Red Hat Enterprise Linux 8

System Design Guide

Designing a RHEL 8 system
Designing a RHEL 8 system
Abstract

This content covers how to start using Red Hat Enterprise Linux 8. To learn about Red Hat Enterprise Linux technology capabilities and limits, see https://access.redhat.com/articles/rhel-limits.
# Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE ......................................................... 24

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION ................................ 25

PART I. DESIGN OF INSTALLATION ............................................................... 26

CHAPTER 1. SUPPORTED RHEL ARCHITECTURES AND SYSTEM REQUIREMENTS ........ 27
   1.1. SUPPORTED ARCHITECTURES ................................................................. 27
   1.2. SYSTEM REQUIREMENTS ................................................................. 27

CHAPTER 2. PREPARING FOR YOUR INSTALLATION ........................................... 28
   2.1. RECOMMENDED STEPS ................................................................. 28
   2.2. RHEL INSTALLATION METHODS ..................................................... 28
   2.3. SYSTEM REQUIREMENTS ................................................................. 29
   2.4. INSTALLATION BOOT MEDIA OPTIONS .......................................... 29
   2.5. TYPES OF INSTALLATION ISO IMAGES ........................................... 30
   2.6. DOWNLOADING A RHEL INSTALLATION ISO IMAGE ............................. 31
      2.6.1. Types of installation ISO images .................................................. 31
      2.6.2. Downloading an ISO image from the Customer Portal ..................... 32
      2.6.3. Downloading an ISO image using curl ........................................ 33
   2.7. CREATING A BOOTABLE INSTALLATION MEDIUM FOR RHEL ............... 34
      2.7.1. Installation boot media options ................................................... 34
      2.7.2. Creating a bootable DVD or CD ................................................... 35
      2.7.3. Creating a bootable USB device on Linux ..................................... 35
      2.7.4. Creating a bootable USB device on Windows ................................ 37
      2.7.5. Creating a bootable USB device on Mac OS X ............................... 38
   2.8. PREPARING AN INSTALLATION SOURCE ........................................... 39
      2.8.1. Types of installation source ....................................................... 39
      2.8.2. Specify the installation source ..................................................... 40
      2.8.3. Ports for network-based installation ......................................... 40
      2.8.4. Creating an installation source on an NFS server ........................... 41
      2.8.5. Creating an installation source using HTTP or HTTPS ..................... 42
      2.8.6. Creating an installation source using FTP .................................... 44
      2.8.7. Preparing a hard drive as an installation source ............................ 46

CHAPTER 3. GETTING STARTED ........................................................................ 48
   3.1. BOOTING THE INSTALLATION ............................................................ 48
      3.1.1. Boot menu ............................................................................. 48
      3.1.2. Types of boot options ............................................................... 49
      3.1.3. Editing the boot: prompt in BIOS .............................................. 50
      3.1.4. Editing the > prompt ................................................................. 50
      3.1.5. Editing the GRUB2 menu ............................................................ 51
      3.1.6. Booting the installation from a USB, CD, or DVD ......................... 51
      3.1.7. Booting the installation from a network using PXE ......................... 52
   3.2. INSTALLING RHEL USING AN ISO IMAGE FROM THE CUSTOMER PORTAL .... 54
   3.3. REGISTERING AND INSTALLING RHEL FROM THE CDN USING THE GUI ...... 55
      3.3.1. What is the Content Delivery Network ....................................... 55
      3.3.2. Registering and installing RHEL from the CDN ............................... 56
      3.3.2.1. Installation source repository after system registration ............... 58
      3.3.3. Verifying your system registration from the CDN ......................... 59
      3.3.4. Unregistering your system from the CDN ..................................... 60
   3.4. COMPLETING THE INSTALLATION ..................................................... 61
CHAPTER 4. CUSTOMIZING YOUR INSTALLATION .................................................. 62
4.1. CONFIGURING LANGUAGE AND LOCATION SETTINGS 62
4.2. CONFIGURING LOCALIZATION OPTIONS 63

4.3. CONFIGURING SYSTEM OPTIONS 65
  4.3.1. Configuring installation destination 65
  4.3.2. Configuring boot loader 69
  4.3.3. Configuring Kdump 70
  4.3.4. Configuring network and host name options 70
    4.3.4.1. Adding a virtual network interface 72
    4.3.4.2. Editing network interface configuration 72
    4.3.4.3. Enabling or Disabling the Interface Connection 73
    4.3.4.4. Setting up Static IPv4 or IPv6 Settings 73
  4.3.5. Configuring Routes 74
  4.3.4.6. Additional resources 74

4.3.5. Configuring Connect to Red Hat 75
  4.3.5.1. Introduction to System Purpose 75
  4.3.5.2. Configuring Connect to Red Hat options 76
  4.3.5.3. Installation source repository after system registration 77
  4.3.5.4. Verifying your system registration from the CDN 78
  4.3.5.5. Unregistering your system from the CDN 79
  4.3.5.6. Additional resources 80

4.3.6. Installing System Aligned with a Security Policy 80
  4.3.6.1. About security policy 80
  4.3.6.2. Configuring a security policy 80
  4.3.6.3. Additional resources 81

4.4. CONFIGURING SOFTWARE SETTINGS 81
  4.4.1. Configuring installation source 81
  4.4.2. Configuring software selection 84

4.5. CONFIGURING STORAGE DEVICES 85
  4.5.1. Storage device selection 85
  4.5.2. Filtering storage devices 86
  4.5.3. Using advanced storage options 86
    4.5.3.1. Discovering and starting an iSCSI session 87
    4.5.3.2. Configuring FCoE parameters 88
    4.5.3.3. Configuring DASD storage devices 89
    4.5.3.4. Configuring FCP devices 90
  4.5.4. Installing to an NVDIMM device 91
    4.5.4.1. Criteria for using an NVDIMM device as an installation target 91
    4.5.4.2. Configuring an NVDIMM device using the graphical installation mode 91

4.6. CONFIGURING MANUAL PARTITIONING 92
  4.6.1. Starting manual partitioning 93
  4.6.2. Adding a mount point file system 94
  4.6.3. Configuring storage for a mount point file system 95
  4.6.4. Customizing a mount point file system 96
  4.6.5. Preserving the /home directory 98
  4.6.6. Creating software RAID 99
  4.6.7. Creating an LVM logical volume 100
  4.6.8. Configuring an LVM logical volume 101

4.7. CONFIGURING A ROOT PASSWORD 102

4.8. CREATING A USER ACCOUNT 103

4.9. EDITING ADVANCED USER SETTINGS 104

CHAPTER 5. COMPLETING POST-INSTALLATION TASKS .............................................. 106
7.4. CREATING SYSTEM IMAGES WITH IMAGE BUILDER WEB CONSOLE INTERFACE
  7.4.1. Accessing Image Builder GUI in the RHEL web console
  7.4.2. Creating an Image Builder blueprint in the web console interface
  7.4.3. Editing an Image Builder blueprint in the web console interface
  7.4.4. Adding users and groups to an Image Builder blueprint in the web console interface
  7.4.5. Creating a system image with Image Builder in the web console interface
  7.4.6. Adding a source to a blueprint
  7.4.7. Creating a user account for a blueprint
  7.4.8. Creating a user account with SSH key

7.5. PREPARING AND UPLOADING CLOUD IMAGES WITH IMAGE BUILDER
  7.5.1. Preparing for uploading AWS AMI images
  7.5.2. Uploading an AMI image to AWS in the CLI
  7.5.3. Pushing images to AWS Cloud AMI
  7.5.4. Preparing for uploading Azure VHD images
  7.5.5. Uploading VHD images to Azure
  7.5.6. Uploading VMDK images to vSphere
  7.5.7. Pushing VMWare images to vSphere
  7.5.8. Pushing VHD images to Azure cloud
  7.5.9. Uploading QCOW2 image to OpenStack
  7.5.10. Preparing for uploading images to Alibaba
  7.5.11. Uploading images to Alibaba
  7.5.12. Importing images to Alibaba
  7.5.13. Creating an instance of a custom image using Alibaba

CHAPTER 8. PERFORMING AN AUTOMATED INSTALLATION USING KICKSTART
  8.1. KICKSTART INSTALLATION BASICS
    8.1.1. What are Kickstart installations
    8.1.2. Automated installation workflow
  8.2. CREATING KICKSTART FILES
    8.2.1. Creating a Kickstart file with the Kickstart configuration tool
    8.2.2. Creating a Kickstart file by performing a manual installation
    8.2.3. Converting a RHEL 7 Kickstart file for RHEL 8 installation
    8.2.4. Creating a custom image using Image Builder
  8.3. MAKING KICKSTART FILES AVAILABLE TO THE INSTALLATION PROGRAM
    8.3.1. Ports for network-based installation
    8.3.2. Making a Kickstart file available on an NFS server
    8.3.3. Making a Kickstart file available on an HTTP or HTTPS server
    8.3.4. Making a Kickstart file available on an FTP server
    8.3.5. Making a Kickstart file available on a local volume
    8.3.6. Making a Kickstart file available on a local volume for automatic loading
  8.4. CREATING INSTALLATION SOURCES FOR KICKSTART INSTALLATIONS
    8.4.1. Types of installation source
    8.4.2. Ports for network-based installation
    8.4.3. Creating an installation source on an NFS server
    8.4.4. Creating an installation source using HTTP or HTTPS
    8.4.5. Creating an installation source using FTP
  8.5. STARTING KICKSTART INSTALLATIONS
    8.5.1. Starting a Kickstart installation manually
    8.5.2. Starting a Kickstart installation automatically using PXE
    8.5.3. Starting a Kickstart installation automatically using a local volume
  8.6. CONSOLES AND LOGGING DURING INSTALLATION
  8.7. MAINTAINING KICKSTART FILES
    8.7.1. Installing Kickstart maintenance tools
8.7.2. Verifying a Kickstart file
8.8. REGISTERING AND INSTALLING RHEL FROM THE CDN USING KICKSTART
8.8.1. Registering and installing RHEL from the CDN
8.8.2. Verifying your system registration from the CDN
8.8.3. Unregistering your system from the CDN
8.9. PERFORMING A REMOTE RHEL INSTALLATION USING VNC
8.9.1. Overview
8.9.2. Considerations
8.9.3. Performing a remote RHEL installation in VNC Direct mode
8.9.4. Performing a remote RHEL installation in VNC Connect mode

CHAPTER 9. ADVANCED CONFIGURATION OPTIONS .............................................. 248
9.1. CONFIGURING SYSTEM PURPOSE
  9.1.1. Overview
  9.1.2. Configuring System Purpose in a Kickstart file
  9.1.3. Additional resources
9.2. UPDATING DRIVERS DURING INSTALLATION
  9.2.1. Prerequisite
  9.2.2. Overview
  9.2.3. Types of driver update
  9.2.4. Preparing a driver update
  9.2.5. Performing an automatic driver update
  9.2.6. Performing an assisted driver update
  9.2.7. Performing a manual driver update
  9.2.8. Disabling a driver
9.3. PREPARING TO INSTALL FROM THE NETWORK USING PXE
  9.3.1. Network install overview
  9.3.2. Configuring a TFTP server for BIOS-based clients
  9.3.3. Configuring a TFTP server for UEFI-based clients
  9.3.4. Configuring a network server for IBM Power systems
9.4. BOOT OPTIONS
  9.4.1. Types of boot options
  9.4.2. Editing boot options
    9.4.2.1. Editing the boot: prompt in BIOS
    9.4.2.2. Editing the > prompt
    9.4.2.3. Editing the GRUB2 menu
  9.4.3. Installation source boot options
  9.4.4. Network boot options
  9.4.5. Console boot options
  9.4.6. Debug boot options
  9.4.7. Storage boot options
  9.4.8. Kickstart boot options
  9.4.9. Advanced installation boot options
  9.4.10. Deprecated boot options
  9.4.11. Removed boot options

CHAPTER 10. KICKSTART REFERENCES ................................................................. 282
APPENDIX I. KICKSTART SCRIPT FILE FORMAT REFERENCE ............................ 283
I.1. KICKSTART FILE FORMAT
I.2. PACKAGE SELECTION IN KICKSTART
  I.2.1. Package selection section
  I.2.2. Package selection commands
  I.2.3. Common package selection options
I.2.4. Options for specific package groups

I.3. SCRIPTS IN KICKSTART FILE
  I.3.1. %pre script
    I.3.1.1. %pre script section options
  I.3.2. %pre-install script
    I.3.2.1. %pre-install script section options
  I.3.3. %post script
    I.3.3.1. %post script section options
    I.3.3.2. Example: Mounting NFS in a post-install script
    I.3.3.3. Example: Running subscription-manager as a post-install script

I.4. ANACONDA CONFIGURATION SECTION

I.5. KICKSTART ERROR HANDLING SECTION

I.6. KICKSTART ADD-ON SECTIONS

APPENDIX J. KICKSTART COMMANDS AND OPTIONS REFERENCE

J.1. KICKSTART CHANGES
  J.1.1. Deprecated Kickstart commands and options
  J.1.2. Removed Kickstart commands and options

J.2. KICKSTART COMMANDS FOR INSTALLATION PROGRAM CONFIGURATION AND FLOW CONTROL
  J.2.1. cdrom
  J.2.2. cmdline
  J.2.3. driverdisk
  J.2.4. eula
  J.2.5. firstboot
  J.2.6. graphical
  J.2.7. halt
  J.2.8. harddrive
  J.2.9. install (deprecated)
  J.2.10. liveimg
  J.2.11. logging
  J.2.12. mediacheck
  J.2.13. nfs
  J.2.14. ostreesetup
  J.2.15. poweroff
  J.2.16. reboot
  J.2.17. rhsm
  J.2.18. shutdown
  J.2.19. sshpw
  J.2.20. text
  J.2.21. url
  J.2.22. vnc
  J.2.23. %include
  J.2.24. %ksappend

J.3. KICKSTART COMMANDS FOR SYSTEM CONFIGURATION
  J.3.1. auth or authconfig (deprecated)
  J.3.2. authselect
  J.3.3. firewall
  J.3.4. group
  J.3.5. keyboard (required)
  J.3.6. lang (required)
  J.3.7. module
  J.3.8. repo
J.3.9. rootpw (required)
J.3.10. selinux
J.3.11. services
J.3.12. skipx
J.3.13. sshkey
J.3.14. syspurpose
J.3.15. timezone (required)
J.3.16. user
J.3.17. xconfig

J.4. KICKSTART COMMANDS FOR NETWORK CONFIGURATION
J.4.1. network (optional)
J.4.2. realm

J.5. KICKSTART COMMANDS FOR HANDLING STORAGE
J.5.1. device (deprecated)
J.5.2. autopart
J.5.3. bootloader (required)
J.5.4. zipl
J.5.5. clearpart
J.5.6. fcoe
J.5.7. ignoredisk
J.5.8. iscsi
J.5.9. iscsiname
J.5.10. logvol
J.5.11. mount
J.5.12. nvdimm
J.5.13. part or partition
J.5.14. raid
J.5.15. requart
J.5.16. snapshot
J.5.17. volgroup
J.5.18. zerombr
J.5.19. zfcp

J.6. KICKSTART COMMANDS FOR ADDONS SUPPLIED WITH THE RHEL INSTALLATION PROGRAM
J.6.1. %addon com_redhat_kdump
J.6.2. %addon org_fedora_oscap

J.7. COMMANDS USED IN ANACONDA
J.7.1. pwpolicy

J.8. KICKSTART COMMANDS FOR SYSTEM RECOVERY
J.8.1. rescue

PART II. DESIGN OF SECURITY

CHAPTER 11. OVERVIEW OF SECURITY HARDENING IN RHEL

11.1. WHAT IS COMPUTER SECURITY?
11.2. STANDARDIZING SECURITY
11.3. CRYPTOGRAPHIC SOFTWARE AND CERTIFICATIONS
11.4. SECURITY CONTROLS
11.4.1. Physical controls
11.4.2. Technical controls
11.4.3. Administrative controls
11.5. VULNERABILITY ASSESSMENT
11.5.1. Defining assessment and testing
11.5.2. Establishing a methodology for vulnerability assessment

358
11.5.3. Vulnerability assessment tools
11.6. SECURITY THREATS
11.6.1. Threats to network security
11.6.2. Threats to server security
11.6.3. Threats to workstation and home PC security
11.7. COMMON EXPLOITS AND ATTACKS

CHAPTER 12. SECURING RHEL DURING INSTALLATION ........................................ 370
12.1. BIOS AND UEFI SECURITY
  12.1.1. BIOS passwords
  12.1.2. Non-BIOS-based systems security
12.2. DISK PARTITIONING
12.3. RESTRICTING NETWORK CONNECTIVITY DURING THE INSTALLATION PROCESS
12.4. INSTALLING THE MINIMUM AMOUNT OF PACKAGES REQUIRED
12.5. POST-INSTALLATION PROCEDURES

CHAPTER 13. USING SYSTEM-WIDE CRYPTOGRAPHIC POLICIES ............................. 373
13.1. SYSTEM-WIDE CRYPTOGRAPHIC POLICIES
  Tool for managing crypto policies
  Strong crypto defaults by removing insecure cipher suites and protocols
  Cipher suites and protocols disabled in all policy levels
  Cipher suites and protocols enabled in the crypto-policies levels
13.2. SWITCHING THE SYSTEM-WIDE CRYPTOGRAPHIC POLICY TO MODE COMPATIBLE WITH EARLIER RELEASES
13.3. SWITCHING THE SYSTEM TO FIPS MODE
13.4. ENABLING FIPS MODE IN A CONTAINER
13.5. LIST OF RHEL APPLICATIONS USING CRYPTOGRAPHY THAT IS NOT COMPLIANT WITH FIPS 140-2
13.6. EXCLUDING AN APPLICATION FROM FOLLOWING SYSTEM-WIDE CRYPTO POLICIES
  13.6.1. Examples of opting out of system-wide crypto policies
13.7. CUSTOMIZING SYSTEM-WIDE CRYPTOGRAPHIC POLICIES WITH SUBPOLICIES
13.8. DISABLING SHA-1 BY CUSTOMIZING A SYSTEM-WIDE CRYPTOGRAPHIC POLICY
13.9. CREATING AND SETTING A CUSTOM SYSTEM-WIDE CRYPTOGRAPHIC POLICY
13.10. ADDITIONAL RESOURCES

CHAPTER 14. CONFIGURING APPLICATIONS TO USE CRYPTOGRAPHIC HARDWARE THROUGH PKCS #11 . 385
14.1. CRYPTOGRAPHIC HARDWARE SUPPORT THROUGH PKCS #11
14.2. USING SSH KEYS STORED ON A SMART CARD
14.3. CONFIGURING APPLICATIONS TO AUTHENTICATE USING CERTIFICATES FROM SMART CARDS
14.4. USING HSMS PROTECTING PRIVATE KEYS IN APACHE
14.5. USING HSMS PROTECTING PRIVATE KEYS IN NGINX
14.6. ADDITIONAL RESOURCES

CHAPTER 15. USING SHARED SYSTEM CERTIFICATES ........................................... 389
15.1. THE SYSTEM-WIDE TRUST STORE
15.2. ADDING NEW CERTIFICATES
15.3. MANAGING TRUSTED SYSTEM CERTIFICATES
15.4. ADDITIONAL RESOURCES

CHAPTER 16. SCANNING THE SYSTEM FOR SECURITY COMPLIANCE AND VULNERABILITIES .......... 392
16.1. CONFIGURATION COMPLIANCE TOOLS IN RHEL
16.2. RED HAT SECURITY ADVISORIES OVAL FEED
16.3. VULNERABILITY SCANNING
  16.3.1. Red Hat Security Advisories OVAL feed
26.5.9.3. Removing a redirected port
26.5.9.4. Removing TCP port 80 forwarded to port 88 on the same machine
26.5.10. Managing ICMP requests
  26.5.10.1. Listing and blocking ICMP requests
  26.5.10.2. Configuring the ICMP filter using GUI
26.5.11. Setting and controlling IP sets using firewalld
  26.5.11.1. Configuring IP set options using CLI
26.5.12. Prioritizing rich rules
  26.5.12.1. How the priority parameter organizes rules into different chains
  26.5.12.2. Setting the priority of a rich rule
26.5.13. Configuring firewall lockdown
  26.5.13.1. Configuring lockdown using CLI
  26.5.13.2. Configuring lockdown allowlist options using CLI
  26.5.13.3. Configuring lockdown allowlist options using configuration files
26.5.14. Enabling traffic forwarding between different interfaces or sources within a firewalld zone
  26.5.14.1. The difference between intra-zone forwarding and zones with the default target set to ACCEPT
  26.5.14.2. Using intra-zone forwarding to forward traffic between an Ethernet and Wi-Fi network
26.5.15. Additional resources
26.6. GETTING STARTED WITH NFTABLES
  26.6.1. Migrating from iptables to nftables
    26.6.1.1. When to use firewalld, nftables, or iptables
    26.6.1.2. Converting iptables rules to nftables rules
    26.6.1.3. Comparison of common iptables and nftables commands
  26.6.2. Writing and executing nftables scripts
    26.6.2.1. Supported nftables script formats
    26.6.2.2. Running nftables scripts
    26.6.2.3. Using comments in nftables scripts
    26.6.2.4. Using variables in an nftables script
    26.6.2.5. Including files in an nftables script
    26.6.2.6. Automatically loading nftables rules when the system boots
  26.6.3. Creating and managing nftables tables, chains, and rules
    26.6.3.1. Standard chain priority values and textual names
    26.6.3.2. Displaying the nftables rule set
    26.6.3.3. Creating an nftables table
    26.6.3.4. Creating an nftables chain
    26.6.3.5. Appending a rule to the end of an nftables chain
    26.6.3.6. Inserting a rule at the beginning of an nftables chain
    26.6.3.7. Inserting a rule at a specific position of an nftables chain
  26.6.4. Configuring NAT using nftables
    26.6.4.1. The different NAT types: masquerading, source NAT, destination NAT, and redirect
    26.6.4.2. Configuring masquerading using nftables
    26.6.4.3. Configuring source NAT using nftables
    26.6.4.4. Configuring destination NAT using nftables
    26.6.4.5. Configuring a redirect using nftables
  26.6.5. Using sets in nftables commands
    26.6.5.1. Using anonymous sets in nftables
    26.6.5.2. Using named sets in nftables
    26.6.5.3. Additional resources
  26.6.6. Using verdict maps in nftables commands
    26.6.6.1. Using anonymous maps in nftables
    26.6.6.2. Using named maps in nftables
    26.6.6.3. Additional resources
26.6.7. Configuring port forwarding using nftables
  26.6.7.1. Forwarding incoming packets to a different local port
  26.6.7.2. Forwarding incoming packets on a specific local port to a different host
  26.6.8. Using nftables to limit the amount of connections
  26.6.8.1. Limiting the number of connections using nftables
  26.6.8.2. Blocking IP addresses that attempt more than ten new incoming TCP connections within one minute

26.6.9. Debugging nftables rules
  26.6.9.1. Creating a rule with a counter
  26.6.9.2. Adding a counter to an existing rule
  26.6.9.3. Monitoring packets that match an existing rule
  26.6.10. Backing up and restoring the nftables rule set
    26.6.10.1. Backing up the nftables rule set to a file
    26.6.10.2. Restoring the nftables rule set from a file
  26.6.11. Additional resources

PART IV. DESIGN OF HARD DISK .............................................................. 584

CHAPTER 27. OVERVIEW OF AVAILABLE FILE SYSTEMS ........................................ 585
  27.1. TYPES OF FILE SYSTEMS
  27.2. LOCAL FILE SYSTEMS
  27.3. THE XFS FILE SYSTEM
  27.4. THE EXT4 FILE SYSTEM
  27.5. COMPARISON OF XFS AND EXT4
  27.6. CHOOSING A LOCAL FILE SYSTEM
  27.7. NETWORK FILE SYSTEMS
  27.8. SHARED STORAGE FILE SYSTEMS
  27.9. CHOOSING BETWEEN NETWORK AND SHARED STORAGE FILE SYSTEMS
  27.10. VOLUME-MANAGING FILE SYSTEMS

CHAPTER 28. MOUNTING NFS SHARES ....................................................... 593
  28.1. INTRODUCTION TO NFS
  28.2. SUPPORTED NFS VERSIONS
    Default NFS version
    Features of minor NFS versions
  28.3. SERVICES REQUIRED BY NFS
    The RPC services with NFSv4
  28.4. NFS HOST NAME FORMATS
  28.5. INSTALLING NFS
  28.6. DISCOVERING NFS EXPORTS
  28.7. MOUNTING AN NFS SHARE WITH MOUNT
  28.8. COMMON NFS MOUNT OPTIONS
  28.9. ADDITIONAL RESOURCES

CHAPTER 29. EXPORTING NFS SHARES ...................................................... 599
  29.1. INTRODUCTION TO NFS
  29.2. SUPPORTED NFS VERSIONS
    Default NFS version
    Features of minor NFS versions
  29.3. THE TCP AND UDP PROTOCOLS IN NFSV3 AND NFSV4
  29.4. SERVICES REQUIRED BY NFS
    The RPC services with NFSv4
  29.5. NFS HOST NAME FORMATS
  29.6. NFS SERVER CONFIGURATION
32.2.5. MBR partition types
32.2.6. GUID Partition Table
32.2.7. Creating a partition table on a disk with parted

32.3. CREATING A PARTITION

32.3.1. Considerations before modifying partitions on a disk
The maximum number of partitions
The maximum size of a partition
Size alignment

32.3.2. Partition types
Partition types or flags
Partition file system type

32.3.3. Partition naming scheme

32.3.4. Mount points and disk partitions

32.3.5. Creating a partition with parted

32.3.6. Setting a partition type with fdisk

32.3.7. Creating a partition with parted

32.4. REMOVING A PARTITION

32.4.1. Considerations before modifying partitions on a disk
The maximum number of partitions
The maximum size of a partition
Size alignment

32.4.2. Removing a partition with parted

32.5. RESIZING A PARTITION

32.5.1. Considerations before modifying partitions on a disk
The maximum number of partitions
The maximum size of a partition
Size alignment

32.5.2. Resizing a partition with parted

32.6. STRATEGIES FOR REPARTITIONING A DISK

32.6.1. Using unpartitioned free space

32.6.2. Using space from an unused partition

32.6.3. Using free space from an active partition

32.6.3.1. Destructive repartitioning

32.6.3.2. Non-destructive repartitioning

32.6.3.2.1. Compressing existing data

32.6.3.2.2. Resizing the existing partition

32.6.3.2.3. Creating new partitions

CHAPTER 33. GETTING STARTED WITH XFS .................................................. 646

33.1. THE XFS FILE SYSTEM

33.2. COMPARISON OF TOOLS USED WITH EXT4 AND XFS

CHAPTER 34. MOUNTING FILE SYSTEMS ................................................. 648

34.1. THE LINUX MOUNT MECHANISM

34.2. LISTING CURRENTLY MOUNTED FILE SYSTEMS

34.3. MOUNTING A FILE SYSTEM WITH MOUNT

34.4. MOVING A MOUNT POINT

34.5. UNMOUNTING A FILE SYSTEM WITH UMTOUNT

34.6. COMMON MOUNT OPTIONS

CHAPTER 35. SHARING A MOUNT ON MULTIPLE MOUNT POINTS ............... 653

35.1. TYPES OF SHARED MOUNTS

35.2. CREATING A PRIVATE MOUNT POINT DUPLICATE

35.3. CREATING A SHARED MOUNT POINT DUPLICATE

35.4. CREATING A SLAVE MOUNT POINT DUPLICATE
35.5. PREVENTING A MOUNT POINT FROM BEING DUPLICATED 657

CHAPTER 36. PERSISTENTLY MOUNTING FILE SYSTEMS ............................... 658
36.1. THE /ETC/FSTAB FILE 658
36.2. ADDING A FILE SYSTEM TO /ETC/FSTAB 658

CHAPTER 37. PERSISTENTLY MOUNTING A FILE SYSTEM USING RHEL SYSTEM ROLES ............................. 660
37.1. EXAMPLE ANSIBLE PLAYBOOK TO PERSISTENTLY MOUNT A FILE SYSTEM 660

CHAPTER 38. MOUNTING FILE SYSTEMS ON DEMAND .................................. 661
38.1. THE AUTOFS SERVICE 661
38.2. THE AUTOFS CONFIGURATION FILES 661
38.3. CONFIGURING AUTOFS MOUNT POINTS 663
38.4. AUTOMOUNTING NFS SERVER USER HOME DIRECTORIES WITH AUTOFS SERVICE 664
38.5. OVERRIDING OR AUGMENTING AUTOFS SITE CONFIGURATION FILES 664
38.6. USING LDAP TO STORE AUTOMOUNTER MAPS 666
38.7. USING SYSTEMD.AUTOMOUNT TO MOUNT A FILE SYSTEM ON DEMAND WITH /ETC/FSTAB 667
38.8. USING SYSTEMD.AUTOMOUNT TO MOUNT A FILE SYSTEM ON DEMAND WITH A MOUNT UNIT 668

CHAPTER 39. USING SSSD COMPONENT FROM IDM TO CACHE THE AUTOFS MAPS .................................. 670
39.1. CONFIGURING AUTOFS MANUALLY TO USE IDM SERVER AS AN LDAP SERVER 670
39.2. CONFIGURING SSSD TO CACHE AUTOFS MAPS 671

CHAPTER 40. SETTING READ-ONLY PERMISSIONS FOR THE ROOT FILE SYSTEM ................................. 673
40.1. FILES AND DIRECTORIES THAT ALWAYS RETAIN WRITE PERMISSIONS 673
40.2. CONFIGURING THE ROOT FILE SYSTEM TO MOUNT WITH READ-ONLY PERMISSIONS ON BOOT 674

CHAPTER 41. MANAGING STORAGE DEVICES ...................................................... 675
41.1. SETTING UP STRATIS FILE SYSTEMS 675
41.1.1. What is Stratis 675
41.1.2. Components of a Stratis volume 675
41.1.3. Block devices usable with Stratis 676
    Supported devices 676
    Unsupported devices 677
41.1.4. Installing Stratis 677
41.1.5. Creating an unencrypted Stratis pool 677
41.1.6. Creating an encrypted Stratis pool 678
41.1.7. Binding a Stratis pool to NBDE 679
41.1.8. Binding a Stratis pool to TPM 680
41.1.9. Unlocking an encrypted Stratis pool with kernel keyring 681
41.1.10. Unlocking an encrypted Stratis pool with Clevis 681
41.1.11. Unbinding a Stratis pool from supplementary encryption 682
41.1.12. Creating a Stratis file system 682
41.1.13. Mounting a Stratis file system 683
41.1.14. Persistently mounting a Stratis file system 683
41.2. EXTENDING A STRATIS VOLUME WITH ADDITIONAL BLOCK DEVICES 685
41.2.1. Components of a Stratis volume 685
41.2.2. Adding block devices to a Stratis pool 686
41.2.3. Additional resources 686
41.3. MONITORING STRATIS FILE SYSTEMS 686
41.3.1. Stratis sizes reported by different utilities 686
41.3.2. Displaying information about Stratis volumes 687
41.3.3. Additional resources 687
41.4. USING SNAPSHOTS ON STRATIS FILE SYSTEMS 688
41.4.1. Characteristics of Stratis snapshots 688
41.4.2. Creating a Stratis snapshot
41.4.3. Accessing the content of a Stratis snapshot
41.4.4. Reverting a Stratis file system to a previous snapshot
41.4.5. Removing a Stratis snapshot
41.4.6. Additional resources

41.5. REMOVING STRATIS FILE SYSTEMS
41.5.1. Components of a Stratis volume
41.5.2. Removing a Stratis file system
41.5.3. Removing a Stratis pool
41.5.4. Additional resources

41.6. GETTING STARTED WITH SWAP
41.6.1. Overview of swap space
41.6.2. Recommended system swap space
41.6.3. Extending swap on an LVM2 logical volume
41.6.4. Creating an LVM2 logical volume for swap
41.6.5. Creating a swap file
41.6.6. Reducing swap on an LVM2 logical volume
41.6.7. Removing an LVM2 logical volume for swap
41.6.8. Removing a swap file

CHAPTER 42. DEDUPLICATING AND COMPRESSING STORAGE

42.1. DEPLOYING VDO
42.1.1. Introduction to VDO
42.1.2. VDO deployment scenarios
   KVM
   File systems
   Placement of VDO on iSCSI
   LVM
   Encryption
42.1.3. Components of a VDO volume
42.1.4. The physical and logical size of a VDO volume
42.1.5. Slab size in VDO
42.1.6. VDO requirements
   42.1.6.1. VDO memory requirements
   42.1.6.2. VDO storage space requirements
   42.1.6.3. Placement of VDO in the storage stack
   42.1.6.4. Examples of VDO requirements by physical size
42.1.7. Installing VDO
42.1.8. Creating a VDO volume
42.1.9. Mounting a VDO volume
42.1.10. Enabling periodic block discard
42.1.11. Monitoring VDO

42.2. MAINTAINING VDO
42.2.1. Managing free space on VDO volumes
   42.2.1.1. The physical and logical size of a VDO volume
   42.2.1.2. Thin provisioning in VDO
   42.2.1.3. Monitoring VDO
   42.2.1.4. Reclaiming space for VDO on file systems
   42.2.1.5. Reclaiming space for VDO without a file system
   42.2.1.6. Reclaiming space for VDO on Fibre Channel or Ethernet network
42.2.2. Starting or stopping VDO volumes
   42.2.2.1. Started and activated VDO volumes
   42.2.2.2. Starting a VDO volume
42.2.2.3. Stopping a VDO volume
42.2.2.4. Additional resources
42.2.3. Automatically starting VDO volumes at system boot
  42.2.3.1. Started and activated VDO volumes
  42.2.3.2. Activating a VDO volume
  42.2.3.3. Deactivating a VDO volume
42.2.4. Selecting a VDO write mode
  42.2.4.1. VDO write modes
  42.2.4.2. The internal processing of VDO write modes
  42.2.4.3. Checking the write mode on a VDO volume
  42.2.4.4. Checking for a volatile cache
  42.2.4.5. Setting a VDO write mode
42.2.5. Recovering a VDO volume after an unclean shutdown
  42.2.5.1. VDO write modes
  42.2.5.2. VDO volume recovery
    Automatic and manual recovery
  42.2.5.3. VDO operating modes
  42.2.5.4. Recovering a VDO volume online
  42.2.5.5. Forcing an offline rebuild of a VDO volume metadata
  42.2.5.6. Removing an unsuccessfully created VDO volume
42.2.6. Optimizing the UDS index
  42.2.6.1. Components of a VDO volume
  42.2.6.2. The UDS index
  42.2.6.3. Recommended UDS index configuration
42.2.7. Enabling or disabling deduplication in VDO
  42.2.7.1. Deduplication in VDO
  42.2.7.2. Enabling deduplication on a VDO volume
  42.2.7.3. Disabling deduplication on a VDO volume
42.2.8. Enabling or disabling compression in VDO
  42.2.8.1. Compression in VDO
  42.2.8.2. Enabling compression on a VDO volume
  42.2.8.3. Disabling compression on a VDO volume
42.2.9. Increasing the size of a VDO volume
  42.2.9.1. The physical and logical size of a VDO volume
  42.2.9.2. Thin provisioning in VDO
  42.2.9.3. Increasing the logical size of a VDO volume
  42.2.9.4. Increasing the physical size of a VDO volume
42.2.10. Removing VDO volumes
  42.2.10.1. Removing a working VDO volume
  42.2.10.2. Removing an unsuccessfully created VDO volume
42.2.11. Additional resources
42.3. DISCARDING UNUSED BLOCKS
  42.3.1. Block discard operations
    Requirements
  42.3.2. Types of block discard operations
    Recommendations
  42.3.3. Performing batch block discard
  42.3.4. Enabling online block discard
  42.3.5. Enabling periodic block discard
42.4. USING THE WEB CONSOLE FOR MANAGING VIRTUAL DATA OPTIMIZER VOLUMES
  42.4.1. VDO volumes in the web console
  42.4.2. Creating VDO volumes in the web console
  42.4.3. Formatting VDO volumes in the web console
### PART V. DESIGN OF LOG FILE

#### CHAPTER 43. AUDITING THE SYSTEM

- 43.1. LINUX AUDIT
- 43.2. AUDIT SYSTEM ARCHITECTURE
- 43.3. CONFIGURING AUDITD FOR A SECURE ENVIRONMENT
- 43.4. STARTING AND CONTROLLING AUDITD
- 43.5. UNDERSTANDING AUDIT LOG FILES
- 43.6. USING AUDITCTL FOR DEFINING AND EXECUTING AUDIT RULES
- 43.7. DEFINING PERSISTENT AUDIT RULES
- 43.8. USING PRE-CONFIGURED RULES FILES
- 43.9. USING AUGENRULES TO DEFINE PERSISTENT RULES
- 43.10. DISABLING AUGENRULES
- 43.11. SETTING UP AUDIT TO MONITOR SOFTWARE UPDATES
- 43.12. MONITORING USER LOGIN TIMES WITH AUDIT
- 43.13. ADDITIONAL RESOURCES

#### PART VI. DESIGN OF KERNEL

#### CHAPTER 44. THE LINUX KERNEL RPM

- 44.1. WHAT AN RPM IS
  - Types of RPM packages
- 44.2. THE LINUX KERNEL RPM PACKAGE OVERVIEW
- 44.3. DISPLAYING CONTENTS OF THE KERNEL PACKAGE

#### CHAPTER 45. UPDATING KERNEL WITH YUM

- 45.1. WHAT IS THE KERNEL
- 45.2. WHAT IS YUM
- 45.3. UPDATING THE KERNEL
- 45.4. INSTALLING THE KERNEL

#### CHAPTER 46. CONFIGURING KERNEL COMMAND-LINE PARAMETERS

- 46.1. UNDERSTANDING KERNEL COMMAND-LINE PARAMETERS
- 46.2. WHAT GRUBBY IS
- 46.3. WHAT BOOT ENTRIES ARE
- 46.4. CHANGING KERNEL COMMAND-LINE PARAMETERS FOR ALL BOOT ENTRIES
- 46.5. CHANGING KERNEL COMMAND-LINE PARAMETERS FOR A SINGLE BOOT ENTRY
- 46.6. CHANGING KERNEL COMMAND-LINE PARAMETERS TEMPORARILY AT BOOT TIME

#### CHAPTER 47. CONFIGURING KERNEL PARAMETERS AT RUNTIME

- 47.1. WHAT ARE KERNEL PARAMETERS
- 47.2. CONFIGURING KERNEL PARAMETERS TEMPORARILY WITH SYSCTL
- 47.3. CONFIGURING KERNEL PARAMETERS PERMANENTLY WITH SYSCTL
- 47.4. USING CONFIGURATION FILES IN /ETC/SYSCTL.D/ TO ADJUST KERNEL PARAMETERS
- 47.5. CONFIGURING KERNEL PARAMETERS TEMPORARILY THROUGH /PROC/SYS/

#### CHAPTER 48. INSTALLING AND CONFIGURING KDUMP

- 48.1. INSTALLING KDUMP
  - 48.1.1. What is kdump
  - 48.1.2. Installing kdump using Anaconda
  - 48.1.3. Installing kdump on the command line
- 48.2. CONFIGURING KDUMP ON THE COMMAND LINE
  - 48.2.1. Estimating the kdump size
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>48.2.2. Configuring kdump memory usage</td>
<td>777</td>
</tr>
<tr>
<td>48.2.3. Configuring the kdump target</td>
<td>779</td>
</tr>
<tr>
<td>48.2.4. Configuring the kdump core collector</td>
<td>781</td>
</tr>
<tr>
<td>48.2.5. Configuring the kdump default failure responses</td>
<td>782</td>
</tr>
<tr>
<td>48.2.6. Testing the kdump configuration</td>
<td>782</td>
</tr>
<tr>
<td>48.3. ENABLING KDUMP</td>
<td>783</td>
</tr>
<tr>
<td>48.4. CONFIGURING KDUMP IN THE WEB CONSOLE</td>
<td>784</td>
</tr>
<tr>
<td>48.4.1. Additional resources</td>
<td>784</td>
</tr>
<tr>
<td>48.4.2. Configuring kdump memory usage and target location in web console</td>
<td>784</td>
</tr>
<tr>
<td>48.5. SUPPORTED KDUMP CONFIGURATIONS AND TARGETS</td>
<td>786</td>
</tr>
<tr>
<td>48.5.1. Memory requirements for kdump</td>
<td>786</td>
</tr>
<tr>
<td>48.5.2. Minimum threshold for automatic memory reservation</td>
<td>787</td>
</tr>
<tr>
<td>48.5.3. Supported kdump targets</td>
<td>788</td>
</tr>
<tr>
<td>48.5.4. Supported kdump filtering levels</td>
<td>789</td>
</tr>
<tr>
<td>48.5.5. Supported default failure responses</td>
<td>789</td>
</tr>
<tr>
<td>48.5.6. Using final_action parameter</td>
<td>790</td>
</tr>
<tr>
<td>48.6. TESTING THE KDUMP CONFIGURATION</td>
<td>790</td>
</tr>
<tr>
<td>48.7. USING KEXEC TO BOOT INTO A DIFFERENT KERNEL</td>
<td>791</td>
</tr>
<tr>
<td>48.8. PREVENTING KERNEL DRIVERS FROM LOADING FOR KDUMP</td>
<td>792</td>
</tr>
<tr>
<td>48.9. RUNNING KDUMP ON SYSTEMS WITH ENCRYPTED DISK</td>
<td>793</td>
</tr>
<tr>
<td>48.10. FIRMWARE ASSISTED DUMP MECHANISMS</td>
<td>794</td>
</tr>
<tr>
<td>48.10.1. Firmware assisted dump on IBM PowerPC hardware</td>
<td>794</td>
</tr>
<tr>
<td>48.10.2. Enabling firmware assisted dump mechanism</td>
<td>794</td>
</tr>
<tr>
<td>48.10.3. Firmware assisted dump mechanisms on IBM Z hardware</td>
<td>795</td>
</tr>
<tr>
<td>48.10.4. Using sadump on Fujitsu PRIMEQUEST systems</td>
<td>796</td>
</tr>
<tr>
<td>48.11. ANALYZING A CORE DUMP</td>
<td>796</td>
</tr>
<tr>
<td>48.11.1. Installing the crash utility</td>
<td>796</td>
</tr>
<tr>
<td>48.11.2. Running and exiting the crash utility</td>
<td>797</td>
</tr>
<tr>
<td>48.11.3. Displaying various indicators in the crash utility</td>
<td>798</td>
</tr>
<tr>
<td>48.11.4. Using Kernel Oops Analyzer</td>
<td>801</td>
</tr>
<tr>
<td>48.11.5. The Kdump Helper tool</td>
<td>802</td>
</tr>
<tr>
<td>48.12. USING EARLY KDUMP TO CAPTURE BOOT TIME CRASHES</td>
<td>802</td>
</tr>
<tr>
<td>48.12.1. What is early kdump</td>
<td>802</td>
</tr>
<tr>
<td>48.12.2. Enabling early kdump</td>
<td>802</td>
</tr>
</tbody>
</table>
MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Please let us know how we could make it better.

- For simple comments on specific passages:
  1. Make sure you are viewing the documentation in the *Multi-page HTML* format. In addition, ensure you see the **Feedback** button in the upper right corner of the document.
  2. Use your mouse cursor to highlight the part of text that you want to comment on.
  3. Click the **Add Feedback** pop-up that appears below the highlighted text.
  4. Follow the displayed instructions.

- For submitting feedback via Bugzilla, create a new ticket:
  1. Go to the [Bugzilla](https://bugzilla.redhat.com) website.
  2. As the Component, use **Documentation**.
  3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
  4. Click **Submit Bug**.
PART I. DESIGN OF INSTALLATION
CHAPTER 1. SUPPORTED RHEL ARCHITECTURES AND SYSTEM REQUIREMENTS

Red Hat Enterprise Linux 8 delivers a stable, secure, consistent foundation across hybrid cloud deployments with the tools needed to deliver workloads faster with less effort. It can be deployed as a guest on supported hypervisors and Cloud provider environments as well as deployed on physical infrastructure, so your applications can take advantage of innovations in the leading hardware architecture platforms.

This section contains information about the supported architectures and the system requirements for installing Red Hat Enterprise Linux.

1.1. SUPPORTED ARCHITECTURES

Red Hat Enterprise Linux supports the following architectures:

- AMD, Intel, and ARM 64-bit architectures
- IBM Power Systems, Little Endian
  - IBM Power System LC servers
  - IBM Power System AC servers
  - IBM Power System L servers
- 64-bit IBM Z

1.2. SYSTEM REQUIREMENTS

If this is a first-time install of Red Hat Enterprise Linux it is recommended that you review the guidelines provided for system, hardware, security, memory, and RAID before installing. See System requirements reference for more information.

If you want to use your system as a virtualization host, review the necessary hardware requirements for virtualization.

Additional resources

- Security hardening
- Composing a customized RHEL system image
CHAPTER 2. PREPARING FOR YOUR INSTALLATION

Before you begin to install Red Hat Enterprise Linux, review the following sections to prepare your setup for the installation.

2.1. RECOMMENDED STEPS

Preparing for your RHEL installation consists of the following steps:

Steps

1. Review and determine the installation method.
2. Check system requirements.
3. Review the installation boot media options.
4. Download the required installation ISO image.
5. Create a bootable installation medium.
6. Prepare the installation source*

*Only required for the Boot ISO (minimal install) image if you are not using the Content Delivery Network (CDN) to download the required software packages.

2.2. RHEL INSTALLATION METHODS

You can install Red Hat Enterprise Linux using any of the following methods:

- GUI-based installations
- System or cloud image-based installations
- Advanced installations

NOTE

This document provides details about installing RHEL using the user interfaces (GUI).

GUI-based installations

The following GUI-based installation methods are available:

- **Install RHEL using an ISO image from the Customer Portal**: Install Red Hat Enterprise Linux by downloading the DVD ISO image file from the Customer Portal. Registration is performed after the GUI installation completes. This installation method is also supported by Kickstart.

- **Register and install RHEL from the Content Delivery Network**: Register your system, attach subscriptions, and install Red Hat Enterprise Linux from the Content Delivery Network (CDN). This installation method is supported by the Boot ISO and DVD ISO image files; however, it is recommended that you use the Boot ISO image file as the installation source defaults to CDN for the Boot ISO image file. Registration is performed before the installation packages are downloaded and installed from the CDN. This installation method is also supported by Kickstart.
IMPORTANT

You can customize the RHEL installation for your specific requirements using the GUI. You can select additional options for specific environment requirements, for example, Connect to Red Hat, software selection, partitioning, security, and many more. For more information, see Customizing your installation.

System or cloud image-based installations

You can use system or cloud image-based installation methods only in virtual and cloud environments.

To perform a system or cloud image-based installation, use Red Hat Image Builder. Image Builder creates customized system images of Red Hat Enterprise Linux, including the system images for cloud deployment.

For more information about installing RHEL using Image Builder, see the Composing a customized RHEL system image document.

Advanced installations

The following advanced installation methods are available:

- **Perform an automated RHEL installation using Kickstart**: Install Red Hat Enterprise Linux using Kickstart. Kickstart is an automated installation that allows you to execute unattended operating system installation tasks.

- **Perform a remote RHEL installation using VNC**: The RHEL installation program offers two VNC installation modes: Direct and Connect. Once a connection is established, the two modes do not differ. The mode you select depends on your environment.

- **Install RHEL from the network using PXE**: A network installation allows you to install Red Hat Enterprise Linux to a system that has access to an installation server. At a minimum, two systems are required for a network installation.

Additional resources

- For more information about the advanced installation methods, see the Performing an advanced RHEL installation document.

2.3. SYSTEM REQUIREMENTS

If this is a first-time install of Red Hat Enterprise Linux it is recommended that you review the guidelines provided for system, hardware, security, memory, and RAID before installing. See System requirements reference for more information.

If you want to use your system as a virtualization host, review the necessary hardware requirements for virtualization.

Additional resources

- Security hardening
- Composing a customized RHEL system image

2.4. INSTALLATION BOOT MEDIA OPTIONS
There are several options available to boot the Red Hat Enterprise Linux installation program.

**Full installation DVD or USB flash drive**
Create a full installation DVD or USB flash drive using the **DVD ISO** image. The DVD or USB flash drive can be used as a boot device and as an installation source for installing software packages. Due to the size of the DVD ISO image, a DVD or USB flash drive are the recommended media types.

**Minimal installation DVD, CD, or USB flash drive**
Create a minimal installation CD, DVD, or USB flash drive using the **Boot ISO** image, which contains only the minimum files necessary to boot the system and start the installation program.

**IMPORTANT**
If you are not using the Content Delivery Network (CDN) to download the required software packages, the **Boot ISO** image requires an installation source that contains the required software packages.

**PXE Server**
A preboot execution environment (PXE) server allows the installation program to boot over the network. After a system boot, you must complete the installation from a different installation source, such as a local hard drive or a network location.

**Image Builder**
Image Builder allows to create customized system and cloud images to install Red Hat Enterprise Linux in virtual and cloud environment.

**Additional resources**
- Performing an advanced RHEL installation
- Composing a customized RHEL system image

### 2.5. TYPES OF INSTALLATION ISO IMAGES

Two types of Red Hat Enterprise Linux 8 installation ISO images are available from the Red Hat Customer Portal.

**DVD ISO image file**
A full installation program that contains the BaseOS and AppStream repositories and allows you to complete the installation without additional repositories.

**IMPORTANT**
You can use a Binary DVD for 64-bit IBM Z to boot the installation program using a SCSI DVD drive, or as an installation source.

**Boot ISO image file**
The Boot ISO image is a minimal installation that can be used to install RHEL in two different ways:

a. When registering and installing RHEL from the Content Delivery Network (CDN).

b. As a minimal image that requires access to the BaseOS and AppStream repositories to install software packages. The repositories are part of the DVD ISO image that is available
for download from the Red Hat Customer Portal. Download and unpack the DVD ISO image to access the repositories.

The following table contains information about the images that are available for the supported architectures.

### Table 2.1. Boot and installation images

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Installation DVD</th>
<th>Boot DVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD64 and Intel 64</td>
<td>x86_64 DVD ISO image file</td>
<td>x86_64 Boot ISO image file</td>
</tr>
<tr>
<td>ARM 64</td>
<td>AArch64 DVD ISO image file</td>
<td>AArch64 Boot ISO image file</td>
</tr>
<tr>
<td>IBM POWER</td>
<td>ppc64le DVD ISO image file</td>
<td>ppc64le Boot ISO image file</td>
</tr>
<tr>
<td>64-bit IBM Z</td>
<td>s390x DVD ISO image file</td>
<td>s390x Boot ISO image file</td>
</tr>
</tbody>
</table>

### 2.6. DOWNLOADING A RHEL INSTALLATION ISO IMAGE

This section contains instructions about downloading a Red Hat Enterprise Linux installation image from the Red Hat Customer Portal or by using the `curl` command.

#### 2.6.1. Types of installation ISO images

Two types of Red Hat Enterprise Linux 8 installation ISO images are available from the Red Hat Customer Portal.

**DVD ISO image file**

A full installation program that contains the BaseOS and AppStream repositories and allows you to complete the installation without additional repositories.

**IMPORTANT**

You can use a Binary DVD for 64-bit IBM Z to boot the installation program using a SCSI DVD drive, or as an installation source.

**Boot ISO image file**

The Boot ISO image is a minimal installation that can be used to install RHEL in two different ways:

a. When registering and installing RHEL from the Content Delivery Network (CDN).

b. As a minimal image that requires access to the BaseOS and AppStream repositories to install software packages. The repositories are part of the DVD ISO image that is available for download from the Red Hat Customer Portal. Download and unpack the DVD ISO image to access the repositories.

The following table contains information about the images that are available for the supported architectures.
Table 2.2. Boot and installation images

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Installation DVD</th>
<th>Boot DVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD64 and Intel 64</td>
<td>x86_64 DVD ISO image file</td>
<td>x86_64 Boot ISO image file</td>
</tr>
<tr>
<td>ARM 64</td>
<td>AArch64 DVD ISO image file</td>
<td>AArch64 Boot ISO image file</td>
</tr>
<tr>
<td>IBM POWER</td>
<td>ppc64le DVD ISO image file</td>
<td>ppc64le Boot ISO image file</td>
</tr>
<tr>
<td>64-bit IBM Z</td>
<td>s390x DVD ISO image file</td>
<td>s390x Boot ISO image file</td>
</tr>
</tbody>
</table>

2.6.2. Downloading an ISO image from the Customer Portal

This procedure describes how to download a Red Hat Enterprise Linux 8 ISO image file from the Red Hat Customer Portal.

NOTE

- The Boot ISO image is a minimal image file that supports registering your system, attaching subscriptions, and installing RHEL from the Content Delivery Network (CDN).
- The DVD ISO image file contains all repositories and software packages and does not require any additional configuration.

Prerequisites

- You have an active Red Hat subscription.
- You are logged in to the Product Downloads section of the Red Hat Customer Portal at Product Downloads.

Procedure

1. From the Product Downloads page, select the By Category tab.
2. Click the Red Hat Enterprise Linux 8 link. The Download Red Hat Enterprise Linux web page opens.
3. From the Product Variant drop-down menu, select the variant that you require.
   a. Optional: Select the Packages tab to view the packages contained in the selected variant. For information on the packages available in Red Hat Enterprise Linux 8, see the Package Manifest document.
4. The Version drop-down menu defaults to the latest version for the selected variant.
5. The Architecture drop-down menu displays the supported architecture. The Product Software tab displays the image files, which include:
   - Red Hat Enterprise Linux Binary DVD image.
- Red Hat Enterprise Linux Boot ISO image.

Additional images may be available, for example, preconfigured virtual machine images, but they are beyond the scope of this document.

6. Click **Download Now** beside the ISO image that you require.

### 2.6.3. Downloading an ISO image using curl

This section explains how to download installation images using the **curl** command.

#### Prerequisites

- Install **curl** and **jq** package:
  - If your distribution uses the **yum** package manager:
    ```bash
    # yum install curl
    # yum install jq
    ```
  - If your distribution uses the **apt** package manager:
    ```bash
    # apt update
    # apt install curl
    # apt install jq
    ```

- If your Linux distribution does not use **yum**, **dnf**, or **apt**, or if you do not use Linux, download the most appropriate software package from the [curl website](https://curl.se).


- Checksum of the file you want to download from [Product Downloads](https://access.redhat.com).

#### Procedure

1. Create a bash file with the following content:

```bash
#!/bin/bash
# set the offline token and checksum parameters
offline_token="<offline_token>"
checksum=<checksum>

# get an access token
access_token=$(curl https://sso.redhat.com/auth/realms/redhat-external/protocol/openid-connect/token -d grant_type=refresh_token -d client_id=rhsm-api -d refresh_token=$offline_token | jq -r '.access_token')

# get the filename and download url
filename=$(echo $image | jq -r .body.filename)
url=$(echo $image | jq -r .body.href)

# download the file
curl $url -o $filename
```
In this text above, replace the `offline_token` with the token collected from the Red Hat API portal and checksum value taken from the *Product Downloads* page.

2. Make this file executable.

   ```sh
   $ chmod u+x FILEPATH/FILENAME.sh
   ```

3. Open a terminal window and execute the bash file.

   ```sh
   $ ./FILEPATH/FILENAME.sh
   ```

![WARNING]

**WARNING**

Use password management that is consistent with networking best practices.

- Do not store passwords or credentials in a plain text.
- Keep the token safe against unauthorized use.

**Additional resources**

- [Getting started with Red Hat APIs](#)

### 2.7. CREATING A BOOTABLE INSTALLATION MEDIUM FOR RHEL

This section contains information about using the ISO image file that you have downloaded to create a bootable physical installation medium, such as a USB, DVD, or CD. For more information about downloading the ISO images, see [Downloading the installation ISO image](#).

**NOTE**

By default, the `inst.stage2=` boot option is used on the installation medium and is set to a specific label, for example, `inst.stage2=hd:LABEL=RHEL8\x86_64`. If you modify the default label of the file system containing the runtime image, or if you use a customized procedure to boot the installation system, you must verify that the label is set to the correct value.

#### 2.7.1. Installation boot media options

There are several options available to boot the Red Hat Enterprise Linux installation program.

**Full installation DVD or USB flash drive**

Create a full installation DVD or USB flash drive using the **DVD ISO** image. The DVD or USB flash drive can be used as a boot device and as an installation source for installing software packages. Due to the size of the DVD ISO image, a DVD or USB flash drive are the recommended media types.

**Minimal installation DVD, CD, or USB flash drive**

Create a minimal installation CD, DVD, or USB flash drive using the **Boot ISO** image, which contains only the minimum files necessary to boot the system and start the installation program.
IMPORTANT

If you are not using the Content Delivery Network (CDN) to download the required software packages, the **Boot ISO** image requires an installation source that contains the required software packages.

PXE Server

A *preboot execution environment* (PXE) server allows the installation program to boot over the network. After a system boot, you must complete the installation from a different installation source, such as a local hard drive or a network location.

Image Builder

Image Builder allows to create customized system and cloud images to install Red Hat Enterprise Linux in virtual and cloud environment.

Additional resources

- Performing an advanced RHEL installation
- Composing a customized RHEL system image

2.7.2. Creating a bootable DVD or CD

You can create a bootable installation DVD or CD using burning software and a CD/DVD burner. The exact steps to produce a DVD or CD from an ISO image file vary greatly, depending on the operating system and disc burning software installed. Consult your system’s burning software documentation for the exact steps to burn a CD or DVD from an ISO image file.

WARNING

You can create a bootable DVD or CD using either the DVD ISO image (full install) or the Boot ISO image (minimal install). However, the DVD ISO image is larger than 4.7 GB, and as a result, it might not fit on a single or dual-layer DVD. Check the size of the DVD ISO image file before you proceed. A USB key is recommended when using the DVD ISO image to create bootable installation media.

2.7.3. Creating a bootable USB device on Linux

Follow this procedure to create a bootable USB device on a Linux system.

NOTE

This procedure is destructive and data on the USB flash drive is destroyed without a warning.

Prerequisites

- You have downloaded an installation ISO image as described in [Downloading the installation ISO image](#).
The DVD ISO image is larger than 4.7 GB, so a USB flash drive that is large enough to hold the ISO image is required.

**Procedure**

1. Connect the USB flash drive to the system.

2. Open a terminal window and run the `dmesg` command:

   ```
   $ dmesg | tail
   ```

   The `dmesg` command returns a log that details all recent events. Messages resulting from the attached USB flash drive are displayed at the bottom of the log. Record the name of the connected device.

3. Switch to user root:

   ```
   $ su -
   ```

4. Enter your root password when prompted.

5. Find the device node assigned to the drive. In this example, the drive name is `sdd`.

   ```
   # dmesg | tail
   [288954.686557] usb 2-1.8: New USB device strings: Mfr=0, Product=1, SerialNumber=2
   [288954.686559] usb 2-1.8: Product: USB Storage
   [288954.686562] usb 2-1.8: SerialNumber: 000000009225
   [288954.712590] usb-storage 2-1.8:1.0: USB Mass Storage device detected
   [288954.712809] scsi host6: usb-storage 2-1.8:1.0
   [288954.716682] usbcore: registered new interface driver usb-storage
   [288955.717140] scsi 6:0:0:0: Direct-Access Generic STORAGE DEVICE 9228 PQ: 0 ANSI: 0
   [288955.717745] sd 6:0:0:0: Attached scsi generic sg4 type 0
   [288961.876382] sd 6:0:0:0: sdd Attached SCSI removable disk
   ```

6. Run the `dd` command to write the ISO image directly to the USB device.

   ```
   # dd if=/image_directory/image.iso of=/dev/device
   ```

   Replace `/image_directory/image.iso` with the full path to the ISO image file that you downloaded, and replace `device` with the device name that you retrieved with the `dmesg` command. In this example, the full path to the ISO image is `/home/testuser/Downloads/rhel-8-x86_64-boot.iso`, and the device name is `sdd`:

   ```
   # dd if=/home/testuser/Downloads/rhel-8-x86_64-boot.iso of=/dev/sdd
   ```

   **NOTE**

   Ensure that you use the correct device name, and not the name of a partition on the device. Partition names are usually device names with a numerical suffix. For example, `sdd` is a device name, and `sdd1` is the name of a partition on the device `sdd`.  

Red Hat Enterprise Linux 8 System Design Guide
7. Wait for the **dd** command to finish writing the image to the device. The data transfer is complete when the # prompt appears. When the prompt is displayed, log out of the root account and unplug the USB drive. The USB drive is now ready to be used as a boot device.

### 2.7.4. Creating a bootable USB device on Windows

Follow the steps in this procedure to create a bootable USB device on a Windows system. The procedure varies depending on the tool. Red Hat recommends using Fedora Media Writer, available for download at [https://github.com/FedoraQt/MediaWriter/releases](https://github.com/FedoraQt/MediaWriter/releases).

**NOTE**

- Fedora Media Writer is a community product and is not supported by Red Hat. You can report any issues with the tool at [https://github.com/FedoraQt/MediaWriter/issues](https://github.com/FedoraQt/MediaWriter/issues).
- This procedure is destructive and data on the USB flash drive is destroyed without a warning.

### Prerequisites

- You have downloaded an installation ISO image as described in [Downloading the installation ISO image](#).
- The DVD ISO image is larger than 4.7 GB, so a USB flash drive that is large enough to hold the ISO image is required.

### Procedure

1. Download and install Fedora Media Writer from [https://github.com/FedoraQt/MediaWriter/releases](https://github.com/FedoraQt/MediaWriter/releases).

   **NOTE**


2. Connect the USB flash drive to the system.

3. Open Fedora Media Writer.

4. From the main window, click **Custom Image** and select the previously downloaded Red Hat Enterprise Linux ISO image.

5. From **Write Custom Image** window, select the drive that you want to use.

6. Click **Write to disk**. The boot media creation process starts. Do not unplug the drive until the operation completes. The operation may take several minutes, depending on the size of the ISO image, and the write speed of the USB drive.

7. When the operation completes, unmount the USB drive. The USB drive is now ready to be used as a boot device.
2.7.5. Creating a bootable USB device on Mac OS X

Follow the steps in this procedure to create a bootable USB device on a Mac OS X system.

**NOTE**

This procedure is destructive and data on the USB flash drive is destroyed without a warning.

**Prerequisites**

- You have downloaded an installation ISO image as described in [Downloading the installation ISO image](#).
- The DVD ISO image is larger than 4.7 GB, so a USB flash drive that is large enough to hold the ISO image is required.

**Procedure**

1. Connect the USB flash drive to the system.

2. Identify the device path with the `diskutil list` command. The device path has the format of `/dev/disknumber`, where number is the number of the disk. The disks are numbered starting at zero (0). Typically, Disk 0 is the OS X recovery disk, and Disk 1 is the main OS X installation. In the following example, the USB device is `disk2`:

   ```bash
   $ diskutil list
   /dev/disk0
   #: TYPE NAME SIZE IDENTIFIER
   0: GUID_partition_scheme *500.3 GB disk0
   1: EFI EFI 209.7 MB disk0s1
   2: Apple_CoreStorage 400.0 GB disk0s2
   3: Apple_Boot Recovery HD 650.0 MB disk0s3
   4: Apple_CoreStorage 98.8 GB disk0s4
   5: Apple_Boot Recovery HD 650.0 MB disk0s5
   /dev/disk1
   #: TYPE NAME SIZE IDENTIFIER
   0: Apple_HFS YosemiteHD *399.6 GB disk1
   Logical Volume on disk0s1
   8A142795-8036-48DF-9FC5-84506DFBB7B2
   Unlocked Encrypted
   /dev/disk2
   #: TYPE NAME SIZE IDENTIFIER
   0: FDisk_partition_scheme *8.1 GB disk2
   1: Windows_NTFS SanDisk USB 8.1 GB disk2s1
   
   To identify your USB flash drive, compare the NAME, TYPE and SIZE columns to your flash drive. For example, the NAME should be the title of the flash drive icon in the Finder tool. You can also compare these values to those in the information panel of the flash drive.

3. Use the `diskutil unmountDisk` command to unmount the flash drive’s filesystem volumes:

   ```bash
   $ diskutil unmountDisk /dev/disknumber
   Unmount of all volumes on disknumber was successful
   ```
When the command completes, the icon for the flash drive disappears from your desktop. If the icon does not disappear, you may have selected the wrong disk. Attempting to unmount the system disk accidentally returns a **failed to unmount** error.

5. Log in as root:

   $ su -

6. Enter your root password when prompted.

7. Use the `dd` command as a parameter of the `sudo` command to write the ISO image to the flash drive:

   # sudo dd if=/path/to/image.iso of=/dev/rdisknumber

   **NOTE**

   Mac OS X provides both a block (/dev/disk*) and character device (/dev/rdisk*) file for each storage device. Writing an image to the /dev/rdisknumber character device is faster than writing to the /dev/disknumber block device.

8. To write the `/Users/user_name/Downloads/rhel-8-x86_64-boot.iso` file to the `/dev/rdisk2` device, run the following command:

   # sudo dd if=/Users/user_name/Downloads/rhel-8-x86_64-boot.iso of=/dev/rdisk2

9. Wait for the `dd` command to finish writing the image to the device. The data transfer is complete when the `#` prompt appears. When the prompt is displayed, log out of the root account and unplug the USB drive. The USB drive is now ready to be used as a boot device.

### 2.8. PREPARING AN INSTALLATION SOURCE

The Boot ISO image file does not include any repositories or software packages; it contains only the installation program and the tools required to boot the system and start the installation. This section contains information about creating an installation source for the Boot ISO image using the DVD ISO image that contains the required repositories and software packages.

**IMPORTANT**

An installation source is required for the Boot ISO image file only if you decide not to register and install RHEL from the Content Delivery Network (CDN).

#### 2.8.1. Types of installation source

You can use one of the following installation sources for minimal boot images:

- **DVD**: Burn the DVD ISO image to a DVD. The DVD will be automatically used as the installation source (software package source).

- **Hard drive or USB drive**: Copy the DVD ISO image to the drive and configure the installation program to install the software packages from the drive. If you use a USB drive, verify that it is connected to the system before the installation begins. The installation program cannot detect media after the installation begins.
○ **Hard drive limitation:** The DVD ISO image on the hard drive must be on a partition with a file system that the installation program can mount. The supported file systems are `xfs`, `ext2`, `ext3`, `ext4`, and `vfat` (FAT32).

**WARNING**

On Microsoft Windows systems, the default file system used when formatting hard drives is NTFS. The exFAT file system is also available. However, neither of these file systems can be mounted during the installation. If you are creating a hard drive or a USB drive as an installation source on Microsoft Windows, verify that you formatted the drive as FAT32. Note that the FAT32 file system cannot store files larger than 4 GiB.

In Red Hat Enterprise Linux 8, you can enable installation from a directory on a local hard drive. To do so, you need to copy the contents of the DVD ISO image to a directory on a hard drive and then specify the directory as the installation source instead of the ISO image. For example:

```
inst.repo=hd:<device>:<path to the directory>
```

- **Network location:** Copy the DVD ISO image or the installation tree (extracted contents of the DVD ISO image) to a network location and perform the installation over the network using the following protocols:
  - **NFS:** The DVD ISO image is in a Network File System (NFS) share.
  - **HTTPS, HTTP or FTP:** The installation tree is on a network location that is accessible over HTTP, HTTPS or FTP.

### 2.8.2. Specify the installation source

You can specify the installation source using any of the following methods:

- **User interface:** Select the installation source in the **Installation Source** window of the graphical install. For more information, see **Configuring installation source**

- **Boot option:** Configure a custom boot option to specify the installation source. For more information, see, **Boot options preference**

- **Kickstart file:** Use the `install` command in a Kickstart file to specify the installation source. See the **Performing an advanced RHEL installation** document for more information.

### 2.8.3. Ports for network-based installation

The following table lists the ports that must be open on the server providing the files for each type of network-based installation.

**Table 2.3. Ports for network-based installation**
<table>
<thead>
<tr>
<th>Protocol used</th>
<th>Ports to open</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>80</td>
</tr>
<tr>
<td>HTTPS</td>
<td>443</td>
</tr>
<tr>
<td>FTP</td>
<td>21</td>
</tr>
<tr>
<td>NFS</td>
<td>2049, 111, 20048</td>
</tr>
<tr>
<td>TFTP</td>
<td>69</td>
</tr>
</tbody>
</table>

**Additional resources**

- [Securing networks](#)

**2.8.4. Creating an installation source on an NFS server**

Follow the steps in this procedure to place the installation source on an NFS server. Use this installation method to install multiple systems from a single source, without having to connect to physical media.

**Prerequisites**

- You have administrator level access to a server with Red Hat Enterprise Linux8, and this server is on the same network as the system to be installed.
- You have downloaded a Binary DVD image. See [Downloading the installation ISO image](#) from the [Performing a standard RHEL installation](#) document for more information.
- You have created a bootable CD, DVD, or USB device from the image file. See [Creating installation media](#) from the [Performing a standard RHEL installation](#) document for more information.
- You have verified that your firewall allows the system you are installing to access the remote installation source. See [Ports for network-based installation](#) from the [Performing a standard RHEL installation](#) document for more information.

**Procedure**

1. Install the `nfs-utils` package:
   ```
   # yum install nfs-utils
   ```
2. Copy the DVD ISO image to a directory on the NFS server.
3. Open the `/etc/exports` file using a text editor and add a line with the following syntax:
   ```
   /exported_directory/ clients
   ```
4. Replace `/exported_directory/` with the full path to the directory with the ISO image. Replace `clients` with the host name or IP address of the target system, the subnetwork that all target
systems can use to access the ISO image, or the asterisk sign (*) if you want to allow any system with network access to the NFS server to use the ISO image. See the `exports(5)` man page for detailed information about the format of this field.

A basic configuration that makes the `/rhel8-install/` directory available as read-only to all clients is:

```
/rhel8-install *
```

5. Save the `/etc/exports` file and exit the text editor.

6. Start the nfs service:

   ```
   # systemctl start nfs-server.service
   
   # systemctl reload nfs-server.service
   ```

   The ISO image is now accessible over NFS and ready to be used as an installation source.

**NOTE**

When configuring the installation source, use `nfs:` as the protocol, the server host name or IP address, the colon sign (:), and the directory holding the ISO image. For example, if the server host name is `myserver.example.com` and you have saved the ISO image in `/rhel8-install/`, specify `nfs:myserver.example.com:/rhel8-install/` as the installation source.

### 2.8.5. Creating an installation source using HTTP or HTTPS

Follow the steps in this procedure to create an installation source for a network-based installation using an installation tree, which is a directory containing extracted contents of the DVD ISO image and a valid `.treeinfo` file. The installation source is accessed over HTTP or HTTPS.

**Prerequisites**

- You have administrator level access to a server with Red Hat Enterprise Linux8, and this server is on the same network as the system to be installed.

- You have downloaded a Binary DVD image. See Downloading the installation ISO image from the Performing a standard RHEL installation document for more information.

- You have created a bootable CD, DVD, or USB device from the image file. See Creating installation media from the Performing a standard RHEL installation document for more information.

- You have verified that your firewall allows the system you are installing to access the remote installation source. See Ports for network-based installation from the Performing a standard RHEL installation document for more information.

**Procedure**

1. To create an installation source using `http`, install the `httpd` package:
# yum install httpd

To create an installation source using https, install httpd and mod_ssl packages:

# yum install httpd mod_ssl

## WARNING

If your Apache web server configuration enables SSL security, verify that you enable only the TLSv1 protocol, and disable SSLv2 and SSLv3. This is due to the POODLE SSL vulnerability (CVE-2014-3566). See https://access.redhat.com/solutions/1232413 for details.

## IMPORTANT

If you use an HTTPS server with a self-signed certificate, you must boot the installation program with the noverifyssl option.

2. Copy the DVD ISO image to the HTTP(S) server.

3. Mount the DVD ISO image, using the mount command, to a suitable directory:

```bash
# mkdir /mnt/rhel8-install/
# mount -o loop,ro -t iso9660 /image_directory/image.iso /mnt/rhel8-install/
```

Replace /image_directory/image.iso with the path to the DVD ISO image.

4. Copy the files from the mounted image to the HTTP(S) server root. This command creates the /var/www/html/rhel8-install/ directory with the contents of the image.

```bash
# cp -r /mnt/rhel8-install/ /var/www/html/
```

This command creates the /var/www/html/rhel8-install/ directory with the content of the image. Note that some copying methods can skip the .treeinfo file which is required for a valid installation source. Running the cp command for whole directories as shown in this procedure will copy .treeinfo correctly.

5. Start the httpd service:

```bash
# systemctl start httpd.service
```

The installation tree is now accessible and ready to be used as the installation source.
When configuring the installation source, use http:// or https:// as the protocol, the server host name or IP address, and the directory that contains the files from the ISO image, relative to the HTTP server root. For example, if you are using HTTP, the server host name is myserver.example.com, and you have copied the files from the image to /var/www/html/rhel8-install, specify http://myserver.example.com/rhel8-install/ as the installation source.

Additional resources

- Deploying different types of servers

### 2.8.6. Creating an installation source using FTP

Follow the steps in this procedure to create an installation source for a network-based installation using an installation tree, which is a directory containing extracted contents of the DVD ISO image and a valid .treeinfo file. The installation source is accessed over FTP.

**Prerequisites**

- You have administrator level access to a server with Red Hat Enterprise Linux8, and this server is on the same network as the system to be installed.
- You have downloaded a Binary DVD image. See Downloading the installation ISO image from the Performing a standard RHEL installation document for more information.
- You have created a bootable CD, DVD, or USB device from the image file. See Creating installation media from the Performing a standard RHEL installation document for more information.
- You have verified that your firewall allows the system you are installing to access the remote installation source. See Ports for network-based installation from the Performing a standard RHEL installation document for more information.

**Procedure**

1. Install the vsftpd package by running the following command as root:

   ```bash
   # yum install vsftpd
   ```

2. Open and edit the /etc/vsftpd/vsftpd.conf configuration file in a text editor.

   a. Change the line anonymous_enable=NO to anonymous_enable=YES

   b. Change the line write-enable=YES to write-enable=NO.

   c. Add lines pasv_min_port=\texttt{min\_port} and \texttt{pasp\_max\_port=max\_port}. Replace \texttt{min\_port} and \texttt{max\_port} with the port number range used by FTP server in passive mode, e. g. 10021 and 10031.

   This step can be necessary in network environments featuring various firewall/NAT setups.

   d. Optionally, add custom changes to your configuration. For available options, see the vsftpd.conf(5) man page. This procedure assumes that default options are used.
3. Configure the server firewall.
   a. Enable the firewall:
      ```
      # systemctl enable firewalld
      # systemctl start firewalld
      ```
   b. Enable in your firewall the FTP port and port range from previous step:
      ```
      # firewall-cmd --add-port min_port-max_port/tcp --permanent
      # firewall-cmd --add-service ftp --permanent
      # firewall-cmd --reload
      ```
      Replace `min_port-max_port` with the port numbers you entered into the `/etc/vsftpd/vsftpd.conf` configuration file.

4. Copy the DVD ISO image to the FTP server.

5. Mount the DVD ISO image, using the mount command, to a suitable directory:
   ```
   # mkdir /mnt/rhel8-install
   # mount -o loop,ro -t iso9660 /image-directory/image.iso /mnt/rhel8-install
   ```
   Replace `/image-directory/image.iso` with the path to the DVD ISO image.

6. Copy the files from the mounted image to the FTP server root:
   ```
   # mkdir /var/ftp/rhel8-install
   # cp -r /mnt/rhel8-install /var/ftp/
   ```
   This command creates the `/var/ftp/rhel8-install/` directory with the content of the image. Note that some copying methods can skip the `.treeinfo` file which is required for a valid installation source. Running the `cp` command for whole directories as shown in this procedure will copy `.treeinfo` correctly.

7. Make sure that the correct SELinux context and access mode is set on the copied content:
   ```
   # restorecon -r /var/ftp/rhel8-install
   # find /var/ftp/rhel8-install -type f -exec chmod 444 {} ;
   # find /var/ftp/rhel8-install -type d -exec chmod 755 {} ;
   ```

8. Start the `vsftpd` service:
# systemctl start vsftpd.service

If the service was running before you changed the `/etc/vsftpd/vsftpd.conf` file, restart the service to load the edited file:

# systemctl restart vsftpd.service

Enable the `vsftpd` service to start during the boot process:

# systemctl enable vsftpd

The installation tree is now accessible and ready to be used as the installation source.

NOTE

When configuring the installation source, use `ftp://` as the protocol, the server host name or IP address, and the directory in which you have stored the files from the ISO image, relative to the FTP server root. For example, if the server host name is `myserver.example.com` and you have copied the files from the image to `/var/ftp/rhel8-install/`, specify `ftp://myserver.example.com/rhel8-install/` as the installation source.

2.8.7. Preparing a hard drive as an installation source

This module describes how to install RHEL using a hard drive as an installation source with `ext2`, `ext3`, `ext4`, or `XFS` file systems. You can use this method for the systems without network access and the optical drive. Hard drive installations use an ISO image of the installation DVD. An ISO image is a file that contains an exact copy of the content of a DVD. With this file present on a hard drive, you can choose Hard drive as the installation source when you boot the installation program.

- To check the file system of a hard drive partition on a Windows operating system, use the **Disk Management** tool.

- To check the file system of a hard drive partition on a Linux operating system, use the **parted** tool.

NOTE

You cannot use ISO files on LVM (Logical Volume Management) partitions.

Procedure

1. Download an ISO image of the Red Hat Enterprise Linux installation DVD. Alternatively, if you have the DVD on physical media, you can create an image of an ISO with the following command on a Linux system:

   ```bash
   dd if=/dev/dvd of=/path_to_image/name_of_image.iso
   ```

   where `dvd` is your DVD drive device name, `name_of_image` is the name you give to the resulting ISO image file, and `path_to_image` is the path to the location on your system where you want to store the image.

2. Copy and paste the ISO image onto the system hard drive or a USB drive.
3. Use a SHA256 checksum program to verify that the ISO image that you copied is intact. Many SHA256 checksum programs are available for various operating systems. On a Linux system, run:

```
$ sha256sum /path_to_image/name_of_image.iso
```

where `name_of_image` is the name of the ISO image file. The SHA256 checksum program displays a string of 64 characters called a hash. Compare this hash to the hash displayed for this particular image on the Downloads page in the Red Hat Customer Portal. The two hashes should be identical.

4. Specify the HDD installation source on the kernel command line before starting the installation:

```
inst.repo=hd:<device>:/path_to_image/name_of_image.iso
```

**Additional resources**

- Specify the installation source
CHAPTER 3. GETTING STARTED

To get started with the installation, first review the boot menu and the available boot options. Then, depending on the choice you make, proceed to boot the installation.

3.1. BOOTING THE INSTALLATION

After you have created bootable media you are ready to boot the Red Hat Enterprise Linux installation.

3.1.1. Boot menu

The Red Hat Enterprise Linux boot menu is displayed using GRand Unified Bootloader version 2 (GRUB2) when your system has completed loading the boot media.

Figure 3.1. Red Hat Enterprise Linux boot menu

The boot menu provides several options in addition to launching the installation program. If you do not make a selection within 60 seconds, the default boot option (highlighted in white) is run. To select a different option, use the arrow keys on your keyboard to make your selection and press the Enter key.

You can customize boot options for a particular menu entry:

- **On BIOS-based systems**: Press the Tab key and add custom boot options to the command line. You can also access the boot: prompt by pressing the Esc key but no required boot options are preset. In this scenario, you must always specify the Linux option before using any other boot options.
On UEFI-based systems: Press the e key and add custom boot options to the command line. When ready press Ctrl+X to boot the modified option.

Table 3.1. Boot menu options

<table>
<thead>
<tr>
<th>Boot menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install Red Hat Enterprise Linux 8</td>
<td>Use this option to install Red Hat Enterprise Linux using the graphical installation program. For more information, Installing RHEL using an ISO image from the Customer Portal</td>
</tr>
<tr>
<td>Test this media &amp; install Red Hat Enterprise Linux 8</td>
<td>Use this option to check the integrity of the installation media. For more information, see Verifying a boot media</td>
</tr>
<tr>
<td>Troubleshooting &gt;</td>
<td>Use this option to resolve various installation issues. Press Enter to display its contents.</td>
</tr>
</tbody>
</table>

Table 3.2. Troubleshooting options

<table>
<thead>
<tr>
<th>Troubleshooting option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Troubleshooting &gt; Install Red Hat Enterprise Linux 8 in basic graphics mode</td>
<td>Use this option to install Red Hat Enterprise Linux in graphical mode even if the installation program is unable to load the correct driver for your video card. If your screen is distorted when using the Install Red Hat Enterprise Linux 8 option, restart your system and use this option. For more information, see Cannot boot into graphical installation</td>
</tr>
<tr>
<td>Troubleshooting &gt; Rescue a Red Hat Enterprise Linux system</td>
<td>Use this option to repair any issues that prevent you from booting. For more information, see Using a rescue mode</td>
</tr>
<tr>
<td>Troubleshooting &gt; Run a memory test</td>
<td>Use this option to run a memory test on your system. Press Enter to display its contents. For more information, see memtest86</td>
</tr>
<tr>
<td>Troubleshooting &gt; Boot from local drive</td>
<td>Use this option to boot the system from the first installed disk. If you booted this disk accidentally, use this option to boot from the hard disk immediately without starting the installation program.</td>
</tr>
</tbody>
</table>

3.1.2. Types of boot options

There are two types of boot options; those with an equals "=" sign, and those without an equals "=" sign. Boot options are appended to the boot command line and multiple options must be separated by a single space. Boot options that are specific to the installation program always start with inst.

Options with an equals "=" sign
You must specify a value for boot options that use the = symbol. For example, the `inst.vncpassword=` option must contain a value, in this case, a password. The correct syntax for this example is `inst.vncpassword=password`.

**Options without an equals "=" sign**

This boot option does not accept any values or parameters. For example, the `rd.live.check` option forces the installation program to verify the installation media before starting the installation. If this boot option is present, the verification is performed; if the boot option is not present, the verification is skipped.

### 3.1.3. Editing the boot: prompt in BIOS

When using the `boot:` prompt, the first option must always specify the installation program image file that you want to load. In most cases, you can specify the image using the keyword. You can specify additional options according to your requirements.

**Prerequisites**

- You have created bootable installation media (USB, CD or DVD).
- You have booted the installation from the media, and the installation boot menu is open.

**Procedure**

1. With the boot menu open, press the **Esc** key on your keyboard.
2. The `boot:` prompt is now accessible.
3. Press the **Tab** key on your keyboard to display the help commands.
4. Press the **Enter** key on your keyboard to start the installation with your options. To return from the `boot:` prompt to the boot menu, restart the system and boot from the installation media again.

**NOTE**

The `boot:` prompt also accepts `dracut` kernel options. A list of options is available in the `dracut.cmdline(7)` man page.

### 3.1.4. Editing the > prompt

You can use the `>` prompt to edit predefined boot options. For example, select **Test this media and install Red Hat Enterprise Linux 8** from the boot menu to display a full set of options.

**NOTE**

This procedure is for BIOS-based AMD64 and Intel 64 systems.

**Prerequisites**

- You have created bootable installation media (USB, CD or DVD).
- You have booted the installation from the media, and the installation boot menu is open.

**Procedure**
1. From the boot menu, select an option and press the **Tab** key on your keyboard. The > prompt is accessible and displays the available options.

2. Append the options that you require to the > prompt.

3. Press the **Enter** key on your keyboard to start the installation.

4. Press the **Esc** key on your keyboard to cancel editing and return to the boot menu.

### 3.1.5. Editing the GRUB2 menu

The GRUB2 menu is available on UEFI-based AMD64, Intel 64, and 64-bit ARM systems.

**Prerequisites**

- You have created bootable installation media (USB, CD or DVD).
- You have booted the installation from the media, and the installation boot menu is open.

**Procedure**

1. From the boot menu window, select the required option and press the **e** key on your keyboard.

2. Move the cursor to the kernel command line. On UEFI systems, the kernel command line starts with `linuxefi`.

3. Move the cursor to the end of the `linuxefi` kernel command line.

4. Edit the parameters as required. For example, to configure one or more network interfaces, add the `ip=` parameter at the end of the `linuxefi` kernel command line, followed by the required value.

5. When you finish editing, press **Ctrl+X** on your keyboard to start the installation using the specified options.

### 3.1.6. Booting the installation from a USB, CD, or DVD

Follow the steps in this procedure to boot the Red Hat Enterprise Linux installation using a USB, CD, or DVD. The following steps are generic. Consult your hardware manufacturer’s documentation for specific instructions.

**Prerequisite**

You have created bootable installation media (USB, CD or DVD). See [Creating a bootable DVD or CD](#) for more information.

**Procedure**

1. Power off the system to which you are installing Red Hat Enterprise Linux.

2. Disconnect any drives from the system.

3. Power on the system.

4. Insert the bootable installation media (USB, DVD, or CD).
5. Power off the system but do not remove the boot media.

6. Power on the system.

**NOTE**

You might need to press a specific key or combination of keys to boot from the media or configure the Basic Input/Output System (BIOS) of your system to boot from the media. For more information, see the documentation that came with your system.

7. The **Red Hat Enterprise Linux boot** window opens and displays information about a variety of available boot options.

8. Use the arrow keys on your keyboard to select the boot option that you require, and press **Enter** to select the boot option. The **Welcome to Red Hat Enterprise Linux** window opens and you can install Red Hat Enterprise Linux using the graphical user interface.

**NOTE**

The installation program automatically begins if no action is performed in the boot window within 60 seconds.

9. Optionally, edit the available boot options:

   a. **UEFI-based systems**: Press **E** to enter edit mode. Change the predefined command line to add or remove boot options. Press **Enter** to confirm your choice.

   b. **BIOS-based systems**: Press the **Tab** key on your keyboard to enter edit mode. Change the predefined command line to add or remove boot options. Press **Enter** to confirm your choice.

Additional Resources

- Graphical installation
- Custom boot options

### 3.1.7. Booting the installation from a network using PXE

When installing Red Hat Enterprise Linux on a large number of systems simultaneously, the best approach is to boot from a PXE server and install from a source in a shared network location. Follow the steps in this procedure to boot the Red Hat Enterprise Linux installation from a network using PXE.

**NOTE**

To boot the installation process from a network using PXE, you must use a physical network connection, for example, Ethernet. You cannot boot the installation process with a wireless connection.

**Prerequisites**

- You have configured a TFTP server, and there is a network interface in your system that supports PXE. See **Additional resources** for more information.
You have configured your system to boot from the network interface. This option is in the BIOS, and can be labeled **Network Boot** or **Boot Services**.

You have verified that the BIOS is configured to boot from the specified network interface. Some BIOS systems specify the network interface as a possible boot device, but do not support the PXE standard. See your hardware’s documentation for more information. When you have properly enabled PXE booting, the system can boot the Red Hat Enterprise Linux installation program without any other media.

### Procedure

1. Verify that the network cable is attached. The link indicator light on the network socket should be lit, even if the computer is not switched on.

2. Switch on the system.
   Depending on your hardware, some network setup and diagnostic information can be displayed before your system connects to a PXE server. When connected, a menu is displayed according to the PXE server configuration.

3. Press the number key that corresponds to the option that you require.
   
   **NOTE**

   In some instances, boot options are not displayed. If this occurs, press the **Enter** key on your keyboard or wait until the boot window opens.

   The **Red Hat Enterprise Linux boot** window opens and displays information about a variety of available boot options.

4. Use the arrow keys on your keyboard to select the boot option that you require, and press **Enter** to select the boot option. The **Welcome to Red Hat Enterprise Linux** window opens and you can install Red Hat Enterprise Linux using the graphical user interface.
   
   **NOTE**

   The installation program automatically begins if no action is performed in the boot window within 60 seconds.

5. Optionally, edit the available boot options:
   a. **UEFI-based systems**: Press **E** to enter edit mode. Change the predefined command line to add or remove boot options. Press **Enter** to confirm your choice.
   b. **BIOS-based systems**: Press the **Tab** key on your keyboard to enter edit mode. Change the predefined command line to add or remove boot options. Press **Enter** to confirm your choice.

### Additional Resources

- **Performing an advanced RHEL installation**

- Refer to the Boot Options Reference for more information about the list of available boot options you can use on the boot command line.
3.2. INSTALLING RHEL USING AN ISO IMAGE FROM THE CUSTOMER PORTAL

Use this procedure to install RHEL using a DVD ISO image that you downloaded from the Customer Portal. The steps provide instructions to follow the RHEL Installation Program.

**WARNING**

When performing a GUI installation using the DVD ISO image file, a race condition in the installer can sometimes prevent the installation from proceeding until you register the system using the Connect to Red Hat feature. For more information, see BZ#1823578 in the Known Issues section of the RHEL Release Notes document.

**Prerequisites**

- You have downloaded the DVD ISO image file from the Customer Portal. See Downloading beta installation images for more information.
- You have created bootable installation media. See Creating a bootable DVD or CD for more information.
- You have booted the installation program and the boot menu is displayed. See Booting the installer

**Procedure**

1. From the boot menu, select **Install Red Hat Enterprise Linux 8** and press **Enter** on your keyboard.

2. In the **Welcome to Red Hat Enterprise Linux 8** window, select your language and location, and click **Continue**. The **Installation Summary** window opens and displays the default values for each setting.

3. Select **System > Installation Destination** and in the **Local Standard Disks** pane, select the target disk and then click **Done**. The default settings are selected for the storage configuration.

4. Select **System > Network & Host Name** The **Network and Hostname** window opens.

5. In the **Network and Hostname** window, toggle the **Ethernet** switch to **ON**, and then click **Done**. The installer connects to an available network and configures the devices available on the network. If required, from the list of networks available, you can choose a desired network and configure the devices that are available on that network.


7. In the **Root Password** window, type the password that you want to set for the root account, and then click **Done**. A root password is required to finish the installation process and to log in to the system administrator user account.
8. Optional: Select User Settings > User Creation to create a user account for the installation process to complete. In place of the root account, you can use this user account to perform any system administrative tasks.

9. In the Create User window, perform the following, and then click Done.
   a. Type a name and user name for the account that you want to create.
   b. Select the Make this user administrator and the Require a password to use this account check boxes. The installation program adds the user to the wheel group, and creates a password protected user account with default settings. It is recommended to create a password protected administrative user account.

10. Click Begin Installation to start the installation, and wait for the installation to complete. It might take a few minutes.

11. When the installation process is complete, click Reboot to restart the system.

12. Remove any installation media if it is not ejected automatically upon reboot. Red Hat Enterprise Linux8 starts after your system’s normal power-up sequence is complete. If your system was installed on a workstation with the X Window System, applications to configure your system are launched. These applications guide you through initial configuration and you can set your system time and date, register your system with Red Hat, and more. If the X Window System is not installed, a login: prompt is displayed.

   **NOTE**
   
   If you have installed a Red Hat Enterprise Linux Beta release, on systems having UEFI Secure Boot enabled, then add the Beta public key to the system’s Machine Owner Key (MOK) list.

13. From the Initial Setup window, accept the licensing agreement and register your system.

**Additional resources**

- Performing a standard RHEL installation
- Installation boot media options

### 3.3. REGISTERING AND INSTALLING RHEL FROM THE CDN USING THE GUI

This section contains information about how to register your system, attach RHEL subscriptions, and install RHEL from the Red Hat Content Delivery Network (CDN) using the GUI.

#### 3.3.1. What is the Content Delivery Network

The Red Hat Content Delivery Network (CDN), available from cdn.redhat.com, is a geographically distributed series of static web servers that contain content and errata that is consumed by systems. The content can be consumed directly, such as using a system registered to Red Hat Subscription Management. The CDN is protected by x.509 certificate authentication to ensure that only valid users have access. When a system is registered to Red Hat Subscription Management, the attached subscriptions govern which subset of the CDN the system can access.

Registering and installing RHEL from the CDN provides the following benefits:
The CDN installation method supports the Boot ISO and the DVD ISO image files. However, the use of the smaller Boot ISO image file is recommended as it consumes less space than the larger DVD ISO image file.

The CDN uses the latest packages resulting in a fully up-to-date system right after installation. There is no requirement to install package updates immediately after installation as is often the case when using the DVD ISO image file.

Integrated support for connecting to Red Hat Insights and enabling System Purpose.

Registering and installing RHEL from the CDN is supported by the GUI and Kickstart. For information about how to register and install RHEL using the GUI, see the *Performing a standard RHEL installation* document. For information about how to register and install RHEL using Kickstart, see the *Performing an advanced RHEL installation* document.

### 3.3.2. Registering and installing RHEL from the CDN

Use this procedure to register your system, attach RHEL subscriptions, and install RHEL from the Red Hat Content Delivery Network (CDN) using the GUI.

**IMPORTANT**

The CDN feature is supported by the *Boot ISO* and *DVD ISO* image files. However, it is recommended that you use the *Boot ISO* image file as the installation source defaults to CDN for the Boot ISO image file.

**Prerequisites**

- Your system is connected to a network that can access the CDN.
- You have downloaded the *Boot ISO* image file from the Customer Portal.
- You have created bootable installation media.
- You have booted the installation program and the boot menu is displayed. Note that the installation repository used after system registration is dependent on how the system was booted.

**Procedure**

1. From the boot menu, select **Install Red Hat Enterprise Linux 8** and press **Enter** on your keyboard.

2. In the **Welcome to Red Hat Enterprise Linux 8** window, select your language and location, and click **Continue**. The **Installation Summary** window opens and displays the default values for each setting.

3. Select **System** > **Installation Destination** and in the **Local Standard Disks** pane, select the target disk and then click **Done**. The default settings are selected for the storage configuration. For more information about customizing the storage settings, see Configuring software settings, Storage devices, Manual partitioning.

4. Select **System** > **Network & Host Name** The **Network and Hostname** window opens.

5. In the **Network and Hostname** window, toggle the **Ethernet** switch to **ON**, and then click **Done**. The installer connects to an available network and configures the devices available on the...
If required, from the list of networks available, you can choose a desired network and configure the devices that are available on that network. For more information about configuring a network or network devices, see Network hostname.


7. In the Connect to Red Hat window, perform the following steps:

   a. Select the Authentication method, and provide the details based on the method you select.
      For Account authentication method: Enter your Red Hat Customer Portal username and password details.
      For Activation Key authentication method: Enter your organization ID and activation key. You can enter more than one activation key, separated by a comma, as long as the activation keys are registered to your subscription.

   b. Select the Set System Purpose check box, and then select the required Role, SLA, and Usage from the corresponding drop-down lists.
      With System Purpose you can record the intended use of a Red Hat Enterprise Linux 8 system, and ensure that the entitlement server auto-attaches the most appropriate subscription to your system.

   c. The Connect to Red Hat Insights check box is enabled by default. Clear the check box if you do not want to connect to Red Hat Insights.
      Red Hat Insights is a Software-as-a-Service (SaaS) offering that provides continuous, in-depth analysis of registered Red Hat-based systems to proactively identify threats to security, performance and stability across physical, virtual and cloud environments, and container deployments.

   d. Optionally, expand Options, and select the network communication type.
      - Select the Use HTTP proxy check box if your network environment allows external Internet access only or accesses the content servers through an HTTP proxy.
      - If you are running Satellite Server or performing internal testing, select the Custom server URL and Custom base URL check boxes and enter the required details.
        The Custom server URL field does not require the HTTP protocol, for example nameofhost.com. However, the Custom base URL field requires the HTTP protocol.
        To change the Custom base URL after registration, you must unregister, provide the new details, and then re-register.

   e. Click Register. When the system is successfully registered and subscriptions are attached, the Connect to Red Hat window displays the attached subscription details. Depending on the amount of subscriptions, the registration and attachment process might take up to a minute to complete.

   f. Click Done.
      A Registered message is displayed under Connect to Red Hat


9. In the Root Password window, type the password that you want to set for the root account, and then click Done. A root password is required to finish the installation process and to log in to the system administrator user account.
   For more details about the requirements and recommendations for creating a password, see Configuring a root password.
10. Optional: Select **User Settings > User Creation** to create a user account for the installation process to complete. In place of the root account, you can use this user account to perform any system administrative tasks.

11. In the **Create User** window, perform the following, and then click **Done**.

   a. Type a name and user name for the account that you want to create.

   b. Select the **Make this user administrator** and the **Require a password to use this account** check boxes. The installation program adds the user to the wheel group, and creates a password protected user account with default settings. It is recommended to create a password protected administrative user account.

      For more information about editing the default settings for a user account, see **Creating a user account**.

12. Click **Begin Installation** to start the installation, and wait for the installation to complete. It might take a few minutes.

13. When the installation process is complete, click **Reboot** to restart the system.

14. Remove any installation media if it is not ejected automatically upon reboot.

   Red Hat Enterprise Linux 8 starts after your system’s normal power-up sequence is complete. If your system was installed on a workstation with the X Window System, applications to configure your system are launched. These applications guide you through initial configuration and you can set your system time and date, register your system with Red Hat, and more. If the X Window System is not installed, a **login:** prompt is displayed.

   **NOTE**

   If you have installed a Red Hat Enterprise Linux Beta release, on systems having UEFI Secure Boot enabled, then add the Beta public key to the system’s Machine Owner Key (MOK) list.

15. From the **Initial Setup** window, accept the licensing agreement and register your system.

**Additional resources**

- How to customize your network, connect to Red Hat, system purpose, installation destination, KDUMP, and security policy
- Red Hat Insights product documentation
- Understanding Activation Keys
- Using an HTTP proxy

**3.3.2.1. Installation source repository after system registration**

The installation source repository used after system registration is dependent on how the system was booted.

**System booted from the Boot ISO or the DVD ISO image file**

If you booted the RHEL installation using either the **Boot ISO** or the **DVD ISO** image file with the default boot parameters, the installation program automatically switches the installation source repository to the CDN after registration.
System booted with the `inst.repo=<URL>` boot parameter

If you booted the RHEL installation with the `inst.repo=<URL>` boot parameter, the installation program does not automatically switch the installation source repository to the CDN after registration. If you want to use the CDN to install RHEL, you must manually switch the installation source repository to the CDN by selecting the Red Hat CDN option in the Installation Source window of the graphical installation. If you do not manually switch to the CDN, the installation program installs the packages from the repository specified on the kernel command line.

**IMPORTANT**

- You can switch the installation source repository to the CDN using the `rhsm` Kickstart command only if you do not specify an installation source using `inst.repo=` on the kernel command line or the `url` command in the Kickstart file. You must use `inst.stage2=<URL>` on the kernel command line to fetch the installation image, but not specify the installation source.

- An installation source URL specified using a boot option or included in a Kickstart file takes precedence over the CDN, even if the Kickstart file contains the `rhsm` command with valid credentials. The system is registered, but it is installed from the URL installation source. This ensures that earlier installation processes operate as normal.

### 3.3.3. Verifying your system registration from the CDN

Use this procedure to verify that your system is registered to the CDN using the GUI.

**WARNING**

You can only verify your registration from the CDN if you have **not** clicked the Begin Installation button from the Installation Summary window. Once the Begin Installation button is clicked, you cannot return to the Installation Summary window to verify your registration.

**Prerequisite**

- You have completed the registration process as documented in the Register and install from CDN using GUI and Registered is displayed under Connect to Red Hat on the Installation Summary window.

**Procedure**

1. From the Installation Summary window, select Connect to Red Hat

2. The window opens and displays a registration summary:
   - **Method**
     - The registered account name or activation keys are displayed.
   - **System Purpose**
     - If set, the role, SLA, and usage details are displayed.
Insights
   If enabled, the Insights details are displayed.

Number of subscriptions
   The number of subscriptions attached are displayed. Note: In the simple content access mode, no subscription being listed is a valid behavior.

3. Verify that the registration summary matches the details that were entered.

Additional resources
   - Simple Content Access

3.3.4. Unregistering your system from the CDN

Use this procedure to unregister your system from the CDN using the GUI.

WARNING
   - You can unregister from the CDN if you have not clicked the Begin Installation button from the Installation Summary window. Once the Begin Installation button is clicked, you cannot return to the Installation Summary window to unregister your registration.

   - When unregistering, the installation program switches to the first available repository, in the following order:
      a. The URL used in the inst.repo=<URL> boot parameter on the kernel command line.
      b. An automatically detected repository on the installation media (USB or DVD).

Prerequisite
   - You have completed the registration process as documented in the Registering and installing RHEL from the CDN and Registered is displayed under Connect to Red Hat on the Installation Summary window.

Procedure
   1. From the Installation Summary window, select Connect to Red Hat
   2. The Connect to Red Hat window opens and displays a registration summary:
      Method
         The registered account name or activation keys used are displayed.
      System Purpose
         If set, the role, SLA, and usage details are displayed.
Insights
If enabled, the Insights details are displayed.

Number of subscriptions
The number of subscriptions attached are displayed. Note: In the simple content access mode, no subscription being listed is a valid behavior.

3. Click **Unregister** to remove the registration from the CDN. The original registration details are displayed with a **Not registered** message displayed in the lower-middle part of the window.

4. Click **Done** to return to the **Installation Summary** window.

5. **Connect to Red Hat** displays a **Not registered** message, and **Software Selection** displays a **Red Hat CDN requires registration** message.

**NOTE**
After unregistering, it is possible to register your system again. Click **Connect to Red Hat**. The previously entered details are populated. Edit the original details, or update the fields based on the account, purpose, and connection. Click **Register** to complete.

**3.4. COMPLETING THE INSTALLATION**

Wait for the installation to complete. It might take a few minutes.

After the installation is complete, remove any installation media if it is not ejected automatically upon reboot.

Red Hat Enterprise Linux 8 starts after your system’s normal power-up sequence is complete. If your system was installed on a workstation with the X Window System, applications to configure your system are launched. These applications guide you through initial configuration and you can set your system time and date, register your system with Red Hat, and more. If the X Window System is not installed, a **login:** prompt is displayed.

To learn how to complete initial setup, register, and secure your system, see the **Completing post-installation tasks** section of the **Performing a standard RHEL installation** document.
CHAPTER 4. CUSTOMIZING YOUR INSTALLATION

When installing Red Hat Enterprise Linux, you can customize location, software, and system settings and parameters, using the Installation Summary window.

The Installation Summary window contains the following categories:

LOCALIZATION

You can configure Keyboard, Language Support, and Time and Date.

SOFTWARE

You can configure Connect to Red Hat, Installation Source, and Software Selection.

SYSTEM

You can configure Installation Destination, KDUMP, Network and Host Name, and Security Policy.

USER SETTINGS

You can configure a root password to log in to the administrator account that is used for system administration tasks, and create a user account to login to the system.

A category has a different status depending on where it is in the installation program.

Table 4.1. Category status

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yellow triangle with an exclamation mark and red text</td>
<td>Requires attention before installation. For example, Network &amp; Host Name requires attention before you can register and download from the Content Delivery Network (CDN).</td>
</tr>
<tr>
<td>Grayed out and with a warning symbol (yellow triangle with an exclamation mark)</td>
<td>The installation program is configuring a category and you must wait for it to finish before accessing the window.</td>
</tr>
</tbody>
</table>

NOTE

A warning message is displayed at the bottom of the Installation Summary window and the Begin Installation button is disabled until you configure all of the required categories.

This section contains information about customizing your Red Hat Enterprise Linux installation using the Graphical User Interface (GUI). The GUI is the preferred method of installing Red Hat Enterprise Linux when you boot the system from a CD, DVD, or USB flash drive, or from a network using PXE.

NOTE

There may be some variance between the online help and the content that is published on the Customer Portal. For the latest updates, see the installation content on the Customer Portal.

4.1. CONFIGURING LANGUAGE AND LOCATION SETTINGS
The installation program uses the language that you selected during installation.

Prerequisites

1. You created installation media. See Creating a bootable DVD or CD
2. You specified an installation source if you are using the Boot ISO image file. See Preparing an installation source
3. You booted the installation. See Booting the installer

Procedure

1. From the left-hand pane of the Welcome to Red Hat Enterprise Linux window, select a language. Alternatively, type your preferred language into the Search field.

   **NOTE**

   A language is pre-selected by default. If network access is configured, that is, if you booted from a network server instead of local media, the pre-selected language is determined by the automatic location detection feature of the GeoIP module. If you used the `inst.lang=` option on the boot command line or in your PXE server configuration, then the language that you define with the boot option is selected.

2. From the right-hand pane of the Welcome to Red Hat Enterprise Linux window, select a location specific to your region.
3. Click Continue to proceed to the Graphical installations window.
4. If you are installing a pre-release version of Red Hat Enterprise Linux, a warning message is displayed about the pre-release status of the installation media.
   a. To continue with the installation, click I want to proceed, or
   b. To quit the installation and reboot the system, click I want to exit.

Additional resources

- Configuring localization settings

4.2. CONFIGURING LOCALIZATION OPTIONS

This section contains information about configuring your keyboard, language support, and time and date settings.

**IMPORTANT**

If you use a layout that cannot accept Latin characters, such as Russian, add the English (United States) layout and configure a keyboard combination to switch between the two layouts. If you select a layout that does not have Latin characters, you might be unable to enter a valid root password and user credentials later in the installation process. This might prevent you from completing the installation.
CONFIGURING KEYBOARD, LANGUAGE, AND TIME AND DATE SETTINGS

Keyboard, Language, and Time and Date Settings are configured by default as part of Installing RHEL using Anaconda. To change any of the settings, complete the following steps, otherwise proceed to Configuring software settings.

Procedure

1. Configure keyboard settings:
   a. From the Installation Summary window, click Keyboard. The default layout depends on the option selected in Installing RHEL using Anaconda.
   b. Click + to open the Add a Keyboard Layout window and change to a different layout.
   c. Select a layout by browsing the list or use the Search field.
   d. Select the required layout and click Add. The new layout appears under the default layout.
   e. Click Options to optionally configure a keyboard switch that you can use to cycle between available layouts. The Layout Switching Options window opens.
   f. To configure key combinations for switching, select one or more key combinations and click OK to confirm your selection.
   
   NOTE
   When you select a layout, click the Keyboard button to open a new dialog box that displays a visual representation of the selected layout.
   
   g. Click Done to apply the settings and return to Graphical installations.

2. Configure language settings:
   a. From the Installation Summary window, click Language Support. The Language Support window opens. The left pane lists the available language groups. If at least one language from a group is configured, a check mark is displayed and the supported language is highlighted.
   b. From the left pane, click a group to select additional languages, and from the right pane, select regional options. Repeat this process for languages that you require.
   c. Click Done to apply the changes and return to Graphical installations.

3. Configure time and date settings:
   a. From the Installation Summary window, click Time & Date. The Time & Date window opens.
NOTE

The **Time & Date** settings are configured by default based on the settings you selected in *Installing RHEL using Anaconda*.

The list of cities and regions come from the Time Zone Database (**tzdata**) public domain that is maintained by the Internet Assigned Numbers Authority (IANA). Red Hat can not add cities or regions to this database. You can find more information at the [IANA official website](https://en.wikipedia.org/wiki/List_of_tz_database_time_zones).

b. From the **Region** drop-down menu, select a region.

NOTE

Select **Etc** as your region to configure a time zone relative to Greenwich Mean Time (GMT) without setting your location to a specific region.

c. From the **City** drop-down menu, select the city, or the city closest to your location in the same time zone.

d. Toggle the **Network Time** switch to enable or disable network time synchronization using the Network Time Protocol (NTP).

NOTE

Enabling the Network Time switch keeps your system time correct as long as the system can access the internet. By default, one NTP pool is configured; you can add a new option, or disable or remove the default options by clicking the gear wheel button next to the **Network Time** switch.

e. Click **Done** to apply the changes and return to *Graphical installations*.

NOTE

If you disable network time synchronization, the controls at the bottom of the window become active, allowing you to set the time and date manually.

### 4.3. CONFIGURING SYSTEM OPTIONS

This section contains information about configuring Installation Destination, KDUMP, Network and Host Name, and Security Policy.

#### 4.3.1. Configuring installation destination

Use the **Installation Destination** window to configure the storage options, for example, the disks that you want to use as the installation target for your Red Hat Enterprise Linux installation. You must select at least one disk.
WARNING

Back up your data if you plan to use a disk that already contains data. For example, if you want to shrink an existing Microsoft Windows partition and install Red Hat Enterprise Linux as a second system, or if you are upgrading a previous release of Red Hat Enterprise Linux. Manipulating partitions always carries a risk. For example, if the process is interrupted or fails for any reason data on the disk can be lost.

IMPORTANT

Special cases

- Some BIOS types do not support booting from a RAID card. In these instances, the /boot partition must be created on a partition outside of the RAID array, such as on a separate hard drive. It is necessary to use an internal hard drive for partition creation with problematic RAID cards. A /boot partition is also necessary for software RAID setups. If you choose to partition your system automatically, you should manually edit your /boot partition.

- To configure the Red Hat Enterprise Linux boot loader to chain load from a different boot loader, you must specify the boot drive manually by clicking the Full disk summary and bootloader link from the Installation Destination window.

- When you install Red Hat Enterprise Linux on a system with both multipath and non-multipath storage devices, the automatic partitioning layout in the installation program creates volume groups that contain a mix of multipath and non-multipath devices. This defeats the purpose of multipath storage. It is recommended that you select either multipath or non-multipath devices on the Installation Destination window. Alternatively, proceed to manual partitioning.

Prerequisite

The Installation Summary window is open.

Procedure

1. From the Installation Summary window, click Installation Destination. The Installation Destination window opens.
   a. From the Local Standard Disks section, select the storage device that you require; a white check mark indicates your selection. Disks without a white check mark are not used during the installation process; they are ignored if you choose automatic partitioning, and they are not available in manual partitioning.
NOTE

All locally available storage devices (SATA, IDE and SCSI hard drives, USB flash and external disks) are displayed under Local Standard Disks. Any storage devices connected after the installation program has started are not detected. If you use a removable drive to install Red Hat Enterprise Linux, your system is unusable if you remove the device.

b. Optional: Click the Refresh link in the lower right-hand side of the window if you want to configure additional local storage devices to connect new hard drives. The Rescan Disks dialog box opens.

NOTE

All storage changes that you make during the installation are lost when you click Rescan Disks.

i. Click Rescan Disks and wait until the scanning process completes.

ii. Click OK to return to the Installation Destination window. All detected disks including any new ones are displayed under the Local Standard Disks section.

2. Optional: To add a specialized storage device, click Add a disk.... The Storage Device Selection window opens and lists all storage devices that the installation program has access to.

3. Optional: Under Storage Configuration, select the Automatic radio button.

IMPORTANT

Automatic partitioning is the recommended method of partitioning your storage.

You can also configure custom partitioning, for more details see Configuring manual partitioning

4. Optional: To reclaim space from an existing partitioning layout, select the I would like to make additional space available check box. For example, if a disk you want to use already contains a different operating system and you want to make this system’s partitions smaller to allow more room for Red Hat Enterprise Linux.

5. Optional: Select Encrypt my data to encrypt all partitions except the ones needed to boot the system (such as /boot) using Linux Unified Key Setup (LUKS). Encrypting your hard drive is recommended.

   a. Click Done. The Disk Encryption Passphrase dialog box opens.

      i. Type your passphrase in the Passphrase and Confirm fields.

      ii. Click Save Passphrase to complete disk encryption.
WARNING

If you lose the LUKS passphrase, any encrypted partitions and their data is completely inaccessible. There is no way to recover a lost passphrase. However, if you perform a Kickstart installation, you can save encryption passphrases and create backup encryption passphrases during the installation. See the Performing an advanced RHEL installation document for information.

6. Optional: Click the Full disk summary and bootloader link in the lower left-hand side of the window to select which storage device contains the boot loader.
   For more information, see Boot loader installation.

   NOTE
   In most cases it is sufficient to leave the boot loader in the default location. Some configurations, for example, systems that require chain loading from another boot loader require the boot drive to be specified manually.

7. Click Done.
   a. If you selected automatic partitioning and I would like to make additional space available, or if there is not enough free space on your selected hard drives to install Red Hat Enterprise Linux, the Reclaim Disk Space dialog box opens when you click Done, and lists all configured disk devices and all partitions on those devices. The dialog box displays information about how much space the system needs for a minimal installation and how much space you have reclaimed.

   WARNING
   If you delete a partition, all data on that partition is lost. If you want to preserve your data, use the Shrink option, not the Delete option.

   b. Review the displayed list of available storage devices. The Reclaimable Space column shows how much space can be reclaimed from each entry.

   c. To reclaim space, select a disk or partition, and click either the Delete button to delete that partition, or all partitions on a selected disk, or click Shrink to use free space on a partition while preserving the existing data.

   NOTE
   Alternatively, you can click Delete all, this deletes all existing partitions on all disks and makes this space available to Red Hat Enterprise Linux. Existing data on all disks is lost.
d. Click **Reclaim space** to apply the changes and return to **Graphical installations.**

**IMPORTANT**

No disk changes are made until you click **Begin Installation** on the **Installation Summary** window. The **Reclaim Space** dialog only marks partitions for resizing or deletion; no action is performed.

Additional resources

- How to use dm-crypt on IBM Z, LinuxONE and with the PAES cipher

### 4.3.2. Configuring boot loader

Red Hat Enterprise Linux uses GRand Unified Bootloader version 2 (**GRUB2**) as the boot loader for AMD64 and Intel 64, IBM Power Systems, and ARM. For 64-bit IBM Z, the **zipl** boot loader is used.

The boot loader is the first program that runs when the system starts and is responsible for loading and transferring control to an operating system. **GRUB2** can boot any compatible operating system (including Microsoft Windows) and can also use chain loading to transfer control to other boot loaders for unsupported operating systems.

**WARNING**

Installing **GRUB2** may overwrite your existing boot loader.

If an operating system is already installed, the Red Hat Enterprise Linux installation program attempts to automatically detect and configure the boot loader to start the other operating system. If the boot loader is not detected, you can manually configure any additional operating systems after you finish the installation.

If you are installing a Red Hat Enterprise Linux system with more than one disk, you might want to manually specify the disk where you want to install the boot loader.

**Procedure**

1. From the **Installation Destination** window, click the **Full disk summary and bootloader** link. The **Selected Disks** dialog box opens. The boot loader is installed on the device of your choice, or on a UEFI system; the **EFI system partition** is created on the target device during guided partitioning.

2. To change the boot device, select a device from the list and click **Set as Boot Device**. You can set only one device as the boot device.

3. To disable a new boot loader installation, select the device currently marked for boot and click **Do not install boot loader**. This ensures **GRUB2** is not installed on any device.
WARNING
If you choose not to install a boot loader, you cannot boot the system directly and you must use another boot method, such as a standalone commercial boot loader application. Use this option only if you have another way to boot your system.

The boot loader may also require a special partition to be created, depending on if your system uses BIOS or UEFI firmware, or if the boot drive has a GUID Partition Table (GPT) or a Master Boot Record (MBR, also known as msdos) label. If you use automatic partitioning, the installation program creates the partition.

4.3.3. Configuring Kdump

Kdump is a kernel crash-dumping mechanism. In the event of a system crash, Kdump captures the contents of the system memory at the moment of failure. This captured memory can be analyzed to find the cause of the crash. If Kdump is enabled, it must have a small portion of the system’s memory (RAM) reserved to itself. This reserved memory is not accessible to the main kernel.

Procedure

1. From the Installation Summary window, click Kdump. The Kdump window opens.
2. Select the Enable kdump check box.
3. Select either the Automatic or Manual memory reservation setting.
   a. If you select Manual, enter the amount of memory (in megabytes) that you want to reserve in the Memory to be reserved field using the + and - buttons. The Usable System Memory readout below the reservation input field shows how much memory is accessible to your main system after reserving the amount of RAM that you select.
4. Click Done to apply the settings and return to Graphical installations.

NOTE
The amount of memory that you reserve is determined by your system architecture (AMD64 and Intel 64 have different requirements than IBM Power) as well as the total amount of system memory. In most cases, automatic reservation is satisfactory.

IMPORTANT
Additional settings, such as the location where kernel crash dumps will be saved, can only be configured after the installation using either the system-config-kdump graphical interface, or manually in the /etc/kdump.conf configuration file.

4.3.4. Configuring network and host name options

Use the Network and Host name window to configure network interfaces. Options that you select here are available both during the installation for tasks such as downloading packages from a remote location, and on the installed system.
Follow the steps in this procedure to configure your network and host name.

**Procedure**

1. From the **Installation Summary** window, click **Network and Host Name**.

2. From the list in the left-hand pane, select an interface. The details are displayed in the right-hand pane.

   **NOTE**
   - There are several types of network device naming standards used to identify network devices with persistent names, for example, `em1` and `wl3sp0`. For information about these standards, see the *Configuring and managing networking* document.

3. Toggle the **ON/OFF** switch to enable or disable the selected interface.

   **NOTE**
   - The installation program automatically detects locally accessible interfaces, and you cannot add or remove them manually.

4. Click + to add a virtual network interface, which can be either: Team, Bond, Bridge, or VLAN.

5. Click - to remove a virtual interface.

6. Click **Configure** to change settings such as IP addresses, DNS servers, or routing configuration for an existing interface (both virtual and physical).

7. Type a host name for your system in the **Host Name** field.

   **NOTE**
   - The host name can be either a fully-qualified domain name (FQDN) in the format `hostname.domainname`, or a short host name with no domain name. Many networks have a Dynamic Host Configuration Protocol (DHCP) service that automatically supplies connected systems with a domain name. To allow the DHCP service to assign the domain name to this machine, specify only the short host name. The value `localhost` means that no specific static host name for the target system is configured, and the actual host name of the installed system is configured during the processing of the network configuration, for example, by `NetworkManager` using DHCP or DNS.
   - Host names can only contain alpha-numeric characters and `-` or `. `. Host names cannot start or end with `-` and `. `. 

8. Click **Apply** to apply the host name to the installer environment.

9. Alternatively, in the **Network and Hostname** window, you can choose the Wireless option. Click **Select network** in the right-hand pane to select your wifi connection, enter the password if required, and click **Done**.
4.3.4.1. Adding a virtual network interface

This procedure describes how to add a virtual network interface.

Procedure

1. From the **Network & Host name** window, click the + button to add a virtual network interface. The **Add a device** dialog opens.

2. Select one of the four available types of virtual interfaces:

   - **Bond**: NIC (Network Interface Controller) Bonding, a method to bind multiple physical network interfaces together into a single bonded channel.
   - **Bridge**: Represents NIC Bridging, a method to connect multiple separate networks into one aggregate network.
   - **Team**: NIC Teaming, a new implementation to aggregate links, designed to provide a small kernel driver to implement the fast handling of packet flows, and various applications to do everything else in user space.
   - **Vlan (Virtual LAN)**: A method to create multiple distinct broadcast domains which are mutually isolated.

3. Select the interface type and click **Add**. An editing interface dialog box opens, allowing you to edit any available settings for your chosen interface type. For more information see **Editing network interface**.

4. Click **Save** to confirm the virtual interface settings and return to the **Network & Host name** window.

   **NOTE**

   If you need to change the settings of a virtual interface, select the interface and click **Configure**.

4.3.4.2. Editing network interface configuration

This section contains information about the most important settings for a typical wired connection used during installation. Configuration of other types of networks is broadly similar, although the specific configuration parameters might be different.

   **NOTE**

   On 64-bit IBM Z, you cannot add a new connection as the network subchannels need to be grouped and set online beforehand, and this is currently done only in the booting phase.

Procedure

1. To configure a network connection manually, select the interface from the **Network and Host name** window and click **Configure**.

   An editing dialog specific to the selected interface opens.
4.3.4.3. Enabling or Disabling the Interface Connection

Follow the steps in this procedure to enable or disable an interface connection.

Procedure

1. Click the General tab.

2. Select the Connect automatically with priority check box to enable connection by default. Keep the default priority setting at 0.

   IMPORTANT
   - When enabled on a wired connection, the system automatically connects during startup or reboot. On a wireless connection, the interface attempts to connect to any known wireless networks in range. For further information about NetworkManager, including the `nm-connection-editor` tool, see the Configuring and managing networking document.
   - You can enable or disable all users on the system from connecting to this network using the All users may connect to this network option. If you disable this option, only root will be able to connect to this network.
   - It is not possible to only allow a specific user other than root to use this interface, as no other users are created at this point during the installation. If you need a connection for a different user, you must configure it after the installation.

3. Click Save to apply the changes and return to the Network and Host name window.

4.3.4.4. Setting up Static IPv4 or IPv6 Settings

By default, both IPv4 and IPv6 are set to automatic configuration depending on current network settings. This means that addresses such as the local IP address, DNS address, and other settings are detected automatically when the interface connects to a network. In many cases, this is sufficient, but you can also provide static configuration in the IPv4 Settings and IPv6 Settings tabs. Complete the following steps to configure IPv4 or IPv6 settings:

Procedure

1. To set static network configuration, navigate to one of the IPv Settings tabs and from the Method drop-down menu, select a method other than Automatic, for example, Manual. The Addresses pane is enabled.
NOTE

In the IPv6 Settings tab, you can also set the method to Ignore to disable IPv6 on this interface.

2. Click Add and enter your address settings.

3. Type the IP addresses in the Additional DNS servers field; it accepts one or more IP addresses of DNS servers, for example, 10.0.0.1,10.0.0.8.

4. Select the Require IPvX addressing for this connection to complete check box.

NOTE

Select this option in the IPv4 Settings or IPv6 Settings tabs to allow this connection only if IPv4 or IPv6 was successful. If this option remains disabled for both IPv4 and IPv6, the interface is able to connect if configuration succeeds on either IP protocol.

5. Click Save to apply the changes and return to the Network & Host name window.

4.3.4.5. Configuring Routes

Complete the following steps to configure routes.

Procedure

1. In the IPv4 Settings and IPv6 Settings tabs, click Routes to configure routing settings for a specific IP protocol on an interface. An editing routes dialog specific to the interface opens.

2. Click Add to add a route.

3. Select the Ignore automatically obtained routes check box to configure at least one static route and to disable all routes not specifically configured.

4. Select the Use this connection only for resources on its network check box to prevent the connection from becoming the default route.

NOTE

This option can be selected even if you did not configure any static routes. This route is used only to access certain resources, such as intranet pages that require a local or VPN connection. Another (default) route is used for publicly available resources. Unlike the additional routes configured, this setting is transferred to the installed system. This option is useful only when you configure more than one interface.

5. Click OK to save your settings and return to the editing routes dialog that is specific to the interface.

6. Click Save to apply the settings and return to the Network and Host Name window.

4.3.4.6. Additional resources
4.3.5. Configuring Connect to Red Hat

The Red Hat Content Delivery Network (CDN), available from cdn.redhat.com, is a geographically distributed series of static web servers that contain content and errata that is consumed by systems. The content can be consumed directly, such as using a system registered to Red Hat Subscription Management. The CDN is protected by x.509 certificate authentication to ensure that only valid users have access. When a system is registered to Red Hat Subscription Management, the attached subscriptions govern which subset of the CDN the system can access.

Registering and installing RHEL from the CDN provides the following benefits:

- The CDN installation method supports the Boot ISO and the DVD ISO image files. However, the use of the smaller Boot ISO image file is recommended as it consumes less space than the larger DVD ISO image file.
- The CDN uses the latest packages resulting in a fully up-to-date system right after installation. There is no requirement to install package updates immediately after installation as is often the case when using the DVD ISO image file.
- Integrated support for connecting to Red Hat Insights and enabling System Purpose.

4.3.5.1. Introduction to System Purpose

System Purpose is an optional but recommended feature of the Red Hat Enterprise Linux installation. You use System Purpose to record the intended use of a Red Hat Enterprise Linux 8 system, and ensure that the entitlement server auto-attaches the most appropriate subscription to your system.

Benefits include:

- In-depth system-level information for system administrators and business operations.
- Reduced overhead when determining why a system was procured and its intended purpose.
- Improved customer experience of Subscription Manager auto-attach as well as automated discovery and reconciliation of system usage.

You can enter System Purpose data in one of the following ways:

- During image creation
- During a GUI installation when using the **Connect to Red Hat** screen to register your system and attach your Red Hat subscription
- During a Kickstart installation when using Kickstart automation scripts
- After installation using the **syspurpose** command-line (CLI) tool

To record the intended purpose of your system, you can configure the following components of System Purpose. The selected values are used by the entitlement server upon registration to attach the most suitable subscription for your system.

- **Role**
  - Red Hat Enterprise Linux Server
- Red Hat Enterprise Linux Workstation
- Red Hat Enterprise Linux Compute Node

**Service Level Agreement**
- Premium
- Standard
- Self-Support

**Usage**
- Production
- Development/Test
- Disaster Recovery

**Additional resources**
- Composing a customized RHEL system image
- Performing an advanced RHEL installation
- Using and Configuring Red Hat Subscription Manager

### 4.3.5.2. Configuring Connect to Red Hat options

Use the following procedure to configure the Connect to Red Hat options in the GUI.

**NOTE**

You can register to the CDN using either your Red Hat account or your activation key details.

**Procedure**

1. Click **Account**.
   - a. Enter your Red Hat Customer Portal username and password details.

2. Optional: Click **Activation Key**.
   - a. Enter your organization ID and activation key. You can enter more than one activation key, separated by a comma, as long as the activation keys are registered to your subscription.

3. Select the **Set System Purpose** check box. System Purpose enables the entitlement server to determine and automatically attach the most appropriate subscription to satisfy the intended use of Red Hat Enterprise Linux 8 system.
   - a. Select the required **Role**, **SLA**, and **Usage** from the corresponding drop-down lists.

4. The **Connect to Red Hat Insights** check box is enabled by default. Clear the check box if you do not want to connect to Red Hat Insights.
NOTE
Red Hat Insights is a Software-as-a-Service (SaaS) offering that provides continuous, in-depth analysis of registered Red Hat-based systems to proactively identify threats to security, performance and stability across physical, virtual and cloud environments, and container deployments.

5. Optional: Expand Options.
   a. Select the Use HTTP proxy check box if your network environment only allows external Internet access or access to content servers through an HTTP proxy. Clear the Use HTTP proxy check box if an HTTP proxy is not used.
   b. If you are running Satellite Server or performing internal testing, select the Custom Server URL and Custom base URL check boxes and enter the required details.

   IMPORTANT
   - The Custom Server URL field does not require the HTTP protocol, for example nameofhost.com. However, the Custom base URL field requires the HTTP protocol.
   - To change the Custom base URL after registration, you must unregister, provide the new details, and then re-register.

6. Click Register to register the system. When the system is successfully registered and subscriptions are attached, the Connect to Red Hat window displays the attached subscription details.

   NOTE
   Depending on the amount of subscriptions, the registration and attachment process might take up to a minute to complete.

7. Click Done to return to the Installation Summary window.
   a. A Registered message is displayed under Connect to Red Hat

4.3.5.3. Installation source repository after system registration

The installation source repository used after system registration is dependent on how the system was booted.

System booted from the Boot ISO or the DVD ISO image file
If you booted the RHEL installation using either the Boot ISO or the DVD ISO image file with the default boot parameters, the installation program automatically switches the installation source repository to the CDN after registration.

System booted with the inst.repo=<URL> boot parameter
If you booted the RHEL installation with the inst.repo=<URL> boot parameter, the installation program does not automatically switch the installation source repository to the CDN after registration. If you want to use the CDN to install RHEL, you must manually switch the installation source repository to the CDN by selecting the Red Hat CDN option in the Installation Source window of the graphical installation. If you do not manually switch to the CDN, the installation program installs the packages from the repository specified on the kernel command line.
IMPORTANT

- You can switch the installation source repository to the CDN using the `rhsm` Kickstart command only if you do not specify an installation source using `inst.repo=` on the kernel command line or the `url` command in the Kickstart file. You must use `inst.stage2=<URL>` on the kernel command line to fetch the installation image, but not specify the installation source.

- An installation source URL specified using a boot option or included in a Kickstart file takes precedence over the CDN, even if the Kickstart file contains the `rhsm` command with valid credentials. The system is registered, but it is installed from the URL installation source. This ensures that earlier installation processes operate as normal.

4.3.5.4. Verifying your system registration from the CDN

Use this procedure to verify that your system is registered to the CDN using the GUI.

WARNING

You can only verify your registration from the CDN if you have not clicked the Begin Installation button from the Installation Summary window. Once the Begin Installation button is clicked, you cannot return to the Installation Summary window to verify your registration.

Prerequisite

- You have completed the registration process as documented in the Register and install from CDN using GUI and Registered is displayed under Connect to Red Hat on the Installation Summary window.

Procedure

1. From the Installation Summary window, select Connect to Red Hat

2. The window opens and displays a registration summary:
   
   **Method**
   
   The registered account name or activation keys are displayed.

   **System Purpose**
   
   If set, the role, SLA, and usage details are displayed.

   **Insights**
   
   If enabled, the Insights details are displayed.

   **Number of subscriptions**
   
   The number of subscriptions attached are displayed. Note: In the simple content access mode, no subscription being listed is a valid behavior.

3. Verify that the registration summary matches the details that were entered.
4.3.5.5. Unregistering your system from the CDN

Use this procedure to unregister your system from the CDN using the GUI.

**WARNING**
- You can unregister from the CDN if you have not clicked the Begin Installation button from the Installation Summary window. Once the Begin Installation button is clicked, you cannot return to the Installation Summary window to unregister your registration.
- When unregistering, the installation program switches to the first available repository, in the following order:
  a. The URL used in the inst.repo=<URL> boot parameter on the kernel command line.
  b. An automatically detected repository on the installation media (USB or DVD).

**Prerequisite**
- You have completed the registration process as documented in the Registering and installing RHEL from the CDN and Registered is displayed under Connect to Red Hat on the Installation Summary window.

**Procedure**
1. From the Installation Summary window, select Connect to Red Hat.
2. The Connect to Red Hat window opens and displays a registration summary:
   - **Method**
     The registered account name or activation keys used are displayed.
   - **System Purpose**
     If set, the role, SLA, and usage details are displayed.
   - **Insights**
     If enabled, the Insights details are displayed.
   - **Number of subscriptions**
     The number of subscriptions attached are displayed. Note: In the simple content access mode, no subscription being listed is a valid behavior.
3. Click Unregister to remove the registration from the CDN. The original registration details are displayed with a Not registered message displayed in the lower-middle part of the window.
4. Click **Done** to return to the **Installation Summary** window.

5. **Connect to Red Hat** displays a *Not registered* message, and **Software Selection** displays a *Red Hat CDN requires registration* message.

   **NOTE**

   After unregistering, it is possible to register your system again. Click **Connect to Red Hat**. The previously entered details are populated. Edit the original details, or update the fields based on the account, purpose, and connection. Click **Register** to complete.

### 4.3.5.6. Additional resources

- For information about Red Hat Insights, see the [Red Hat Insights product documentation](#).
- For information about Activation Keys, see the [Understanding Activation Keys](#) chapter of the [Using Red Hat Subscription Management](#) document.
- For information about how to set up an HTTP proxy for Subscription Manager, see the [Using an HTTP proxy](#) chapter of the [Using and Configuring Red Hat Subscription Manager](#) document.

### 4.3.6. Installing System Aligned with a Security Policy

This section contains information about applying Red Hat Enterprise Linux 8 security policy during installation and how to configure it for use on your system before the first boot.

#### 4.3.6.1. About security policy

The Red Hat Enterprise Linux includes OpenSCAP suite to enable automated configuration of the system in alignment with a particular security policy. The policy is implemented using the Security Content Automation Protocol (SCAP) standard. The packages are available in the AppStream repository. However, by default, the installation and post-installation process does not enforce any policies and therefore does not involve any checks unless specifically configured.

Applying a security policy is not a mandatory feature of the installation program. If you apply a security policy to the system, it is installed using restrictions and recommendations defined in the profile that you selected. The *openscap-scanner* and *scap-security-guide* packages are added to your package selection, providing a preinstalled tool for compliance and vulnerability scanning.

When you select a security policy, the Anaconda GUI installer requires the configuration to adhere to the policy’s requirements. There might be conflicting package selections, as well as separate partitions defined. Only after all the requirements are met, you can start the installation.

At the end of the installation process, the selected OPenSCAP security policy automatically hardens the system and scans it to verify compliance, saving the scan results to the `/root/openscap_data` directory on the installed system.

   **NOTE**

   By default, the installer uses the content of the *scap-security-guide* package bundled in the installation image. You can also load external content from an HTTP, HTTPS, or FTP server.

#### 4.3.6.2. Configuring a security policy
Complete the following steps to configure a security policy.

**Prerequisite**

The **Installation Summary** window is open.

**Procedure**

1. From the **Installation Summary** window, click **Security Policy**. The **Security Policy** window opens.

2. To enable security policies on the system, toggle the **Apply security policy** switch to **ON**.

3. Select one of the profiles listed in the top pane.

4. Click **Select profile**. Profile changes that you must apply before installation appear in the bottom pane.

5. Click **Change content** to use a custom profile. A separate window opens allowing you to enter a URL for valid security content.
   a. Click **Fetch** to retrieve the URL.
   b. Click **Use SCAP Security Guide** to return to the **Security Policy** window.

   **NOTE**
   
   You can load custom profiles from an **HTTP**, **HTTPS**, or **FTP** server. Use the full address of the content including the protocol, such as **http://**. A network connection must be active before you can load a custom profile. The installation program detects the content type automatically.

6. Click **Done** to apply the settings and return to the **Installation Summary** window.

**4.3.6.3. Additional resources**

- **scap-security-guide(8)** - The manual page for the **scap-security-guide** project contains information about SCAP security profiles, including examples on how to utilize the provided benchmarks using the OpenSCAP utility.

- Red Hat Enterprise Linux security compliance information is available in the **Security hardening** document.

**4.4. CONFIGURING SOFTWARE SETTINGS**

This section contains information about configuring your installation source and software selection settings, and activating a repository.

**4.4.1. Configuring installation source**

Complete the steps in this procedure to configure an installation source from either auto-detected installation media, Red Hat CDN, or the network.
NOTE

When the Installation Summary window first opens, the installation program attempts to configure an installation source based on the type of media that was used to boot the system. The full Red Hat Enterprise Linux Server DVD configures the source as local media.

Prerequisites

- You have downloaded the full installation image. Downloading a RHEL installation ISO image
- You have created a bootable physical media. Creating a bootable CD or DVD
- The Installation Summary window is open.

Procedure

1. From the Installation Summary window, click Installation Source. The Installation Source window opens.
   a. Review the Auto-detected installation media section to verify the details. This option is selected by default if you started the installation program from media containing an installation source, for example, a DVD.
   b. Click Verify to check the media integrity.
   c. Review the Additional repositories section and note that the AppStream checkbox is selected by default.

   IMPORTANT

   - No additional configuration is necessary as the BaseOS and AppStream repositories are installed as part of the full installation image.
   - Do not disable the AppStream repository check box if you want a full Red Hat Enterprise Linux8 installation.

2. Optional: Select the Red Hat CDN option to register your system, attach RHEL subscriptions, and install RHEL from the Red Hat Content Delivery Network (CDN). For more information, see the Registering and installing RHEL from the CDN section.

3. Optional: Select the On the network option to download and install packages from a network location instead of local media.

NOTE

- If you do not want to download and install additional repositories from a network location, proceed to Configuring software selection.
- This option is available only when a network connection is active. See Configuring network and host name options for information about how to configure network connections in the GUI.
a. Select the **On the network** drop-down menu to specify the protocol for downloading packages. This setting depends on the server that you want to use.

b. Type the server address (without the protocol) into the address field. If you choose NFS, a second input field opens where you can specify custom **NFS mount options**. This field accepts options listed in the `nfs(5)` man page.

### IMPORTANT

When selecting an NFS installation source, you must specify the address with a colon (:) character separating the host name from the path. For example:

```
server.example.com:/path/to/directory
```

### NOTE

The following steps are optional and are only required if you use a proxy for network access.

c. Click **Proxy setup...** to configure a proxy for an HTTP or HTTPS source.

d. Select the **Enable HTTP proxy** check box and type the URL into the **Proxy Host** field.

e. Select the **Use Authentication** check box if the proxy server requires authentication.

f. Type in your user name and password.

g. Click **OK** to finish the configuration and exit the **Proxy Setup...** dialog box.

### NOTE

If your HTTP or HTTPS URL refers to a repository mirror, select the required option from the **URL type** drop-down list. All environments and additional software packages are available for selection when you finish configuring the sources.

4. Click **+** to add a repository.

5. Click **-** to delete a repository.

6. Click the **arrow** icon to revert the current entries to the setting when you opened the **Installation Source** window.

7. To activate or deactivate a repository, click the check box in the **Enabled** column for each entry in the list.

### NOTE

You can name and configure your additional repository in the same way as the primary repository on the network.

8. Click **Done** to apply the settings and return to the **Installation Summary** window.
4.4.2. Configuring software selection

Use the Software Selection window to select the software packages that you require. The packages are organized by Base Environment and Additional Software.

- **Base Environment** contains predefined packages. You can select only one base environment, and availability is dependent on the installation ISO image that is used as the installation source.

- **Additional Software for Selected Environment** contains additional software packages for the base environment. You can select multiple software packages.

Use a predefined environment and additional software to customize your system. However, in a standard installation, you cannot select individual packages to install. To view the packages contained in a specific environment, see the `repository/repoDATA/*/comps-repository.architecture.xml` file on your installation source media (DVD, CD, USB). The XML file contains details of the packages installed as part of a base environment. Available environments are marked by the `<environment>` tag, and additional software packages are marked by the `<group>` tag.

If you are unsure about which packages to install, Red Hat recommends that you select the Minimal Install base environment. Minimal install installs a basic version of Red Hat Enterprise Linux with only a minimal amount of additional software. After the system finishes installing and you log in for the first time, you can use the YUM package manager to install additional software. For more information about YUM package manager, see the Configuring basic system settings document.

NOTE

- The `yum group list` command lists all package groups from yum repositories. See the Configuring basic system settings document for more information.

- If you need to control which packages are installed, you can use a Kickstart file and define the packages in the `%packages` section. See the Performing an advanced RHEL installation document for information about installing Red Hat Enterprise Linux using Kickstart.

Prerequisites

- You have configured the installation source.

- The installation program downloaded package metadata.

- The Installation Summary window is open.

Procedure

1. From the Installation Summary window, click Software Selection. The Software Selection window opens.

2. From the Base Environment pane, select a base environment. You can select only one base environment.

NOTE

The Server with GUI base environment is the default base environment and it launches the Initial Setup application after the installation completes and you restart the system.
3. From the Additional Software for Selected Environment pane, select one or more options.

4. Click Done to apply the settings and return to Graphical installations.

4.5. CONFIGURING STORAGE DEVICES

You can install Red Hat Enterprise Linux on a large variety of storage devices. You can configure basic, locally accessible, storage devices in the Installation Destination window. Basic storage devices directly connected to the local system, such as hard disk drives and solid-state drives, are displayed in the Local Standard Disks section of the window. On 64-bit IBM Z, this section contains activated Direct Access Storage Devices (DASDs).

**WARNING**

A known issue prevents DASDs configured as HyperPAV aliases from being automatically attached to the system after the installation is complete. These storage devices are available during the installation, but are not immediately accessible after you finish installing and reboot. To attach HyperPAV alias devices, add them manually to the `/etc/dasd.conf` configuration file of the system.

4.5.1. Storage device selection

The storage device selection window lists all storage devices that the installation program can access. Depending on your system and available hardware, some tabs might not be displayed. The devices are grouped under the following tabs:

**Multipath Devices**

Storage devices accessible through more than one path, such as through multiple SCSI controllers or Fiber Channel ports on the same system.

**IMPORTANT**

The installation program only detects multipath storage devices with serial numbers that are 16 or 32 characters long.

**Other SAN Devices**

Devices available on a Storage Area Network (SAN).

**Firmware RAID**

Storage devices attached to a firmware RAID controller.

**NVDIMM Devices**

Under specific circumstances, Red Hat Enterprise Linux 8 can boot and run from (NVDIMM) devices in sector mode on the Intel 64 and AMD64 architectures.

**System z Devices**

Storage devices, or Logical Units (LUNs), attached through the zSeries Linux FCP (Fiber Channel Protocol) driver.
4.5.2. Filtering storage devices

In the storage device selection window you can filter storage devices either by their World Wide Identifier (WWID) or by the port, target, or logical unit number (LUN).

**Prerequisite**

The Installation Summary window is open.

**Procedure**

1. From the Installation Summary window, click Installation Destination. The Installation Destination window opens, listing all available drives.

2. Under the Specialized & Network Disks section, click Add a disk…. The storage devices selection window opens.

3. Click the Search by tab to search by port, target, LUN, or WWID. Searching by WWID or LUN requires additional values in the corresponding input text fields.

4. Select the option that you require from the Search drop-down menu.

5. Click Find to start the search. Each device is presented on a separate row with a corresponding check box.

6. Select the check box to enable the device that you require during the installation process. Later in the installation process you can choose to install Red Hat Enterprise Linux on any of the selected devices, and you can choose to mount any of the other selected devices as part of the installed system automatically.

   **NOTE**

   - Selected devices are not automatically erased by the installation process and selecting a device does not put the data stored on the device at risk.
   - You can add devices to the system after installation by modifying the /etc/fstab file.

7. Click Done to return to the Installation Destination window.

   **IMPORTANT**

   Any storage devices that you do not select are hidden from the installation program entirely. To chain load the boot loader from a different boot loader, select all the devices present.

4.5.3. Using advanced storage options

To use an advanced storage device, you can configure an iSCSI (SCSI over TCP/IP) target or FCoE (Fibre Channel over Ethernet) SAN (Storage Area Network).

To use iSCSI storage devices for the installation, the installation program must be able to discover them as iSCSI targets and be able to create an iSCSI session to access them. Each of these steps might require a user name and password for Challenge Handshake Authentication Protocol (CHAP) authentication. Additionally, you can configure an iSCSI target to authenticate the iSCSI initiator on the
system to which the target is attached (reverse CHAP), both for discovery and for the session. Used together, CHAP and reverse CHAP are called mutual CHAP or two-way CHAP. Mutual CHAP provides the greatest level of security for iSCSI connections, particularly if the user name and password are different for CHAP authentication and reverse CHAP authentication.

**NOTE**

Repeat the iSCSI discovery and iSCSI login steps to add all required iSCSI storage. You cannot change the name of the iSCSI initiator after you attempt discovery for the first time. To change the iSCSI initiator name, you must restart the installation.

# 4.5.3.1. Discovering and starting an iSCSI session

Complete the following steps to discover and start an iSCSI session.

**Prerequisites**

- The Installation Summary window is open.

**Procedure**

1. From the Installation Summary window, click Installation Destination. The Installation Destination window opens, listing all available drives.

2. Under the Specialized & Network Disks section, click Add a disk…. The storage devices selection window opens.

3. Click Add iSCSI target…. The Add iSCSI Storage Target window opens.

**IMPORTANT**

You cannot place the /boot partition on iSCSI targets that you have manually added using this method - an iSCSI target containing a /boot partition must be configured for use with iBFT. However, in instances where the installed system is expected to boot from iSCSI with iBFT configuration provided by a method other than firmware iBFT, for example using iPXE, you can remove the /boot partition restriction using the inst.nonibftiscsiboot installer boot option.

4. Enter the IP address of the iSCSI target in the Target IP Address field.

5. Type a name in the iSCSI Initiator Name field for the iSCSI initiator in iSCSI qualified name (IQN) format. A valid IQN entry contains the following information:

   - The string iqn. (note the period).

   - A date code that specifies the year and month in which your organization’s Internet domain or subdomain name was registered, represented as four digits for the year, a dash, and two digits for the month, followed by a period. For example, represent September 2010 as 2010-09.

   - Your organization’s Internet domain or subdomain name, presented in reverse order with the top-level domain first. For example, represent the subdomain storage.example.com as com.example.storage.
A colon followed by a string that uniquely identifies this particular iSCSI initiator within your domain or subdomain. For example, `:diskarrays-sn-a8675309`.

A complete IQN is as follows: `iqn.2010-09.storage.example.com:diskarrays-sn-a8675309`.

The installation program prepopulates the iSCSI Initiator Name field with a name in this format to help you with the structure. For more information about IQNs, see 3.2.6. *iSCSI Names in RFC 3720 - Internet Small Computer Systems Interface (iSCSI)* available from tools.ietf.org and 1. *iSCSI Names and Addresses in RFC 3721 - Internet Small Computer Systems Interface (iSCSI) Naming and Discovery* available from tools.ietf.org.

6. Select the Discovery Authentication Type drop-down menu to specify the type of authentication to use for iSCSI discovery. The following options are available:
   - No credentials
   - CHAP pair
   - CHAP pair and a reverse pair

7. a. If you selected CHAP pair as the authentication type, enter the user name and password for the iSCSI target in the CHAP Username and CHAP Password fields.
   b. If you selected CHAP pair and a reverse pair as the authentication type, enter the user name and password for the iSCSI target in the CHAP Username and CHAP Password field, and the user name and password for the iSCSI initiator in the Reverse CHAP Username and Reverse CHAP Password fields.

8. Optionally, select the Bind targets to network interfaces check box.

9. Click Start Discovery.
   The installation program attempts to discover an iSCSI target based on the information provided. If discovery succeeds, the Add iSCSI Storage Target window displays a list of all iSCSI nodes discovered on the target.

10. Select the check boxes for the node that you want to use for installation.

    **NOTE**
    The Node login authentication type menu contains the same options as the Discovery Authentication Type menu. However, if you need credentials for discovery authentication, use the same credentials to log in to a discovered node.

11. Click the additional Use the credentials from discovery drop-down menu. When you provide the proper credentials, the Log In button becomes available.

12. Click Log In to initiate an iSCSI session.

### 4.5.3.2. Configuring FCoE parameters

Complete the following steps to configure FCoE parameters.

**Prerequisite**

The Installation Summary window is open.

**Procedure**
1. From the Installation Summary window, click Installation Destination. The Installation Destination window opens, listing all available drives.

2. Under the Specialized & Network Disks section, click Add a disk.... The storage devices selection window opens.

3. Click Add FCoE SAN.... A dialog box opens for you to configure network interfaces for discovering FCoE storage devices.

4. Select a network interface that is connected to an FCoE switch in the NIC drop-down menu.

5. Click Add FCoE disk(s) to scan the network for SAN devices.

6. Select the required check boxes:

   - **Use DCB:** Data Center Bridging (DCB) is a set of enhancements to the Ethernet protocols designed to increase the efficiency of Ethernet connections in storage networks and clusters. Select the check box to enable or disable the installation program’s awareness of DCB. Enable this option only for network interfaces that require a host-based DCBX client. For configurations on interfaces that use a hardware DCBX client, disable the check box.

   - **Use auto vlan:** Auto VLAN is enabled by default and indicates whether VLAN discovery should be performed. If this check box is enabled, then the FIP (FCoE Initiation Protocol) VLAN discovery protocol runs on the Ethernet interface when the link configuration has been validated. If they are not already configured, network interfaces for any discovered FCoE VLANs are automatically created and FCoE instances are created on the VLAN interfaces.

7. Discovered FCoE devices are displayed under the Other SAN Devices tab in the Installation Destination window.

4.5.3.3. Configuring DASD storage devices

Complete the following steps to configure DASD storage devices.

**Prerequisite**

The Installation Summary window is open.

**Procedure**

1. From the Installation Summary window, click Installation Destination. The Installation Destination window opens, listing all available drives.

2. Under the Specialized & Network Disks section, click Add a disk.... The storage devices selection window opens.

3. Click Add DASD. The Add DASD Storage Target dialog box opens and prompts you to specify a device number, such as 0.0.0204, and attach additional DASDs that were not detected when the installation started.

4. Type the device number of the DASD that you want to attach in the Device number field.

5. Click Start Discovery.
NOTE

- If a DASD with the specified device number is found and if it is not already attached, the dialog box closes and the newly-discovered drives appear in the list of drives. You can then select the check boxes for the required devices and click **Done**. The new DASDs are available for selection, marked as **DASD device 0.0.xxxx** in the **Local Standard Disks** section of the **Installation Destination** window.

- If you entered an invalid device number, or if the DASD with the specified device number is already attached to the system, an error message appears in the dialog box, explaining the error and prompting you to try again with a different device number.

**4.5.3.4. Configuring FCP devices**

FCP devices enable 64-bit IBM Z to use SCSI devices rather than, or in addition to, Direct Access Storage Device (DASD) devices. FCP devices provide a switched fabric topology that enables 64-bit IBM Z systems to use SCSI LUNs as disk devices in addition to traditional DASD devices.

**Prerequisites**

- The **Installation Summary** window is open.

- For an FCP-only installation, remove the **DASD=** option from the CMS configuration file or the **rd.dasd=** option from the parameter file to indicate that no DASD is present.

**Procedure**

1. From the **Installation Summary** window, click **Installation Destination**. The **Installation Destination** window opens, listing all available drives.

2. Under the **Specialized & Network Disks** section, click **Add a disk…**. The storage devices selection window opens.

3. Click **Add ZFCP LUN**. The **Add zFCP Storage Target** dialog box opens allowing you to add a FCP (Fibre Channel Protocol) storage device. 64-bit IBM Z requires that you enter any FCP device manually so that the installation program can activate FCP LUNs. You can enter FCP devices either in the graphical installation, or as a unique parameter entry in the parameter or CMS configuration file. The values that you enter must be unique to each site that you configure.

4. Type the 4 digit hexadecimal device number in the **Device number** field.

5. Type the 16 digit hexadecimal World Wide Port Number (WWPN) in the **WWPN** field.

6. Type the 16 digit hexadecimal FCP LUN identifier in the **LUN** field.

7. Click **Start Discovery** to connect to the FCP device.

The newly-added devices are displayed in the **System z Devices** tab of the **Installation Destination** window.
NOTE

- Interactive creation of an FCP device is only possible in graphical mode. It is not possible to configure an FCP device interactively in text mode installation.
- Use only lower-case letters in hex values. If you enter an incorrect value and click Start Discovery, the installation program displays a warning. You can edit the configuration information and retry the discovery attempt.
- For more information about these values, consult the hardware documentation and check with your system administrator.

4.5.4. Installing to an NVDIMM device

Non-Volatile Dual In-line Memory Module (NVDIMM) devices combine the performance of RAM with disk-like data persistence when no power is supplied. Under specific circumstances, Red Hat Enterprise Linux 8 can boot and run from NVDIMM devices.

4.5.4.1. Criteria for using an NVDIMM device as an installation target

You can install Red Hat Enterprise Linux 8 to Non-Volatile Dual In-line Memory Module (NVDIMM) devices in sector mode on the Intel 64 and AMD64 architectures, supported by the nd_pmem driver.

Conditions for using an NVDIMM device as storage

To use an NVDIMM device as storage, the following conditions must be satisfied:

- The architecture of the system is Intel 64 or AMD64.
- The NVDIMM device is configured to sector mode. The installation program can reconfigure NVDIMM devices to this mode.
- The NVDIMM device must be supported by the nd_pmem driver.

Conditions for booting from an NVDIMM Device

Booting from an NVDIMM device is possible under the following conditions:

- All conditions for using the NVDIMM device as storage are satisfied.
- The system uses UEFI.
- The NVDIMM device must be supported by firmware available on the system, or by an UEFI driver. The UEFI driver may be loaded from an option ROM of the device itself.
- The NVDIMM device must be made available under a namespace.

Utilize the high performance of NVDIMM devices during booting, place the /boot and /boot/efi directories on the device. The Execute-in-place (XIP) feature of NVDIMM devices is not supported during booting and the kernel is loaded into conventional memory.

4.5.4.2. Configuring an NVDIMM device using the graphical installation mode

A Non-Volatile Dual In-line Memory Module (NVDIMM) device must be properly configured for use by Red Hat Enterprise Linux 8 using the graphical installation.
**WARNING**

Reconfiguration of a NVDIMM device process destroys any data stored on the device.

---

**Prerequisites**

- A NVDIMM device is present on the system and satisfies all the other conditions for usage as an installation target.
- The installation has booted and the **Installation Summary** window is open.

**Procedure**

1. From the **Installation Summary** window, click **Installation Destination**. The **Installation Destination** window opens, listing all available drives.
2. Under the **Specialized & Network Disks** section, click **Add a disk...**. The storage devices selection window opens.
3. Click the **NVDIMM Devices** tab.
4. To reconfigure a device, select it from the list. If a device is not listed, it is not in sector mode.
5. Click **Reconfigure NVDIMM...**. A reconfiguration dialog opens.
6. Enter the sector size that you require and click **Start Reconfiguration**. The supported sector sizes are 512 and 4096 bytes.
7. When reconfiguration completes click **OK**.
8. Select the device check box.
9. Click **Done** to return to the **Installation Destination** window. The NVDIMM device that you reconfigured is displayed in the **Specialized & Network Disks** section.
10. Click **Done** to return to the **Installation Summary** window.

The NVDIMM device is now available for you to select as an installation target. Additionally, if the device meets the requirements for booting, you can set the device as a boot device.

### 4.6. CONFIGURING MANUAL PARTITIONING

You can use manual partitioning to configure your disk partitions and mount points and define the file system that Red Hat Enterprise Linux is installed on.
NOTE

Before installation, you should consider whether you want to use partitioned or unpartitioned disk devices. For more information, see the Knowledgebase article at https://access.redhat.com/solutions/163853.

An installation of Red Hat Enterprise Linux requires a minimum of one partition but Red Hat recommends using at least the following partitions or volumes: /, /home, /boot, and swap. You can also create additional partitions and volumes as you require.

WARNING

To prevent data loss it is recommended that you back up your data before proceeding. If you are upgrading or creating a dual-boot system, you should back up any data you want to keep on your storage devices.

4.6.1. Starting manual partitioning

Prerequisites

- The Installation Summary screen is currently displayed.
- All disks are available to the installation program.

Procedure

1. Select disks for installation:
   a. Click Installation Destination to open the Installation Destination window.
   b. Select the disks that you require for installation by clicking the corresponding icon. A selected disk has a check-mark displayed on it.
   c. Under Storage Configuration, select the Custom radio-button.
   d. Optional: To enable storage encryption with LUKS, select the Encrypt my data check box.
   e. Click Done.

2. If you selected to encrypt the storage, a dialog box for entering a disk encryption passphrase opens. Type in the LUKS passphrase:
   a. Enter the passphrase in the two text fields. To switch keyboard layout, use the keyboard icon.
b. Click **Save Passphrase**. The **Manual Partitioning** window opens.

3. Detected mount points are listed in the left-hand pane. The mount points are organized by detected operating system installations. As a result, some file systems may be displayed multiple times if a partition is shared among several installations.

   a. Select the mount points in the left pane; the options that can be customized are displayed in the right pane.

**NOTE**

- If your system contains existing file systems, ensure that enough space is available for the installation. To remove any partitions, select them in the list and click the • button.
  The dialog has a check box that you can use to remove all other partitions used by the system to which the deleted partition belongs.

- If there are no existing partitions and you want to create the recommended set of partitions as a starting point, select your preferred partitioning scheme from the left pane (default for Red Hat Enterprise Linux is LVM) and click the **Click here to create them automatically** link.
  A /boot partition, a / (root) volume, and a swap volume proportionate to the size of the available storage are created and listed in the left pane.
  These are the recommended file systems for a typical installation, but you can add additional file systems and mount points.

   b. Click **Done** to confirm any changes and return to the **Installation Summary** window.

4.6.2. Adding a mount point file system

Complete the following steps to add multiple mount point file systems.

**Prerequisites**

- Plan for your partitions:
  - To avoid problems with space allocation, first create small partitions with known fixed sizes, such as /boot, and then create the remaining partitions, letting the installation program allocate the remaining capacity to them.
  - If you want to install the system on multiple disks, or if your disks differ in size and a particular partition must be created on the first disk detected by BIOS, then create these partitions first.
Procedure

1. Click + to create a new mount point file system. The Add a New Mount Point dialog opens.

2. Select one of the preset paths from the Mount Point drop-down menu or type your own; for example, select / for the root partition or /boot for the boot partition.

3. Enter the size of the file system in to the Desired Capacity field; for example, 2GiB.

   **WARNING**

   If you do not specify a value in the Desired Capacity field, or if you specify a size bigger than available space, then all remaining free space is used.

4. Click Add mount point to create the partition and return to the Manual Partitioning window.

### 4.6.3. Configuring storage for a mount point file system

This procedure describes how to set the partitioning scheme for each mount point that was created manually. The available options are Standard Partition, LVM, and LVM Thin Provisioning.

**NOTE**

- Btrfs support has been removed in Red Hat Enterprise Linux 8.
- The /boot partition is always located on a standard partition, regardless of the value selected.

Procedure

1. To change the devices that a single non-LVM mount point should be located on, select the required mount point from the left-hand pane.

2. Under the Device(s) heading, click Modify…. The Configure Mount Point dialog opens.

3. Select one or more devices and click Select to confirm your selection and return to the Manual Partitioning window.

4. Click Update Settings to apply the changes.

5. In the lower left-hand side of the Manual Partitioning window, click the storage device selected link to open the Selected Disks dialog and review disk information.

**NOTE**

Click the Rescan button (circular arrow button) to refresh all local disks and partitions; this is only required after performing advanced partition configuration outside the installation program. Clicking the Rescan Disks button resets all configuration changes made in the installation program.
4.6.4. Customizing a mount point file system

You can customize a partition or volume if you want to set specific settings.

**IMPORTANT**

If /usr or /var is partitioned separately from the rest of the root volume, the boot process becomes much more complex as these directories contain critical components. In some situations, such as when these directories are placed on an iSCSI drive or an FCoE location, the system is unable to boot, or hangs with a *Device is busy* error when powering off or rebooting.

This limitation only applies to /usr or /var, not to directories below them. For example, a separate partition for /var/www works successfully.

**Procedure**

1. From the left pane, select the mount point.

![Figure 4.1. Customizing Partitions](image)

   - **Procedure**
     - **1.** From the left pane, select the mount point.
     - **2.** From the right-hand pane, you can customize the following options:
       - **a.** Enter the file system mount point into the **Mount Point** field. For example, if a file system is the root file system, enter `/`; enter `/boot` for the `/boot` file system, and so on. For a swap file system, do not set the mount point as setting the file system type to swap is sufficient.
       - **b.** Enter the size of the file system in the **Desired Capacity** field. You can use common size units such as KiB or GiB. The default is MiB if you do not set any other unit.
c. Select the device type that you require from the drop-down Device Type menu: Standard Partition, LVM, or LVM Thin Provisioning.

WARNING
The installation program does not support overprovisioned LVM thin pools.

NOTE
RAID is available only if two or more disks are selected for partitioning. If you choose RAID, you can also set the RAID Level. Similarly, if you select LVM, you can specify the Volume Group.

d. Select the Encrypt check box to encrypt the partition or volume. You must set a password later in the installation program. The LUKS Version drop-down menu is displayed.

e. Select the LUKS version that you require from the drop-down menu.

f. Select the appropriate file system type for this partition or volume from the File system drop-down menu.

NOTE
Support for VFAT file system is not available for Linux system partitions. For example, /, /var, /usr, and so on.

g. Select the Reformat check box to format an existing partition, or clear the Reformat check box to retain your data. The newly-created partitions and volumes must be reformatted, and the check box cannot be cleared.

h. Type a label for the partition in the Label field. Use labels to easily recognize and address individual partitions.

i. Type a name in the Name field.

NOTE
Note that standard partitions are named automatically when they are created and you cannot edit the names of standard partitions. For example, you cannot edit the /boot name sda1.

3. Click Update Settings to apply your changes and if required, select another partition to customize. Changes are not applied until you click Begin Installation from the Installation Summary window.

NOTE
Click Reset All to discard your partition changes.
4. Click **Done** when you have created and customized all file systems and mount points. If you choose to encrypt a file system, you are prompted to create a passphrase. A **Summary of Changes** dialog box opens, displaying a summary of all storage actions for the installation program.

5. Click **Accept Changes** to apply the changes and return to the **Installation Summary** window.

### 4.6.5. Preserving the `/home` directory

In a Red Hat Enterprise Linux 8 graphical installation, you can preserve the `/home` directory that was used on your RHEL 7 system.

#### WARNING

Preserving `/home` is only possible if the `/home` directory is located on a separate `/home` partition on your RHEL 7 system.

Preserving the `/home` directory that includes various configuration settings, makes it possible that the GNOME Shell environment on the new Red Hat Enterprise Linux 8 system is set in the same way as it was on your RHEL 7 system. Note that this applies only for users on Red Hat Enterprise Linux 8 with the same user name and ID as on the previous RHEL 7 system.

Complete this procedure to preserve the `/home` directory from your RHEL 7 system.

**Prerequisites**

- RHEL 7 system is installed on your computer.
- The `/home` directory is located on a separate `/home` partition on your RHEL 7 system.
- The Red Hat Enterprise Linux 8 **Installation Summary** window is currently displayed.

**Procedure**

1. Click **Installation Destination** to open the **Installation Destination** window.

2. Under **Storage Configuration**, select the **Custom** radio button. Click **Done**.

3. Click **Done**, the **Manual Partitioning** window opens.

4. Choose the `/home` partition, fill in `/home` under **Mount Point**: and clear the **Reformat** check box.
Figure 4.2. Ensuring that /home is not formatted

5. Optional: You can also customize various aspects of the /home partition required for your Red Hat Enterprise Linux 8 system as described in Customizing a mount point file system. However, to preserve /home from your RHEL 7 system, it is necessary to clear the Reformat check box.

6. After you customized all partitions according to your requirements, click Done. The Summary of changes dialog box opens.

7. Verify that the Summary of changes dialog box does not show any change for /home. This means that the /home partition is preserved.

8. Click Accept Changes to apply the changes, and return to the Installation Summary window.

4.6.6. Creating software RAID

Follow the steps in this procedure to create a Redundant Arrays of Independent Disks (RAID) device. RAID devices are constructed from multiple storage devices that are arranged to provide increased performance and, in some configurations, greater fault tolerance.

A RAID device is created in one step and disks are added or removed as necessary. You can configure one RAID partition for each physical disk in your system, so the number of disks available to the installation program determines the levels of RAID device available. For example, if your system has two hard drives, you cannot create a RAID 10 device, as it requires a minimum of three separate disks.

NOTE

On 64-bit IBM Z, the storage subsystem uses RAID transparently. You do not have to configure software RAID manually.
Prerequisites

- You have selected two or more disks for installation before RAID configuration options are visible. At least two disks are required to create a RAID device.
- You have created a mount point. By configuring a mount point, you configure the RAID device.
- You have selected the Custom radio button on the Installation Destination window.

Procedure

1. From the left pane of the Manual Partitioning window, select the required partition.
2. Under the Device(s) section, click Modify. The Configure Mount Point dialog box opens.
3. Select the disks that you want to include in the RAID device and click Select.
4. Click the Device Type drop-down menu and select RAID.
5. Click the File System drop-down menu and select your preferred file system type.
6. Click the RAID Level drop-down menu and select your preferred level of RAID.
7. Click Update Settings to save your changes.
8. Click Done to apply the settings and return to the Installation Summary window.

A message is displayed at the bottom of the window if the specified RAID level requires more disks.

To learn more about soft corruption and how you can protect your data when configuring a RAID LV, see Using DM integrity with RAID LV.

4.6.7. Creating an LVM logical volume

Logical Volume Management (LVM) presents a simple logical view of underlying physical storage space, such as hard drives or LUNs. Partitions on physical storage are represented as physical volumes that you can group together into volume groups. You can divide each volume group into multiple logical volumes, each of which is analogous to a standard disk partition. Therefore, LVM logical volumes function as partitions that can span multiple physical disks.

NOTE

LVM configuration is available only in the graphical installation program.

IMPORTANT

During text-mode installation, LVM configuration is not available. To create an LVM configuration, press Ctrl+Alt+F2 to use a shell prompt in a different virtual console. You can run vgcreate and lvm commands in this shell. To return to the text-mode installation, press Ctrl+Alt+F1.

Procedure

1. From the left-hand pane of the Manual Partitioning window, select the mount point.
2. Click the Device Type drop-down menu and select LVM. The Volume Group drop-down menu is displayed with the newly-created volume group name.

**NOTE**

You cannot specify the size of the volume group’s physical extents in the configuration dialog. The size is always set to the default value of 4 MiB. If you want to create a volume group with different physical extents, you must create it manually by switching to an interactive shell and using the `vgcreate` command, or use a Kickstart file with the `volgroup --pesize=size` command. See the Performing an advanced RHEL installation document for more information about Kickstart.

Additional resources

- Configuring and managing logical volumes

### 4.6.8. Configuring an LVM logical volume

Follow the steps in this procedure to configure a newly-created LVM logical volume.

**WARNING**

Placing the `/boot` partition on an LVM volume is not supported.

**Procedure**

1. From the left-hand pane of the Manual Partitioning window, select the mount point.

2. Click the Device Type drop-down menu and select LVM. The Volume Group drop-down menu is displayed with the newly-created volume group name.

3. Click Modify to configure the newly-created volume group. The Configure Volume Group dialog box opens.

**NOTE**

You cannot specify the size of the volume group’s physical extents in the configuration dialog. The size is always set to the default value of 4 MiB. If you want to create a volume group with different physical extents, you must create it manually by switching to an interactive shell and using the `vgcreate` command, or use a Kickstart file with the `volgroup --pesize=size` command. See the Performing an advanced RHEL installation document for more information about Kickstart.

4. From the RAID Level drop-down menu, select the RAID level that you require. The available RAID levels are the same as with actual RAID devices.

5. Select the Encrypt check box to mark the volume group for encryption.
6. From the **Size policy** drop-down menu, select the size policy for the volume group. The available policy options are:

- **Automatic**: The size of the volume group is set automatically so that it is large enough to contain the configured logical volumes. This is optimal if you do not need free space within the volume group.

- **As large as possible**: The volume group is created with maximum size, regardless of the size of the configured logical volumes it contains. This is optimal if you plan to keep most of your data on LVM and later need to increase the size of some existing logical volumes, or if you need to create additional logical volumes within this group.

- **Fixed**: You can set an exact size of the volume group. Any configured logical volumes must then fit within this fixed size. This is useful if you know exactly how large you need the volume group to be.

7. Click **Save** to apply the settings and return to the **Manual Partitioning** window.

8. Click **Update Settings** to save your changes.

9. Click **Done** to return to the **Installation Summary** window.

### Additional resources

- How to use dm-crypt on IBM Z, LinuxONE and with the PAES cipher

### 4.7. CONFIGURING A ROOT PASSWORD

You must configure a root password to finish the installation process and to log in to the administrator (also known as superuser or root) account that is used for system administration tasks. These tasks include installing and updating software packages and changing system-wide configuration such as network and firewall settings, storage options, and adding or modifying users, groups and file permissions.

**IMPORTANT**

- Use one or both of the following ways to gain root privileges to the installed system:
  - Use a root account.
  - Create a user account with administrative privileges (member of the wheel group). The root account is always created during the installation. Switch to the administrator account only when you need to perform a task that requires administrator access.

**WARNING**

The root account has complete control over the system. If unauthorized personnel gain access to the account, they can access or delete users' personal files.
Procedure

1. From the Installation Summary window, select User Settings > Root Password The Root Password window opens.

2. Type your password in the Root Password field. The requirements and recommendations for creating a strong root password are:
   - Must be at least eight characters long
   - May contain numbers, letters (upper and lower case) and symbols
   - Is case-sensitive

3. Type the same password in the Confirm field.

4. Click Done to confirm your root password and return to the Installation Summary window.

   **NOTE**
   
   If you proceeded with a weak password, you must click Done twice.

4.8. CREATING A USER ACCOUNT

It is recommended that you create a user account to finish the installation. If you do not create a user account, you must log in to the system as root directly, which is not recommended.

Procedure


2. Type the user account name in to the Full name field, for example: John Smith.

3. Type the username in to the User name field, for example: jsmith.

   **NOTE**
   
   The User name is used to log in from a command line; if you install a graphical environment, then your graphical login manager uses the Full name.

4. Select the Make this user administrator check box if the user requires administrative rights (the installation program adds the user to the wheel group).

   **IMPORTANT**
   
   An administrator user can use the sudo command to perform tasks that are only available to root using the user password, instead of the root password. This may be more convenient, but it can also cause a security risk.

5. Select the Require a password to use this account check box.
6. Type a password into the Password field.

7. Type the same password into the Confirm password field.

8. Click Done to apply the changes and return to the Installation Summary window.

4.9. EDITING ADVANCED USER SETTINGS

This procedure describes how to edit the default settings for the user account in the Advanced User Configuration dialog box.

Procedure

1. On the Create User window, click Advanced.

2. Edit the details in the Home directory field, if required. The field is populated by default with /home/username.

3. In the User and Groups IDs section you can:
   a. Select the Specify a user ID manually check box and use + or - to enter the required value.

   ![NOTE]
   The default value is 1000. User IDs (UIDs) 0–999 are reserved by the system so they cannot be assigned to a user.

   ![NOTE]
   The default group name is the same as the user name, and the default Group ID (GID) is 1000. GIDs 0–999 are reserved by the system so they cannot be assigned to a user group.

   b. Select the Specify a group ID manually check box and use + or - to enter the required value.

   ![NOTE]
   The default group name is the same as the user name, and the default Group ID (GID) is 1000. GIDs 0–999 are reserved by the system so they cannot be assigned to a user group.

4. Specify additional groups as a comma-separated list in the Group Membership field. Groups that do not already exist are created; you can specify custom GIDs for additional groups in parentheses. If you do not specify a custom GID for a new group, the new group receives a GID automatically.
NOTE

The user account created always has one default group membership (the user’s default group with an ID set in the Specify a group ID manually field).

5. Click Save Changes to apply the updates and return to the Create User window.
CHAPTER 5. COMPLETING POST-INSTALLATION TASKS

This section describes how to complete the following post-installation tasks:

- Completing initial setup
- Registering your system

**NOTE**

Depending on your requirements, there are several methods to register your system. Most of these methods are completed as part of post-installation tasks. However, the Red Hat Content Delivery Network (CDN) registers your system and attaches RHEL subscriptions before the installation process starts. See Registering and installing RHEL from the CDN for more information.

- Securing your system

5.1. COMPLETING INITIAL SETUP

This section contains information about how to complete initial setup on a Red Hat Enterprise Linux 8 system.

**IMPORTANT**

- If you selected the Server with GUI base environment during installation, the Initial Setup window opens the first time you reboot your system after the installation process is complete.

- If you registered and installed RHEL from the CDN, the Subscription Manager option displays a note that all installed products are covered by valid entitlements.

The information displayed in the Initial Setup window might vary depending on what was configured during installation. At a minimum, the Licensing and Subscription Manager options are displayed.

**Prerequisites**

- You have completed the graphical installation according to the recommended workflow described on Installing RHEL using an ISO image from the Customer Portal.

- You have an active, non-evaluation Red Hat Enterprise Linux subscription.

**Procedure**

1. From the Initial Setup window, select Licensing Information. The License Agreement window opens and displays the licensing terms for Red Hat Enterprise Linux.

2. Review the license agreement and select the I accept the license agreement checkbox.
NOTE

You must accept the license agreement. Exiting Initial Setup without completing this step causes a system restart. When the restart process is complete, you are prompted to accept the license agreement again.

3. Click Done to apply the settings and return to the Initial Setup window.

NOTE

If you did not configure network settings, you cannot register your system immediately. In this case, click Finish Configuration. Red Hat Enterprise Linux 8 starts and you can login, activate access to the network, and register your system. See Subscription manager post installation for more information. If you configured network settings, as described in Network hostname, you can register your system immediately, as shown in the following steps:

4. From the Initial Setup window, select Subscription Manager.

IMPORTANT

If you registered and installed RHEL from the CDN, the Subscription Manager option displays a note that all installed products are covered by valid entitlements.

5. The Subscription Manager graphical interface opens and displays the option you are going to register, which is: subscription.rhsm.redhat.com.

6. Click Next.

7. Enter your Login and Password details and click Register.

8. Confirm the Subscription details and click Attach. You must receive the following confirmation message: Registration with Red Hat Subscription Management is Done!

9. Click Done. The Initial Setup window opens.

10. Click Finish Configuration. The login window opens.

11. Configure your system. See the Configuring basic system settings document for more information.

Additional resources

Depending on your requirements, there are five methods to register your system:

- Using the Red Hat Content Delivery Network (CDN) to register your system, attach RHEL subscriptions, and install Red Hat Enterprise Linux. See Register and install from CDN using GUI for more information.
- During installation using Initial Setup.
- After installation using the command line.
After installation using the Subscription Manager user interface. See Subscription manager post install UI for more information.

After installation using Registration Assistant. Registration Assistant is designed to help you choose the most suitable registration option for your Red Hat Enterprise Linux environment. See https://access.redhat.com/labs/registrationassistant/ for more information.

5.2. REGISTERING YOUR SYSTEM USING THE COMMAND LINE

This section contains information about how to register your Red Hat Enterprise Linux 8 subscription using the command line.

NOTE

When auto-attaching a system, the subscription service checks if the system is physical or virtual, as well as how many sockets are on the system. A physical system usually consumes two entitlements, a virtual system usually consumes one. One entitlement is consumed per two sockets on a system.

Prerequisites

- You have an active, non-evaluation Red Hat Enterprise Linux subscription.
- Your Red Hat subscription status is verified.
- You have never previously received a Red Hat Enterprise Linux 8 subscription.
- You have activated your subscription before attempting to download entitlements from the Customer Portal. You need an entitlement for each instance that you plan to use. Red Hat Customer Service is available if you need help activating your subscription.
- You have successfully installed Red Hat Enterprise Linux 8 and logged into the system as root.

Procedure

1. Open a terminal window and register your Red Hat Enterprise Linux system using your Red Hat Customer Portal username and password:

   ```
   # subscription-manager register --username [username] --password [password]
   ```

2. When the system is successfully registered, an output similar to the following is displayed:

   ```
   # The system has been registered with ID: 123456abcdef
   # The registered system name is: localhost.localdomain
   ```

3. Set the role for the system, for example:

   ```
   # subscription-manager role --set="Red Hat Enterprise Linux Server"
   ```
Available roles depend on the subscriptions that have been purchased by the organization and the architecture of the Red Hat Enterprise Linux 8 system. You can set one of the following roles: **Red Hat Enterprise Linux Server**, **Red Hat Enterprise Linux Workstation**, or **Red Hat Enterprise Linux Compute Node**.

4. Set the service level for the system, for example:

   ```bash
   # subscription-manager service-level --set="Premium"
   ```

5. Set the usage for the system, for example:

   ```bash
   # subscription-manager usage --set="Production"
   ```

6. Attach the system to an entitlement that matches the host system architecture:

   ```bash
   # subscription-manager attach --auto
   ```

7. When a subscription is successfully attached, an output similar to the following is displayed:

   **Installed Product Current Status:**
   
   **Product Name:** Red Hat Enterprise Linux for x86_64
   
   **Status:** Subscribed

---

**NOTE**

An alternative method for registering your Red Hat Enterprise Linux 8 system is by logging in to the system as a **root** user and using the Subscription Manager graphical user interface.

### 5.3. REGISTERING YOUR SYSTEM USING THE SUBSCRIPTION MANAGER USER INTERFACE

This section contains information about how to register your Red Hat Enterprise Linux 8 system using the Subscription Manager User Interface to receive updates and access package repositories.

**Prerequisites**

- You have completed the graphical installation as per the recommended workflow described on [Installing RHEL using an ISO image from the Customer Portal](#).
- You have an active, non-evaluation Red Hat Enterprise Linux subscription.
- Your Red Hat subscription status is verified.

**Procedure**

1. Log in to your system.

2. From the top left-hand side of the window, click **Activities**.

3. From the menu options, click the **Show Applications** icon.
4. Click the Red Hat Subscription Manager icon, or enter Red Hat Subscription Manager in the search.

5. Enter your administrator password in the Authentication Required dialog box.

   **NOTE**
   
   Authentication is required to perform privileged tasks on the system.

6. The Subscriptions window opens, displaying the current status of Subscriptions, System Purpose, and installed products. Unregistered products display a red X.

7. Click the Register button.

8. The Register System dialog box opens. Enter your Customer Portal credentials and click the Register button.

   The Register button in the Subscriptions window changes to Unregister and installed products display a green X. You can troubleshoot an unsuccessful registration from a terminal window using the subscription-manager status command.

**Additional resources**

- *Using and Configuring Red Hat Subscription Manager*
- *Configuring Virtual Machine Subscriptions in Red Hat Subscription Management*

### 5.4. REGISTRATION ASSISTANT

Registration Assistant is designed to help you choose the most suitable registration option for your Red Hat Enterprise Linux environment. See [https://access.redhat.com/labs/registrationassistant/](https://access.redhat.com/labs/registrationassistant/) for more information.

### 5.5. CONFIGURING SYSTEM PURPOSE USING THE SYSPURPOSE COMMAND-LINE TOOL

System Purpose is an optional but recommended feature of the Red Hat Enterprise Linux installation. You can use System Purpose to record the intended use of a Red Hat Enterprise Linux 8 system, and ensure that the entitlement server auto-attaches the most appropriate subscription to your system. The syspurpose command-line tool is part of the python3_syspurpose.rpm package. If System Purpose was not configured during the installation process, you can use the syspurpose command-line tool after installation to set the required attributes.

**Prerequisites**

- You installed and registered your Red Hat Enterprise Linux 8 system, but System Purpose is not configured.
- You are logged in as a root user.
- The python3_syspurpose.rpm package is available on your system.
NOTE

If your system is registered but has subscriptions that do not satisfy the required purpose, you can run the `subscription-manager remove --all` command to remove attached subscriptions. You can then use the `syspurpose` command-line tool to set the required purpose attributes, and run `subscription-manager attach --auto` to entitle the system with the updated attributes.

Procedure

Complete the steps in this procedure to configure System Purpose after installation using the `syspurpose` command-line tool. The selected values are used by the entitlement server to attach the most suitable subscription to your system.

1. From a terminal window, run the following command to set the intended role of the system:

   ```
   # syspurpose set-role "VALUE"
   ```

   Replace `VALUE` with the role that you want to assign:

   - Red Hat Enterprise Linux Server
   - Red Hat Enterprise Linux Workstation
   - Red Hat Enterprise Linux Compute Node

   For example:

   ```
   # syspurpose set-role "Red Hat Enterprise Linux Server"
   ```

   a. Optional: Run the following command to unset the role:

   ```
   # syspurpose unset-role
   ```

2. Run the following command to set the intended Service Level Agreement (SLA) of the system:

   ```
   # syspurpose set-sla "VALUE"
   ```

   Replace `VALUE` with the SLA that you want to assign:

   - Premium
   - Standard
   - Self-Support

   For example:

   ```
   # syspurpose set-sla "Standard"
   ```

   a. Optional: Run the following command to unset the SLA:

   ```
   # syspurpose unset-sla
   ```
3. Run the following command to set the intended usage of the system:

```bash
# syspurpose set-usage "VALUE"
```

Replace `VALUE` with the usage that you want to assign:

- **Production**
- **Disaster Recovery**
- **Development/Test**

For example:

```bash
# syspurpose set-usage "Production"
```

a. Optional: Run the following command to unset the usage:

```bash
# syspurpose unset-usage
```

4. Run the following command to show the current system purpose properties:

```bash
# syspurpose show
```

a. Optional: Run the following command to access the `syspurpose` man page:

```bash
# man syspurpose
```

### 5.6. SECURING YOUR SYSTEM

Complete the following security-related steps immediately after you install Red Hat Enterprise Linux.

**Prerequisites**

- You have completed the graphical installation. See [Installing RHEL using an ISO image from the Customer Portal](#).

**Procedure**

1. To update your system, run the following command as root:

```bash
# yum update
```

2. Even though the firewall service, `firewalld`, is automatically enabled with the installation of Red Hat Enterprise Linux, there are scenarios where it might be explicitly disabled, for example in a Kickstart configuration. In that scenario, it is recommended that you re-enable the firewall. To start `firewalld`, run the following commands as root:

```bash
# systemctl start firewalld
# systemctl enable firewalld
```
3. To enhance security, disable services that you do not need. For example, if your system has no printers installed, disable the cups service using the following command:

```
# systemctl mask cups
```

To review active services, run the following command:

```
$ systemctl list-units | grep service
```

## 5.7. DEPLOYING SYSTEMS THAT ARE COMPLIANT WITH A SECURITY PROFILE IMMEDIATELY AFTER AN INSTALLATION

You can use the OpenSCAP suite to deploy RHEL systems that are compliant with a security profile, such as OSPP, PCI-DSS, and HIPAA profile, immediately after the installation process. Using this deployment method, you can apply specific rules that cannot be applied later using remediation scripts, for example, a rule for password strength and partitioning.

### 5.7.1. Profiles not compatible with Server with GUI

Certain security profiles provided as part of the SCAP Security Guide are not compatible with the extended package set included in the Server with GUI base environment. Therefore, do not select Server with GUI when installing systems compliant with one of the following profiles:

<table>
<thead>
<tr>
<th>Profile name</th>
<th>Profile ID</th>
<th>Justification</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIS Red Hat Enterprise Linux 8 Benchmark for Level 2 - Server</td>
<td>xccdf_org.ssgproj ect.content_profile_cis</td>
<td>Packages <code>xorg-x11-server-Xorg</code>, <code>xorg-x11-server-common</code>, <code>xorg-x11-server-utils</code>, and <code>xorg-x11-server-Xwayland</code> are part of the Server with GUI package set, but the policy requires their removal.</td>
<td></td>
</tr>
<tr>
<td>CIS Red Hat Enterprise Linux 8 Benchmark for Level 1 - Server</td>
<td>xccdf_org.ssgproj ect.content_profile_cis_server_l1</td>
<td>Packages <code>xorg-x11-server-Xorg</code>, <code>xorg-x11-server-common</code>, <code>xorg-x11-server-utils</code>, and <code>xorg-x11-server-Xwayland</code> are part of the Server with GUI package set, but the policy requires their removal.</td>
<td></td>
</tr>
</tbody>
</table>
### 5.7.2. Deploying baseline-compliant RHEL systems using the graphical installation

Use this procedure to deploy a RHEL system that is aligned with a specific baseline. This example uses Protection Profile for General Purpose Operating System (OSPP).

#### WARNING

Certain security profiles provided as part of the SCAP Security Guide are not compatible with the extended package set included in the Server with GUI base environment. For additional details, see Profiles not compatible with a GUI server.

#### Prerequisites

- You have booted into the graphical installation program. Note that the OSCAP Anaconda Add-on does not support interactive text-only installation.
- You have accessed the Installation Summary window.

#### Procedure

1. From the Installation Summary window, click Software Selection. The Software Selection window opens.
2. From the **Base Environment** pane, select the **Server** environment. You can select only one base environment.

3. Click **Done** to apply the setting and return to the **Installation Summary** window.

4. Click **Security Policy**. The **Security Policy** window opens.

5. To enable security policies on the system, toggle the **Apply security policy** switch to **ON**.

6. Select **Protection Profile for General Purpose Operating Systems** from the profile pane.

7. Click **Select Profile** to confirm the selection.

8. Confirm the changes in the **Changes that were done or need to be done** pane that is displayed at the bottom of the window. Complete any remaining manual changes.

9. Because OSPP has strict partitioning requirements that must be met, create separate partitions for /boot, /home, /var, /var/log, /var/tmp, and /var/log/audit.

10. Complete the graphical installation process.

**NOTE**

The graphical installation program automatically creates a corresponding Kickstart file after a successful installation. You can use the /root/anaconda-ks.cfg file to automatically install OSPP-compliant systems.

**Verification**

- To check the current status of the system after installation is complete, reboot the system and start a new scan:

```
# oscap xccdf eval --profile ospp --report eval_postinstall_report.html
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

**Additional resources**

- Configuring manual partitioning

**5.7.3. Deploying baseline-compliant RHEL systems using Kickstart**

Use this procedure to deploy RHEL systems that are aligned with a specific baseline. This example uses Protection Profile for General Purpose Operating System (OSPP).

**Prerequisites**

- The **scap-security-guide** package is installed on your RHEL 8 system.

**Procedure**

2. Update the partitioning scheme to fit your configuration requirements. For OSPP compliance, the separate partitions for `/boot`, `/home`, `/var`, `/var/log`, `/var/tmp`, and `/var/log/audit` must be preserved, and you can only change the size of the partitions.

3. Start a Kickstart installation as described in Performing an automated installation using Kickstart.

**IMPORTANT**

Passwords in Kickstart files are not checked for OSPP requirements.

**Verification**

1. To check the current status of the system after installation is complete, reboot the system and start a new scan:

   ```bash
   # oscap xccdf eval --profile ospp --report eval_postinstall_report.html
   /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
   ```

**Additional resources**

- OSCAP Anaconda Addon

**5.8. NEXT STEPS**

When you have completed the required post-installation steps, you can configure basic system settings. For information about completing tasks such as installing software with yum, using systemd for service management, managing users, groups, and file permissions, using chrony to configure NTP, and working with Python 3, see the Configuring basic system settings document.
APPENDIX A. TROUBLESHOOTING

The following sections cover various troubleshooting information that might be helpful when diagnosing issues during different stages of the installation process.
APPENDIX B. TOOLS AND TIPS FOR TROUBLESHOOTING AND BUG REPORTING

The troubleshooting information in the following sections might be helpful when diagnosing issues at the start of the installation process. The following sections are for all supported architectures. However, if an issue is for a particular architecture, it is specified at the start of the section.

B.1. Dracut

Dracut is a tool that manages the initramfs image during the Linux operating system boot process. The dracut emergency shell is an interactive mode that can be initiated while the initramfs image is loaded. You can run basic troubleshooting commands from the dracut emergency shell. For more information, see the Troubleshooting section of the dracut man page.

B.2. Using installation log files

For debugging purposes, the installation program logs installation actions in files that are located in the /tmp directory. These log files are listed in the following table.

Table B.1. Log files generated during the installation

<table>
<thead>
<tr>
<th>Log file</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>/tmp/anaconda.log</td>
<td>General messages.</td>
</tr>
<tr>
<td>/tmp/program.log</td>
<td>All external programs run during the installation.</td>
</tr>
<tr>
<td>/tmp/storage.log</td>
<td>Extensive storage module information.</td>
</tr>
<tr>
<td>/tmp/packaging.log</td>
<td>yum and rpm package installation messages.</td>
</tr>
<tr>
<td>/tmp/dbus.log</td>
<td>Information about the dbus session that is used for installation program modules.</td>
</tr>
<tr>
<td>/tmp/sensitive-info.log</td>
<td>Configuration information that is not part of other logs and not copied to the installed system.</td>
</tr>
<tr>
<td>/tmp/syslog</td>
<td>Hardware-related system messages. This file contains messages from other Anaconda files.</td>
</tr>
</tbody>
</table>

If the installation fails, the messages are consolidated into /tmp/anaconda-tb-identifier, where identifier is a random string. After a successful installation, these files are copied to the installed system under the directory /var/log/anaconda/. However, if the installation is unsuccessful, or if the inst.nosave=all or inst.nosave=logs options are used when booting the installation system, these logs only exist in the installation program’s RAM disk. This means that the logs are not saved permanently and are lost when the system is powered down. To store them permanently, copy the files to another system on the network or copy them to a mounted storage device such as a USB flash drive.

B.2.1. Creating pre-installation log files

Use this procedure to set the inst.debug option to create log files before the installation process starts.
Use this procedure to set the `inst.debug` option to create log files before the installation process starts. These log files contain, for example, the current storage configuration.

**Prerequisites**

- The Red Hat Enterprise Linux boot menu is displayed.

**Procedure**

1. Select the **Install Red Hat Enterprise Linux** option from the boot menu.
2. Press the **Tab** key on BIOS-based systems or the **e** key on UEFI-based systems to edit the selected boot options.
3. Append `inst.debug` to the options. For example:
   ```
   vmlinuz ... inst.debug
   ```
4. Press the **Enter** key on your keyboard. The system stores the pre-installation log files in the `/tmp/pre-anaconda-logs/` directory before the installation program starts.
5. To access the log files, switch to the console.
6. Change to the `/tmp/pre-anaconda-logs/` directory:
   ```
   # cd /tmp/pre-anaconda-logs/
   ```

**Additional resources**

- Custom boot options
- Console logging during installation

### B.2.2. Transferring installation log files to a USB drive

Use this procedure to transfer installation log files to a USB drive.

**Prerequisites**

- Back up any data on the USB drive before using this procedure.
- You are logged into a root account and you have access to the installation program’s temporary file system.

**Procedure**

1. Press `Ctrl + Alt + F2` to access a shell prompt on the system you are installing.
2. Connect a USB flash drive to the system and run the `dmesg` command:
   ```
   # dmesg
   ```

   A log detailing all recent events is displayed. At the end of this log, a set of messages is displayed. For example:
3. Note the name of the connected device. In the above example, it is `sdb`.

4. Navigate to the `/mnt` directory and create a new directory that serves as the mount target for the USB drive. This example uses the name `usb`:

   ```
   # mkdir usb
   ```

5. Mount the USB flash drive onto the newly created directory. In most cases, you do not want to mount the whole drive, but a partition on it. Do not use the name `sdb`, use the name of the partition you want to write the log files to. In this example, the name `sdb1` is used:

   ```
   # mount /dev/sdb1 /mnt/usb
   ```

6. Verify that you mounted the correct device and partition by accessing it and listing its contents:

   ```
   # cd /mnt/usb
   # ls
   ```

7. Copy the log files to the mounted device.

   ```
   # cp /tmp/*log /mnt/usb
   ```

8. Unmount the USB flash drive. If you receive an error message that the target is busy, change your working directory to outside the mount (for example, `/`).

   ```
   # umount /mnt/usb
   ```

B.2.3. Transferring installation log files over the network

Use this procedure to transfer installation log files over the network.

Prerequisites

- You are logged into a root account and you have access to the installation program’s temporary file system.

Procedure

1. Press `Ctrl + Alt + F2` to access a shell prompt on the system you are installing.

2. Switch to the `/tmp` directory where the log files are located:

   ```
   # cd /tmp
   ```

3. Copy the log files onto another system on the network using the `scp` command:

   ```
   # scp *log user@address:path
   ```

   a. Replace `user` with a valid user name on the target system, `address` with the target system’s
address or host name, and **path** with the path to the directory where you want to save the log files. For example, if you want to log in as **john** on a system with an IP address of 192.168.0.122 and place the log files into the `/home/john/logs/` directory on that system, the command is as follows:

```
# scp *log john@192.168.0.122:/home/john/logs/
```

When connecting to the target system for the first time, the SSH client asks you to confirm that the fingerprint of the remote system is correct and that you want to continue:

```
The authenticity of host '192.168.0.122 (192.168.0.122)' can't be established.
Are you sure you want to continue connecting (yes/no)?
```

b. Type **yes** and press **Enter** to continue. Provide a valid password when prompted. The files are transferred to the specified directory on the target system.

### B.3. Detecting memory faults using the Memtest86 application

Faults in memory (RAM) modules can cause your system to fail unpredictably. In certain situations, memory faults might only cause errors with particular combinations of software. For this reason, you should test your system’s memory before you install Red Hat Enterprise Linux.

**NOTE**

Red Hat Enterprise Linux includes the **Memtest86+** memory testing application for BIOS systems only. Support for UEFI systems is currently unavailable.

### B.3.1. Running Memtest86

Use this procedure to run the **Memtest86** application to test your system’s memory for faults before you install Red Hat Enterprise Linux.

**Prerequisites**

- You have accessed the Red Hat Enterprise Linux boot menu.

**Procedure**

1. From the Red Hat Enterprise Linux boot menu, select **Troubleshooting > Run a memory test**

   The **Memtest86** application window is displayed and testing begins immediately. By default, **Memtest86** performs ten tests in every pass. After the first pass is complete, a message is displayed in the lower part of the window informing you of the current status. Another pass starts automatically.

   If **Memtest86+** detects an error, the error is displayed in the central pane of the window and is highlighted in red. The message includes detailed information such as which test detected a problem, the memory location that is failing, and others. In most cases, a single successful pass of all 10 tests is sufficient to verify that your RAM is in good condition. In rare circumstances, however, errors that went undetected during the first pass might appear on subsequent passes. To perform a thorough test on important systems, run the tests overnight or for a few days to complete multiple passes.
The amount of time it takes to complete a single full pass of Memtest86+ varies depending on your system’s configuration, notably the RAM size and speed. For example, on a system with 2 GiB of DDR2 memory at 667 MHz, a single pass takes 20 minutes to complete.

2. Optional: Follow the on-screen instructions to access the Configuration window and specify a different configuration.

3. To halt the tests and reboot your computer, press the Esc key at any time.

Additional resources

- How to use Memtest86

B.4. Verifying boot media

Verifying ISO images helps to avoid problems that are sometimes encountered during installation. These sources include DVD and ISO images stored on a hard drive or NFS server. Use this procedure to test the integrity of an ISO-based installation source before using it to install Red Hat Enterprise Linux.

Prerequisites

- You have accessed the Red Hat Enterprise Linux boot menu.

Procedure

1. From the boot menu, select Test this media & install Red Hat Enterprise Linux 8.1 to test the boot media.

2. The boot process tests the media and highlights any issues.

3. Optional: You can start the verification process by appending rd.live.check to the boot command line.

B.5. Consoles and logging during installation

The Red Hat Enterprise Linux installer uses the tmux terminal multiplexer to display and control several windows in addition to the main interface. Each of these windows serve a different purpose; they display several different logs, which can be used to troubleshoot issues during the installation process. One of the windows provides an interactive shell prompt with root privileges, unless this prompt was specifically disabled using a boot option or a Kickstart command.

NOTE

In general, there is no reason to leave the default graphical installation environment unless you need to diagnose an installation problem.

The terminal multiplexer is running in virtual console 1. To switch from the actual installation environment to tmux, press Ctrl+Alt+F1. To go back to the main installation interface which runs in virtual console 6, press Ctrl+Alt+F6.
NOTE

If you choose text mode installation, you will start in virtual console 1 (tmux), and switching to console 6 will open a shell prompt instead of a graphical interface.

The console running tmux has five available windows; their contents are described in the following table, along with keyboard shortcuts. Note that the keyboard shortcuts are two-part: first press Ctrl+b, then release both keys, and press the number key for the window you want to use.

You can also use Ctrl+b n, Alt+ Tab, and Ctrl+b p to switch to the next or previous tmux window, respectively.

Table B.2. Available tmux windows

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl+b 1</td>
<td>Main installation program window. Contains text-based prompts (during text mode installation or if you use VNC direct mode), and also some debugging information.</td>
</tr>
<tr>
<td>Ctrl+b 2</td>
<td>Interactive shell prompt with root privileges.</td>
</tr>
<tr>
<td>Ctrl+b 3</td>
<td>Installation log; displays messages stored in /tmp/anaconda.log.</td>
</tr>
<tr>
<td>Ctrl+b 4</td>
<td>Storage log; displays messages related to storage devices and configuration, stored in /tmp/storage.log.</td>
</tr>
<tr>
<td>Ctrl+b 5</td>
<td>Program log; displays messages from utilities executed during the installation process, stored in /tmp/program.log.</td>
</tr>
</tbody>
</table>

B.6. Saving screenshots

You can press Shift+Print Screen at any time during the graphical installation to capture the current screen. The screenshots are saved to /tmp/anaconda-screenshots.

B.7. Display settings and device drivers

Some video cards have trouble booting into the Red Hat Enterprise Linux graphical installation program. If the installation program does not run using its default settings, it attempts to run in a lower resolution mode. If that fails, the installation program attempts to run in text mode. There are several possible solutions to resolve display issues, most of which involve specifying custom boot options.

For more information, see Console boot options.

Table B.3. Solutions
<table>
<thead>
<tr>
<th>Solution</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use the basic graphics mode</td>
<td>You can attempt to perform the installation using the basic graphics driver. To do this, either select <strong>Troubleshooting &gt; Install Red Hat Enterprise Linux in basic graphics mode</strong> from the boot menu, or edit the installation program’s boot options and append <strong>inst.xdriver=vesa</strong> at the end of the command line.</td>
</tr>
<tr>
<td>Specify the display resolution manually</td>
<td>If the installation program fails to detect your screen resolution, you can override the automatic detection and specify it manually. To do this, append the <strong>inst.resolution=x</strong> option at the boot menu, where x is your display’s resolution, for example, 1024x768.</td>
</tr>
<tr>
<td>Use an alternate video driver</td>
<td>You can attempt to specify a custom video driver, overriding the installation program’s automatic detection. To specify a driver, use the <strong>inst.xdriver=x</strong> option, where x is the device driver you want to use (for example, nouveau)*.</td>
</tr>
<tr>
<td>Perform the installation using VNC</td>
<td>If the above options fail, you can use a separate system to access the graphical installation over the network, using the Virtual Network Computing (VNC) protocol. For details on installing using VNC, see the <strong>Performing a remote RHEL installation using VNC</strong> section of the <strong>Performing an advanced RHEL installation</strong> document.</td>
</tr>
</tbody>
</table>

*If specifying a custom video driver solves your problem, you should report it as a bug at [https://bugzilla.redhat.com](https://bugzilla.redhat.com) under the **anaconda** component. The installation program should be able to detect your hardware automatically and use the appropriate driver without intervention.

**B.8. Reporting error messages to Red Hat Customer Support**

If the graphical installation encounters an error, it displays the **unknown error** dialog box. You can send information about the error to Red Hat Customer Support. To send a report, you must enter your Customer Portal credentials. If you do not have a Customer Portal account, you can register at [https://www.redhat.com/wapps/ugc/register.html](https://www.redhat.com/wapps/ugc/register.html). Automated error reporting requires a network connection.

**Prerequisite**

The graphical installation program encountered an error and displayed the **unknown error** dialog box.

**Procedure**

1. From the **unknown error** dialog box, click **Report Bug** to report the problem, or **Quit** to exit the installation.
   a. Optionally, click **More Info...** to display a detailed output that might help determine the cause of the error. If you are familiar with debugging, click **Debug**. This displays the virtual
terminal tty1, where you can request additional information. To return to the graphical interface from tty1, use the continue command.

2. Click **Report a bug to Red Hat Customer Support**.

3. The **Red Hat Customer Support - Reporting Configuration** dialog box is displayed. From the **Basic** tab, enter your Customer Portal user name and password. If your network settings require you to use an HTTP or HTTPS proxy, you can configure it by selecting the **Advanced** tab and entering the address of the proxy server.

4. Complete all fields and click **OK**.

5. A text box is displayed. Explain each step that was taken before the **unknown error** dialog box was displayed.

6. Select an option from the **How reproducible is this problem** drop-down menu and provide additional information in the text box.

7. Click **Forward**.

8. Verify that all the information you provided is in the **Comment** tab. The other tabs include information such as your system’s host name and other details about your installation environment. You can remove any of the information that you do not want to send to Red Hat, but be aware that providing less detail might affect the investigation of the issue.

9. Click **Forward** when you have finished reviewing all tabs.

10. A dialog box displays all the files that will be sent to Red Hat. Clear the check boxes beside the files that you do not want to send to Red Hat. To add a file, click **Attach a file**.

11. Select the check box **I have reviewed the data and agree with submitting it**.

12. Click **Forward** to send the report and attachments to Red Hat.

13. Click **Show log** to view the details of the reporting process or click **Close** to return to the **unknown error** dialog box.

14. Click **Quit** to exit the installation.

### A.1. TROUBLESHOOTING DURING THE INSTALLATION

The troubleshooting information in the following sections might be helpful when diagnosing issues during the installation process. The following sections are for all supported architectures. However, if an issue is for a particular architecture, it is specified at the start of the section.

#### A.1.1. Disks are not detected

If the installation program cannot find a writable storage device to install to, it returns the following error message in the **Installation Destination** window: **No disks detected. Please shut down the computer, connect at least one disk, and restart to complete installation.**

Check the following items:

- Your system has at least one storage device attached.
If your system uses a hardware RAID controller; verify that the controller is properly configured and working as expected. See your controller’s documentation for instructions.

If you are installing into one or more iSCSI devices and there is no local storage present on the system, verify that all required LUNs are presented to the appropriate Host Bus Adapter (HBA).

If the error message is still displayed after rebooting the system and starting the installation process, the installation program failed to detect the storage. In many cases the error message is a result of attempting to install on an iSCSI device that is not recognized by the installation program.

In this scenario, you must perform a driver update before starting the installation. Check your hardware vendor’s website to determine if a driver update is available. For more general information on driver updates, see the Updating drivers during installation section of the Performing an advanced RHEL installation document.

You can also consult the Red Hat Hardware Compatibility List, available at https://access.redhat.com/ecosystem/search/#/category/Server.

A.1.2. Reporting error messages to Red Hat Customer Support

If the graphical installation encounters an error, it displays the unknown error dialog box. You can send information about the error to Red Hat Customer Support. To send a report, you must enter your Customer Portal credentials. If you do not have a Customer Portal account, you can register at https://www.redhat.com/wapps/ugc/register.html. Automated error reporting requires a network connection.

Prerequisite

The graphical installation program encountered an error and displayed the unknown error dialog box.

Procedure

1. From the unknown error dialog box, click Report Bug to report the problem, or Quit to exit the installation.
   a. Optionally, click More Info... to display a detailed output that might help determine the cause of the error. If you are familiar with debugging, click Debug. This displays the virtual terminal tty1, where you can request additional information. To return to the graphical interface from tty1, use the continue command.


3. The Red Hat Customer Support - Reporting Configuration dialog box is displayed. From the Basic tab, enter your Customer Portal user name and password. If your network settings require you to use an HTTP or HTTPS proxy, you can configure it by selecting the Advanced tab and entering the address of the proxy server.

4. Complete all fields and click OK.

5. A text box is displayed. Explain each step that was taken before the unknown error dialog box was displayed.

6. Select an option from the How reproducible is this problem drop-down menu and provide additional information in the text box.

7. Click Forward.
8. Verify that all the information you provided is in the **Comment** tab. The other tabs include information such as your system's host name and other details about your installation environment. You can remove any of the information that you do not want to send to Red Hat, but be aware that providing less detail might affect the investigation of the issue.

9. Click **Forward** when you have finished reviewing all tabs.

10. A dialog box displays all the files that will be sent to Red Hat. Clear the check boxes beside the files that you do not want to send to Red Hat. To add a file, click **Attach a file**.

11. Select the check box **I have reviewed the data and agree with submitting it**.

12. Click **Forward** to send the report and attachments to Red Hat.

13. Click **Show log** to view the details of the reporting process or click **Close** to return to the **unknown error** dialog box.

14. Click **Quit** to exit the installation.

### A.1.3. Partitioning issues for IBM Power Systems

**NOTE**

This issue is for IBM Power Systems.

If you manually created partitions, but cannot move forward in the installation process, you might not have created all the partitions that are necessary for the installation to proceed. At a minimum, you must have the following partitions:

- **/ (root) partition**
- **PReP boot partition**
- **/boot** partition (only if the root partition is an LVM logical volume)

**Additional resources**

- See [Recommended Partitioning Scheme](#)
APPENDIX C. TROUBLESHOOTING

The troubleshooting information in the following sections might be helpful when diagnosing issues after the installation process. The following sections are for all supported architectures. However, if an issue is for a particular architecture, it is specified at the start of the section.

C.1. Resuming an interrupted download attempt

You can resume an interrupted download using the `curl` command.

**Prerequisite**

- You have navigated to the **Product Downloads** section of the Red Hat Customer Portal at [https://access.redhat.com/downloads](https://access.redhat.com/downloads), and selected the required variant, version, and architecture.

- You have right-clicked on the required ISO file, and selected **Copy Link Location** to copy the URL of the ISO image file to your clipboard.

**Procedure**

1. Download the ISO image from the new link. Add the `--continue-at -` option to automatically resume the download:

   ```
   $ curl --output directory-path/filename.iso 'new_copied_link_location' --continue-at -
   ```

2. Use a checksum utility such as `sha256sum` to verify the integrity of the image file after the download finishes:

   ```
   $ sha256sum rhel-x.x-x86_64-dvd.iso '85a...46c rhel-x.x-x86_64-dvd.iso'
   ```

   Compare the output with reference checksums provided on the Red Hat Enterprise Linux **Product Download** web page.

**Example C.1. Resuming an interrupted download attempt**

The following is an example of a `curl` command for a partially downloaded ISO image:

```
$ curl --output _rhel-x.x-x86_64-dvd.iso 'https://access.cdn.redhat.com/content/origin/files/sha256/85/85a...46c/rhel-x.x-x86_64-dvd.iso?_auth=141...963' --continue-at -
```

C.2. Disks are not detected

If the installation program cannot find a writable storage device to install to, it returns the following error message in the **Installation Destination** window: **No disks detected. Please shut down the computer, connect at least one disk, and restart to complete installation.**

Check the following items:

- Your system has at least one storage device attached.
If your system uses a hardware RAID controller; verify that the controller is properly configured and working as expected. See your controller’s documentation for instructions.

If you are installing into one or more iSCSI devices and there is no local storage present on the system, verify that all required LUNs are presented to the appropriate Host Bus Adapter (HBA).

If the error message is still displayed after rebooting the system and starting the installation process, the installation program failed to detect the storage. In many cases the error message is a result of attempting to install on an iSCSI device that is not recognized by the installation program.

In this scenario, you must perform a driver update before starting the installation. Check your hardware vendor’s website to determine if a driver update is available. For more general information on driver updates, see the Updating drivers during installation section of the Performing an advanced RHEL installation document.

You can also consult the Red Hat Hardware Compatibility List, available at https://access.redhat.com/ecosystem/search/#/category/Server.

C.3. Cannot boot with a RAID card

If you cannot boot your system after the installation, you might need to reinstall and repartition your system’s storage. Some BIOS types do not support booting from RAID cards. After you finish the installation and reboot the system for the first time, a text-based screen displays the boot loader prompt (for example, grub>) and a flashing cursor might be displayed. If this is the case, you must repartition your system and move your /boot partition and the boot loader outside of the RAID array. The /boot partition and the boot loader must be on the same drive. Once these changes have been made, you should be able to finish your installation and boot the system properly.

C.4. Graphical boot sequence is not responding

When rebooting your system for the first time after installation, the system might be unresponsive during the graphical boot sequence. If this occurs, a reset is required. In this scenario, the boot loader menu is displayed successfully, but selecting any entry and attempting to boot the system results in a halt. This usually indicates that there is a problem with the graphical boot sequence. To resolve the issue, you must disable the graphical boot by temporarily altering the setting at boot time before changing it permanently.

Procedure: Disabling the graphical boot temporarily

1. Start your system and wait until the boot loader menu is displayed. If you set your boot timeout period to 0, press the Esc key to access it.

2. From the boot loader menu, use your cursor keys to highlight the entry you want to boot. Press the Tab key on BIOS-based systems or the e key on UEFI-based systems to edit the selected entry options.

3. In the list of options, find the kernel line - that is, the line beginning with the keyword linux. On this line, locate and delete rhgb.

4. Press F10 or Ctrl+X to boot your system with the edited options.

If the system started successfully, you can log in normally. However, if you do not disable graphical boot permanently, you must perform this procedure every time the system boots.

Procedure: Disabling the graphical boot permanently
1. Log in to the root account on your system.

2. Use the grubby tool to find the default GRUB2 kernel:
   
   ```
   # grubby --default-kernel
   /boot/vmlinuz-4.18.0-94.el8.x86_64
   ```

3. Use the grubby tool to remove the **rhgb** boot option from the default kernel in your GRUB2 configuration. For example:

   ```
   # grubby --remove-args="rhgb" --update-kernel /boot/vmlinuz-4.18.0-94.el8.x86_64
   ```

4. Reboot the system. The graphical boot sequence is no longer used. If you want to enable the graphical boot sequence, follow the same procedure, replacing the **--remove-args="rhgb"** parameter with the **--args="rhgb"** parameter. This restores the **rhgb** boot option to the default kernel in your GRUB2 configuration.

### C.5. X server fails after log in

An X server is a program in the X Window System that runs on local machines, that is, the computers used directly by users. X server handles all access to the graphics cards, display screens and input devices, typically a keyboard and mouse on those computers. The X Window System, often referred to as X, is a complete, cross-platform and free client-server system for managing GUIs on single computers and on networks of computers. The client-server model is an architecture that divides the work between two separate but linked applications, referred to as clients and servers.*

If X server crashes after login, one or more of the file systems might be full. To troubleshoot the issue, execute the following command:

```
$ df -h
```

The output verifies which partition is full - in most cases, the problem is on the `/home` partition. The following is a sample output of the `df` command:

```
Filesystem Size Used Avail Use% Mounted on
/devtmpfs 396M 0 396M 0% /dev
tmpfs 411M 0 411M 0% /dev/shm
tmpfs 411M 6.7M 405M 2% /run
tmpfs 411M 0 411M 0% /sys/fs/cgroup
/devmapper/rhel-root 17G 4.1G 13G 25% /
/dev/sda1 1014M 173M 842M 17% /boot
tmpfs 83M 20K 83M 1% /run/user/42
tmpfs 83M 84K 83M 1% /run/user/1000
/dev/dm-4 90G 90G 0 100% /home
```

In the example, you can see that the `/home` partition is full, which causes the failure. Remove any unwanted files. After you free up some disk space, start X using the `startx` command. For additional information about `df` and an explanation of the options available, such as the `-h` option used in this example, see the `df(1)` man page.

*Source: [http://www.linfo.org/x_server.html](http://www.linfo.org/x_server.html)

### C.6. RAM is not recognized
In some scenarios, the kernel does not recognize all memory (RAM), which causes the system to use less memory than is installed. You can find out how much RAM is being utilized using the `free -m` command. If the total amount of memory does not match your expectations, it is likely that at least one of your memory modules is faulty. On BIOS-based systems, you can use the `Memtest86+` utility to test your system’s memory.

Some hardware configurations have part of the system’s RAM reserved, and as a result, it is unavailable to the system. Some laptop computers with integrated graphics cards reserve a portion of memory for the GPU. For example, a laptop with 4 GiB of RAM and an integrated Intel graphics card shows roughly 3.7 GiB of available memory. Additionally, the kdump crash kernel dumping mechanism, which is enabled by default on most Red Hat Enterprise Linux systems, reserves some memory for the secondary kernel used in case of a primary kernel failure. This reserved memory is not displayed as available when using the `free` command.

**Procedure: Manually configuring the memory**

Use this procedure to manually set the amount of memory using the `mem=` kernel option.

1. Start your system and wait until the boot loader menu is displayed. If you set your boot timeout period to 0, press the `Esc` key to access it.

2. From the boot loader menu, use your cursor keys to highlight the entry you want to boot, and press the `Tab` key on BIOS-based systems or the `e` key on UEFI-based systems to edit the selected entry options.

3. In the list of options, find the kernel line - that is, the line beginning with the keyword `linux`. Append the following option to the end of this line:

   ```
   mem=xxM
   ```

4. Replace `xx` with the amount of RAM you have in MiB.

5. Press `F10` or `Ctrl`+`X` to boot your system with the edited options.

6. Wait for the system to boot and then log in.

7. Open a command line and execute the free `-m` command again. If the total amount of RAM displayed by the command matches your expectations, append the following to the line beginning with `GRUB_CMDLINE_LINUX` in the `/etc/default/grub` file to make the change permanent:

   ```
   # grub2-mkconfig --output=/boot/grub2/grub.cfg
   ```

**C.7. System is displaying signal 11 errors**

A signal 11 error, commonly known as a segmentation fault means that a program accessed a memory location that it was not assigned. A signal 11 error can occur due to a bug in one of the software programs that are installed, or faulty hardware. If you receive a signal 11 error during the installation process, verify that you are using the most recent installation images and prompt the installation program to verify them to ensure they are not corrupt.

For more information, see [Verifying Boot media](#).

Faulty installation media (such as an improperly burned or scratched optical disk) are a common cause of signal 11 errors. Verifying the integrity of the installation media is recommended before every installation. For information about obtaining the most recent installation media, see [Downloading the](#)
installation ISO image.

To perform a media check before the installation starts, append the \texttt{rd.live.check} boot option at the boot menu. If you performed a media check without any errors and you still have issues with segmentation faults, it usually indicates that your system encountered a hardware error. In this scenario, the problem is most likely in the system’s memory (RAM). This can be a problem even if you previously used a different operating system on the same computer without any errors.

\textbf{NOTE}

For AMD and Intel 64-bit and 64-bit ARM architectures: On BIOS-based systems, you can use the \texttt{Memtest86+} memory testing module included on the installation media to perform a thorough test of your system’s memory.

For more information, see Detecting memory faults using the Memtest86 application.

Other possible causes are beyond this document’s scope. Consult your hardware manufacturer’s documentation and also see the Red Hat Hardware Compatibility List, available online at https://access.redhat.com/ecosystem/search/#/category/Server.

\section*{C.8. Unable to IPL from network storage space}

\textbf{NOTE}

- This issue is for IBM Power Systems.
- The \texttt{PReP} Boot partitions are not required on PowerNV systems.

If you experience difficulties when trying to IPL from Network Storage Space (*NWSSTG), it is most likely due to a missing PReP partition. In this scenario, you must reinstall the system and create this partition during the partitioning phase or in the Kickstart file.

\section*{C.9. Using XDMCP}

There are scenarios where you have installed the X Window System and want to log in to your Red Hat Enterprise Linux system using a graphical login manager. Use this procedure to enable the X Display Manager Control Protocol (XDMCP) and remotely log in to a desktop environment from any X-compatible client, such as a network-connected workstation or X11 terminal.

\textbf{NOTE}

XDMCP is not supported by the Wayland protocol. For more information, see the Using the desktop environment in RHEL document.

\textbf{NOTE}

This issue is for 64-bit IBM Z.

\textbf{Procedure}

1. Open the \texttt{/etc/gdm/custom.conf} configuration file in a plain text editor such as \texttt{vi} or \texttt{nano}. 
2. In the `custom.conf` file, locate the section starting with `[xdmcp]`. In this section, add the following line:

   ```
   Enable=true
   ```

3. Save the file and exit the text editor.

4. Restart the X Window System. To do this, either reboot the system, or restart the GNOME Display Manager using the following command as root:

   ```
   # systemctl restart gdm.service
   ```

5. Wait for the login prompt and log in using your user name and password. The X Window System is now configured for XDMCP. You can connect to it from another workstation (client) by starting a remote X session using the X command on the client workstation. For example:

   ```
   $ X :1 -query address
   ```

6. Replace `address` with the host name of the remote X11 server. The command connects to the remote X11 server using XDMCP and displays the remote graphical login screen on display :1 of the X11 server system (usually accessible by pressing `Ctrl-Alt-F8`). You can also access remote desktop sessions using a nested X11 server, which opens the remote desktop as a window in your current X11 session. You can use Xnest to open a remote desktop nested in a local X11 session. For example, run Xnest using the following command, replacing `address` with the host name of the remote X11 server:

   ```
   $ Xnest :1 -query address
   ```

**Additional resources**

- X Window System documentation

**C.10. Using rescue mode**

The installation program’s rescue mode is a minimal Linux environment that can be booted from the Red Hat Enterprise Linux DVD or other boot media. It contains command-line utilities for repairing a wide variety of issues. Rescue mode can be accessed from the Troubleshooting menu of the boot menu. In this mode, you can mount file systems as read-only, blacklist or add a driver provided on a driver disc, install or upgrade system packages, or manage partitions.

**NOTE**

The installation program’s rescue mode is different from rescue mode (an equivalent to single-user mode) and emergency mode, which are provided as parts of the systemd system and service manager.

To boot into rescue mode, you must be able to boot the system using one of the Red Hat Enterprise Linux boot media, such as a minimal boot disc or USB drive, or a full installation DVD.
IMPORTANT

Advanced storage, such as iSCSI or zFCP devices, must be configured either using **dracut** boot options such as `rd.zfcp=` or `root=iscsi: options`, or in the CMS configuration file on 64-bit IBM Z. It is not possible to configure these storage devices interactively after booting into rescue mode. For information about **dracut** boot options, see the **dracut.cmdline(7)** man page.

C.10.1. Booting into rescue mode

This procedure describes how to boot into rescue mode.

Procedure

1. Boot the system from either minimal boot media, or a full installation DVD or USB drive, and wait for the boot menu to be displayed.

2. From the boot menu, either select **Troubleshooting > Rescue a Red Hat Enterprise Linux system** option, or append the **inst.rescue** option to the boot command line. To enter the boot command line, press the **Tab** key on BIOS-based systems or the **e** key on UEFI-based systems.

3. Optional: If your system requires a third-party driver provided on a driver disc to boot, append the **inst.dd=driver_name** to the boot command line:

   ```
   inst.rescue inst.dd=driver_name
   ```

4. Optional: If a driver that is part of the Red Hat Enterprise Linux distribution prevents the system from booting, append the **modprobe.blacklist=** option to the boot command line:

   ```
   inst.rescue modprobe.blacklist=driver_name
   ```

5. Press **Enter** (BIOS-based systems) or **Ctrl+X** (UEFI-based systems) to boot the modified option. Wait until the following message is displayed:

   The rescue environment will now attempt to find your Linux installation and mount it under the directory: `/mnt/sysroot/`. You can then make any changes required to your system. Choose 1 to proceed with this step. You can choose to mount your file systems read-only instead of read-write by choosing 2. If for some reason this process does not work choose 3 to skip directly to a shell.

   1) Continue
   2) Read-only mount
   3) Skip to shell
   4) Quit (Reboot)

   If you select 1, the installation program attempts to mount your file system under the directory `/mnt/sysroot/`. You are notified if it fails to mount a partition. If you select 2, it attempts to mount your file system under the directory `/mnt/sysroot/`, but in read-only mode. If you select 3, your file system is not mounted.

   For the system root, the installer supports two mount points `/mnt/sysimage` and `/mnt/sysroot`. The `/mnt/sysroot` path is used to mount `/` of the target system. Usually, the physical root and the system root are the same, so `/mnt/sysroot` is attached to the same file system as
The only exceptions are rpm-ostree systems, where the system root changes based on the deployment. Then, /mnt/sysroot is attached to a subdirectory of /mnt/sysimage. It is recommended to use /mnt/sysroot for chroot.

6. Select 1 to continue. Once your system is in rescue mode, a prompt appears on VC (virtual console) 1 and VC 2. Use the Ctrl+Alt+F1 key combination to access VC 1 and Ctrl+Alt+F2 to access VC 2:

   sh-4.2#

7. Even if your file system is mounted, the default root partition while in rescue mode is a temporary root partition, not the root partition of the file system used during normal user mode (multi-user.target or graphical.target). If you selected to mount your file system and it mounted successfully, you can change the root partition of the rescue mode environment to the root partition of your file system by executing the following command:

   sh-4.2# chroot /mnt/sysroot

   This is useful if you need to run commands, such as rpm, that require your root partition to be mounted as /. To exit the chroot environment, type exit to return to the prompt.

8. If you selected 3, you can still try to mount a partition or LVM2 logical volume manually inside rescue mode by creating a directory, such as /directory/, and typing the following command:

   sh-4.2# mount -t xfs /dev/mapper/VolGroup00-LogVol02 /directory

   In the above command, /directory/ is the directory that you created and /dev/mapper/VolGroup00-LogVol02 is the LVM2 logical volume you want to mount. If the partition is a different type than XFS, replace the xfs string with the correct type (such as ext4).

9. If you do not know the names of all physical partitions, use the following command to list them:

   sh-4.2# fdisk -l

   If you do not know the names of all LVM2 physical volumes, volume groups, or logical volumes, use the pvdisplay, vgdisplay or lvdisplay commands.

C.10.2. Using an SOS report in rescue mode

The sosreport command-line utility collects configuration and diagnostic information, such as the running kernel version, loaded modules, and system and service configuration files from the system. The utility output is stored in a tar archive in the /var/tmp/ directory. The sosreport utility is useful for analyzing system errors and troubleshooting. Use this procedure to capture an sosreport output in rescue mode.

**Prerequisites**

- You have booted into rescue mode.
- You have mounted the installed system / (root) partition in read-write mode.
- You have contacted Red Hat Support about your case and received a case number.

**Procedure**
1. Change the root directory to the `/mnt/sysroot/` directory:

   ```bash
   sh-4.2# chroot /mnt/sysroot/
   ```

2. Execute `sosreport` to generate an archive with system configuration and diagnostic information:

   ```bash
   sh-4.2# sosreport
   ```

   **IMPORTANT**
   
   `sosreport` prompts you to enter your name and the case number you received from Red Hat Support. Use only letters and numbers because adding any of the following characters or spaces could render the report unusable:

   ```
   # % & { } 
   ```

3. Optional: If you want to transfer the generated archive to a new location using the network, it is necessary to have a network interface configured. In this scenario, use the dynamic IP addressing as no other steps required. However, when using static addressing, enter the following command to assign an IP address (for example `10.13.153.64/23`) to a network interface, for example `dev eth0`:

   ```bash
   bash-4.2# ip addr add 10.13.153.64/23 dev eth0
   ```

4. Exit the chroot environment:

   ```bash
   sh-4.2# exit
   ```

5. Store the generated archive in a new location, from where it can be easily accessible:

   ```bash
   sh-4.2# cp /mnt/sysroot/var/tmp/sosreport new_location
   ```

6. For transferring the archive through the network, use the `scp` utility:

   ```bash
   sh-4.2# scp /mnt/sysroot/var/tmp/sosreport username@hostname:sosreport
   ```

**Additional resources**

- What is an sosreport and how to create one in Red Hat Enterprise Linux?
- How to generate sosreport from the rescue environment
- How do I make sosreport write to an alternative location?
- Sosreport fails. What data should I provide in its place?

**C.10.3. Reinstalling the GRUB2 boot loader**

In some scenarios, the GRUB2 boot loader is mistakenly deleted, corrupted, or replaced by other operating systems. Use this procedure to reinstall GRUB2 on the master boot record (MBR) on AMD64 and Intel 64 systems with BIOS, or on the little-endian variants of IBM Power Systems with Open
Firmware.

Prerequisites

- You have booted into rescue mode.
- You have mounted the installed system / (root) partition in read-write mode.
- You have mounted the /boot mount point in read-write mode.

Procedure

1. Change the root partition:
   
   sh-4.2# chroot /mnt/sysroot/

2. Reinstall the GRUB2 boot loader, where the install_device block device was installed:

   sh-4.2# /sbin/grub2-install install_device

   **IMPORTANT**

   Running the grub2-install command could lead to the machine being unbootable if all the following conditions apply:

   - The system is an AMD64 or Intel 64 with Extensible Firmware Interface (EFI).
   - Secure Boot is enabled.

   After you run the grub2-install command, you cannot boot the AMD64 or Intel 64 systems that have Extensible Firmware Interface (EFI) and Secure Boot enabled. This issue occurs because the grub2-install command installs an unsigned GRUB2 image that boots directly instead of using the shim application. When the system boots, the shim application validates the image signature, which when not found fails to boot the system.

3. Reboot the system.

C.10.4. Using RPM to add or remove a driver

Missing or malfunctioning drivers cause problems when booting the system. Rescue mode provides an environment in which you can add or remove a driver even when the system fails to boot. Wherever possible, it is recommended that you use the RPM package manager to remove malfunctioning drivers or to add updated or missing drivers. Use the following procedures to add or remove a driver.

**IMPORTANT**

When you install a driver from a driver disc, the driver disc updates all initramfs images on the system to use this driver. If a problem with a driver prevents a system from booting, you cannot rely on booting the system from another initramfs image.

Procedure: Adding a driver using RPM

Use this procedure to add a driver.
Prerequisites

- You have booted into rescue mode.
- You have mounted the installed system in read-write mode.

1. Make the RPM package that contains the driver available. For example, mount a CD or USB flash drive and copy the RPM package to a location of your choice under /mnt/sysroot/, for example: /mnt/sysroot/root/drivers/.

2. Change the root directory to /mnt/sysroot/:
   
   ```
   sh-4.2# chroot /mnt/sysroot/
   ```

3. Use the `rpm -ivh` command to install the driver package. For example, run the following command to install the `xorg-x11-drv-wacom` driver package from /root/drivers/:
   
   ```
   sh-4.2# rpm -ivh /root/drivers/xorg-x11-drv-wacom-0.23.0-6.el7.x86_64.rpm
   ```

   **NOTE**
   
   The /root/drivers/ directory in this chroot environment is the /mnt/sysroot/root/drivers/ directory in the original rescue environment.

4. Exit the chroot environment:
   
   ```
   sh-4.2# exit
   ```

Procedure: Removing a driver using RPM

Use this procedure to remove a driver.

Prerequisites

- You have booted into rescue mode.
- You have mounted the installed system in read-write mode.

1. Change the root directory to the /mnt/sysroot/ directory:
   
   ```
   sh-4.2# chroot /mnt/sysroot/
   ```

2. Use the `rpm -e` command to remove the driver package. For example, to remove the `xorg-x11-drw-wacom` driver package, run:
   
   ```
   sh-4.2# rpm -e xorg-x11-drw-wacom
   ```

3. Exit the chroot environment:
   
   ```
   sh-4.2# exit
   ```

   If you cannot remove a malfunctioning driver for some reason, you can instead blacklist the driver so that it does not load at boot time.
4. When you have finished adding and removing drivers, reboot the system.

C.11. ip= boot option returns an error

Using the ip= boot option format ip=[ip address] for example, ip=192.168.1.1 returns the error message Fatal for argument 'ip=[insert ip here]'

In previous releases of Red Hat Enterprise Linux, the boot option format was:

```
ip=192.168.1.15 netmask=255.255.255.0 gateway=192.168.1.254 nameserver=192.168.1.250 hostname=myhost1
```

However, in Red Hat Enterprise Linux 8, the boot option format is:

```
ip=192.168.1.15::192.168.1.254:255.255.255.0:myhost1::none: nameserver=192.168.1.250
```

To resolve the issue, use the format: ip=ip::gateway:netmask:hostname:interface:none where:

- ip specifies the client ip address. You can specify IPv6 addresses in square brackets, for example, [2001:DB8::1].
- gateway is the default gateway. IPv6 addresses are also accepted.
- netmask is the netmask to be used. This can be either a full netmask, for example, 255.255.255.0, or a prefix, for example, 64.
- hostname is the host name of the client system. This parameter is optional.

Additional resources

- Network boot options

C.12. Cannot boot into the graphical installation on iLO or iDRAC devices

The graphical installer for a remote ISO installation on iLO or iDRAC devices may not be available due to a slow internet connection. To proceed with the installation in this case, you can choose one of the following methods:

1. Avoid the timeout. To do so:
   - Press the Tab key in case of BIOS usage, or the e key in case of UEFI usage when booting from an installation media. That will allow you to modify the kernel command line arguments.
   - To proceed with the installation, append the rd.live.ram=1 and press Enter in case of BIOS usage, or Ctrl+x in case of UEFI usage. This might take longer time to load the installation program.

2. Another option to extend the loading time for the graphical installer is to set the inst.xtimeout kernel argument in seconds:

```
inst.xtimeout=N
```

3. You can install the system in text mode. For more details, see Installing RHEL8 in text mode.
4. In the remote management console, such as iLO or iDRAC, instead of a local media source, use the direct URL to the installation ISO file from the Download center on the Red Hat Customer Portal. You must be logged in to access this section.

C.13. Rootfs image is not initramfs

If you get the following message on the console during booting the installer, the transfer of the installer initrd.img might have had errors:

[...] rootfs image is not initramfs

To resolve this issue, download initrd again or run the sha256sum with initrd.img and compare it with the checksum stored in the .treeinfo file on the installation medium, for example,

```bash
$ sha256sum dvd/images/pxeboot/initrd.img
fd1a70321c06e25a1ed6bf3d8779371b768d5972078eb72b2c78c925067b5d8
dvd/images/pxeboot/initrd.img
```

To view the checksum in .treeinfo:

```bash
$ grep sha256 dvd/.treeinfo
images/efiboot.img = sha256:d357d5063b96226d643c41c9025529554a422acb43a4394e4ebcaa779cc7a917
images/install.img = sha256:8c032357271c04e34dd81c97d008a2d6f62cfc525aef8c31459e21bf3397514
images/pxeboot/initrd.img = sha256:fd1a70321c06e25a1ed6bf3d8779371b768d5972078eb72b2c78c925067b5d8
images/pxeboot/vmlinuz = sha256:b8510ea4212220e85351cbb7f2ebc2b1b0804a6d40cc93307c165e16d1095db
```

Despite having correct initrd.img, if you get the following kernel messages during booting the installer, often a boot parameter is missing or mis-spelled, and the installer could not load stage2, typically referred to by the inst.repo= parameter, providing the full installer initial ramdisk for its in-memory root file system:

```bash
[...] No filesystem could mount root, tried:
[...] Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(1,0)
[...] CPU: 0 PID: 1 Comm: swapper/0 Not tainted 5.14.0-55.el9.s390x #1
[...] ...
[...] Call Trace:
[...] [<-->] show_trace+0x.../0x...
[...] [<-->] show_stack+0x.../0x...
[...] [<-->] panic+0x.../0x...
[...] [<-->] mount_block_root+0x.../0x...
[...] [<-->] prepare_namespace+0x.../0x...
[...] [<-->] kernel_init_freeable+0x.../0x...
[...] [<-->] kernel_init+0x.../0x...
[...] [<-->] kernel_thread Starter+0x.../0x...
[...] [<-->] kernel_thread Starter+0x.../0x...
```

To resolve this issue, check

- if the installation source specified is correct on the kernel command line (inst.repo=) or in the kickstart file
• the network configuration is specified on the kernel command line (if the installation source is specified as network)
• the network installation source is accessible from another system
APPENDIX D. SYSTEM REQUIREMENTS REFERENCE

This section provides information and guidelines for hardware, installation target, system, memory, and RAID when installing Red Hat Enterprise Linux.

D.1. HARDWARE COMPATIBILITY

Red Hat works closely with hardware vendors on supported hardware.

- To verify that your hardware is supported, see the Red Hat Hardware Compatibility List, available at https://access.redhat.com/ecosystem/search/#/category/Server.

- To view supported memory sizes or CPU counts, see https://access.redhat.com/articles/rhel-limits for information.

D.2. SUPPORTED INSTALLATION TARGETS

An installation target is a storage device that stores Red Hat Enterprise Linux and boots the system. Red Hat Enterprise Linux supports the following installation targets for AMD64, Intel 64, and 64-bit ARM systems:

- Storage connected by a standard internal interface, such as SCSI, SATA, or SAS

- BIOS/firmware RAID devices

- NVDIMM devices in sector mode on the Intel64 and AMD64 architectures, supported by the nd_pmem driver.

- Fibre Channel Host Bus Adapters and multipath devices. Some can require vendor-provided drivers.

- Xen block devices on Intel processors in Xen virtual machines.

- VirtIO block devices on Intel processors in KVM virtual machines.

Red Hat does not support installation to USB drives or SD memory cards. For information about support for third-party virtualization technologies, see the Red Hat Hardware Compatibility List.

D.3. SYSTEM SPECIFICATIONS

The Red Hat Enterprise Linux installation program automatically detects and installs your system’s hardware, so you should not have to supply any specific system information. However, for certain Red Hat Enterprise Linux installation scenarios, it is recommended that you record system specifications for future reference. These scenarios include:

Installing RHEL with a customized partition layout

Record: The model numbers, sizes, types, and interfaces of the hard drives attached to the system. For example, Seagate ST3320613AS 320 GB on SATA0, Western Digital WD7500AAKS 750 GB on SATA1.

Installing RHEL as an additional operating system on an existing system

Record: Partitions used on the system. This information can include file system types, device node names, file system labels, and sizes, and allows you to identify specific partitions during the partitioning process. If one of the operating systems is a Unix operating system, Red Hat Enterprise Linux may
report the device names differently. Additional information can be found by executing the equivalent of the `mount` command and the `blkid` command, and in the `/etc/fstab` file.

If multiple operating systems are installed, the Red Hat Enterprise Linux installation program attempts to automatically detect them, and to configure boot loader to boot them. You can manually configure additional operating systems if they are not detected automatically.

See Configuring boot loader in Configuring software settings for more information.

**Installing RHEL from an image on a local hard drive**

**Record:** The hard drive and directory that holds the image.

**Installing RHEL from a network location**

If the network has to be configured manually, that is, DHCP is not used.

**Record:**

- IP address
- Netmask
- Gateway IP address
- Server IP addresses, if required

Contact your network administrator if you need assistance with networking requirements.

**Installing RHEL on an iSCSI target**

**Record:** The location of the iSCSI target. Depending on your network, you may need a CHAP user name and password, and a reverse CHAP user name and password.

**Installing RHEL if the system is part of a domain**

Verify that the domain name is supplied by the DHCP server. If it is not, enter the domain name during installation.

**D.4. DISK AND MEMORY REQUIREMENTS**

If several operating systems are installed, it is important that you verify that the allocated disk space is separate from the disk space required by Red Hat Enterprise Linux.

**Note**

- For AMD64, Intel 64, and 64-bit ARM, at least two partitions (`/` and `swap`) must be dedicated to Red Hat Enterprise Linux.
- For IBM Power Systems servers, at least three partitions (`/`, `swap`, and a PReP boot partition) must be dedicated to Red Hat Enterprise Linux.
- The Prep Boot partitions are not required on PowerNV systems.

You must have a minimum of 10 GiB of available disk space. To install Red Hat Enterprise Linux, you must have a minimum of 10 GiB of space in either unpartitioned disk space or in partitions that can be deleted.
See Partitioning reference

Table D.1. Minimum RAM requirements

<table>
<thead>
<tr>
<th>Installation type</th>
<th>Recommended minimum RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local media installation (USB, DVD)</td>
<td>● 1.5 GiB for aarch64, s390x and x86_64 architectures</td>
</tr>
<tr>
<td></td>
<td>● 3 GiB for ppc64le architecture</td>
</tr>
<tr>
<td>NFS network installation</td>
<td>● 1.5 GiB for aarch64, s390x and x86_64 architectures</td>
</tr>
<tr>
<td></td>
<td>● 3 GiB for ppc64le architecture</td>
</tr>
<tr>
<td>HTTP, HTTPS or FTP network installation</td>
<td>● 3 GiB for s390x and x86_64 architectures</td>
</tr>
<tr>
<td></td>
<td>● 4 GiB for aarch64 and ppc64le architectures</td>
</tr>
</tbody>
</table>

**NOTE**

It is possible to complete the installation with less memory than the recommended minimum requirements. The exact requirements depend on your environment and installation path. It is recommended that you test various configurations to determine the minimum required RAM for your environment. Installing Red Hat Enterprise Linux using a Kickstart file has the same recommended minimum RAM requirements as a standard installation. However, additional RAM may be required if your Kickstart file includes commands that require additional memory, or write data to the RAM disk. See the Performing an advanced RHEL installation document for more information.

D.5. UEFI SECURE BOOT AND BETA RELEASE REQUIREMENTS

If you plan to install a Beta release of Red Hat Enterprise Linux, on systems having UEFI Secure Boot enabled, then first disable the UEFI Secure Boot option and then begin the installation.

UEFI Secure Boot requires that the operating system kernel is signed with a recognized private key, which the system’s firmware verifies using the corresponding public key. For Red Hat Enterprise Linux Beta releases, the kernel is signed with a Red Hat Beta-specific public key, which the system fails to recognize by default. As a result, the system fails to even boot the installation media.
E.1. SUPPORTED DEVICE TYPES

Standard partition
A standard partition can contain a file system or swap space. Standard partitions are most commonly used for /boot and the BIOS Boot and EFI System partitions. LVM logical volumes are recommended for most other uses.

LVM
Choosing LVM (or Logical Volume Management) as the device type creates an LVM logical volume. LVM can improve performance when using physical disks, and it allows for advanced setups such as using multiple physical disks for one mount point, and setting up software RAID for increased performance, reliability, or both.

LVM thin provisioning
Using thin provisioning, you can manage a storage pool of free space, known as a thin pool, which can be allocated to an arbitrary number of devices when needed by applications. You can dynamically expand the pool when needed for cost-effective allocation of storage space.

WARNING
The installation program does not support overprovisioned LVM thin pools.

E.2. SUPPORTED FILE SYSTEMS

This section describes the file systems available in Red Hat Enterprise Linux.

xfs
XFS is a highly scalable, high-performance file system that supports file systems up to 16 exabytes (approximately 16 million terabytes), files up to 8 exabytes (approximately 8 million terabytes), and directory structures containing tens of millions of entries. XFS also supports metadata journaling, which facilitates quicker crash recovery. The maximum supported size of a single XFS file system is 500 TB. XFS is the default and recommended file system on Red Hat Enterprise Linux. The XFS filesystem cannot be shrunk to get free space.

ext4
The ext4 file system is based on the ext3 file system and features a number of improvements. These include support for larger file systems and larger files, faster and more efficient allocation of disk space, no limit on the number of subdirectories within a directory, faster file system checking, and more robust journaling. The maximum supported size of a single ext4 file system is 50 TB.

ext3
The ext3 file system is based on the ext2 file system and has one main advantage - journaling. Using a journaling file system reduces the time spent recovering a file system after it terminates unexpectedly, as there is no need to check the file system for metadata consistency by running the fsck utility every time.

ext2
An ext2 file system supports standard Unix file types, including regular files, directories, or symbolic links. It provides the ability to assign long file names, up to 255 characters.

**swap**
Swap partitions are used to support virtual memory. In other words, data is written to a swap partition when there is not enough RAM to store the data your system is processing.

**vfat**
The VFAT file system is a Linux file system that is compatible with Microsoft Windows long file names on the FAT file system.

**NOTE**
Support for VFAT file system is not available for Linux system partitions. For example, `/`, `/var`, `/usr` and so on.

**BIOS Boot**
A very small partition required for booting from a device with a GUID partition table (GPT) on BIOS systems and UEFI systems in BIOS compatibility mode.

**EFI System Partition**
A small partition required for booting a device with a GUID partition table (GPT) on a UEFI system.

**PReP**
This small boot partition is located on the first partition of the hard drive. The PReP boot partition contains the GRUB2 boot loader, which allows other IBM Power Systems servers to boot Red Hat Enterprise Linux.

**NOTE**
- The PReP Boot partitions are not required on PowerNV systems.

### E.3. SUPPORTED RAID TYPES

RAID stands for Redundant Array of Independent Disks, a technology which allows you to combine multiple physical disks into logical units. Some setups are designed to enhance performance at the cost of reliability, while others improve reliability at the cost of requiring more disks for the same amount of available space.

This section describes supported software RAID types which you can use with LVM and LVM Thin Provisioning to set up storage on the installed system.

**RAID 0**
- Performance: Distributes data across multiple disks. RAID 0 offers increased performance over standard partitions and can be used to pool the storage of multiple disks into one large virtual device.
- Note that RAID 0 offers no redundancy and that the failure of one device in the array destroys data in the entire array. RAID 0 requires at least two disks.

**RAID 1**
- Redundancy: Mirrors all data from one partition onto one or more other disks. Additional devices in the array provide increasing levels of redundancy. RAID 1 requires at least two disks.

**RAID 4**
- Error checking: Distributes data across multiple disks and uses one disk in the array to store parity
information which safeguards the array in case any disk in the array fails. As all parity information is stored on one disk, access to this disk creates a “bottleneck” in the array’s performance. RAID 4 requires at least three disks.

**RAID 5**

Distributed error checking: Distributes data and parity information across multiple disks. RAID 5 offers the performance advantages of distributing data across multiple disks, but does not share the performance bottleneck of RAID 4 as the parity information is also distributed through the array. RAID 5 requires at least three disks.

**RAID 6**

Redundant error checking: RAID 6 is similar to RAID 5, but instead of storing only one set of parity data, it stores two sets. RAID 6 requires at least four disks.

**RAID 10**

Performance and redundancy: RAID 10 is nested or hybrid RAID. It is constructed by distributing data over mirrored sets of disks. For example, a RAID 10 array constructed from four RAID partitions consists of two mirrored pairs of striped partitions. RAID 10 requires at least four disks.

### E.4. RECOMMENDED PARTITIONING SCHEME

Red Hat recommends that you create separate file systems at the following mount points. However, if required, you can also create the file systems at `/usr`, `/var`, and `/tmp` mount points.

- `/boot`
- `/ (root)`
- `/home`
- `swap`
- `/boot/efi`
- `PReP`

This partition scheme is recommended for bare metal deployments and it does not apply to virtual and cloud deployments.

**/boot partition - recommended size at least 1 GiB**

The partition mounted on `/boot` contains the operating system kernel, which allows your system to boot Red Hat Enterprise Linux 8, along with files used during the bootstrap process. Due to the limitations of most firmwares, creating a small partition to hold these is recommended. In most scenarios, a 1 GiB boot partition is adequate. Unlike other mount points, using an LVM volume for `/boot` is not possible – `/boot` must be located on a separate disk partition.

![WARNING]

Normally, the `/boot` partition is created automatically by the installation program. However, if the `/ (root)` partition is larger than 2 TiB and (U)EFI is used for booting, you need to create a separate `/boot` partition that is smaller than 2 TiB to boot the machine successfully.
NOTE

If you have a RAID card, be aware that some BIOS types do not support booting from the RAID card. In such a case, the /boot partition must be created on a partition outside of the RAID array, such as on a separate hard drive.

root - recommended size of 10 GiB

This is where "/", or the root directory, is located. The root directory is the top-level of the directory structure. By default, all files are written to this file system unless a different file system is mounted in the path being written to, for example, /boot or /home.

While a 5 GiB root file system allows you to install a minimal installation, it is recommended to allocate at least 10 GiB so that you can install as many package groups as you want.

IMPORTANT

Do not confuse the / directory with the /root directory. The /root directory is the home directory of the root user. The /root directory is sometimes referred to as slash root to distinguish it from the root directory.

/home - recommended size at least 1 GiB

To store user data separately from system data, create a dedicated file system for the /home directory. Base the file system size on the amount of data that is stored locally, number of users, and so on. You can upgrade or reinstall Red Hat Enterprise Linux 8 without erasing user data files. If you select automatic partitioning, it is recommended to have at least 55 GiB of disk space available for the installation, to ensure that the /home file system is created.

swap partition - recommended size at least 1 GiB

Swap file systems support virtual memory; data is written to a swap file system when there is not enough RAM to store the data your system is processing. Swap size is a function of system memory workload, not total system memory and therefore is not equal to the total system memory size. It is important to analyze what applications a system will be running and the load those applications will serve in order to determine the system memory workload. Application providers and developers can provide guidance.

When the system runs out of swap space, the kernel terminates processes as the system RAM memory is exhausted. Configuring too much swap space results in storage devices being allocated but idle and is a poor use of resources. Too much swap space can also hide memory leaks. The maximum size for a swap partition and other additional information can be found in the mkswap(8) manual page.

The following table provides the recommended size of a swap partition depending on the amount of RAM in your system and if you want sufficient memory for your system to hibernate. If you let the installation program partition your system automatically, the swap partition size is established using these guidelines. Automatic partitioning setup assumes hibernation is not in use. The maximum size of the swap partition is limited to 10 percent of the total size of the hard drive, and the installation program cannot create swap partitions more than 1TiB. To set up enough swap space to allow for hibernation, or if you want to set the swap partition size to more than 10 percent of the system's storage space, or more than 1TiB, you must edit the partitioning layout manually.

<table>
<thead>
<tr>
<th>RAM Size</th>
<th>Recommended Swap Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;512 GB</td>
<td>2 GB</td>
</tr>
<tr>
<td>512 GB - 1 TiB</td>
<td>3 GB</td>
</tr>
<tr>
<td>&gt;1 TiB</td>
<td>4 GB</td>
</tr>
</tbody>
</table>

Table E.1. Recommended system swap space
<table>
<thead>
<tr>
<th>Amount of RAM in the system</th>
<th>Recommended swap space</th>
<th>Recommended swap space if allowing for hibernation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than 2 GiB</td>
<td>2 times the amount of RAM</td>
<td>3 times the amount of RAM</td>
</tr>
<tr>
<td>2 GiB - 8 GiB</td>
<td>Equal to the amount of RAM</td>
<td>2 times the amount of RAM</td>
</tr>
<tr>
<td>8 GiB - 64 GiB</td>
<td>4 GiB to 0.5 times the amount of RAM</td>
<td>1.5 times the amount of RAM</td>
</tr>
<tr>
<td>More than 64 GiB</td>
<td>Workload dependent (at least 4GiB)</td>
<td>Hibernation not recommended</td>
</tr>
</tbody>
</table>

/\texttt{boot/efi} partition - recommended size of 200 MiB

UEFI-based AMD64, Intel 64, and 64-bit ARM require a 200 MiB EFI system partition. The recommended minimum size is 200 MiB, the default size is 600 MiB, and the maximum size is 600 MiB. BIOS systems do not require an EFI system partition.

At the border between each range, for example, a system with 2 GiB, 8 GiB, or 64 GiB of system RAM, discretion can be exercised with regard to chosen swap space and hibernation support. If your system resources allow for it, increasing the swap space can lead to better performance.

Distributing swap space over multiple storage devices – particularly on systems with fast drives, controllers and interfaces – also improves swap space performance.

Many systems have more partitions and volumes than the minimum required. Choose partitions based on your particular system needs.

**NOTE**

- Only assign storage capacity to those partitions you require immediately. You can allocate free space at any time, to meet needs as they occur.

- If you are unsure about how to configure partitions, accept the automatic default partition layout provided by the installation program.

PReP boot partition - recommended size of 4 to 8 MiB

When installing Red Hat Enterprise Linux on IBM Power System servers, the first partition of the hard drive should include a PReP boot partition. This contains the GRUB2 boot loader, which allows other IBM Power Systems servers to boot Red Hat Enterprise Linux.

**NOTE**

- The PReP Boot partitions are not required on PowerNV systems.

### E.5. ADVICE ON PARTITIONS

There is no best way to partition every system; the optimal setup depends on how you plan to use the system being installed. However, the following tips may help you find the optimal layout for your needs:
• Create partitions that have specific requirements first, for example, if a particular partition must be on a specific disk.

• Consider encrypting any partitions and volumes which might contain sensitive data. Encryption prevents unauthorized people from accessing the data on the partitions, even if they have access to the physical storage device. In most cases, you should at least encrypt the /home partition, which contains user data.

• In some cases, creating separate mount points for directories other than /, /boot and /home may be useful; for example, on a server running a MySQL database, having a separate mount point for /var/lib/mysql allows you to preserve the database during a re-installation without having to restore it from backup afterward. However, having unnecessary separate mount points will make storage administration more difficult.

• Some special restrictions apply to certain directories with regards on which partitioning layouts can they be placed. Notably, the /boot directory must always be on a physical partition (not on an LVM volume).

• If you are new to Linux, consider reviewing the Linux Filesystem Hierarchy Standard for information about various system directories and their contents.

• Each kernel requires approximately: 60MiB (initrd 34MiB, 11MiB vmlinuz, and 5MiB System.map)

• For rescue mode: 100MiB (initrd 76MiB, 11MiB vmlinuz, and 5MiB System map)

• When kdump is enabled in system it will take approximately another 40MiB (another initrd with 33MiB)
  The default partition size of 1 GiB for /boot should suffice for most common use cases. However, it is recommended that you increase the size of this partition if you are planning on retaining multiple kernel releases or errata kernels.

• The /var directory holds content for a number of applications, including the Apache web server, and is used by the YUM package manager to temporarily store downloaded package updates. Make sure that the partition or volume containing /var has at least 3 GiB.

• The /usr directory holds the majority of software on a typical Red Hat Enterprise Linux installation. The partition or volume containing this directory should therefore be at least 5 GiB for minimal installations, and at least 10 GiB for installations with a graphical environment.

• If /usr or /var is partitioned separately from the rest of the root volume, the boot process becomes much more complex because these directories contain boot-critical components. In some situations, such as when these directories are placed on an iSCSI drive or an FCoE location, the system may either be unable to boot, or it may hang with a Device is busy error when powering off or rebooting.
  This limitation only applies to /usr or /var, not to directories under them. For example, a separate partition for /var/www works without issues.

  IMPORTANT

Some security policies require the separation of /usr and /var, even though it makes administration more complex.

• Consider leaving a portion of the space in an LVM volume group unallocated. This unallocated space gives you flexibility if your space requirements change but you do not wish to remove data from other volumes. You can also select the LVM Thin Provisioning device type for the partition to have the unused space handled automatically by the volume.
- The size of an XFS file system cannot be reduced - if you need to make a partition or volume with this file system smaller, you must back up your data, destroy the file system, and create a new, smaller one in its place. Therefore, if you plan to alter your partitioning layout later, you should use the ext4 file system instead.

- Use Logical Volume Management (LVM) if you anticipate expanding your storage by adding more hard drives or expanding virtual machine hard drives after the installation. With LVM, you can create physical volumes on the new drives, and then assign them to any volume group and logical volume as you see fit - for example, you can easily expand your system’s /home (or any other directory residing on a logical volume).

- Creating a BIOS Boot partition or an EFI System Partition may be necessary, depending on your system’s firmware, boot drive size, and boot drive disk label. Note that you cannot create a BIOS Boot or EFI System Partition in graphical installation if your system does not require one - in that case, they are hidden from the menu.

- If you need to make any changes to your storage configuration after the installation, Red Hat Enterprise Linux repositories offer several different tools which can help you do this. If you prefer a command-line tool, try system-storage-manager.

Additional resources

- How to use dm-crypt on IBM Z, LinuxONE and with the PAES cipher

### E.6. SUPPORTED HARDWARE STORAGE

It is important to understand how storage technologies are configured and how support for them may have changed between major versions of Red Hat Enterprise Linux.

**Hardware RAID**

Any RAID functions provided by the mainboard of your computer, or attached controller cards, need to be configured before you begin the installation process. Each active RAID array appears as one drive within Red Hat Enterprise Linux.

**Software RAID**

On systems with more than one hard drive, you can use the Red Hat Enterprise Linux installation program to operate several of the drives as a Linux software RAID array. With a software RAID array, RAID functions are controlled by the operating system rather than the dedicated hardware.

**NOTE**

When a pre-existing RAID array’s member devices are all unpartitioned disks/drives, the installation program treats the array as a disk and there is no method to remove the array.

**USB Disks**

You can connect and configure external USB storage after installation. Most devices are recognized by the kernel, but some devices may not be recognized. If it is not a requirement to configure these disks during installation, disconnect them to avoid potential problems.

**NVDIMM devices**

To use a Non-Volatile Dual In-line Memory Module (NVDIMM) device as storage, the following conditions must be satisfied:
- Version of Red Hat Enterprise Linux is 7.6 or later.
- The architecture of the system is Intel 64 or AMD64.
- The device is configured to sector mode. Anaconda can reconfigure NVDIMM devices to this mode.
- The device must be supported by the nd_pmem driver.

Booting from an NVDIMM device is possible under the following additional conditions:

- The system uses UEFI.
- The device must be supported by firmware available on the system, or by a UEFI driver. The UEFI driver may be loaded from an option ROM of the device itself.
- The device must be made available under a namespace.

To take advantage of the high performance of NVDIMM devices during booting, place the /boot and /boot/efi directories on the device.

**NOTE**

The Execute-in-place (XIP) feature of NVDIMM devices is not supported during booting and the kernel is loaded into conventional memory.

**Considerations for Intel BIOS RAID Sets**

Red Hat Enterprise Linux uses `mdraid` for installing on Intel BIOS RAID sets. These sets are automatically detected during the boot process and their device node paths can change across several booting processes. It is recommended that you replace device node paths (such as /dev/sda) with file system labels or device UUIDs. You can find the file system labels and device UUIDs using the `blkid` command.
This section contains information about some of the boot options that you can use to modify the default behavior of the installation program. For Kickstart and advanced boot options, see the Performing an advanced RHEL installation document.

F.1. INSTALLATION SOURCE BOOT OPTIONS

This section contains information about the various installation source boot options.

**inst.repo=**

The `inst.repo=` boot option specifies the installation source, that is, the location providing the package repositories and a valid `.treeinfo` file that describes them. For example: `inst.repo=cdrom`. The target of the `inst.repo=` option must be one of the following installation media:

- an installable tree, which is a directory structure containing the installation program images, packages, and repository data as well as a valid `.treeinfo` file
- a DVD (a physical disk present in the system DVD drive)
- an ISO image of the full Red Hat Enterprise Linux installation DVD, placed on a hard drive or a network location accessible to the system.

Use the `inst.repo=` boot option to configure different installation methods using different formats. The following table contains details of the `inst.repo=` boot option syntax:

### Table F.1. `inst.repo=` installation source boot options

<table>
<thead>
<tr>
<th>Source type</th>
<th>Boot option format</th>
<th>Source format</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD/DVD drive</td>
<td><code>inst.repo=cdrom:&lt;device&gt;</code></td>
<td>Installation DVD as a physical disk. [a]</td>
</tr>
<tr>
<td>Mountable device</td>
<td><code>inst.repo=hd:&lt;device&gt;:/&lt;path&gt;</code></td>
<td>Image file of the installation DVD.</td>
</tr>
<tr>
<td>NFS Server</td>
<td><code>inst.repo=nfs:[options:]&lt;server&gt;:/&lt;path&gt;</code></td>
<td>Image file of the installation DVD, or an installation tree, which is a complete copy of the directories and files on the installation DVD. [b]</td>
</tr>
<tr>
<td>HTTP Server</td>
<td><code>inst.repo=http://&lt;host&gt;:/&lt;path&gt;</code></td>
<td>Installation tree, which is a complete copy of the directories and files on the installation DVD.</td>
</tr>
<tr>
<td>HTTPS Server</td>
<td><code>inst.repo=https://&lt;host&gt;:/&lt;path&gt;</code></td>
<td></td>
</tr>
<tr>
<td>FTP Server</td>
<td><code>inst.repo=ftp://&lt;username&gt;:@&lt;host&gt;:/&lt;password&gt;/&lt;path&gt;</code></td>
<td></td>
</tr>
<tr>
<td>HMC</td>
<td><code>inst.repo=hmc</code></td>
<td></td>
</tr>
</tbody>
</table>
[a] If device is left out, installation program automatically searches for a drive containing the installation DVD.

[b] The NFS Server option uses NFS protocol version 3 by default. To use a different version, add `nfsvers=X` to `options`, replacing X with the version number that you want to use.

Set disk device names with the following formats:

- Kernel device name, for example `/dev/sda1` or `sdb2`
- File system label, for example `LABEL=Flash` or `LABEL=RHEL8`
- File system UUID, for example `UUID=8176c7bf-04ff-403a-a832-9557f94e61db`  
  Non-alphanumeric characters must be represented as `\xNN`, where `NN` is the hexadecimal representation of the character. For example, `\x20` is a white space (" ").

**inst.addrepo=**

Use the `inst.addrepo=` boot option to add an additional repository that can be used as another installation source along with the main repository (`inst.repo=`). You can use the `inst.addrepo=` boot option multiple times during one boot. The following table contains details of the `inst.addrepo=` boot option syntax.

**NOTE**

The `REPO_NAME` is the name of the repository and is required in the installation process. These repositories are only used during the installation process; they are not installed on the installed system.

For more information about unified ISO, see **Unified ISO**

**Table F.2. inst.addrepo installation source boot options**

<table>
<thead>
<tr>
<th>Installation source</th>
<th>Boot option format</th>
<th>Additional information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installable tree at a URL</td>
<td><code>inst.addrepo=REPO_NAME, [http,https,ftp]://&lt;host&gt;/&lt;path&gt;</code></td>
<td>Looks for the installable tree at a given URL.</td>
</tr>
<tr>
<td>Installable tree at an NFS path</td>
<td><code>inst.addrepo=REPO_NAME,nfs://&lt;server&gt;/&lt;path&gt;</code></td>
<td>Looks for the installable tree at a given NFS path. A colon is required after the host. The installation program passes everything after <code>nfs://</code> directly to the mount command instead of parsing URLs according to RFC 2224.</td>
</tr>
</tbody>
</table>
### Installation source

<table>
<thead>
<tr>
<th>Boot option format</th>
<th>Additional information</th>
</tr>
</thead>
<tbody>
<tr>
<td>inst.addrepo=REPO_NAME,file://&lt;path&gt;</td>
<td>Looks for the installable tree at the given location in the installation environment. To use this option, the repository must be mounted before the installation program attempts to load the available software groups. The benefit of this option is that you can have multiple repositories on one bootable ISO, and you can install both the main repository and additional repositories from the ISO. The path to the additional repositories is /run/install/source/REPO_ISO_PATH. Additional, you can mount the repository directory in the %pre section in the Kickstart file. The path must be absolute and start with /, for example inst.addrepo=REPO_NAME,file:///&lt;path&gt;</td>
</tr>
</tbody>
</table>

| inst.addrepo=REPO_NAME,h d:<device>:<path> | Mounts the given <device> partition and installs from the ISO that is specified by the <path>. If the <path> is not specified, the installation program looks for a valid installation ISO on the <device>. This installation method requires an ISO with a valid installable tree. |

### inst.stage2=

The inst.stage2= boot option specifies the location of the installation program’s runtime image. This option expects the path to a directory that contains a valid .treeinfo file and reads the runtime image location from the .treeinfo file. If the .treeinfo file is not available, the installation program attempts to load the image from images/install.img. When the inst.stage2 option is not specified, the installation program attempts to use the location specified with the inst.repo option.

Use this option when you want to manually specify the installation source in the installation program at a later time. For example, when you want to select the Content Delivery Network (CDN) as an installation source. The installation DVD and Boot ISO already contain a correct inst.stage2 option to boot the installation program from the respective ISO.

If you want to specify an installation source, use the inst.repo= option instead.
NOTE

By default, the `inst.stage2=` boot option is used on the installation media and is set to a specific label; for example, `inst.stage2=hd:LABEL=RHEL-x-0-0-BaseOS-x86_64`. If you modify the default label of the file system that contains the runtime image, or if you use a customized procedure to boot the installation system, verify that the `inst.stage2=` boot option is set to the correct value.

`inst.noverifyssl`

Use the `inst.noverifyssl` boot option to prevent the installer from verifying SSL certificates for all HTTPS connections with the exception of additional Kickstart repositories, where `--noverifyssl` can be set per repository.

For example, if your remote installation source is using self-signed SSL certificates, the `inst.noverifyssl` boot option enables the installer to complete the installation without verifying the SSL certificates.

**Example when specifying the source using `inst.stage2=`**

```
inst.stage2=https://hostname/path_to_install_image/ inst.noverifyssl
```

**Example when specifying the source using `inst.repo=`**

```
inst.repo=https://hostname/path_to_install_repository/ inst.noverifyssl
```

`inst.stage2.all`

The `inst.stage2.all` boot option is used to specify several HTTP, HTTPS, or FTP sources. You can use the `inst.stage2=` boot option multiple times with the `inst.stage2.all` option to fetch the image from the sources sequentially until one succeeds. For example:

```
inst.stage2.all
inst.stage2=http://hostname1/path_to_install_tree/
inst.stage2=http://hostname2/path_to_install_tree/
inst.stage2=http://hostname3/path_to_install_tree/
```

`inst.dd=`

The `inst.dd=` boot option is used to perform a driver update during the installation. For more information on how to update drivers during installation, see the *Performing an advanced RHEL installation* document.

`inst.repo=hmc`

When booting from a Binary DVD, the installation program prompts you to enter additional kernel parameters. To set the DVD as an installation source, append the `inst.repo=hmc` option to the kernel parameters. The installation program then enables support element (SE) and hardware management console (HMC) file access, fetches the images for stage2 from the DVD, and provides access to the packages on the DVD for software selection. This option eliminates the requirement of an external network setup and expands the installation options.

`inst.proxy=`

The `inst.proxy=` boot option is used when performing an installation from a HTTP, HTTPS, and FTP protocol. For example:

```
[PROTOCOL://][USERNAME[:PASSWORD]@]HOST[:PORT]
```
**inst.nosave**

Use the `inst.nosave` boot option to control the installation logs and related files that are not saved to the installed system, for example `input_ks`, `output_ks`, `all_ks`, `logs` and `all`. Multiple values can be combined as a comma-separated list, for example: `input_ks,logs`.

**NOTE**

The `inst.nosave` boot option is used for excluding files from the installed system that can't be removed by a Kickstart `%post` script, such as logs and input/output Kickstart results.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>input_ks</code></td>
<td>Disables the ability to save the input Kickstart results.</td>
</tr>
<tr>
<td><code>output_ks</code></td>
<td>Disables the ability to save the output Kickstart results generated by the installation program.</td>
</tr>
<tr>
<td><code>all_ks</code></td>
<td>Disables the ability to save the input and output Kickstart results.</td>
</tr>
<tr>
<td><code>logs</code></td>
<td>Disables the ability to save all installation logs.</td>
</tr>
<tr>
<td><code>all</code></td>
<td>Disables the ability to save all Kickstart results, and all logs.</td>
</tr>
</tbody>
</table>

**inst.multilib**

Use the `inst.multilib` boot option to set DNF's `multilib_policy` to `all`, instead of `best`.

**inst.memcheck**

The `inst.memcheck` boot option performs a check to verify that the system has enough RAM to complete the installation. If there isn't enough RAM, the installation process is stopped. The system check is approximate and memory usage during installation depends on the package selection, user interface, for example graphical or text, and other parameters.

**inst.nomemcheck**

The `inst.nomemcheck` boot option does not perform a check to verify if the system has enough RAM to complete the installation. Any attempt to perform the installation with less than the recommended minimum amount of memory is unsupported, and might result in the installation process failing.

**F.2. NETWORK BOOT OPTIONS**

This section contains information about commonly used network boot options.
NOTE

Initial network initialization is handled by `dracut`. For a complete list, see the `dracut.cmdline(7)` man page.

`ip=`

Use the `ip=` boot option to configure one or more network interfaces. To configure multiple interfaces, you can use the `ip` option multiple times, once for each interface; to do so, you must use the `rd.neednet=1` option, and you must specify a primary boot interface using the `bootdev` option. Alternatively, you can use the `ip` option once, and then use Kickstart to set up further interfaces. This option accepts several different formats. The following tables contain information about the most common options.

NOTE

In the following tables:

- The `ip` parameter specifies the client IP address and requires square brackets, for example `[2001:db8::99]`.
- The `gateway` parameter is the default gateway. IPv6 addresses are also accepted.
- The `netmask` parameter is the netmask to be used. This can be either a full netmask (for example, 255.255.255.0) or a prefix (for example, 64).
- The `hostname` parameter is the host name of the client system. This parameter is optional.

Table F.4. Network interface configuration boot option formats

<table>
<thead>
<tr>
<th>Configuration method</th>
<th>Boot option format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic configuration of any interface</td>
<td><code>ip=method</code></td>
</tr>
<tr>
<td>Automatic configuration of a specific interface</td>
<td><code>ip=interface:method</code></td>
</tr>
<tr>
<td>Static configuration</td>
<td><code>ip=ip::gateway:netmask:hostname:interface:none</code></td>
</tr>
</tbody>
</table>

NOTE

The method `automatic configuration of a specific interface with an override` brings up the interface using the specified method of automatic configuration, such as `dhcp`, but overrides the automatically-obtained IP address, gateway, netmask, host name or other specified parameters. All parameters are optional, so specify only the parameters that you want to override.

The `method` parameter can be any of the following:
Table F.5. Automatic interface configuration methods

<table>
<thead>
<tr>
<th>Automatic configuration method</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHCP</td>
<td>dhcp</td>
</tr>
<tr>
<td>IPv6 DHCP</td>
<td>dhcp6</td>
</tr>
<tr>
<td>IPv6 automatic configuration</td>
<td>auto6</td>
</tr>
<tr>
<td>iSCSI Boot Firmware Table (iBFT)</td>
<td>ibft</td>
</tr>
</tbody>
</table>

**NOTE**
- If you use a boot option that requires network access, such as `inst.ks=http://host/path`, without specifying the `ip` option, the installation program uses `ip=dhcp`.
- To connect to an iSCSI target automatically, you must activate a network device for accessing the target. The recommended way to activate a network is to use the `ip=ibft` boot option.

**nameserver=**

The `nameserver=` option specifies the address of the name server. You can use this option multiple times.

**NOTE**

The `ip=` parameter requires square brackets. However, an IPv6 address does not work with square brackets. An example of the correct syntax to use for an IPv6 address is `nameserver=2001:db8::1`.

**bootdev=**

The `bootdev=` option specifies the boot interface. This option is mandatory if you use more than one `ip` option.

**ifname=**

The `ifname=` options assigns an interface name to a network device with a given MAC address. You can use this option multiple times. The syntax is `ifname=interface:MAC`. For example:

```
ifname=eth0:01:23:45:67:89:ab
```

**NOTE**

The `ifname=` option is the only supported way to set custom network interface names during installation.

**inst.dhcpclass=**
The `inst.dhcpclass=` option specifies the DHCP vendor class identifier. The `dhcpd` service sees this value as `vendor-class-identifier`. The default value is `anaconda-$(uname -srm)`.

`inst.waitfornet=`

Using the `inst.waitfornet=SECONDS` boot option causes the installation system to wait for network connectivity before installation. The value given in the `SECONDS` argument specifies the maximum amount of time to wait for network connectivity before timing out and continuing the installation process even if network connectivity is not present.

`vlan=`

Use the `vlan=` option to configure a Virtual LAN (VLAN) device on a specified interface with a given name. The syntax is `vlan=name:interface`. For example:

```
vlan=vlan5:enp0s1
```

This configures a VLAN device named `vlan5` on the `enp0s1` interface. The name can take the following forms:

<table>
<thead>
<tr>
<th>Naming scheme</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN_PLUS_VID</td>
<td>vlan0005</td>
</tr>
<tr>
<td>VLAN_PLUS_VID_NO_PAD</td>
<td>vlan5</td>
</tr>
<tr>
<td>DEV_PLUS_VID</td>
<td>enp0s1.0005</td>
</tr>
<tr>
<td>DEV_PLUS_VID_NO_PAD</td>
<td>enp0s1.5</td>
</tr>
</tbody>
</table>

`bond=`

Use the `bond=` option to configure a bonding device with the following syntax: `bond=name[:interfaces][:options]`. Replace `name` with the bonding device name, `interfaces` with a comma-separated list of physical (Ethernet) interfaces, and `options` with a comma-separated list of bonding options. For example:

```
bond=bond0:enp0s1,enp0s2:mode=active-backup,tx_queues=32,downdelay=5000
```

For a list of available options, execute the `modinfo` bonding command.

`team=`

Use the `team=` option to configure a team device with the following syntax: `team=name:interfaces`. Replace `name` with the desired name of the team device and `interfaces` with a comma-separated list of physical (Ethernet) devices to be used as underlying interfaces in the team device. For example:

```
team=team0:enp0s1,enp0s2
```

**IMPORTANT**

`bridge=`
Use the `bridge=` option to configure a bridge device with the following syntax: `bridge=name:interfaces`. Replace `name` with the desired name of the bridge device and `interfaces` with a comma-separated list of physical (Ethernet) devices to be used as underlying interfaces in the bridge device. For example:

```
bridge=bridge0:enp0s1,enp0s2
```

Additional resources

- **Configuring and managing networking**

### F.3. CONSOLE BOOT OPTIONS

This section contains information about configuring boot options for your console, monitor display, and keyboard.

**console=**

Use the `console=` option to specify a device that you want to use as the primary console. For example, to use a console on the first serial port, use `console=ttys0`. While using the `console=` argument, the installation will be started with a text UI similar when you boot with the `inst.text` option. You can use the `console=` option multiple times. If you do, the boot message is displayed on all specified consoles, but only the last one is used by the installation program. For example, if you specify `console=ttys0 console=ttys1`, the installation program uses `ttys1`.

**inst.lang=**

Use the `inst.lang=` option to set the language that you want to use during the installation. The `locale -a | grep _` or `localectl list-locales | grep _` commands return a list of locales.

**inst.singlelang**

Use the `inst.singlelang` option to install in single language mode, which results in no available interactive options for the installation language and language support configuration. If a language is specified using the `inst.lang` boot option or the `lang` Kickstart command, then it is used. If no language is specified, the installation program defaults to `en_US.UTF-8`.

**inst.geoloc=**

Use the `inst.geoloc=` option to configure geolocation usage in the installation program. Geolocation is used to preset the language and time zone, and uses the following syntax: `inst.geoloc=value`. The `value` can be any of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Boot option format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable geolocation</td>
<td><code>inst.geoloc=0</code></td>
</tr>
<tr>
<td>Use the Fedora GeoIP API</td>
<td><code>inst.geoloc=provider_fedora_geoip</code></td>
</tr>
<tr>
<td>Use the Hostip.info GeoIP API</td>
<td><code>inst.geoloc=provider_hostip</code></td>
</tr>
</tbody>
</table>

If you do not specify the `inst.geoloc=` option, the installation program uses `provider_fedora_geoip`.

**inst.keymap=**
Use the `inst.keymap=` option to specify the keyboard layout that you want to use for the installation.

**inst.cmdline**

Use the `inst.cmdline` option to force the installation program to run in command-line mode. This mode does not allow any interaction, and you must specify all options in a Kickstart file or on the command line.

**inst.graphical**

Use the `inst.graphical` option to force the installation program to run in graphical mode. This mode is the default.

**inst.text**

Use the `inst.text` option to force the installation program to run in text mode instead of graphical mode.

**inst.noninteractive**

Use the `inst.noninteractive` boot option to run the installation program in a non-interactive mode. User interaction is not permitted in the non-interactive mode, and `inst.noninteractive` can be used with a graphical or text installation. When the `inst.noninteractive` option is used in text mode it behaves the same as the `inst.cmdline` option.

**inst.resolution=**

Use the `inst.resolution=` option to specify the screen resolution in graphical mode. The format is `NxM`, where `N` is the screen width and `M` is the screen height (in pixels). The lowest supported resolution is 1024x768.

**inst.vnc**

Use the `inst.vnc` option to run the graphical installation using VNC. You must use a VNC client application to interact with the installation program. When VNC sharing is enabled, multiple clients can connect. A system installed using VNC starts in text mode.

**inst.vncpassword=**

Use the `inst.vncpassword=` option to set a password on the VNC server that is used by the installation program.

**inst.vncconnect=**

Use the `inst.vncconnect=` option to connect to a listening VNC client at the given host location. For example `inst.vncconnect=<host>[:<port>]` The default port is 5900. This option can be used with `vncviewer -listen`.

**inst.xdriver=**

Use the `inst.xdriver= option to specify the name of the X driver that you want to use both during installation and on the installed system.

**inst.usefbx**

Use the `inst.usefbx` option to prompt the installation program to use the frame buffer X driver instead of a hardware-specific driver. This option is equivalent to `inst.xdriver=fbdev`.

**modprobe.blacklist=**

Use the `modprobe.blacklist=` option to blacklist or completely disable one or more drivers. Drivers (mods) that you disable using this option cannot load when the installation starts, and after the installation finishes, the installed system retains these settings. You can find a list of the blocklisted drivers in the `/etc/modprobe.d/` directory. Use a comma-separated list to disable multiple drivers. For example:

```
modprobe.blacklist=ahci,firewire_ohci
```

**inst.xtimeout=**
Use the `inst.xtimeout=` option to specify the timeout in seconds for starting X server.

**inst.sshd**

Use the `inst.sshd` option to start the sshd service during installation, so that you can connect to the system during the installation using SSH, and monitor the installation progress. For more information about SSH, see the ssh(1) man page. By default, the sshd option is automatically started only on the 64-bit IBM Z architecture. On other architectures, sshd is not started unless you use the `inst.sshd` option.

**NOTE**

During installation, the root account has no password by default. You can set a root password during installation with the sshpw Kickstart command.

**inst.kdump_addon=**

Use the `inst.kdump_addon=` option to enable or disable the Kdump configuration screen (add-on) in the installation program. This screen is enabled by default; use `inst.kdump_addon=off` to disable it. Disabling the add-on disables the Kdump screens in both the graphical and text-based interface as well as the `%addon com_redhat_kdump` Kickstart command.

### F.4. DEBUG BOOT OPTIONS

This section contains information about the options that you can use when debugging issues.

**inst.rescue**

Use the `inst.rescue` option to run the rescue environment. The option is useful for trying to diagnose and fix systems. For example, you can repair a filesystem in rescue mode.

**inst.updates=**

Use the `inst.updates=` option to specify the location of the `updates.img` file that you want to apply during installation. There are a number of sources for the updates.

**Table F.8. inst.updates= source updates**

<table>
<thead>
<tr>
<th>Source</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updates from a network</td>
<td>The easiest way to use <code>inst.updates=</code> is to specify the network location of <code>updates.img</code>. This does not require any modification to the installation tree. To use this method, edit the kernel command line to include <code>inst.updates</code>.</td>
<td><code>inst.updates=http://some.website.com/path/to/updates.img</code>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updates from a network</td>
<td>The easiest way to use <code>inst.updates=</code> is to specify the network location of <code>updates.img</code>. This does not require any modification to the installation tree. To use this method, edit the kernel command line to include <code>inst.updates</code>.</td>
<td><code>inst.updates=http://some.website.com/path/to/updates.img</code>.</td>
</tr>
</tbody>
</table>
Updates from a disk image

You can save an `updates.img` on a floppy drive or a USB key. This can be done only with an `ext2` filesystem type of `updates.img`. To save the contents of the image on your floppy drive, insert the floppy disc and run the command.

```
  dd if=updates.img of=/dev/fd0 bs=72k count=20
```
To use a USB key or flash media, replace `/dev/fd0` with the device name of your USB key.

Updates from an installation tree

If you are using a CD, hard drive, HTTP, or FTP install, you can save the `updates.img` in the installation tree so that all installations can detect the `.img` file. Save the file in the `images/` directory. The file name must be `updates.img`.

For NFS installs, there are two options: You can either save the image in the `images/` directory, or in the `RHupdates/` directory in the installation tree.

<table>
<thead>
<tr>
<th>Source</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updates from a disk image</td>
<td>You can save an <code>updates.img</code> on a floppy drive or a USB key. This can be done only with an <code>ext2</code> filesystem type of <code>updates.img</code>. To save the contents of the image on your floppy drive, insert the floppy disc and run the command.</td>
<td><code>dd if=updates.img of=/dev/fd0 bs=72k count=20</code>. To use a USB key or flash media, replace <code>/dev/fd0</code> with the device name of your USB key.</td>
</tr>
<tr>
<td>Updates from an installation tree</td>
<td>If you are using a CD, hard drive, HTTP, or FTP install, you can save the <code>updates.img</code> in the installation tree so that all installations can detect the <code>.img</code> file. Save the file in the <code>images/</code> directory. The file name must be <code>updates.img</code>.</td>
<td>For NFS installs, there are two options: You can either save the image in the <code>images/</code> directory, or in the <code>RHupdates/</code> directory in the installation tree.</td>
</tr>
</tbody>
</table>

### inst.loglevel=

Use the `inst.loglevel=` option to specify the minimum level of messages logged on a terminal. This concerns only terminal logging; log files always contain messages of all levels. Possible values for this option from the lowest to highest level are: `debug`, `info`, `warning`, `error` and `critical`. The default value is `info`, which means that by default, the logging terminal displays messages ranging from `info` to `critical`.

### inst.syslog=

When installation starts, the `inst.syslog=` option sends log messages to the `syslog` process on the specified host. The remote `syslog` process must be configured to accept incoming connections.

### inst.virtiolog=

Use the `inst.virtiolog=` option to specify the virtio port (a character device at `/dev/virtio-ports/name`) that you want to use for forwarding logs. The default value is `org.fedoraproject.anaconda.log.0`; if this port is present, it is used.

### inst.zram=

The `inst.zram=` option controls the usage of zRAM swap during installation. The option creates a compressed block device inside the system RAM and uses it for swap space instead of the hard drive. This allows the installation program to run with less available memory than is possible without compression, and it might also make the installation faster. By default, swap on zRAM is enabled on systems with 2 GiB or less RAM, and disabled on systems with more than 2 GiB of memory. You can use this option to change this behavior; on a system with more than 2 GiB RAM, use `inst.zram=1` to enable the feature, and on systems with 2 GiB or less memory, use `inst.zram=0` to disable the feature.

### rd.live.ram

If the `rd.live.ram` option is specified, the stage 2 image is copied into RAM. If the `rd.live.ram` option is specified, the stage 2 image (`images/install.img`) is copied into RAM. This increases the memory required for installation by the size of the image. The size may vary from 400 to 800MB.

### inst.nokill

The `inst.nokill` option is a debugging option that prevents the installation program from rebooting when a fatal error occurs, or at the end of the installation process. Use the `inst.nokill` option to capture installation logs which would be lost upon reboot.
inst.noshell

Use **inst.noshell** option if you do not want a shell on terminal session 2 (tty2) during installation.

inst.notmux

Use **inst.notmux** option if you do not want to use tmux during installation. The output is generated without terminal control characters and is meant for non-interactive uses.

inst.remotelog=

You can use the **inst.remotelog=** option to send all of the logs to a remote **host:port** using a TCP connection. The connection is retired if there is no listener and the installation proceeds as normal.

F.5. STORAGE BOOT OPTIONS

inst.nodmraid

Use the **inst.nodmraid** option to disable **dmraid** support.

**WARNING**

Use this option with caution. If you have a disk that is incorrectly identified as part of a firmware RAID array, it might have some stale RAID metadata on it that must be removed using the appropriate tool, for example, **dmraid** or **wipefs**.

inst.nompath

Use the **inst.nompath** option to disable support for multipath devices. This option can be used for systems on which a false-positive is encountered which incorrectly identifies a normal block device as a multipath device. There is no other reason to use this option.

**WARNING**

Use this option with caution. You should not use this option with multipath hardware. Using this option to attempt to install to a single path of a multipath is not supported.

inst.gpt

The **inst.gpt** boot option forces the installation program to install partition information to a GUID Partition Table (GPT) instead of a Master Boot Record (MBR). This option is not valid on UEFI-based systems, unless they are in BIOS compatibility mode. Normally, BIOS-based systems and UEFI-based systems in BIOS compatibility mode attempt to use the MBR schema for storing partitioning information, unless the disk is $2^{32}$ sectors in size or larger. Disk sectors are typically 512 bytes in size, meaning that this is usually equivalent to 2 TiB. Using the **inst.gpt** boot option changes this behavior, allowing a GPT to be written to smaller disks.

F.6. DEPRECATED BOOT OPTIONS
This section contains information about deprecated boot options. These options are still accepted by
the installation program but they are deprecated and are scheduled to be removed in a future release of
Red Hat Enterprise Linux.

method
The method option is an alias for inst.repo.

dns
Use nameserver instead of dns. Note that nameserver does not accept comma-separated lists; use
multiple nameserver options instead.

netmask, gateway, hostname
The netmask, gateway, and hostname options are provided as part of the ip option.

ip=bootif
A PXE-supplied BOOTIF option is used automatically, so there is no requirement to use ip=bootif.

ksdevice

Table F.9. Values for the ksdevice boot option

<table>
<thead>
<tr>
<th>Value</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not present</td>
<td>N/A</td>
</tr>
<tr>
<td>ksdevice=link</td>
<td>Ignored as this option is the same as the default behavior</td>
</tr>
<tr>
<td>ksdevice=bootif</td>
<td>Ignored as this option is the default if BOOTIF= is present</td>
</tr>
<tr>
<td>ksdevice=ibft</td>
<td>Replaced with ip=ibft. See ip for details</td>
</tr>
<tr>
<td>ksdevice=&lt;MAC&gt;</td>
<td>Replaced with BOOTIF=${MAC:/-}</td>
</tr>
<tr>
<td>ksdevice=&lt;DEV&gt;</td>
<td>Replaced with bootdev</td>
</tr>
</tbody>
</table>

F.7. REMOVED BOOT OPTIONS

This section contains the boot options that have been removed from Red Hat Enterprise Linux.

NOTE

dracut provides advanced boot options. For more information about dracut, see the
dracut.cmdline(7) man page.

askmethod, asknetwork

initramfs is completely non-interactive, so the askmethod and asknetwork options have been
removed. Instead, use inst.repo or specify the appropriate network options.

blacklist, nofirewire
The `modprobe` option handles blocklisting kernel modules; use `modprobe.blacklist=<mod1>, <mod2>`. You can blocklist the firewire module by using `modprobe.blacklist=firewire_ohci`.

`inst.headless=`

The `headless=` option specified that the system that is being installed to does not have any display hardware, and that the installation program is not required to look for any display hardware.

`inst.decorated`

The `inst.decorated` option was used to specify the graphical installation in a decorated window. By default, the window is not decorated, so it doesn’t have a title bar, resize controls, and so on. This option was no longer required.

`repo=nfsiso`

Use the `inst.repo=nfs:` option.

`serial`

Use the `console=ttyS0` option.

`updates`

Use the `inst.updates` option.

`essid, wepkey, wpakey`

Dracut does not support wireless networking.

`ethtool`

This option was no longer required.

`gdb`

This option was removed as there are many options available for debugging dracut-based `initramfs`.

`inst.mediacheck`

Use the `dracut option rd.live.check` option.

`ks=floppy`

Use the `inst.ks=hd:<device>` option.

`display`

For a remote display of the UI, use the `inst.vnc` option.

`utf8`

This option was no longer required as the default TERM setting behaves as expected.

`noipv6`

Ipv6 is built into the kernel and cannot be removed by the installation program. You can disable ipv6 using `ipv6.disable=1`. This setting is used by the installed system.

`upgradeany`

This option was no longer required as the installation program no longer handles upgrades.
APPENDIX G. CHANGING A SUBSCRIPTION SERVICE

To manage the subscriptions, you can register a RHEL system with either Red Hat Subscription Management Server or Red Hat Satellite Server. If required, you can change the subscription service at a later point. To change the subscription service under which you are registered, unregister the system from the current service and then register it with a new service.

This section contains information about how to unregister your RHEL system from the Red Hat Subscription Management Server and Red Hat Satellite Server.

**Prerequisites**

You have registered your system with any one of the following:

- Red Hat Subscription Management Server
- Red Hat Satellite Server

**NOTE**

To receive the system updates, register your system with either of the management server.

G.1. UNREGISTERING FROM SUBSCRIPTION MANAGEMENT SERVER

This section contains information about how to unregister a RHEL system from Red Hat Subscription Management Server, using a command line and the Subscription Manager user interface.

G.1.1. Unregistering using command line

Use the `unregister` command to unregister a RHEL system from Red Hat Subscription Management Server.

**Procedure**

1. Run the unregister command as a root user, without any additional parameters.

   ```bash
   # subscription-manager unregister
   ```

2. When prompted, provide a root password.

   The system is unregistered from the Subscription Management Server, and the status 'The system is currently not registered' is displayed with the Register button enabled.

   **NOTE**

   To continue uninterrupted services, re-register the system with either of the management services. If you do not register the system with a management service, you may fail to receive the system updates. For more information about registering a system, see Registering your system using the command line

**Additional resources**

- Using and Configuring Red Hat Subscription Manager
G.1.2. Unregistering using Subscription Manager user interface

This section contains information about how to unregister a RHEL system from Red Hat Subscription Management Server, using Subscription Manager user interface.

Procedure

1. Log in to your system.

2. From the top left-hand side of the window, click Activities.

3. From the menu options, click the Show Applications icon.

4. Click the Red Hat Subscription Manager icon, or enter Red Hat Subscription Manager in the search.

5. Enter your administrator password in the Authentication Required dialog box. The Subscriptions window appears and displays the current status of Subscriptions, System Purpose, and installed products. Unregistered products display a red X.

   **NOTE**

   Authentication is required to perform privileged tasks on the system.

6. Click the Unregister button.

The system is unregistered from the Subscription Management Server, and the status 'The system is currently not registered' is displayed with the Register button enabled.

   **NOTE**

   To continue uninterrupted services, re-register the system with either of the management services. If you do not register the system with a management service, you may fail to receive the system updates. For more information about registering a system, see Registering your system using the Subscription Manager User Interface

Additional resources

- Using and Configuring Red Hat Subscription Manager

G.2. UNREGISTERING FROM SATELLITE SERVER

To unregister a Red Hat Enterprise Linux system from Satellite Server, remove the system from Satellite Server.

For more information, see Removing a Host from Red Hat Satellite in the Managing Hosts guide from Satellite Server documentation.
APPENDIX H. iSCSI DISKS IN INSTALLATION PROGRAM

The Red Hat Enterprise Linux installer can discover and log in to iSCSI disks in two ways:

- When the installer starts, it checks if the BIOS or add-on boot ROMs of the system support iSCSI Boot Firmware Table (iBFT), a BIOS extension for systems that can boot from iSCSI. If the BIOS supports iBFT, the installer reads the iSCSI target information for the configured boot disk from the BIOS and logs in to this target, making it available as an installation target.

  IMPORTANT

  To connect automatically to an iSCSI target, activate a network device for accessing the target. To do so, use `ip=ibft` boot option. For more information, see Network boot options.

- You can discover and add iSCSI targets manually in the installer’s graphical user interface. For more information, see Configuring storage devices.

  IMPORTANT

  You cannot place the `/boot` partition on iSCSI targets that you have manually added using this method - an iSCSI target containing a `/boot` partition must be configured for use with iBFT. However, in instances where the installed system is expected to boot from iSCSI with iBFT configuration provided by a method other than firmware iBFT, for example using iPXE, you can remove the `/boot` partition restriction using the `inst.nonibftiscsiboot` installer boot option.

While the installer uses `iscsiadm` to find and log into iSCSI targets, `iscsiadm` automatically stores any information about these targets in the `iscsiadm` iSCSI database. The installer then copies this database to the installed system and marks any iSCSI targets that are not used for root partition, so that the system automatically logs in to them when it starts. If the root partition is placed on an iSCSI target, initrd logs into this target and the installer does not include this target in start up scripts to avoid multiple attempts to log into the same target.
CHAPTER 6. BOOTING A BETA SYSTEM WITH UEFI SECURE BOOT

To enhance the security of your operating system, use the UEFI Secure Boot feature for signature verification when booting a Red Hat Enterprise Linux Beta release on systems having UEFI Secure Boot enabled.

6.1. UEFI SECURE BOOT AND RHEL BETA RELEASES

UEFI Secure Boot requires that the operating system kernel is signed with a recognized private key. UEFI Secure Boot then verifies the signature using the corresponding public key.

For Red Hat Enterprise Linux Beta releases, the kernel is signed with a Red Hat Beta-specific private key. UEFI Secure Boot attempts to verify the signature using the corresponding public key, but because the hardware does not recognize the Beta private key, Red Hat Enterprise Linux Beta release system fails to boot. Therefore, to use UEFI Secure Boot with a Beta release, add the Red Hat Beta public key to your system using the Machine Owner Key (MOK) facility.

6.2. ADDING A BETA PUBLIC KEY FOR UEFI SECURE BOOT

This section contains information about how to add a Red Hat Enterprise Linux Beta public key for UEFI Secure Boot.

Prerequisites

- UEFI Secure Boot is disabled on the system.
- The Red Hat Enterprise Linux Beta release is installed, and Secure Boot is disabled even after system reboot.
- You are logged in to the system, and the tasks in the Initial Setup window are complete.

Procedure

1. Begin to enroll the Red Hat Beta public key in the system’s Machine Owner Key (MOK) list:

   ```
   # mokutil --import /usr/share/doc/kernel-keys/$(uname -r)/kernel-signing-ca.cer
   
   $(uname -r) is replaced by the kernel version - for example, 4.18.0-80.el8.x86_64.
   ```

2. Enter a password when prompted.

3. Reboot the system and press any key to continue the startup. The Shim UEFI key management utility starts during the system startup.

4. Select Enroll MOK.

5. Select Continue.

6. Select Yes and enter the password. The key is imported into the system’s firmware.

7. Select Reboot.

8. Enable Secure Boot on the system.
6.3. REMOVING A BETA PUBLIC KEY

If you plan to remove the Red Hat Enterprise Linux Beta release, and install a Red Hat Enterprise Linux General Availability (GA) release, or a different operating system, then remove the Beta public key.

The procedure describes how to remove a Beta public key.

Procedure

1. Begin to remove the Red Hat Beta public key from the system’s Machine Owner Key (MOK) list:
   
   ```bash
   # mokutil --reset
   ```

2. Enter a password when prompted.

3. Reboot the system and press any key to continue the startup. The Shim UEFI key management utility starts during the system startup.

4. Select **Reset MOK**.

5. Select **Continue**.

6. Select **Yes** and enter the password that you had specified in step 2. The key is removed from the system's firmware.

7. Select **Reboot**.

Red Hat Enterprise Linux 8 System Design Guide

172
CHAPTER 7. COMPOSING A CUSTOMIZED RHEL SYSTEM IMAGE

7.1. IMAGE BUILDER DESCRIPTION

7.1.1. Introduction to Image Builder

You can use Image Builder to create customized system images of Red Hat Enterprise Linux, including system images prepared for deployment on cloud platforms. Image Builder automatically handles details of setup for each output type and is thus easier to use and faster to work with than manual methods of image creation. You can access Image Builder functionality through a command-line interface in the composer-cli tool, or a graphical user interface in the RHEL web console.

As of Red Hat Enterprise Linux 8.3, the osbuild-composer backend replaces lorax-composer. The new service provides REST APIs for image building. As a result, users can benefit from a more reliable backend and more predictable output images.

Image Builder runs as a system service osbuild-composer. You can interact with this service through two interfaces:

- CLI tool composer-cli for running commands in the terminal. This method is preferred.
- GUI plugin for the RHEL web console.

7.1.2. Image Builder terminology

Blueprint

Blueprints define customized system images by listing packages and customizations that will be part of the system. Blueprints can be edited and they are versioned. When a system image is created from a blueprint, the image is associated with the blueprint in the Image Builder interface of the RHEL web console.

Blueprints are presented to the user as plain text in the Tom’s Obvious, Minimal Language (TOML) format.

Compose

Composes are individual builds of a system image, based on a particular version of a particular blueprint. Compose as a term refers to the system image, the logs from its creation, inputs, metadata, and the process itself.

Customizations

Customizations are specifications for the system, which are not packages. This includes users, groups, and SSH keys.

7.1.3. Image Builder output formats

Image Builder can create images in multiple output formats shown in the following table. To check the supported types, run the command:

```
# composer-cli compose types
```

Table 7.1. Image Builder output formats
<table>
<thead>
<tr>
<th>Description</th>
<th>CLI name</th>
<th>file extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>QEMU QCOW2 Image</td>
<td>qcow2</td>
<td>.qcow2</td>
</tr>
<tr>
<td>TAR Archive</td>
<td>tar</td>
<td>.tar</td>
</tr>
<tr>
<td>Amazon Machine Image Disk</td>
<td>ami</td>
<td>.raw</td>
</tr>
<tr>
<td>Azure Disk Image</td>
<td>vhd</td>
<td>.vhd</td>
</tr>
<tr>
<td>VMware Virtual Machine Disk</td>
<td>vmdk</td>
<td>.vmdk</td>
</tr>
<tr>
<td>Openstack</td>
<td>openstack</td>
<td>.qcow2</td>
</tr>
<tr>
<td>RHEL for Edge Commit</td>
<td>edge-commit</td>
<td>.tar</td>
</tr>
<tr>
<td>RHEL for Edge Container</td>
<td>edge-container</td>
<td>.tar</td>
</tr>
<tr>
<td>RHEL for Edge Installer</td>
<td>edge-installer</td>
<td>.iso</td>
</tr>
<tr>
<td>RHEL for Edge Raw</td>
<td>edge-raw-image</td>
<td>.tar</td>
</tr>
<tr>
<td>RHEL for Edge Simplified Installer</td>
<td>edge-simplified-installer</td>
<td>.iso</td>
</tr>
<tr>
<td>ISO image</td>
<td>image-installer</td>
<td>.iso</td>
</tr>
</tbody>
</table>

### 7.1.4. Image Builder system requirements

The environment where Image Builder runs, for example a dedicated virtual machine, must meet requirements listed in the following table.

**Table 7.2. Image Builder system requirements**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Minimal Required Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>System type</td>
<td>A dedicated virtual machine</td>
</tr>
<tr>
<td>Processor</td>
<td>2 cores</td>
</tr>
<tr>
<td>Memory</td>
<td>4 GiB</td>
</tr>
<tr>
<td>Disk space</td>
<td>20 GiB</td>
</tr>
<tr>
<td>Access privileges</td>
<td>Administrator level (root)</td>
</tr>
<tr>
<td>Network</td>
<td>Connectivity to Internet</td>
</tr>
</tbody>
</table>
7.2. INSTALLING IMAGE BUILDER

Image Builder is a tool for creating custom system images. Before using Image Builder, you must install Image Builder in a virtual machine.

7.2.1. Image Builder system requirements

The environment where Image Builder runs, for example a dedicated virtual machine, must meet requirements listed in the following table.

Table 7.3. Image Builder system requirements

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Minimal Required Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>System type</td>
<td>A dedicated virtual machine</td>
</tr>
<tr>
<td>Processor</td>
<td>2 cores</td>
</tr>
<tr>
<td>Memory</td>
<td>4 GiB</td>
</tr>
<tr>
<td>Disk space</td>
<td>20 GiB</td>
</tr>
<tr>
<td>Access privileges</td>
<td>Administrator level (root)</td>
</tr>
<tr>
<td>Network</td>
<td>Connectivity to Internet</td>
</tr>
</tbody>
</table>

7.2.2. Installing Image Builder in a virtual machine

To install Image Builder on a dedicated virtual machine, follow these steps:

Prerequisites

- Connect to the virtual machine.
- The virtual machine for Image Builder must be installed, subscribed to RHSM or Red Hat Satellite, and running.

Procedure

1. Install the Image Builder and other necessary packages on the virtual machine:
- osbuild-composer - supported from RHEL 8.3 onward
- composer-cli
- cockpit-composer
- bash-completion

```bash
# yum install osbuild-composer composer-cli cockpit-composer bash-completion
```

The web console is installed as a dependency of the cockpit-composer package.

2. Enable Image Builder to start after each reboot:

```bash
# systemctl enable --now osbuild-composer.socket
# systemctl enable --now cockpit.socket
```

The osbuild-composer and cockpit services start automatically on first access.

3. Load the shell configuration script so that the autocomplete feature for the composer-cli command starts working immediately without reboot:

```bash
$ source /etc/bash_completion.d/composer-cli
```

**IMPORTANT**

The osbuild-composer package is the new backend engine that will be the preferred default and focus of all new functionality beginning with Red Hat Enterprise Linux 8.3 and later. The previous backend lorax-composer package is considered deprecated, will only receive select fixes for the remainder of the Red Hat Enterprise Linux 8 life cycle and will be omitted from future major releases. It is recommended to uninstall lorax-composer in favor of osbuild-composer.

### 7.2.3. Reverting to lorax-composer Image Builder backend

The osbuild-composer backend, though much more extensible, does not currently achieve feature parity with the previous lorax-composer backend.

To revert to the previous backend, follow the steps:

**Prerequisites**

- You have installed the osbuild-composer package

**Procedure**

1. Remove the osbuild-composer backend.

```bash
# yum remove osbuild-composer
```

2. In the /etc/yum.conf file, add an exclude entry for osbuild-composer package.

```bash
# cat /etc/yum.conf
```
3. Install the **lorax-composer** package.

```
# yum install lorax-composer
```

4. Enable and start the **lorax-composer** service to start after each reboot.

```
# systemctl enable --now lorax-composer.socket
# systemctl start lorax-composer
```

**Additional resources**

- Create a Case at Red Hat Support.

### 7.3. CREATING SYSTEM IMAGES WITH IMAGE BUILDER COMMAND-LINE INTERFACE

Image Builder is a tool for creating custom system images. To control Image Builder and create your custom system images, use the command-line interface which is currently the preferred method to use Image Builder.

#### 7.3.1. Image Builder command-line interface

Image Builder command-line interface is currently the preferred method to use Image Builder. It offers more functionality than the Web console interface. To use this interface, run the **composer-cli** command with suitable options and subcommands.

The workflow for the command-line interface can be summarized as follows:

1. Export (save) the blueprint definition to a plain text file
2. Edit this file in a text editor
3. Import (push) the blueprint text file back into Image Builder
4. Run a compose to build an image from the blueprint
5. Export the image file to download it

Apart from the basic subcommands to achieve this procedure, the **composer-cli** command offers many subcommands to examine the state of configured blueprints and composes.

To run the **composer-cli** command as non-root, user must be in the **weldr** or **root** groups.

#### 7.3.2. Creating an Image Builder blueprint with command-line interface

This procedure describes how to create a new Image Builder blueprint using the command-line interface.
Procedure

1. Create a plain text file with the following contents:

```toml
name = "BLUEPRINT-NAME"
description = "LONG FORM DESCRIPTION TEXT"
version = "0.0.1"
modules = []
groups = []
```

Replace `BLUEPRINT-NAME` and `LONG FORM DESCRIPTION TEXT` with a name and description for your blueprint.

Replace `0.0.1` with a version number according to the Semantic Versioning scheme.

2. For every package that you want to be included in the blueprint, add the following lines to the file:

```toml
[[packages]]
name = "package-name"
version = "package-version"
```

Replace `package-name` with name of the package, such as `httpd`, `gdb-doc`, or `coreutils`.

Replace `package-version` with a version to use. This field supports dnf version specifications:

- For a specific version, use the exact version number such as `8.30`.
- For latest available version, use the asterisk `*`.
- For a latest minor version, use format such as `8.*`.

3. Blueprints can be customized in a number of ways. For this example, Simultaneous Multi Threading (SMT) can be disabled by performing the steps below. For additional customizations available, please see Supported Image Customizations.

```toml
[customizations.kernel]
append = "nosmt=force"
```

4. Save the file as `BLUEPRINT-NAME.toml` and close the text editor.

5. Push (import) the blueprint:

```bash
# composer-cli blueprints push BLUEPRINT-NAME.toml
```

Replace `BLUEPRINT-NAME` with the value you used in previous steps.

6. To verify that the blueprint has been pushed and exists, list the existing blueprints:

```bash
# composer-cli blueprints list
```

7. Check whether the components and versions listed in the blueprint and their dependencies are valid:

```bash
# composer-cli blueprints depsolve BLUEPRINT-NAME
```
NOTE

You are able to create images using the composer-cli command as non-root. To do so, add your user to the weldr or root groups. To add your user to the weldr group, perform the following steps:

```bash
# usermod -a -G weldr user
$ newgrp weldr
```

7.3.3. Editing an Image Builder blueprint with command-line interface

This procedure describes how to edit an existing Image Builder blueprint in the command-line interface.

Procedure

1. Save (export) the blueprint to a local text file:

```bash
# composer-cli blueprints save BLUEPRINT-NAME
```

2. Edit the BLUEPRINT-NAME.toml file with a text editor of your choice and make your changes.

3. Before finishing with the edits, make sure the file is a valid blueprint:
   a. Remove this line, if present:

   ```toml
   packages = []
   ```
   b. Increase the version number. Remember that Image Builder blueprint versions must use the Semantic Versioning scheme. Note also that if you do not change the version, the patch component of version is increased automatically.
   c. Check if the contents are valid TOML specifications. See the TOML documentation for more information.

   NOTE

   TOML documentation is a community product and is not supported by Red Hat. You can report any issues with the tool at https://github.com/toml-lang/toml/issues

4. Save the file and close the editor.

5. Push (import) the blueprint back into Image Builder:

```bash
# composer-cli blueprints push BLUEPRINT-NAME.toml
```

Note that you must supply the file name including the .toml extension, while in other commands you use only the name of the blueprint.

6. To verify that the contents uploaded to Image Builder match your edits, list the contents of blueprint:

```bash
# composer-cli blueprints show BLUEPRINT-NAME
```
7. Check whether the components and versions listed in the blueprint and their dependencies are valid:

```bash
# composer-cli blueprints depsolve BLUEPRINT-NAME
```

### 7.3.4. Creating a system image with Image Builder in the command-line interface

This procedure shows how to build a custom image using the Image Builder command-line interface.

**Prerequisites**

- You have a blueprint prepared for the image.

**Procedure**

1. Start the compose:

```bash
# composer-cli compose start BLUEPRINT-NAME IMAGE-TYPE
```

Replace `BLUEPRINT-NAME` with name of the blueprint, and `IMAGE-TYPE` with the type of image. For possible values, see output of the `composer-cli compose types` command.

The compose process starts in the background and the UUID of the compose is shown.

2. Wait until the compose is finished. Please, notice that this may take several minutes.

To check the status of the compose:

```bash
# composer-cli compose status
```

A finished compose shows a status value `FINISHED`. Identify the compose in the list by its UUID.

3. Once the compose is finished, download the resulting image file:

```bash
# composer-cli compose image UUID
```

Replace `UUID` with the UUID value shown in the previous steps.

You can also download the logs using the `composer-cli compose logs UUID` command, or the metadata using the `composer-cli compose metadata UUID` command.

### 7.3.5. Basic Image Builder command-line commands

The Image Builder command-line interface offers the following subcommands.

**Blueprint manipulation**

- List all available blueprints

```bash
# composer-cli blueprints list
```

- Show a blueprint contents in the TOML format

```bash
# composer-cli blueprints show BLUEPRINT-NAME
```
Save (export) blueprint contents in the TOML format into a file `BLUEPRINT-NAME.toml`

```
# composer-cli blueprints save BLUEPRINT-NAME
```

Remove a blueprint

```
# composer-cli blueprints delete BLUEPRINT-NAME
```

Push (import) a blueprint file in the TOML format into Image Builder

```
# composer-cli blueprints push BLUEPRINT-NAME
```

Composing images from blueprints

List the available image types

```
# composer-cli compose types
```

Start a compose

```
# composer-cli compose start BLUEPRINT COMPOSE-TYPE
```

Replace `BLUEPRINT` with name of the blueprint to build and `COMPOSE-TYPE` with the output image type.

List all composes

```
# composer-cli compose list
```

List all composes and their status

```
# composer-cli compose status
```

Cancel a running compose

```
# composer-cli compose cancel COMPOSE-UUID
```

Delete a finished compose

```
# composer-cli compose delete COMPOSE-UUID
```

Show detailed information about a compose

```
# composer-cli compose info COMPOSE-UUID
```

Download image file of a compose

```
# composer-cli compose image COMPOSE-UUID
```
Additional resources

- The `composer-cli(1)` manual page provides a full list of the available subcommands and options:

  ```
  $ man composer-cli
  ```

- The `composer-cli` command provides help on the subcommands and options:

  ```
  # composer-cli help
  ```

### 7.3.6. Image Builder blueprint format

Image Builder blueprints are presented to the user as plain text in the Tom’s Obvious, Minimal Language (TOML) format.

The elements of a typical blueprint file include:

#### The blueprint metadata

```toml
name = "$BLUEPRINT-NAME"
description = "$LONG FORM DESCRIPTION TEXT"
version = "$VERSION"
```

Replace `BLUEPRINT-NAME` and `LONG FORM DESCRIPTION TEXT` with a name and description for your blueprint.

Replace `VERSION` with a version number according to the [Semantic Versioning](https://semver.org/) scheme.

This part is present only once for the whole blueprint file.

The entry `modules` describe the package names and matching version glob to be installed into the image.

The entry `group` describes a group of packages to be installed into the image. Groups categorize their packages in:

- Mandatory
- Default
- Optional
  Blueprints installs the mandatory packages. There is no mechanism for selecting optional packages.

#### Groups to include in the image

```toml
[[groups]]
name = "group-name"
```

Replace `group-name` with the name of the group, such as `anaconda-tools`, `widget`, `wheel` or `users`.

#### Packages to include in the image
Replace package-name with the name of the package, such as httpd, gdb-doc, or coreutils. Replace package-version with a version to use. This field supports dnf version specifications:

- For a specific version, use the exact version number such as 8.30.
- For latest available version, use the asterisk *.
- For the latest minor version, use format such as 8.*.

Repeat this block for every package to include.

### 7.3.7. Supported Image Customizations

A number of image customizations are supported at this time within blueprints. In order to make use of these options, they must be initially configured in the blueprint and imported (pushed) to Image Builder.

**NOTE**

These customizations are not currently supported within the accompanying cockpit-composer GUI.

Set the image hostname

```bash
[customizations]
hostname = "baseimage"
```

User specifications for the resulting system image

```bash
[[customizations.user]]
name = "USER-NAME"
description = "USER-DESCRIPTION"
password = "PASSWORD-HASH"
key = "PUBLIC-SSH-KEY"
home = "/home/USER-NAME/
shell = "/usr/bin/bash"
groups = ["users", "wheel"]
uid = NUMBER
gid = NUMBER
```

**NOTE**

The GID is optional and must already exist in the image, be created by a package, or be created by the blueprint `[[customizations.group]]` entry.
IMPORTANT

To generate the hash, you must install python3 on your system. The following command will install the python3 package.

```
# yum install python3
```

Replace PASSWORD-HASH with the actual password hash. To generate the hash, use a command such as:

```
$ python3 -c "import crypt, getpass; pw = getpass.getpass(); print(crypt.crypt(pw) if (pw == getpass.getpass("Confirm: ")) else exit())"
```

Replace PUBLIC-SSH-KEY with the actual public key.

Replace the other placeholders with suitable values.

Leave out any of the lines as needed, only the user name is required.

Repeat this block for every user to include.

Group specifications for the resulting system image

```
[[customizations.group]]
name = "GROUP-NAME"
gid = NUMBER
```

Repeat this block for every group to include.

Set an existing users ssh key

```
[[customizations.sshkey]]
user = "root"
key = "PUBLIC-SSH-KEY"
```

NOTE

This option is only applicable for existing users. To create a user and set an ssh key, use the User specifications for the resulting system image customization.

Append a kernel boot parameter option to the defaults

```
[customizations.kernel]
append = "KERNEL-OPTION"
```

Define a kernel name to be used in an image

```
[customizations.kernel.name]
name = "KERNEL-NAME"
```

Set the timezone and the Network Time Protocol (NTP) servers for the resulting system image
If you do not set a timezone, the system uses *Universal Time, Coordinated (UTC)* as default. Setting NTP servers is optional.

**Set the locale settings for the resulting system image**

```ini
[customizations.locale]
languages = ["LANGUAGE"]
keyboard = "KEYBOARD"
```

Setting both language and keyboard options is mandatory. You can add multiple languages. The first language you add will be the primary language and the other languages will be secondary.

**Set the firewall for the resulting system image**

```ini
[customizations.firewall]
port = ["PORTS"]
```

You can use the numeric ports, or their names from the */etc/services* file to enable lists.

**Customize the firewall services**

Review the available firewall services.

```
$ firewall-cmd --get-services
```

In the blueprint, under section `customizations.firewall.service`, specify the firewall services that you want to customize.

```ini
[customizations.firewall.services]
enabled = ["SERVICES"]
disabled = ["SERVICES"]
```

The services listed in `firewall.services` are different from the names available in the */etc/services* file.

You can optionally customize the firewall services for the system image that you plan to create.

**NOTE**

If you do not want to customize the firewall services, omit the `customizations.firewall` and `customizations.firewall.services` sections from the blueprint.

**Set which services to enable during the boot time**

```ini
[customizations.services]
enabled = ["SERVICES"]
disabled = ["SERVICES"]
```
You can control which services to enable during the boot time. Some image types already have services enabled or disabled so that the image works correctly and this setup cannot be overridden.

**NOTE**

Each time a build starts, it clones the repository. If you refer to a repository with a large amount of history, it might take a while to clone and use a significant amount of disk space. Also, the clone is temporary and is removed once the RPM package is created.

**Specify a custom filesystem configuration**

You can specify a custom filesystem configuration in your blueprints and thus create images with a specific disk layout, instead of using the default layout configuration. By using the non-default layout configuration in your blueprints, you can benefit from:

- security benchmark compliance
- protection against out-of-disk errors
- performance
- consistency with existing setups

Customize the filesystem configuration in your blueprint:

```bash
[[customizations.filesystem]]
  mountpoint = "MOUNTPOINT"
  size = MINIMUM-PARTITION-SIZE
```

The following mountpoints and their sub-directories are supported:

- `/` - the root mount point
- `/var`
- `/home`
- `/opt`
- `/srv`
- `/usr`
- `/app`
- `/data`

**NOTE**

Customizing mount points is only supported in the RHEL 8.5 and RHEL 9.0 distributions, using the CLI. In early distributions, you can only specify the root partition as a mount point and specify the size argument as an alias for the image size.

### 7.3.8. Installed Packages
When you create a system image using Image Builder, by default, the system installs a set of base packages. The base list of packages are the members of the **comps core** group. By default, Image Builder uses the **core yum** group.

### Table 7.4. Default packages to support image type creation

<table>
<thead>
<tr>
<th>Image type</th>
<th>Default Packages</th>
</tr>
</thead>
<tbody>
<tr>
<td>ami</td>
<td>checkpolicy, chrony, cloud-init, cloud-utils-growpart,</td>
</tr>
<tr>
<td></td>
<td>@Core, dhcp-client, gdisk, insights-client, kernel,</td>
</tr>
<tr>
<td></td>
<td>langpacks-en, net-tools, NetworkManager, redhat-release, redhat-release-eula,</td>
</tr>
<tr>
<td></td>
<td>rng-tools, rsync, selinux-policy-targeted, tar, yum-utils</td>
</tr>
<tr>
<td>openstack</td>
<td>@Core, langpacks-en</td>
</tr>
<tr>
<td>qcow2</td>
<td>@Core, chrony, dnf, kernel, yum, nfs-utils, dnf-utils,</td>
</tr>
<tr>
<td></td>
<td>cloud-init, python3-jsonschema, qemu-guest-agent,</td>
</tr>
<tr>
<td></td>
<td>cloud-utils-growpart, dracut-norescue, tar, tcpdump,</td>
</tr>
<tr>
<td></td>
<td>rsync, dnf-plugin-spacewalk, rhn-client-tools, rhnlib,</td>
</tr>
<tr>
<td></td>
<td>rhnsd, rhn-setup, NetworkManager, dhcp-client, cockpit-ws, cockpit-system,</td>
</tr>
<tr>
<td></td>
<td>subscription-manager-cockpit, redhat-release, redhat-release-eula, rng-tools,</td>
</tr>
<tr>
<td></td>
<td>insights-client</td>
</tr>
<tr>
<td>rhel-edge-commit</td>
<td>glibc, glibc-minimal-langpack, nss-altfiles, kernel,</td>
</tr>
<tr>
<td></td>
<td>dracut-config-generic, dracut-network, basesystem, bash, platform-python,</td>
</tr>
<tr>
<td></td>
<td>shadow-utils, chrony, setup, sudo, systemd, coreutils, util-linux, curl,</td>
</tr>
<tr>
<td></td>
<td>vim-minimal, rpm, rpm-ostree, polkit, lvm2, cryptosetup, pinentry,</td>
</tr>
<tr>
<td></td>
<td>e2fsprogs, dosfstools, keyutils, gnupg2, attr, xz, gzip, firewalld, iptables,</td>
</tr>
<tr>
<td></td>
<td>NetworkManager, NetworkManager-wifi, NetworkManager-wwan, wpa_supplicant,</td>
</tr>
<tr>
<td></td>
<td>dnsmasq, traceroute, hostname, iproute, iptutils, openssh-clients, procps-ng,</td>
</tr>
<tr>
<td></td>
<td>rootfiles, openssh-server, passwd, policycoreutils, policycoreutils-python-utils,</td>
</tr>
<tr>
<td></td>
<td>selinux-policy-targeted, setools-console, less, tar, rsync, fwupd, usbguard,</td>
</tr>
<tr>
<td></td>
<td>bash-completion, tmux, ima-evm-utils, audit, ng-tools, podman, container-selinux,</td>
</tr>
<tr>
<td></td>
<td>skopeo, criu, slirp4netns, fuse-overlayfs, clevis, clevis-dracut,</td>
</tr>
<tr>
<td></td>
<td>clevis-luks, greenboot, greenboot-grub2, greenboot-rpm-ostree-grub2, greenboot-</td>
</tr>
<tr>
<td></td>
<td>reboot, greenboot-status</td>
</tr>
<tr>
<td>tar</td>
<td>policycoreutils, selinux-policy-targeted</td>
</tr>
<tr>
<td>vhd</td>
<td>@Core, langpacks-en</td>
</tr>
<tr>
<td>vmdk</td>
<td>@Core, chrony, firewalld, kernel, langpacks-en, open-vm-tools, selinux-policy-targeted</td>
</tr>
</tbody>
</table>
NOTE

When you add additional components to your blueprint, you must make sure that the packages in the components you added do not conflict with any other package components, otherwise the system fails to solve dependencies. As a consequence, you are not able to create your customized image.

Additional resources

- Image Builder description

7.3.9. Enabled Services

When you configure the custom image, the services enabled are the defaults services for the RHEL release you are running osbuild-composer from, additionally the services enabled for specific image types.

For example, the .ami image type enables the services sshd, chronyd and cloud-init and without these services, the custom image does not boot.

Table 7.5. Enabled services to support image type creation

<table>
<thead>
<tr>
<th>Image type</th>
<th>Enabled Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>ami</td>
<td>sshd, cloud-init, cloud-init-local, cloud-config, cloud-final</td>
</tr>
<tr>
<td>openstack</td>
<td>sshd, cloud-init, cloud-init-local, cloud-config, cloud-final</td>
</tr>
<tr>
<td>qcow2</td>
<td>cloud-init</td>
</tr>
<tr>
<td>rhel-edge-commit</td>
<td>No extra service enables by default</td>
</tr>
<tr>
<td>tar</td>
<td>No extra service enables by default</td>
</tr>
<tr>
<td>vhd</td>
<td>sshd, chronyd, waagent, cloud-init, cloud-init-local, cloud-config, cloud-final</td>
</tr>
<tr>
<td>vmdk</td>
<td>sshd, chronyd, vmtoolsd</td>
</tr>
</tbody>
</table>

Note: You can customize which services to enable during the system boot. However, for image types with services enabled by default, the customization does not override this feature.

Additional resources

- Supported Image Customizations

7.4. CREATING SYSTEM IMAGES WITH IMAGE BUILDER WEB CONSOLE INTERFACE
Image Builder is a tool for creating custom system images. To control Image Builder and create your custom system images, you can use the web console interface. Note that the command line interface is the currently preferred alternative, because it offers more features.

### 7.4.1. Accessing Image Builder GUI in the RHEL web console

The `cockpit-composer` plugin for the RHEL web console enables users to manage Image Builder blueprints and composes with a graphical interface. Note that the preferred method for controlling Image Builder is at the moment using the command-line interface.

**Prerequisites**

- You must have root access to the system.

**Procedure**

1. Open [https://localhost:9090/](https://localhost:9090/) in a web browser on the system where Image Builder is installed. For more information on how to remotely access Image Builder, see *Managing systems using the RHEL web console* document.

2. Log into the web console with credentials for a user account with sufficient privileges on the system.

3. To display the Image Builder controls, click the **Image Builder** icon, which is in the upper-left corner of the window. The Image Builder view opens, listing existing blueprints.

### 7.4.2. Creating an Image Builder blueprint in the web console interface

To describe the customized system image, create a blueprint first.

**Prerequisites**

- You have opened the Image Builder interface of the RHEL web console in a browser.

**Procedure**

1. Click **Create Blueprint** in the top right corner. A pop-up appears with fields for the blueprint name and description.

2. Fill in the name of the blueprint, its description, then click **Create**. The screen changes to blueprint editing mode.

3. Add components that you want to include in the system image:
   
   a. On the left, enter all or part of the component name in the **Available Components** field and press **Enter**. The search is added to the list of filters under the text entry field, and the list of components below is reduced to those that match the search. If the list of components is too long, add further search terms in the same way.

   b. The list of components is paged. To move to other result pages, use the arrows and entry field above the component list.
Click the name of the component you intend to use to display its details. The right pane fills with details of the components, such as its version and dependencies.

d. Select the version you want to use in the **Component Options** box, with the **Version Release** dropdown.

e. Click **Add** in the top left.

f. If you added a component by mistake, remove it by clicking the ... button at the far right of its entry in the right pane, and select **Remove** in the menu.

**NOTE**

If you do not intend to select a version for some components, you can skip the component details screen and version selection by clicking the + buttons on the right side of the component list.

4. To save the blueprint, click **Commit** in the top right. A dialog with a summary of the changes pops up. Click **Commit**.

   A small pop-up on the right informs you of the saving progress and then the result.

5. To exit the editing screen, click **Back to Blueprints** in the top left.

   The Image Builder view opens, listing existing blueprints.

### 7.4.3. Editing an Image Builder blueprint in the web console interface

To change the specifications for a custom system image, edit the corresponding blueprint.

**Prerequisites**

- You have opened the Image Builder interface of the RHEL web console in a browser.
- A blueprint exists.

**Procedure**

1. Locate the blueprint that you want to edit by entering its name or a part of it into the search box at top left, and press **Enter**.

   The search is added to the list of filters under the text entry field, and the list of blueprints below is reduced to those that match the search.

   If the list of blueprints is too long, add further search terms in the same way.

2. On the right side of the blueprint, press the **Edit Blueprint** button that belongs to the blueprint.

   The view changes to the blueprint editing screen.

3. Remove unwanted components by clicking their ... button at the far right of its entry in the right pane, and select **Remove** in the menu.

4. Change version of existing components:

   a. On the Blueprint Components search field, enter component name or a part of it into the field under the heading **Blueprint Components** and press **Enter**.

      The search is added to the list of filters under the text entry field, and the list of components below is reduced to those that match the search.
If the list of components is too long, add further search terms in the same way.

b. Click the button at the far right of the component entry, and select View in the menu. A component details screen opens in the right pane.

c. Select the desired version in the Version Release drop-down menu and click Apply Change in top right. The change is saved and the right pane returns to listing the blueprint components.

5. Add new components:

a. On the left, enter component name or a part of it into the field under the heading Available Components and press Enter. The search is added to the list of filters under the text entry field, and the list of components below is reduced to those that match the search.

If the list of components is too long, add further search terms in the same way.

b. The list of components is paged. To move to other result pages, use the arrows and entry field above the component list.

c. Click the name of the component you intend to use to display its details. The right pane fills with details of the components, such as its version and dependencies.

d. Select the version you want to use in the Component Options box, with the Version Release drop-down menu.

e. Click Add in the top right.

f. If you added a component by mistake, remove it by clicking the button at the far right of its entry in the right pane, and select Remove in the menu.

NOTE

If you do not intend to select a version for some components, you can skip the component details screen and version selection by clicking the + buttons on the right side of the component list.

6. Commit a new version of the blueprint with your changes:

a. Click the Commit button in top right. A pop-up window with a summary of your changes appears.

b. Review your changes and confirm them by clicking Commit. A small pop-up on the right informs you of the saving progress and the results. A new version of the blueprint is created.

c. In the top left, click Back to Blueprints to exit the editing screen. The Image Builder view opens, listing existing blueprints.

7.4.4. Adding users and groups to an Image Builder blueprint in the web console interface

Adding customizations such as users and groups to blueprints in the web console interface is currently not possible. To work around this limitation, use the Terminal tab in web console to use the command-line interface (CLI) workflow.
Prerequisites

- A blueprint must exist.
- A CLI text editor such as **vim**, **nano**, or **emacs** must be installed. To install them:
  
  ```sh
  # yum install editor-name
  ```

Procedure

1. Find out the name of the blueprint: Open the Image Builder (**Image builder**) tab on the left in the RHEL web console to see the name of the blueprint.

2. Navigate to the CLI in web console: Open the system administration tab on the left, then select the last item **Terminal** from the list on the left.

3. Enter the super-user (root) mode:

   ```sh
   $ sudo bash
   ```

   Provide your credentials when asked. Note that the terminal does not reuse your credentials you entered when logging into the web console.

   A new shell with root privileges starts in your home directory.

4. Export the blueprint to a file:

   ```sh
   # composer-cli blueprints save BLUEPRINT-NAME
   ```

5. Edit the file `BLUEPRINT-NAME.toml` with a CLI text editor of your choice and add the users and groups.

   **IMPORTANT**

   RHEL web console does not have any built-in feature to edit text files on the system, so the use of a CLI text editor is required for this step.

   a. For every user to be added, add this block to the file:

   ```toml
   [[customizations.user]]
   name = "USER-NAME"
   description = "USER-DESCRIPTION"
   password = "PASSWORD-HASH"
   key = "ssh-rsa (...) key-name"
   home = "/home/USER-NAME/"
   shell = "/usr/bin/bash"
   groups = ["users", "wheel"]
   uid = NUMBER
   gid = NUMBER
   ```

   Replace `PASSWORD-HASH` with the actual password hash. To generate the hash, use a command such as this:
$ python3 -c 'import crypt, getpass; pw = getpass.getpass(); print(crypt.crypt(pw) if (pw == getpass.getpass("Confirm: ")) else exit())'

Replace `ssh-rsa (...) key-name` with the actual public key.

Replace the other placeholders with suitable values.

Leave out any of the lines as needed, only the user name is required.

b. For every user group to be added, add this block to the file:

```python
[[customizations.group]]
name = "GROUP-NAME"
gid = NUMBER
```

c. Increase the version number.

d. Save the file and close the editor.

6. Import the blueprint back into Image Builder:

```bash
# composer-cli blueprints push BLUEPRINT-NAME.toml
```

Note that you must supply the file name including the `.toml` extension, while in other commands you use only the name of the blueprint.

7. To verify that the contents uploaded to Image Builder match your edits, list the contents of blueprint:

```bash
# composer-cli blueprints show BLUEPRINT-NAME
```

Check if the version matches what you put in the file and if your customizations are present.

**IMPORTANT**

The Image Builder plugin for RHEL web console does not show any information that could be used to verify that the changes have been applied, unless you also edited the packages included in the blueprint.

8. Exit the privileged shell:

```bash
# exit
```

9. Open the Image Builder (Image builder) tab on the left and refresh the page, in all browsers and all tabs where it was opened.

This prevents state cached in the loaded page from accidentally reverting your changes.

### 7.4.5. Creating a system image with Image Builder in the web console interface

The following steps below describe creating a system image.

**Prerequisites**
You have opened the Image Builder interface of the RHEL web console in a browser.

A blueprint exists.

**Procedure**

1. Locate the blueprint that you want to build an image by entering its name or a part of it into the search box at top left, and press **Enter**.
   The search is added to the list of filters under the text entry field, and the list of blueprints below is reduced to those that match the search.

   If the list of blueprints is too long, add further search terms in the same way.

2. On the right side of the blueprint, press the **Create Image** button that belongs to the blueprint. A pop-up window appears.

3. Select the image type and press **Create**.
   A small pop-up in the top right informs you that the image creation has been added to the queue.

4. Click the name of the blueprint.
   A screen with details of the blueprint opens.

5. Click the **Images** tab to switch to it. The image that is being created is listed with the status **In Progress**.

   **NOTE**

   Image creation takes a longer time, measured in minutes. There is no indication of progress while the image is created.

   To abort image creation, press its **Stop** button on the right.

6. Once the image is successfully created, the **Stop** button is replaced by a **Download** button. Click this button to download the image to your system.

**7.4.6. Adding a source to a blueprint**

The sources defined in Image Builder provide the contents that you can add to blueprints. These sources are global and therefore available to all blueprints. The System sources are repositories that are set up locally on your computer and cannot be removed from Image Builder. You can add additional custom sources and thus be able to access other contents than the System sources available on your system.

The following steps describe how to add a Source to your local system.

**Prerequisites**

- You have opened the Image Builder interface of the RHEL web console in a browser.

**Procedure**

1. Click the **Manage Sources** button in the top right corner.
A pop-up window appears with the available sources, their names and descriptions.

2. On the right side of the pop-up window, click the Add Source button.

3. Add the desired Source name, the Source path, and the Source Type. The Security field is optional.
4. Click Add Source button. The screen shows the available sources window and lists the source you have added.

As a result, the new System source is available and ready to be used or edited.

7.4.7. Creating a user account for a blueprint

The images created by Image Builder have the root account locked and no other accounts included. Such configuration is provided in order to ensure that you cannot accidentally build and deploy an image without a password. Image Builder enables you to create a user account with password for a blueprint so that you can log in to the image created from the blueprint.

Prerequisites

- You have opened the Image Builder interface of the RHEL web console in a browser.
- You have an existing blueprint.

Procedure

1. Locate the blueprint that you want to create a user account for by entering its name or a part of it into the search box at the top left, and press Enter.
   The search is added to the list of filters under the text entry field, and the list of blueprints below is reduced to those that match the search.

2. Click on the blueprint name to display the blueprint details.

3. Click Create User Account.
   This will open a window with fields for user account creation.
4. Fill in the details. Notice that when you insert the name, the **User name** field autocompletes, suggesting a username.

5. Once you have inserted all the desired details, click **Create**.

6. The created user account appears showing all the information you have inserted.

7. To create further user accounts for the blueprint, repeat the process.

**7.4.8. Creating a user account with SSH key**

The images created by Image Builder have the root account locked and no other accounts included. Such configuration is provided in order to ensure that images are secure, by not having a default
password. Image Builder enables you to create a user account with SSH key for a blueprint so that you can authenticate to the image that you created from the blueprint. To do so, first, create a blueprint. Then, you will create a user account with a password and an SSH key. The following example shows how to create a Server administrator user with an SSH key configured.

Prerequisites

- You have created an SSH key that will be paired with the created user later on in the process.
- You have opened the Image Builder interface of the RHEL web console in a browser.
- You have an existing blueprint

Procedure

1. Locate the blueprint that you want to create a user account for by entering its name or a part of it into the search box at the top left, and press Enter. The search is added to the list of filters under the text entry field, and the list of blueprints below is reduced to those that match the search.

2. Click on the blueprint name to display the blueprint details.

3. Click Create User Account.
   This will open a window with fields for user account creation.
4. Fill in the details. Notice that when you insert the name, the **User name** field autocomplete, suggesting a username. If you want to provide administrators rights to the user account you are creating, check the **Role** field.

Paste the content of your public SSH key file.

5. Once you have inserted all the desired details, click **Create**.

6. The new user account will appear in the user list, showing all the information you have inserted.

7. If you want to create more user accounts for the blueprint, repeat the process.

**Additional resources**
7.5. PREPARING AND UPLOADING CLOUD IMAGES WITH IMAGE BUILDER

Image Builder can create custom system images ready for use in clouds of various providers. To use your customized RHEL system image in a cloud, create the system image with Image Builder using the respective output type, configure your system for uploading the image, and upload the image to your cloud account. From Red Hat Enterprise Linux 8.3, the ability to push customized image clouds through the Image Builder application in the RHEL web console is available for a subset of the service providers that we support, such as AWS and Azure clouds. See Pushing images to AWS Cloud AMI and Pushing VHD images to Azure cloud.

7.5.1. Preparing for uploading AWS AMI images

This describes steps to configure a system for uploading AWS AMI images.

Prerequisites

- You must have an Access Key ID configured in the AWS IAM account manager.
- You must have a writable S3 bucket prepared.

Procedure

1. Install Python 3 and the pip tool:
   
   ```
   # yum install python3
   # yum install python3-pip
   ```

2. Install the AWS command-line tools with pip:
   
   ```
   # pip3 install awscli
   ```

3. Run the following command to set your profile. The terminal prompts you to provide your credentials, region and output format:
   
   ```
   $ aws configure
   AWS Access Key ID [None]:
   AWS Secret Access Key [None]:
   Default region name [None]:
   Default output format [None]:
   ```

4. Define a name for your bucket and use the following command to create a bucket:
   
   ```
   $ BUCKET=bucketname
   $ aws s3 mb s3://$BUCKET
   ```

   Replace `bucketname` with the actual bucket name. It must be a globally unique name. As a result, your bucket is created.
5. Then, to grant permission to access the S3 bucket, create a **vmimport** S3 Role in IAM, if you have not already done so in the past:

```bash


$ aws iam create-role --role-name vmimport --assume-role-policy-document file://trust-policy.json
$ aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document file://role-policy.json
```

### Additional resources

- **Using high-level (s3) commands with the AWS CLI**

#### 7.5.2. Uploading an AMI image to AWS in the CLI

You can use Image Builder to build `.ami` images and push them directly to Amazon AWS Cloud service provider using the CLI.

### Prerequisites

- You have an **Access Key ID** configured in the **AWS IAM** account manager.

- You have a writable **S3 bucket** prepared.

- You have a defined blueprint.

### Procedure

1. Using the text editor of your choice, create a configuration file with the following content:

```toml
provider = "aws"

[settings]
accessKeyId = "AWS_ACCESS_KEY_ID"
secretAccessKey = "AWS_SECRET_ACCESS_KEY"
bucket = "AWS_BUCKET"
region = "AWS_REGION"
key = "IMAGE_KEY"
```

Replace values in the fields with your credentials for **accessKeyId**, **secretAccessKey**, **bucket**, **region**. The **IMAGE_KEY** value is the name of your VM Image to be uploaded to EC2.

2. Save the file as `CONFIGURATION-FILE.toml` and close the text editor.

3. Start the compose:

```bash
# composer-cli compose start BLUEPRINT-NAME IMAGE-TYPE IMAGE_KEY CONFIGURATION-FILE.toml
```
Replace:

- **BLUEPRINT-NAME** with the name of the blueprint you created
- **IMAGE-TYPE** with the `ami` image type.
- **IMAGE_KEY** with the name of your VM Image to be uploaded to EC2.
- **CONFIGURATION-FILE.toml** with the name of the configuration file of the cloud provider.

**NOTE**

You must have the correct IAM settings for the bucket you are going to send your customized image to. You have to set up a policy to your bucket before you are able to upload images to it.

4. Check the status of the image build and upload it to AWS:

```bash
# composer-cli compose status
```

After the image upload process is complete, you can see the "FINISHED" status.

**Verification**

To confirm that the image upload was successful:

1. Access **EC2** on the menu and choose the correct region in the AWS console. The image must have the "available" status, to indicate that it was successfully uploaded.

2. On the dashboard, select your image and click **Launch**.

**Additional Resources**

- Required service role

**7.5.3. Pushing images to AWS Cloud AMI**

The ability to push the output image that you create to **AWS Cloud AMI** is available this time. This describes steps to push `.ami` images you create using Image Builder to Amazon AWS Cloud service provider.

**Prerequisites**

- You must have **root** or **wheel** group user access to the system.
- You have opened the Image Builder interface of the RHEL 8 web console in a browser.
- You must have an Access Key ID configured in the **AWS IAM** account manager.
- You must have a writable **S3 bucket** prepared.

**Procedure**
1. Click **Create blueprint** to create a blueprint. See *Creating an Image Builder blueprint in the web console interface*.

2. Select the components and packages that you want as part of the image you are creating.

3. Click **Commit** to commit the changes you made to the blueprint.
   A small pop-up on the superior right side informs you of the saving progress and then the result of the changes you committed.

4. Click **blueprint name** link on the left banner.

5. Select the tab **Images**.

6. Click **Create Image** to create your customized image.
   A pop-up window opens.
   a. From the "**Type**" drop-down menu list, select the "Amazon Machine Image Disk (.ami)" image.
   b. Check the "**Upload to AWS**" check box to upload your image to the AWS Cloud and click **Next**.
   c. To authenticate your access to AWS, type your "AWS access key ID" and "AWS secret access key" in the corresponding fields. Click **Next**.

   **NOTE**
   You can view your AWS secret access key only when you create a new Access Key ID. If you do not know your Secret Key, generate a new Access Key ID.

   d. Type the name of the image in the "**Image name**" field, type the Amazon bucket name in the "Amazon S3 bucket name" field and type the "AWS region" field for the bucket you are going to add your customized image to. Click **Next**.

   e. Review the information you provided and once you are satisfied, click **Finish**.
   Optionally, you can click **Back** to modify any incorrect detail.

   **NOTE**
   You must have the correct IAM settings for the bucket you are going to send your customized image. We are using the IAM Import and Export, so you have to set up a **policy** to your bucket before you are able to upload images to it.
   For more information, see *Required Permissions for IAM Users*.

7. A small pop-up on the superior right side informs you of the saving progress. It also informs that the image creation has been initiated, the progress of this image creation and the subsequent upload to the AWS Cloud.
   Once the process is complete, you can see the "**Image build complete**" status.

8. Click **Service→EC2** on the menu and choose the **correct region** in the AWS console. The image must have the “Available” status, to indicate that it is uploaded.

9. On the dashboard, select your image and click **Launch**.
10. A new window opens. Choose an instance type according to the resources you need to launch your image. Click **Review and Launch**.

11. Review your instance launch details. You can edit each section if you need to make any change. Click **Launch**.

12. Before you launch the instance, you must select a public key to access it. You can either use the key pair you already have or you can create a new key pair. Alternatively, you can use **Image Builder** to add a user to the image with a preset public key. See [Creating a user account with SSH key](#) for more details.

Follow the next steps to create a new key pair in EC2 and attach it to the new instance.

   a. From the drop-down menu list, select "Create a new key pair".
   
   b. Enter the name to the new key pair. It generates a new key pair.
   
   c. Click "Download Key Pair" to save the new key pair on your local system.

13. Then, you can click **Launch Instance** to launch your instance.

You can check the status of the instance, it shows as "Initializing".

14. Once the instance status is "running", the **Connect** button becomes available.

15. Click **Connect**. A popup window appears with instructions on how to connect using SSH.

   a. Select the preferred connection method to "A standalone SSH client" and open a terminal.
   
   b. In the location you store your private key, make sure that your key is publicly viewable for SSH to work. To do so, run the command:

   ```bash
   $ chmod 400 <your-instance-name.pem>_
   ```

   c. Connect to your instance using its Public DNS:

   ```bash
   $ ssh -i "<your-instance-name.pem>" ec2-user@<your-instance-IP-address>
   ```

   d. Type "yes" to confirm that you want to continue connecting. As a result, you are connected to your instance using SSH.

**Verification steps**

1. Check if you are able to perform any action while connected to your instance using SSH.

**Additional resources**

- Open a case on Red Hat Customer Portal
- Connecting to your Linux instance using SSH
- Open support cases

**7.5.4. Preparing for uploading Azure VHD images**

This describes steps to upload an VHD image to Azure.
Prerequisites

- You must have a usable Azure resource group and storage account.

Procedure

1. Install python2:

   ```
   # yum install python2
   ```

   **NOTE**
   
   python2 package must be installed because since the AZ CLI depends specifically on python 2.7

2. Import the Microsoft repository key:

   ```
   # rpm --import https://packages.microsoft.com/keys/microsoft.asc
   ```

3. Create a local azure-cli repository information:

   ```
   # sh -c 'echo -e "[azure-cli]\nname=Azure CLI\nbaseurl=https://packages.microsoft.com/yumrepos/azure-cli\nenabled=1\ngpgcheck=1\ngpgkey=https://packages.microsoft.com/keys/microsoft.asc" > /etc/yum.repos.d/azure-cli.repo'
   ```

4. Install the Azure CLI:

   ```
   # yumdownloader azure-cli
   # rpm -ivh --nodeps azure-cli-2.0.64-1.el7.x86_64.rpm
   ```

   **NOTE**
   
   The downloaded version of the Azure CLI package may vary depending on the current downloaded version.

5. Run the Azure CLI:

   ```
   $ az login
   ```

   The terminal shows the message 'Note, we have launched a browser for you to login. For old experience with device code, use "az login --use-device-code" and open a browser where you can login.

   **NOTE**
   
   If you are running a remote (SSH) session, the link will not open in the browser. In this case, you can use the link provided and thus be able to login and authenticate your remote session. To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code XXXXXXXXXX to authenticate.
6. List the keys for the storage account in Azure:

```bash
$ GROUP=resource-group-name
$ ACCOUNT=storage-account-name
$ az storage account keys list --resource-group $GROUP --account-name $ACCOUNT
```

Replace `resource-group-name` with name of the Azure resource group and `storage-account-name` with name of the Azure storage account.

**NOTE**

You can list the available resources using the command:

```bash
$ az resource list
```

7. Make note of value `key1` in the output of the previous command, and assign it to an environment variable:

```bash
$ KEY1=value
```

8. Create a storage container:

```bash
$ CONTAINER=storage-account-name
$ az storage container create --account-name $ACCOUNT --account-key $KEY1 --name $CONTAINER
```

Replace `storage-account-name` with name of the storage account.

**Additional resources**

- [Azure CLI](#)

### 7.5.5. Uploading VHD images to Azure

This describes steps to upload an VHD image to Azure.

**Prerequisites**

- Your system must be set up for uploading Azure VHD images.
- You must have an Azure VHD image created by Image Builder. Use the `vhd` output type in CLI or `Azure Disk Image (.vhd)` in GUI when creating the image.

**Procedure**

1. Push the image to Azure and create an instance from it:

```bash
$ VHD=25ccb8dd-3872-477f-9e3d-c2970cd4bbaf-disk.vhd
$ az storage blob upload --account-name $ACCOUNT --container-name $CONTAINER --file $VHD --name $VHD --type page
...
2. Once the upload to the Azure BLOB completes, create an Azure image from it:

```
$ az image create --resource-group $GROUP --name $VHD --os-type linux --location eastus
  --source https://$ACCOUNT.blob.core.windows.net/$CONTAINER/$VHD
  - Running ...
```

3. Create an instance either with the Azure portal, or a command similar to the following:

```
$ az vm create --resource-group $GROUP --location eastus --name $VHD --image $VHD
  --admin-username azure-user --generate-ssh-keys
  - Running ...
```

4. Use your private key via SSH to access the resulting instance. Log in as `azure-user`.

### 7.5.6. Uploading VMDK images to vSphere

Image Builder can generate images suitable for uploading to a VMware ESXi or vSphere system. This describes steps to upload an VMDK image to VMware vSphere.

**NOTE**

Because VMWare deployments typically do not have `cloud-init` configured to inject user credentials to virtual machines, we must perform that task ourselves on the blueprint.

**Prerequisites**

- You must have an VMDK image created by Image Builder. Use the `vmdk` output type in CLI or `VMware Virtual Machine Disk (.vmdk)` in GUI when creating the image.

**Procedure**

1. Upload the image into vSphere via HTTP. Click on **Upload Files** in the vCenter:
2. When you create a VM, on the **Device Configuration**, delete the default **New Hard Disk** and use the drop-down to select an **Existing Hard Disk** disk image:

3. Make sure you use an **IDE** device as the **Virtual Device Node** for the disk you create. The default value **SCSI** results in an unbootable virtual machine.
7.5.7. Pushing VMWare images to vSphere

You can build VMWare images and push them directly to your vSphere instance, to avoid having to download the image file and push it manually. This describes steps to push .vmdk images you create using Image Builder directly to vSphere instances service provider.

Prerequisites

- You have root or wheel group user access to the system.
- You have opened the Image Builder interface of the RHEL 8 web console in a browser.
- You have a vSphere Account.

Procedure

1. Click Create blueprint.
   See Creating an Image Builder blueprint in the web console interface.

2. Select the components and packages that you want as part of the image you are creating.

3. Click Commit to commit the changes you made to the blueprint.
   A pop-up on the upper right side informs you of the saving progress and then the result of the changes you committed.

4. Click blueprint name link on the left banner.

5. Select the Customizations tab to create a user account for the blueprint.
   See Creating a user account for a blueprint.

6. Select the Images tab.

7. Click Create Image to create your customized image.
   The Image type window opens.

8. In the Image type window:
   a. From the dropdown menu, select the Type: VMWare VSphere (.vmdk).
   b. Check the Upload to VMware checkbox to upload your image to the vSphere.
   c. Optional: Set the size of the image you want to instantiate. The minimal default size is 2GB.
   d. Click Next.

9. In the Upload to VMware window, under Authentication, enter the following details:
   a. Username: username of the vSphere account.
   b. Password: pasword of the vSphere account.

10. In the Upload to VMware window, under Destination, enter the following details:
    a. Image name: a name for the image to be uploaded.
    b. Host: The URL of your VMWare vSphere where the image will be uploaded.
c. **Cluster:** The name of the cluster where the image will be uploaded.

d. **Data center:** The name of the datacenter where the image will be uploaded.

e. **Data store:** The name of the Data store where the image will be uploaded.

f. Click **Next.**

11. In the **Review** window, review the details about the image creation. Once you are satisfied, click **Finish.**

You can click **Back** to modify any incorrect detail.

Image Builder adds the compose of a RHEL 8.4 vSphere image to the queue, and creates and uploads the image to the **Cluster** on the vSphere instance you specified.

**NOTE**

The image build and upload processes take a few minutes to complete.

Once the process is complete, you can see the **Image build complete** status.

**Verification**

After the image status upload is completed successfully, you can create a Virtual Machine (VM) from the image you uploaded and login into it. Follow the steps for that:

1. Access VMWare vSphere Client.

2. Search for the image in the **Cluster** on the vSphere instance you specified.

3. You can create a new Virtual Machine from the image you uploaded. For that:
   a. Select the image you uploaded.

   b. Click the right button on the selected image.

   c. Click **New Virtual Machine.**

   A **New Virtual Machine** window opens.

   In the **New Virtual Machine** window, provide the following details:

   i. Select a creation type: You can choose to create a New Virtual Machine.

   ii. Select a name and a folder: For example, Virtual Machine name: **vSphere Virtual Machine** and location of your choice inside vSphere Client.

   iii. Select a computer resource: choose a destination computer resource for this operation.

   iv. Select storage: For example, select NFS-Node1

   v. Select compatibility: The image should be BIOS only.

   vi. Select a guest OS: For example, select **Linux and _Red Hat Fedora (64-bit).**

   vii. **Customize hardware:** When you create a VM, on the **Device Configuration** button on the upper right, delete the default New Hard Disk and use the drop-down to select an Existing Hard Disk disk image:
viii. Ready to complete: Review the details and click **Finish** to create the image.

d. Navigate to the **VMs** tab.

  i. From the list, select the VM you created.

  ii. Click the **Start** button from the panel. A new window appears, showing the VM image loading.

  iii. Log in with the credentials you created for the blueprint.

  iv. You can verify if the packages you added to the blueprint are installed. For example:

        $ rpm -qa | grep firefox

**Additional resources**

- [Installing the vSphere Client](#).

### 7.5.8. Pushing VHD images to Azure cloud

The ability to push the output image you create to the Azure Blob Storage is available. This section describes steps to push `.vhd` images you create using Image Builder to Azure Cloud service provider.

**Prerequisites**

- You must have root access to the system.
- You have opened the Image Builder interface of the RHEL 8 web console in a browser.
- You must have a **Storage Account** created.
- You must have a writable **Blob Storage** prepared.

**Procedure**

1. Click **Create blueprint** to create a blueprint. See [Creating an Image Builder blueprint in the web console interface](#).

2. Select the components and packages that you want as part of the image you are creating.

3. Click **Commit** to commit the changes you made to the blueprint.

   A small pop-up on the upper right side informs you of the saving progress and then the result of the changes you committed.

4. Click **blueprint name** link on the left banner.

5. Select the tab **Images**.

6. Click **Create Image** to create your customized image.

   A pop-up window opens.

   a. From the "**Type**" drop-down menu list, select the **Azure Disk Image (.vhd)** image.

   b. Check the "**Upload to Azure**" check box to upload your image to the Azure Cloud and click **Next**.
c. To authenticate your access to Azure, type your "Storage account" and "Storage access key" in the corresponding fields. Click Next.
   You can find your Storage account details in the Settings→Access Key menu list.

d. Type a "Image name" to be used for the image file that will be uploaded and the Blob "Storage container" in which the image file you want to push the image into. Click Next.

e. Review the information you provided and once you are satisfied, click Finish. Optionally, you can click Back to modify any incorrect detail.

7. A small pop-up on the upper right side displays when the image creation process starts with the message: "Image creation has been added to the queue". Once the image process creation is complete, click the blueprint you created an image from. You can see the "Image build complete" status for the image you created within the Images tab.

8. To access the image you pushed into Azure Cloud, access Azure Portal.

9. On the search bar, type Images and select the first entry under Services. You are redirected to the Image dashboard.

10. Click +Add. You are redirected to the Create an Image dashboard. Insert the below details:

a. Name: Choose a name for your new image.

b. Resource Group: Select a resource group.

c. Location: Select the location that matches the regions assigned to your storage account. Otherwise you will not be able to select a blob.

d. OS Type: Set the OS type to Linux.

e. VM Generation: Keep the VM generation set on Gen 1.

f. Storage Blob: Click Browse on the right of Storage blob input. Use the dialog to find the image you uploaded earlier. Keep the remaining fields as in the default choice.

11. Click Create to create the image. Once the image is created, you can see the message "Successfully created image" in the upper right corner.

12. Click Refresh to see your new image and open your newly created image.

13. Click + Create VM. You are redirected to the Create a virtual machine dashboard.

14. In the Basic tab, under Project Details, your Subscription and the Resource Group are already pre-set. If you want to create a new resource Group

a. Click Create new. A pop-up prompts you to create the Resource Group Name container.

b. Insert a name and click OK. If you want to keep the Resource Group that is already pre-set.

15. Under Instance Details, insert:
a. **Virtual machine name**

b. **Region**

c. **Image**: The image you created is pre-selected by default.

d. **Size**: Choose a VM size that better suits your needs. Keep the remaining fields as in the default choice.

16. Under **Administrator account**, enter the below details:

   a. **Username**: the name of the account administrator.

   b. **SSH public key source** from the drop-down menu, select **Generate new key pair**. You can either use the key pair you already have or you can create a new key pair. Alternatively, you can use **Image Builder** to add a user to the image with a preset public key. See [Creating a user account with SSH key](#) for more details.

   c. **Key pair name**: insert a name for the key pair.

17. Under **Inbound port rules**, select:

   a. **Public inbound ports**: Allow selected ports.

   b. **Select inbound ports**: Use the default set **SSH (22)**.

18. Click **Review + Create**. You are redirected to the **Review + create** tab and receive a confirmation that the validation passed.

19. Review the details and click **Create**. Optionally, you can click **Previous** to fix previous options selected.

20. A pop-up **generates new key pair** window opens. Click **Download private key and create resources**. Save the key file as "yourKey.pem".

21. Once the deployment is complete, click **Go to resource**.

22. You are redirected to a new window with your VM details. Select the public IP address on the top right side of the page and copy it to your clipboard.

Now, to create an SSH connection with the VM to connect to the Virtual Machine.

1. Open a terminal.

2. At your prompt, open an SSH connection to your virtual machine. Replace the IP address with the one from your VM, and replace the path to the .pem with the path to where the key file was downloaded.

   ```bash
   # ssh -i ./Downloads/yourKey.pem azureuser@10.111.12.123
   ```

3. You are required to confirm if you want to continue to connect. Type yes to continue.

As a result, the output image you pushed to the Azure Storage Blob is ready to be provisioned.

**Additional resources**
7.5.9. Uploading QCOW2 image to OpenStack

Image Builder can generate images suitable for uploading to OpenStack cloud deployments, and starting instances there. This describes steps to upload an QCOW2 image to OpenStack.

Prerequisites

- You must have an OpenStack-specific image created by Image Builder. Use the `openstack` output type in CLI or OpenStack Image (.qcow2) in GUI when creating the image.

```
WARNING

Image Builder also offers a generic QCOW2 image type output format as `qcow2` or QEMU QCOW2 Image (.qcow2). Do not mistake it with the OpenStack image type which is also in the QCOW2 format, but contains further changes specific to OpenStack.
```

Procedure

1. Upload the image to OpenStack and start an instance from it. Use the `Images` interface to do this:
2. Start an instance with that image:
3. You can run the instance using any mechanism (CLI or OpenStack web UI) from the snapshot. Use your private key via SSH to access the resulting instance. Log in as `cloud-user`.

### 7.5.10. Preparing for uploading images to Alibaba

This section describes steps to verify custom images that you can deploy on Alibaba Cloud. The images will need a specific configuration to boot successfully, because Alibaba Cloud requests the custom images to meet certain requirements before you use it. For this, it is recommended that you use the Alibaba `image_check` tool.

**NOTE**

The custom image verification is an optional task. Image Builder generates images that conform to Alibaba’s requirements.

**Prerequisites**

- You must have an Alibaba image created by Image Builder.

**Procedure**

1. Connect to the system containing the image you want to check it by the Alibaba `image_check` tool.
2. Download the `image_check` tool:

   ```
   $ curl -O http://docs-aliyun.cn-hangzhou.oss.aliyun-inc.com/assets/attach/73848/cn_zh/1557459863884/image_check
   ```

3. Change the file permission of the image compliance tool:

   ```
   # chmod +x image_check
   ```

4. Run the command to start the image compliance tool checkup:

   ```
   # ./image_check
   ```

   The tool verifies the system configuration and generates a report that is displayed on your screen. The `image_check` tool saves this report in the same folder where the image compliance tool is running.

5. If any of the Detection Items fail, follow the instructions to correct it. For more information, see link: Detection items section.

Additional resources

- Image Compliance Tool

7.5.11. Uploading images to Alibaba

This section describes how to upload an Alibaba image to Object Storage Service (OSS).

Prerequisites

- Your system is set up for uploading Alibaba images.
- You must have an Alibaba image created by Image Builder. Use the `ami` output type on RHEL 7 or Alibaba on RHEL 8 when creating the image.
- You have a bucket. See Creating a bucket.
- You have an active Alibaba Account.
- You activated OSS.

Procedure

1. Log in to the OSS console.

2. On the left side Bucket menu, select the bucket to which you want to upload an image.

3. On the right upper menu, click the Files tab.

4. Click Upload. A window dialog opens on the right side. Choose the following information:
   
   - **Upload To**: Choose to upload the file to the Current directory or to a Specified directory.
   - **File ACL**: Choose the type of permission of the uploaded file.
5. Click **Upload**.

6. Choose the image you want to upload.

7. Click **Open**.

As a result, the custom image is uploaded to OSS Console.

**Additional resources**

- Upload an object
- Creating an instance from custom images
- Importing images

### 7.5.12. Importing images to Alibaba

This section describes how to import an Alibaba image to Elastic Cloud Console (ECS).

**Prerequisites**

- You have uploaded the image to Object Storage Service (OSS).

**Procedure**

1. Log in to the **ECS console**.
   
   i. On the left side menu, click **Images**.
   
   ii. On the right upper side, click **Import Image**. A window dialog opens.
   
   iii. Confirm that you have set up the correct region where the image is located. Enter the following information:
      
      a. **OSS Object Address**: See how to obtain **OSS Object Address**.
      
      b. **Image Name**:
      
      c. **Operating System**:
      
      d. **System Disk Size**:
      
      e. **System Architecture**:
      
      f. **Platform**: Red Hat

   iv. Optionally, provide the following details:
      
      g. **Image Format**: qcow2 or ami, depending on the uploaded image format.
      
      h. **Image Description**:
      
      i. **Add Images of Data Disks**
      
      The address can be determined in the OSS management console after selecting the required bucket in the left menu, select Files section and then click the **Details** link on the right for the appropriate image. A window will appear on the right side of the screen,
2. Click **OK**.

**NOTE**

The importing process time can vary depending on the image size.

As a result, the custom image is imported to ECS Console. You can create an instance from the custom image.

**Additional resources**

- Notes for importing images
- Creating an instance from custom images
- Upload an object

**7.5.13. Creating an instance of a custom image using Alibaba**

You can create instances of the custom image using Alibaba ECS Console.

**Prerequisites**

- You have activated OSS and uploaded your custom image.
- You have successfully imported your image to ECS Console.

**Procedure**

1. Log in to the [ECS console](https://example.com).
2. On the left side menu, choose **Instances**.
3. In the top corner, click **Create Instance**. You are redirected to a new window.
4. Fill in all the required information. See [Creating an instance by using the wizard](https://example.com) for more details.
5. Click **Create Instance** and confirm the order.

**NOTE**

You can see the option **Create Order** instead of **Create Instance**, depending on your subscription.

As a result, you have an active instance ready for deployment.

**Additional resources**

- Creating an instance by using a custom image
- Create an instance by using the wizard
CHAPTER 8. PERFORMING AN AUTOMATED INSTALLATION USING KICKSTART

8.1. KICKSTART INSTALLATION BASICS

The following provides basic information about Kickstart and how to use it to automate installing Red Hat Enterprise Linux.

8.1.1. What are Kickstart installations

Kickstart provides a way to automate the RHEL installation process, either partially or fully.

Kickstart files contain some or all of the RHEL installation options. For example, the time zone, how the drives should be partitioned, or which packages should be installed. Providing a prepared Kickstart file allows an installation without the need for any user intervention. This is especially useful when deploying Red Hat Enterprise Linux on a large number of systems at once.

Kickstart files also provide more options regarding software selection. When installing Red Hat Enterprise Linux manually using the graphical installation interface, the software selection is limited to pre-defined environments and add-ons. A Kickstart file allows you to install or remove individual packages as well.

Kickstart files can be kept on a single server system and read by individual computers during the installation. This installation method supports the use of a single Kickstart file to install Red Hat Enterprise Linux on multiple machines, making it ideal for network and system administrators.

All Kickstart scripts and log files of their execution are stored in the /tmp directory of the newly installed system to assist with debugging installation issues. The kickstart used for installation as well as the Anaconda generated output kickstart are stored in [filename]/root` on the target system and that logs from kickstart scriptlet execution are stored in /var/log/anaconda.

NOTE

In previous versions of Red Hat Enterprise Linux, Kickstart could be used for upgrading systems. Starting with Red Hat Enterprise Linux 7, this functionality has been removed and system upgrades are instead handled by specialized tools. For details on upgrading to Red Hat Enterprise Linux 8, see Upgrading from RHEL 7 to RHEL 8 and Considerations in adopting RHEL.

8.1.2. Automated installation workflow

Kickstart installations can be performed using a local DVD, a local hard drive, or a NFS, FTP, HTTP, or HTTPS server. This section provides a high level overview of Kickstart usage.

1. Create a Kickstart file. You can write it by hand, copy a Kickstart file saved after a manual installation, or use an online generator tool to create the file, and edit it afterward. See Creating Kickstart files.

2. Make the Kickstart file available to the installation program on removable media, a hard drive or a network location using an HTTP(S), FTP, or NFS server. See Making Kickstart files available to the installation program.

3. Create the boot medium which will be used to begin the installation. See Creating a bootable installation medium and Preparing to install from the network using PXE.
4. Make the installation source available to the installation program. See Creating installation sources for Kickstart installations.

5. Start the installation using the boot medium and the Kickstart file. See Starting Kickstart installations.

If the Kickstart file contains all mandatory commands and sections, the installation finishes automatically. If one or more of these mandatory parts are missing, or if an error occurs, the installation requires manual intervention to finish.

NOTE

If you plan to install a Beta release of Red Hat Enterprise Linux, on systems having UEFI Secure Boot enabled, then first disable the UEFI Secure Boot option and then begin the installation.

UEFI Secure Boot requires that the operating system kernel is signed with a recognized private key, which the system’s firmware verifies using the corresponding public key. For Red Hat Enterprise Linux Beta releases, the kernel is signed with a Red Hat Beta-specific private key, which the system fails to recognize by default. As a result, the system fails to boot the installation media.

8.2. CREATING KICKSTART FILES

You can create a Kickstart file using the following methods:

- Use the online Kickstart configuration tool.
- Copy the Kickstart file created as a result of a manual installation.
- Write the entire Kickstart file manually. Note that editing an already existing file from the other methods is faster, so this method is not recommended.
- Convert the Red Hat Enterprise Linux 7 Kickstart file for Red Hat Enterprise Linux 8 installation.
- In case of virtual and cloud environment, create a custom system image, using Image Builder.

Note that some highly specific installation options can be configured only by manual editing of the Kickstart file.

8.2.1. Creating a Kickstart file with the Kickstart configuration tool

Users with a Red Hat Customer Portal account can use the Kickstart Generator tool in the Customer Portal Labs to generate Kickstart files online. This tool will walk you through the basic configuration and enables you to download the resulting Kickstart file.

NOTE

The tool currently does not support any advanced partitioning.

Prerequisites

- You must have a Red Hat Customer Portal account and an active Red Hat subscription.
Procedure

1. Open the Kickstart generator lab information page at https://access.redhat.com/labsinfo/kickstartconfig

2. Click the Go to Application button to the left of heading and wait for the next page to load.

3. Select Red Hat Enterprise Linux 8 in the drop-down menu and wait for the page to update.

4. Describe the system to be installed using the fields in the form.
   You can use the links on the left side of the form to quickly navigate between sections of the form.

5. To download the generated Kickstart file, click the red Download button at the top of the page.
   Your web browser saves the file.

8.2.2. Creating a Kickstart file by performing a manual installation

The recommended approach to creating Kickstart files is to use the file created by a manual installation of Red Hat Enterprise Linux. After an installation completes, all choices made during the installation are saved into a Kickstart file named anaconda-ks.cfg, located in the /root/ directory on the installed system. You can use this file to reproduce the installation in the same way as before. Alternatively, copy this file, make any changes you need, and use the resulting configuration file for further installations.

Procedure

1. Install RHEL. For more details, see Performing a standard RHEL installation. During the installation, create a user with administrator privileges.

2. Finish the installation and reboot into the installed system.

3. Log into the system with the administrator account.

4. Copy the file /root/anaconda-ks.cfg to a location of your choice.

   IMPORTANT
   The file contains information about users and passwords.

   • To display the file contents in terminal:

     # cat /root/anaconda-ks.cfg

     You can copy the output and save to another file of your choice.

   • To copy the file to another location, use the file manager. Remember to change permissions on the copy, so that the file can be read by non-root users.

   Additional resources
   • Performing a standard RHEL installation

8.2.3. Converting a RHEL 7 Kickstart file for RHEL 8 installation
You can use the Kickstart Converter tool to convert a RHEL 7 Kickstart file for use in a new RHEL 8 installation. For more information about the tool and how to use it to convert a RHEL 7 Kickstart file, see https://access.redhat.com/labs/kickstartconvert/

8.2.4. Creating a custom image using Image Builder

You can use Red Hat Image Builder to create a customized system image for virtual and cloud deployments.

For more information about creating customized images, using Image Builder, see Composing a customized RHEL system image document.

8.3. MAKING KICKSTART FILES AVAILABLE TO THE INSTALLATION PROGRAM

The following provides information about making the Kickstart file available to the installation program on the target system.

8.3.1. Ports for network-based installation

The following table lists the ports that must be open on the server providing the files for each type of network-based installation.

Table 8.1. Ports for network-based installation

<table>
<thead>
<tr>
<th>Protocol used</th>
<th>Ports to open</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>80</td>
</tr>
<tr>
<td>HTTPS</td>
<td>443</td>
</tr>
<tr>
<td>FTP</td>
<td>21</td>
</tr>
<tr>
<td>NFS</td>
<td>2049, 111, 20048</td>
</tr>
<tr>
<td>TFTP</td>
<td>69</td>
</tr>
</tbody>
</table>

Additional resources

- Securing networks

8.3.2. Making a Kickstart file available on an NFS server

This procedure describes how to store the Kickstart script file on an NFS server. This method enables you to install multiple systems from a single source without having to use physical media for the Kickstart file.

Prerequisites

- You must have administrator level access to a server with Red Hat Enterprise Linux 8 on the local network.
The system to be installed must be able to connect to the server.

Firewall on the server must allow connections from the system you are installing to.

**Procedure**

1. Install the `nfs-utils` package by running the following command as root:

   ```
   # yum install nfs-utils
   ```

2. Copy the Kickstart file to a directory on the NFS server.

3. Open the `/etc/exports` file using a text editor and add a line with the following syntax:

   ```
   /exported_directory/ clients
   ```

4. Replace `/exported_directory/` with the full path to the directory holding the Kickstart file. Instead of `clients`, use the host name or IP address of the computer that is to be installed from this NFS server, the subnetwork from which all computers are to have access the ISO image, or the asterisk sign (`*`) if you want to allow any computer with network access to the NFS server to use the ISO image. See the `exports(5)` man page for detailed information about the format of this field.

   A basic configuration that makes the `/rhel8-install/` directory available as read-only to all clients is:

   ```
   /rhel8-install *
   ```

5. Save the `/etc/exports` file and exit the text editor.

6. Start the nfs service:

   ```
   # systemctl start nfs-server.service
   ```

   If the service was running before you changed the `/etc/exports` file, enter the following command, in order for the running NFS server to reload its configuration:

   ```
   # systemctl reload nfs-server.service
   ```

   The Kickstart file is now accessible over NFS and ready to be used for installation.

**NOTE**

When specifying the Kickstart source, use `nfs:` as the protocol, the server’s host name or IP address, the colon sign (`:`), and the path inside directory holding the file. For example, if the server’s host name is `myserver.example.com` and you have saved the file in `/rhel8-install/my-ks.cfg`, specify `inst.ks=nfs:myserver.example.com:/rhel8-install/my-ks.cfg` as the installation source boot option.

**Additional resources**

- Preparing to install from the network using PXE

**8.3.3. Making a Kickstart file available on an HTTP or HTTPS server**
This procedure describes how to store the Kickstart script file on an HTTP or HTTPS server. This method enables you to install multiple systems from a single source without having to use physical media for the Kickstart file.

Prerequisites

- You must have administrator level access to a server with Red Hat Enterprise Linux 8 on the local network.
- The system to be installed must be able to connect to the server.
- Firewall on the server must allow connections from the system you are installing to.

Procedure

1. To store the Kickstart file on an HTTP, install the `httpd` package:
   ```bash
   # yum install httpd
   ```
   To store the Kickstart file on an HTTPS, install `httpd` and `mod_ssl` packages:
   ```bash
   # yum install httpd mod_ssl
   ```

   **WARNING**
   If your Apache web server configuration enables SSL security, verify that you only enable the TLSv1 protocol, and disable SSLv2 and SSLv3. This is due to the POODLE SSL vulnerability (CVE-2014-3566). See [https://access.redhat.com/solutions/1232413](https://access.redhat.com/solutions/1232413) for details.

   **IMPORTANT**
   If you use an HTTPS server with a self-signed certificate, you must boot the installation program with the `inst.noverifyssl` option.

2. Copy the Kickstart file to the HTTP(S) server into a subdirectory of the `/var/www/html/` directory.

3. Start the `httpd` service:
   ```bash
   # systemctl start httpd.service
   ```
   The Kickstart file is now accessible and ready to be used for installation.
NOTE

When specifying the location of the Kickstart file, use http:// or https:// as the protocol, the server’s host name or IP address, and the path of the Kickstart file, relative to the HTTP server root. For example, if you are using HTTP, the server’s host name is myserver.example.com, and you have copied the Kickstart file as /var/www/html/rhel8-install/my-ks.cfg, specify http://myserver.example.com/rhel8-install/my-ks.cfg as the file location.

Additional resources

- Deploying different types of servers

8.3.4. Making a Kickstart file available on an FTP server

This procedure describes how to store the Kickstart script file on an FTP server. This method enables you to install multiple systems from a single source without having to use physical media for the Kickstart file.

Prerequisites

- You must have administrator level access to a server with Red Hat Enterprise Linux 8 on the local network.
- The system to be installed must be able to connect to the server.
- Firewall on the server must allow connections from the system you are installing to.

Procedure

1. Install the vsftpd package by running the following command as root:

   # yum install vsftpd

2. Open and edit the /etc/vsftpd/vsftpd.conf configuration file in a text editor.
   a. Change the line anonymous_enable=NO to anonymous_enable=YES
   b. Change the line write_enable=YES to write_enable=NO.
   c. Add lines pasv_min_port=\textit{min\_port} and pasv_max_port=\textit{max\_port}. Replace \textit{min\_port} and \textit{max\_port} with the port number range used by FTP server in passive mode, e.g., 10021 and 10031.
      This step can be necessary in network environments featuring various firewall/NAT setups.
   d. Optionally, add custom changes to your configuration. For available options, see the vsftpd.conf(5) man page. This procedure assumes that default options are used.
3. Configure the server firewall.
   
   a. Enable the firewall:
      
      ```
      # systemctl enable firewalld
      # systemctl start firewalld
      ```
   
   b. Enable in your firewall the FTP port and port range from previous step:
      
      ```
      # firewall-cmd --add-port min_port-max_port/tcp --permanent
      # firewall-cmd --add-service ftp --permanent
      # firewall-cmd --reload
      ```

      Replace `min_port-max_port` with the port numbers you entered into the `/etc/vsftpd/vsftpd.conf` configuration file.

4. Copy the Kickstart file to the FTP server into the `/var/ftp/` directory or its subdirectory.

5. Make sure that the correct SELinux context and access mode is set on the file:

   ```
   # restorecon -r /var/ftp/your-kickstart-file.ks
   # chmod 444 /var/ftp/your-kickstart-file.ks
   ```

6. Start the `vsftpd` service:

   ```
   # systemctl start vsftpd.service
   ```

   If the service was running before you changed the `/etc/vsftpd/vsftpd.conf` file, restart the service to load the edited file:

   ```
   # systemctl restart vsftpd.service
   ```

   Enable the `vsftpd` service to start during the boot process:

   ```
   # systemctl enable vsftpd
   ```

   The Kickstart file is now accessible and ready to be used for installations by systems on the same network.
NOTE

When configuring the installation source, use ftp:// as the protocol, the server’s host name or IP address, and the path of the Kickstart file, relative to the FTP server root. For example, if the server’s host name is myserver.example.com and you have copied the file to /var/ftp/my-ks.cfg, specify ftp://myserver.example.com/my-ks.cfg as the installation source.

8.3.5. Making a Kickstart file available on a local volume

This procedure describes how to store the Kickstart script file on a volume on the system to be installed. This method enables you to bypass the need for another system.

Prerequisites

- You must have a drive that can be moved to the machine to be installed, such as a USB stick.
- The drive must contain a partition that can be read by the installation program. The supported types are ext2, ext3, ext4, xfs, and fat.
- The drive must be already connected to the system and its volumes mounted.

Procedure

1. List volume information and note the UUID of the volume to which you want to copy the Kickstart file.

   # lsblk -l -p -o name,rm,ro,hotplug,size,type,mountpoint,uuid

2. Navigate to the file system on the volume.

3. Copy the Kickstart file to this file system.

4. Make a note of the string to use later with the inst.ks= option. This string is in the form hd:UUID=volume-UUID:path/to/kickstart-file.cfg. Note that the path is relative to the file system root, not to the / root of file system hierarchy. Replace volume-UUID with the UUID you noted earlier.

5. Unmount all drive volumes:

   # umount /dev/xyz ...

   Add all the volumes to the command, separated by spaces.

8.3.6. Making a Kickstart file available on a local volume for automatic loading

A specially named Kickstart file can be present in the root of a specially named volume on the system to be installed. This lets you bypass the need for another system, and makes the installation program load the file automatically.

Prerequisites

- You must have a drive that can be moved to the machine to be installed, such as a USB stick.
The drive must contain a partition that can be read by the installation program. The supported types are **ext2**, **ext3**, **ext4**, **xfs**, and **fat**.

The drive must be already connected to the system and its volumes mounted.

**Procedure**

1. List volume information to which you want to copy the Kickstart file.
   
   ```bash
   # lsblk -l -p
   ```

2. Navigate to the file system on the volume.

3. Copy the Kickstart file into the root of this file system.

4. Rename the Kickstart file to **ks.cfg**.

5. Rename the volume as **OEMDRV**:
   
   - For **ext2**, **ext3**, and **ext4** file systems:
     
     ```bash
     # e2label /dev/xyz OEMDRV
     ```
   
   - For the XFS file system:
     
     ```bash
     # xfs_admin -L OEMDRV /dev/xyz
     ```
   
   Replace `/dev/xyz` with the path to the volume’s block device.

6. Unmount all drive volumes:
   
   ```bash
   # umount /dev/xyz ...
   ```

   Add all the volumes to the command, separated by spaces.

---

**8.4. CREATING INSTALLATION SOURCES FOR KICKSTART INSTALLATIONS**

This section describes how to create an installation source for the Boot ISO image using the DVD ISO image that contains the required repositories and software packages.

**8.4.1. Types of installation source**

You can use one of the following installation sources for minimal boot images:

- **DVD**: Burn the DVD ISO image to a DVD. The DVD will be automatically used as the installation source (software package source).

- **Hard drive or USB drive**: Copy the DVD ISO image to the drive and configure the installation program to install the software packages from the drive. If you use a USB drive, verify that it is connected to the system before the installation begins. The installation program cannot detect media after the installation begins.

  - **Hard drive limitation**: The DVD ISO image on the hard drive must be on a partition with a file system that the installation program can mount. The supported file systems are **xfs**.
The system that the installation program can mount. The supported file systems are ext4, ext2, ext3, ext4, and vfat (FAT32).

**WARNING**

On Microsoft Windows systems, the default file system used when formatting hard drives is NTFS. The exFAT file system is also available. However, neither of these file systems can be mounted during the installation. If you are creating a hard drive or a USB drive as an installation source on Microsoft Windows, verify that you formatted the drive as FAT32. Note that the FAT32 file system cannot store files larger than 4 GiB.

In Red Hat Enterprise Linux 8, you can enable installation from a directory on a local hard drive. To do so, you need to copy the contents of the DVD ISO image to a directory on a hard drive and then specify the directory as the installation source instead of the ISO image. For example:

`inst.repo=hd:<device>:<path to the directory>`

- **Network location:** Copy the DVD ISO image or the installation tree (extracted contents of the DVD ISO image) to a network location and perform the installation over the network using the following protocols:
  - **NFS:** The DVD ISO image is in a Network File System (NFS) share.
  - **HTTPS, HTTP or FTP:** The installation tree is on a network location that is accessible over HTTP, HTTPS or FTP.

### 8.4.2. Ports for network-based installation

The following table lists the ports that must be open on the server providing the files for each type of network-based installation.

**Table 8.2. Ports for network-based installation**

<table>
<thead>
<tr>
<th>Protocol used</th>
<th>Ports to open</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>80</td>
</tr>
<tr>
<td>HTTPS</td>
<td>443</td>
</tr>
<tr>
<td>FTP</td>
<td>21</td>
</tr>
<tr>
<td>NFS</td>
<td>2049, 111, 20048</td>
</tr>
<tr>
<td>TFTP</td>
<td>69</td>
</tr>
</tbody>
</table>

**Additional resources**

- **Securing networks**
8.4.3. Creating an installation source on an NFS server

Follow the steps in this procedure to place the installation source on an NFS server. Use this installation method to install multiple systems from a single source, without having to connect to physical media.

Prerequisites

- You have administrator level access to a server with Red Hat Enterprise Linux 8, and this server is on the same network as the system to be installed.
- You have downloaded a Binary DVD image. See Downloading the installation ISO image from the Performing a standard RHEL installation document for more information.
- You have created a bootable CD, DVD, or USB device from the image file. See Creating installation media from the Performing a standard RHEL installation document for more information.
- You have verified that your firewall allows the system you are installing to access the remote installation source. See Ports for network-based installation from the Performing a standard RHEL installation document for more information.

Procedure

1. Install the nfs-utils package:

```
# yum install nfs-utils
```

2. Copy the DVD ISO image to a directory on the NFS server.

3. Open the /etc/exports file using a text editor and add a line with the following syntax:

```
/exported_directory/ clients
```

4. Replace /exported_directory/ with the full path to the directory with the ISO image. Replace clients with the host name or IP address of the target system, the subnetwork that all target systems can use to access the ISO image, or the asterisk sign (*) if you want to allow any system with network access to the NFS server to use the ISO image. See the exports(5) man page for detailed information about the format of this field. A basic configuration that makes the /rhel8-install/ directory available as read-only to all clients is:

```
/rhel8-install *
```

5. Save the /etc/exports file and exit the text editor.

6. Start the nfs service:

```
# systemctl start nfs-server.service
```

If the service was running before you changed the /etc/exports file, run the following command for the running NFS server to reload its configuration:

```
# systemctl reload nfs-server.service
```
The ISO image is now accessible over NFS and ready to be used as an installation source.

**NOTE**

When configuring the installation source, use **nfs**: as the protocol, the server host name or IP address, the colon sign (:), and the directory holding the ISO image. For example, if the server host name is **myserver.example.com** and you have saved the ISO image in `/rhel8-install/`, specify **nfs:myserver.example.com:/rhel8-install/** as the installation source.

8.4.4. Creating an installation source using HTTP or HTTPS

Follow the steps in this procedure to create an installation source for a network-based installation using an installation tree, which is a directory containing extracted contents of the DVD ISO image and a valid `.treeinfo` file. The installation source is accessed over HTTP or HTTPS.

**Prerequisites**

- You have administrator level access to a server with Red Hat Enterprise Linux 8, and this server is on the same network as the system to be installed.

- You have downloaded a Binary DVD image. See Downloading the installation ISO image from the Performing a standard RHEL installation document for more information.

- You have created a bootable CD, DVD, or USB device from the image file. See Creating installation media from the Performing a standard RHEL installation document for more information.

- You have verified that your firewall allows the system you are installing to access the remote installation source. See Ports for network-based installation from the Performing a standard RHEL installation document for more information.

**Procedure**

1. To create an installation source using **http**, install the **httpd** package:

   ```
   # yum install httpd
   ```

   To create an installation source using **https**, install **httpd** and **mod_ssl** packages:

   ```
   # yum install httpd mod_ssl
   ```

**WARNING**

If your Apache web server configuration enables SSL security, verify that you enable only the TLSv1 protocol, and disable SSLv2 and SSLv3. This is due to the POODLE SSL vulnerability (CVE-2014-3566). See https://access.redhat.com/solutions/1232413 for details.
IMPORTANT

If you use an HTTPS server with a self-signed certificate, you must boot the
installation program with the `noverifyssl` option.

2. Copy the DVD ISO image to the HTTP(S) server.

3. Mount the DVD ISO image, using the `mount` command, to a suitable directory:

   ```
   # mkdir /mnt/rhel8-install/
   # mount -o loop,ro -t iso9660 /image_directory/image.iso /mnt/rhel8-install/
   ```

   Replace `/image_directory/image.iso` with the path to the DVD ISO image.

4. Copy the files from the mounted image to the HTTP(S) server root. This command creates the
   `/var/www/html/rhel8-install/` directory with the contents of the image.

   ```
   # cp -r /mnt/rhel8-install/ /var/www/html/
   ```

   This command creates the `/var/www/html/rhel8-install/` directory with the content of the
   image. Note that some copying methods can skip the `.treeinfo` file which is required for a valid
   installation source. Running the `cp` command for whole directories as shown in this procedure
   will copy `.treeinfo` correctly.

5. Start the `httpd` service:

   ```
   # systemctl start httpd.service
   ```

   The installation tree is now accessible and ready to be used as the installation source.

NOTE

When configuring the installation source, use `http://` or `https://` as the protocol,
the server host name or IP address, and the directory that contains the files from
the ISO image, relative to the HTTP server root. For example, if you are using
HTTP, the server host name is `myserver.example.com`, and you have copied the
files from the image to `/var/www/html/rhel8-install/`, specify
`http://myserver.example.com/rhel8-install/` as the installation source.

Additional resources

- Deploying different types of servers

8.4.5. Creating an installation source using FTP

Follow the steps in this procedure to create an installation source for a network-based installation using
an installation tree, which is a directory containing extracted contents of the DVD ISO image and a valid
`.treeinfo` file. The installation source is accessed over FTP.

Prerequisites

- You have administrator level access to a server with Red Hat Enterprise Linux8, and this server is
  on the same network as the system to be installed.
- You have downloaded a Binary DVD image. See [Downgrading the installation ISO image](#) from the Performing a standard RHEL installation document for more information.

- You have created a bootable CD, DVD, or USB device from the image file. See Creating installation media from the Performing a standard RHEL installation document for more information.

- You have verified that your firewall allows the system you are installing to access the remote installation source. See [Ports for network-based installation](#) from the Performing a standard RHEL installation document for more information.

### Procedure

1. Install the `vsftpd` package by running the following command as root:

   ```bash
   # yum install vsftpd
   ```

2. Open and edit the `/etc/vsftpd/vsftpd.conf` configuration file in a text editor.

   a. Change the line `anonymous_enable=NO` to `anonymous_enable=YES`.

   b. Change the line `write_enable=YES` to `write_enable=NO`.

   c. Add lines `pasmv_min_port=min_port` and `pasmv_max_port=max_port` Replace `min_port` and `max_port` with the port number range used by FTP server in passive mode, e.g. 10021 and 10031.

   d. Optionally, add custom changes to your configuration. For available options, see the `vsftpd.conf(5)` man page. This procedure assumes that default options are used.

   ![WARNING]

   **WARNING**

   If you configured SSL/TLS security in your `vsftpd.conf` file, ensure that you enable only the TLSv1 protocol, and disable SSLv2 and SSLv3. This is due to the POODLE SSL vulnerability (CVE-2014-3566). See https://access.redhat.com/solutions/1234773 for details.

3. Configure the server firewall.

   a. Enable the firewall:

   ```bash
   # systemctl enable firewalld
   # systemctl start firewalld
   ```

   b. Enable in your firewall the FTP port and port range from previous step:

   ```bash
   # firewall-cmd --add-port min_port-max_port/tcp --permanent
   # firewall-cmd --add-service ftp --permanent
   # firewall-cmd --reload
   ```
4. Copy the DVD ISO image to the FTP server.

5. Mount the DVD ISO image, using the mount command, to a suitable directory:

   ```
   # mkdir /mnt/rhel8-install
   # mount -o loop,ro -t iso9660 /image-directory/image.iso /mnt/rhel8-install
   ```

Replace `/image-directory/image.iso` with the path to the DVD ISO image.

6. Copy the files from the mounted image to the FTP server root:

   ```
   # mkdir /var/ftp/rhel8-install
   # cp -r /mnt/rhel8-install /var/ftp/
   ```

This command creates the `/var/ftp/rhel8-install/` directory with the content of the image. Note that some copying methods can skip the `.treeinfo` file which is required for a valid installation source. Running the `cp` command for whole directories as shown in this procedure will copy `.treeinfo` correctly.

7. Make sure that the correct SELinux context and access mode is set on the copied content:

   ```
   # restorecon -r /var/ftp/rhel8-install
   # find /var/ftp/rhel8-install -type f -exec chmod 444 {} \;
   # find /var/ftp/rhel8-install -type d -exec chmod 755 {} \;
   ```

8. Start the `vsftpd` service:

   ```
   # systemctl start vsftpd.service
   ```

   If the service was running before you changed the `/etc/vsftpd/vsftpd.conf` file, restart the service to load the edited file:

   ```
   # systemctl restart vsftpd.service
   ```

Enable the `vsftpd` service to start during the boot process:

   ```
   # systemctl enable vsftpd
   ```

The installation tree is now accessible and ready to be used as the installation source.

**NOTE**

When configuring the installation source, use `ftp://` as the protocol, the server host name or IP address, and the directory in which you have stored the files from the ISO image, relative to the FTP server root. For example, if the server host name is `myserver.example.com` and you have copied the files from the image to `/var/ftp/rhel8-install/`, specify `ftp://myserver.example.com/rhel8-install/` as the installation source.

### 8.5. STARTING KICKSTART INSTALLATIONS
You can start Kickstart installations in multiple ways:

- Manually by entering the installation program boot menu and specifying the options including Kickstart file there.
- Automatically by editing the boot options in PXE boot.
- Automatically by providing the file on a volume with specific name.

Learn how to perform each of these methods in the following sections.

8.5.1. Starting a Kickstart installation manually

This section explains how to start a Kickstart installation manually, which means some user interaction is required (adding boot options at the `boot:` prompt). Use the boot option `inst.ks=location` when booting the installation system, replacing location with the location of your Kickstart file. The exact way to specify the boot option depends on your system’s architecture.

Prerequisites

- You have a Kickstart file ready in a location accessible from the system to be installed

Procedure

1. Boot the system using a local media (a CD, DVD, or a USB flash drive).

2. At the boot prompt, specify the required boot options.
   a. If the Kickstart file or a required repository is in a network location, you may need to configure the network using the `ip=` option. The installer tries to configure all network devices using the DHCP protocol by default without this option.
   b. Add the `inst.ks=boot` option and the location of the Kickstart file.
   c. In order to access a software source from which necessary packages will be installed, you may need to add the `inst.repo=` option. If you do not specify this option, you must specify the installation source in the Kickstart file.

   For information about editing boot options, see Editing boot options.

3. Start the installation by confirming your added boot options.

   The installation begins now, using the options specified in the Kickstart file. If the Kickstart file is valid and contains all required commands, the installation is completely automated from this point forward.

   **NOTE**

   If you have installed a Red Hat Enterprise Linux Beta release, on systems having UEFI Secure Boot enabled, then add the Beta public key to the system’s Machine Owner Key (MOK) list. For more information about UEFI Secure Boot and Red Hat Enterprise Linux Beta releases, see the Completing post-installation tasks section of the Performing a standard RHEL installation document.

8.5.2. Starting a Kickstart installation automatically using PXE
AMD64, Intel 64, and 64-bit ARM systems and IBM Power Systems servers have the ability to boot using a PXE server. When you configure the PXE server, you can add the boot option into the boot loader configuration file, which in turn lets you start the installation automatically. Using this approach, it is possible to automate the installation completely, including the boot process.

This procedure is intended as a general reference; detailed steps differ based on your system’s architecture, and not all options are available on all architectures (for example, you cannot use PXE boot on 64-bit IBM Z).

**Prerequisites**

- You must have a Kickstart file ready in a location accessible from the system to be installed.
- You must have a PXE server which can be used to boot the system and begin the installation.

**Procedure**

1. Open the boot loader configuration file on your PXE server, and add the `inst.ks=` boot option to the appropriate line. The name of the file and its syntax depends on your system’s architecture and hardware:

   - On AMD64 and Intel 64 systems with BIOS, the file name can be either default or based on your system’s IP address. In this case, add the `inst.ks=` option to the append line in the installation entry. A sample append line in the configuration file looks similar to the following:

     ```
     append initrd=initrd.img inst.ks=http://10.32.5.1/mnt/archive/RHEL-8/8.x/x86_64/kickstarts/ks.cfg
     ```

   - On systems using the GRUB2 boot loader (AMD64, Intel 64, and 64-bit ARM systems with UEFI firmware and IBM Power Systems servers), the file name will be `grub.cfg`. In this file, append the `inst.ks=` option to the kernel line in the installation entry. A sample kernel line in the configuration file will look similar to the following:

     ```
     kernel vmlinuz inst.ks=http://10.32.5.1/mnt/archive/RHEL-8/8.x/x86_64/kickstarts/ks.cfg
     ```

2. Boot the installation from the network server.

   The installation begins now, using the installation options specified in the Kickstart file. If the Kickstart file is valid and contains all required commands, the installation is completely automated.

   **NOTE**

   If you have installed a Red Hat Enterprise Linux Beta release, on systems having UEFI Secure Boot enabled, then add the Beta public key to the system’s Machine Owner Key (MOK) list.

   For more information about UEFI Secure Boot and Red Hat Enterprise Linux Beta releases, see the Completing post-installation tasks section of the Performing a standard RHEL installation document.

8.5.3. Starting a Kickstart installation automatically using a local volume
You can start a Kickstart installation by putting a Kickstart file with a specific name on a specifically labelled storage volume.

**Prerequisites**

- You must have a volume prepared with label **OEMDRV** and the Kickstart file present in its root as **ks.cfg**.
- A drive containing this volume must be available on the system as the installation program boots.

**Procedure**

1. Boot the system using a local media (a CD, DVD, or a USB flash drive).
2. At the boot prompt, specify the required boot options.
   
   a. If a required repository is in a network location, you may need to configure the network using the **ip** option. The installer tries to configure all network devices using the DHCP protocol by default without this option.
   
   b. In order to access a software source from which necessary packages will be installed, you may need to add the **inst.repo** option. If you do not specify this option, you must specify the installation source in the Kickstart file.
3. Start the installation by confirming your added boot options. The installation begins now, and the Kickstart file is automatically detected and used to start an automated Kickstart installation.

**NOTE**

If you have installed a Red Hat Enterprise Linux Beta release, on systems having UEFI Secure Boot enabled, then add the Beta public key to the system’s Machine Owner Key (MOK) list. For more information about UEFI Secure Boot and Red Hat Enterprise Linux Beta releases, see the Completing post-installation tasks section of the Performing a standard RHEL installation document.

### 8.6. CONSOLES AND LOGGING DURING INSTALLATION

The Red Hat Enterprise Linux installer uses the **tmux** terminal multiplexer to display and control several windows in addition to the main interface. Each of these windows serve a different purpose; they display several different logs, which can be used to troubleshoot issues during the installation process. One of the windows provides an interactive shell prompt with root privileges, unless this prompt was specifically disabled using a boot option or a Kickstart command.

**NOTE**

In general, there is no reason to leave the default graphical installation environment unless you need to diagnose an installation problem.

The terminal multiplexer is running in virtual console 1. To switch from the actual installation environment to **tmux**, press **Ctrl+Alt+F1**. To go back to the main installation interface which runs in virtual console 6, press **Ctrl+Alt+F6**.
If you choose text mode installation, you will start in virtual console 1 (**tmux**), and switching to console 6 will open a shell prompt instead of a graphical interface.

The console running **tmux** has five available windows; their contents are described in the following table, along with keyboard shortcuts. Note that the keyboard shortcuts are two-part: first press **Ctrl+b**, then release both keys, and press the number key for the window you want to use.

You can also use **Ctrl+b n**, **Alt+Tab**, and **Ctrl+b p** to switch to the next or previous **tmux** window, respectively.

### Table 8.3. Available tmux windows

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl+b 1</td>
<td>Main installation program window. Contains text-based prompts (during text mode installation or if you use VNC direct mode), and also some debugging information.</td>
</tr>
<tr>
<td>Ctrl+b 2</td>
<td>Interactive shell prompt with <strong>root</strong> privileges.</td>
</tr>
<tr>
<td>Ctrl+b 3</td>
<td>Installation log; displays messages stored in <strong>/tmp/anaconda.log</strong>.</td>
</tr>
<tr>
<td>Ctrl+b 4</td>
<td>Storage log; displays messages related to storage devices and configuration, stored in <strong>/tmp/storage.log</strong>.</td>
</tr>
<tr>
<td>Ctrl+b 5</td>
<td>Program log; displays messages from utilities executed during the installation process, stored in <strong>/tmp/program.log</strong>.</td>
</tr>
</tbody>
</table>

### 8.7. MAINTAINING KICKSTART FILES

You can run automated checks on Kickstart files. Typically, you will want to verify that a new or problematic Kickstart file is valid.

#### 8.7.1. Installing Kickstart maintenance tools

To use the Kickstart maintenance tools, you must install the package that contains them.

**Procedure**

- Install the **pykickstart** package:

  ```
  # yum install pykickstart
  ```

#### 8.7.2. Verifying a Kickstart file
Use the `ksvalidator` command line utility to verify that your Kickstart file is valid. This is useful when you make extensive changes to a Kickstart file. Use the `-v RHEL8` option in the `ksvalidator` command to acknowledge new commands of the RHEL8 class.

**Procedure**

- Run `ksvalidator` on your Kickstart file:

  ```
  $ ksvalidator -v RHEL8 /path/to/kickstart.ks
  ```

  Replace `/path/to/kickstart.ks` with the path to the Kickstart file you want to verify.

**IMPORTANT**

The validation tool cannot guarantee the installation will be successful. It ensures only that the syntax is correct and that the file does not include deprecated options. It does not attempt to validate the `%pre`, `%post` and `%packages` sections of the Kickstart file.

**Additional resources**

- The `ksvalidator(1)` man page

### 8.8. REGISTERING AND INSTALLING RHEL FROM THE CDN USING KICKSTART

This section contains information about how to register your system, attach RHEL subscriptions, and install from the Red Hat Content Delivery Network (CDN) using Kickstart.

#### 8.8.1. Registering and installing RHEL from the CDN

Use this procedure to register your system, attach RHEL subscriptions, and install from the Red Hat Content Delivery Network (CDN) using the `rhsm` Kickstart command, which supports the `syspurpose` command as well as Red Hat Insights. The `rhsm` Kickstart command removes the requirement of using custom `%post` scripts when registering the system.

**IMPORTANT**

The CDN feature is supported by the Boot ISO and DVD ISO image files. However, it is recommended that you use the Boot ISO image file as the installation source defaults to CDN for the Boot ISO image file.

**Prerequisites**

- Your system is connected to a network that can access the CDN.
- You created a Kickstart file and made it available to the installation program on removable media, a hard drive or a network location using an HTTP(S), FTP, or NFS server.
- The Kickstart file is in a location that is accessible by the system that is to be installed.
- You created the boot media used to begin the installation and made the installation source available to the installation program.
IMPORTANT

- The installation source repository used after system registration is dependent on how the system was booted. For more information, see the Installation source repository after system registration section in the Performing a standard RHEL installation document.

- Repository configuration is not required in a Kickstart file as your subscription governs which CDN subset and repositories the system can access.

Procedure

1. Open the Kickstart file.

2. Edit the file to add the rhsm Kickstart command and its options to the file:

   **Organization (required)**
   Enter the organization id. An example is:

   ```bash
   --organization=1234567
   ```

   **NOTE**
   For security reasons, Red Hat username and password account details are not supported by Kickstart when registering and installing from the CDN.

   **Activation Key (required)**
   Enter the Activation Key. You can enter multiple keys as long as the activation keys are registered to your subscription. An example is:

   ```bash
   --activation-key="Test_key_1" --activation-key="Test_key_2"
   ```

   **Red Hat Insights (optional)**
   Connect the target system to Red Hat Insights.

   **NOTE**
   Red Hat Insights is a Software-as-a-Service (SaaS) offering that provides continuous, in-depth analysis of registered Red Hat-based systems to proactively identify threats to security, performance and stability across physical, virtual and cloud environments, and container deployments. Unlike manual installation using the installer GUI, connecting to Red Hat Insights is not enabled by default when using Kickstart.

   An example is:

   ```bash
   --connect-to-insights
   ```

   **HTTP proxy (optional)**
   Set the HTTP proxy. An example is:
--proxy="user:password@hostname:9000"

**NOTE**

Only the hostname is mandatory. If the proxy is required to run on a default port with no authentication, then the option is: `--proxy="hostname"`

**System Purpose (optional)**

Set the System Purpose role, SLA, and usage using the command:

```
syspurpose --role="Red Hat Enterprise Linux Server" --sla="Premium" --usage="Production"
```

**Example**

The following example displays a minimal Kickstart file with all `rhsm` Kickstart command options.

```
graphical
lang en_US.UTF-8
keyboard us
rootpw 12345
timezone America/New_York
zerombr
clearpart --all --initlabel
autopart
syspurpose --role="Red Hat Enterprise Linux Server" --sla="Premium" --usage="Production"
rhsm --organization="12345" --activation-key="test_key" --connect-to-insights --proxy="user:password@hostname:9000"
reboot
%packages
vim
%end
```

3. Save the Kickstart file and start the installation process.

**Additional resources**

- Configuring System Purpose
- Starting Kickstart installations
- *Red Hat Insights product documentation*
- Understanding Activation Keys
- Using an HTTP proxy

**8.8.2. Verifying your system registration from the CDN**

Use this procedure to verify that your system is registered to the CDN.
Prerequisites

- You have completed the registration and installation process as documented in Register and install using CDN.
- You have started the Kickstart installation as documented in Starting Kickstart installations.
- The installed system has rebooted and a terminal window is open.

Procedure

1. From the terminal window, log in as a root user and verify the registration:

   ```
   # subscription-manager list
   ```

   The output displays the attached subscription details, for example:

   ```
   Installed Product Status
   Product Name: Red Hat Enterprise Linux for x86_64
   Product ID: 486
   Version: X
   Arch: x86_64
   Status: Subscribed
   Status Details
   Starts: 11/4/2019
   Ends: 11/4/2020
   ```

2. To view a detailed report, run the command:

   ```
   # subscription-manager list --consumed
   ```

8.8.3. Unregistering your system from the CDN

Use this procedure to unregister your system from the Red Hat CDN.

Prerequisites

- You have completed the registration and installation process as documented in Registering and installing RHEL from the CDN.
- You have started the Kickstart installation as documented in Starting Kickstart installations.
- The installed system has rebooted and a terminal window is open.

Procedure

1. From the terminal window, log in as a root user and unregister:

   ```
   # subscription-manager unregister
   ```

   The attached subscription is unregistered from the system and the connection to CDN is removed.
8.9. PERFORMING A REMOTE RHEL INSTALLATION USING VNC

This section describes how to perform a remote RHEL installation using Virtual Network Computing (VNC).

8.9.1. Overview

The graphical user interface is the recommended method of installing RHEL when you boot the system from a CD, DVD, or USB flash drive, or from a network using PXE. However, many enterprise systems, for example, IBM Power Systems and 64-bit IBM Z, are located in remote data center environments that are run autonomously and are not connected to a display, keyboard, and mouse. These systems are often referred to as headless systems and they are typically controlled over a network connection. The RHEL installation program includes a Virtual Network Computing (VNC) installation that runs the graphical installation on the target machine, but control of the graphical installation is handled by another system on the network. The RHEL installation program offers two VNC installation modes: Direct and Connect. Once a connection is established, the two modes do not differ. The mode you select depends on your environment.

Direct mode

In Direct mode, the RHEL installation program is configured to start on the target system and wait for a VNC viewer that is installed on another system before proceeding. As part of the Direct mode installation, the IP address and port are displayed on the target system. You can use the VNC viewer to connect to the target system remotely using the IP address and port, and complete the graphical installation.

Connect mode

In Connect mode, the VNC viewer is started on a remote system in listening mode. The VNC viewer waits for an incoming connection from the target system on a specified port. When the RHEL installation program starts on the target system, the system host name and port number are provided by using a boot option or a Kickstart command. The installation program then establishes a connection with the listening VNC viewer using the specified system host name and port number. To use Connect mode, the system with the listening VNC viewer must be able to accept incoming network connections.

8.9.2. Considerations

Consider the following items when performing a remote RHEL installation using VNC:

- **VNC client application:** A VNC client application is required to perform both a VNC Direct and Connect installation. VNC client applications are available in the repositories of most Linux distributions, and free VNC client applications are also available for other operating systems such as Windows. The following VNC client applications are available in RHEL:
  - **tigervnc** is independent of your desktop environment and is installed as part of the tigervnc package.
  - **vinagre** is part of the GNOME desktop environment and is installed as part of the vinagre package.

  **NOTE**

  A VNC server is included in the installation program and doesn’t need to be installed.

- **Network and firewall:**
• If the target system is not allowed inbound connections by a firewall, then you must use Connect mode or disable the firewall. Disabling a firewall can have security implications.

• If the system that is running the VNC viewer is not allowed incoming connections by a firewall, then you must use Direct mode, or disable the firewall. Disabling a firewall can have security implications. See the Security hardening document for more information on configuring the firewall.

• Custom Boot Options: You must specify custom boot options to start a VNC installation and the installation instructions might differ depending on your system architecture.

• VNC in Kickstart installations: You can use VNC-specific commands in Kickstart installations. Using only the `vnc` command runs a RHEL installation in Direct mode. Additional options are available to set up an installation using Connect mode.

8.9.3. Performing a remote RHEL installation in VNC Direct mode

Use this procedure to perform a remote RHEL installation in VNC Direct mode. Direct mode expects the VNC viewer to initiate a connection to the target system that is being installed with RHEL. In this procedure, the system with the VNC viewer is called the remote system. You are prompted by the RHEL installation program to initiate the connection from the VNC viewer on the remote system to the target system.

NOTE

This procedure uses TigerVNC as the VNC viewer. Specific instructions for other viewers might differ, but the general principles apply.

Prerequisites

• As root, you have installed a VNC viewer on a remote system, for example:

```bash
# yum install tigervnc
```

• You have set up a network boot server and booted the installation on the target system.

Procedure

1. From the RHEL boot menu on the target system, press the Tab key on your keyboard to edit the boot options.

2. Append the `inst.vnc` option to the end of the command line.

   a. If you want to restrict VNC access to the system that is being installed, add the `inst.vncpassword=PASSWORD` boot option to the end of the command line. Replace `PASSWORD` with the password you want to use for the installation. The VNC password must be between 6 and 8 characters long.

   IMPORTANT

   Use a temporary password for the `inst.vncpassword=` option. It should not be an existing or root password.

   b. If you want to disable VNC access completely, add the `inst.vnc=0` boot option to the end of the command line.
3. Press **Enter** to start the installation. The target system initializes the installation program and starts the necessary services. When the system is ready, a message is displayed providing the IP address and port number of the system.

4. Open the VNC viewer on the remote system.

5. Enter the IP address and the port number into the **VNC server** field.

6. Click **Connect**.

7. Enter the VNC password and click **OK**. A new window opens with the VNC connection established, displaying the RHEL installation menu. From this window, you can install RHEL on the target system using the graphical user interface.

### 8.9.4. Performing a remote RHEL installation in VNC Connect mode

Use this procedure to perform a remote RHEL installation in VNC Connect mode. In Connect mode, the target system that is being installed with RHEL initiates a connect to the VNC viewer that is installed on another system. In this procedure, the system with the VNC viewer is called the **remote** system.

**NOTE**

This procedure uses **TigerVNC** as the VNC viewer. Specific instructions for other viewers might differ, but the general principles apply.

#### Prerequisites

- As root, you have installed a VNC viewer on a remote system, for example:

  ```bash
  # yum install tigervnc
  ```

- You have set up a network boot server to start the installation on the target system.

- You have configured the target system to use the boot options for a VNC Connect installation.

- You have verified that the remote system with the VNC viewer is configured to accept an incoming connection on the required port. Verification is dependent on your network and system configuration. For more information, see the Security hardening and Securing networks documents.

#### Procedure

1. Start the VNC viewer on the remote system in **listening mode** by running the following command:

   ```bash
   $ vncviewer -listen PORT
   ```

2. Replace PORT with the port number used for the connection.

3. The terminal displays a message indicating that it is waiting for an incoming connection from the target system.

   TigerVNC Viewer 64-bit v1.8.0
   Built on: 2017-10-12 09:20
   Copyright (C) 1999-2017 TigerVNC Team and many others (see README.txt)
4. Boot the target system from the network.

5. From the RHEL boot menu on the target system, press the Tab key on your keyboard to edit the boot options.

6. Append the `inst.vnc inst.vncconnect=HOST:PORT` option to the end of the command line.

7. Replace `HOST` with the IP address of the remote system that is running the listening VNC viewer, and `PORT` with the port number that the VNC viewer is listening on.

8. Press Enter to start the installation. The system initializes the installation program and starts the necessary services. When the initialization process is finished, the installation program attempts to connect to the IP address and port provided.

9. When the connection is successful, a new window opens with the VNC connection established, displaying the RHEL installation menu. From this window, you can install RHEL on the target system using the graphical user interface.
CHAPTER 9. ADVANCED CONFIGURATION OPTIONS

9.1. CONFIGURING SYSTEM PURPOSE

You use System Purpose to record the intended use of a Red Hat Enterprise Linux 8 system. Setting System Purpose enables the entitlement server to auto-attach the most appropriate subscription. This section describes how to configure System Purpose using Kickstart.

Benefits include:

- In-depth system-level information for system administrators and business operations.
- Reduced overhead when determining why a system was procured and its intended purpose.
- Improved customer experience of Subscription Manager auto-attach as well as automated discovery and reconciliation of system usage.

9.1.1. Overview

You can enter System Purpose data in one of the following ways:

- During image creation
- During a GUI installation when using the Connect to Red Hat screen to register your system and attach your Red Hat subscription
- During a Kickstart installation when using the syspurpose Kickstart command
- After installation using the syspurpose command-line (CLI) tool

To record the intended purpose of your system, you can configure the following components of System Purpose. The selected values are used by the entitlement server upon registration to attach the most suitable subscription for your system.

Role

- Red Hat Enterprise Linux Server
- Red Hat Enterprise Linux Workstation
- Red Hat Enterprise Linux Compute Node

Service Level Agreement

- Premium
- Standard
- Self-Support

Usage

- Production
- Development/Test
9.1.2. Configuring System Purpose in a Kickstart file

Follow the steps in this procedure to configure System Purpose during the installation. To do so, use the `syspurpose` Kickstart command in the Kickstart configuration file.

Even though System Purpose is an optional feature of the Red Hat Enterprise Linux installation program, we strongly recommend that you configure System Purpose to auto-attach the most appropriate subscription.

NOTE
You can also enable System Purpose after the installation is complete. To do so use the `syspurpose` command-line tool. The `syspurpose` tool commands are different from the `syspurpose` Kickstart commands.

The following actions are available for the `syspurpose` Kickstart command:

**role**
Set the intended role of the system. This action uses the following format:

```
syspurpose --role=
```

The assigned role can be:

- Red Hat Enterprise Linux Server
- Red Hat Enterprise Linux Workstation
- Red Hat Enterprise Linux Compute Node

**SLA**
Set the intended SLA of the system. This action uses the following format:

```
syspurpose --sla=
```

The assigned sla can be:

- Premium
- Standard
- Self-Support
usage
Set the intended usage of the system. This action uses the following format:

```
syspurpose --usage=
```

The assigned usage can be:

- Production
- Development/Test
- Disaster Recovery

addon
Any additional layered products or features. To add multiple items specify `--addon` multiple times, once per layered product/feature. This action uses the following format:

```
syspurpose --addon=
```

9.1.3. Additional resources
- Configuring System Purpose using the syspurpose command-line tool

9.2. UPDATING DRIVERS DURING INSTALLATION

This section describes how to complete a driver update during the Red Hat Enterprise Linux installation process.

**NOTE**
This is an optional step of the installation process. Red Hat recommends that you do not perform a driver update unless it is necessary.

9.2.1. Prerequisite

You have been notified by Red Hat, your hardware vendor, or a trusted third-party vendor that a driver update is required during Red Hat Enterprise Linux installation.

9.2.2. Overview

Red Hat Enterprise Linux supports drivers for many hardware devices but some newly-released drivers may not be supported. A driver update should only be performed if an unsupported driver prevents the installation from completing. Updating drivers during installation is typically only required to support a particular configuration. For example, installing drivers for a storage adapter card that provides access to your system’s storage devices.
9.2.3. Types of driver update

Red Hat, your hardware vendor, or a trusted third party provides the driver update as an ISO image file. Once you receive the ISO image file, choose the type of driver update.

Types of driver update

Automatic

The recommended driver update method; a storage device (including a CD, DVD, or USB flash drive) labeled OEMDRV is physically connected to the system. If the OEMDRV storage device is present when the installation starts, it is treated as a driver update disk, and the installation program automatically loads its drivers.

Assisted

The installation program prompts you to locate a driver update. You can use any local storage device with a label other than OEMDRV. The inst.dd boot option is specified when starting the installation. If you use this option without any parameters, the installation program displays all of the storage devices connected to the system, and prompts you to select a device that contains a driver update.

Manual

Manually specify a path to a driver update image or an RPM package. You can use any local storage device with a label other than OEMDRV, or a network location accessible from the installation system. The inst.dd=location boot option is specified when starting the installation, where location is the path to a driver update disk or ISO image. When you specify this option, the installation program attempts to load any driver updates found at the specified location. With manual driver updates, you can specify local storage devices, or a network location (HTTP, HTTPS or FTP server).

NOTE

- You can use both inst.dd=location and inst.dd simultaneously, where location is the path to a driver update disk or ISO image. In this scenario, the installation program attempts to load any available driver updates from the location and also prompts you to select a device that contains the driver update.

- Initialize the network using the ip= option when loading a driver update from a network location.

Limitations

On UEFI systems with the Secure Boot technology enabled, all drivers must be signed with a valid certificate. Red Hat drivers are signed by one of Red Hat’s private keys and authenticated by its corresponding public key in the kernel. If you load additional, separate drivers, verify that they are signed.

9.2.4. Preparing a driver update
This procedure describes how to prepare a driver update on a CD and DVD.

**Prerequisites**

- You received the driver update ISO image from Red Hat, your hardware vendor, or a trusted third-party vendor.
- You burned the driver update ISO image to a CD or DVD.

**WARNING**

If only a single ISO image file ending in `.iso` is available on the CD or DVD, the burn process has not been successful. See your system's burning software documentation for instructions on how to burn ISO images to a CD or DVD.

**Procedure**

1. Insert the driver update CD or DVD into your system's CD/DVD drive, and browse it using the system's file manager tool.

2. Verify that a single file `rhdd3` is available. `rhdd3` is a signature file that contains the driver description and a directory named `rpms`, which contains the RPM packages with the actual drivers for various architectures.

9.2.5. Performing an automatic driver update

This procedure describes how to perform an automatic driver update during installation.

**Prerequisites**

- You have placed the driver update image on a standard disk partition with an `OEMDRV` label or burnt the `OEMDRV` driver update image to a CD or DVD. Advanced storage, such as RAID or LVM volumes, may not be accessible during the driver update process.
- You have connected a block device with an `OEMDRV` volume label to your system, or inserted the prepared CD or DVD into your system's CD/DVD drive before starting the installation process.

**Procedure**

1. Once you have completed the prerequisite steps, the drivers are automatically loaded when the installation program starts, and installed on the system during the installation process.

9.2.6. Performing an assisted driver update

This procedure describes how to perform an assisted driver update during installation.

**Prerequisites**
You have connected a block device without an **OEMDRV** volume label to your system and copied the driver disk image to this device, or you have prepared a driver update CD or DVD and inserted it into your system’s CD/DVD drive before starting the installation process.

**NOTE**

If you burned an ISO image file to a CD or DVD but it does not have the **OEMDRV** volume label, you can use the *inst.dd* option with no arguments. The installation program provides an option to scan and select drivers from the CD or DVD. In this scenario, the installation program does not prompt you to select a driver update ISO image. Another scenario is to use the CD or DVD with the *inst.dd=location* boot option; this allows the installation program to automatically scan the CD or DVD for driver updates. For more information, see Performing a manual driver update.

**Procedure**

1. From the boot menu window, press the Tab key on your keyboard to display the boot command line.

2. Append the *inst.dd* boot option to the command line and press Enter to execute the boot process.

3. From the menu, select a local disk partition or a CD or DVD device. The installation program scans for ISO files, or driver update RPM packages.


   **NOTE**

   This step is not required if the selected device or partition contains driver update RPM packages rather than an ISO image file, for example, an optical drive containing a driver update CD or DVD.

5. Select the required drivers.

   a. Use the number keys on your keyboard to toggle the driver selection.

   b. Press c to install the selected driver. The selected driver is loaded and the installation process starts.

9.2.7. Performing a manual driver update

This procedure describes how to perform a manual driver update during installation.

**Prerequisites**

- Place the driver update ISO image file on a USB flash drive or a web server, and connect it to your computer.

**Procedure**

1. From the boot menu window, press the Tab key on your keyboard to display the boot command line.

2. Append the *inst.dd=location* boot option to the command line, where location is a path to the
driver update. Typically, the image file is located on a web server, for example, http://server.example.com/dd.iso, or on a USB flash drive, for example, /dev/sdb1. It is also possible to specify an RPM package containing the driver update, for example http://server.example.com/dd.rpm.

3. Press Enter to execute the boot process. The drivers available at the specified location are automatically loaded and the installation process starts.

Additional resources

- The inst.dd boot option

9.2.8. Disabling a driver

This procedure describes how to disable a malfunctioning driver.

Prerequisites

- You have booted the installation program boot menu.

Procedure

1. From the boot menu, press the Tab key on your keyboard to display the boot command line.

2. Append the modprobe.blacklist=driver_name boot option to the command line.

3. Replace driver_name with the name of the driver or drivers you want to disable, for example:

   ```
   modprobe.blacklist=ahci
   ```

   Drivers disabled using the modprobe.blacklist= boot option remain disabled on the installed system and appear in the /etc/modprobe.d/anaconda-blacklist.conf file.

4. Press Enter to execute the boot process.

9.3. PREPARING TO INSTALL FROM THE NETWORK USING PXE

This section describes how to configure TFTP and DHCP on a PXE server to enable PXE boot and network installation.

9.3.1. Network install overview

A network installation allows you to install Red Hat Enterprise Linux to a system that has access to an installation server. At a minimum, two systems are required for a network installation:

**PXE Server**

A system running a DHCP server, a TFTP server, and an HTTP, HTTPS, FTP, or NFS server. While each server can run on a different physical system, the procedures in this section assume a single system is running all servers.

**Client**

The system to which you are installing Red Hat Enterprise Linux. Once installation starts, the client queries the DHCP server, receives the boot files from the TFTP server, and downloads the installation image from the HTTP, HTTPS, FTP or NFS server. Unlike other installation methods, the
client does not require any physical boot media for the installation to start.

NOTE
To boot a client from the network, configure it in BIOS/UEFI or a quick boot menu. On some hardware, the option to boot from a network might be disabled, or not available.

The workflow steps to prepare to install Red Hat Enterprise Linux from a network using PXE are as follows:

Steps

1. Export the installation ISO image or the installation tree to an NFS, HTTPS, HTTP, or FTP server.

2. Configure the TFTP server and DHCP server, and start the TFTP service on the PXE server.

3. Boot the client and start the installation.

IMPORTANT
The GRUB2 boot loader supports a network boot from HTTP in addition to a TFTP server. Sending the boot files, which are the kernel and initial RAM disk `vmlinuz` and `initrd`, over this protocol might be slow and result in timeout failures. An HTTP server does not carry this risk, but it is recommended that you use a TFTP server when sending the boot files.

Additional resources

- Creating installation sources for Kickstart installations
- Configuring a TFTP server for BIOS-based clients
- Configuring a TFTP server for UEFI-based clients
- Configuring a network server for IBM Power systems
- Red Hat Satellite product documentation

9.3.2. Configuring a TFTP server for BIOS-based clients

Use this procedure to configure a TFTP server and DHCP server and start the TFTP service on the PXE server for BIOS-based AMD and Intel 64-bit systems.

IMPORTANT
All configuration files in this section are examples. Configuration details vary and are dependent on the architecture and specific requirements.

Procedure

1. As root, install the following packages. If you already have a DHCP server configured in your network, exclude the `dhcp-server` packages:
# yum install tftp-server dhcp-server

2. Allow incoming connections to the **tftp service** in the firewall:

```bash
# firewall-cmd --add-service=tftp
```

**NOTE**

- This command enables temporary access until the next server reboot. To enable permanent access, add the **--permanent** option to the command.

- Depending on the location of the installation ISO file, you might have to allow incoming connections for HTTP or other services.

3. Configure your DHCP server to use the boot images packaged with **SYSLINUX** as shown in the following example `/etc/dhcp/dhcpd.conf` file. Note that if you already have a DHCP server configured, then perform this step on the DHCP server.

```plaintext
option space pxelinus;
option pxelinus.magic code 208 = string;
option pxelinus.configfile code 209 = text;
option pxelinus.pathprefix code 210 = text;
option pxelinus.reboottime code 211 = unsigned integer 32;
option architecture-type code 93 = unsigned integer 16;

subnet 10.0.0.0 netmask 255.255.255.0 {
    option routers 10.0.0.254;
    range 10.0.0.2 10.0.0.253;

    class "pxeclients" {
        match if substring (option vendor-class-identifier, 0, 9) = "PXEClient";
        next-server 10.0.0.1;

        if option architecture-type = 00:07 {
            filename "BOOTX64.efi";
        } else {
            filename "pxelinus/pxelinus.0";
        }
    }
}
```

4. Access the **pxelinus.0** file from the **SYSLINUX** package in the DVD ISO image file, where `my_local_directory` is the name of the directory that you create:

```bash
# mount -t iso9660 /path_to_image/name_of_image.iso /mount_point -o loop,ro

# cp -pr /mount_point/BaseOS/Packages/symlinus-tftpboot-version-architecture.rpm
/my_local_directory

# umount /mount_point
```

5. Extract the package:
# rpm2cpio syslinux-tftpboot-version-architecture.rpm | cpio -dimv

6. Create a `pxelinux/` directory in `tftpboot/` and copy all the files from the directory into the `pxelinux/` directory:

```bash
# mkdir /var/lib/tftpboot/pxelinux

# cp my_local_directory/tftpboot/* /var/lib/tftpboot/pxelinux
```

7. Create the directory `pxelinux.cfg/` in the `pxelinux/` directory:

```bash
# mkdir /var/lib/tftpboot/pxelinux/pxelinux.cfg
```

8. Create a configuration file named `default` and add it to the `pxelinux.cfg/` directory as shown in the following example:

```bash
default vesamenu.c32
prompt 1
timeout 600

display boot.msg

label linux
 menu label ^Install system
 menu default
 kernel images/RHEL-8.1/vmlinuz
 append initrd=images/RHEL-8.1/initrd.img ip=dhcp inst.repo=http://10.32.5.1/RHEL-8.1/x86_64/iso-contents-root/
 label vesa
 menu label Install system with ^basic video driver
 kernel images/RHEL-8.1/vmlinuz
 append initrd=images/RHEL-8.1/initrd.img ip=dhcp inst.xdriver=vesa nomodeset
 inst.repo=http://10.32.5.1/RHEL-8.1/x86_64/iso-contents-root/
 label rescue
 menu label ^Rescue installed system
 kernel images/RHEL-8.1/vmlinuz
 append initrd=images/RHEL-8.1/initrd.img rescue
 label local
 menu label Boot from ^local drive
 localboot 0xffff
```

**NOTE**

- The installation program cannot boot without its runtime image. Use the `inst.stage2` boot option to specify location of the image. Alternatively, you can use the `inst.repo=` option to specify the image as well as the installation source.

- The installation source location used with `inst.repo` must contain a valid `.treeinfo` file.

- When you select the RHEL8 installation DVD as the installation source, the `.treeinfo` file points to the BaseOS and the AppStream repositories. You can use a single `inst.repo` option to load both repositories.
9. Create a subdirectory to store the boot image files in the `/var/lib/tftpboot/` directory, and copy the boot image files to the directory. In this example, the directory is `/var/lib/tftpboot/pxelinux/images/RHEL-8.1/`:

```bash
# mkdir -p /var/lib/tftpboot/pxelinux/images/RHEL-8.1/
# cp /path_to_x86_64_images/pxeboot/{vmlinuz,initrd.img} /var/lib/tftpboot/pxelinux/images/RHEL-8.1/
```

10. On the DHCP server, start and enable the `dhcpd` service. If you have configured a DHCP server on the localhost, then start and enable the `dhcpd` service on the localhost.

```bash
# systemctl start dhcpd
# systemctl enable dhcpd
```

11. Start and enable the `tftp.socket` service:

```bash
# systemctl start tftp.socket
# systemctl enable tftp.socket
```

The PXE boot server is now ready to serve PXE clients. You can start the client, which is the system to which you are installing Red Hat Enterprise Linux, select PXE Boot when prompted to specify a boot source, and start the network installation.

### 9.3.3. Configuring a TFTP server for UEFI-based clients

Use this procedure to configure a TFTP server and DHCP server and start the TFTP service on the PXE server for UEFI-based AMD64, Intel 64, and 64-bit ARM systems.

**IMPORTANT**

- All configuration files in this section are examples. Configuration details vary and are dependent on the architecture and specific requirements.
- Red Hat Enterprise Linux 8 UEFI PXE boot supports a lowercase file format for a MAC-based grub menu file. For example, the MAC address file format for grub2 is `grub.cfg-01-aa-bb-cc-dd-ee-ff`

**Procedure**

1. As root, install the following packages. If you already have a DHCP server configured in your network, exclude the `dhcp-server` packages.

   ```bash
   # yum install tftp-server dhcp-server
   ```

2. Allow incoming connections to the `tftp service` in the firewall:

   ```bash
   # firewall-cmd --add-service=tftp
   ```
NOTE

- This command enables temporary access until the next server reboot. To enable permanent access, add the `--permanent` option to the command.

- Depending on the location of the installation ISO file, you might have to allow incoming connections for HTTP or other services.

3. Configure your DHCP server to use the boot images packaged with `shim` as shown in the following example `/etc/dhcp/dhcpd.conf` file. Note that if you already have a DHCP server configured, then perform this step on the DHCP server.

```bash
option space pxelinux;
option pxelinux.magic code 208 = string;
option pxelinux.configfile code 209 = text;
option pxelinux.pathprefix code 210 = text;
option pxelinux.reboottime code 211 = unsigned integer 32;
option architecture-type code 93 = unsigned integer 16;

subnet 10.0.0.0 netmask 255.255.255.0 {
    option routers 10.0.0.254;
    range 10.0.0.2 10.0.0.253;

    class "pxeclients" {
        match if substring (option vendor-class-identifier, 0, 9) = "PXEClient";
        next-server 10.0.0.1;

        if option architecture-type = 00:07 {
            filename "BOOTX64.efi";
        } else {
            filename "pxelinux/pxelinux.0";
        }
    }
}
```

4. Access the `BOOTX64.efi` file from the `shim` package, and the `grubx64.efi` file from the `grub2-efi` package in the DVD ISO image file where `my_local_directory` is the name of the directory that you create:

```bash
# mount -t iso9660 /path_to_image/name_of_image.iso /mount_point -o loop,ro

# cp -pr /mount_point/BaseOS/Packages/shim-version-architecture.rpm /my_local_directory

# cp -pr /mount_point/BaseOS/Packages/grub2-efi-version-architecture.rpm /my_local_directory

# umount /mount_point
```

5. Extract the packages:

```bash
# rpm2cpio shim-version-architecture.rpm | cpio -dimv

# rpm2cpio grub2-efi-version-architecture.rpm | cpio -dimv
```
6. Copy the EFI boot images from your boot directory. Replace ARCH with shim or grub followed by the architecture, for example, grubx64.

```bash
# mkdir /var/lib/tftpboot/uefi
# cp my_local_directory/boot/efi/EFI/redhat/ARCH.efi /var/lib/tftpboot/uefi/
```

7. Add a configuration file named `grub.cfg` to the `tftpboot` directory as shown in the following example:

```bash
set timeout=60
menuentry 'RHEL 8' {
    linuxefi images/RHEL-8.1/vmlinuz
    ip=dhcp
    inst.repo=http://10.32.5.1/RHEL-8.1/x86_64/iso-contents-root/
    initrddefi images/RHEL-8.1/initrd.img
}
```

**NOTE**

- The installation program cannot boot without its runtime image. Use the `inst.stage2` boot option to specify location of the image. Alternatively, you can use the `inst.repo=` option to specify the image as well as the installation source.

- The installation source location used with `inst.repo` must contain a valid `.treeinfo` file.

- When you select the RHEL8 installation DVD as the installation source, the `.treeinfo` file points to the BaseOS and the AppStream repositories. You can use a single `inst.repo` option to load both repositories.

8. Create a subdirectory to store the boot image files in the `/var/lib/tftpboot/` directory, and copy the boot image files to the directory. In this example, the directory is `/var/lib/tftpboot/images/RHEL-8.1/`:

```bash
# mkdir -p /var/lib/tftpboot/images/RHEL-8.1/
# cp /path_to_x86_64_images/pxeboot/{vmlinuz,initrd.img} /var/lib/tftpboot/images/RHEL-8.1/
```

9. On the DHCP server, start and enable the `dhcpd` service. If you have configured a DHCP server on the localhost, then start and enable the `dhcpp` service on the localhost.

```bash
# systemctl start dhcpd
# systemctl enable dhcpd
```

10. Start and enable the `tftp.socket` service:

```bash
# systemctl start tftp.socket
# systemctl enable tftp.socket
```

The PXE boot server is now ready to serve PXE clients. You can start the client, which is the system to which you are installing Red Hat Enterprise Linux, select PXE Boot when prompted to specify a boot source, and start the network installation.

**Additional resources**
9.3.4. Configuring a network server for IBM Power systems

Use this procedure to configure a network boot server for IBM Power systems using GRUB2.

**IMPORTANT**
All configuration files in this section are examples. Configuration details vary and are dependent on the architecture and specific requirements.

**Procedure**

1. As root, install the following packages. If you already have a DHCP server configured in your network, exclude the dhcp-server packages.

   ```bash
   # yum install tftp-server dhcp-server
   ```

2. Allow incoming connections to the *tftp service* in the firewall:

   ```bash
   # firewall-cmd --add-service=tftp
   ```

   **NOTE**
   - This command enables temporary access until the next server reboot. To enable permanent access, add the *--permanent* option to the command.
   - Depending on the location of the installation ISO file, you might have to allow incoming connections for HTTP or other services.

3. Create a GRUB2 network boot directory inside the tftp root:

   ```bash
   # grub2-mknetdir --net-directory=/var/lib/tftpboot
   ```

   Netboot directory for powerpc-ieee1275 created. Configure your DHCP server to point to /boot/grub2/powerpc-ieee1275/core.elf

   **NOTE**
   The command output informs you of the file name that needs to be configured in your DHCP configuration, described in this procedure.

   a. If the PXE server runs on an x86 machine, the *grub2-ppc64-modules* must be installed before creating a GRUB2 network boot directory inside the tftp root:

   ```bash
   # yum install grub2-ppc64-modules
   ```

4. Create a GRUB2 configuration file: `/var/lib/tftpboot/boot/grub2/grub.cfg` as shown in the following example:

   ```
   set default=0
   set timeout=5
   ```
Welcome to the Red Hat Enterprise Linux 8 installer!

```bash
menuentry 'Red Hat Enterprise Linux 8' {
    linux grub2-ppc64/vmlinuz ro ip=dhcp inst.repo=http://10.32.5.1/RHEL-8.1/x86_64/iso-contents-root/
    initrd grub2-ppc64/initrd.img
}
```

NOTE

- The installation program cannot boot without its runtime image. Use the `inst.stage2` boot option to specify location of the image. Alternatively, you can use the `inst.repo=` option to specify the image as well as the installation source.

- The installation source location used with `inst.repo` must contain a valid `.treeinfo` file.

- When you select the RHEL8 installation DVD as the installation source, the `.treeinfo` file points to the BaseOS and the AppStream repositories. You can use a single `inst.repo` option to load both repositories.

5. Mount the DVD ISO image using the command:

```bash
# mount -t iso9660 /path_to_image/name_of_iso/ /mount_point -o loop,ro
```

6. Create a directory and copy the `initrd.img` and `vmlinuz` files from DVD ISO image into it, for example:

```bash
# cp /mount_point/ppc/ppc64/{initrd.img,vmlinuz} /var/lib/tftpboot/grub2-ppc64/
```

7. Configure your DHCP server to use the boot images packaged with GRUB2 as shown in the following example. Note that if you already have a DHCP server configured, then perform this step on the DHCP server.

```bash
subnet 192.168.0.1 netmask 255.255.255.0 {
    allow bootp;
    option routers 192.168.0.5;
    group { #BOOTP POWER clients
        filename "boot/grub2/powerpc-ieee1275/core.elf";
        host client1 {
            hardware ethernet 01:23:45:67:89:ab;
            fixed-address 192.168.0.112;
        }
    }
}
```

8. Adjust the sample parameters `subnet`, `netmask`, `routers`, `fixed-address` and `hardware ethernet` to fit your network configuration. Note the `file name` parameter; this is the file name that was outputted by the `grub2-mknetdir` command earlier in this procedure.

9. On the DHCP server, start and enable the `dhcpcd` service. If you have configured a DHCP server on the localhost, then start and enable the `dhcpcd` service on the localhost.
# systemctl start dhcpd
# systemctl enable dhcpd

10. Start and enable the `tftp.socket` service:

# systemctl start tftp.socket
# systemctl enable tftp.socket

The PXE boot server is now ready to serve PXE clients. You can start the client, which is the system to which you are installing Red Hat Enterprise Linux, select **PXE Boot** when prompted to specify a boot source, and start the network installation.

## 9.4. BOOT OPTIONS

This section contains information about some of the boot options that you can use to modify the default behavior of the installation program. For a full list of boot options, see the [upstream boot option content](#).

### 9.4.1. Types of boot options

There are two types of boot options; those with an equals "=" sign, and those without an equals "=" sign. Boot options are appended to the boot command line and multiple options must be separated by a single space. Boot options that are specific to the installation program always start with **inst**.

**Options with an equals "=" sign**

You must specify a value for boot options that use the `=` symbol. For example, the `inst.vncpassword=` option must contain a value, in this case, a password. The correct syntax for this example is `inst.vncpassword=password`.

**Options without an equals "=" sign**

This boot option does not accept any values or parameters. For example, the `rd.live.check` option forces the installation program to verify the installation media before starting the installation. If this boot option is present, the verification is performed; if the boot option is not present, the verification is skipped.

### 9.4.2. Editing boot options

This section contains information about the different ways that you can edit boot options from the boot menu. The boot menu opens after you boot the installation media.

#### 9.4.2.1. Editing the boot: prompt in BIOS

When using the `boot:` prompt, the first option must always specify the installation program image file that you want to load. In most cases, you can specify the image using the keyword. You can specify additional options according to your requirements.

**Prerequisites**

- You have created bootable installation media (USB, CD or DVD).
- You have booted the installation from the media, and the installation boot menu is open.

**Procedure**

# systemctl start dhcpd
# systemctl enable dhcpd
1. With the boot menu open, press the Esc key on your keyboard.

2. The boot: prompt is now accessible.

3. Press the Tab key on your keyboard to display the help commands.

4. Press the Enter key on your keyboard to start the installation with your options. To return from the boot: prompt to the boot menu, restart the system and boot from the installation media again.

NOTE

The boot: prompt also accepts dracut kernel options. A list of options is available in the dracut.cmdline(7) man page.

9.4.2.2. Editing the > prompt

You can use the > prompt to edit predefined boot options. For example, select Test this media and install Red Hat Enterprise Linux 8 from the boot menu to display a full set of options.

NOTE

This procedure is for BIOS-based AMD64 and Intel 64 systems.

Prerequisites

- You have created bootable installation media (USB, CD or DVD).
- You have booted the installation from the media, and the installation boot menu is open.

Procedure

1. From the boot menu, select an option and press the Tab key on your keyboard. The > prompt is accessible and displays the available options.

2. Append the options that you require to the > prompt.

3. Press the Enter key on your keyboard to start the installation.

4. Press the Esc key on your keyboard to cancel editing and return to the boot menu.

9.4.2.3. Editing the GRUB2 menu

The GRUB2 menu is available on UEFI-based AMD64, Intel 64, and 64-bit ARM systems.

Prerequisites

- You have created bootable installation media (USB, CD or DVD).
- You have booted the installation from the media, and the installation boot menu is open.

Procedure

1. From the boot menu window, select the required option and press the e key on your keyboard.
2. Move the cursor to the kernel command line. On UEFI systems, the kernel command line starts with `linuxefi`.

3. Move the cursor to the end of the `linuxefi` kernel command line.

4. Edit the parameters as required. For example, to configure one or more network interfaces, add the `ip=` parameter at the end of the `linuxefi` kernel command line, followed by the required value.

5. When you finish editing, press `Ctrl+X` on your keyboard to start the installation using the specified options.

### 9.4.3. Installation source boot options

This section contains information about the various installation source boot options.

**inst.repo=**

The `inst.repo=` boot option specifies the installation source, that is, the location providing the package repositories and a valid `.treeinfo` file that describes them. For example: `inst.repo=cdrom`. The target of the `inst.repo=` option must be one of the following installation media:

- an installable tree, which is a directory structure containing the installation program images, packages, and repository data as well as a valid `.treeinfo` file
- a DVD (a physical disk present in the system DVD drive)
- an ISO image of the full Red Hat Enterprise Linux installation DVD, placed on a hard drive or a network location accessible to the system.

Use the `inst.repo=` boot option to configure different installation methods using different formats. The following table contains details of the `inst.repo=` boot option syntax:

<table>
<thead>
<tr>
<th>Table 9.1. <code>inst.repo=</code> installation source boot options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Source type</strong></td>
</tr>
<tr>
<td>CD/DVD drive</td>
</tr>
<tr>
<td>Mountable device</td>
</tr>
<tr>
<td>NFS Server</td>
</tr>
<tr>
<td>HTTP Server</td>
</tr>
<tr>
<td>HTTPS Server</td>
</tr>
</tbody>
</table>
Set disk device names with the following formats:

- Kernel device name, for example /dev/sda1 or sdb2
- File system label, for example LABEL=Flash or LABEL=RHEL8
- File system UUID, for example UUID=8176c7bf-04ff-403a-a832-9557f94e61db
  Non-alphanumeric characters must be represented as \xNN, where N/N is the hexadecimal representation of the character. For example, \x20 is a white space (" ").

`inst.addrepo=`

Use the `inst.addrepo=` boot option to add an additional repository that can be used as another installation source along with the main repository (`inst.repo=`). You can use the `inst.addrepo=` boot option multiple times during one boot. The following table contains details of the `inst.addrepo=` boot option syntax.

**NOTE**

The REPO_NAME is the name of the repository and is required in the installation process. These repositories are only used during the installation process; they are not installed on the installed system.

For more information about unified ISO, see Unified ISO

**Table 9.2. inst.addrepo installation source boot options**

<table>
<thead>
<tr>
<th>Installation source</th>
<th>Boot option format</th>
<th>Additional information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installable tree at a URL</td>
<td><code>inst.addrepo=REPO_NAME, [http,https,ftp]:/&lt;host&gt;/&lt;path&gt;</code></td>
<td>Looks for the installable tree at a given URL.</td>
</tr>
<tr>
<td>Installation source</td>
<td>Boot option format</td>
<td>Additional information</td>
</tr>
<tr>
<td>---------------------------------------------------------</td>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Installable tree at an NFS path</td>
<td><code>inst.addrepo=REPO_NAME,nfs://&lt;server&gt;/&lt;path&gt;</code></td>
<td>Looks for the installable tree at a given NFS path. A colon is required after the host. The installation program passes everything after <code>nfs://</code> directly to the mount command instead of parsing URLs according to RFC 2224.</td>
</tr>
<tr>
<td>Installable tree in the installation environment</td>
<td><code>inst.addrepo=REPO_NAME,file://&lt;path&gt;</code></td>
<td>Looks for the installable tree at the given location in the installation environment. To use this option, the repository must be mounted before the installation program attempts to load the available software groups. The benefit of this option is that you can have multiple repositories on one bootable ISO, and you can install both the main repository and additional repositories from the ISO. The path to the additional repositories is <code>/run/install/source/REPO_ISO_PATH</code>. Additional, you can mount the repository directory in the <code>%pre</code> section in the Kickstart file. The path must be absolute and start with <code>/</code>, for example <code>inst.addrepo=REPO_NAME,file:///&lt;path&gt;</code></td>
</tr>
<tr>
<td>Hard Drive</td>
<td><code>inst.addrepo=REPO_NAME,h:&lt;device&gt;:&lt;path&gt;</code></td>
<td>Mounts the given <code>&lt;device&gt;</code> partition and installs from the ISO that is specified by the <code>&lt;path&gt;</code>. If the <code>&lt;path&gt;</code> is not specified, the installation program looks for a valid installation ISO on the <code>&lt;device&gt;</code>. This installation method requires an ISO with a valid installable tree.</td>
</tr>
</tbody>
</table>

**inst.stage2=**

The `inst.stage2=` boot option specifies the location of the installation program’s runtime image. This option expects the path to a directory that contains a valid `.treeinfo` file and reads the runtime image location from the `.treeinfo` file. If the `.treeinfo` file is not available, the installation program attempts to load the image from `images/install.img`.

When the `inst.stage2` option is not specified, the installation program attempts to use the location specified with the `inst.repo` option.
Use this option when you want to manually specify the installation source in the installation program at a later time. For example, when you want to select the Content Delivery Network (CDN) as an installation source. The installation DVD and Boot ISO already contain a correct `inst.stage2` option to boot the installation program from the respective ISO.

If you want to specify an installation source, use the `inst.repo=` option instead.

NOTE

By default, the `inst.stage2=` boot option is used on the installation media and is set to a specific label; for example, `inst.stage2=hd:LABEL=RHEL-x-0-0-BaseOS-x86_64`. If you modify the default label of the file system that contains the runtime image, or if you use a customized procedure to boot the installation system, verify that the `inst.stage2=` boot option is set to the correct value.

`inst.noverifyssl`

Use the `inst.noverifyssl` boot option to prevent the installer from verifying SSL certificates for all HTTPS connections with the exception of additional Kickstart repositories, where `--noverifyssl` can be set per repository.

For example, if your remote installation source is using self-signed SSL certificates, the `inst.noverifyssl` boot option enables the installer to complete the installation without verifying the SSL certificates.

**Example when specifying the source using `inst.stage2=`**

```bash
inst.stage2=https://hostname/path_to_install_image/ inst.noverifyssl
```

**Example when specifying the source using `inst.repo=`**

```bash
inst.repo=https://hostname/path_to_install_repository/ inst.noverifyssl
```

`inst.stage2.all`

The `inst.stage2.all` boot option is used to specify several HTTP, HTTPS, or FTP sources. You can use the `inst.stage2=` boot option multiple times with the `inst.stage2.all` option to fetch the image from the sources sequentially until one succeeds. For example:

```bash
inst.stage2.all
inst.stage2=http://hostname1/path_to_install_tree/
inst.stage2=http://hostname2/path_to_install_tree/
inst.stage2=http://hostname3/path_to_install_tree/
```

`inst.dd=`

The `inst.dd=` boot option is used to perform a driver update during the installation. For more information on how to update drivers during installation, see the *Performing an advanced RHEL installation* document.

`inst.repo=hmc`

When booting from a Binary DVD, the installation program prompts you to enter additional kernel parameters. To set the DVD as an installation source, append the `inst.repo=hmc` option to the kernel parameters. The installation program then enables support element (SE) and hardware
management console (HMC) file access, fetches the images for stage2 from the DVD, and provides access to the packages on the DVD for software selection. This option eliminates the requirement of an external network setup and expands the installation options.

**inst.proxy=**

The *inst.proxy=* boot option is used when performing an installation from a HTTP, HTTPS, and FTP protocol. For example:

```
[PROTOCOL://][USERNAME[:PASSWORD]@]HOST[:PORT]
```

**inst.nosave=**

Use the *inst.nosave=* boot option to control the installation logs and related files that are not saved to the installed system, for example *input_ks*, *output_ks*, *all_ks*, *logs* and *all*. Multiple values can be combined as a comma-separated list, for example: *input_ks,logs*.

**NOTE**

The *inst.nosave* boot option is used for excluding files from the installed system that can’t be removed by a Kickstart %post script, such as logs and input/output Kickstart results.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>input_ks</td>
<td>Disables the ability to save the input Kickstart results.</td>
</tr>
<tr>
<td>output_ks</td>
<td>Disables the ability to save the output Kickstart results generated by the installation program.</td>
</tr>
<tr>
<td>all_ks</td>
<td>Disables the ability to save the input and output Kickstart results.</td>
</tr>
<tr>
<td>logs</td>
<td>Disables the ability to save all installation logs.</td>
</tr>
<tr>
<td>all</td>
<td>Disables the ability to save all Kickstart results, and all logs.</td>
</tr>
</tbody>
</table>

**inst.multilib**

Use the *inst.multilib* boot option to set DNF’s *multilib_policy* to *all*, instead of *best*.

**inst.memcheck**

The *inst.memcheck* boot option performs a check to verify that the system has enough RAM to complete the installation. If there isn’t enough RAM, the installation process is stopped. The system check is approximate and memory usage during installation depends on the package selection, user interface, for example graphical or text, and other parameters.

**inst.nomemcheck**

The *inst.nomemcheck* boot option does not perform a check to verify if the system has enough RAM to complete the installation. Any attempt to perform the installation with less than the recommended minimum amount of memory is unsupported, and might result in the installation
process failing.

9.4.4. Network boot options

This section contains information about commonly used network boot options.

NOTE

Initial network initialization is handled by `dracut`. For a complete list, see the `dracut.cmdline(7)` man page.

ip=

Use the `ip=` boot option to configure one or more network interfaces. To configure multiple interfaces, you can use the `ip` option multiple times, once for each interface; to do so, you must use the `rd.neednet=1` option, and you must specify a primary boot interface using the `bootdev` option. Alternatively, you can use the `ip` option once, and then use Kickstart to set up further interfaces. This option accepts several different formats. The following tables contain information about the most common options.

NOTE

In the following tables:

- The `ip` parameter specifies the client IP address and requires square brackets, for example `[2001:db8::99].`
- The `gateway` parameter is the default gateway. IPv6 addresses are also accepted.
- The `netmask` parameter is the netmask to be used. This can be either a full netmask (for example, 255.255.255.0) or a prefix (for example, 64).
- The `hostname` parameter is the host name of the client system. This parameter is optional.

Table 9.4. Network interface configuration boot option formats

<table>
<thead>
<tr>
<th>Configuration method</th>
<th>Boot option format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic configuration of any interface</td>
<td><code>ip=method</code></td>
</tr>
<tr>
<td>Automatic configuration of a specific interface</td>
<td><code>ip=interface:method</code></td>
</tr>
<tr>
<td>Static configuration</td>
<td><code>ip=ip::gateway:netmask:hostname:interface:none</code></td>
</tr>
</tbody>
</table>
The method **automatic configuration of a specific interface with an override** brings up the interface using the specified method of automatic configuration, such as `dhcp`, but overrides the automatically-obtained IP address, gateway, netmask, host name or other specified parameters. All parameters are optional, so specify only the parameters that you want to override.

The **method** parameter can be any of the following:

**Table 9.5. Automatic interface configuration methods**

<table>
<thead>
<tr>
<th>Automatic configuration method</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHCP</td>
<td><code>dhcp</code></td>
</tr>
<tr>
<td>IPv6 DHCP</td>
<td><code>dhcp6</code></td>
</tr>
<tr>
<td>IPv6 automatic configuration</td>
<td><code>auto6</code></td>
</tr>
<tr>
<td>iSCSI Boot Firmware Table (iBFT)</td>
<td><code>ibft</code></td>
</tr>
</tbody>
</table>

**NOTE**

- If you use a boot option that requires network access, such as `inst.ks=http://host/path`, without specifying the `ip` option, the installation program uses `ip=dhcp`.
- To connect to an iSCSI target automatically, you must activate a network device for accessing the target. The recommended way to activate a network is to use the `ip=ibft` boot option.

**nameserver=**

The `nameserver=` option specifies the address of the name server. You can use this option multiple times.

**NOTE**

The `ip=` parameter requires square brackets. However, an IPv6 address does not work with square brackets. An example of the correct syntax to use for an IPv6 address is `nameserver=2001:db8::1`.

**bootdev=**

The `bootdev=` option specifies the boot interface. This option is mandatory if you use more than one `ip` option.

**ifname=**

The `ifname=` options assigns an interface name to a network device with a given MAC address. You can use this option multiple times. The syntax is `ifname=interface:MAC`. For example:

`ifname=eth0:01:23:45:67:89:ab`
The `ifname=` option is the only supported way to set custom network interface names during installation.

**inst.dhcpclass=**

The `inst.dhcpclass=` option specifies the DHCP vendor class identifier. The `dhcppd` service sees this value as `vendor-class-identifier`. The default value is `anaconda-$(uname -srm)`.

**inst.waitfornet=**

Using the `inst.waitfornet=SECONDS` boot option causes the installation system to wait for network connectivity before installation. The value given in the `SECONDS` argument specifies the maximum amount of time to wait for network connectivity before timing out and continuing the installation process even if network connectivity is not present.

**vlan=**

Use the `vlan=` option to configure a Virtual LAN (VLAN) device on a specified interface with a given name. The syntax is `vlan=name:interface`. For example:

```
  vlan=vlan5:enp0s1
```

This configures a VLAN device named `vlan5` on the `enp0s1` interface. The name can take the following forms:

<table>
<thead>
<tr>
<th>Naming scheme</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN_PLUS_VID</td>
<td>vlan0005</td>
</tr>
<tr>
<td>VLAN_PLUS_VID_NO_PAD</td>
<td>vlan5</td>
</tr>
<tr>
<td>DEV_PLUS_VID</td>
<td>enp0s1.0005</td>
</tr>
<tr>
<td>DEV_PLUS_VID_NO_PAD</td>
<td>enp0s1.5</td>
</tr>
</tbody>
</table>

**bond=**

Use the `bond=` option to configure a bonding device with the following syntax: `bond=name[:interfaces][:options]`. Replace `name` with the bonding device name, `interfaces` with a comma-separated list of physical (Ethernet) interfaces, and `options` with a comma-separated list of bonding options. For example:

```
  bond=bond0:enp0s1,enp0s2:mode=active-backup,tx_queues=32,downdelay=5000
```

For a list of available options, execute the `modinfo` bonding command.

**team=**

Use the `team=` option to configure a team device with the following syntax: `team=name:interfaces`. Replace `name` with the desired name of the team device and `interfaces` with a comma-separated list of physical (Ethernet) devices to be used as underlying interfaces in the team device. For example:
bridge=

Use the `bridge=` option to configure a bridge device with the following syntax: `bridge=name:interfaces`. Replace `name` with the desired name of the bridge device and `interfaces` with a comma-separated list of physical (Ethernet) devices to be used as underlying interfaces in the bridge device. For example:

```
bridge=bridge0:enp0s1,enp0s2
```

Additional resources

- Configuring and managing networking

9.4.5. Console boot options

This section contains information about configuring boot options for your console, monitor display, and keyboard.

console=

Use the `console=` option to specify a device that you want to use as the primary console. For example, to use a console on the first serial port, use `console=ttyS0`. While using the `console=` argument, the installation will be started with a text UI similar when you boot with the `inst.text` option. You can use the `console=` option multiple times. If you do, the boot message is displayed on all specified consoles, but only the last one is used by the installation program. For example, if you specify `console=ttyS0 console=ttyS1`, the installation program uses `ttyS1`.

inst.lang=

Use the `inst.lang=` option to set the language that you want to use during the installation. The `locale -a | grep _` or `localectl list-locales | grep _` commands return a list of locales.

inst.singlelang

Use the `inst.singlelang` option to install in single language mode, which results in no available interactive options for the installation language and language support configuration. If a language is specified using the `inst.lang` boot option or the `lang` Kickstart command, then it is used. If no language is specified, the installation program defaults to `en_US.UTF-8`.

inst.geoloc=

Use the `inst.geoloc=` option to configure geolocation usage in the installation program. Geolocation is used to preset the language and time zone, and uses the following syntax: `inst.geoloc=value`. The `value` can be any of the following parameters:

<table>
<thead>
<tr>
<th>Value</th>
<th>Boot option format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable geolocation</td>
<td><code>inst.geoloc=0</code></td>
</tr>
<tr>
<td>Use the Fedora GeoIP API</td>
<td><code>inst.geoloc=provider_fedora_geoiop</code></td>
</tr>
</tbody>
</table>

Table 9.7. Values for the `inst.geoloc` boot option
<table>
<thead>
<tr>
<th>Value</th>
<th>Boot option format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use the Hostip.info GeoIP API</td>
<td><code>inst.geoloc=provider_hostip</code></td>
</tr>
</tbody>
</table>

If you do not specify the `inst.geoloc=` option, the installation program uses `provider_fedora_geoip`.

`inst.keymap=`

Use the `inst.keymap=` option to specify the keyboard layout that you want to use for the installation.

`inst.cmdline`

Use the `inst.cmdline` option to force the installation program to run in command-line mode. This mode does not allow any interaction, and you must specify all options in a Kickstart file or on the command line.

`inst.graphical`

Use the `inst.graphical` option to force the installation program to run in graphical mode. This mode is the default.

`inst.text`

Use the `inst.text` option to force the installation program to run in text mode instead of graphical mode.

`inst.noninteractive`

Use the `inst.noninteractive` boot option to run the installation program in a non-interactive mode. User interaction is not permitted in the non-interactive mode, and `inst.noninteractive` can be used with a graphical or text installation. When the `inst.noninteractive` option is used in text mode it behaves the same as the `inst.cmdline` option.

`inst.resolution=`

Use the `inst.resolution=` option to specify the screen resolution in graphical mode. The format is `NxM`, where `N` is the screen width and `M` is the screen height (in pixels). The lowest supported resolution is 1024x768.

`inst.vnc`

Use the `inst.vnc` option to run the graphical installation using VNC. You must use a VNC client application to interact with the installation program. When VNC sharing is enabled, multiple clients can connect. A system installed using VNC starts in text mode.

`inst.vncpassword=`

Use the `inst.vncpassword=` option to set a password on the VNC server that is used by the installation program.

`inst.vncconnect=`

Use the `inst.vncconnect=` option to connect to a listening VNC client at the given host location. For example `inst.vncconnect=<host>[:<port>]` The default port is 5900. This option can be used with `vncviewer -listen`.

`inst.xdriver=`

Use the `inst.xdriver=` option to specify the name of the X driver that you want to use both during installation and on the installed system.

`inst.usefbx`

Use the `inst.usefbx` option to prompt the installation program to use the frame buffer X driver instead of a hardware-specific driver. This option is equivalent to `inst.xdriver=fbdev`.

`modprobe.blacklist=`

Use the `modprobe.blacklist=` option to blacklist or completely disable one or more drivers. Drivers
(mods) that you disable using this option cannot load when the installation starts, and after the installation finishes, the installed system retains these settings. You can find a list of the blocklisted drivers in the `/etc/modprobe.d/` directory. Use a comma-separated list to disable multiple drivers. For example:

```bash
modprobe.blacklist=ahci,firewire_ohci
```

`inst.xtimeout=`

Use the `inst.xtimeout=` option to specify the timeout in seconds for starting X server.

`inst.sshd`

Use the `inst.sshd` option to start the `sshd` service during installation, so that you can connect to the system during the installation using SSH, and monitor the installation progress. For more information about SSH, see the `ssh(1)` man page. By default, the `sshd` option is automatically started only on the 64-bit IBM Z architecture. On other architectures, `sshd` is not started unless you use the `inst.sshd` option.

**NOTE**

During installation, the root account has no password by default. You can set a root password during installation with the `sshpw` Kickstart command.

`inst.kdump_addon=`

Use the `inst.kdump_addon=` option to enable or disable the Kdump configuration screen (add-on) in the installation program. This screen is enabled by default; use `inst.kdump_addon=off` to disable it. Disabling the add-on disables the Kdump screens in both the graphical and text-based interface as well as the `%addon com_redhat_kdump` Kickstart command.

### 9.4.6. Debug boot options

This section contains information about the options that you can use when debugging issues.

`inst.rescue`

Use the `inst.rescue` option to run the rescue environment. The option is useful for trying to diagnose and fix systems. For example, you can *repair a filesystem in rescue mode*.

`inst.updates=`

Use the `inst.updates=` option to specify the location of the `updates.img` file that you want to apply during installation. There are a number of sources for the updates.

**Table 9.8. `inst.updates=` source updates**

<table>
<thead>
<tr>
<th>Source</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updates from a network</td>
<td>The easiest way to use <code>inst.updates=</code> is to specify the network location of <code>updates.img</code>. This does not require any modification to the installation tree. To use this method, edit the kernel command line to include <code>inst.updates</code>.</td>
<td><code>inst.updates=http://some.website.com/path/to/updates.img</code>.</td>
</tr>
</tbody>
</table>
Updates from a disk image

You can save an `updates.img` on a floppy drive or a USB key. This can be done only with an `ext2` filesystem type of `updates.img`. To save the contents of the image on your floppy drive, insert the floppy disc and run the command:

```
dd if=updates.img of=/dev/fd0 bs=72k count=20
```

To use a USB key or flash media, replace `/dev/fd0` with the device name of your USB key.

Updates from an installation tree

If you are using a CD, hard drive, HTTP, or FTP install, you can save the `updates.img` in the installation tree so that all installations can detect the `.img` file. Save the file in the `images/` directory. The file name must be `updates.img`.

For NFS installs, there are two options: You can either save the image in the `images/` directory, or in the `RHupdates/` directory in the installation tree.

### Source Description Example

<table>
<thead>
<tr>
<th>Source</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updates from a disk image</td>
<td>You can save an <code>updates.img</code> on a floppy drive or a USB key. This can be</td>
<td><code>dd if=updates.img of=/dev/fd0 bs=72k count=20</code>. To use a USB key or</td>
</tr>
<tr>
<td></td>
<td>done only with an <code>ext2</code> filesystem type of <code>updates.img</code>. To save the</td>
<td>flash media, replace <code>/dev/fd0</code> with the device name of your USB key.</td>
</tr>
<tr>
<td></td>
<td>contents of the image on your floppy drive, insert the floppy disc and run</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the command.</td>
<td></td>
</tr>
<tr>
<td>Updates from an installation tree</td>
<td>If you are using a CD, hard drive, HTTP, or FTP install, you can save the</td>
<td>For NFS installs, there are two options: You can either save the image</td>
</tr>
<tr>
<td></td>
<td><code>updates.img</code> in the installation tree so that all installations can detect</td>
<td>in the <code>images/</code> directory, or in the <code>RHupdates/</code> directory in the</td>
</tr>
<tr>
<td></td>
<td>the <code>.img</code> file. Save the file in the <code>images/</code> directory. The file name</td>
<td>installation tree.</td>
</tr>
<tr>
<td></td>
<td>must be <code>updates.img</code>.</td>
<td></td>
</tr>
</tbody>
</table>

### inst.loglevel=

Use the `inst.loglevel=` option to specify the minimum level of messages logged on a terminal. This concerns only terminal logging; log files always contain messages of all levels. Possible values for this option from the lowest to highest level are: `debug`, `info`, `warning`, `error` and `critical`. The default value is `info`, which means that by default, the logging terminal displays messages ranging from `info` to `critical`.

### inst.syslog=

When installation starts, the `inst.syslog=` option sends log messages to the `syslog` process on the specified host. The remote `syslog` process must be configured to accept incoming connections.

### inst.virtiolog=

Use the `inst.virtiolog=` option to specify the virtio port (a character device at `/dev/virtio-ports/name) that you want to use for forwarding logs. The default value is `org.fedoraproject.anaconda.log.0`; if this port is present, it is used.

### inst.zram=

The `inst.zram=` option controls the usage of zRAM swap during installation. The option creates a compressed block device inside the system RAM and uses it for swap space instead of the hard drive. This allows the installation program to run with less available memory than is possible without compression, and it might also make the installation faster. By default, swap on zRAM is enabled on systems with 2 GiB or less RAM, and disabled on systems with more than 2 GiB of memory. You can
use this option to change this behavior; on a system with more than 2 GiB RAM, use \texttt{inst.zram=1} to enable the feature, and on systems with 2 GiB or less memory, use \texttt{inst.zram=0} to disable the feature.

\texttt{rd.live.ram}

If the \texttt{rd.live.ram} option is specified, the stage 2 image is copied into RAM. If the \texttt{rd.live.ram} option is specified, the stage 2 image (\texttt{images/install.img}) is copied into RAM. This increases the memory required for installation by the size of the image. The size may vary from 400 to 800MB.

\texttt{inst.nokill}

The \texttt{inst.nokill} option is a debugging option that prevents the installation program from rebooting when a fatal error occurs, or at the end of the installation process. Use the \texttt{inst.nokill} option to capture installation logs which would be lost upon reboot.

\texttt{inst.noshell}

Use \texttt{inst.noshell} option if you do not want a shell on terminal session 2 (tty2) during installation.

\texttt{inst.notmux}

Use \texttt{inst.notmux} option if you do not want to use tmux during installation. The output is generated without terminal control characters and is meant for non-interactive uses.

\texttt{inst.remotelog=}

You can use the \texttt{inst.remotelog=} option to send all of the logs to a remote host:port using a TCP connection. The connection is retired if there is no listener and the installation proceeds as normal.

\section*{9.4.7. Storage boot options}

\texttt{inst.nodmraid}

Use the \texttt{inst.nodmraid} option to disable \texttt{dmraid} support.

\begin{warning}
Use this option with caution. If you have a disk that is incorrectly identified as part of a firmware RAID array, it might have some stale RAID metadata on it that must be removed using the appropriate tool, for example, \texttt{dmraid} or \texttt{wipefs}.
\end{warning}

\texttt{inst.nompath}

Use the \texttt{inst.nompath} option to disable support for multipath devices. This option can be used for systems on which a false-positive is encountered which incorrectly identifies a normal block device as a multipath device. There is no other reason to use this option.

\begin{warning}
Use this option with caution. You should not use this option with multipath hardware. Using this option to attempt to install to a single path of a multipath is not supported.
\end{warning}
The **inst.gpt** boot option forces the installation program to install partition information to a GUID Partition Table (GPT) instead of a Master Boot Record (MBR). This option is not valid on UEFI-based systems, unless they are in BIOS compatibility mode. Normally, BIOS-based systems and UEFI-based systems in BIOS compatibility mode attempt to use the MBR schema for storing partitioning information, unless the disk is $2^{32}$ sectors in size or larger. Disk sectors are typically 512 bytes in size, meaning that this is usually equivalent to 2 TiB. Using the **inst.gpt** boot option changes this behavior, allowing a GPT to be written to smaller disks.

### 9.4.8. Kickstart boot options

This section contains information about the Kickstart boot options.

**inst.ks=**

Use the **inst.ks=** boot option to define the location of a Kickstart file that you want to use to automate the installation. You can then specify locations using any of the **inst.repo** formats. If you specify a device and not a path, the installation program looks for the Kickstart file in `/ks.cfg` on the device that you specify. If you use this option without specifying a device, the installation program uses the following option:

```
inst.ks=nfs:next-server:/filename
```

In the previous example, `next-server` is the DHCP next-server option or the IP address of the DHCP server itself, and `filename` is the DHCP filename option, or `/kickstart/`. If the given file name ends with the `/` character, `ip-kickstart` is appended. The following table contains an example.

<table>
<thead>
<tr>
<th>DHCP server address</th>
<th>Client address</th>
<th>Kickstart file location</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.122.1</td>
<td>192.168.122.100</td>
<td>192.168.122.1:/kickstart/192.168.12 2.100-kickstart</td>
</tr>
</tbody>
</table>

If a volume with a label of **OEMDRV** is present, the installation program attempts to load a Kickstart file named `ks.cfg`. If your Kickstart file is in this location, you do not need to use the **inst.ks=** boot option.

**inst.ks.all**

Specify this option to sequentially try multiple Kickstart file locations provided by multiple **inst.ks** options. The first successful location is used. This applies only to locations of type **http**, **https** or **ftp**, other locations are ignored.

**inst.ks.sendmac**

Use the **inst.ks.sendmac** option to add headers to outgoing HTTP requests that contain the MAC addresses of all network interfaces. For example:

```
X-RHN-Provisioning-MAC-0: eth0 01:23:45:67:89:ab
```

This can be useful when using **inst.ks=http** to provision systems.

**inst.ks.sendsn**
Use the `inst.ks.sendsn` option to add a header to outgoing HTTP requests. This header contains the system serial number, read from `/sys/class/dmi/id/product_serial`. The header has the following syntax:

```
X-System-Serial-Number: R8VA23D
```

Additional resources

- Full list of boot options

### 9.4.9. Advanced installation boot options

This section contains information about advanced installation boot options.

#### inst.kexec

The `inst.kexec` option allows the installation program to use the `kexec` system call at the end of the installation, instead of performing a reboot. The `inst.kexec` option loads the new system immediately, and bypasses the hardware initialization normally performed by the BIOS or firmware.

**IMPORTANT**

This option is deprecated and available as a Technology Preview only. For information on Red Hat scope of support for Technology Preview features, see the Technology Preview Features Support Scope document.

When `kexec` is used, device registers which would normally be cleared during a full system reboot, might stay filled with data, which could potentially create issues for some device drivers.

#### inst.multilib

Use the `inst.multilib` boot option to configure the system for multilib packages, that is, to allow installing 32-bit packages on a 64-bit AMD64 or Intel 64 system. Normally, on an AMD64 or Intel 64 system, only packages for this architecture (marked as x86_64) and packages for all architectures (marked as noarch) are installed. When you use the `inst.multilib` boot option, packages for 32-bit AMD or Intel systems (marked as i686) are automatically installed.

This applies only to packages directly specified in the `%packages` section. If a package is installed as a dependency, only the exact specified dependency is installed. For example, if you are installing the `bash` package which depends on the `glibc` package, the former is installed in multiple variants, while the latter is installed only in variants that the bash package requires.

#### selinux=0

By default, SELinux operates in permissive mode in the installation program, and in enforcing mode in the installed system. The `selinux=0` boot option disables the use of SELinux in the installation program and the installed system.

**NOTE**

The `selinux=0` and `inst.selinux=0` options are not the same. The `selinux=0` option disables the use of SELinux in the installation program and the installed system. The `inst.selinux=0` option disables SELinux only in the installation program. By default, SELinux operates in permissive mode in the installation program, so disabling SELinux has little effect.
Use the `inst.nonibftiscsiboot` boot option to place the boot loader on iSCSI devices that were not configured in the iSCSI Boot Firmware Table (iBFT).

### 9.4.10. Deprecated boot options

This section contains information about deprecated boot options. These options are still accepted by the installation program but they are deprecated and are scheduled to be removed in a future release of Red Hat Enterprise Linux.

**method**

The `method` option is an alias for `inst.repo`.

**dns**

Use `nameserver` instead of `dns`. Note that nameserver does not accept comma-separated lists; use multiple nameserver options instead.

**netmask, gateway, hostname**

The `netmask`, `gateway`, and `hostname` options are provided as part of the `ip` option.

**ip=bootif**

A PXE-supplied `BOOTIF` option is used automatically, so there is no requirement to use `ip=bootif`.

**ksdevice**

<table>
<thead>
<tr>
<th>Value</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not present</td>
<td>N/A</td>
</tr>
<tr>
<td><code>ksdevice=link</code></td>
<td>Ignored as this option is the same as the default behavior</td>
</tr>
<tr>
<td><code>ksdevice=bootif</code></td>
<td>Ignored as this option is the default if <code>BOOTIF=</code> is present</td>
</tr>
<tr>
<td><code>ksdevice=ibft</code></td>
<td>Replaced with <code>ip=ibft</code>. See <code>ip</code> for details</td>
</tr>
<tr>
<td><code>ksdevice=&lt;MAC&gt;</code></td>
<td>Replaced with <code>BOOTIF=${MAC://-}</code></td>
</tr>
<tr>
<td><code>ksdevice=&lt;DEV&gt;</code></td>
<td>Replaced with <code>bootdev</code></td>
</tr>
</tbody>
</table>

### 9.4.11. Removed boot options

This section contains the boot options that have been removed from Red Hat Enterprise Linux.

**NOTE**

`dracut` provides advanced boot options. For more information about `dracut`, see the `dracut.cmdline(7)` man page.
askmethod, asknetwork

initramfs is completely non-interactive, so the askmethod and asknetwork options have been removed. Instead, use inst.repo or specify the appropriate network options.

blacklist, nofirewire

The modprobe option handles blocklisting kernel modules; use modprobe.blacklist=<mod1>, <mod2>. You can blocklist the firewire module by using modprobe.blacklist=firewire_ohci.

inst.headless=

The headless option specified that the system that is being installed to does not have any display hardware, and that the installation program is not required to look for any display hardware.

inst.decorated

The inst.decorated option was used to specify the graphical installation in a decorated window. By default, the window is not decorated, so it doesn’t have a title bar, resize controls, and so on. This option was no longer required.

repo=nfsiso

Use the inst.repo=nfs: option.

serial

Use the console=ttys0 option.

updates

Use the inst.updates option.

essid, wepkey, wpakey

Dracut does not support wireless networking.

ethtool

This option was no longer required.

gdb

This option was removed as there are many options available for debugging dracut-based initramfs.

inst.mediacheck

Use the dracut option rd.live.check option.

ks=floppy

Use the inst.ks=hd:<device> option.

display

For a remote display of the UI, use the inst.vnc option.

utf8

This option was no longer required as the default TERM setting behaves as expected.

noipv6

ipv6 is built into the kernel and cannot be removed by the installation program. You can disable ipv6 using ipv6.disable=1. This setting is used by the installed system.

upgradeany

This option was no longer required as the installation program no longer handles upgrades.
APPENDIX I. KICKSTART SCRIPT FILE FORMAT REFERENCE

This reference describes in detail the kickstart file format.

I.1. KICKSTART FILE FORMAT

Kickstart scripts are plain text files that contain keywords recognized by the installation program, which serve as directions for the installation. Any text editor able to save files as ASCII text, such as Gedit or vim on Linux systems or Notepad on Windows systems, can be used to create and edit Kickstart files. The file name of your Kickstart configuration does not matter; however, it is recommended to use a simple name as you will need to specify this name later in other configuration files or dialogs.

**Commands**

Commands are keywords that serve as directions for installation. Each command must be on a single line. Commands can take options. Specifying commands and options is similar to using Linux commands in shell.

**Sections**

Certain special commands that begin with the percent `%` character start a section. Interpretation of commands in sections is different from commands placed outside sections. Every section must be finished with `%end` command.

**Section types**

The available sections are:

- **Add-on sections.** These sections use the `%addon addon_name` command.
- **Package selection sections.** Starts with `%packages`. Use it to list packages for installation, including indirect means such as package groups or modules.
- **Script sections.** These start with `%pre`, `%pre-install`, `%post`, and `%onerror`. These sections are not required.

**Command section**

The command section is a term used for the commands in the Kickstart file that are not part of any script section or `%packages` section.

**Script section count and ordering**

All sections except the command section are optional and can be present multiple times. When a particular type of script section is to be evaluated, all sections of that type present in the Kickstart are evaluated in order of appearance: two `%post` sections are evaluated one after another, in the order as they appear. However, you do not have to specify the various types of script sections in any order: it does not matter if there are `%post` sections before `%pre` sections.

**Comments**

Kickstart comments are lines starting with the hash `#` character. These lines are ignored by the installation program.

Items that are not required can be omitted. Omitting any required item results in the installation program changing to the interactive mode so that the user can provide an answer to the related item, just as during a regular interactive installation. It is also possible to declare the kickstart script as non-interactive with the `cmdline` command. In non-interactive mode, any missing answer aborts the installation process.
NOTE

If user interaction is needed during kickstart installation in text or graphical mode, enter only the windows where updates are mandatory to complete the installation. Entering spokes might lead to resetting the kickstart configuration. Resetting of the configuration applies specifically to the kickstart commands related to storage after entering the Installation Destination window.

I.2. PACKAGE SELECTION IN KICKSTART

Kickstart uses sections started by the %packages command for selecting packages to install. You can install packages, groups, environments, module streams, and module profiles this way.

I.2.1. Package selection section

Use the %packages command to begin a Kickstart section which describes the software packages to be installed. The %packages section must end with the %end command.

You can specify packages by environment, group, module stream, module profile, or by their package names. Several environments and groups that contain related packages are defined. See the repository/repodata/*-comps-repository.architecture.xml file on the Red Hat Enterprise Linux 8 Installation DVD for a list of environments and groups.

The *-comps-repository.architecture.xml file contains a structure describing available environments (marked by the <environment> tag) and groups (the <group> tag). Each entry has an ID, user visibility value, name, description, and package list. If the group is selected for installation, the packages marked mandatory in the package list are always installed, the packages marked default are installed if they are not specifically excluded elsewhere, and the packages marked optional must be specifically included elsewhere even when the group is selected.

You can specify a package group or environment using either its ID (the <id> tag) or name (the <name> tag).

If you are not sure what package should be installed, Red Hat recommends you to select the Minimal Install environment. Minimal Install provides only the packages which are essential for running Red Hat Enterprise Linux 8. This will substantially reduce the chance of the system being affected by a vulnerability. If necessary, additional packages can be added later after the installation. For more details on Minimal Install, see the Installing the Minimum Amount of Packages Required section of the Security Hardening document. Note that Initial Setup can not run after a system is installed from a Kickstart file unless a desktop environment and the X Window System were included in the installation and graphical login was enabled.

IMPORTANT

To install a 32-bit package on a 64-bit system:

- specify the --multilib option for the %packages section
- append the package name with the 32-bit architecture for which the package was built; for example, glibc.i686

I.2.2. Package selection commands

These commands can be used within the %packages section of a Kickstart file.
Specifying an environment

Specify an entire environment to be installed as a line starting with the `@^` symbols:

```
%packages
@^Infrastructure Server
@end
```

This installs all packages which are part of the *Infrastructure Server* environment. All available environments are described in the `repository/repodata/*/comps-repository.architecture.xml` file on the Red Hat Enterprise Linux 8 Installation DVD.

Only a single environment should be specified in the Kickstart file. If more environments are specified, only the last specified environment is used.

Specifying groups

Specify groups, one entry to a line, starting with an `@` symbol, and then the full group name or group id as given in the `*-comps-repository.architecture.xml` file. For example:

```
%packages
@X Window System
@Desktop
@Sound and Video
@end
```

The *Core* group is always selected - it is not necessary to specify it in the `%packages` section.

Specifying individual packages

Specify individual packages by name, one entry to a line. You can use the asterisk character (*) as a wildcard in package names. For example:

```
%packages
sqlite
curl
aspell
docbook*
@end
```

The `docbook*` entry includes the packages `docbook-dtds` and `docbook-style` that match the pattern represented with the wildcard.

Specifying profiles of module streams

Specify profiles for module streams, one entry to a line, using the syntax for profiles:

```
%packages
@module:stream/profile
@end
```

This installs all packages listed in the specified profile of the module stream.

- When a module has a default stream specified, you can leave it out. When the default stream is not specified, you must specify it.
• When a module stream has a default profile specified, you can leave it out. When the default profile is not specified, you must specify it.

• Installing a module multiple times with different streams is not possible.

• Installing multiple profiles of the same module and stream is possible.

Modules and groups use the same syntax starting with the @ symbol. When a module and a package group exist with the same name, the module takes precedence.

In Red Hat Enterprise Linux 8, modules are present only in the AppStream repository. To list available modules, use the `yum module list` command on an installed Red Hat Enterprise Linux 8 system.

It is also possible to enable module streams using the `module` Kickstart command and then install packages contained in the module stream by naming them directly.

**Excluding environments, groups, or packages**

Use a leading dash (-) to specify packages or groups to exclude from the installation. For example:

```
%packages
-@Graphical Administration Tools
-autofs
-ipa*compat
%end
```

**IMPORTANT**

Installing all available packages using only * in a Kickstart file is not supported.

You can change the default behavior of the `%packages` section by using several options. Some options work for the entire package selection, others are used with only specific groups.

**Additional resources**

• [Installing software](#)

• [Installing, managing, and removing user-space components](#)

**I.2.3. Common package selection options**

The following options are available for the `%packages` sections. To use an option, append it to the start of the package selection section. For example:

```
%packages --multilib --ignoremissing
```

**--default**

Install the default set of packages. This corresponds to the package set which would be installed if no other selections were made in the Package Selection screen during an interactive installation.

**--excludedocs**
Do not install any documentation contained within packages. In most cases, this excludes any files normally installed in the `/usr/share/doc` directory, but the specific files to be excluded depend on individual packages.

**--ignoremissing**
Ignore any packages, groups, module streams, module profiles, and environments missing in the installation source, instead of halting the installation to ask if the installation should be aborted or continued.

**--instLangs**
Specify a list of languages to install. Note that this is different from package group level selections. This option does not describe which package groups should be installed; instead, it sets RPM macros controlling which translation files from individual packages should be installed.

**--multilib**
Configure the installed system for multilib packages, to allow installing 32-bit packages on a 64-bit system, and install packages specified in this section as such. Normally, on an AMD64 and Intel 64 system, you can install only the x86_64 and the noarch packages. However, with the `--multilib` option, you can automatically install the 32-bit AMD and the i686 Intel system packages available, if any.

This only applies to packages explicitly specified in the `%packages` section. Packages which are only being installed as dependencies without being specified in the Kickstart file are only installed in architecture versions in which they are needed, even if they are available for more architectures.

User can configure Anaconda to install packages in **multilib** mode during the installation of the system. Use one of the following options to enable **multilib** mode:

1. Configure Kickstart file with the following lines:

   ```
   %packages --multilib --default
   %end
   ```

2. Add the inst.multilib boot option during booting the installation image.

**--nocore**
Disables installation of the `@Core` package group which is otherwise always installed by default. Disabling the `@Core` package group with `--nocore` should be only used for creating lightweight containers; installing a desktop or server system with `--nocore` will result in an unusable system.

**NOTES**

- Using `@Core` to exclude packages in the `@Core` package group does not work. The only way to exclude the `@Core` package group is with the `--nocore` option.

- The `@Core` package group is defined as a minimal set of packages needed for installing a working system. It is not related in any way to core packages as defined in the Package Manifest and Scope of Coverage Details.

**--excludeWeakdeps**
Disables installation of packages from weak dependencies. These are packages linked to the selected package set by Recommends and Supplements flags. By default weak dependencies will be installed.
--retries=
Sets the number of times YUM will attempt to download packages (retries). The default value is 10. This option only applies during the installation, and will not affect YUM configuration on the installed system.

--timeout=
Sets the YUM timeout in seconds. The default value is 30. This option only applies during the installation, and will not affect YUM configuration on the installed system.

I.2.4. Options for specific package groups
The options in this list only apply to a single package group. Instead of using them at the %packages command in the Kickstart file, append them to the group name. For example:

```plaintext
%packages
@Graphical Administration Tools --optional
%end
```

--nodefaults
Only install the group’s mandatory packages, not the default selections.

--optional
Install packages marked as optional in the group definition in the *-comps-repository.architecture.xml file, in addition to installing the default selections.

Note that some package groups, such as Scientific Support, do not have any mandatory or default packages specified - only optional packages. In this case the --optional option must always be used, otherwise no packages from this group will be installed.

IMPORTANT
The --nodefaults and --optional options cannot be used together. You can install only mandatory packages during the installation using --nodefaults and install the optional packages on the installed system post installation.

I.3. SCRIPTS IN KICKSTART FILE
A kickstart file can include the following scripts:

- %pre
- %pre-install
- %post

This section provides the following details about the scripts:

- Execution time
- Types of commands that can be included in the script
- Purpose of the script
- Script options
I.3.1. %pre script

The %pre scripts are run on the system immediately after the Kickstart file has been loaded, but before it is completely parsed and installation begins. Each of these sections must start with %pre and end with %end.

The %pre script can be used for activation and configuration of networking and storage devices. It is also possible to run scripts, using interpreters available in the installation environment. Adding a %pre script can be useful if you have networking and storage that needs special configuration before proceeding with the installation, or have a script that, for example, sets up additional logging parameters or environment variables.

Debugging problems with %pre scripts can be difficult, so it is recommended only to use a %pre script when necessary.

Commands related to networking, storage, and file systems are available to use in the %pre script, in addition to most of the utilities in the installation environment /sbin and /bin directories.

You can access the network in the %pre section. However, the name service has not been configured at this point, so only IP addresses work, not URLs.

NOTE

The pre script does not run in the chroot environment.

I.3.1.1. %pre script section options

The following options can be used to change the behavior of pre-installation scripts. To use an option, append it to the %pre line at the beginning of the script. For example:

```
%pre --interpreter=/usr/libexec/platform-python
   -- Python script omitted --
%end
```

--interpreter=

Allows you to specify a different scripting language, such as Python. Any scripting language available on the system can be used; in most cases, these are /usr/bin/sh, /usr/bin/bash, and /usr/libexec/platform-python.

Note that the platform-python interpreter uses Python version 3.6. You must change your Python scripts from previous RHEL versions for the new path and version. Additionally, platform-python is meant for system tools: Use the python36 package outside the installation environment. For more details about Python in Red Hat Enterprise Linux, see Introduction to Python in Configuring basic system settings.

--erroronfail

Displays an error and halts the installation if the script fails. The error message will direct you to where the cause of the failure is logged. The installed system might get into an unstable and unbootable state. You can use the inst.nokill option to debug the script.

--log=

Logs the script’s output into the specified log file. For example:

```
%pre --log=/tmp/ks-pre.log
```
I.3.2. %pre-install script

The commands in the pre-install script are run after the following tasks are complete:

- System is partitioned
- Filesystems are created and mounted under /mnt/sysroot
- Network has been configured according to any boot options and kickstart commands

Each of the %pre-install sections must start with %pre-install and end with %end.

The %pre-install scripts can be used to modify the installation, and to add users and groups with guaranteed IDs before package installation.

It is recommended to use the %post scripts for any modifications required in the installation. Use the %pre-install script only if the %post script falls short for the required modifications.

Note: The pre-install script does not run in chroot environment.

I.3.2.1. %pre-install script section options

The following options can be used to change the behavior of pre-install scripts. To use an option, append it to the %pre-install line at the beginning of the script. For example:

```
%pre-install --interpreter=/usr/libexec/platform-python
-- Python script omitted --
%end
```

Note that you can have multiple %pre-install sections, with same or different interpreters. They are evaluated in their order of appearance in the Kickstart file.

--interpreter=

Allows you to specify a different scripting language, such as Python. Any scripting language available on the system can be used; in most cases, these are /usr/bin/sh, /usr/bin/bash, and /usr/libexec/platform-python.

Note that the platform-python interpreter uses Python version 3.6. You must change your Python scripts from previous RHEL versions for the new path and version. Additionally, platform-python is meant for system tools: Use the python36 package outside the installation environment. For more details about Python in Red Hat Enterprise Linux, see Introduction to Python in Configuring basic system settings.

--erroronfail

Displays an error and halts the installation if the script fails. The error message will direct you to where the cause of the failure is logged. The installed system might get into an unstable and unbootable state. You can use the inst.nokill option to debug the script.

--log=

Logs the script’s output into the specified log file. For example:

```
%pre-install --log=/mnt/sysroot/root/ks-pre.log
```

I.3.3. %post script
The %post script is a post-installation script that is run after the installation is complete, but before the system is rebooted for the first time. You can use this section to run tasks such as system subscription.

You have the option of adding commands to run on the system once the installation is complete, but before the system is rebooted for the first time. This section must start with %post and end with %end.

The %post section is useful for functions such as installing additional software or configuring an additional name server. The post-install script is run in a chroot environment, therefore, performing tasks such as copying scripts or RPM packages from the installation media do not work by default. You can change this behavior using the --nochroot option as described below. Then the %post script will run in the installation environment, not in chroot on the installed target system.

Because post-install script runs in a chroot environment, most systemctl commands will refuse to perform any action.

Note that during execution of the %post section, the installation media must be still inserted.

I.3.3.1. %post script section options

The following options can be used to change the behavior of post-installation scripts. To use an option, append it to the %post line at the beginning of the script. For example:

```
%post --interpreter=/usr/libexec/platform-python
-- Python script omitted --
%end

--interpreter=

Allows you to specify a different scripting language, such as Python. For example:

```
%post --interpreter=/usr/libexec/platform-python

Any scripting language available on the system can be used; in most cases, these are /usr/bin/sh, /usr/bin/bash, and /usr/libexec/platform-python.

Note that the platform-python interpreter uses Python version 3.6. You must change your Python scripts from previous RHEL versions for the new path and version. Additionally, platform-python is meant for system tools: Use the python36 package outside the installation environment. For more details about Python in Red Hat Enterprise Linux, see Introduction to Python in Configuring basic system settings.

--nochroot

Allows you to specify commands that you would like to run outside of the chroot environment. The following example copies the file /etc/resolv.conf to the file system that was just installed.

```
%post --nochroot
cp /etc/resolv.conf /mnt/sysroot/etc/resolv.conf
%end

--erroronfail

Displays an error and halts the installation if the script fails. The error message will direct you to where the cause of the failure is logged. The installed system might get into an unstable and unbootable state. You can use the inst.nokill option to debug the script.

--log=
Logs the script's output into the specified log file. Note that the path of the log file must take into account whether or not you use the `--nochroot` option. For example, without `--nochroot`:

```
%post --log=/root/ks-post.log
```

and with `--nochroot`:

```
%post --nochroot --log=/mnt/sysroot/root/ks-post.log
```

### I.3.3.2. Example: Mounting NFS in a post-install script

This example of a `%post` section mounts an NFS share and executes a script named `runme` located at `/usr/new-machines/` on the share. Note that NFS file locking is not supported while in Kickstart mode, therefore the `-o nolock` option is required.

```
# Start of the %post section with logging into /root/ks-post.log
%post --log=/root/ks-post.log

# Mount an NFS share
mkdir /mnt/temp
mount -o nolock 10.10.0.2:/usr/new-machines /mnt/temp
openvt -s -w -- /mnt/temp/runme
umount /mnt/temp

# End of the %post section
%end
```

### I.3.3.3. Example: Running subscription-manager as a post-install script

One of the most common uses of post-installation scripts in Kickstart installations is automatic registration of the installed system using Red Hat Subscription Manager. The following is an example of automatic subscription in a `%post` script:

```
%post --log=/root/ks-post.log
subscription-manager register --username=admin@example.com --password=secret --auto-attach
%end
```

The subscription-manager command-line script registers a system to a Red Hat Subscription Management server (Customer Portal Subscription Management, Satellite 6, or CloudForms System Engine). This script can also be used to assign or attach subscriptions automatically to the system that best-match that system. When registering to the Customer Portal, use the Red Hat Network login credentials. When registering to Satellite 6 or CloudForms System Engine, you may also need to specify more subscription-manager options like `--serverurl`, `--org`, `--environment` as well as credentials provided by your local administrator. Note that credentials in the form of an `--org` `--activationkey` combination is a good way to avoid exposing `--username` `--password` values in shared kickstart files.

Additional options can be used with the registration command to set a preferred service level for the system and to restrict updates and errata to a specific minor release version of RHEL for customers with Extended Update Support subscriptions that need to stay fixed on an older stream.

I.4. ANACONDA CONFIGURATION SECTION

Additional installation options can be configured in the %anaconda section of your Kickstart file. This section controls the behavior of the user interface of the installation system.

This section must be placed towards the end of the Kickstart file, after Kickstart commands, and must start with %anaconda and end with %end.

Currently, the only command that can be used in the %anaconda section is pwpolicy.

Example I.1. Sample %anaconda script

The following is an example %anaconda section:

```
%anaconda
   pwpolicy root --minlen=10 --strict
%end
```

This example %anaconda section sets a password policy which requires that the root password be at least 10 characters long, and strictly forbids passwords which do not match this requirement.

I.5. KICKSTART ERROR HANDLING SECTION

Starting with Red Hat Enterprise Linux 7, Kickstart installations can contain custom scripts which are run when the installation program encounters a fatal error. For example, an error in a package that has been requested for installation, failure to start VNC when specified, or an error when scanning storage devices. Installation cannot continue after such an error has occurred. The installation program will run all %onerror scripts in the order they are provided in the Kickstart file. In addition, %onerror scripts will be run in the event of a traceback.

Each %onerror script is required to end with %end.

Error handling sections accept the following options:

--erroronfail

Displays an error and halts the installation if the script fails. The error message will direct you to where the cause of the failure is logged. The installed system might get into an unstable and unbootable state. You can use the inst.nokill option to debug the script.

--interpreter=

Allows you to specify a different scripting language, such as Python. For example:

```
%onerror --interpreter=/usr/libexec/platform-python
```

Any scripting language available on the system can be used; in most cases, these are /usr/bin/sh, /usr/bin/bash, and /usr/libexec/platform-python.

Note that the platform-python interpreter uses Python version 3.6. You must change your Python scripts from previous RHEL versions for the new path and version. Additionally, platform-python is meant for system tools: Use the python36 package outside the installation environment. For more details about Python in Red Hat Enterprise Linux, see Introduction to Python in Configuring basic system settings.
--log=
Logs the script’s output into the specified log file.

I.6. KICKSTART ADD-ON SECTIONS

Starting with Red Hat Enterprise Linux 7, Kickstart installations support add-ons. These add-ons can expand the basic Kickstart (and Anaconda) functionality in many ways.

To use an add-on in your Kickstart file, use the `%addon addon_name options` command, and finish the command with an `%end` statement, similar to pre-installation and post-installation script sections. For example, if you want to use the Kdump add-on, which is distributed with Anaconda by default, use the following commands:

```
%addon com_redhat_kdump --enable --reserve-mb=auto
%end
```

The `%addon` command does not include any options of its own – all options are dependent on the actual add-on.
APPENDIX J. KICKSTART COMMANDS AND OPTIONS REFERENCE

This reference is a complete list of all Kickstart commands supported by the Red Hat Enterprise Linux installation program program. The commands are sorted alphabetically in a few broad categories. If a command can fall under multiple categories, it is listed in all of them.

J.1. KICKSTART CHANGES

The following sections describe the changes in Kickstart commands and options in Red Hat Enterprise Linux 8.

auth or authconfig is deprecated in RHEL 8

The auth or authconfig Kickstart command is deprecated in Red Hat Enterprise Linux 8 because the authconfig tool and package have been removed.

Similarly to authconfig commands issued on command line, authconfig commands in Kickstart scripts now use the authselect-compat tool to run the new authselect tool. For a description of this compatibility layer and its known issues, see the manual page authselect-migration(7). The installation program will automatically detect use of the deprecated commands and install on the system the authselect-compat package to provide the compatibility layer.

Kickstart no longer supports Btrfs

The Btrfs file system is not supported from Red Hat Enterprise Linux 8. As a result, the Graphical User Interface (GUI) and the Kickstart commands no longer support Btrfs.

Using Kickstart files from previous RHEL releases

If you are using Kickstart files from previous RHEL releases, see the Repositories section of the Considerations in adopting RHEL 8 document for more information about the Red Hat Enterprise Linux 8 BaseOS and AppStream repositories.

J.1.1. Deprecated Kickstart commands and options

The following Kickstart commands and options have been deprecated in Red Hat Enterprise Linux 8.

Where only specific options are listed, the base command and its other options are still available and not deprecated.

- auth or authconfig - use authselect instead
- device
- deviceprobe
- dmraid
- install - use the subcommands or methods directly as commands
- multipath
- bootloader --upgrade
- ignoredisk --interactive
- partition --active
- reboot --kexec

Except the auth or authconfig command, using the commands in Kickstart files prints a warning in the logs.

You can turn the deprecated command warnings into errors with the inst.ksstrict boot option, except for the auth or authconfig command.

J.1.2. Removed Kickstart commands and options

The following Kickstart commands and options have been completely removed in Red Hat Enterprise Linux 8. Using them in Kickstart files will cause an error.

- device
- deviceprobe
- dmraid
- install - use the subcommands or methods directly as commands
- multipath
- bootloader --upgrade
- ignoredisk --interactive
- partition --active
- harddrive --biospart
- upgrade (This command had already previously been deprecated.)
- btrfs
- part/partition btrfs
- part --fstype btrfs or partition --fstype btrfs
- logvol --fstype btrfs
- raid --fstype btrfs
- unsupported_hardware

Where only specific options and values are listed, the base command and its other options are still available and not removed.

J.2. KICKSTART COMMANDS FOR INSTALLATION PROGRAM CONFIGURATION AND FLOW CONTROL

The Kickstart commands in this list control the mode and course of installation, and what happens at its end.
J.2.1. cdrom

The cdrom Kickstart command is optional. It performs the installation from the first optical drive on the system.

Syntax

cdrom

Notes

- Previously, the cdrom command had to be used together with the install command. The install command has been deprecated and cdrom can be used on its own, because it implies install.
- This command has no options.
- To actually run the installation, one of cdrom, harddrive, hmc, nfs, liveimg, or url must be specified.

J.2.2. cmdline

The cmdline Kickstart command is optional. It performs the installation in a completely non-interactive command line mode. Any prompt for interaction halts the installation.

Syntax

cmdline

Notes

- For a fully automatic installation, you must either specify one of the available modes (graphical, text, or cmdline) in the Kickstart file, or you must use the console= boot option. If no mode is specified, the system will use graphical mode if possible, or prompt you to choose from VNC and text mode.
- This command has no options.
- This mode is useful on 64-bit IBM Z systems with the x3270 terminal.

J.2.3. driverdisk

The driverdisk Kickstart command is optional. Use it to provide additional drivers to the installation program.

Driver disks can be used during Kickstart installations to provide additional drivers not included by default. You must copy the driver disks contents to the root directory of a partition on the system’s hard drive. Then, you must use the driverdisk command to specify that the installation program should look for a driver disk and its location.

Syntax

driverdisk [partition|--source=url|--biospart=biospart]
Options

You must specify the location of driver disk in one way out of these:

- **partition** - Partition containing the driver disk. Note that the partition must be specified as a full path (for example, `/dev/sdb1`), not just the partition name (for example, `sdb1`).

- **--source** - URL for the driver disk. Examples include:
  
  - `driverdisk --source=ftp://path/to/dd.img`
  - `driverdisk --source=http://path/to/dd.img`
  - `driverdisk --source=nfs:host:/path/to/dd.img`

- **--biospart** - BIOS partition containing the driver disk (for example, `82p2`).

Notes

Driver disks can also be loaded from a hard disk drive or a similar device instead of being loaded over the network or from `initrd`. Follow this procedure:

1. Load the driver disk on a hard disk drive, a USB or any similar device.
2. Set the label, for example, `DD`, to this device.
3. Add the following line to your Kickstart file:

   ```
   driverdisk LABEL=DD:/e1000.rpm
   ```

Replace `DD` with a specific label and replace `e1000.rpm` with a specific name. Use anything supported by the `inst.repo` command instead of `LABEL` to specify your hard disk drive.

J.2.4. eula

The `eula` Kickstart command is optional. Use this option to accept the End User License Agreement (EULA) without user interaction. Specifying this option prevents Initial Setup from prompting you to accept the license agreement after you finish the installation and reboot the system for the first time.

**Syntax**

```
eula [--agreed]
```

**Options**

- **--agreed** (required) - Accept the EULA. This option must always be used, otherwise the `eula` command is meaningless.

J.2.5. firstboot

The `firstboot` Kickstart command is optional. It determines whether the Initial Setup application starts the first time the system is booted. If enabled, the `initial-setup` package must be installed. If not specified, this option is disabled by default.

**Syntax**

```
firstboot OPTIONS
```
Options

- `--enable` or `--enabled` - Initial Setup is started the first time the system boots.
- `--disable` or `--disabled` - Initial Setup is not started the first time the system boots.
- `--reconfig` - Enable the Initial Setup to start at boot time in reconfiguration mode. This mode enables the root password, time & date, and networking & host name configuration options in addition to the default ones.

J.2.6. graphical

The **graphical** Kickstart command is optional. It performs the installation in graphical mode. This is the default.

Syntax

```
graphical [--non-interactive]
```

Options

- `--non-interactive` - Performs the installation in a completely non-interactive mode. This mode will terminate the installation when user interaction is required.

Notes

- For a fully automatic installation, you must either specify one of the available modes (**graphical**, **text**, or **cmdline**) in the Kickstart file, or you must use the `console=` boot option. If no mode is specified, the system will use graphical mode if possible, or prompt you to choose from VNC and text mode.

J.2.7. halt

The **halt** Kickstart command is optional.

Halt the system after the installation has successfully completed. This is similar to a manual installation, where Anaconda displays a message and waits for the user to press a key before rebooting. During a Kickstart installation, if no completion method is specified, this option is used as the default.

Syntax

```
halt
```

Notes

- The `halt` command is equivalent to the `shutdown -H` command. For more details, see the `shutdown(8)` man page.
- For other completion methods, see the `poweroff`, `reboot`, and `shutdown` commands.
- This command has no options.
J.2.8. harddrive

The harddrive Kickstart command is optional. It performs the installation from a Red Hat installation tree or full installation ISO image on a local drive. The drive must be formatted with a file system the installation program can mount: ext2, ext3, ext4, vfat, or xfs.

Syntax

```
harddrive OPTIONS
```

Options

- `--partition=` - Partition to install from (such as sdb2).
- `--dir=` - Directory containing the variant directory of the installation tree, or the ISO image of the full installation DVD.

Example

```
harddrive --partition=hdb2 --dir=/tmp/install-tree
```

Notes

- Previously, the harddrive command had to be used together with the install command. The install command has been deprecated and harddrive can be used on its own, because it implies install.
- To actually run the installation, one of cdrom, harddrive, hmc, nfs, liveimg, or url must be specified.

J.2.9. install (deprecated)

IMPORTANT

The install Kickstart command is deprecated in Red Hat Enterprise Linux 8. Use its methods as separate commands.

The install Kickstart command is optional. It specifies the default installation mode.

Syntax

```
install installation_method
```

Notes

- The install command must be followed by an installation method command. The installation method command must be on a separate line.
- The methods include:
  - cdrom
J.2.10. liveimg

The liveimg Kickstart command is optional. It performs the installation from a disk image instead of packages.

Syntax

```
liveimg --url=SOURCE [OPTIONS]
```

Mandatory options

- `--url=` - The location to install from. Supported protocols are HTTP, HTTPS, FTP, and file.

Optional options

- `--url=` - The location to install from. Supported protocols are HTTP, HTTPS, FTP, and file.
- `--proxy=` - Specify an HTTP, HTTPS or FTP proxy to use while performing the installation.
- `--checksum=` - An optional argument with the SHA256 checksum of the image file, used for verification.
- `--noverifyssl` - Disable SSL verification when connecting to an HTTPS server.

Example

```
liveimg --url=file:///images/install/squashfs.img --
checksum=03825f567f17705100de3308a20354b4d81ac9d8bed4bb4692b2381045e56197 --
noverifyssl
```

Notes

- The image can be the squashfs.img file from a live ISO image, a compressed tar file (.tar, .tbz, .tgz, .txz, .tar.bz2, .tar.gz, or .tar.xz), or any file system that the installation media can mount. Supported file systems are ext2, ext3, ext4, vfat, and xfs.
- When using the liveimg installation mode with a driver disk, drivers on the disk will not automatically be included in the installed system. If necessary, these drivers should be installed manually, or in the %post section of a kickstart script.
- To actually run the installation, one of cdrom, harddrive, hmc, nfs, liveimg, or url must be specified.
Previously, the `liveimg` command had to be used together with the `install` command. The `install` command has been deprecated and `liveimg` can be used on its own, because it implies `install`.

### J.2.11. logging

The **logging** Kickstart command is optional. It controls the error logging of Anaconda during installation. It has no effect on the installed system.

**NOTE**

Logging is supported over TCP only. For remote logging, ensure that the port number that you specify in `--port=` option is open on the remote server. The default port is 514.

**Syntax**

```
logging OPTIONS
```

**Optional options**

- `--host=` - Send logging information to the given remote host, which must be running a syslogd process configured to accept remote logging.

- `--port=` - If the remote syslogd process uses a port other than the default, set it using this option.

- `--level=` - Specify the minimum level of messages that appear on tty3. All messages are still sent to the log file regardless of this level, however. Possible values are `debug`, `info`, `warning`, `error`, or `critical`.

### J.2.12. mediacheck

The **mediacheck** Kickstart command is optional. This command forces the installation program to perform a media check before starting the installation. This command requires that installations be attended, so it is disabled by default.

**Syntax**

```
mediacheck
```

**Notes**

- This Kickstart command is equivalent to the `rd.live.check` boot option.

- This command has no options.

### J.2.13. nfs

The **nfs** Kickstart command is optional. It performs the installation from a specified NFS server.

**Syntax**

---

Red Hat Enterprise Linux 8 System Design Guide
nfs OPTIONS

Options

- `--server=` - Server from which to install (host name or IP).
- `--dir=` - Directory containing the `variant` directory of the installation tree.
- `--opts=` - Mount options to use for mounting the NFS export. (optional)

Example

```
nfs --server=nfsserver.example.com --dir=/tmp/install-tree
```

Notes

- Previously, the `nfs` command had to be used together with the `install` command. The `install` command has been deprecated and `nfs` can be used on its own, because it implies `install`.
- To actually run the installation, one of `cdrom`, `harddrive`, `hmc`, `nfs`, `liveimg`, or `url` must be specified.

J.2.14. ostreesetup

The `ostreesetup` Kickstart command is optional. It is used to set up OStree-based installations.

Syntax

```
ostreesetup --osname=OSNAME [--remote=REMOTE] --url=URL --ref=REF [--nogpg]
```

Mandatory options:

- `--osname=OSNAME` - Management root for OS installation.
- `--url=URL` - URL of the repository to install from.
- `--ref=REF` - Name of the branch from the repository to be used for installation.

Optional options:

- `--remote=REMOTE` - Management root for OS installation.
- `--nogpg` - Disable GPG key verification.

Notes

- For more information about the OStree tools, see the upstream documentation: https://ostree.readthedocs.io/en/latest/

J.2.15. poweroff

The `poweroff` Kickstart command is optional. It shuts down and powers off the system after the
The `poweroff` Kickstart command is optional. It shuts down and powers off the system after the installation has successfully completed. Normally during a manual installation, Anaconda displays a message and waits for the user to press a key before rebooting.

**Syntax**

```
poweroff
```

**Notes**

- The `poweroff` option is equivalent to the `shutdown -P` command. For more details, see the `shutdown(8)` man page.
- For other completion methods, see the `halt`, `reboot`, and `shutdown` Kickstart commands. The `halt` option is the default completion method if no other methods are explicitly specified in the Kickstart file.
- The `poweroff` command is highly dependent on the system hardware in use. Specifically, certain hardware components such as the BIOS, APM (advanced power management), and ACPI (advanced configuration and power interface) must be able to interact with the system kernel. Consult your hardware documentation for more information on your system’s APM/ACPI abilities.
- This command has no options.

### J.2.16. reboot

The `reboot` Kickstart command is optional. It instructs the installation program to reboot after the installation is successfully completed (no arguments). Normally, Kickstart displays a message and waits for the user to press a key before rebooting.

**Syntax**

```
reboot OPTIONS
```

**Options**

- `--eject` - Attempt to eject the bootable media (DVD, USB, or other media) before rebooting.
- `--kexec` - Uses the `kexec` system call instead of performing a full reboot, which immediately loads the installed system into memory, bypassing the hardware initialization normally performed by the BIOS or firmware.

**IMPORTANT**

This option is deprecated and available as a Technology Preview only. For information on Red Hat scope of support for Technology Preview features, see the Technology Preview Features Support Scope document.

When `kexec` is used, device registers (which would normally be cleared during a full system reboot) might stay filled with data, which could potentially create issues for some device drivers.

**Notes**
• Use of the reboot option might result in an endless installation loop, depending on the installation media and method.

• The reboot option is equivalent to the shutdown -r command. For more details, see the shutdown(8) man page.

• Specify reboot to automate installation fully when installing in command line mode on 64-bit IBM Z.

• For other completion methods, see the halt, poweroff, and shutdown Kickstart options. The halt option is the default completion method if no other methods are explicitly specified in the Kickstart file.

J.2.17. rhsm

The rhsm Kickstart command is optional. It instructs the installation program to register and install RHEL from the CDN.

NOTE

The rhsm Kickstart command removes the requirement of using custom %post scripts when registering the system.

Options

• --organization= - Uses the organization id to register and install RHEL from the CDN.

• --activation-key= - Uses the activation key to register and install RHEL from the CDN. Option can be used multiple times, once per activation key, as long as the activation keys used are registered to your subscription.

• --connect-to-insights - Connects the target system to Red Hat Insights.

• --proxy= - Sets the HTTP proxy.

• --server-hostname= - Sets Satellite instance URL. Use this option if you want to register to a Satellite instance instead to Red Hat subscription infrastructure.

J.2.18. shutdown

The shutdown Kickstart command is optional. It shuts down the system after the installation has successfully completed.

Syntax

```
shutdown
```

Notes

• The shutdown Kickstart option is equivalent to the shutdown command. For more details, see the shutdown(8) man page.

• For other completion methods, see the halt, poweroff, and reboot Kickstart options. The halt option is the default completion method if no other methods are explicitly specified in the Kickstart file.
This command has no options.

J.2.19. sshpw

The `sshpw` Kickstart command is optional.

During the installation, you can interact with the installation program and monitor its progress over an SSH connection. Use the `sshpw` command to create temporary accounts through which to log on. Each instance of the command creates a separate account that exists only in the installation environment. These accounts are not transferred to the installed system.

**Syntax**

```
sshpw --username=name [OPTIONS] password
```

**Mandatory options**

- `--username=name` - Provides the name of the user. This option is required.
- `password` - The password to use for the user. This option is required.

**Optional options**

- `--iscrypted` - If this option is present, the password argument is assumed to already be encrypted. This option is mutually exclusive with `--plaintext`. To create an encrypted password, you can use Python:

  ```
  $ python3 -c 'import crypt, getpass; pw=getpass.getpass(); print(crypt.crypt(pw) if pw==getpass.getpass("Confirm: ") else exit())'
  ```

  This generates a sha512 crypt-compatible hash of your password using a random salt.
- `--plaintext` - If this option is present, the password argument is assumed to be in plain text. This option is mutually exclusive with `--iscrypted`
- `--lock` - If this option is present, this account is locked by default. This means that the user will not be able to log in from the console.
- `--sshkey` - If this is option is present, then the `<password>` string is interpreted as an ssh key value.

**Notes**

- By default, the ssh server is not started during the installation. To make ssh available during the installation, boot the system with the kernel boot option `inst.sshd`.
- If you want to disable root ssh access, while allowing another user ssh access, use the following:

  ```
  sshpw --username=example_username example_password --plaintext
  sshpw --username=root example_password --lock
  ```

- To simply disable root ssh access, use the following:

  ```
  sshpw --username=root example_password --lock
  ```
J.2.20. text

The text Kickstart command is optional. It performs the Kickstart installation in text mode. Kickstart installations are performed in graphical mode by default.

Syntax

```
  text [--non-interactive]
```

Options

- **--non-interactive** - Performs the installation in a completely non-interactive mode. This mode will terminate the installation when user interaction is required.

Notes

- Note that for a fully automatic installation, you must either specify one of the available modes (graphical, text, or cmdline) in the Kickstart file, or you must use the console= boot option. If no mode is specified, the system will use graphical mode if possible, or prompt you to choose from VNC and text mode.

J.2.21. url

The url Kickstart command is optional. It is used to install from an installation tree image on a remote server using the FTP, HTTP, or HTTPS protocol. You can only specify one URL.

Syntax

```
  url --url=FROM [OPTIONS]
```

Mandatory options

- **--url=FROM** - Specifies the HTTP, HTTPS, FTP, or file location to install from.

Optional options

- **--mirrorlist=** - Specifies the mirror URL to install from.
- **--proxy=** - Specifies an HTTP, HTTPS, or FTP proxy to use during the installation.
- **--noverifyssl** - Disables SSL verification when connecting to an HTTPS server.
- **--metalink=URL** - Specifies the metalink URL to install from. Variable substitution is done for $releasever and $basearch in the URL.

Examples

- To install from a HTTP server:
  ```
  url --url=http://server/path
  ```
- To install from a FTP server:
url --url=ftp://username:password@server/path

- To install from a local file:
  
  liveimg --url=file:///images/install/squashfs.img --noverifyssl

Notes

- Previously, the url command had to be used together with the install command. The install command has been deprecated and url can be used on its own, because it implies install.

- To actually run the installation, one of cdrom, harddrive, hmc, nfs, liveimg, or url must be specified.

J.2.22. vnc

The vnc Kickstart command is optional. It allows the graphical installation to be viewed remotely through VNC.

This method is usually preferred over text mode, as there are some size and language limitations in text installations. With no additional options, this command starts a VNC server on the installation system with no password and displays the details required to connect to it.

Syntax

vnc [--host=host_name] [--port=port] [--password=password]

Options

--host=
  Connect to the VNC viewer process listening on the given host name.

--port=
  Provide a port that the remote VNC viewer process is listening on. If not provided, Anaconda uses the VNC default port of 5900.

--password=
  Set a password which must be provided to connect to the VNC session. This is optional, but recommended.

Additional resources

- Performing a remote RHEL installation using VNC

J.2.23. %include

The %include Kickstart command is optional.

Use the %include command to include the contents of another file in the Kickstart file as if the contents were at the location of the %include command in the Kickstart file.

This inclusion is evaluated only after the %pre script sections and can thus be used to include files generated by scripts in the %pre sections. To include files before evaluation of %pre sections, use the %ksappend command.
%include path/to/file

J.2.24. %ksappend

The %ksappend Kickstart command is optional.

Use the %ksappend command to include the contents of another file in the Kickstart file as if the contents were at the location of the %ksappend command in the Kickstart file.

This inclusion is evaluated before the %pre script sections, unlike inclusion with the %include command.

%ksappend path/to/file

J.3. KICKSTART COMMANDS FOR SYSTEM CONFIGURATION

The Kickstart commands in this list configure further details on the resulting system such as users, repositories, or services.

J.3.1. auth or authconfig (deprecated)

IMPORTANT

Use the new authselect command instead of the deprecated auth or authconfig Kickstart command. auth and authconfig are available only for limited backwards compatibility.

The auth or authconfig Kickstart command is optional. It sets up the authentication options for the system using the authconfig tool, which can also be run on the command line after the installation finishes.

authconfig [OPTIONS]

Notes

- Previously, the auth or authconfig Kickstart commands called the authconfig tool. This tool has been deprecated in Red Hat Enterprise Linux 8. These Kickstart commands now use the authselect-compat tool to call the new authselect tool. For a description of the compatibility layer and its known issues, see the manual page authselect-migration(7). The installation program will automatically detect use of the deprecated commands and install on the system the authselect-compat package to provide the compatibility layer.

- Passwords are shadowed by default.
When using OpenLDAP with the SSL protocol for security, make sure that the SSLv2 and SSLv3 protocols are disabled in the server configuration. This is due to the POODLE SSL vulnerability (CVE-2014-3566). See https://access.redhat.com/solutions/1234843 for details.

J.3.2. authselect

The authselect Kickstart command is optional. It sets up the authentication options for the system using the authselect command, which can also be run on the command line after the installation finishes.

Syntax

```
authselect [OPTIONS]
```

Notes

- This command passes all options to the authselect command. Refer to the authselect(8) manual page and the authselect --help command for more details.
- This command replaces the deprecated auth or authconfig commands deprecated in Red Hat Enterprise Linux 8 together with the authconfig tool.
- Passwords are shadowed by default.
- When using OpenLDAP with the SSL protocol for security, make sure that the SSLv2 and SSLv3 protocols are disabled in the server configuration. This is due to the POODLE SSL vulnerability (CVE-2014-3566). See https://access.redhat.com/solutions/1234843 for details.

J.3.3. firewall

The firewall Kickstart command is optional. It specifies the firewall configuration for the installed system.

Syntax

```
firwall --enabled|--disabled [incoming] [OPTIONS]
```

Mandatory options

- `--enabled` or `--enable` - Reject incoming connections that are not in response to outbound requests, such as DNS replies or DHCP requests. If access to services running on this machine is needed, you can choose to allow specific services through the firewall.
- `--disabled` or `--disable` - Do not configure any iptables rules.

Optional options

- `--trust` - Listing a device here, such as `em1`, allows all traffic coming to and from that device to go through the firewall. To list more than one device, use the option more times, such as `--trust em1 --trust em2`. Do not use a comma-separated format such as `--trust em1, em2`.
- `--remove-service` - Do not allow services through the firewall.
- `incoming` - Replace with one or more of the following to allow the specified services through the firewall.
--ssh
--smtp
--http
--ftp

- **--port** - You can specify that ports be allowed through the firewall using the port:protocol format. For example, to allow IMAP access through your firewall, specify `imap:tcp`. Numeric ports can also be specified explicitly; for example, to allow UDP packets on port 1234 through, specify `1234:udp`. To specify multiple ports, separate them by commas.

- **--service** - This option provides a higher-level way to allow services through the firewall. Some services (like `cups`, `avahi`, and so on.) require multiple ports to be open or other special configuration in order for the service to work. You can specify each individual port with the **--port** option, or specify **--service** and open them all at once. Valid options are anything recognized by the `firewall-offline-cmd` program in the `firewalld` package. If the `firewalld` service is running, `firewall-cmd --get-services` provides a list of known service names.

- **--use-system-defaults** - Do not configure the firewall at all. This option instructs anaconda to do nothing and allows the system to rely on the defaults that were provided with the package or ostree. If this option is used with other options then all other options will be ignored.

### J.3.4. group

The **group** Kickstart command is optional. It creates a new user group on the system.

```
group --name=name [--gid=gid]
```

**Mandatory options**

- **--name** - Provides the name of the group.

**Optional options**

- **--gid** - The group’s GID. If not provided, defaults to the next available non-system GID.

**Notes**

- If a group with the given name or GID already exists, this command fails.
- The **user** command can be used to create a new group for the newly created user.

### J.3.5. keyboard (required)

The **keyboard** Kickstart command is required. It sets one or more available keyboard layouts for the system.

**Syntax**

```
keyboard --vckeymap|--xlayouts OPTIONS
```
Options

- **--vckeymap=** - Specify a VConsole keymap which should be used. Valid names correspond to the list of files in the `/usr/lib/kbd/keymaps/xkb/` directory, without the `.map.gz` extension.

- **--xlayouts=** - Specify a list of X layouts that should be used as a comma-separated list without spaces. Accepts values in the same format as `setxkbmap(1)`, either in the `layout` format (such as `cz`), or in the `layout (variant)` format (such as `cz (qwerty)`). All available layouts can be viewed on the `xkeyboard-config(7)` man page under **Layouts**.

- **--switch=** - Specify a list of layout-switching options (shortcuts for switching between multiple keyboard layouts). Multiple options must be separated by commas without spaces. Accepts values in the same format as `setxkbmap(1)`. Available switching options can be viewed on the `xkeyboard-config(7)` man page under **Options**.

Notes

- Either the **--vckeymap** or the **--xlayouts** option must be used.

Example

The following example sets up two keyboard layouts (**English (US)** and **Czech (qwerty)**) using the **--xlayouts** option, and allows to switch between them using **Alt+Shift**:

```
keyboard --xlayouts=us,'cz (qwerty)' --switch=grp:alt_shift_toggle
```

J.3.6. lang (required)

The **lang** Kickstart command is required. It sets the language to use during installation and the default language to use on the installed system.

Syntax

```
lang language [--addsupport=language,...]
```

Mandatory options

- **language** - Install support for this language and set it as system default.

Optional options

- **--addsupport=** - Add support for additional languages. Takes the form of comma-separated list without spaces. For example:

```
lang en_US --addsupport=cs_CZ,de_DE,en_UK
```

Notes

- The **locale -a | grep _** or **locale --list-locales | grep _** commands return a list of supported locales.

- Certain languages (for example, Chinese, Japanese, Korean, and Indic languages) are not supported during text-mode installation. If you specify one of these languages with the **lang**
command, the installation process continues in English, but the installed system uses your selection as its default language.

Example
To set the language to English, the Kickstart file should contain the following line:

```
lang en_US
```

J.3.7. module
The module Kickstart command is optional. Use this command to enable a package module stream within kickstart script.

Syntax
```
module --name=NAME [--stream=STREAM]
```

Mandatory options
--name=
Specifies the name of the module to enable. Replace NAME with the actual name.

Optional options
--stream=
Specifies the name of the module stream to enable. Replace STREAM with the actual name. You do not need to specify this option for modules with a default stream defined. For modules without a default stream, this option is mandatory and leaving it out results in an error. Enabling a module multiple times with different streams is not possible.

Notes
- Using a combination of this command and the %packages section allows you to install packages provided by the enabled module and stream combination, without specifying the module and stream explicitly. Modules must be enabled before package installation. After enabling a module with the module command, you can install the packages enabled by this module by listing them in the %packages section.
- A single module command can enable only a single module and stream combination. To enable multiple modules, use multiple module commands. Enabling a module multiple times with different streams is not possible.
- In Red Hat Enterprise Linux 8, modules are present only in the AppStream repository. To list available modules, use the yum module list command on an installed Red Hat Enterprise Linux 8 system with a valid subscription.

Additional resources
- Installing, managing, and removing user-space components

J.3.8. repo
The repo Kickstart command is optional. It configures additional yum repositories that can be used as sources for package installation. You can add multiple repo lines.

Syntax

```
repo --name=repoid [--baseurl=url|--mirrorlist=url|--metalink=url] [OPTIONS]
```

Mandatory options

- `--name=` - The repository id. This option is required. If a repository has a name which conflicts with another previously added repository, it is ignored. Because the installation program uses a list of preset repositories, this means that you cannot add repositories with the same names as the preset ones.

URL options

These options are mutually exclusive and optional. The variables that can be used in yum repository configuration files are not supported here. You can use the strings `$releasever` and `$basearch` which are replaced by the respective values in the URL.

- `--baseurl=` - The URL to the repository.
- `--mirrorlist=` - The URL pointing at a list of mirrors for the repository.
- `--metalink=` - The URL with metalink for the repository.

Optional options

- `--install` - Save the provided repository configuration on the installed system in the `/etc/yum.repos.d/` directory. Without using this option, a repository configured in a Kickstart file will only be available during the installation process, not on the installed system.

- `--cost=` - An integer value to assign a cost to this repository. If multiple repositories provide the same packages, this number is used to prioritize which repository will be used before another. Repositories with a lower cost take priority over repositories with higher cost.

- `--excludepkgs=` - A comma-separated list of package names that must not be pulled from this repository. This is useful if multiple repositories provide the same package and you want to make sure it comes from a particular repository. Both full package names (such as `publican`) and globs (such as `gnome-*`) are accepted.

- `--includepkgs=` - A comma-separated list of package names and globs that are allowed to be pulled from this repository. Any other packages provided by the repository will be ignored. This is useful if you want to install just a single package or set of packages from a repository while excluding all other packages the repository provides.

- `--proxy=[protocol://][username[:password]@]host[:port]` - Specify an HTTP/HTTPS/FTP proxy to use just for this repository. This setting does not affect any other repositories, nor how the `install.img` is fetched on HTTP installations.

- `--noverifyssl` - Disable SSL verification when connecting to an HTTPS server.

Notes

- Repositories used for installation must be stable. The installation can fail if a repository is modified before the installation concludes.
J.3.9. rootpw (required)

The rootpw Kickstart command is required. It sets the system’s root password to the `password` argument.

Syntax

```
rootpw [-iscrypted|--plaintext] [--lock] password
```

Mandatory options

- `password` - Password specification. Either plain text or encrypted string. See `--iscrypted` and `--plaintext` below.

Options

- `--iscrypted` - If this option is present, the password argument is assumed to already be encrypted. This option is mutually exclusive with `--plaintext`. To create an encrypted password, you can use python:

  ```
  python -c 'import crypt, getpass; pw = getpass.getpass(); print(crypt.crypt(pw) if pw == getpass.getpass("Confirm: ") else exit())'
  ```

  This generates a sha512 crypt-compatible hash of your password using a random salt.

- `--plaintext` - If this option is present, the password argument is assumed to be in plain text. This option is mutually exclusive with `--iscrypted`.

- `--lock` - If this option is present, the root account is locked by default. This means that the root user will not be able to log in from the console. This option will also disable the Root Password screens in both the graphical and text-based manual installation.

J.3.10. selinux

The selinux Kickstart command is optional. It sets the state of SELinux on the installed system. The default SELinux policy is enforcing.

Syntax

```
selinux [--disabled|--enforcing|--permissive]
```

Options

- `--enforcing` - Enables SELinux with the default targeted policy being enforcing.
- `--permissive` - Outputs warnings based on the SELinux policy, but does not actually enforce the policy.
- `--disabled` - Disables SELinux completely on the system.

Additional resources
Using SElinux

J.3.11. services

The `services` Kickstart command is optional. It modifies the default set of services that will run under the default systemd target. The list of disabled services is processed before the list of enabled services. Therefore, if a service appears on both lists, it will be enabled.

Syntax

```
services [--disabled=list] [--enabled=list]
```

Options

- `--disabled=` - Disable the services given in the comma separated list.
- `--enabled=` - Enable the services given in the comma separated list.

Notes

- Do not include spaces in the list of services. If you do, Kickstart will enable or disable only the services up to the first space. For example:

  ```
  services --disabled=auditd,cups,smartd,nfslock
  ```

  That disables only the `auditd` service. To disable all four services, this entry must include no spaces:

  ```
  services --disabled=auditd,cups,smartd,nfslock
  ```

J.3.12. skipx

The `skipx` Kickstart command is optional. If present, X is not configured on the installed system.

If you install a display manager among your package selection options, this package creates an X configuration, and the installed system defaults to `graphical.target`. That overrides the effect of the `skipx` option.

Syntax

```
skipx
```

Notes

- This command has no options.

J.3.13. sshkey

The `sshkey` Kickstart command is optional. It adds a SSH key to the `authorized_keys` file of the specified user on the installed system.

Syntax
sshkey --username="user "ssh_key"

Mandatory options

- **--username=** - The user for which the key will be installed.
- **ssh_key** - The complete SSH key fingerprint. It must be wrapped with quotes.

### J.3.14. syspurpose

The *syspurpose* Kickstart command is optional. Use it to set the system purpose which describes how the system will be used after installation. This information helps apply the correct subscription entitlement to the system.

#### Syntax

```
syspurpose [OPTIONS]
```

#### Options

- **--role=** - Set the intended system role. Available values are:
  - Red Hat Enterprise Linux Server
  - Red Hat Enterprise Linux Workstation
  - Red Hat Enterprise Linux Compute Node
- **--sla=** - Set the Service Level Agreement. Available values are:
  - Premium
  - Standard
  - Self-Support
- **--usage=** - The intended usage of the system. Available values are:
  - Production
  - Disaster Recovery
  - Development/Test
- **--addon=** - Specifies additional layered products or features. You can use this option multiple times.

#### Notes

- Enter the values with spaces and enclose them in double quotes:
  ```
syspurpose --role="Red Hat Enterprise Linux Server"
  ```
While it is strongly recommended that you configure System Purpose, it is an optional feature of the Red Hat Enterprise Linux installation program. If you want to enable System Purpose after the installation completes, you can do so using the \texttt{syspurpose} command-line tool.

\section*{J.3.15. timezone (required)}

The \texttt{timezone} Kickstart command is required. It sets the system time zone.

\textbf{Syntax}

\begin{verbatim}
timezone timezone [OPTIONS]
\end{verbatim}

\textbf{Mandatory options}

- \texttt{timezone} - the time zone to set for the system.

\textbf{Optional options}

- \texttt{--utc} - If present, the system assumes the hardware clock is set to UTC (Greenwich Mean) time.
- \texttt{--nontp} - Disable the NTP service automatic starting.
- \texttt{--ntpservers=} - Specify a list of NTP servers to be used as a comma-separated list without spaces.

\textbf{Notes}

In Red Hat Enterprise Linux 8, time zone names are validated using the \texttt{pytz.all_timezones} list, provided by the \texttt{pytz} package. In previous releases, the names were validated against \texttt{pytz.common_timezones}, which is a subset of the currently used list. Note that the graphical and text mode interfaces still use the more restricted \texttt{pytz.common_timezones} list; you must use a Kickstart file to use additional time zone definitions.

\section*{J.3.16. user}

The \texttt{user} Kickstart command is optional. It creates a new user on the system.

\textbf{Syntax}

\begin{verbatim}
user --name=username [OPTIONS]
\end{verbatim}

\textbf{Mandatory options}

- \texttt{--name=} - Provides the name of the user. This option is required.

\textbf{Optional options}

- \texttt{--gecos=} - Provides the GECOS information for the user. This is a string of various system-specific fields separated by a comma. It is frequently used to specify the user’s full name, office number, and so on. See the \texttt{passwd(5)} man page for more details.
- \texttt{--groups=} - In addition to the default group, a comma separated list of group names the user should belong to. The groups must exist before the user account is created. See the \texttt{group} command.
- `--homdir=` - The home directory for the user. If not provided, this defaults to `/home/username`.

- `--lock` - If this option is present, this account is locked by default. This means that the user will not be able to log in from the console. This option will also disable the Create User screens in both the graphical and text-based manual installation.

- `--password=` - The new user’s password. If not provided, the account will be locked by default.

- `--iscrypted` - If this option is present, the password argument is assumed to already be encrypted. This option is mutually exclusive with `--plaintext`. To create an encrypted password, you can use python:

  ```
  $ python -c 'import crypt, getpass;pw=getpass.getpass();print(crypt.crypt(pw) if pw==getpass.getpass("Confirm: ") else exit())'
  ```

  This generates a sha512 crypt-compatible hash of your password using a random salt.

- `--plaintext` - If this option is present, the password argument is assumed to be in plain text. This option is mutually exclusive with `--iscrypted`

- `--shell=` - The user’s login shell. If not provided, the system default is used.

- `--uid=` - The user’s UID (User ID). If not provided, this defaults to the next available non-system UID.

- `--gid=` - The GID (Group ID) to be used for the user’s group. If not provided, this defaults to the next available non-system group ID.

**Notes**

- Consider using the `--uid` and `--gid` options to set IDs of regular users and their default groups at range starting at 5000 instead of 1000. That is because the range reserved for system users and groups, 0-999, might increase in the future and thus overlap with IDs of regular users. For changing the minimum UID and GID limits after the installation, which ensures that your chosen UID and GID ranges are applied automatically on user creation, see the Setting default permissions for new files using umask section of the Configuring basic system settings document.

- Files and directories are created with various permissions, dictated by the application used to create the file or directory. For example, the `mkdir` command creates directories with all permissions enabled. However, applications are prevented from granting certain permissions to newly created files, as specified by the user file-creation mask setting.

  The user file-creation mask can be controlled with the `umask` command. The default setting of the user file-creation mask for new users is defined by the `UMASK` variable in the `/etc/login.defs` configuration file on the installed system. If unset, it defaults to `022`. This means that by default when an application creates a file, it is prevented from granting write permission to users other than the owner of the file. However, this can be overridden by other settings or scripts. More information can be found in the Setting default permissions for new files using umask section of the Configuring basic system settings document.

### J.3.17. xconfig

The `xconfig` Kickstart command is optional. It configures the X Window System.

**Syntax**
xconfig [--startxonboot]

Options

- **--startxonboot** - Use a graphical login on the installed system.

Notes

- Because Red Hat Enterprise Linux 8 does not include the KDE Desktop Environment, do not use the **--defaultdesktop** documented in upstream.

### J.4. KICKSTART COMMANDS FOR NETWORK CONFIGURATION

The Kickstart commands in this list let you configure networking on the system.

#### J.4.1. network (optional)

Use the optional **network** Kickstart command to configure network information for the target system and activate the network devices in the installation environment. The device specified in the first **network** command is activated automatically. You can also explicitly require a device to be activated using the **--activate** option.

**Syntax**

```
network OPTIONS
```

**Options**

- **--activate** - activate this device in the installation environment.
  
  If you use the **--activate** option on a device that has already been activated (for example, an interface you configured with boot options so that the system could retrieve the Kickstart file) the device is reactivated to use the details specified in the Kickstart file.

  Use the **--nodefroute** option to prevent the device from using the default route.

- **--no-activate** - do not activate this device in the installation environment.
  
  By default, Anaconda activates the first network device in the Kickstart file regardless of the **--activate** option. You can disable the default setting by using the **--no-activate** option.

- **--bootproto** - One of **dhcp**, **bootp**, **ibft**, or **static**. The default option is **dhcp**; the **dhcp** and **bootp** options are treated the same. To disable **ipv4** configuration of the device, use the **--noipv4** option.

  **NOTE**

  This option configures ipv4 configuration of the device. For ipv6 configuration use **--ipv6** and **--ipv6gateway** options.

  The DHCP method uses a DHCP server system to obtain its networking configuration. The BOOTP method is similar, requiring a BOOTP server to supply the networking configuration. To direct a system to use DHCP:
network --bootproto=dhcp

To direct a machine to use BOOTP to obtain its networking configuration, use the following line in the Kickstart file:

network --bootproto=bootp

To direct a machine to use the configuration specified in iBFT, use:

network --bootproto=ibft

The static method requires that you specify at least the IP address and netmask in the Kickstart file. This information is static and is used during and after the installation.

All static networking configuration information must be specified on one line; you cannot wrap lines using a backslash (\) as you can on a command line.

network --bootproto=static --ip=10.0.2.15 --netmask=255.255.255.0 --gateway=10.0.2.254 --nameserver=10.0.2.1

You can also configure multiple nameservers at the same time. To do so, use the --nameserver= option once, and specify each of their IP addresses, separated by commas:

network --bootproto=static --ip=10.0.2.15 --netmask=255.255.255.0 --gateway=10.0.2.254 --nameserver=192.168.2.1,192.168.3.1

- **--device=** - specifies the device to be configured (and eventually activated in Anaconda) with the network command.
  If the --device= option is missing on the first use of the network command, the value of the inst.ks.device= Anaconda boot option is used, if available. Note that this is considered deprecated behavior; in most cases, you should always specify a --device= for every network command.
The behavior of any subsequent `network` command in the same Kickstart file is unspecified if its `--device=` option is missing. Make sure you specify this option for any `network` command beyond the first.

You can specify a device to be activated in any of the following ways:

- the device name of the interface, for example, `em1`
- the MAC address of the interface, for example, `01:23:45:67:89:ab`
- the keyword `link`, which specifies the first interface with its link in the **up** state
- the keyword `bootif`, which uses the MAC address that pxelinux set in the `BOOTIF` variable. Set `IPAPPEND 2` in your `pxelinux.cfg` file to have pxelinux set the `BOOTIF` variable.

For example:

```bash
network --bootproto=dhcp --device=em1
```

- `--ip=` - IP address of the device.
- `--ipv6=` - IPv6 address of the device, in the form of `address[/prefix length]` - for example, `3ffe:ffff:0:1::1/128`. If `prefix` is omitted, `64` is used. You can also use `auto` for automatic configuration, or `dhcp` for DHCPv6-only configuration (no router advertisements).
- `--gateway=` - Default gateway as a single IPv4 address.
- `--ipv6gateway=` - Default gateway as a single IPv6 address.
- `--nodefroute` - Prevents the interface being set as the default route. Use this option when you activate additional devices with the `--activate=` option, for example, a NIC on a separate subnet for an iSCSI target.
- `--nameserver=` - DNS name server, as an IP address. To specify more than one name server, use this option once, and separate each IP address with a comma.
- `--netmask=` - Network mask for the installed system.
- `--hostname=` - Used to configure the target system’s host name. The host name can either be a fully qualified domain name (FQDN) in the format `hostname.domainname`, or a short host name without the domain. If you use a Dynamic Host Configuration Protocol (DHCP) service to automatically assign a domain name to connected systems, only specify the short host name. If you only want to configure the target system’s host name, use the `--hostname` option in the `network` command and do not include any other option.

If you provide additional options when configuring the host name, the `network` command configures a device using the options specified. If you do not specify which device to configure using the `--device` option, the default `--device link` value is used. Additionally, if you do not specify the protocol using the `--bootproto` option, the device is configured to use DHCP by default.
IMPORTANT

If your network does not provide a DHCP service, always use the FQDN as the system’s host name.

- **--ethtool=** - Specifies additional low-level settings for the network device which will be passed to the ethtool program.
- **--onboot=** - Whether or not to enable the device at boot time.
- **--dhcpclass=** - The DHCP class.
- **--mtu=** - The MTU of the device.
- **--noipv4** - Disable IPv4 on this device.
- **--noipv6** - Disable IPv6 on this device.
- **--bondslaves=** - When this option is used, the bond device specified by the **--device=** option is created using secondary devices defined in the **--bondslaves=** option. For example:

  network --device=bond0 --bondslaves=em1,em2

  The above command creates a bond device named **bond0** using the **em1** and **em2** interfaces as its secondary devices.

- **--bondopts=** - a list of optional parameters for a bonded interface, which is specified using the **- bondslaves=** and **--device=** options. Options in this list must be separated by commas ("","") or semicolons (";"). If an option itself contains a comma, use a semicolon to separate the options. For example:

  network --bondopts=mode=active-backup,balance-rr;primary=eth1

  IMPORTANT

  The **--bondopts=mode** parameter only supports full mode names such as **balance-rr** or **broadcast**, not their numerical representations such as 0 or 3. For the list of available and supported modes, see Configuring and Managing Networking Guide.

- **--vlanid=** - Specifies virtual LAN (VLAN) ID number (802.1q tag) for the device created using the device specified in **--device=** as a parent. For example, **network --device=em1 -- vlanid=171** creates a virtual LAN device **em1.171**.

- **--interfacenames=** - Specify a custom interface name for a virtual LAN device. This option should be used when the default name generated by the **--vlanid=** option is not desirable. This option must be used along with **--vlanid=**. For example:

  network --device=em1 --vlanid=171 --interfacename=vlan171

  The above command creates a virtual LAN interface named **vlan171** on the **em1** device with an ID of 171.

  The interface name can be arbitrary (for example, **my-vlan**), but in specific cases, the following conventions must be followed:
If the name contains a dot (.), it must take the form of NAME.ID. The NAME is arbitrary, but the ID must be the VLAN ID. For example: em1.171 or my-vlan.171.

- Names starting with vlan must take the form of vlanID - for example, vlan171.

- `--teamslaves=` - Team device specified by the `--device=` option will be created using secondary devices specified in this option. Secondary devices are separated by commas. A secondary device can be followed by its configuration, which is a single-quoted JSON string with double quotes escaped by the \ character. For example:

  ```
  network --teamslaves="p3p1{"prio": -10, \"sticky\": true},p3p2{"prio": 100}" 
  ```

  See also the `--teamconfig=` option.

- `--teamconfig=` - Double-quoted team device configuration which is a JSON string with double quotes escaped by the \ character. The device name is specified by `--device=` option and its secondary devices and their configuration by `--teamslaves=` option. For example:

  ```
  network --device team0 --activate --bootproto static --ip=10.34.102.222 --netmask=255.255.255.0 --gateway=10.34.102.254 --nameserver=10.34.39.2 --teamslaves="p3p1{"prio": -10, \"sticky\": true},p3p2{"prio": 100}" --teamconfig="{"runner": {\"name\": \"activebackup\"}}"
  ```

- `--bridgeslaves=` - When this option is used, the network bridge with device name specified using the `--device=` option will be created and devices defined in the `--bridgeslaves=` option will be added to the bridge. For example:

  ```
  network --device=bridge0 --bridgeslaves=em1
  ```

- `--bridgeopts=` - An optional comma-separated list of parameters for the bridged interface. Available values are stp, priority, forward-delay, hello-time, max-age, and ageing-time. For information about these parameters, see the bridge setting table in the nm-settings(5) man page or at Network Configuration Setting Specification. Also see the Configuring and managing networking document for general information about network bridging.

- `--bindto=mac` - Bind the device configuration (ifcfg) file on the installed system to the device MAC address (HWADDR) instead of the default binding to the interface name (DEVICE). Note that this option is independent of the `--device=` option - `--bindto=mac` will be applied even if the same network command also specifies a device name, link, or bootif.

**Notes**

- The ethN device names such as eth0 are no longer available in Red Hat Enterprise Linux 8 due to changes in the naming scheme. For more information about the device naming scheme, see the upstream document Predictable Network Interface Names.

- If you used a Kickstart option or a boot option to specify an installation repository on a network, but no network is available at the start of the installation, the installation program displays the Network Configuration window to set up a network connection prior to displaying the Installation Summary window. For more details, see the Configuring network and host name options section of the Performing a standard RHEL installation document.

**J.4.2. realm**
The `realm` Kickstart command is optional. Use it to join an Active Directory or IPA domain. For more information about this command, see the `join` section of the `realm(8)` man page.

Syntax

```
realm join [OPTIONS] domain
```

Mandatory options

- `domain` - The domain to join.

Options

- `--computer-ou=OU=` - Provide the distinguished name of an organizational unit in order to create the computer account. The exact format of the distinguished name depends on the client software and membership software. The root DSE portion of the distinguished name can usually be left out.
- `--no-password` - Join automatically without a password.
- `--one-time-password=` - Join using a one-time password. This is not possible with all types of realm.
- `--client-software=` - Only join realms which can run this client software. Valid values include `sssd` and `winbind`. Not all realms support all values. By default, the client software is chosen automatically.
- `--server-software=` - Only join realms which can run this server software. Possible values include `active-directory` or `freeipa`.
- `--membership-software=` - Use this software when joining the realm. Valid values include `samba` and `adcli`. Not all realms support all values. By default, the membership software is chosen automatically.

### J.5. KICKSTART COMMANDS FOR HANDLING STORAGE

The Kickstart commands in this section configure aspects of storage such as devices, disks, partitions, LVM, and filesystems.

#### J.5.1. device (deprecated)

The `device` Kickstart command is optional. Use it to load additional kernel modules.

On most PCI systems, the installation program automatically detects Ethernet and SCSI cards. However, on older systems and some PCI systems, Kickstart requires a hint to find the proper devices. The `device` command, which tells the installation program to install extra modules, uses the following format:

Syntax

```
device moduleName --opts=options
```

Options
| moduleName | Replace with the name of the kernel module which should be installed. |
| --opts= | Options to pass to the kernel module. For example: |
| device --opts="aic152x=0x340 io=11" |

J.5.2. autopart

The autopart Kickstart command is optional. It automatically creates partitions.

The automatically created partitions are: a root (/) partition (1 GiB or larger), a swap partition, and an appropriate /boot partition for the architecture. On large enough drives (50 GiB and larger), this also creates a /home partition.

Syntax

autopart OPTIONS

Options

- **--type=** - Selects one of the predefined automatic partitioning schemes you want to use. Accepts the following values:
  - lvm: The LVM partitioning scheme.
  - plain: Regular partitions with no LVM.
  - thinp: The LVM Thin Provisioning partitioning scheme.
- **--fstype=** - Selects one of the available file system types. The available values are ext2, ext3, ext4, xfs, and vfat. The default file system is xfs.
- **--nohome** - Disables automatic creation of the /home partition.
- **--nolvm** - Do not use LVM for automatic partitioning. This option is equal to --type=plain.
- **--noboot** - Do not create a /boot partition.
- **--noswap** - Do not create a swap partition.
- **--encrypted** - Encrypts all partitions with Linux Unified Key Setup (LUKS). This is equivalent to checking the Encrypt partitions check box on the initial partitioning screen during a manual graphical installation.

**NOTE**

When encrypting one or more partitions, Anaconda attempts to gather 256 bits of entropy to ensure the partitions are encrypted securely. Gathering entropy can take some time - the process will stop after a maximum of 10 minutes, regardless of whether sufficient entropy has been gathered.

The process can be sped up by interacting with the installation system (typing on the keyboard or moving the mouse). If you are installing in a virtual machine, you can also attach a virtio-rng device (a virtual random number generator) to the guest.
- **--luks-version=LUKS_VERSION** - Specifies which version of LUKS format should be used to encrypt the filesystem. This option is only meaningful if **--encrypted** is specified.

- **--passphrase=** - Provides a default system-wide passphrase for all encrypted devices.

- **--escrowcert=URL_of_X.509_certificate** - Stores data encryption keys of all encrypted volumes as files in /root, encrypted using the X.509 certificate from the URL specified with **URL_of_X.509_certificate**. The keys are stored as a separate file for each encrypted volume. This option is only meaningful if **--encrypted** is specified.

- **--backuppassphrase** - Adds a randomly-generated passphrase to each encrypted volume. Store these passphrases in separate files in /root, encrypted using the X.509 certificate specified with **--escrowcert**. This option is only meaningful if **--escrowcert** is specified.

- **--cipher=** - Specifies the type of encryption to use if the Anaconda default aes-xts-plain64 is not satisfactory. You must use this option together with the **--encrypted** option; by itself it has no effect. Available types of encryption are listed in the Security hardening document, but Red Hat strongly recommends using either aes-xts-plain64 or aes-cbc-essiv:sha256.

- **--pbkdf=PBKDF** - Sets Password-Based Key Derivation Function (PBKDF) algorithm for LUKS keystore. See also the man page cryptsetup(8). This option is only meaningful if **--encrypted** is specified.

- **--pbkdf-memory=PBKDF_MEMORY** - Sets the memory cost for PBKDF. See also the man page cryptsetup(8). This option is only meaningful if **--encrypted** is specified.

- **--pbkdf-time=PBKDF_TIME** - Sets the number of milliseconds to spend with PBKDF passphrase processing. See also **--iter-time** in the man page cryptsetup(8). This option is only meaningful if **--encrypted** is specified, and is mutually exclusive with **--pbkdf-iterations**.

- **--pbkdf-iterations=PBKDF_ITERATIONS** - Sets the number of iterations directly and avoids PBKDF benchmark. See also **--pbkdf-force-iterations** in the man page cryptsetup(8). This option is only meaningful if **--encrypted** is specified, and is mutually exclusive with **--pbkdf-time**.

**Notes**

- The **autopart** option cannot be used together with the **part/partition, raid, logvol, or volgroup** options in the same Kickstart file.

- The **autopart** command is not mandatory, but you must include it if there are no **part or mount** commands in your Kickstart script.

- It is recommended to use the **autopart --nohome** Kickstart option when installing on a single FBA DASD of the CMS type. This ensures that the installation program does not create a separate /home partition. The installation then proceeds successfully.

- If you lose the LUKS passphrase, any encrypted partitions and their data is completely inaccessible. There is no way to recover a lost passphrase. However, you can save encryption passphrases with the **--escrowcert** and create backup encryption passphrases with the **--backuppassphrase** options.

- Ensure that the disk sector sizes are consistent when using **autopart, autopart --type=lvm, or autopart=thinp**.

**J.5.3. bootloader (required)**
The **bootloader** Kickstart command is required. It specifies how the boot loader should be installed.

### Syntax

```
bootloader [OPTIONS]
```

### Options

- **--append=** - Specifies additional kernel parameters. To specify multiple parameters, separate them with spaces. For example:

```
bootloader --location=mbr --append="hdd=ide-scsi ide=nodma"
```

The *rhgb* and *quiet* parameters are automatically added when the *plymouth* package is installed, even if you do not specify them here or do not use the **--append=** command at all. To disable this behavior, explicitly disallow installation of *plymouth*:

```
%packages
-plymouth
%end
```

This option is useful for disabling mechanisms which were implemented to mitigate the Meltdown and Spectre speculative execution vulnerabilities found in most modern processors (CVE-2017-5754, CVE-2017-5753, and CVE-2017-5715). In some cases, these mechanisms may be unnecessary, and keeping them enabled causes decreased performance with no improvement in security. To disable these mechanisms, add the options to do so into your Kickstart file - for example, `bootloader --append="nopti noibrs noibpb"` on AMD64/Intel 64 systems.

---

**WARNING**

Ensure your system is not at risk of attack before disabling any of the vulnerability mitigation mechanisms. See the Red Hat vulnerability response article for information about the Meltdown and Spectre vulnerabilities.

---

- **--boot-drive=** - Specifies which drive the boot loader should be written to, and therefore which drive the computer will boot from. If you use a multipath device as the boot drive, specify the device using its disk/by-id/dm-uuid-mpath-WWID name.

---

**IMPORTANT**

The **--boot-drive=** option is currently being ignored in Red Hat Enterprise Linux installations on 64-bit IBM Z systems using the *zipl* boot loader. When *zipl* is installed, it determines the boot drive on its own.
--leavebootorder - The installation program will add Red Hat Enterprise Linux 8 to the top of the list of installed systems in the boot loader, and preserve all existing entries as well as their order.

**IMPORTANT**
This option is applicable for Power systems only and UEFI systems should not use this option.

--driveorder= - Specifies which drive is first in the BIOS boot order. For example:

```bash
bootloader --driveorder=sda,hda
```

--location= - Specifies where the boot record is written. Valid values are the following:

- **mbr** - The default option. Depends on whether the drive uses the Master Boot Record (MBR) or GUID Partition Table (GPT) scheme:
  - On a GPT-formatted disk, this option installs stage 1.5 of the boot loader into the BIOS boot partition.
  - On an MBR-formatted disk, stage 1.5 is installed into the empty space between the MBR and the first partition.

- **partition** - Install the boot loader on the first sector of the partition containing the kernel.

- **none** - Do not install the boot loader.

In most cases, this option does not need to be specified.

--nombr - Do not install the boot loader to the MBR.

--password= - If using GRUB2, sets the boot loader password to the one specified with this option. This should be used to restrict access to the GRUB2 shell, where arbitrary kernel options can be passed.

If a password is specified, GRUB2 also asks for a user name. The user name is always **root**.

--iscrypted - Normally, when you specify a boot loader password using the `--password=` option, it is stored in the Kickstart file in plain text. If you want to encrypt the password, use this option and an encrypted password.

To generate an encrypted password, use the `grub2-mkpasswd-pbkdf2` command, enter the password you want to use, and copy the command’s output (the hash starting with `grub.pbkdf2`) into the Kickstart file. An example **bootloader** Kickstart entry with an encrypted password looks similar to the following:

```bash
bootloader --iscrypted --password=grub.pbkdf2.sha512.1000.5520C6C9832F3AC3D149AC0B24BE69E2D4FB0DBEEDBD29CA1D30A044DE2645C4C7A291E585D4DC43F8A4D82479F8B95CA4BA4381F8550510B75E8E0BB2938990.C688B6F0EF935701FF9BD1A8EC7FE5BD233799C98F28420C5CC8F1A2A233DE22C83705BB614EA17F3FDFDF4AC2161CEA3384E56EB38A2E39102F5334C47405E
```

--timeout= - Specifies the amount of time the boot loader waits before booting the default option (in seconds).

--default= - Sets the default boot image in the boot loader configuration.
- **--extlinux** - Use the extlinux boot loader instead of GRUB2. This option only works on systems supported by extlinux.

- **--disabled** - This option is a stronger version of **--location=none**. While **--location=none** simply disables boot loader installation, **--disabled** disables boot loader installation and also disables installation of the package containing the boot loader, thus saving space.

**Notes**

- Red Hat recommends setting up a boot loader password on every system. An unprotected boot loader can allow a potential attacker to modify the system’s boot options and gain unauthorized access to the system.

- In some cases, a special partition is required to install the boot loader on AMD64, Intel 64, and 64-bit ARM systems. The type and size of this partition depends on whether the disk you are installing the boot loader to uses the Master Boot Record (MBR) or a GUID Partition Table (GPT) schema. For more information, see the Configuring boot loader section of the Performing a standard RHEL installation document.

- Device names in the $sdX$ (or $/dev/sdX$) format are not guaranteed to be consistent across reboots, which can complicate usage of some Kickstart commands. When a command calls for a device node name, you can instead use any item from $/dev/disk$. For example, instead of:

  ```
  part / --fstype=xfs --onpart=sdta1
  ```

  You can use an entry similar to one of the following:

  ```
  part / --fstype=xfs --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0:part1
  
  part / --fstype=xfs --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
  ```

  This way the command will always target the same storage device. This is especially useful in large storage environments. See the chapter Overview of persistent naming attributes in the Managing storage devices document for more in-depth information about different ways to consistently refer to storage devices.

- The **--upgrade** option is deprecated in Red Hat Enterprise Linux 8.

**J.5.4. zipl**

The zipl Kickstart command is optional. It specifies the ZIPL configuration for 64-bit IBM Z.

**Options**

- **--secure-boot** - Enables secure boot if it is supported by the installing system.

  **NOTE**

  When installed on a system that is later than IBM z14, the installed system cannot be booted from an IBM z14 or earlier model.

- **--force-secure-boot** - Enables secure boot unconditionally.
NOTE
Installation is not supported on IBM z14 and earlier models.

• --no-secure-boot - Disables secure boot.

NOTE
Secure Boot is not supported on IBM z14 and earlier models. Use --no-secure-boot if you intend to boot the installed system on IBM z14 and earlier models.

J.5.5. clearpart
The clearpart Kickstart command is optional. It removes partitions from the system, prior to creation of new partitions. By default, no partitions are removed.

Syntax

```
| clearpart OPTIONS |
```

Options

• --all - Erases all partitions from the system. This option will erase all disks which can be reached by the installation program, including any attached network storage. Use this option with caution.

You can prevent clearpart from wiping storage you want to preserve by using the --drives= option and specifying only the drives you want to clear, by attaching network storage later (for example, in the %post section of the Kickstart file), or by blocklisting the kernel modules used to access network storage.

• --drives= - Specifies which drives to clear partitions from. For example, the following clears all the partitions on the first two drives on the primary IDE controller:

```
clearpart --drives=hda,hdb --all
```

To clear a multipath device, use the format disk/by-id/scsi-WWID, where WWID is the world-wide identifier for the device. For example, to clear a disk with WWID 58095BEC5510947BE8C0360F604351918, use:

```
clearpart --drives=disk/by-id/scsi-58095BEC5510947BE8C0360F604351918
```

This format is preferable for all multipath devices, but if errors arise, multipath devices that do not use logical volume management (LVM) can also be cleared using the format disk/by-id/dm-uuid-mpath-WWID, where WWID is the world-wide identifier for the device. For example, to clear a disk with WWID 2416CD96995134CA5D787F00A5AA11017, use:

```
clearpart --drives=disk/by-id/dm-uuid-mpath-2416CD96995134CA5D787F00A5AA11017
```

Never specify multipath devices by device names like mpatha. Device names such as this are not specific to a particular disk. The disk named /dev/mpatha during installation might not be the one that you expect it to be. Therefore, the clearpart command could target the wrong disk.
• **--initlabel** - Initializes a disk (or disks) by creating a default disk label for all disks in their respective architecture that have been designated for formatting (for example, msdos for x86). Because **--initlabel** can see all disks, it is important to ensure only those drives that are to be formatted are connected.

```
clearpart --initlabel --drives=names_of_disks
```

For example:

```
clearpart --initlabel --drives=dasda,dasdb,dasdc
```

• **--list=** - Specifies which partitions to clear. This option overrides the **--all** and **--linux** options if used. Can be used across different drives. For example:

```
clearpart --list=sda2,sda3,sdb1
```

• **--disklabel=LABEL** - Set the default disklabel to use. Only disklabels supported for the platform will be accepted. For example, on the 64-bit Intel and AMD architectures, the **msdos** and **gpt** disklabels are accepted, but **dasd** is not accepted.

• **--linux** - Erases all Linux partitions.

• **--none** (default) - Do not remove any partitions.

• **--cdl** - Reformat any LDL DASDs to CDL format.

**Notes**

• Device names in the **sdX** (or **/dev/sdX**) format are not guaranteed to be consistent across reboots, which can complicate usage of some Kickstart commands. When a command calls for a device node name, you can instead use any item from **/dev/disk**. For example, instead of:

```
part / --fstype=xfs --onpart=sda1
```

You could use an entry similar to one of the following:

```
part / --fstype=xfs --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-part1
```

```
part / --fstype=xfs --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
```

This way the command will always target the same storage device. This is especially useful in large storage environments. See the chapter [Overview of persistent naming attributes](#) in the *Managing storage devices* document for more in-depth information about different ways to consistently refer to storage devices.

• If the **clearpart** command is used, then the **part --onpart** command cannot be used on a logical partition.

### J.5.6. fcoe

The **fcoe** Kickstart command is optional. It specifies which FCoE devices should be activated automatically in addition to those discovered by Enhanced Disk Drive Services (EDD).
Syntax

```
fcoe --nic=name [OPTIONS]
```

**Options**

- **--nic=** (required) - The name of the device to be activated.
- **--dcb=** - Establish Data Center Bridging (DCB) settings.
- **--autovlan** - Discover VLANs automatically. This option is enabled by default.

### J.5.7. ignoredisk

The `ignoredisk` Kickstart command is optional. It causes the installation program to ignore the specified disks.

This is useful if you use automatic partitioning and want to be sure that some disks are ignored. For example, without `ignoredisk`, attempting to deploy on a SAN-cluster the Kickstart would fail, as the installation program detects passive paths to the SAN that return no partition table.

**Syntax**

```
ignoredisk --drives=drive1,drive2,... | --only-use=drive
```

**Options**

- **--drives=driveN,...** - Replace `driveN` with one of `sda, sdb,..., hda,...` and so on.
- **--only-use=driveN,...** - Specifies a list of disks for the installation program to use. All other disks are ignored. For example, to use disk `sda` during installation and ignore all other disks:

  ```
  ignoredisk --only-use=sda
  ```

To include a multipath device that does not use LVM:

  ```
  ignoredisk --only-use=disk/by-id/dm-uuid-mpath-2416CD96995134CA5D787F00A5AA11017
  ```

To include a multipath device that uses LVM:

  ```
  ignoredisk --only-use=/dev/disk/by-id/dm-uuid-mpath-
  ```

  ```
  bootloader --location=mbr
  ```

You must specify only one of the **--drives** or **--only-use**.

**Notes**

- The **--interactive** option is deprecated in Red Hat Enterprise Linux 8. This option allowed users to manually navigate the advanced storage screen.
- To ignore a multipath device that does not use logical volume management (LVM), use the
format disk/by-id/dm-uuid-mpath-WWID, where WWID is the world-wide identifier for the device. For example, to ignore a disk with WWID 2416CD96995134CA5D787F00A5AA11017, use:

```
ignoredisk --drives=disk/by-id/dm-uuid-mpath-2416CD96995134CA5D787F00A5AA11017
```

- Never specify multipath devices by device names like mpatha. Device names such as this are not specific to a particular disk. The disk named /dev/mpatha during installation might not be the one that you expect it to be. Therefore, the clearpart command could target the wrong disk.

- Device names in the sdX (or /dev/sdX) format are not guaranteed to be consistent across reboots, which can complicate usage of some Kickstart commands. When a command calls for a device node name, you can instead use any item from /dev/disk. For example, instead of:

```
part / --fstype=xfs --onpart=sda1
```

You can use an entry similar to one of the following:

```
part / --fstype=xfs --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-part1
```

```
part / --fstype=xfs --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
```

This way the command will always target the same storage device. This is especially useful in large storage environments. See the chapter Overview of persistent naming attributes in the Managing storage devices document for more in-depth information about different ways to consistently refer to storage devices.

### J.5.8. iscsi

The iscsi Kickstart command is optional. It specifies additional iSCSI storage to be attached during installation.

**Syntax**

```
iscsi --ipaddr=address [OPTIONS]
```

**Mandatory options**

- `--ipaddr=` (required) - the IP address of the target to connect to.

**Optional options**

- `--port=` (required) - the port number. If not present, `--port=3260` is used automatically by default.

- `--target=` - the target IQN (iSCSI Qualified Name).

- `--iface=` - bind the connection to a specific network interface instead of using the default one determined by the network layer. Once used, it must be specified in all instances of the iscsi command in the entire Kickstart file.

- `--user=` - the user name required to authenticate with the target.
• `--password=` - the password that corresponds with the user name specified for the target

• `--reverse-user=` - the user name required to authenticate with the initiator from a target that uses reverse CHAP authentication

• `--reverse-password=` - the password that corresponds with the user name specified for the initiator

Notes

• If you use the `iscsi` command, you must also assign a name to the iSCSI node, using the `iscsiname` command. The `iscsiname` command must appear before the `iscsi` command in the Kickstart file.

• Wherever possible, configure iSCSI storage in the system BIOS or firmware (iBFT for Intel systems) rather than use the `iscsi` command. Anaconda automatically detects and uses disks configured in BIOS or firmware and no special configuration is necessary in the Kickstart file.

• If you must use the `iscsi` command, ensure that networking is activated at the beginning of the installation, and that the `iscsi` command appears in the Kickstart file before you refer to iSCSI disks with commands such as `clearpart` or `ignoredisk`.

J.5.9. iscsiname

The `iscsiname` Kickstart command is optional. It assigns a name to an iSCSI node specified by the `iscsi` command.

Syntax

```
iscsiname iqname
```

Options

• `iqname` - Name to assign to the iSCSI node.

Notes

• If you use the `iscsi` command in your Kickstart file, you must specify `iscsiname` earlier in the Kickstart file.

J.5.10. logvol

The `logvol` Kickstart command is optional. It creates a logical volume for Logical Volume Management (LVM).

Syntax

```
logvol mntpoint --vgname=name --name=name [OPTIONS]
```

Mandatory options

`mntpoint`

The mount point where the partition is mounted. Must be of one of the following forms:
- `/path`
  For example, `/` or `/home`

- `swap`
  The partition is used as swap space.

  To determine the size of the swap partition automatically, use the `--recommended` option:

  ```
  swap --recommended
  ```

  To determine the size of the swap partition automatically and also allow extra space for your system to hibernate, use the `--hibernation` option:

  ```
  swap --hibernation
  ```

  The size assigned will be equivalent to the swap space assigned by `--recommended` plus the amount of RAM on your system.

  For the swap sizes assigned by these commands, see Recommended Partitioning Scheme for AMD64, Intel 64, and 64-bit ARM systems.

  ```
  --vgname=name
  Name of the volume group.

  --name=name
  Name of the logical volume.
  ```

Optional options

- `--noformat`
  Use an existing logical volume and do not format it.

- `--useexisting`
  Use an existing logical volume and reformat it.

- `--fstype=`
  Sets the file system type for the logical volume. Valid values are `xfs`, `ext2`, `ext3`, `ext4`, `swap`, and `vfat`.

- `--fsoptions=`
  Specifies a free form string of options to be used when mounting the filesystem. This string will be copied into the `/etc/fstab` file of the installed system and should be enclosed in quotes.

  **NOTE**

  In the EFI system partition (`/boot/efi`), anaconda hard codes the value and ignores the users specified `--fsoptions` values.

- `--mkfsoptions=`
  Specifies additional parameters to be passed to the program that makes a filesystem on this partition. No processing is done on the list of arguments, so they must be supplied in a format that can be passed directly to the mkfs program. This means multiple options should be comma-separated or surrounded by double quotes, depending on the filesystem.

- `--fsprofile=`
Specifies a usage type to be passed to the program that makes a filesystem on this partition. A usage type defines a variety of tuning parameters to be used when making a filesystem. For this option to work, the filesystem must support the concept of usage types and there must be a configuration file that lists valid types. For ext2, ext3, and ext4, this configuration file is /etc/mke2fs.conf.

--label=
Sets a label for the logical volume.

--grow
Extends the logical volume to occupy the available space (if any), or up to the maximum size specified, if any. The option must be used only if you have pre-allocated a minimum storage space in the disk image, and would want the volume to grow and occupy the available space. In a physical environment, this is an one-time-action. However, in a virtual environment, the volume size increases as and when the virtual machine writes any data to the virtual disk.

--size=
The size of the logical volume in MiB. This option cannot be used together with the --percent= option.

--percent=
The size of the logical volume, as a percentage of the free space in the volume group after any statically-sized logical volumes are taken into account. This option cannot be used together with the --size= option.

**IMPORTANT**
When creating a new logical volume, you must either specify its size statically using the --size= option, or as a percentage of remaining free space using the --percent= option. You cannot use both of these options on the same logical volume.

--maxsize=
The maximum size in MiB when the logical volume is set to grow. Specify an integer value here such as 500 (do not include the unit).

--recommended
Use this option when creating a logical volume to determine the size of this volume automatically, based on your system's hardware.
For details about the recommended scheme, see Recommended Partitioning Scheme for AMD64, Intel 64, and 64-bit ARM systems.

--resize
Resize a logical volume. If you use this option, you must also specify --useexisting and --size.

--encrypted
Specifies that this logical volume should be encrypted with Linux Unified Key Setup (LUKS), using the passphrase provided in the --passphrase= option. If you do not specify a passphrase, the installation program uses the default, system-wide passphrase set with the autopart --passphrase command, or stops the installation and prompts you to provide a passphrase if no default is set.
When encrypting one or more partitions, Anaconda attempts to gather 256 bits of entropy to ensure the partitions are encrypted securely. Gathering entropy can take some time - the process will stop after a maximum of 10 minutes, regardless of whether sufficient entropy has been gathered.

The process can be sped up by interacting with the installation system (typing on the keyboard or moving the mouse). If you are installing in a virtual machine, you can also attach a `virtio-rng` device (a virtual random number generator) to the guest.

---passphrase=

Specifies the passphrase to use when encrypting this logical volume. You must use this option together with the **--encrypted** option; it has no effect by itself.

---cipher=

Specifies the type of encryption to use if the Anaconda default `aes-xts-plain64` is not satisfactory. You must use this option together with the **--encrypted** option; by itself it has no effect. Available types of encryption are listed in the Security hardening document, but Red Hat strongly recommends using either `aes-xts-plain64` or `aes-cbc-essiv:sha256`.

---escrowcert=URL_of_X.509_certificate

Store data encryption keys of all encrypted volumes as files in `/root`, encrypted using the X.509 certificate from the URL specified with `URL_of_X.509_certificate`. The keys are stored as a separate file for each encrypted volume. This option is only meaningful if **--encrypted** is specified.

---luks-version=LUKS_VERSION

Specifies which version of LUKS format should be used to encrypt the filesystem. This option is only meaningful if **--encrypted** is specified.

---backuppassphrase

Add a randomly-generated passphrase to each encrypted volume. Store these passphrases in separate files in `/root`, encrypted using the X.509 certificate specified with **--escrowcert**. This option is only meaningful if **--escrowcert** is specified.

---pbkdf=PBKDF

Sets Password-Based Key Derivation Function (PBKDF) algorithm for LUKS keyslot. See also the man page `cryptsetup(8)`. This option is only meaningful if **--encrypted** is specified.

---pbkdf-memory=PBKDF_MEMORY

Sets the memory cost for PBKDF. See also the man page `cryptsetup(8)`. This option is only meaningful if **--encrypted** is specified.

---pbkdf-time=PBKDF_TIME

Sets the number of milliseconds to spend with PBKDF passphrase processing. See also **--iter-time** in the man page `cryptsetup(8)`. This option is only meaningful if **--encrypted** is specified, and is mutually exclusive with **--pbkdf-iterations**.

---pbkdf-iterations=PBKDF_ITERATIONS

Sets the number of iterations directly and avoids PBKDF benchmark. See also **--pbkdf-force-iterations** in the man page `cryptsetup(8)`. This option is only meaningful if **--encrypted** is specified, and is mutually exclusive with **--pbkdf-time**.

---thinpool

 Creates a thin pool logical volume. (Use a mount point of none)

---metadatasize=size

Specify the metadata area size (in MiB) for a new thin pool device.
specify the chunk size (in KiB) for a new thin pool device.

--thin
Create a thin logical volume. (Requires use of --poolname)

--poolname=name
Specify the name of the thin pool in which to create a thin logical volume. Requires the --thin option.

--profile=name
Specify the configuration profile name to use with thin logical volumes. If used, the name will also be included in the metadata for the given logical volume. By default, the available profiles are default and thin-performance and are defined in the /etc/lvm/profile/ directory. See the lvm(8) man page for additional information.

--cachelevs=
A comma-separated list of physical volumes which should be used as a cache for this volume.

--cachemode=
Specify which mode should be used to cache this logical volume - either writeback or writethrough.

**NOTE**
For more information about cached logical volumes and their modes, see the lvmcache(7) man page.

--cachesize=
Size of cache attached to the logical volume, specified in MiB. This option requires the --cachelevs= option.

Notes
- Do not use the dash (-) character in logical volume and volume group names when installing Red Hat Enterprise Linux using Kickstart. If this character is used, the installation finishes normally, but the /dev/mapper/ directory will list these volumes and volume groups with every dash doubled. For example, a volume group named volgrp-01 containing a logical volume named logvol-01 will be listed as /dev/mapper/volgrp-01-logvol-01. This limitation only applies to newly created logical volume and volume group names. If you are reusing existing ones using the --noformat option, their names will not be changed.

- If you lose the LUKS passphrase, any encrypted partitions and their data is completely inaccessible. There is no way to recover a lost passphrase. However, you can save encryption passphrases with the --escrowcert and create backup encryption passphrases with the --backuppassphrase options.

Examples
- Create the partition first, create the logical volume group, and then create the logical volume:

```
part pv.01 --size 3000
volgroup myvg pv.01
logvol / --vgname=myvg --size=2000 --name=rootvol
```

- Create the partition first, create the logical volume group, and then create the logical volume to occupy 90% of the remaining space in the volume group:
part pv.01 --size 1 --grow
volgroup myvg pv.01
logvol / --vgname=myvg --name=rootvol --percent=90

Additional resources

- Configuring and managing logical volumes

J.5.11. mount

The `mount` Kickstart command is optional. It assigns a mount point to an existing block device, and optionally reformats it to a given format.

**Syntax**

```
mount [OPTIONS] device mountpoint
```

**Mandatory options:**

- `device` - The block device to mount.

- `mountpoint` - Where to mount the `device`. It must be a valid mount point, such as `/` or `/usr`, or `none` if the device is unmountable (for example `swap`).

**Optional options:**

- `--reformat=` - Specifies a new format (such as `ext4`) to which the device should be reformatted.

- `--mkfsoptions=` - Specifies additional options to be passed to the command which creates the new file system specified in `--reformat=`. The list of options provided here is not processed, so they must be specified in a format that can be passed directly to the `mkfs` program. The list of options should be either comma-separated or surrounded by double quotes, depending on the file system. See the `mkfs` man page for the file system you want to create (for example `mkfs.ext4(8)` or `mkfs.xfs(8)`) for specific details.

- `--mountoptions=` - Specifies a free form string that contains options to be used when mounting the file system. The string will be copied to the `/etc/fstab` file on the installed system and should be enclosed in double quotes. See the `mount(8)` man page for a full list of mount options, and `fstab(5)` for basics.

**Notes**

- Unlike most other storage configuration commands in Kickstart, `mount` does not require you to describe the entire storage configuration in the Kickstart file. You only need to ensure that the described block device exists on the system. However, if you want to create the storage stack with all the devices mounted, you must use other commands such as `part` to do so.

- You can not use `mount` together with other storage-related commands such as `part`, `logvol`, or `autopart` in the same Kickstart file.

J.5.12. nvdimm

The `nvdimm` Kickstart command is optional. It performs an action on Non-Volatile Dual In-line Memory Module (NVDIMM) devices.
Syntax

```
nvdimm action [OPTIONS]
```

### Actions

- **reconfigure** - Reconfigure a specific NVDIMM device into a given mode. Additionally, the specified device is implicitly marked as to be used, so a subsequent `nvdimm use` command for the same device is redundant. This action uses the following format:

  ```
  nvdimm reconfigure [--namespace=NAMESPACE] [--mode=MODE] [--sectorsize=SECTORSIZE]
  ```

  - **--namespace** - The device specification by namespace. For example:
    ```
    nvdimm reconfigure --namespace=namespace0.0 --mode=sector --sectorsize=512
    ```
  - **--mode** - The mode specification. Currently, only the value `sector` is available.
  - **--sectorsize** - Size of a sector for sector mode. For example:
    ```
    nvdimm reconfigure --namespace=namespace0.0 --mode=sector --sectorsize=512
    ```

  The supported sector sizes are 512 and 4096 bytes.

- **use** - Specify a NVDIMM device as a target for installation. The device must be already configured to the sector mode by the `nvdimm reconfigure` command. This action uses the following format:

  ```
  nvdimm use [--namespace=NAMESPACE|--blockdevs=DEVICES]
  ```

  - **--namespace** - Specifies the device by namespace. For example:
    ```
    nvdimm use --namespace=namespace0.0
    ```
  - **--blockdevs** - Specifies a comma-separated list of block devices corresponding to the NVDIMM devices to be used. The asterisk * wildcard is supported. For example:
    ```
    nvdimm use --blockdevs=pmem0s,pmem1s
    nvdimm use --blockdevs=pmem*
    ```

### Notes

- By default, all NVDIMM devices are ignored by the installation program. You must use the `nvdimm` command to enable installation on these devices.

#### J.5.13. part or partition

The `part` or `partition` Kickstart command is required. It creates a partition on the system.

**Syntax**
**part|partition**  
`mntpoint --name=name --device=device --rule=rule [OPTIONS]`

**Options**

- **mntpoint** - Where the partition is mounted. The value must be of one of the following forms:
  - `/path`
    For example, `/`, `/usr`, `/home`
  - **swap**
    The partition is used as swap space.
    To determine the size of the swap partition automatically, use the **--recommended** option:
    ```
    swap --recommended
    ```
    The size assigned will be effective but not precisely calibrated for your system.
    To determine the size of the swap partition automatically but also allow extra space for your system to hibernate, use the **--hibernation** option:
    ```
    swap --hibernation
    ```
    The size assigned will be equivalent to the swap space assigned by **--recommended** plus the amount of RAM on your system.
    For the swap sizes assigned by these commands, see Section E.4, “Recommended partitioning scheme” for AMD64, Intel 64, and 64-bit ARM systems.
  - **raid.id**
    The partition is used for software RAID (see **raid**).
  - **pv.id**
    The partition is used for LVM (see **logvol**).
  - **biosboot**
    The partition will be used for a BIOS Boot partition. A 1 MiB BIOS boot partition is necessary on BIOS-based AMD64 and Intel 64 systems using a GUID Partition Table (GPT); the boot loader will be installed into it. It is not necessary on UEFI systems. See also the **bootloader** command.
  - **/boot/efi**
    An EFI System Partition. A 50 MiB EFI partition is necessary on UEFI-based AMD64, Intel 64, and 64-bit ARM; the recommended size is 200 MiB. It is not necessary on BIOS systems. See also the **bootloader** command.
- **--size=** - The minimum partition size in MiB. Specify an integer value here such as **500** (do not include the unit).

**IMPORTANT**

If the **--size** value is too small, the installation fails. Set the **--size** value as the minimum amount of space you require. For size recommendations, see Section E.4, “Recommended partitioning scheme”.
• **--grow** - Tells the partition to grow to fill available space (if any), or up to the maximum size setting, if one is specified.

**NOTE**

If you use **--grow** without setting **--maxsize** on a swap partition, Anaconda limits the maximum size of the swap partition. For systems that have less than 2 GiB of physical memory, the imposed limit is twice the amount of physical memory. For systems with more than 2 GiB, the imposed limit is the size of physical memory plus 2GiB.

• **--maxsize** - The maximum partition size in MiB when the partition is set to grow. Specify an integer value here such as 500 (do not include the unit).

• **--noformat** - Specifies that the partition should not be formatted, for use with the **--onpart** command.

• **--onpart** or **--usepart** - Specifies the device on which to place the partition. Uses an existing blank device and format it to the new specified type. For example:

```
partition /home --onpart=hda1
```

puts /home on /dev/hda1.

These options can also add a partition to a logical volume. For example:

```
partition pv.1 --onpart=hda2
```

The device must already exist on the system; the **--onpart** option will not create it.

It is also possible to specify an entire drive, rather than a partition, in which case Anaconda will format and use the drive without creating a partition table. Note, however, that installation of GRUB2 is not supported on a device formatted in this way, and must be placed on a drive with a partition table.

```
partition pv.1 --onpart=hdb
```

• **--ondisk** or **--ondrive** - Creates a partition (specified by the **part** command) on an existing disk. This command always creates a partition. Forces the partition to be created on a particular disk. For example, **--ondisk=sdb** puts the partition on the second SCSI disk on the system.

To specify a multipath device that does not use logical volume management (LVM), use the format **disk/by-id/dm-uuid-mpath-**WWID**, where **WWID** is the world-wide identifier for the device. For example, to specify a disk with WWID **2416CD96995134CA5D787F00A5AA11017**, use:

```
part / --fstype=xfs --grow --asprimary --size=8192 --ondisk=disk/by-id/dm-uuid-mpath-2416CD96995134CA5D787F00A5AA11017
```
WARNING

Never specify multipath devices by device names like `mpatha`. Device names such as this are not specific to a particular disk. The disk named `/dev/mpatha` during installation might not be the one that you expect it to be. Therefore, the `part` command could target the wrong disk.

- **--asprimary** - Forces the partition to be allocated as a primary partition. If the partition cannot be allocated as primary (usually due to too many primary partitions being already allocated), the partitioning process fails. This option only makes sense when the disk uses a Master Boot Record (MBR); for GUID Partition Table (GPT)-labeled disks this option has no meaning.

- **--fsprofile** - Specifies a usage type to be passed to the program that makes a filesystem on this partition. A usage type defines a variety of tuning parameters to be used when making a filesystem. For this option to work, the filesystem must support the concept of usage types and there must be a configuration file that lists valid types. For `ext2`, `ext3`, `ext4`, this configuration file is `/etc/mke2fs.conf`.

- **--mkfsoptions** - Specifies additional parameters to be passed to the program that makes a filesystem on this partition. This is similar to `--fsprofile` but works for all filesystems, not just the ones that support the profile concept. No processing is done on the list of arguments, so they must be supplied in a format that can be passed directly to the mkfs program. This means multiple options should be comma-separated or surrounded by double quotes, depending on the filesystem.

- **--fstype** - Sets the file system type for the partition. Valid values are `xfs`, `ext2`, `ext3`, `ext4`, `swap`, `vfat`, `efi` and `biosboot`.

- **--fsoptions** - Specifies a free form string of options to be used when mounting the filesystem. This string will be copied into the `/etc/fstab` file of the installed system and should be enclosed in quotes.

**NOTE**

In the EFI system partition (`/boot/efi`), anaconda hard codes the value and ignores the users specified `--fsoptions` values.

- **--label** - assign a label to an individual partition.

- **--recommended** - Determine the size of the partition automatically.
  For details about the recommended scheme, see Section E.4, “Recommended partitioning scheme” for AMD64, Intel 64, and 64-bit ARM.

**IMPORTANT**

This option can only be used for partitions which result in a file system such as the `/boot` partition and `swap` space. It cannot be used to create LVM physical volumes or RAID members.

- **--onbiosdisk** - Forces the partition to be created on a particular disk as discovered by the BIOS.
- **--encrypted** - Specifies that this partition should be encrypted with Linux Unified Key Setup (LUKS), using the passphrase provided in the **--passphrase** option. If you do not specify a passphrase, Anaconda uses the default, system-wide passphrase set with the `autopart --passphrase` command, or stops the installation and prompts you to provide a passphrase if no default is set.

**NOTE**

When encrypting one or more partitions, Anaconda attempts to gather 256 bits of entropy to ensure the partitions are encrypted securely. Gathering entropy can take some time - the process will stop after a maximum of 10 minutes, regardless of whether sufficient entropy has been gathered.

The process can be sped up by interacting with the installation system (typing on the keyboard or moving the mouse). If you are installing in a virtual machine, you can also attach a `virtio-rng` device (a virtual random number generator) to the guest.

- **--luks-version=LUKS_VERSION** - Specifies which version of LUKS format should be used to encrypt the filesystem. This option is only meaningful if **--encrypted** is specified.

- **--passphrase=** - Specifies the passphrase to use when encrypting this partition. You must use this option together with the **--encrypted** option; by itself it has no effect.

- **--cipher=** - Specifies the type of encryption to use if the Anaconda default `aes-xts-plain64` is not satisfactory. You must use this option together with the **--encrypted** option; by itself it has no effect. Available types of encryption are listed in the Security hardening document, but Red Hat strongly recommends using either `aes-xts-plain64` or `aes-cbc-essiv:sha256`.

- **--escrowcert=URL_of_X.509_certificate** - Store data encryption keys of all encrypted partitions as files in `/root`, encrypted using the X.509 certificate from the URL specified with `URL_of_X.509_certificate`. The keys are stored as a separate file for each encrypted partition. This option is only meaningful if **--encrypted** is specified.

- **--backuppassphrase** - Add a randomly-generated passphrase to each encrypted partition. Store these passphrases in separate files in `/root`, encrypted using the X.509 certificate specified with `--escrowcert`. This option is only meaningful if `--escrowcert` is specified.

- **--pbkdf=PBKDF** - Sets Password-Based Key Derivation Function (PBKDF) algorithm for LUKS keyslot. See also the man page `cryptsetup(8)`. This option is only meaningful if **--encrypted** is specified.

- **--pbkdf-memory=PBKDF_MEMORY** - Sets the memory cost for PBKDF. See also the man page `cryptsetup(8)`. This option is only meaningful if **--encrypted** is specified.

- **--pbkdf-time=PBKDF_TIME** - Sets the number of milliseconds to spend with PBKDF passphrase processing. See also **--iter-time** in the man page `cryptsetup(8)`. This option is only meaningful if **--encrypted** is specified, and is mutually exclusive with **--pbkdf-iterations**.

- **--pbkdf-iterations=PBKDF_ITERATIONS** - Sets the number of iterations directly and avoids PBKDF benchmark. See also **--pbkdf-force-iterations** in the man page `cryptsetup(8)`. This option is only meaningful if **--encrypted** is specified, and is mutually exclusive with **--pbkdf-time**.

- **--resize=** - Resize an existing partition. When using this option, specify the target size (in MiB) using the **--size=** option and the target partition using the **--onpart=** option.
Notes

- The `part` command is not mandatory, but you must include either `part`, `autopart` or `mount` in your Kickstart script.

- The `--active` option is deprecated in Red Hat Enterprise Linux 8.

- If partitioning fails for any reason, diagnostic messages appear on virtual console 3.

- All partitions created are formatted as part of the installation process unless `--noformat` and `--onpart` are used.

- Device names in the `sdX` (or `/dev/sdX`) format are not guaranteed to be consistent across reboots, which can complicate usage of some Kickstart commands. When a command calls for a device node name, you can instead use any item from `/dev/disk`. For example, instead of:

  ```bash
  part / --fstype=xfs --onpart=sda1
  ```

  You could use an entry similar to one of the following:

  ```bash
  part / --fstype=xfs --onpart=/dev/disk/by-path/pci-0000:00:05.0-scsi-0:0:0:0-0:0:0:0-part1
  part / --fstype=xfs --onpart=/dev/disk/by-id/ata-ST3160815AS_6RA0C882-part1
  ```

  This way the command will always target the same storage device. This is especially useful in large storage environments. See the chapter Overview of persistent naming attributes in the Managing storage devices document for more in-depth information about different ways to consistently refer to storage devices.

- If you lose the LUKS passphrase, any encrypted partitions and their data is completely inaccessible. There is no way to recover a lost passphrase. However, you can save encryption passphrases with the `--escrowcert` and create backup encryption passphrases with the `--backuppassphrase` options.

J.5.14. raid

The `raid` Kickstart command is optional. It assembles a software RAID device.

Syntax

```bash
raid mntpoint --level=level --device=device-name partitions*
```

Options

- `mntpoint` - Location where the RAID file system is mounted. If it is `/`, the RAID level must be 1 unless a boot partition (`/boot`) is present. If a boot partition is present, the `/boot` partition must be level 1 and the root (`/`) partition can be any of the available types. The `partitions*` (which denotes that multiple partitions can be listed) lists the RAID identifiers to add to the RAID array.
IMPORTANT

- On IBM Power Systems, if a RAID device has been prepared and has not been reformatted during the installation, ensure that the RAID metadata version is 0.90 if you intend to put the `/boot` and PReP partitions on the RAID device. The default Red Hat Enterprise Linux 7 `mdadm` metadata version is not supported for the boot device.

- The PReP Boot partitions are not required on PowerNV systems.

- `--level=` - RAID level to use (0, 1, 4, 5, 6, or 10). See Section E.3, "Supported RAID types" for information about various available RAID levels.

- `--device=` - Name of the RAID device to use - for example, `--device=root`.

IMPORTANT

Do not use `mdraid` names in the form of `md0` - these names are not guaranteed to be persistent. Instead, use meaningful names such as `root` or `swap`. Using meaningful names creates a symbolic link from `/dev/md/name` to whichever `/dev/md/X` node is assigned to the array.

If you have an old (v0.90 metadata) array that you cannot assign a name to, you can specify the array by a filesystem label or UUID (for example, `--device=rhel7-root --label=rhel7-root`).

- `--chunksize=` - Sets the chunk size of a RAID storage in KiB. In certain situations, using a different chunk size than the default (512 Kib) can improve the performance of the RAID.

- `--spares=` - Specifies the number of spare drives allocated for the RAID array. Spare drives are used to rebuild the array in case of drive failure.

- `--fsprofile=` - Specifies a usage type to be passed to the program that makes a filesystem on this partition. A usage type defines a variety of tuning parameters to be used when making a filesystem. For this option to work, the filesystem must support the concept of usage types and there must be a configuration file that lists valid types. For ext2, ext3, and ext4, this configuration file is `/etc/mke2fs.conf`.

- `--fstype=` - Sets the file system type for the RAID array. Valid values are `xfs`, `ext2`, `ext3`, `ext4`, `swap`, and `vfat`.

- `--fsoptions=` - Specifies a free form string of options to be used when mounting the filesystem. This string will be copied into the `/etc/fstab` file of the installed system and should be enclosed in quotes.

NOTE

In the EFI system partition (`/boot/efi`), anaconda hard codes the value and ignores the users specified `--fsoptions` values.

- `--mkfsoptions=` - Specifies additional parameters to be passed to the program that makes a filesystem on this partition. No processing is done on the list of arguments, so they must be supplied in a format that can be passed directly to the mkfs program. This means multiple options should be comma-separated or surrounded by double quotes, depending on the filesystem.
- **--label=** - Specify the label to give to the filesystem to be made. If the given label is already in use by another filesystem, a new label will be created.

- **--noformat** - Use an existing RAID device and do not format the RAID array.

- **--useexisting** - Use an existing RAID device and reformat it.

- **--encrypted** - Specifies that this RAID device should be encrypted with Linux Unified Key Setup (LUKS), using the passphrase provided in the **--passphrase** option. If you do not specify a passphrase, Anaconda uses the default, system-wide passphrase set with the `autopart --passphrase` command, or stops the installation and prompts you to provide a passphrase if no default is set.

**NOTE**

When encrypting one or more partitions, Anaconda attempts to gather 256 bits of entropy to ensure the partitions are encrypted securely. Gathering entropy can take some time - the process will stop after a maximum of 10 minutes, regardless of whether sufficient entropy has been gathered.

The process can be sped up by interacting with the installation system (typing on the keyboard or moving the mouse). If you are installing in a virtual machine, you can also attach a `virtio-rng` device (a virtual random number generator) to the guest.

- **--luks-version=LUKS_VERSION** - Specifies which version of LUKS format should be used to encrypt the filesystem. This option is only meaningful if **--encrypted** is specified.

- **--cipher=** - Specifies the type of encryption to use if the Anaconda default `aes-xts-plain64` is not satisfactory. You must use this option together with the **--encrypted** option; by itself it has no effect. Available types of encryption are listed in the Security hardening document, but Red Hat strongly recommends using either `aes-xts-plain64` or `aes-cbc-essiv:sha256`.

- **--passphrase=** - Specifies the passphrase to use when encrypting this RAID device. You must use this option together with the **--encrypted** option; by itself it has no effect.

- **--escrowcert=URL_of_X.509_certificate** - Store the data encryption key for this device in a file in `/root`, encrypted using the X.509 certificate from the URL specified with `URL_of_X.509_certificate`. This option is only meaningful if **--encrypted** is specified.

- **--backuppassphrase** - Add a randomly-generated passphrase to this device. Store the passphrase in a file in `/root`, encrypted using the X.509 certificate specified with **--escrowcert**. This option is only meaningful if **--escrowcert** is specified.

- **--pbkdf=PBKDF** - Sets Password-Based Key Derivation Function (PBKDF) algorithm for LUKS keyslot. See also the man page `cryptsetup(8)`. This option is only meaningful if **--encrypted** is specified.

- **--pbkdf-memory=PBKDF_MEMORY** - Sets the memory cost for PBKDF. See also the man page `cryptsetup(8)`. This option is only meaningful if **--encrypted** is specified.

- **--pbkdf-time=PBKDF_TIME** - Sets the number of milliseconds to spend with PBKDF passphrase processing. See also **--iter-time** in the man page `cryptsetup(8)`. This option is only meaningful if **--encrypted** is specified, and is mutually exclusive with **--pbkdf-iterations**.

- **--pbkdf-iterations=PBKDF_ITERATIONS** - Sets the number of iterations directly and avoids
PBKDF benchmark. See also --pbkdf-force-iterations in the man page cryptsetup(8). This option is only meaningful if --encrypted is specified, and is mutually exclusive with --pbkdf-time.

Example

The following example shows how to create a RAID level 1 partition for /, and a RAID level 5 for /home, assuming there are three SCSI disks on the system. It also creates three swap partitions, one on each drive.

```
part raid.01 --size=6000 --ondisk=sda
part raid.02 --size=6000 --ondisk=sdb
part raid.03 --size=6000 --ondisk=sdc
part swap --size=512 --ondisk=sda
part swap --size=512 --ondisk=sdb
part swap --size=512 --ondisk=sdc
part raid.11 --size=1 --grow --ondisk=sda
part raid.12 --size=1 --grow --ondisk=sdb
part raid.13 --size=1 --grow --ondisk=sdc
raid / --level=1 --device=rhel8-root --label=rhel8-root raid.01 raid.02 raid.03
raid /home --level=5 --device=rhel8-home --label=rhel8-home raid.11 raid.12 raid.13
```

Notes

- If you lose the LUKS passphrase, any encrypted partitions and their data is completely inaccessible. There is no way to recover a lost passphrase. However, you can save encryption passphrases with the --escrowcert and create backup encryption passphrases with the --backup passphrase options.

J.5.15. reqpart

The reqpart Kickstart command is optional. It automatically creates partitions required by your hardware platform. These include a /boot/efi partition for systems with UEFI firmware, a biosboot partition for systems with BIOS firmware and GPT, and a PRePBoot partition for IBM Power Systems.

Syntax

```
reqpart [--add-boot]
```

Options

- `--add-boot` - Creates a separate /boot partition in addition to the platform-specific partition created by the base command.

Notes

- This command cannot be used together with autopart, because autopart does everything the reqpart command does and, in addition, creates other partitions or logical volumes such as / and swap. In contrast with autopart, this command only creates platform-specific partitions and leaves the rest of the drive empty, allowing you to create a custom layout.

J.5.16. snapshot
The `snapshot` Kickstart command is optional. Use it to create LVM thin volume snapshots during the installation process. This enables you to back up a logical volume before or after the installation.

To create multiple snapshots, add the `snapshot` Kickstart command multiple times.

**Syntax**

```bash
snapshot vg_name/lv_name --name=snapshot_name --when=pre-install|post-install
```

**Options**

- `vg_name/lv_name` - Sets the name of the volume group and logical volume to create the snapshot from.
- `--name=snapshot_name` - Sets the name of the snapshot. This name must be unique within the volume group.
- `--when=pre-install|post-install` - Sets if the snapshot is created before the installation begins or after the installation is completed.

### J.5.17. volgroup

The `volgroup` Kickstart command is optional. It creates a Logical Volume Management (LVM) group.

**Syntax**

```bash
volgroup name [OPTIONS] [partition*]
```

**Mandatory options**

- `name` - Name of the new volume group.

**Options**

- `partition` - Physical volume partitions to use as backing storage for the volume group.
- `--noformat` - Use an existing volume group and do not format it.
- `--useexisting` - Use an existing volume group and reformat it. If you use this option, do not specify a `partition`. For example:
  ```bash
  volgroup rhel00 --useexisting --noformat
  ```
- `--pesize=` - Set the size of the volume group’s physical extents in KiB. The default value is 4096 (4 MiB), and the minimum value is 1024 (1 MiB).
- `--reserved-space=` - Specify an amount of space to leave unused in a volume group in MiB. Applicable only to newly created volume groups.
- `--reserved-percent=` - Specify a percentage of total volume group space to leave unused. Applicable only to newly created volume groups.

**Notes**
Create the partition first, then create the logical volume group, and then create the logical volume. For example:

```
part pv.01 --size 10000
volgroup my_volgrp pv.01
logvol / --vgname=my_volgrp --size=2000 --name=root
```

Do not use the dash (-) character in logical volume and volume group names when installing Red Hat Enterprise Linux using Kickstart. If this character is used, the installation finishes normally, but the `/dev/mapper/` directory will list these volumes and volume groups with every dash doubled. For example, a volume group named `volgrp-01` containing a logical volume named `logvol-01` will be listed as `/dev/mapper/volgrp--01-logvol--01`.

This limitation only applies to newly created logical volume and volume group names. If you are reusing existing ones using the `--noformat` option, their names will not be changed.

### J.5.18. zerombr

The `zerombr` Kickstart command is optional. The `zerombr` initializes any invalid partition tables that are found on disks and destroys all of the contents of disks with invalid partition tables. This command is required when performing an installation on an 64-bit IBM Z system with unformatted Direct Access Storage Device (DASD) disks, otherwise the unformatted disks are not formatted and used during the installation.

**Syntax**

```
zerombr
```

**Notes**

- On 64-bit IBM Z, if `zerombr` is specified, any Direct Access Storage Device (DASD) visible to the installation program which is not already low-level formatted is automatically low-level formatted with `dasdfmt`. The command also prevents user choice during interactive installations.
- If `zerombr` is not specified and there is at least one unformatted DASD visible to the installation program, a non-interactive Kickstart installation exits unsuccessfully.
- If `zerombr` is not specified and there is at least one unformatted DASD visible to the installation program, an interactive installation exits if the user does not agree to format all visible and unformatted DASDs. To circumvent this, only activate those DASDs that you will use during installation. You can always add more DASDs after installation is complete.
- This command has no options.

### J.5.19. zfcp

The `zfcp` Kickstart command is optional. It defines a Fibre channel device.

This option only applies on 64-bit IBM Z. All of the options described below must be specified.

**Syntax**

```
zfcp --devnum=devnum --wwpn=wwpn --fcplun=lun
```
Options

- **--devnum=** - The device number (zFCP adapter device bus ID).
- **--wwpn=** - The device’s World Wide Port Name (WWPN). Takes the form of a 16-digit number, preceded by `0x`.
- **--fcplun=** - The device’s Logical Unit Number (LUN). Takes the form of a 16-digit number, preceded by `0x`.

Example

```
zfcp --devnum=0.0.4000 --wwpn=0x5005076300C213e9 --fcplun=0x5022000000000000
```

J.6. KICKSTART COMMANDS FOR ADDONS SUPPLIED WITH THE RHEL INSTALLATION PROGRAM

The Kickstart commands in this section are related to add-ons supplied by default with the Red Hat Enterprise Linux installation program: Kdump and OpenSCAP.

J.6.1. `%addon com_redhat_kdump`

The `%addon com_redhat_kdump` Kickstart command is optional. This command configures the kdump kernel crash dumping mechanism.

Syntax

```
%addon com_redhat_kdump [OPTIONS]
%end
```

**NOTE**

The syntax for this command is unusual because it is an add-on rather than a built-in Kickstart command.

Notes

Kdump is a kernel crash dumping mechanism that allows you to save the contents of the system’s memory for later analysis. It relies on `kexec`, which can be used to boot a Linux kernel from the context of another kernel without rebooting the system, and preserve the contents of the first kernel’s memory that would otherwise be lost.

In case of a system crash, `kexec` boots into a second kernel (a capture kernel). This capture kernel resides in a reserved part of the system memory. Kdump then captures the contents of the crashed kernel’s memory (a crash dump) and saves it to a specified location. The location cannot be configured using this Kickstart command; it must be configured after the installation by editing the `/etc/kdump.conf` configuration file.

For more information about Kdump, see the Installing and configuring kdump chapter of the Managing, monitoring and updating the kernel document.

Options
• **--enable** - Enable kdump on the installed system.

• **--disable** - Disable kdump on the installed system.

• **--reserve-mb** - The amount of memory you want to reserve for kdump, in MiB. For example:

  ```
  %addon com_redhat_kdump --enable --reserve-mb=128
  %end
  ```

  You can also specify `auto` instead of a numeric value. In that case, the installation program will determine the amount of memory automatically based on the criteria described in the Memory requirements for kdump section of the Managing, monitoring and updating the kernel document.

  If you enable kdump and do not specify a **--reserve-mb** option, the value `auto` will be used.

• **--enablefadump** - Enable firmware-assisted dumping on systems which allow it (notably, IBM Power Systems servers).

### J.6.2. %addon org_fedora_oscap

The **%addon org_fedora_oscap** Kickstart command is optional.

The OpenSCAP installation program add-on is used to apply SCAP (Security Content Automation Protocol) content - security policies - on the installed system. This add-on has been enabled by default since Red Hat Enterprise Linux 7.2. When enabled, the packages necessary to provide this functionality will automatically be installed. However, by default, no policies are enforced, meaning that no checks are performed during or after installation unless specifically configured.

**IMPORTANT**

Applying a security policy is not necessary on all systems. This command should only be used when a specific policy is mandated by your organization rules or government regulations.

Unlike most other commands, this add-on does not accept regular options, but uses key-value pairs in the body of the **%addon** definition instead. These pairs are whitespace-agnostic. Values can be optionally enclosed in single quotes (') or double quotes (").

**Syntax**

```
%addon org_fedora_oscap
  key = value
%end
```

**Keys**

The following keys are recognized by the add-on:

**content-type**

Type of the security content. Possible values are **datastream**, **archive**, **rpm**, and **scap-security-guide**.

If the **content-type** is **scap-security-guide**, the add-on will use content provided by the **scap-security-guide** package, which is present on the boot media. This means that all other keys except **profile** will have no effect.
content-url

Location of the security content. The content must be accessible using HTTP, HTTPS, or FTP; local storage is currently not supported. A network connection must be available to reach content definitions in a remote location.

datastream-id

ID of the data stream referenced in the content-url value. Used only if content-type is datastream.

xccdf-id

ID of the benchmark you want to use.

content-path

Path to the datastream or the XCCDF file which should be used, given as a relative path in the archive.

profile

ID of the profile to be applied. Use default to apply the default profile.

fingerprint

A MD5, SHA1 or SHA2 checksum of the content referenced by content-url.

tailoring-path

Path to a tailoring file which should be used, given as a relative path in the archive.

Examples

- The following is an example %addon org_fedora_oscap section which uses content from the scap-security-guide on the installation media:

  Example J.1. Sample OpenSCAP Add-on Definition Using SCAP Security Guide

  %addon org_fedora_oscap
  content-type = scap-security-guide
  profile = xccdf_org.ssgproject.content_profile_pci-dss
  %end

- The following is a more complex example which loads a custom profile from a web server:

  Example J.2. Sample OpenSCAP Add-on Definition Using a Datastream

  %addon org_fedora_oscap
  content-type = datastream
  content-url = http://www.example.com/scap/testing_ds.xml
  datastream-id = scap_example.com_datastream_testing
  xccdf-id = scap_example.com_cref_xccdf.xml
  profile = scap_example.com_profile_my_profile
  fingerprint = 240f2f18222fda98856c3b4fc50c4195
  %end

Additional resources

- Security Hardening
- OpenSCAP installation program add-on
J.7. COMMANDS USED IN ANACONDA

The `pwpolicy` command is an Anaconda UI specific command that can be used only in the `%anaconda` section of the kickstart file.

J.7.1. pwpolicy

The `pwpolicy` Kickstart command is optional. Use this command to enforce a custom password policy during installation. The policy requires you to create passwords for the root, users, or the luks user accounts. The factors such as password length and strength decide the validity of a password.

Syntax

```
pwpolicy name [--minlen=length] [--minquality=quality] [--strict|--nostrict] [--emptyok|--noempty] [--changesok|--nochanges]
```

Mandatory options

- `name` - Replace with either `root`, `user` or `luks` to enforce the policy for the `root` password, user passwords, or LUKS passphrase, respectively.

Optional options

- `--minlen=` - Sets the minimum allowed password length, in characters. The default is 6.
- `--minquality=` - Sets the minimum allowed password quality as defined by the `libpwquality` library. The default value is 1.
- `--strict` - Enables strict password enforcement. Passwords which do not meet the requirements specified in `--minquality=` and `--minlen=` will not be accepted. This option is disabled by default.
- `--notstrict` - Passwords which do not meet the minimum quality requirements specified by the `--minquality=` and `--minlen=` options will be allowed, after `Done` is clicked twice in the GUI. For text mode interface, a similar mechanism is used.
- `--emptyok` - Allows the use of empty passwords. Enabled by default for user passwords.
- `--notempty` - Disallows the use of empty passwords. Enabled by default for the root password and the LUKS passphrase.
- `--changesok` - Allows changing the password in the user interface, even if the Kickstart file already specifies a password. Disabled by default.
- `--nochanges` - Disallows changing passwords which are already set in the Kickstart file. Enabled by default.

Notes

- The `pwpolicy` command is an Anaconda-UI specific command that can be used only in the `%anaconda` section of the kickstart file.
The libpwquality library is used to check minimum password requirements (length and quality). You can use the pwscore and pwmake commands provided by the libpwquality package to check the quality score of a password, or to create a random password with a given score. See the pwscore(1) and pwmake(1) man page for details about these commands.

J.8. KICKSTART COMMANDS FOR SYSTEM RECOVERY

The Kickstart command in this section repairs an installed system.

J.8.1. rescue

The rescue Kickstart command is optional. It provides a shell environment with root privileges and a set of system management tools to repair the installation and to troubleshoot the issues like:

- Mount file systems as read-only
- Blocklist or add a driver provided on a driver disc
- Install or upgrade system packages
- Manage partitions

**NOTE**

The Kickstart rescue mode is different from the rescue mode and emergency mode, which are provided as part of the systemd and service manager.

The rescue command does not modify the system on its own. It only sets up the rescue environment by mounting the system under /mnt/sysimage in a read-write mode. You can choose not to mount the system, or to mount it in read-only mode.

**Syntax**

```
rescue [--nomount|--romount]
```

**Options**

- `--nomount` or `--romount` - Controls how the installed system is mounted in the rescue environment. By default, the installation program finds your system and mount it in read-write mode, telling you where it has performed this mount. You can optionally select to not mount anything (the `--nomount` option) or mount in read-only mode (the `--romount` option). Only one of these two options can be used.

**Notes**

To run a rescue mode, make a copy of the Kickstart file, and include the rescue command in it.

Using the rescue command causes the installer to perform the following steps:

1. Run the %pre script.
2. Set up environment for rescue mode.
   The following kickstart commands take effect:
a. updates
b. sshpw
c. logging
d. lang
e. network

3. Set up advanced storage environment.
   The following kickstart commands take effect:
   a. fcoe
   b. iscsi
   c. iscsiname
   d. nvdimm
   e. zfcp

4. Mount the system
   
   ```
   rescue [--nomount|--romount]
   ```

5. Run %post script
   This step is run only if the installed system is mounted in read-write mode.

6. Start shell

7. Reboot system
PART II. DESIGN OF SECURITY
CHAPTER 11. OVERVIEW OF SECURITY HARDENING IN RHEL

Due to the increased reliance on powerful, networked computers to help run businesses and keep track of our personal information, entire industries have been formed around the practice of network and computer security. Enterprises have solicited the knowledge and skills of security experts to properly audit systems and tailor solutions to fit the operating requirements of their organization. Because most organizations are increasingly dynamic in nature, their workers are accessing critical company IT resources locally and remotely, hence the need for secure computing environments has become more pronounced.

Unfortunately, many organizations, as well as individual users, regard security as more of an afterthought, a process that is overlooked in favor of increased power, productivity, convenience, ease of use, and budgetary concerns. Proper security implementation is often enacted postmortem — after an unauthorized intrusion has already occurred. Taking the correct measures prior to connecting a site to an untrusted network, such as the Internet, is an effective means of thwarting many attempts at intrusion.

11.1. WHAT IS COMPUTER SECURITY?

Computer security is a general term that covers a wide area of computing and information processing. Industries that depend on computer systems and networks to conduct daily business transactions and access critical information regard their data as an important part of their overall assets. Several terms and metrics have entered our daily business vocabulary, such as total cost of ownership (TCO), return on investment (ROI), and quality of service (QoS). Using these metrics, industries can calculate aspects such as data integrity and high-availability (HA) as part of their planning and process management costs. In some industries, such as electronic commerce, the availability and trustworthiness of data can mean the difference between success and failure.

11.2. STANDARDIZING SECURITY

Enterprises in every industry rely on regulations and rules that are set by standards-making bodies such as the American Medical Association (AMA) or the Institute of Electrical and Electronics Engineers (IEEE). The same concepts hold true for information security. Many security consultants and vendors agree upon the standard security model known as CIA, or Confidentiality, Integrity, and Availability. This three-tiered model is a generally accepted component to assessing risks of sensitive information and establishing security policy. The following describes the CIA model in further detail:

- Confidentiality – Sensitive information must be available only to a set of pre-defined individuals. Unauthorized transmission and usage of information should be restricted. For example, confidentiality of information ensures that a customer’s personal or financial information is not obtained by an unauthorized individual for malicious purposes such as identity theft or credit fraud.

- Integrity – Information should not be altered in ways that render it incomplete or incorrect. Unauthorized users should be restricted from the ability to modify or destroy sensitive information.

- Availability – Information should be accessible to authorized users any time that it is needed. Availability is a warranty that information can be obtained with an agreed-upon frequency and timeliness. This is often measured in terms of percentages and agreed to formally in Service Level Agreements (SLAs) used by network service providers and their enterprise clients.

11.3. CRYPTOGRAPHIC SOFTWARE AND CERTIFICATIONS
Red Hat Enterprise Linux undergoes several security certifications, such as FIPS 140-2 or Common Criteria (CC), to ensure that industry best practices are followed.

The RHEL 8 core crypto components Knowledgebase article provides an overview of the Red Hat Enterprise Linux 8 core crypto components, documenting which are they, how are they selected, how are they integrated into the operating system, how do they support hardware security modules and smart cards, and how do crypto certifications apply to them.

11.4. SECURITY CONTROLS

Computer security is often divided into three distinct main categories, commonly referred to as controls:

- Physical
- Technical
- Administrative

These three broad categories define the main objectives of proper security implementation. Within these controls are sub-categories that further detail the controls and how to implement them.

11.4.1. Physical controls

Physical control is the implementation of security measures in a defined structure used to deter or prevent unauthorized access to sensitive material. Examples of physical controls are:

- Closed-circuit surveillance cameras
- Motion or thermal alarm systems
- Security guards
- Picture IDs
- Locked and dead-bolted steel doors
- Biometrics (includes fingerprint, voice, face, iris, handwriting, and other automated methods used to recognize individuals)

11.4.2. Technical controls

Technical controls use technology as a basis for controlling the access and usage of sensitive data throughout a physical structure and over a network. Technical controls are far-reaching in scope and encompass such technologies as:

- Encryption
- Smart cards
- Network authentication
- Access control lists (ACLs)
- File integrity auditing software
11.4.3. Administrative controls

Administrative controls define the human factors of security. They involve all levels of personnel within an organization and determine which users have access to what resources and information by such means as:

- Training and awareness
- Disaster preparedness and recovery plans
- Personnel recruitment and separation strategies
- Personnel registration and accounting

11.5. VULNERABILITY ASSESSMENT

Given time, resources, and motivation, an attacker can break into nearly any system. All of the security procedures and technologies currently available cannot guarantee that any systems are completely safe from intrusion. Routers help secure gateways to the Internet. Firewalls help secure the edge of the network. Virtual Private Networks safely pass data in an encrypted stream. Intrusion detection systems warn you of malicious activity. However, the success of each of these technologies is dependent upon a number of variables, including:

- The expertise of the staff responsible for configuring, monitoring, and maintaining the technologies.
- The ability to patch and update services and kernels quickly and efficiently.
- The ability of those responsible to keep constant vigilance over the network.

Given the dynamic state of data systems and technologies, securing corporate resources can be quite complex. Due to this complexity, it is often difficult to find expert resources for all of your systems. While it is possible to have personnel knowledgeable in many areas of information security at a high level, it is difficult to retain staff who are experts in more than a few subject areas. This is mainly because each subject area of information security requires constant attention and focus. Information security does not stand still.

A vulnerability assessment is an internal audit of your network and system security; the results of which indicate the confidentiality, integrity, and availability of your network. Typically, vulnerability assessment starts with a reconnaissance phase, during which important data regarding the target systems and resources is gathered. This phase leads to the system readiness phase, whereby the target is essentially checked for all known vulnerabilities. The readiness phase culminates in the reporting phase, where the findings are classified into categories of high, medium, and low risk; and methods for improving the security (or mitigating the risk of vulnerability) of the target are discussed.

If you were to perform a vulnerability assessment of your home, you would likely check each door to your home to see if they are closed and locked. You would also check every window, making sure that they closed completely and latch correctly. This same concept applies to systems, networks, and electronic data. Malicious users are the thieves and vandals of your data. Focus on their tools, mentality, and motivations, and you can then react swiftly to their actions.

11.5.1. Defining assessment and testing

Vulnerability assessments may be broken down into one of two types: outside looking in and inside looking around.
When performing an outside-looking-in vulnerability assessment, you are attempting to compromise your systems from the outside. Being external to your company provides you with the cracker’s point of view. You see what a cracker sees—publicly-routable IP addresses, systems on your DMZ, external interfaces of your firewall, and more. DMZ stands for “demilitarized zone”, which corresponds to a computer or small subnetwork that sits between a trusted internal network, such as a corporate private LAN, and an untrusted external network, such as the public Internet. Typically, the DMZ contains devices accessible to Internet traffic, such as web (HTTP) servers, FTP servers, SMTP (e-mail) servers and DNS servers.

When you perform an inside-looking-around vulnerability assessment, you are at an advantage since you are internal and your status is elevated to trusted. This is the point of view you and your co-workers have once logged on to your systems. You see print servers, file servers, databases, and other resources.

There are striking distinctions between the two types of vulnerability assessments. Being internal to your company gives you more privileges than an outsider. In most organizations, security is configured to keep intruders out. Very little is done to secure the internals of the organization (such as departmental firewalls, user-level access controls, and authentication procedures for internal resources). Typically, there are many more resources when looking around inside as most systems are internal to a company. Once you are outside the company, your status is untrusted. The systems and resources available to you externally are usually very limited.

Consider the difference between vulnerability assessments and penetration tests. Think of a vulnerability assessment as the first step to a penetration test. The information gleaned from the assessment is used for testing. Whereas the assessment is undertaken to check for holes and potential vulnerabilities, the penetration testing actually attempts to exploit the findings.

Assessing network infrastructure is a dynamic process. Security, both information and physical, is dynamic. Performing an assessment shows an overview, which can turn up false positives and false negatives. A false positive is a result, where the tool finds vulnerabilities which in reality do not exist. A false negative is when it omits actual vulnerabilities.

Security administrators are only as good as the tools they use and the knowledge they retain. Take any of the assessment tools currently available, run them against your system, and it is almost a guarantee that there are some false positives. Whether by program fault or user error, the result is the same. The tool may find false positives, or, even worse, false negatives.

Now that the difference between a vulnerability assessment and a penetration test is defined, take the findings of the assessment and review them carefully before conducting a penetration test as part of your new best practices approach.

**WARNING**

Do not attempt to exploit vulnerabilities on production systems. Doing so can have adverse effects on productivity and efficiency of your systems and network.

The following list examines some of the benefits of performing vulnerability assessments.

- Creates proactive focus on information security.
- Finds potential exploits before crackers find them.
- Results in systems being kept up to date and patched.
• Promotes growth and aids in developing staff expertise.

• Abates financial loss and negative publicity.

11.5.2. Establishing a methodology for vulnerability assessment

To aid in the selection of tools for a vulnerability assessment, it is helpful to establish a vulnerability assessment methodology. Unfortunately, there is no predefined or industry approved methodology at this time; however, common sense and best practices can act as a sufficient guide.

What is the target? Are we looking at one server, or are we looking at our entire network and everything within the network? Are we external or internal to the company? The answers to these questions are important as they help determine not only which tools to select but also the manner in which they are used.

To learn more about establishing methodologies, see the following website:

• https://www.owasp.org/ — The Open Web Application Security Project

11.5.3. Vulnerability assessment tools

An assessment can start by using some form of an information-gathering tool. When assessing the entire network, map the layout first to find the hosts that are running. Once located, examine each host individually. Focusing on these hosts requires another set of tools. Knowing which tools to use may be the most crucial step in finding vulnerabilities.

The following tools are just a small sampling of the available tools:

• **Nmap** is a popular tool that can be used to find host systems and open ports on those systems. To install Nmap from the AppStream repository, enter the `yum install nmap` command as the root user. See the `nmap(1)` man page for more information.

• The tools from the OpenSCAP suite, such as the `oscap` command-line utility and the `scap-workbench` graphical utility, provides a fully automated compliance audit. See Scanning the system for security compliance and vulnerabilities for more information.

• Advanced Intrusion Detection Environment (AIDE) is a utility that creates a database of files on the system, and then uses that database to ensure file integrity and detect system intrusions. See Checking integrity with AIDE for more information.

11.6. SECURITY THREATS

11.6.1. Threats to network security

Bad practices when configuring the following aspects of a network can increase the risk of an attack.

**Insecure architectures**

A misconfigured network is a primary entry point for unauthorized users. Leaving a trust-based, open local network vulnerable to the highly-insecure Internet is much like leaving a door ajar in a crime-ridden neighborhood — nothing may happen for an arbitrary amount of time, but someone exploits the opportunity eventually.

**Broadcast networks**

System administrators often fail to realize the importance of networking hardware in their security
schemes. Simple hardware, such as hubs and routers, relies on the broadcast or non-switched principle; that is, whenever a node transmits data across the network to a recipient node, the hub or router sends a broadcast of the data packets until the recipient node receives and processes the data. This method is the most vulnerable to address resolution protocol (ARP) or media access control (MAC) address spoofing by both outside intruders and unauthorized users on local hosts.

Centralized servers

Another potential networking pitfall is the use of centralized computing. A common cost-cutting measure for many businesses is to consolidate all services to a single powerful machine. This can be convenient as it is easier to manage and costs considerably less than multiple-server configurations. However, a centralized server introduces a single point of failure on the network. If the central server is compromised, it may render the network completely useless or worse, prone to data manipulation or theft. In these situations, a central server becomes an open door that allows access to the entire network.

11.6.2. Threats to server security

Server security is as important as network security because servers often hold a great deal of an organization’s vital information. If a server is compromised, all of its contents may become available for the cracker to steal or manipulate at will. The following sections detail some of the main issues.

Unused services and open ports

A full installation of Red Hat Enterprise Linux 8 contains more than 1000 applications and library packages. However, most server administrators do not opt to install every single package in the distribution, preferring instead to install a base installation of packages, including several server applications.

A common occurrence among system administrators is to install the operating system without paying attention to what programs are actually being installed. This can be problematic because unneeded services may be installed, configured with the default settings, and possibly turned on. This can cause unwanted services, such as Telnet, DHCP, or DNS, to run on a server or workstation without the administrator realizing it, which in turn can cause unwanted traffic to the server or even a potential pathway into the system for crackers.

Unpatched services

Most server applications that are included in a default installation are solid, thoroughly tested pieces of software. Having been in use in production environments for many years, their code has been thoroughly refined and many of the bugs have been found and fixed.

However, there is no such thing as perfect software and there is always room for further refinement. Moreover, newer software is often not as rigorously tested as one might expect, because of its recent arrival to production environments or because it may not be as popular as other server software.

Developers and system administrators often find exploitable bugs in server applications and publish the information on bug tracking and security-related websites such as the Bugtraq mailing list (http://www.securityfocus.com) or the Computer Emergency Response Team (CERT) website (http://www.cert.org). Although these mechanisms are an effective way of alerting the community to security vulnerabilities, it is up to system administrators to patch their systems promptly. This is particularly true because crackers have access to these same vulnerability tracking services and will use the information to crack unpatched systems whenever they can. Good system administration requires vigilance, constant bug tracking, and proper system maintenance to ensure a more secure computing environment.

Inattentive administration
Administrators who fail to patch their systems are one of the greatest threats to server security. This applies as much to inexperienced administrators as it does to overconfident or unmotivated administrators.

Some administrators fail to patch their servers and workstations, while others fail to watch log messages from the system kernel or network traffic. Another common error is when default passwords or keys to services are left unchanged. For example, some databases have default administration passwords because the database developers assume that the system administrator changes these passwords immediately after installation. If a database administrator fails to change this password, even an inexperienced cracker can use a widely-known default password to gain administrative privileges to the database. These are only a few examples of how inattentive administration can lead to compromised servers.

**Inherently insecure services**

Even the most vigilant organization can fall victim to vulnerabilities if the network services they choose are inherently insecure. For instance, there are many services developed under the assumption that they are used over trusted networks; however, this assumption fails as soon as the service becomes available over the Internet – which is itself inherently untrusted.

One category of insecure network services are those that require unencrypted user names and passwords for authentication. Telnet and FTP are two such services. If packet sniffing software is monitoring traffic between the remote user and such a service user names and passwords can be easily intercepted.

Inherently, such services can also more easily fall prey to what the security industry terms the *man-in-the-middle* attack. In this type of attack, a cracker redirects network traffic by tricking a cracked name server on the network to point to his machine instead of the intended server. Once someone opens a remote session to the server, the attacker’s machine acts as an invisible conduit, sitting quietly between the remote service and the unsuspecting user capturing information. In this way a cracker can gather administrative passwords and raw data without the server or the user realizing it.

Another category of insecure services include network file systems and information services such as NFS or NIS, which are developed explicitly for LAN usage but are, unfortunately, extended to include WANs (for remote users). NFS does not, by default, have any authentication or security mechanisms configured to prevent a cracker from mounting the NFS share and accessing anything contained therein. NIS, as well, has vital information that must be known by every computer on a network, including passwords and file permissions, within a plain text ASCII or DBM (ASCII-derived) database. A cracker who gains access to this database can then access every user account on a network, including the administrator’s account.

By default, Red Hat Enterprise Linux 8 is released with all such services turned off. However, since administrators often find themselves forced to use these services, careful configuration is critical.

**11.6.3. Threats to workstation and home PC security**

Workstations and home PCs may not be as prone to attack as networks or servers, but because they often contain sensitive data, such as credit card information, they are targeted by system crackers. Workstations can also be co-opted without the user’s knowledge and used by attackers as “bot” machines in coordinated attacks. For these reasons, knowing the vulnerabilities of a workstation can save users the headache of reinstalling the operating system, or worse, recovering from data theft.

**Bad passwords**

Bad passwords are one of the easiest ways for an attacker to gain access to a system.

**Vulnerable client applications**
Although an administrator may have a fully secure and patched server, that does not mean remote users are secure when accessing it. For instance, if the server offers Telnet or FTP services over a public network, an attacker can capture the plain text user names and passwords as they pass over the network, and then use the account information to access the remote user’s workstation.

Even when using secure protocols, such as SSH, a remote user may be vulnerable to certain attacks if they do not keep their client applications updated. For instance, SSH protocol version 1 clients are vulnerable to an X-forwarding attack from malicious SSH servers. Once connected to the server, the attacker can quietly capture any keystrokes and mouse clicks made by the client over the network. This problem was fixed in the SSH version 2 protocol, but it is up to the user to keep track of what applications have such vulnerabilities and update them as necessary.

11.7. COMMON EXPLOITS AND ATTACKS

The following table details some of the most common exploits and entry points used by intruders to access organizational network resources. Key to these common exploits are the explanations of how they are performed and how administrators can properly safeguard their network against such attacks.

Table 11.1. Common exploits

<table>
<thead>
<tr>
<th>Exploit</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null or default passwords</td>
<td>Leaving administrative passwords blank or using a default password set by the product vendor. This is most common in hardware such as routers and firewalls, but some services that run on Linux can contain default administrator passwords as well (though Red Hat Enterprise Linux 8 does not ship with them).</td>
<td>Commonly associated with networking hardware such as routers, firewalls, VPNs, and network attached storage (NAS) appliances. Common in many legacy operating systems, especially those that bundle services (such as UNIX and Windows.) Administrators sometimes create privileged user accounts in a rush and leave the password null, creating a perfect entry point for malicious users who discover the account.</td>
</tr>
<tr>
<td>Default shared keys</td>
<td>Secure services sometimes package default security keys for development or evaluation testing purposes. If these keys are left unchanged and are placed in a production environment on the Internet, all users with the same default keys have access to that shared-key resource, and any sensitive information that it contains.</td>
<td>Most common in wireless access points and preconfigured secure server appliances.</td>
</tr>
<tr>
<td>Exploit</td>
<td>Description</td>
<td>Notes</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IP spoofing</td>
<td>A remote machine acts as a node on your local network, finds vulnerabilities with your servers, and installs a backdoor program or Trojan horse to gain control over your network resources.</td>
<td>Spoofing is quite difficult as it involves the attacker predicting TCP/IP sequence numbers to coordinate a connection to target systems, but several tools are available to assist crackers in performing such a vulnerability. Depends on target system running services (such as <code>rsh</code>, <code>telnet</code>, FTP and others) that use source-based authentication techniques, which are not recommended when compared to PKI or other forms of encrypted authentication used in <code>ssh</code> or <code>SSL/TLS</code>.</td>
</tr>
<tr>
<td>Eavesdropping</td>
<td>Collecting data that passes between two active nodes on a network by eavesdropping on the connection between the two nodes.</td>
<td>This type of attack works mostly with plain text transmission protocols such as Telnet, FTP, and HTTP transfers. Remote attacker must have access to a compromised system on a LAN in order to perform such an attack; usually the cracker has used an active attack (such as IP spoofing or man-in-the-middle) to compromise a system on the LAN. Preventative measures include services with cryptographic key exchange, one-time passwords, or encrypted authentication to prevent password snooping; strong encryption during transmission is also advised.</td>
</tr>
<tr>
<td>Exploit</td>
<td>Description</td>
<td>Notes</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Service vulnerabilities</td>
<td>An attacker finds a flaw or loophole in a service run over the Internet; through this vulnerability, the attacker compromises the entire system and any data that it may hold, and could possibly compromise other systems on the network.</td>
<td>HTTP-based services such as CGI are vulnerable to remote command execution and even interactive shell access. Even if the HTTP service runs as a non-privileged user such as “nobody”, information such as configuration files and network maps can be read, or the attacker can start a denial of service attack which drains system resources or renders it unavailable to other users. Services sometimes can have vulnerabilities that go unnoticed during development and testing; these vulnerabilities (such as buffer overflows, where attackers crash a service using arbitrary values that fill the memory buffer of an application, giving the attacker an interactive command prompt from which they may execute arbitrary commands) can give complete administrative control to an attacker. Administrators should make sure that services do not run as the root user, and should stay vigilant of patches and errata updates for applications from vendors or security organizations such as CERT and CVE.</td>
</tr>
</tbody>
</table>
### Application vulnerabilities

<table>
<thead>
<tr>
<th>Exploit</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application vulnerabilities</td>
<td>Attackers find faults in desktop and workstation applications (such as email clients) and execute arbitrary code, implant Trojan horses for future compromise, or crash systems. Further exploitation can occur if the compromised workstation has administrative privileges on the rest of the network.</td>
<td>Workstations and desktops are more prone to exploitation as workers do not have the expertise or experience to prevent or detect a compromise; it is imperative to inform individuals of the risks they are taking when they install unauthorized software or open unsolicited email attachments. Safeguards can be implemented such that email client software does not automatically open or execute attachments. Additionally, the automatic update of workstation software using Red Hat Network; or other system management services can alleviate the burdens of multi-seat security deployments.</td>
</tr>
</tbody>
</table>

### Denial of Service (DoS) attacks

<table>
<thead>
<tr>
<th>Exploit</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denial of Service (DoS) attacks</td>
<td>Attacker or group of attackers coordinate against an organization's network or server resources by sending unauthorized packets to the target host (either server, router, or workstation). This forces the resource to become unavailable to legitimate users.</td>
<td>The most reported DoS case in the US occurred in 2000. Several highly-trafficked commercial and government sites were rendered unavailable by a coordinated ping flood attack using several compromised systems with high bandwidth connections acting as zombies, or redirected broadcast nodes. Source packets are usually forged (as well as rebroadcast), making investigation as to the true source of the attack difficult. Advances in ingress filtering (RFC 2267) using the nftables packet-filtering framework and Network Intrusion Detection Systems such as snort assist administrators in tracking down and preventing distributed DoS attacks.</td>
</tr>
</tbody>
</table>
CHAPTER 12. SECURING RHEL DURING INSTALLATION

Security begins even before you start the installation of Red Hat Enterprise Linux. Configuring your system securely from the beginning makes it easier to implement additional security settings later.

12.1. BIOS AND UEFI SECURITY

Password protection for the BIOS (or BIOS equivalent) and the boot loader can prevent unauthorized users who have physical access to systems from booting using removable media or obtaining root privileges through single user mode. The security measures you should take to protect against such attacks depends both on the sensitivity of the information on the workstation and the location of the machine.

For example, if a machine is used in a trade show and contains no sensitive information, then it may not be critical to prevent such attacks. However, if an employee's laptop with private, unencrypted SSH keys for the corporate network is left unattended at that same trade show, it could lead to a major security breach with ramifications for the entire company.

If the workstation is located in a place where only authorized or trusted people have access, however, then securing the BIOS or the boot loader may not be necessary.

12.1.1. BIOS passwords

The two primary reasons for password protecting the BIOS of a computer are:

1. Preventing changes to BIOS settings – If an intruder has access to the BIOS, they can set it to boot from a CD-ROM or a flash drive. This makes it possible for them to enter rescue mode or single user mode, which in turn allows them to start arbitrary processes on the system or copy sensitive data.

2. Preventing system booting – Some BIOSes allow password protection of the boot process. When activated, an attacker is forced to enter a password before the BIOS launches the boot loader.

Because the methods for setting a BIOS password vary between computer manufacturers, consult the computer’s manual for specific instructions.

If you forget the BIOS password, it can either be reset with jumpers on the motherboard or by disconnecting the CMOS battery. For this reason, it is good practice to lock the computer case if possible. However, consult the manual for the computer or motherboard before attempting to disconnect the CMOS battery.

12.1.2. Non-BIOS-based systems security

Other systems and architectures use different programs to perform low-level tasks roughly equivalent to those of the BIOS on x86 systems. For example, the Unified Extensible Firmware Interface (UEFI) shell.

For instructions on password protecting BIOS-like programs, see the manufacturer’s instructions.

12.2. DISK PARTITIONING

Red Hat recommends creating separate partitions for the /boot, /, /home, /tmp, and /var/tmp/ directories.
/boot

This partition is the first partition that is read by the system during boot up. The boot loader and kernel images that are used to boot your system into Red Hat Enterprise Linux 8 are stored in this partition. This partition should not be encrypted. If this partition is included in / and that partition is encrypted or otherwise becomes unavailable then your system is not able to boot.

/home

When user data (/home) is stored in / instead of in a separate partition, the partition can fill up causing the operating system to become unstable. Also, when upgrading your system to the next version of Red Hat Enterprise Linux 8 it is a lot easier when you can keep your data in the /home partition as it is not be overwritten during installation. If the root partition (/) becomes corrupt your data could be lost forever. By using a separate partition there is slightly more protection against data loss. You can also target this partition for frequent backups.

/tmp and /var/tmp/

Both the /tmp and /var/tmp/ directories are used to store data that does not need to be stored for a long period of time. However, if a lot of data floods one of these directories it can consume all of your storage space. If this happens and these directories are stored within / then your system could become unstable and crash. For this reason, moving these directories into their own partitions is a good idea.

NOTE

During the installation process, you have an option to encrypt partitions. You must supply a passphrase. This passphrase serves as a key to unlock the bulk encryption key, which is used to secure the partition’s data.

12.3. RESTRICTING NETWORK CONNECTIVITY DURING THE INSTALLATION PROCESS

When installing Red Hat Enterprise Linux 8, the installation medium represents a snapshot of the system at a particular time. Because of this, it may not be up-to-date with the latest security fixes and may be vulnerable to certain issues that were fixed only after the system provided by the installation medium was released.

When installing a potentially vulnerable operating system, always limit exposure only to the closest necessary network zone. The safest choice is the “no network” zone, which means to leave your machine disconnected during the installation process. In some cases, a LAN or intranet connection is sufficient while the Internet connection is the riskiest. To follow the best security practices, choose the closest zone with your repository while installing Red Hat Enterprise Linux 8 from a network.

12.4. INSTALLING THE MINIMUM AMOUNT OF PACKAGES REQUIRED

It is best practice to install only the packages you will use because each piece of software on your computer could possibly contain a vulnerability. If you are installing from the DVD media, take the opportunity to select exactly what packages you want to install during the installation. If you find you need another package, you can always add it to the system later.

12.5. POST-INSTALLATION PROCEDURES

The following steps are the security-related procedures that should be performed immediately after installation of Red Hat Enterprise Linux 8.

- Update your system. Enter the following command as root:
Even though the firewall service, `firewalld`, is automatically enabled with the installation of Red Hat Enterprise Linux, there are scenarios where it might be explicitly disabled, for example in the kickstart configuration. In such a case, it is recommended to consider re-enabling the firewall.

To start `firewalld` enter the following commands as root:

```bash
# systemctl start firewalld
# systemctl enable firewalld
```

To enhance security, disable services you do not need. For example, if there are no printers installed on your computer, disable the `cups` service using the following command:

```bash
# systemctl disable cups
```

To review active services, enter the following command:

```bash
$ systemctl list-units | grep service
```

---

[1] Since system BIOSes differ between manufacturers, some may not support password protection of either type, while others may support one type but not the other.
CHAPTER 13. USING SYSTEM-WIDE CRYPTOGRAPHIC POLICIES

The system-wide cryptographic policies is a system component that configures the core cryptographic subsystems, covering the TLS, IPSec, SSH, DNSSec, and Kerberos protocols. It provides a small set of policies, which the administrator can select.

13.1. SYSTEM-WIDE CRYPTOGRAPHIC POLICIES

When a system-wide policy is set up, applications in RHEL follow it and refuse to use algorithms and protocols that do not meet the policy, unless you explicitly request the application to do so. That is, the policy applies to the default behavior of applications when running with the system-provided configuration but you can override it if required.

RHEL 8 contains the following predefined policies:

<table>
<thead>
<tr>
<th>Policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT</td>
<td>The default system-wide cryptographic policy level offers secure settings for current threat models. It allows the TLS 1.2 and 1.3 protocols, as well as the IKEv2 and SSH2 protocols. The RSA keys and Diffie-Hellman parameters are accepted if they are at least 2048 bits long.</td>
</tr>
<tr>
<td>LEGACY</td>
<td>This policy ensures maximum compatibility with Red Hat Enterprise Linux 5 and earlier; it is less secure due to an increased attack surface. In addition to the DEFAULT level algorithms and protocols, it includes support for the TLS 1.0 and 1.1 protocols. The algorithms DSA, 3DES, and RC4 are allowed, while RSA keys and Diffie-Hellman parameters are accepted if they are at least 1023 bits long.</td>
</tr>
<tr>
<td>FUTURE</td>
<td>A conservative security level that is believed to withstand any near-term future attacks. This level does not allow the use of SHA-1 in signature algorithms. It allows the TLS 1.2 and 1.3 protocols, as well as the IKEv2 and SSH2 protocols. The RSA keys and Diffie-Hellman parameters are accepted if they are at least 3072 bits long.</td>
</tr>
<tr>
<td>FIPS</td>
<td>A policy level that conforms with the FIPS 140-2 requirements. This is used internally by the fips-mode-setup tool, which switches the RHEL system into FIPS mode.</td>
</tr>
</tbody>
</table>

Red Hat continuously adjusts all policy levels so that all libraries, except when using the LEGACY policy, provide secure defaults. Even though the LEGACY profile does not provide secure defaults, it does not include any algorithms that are easily exploitable. As such, the set of enabled algorithms or acceptable key sizes in any provided policy may change during the lifetime of Red Hat Enterprise Linux.

Such changes reflect new security standards and new security research. If you must ensure interoperability with a specific system for the whole lifetime of Red Hat Enterprise Linux, you should opt-out from cryptographic-policies for components that interact with that system or re-enable specific algorithms using custom policies.
IMPORTANT

Because a cryptographic key used by a certificate on the Customer Portal API does not meet the requirements by the FUTURE system-wide cryptographic policy, the redhat-support-tool utility does not work with this policy level at the moment.

To work around this problem, use the DEFAULT crypto policy while connecting to the Customer Portal API.

NOTE

The specific algorithms and ciphers described in the policy levels as allowed are available only if an application supports them.

Tool for managing crypto policies
To view or change the current system-wide cryptographic policy, use the update-crypto-policies tool, for example:

```
$ update-crypto-policies --show
DEFAULT
# update-crypto-policies --set FUTURE
Setting system policy to FUTURE
```

To ensure that the change of the cryptographic policy is applied, restart the system.

Strong crypto defaults by removing insecure cipher suites and protocols
The following list contains cipher suites and protocols removed from the core cryptographic libraries in Red Hat Enterprise Linux 8. They are not present in the sources, or their support is disabled during the build, so applications cannot use them.

- DES (since RHEL 7)
- All export grade cipher suites (since RHEL 7)
- MD5 in signatures (since RHEL 7)
- SSLv2 (since RHEL 7)
- SSLv3 (since RHEL 8)
- All ECC curves < 224 bits (since RHEL 6)
- All binary field ECC curves (since RHEL 6)

Cipher suites and protocols disabled in all policy levels
The following cipher suites and protocols are disabled in all crypto policy levels. They can be enabled only by an explicit configuration of individual applications.

- DH with parameters < 1024 bits
- RSA with key size < 1024 bits
- Camellia
- ARIA
- SEED
- IDEA
- Integrity-only cipher suites
- TLS CBC mode cipher suites using SHA-384 HMAC
- AES-CCM8
- All ECC curves incompatible with TLS 1.3, including secp256k1
- IKEv1 (since RHEL 8)

Cipher suites and protocols enabled in the crypto-policies levels
The following table shows the enabled cipher suites and protocols in all four crypto-policies levels.

<table>
<thead>
<tr>
<th></th>
<th>LEGACY</th>
<th>DEFAULT</th>
<th>FIPS</th>
<th>FUTURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>IKEv1</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>3DES</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>RC4</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>DH</td>
<td>min. 1024-bit</td>
<td>min. 2048-bit</td>
<td>min. 2048-bit</td>
<td>min. 3072-bit</td>
</tr>
<tr>
<td>RSA</td>
<td>min. 1024-bit</td>
<td>min. 2048-bit</td>
<td>min. 2048-bit</td>
<td>min. 3072-bit</td>
</tr>
<tr>
<td>DSA</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>TLS v1.0</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>TLS v1.1</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>SHA-1 in digital signatures</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>CBC mode ciphers</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Symmetric ciphers with keys &lt; 256 bits</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>SHA-1 and SHA-224 signatures in certificates</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>
CBC ciphers are disabled for TLS. In a non-TLS scenario, AES-128-CBC is disabled but AES-256-CBC is enabled. To disable also AES-256-CBC, apply a custom subpolicy.

Additional resources

- [update-crypto-policies(8)](man) man page

### 13.2. SWITCHING THE SYSTEM-WIDE CRYPTOGRAPHIC POLICY TO MODE COMPATIBLE WITH EARLIER RELEASES

The default system-wide cryptographic policy in Red Hat Enterprise Linux 8 does not allow communication using older, insecure protocols. For environments that require to be compatible with Red Hat Enterprise Linux 6 and in some cases also with earlier releases, the less secure [LEGACY](default) policy level is available.

**WARNING**

Switching to the [LEGACY](default) policy level results in a less secure system and applications.

### Procedure

1. To switch the system-wide cryptographic policy to the **LEGACY** level, enter the following command as [root](users):

   ```bash
   # update-crypto-policies --set LEGACY
   Setting system policy to LEGACY
   ```

Additional resources

- For the list of available cryptographic policy levels, see the [update-crypto-policies(8)](man) man page.
- For defining custom cryptographic policies, see the [Custom Policies](crypto-policies) section in the [update-crypto-policies(8)](man) man page and the [Crypto Policy Definition Format](crypto-policies) section in the [crypto-policies(7)](man) man page.

### 13.3. SWITCHING THE SYSTEM TO FIPS MODE

The system-wide cryptographic policies contain a policy level that enables cryptographic modules self-checks in accordance with the requirements by the Federal Information Processing Standard (FIPS) Publication 140–2. The [fips-mode-setup](fips-mode) tool that enables or disables FIPS mode internally uses the [FIPS](fips) system-wide cryptographic policy level.
IMPORTANT

Red Hat recommends installing Red Hat Enterprise Linux 8 with FIPS mode enabled, as opposed to enabling FIPS mode later. Enabling FIPS mode during the installation ensures that the system generates all keys with FIPS-approved algorithms and continuous monitoring tests in place.

Procedure

1. To switch the system to FIPS mode:

   ```
   # fips-mode-setup --enable
   Kernel initramdisks are being regenerated. This might take some time.
   Setting system policy to FIPS
   Note: System-wide crypto policies are applied on application start-up.
   It is recommended to restart the system for the change of policies to fully take place.
   FIPS mode will be enabled.
   Please reboot the system for the setting to take effect.
   ```

2. Restart your system to allow the kernel to switch to FIPS mode:

   ```
   # reboot
   ```

Verification

1. After the restart, you can check the current state of FIPS mode:

   ```
   # fips-mode-setup --check
   FIPS mode is enabled.
   ```

Additional resources

- `fips-mode-setup(8)` man page
- Installing a RHEL 8 system with FIPS mode enabled
- List of RHEL applications using cryptography that is not compliant with FIPS 140-2
- Security Requirements for Cryptographic Modules on the National Institute of Standards and Technology (NIST) web site.

13.4. ENABLING FIPS MODE IN A CONTAINER

On systems with FIPS mode enabled, the `podman` utility automatically configures containers to FIPS mode. On systems not in FIPS mode, you can switch a container to FIPS mode by using a single command later.

NOTE

The `fips-mode-setup` command does not work correctly in containers, and it cannot be used to enable or check FIPS mode in this scenario.
Prerequisites

- The host system must be in FIPS mode.

Procedure

- Use the following command in a container that you want to switch to FIPS mode:

  ```bash
  # mount --bind /usr/share/crypto-policies/back-ends/FIPS /etc/crypto-policies/back-ends
  ```

  **NOTE**

  On a RHEL 8.1 system, you can enable FIPS mode in a container by performing the following steps:

  1. Switch the host system to FIPS mode.

  2. Mount the `/etc/system-fips` file on the container from the host.

  3. Set the FIPS cryptographic policy level in the container:

     ```bash
     $ update-crypto-policies --set FIPS
     ```

Additional resources

- Switching the system to FIPS mode.

- Installing a RHEL 8 system with FIPS mode enabled

### 13.5. LIST OF RHEL APPLICATIONS USING CRYPTOGRAPHY THAT IS NOT COMPLIANT WITH FIPS 140-2

Red Hat recommends to utilize libraries from the core crypto components set, as they are guaranteed to pass all relevant crypto certifications, such as FIPS 140-2, and also follow the RHEL system-wide crypto policies.

See the RHEL 8 core crypto components article for an overview of the RHEL 8 core crypto components, the information on how they are selected, how they are integrated into the operating system, how do they support hardware security modules and smart cards, and how do crypto certifications apply to them.

In addition to the following table, in some RHEL 8 Z-stream releases (for example, 8.1.1), the Firefox browser packages have been updated, and they contain a separate copy of the NSS cryptography library. This way, Red Hat wants to avoid the disruption of rebasing such a low-level component in a patch release. As a result, these Firefox packages do not use a FIPS 140-2-validated module.

<table>
<thead>
<tr>
<th>Application</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeRADIUS</td>
<td>The RADIUS protocol uses MD5</td>
</tr>
</tbody>
</table>
### Application

<table>
<thead>
<tr>
<th>Application</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>ghostscript</td>
<td>Custom cryptography implementation (MD5, RC4, SHA-2, AES) to encrypt and decrypt documents</td>
</tr>
<tr>
<td>ipxe</td>
<td>Crypto stack for TLS is compiled in, however, it is unused</td>
</tr>
<tr>
<td>libica</td>
<td>Software fallbacks for various algorithms such as RSA and ECDH through CPACF instructions</td>
</tr>
<tr>
<td>Ovmf (UEFI firmware), Edk2, shim</td>
<td>Full crypto stack (an embedded copy of the OpenSSL library)</td>
</tr>
<tr>
<td>perl-Digest-HMAC</td>
<td>HMAC, HMAC-SHA1, HMAC-MD5</td>
</tr>
<tr>
<td>perl-Digest-SHA</td>
<td>SHA-1, SHA-224, ...</td>
</tr>
<tr>
<td>pidgin</td>
<td>DES, RC4</td>
</tr>
<tr>
<td>qatengine</td>
<td>Mixed hardware and software implementation of cryptographic primitives (RSA, EC, DH, AES, ...)</td>
</tr>
<tr>
<td>samba[a]</td>
<td>AES, DES, RC4</td>
</tr>
<tr>
<td>valgrind</td>
<td>AES, hashes[b]</td>
</tr>
</tbody>
</table>

\[a\] Starting with RHEL 8.3, samba uses FIPS-compliant cryptography.

\[b\] Re-implements in software hardware-offload operations, such as AES-NI.

### 13.6. EXCLUDING AN APPLICATION FROM FOLLOWING SYSTEM-WIDE CRYPTO POLICIES

You can customize cryptographic settings used by your application preferably by configuring supported cipher suites and protocols directly in the application.

You can also remove a symlink related to your application from the `/etc/crypto-policies/back-ends` directory and replace it with your customized cryptographic settings. This configuration prevents the use of system-wide cryptographic policies for applications that use the excluded back end. Furthermore, this modification is not supported by Red Hat.

#### 13.6.1. Examples of opting out of system-wide crypto policies

**wget**

To customize cryptographic settings used by the `wget` network downloader, use `--secure-protocol` and `--ciphers` options. For example:
$ wget --secure-protocol=TLSv1_1 --ciphers="SECURE128" https://example.com

See the HTTPS (SSL/TLS) Options section of the wget(1) man page for more information.

curl

To specify ciphers used by the curl tool, use the --ciphers option and provide a colon-separated list of ciphers as a value. For example:

$ curl https://example.com --ciphers '@SECLEVEL=0:DES-CBC3-SHA:RSA-DES-CBC3-SHA'

See the curl(1) man page for more information.

Firefox

Even though you cannot opt out of system-wide cryptographic policies in the Firefox web browser, you can further restrict supported ciphers and TLS versions in Firefox’s Configuration Editor. Type about:config in the address bar and change the value of the security.tls.version.min option as required. Setting security.tls.version.min to 1 allows TLS 1.0 as the minimum required, security.tls.version.min 2 enables TLS 1.1, and so on.

OpenSSH

To opt out of the system-wide crypto policies for your OpenSSH server, uncomment the line with the CRYPTO_POLICY= variable in the /etc/sysconfig/sshd file. After this change, values that you specify in the Ciphers, MACs, KexAlgorithms, and GSSAPIKexAlgorithms sections in the /etc/ssh/sshd_config file are not overridden. See the sshd_config(5) man page for more information.

To opt out of system-wide crypto policies for your OpenSSH client, perform one of the following tasks:

- For a given user, override the global ssh_config with a user-specific configuration in the ~/.ssh/config file.

- For the entire system, specify the crypto policy in a drop-in configuration file located in the /etc/ssh/sshd_config.d/ directory, with a two-digit number prefix smaller than 50, so that it lexicographically precedes the 50-redhat.conf file, and with a .conf suffix, for example, 49-crypto-policy-override.conf.

See the ssh_config(5) man page for more information.

Libreswan

See the Configuring IPsec connections that opt out of the system-wide crypto policies in the Securing networks document for detailed information.

Additional resources

- update-crypto-policies(8) man page

13.7. CUSTOMIZING SYSTEM-WIDE CRYPTOGRAPHIC POLICIES WITH SUBPOLICIES

Use this procedure to adjust the set of enabled cryptographic algorithms or protocols.

You can either apply custom subpolicies on top of an existing system-wide cryptographic policy or define such a policy from scratch.
The concept of scoped policies allows enabling different sets of algorithms for different back ends. You can limit each configuration directive to specific protocols, libraries, or services.

Furthermore, directives can use asterisks for specifying multiple values using wildcards.

**NOTE**
Customization of system-wide cryptographic policies is available from RHEL 8.2. You can use the concept of scoped policies and the option of using wildcards in RHEL 8.5 and newer.

**Procedure**

1. Checkout to the `/etc/crypto-policies/policies/modules/` directory:

   ```bash
   # cd /etc/crypto-policies/policies/modules/
   ```

2. Create subpolicies for your adjustments, for example:

   ```bash
   # touch MYCRYPTO-1.pmod
   # touch SCOPES-AND-WILDCARDS.pmod
   ```

   **IMPORTANT**
   Use upper-case letters in file names of policy modules.

3. Open the policy modules in a text editor of your choice and insert options that modify the system-wide cryptographic policy, for example:

   ```bash
   # vi MYCRYPTO-1.pmod
   ```

   ```bash
   min_rsa_size = 3072
   hash = SHA2-384 SHA2-512 SHA3-384 SHA3-512
   ```

   ```bash
   # vi SCOPES-AND-WILDCARDS.pmod
   ```

   ```bash
   # Disable the AES-128 cipher, all modes
   cipher = -AES-128-*

   # Disable CHACHA20-POLY1305 for the TLS protocol (OpenSSL, GnuTLS, NSS, and OpenJDK)
   cipher@TLS = -CHACHA20-POLY1305

   # Allow using the FFDHE-1024 group with the SSH protocol (libssh and OpenSSH)
   group@SSH = FFDHE-1024+

   # Disable all CBC mode ciphers for the SSH protocol (libssh and OpenSSH)
   cipher@SSH = -*-CBC

   # Allow the AES-256-CBC cipher in applications using libssh
   cipher@libssh = AES-256-CBC+
   ```
4. Save the changes in the module files.

5. Apply your policy adjustments to the `DEFAULT` system-wide cryptographic policy level:

   ```
   # update-crypto-policies --set DEFAULT:MYCRYPTO-1:SCOPES-AND-WILDCARDS
   ```

6. To make your cryptographic settings effective for already running services and applications, restart the system:

   ```
   # reboot
   ```

Additional resources

- Custom Policies section in the `update-crypto-policies(8)` man page
- Crypto Policy Definition Format section in the `crypto-policies(7)` man page
- How to customize crypto policies in RHEL 8.2 Red Hat blog article

13.8. DISABLING SHA-1 BY CUSTOMIZING A SYSTEM-WIDE CRYPTOGRAPHIC POLICY

Because the SHA-1 hash function has an inherently weak design, and advancing cryptanalysis has made it vulnerable to attacks, RHEL 8 does not use SHA-1 by default. Nevertheless, some third party applications, for example public signatures, still use SHA-1. To disable the use of SHA-1 in signature algorithms on your system, you can use the `NO-SHA1` policy module.

**IMPORTANT**

The `NO-SHA1` policy module disables the SHA-1 hash function only in signatures and not elsewhere. In particular, the `NO-SHA1` module still allows the use of SHA-1 with hash-based message authentication codes (HMAC). This is because HMAC security properties do not rely on collision resistance of the corresponding hash function, and therefore the recent attacks on SHA-1 have a significantly lower impact on the use of SHA-1 for HMAC.

If your scenario requires disabling a specific key exchange (kex) algorithm combination, for example `diffie-hellman-group-exchange-sha1`, but you still want to use both the relevant kex and the algorithm in other combinations, see Steps to disable the `diffie-hellman-group1-sha1` algorithm in SSH for instructions on opting out of system-wide crypto-policies for SSH and configuring SSH directly.

**NOTE**

The module for disabling SHA-1 is available from RHEL 8.3. Customization of system-wide cryptographic policies is available from RHEL 8.2.

Procedure

1. Apply your policy adjustments to the `DEFAULT` system-wide cryptographic policy level:

   ```
   # update-crypto-policies --set DEFAULT:NO-SHA1
   ```
2. To make your cryptographic settings effective for already running services and applications, restart the system:

```
# reboot
```

Additional resources

- **Custom Policies** section in the `update-crypto-policies(8)` man page.
- The **Crypto Policy Definition Format** section in the `crypto-policies(7)` man page.
- [How to customize crypto policies in RHEL 8.2](https://access.redhat.com/blogs/) Red Hat blog article.

### 13.9. CREATING AND SETTING A CUSTOM SYSTEM-WIDE CRYPTOGRAPHIC POLICY

The following steps demonstrate customizing the system-wide cryptographic policies by a complete policy file.

#### NOTE

Customization of system-wide cryptographic policies is available from RHEL 8.2.

**Procedure**

1. Create a policy file for your customizations:

```
# cd /etc/crypto-policies/policies/
# touch MYPOLICY.pol
```

Alternatively, start by copying one of the four predefined policy levels:

```
# cp /usr/share/crypto-policies/policies/DEFAULT.pol /etc/crypto-policies/policies/MYPOLICY.pol
```

2. Edit the file with your custom cryptographic policy in a text editor of your choice to fit your requirements, for example:

```
# vi /etc/crypto-policies/policies/MYPOLICY.pol
```

3. Switch the system-wide cryptographic policy to your custom level:

```
# update-crypto-policies --set MYPOLICY
```

4. To make your cryptographic settings effective for already running services and applications, restart the system:

```
# reboot
```

**Additional resources**

- [Custom Policies](#)
- [Crypto Policy Definition Format](#)
- [How to customize crypto policies in RHEL 8.2](https://access.redhat.com/blogs/) Red Hat blog article.
- **Custom Policies** section in the `update-crypto-policies(8)` man page and the **Crypto Policy Definition Format** section in the `crypto-policies(7)` man page

- **How to customize crypto policies in RHEL** Red Hat blog article

### 13.10. ADDITIONAL RESOURCES

- **System-wide crypto policies in RHEL 8** and **Strong crypto defaults in RHEL 8 and deprecation of weak crypto algorithms** Knowledgebase articles
CHAPTER 14. CONFIGURING APPLICATIONS TO USE CRYPTOGRAPHIC HARDWARE THROUGH PKCS #11

Separating parts of your secret information on dedicated cryptographic devices, such as smart cards and cryptographic tokens for end-user authentication and hardware security modules (HSM) for server applications, provides an additional layer of security. In RHEL, support for cryptographic hardware through the PKCS #11 API is consistent across different applications, and the isolation of secrets on cryptographic hardware is not a complicated task.

14.1. CRYPTOGRAPHIC HARDWARE SUPPORT THROUGH PKCS #11

PKCS #11 (Public-Key Cryptography Standard) defines an application programming interface (API) to cryptographic devices that hold cryptographic information and perform cryptographic functions. These devices are called tokens, and they can be implemented in a hardware or software form.

A PKCS #11 token can store various object types including a certificate; a data object; and a public, private, or secret key. These objects are uniquely identifiable through the PKCS #11 URI scheme.

A PKCS #11 URI is a standard way to identify a specific object in a PKCS #11 module according to the object attributes. This enables you to configure all libraries and applications with the same configuration string in the form of a URI.

RHEL provides the OpenSC PKCS #11 driver for smart cards by default. However, hardware tokens and HSMs can have their own PKCS #11 modules that do not have their counterpart in the system. You can register such PKCS #11 modules with the `p11-kit` tool, which acts as a wrapper over the registered smart card drivers in the system.

To make your own PKCS #11 module work on the system, add a new text file to the `/etc/pkcs11/modules/` directory.

You can add your own PKCS #11 module into the system by creating a new text file in the `/etc/pkcs11/modules/` directory. For example, the OpenSC configuration file in `p11-kit` looks as follows:

```
$ cat /usr/share/p11-kit/modules/opensc.module
module: opensc-pkcs11.so
```

Additional resources

- Consistent PKCS #11 support in Red Hat Enterprise Linux 8
- The PKCS #11 URI Scheme
- Controlling access to smart cards

14.2. USING SSH KEYS STORED ON A SMART CARD

Red Hat Enterprise Linux enables you to use RSA and ECDSA keys stored on a smart card on OpenSSH clients. Use this procedure to enable authentication using a smart card instead of using a password.

Prerequisites

- On the client side, the `opensc` package is installed and the `pcscd` service is running.
Procedure

1. List all keys provided by the OpenSC PKCS #11 module including their PKCS #11 URIs and save the output to the `keys.pub` file:

```
$ ssh-keygen -D pkcs11: > keys.pub
$ ssh-keygen -D pkcs11:
ssh-rsa AAAAB3NzaC1yc2E...KKZMzcQZzx
pkcs11:id=%02;object=SIGN%20pubkey;token=SSH%20key;manufacturer=piv_II?
module-path=/usr/lib64/pkcs11/opensc-pkcs11.so
ecdsa-sha2-nistp256 AAA...J0hkYnnMsM=
pkcs11:id=%01;object=PIV%20AUTH%20pubkey;token=SSH%20key;manufacturer=piv_II?
module-path=/usr/lib64/pkcs11/opensc-pkcs11.so
```

2. To enable authentication using a smart card on a remote server (example.com), transfer the public key to the remote server. Use the `ssh-copy-id` command with `keys.pub` created in the previous step:

```
$ ssh-copy-id -f -i keys.pub username@example.com
```

3. To connect to example.com using the ECDSA key from the output of the `ssh-keygen -D` command in step 1, you can use just a subset of the URI, which uniquely references your key, for example:

```
$ ssh -i "pkcs11:id=%01?module-path=/usr/lib64/pkcs11/opensc-pkcs11.so" example.com
Enter PIN for 'SSH key':
[example.com] $
```

4. You can use the same URI string in the `~/.ssh/config` file to make the configuration permanent:

```
$ cat ~/.ssh/config
IdentityFile "pkcs11:id=%01?module-path=/usr/lib64/pkcs11/opensc-pkcs11.so"
$ ssh example.com
Enter PIN for 'SSH key':
[example.com] $
```

Because OpenSSH uses the `p11-kit-proxy` wrapper and the OpenSC PKCS #11 module is registered to PKCS#11 Kit, you can simplify the previous commands:

```
$ ssh -i "pkcs11:id=%01" example.com
Enter PIN for 'SSH key':
[example.com] $
```

If you skip the `id=` part of a PKCS #11 URI, OpenSSH loads all keys that are available in the proxy module. This can reduce the amount of typing required:

```
$ ssh -i pkcs11: example.com
Enter PIN for 'SSH key':
[example.com] $
```

Additional resources

- Fedora 28: Better smart card support in OpenSSH
14.3. CONFIGURING APPLICATIONS TO AUTHENTICATE USING CERTIFICATES FROM SMART CARDS

Authentication using smart cards in applications may increase security and simplify automation.

- The `wget` network downloader enables you to specify PKCS #11 URIs instead of paths to locally stored private keys, and thus simplifies creating scripts for tasks that require safely stored private keys and certificates. For example:

  ```
  $ wget --private-key 'pkcs11:token=softhsm;id=%01;type=private?pin-value=111111' --
  certificate 'pkcs11:token=softhsm;id=%01;type=cert' https://example.com/
  ```

  See the `wget(1)` man page for more information.

- Specifying PKCS #11 URI for use by the `curl` tool is analogous:

  ```
  $ curl --key 'pkcs11:token=softhsm;id=%01;type=private?pin-value=111111' --cert
  'pkcs11:token=softhsm;id=%01;type=cert' https://example.com/
  ```

  See the `curl(1)` man page for more information.

- The `Firefox` web browser automatically loads the `p11-kit-proxy` module. This means that every supported smart card in the system is automatically detected. For using TLS client authentication, no additional setup is required and keys from a smart card are automatically used when a server requests them.

Using PKCS #11 URIs in custom applications

If your application uses the `GnuTLS` or `NSS` library, support for PKCS #11 URIs is ensured by their built-in support for PKCS #11. Also, applications relying on the `OpenSSL` library can access cryptographic hardware modules thanks to the `openssl-pkcs11` engine.

With applications that require working with private keys on smart cards and that do not use `NSS`, `GnuTLS`, and `OpenSSL`, use `p11-kit` to implement registering PKCS #11 modules.

Additional resources

- `p11-kit(8)` man page.

14.4. USING HSMS PROTECTING PRIVATE KEYS IN APACHE

The `Apache` HTTP server can work with private keys stored on hardware security modules (HSMs), which helps to prevent the keys' disclosure and man-in-the-middle attacks. Note that this usually requires high-performance HSMs for busy servers.

For secure communication in the form of the HTTPS protocol, the `Apache` HTTP server (`httpd`) uses the OpenSSL library. OpenSSL does not support PKCS #11 natively. To utilize HSMs, you have to install the `openssl-pkcs11` package, which provides access to PKCS #11 modules through the engine interface. You can use a PKCS #11 URI instead of a regular file name to specify a server key and a certificate in the `/etc/httpd/conf.d/ssl.conf` configuration file, for example:
Install the **httpd-manual** package to obtain complete documentation for the **Apache** HTTP Server, including TLS configuration. The directives available in the `/etc/httpd/conf.d/ssl.conf` configuration file are described in detail in `/usr/share/httpd/manual/mod/mod_ssl.html`.

### 14.5. USING HSMS PROTECTING PRIVATE KEYS IN NGINX

The **Nginx** HTTP server can work with private keys stored on hardware security modules (HSMs), which helps to prevent the keys’ disclosure and man-in-the-middle attacks. Note that this usually requires high-performance HSMs for busy servers.

Because **Nginx** also uses the OpenSSL for cryptographic operations, support for PKCS #11 must go through the `openssl-pkcs11` engine. **Nginx** currently supports only loading private keys from an HSM, and a certificate must be provided separately as a regular file. Modify the `ssl_certificate` and `ssl_certificate_key` options in the `server` section of the `/etc/nginx/nginx.conf` configuration file:

```plaintext
ssl_certificate     /path/to/cert.pem
ssl_certificate_key "engine:pkcs11:pkcs11:token=softhsm;id=%01;type=private?pin-value=111111;"
```

Note that the `engine:pkcs11:` prefix is needed for the PKCS #11 URI in the **Nginx** configuration file. This is because the other `pkcs11` prefix refers to the engine name.

### 14.6. ADDITIONAL RESOURCES

- `pkcs11.conf(5)` man page.
CHAPTER 15. USING SHARED SYSTEM CERTIFICATES

The shared system certificates storage enables NSS, GnuTLS, OpenSSL, and Java to share a default source for retrieving system certificate anchors and block-list information. By default, the trust store contains the Mozilla CA list, including positive and negative trust. The system allows updating the core Mozilla CA list or choosing another certificate list.

15.1. THE SYSTEM-WIDE TRUST STORE

In Red Hat Enterprise Linux, the consolidated system-wide trust store is located in the /etc/pki/ca-trust/ and /usr/share/pki/ca-trust-source/ directories. The trust settings in /usr/share/pki/ca-trust-source/ are processed with lower priority than settings in /etc/pki/ca-trust/.

Certificate files are treated depending on the subdirectory they are installed to the following directories:

- for trust anchors
  - /usr/share/pki/ca-trust-source/anchors/
  - /etc/pki/ca-trust/source/anchors/
- for distrusted certificates
  - /usr/share/pki/ca-trust-source/blacklist/
  - /etc/pki/ca-trust/source/blacklist/
- for certificates in the extended BEGIN TRUSTED file format
  - /usr/share/pki/ca-trust-source/
  - /etc/pki/ca-trust/source/

NOTE

In a hierarchical cryptographic system, a trust anchor is an authoritative entity which other parties consider being trustworthy. In the X.509 architecture, a root certificate is a trust anchor from which a chain of trust is derived. To enable chain validation, the trusting party must have access to the trust anchor first.

15.2. ADDING NEW CERTIFICATES

To acknowledge applications on your system with a new source of trust, add the corresponding certificate to the system-wide store, and use the update-ca-trust command.

Prerequisites

- The ca-certificates package is present on the system.

Procedure

1. To add a certificate in the simple PEM or DER file formats to the list of CAs trusted on the system, copy the certificate file to the /usr/share/pki/ca-trust-source/anchors/ or /etc/pki/ca-trust/source/anchors/ directory, for example:
# cp ~/certificate-trust-examples/Cert-trust-test-ca.pem /usr/share/pki/ca-trust-source/anchors/

2. To update the system-wide trust store configuration, use the `update-ca-trust` command:

```
# update-ca-trust
```

**NOTE**

While the Firefox browser is able to use an added certificate without executing `update-ca-trust`, Red Hat recommends to use the `update-ca-trust` command after a CA change. Also note that browsers, such as Firefox, Epiphany, or Chromium, cache files, and you might have to clear browser’s cache or restart your browser to load the current system certificates configuration.

15.3. MANAGING TRUSTED SYSTEM CERTIFICATES

The `trust` command provides a convenient way for managing certificates in the shared system-wide trust store.

- To list, extract, add, remove, or change trust anchors, use the `trust` command. To see the built-in help for this command, enter it without any arguments or with the `--help` directive:

```
$ trust
usage: trust command <args>...
Common trust commands are:
list List trust or certificates
extract Extract certificates and trust
extract-compat Extract trust compatibility bundles
anchor Add, remove, change trust anchors
dump Dump trust objects in internal format
See 'trust <command> --help' for more information
```

- To list all system trust anchors and certificates, use the `trust list` command:

```
$ trust list
pkcs11:id=%d2%87%b4%e3%df%37%93%55%f6%56%ea%81%e5%36%cc%8c%1e%3f%bd;type=cert
type: certificate
label: ACCVRAIZ1
trust: anchor
category: authority

pkcs11:id=%a6%b3%e1%2b%2b%49%b6%d7%73%a1%aa%94%f5%01%e7%73%65%4c%ac%50;type=cert
type: certificate
label: ACEDICOM Root
trust: anchor
category: authority
...
• To store a trust anchor into the system-wide trust store, use the `trust anchor` sub-command and specify a path to a certificate. Replace `path.to/certificate.crt` by a path to your certificate and its file name:

```
# trust anchor path.to/certificate.crt
```

• To remove a certificate, use either a path to a certificate or an ID of a certificate:

```
# trust anchor --remove path.to/certificate.crt
# trust anchor --remove "pkcs11:id=%AA%BB%CC%DD%EE;type=cert"
```

**Additional resources**

• All sub-commands of the `trust` commands offer a detailed built-in help, for example:

```
$ trust list --help
usage: trust list --filter=<what>

  --filter=<what>  filter of what to export
    ca-anchors      certificate anchors
...
  --purpose=<usage> limit to certificates usable for the purpose
    server-auth     for authenticating servers
...
```

**15.4. ADDITIONAL RESOURCES**

• `update-ca-trust(8)` and `trust(1)` man pages
CHAPTER 16. SCANNING THE SYSTEM FOR SECURITY COMPLIANCE AND VULNERABILITIES

16.1. CONFIGURATION COMPLIANCE TOOLS IN RHEL

Red Hat Enterprise Linux provides tools that enable you to perform a fully automated compliance audit. These tools are based on the Security Content Automation Protocol (SCAP) standard and are designed for automated tailoring of compliance policies.

- **SCAP Workbench** - The `scap-workbench` graphical utility is designed to perform configuration and vulnerability scans on a single local or remote system. You can also use it to generate security reports based on these scans and evaluations.

- **OpenSCAP** - The OpenSCAP library, with the accompanying `oscap` command-line utility, is designed to perform configuration and vulnerability scans on a local system, to validate configuration compliance content, and to generate reports and guides based on these scans and evaluations.

- **SCAP Security Guide (SSG)** - The `scap-security-guide` package provides the latest collection of security policies for Linux systems. The guidance consists of a catalog of practical hardening advice, linked to government requirements where applicable. The project bridges the gap between generalized policy requirements and specific implementation guidelines.

- **Script Check Engine (SCE)** - SCE is an extension to the SCAP protocol that enables administrators to write their security content using a scripting language, such as Bash, Python, and Ruby. The SCE extension is provided in the `openscap-engine-sce` package. The SCE itself is not part of the SCAP standard.

To perform automated compliance audits on multiple systems remotely, you can use the OpenSCAP solution for Red Hat Satellite.

Additional resources

- `oscap(8), scap-workbench(8), and scap-security-guide(8)` man pages
- Red Hat Security Demos: Creating Customized Security Policy Content to Automate Security Compliance
- Red Hat Security Demos: Defend Yourself with RHEL Security Technologies

16.2. RED HAT SECURITY ADVISORIES OVAL FEED

Red Hat Enterprise Linux security auditing capabilities are based on the Security Content Automation Protocol (SCAP) standard. SCAP is a multi-purpose framework of specifications that supports automated configuration, vulnerability and patch checking, technical control compliance activities, and security measurement.

SCAP specifications create an ecosystem where the format of security content is well-known and standardized although the implementation of the scanner or policy editor is not mandated. This enables organizations to build their security policy (SCAP content) once, no matter how many security vendors they employ.
The Open Vulnerability Assessment Language (OVAL) is the essential and oldest component of SCAP. Unlike other tools and custom scripts, OVAL describes a required state of resources in a declarative manner. OVAL code is never executed directly but using an OVAL interpreter tool called scanner. The declarative nature of OVAL ensures that the state of the assessed system is not accidentally modified.

Like all other SCAP components, OVAL is based on XML. The SCAP standard defines several document formats. Each of them includes a different kind of information and serves a different purpose.


Because of differences between platforms, versions, and other factors, Red Hat Product Security qualitative severity ratings of vulnerabilities do not directly align with the Common Vulnerability Scoring System (CVSS) baseline ratings provided by third parties. Therefore, we recommend that you use the RHSA OVAL definitions instead of those provided by third parties.

The RHSA OVAL definitions are available individually and as a complete package, and are updated within an hour of a new security advisory being made available on the Red Hat Customer Portal.

Each OVAL patch definition maps one-to-one to a Red Hat Security Advisory (RHSA). Because an RHSA can contain fixes for multiple vulnerabilities, each vulnerability is listed separately by its Common Vulnerabilities and Exposures (CVE) name and has a link to its entry in our public bug database.

The RHSA OVAL definitions are designed to check for vulnerable versions of RPM packages installed on a system. It is possible to extend these definitions to include further checks, for example, to find out if the packages are being used in a vulnerable configuration. These definitions are designed to cover software and updates shipped by Red Hat. Additional definitions are required to detect the patch status of third-party software.

Additional resources
- Red Hat and OVAL compatibility
- Red Hat and CVE compatibility
- Notifications and Advisories in the Product Security Overview
- Security Data Metrics

16.3. VULNERABILITY SCANNING

16.3.1. Red Hat Security Advisories OVAL feed

Red Hat Enterprise Linux security auditing capabilities are based on the Security Content Automation Protocol (SCAP) standard. SCAP is a multi-purpose framework of specifications that supports automated configuration, vulnerability and patch checking, technical control compliance activities, and security measurement.

SCAP specifications create an ecosystem where the format of security content is well-known and standardized although the implementation of the scanner or policy editor is not mandated. This enables organizations to build their security policy (SCAP content) once, no matter how many security vendors they employ.

The Open Vulnerability Assessment Language (OVAL) is the essential and oldest component of SCAP.
Unlike other tools and custom scripts, OVAL describes a required state of resources in a declarative manner. OVAL code is never executed directly but using an OVAL interpreter tool called scanner. The declarative nature of OVAL ensures that the state of the assessed system is not accidentally modified.

Like all other SCAP components, OVAL is based on XML. The SCAP standard defines several document formats. Each of them includes a different kind of information and serves a different purpose.


Because of differences between platforms, versions, and other factors, Red Hat Product Security qualitative severity ratings of vulnerabilities do not directly align with the Common Vulnerability Scoring System (CVSS) baseline ratings provided by third parties. Therefore, we recommend that you use the RHSA OVAL definitions instead of those provided by third parties.

The RHSA OVAL definitions are available individually and as a complete package, and are updated within an hour of a new security advisory being made available on the Red Hat Customer Portal.

Each OVAL patch definition maps one-to-one to a Red Hat Security Advisory (RHSA). Because an RHSA can contain fixes for multiple vulnerabilities, each vulnerability is listed separately by its Common Vulnerabilities and Exposures (CVE) name and has a link to its entry in our public bug database.

The RHSA OVAL definitions are designed to check for vulnerable versions of RPM packages installed on a system. It is possible to extend these definitions to include further checks, for example, to find out if the packages are being used in a vulnerable configuration. These definitions are designed to cover software and updates shipped by Red Hat. Additional definitions are required to detect the patch status of third-party software.

Additional resources

- Red Hat and OVAL compatibility
- Red Hat and CVE compatibility
- Notifications and Advisories in the Product Security Overview
- Security Data Metrics

16.3.2. Scanning the system for vulnerabilities

The oscap command-line utility enables you to scan local systems, validate configuration compliance content, and generate reports and guides based on these scans and evaluations. This utility serves as a front end to the OpenSCAP library and groups its functionalities to modules (sub-commands) based on the type of SCAP content it processes.

Prerequisites

- The AppStream repository is enabled.

Procedure

1. Install the openscap-scanner and bzip2 packages:

   ```
   # yum install openscap-scanner bzip2
   ```
2. Download the latest RHSA OVAL definitions for your system:

```bash
```

3. Scan the system for vulnerabilities and save results to the `vulnerability.html` file:

```bash
# oscap oval eval --report vulnerability.html rhel-8.oval.xml
```

**Verification**

- Check the results in a browser of your choice, for example:

```bash
$ firefox vulnerability.html &
```

**Additional resources**

- `oscap(8)` man page
- Red Hat OVAL definitions

16.3.3. Scanning remote systems for vulnerabilities

You can check also remote systems for vulnerabilities with the OpenSCAP scanner using the `oscap-ssh` tool over the SSH protocol.

**Prerequisites**

- The AppStream repository is enabled.
- The `openscap-scanner` package is installed on the remote systems.
- The SSH server is running on the remote systems.

**Procedure**

1. Install the `openscap-utils` and `bzip2` packages:

   ```bash
   # yum install openscap-utils bzip2
   ```

2. Download the latest RHSA OVAL definitions for your system:

   ```bash
   ```

3. Scan a remote system with the `machine1` host name, SSH running on port 22, and the `joesec` user name for vulnerabilities and save results to the `remote-vulnerability.html` file:

   ```bash
   # oscap-ssh joesec@machine1 22 oval eval --report remote-vulnerability.html rhel-8.oval.xml
   ```

**Additional resources**
16.4. CONFIGURATION COMPLIANCE SCANNING

16.4.1. Configuration compliance in RHEL

You can use configuration compliance scanning to conform to a baseline defined by a specific organization. For example, if you work with the US government, you might have to align your systems with the Operating System Protection Profile (OSPP), and if you are a payment processor, you might have to align your systems with the Payment Card Industry Data Security Standard (PCI-DSS). You can also perform configuration compliance scanning to harden your system security.

Red Hat recommends you follow the Security Content Automation Protocol (SCAP) content provided in the SCAP Security Guide package because it is in line with Red Hat best practices for affected components.

The SCAP Security Guide package provides content which conforms to the SCAP 1.2 and SCAP 1.3 standards. The openscap scanner utility is compatible with both SCAP 1.2 and SCAP 1.3 content provided in the SCAP Security Guide package.

IMPORTANT

Performing a configuration compliance scanning does not guarantee the system is compliant.

The SCAP Security Guide suite provides profiles for several platforms in a form of data stream documents. A data stream is a file that contains definitions, benchmarks, profiles, and individual rules. Each rule specifies the applicability and requirements for compliance. RHEL provides several profiles for compliance with security policies. In addition to the industry standard, Red Hat data streams also contain information for remediation of failed rules.

Structure of compliance scanning resources

A profile is a set of rules based on a security policy, such as OSPP, PCI-DSS, and Health Insurance Portability and Accountability Act (HIPAA). This enables you to audit the system in an automated way for compliance with security standards.

You can modify (tailor) a profile to customize certain rules, for example, password length. For more information on profile tailoring, see Customizing a security profile with SCAP Workbench.
### 16.4.2. Possible results of an OpenSCAP scan

Depending on various properties of your system and the data stream and profile applied to an OpenSCAP scan, each rule may produce a specific result. This is a list of possible results with brief explanations of what they mean.

**Table 16.1. Possible results of an OpenSCAP scan**

<table>
<thead>
<tr>
<th>Result</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pass</td>
<td>The scan did not find any conflicts with this rule.</td>
</tr>
<tr>
<td>Fail</td>
<td>The scan found a conflict with this rule.</td>
</tr>
<tr>
<td>Not checked</td>
<td>OpenSCAP does not perform an automatic evaluation of this rule. Check whether your system conforms to this rule manually.</td>
</tr>
<tr>
<td>Not applicable</td>
<td>This rule does not apply to the current configuration.</td>
</tr>
<tr>
<td>Not selected</td>
<td>This rule is not part of the profile. OpenSCAP does not evaluate this rule and does not display these rules in the results.</td>
</tr>
<tr>
<td>Error</td>
<td>The scan encountered an error. For additional information, you can enter the oscap command with the <code>--verbose DEVEL</code> option. Consider opening a bug report.</td>
</tr>
<tr>
<td>Unknown</td>
<td>The scan encountered an unexpected situation. For additional information, you can enter the oscap command with the <code>--verbose DEVEL</code> option. Consider opening a bug report.</td>
</tr>
</tbody>
</table>

### 16.4.3. Viewing profiles for configuration compliance

Before you decide to use profiles for scanning or remediation, you can list them and check their detailed descriptions using the oscap info sub-command.

**Prerequisites**

- The openscap-scanner and scap-security-guide packages are installed.

**Procedure**

1. List all available files with security compliance profiles provided by the SCAP Security Guide project:

   ```
   $ ls /usr/share/xml/scap/ssg/content/
   ssg-firefox-cpe-dictionary.xml  ssg-rhel6-ocil.xml
   ssg-firefox-cpe-oval.xml      ssg-rhel6-oval.xml
   ...  
   ```
2. Display detailed information about a selected data stream using the `oscap info` sub-command. XML files containing data streams are indicated by the `-ds` string in their names. In the Profiles section, you can find a list of available profiles and their IDs:

```
$ oscap info /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
Profiles:

  Title: Health Insurance Portability and Accountability Act (HIPAA)
  Id: xccdf_org.ssgproject.content_profile_hipaa

  Title: PCI-DSS v3.2.1 Control Baseline for Red Hat Enterprise Linux 8
  Id: xccdf_org.ssgproject.content_profile_pci-dss

  Title: OSPP - Protection Profile for General Purpose Operating Systems
  Id: xccdf_org.ssgproject.content_profile_ospp
```

3. Select a profile from the data-stream file and display additional details about the selected profile. To do so, use `oscap info` with the `--profile` option followed by the last section of the ID displayed in the output of the previous command. For example, the ID of the HIPPA profile is: `xccdf_org.ssgproject.content_profile_hipaa`, and the value for the `--profile` option is `hipaa`:

```
$ oscap info --profile hipaa /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml

Profile
Title: Health Insurance Portability and Accountability Act (HIPAA)

Description: The HIPAA Security Rule establishes U.S. national standards to protect individuals’ electronic personal health information that is created, received, used, or maintained by a covered entity.
```

Additional resources

- `scap-security-guide(8)` man page

16.4.4. Assessing configuration compliance with a specific baseline

To determine whether your system conforms to a specific baseline, follow these steps.

Prerequisites

- The `openscap-scanner` and `scap-security-guide` packages are installed
- You know the ID of the profile within the baseline with which the system should comply. To find the ID, see Viewing Profiles for Configuration Compliance.

Procedure

1. Evaluate the compliance of the system with the selected profile and save the scan results in the report.html HTML file, for example:
$ sudo oscap xccdf eval --report report.html --profile hipaa
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml

2. Optional: Scan a remote system with the machine1 host name, SSH running on port 22, and the joesec user name for compliance and save results to the remote-report.html file:

$ oscap-ssh joesec@machine1 22 xccdf eval --report remote_report.html --profile hipaa
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml

Additional resources

- scap-security-guide(8) man page
- Guide to the Secure Configuration of Red Hat Enterprise Linux 8 installed with the scap-security-guide-doc package

16.5. REMEDIATING THE SYSTEM TO ALIGN WITH A SPECIFIC BASELINE

Use this procedure to remediate the RHEL system to align with a specific baseline. This example uses the Health Insurance Portability and Accountability Act (HIPAA) profile.

**WARNING**

If not used carefully, running the system evaluation with the Remediate option enabled might render the system non-functional. Red Hat does not provide any automated method to revert changes made by security-hardening remediations. Remediations are supported on RHEL systems in the default configuration. If your system has been altered after the installation, running remediation might not make it compliant with the required security profile.

Prerequisites

- The scap-security-guide package is installed on your RHEL system.

Procedure

1. Use the oscap command with the --remediate option:

```bash
$ sudo oscap xccdf eval --profile hipaa --remediate /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

2. Restart your system.

Verification
1. Evaluate compliance of the system with the HIPAA profile, and save scan results in the hipaa_report.html file:

   $ oscap xccdf eval --report hipaa_report.html --profile hipaa /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml

Additional resources

- scap-security-guide(8) and oscap(8) man pages

16.6. REMEDIATING THE SYSTEM TO ALIGN WITH A SPECIFIC BASELINE USING THE SSG ANSIBLE PLAYBOOK

Use this procedure to remediate your system with a specific baseline using the Ansible playbook file from the SCAP Security Guide project. This example uses the Health Insurance Portability and Accountability Act (HIPAA) profile.

WARNING

If not used carefully, running the system evaluation with the Remediate option enabled might render the system non-functional. Red Hat does not provide any automated method to revert changes made by security-hardening remediations. Remediations are supported on RHEL systems in the default configuration. If your system has been altered after the installation, running remediation might not make it compliant with the required security profile.

Prerequisites

- The scap-security-guide package is installed.
- The ansible package is installed. See the Ansible Installation Guide for more information.

NOTE

In RHEL 8.6 and later versions, Ansible Engine is replaced with the ansible-core package, which contains only built-in modules. Note that many Ansible remediations use modules from the community and Portable Operating System Interface (POSIX) collections, which are not included in the built-in modules. In this case, you can use Bash remediations as a substitute to Ansible remediations.

Procedure

1. Remediate your system to align with HIPAA using Ansible:

   # ansible-playbook -i localhost, -c local /usr/share/scap-security-guide/ansible/rhel8-playbook-hipaa.yml

2. Restart the system.
1. Evaluate compliance of the system with the HIPAA profile, and save scan results in the `hipaa_report.html` file:

```
# oscap xccdf eval --profile hipaa --report hipaa_report.html
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

### Additional resources

- `scap-security-guide(8)` and `oscap(8)` man pages
- Ansible Documentation

### 16.7. CREATING A REMEDIATION ANSIBLE PLAYBOOK TO ALIGN THE SYSTEM WITH A SPECIFIC BASELINE

You can create an Ansible playbook containing only the remediations that are required to align your system with a specific baseline. This example uses the Health Insurance Portability and Accountability Act (HIPAA) profile. With this procedure, you create a smaller playbook that does not cover already satisfied requirements. By following these steps, you do not modify your system in any way, you only prepare a file for later application.

**NOTE**

In RHEL 8.6, Ansible Engine is replaced by the `ansible-core` package, which contains only built-in modules. Note that many Ansible remediations use modules from the community and Portable Operating System Interface (POSIX) collections, which are not included in the built-in modules. In this case, you can use Bash remediations as a substitute for Ansible remediations.

### Prerequisites

- The `scap-security-guide` package is installed.

### Procedure

1. Scan the system and save the results:

```
# oscap xccdf eval --profile hipaa --results hipaa-results.xml
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

2. Generate an Ansible playbook based on the file generated in the previous step:

```
# oscap xccdf generate fix --fix-type ansible --profile hipaa --output hipaa-remediations.yml
hipaa-results.xml
```

3. The `hipaa-remediations.yml` file contains Ansible remediations for rules that failed during the scan performed in step 1. After reviewing this generated file, you can apply it with the `ansible-playbook hipaa-remediations.yml` command.

### Verification
In a text editor of your choice, review that the `hipaa-remediations.yml` file contains rules that failed in the scan performed in step 1.

**Additional resources**

- `scap-security-guide(8)` and `oscap(8)` man pages
- Ansible Documentation

### 16.8. CREATING A REMEDIATION BASH SCRIPT FOR A LATER APPLICATION

Use this procedure to create a Bash script containing remediations that align your system with a security profile such as HIPAA. Using the following steps, you do not do any modifications to your system, you only prepare a file for later application.

**Prerequisites**

- The `scap-security-guide` package is installed on your RHEL system.

**Procedure**

1. Use the `oscap` command to scan the system and to save the results to an XML file. In the following example, `oscap` evaluates the system against the `hipaa` profile:

   ```
   # oscap xccdf eval --profile hipaa --results hipaa-results.xml
   /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
   ```

2. Generate a Bash script based on the results file generated in the previous step:

   ```
   # oscap xccdf generate fix --profile hipaa --fix-type bash --output hipaa-remediations.sh
   hipaa-results.xml
   ```

3. The `hipaa-remediations.sh` file contains remediations for rules that failed during the scan performed in step 1. After reviewing this generated file, you can apply it with the `./hipaa-remediations.sh` command when you are in the same directory as this file.

**Verification**

- In a text editor of your choice, review that the `hipaa-remediations.sh` file contains rules that failed in the scan performed in step 1.

**Additional resources**

- `scap-security-guide(8), oscap(8), and bash(1)` man pages

### 16.9. SCANNING THE SYSTEM WITH A CUSTOMIZED PROFILE USING SCAP WORKBENCH

SCAP Workbench, which is contained in the `scap-workbench` package, is a graphical utility that enables users to perform configuration and vulnerability scans on a single local or a remote system, perform remediation of the system, and generate reports based on scan evaluations. Note that SCAP
Workbench has limited functionality compared with the oscap command-line utility. SCAP Workbench processes security content in the form of data-stream files.

16.9.1. Using SCAP Workbench to scan and remediate the system

To evaluate your system against the selected security policy, use the following procedure.

Prerequisites

- The scap-workbench package is installed on your system.

Procedure

1. To run SCAP Workbench from the GNOME Classic desktop environment, press the Super key to enter the Activities Overview, type scap-workbench, and then press Enter. Alternatively, use:

```
$ scap-workbench &
```

2. Select a security policy using either of the following options:
   - Load Content button on the starting window
   - Open content from SCAP Security Guide
   - Open Other Content in the File menu, and search the respective XCCDF, SCAP RPM, or data stream file.

3. You can allow automatic correction of the system configuration by selecting the Remediate check box. With this option enabled, SCAP Workbench attempts to change the system configuration in accordance with the security rules applied by the policy. This process should fix the related checks that fail during the system scan.
**WARNING**

If not used carefully, running the system evaluation with the **Remediate** option enabled might render the system non-functional. Red Hat does not provide any automated method to revert changes made by security-hardening remediations. Remediations are supported on RHEL systems in the default configuration. If your system has been altered after the installation, running remediation might not make it compliant with the required security profile.

4. **Scan your system with the selected profile by clicking the **Scan** button.**

5. To store the scan results in form of an XCCDF, ARF, or HTML file, click the **Save Results** combo box. Choose the **HTML Report** option to generate the scan report in human-readable format. The XCCDF and ARF (data stream) formats are suitable for further automatic processing. You can repeatedly choose all three options.

6. To export results-based remediations to a file, use the **Generate remediation role** pop-up menu.

**16.9.2. Customizing a security profile with SCAP Workbench**
You can customize a security profile by changing parameters in certain rules (for example, minimum password length), removing rules that you cover in a different way, and selecting additional rules, to implement internal policies. You cannot define new rules by customizing a profile.

The following procedure demonstrates the use of **SCAP Workbench** for customizing (tailoring) a profile. You can also save the tailored profile for use with the **oscap** command-line utility.

**Prerequisites**

- The **scap-workbench** package is installed on your system.

**Procedure**

1. Run **SCAP Workbench**, and select the profile to customize by using either **Open content from SCAP Security Guide** or **Open Other Content** in the **File** menu.

2. To adjust the selected security profile according to your needs, click the **Customize** button. This opens the new Customization window that enables you to modify the currently selected profile without changing the original data stream file. Choose a new profile ID.

   ![Customize Profile](image)

   **Choose the ID of your profile.**

   **Warning:** Choose it wisely. It cannot be changed later and may be required if you choose to use command line tools or various integrations of OpenSCAP.

   The ID has to have a format of "xccdf_{reverse DNS}_profile_{rest of the ID}.
   For example "xccdf_org.mycorporation_profile_server".

   **New Profile ID**

   ![New Profile ID](image)

   3. Find a rule to modify using either the tree structure with rules organized into logical groups or the **Search** field.

   4. Include or exclude rules using check boxes in the tree structure, or modify values in rules where applicable.
5. Confirm the changes by clicking the OK button.

6. To store your changes permanently, use one of the following options:

   - Save a customization file separately by using **Save Customization Only** in the File menu.

   - Save all security content at once by **Save All** in the File menu.
     If you select the **Into a directory** option, SCAP Workbench saves both the data stream file and the customization file to the specified location. You can use this as a backup solution.

     By selecting the **As RPM** option, you can instruct SCAP Workbench to create an RPM package containing the data stream file and the customization file. This is useful for distributing the security content to systems that cannot be scanned remotely, and for delivering the content for further processing.

     **NOTE**

     Because SCAP Workbench does not support results-based remediations for tailored profiles, use the exported remediations with the **oscap** command-line utility.

### 16.9.3. Additional resources

- **scap-workbench(8)** man page

- SCAP Workbench User Manual

- Deploy customized SCAP policies with Satellite 6.x - a Knowledge Base article on tailoring scripts
16.10. SCANNING CONTAINER AND CONTAINER IMAGES FOR VULNERABILITIES

Use this procedure to find security vulnerabilities in a container or a container image.

**NOTE**

The oscap-podman command is available from RHEL 8.2. For RHEL 8.1 and 8.0, use the workaround described in the Using OpenSCAP for scanning containers in RHEL 8 Knowledgebase article.

Prerequisites

- The openscap-utils package is installed.

Procedure

1. Download the latest RHSA OVAL definitions for your system:

```
```

2. Get the ID of a container or a container image, for example:

```
# podman images
REPOSITORY                            TAG      IMAGE ID       CREATED       SIZE
registry.access.redhat.com/ubi8/ubi   latest   096cae65a207   7 weeks ago   239 MB
```

3. Scan the container or the container image for vulnerabilities and save results to the vulnerability.html file:

```
# oscap-podman 096cae65a207 oval eval --report vulnerability.html rhel-8.oval.xml
```

Note that the oscap-podman command requires root privileges, and the ID of a container is the first argument.

Verification

- Check the results in a browser of your choice, for example:

```
$ firefox vulnerability.html &
```

Additional resources

- For more information, see the oscap-podman(8) and oscap(8) man pages.

16.11. ASSESSING SECURITY COMPLIANCE OF A CONTAINER OR A CONTAINER IMAGE WITH A SPECIFIC BASELINE

Follow these steps to assess compliance of your container or a container image with a specific security baseline, such as Operating System Protection Profile (OSPP), Payment Card Industry Data Security Standard (PCI-DSS), and Health Insurance Portability and Accountability Act (HIPAA).
NOTE

The oscap-podman command is available from RHEL 8.2. For RHEL 8.1 and 8.0, use the workaround described in the Using OpenSCAP for scanning containers in RHEL 8 Knowledgebase article.

Prerequisites

- The openscap-utils and scap-security-guide packages are installed.

Procedure

1. Get the ID of a container or a container image, for example:

   ```bash
   # podman images
   REPOSITORY                            TAG      IMAGE ID       CREATED       SIZE
   registry.access.redhat.com/ubi8/ubi   latest   096cae65a207   7 weeks ago   239 MB
   ```

2. Evaluate the compliance of the container image with the HIPAA profile and save scan results into the report.html HTML file

   ```bash
   # oscap-podman 096cae65a207 xccdf eval --report report.html --profile hipaa /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
   ```

   Replace 096cae65a207 with the ID of your container image and the hipaa value with ospp or pci-dss if you assess security compliance with the OSPP or PCI-DSS baseline. Note that the oscap-podman command requires root privileges.

Verification

- Check the results in a browser of your choice, for example:

  ```bash
  $ firefox report.html &
  ```

NOTE

The rules marked as notapplicable are rules that do not apply to containerized systems. These rules apply only to bare-metal and virtualized systems.

Additional resources

- oscap-podman(8) and scap-security-guide(8) man pages.

16.12. SUPPORTED VERSIONS OF THE SCAP SECURITY GUIDE IN RHEL

Officially supported versions of the SCAP Security Guide are versions provided in the related minor release of RHEL or in the related batch update of RHEL.

Table 16.2. Supported versions of the SCAP Security Guide in RHEL
### 16.13. CHECKING INTEGRITY WITH AIDE

Advanced Intrusion Detection Environment (**AIDE**) is a utility that creates a database of files on the system, and then uses that database to ensure file integrity and detect system intrusions.

#### 16.13.1. Installing AIDE

The following steps are necessary to install **AIDE** and to initiate its database.

**Prerequisites**

- The **AppStream** repository is enabled.
Procedure

1. To install the `aide` package:

   ```bash
   # yum install aide
   ```

2. To generate an initial database:

   ```bash
   # aide --init
   ```

   **NOTE**

   In the default configuration, the `aide --init` command checks just a set of directories and files defined in the `/etc/aide.conf` file. To include additional directories or files in the AIDE database, and to change their watched parameters, edit `/etc/aide.conf` accordingly.

3. To start using the database, remove the `.new` substring from the initial database file name:

   ```bash
   # mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
   ```

4. To change the location of the AIDE database, edit the `/etc/aide.conf` file and modify the `DBDIR` value. For additional security, store the database, configuration, and the `/usr/sbin/aide` binary file in a secure location such as a read-only media.

### 16.13.2. Performing integrity checks with AIDE

**Prerequisites**

- AIDE is properly installed and its database is initialized. See Installing AIDE

**Procedure**

1. To initiate a manual check:

   ```bash
   # aide --check
   Start timestamp: 2018-07-11 12:41:20 +0200 (AIDE 0.16)
   AIDE found differences between database and filesystem!!
   ...
   [trimmed for clarity]
   ```

2. At a minimum, configure the system to run AIDE weekly. Optimally, run AIDE daily. For example, to schedule a daily execution of AIDE at 04:05 a.m. using the `cron` command, add the following line to the `/etc/crontab` file:

   ```bash
   05 4 * * * root /usr/sbin/aide --check
   ```

### 16.13.3. Updating an AIDE database

After verifying the changes of your system such as, package updates or configuration files adjustments, Red Hat recommends updating your baseline AIDE database.
Prerequisites

- **AIDE** is properly installed and its database is initialized. See  [Installing AIDE](#).

Procedure

1. Update your baseline **AIDE** database:

   ```
   # aide --update
   ```

   The `aide --update` command creates the `/var/lib/aide/aide.db.new.gz` database file.

2. To start using the updated database for integrity checks, remove the `.new` substring from the file name.

### 16.13.4. File-integrity tools: AIDE and IMA

Red Hat Enterprise Linux provides several tools for checking and preserving the integrity of files and directories on your system. The following table helps you decide which tool better fits your scenario.

#### Table 16.3. Comparison between AIDE and IMA

<table>
<thead>
<tr>
<th>Question</th>
<th>Advanced Intrusion Detection Environment (AIDE)</th>
<th>Integrity Measurement Architecture (IMA)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What</strong></td>
<td>AIDE is a utility that creates a database of files and directories on the system. This database serves for checking file integrity and detect intrusion detection.</td>
<td>IMA detects if a file is altered by checking file measurement (hash values) compared to previously stored extended attributes.</td>
</tr>
<tr>
<td><strong>How</strong></td>
<td>AIDE uses rules to compare the integrity state of the files and directories.</td>
<td>IMA uses file hash values to detect the intrusion.</td>
</tr>
<tr>
<td><strong>Why</strong></td>
<td>Detection - AIDE detects if a file is modified by verifying the rules.</td>
<td>Detection and Prevention - IMA detects and prevents an attack by replacing the extended attribute of a file.</td>
</tr>
<tr>
<td><strong>Usage</strong></td>
<td>AIDE detects a threat when the file or directory is modified.</td>
<td>IMA detects a threat when someone tries to alter the entire file.</td>
</tr>
<tr>
<td><strong>Extension</strong></td>
<td>AIDE checks the integrity of files and directories on the local system.</td>
<td>IMA ensures security on the local and remote systems.</td>
</tr>
</tbody>
</table>

### 16.13.5. Additional resources

- **aide(1)** man page
-  [Kernel integrity subsystem](#)
16.14. ENCRYPTING BLOCK DEVICES USING LUKS

Disk encryption protects the data on a block device by encrypting it. To access the device’s decrypted contents, a user must provide a passphrase or key as authentication. This is particularly important when it comes to mobile computers and removable media: it helps to protect the device’s contents even if it has been physically removed from the system. The LUKS format is a default implementation of block device encryption in RHEL.

16.14.1. LUKS disk encryption

The Linux Unified Key Setup-on-disk-format (LUKS) enables you to encrypt block devices and it provides a set of tools that simplifies managing the encrypted devices. LUKS allows multiple user keys to decrypt a master key, which is used for the bulk encryption of the partition.

RHEL utilizes LUKS to perform block device encryption. By default, the option to encrypt the block device is unchecked during the installation. If you select the option to encrypt your disk, the system prompts you for a passphrase every time you boot the computer. This passphrase “unlocks” the bulk encryption key that decrypts your partition. If you choose to modify the default partition table, you can choose which partitions you want to encrypt. This is set in the partition table settings.

What LUKS does

- LUKS encrypts entire block devices and is therefore well-suited for protecting contents of mobile devices such as removable storage media or laptop disk drives.
- The underlying contents of the encrypted block device are arbitrary, which makes it useful for encrypting swap devices. This can also be useful with certain databases that use specially formatted block devices for data storage.
- LUKS uses the existing device mapper kernel subsystem.
- LUKS provides passphrase strengthening, which protects against dictionary attacks.
- LUKS devices contain multiple key slots, allowing users to add backup keys or passphrases.

What LUKS does not do

- Disk-encryption solutions like LUKS protect the data only when your system is off. Once the system is on and LUKS has decrypted the disk, the files on that disk are available to anyone who would normally have access to them.
- LUKS is not well-suited for scenarios that require many users to have distinct access keys to the same device. The LUKS1 format provides eight key slots, LUKS2 up to 32 key slots.
- LUKS is not well-suited for applications requiring file-level encryption.

Ciphers

The default cipher used for LUKS is **aes-xts-plain64**. The default key size for LUKS is 512 bits. The default key size for LUKS with Anaconda (XTS mode) is 512 bits. Ciphers that are available are:

- AES - Advanced Encryption Standard
- Twofish (a 128-bit block cipher)
- Serpent
Additional resources

- LUKS Project Home Page
- LUKS On-Disk Format Specification
- FIPS PUB 197

16.14.2. LUKS versions in RHEL

In RHEL, the default format for LUKS encryption is LUKS2. The legacy LUKS1 format remains fully supported and it is provided as a format compatible with earlier RHEL releases.

The LUKS2 format is designed to enable future updates of various parts without a need to modify binary structures. LUKS2 internally uses JSON text format for metadata, provides redundancy of metadata, detects metadata corruption and allows automatic repairs from a metadata copy.

**IMPORTANT**

Do not use LUKS2 in systems that must be compatible with legacy systems that support only LUKS1. Note that RHEL 7 supports the LUKS2 format since version 7.6.

**WARNING**

LUKS2 and LUKS1 use different commands to encrypt the disk. Using the wrong command for a LUKS version might cause data loss.

<table>
<thead>
<tr>
<th>LUKS version</th>
<th>Encryption command</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUKS2</td>
<td>cryptsetup reencrypt</td>
</tr>
<tr>
<td>LUKS1</td>
<td>cryptsetup-reencrypt</td>
</tr>
</tbody>
</table>

**Online re-encryption**

The LUKS2 format supports re-encrypting encrypted devices while the devices are in use. For example, you do not have to unmount the file system on the device to perform the following tasks:

- Change the volume key
- Change the encryption algorithm

When encrypting a non-encrypted device, you must still unmount the file system. You can remount the file system after a short initialization of the encryption.

The LUKS1 format does not support online re-encryption.

**Conversion**
The LUKS2 format is inspired by LUKS1. In certain situations, you can convert LUKS1 to LUKS2. The conversion is not possible specifically in the following scenarios:

- A LUKS1 device is marked as being used by a Policy-Based Decryption (PBD - Clevis) solution. The `cryptsetup` tool refuses to convert the device when some `luksmeta` metadata are detected.
- A device is active. The device must be in the inactive state before any conversion is possible.

### 16.14.3. Options for data protection during LUKS2 re-encryption

LUKS2 provides several options that prioritize performance or data protection during the re-encryption process:

**checksum**

This is the default mode. It balances data protection and performance. This mode stores individual checksums of the sectors in the re-encryption area, so the recovery process can detect which sectors LUKS2 already re-encrypted. The mode requires that the block device sector write is atomic.

**journal**

That is the safest mode but also the slowest. This mode journals the re-encryption area in the binary area, so LUKS2 writes the data twice.

**none**

This mode prioritizes performance and provides no data protection. It protects the data only against safe process termination, such as the `SIGTERM` signal or the user pressing `Ctrl+C`. Any unexpected system crash or application crash might result in data corruption.

You can select the mode using the `--resilience` option of `cryptsetup`.

If a LUKS2 re-encryption process terminates unexpectedly by force, LUKS2 can perform the recovery in one of the following ways:

- Automatically, during the next LUKS2 device open action. This action is triggered either by the `cryptsetup open` command or by attaching the device with `systemd-cryptsetup`.
- Manually, by using the `cryptsetup repair` command on the LUKS2 device.

### 16.14.4. Encrypting existing data on a block device using LUKS2

This procedure encrypts existing data on a not yet encrypted device using the LUKS2 format. A new LUKS header is stored in the head of the device.

#### Prerequisites

- The block device contains a file system.
- You have backed up your data.
**WARNING**

You might lose your data during the encryption process: due to a hardware, kernel, or human failure. Ensure that you have a reliable backup before you start encrypting the data.

---

**Procedure**

1. Unmount all file systems on the device that you plan to encrypt. For example:

   ```
   # umount /dev/sdb1
   ```

2. Make free space for storing a LUKS header. Choose one of the following options that suits your scenario:

   - In the case of encrypting a logical volume, you can extend the logical volume without resizing the file system. For example:

     ```
     # lvextend -L+32M vg00/lv00
     ```

   - Extend the partition using partition management tools, such as `parted`.

   - Shrink the file system on the device. You can use the `resize2fs` utility for the ext2, ext3, or ext4 file systems. Note that you cannot shrink the XFS file system.

3. Initialize the encryption. For example:

   ```
   # cryptsetup reencrypt \
   --encrypt \
   --init-only \
   --reduce-device-size 32M \
   /dev/sdb1 sdb1_encrypted
   ```

   The command asks you for a passphrase and starts the encryption process.

4. Mount the device:

   ```
   # mount /dev/mapper/sdb1_encrypted /mnt/sdb1_encrypted
   ```

5. Start the online encryption:

   ```
   # cryptsetup reencrypt --resume-only /dev/sdb1
   ```

**Additional resources**

- `cryptsetup(8)`, `lvextend(8)`, `resize2fs(8)`, and `parted(8)` man pages

**16.14.5. Encrypting existing data on a block device using LUKS2 with a detached header**
This procedure encrypts existing data on a block device without creating free space for storing a LUKS header. The header is stored in a detached location, which also serves as an additional layer of security. The procedure uses the LUKS2 encryption format.

**Prerequisites**

- The block device contains a file system.
- You have backed up your data.

**WARNING**

You might lose your data during the encryption process: due to a hardware, kernel, or human failure. Ensure that you have a reliable backup before you start encrypting the data.

**Procedure**

1.Unmount all file systems on the device. For example:

   ```
   # umount /dev/sdb1
   ```

2. Initialize the encryption:

   ```
   # cryptsetup reencrypt \
   --encrypt \
   --init-only \
   --header /path/to/header \
   /dev/sdb1 sdb1_encrypted
   ```

   Replace `/path/to/header` with a path to the file with a detached LUKS header. The detached LUKS header has to be accessible so that the encrypted device can be unlocked later.

   The command asks you for a passphrase and starts the encryption process.

3. Mount the device:

   ```
   # mount /dev/mapper/sdb1_encrypted /mnt/sdb1_encrypted
   ```

4. Start the online encryption:

   ```
   # cryptsetup reencrypt --resume-only --header /path/to/header /dev/sdb1
   ```

**Additional resources**

- cryptsetup(8) man page

**16.14.6. Encrypting a blank block device using LUKS2**
This procedure provides information about encrypting a blank block device using the LUKS2 format.

**Prerequisites**
- A blank block device.

**Procedure**

1. Setup a partition as an encrypted LUKS partition:
   ```
   # cryptsetup luksFormat /dev/sdb1
   ```

2. Open an encrypted LUKS partition:
   ```
   # cryptsetup open /dev/sdb1 sdb1_encrypted
   ```
   This unlocks the partition and maps it to a new device using the device mapper. This alerts kernel that *device* is an encrypted device and should be addressed through LUKS using the `/dev/mapper/device_mapped_name` so as not to overwrite the encrypted data.

3. To write encrypted data to the partition, it must be accessed through the device mapped name. To do this, you must create a file system. For example:
   ```
   # mkfs -t ext4 /dev/mapper/sdb1_encrypted
   ```

4. Mount the device:
   ```
   # mount /dev/mapper/sdb1_encrypted mount-point
   ```

**Additional resources**
- `cryptsetup(8)` man page

**16.14.7. Creating a LUKS encrypted volume using the storage role**

You can use the `storage` role to create and configure a volume encrypted with LUKS by running an Ansible playbook.

**Prerequisites**
- You have Red Hat Ansible Engine installed on the system from which you want to run the playbook.
  
  **NOTE**
  You do not have to have Red Hat Ansible Automation Platform installed on the systems on which you want to create the volume.

- You have the `rhel-system-roles` package installed on the Ansible controller.
- You have an inventory file detailing the systems on which you want to deploy a LUKS encrypted volume using the storage System Role.
**Procedure**

1. Create a new `playbook.yml` file with the following content:

   ```yaml
   - hosts: all
     vars:
       storage_volumes:
         - name: barefs
           type: disk
           disks:
             - sdb
           fs_type: xfs
           fs_label: label-name
           mount_point: /mnt/data
           encryption: true
           encryption_password: your-password
     roles:
       - rhel-system-roles.storage
   ```

2. Optional: Verify playbook syntax:

   ```bash
   # ansible-playbook --syntax-check playbook.yml
   ```

3. Run the playbook on your inventory file:

   ```bash
   # ansible-playbook -i inventory.file /path/to/file/playbook.yml
   ```

**Additional resources**

- Encrypting block devices using LUKS
- `/usr/share/ansible/roles/rhel-system-roles.storage/README.md` file

### 16.15. CONFIGURING AUTOMATED UNLOCKING OF ENCRYPTED VOLUMES USING POLICY-BASED DECRYPTION

The Policy-Based Decryption (PBD) is a collection of technologies that enable unlocking encrypted root and secondary volumes of hard drives on physical and virtual machines. PBD uses a variety of unlocking methods, such as user passwords, a Trusted Platform Module (TPM) device, a PKCS #11 device connected to a system, for example, a smart card, or a special network server.

PBD allows combining different unlocking methods into a policy, which makes it possible to unlock the same volume in different ways. The current implementation of the PBD in Red Hat Enterprise Linux consists of the Clevis framework and plug-ins called *pins*. Each pin provides a separate unlocking capability. Currently, the following pins are available:

- **tang** - allows to unlock volumes using a network server
- **tpm2** - allows to unlock volumes using a TPM2 policy
- **sss** - allows to deploy high-availability systems using the Shamir’s Secret Sharing (SSS) cryptographic scheme
The Network Bound Disc Encryption (NBDE) is a subcategory of PBD that allows binding encrypted volumes to a special network server. The current implementation of the NBDE includes a Clevis pin for the Tang server and the Tang server itself.

### 16.15.1. Network-bound disk encryption

In Red Hat Enterprise Linux, NBDE is implemented through the following components and technologies:

**Figure 16.1. NBDE scheme when using a LUKS1-encrypted volume. The luksmeta package is not used for LUKS2 volumes.**

![Diagram](image)

*Tang* is a server for binding data to network presence. It makes a system containing your data available when the system is bound to a certain secure network. Tang is stateless and does not require TLS or authentication. Unlike escrow-based solutions, where the server stores all encryption keys and has knowledge of every key ever used, Tang never interacts with any client keys, so it never gains any identifying information from the client.

*Clevis* is a pluggable framework for automated decryption. In NBDE, Clevis provides automated unlocking of LUKS volumes. The *clevis* package provides the client side of the feature.

A *Clevis pin* is a plug-in into the Clevis framework. One of such pins is a plug-in that implements interactions with the NBDE server — Tang.

Clevis and Tang are generic client and server components that provide network-bound encryption. In Red Hat Enterprise Linux, they are used in conjunction with LUKS to encrypt and decrypt root and non-root storage volumes to accomplish Network-Bound Disk Encryption.

Both client- and server-side components use the *José* library to perform encryption and decryption operations.

When you begin provisioning NBDE, the Clevis pin for Tang server gets a list of the Tang server’s advertised asymmetric keys. Alternatively, since the keys are asymmetric, a list of Tang’s public keys can be distributed out of band so that clients can operate without access to the Tang server. This mode is called *offline provisioning*.

The Clevis pin for Tang uses one of the public keys to generate a unique, cryptographically-strong encryption key. Once the data is encrypted using this key, the key is discarded. The Clevis client should store the state produced by this provisioning operation in a convenient location. This process of encrypting data is the *provisioning step*. 
The LUKS version 2 (LUKS2) is the default disk-encryption format in RHEL, hence, the provisioning state for NBDE is stored as a token in a LUKS2 header. The leveraging of provisioning state for NBDE by the `luksmeta` package is used only for volumes encrypted with LUKS1.

The Clevis pin for Tang supports both LUKS1 and LUKS2 without specification need. Clevis can encrypt plain-text files but you have to use the `cryptsetup` tool for encrypting block devices. See the [Encrypting block devices using LUKS](https://redhat.github.io/pkcs11-python/) for more information.

When the client is ready to access its data, it loads the metadata produced in the provisioning step and it responds to recover the encryption key. This process is the recovery step.

In NBDE, Clevis binds a LUKS volume using a pin so that it can be automatically unlocked. After successful completion of the binding process, the disk can be unlocked using the provided Dracut unlocker.

**NOTE**

If the `kdump` kernel crash dumping mechanism is set to save the content of the system memory to a LUKS-encrypted device, you are prompted for entering a password during the second kernel boot.

### 16.15.2. Installing an encryption client - Clevis

Use this procedure to deploy and start using the Clevis pluggable framework on your system.

**Procedure**

1. To install Clevis and its pins on a system with an encrypted volume:

   ```
   # yum install clevis
   ```

2. To decrypt data, use a `clevis decrypt` command and provide a cipher text in the JSON Web Encryption (JWE) format, for example:

   ```
   $ clevis decrypt < secret.jwe
   ```

**Additional resources**

- `clevis(1)` man page
- Built-in CLI help after entering the `clevis` command without any argument:

  ```
  $ clevis
  Usage: clevis COMMAND [OPTIONS]
  clevis decrypt          Decrypts using the policy defined at encryption time
  clevis encrypt sss      Encrypts using a Shamir's Secret Sharing policy
  clevis encrypt tang     Encrypts using a Tang binding server policy
  clevis encrypt tpm2     Encrypts using a TPM2.0 chip binding policy
  clevis luks bind        Binds a LUKS device using the specified policy
  clevis luks list        Lists pins bound to a LUKSv1 or LUKSv2 device
  clevis luks pass        Returns the LUKE passpharse used for binding a particular slot.
  clevis luks regen       Regenerate LUKS metadata
  ```
16.15.3. Deploying a Tang server with SELinux in enforcing mode

Use this procedure to deploy a Tang server running on a custom port as a confined service in SELinux enforcing mode.

Prerequisites

- The `policycoreutils-python-utils` package and its dependencies are installed.
- The `firewalld` service is running.

Procedure

1. To install the `tang` package and its dependencies, enter the following command as `root`:

```
# yum install tang
```

2. Pick an unoccupied port, for example, `7500/tcp`, and allow the `tangd` service to bind to that port:

```
# semanage port -a -t tangd_port_t -p tcp 7500
```

   Note that a port can be used only by one service at a time, and thus an attempt to use an already occupied port implies the `ValueError: Port already defined` error message.

3. Open the port in the firewall:

```
# firewall-cmd --add-port=7500/tcp
# firewall-cmd --runtime-to-permanent
```

4. Enable the `tangd` service:

```
# systemctl enable tangd.socket
```

5. Create an override file:

```
# systemctl edit tangd.socket
```

6. In the following editor screen, which opens an empty `override.conf` file located in the `/etc/systemd/system/tangd.socket.d/` directory, change the default port for the Tang server from 80 to the previously picked number by adding the following lines:

```
[Socket]
ListenStream=
ListenStream=7500
```

   Save the file and exit the editor.

7. Reload the changed configuration:
# systemctl daemon-reload

8. Check that your configuration is working:

```bash
# systemctl show tangd.socket -p Listen
Listen=[::]:7500 (Stream)
```

9. Start the `tangd` service:

```bash
# systemctl restart tangd.socket
```

Because `tangd` uses the `systemd` socket activation mechanism, the server starts as soon as the first connection comes in. A new set of cryptographic keys is automatically generated at the first start. To perform cryptographic operations such as manual key generation, use the `jose` utility.

Additional resources

- `tang(8)`, `semanage(8)`, `firewall-cmd(1)`, `jose(1)`, `systemd.unit(5)`, and `systemd.socket(5)` man pages

16.15.4. Rotating Tang server keys and updating bindings on clients

Use the following steps to rotate your Tang server keys and update existing bindings on clients. The precise interval at which you should rotate them depends on your application, key sizes, and institutional policy.

Alternatively, you can rotate Tang keys by using the `nbde_server` RHEL system role. See Using the `nbde_server system role for setting up multiple Tang servers` for more information.

Prerequisites

- A Tang server is running.
- The `clevis` and `clevis-luks` packages are installed on your clients.
- Note that `clevis luks list`, `clevis luks report`, and `clevis luks regen` have been introduced in RHEL 8.2.

Procedure

1. Rename all keys in the `/var/db/tang` key database directory to have a leading . to hide them from advertisement. Note that the file names in the following example differs from unique file names in the key database directory of your Tang server:

```bash
# cd /var/db/tang
# ls -l
-rw-r--r--. 1 root root 349 Feb  7 14:55 UV6dqXSwe1bRKG3KbJmdI020hY.jwk
-rw-r--r--. 1 root root 354 Feb  7 14:55 y9hxLTQSiSB5jSEGtNjhY8fDTJU.jwk
# mv UV6dqXSwe1bRKG3KbJmdI020hY.jwk .UV6dqXSwe1bRKG3KbJmdI020hY.jwk
# mv y9hxLTQSiSB5jSEGtNjhY8fDTJU.jwk .y9hxLTQSiSB5jSEGtNjhY8fDTJU.jwk
```

2. Check that you renamed and therefore hid all keys from the Tang server advertisement:
3. Generate new keys using the `/usr/libexec/tangd-keygen` command in `/var/db/tang` on the Tang server:

   ```
   # /usr/libexec/tangd-keygen /var/db/tang
   # ls /var/db/tang
   3ZWS6-cDrCG61UPJS2BMmPU4I54.jwk zyLuX6hijUy_PSeUEFDi7hi38.jwk
   ```

4. Check that your Tang server advertises the signing key from the new key pair, for example:

   ```
   # tang-show-keys 7500
   3ZWS6-cDrCG61UPJS2BMmPU4I54
   ```

5. On your NBDE clients, use the `clevis luks report` command to check if the keys advertised by the Tang server remains the same. You can identify slots with the relevant binding using the `clevis luks list` command, for example:

   ```
   # clevis luks list -d /dev/sda2
   1: tang '{"url":"http://tang.srv"}'
   # clevis luks report -d /dev/sda2 -s 1
   ...
   Report detected that some keys were rotated.
   Do you want to regenerate luks metadata with "clevis luks regen -d /dev/sda2 -s 1"? [ynYN]
   ```

6. To regenerate LUKS metadata for the new keys either press `y` to the prompt of the previous command, or use the `clevis luks regen` command:

   ```
   # clevis luks regen -d /dev/sda2 -s 1
   ```

7. When you are sure that all old clients use the new keys, you can remove the old keys from the Tang server, for example:

   ```
   # cd /var/db/tang
   # rm *.jwk
   ```

   **WARNING**
   Removing the old keys while clients are still using them can result in data loss. If you accidentally remove such keys, use the `clevis luks regen` command on the clients, and provide your LUKS password manually.

Additional resources

- `tang-show-keys(1)`, `clevis-luks-list(1)`, `clevis-luks-report(1)`, and `clevis-luks-regen(1)` man pages
16.15.5. Configuring automated unlocking using a Tang key in the web console

Configure automated unlocking of a LUKS-encrypted storage device using a key provided by a Tang server.

Prerequisites

- The RHEL 8 web console has been installed. For details, see Installing the web console.
- The `cockpit-storaged` package is installed on your system.
- The `cockpit.socket` service is running at port 9090.
- The `clevis`, `tang`, and `clevis-dracut` packages are installed.
- A Tang server is running.

Procedure

1. Open the RHEL web console by entering the following address in a web browser:

   `https://localhost:9090`

   Replace the `localhost` part by the remote server’s host name or IP address when you connect to a remote system.

2. Provide your credentials and click Storage. Select an encrypted device and click Encryption in the Content part:

3. Click + in the Keys section to add a Tang key:

   ![Image of a web console showing the Storage section with an encrypted device selected and the Keys section expanded]

   - Capacity: 14.9 GiB, 16.0 GB, 16013852672 bytes
   - Device File: /dev/ssa

4. Provide the address of your Tang server and a password that unlocks the LUKS-encrypted device. Click Add to confirm:
5. The following dialog window provides a command to verify that the key hash matches. RHEL 8.2 introduced the `tang-show-keys` script, and you can obtain the key hash using the following command on the Tang server running on the port 7500:

```
# tang-show-keys 7500
3ZWS6-cDrCG61UPJS2BMmPU4I54
```

On RHEL 8.1 and earlier, obtain the key hash using the following command:

```
# curl -s localhost:7500/adv | jose fmt -j -g payload -y -o- | jose jwk use -i- -r -u verify -o- | jose jwk thp -i-
3ZWS6-cDrCG61UPJS2BMmPU4I54
```

6. Click Trust key when the key hashes in the web console and in the output of previously listed commands are the same:
Verify key

Make sure the key hash from the Tang server matches:

3ZWS6-cDrCG61UPJS2BMmPU4I54

Manually check with SSH: `ssh localhost tang-show-keys 7500`

If `tang-show-keys` is not available, run the following:

```
ssh localhost "curl -s localhost:7500/adv | 
    jose fmt -j -g payload -y -o - | 
    jose jwk use -i -r -u verify -o - | 
    jose jwk thp -i -"
```

7. To enable the early boot system to process the disk binding, click Terminal at the bottom of the left navigation bar and enter the following commands:

```
# yum install clevis-dracut
# grubby --update-kernel=ALL --args="rd.neednet=1"
# dracut -fv --regenerate-all
```

Verification

1. Check that the newly added Tang key is now listed in the Keys section with the Keyserver type:

<table>
<thead>
<tr>
<th>Partition</th>
<th>Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>/dev/sda1</td>
</tr>
</tbody>
</table>

Options (none)

2. Verify that the bindings are available for the early boot, for example:

```
# lsinitrd | grep clevis
clevis
clevis-pin-sss
clevis-pin-tang
```
Additional resources

- Configuring automated unlocking of encrypted volumes using policy-based decryption

16.15.6. Basic NBDE and TPM2 encryption-client operations

The Clevis framework can encrypt plain-text files and decrypt both ciphertexts in the JSON Web Encryption (JWE) format and LUKS-encrypted block devices. Clevis clients can use either Tang network servers or Trusted Platform Module 2.0 (TPM 2.0) chips for cryptographic operations.

The following commands demonstrate the basic functionality provided by Clevis on examples containing plain-text files. You can also use them for troubleshooting your NBDE or Clevis+TPM deployments.

**Encryption client bound to a Tang server**

- To check that a Clevis encryption client binds to a Tang server, use the `clevis encrypt tang` sub-command:

  ```
  $ clevis encrypt tang '{"url":"http://tang.srv:port"}' < input-plain.txt > secret.jwe
  
  The advertisement contains the following signing keys:

  _OsIk0T-E2l6qjfdDiwVmidoZjA

  Do you wish to trust these keys? [ynYN] y
  
  Change the `http://tang.srv:port` URL in the previous example to match the URL of the server where `tang` is installed. The `secret.jwe` output file contains your encrypted cipher text in the JWE format. This cipher text is read from the `input-plain.txt` input file.

  Alternatively, if your configuration requires a non-interactive communication with a Tang server without SSH access, you can download an advertisement and save it to a file:

  ```
  $ curl -sfg http://tang.srv:port/adv -o adv.jws
  
  Use the advertisement in the `adv.jws` file for any following tasks, such as encryption of files or messages:

  ```
  $ echo 'hello' | clevis encrypt tang '{"url":"http://tang.srv:port","adv":"adv.jws"}'
  
  - To decrypt data, use the `clevis decrypt` command and provide the cipher text (JWE):

  ```
  $ clevis decrypt < secret.jwe > output-plain.txt
  
  Encryption client using TPM 2.0
To encrypt using a TPM 2.0 chip, use the clevis encrypt tpm2 sub-command with the only argument in form of the JSON configuration object:

```
$ clevis encrypt tpm2 '{}' < input-plain.txt > secret.jwe
```

To choose a different hierarchy, hash, and key algorithms, specify configuration properties, for example:

```
$ clevis encrypt tpm2 '{"hash":"sha1","key":"rsa"}' < input-plain.txt > secret.jwe
```

To decrypt the data, provide the ciphertext in the JSON Web Encryption (JWE) format:

```
$ clevis decrypt < secret.jwe > output-plain.txt
```

The pin also supports sealing data to a Platform Configuration Registers (PCR) state. That way, the data can only be unsealed if the PCR hashes values match the policy used when sealing.

For example, to seal the data to the PCR with index 0 and 7 for the SHA-1 bank:

```
$ clevis encrypt tpm2 '{"pcr_bank":"sha1","pcr_ids":"0,7"}' < input-plain.txt > secret.jwe
```

---

**WARNING**

Hashes in PCRs can be rewritten, and you no longer can unlock your encrypted volume. For this reason, add a strong passphrase that enable you to unlock the encrypted volume manually even when a value in a PCR changes.

If the system cannot automatically unlock your encrypted volume after an upgrade of the **shim-x64** package, follow the steps in the **Clevis TPM2 no longer decrypts LUKS devices after a restart** KCS article.

---

**Additional resources**

- clevis-encrypt-tang(1), clevis-luks-unlockers(7), clevis(1), and clevis-encrypt-tpm2(1) man pages
- clevis, clevis decrypt, and clevis encrypt tang commands without any arguments show the built-in CLI help, for example:

```
$ clevis encrypt tang
Usage: clevis encrypt tang CONFIG < PLAINTEXT > JWE
...```

**16.15.7. Configuring manual enrollment of LUKS-encrypted volumes**

Use the following steps to configure unlocking of LUKS-encrypted volumes with NBDE.

**Prerequisites**
A Tang server is running and available.

**Procedure**

1. To automatically unlock an existing LUKS-encrypted volume, install the `clevis-luks` subpackage:

   ```bash
   # yum install clevis-luks
   ``

2. Identify the LUKS-encrypted volume for PBD. In the following example, the block device is referred as `/dev/sda2`:

   ```bash
   # lsblk
   NAME          MAJ:MIN   RM  SIZE法师  TYPE     MOUNTPOINT
   sda           8:0       0   12G  0 disk
     └─sda1      8:1       0   1G  0 part  /boot
     └─sda2      8:2       0   11G  0 part
         └─luks-40e20552-2ade-4954-9d56-565aa7994fb6 253:0  0   11G  0 crypt
             └─rhel-root 253:0  0   9.8G  0 lvm   /
             └─rhel-swap 253:1  0   1.2G  0 lvm  [SWAP]
   ```

3. Bind the volume to a Tang server using the `clevis luks bind` command:

   ```bash
   # clevis luks bind -d /dev/sda2 tang '{"url":"http://tang.srv"}"
   ```

   The advertisement contains the following signing keys:

   ```
   _OsIk0T-E2l6qjfdDiwVmidoZjA
   ```

   Do you wish to trust these keys? [ynYN] y

   You are about to initialize a LUKS device for metadata storage. Attempting to initialize it may result in data loss if data was already written into the LUKS header gap in a different format. A backup is advised before initialization is performed.

   Do you wish to initialize /dev/sda2? [yn] y

   Enter existing LUKS password:

   ```
   This command performs four steps:
   ```

   a. Creates a new key with the same entropy as the LUKS master key.

   b. Encrypts the new key with Clevis.

   c. Stores the Clevis JWE object in the LUKS2 header token or uses LUKSMeta if the non-default LUKS1 header is used.

   d. Enables the new key for use with LUKS.

   **NOTE**

   The binding procedure assumes that there is at least one free LUKS password slot. The `clevis luks bind` command takes one of the slots.

4. The volume can now be unlocked with your existing password as well as with the Clevis policy.
5. To enable the early boot system to process the disk binding, use the `dracut` tool on an already installed system:

```bash
# yum install clevis-dracut
```

In Red Hat Enterprise Linux 8, Clevis produces a generic `initrd` (initial ramdisk) without host-specific configuration options and does not automatically add parameters such as `rd.neednet=1` to the kernel command line. If your configuration relies on a Tang pin that requires network during early boot, use the `--hostonly-cmdline` argument and `dracut` adds `rd.neednet=1` when it detects a Tang binding:

```bash
# dracut -fv --regenerate-all --hostonly-cmdline
```

Alternatively, create a `.conf` file in the `/etc/dracut.conf.d/`, and add the `hostonly_cmdline=yes` option to the file, for example:

```bash
# echo "hostonly_cmdline=yes" > /etc/dracut.conf.d/clevis.conf
```

**NOTE**

You can also ensure that networking for a Tang pin is available during early boot by using the `grubby` tool on the system where Clevis is installed:

```bash
# grubby --update-kernel=ALL --args="rd.neednet=1"
```

Then you can use `dracut` without `--hostonly-cmdline`:

```bash
# dracut -fv --regenerate-all
```

**Verification**

1. To verify that the Clevis JWE object is successfully placed in a LUKS header, use the `clevis luks list` command:

```bash
# clevis luks list -d /dev/sda2
1: tang '{"url":"http://tang.srv:port"}'
```
IMPORTANT

To use NBDE for clients with static IP configuration (without DHCP), pass your network configuration to the `dracut` tool manually, for example:

```
# dracut -fv --regenerate-all --kernel-cmdline
"ip=192.0.2.10::192.0.2.1:255.255.255.0::ens3:none"
```

Alternatively, create a `.conf` file in the `/etc/dracut.conf.d/` directory with the static network information. For example:

```
# cat /etc/dracut.conf.d/static_ip.conf
kernel_cmdline="ip=192.0.2.10::192.0.2.1:255.255.255.0::ens3:none"
```

Regenerate the initial RAM disk image:

```
# dracut -fv --regenerate-all
```

Additional resources

- `clevis-luks-bind(1)` and `dracut.cmdline(7)` man pages.
- RHEL Network boot options

16.15.8. Configuring manual enrollment of LUKS-encrypted volumes using a TPM 2.0 policy

Use the following steps to configure unlocking of LUKS-encrypted volumes by using a Trusted Platform Module 2.0 (TPM 2.0) policy.

Prerequisites

- An accessible TPM 2.0-compatible device.
- A system with the 64-bit Intel or 64-bit AMD architecture.

Procedure

1. To automatically unlock an existing LUKS-encrypted volume, install the `clevis-luks` subpackage:

   ```
   # yum install clevis-luks
   ```

2. Identify the LUKS-encrypted volume for PBD. In the following example, the block device is referred as `/dev/sda2`:

   ```
   # lsblk
   NAME                                  MAJ:MIN  RM  SIZE RO TYPE  MOUNTPOINT
   sda                                    8:0     0 12G  0 disk
   └─sda1                                 8:1     0 1G   0 part  /boot
   └─sda2                                 8:2     0 11G  0 part
   └─luks-40e20552-2ade-4954-9d56-565aa7994fb6 253:0    0     11G  0 crypt
   └─rhel-root                            253:0    0 9.8G  0 lvm  /
   └─rhel-swap                            253:1    0 1.2G  0 lvm  [SWAP]
   ```
3. Bind the volume to a TPM 2.0 device using the `clevis luks bind` command, for example:

```
# clevis luks bind -d /dev/sda2 tpm2 '{"hash":"sha1","key":"rsa"}'
```

Do you wish to initialize /dev/sda2? [yn] y

Enter existing LUKS password:

This command performs four steps:

a. Creates a new key with the same entropy as the LUKS master key.

b. Encrypts the new key with Clevis.

c. Stores the Clevis JWE object in the LUKS2 header token or uses LUKSMeta if the non-default LUKS1 header is used.

d. Enables the new key for use with LUKS.

**NOTE**

The binding procedure assumes that there is at least one free LUKS password slot. The `clevis luks bind` command takes one of the slots.

Alternatively, if you want to seal data to specific Platform Configuration Registers (PCR) states, add the `pcr_bank` and `pcr_ids` values to the `clevis luks bind` command, for example:

```
# clevis luks bind -d /dev/sda2 tpm2
 '{"hash":"sha1","key":"rsa","pcr_bank":"sha1","pcr_ids":"0,1"}'
```

**WARNING**

Because the data can only be unsealed if PCR hashes values match the policy used when sealing and the hashes can be rewritten, add a strong passphrase that enable you to unlock the encrypted volume manually when a value in a PCR changes.

If the system cannot automatically unlock your encrypted volume after an upgrade of the `shim-x64` package, follow the steps in the Clevis TPM2 no longer decrypts LUKS devices after a restart KCS article.

4. The volume can now be unlocked with your existing password as well as with the Clevis policy.

5. To enable the early boot system to process the disk binding, use the `dracut` tool on an already installed system:

```
# yum install clevis-dracut
# dracut -fv --regenerate-all
```
Verification

1. To verify that the Clevis JWE object is successfully placed in a LUKS header, use the `clevis luks list` command:

   ```bash
   # clevis luks list -d /dev/sda2
   1: tpm2 {"hash":"sha1","key":"rsa"}
   ```

Additional resources

- `clevis-luks-bind(1)`, `clevis-encrypt-tpm2(1)`, and `dracut.cmdline(7)` man pages

16.15.9. Removing a Clevis pin from a LUKS-encrypted volume manually

Use the following procedure for manual removing the metadata created by the `clevis luks bind` command and also for wiping a key slot that contains passphrase added by Clevis.

**IMPORTANT**

The recommended way to remove a Clevis pin from a LUKS-encrypted volume is through the `clevis luks unbind` command. The removal procedure using `clevis luks unbind` consists of only one step and works for both LUKS1 and LUKS2 volumes. The following example command removes the metadata created by the binding step and wipe the key slot 1 on the `/dev/sda2` device:

```bash
# clevis luks unbind -d /dev/sda2 -s 1
```

Prerequisites

- A LUKS-encrypted volume with a Clevis binding.

Procedure

1. Check which LUKS version the volume, for example `/dev/sda2`, is encrypted by and identify a slot and a token that is bound to Clevis:

   ```bash
   # cryptsetup luksDump /dev/sda2
   LUKS header information
   Version: 2
   ...
   Keyslots:
   0: luks2
   ...
   1: luks2
   Key: 512 bits
   Priority: normal
   Cipher: aes-xts-plain64
   ...
   Tokens:
   0: clevis
   Keyslot: 1
   ...
   ```

   In the previous example, the Clevis token is identified by 0 and the associated key slot is 1.
2. In case of LUKS2 encryption, remove the token:

```bash
# cryptsetup token remove --token-id 0 /dev/sda2
```

3. If your device is encrypted by LUKS1, which is indicated by the Version: 1 string in the output of the `cryptsetup luksDump` command, perform this additional step with the `luksmeta wipe` command:

```bash
# luksmeta wipe -d /dev/sda2 -s 1
```

4. Wipe the key slot containing the Clevis passphrase:

```bash
# cryptsetup luksKillSlot /dev/sda2 1
```

Additional resources

- clevis-luks-unbind(1), cryptsetup(8), and luksmeta(8) man pages

16.15.10. Configuring automated enrollment of LUKS-encrypted volumes using Kickstart

Follow the steps in this procedure to configure an automated installation process that uses Clevis for the enrollment of LUKS-encrypted volumes.

Procedure

1. Instruct Kickstart to partition the disk such that LUKS encryption has enabled for all mount points, other than `/boot`, with a temporary password. The password is temporary for this step of the enrollment process.

```bash
part /boot --fstype="xfs" --ondisk=vda --size=256
part / --fstype="xfs" --ondisk=vda --grow --encrypted --passphrase=temppass
```

Note that OSPP-compliant systems require a more complex configuration, for example:

```bash
part /boot --fstype="xfs" --ondisk=vda --size=256
part / --fstype="xfs" --ondisk=vda --size=2048 --encrypted --passphrase=temppass
part /var --fstype="xfs" --ondisk=vda --size=1024 --encrypted --passphrase=temppass
part /tmp --fstype="xfs" --ondisk=vda --size=1024 --encrypted --passphrase=temppass
part /home --fstype="xfs" --ondisk=vda --size=2048 --grow --encrypted --passphrase=temppass
part /var/log --fstype="xfs" --ondisk=vda --size=1024 --encrypted --passphrase=temppass
part /var/log/audit --fstype="xfs" --ondisk=vda --size=1024 --encrypted --passphrase=temppass
```

2. Install the related Clevis packages by listing them in the `%packages` section:

```bash
%packages
  clevis-dracut
clevis-luks
clevis-systemd
%end
```
3. Optionally, to ensure that you can unlock the encrypted volume manually when required, add a strong passphrase before you remove the temporary passphrase. See the How to add a passphrase, key, or keyfile to an existing LUKS device article for more information.

4. Call `clevis luks bind` to perform binding in the `%post` section. Afterward, remove the temporary password:

   ```bash
   %post
   clevis luks bind -y -k -d /dev/vda2 \ 
   tang '["url":"http://tang.srv"]' <<< "temppass"
   cryptsetup luksRemoveKey /dev/vda2 <<< "temppass"
   dracut -f --regenerate-all
   %end
   
   If your configuration relies on a Tang pin that requires network during early boot or you use NBDE clients with static IP configurations, you have to modify the `dracut` command as described in Configuring manual enrollment of LUKS-encrypted volumes.

   Note that the `-y` option for the `clevis luks bind` command is available from RHEL 8.3. In RHEL 8.2 and older, replace `-y` by `-f` in the `clevis luks bind` command and download the advertisement from the Tang server:

   ```bash
   %post
   curl -sfg http://tang.srv/adv -o adv.jws
   clevis luks bind -f -k -d /dev/vda2 \ 
   tang '["url":"http://tang.srv","adv":"adv.jws"]' <<< "temppass"
   cryptsetup luksRemoveKey /dev/vda2 <<< "temppass"
   dracut -f --regenerate-all
   %end
   
   WARNING
   The `cryptsetup luksRemoveKey` command prevents any further administration of a LUKS2 device on which you apply it. You can recover a removed master key using the `dmsetup` command only for LUKS1 devices.

   You can use an analogous procedure when using a TPM 2.0 policy instead of a Tang server.

   Additional resources
   - `clevis(1)`, `clevis-luks-bind(1)`, `cryptsetup(8)`, and `dmsetup(8)` man pages
   - Installing Red Hat Enterprise Linux 8 using Kickstart

16.15.11. Configuring automated unlocking of a LUKS-encrypted removable storage device

Use this procedure to set up an automated unlocking process of a LUKS-encrypted USB storage device.
### Procedure

1. To automatically unlock a LUKS-encrypted removable storage device, such as a USB drive, install the `clevis-udisks2` package:

```
# yum install clevis-udisks2
```

2. Reboot the system, and then perform the binding step using the `clevis luks bind` command as described in Configuring manual enrollment of LUKS-encrypted volumes, for example:

```
# clevis luks bind -d /dev/sdb1 tang '{"url":"http://tang.srv"}'
```

3. The LUKS-encrypted removable device can be now unlocked automatically in your GNOME desktop session. The device bound to a Clevis policy can be also unlocked by the `clevis luks unlock` command:

```
# clevis luks unlock -d /dev/sdb1
```

You can use an analogous procedure when using a TPM 2.0 policy instead of a Tang server.

### Additional resources

- `clevis-luks-unlockers(7)` man page

### 16.15.12. Deploying high-availability NBDE systems

Tang provides two methods for building a high-availability deployment:

#### Client redundancy (recommended)

Clients should be configured with the ability to bind to multiple Tang servers. In this setup, each Tang server has its own keys and clients can decrypt by contacting a subset of these servers. Clevis already supports this workflow through its `sss` plug-in. Red Hat recommends this method for a high-availability deployment.

#### Key sharing

For redundancy purposes, more than one instance of Tang can be deployed. To set up a second or any subsequent instance, install the `tang` packages and copy the key directory to the new host using `rsync` over SSH. Note that Red Hat does not recommend this method because sharing keys increases the risk of key compromise and requires additional automation infrastructure.

### 16.15.12.1. High-available NBDE using Shamir’s Secret Sharing

Shamir’s Secret Sharing (SSS) is a cryptographic scheme that divides a secret into several unique parts. To reconstruct the secret, a number of parts is required. The number is called threshold and SSS is also referred to as a thresholding scheme.

Clevis provides an implementation of SSS. It creates a key and divides it into a number of pieces. Each piece is encrypted using another pin including even SSS recursively. Additionally, you define the threshold $t$. If an NBDE deployment decrypts at least $t$ pieces, then it recovers the encryption key and the decryption process succeeds. When Clevis detects a smaller number of parts than specified in the threshold, it prints an error message.

#### 16.15.12.1.1. Example 1: Redundancy with two Tang servers
The following command decrypts a LUKS-encrypted device when at least one of two Tang servers is available:

```bash
# clevis luks bind -d /dev/sda1 sss '{"t":1,"pins":{"tang":[{"url":"http://tang1.srv"},
{"url":"http://tang2.srv"}]}'}
```

The previous command used the following configuration scheme:

```json
{
  "t":1,
  "pins":{
    "tang":[
      {
        "url":"http://tang1.srv"
      },
      {
        "url":"http://tang2.srv"
      }
    ]
  }
}
```

In this configuration, the SSS threshold \( t \) is set to \( 1 \) and the clevis luks bind command successfully reconstructs the secret if at least one from two listed tang servers is available.

16.15.12.1.2. Example 2: Shared secret on a Tang server and a TPM device

The following command successfully decrypts a LUKS-encrypted device when both the tang server and the tpm2 device are available:

```bash
# clevis luks bind -d /dev/sda1 sss '{"t":2,"pins":{"tang":{"url":"http://tang1.srv"}},
"tpm2":
{"pcr_ids":"0,7"}'}
```

The configuration scheme with the SSS threshold 't' set to '2' is now:

```json
{
  "t":2,
  "pins":{
    "tang":[
      {
        "url":"http://tang1.srv"
      }
    ],
    "tpm2":{
      "pcr_ids":"0,7"
    }
  }
}
```

Additional resources

- tang(8) (section High Availability), clevis(1) (section Shamir's Secret Sharing), and clevis-encrypt-sss(1) man pages
16.15.13. Deployment of virtual machines in a NBDE network

The `clevis luks bind` command does not change the LUKS master key. This implies that if you create a LUKS-encrypted image for use in a virtual machine or cloud environment, all the instances that run this image will share a master key. This is extremely insecure and should be avoided at all times.

This is not a limitation of Clevis but a design principle of LUKS. If you wish to have encrypted root volumes in a cloud, you need to make sure that you perform the installation process (usually using Kickstart) for each instance of Red Hat Enterprise Linux in a cloud as well. The images cannot be shared without also sharing a LUKS master key.

If you intend to deploy automated unlocking in a virtualized environment, Red Hat strongly recommends that you use systems such as lorax or virt-install together with a Kickstart file (see Configuring automated enrollment of LUKS-encrypted volumes using Kickstart) or another automated provisioning tool to ensure that each encrypted VM has a unique master key.

NOTE

Automated unlocking with a TPM 2.0 policy is not supported in a virtual machine.

Additional resources

- `clevis-luks-bind(1)` man page

16.15.14. Building automatically-enrollable VM images for cloud environments using NBDE

Deploying automatically-enrollable encrypted images in a cloud environment can provide a unique set of challenges. Like other virtualization environments, it is recommended to reduce the number of instances started from a single image to avoid sharing the LUKS master key.

Therefore, the best practice is to create customized images that are not shared in any public repository and that provide a base for the deployment of a limited amount of instances. The exact number of instances to create should be defined by deployment’s security policies and based on the risk tolerance associated with the LUKS master key attack vector.

To build LUKS-enabled automated deployments, systems such as Lorax or virt-install together with a Kickstart file should be used to ensure master key uniqueness during the image building process.

Cloud environments enable two Tang server deployment options which we consider here. First, the Tang server can be deployed within the cloud environment itself. Second, the Tang server can be deployed outside of the cloud on independent infrastructure with a VPN link between the two infrastructures.

Deploying Tang natively in the cloud does allow for easy deployment. However, given that it shares infrastructure with the data persistence layer of ciphertext of other systems, it may be possible for both the Tang server’s private key and the Clevis metadata to be stored on the same physical disk. Access to this physical disk permits a full compromise of the ciphertext data.

IMPORTANT

For this reason, Red Hat strongly recommends maintaining a physical separation between the location where the data is stored and the system where Tang is running. This separation between the cloud and the Tang server ensures that the Tang server’s private key cannot be accidentally combined with the Clevis metadata. It also provides local control of the Tang server if the cloud infrastructure is at risk.
CHAPTER 16. SCANNING THE SYSTEM FOR SECURITY COMPLIANCE AND VULNERABILITIES

16.15.15. Deploying Tang as a container
The tang container image provides Tang-server decryption capabilities for Clevis clients that run either
in OpenShift Container Platform (OCP) clusters or in separate virtual machines.
Prerequisites
The podman package and its dependencies are installed on the system.
You have logged in on the registry.redhat.io container catalog using the podman login
registry.redhat.io command. See Red Hat Container Registry Authentication for more
information.
The Clevis client is installed on systems containing LUKS-encrypted volumes that you want to
automatically unlock by using a Tang server.
Procedure
1. Pull the tang container image from the registry.redhat.io registry:
# podman pull registry.redhat.io/rhel8/tang
2. Run the container, specify its port, and specify the path to the Tang keys. The previous example
runs the tang container, specifies the port 7500, and indicates a path to the Tang keys of the
/var/db/tang directory:
# podman run -d -p 7500:7500 -v tang-keys:/var/db/tang --name tang
registry.redhat.io/rhel{ProductNumber}/tang
Note that Tang uses port 80 by default but this may collide with other services such as the
Apache HTTP server.
3. [Optional] For increased security, rotate the Tang keys periodically. You can use the tangdrotate-keys script, for example:
# podman run --rm -v tang-keys:/var/db/tang registry.redhat.io/rhel{ProductNumber}/tang
tangd-rotate-keys -v -d /var/db/tang
Rotated key 'rZAMKAseaXBe0rcKXL1hCCIq-DY.jwk' -> .'rZAMKAseaXBe0rcKXL1hCCIqDY.jwk'
Rotated key 'x1AIpc6WmnCU-CabD8_4q18vDuw.jwk' -> .'x1AIpc6WmnCUCabD8_4q18vDuw.jwk'
Created new key GrMMX_WfdqomIU_4RyjpcdlXb0E.jwk
Created new key _dTTfn17sZZqVAp80u3ygFDHtjk.jwk
Keys rotated successfully.
Verification
On a system that contains LUKS-encrypted volumes for automated unlocking by the presence
of the Tang server, check that the Clevis client can encrypt and decrypt a plain-text message
using Tang:
# echo test | clevis encrypt tang '{"url":"http://localhost:7500"}' | clevis decrypt
The advertisement contains the following signing keys:

439


Additional resources

- podman(1), clevis(1), and tang(8) man pages
- For more details on automated unlocking of LUKS-encrypted volumes using Clevis and Tang, see the Configuring automated unlocking of encrypted volumes using policy-based decryption chapter.

16.15.16. Introduction to the Clevis and Tang system roles

RHEL System Roles is a collection of Ansible roles and modules that provide a consistent configuration interface to remotely manage multiple RHEL systems.

RHEL 8.3 introduced Ansible roles for automated deployments of Policy-Based Decryption (PBD) solutions using Clevis and Tang. The rhel-system-roles package contains these system roles, the related examples, and also the reference documentation.

The nbde_client System Role enables you to deploy multiple Clevis clients in an automated way. Note that the nbde_client role supports only Tang bindings, and you cannot use it for TPM2 bindings at the moment.

The nbde_client role requires volumes that are already encrypted using LUKS. This role supports to bind a LUKS-encrypted volume to one or more Network-Bound (NBDE) servers - Tang servers. You can either preserve the existing volume encryption with a passphrase or remove it. After removing the passphrase, you can unlock the volume only using NBDE. This is useful when a volume is initially encrypted using a temporary key or password that you should remove after the system you provision the system.

If you provide both a passphrase and a key file, the role uses what you have provided first. If it does not find any of these valid, it attempts to retrieve a passphrase from an existing binding.

PBD defines a binding as a mapping of a device to a slot. This means that you can have multiple bindings for the same device. The default slot is slot 1.

The nbde_client role provides also the state variable. Use the present value for either creating a new binding or updating an existing one. Contrary to a clevis luks bind command, you can use state: present also for overwriting an existing binding in its device slot. The absent value removes a specified binding.

Using the nbde_server System Role, you can deploy and manage a Tang server as part of an automated disk encryption solution. This role supports the following features:

- Rotating Tang keys
- Deploying and backing up Tang keys

Additional resources
For a detailed reference on Network-Bound Disk Encryption (NBDE) role variables, install the `rhel-system-roles` package, and see the `README.md` and `README.html` files in the `/usr/share/doc/rhel-system-roles/nbde_client/` and `/usr/share/doc/rhel-system-roles/nbde_server/` directories.

For example system-roles playbooks, install the `rhel-system-roles` package, and see the `/usr/share/ansible/roles/rhel-system-roles.nbde_server/examples/` directories.

For more information on RHEL System Roles, see [Introduction to RHEL System Roles](#).

### 16.15.17. Using the `nbde_server` system role for setting up multiple Tang servers

Follow the steps to prepare and apply an Ansible playbook containing your Tang server settings.

**Prerequisites**

- Access and permissions to one or more managed nodes, which are systems you want to configure with the `nbde_server` System Role.

- Access and permissions to a control node, which is a system from which Red Hat Ansible Engine configures other systems.
  
  On the control node:
  
  - Red Hat Ansible Engine is installed.
  - The `rhel-system-roles` package is installed.
  - An inventory file which lists the managed nodes.

**Procedure**

1. Prepare your playbook containing settings for Tang servers. You can either start from the scratch, or use one of the example playbooks from the `/usr/share/ansible/roles/rhel-system-roles.nbde_server/examples/` directory.

   ```
   # cp /usr/share/ansible/roles/rhel-system-roles.nbde_server/examples/simple_deploy.yml ./my-tang-playbook.yml
   ```

2. Edit the playbook in a text editor of your choice, for example:

   ```
   # vi my-tang-playbook.yml
   ```

3. Add the required parameters. The following example playbook ensures deploying of your Tang server and a key rotation:

   ```
   ---
   - hosts: all
     
     vars:
       nbde_server_rotate_keys: yes
     
     roles:
       - rhel-system-roles.nbde_server
   ```
4. Apply the finished playbook:

```
# ansible-playbook -i host1,host2,host3 my-tang-playbook.yml
```

**IMPORTANT**

To ensure that networking for a Tang pin is available during early boot by using the `grubby` tool on the systems where Clevis is installed:

```
# grubby --update-kernel=ALL --args="rd.neednet=1"
```

Additional resources

- For more information, install the `rhel-system-roles` package, and see the `/usr/share/doc/rhel-system-roles/nbde_server/` and `/usr/share/ansible/roles/rhel-system-roles.nbde_server/` directories.

---

16.15.18. Using the `nbde_client` System Role for setting up multiple Clevis clients

Follow the steps to prepare and apply an Ansible playbook containing your Clevis client settings.

**NOTE**

The `nbde_client` System Role supports only Tang bindings. This means that you cannot use it for TPM2 bindings at the moment.

**Prerequisites**

- Access and permissions to one or more managed nodes, which are systems you want to configure with the `nbde_client` System Role.

- Access and permissions to a control node, which is a system from which Red Hat Ansible Engine configures other systems.
  
  On the control node:
  
  - Red Hat Ansible Engine is installed.
  
  - The `rhel-system-roles` package is installed.
  
  - An inventory file which lists the managed nodes.
  
  - Your volumes are already encrypted by LUKS.

**Procedure**

1. Prepare your playbook containing settings for Clevis clients. You can either start from the scratch, or use one of the example playbooks from the `/usr/share/ansible/roles/rhel-system-roles.nbde_client/examples/` directory.

```
# cp /usr/share/ansible/roles/rhel-system-roles.nbde_client/examples/high_availability.yml . /my-clevis-playbook.yml
```

2. Edit the playbook in a text editor of your choice, for example:
# vi my-clevis-playbook.yml

3. Add the required parameters. The following example playbook configures Clevis clients for automated unlocking of two LUKS-encrypted volumes by when at least one of two Tang servers is available:

```yaml
---
- hosts: all
  vars:
    nbde_client_bindings:
      - device: /dev/rhel/root
        encryption_key_src: /etc/luks/keyfile
        servers:
          - http://server1.example.com
          - http://server2.example.com
      - device: /dev/rhel/swap
        encryption_key_src: /etc/luks/keyfile
        servers:
          - http://server1.example.com
          - http://server2.example.com

  roles:
    - rhel-system-roles.nbde_client
```

4. Apply the finished playbook:

```bash
# ansible-playbook -i host1,host2,host3 my-clevis-playbook.yml
```

**IMPORTANT**

To ensure that networking for a Tang pin is available during early boot by using the `grubby` tool on the system where Clevis is installed:

```bash
# grubby --update-kernel=ALL --args="rd.neednet=1"
```

**Additional resources**

- For details about the parameters and additional information about the `nbde_client` System Role, install the `rhel-system-roles` package, and see the `/usr/share/doc/rhel-system-roles/nbde_client/` and `/usr/share/ansible/roles/rhel-system-roles.nbde_client/` directories.

**16.15.19. Additional resources**

- `tang(8)`, `clevis(1)`, `jose(1)`, and `clevis-luks-unlockers(7)` man pages

- How to set up Network-Bound Disk Encryption with multiple LUKS devices (Clevis + Tang unlocking) Knowledgebase article
CHAPTER 17. USING SELINUX

17.1. GETTING STARTED WITH SELINUX

Security Enhanced Linux (SELinux) provides an additional layer of system security. SELinux fundamentally answers the question: May <subject> do <action> to <object>?, for example: May a web server access files in users' home directories?

17.1.1. Introduction to SELinux

The standard access policy based on the user, group, and other permissions, known as Discretionary Access Control (DAC), does not enable system administrators to create comprehensive and fine-grained security policies, such as restricting specific applications to only viewing log files, while allowing other applications to append new data to the log files.

Security Enhanced Linux (SELinux) implements Mandatory Access Control (MAC). Every process and system resource has a special security label called an SELinux context. A SELinux context, sometimes referred to as an SELinux label, is an identifier which abstracts away the system-level details and focuses on the security properties of the entity. Not only does this provide a consistent way of referencing objects in the SELinux policy, but it also removes any ambiguity that can be found in other identification methods. For example, a file can have multiple valid path names on a system that makes use of bind mounts.

The SELinux policy uses these contexts in a series of rules which define how processes can interact with each other and the various system resources. By default, the policy does not allow any interaction unless a rule explicitly grants access.

\[\text{NOTE}\]

Remember that SELinux policy rules are checked after DAC rules. SELinux policy rules are not used if DAC rules deny access first, which means that no SELinux denial is logged if the traditional DAC rules prevent the access.

SELinux contexts have several fields: user, role, type, and security level. The SELinux type information is perhaps the most important when it comes to the SELinux policy, as the most common policy rule which defines the allowed interactions between processes and system resources uses SELinux types and not the full SELinux context. SELinux types end with \_t. For example, the type name for the web server is \textit{httpd\_t}. The type context for files and directories normally found in /var/www/html/ is \textit{httpd\_sys\_content\_t}. The type contexts for files and directories normally found in /tmp and /var/tmp/ is \textit{tmp\_t}. The type context for web server ports is \textit{http\_port\_t}.

There is a policy rule that permits Apache (the web server process running as \textit{httpd\_t}) to access files and directories with a context normally found in /var/www/html/ and other web server directories (\textit{httpd\_sys\_content\_t}). There is no allow rule in the policy for files normally found in /tmp and /var/tmp/, so access is not permitted. With SELinux, even if Apache is compromised, and a malicious script gains access, it is still not able to access the /tmp directory.
As the previous scheme shows, SELinux allows the Apache process running as `httpd_t` to access the `/var/www/html/` directory and it denies the same process to access the `/data/mysql/` directory because there is no allow rule for the `httpd_t` and `mysqld_db_t` type contexts. On the other hand, the MariaDB process running as `mysqld_t` is able to access the `/data/mysql/` directory and SELinux also correctly denies the process with the `mysqld_t` type to access the `/var/www/html/` directory labeled as `httpd_sys_content_t`.

**Additional resources**

- `selinux(8)` man page and man pages listed by the `apropos selinux` command.
- Man pages listed by the `man -k _selinux` command when the `selinux-policy-doc` package is installed.
- The SELinux Coloring Book helps you to better understand SELinux basic concepts.
- SELinux Wiki FAQ

**17.1.2. Benefits of running SELinux**

SELinux provides the following benefits:

- All processes and files are labeled. SELinux policy rules define how processes interact with files, as well as how processes interact with each other. Access is only allowed if an SELinux policy rule exists that specifically allows it.
- Fine-grained access control. Stepping beyond traditional UNIX permissions that are controlled at user discretion and based on Linux user and group IDs, SELinux access decisions are based on all available information, such as an SELinux user, role, type, and, optionally, a security level.
- SELinux policy is administratively-defined and enforced system-wide.
- Improved mitigation for privilege escalation attacks. Processes run in domains, and are therefore separated from each other. SELinux policy rules define how processes access files and other processes. If a process is compromised, the attacker only has access to the normal functions of that process, and to files the process has been configured to have access to. For example, if the Apache HTTP Server is compromised, an attacker cannot use that process to read files in user home directories, unless a specific SELinux policy rule was added or configured to allow such access.
SELinux can be used to enforce data confidentiality and integrity, as well as protecting processes from untrusted inputs.

However, SELinux is not:

- antivirus software,
- replacement for passwords, firewalls, and other security systems,
- all-in-one security solution.

SELinux is designed to enhance existing security solutions, not replace them. Even when running SELinux, it is important to continue to follow good security practices, such as keeping software up-to-date, using hard-to-guess passwords, and firewalls.

17.1.3. SELinux examples

The following examples demonstrate how SELinux increases security:

- The default action is deny. If an SELinux policy rule does not exist to allow access, such as for a process opening a file, access is denied.

- SELinux can confine Linux users. A number of confined SELinux users exist in the SELinux policy. Linux users can be mapped to confined SELinux users to take advantage of the security rules and mechanisms applied to them. For example, mapping a Linux user to the SELinux user_u user, results in a Linux user that is not able to run unless configured otherwise set user ID (setuid) applications, such as sudo and su.

- Increased process and data separation. The concept of SELinux domains allows defining which processes can access certain files and directories. For example, when running SELinux, unless otherwise configured, an attacker cannot compromise a Samba server, and then use that Samba server as an attack vector to read and write to files used by other processes, such as MariaDB databases.

- SELinux helps mitigate the damage made by configuration mistakes. Domain Name System (DNS) servers often replicate information between each other in what is known as a zone transfer. Attackers can use zone transfers to update DNS servers with false information. When running the Berkeley Internet Name Domain (BIND) as a DNS server in Red Hat Enterprise Linux, even if an administrator forgets to limit which servers can perform a zone transfer, the default SELinux policy prevents zone files \[2\] from being updated using zone transfers, by the BIND named daemon itself, and by other processes.

17.1.4. SELinux architecture and packages

SELinux is a Linux Security Module (LSM) that is built into the Linux kernel. The SELinux subsystem in the kernel is driven by a security policy which is controlled by the administrator and loaded at boot. All security-relevant, kernel-level access operations on the system are intercepted by SELinux and examined in the context of the loaded security policy. If the loaded policy allows the operation, it continues. Otherwise, the operation is blocked and the process receives an error.

SELinux decisions, such as allowing or disallowing access, are cached. This cache is known as the Access Vector Cache (AVC). When using these cached decisions, SELinux policy rules need to be checked less, which increases performance. Remember that SELinux policy rules have no effect if DAC rules deny access first. Raw audit messages are logged to the /var/log/audit/audit.log and they start with the type=AVC string.
In RHEL 8, system services are controlled by the `systemd` daemon; `systemd` starts and stops all services, and users and processes communicate with `systemd` using the `systemctl` utility. The `systemd` daemon can consult the SELinux policy and check the label of the calling process and the label of the unit file that the caller tries to manage, and then ask SELinux whether or not the caller is allowed the access. This approach strengthens access control to critical system capabilities, which include starting and stopping system services.

The `systemd` daemon also works as an SELinux Access Manager. It retrieves the label of the process running `systemctl` or the process that sent a D-Bus message to `systemd`. The daemon then looks up the label of the unit file that the process wanted to configure. Finally, `systemd` can retrieve information from the kernel if the SELinux policy allows the specific access between the process label and the unit file label. This means a compromised application that needs to interact with `systemd` for a specific service can now be confined by SELinux. Policy writers can also use these fine-grained controls to confine administrators.

If a process is sending a D-Bus message to another process and if the SELinux policy does not allow the D-Bus communication of these two processes, then the system prints a USER_AVC denial message, and the D-Bus communication times out. Note that the D-Bus communication between two processes works bidirectionally.

**IMPORTANT**

To avoid incorrect SELinux labeling and subsequent problems, ensure that you start services using a `systemctl start` command.

RHEL 8 provides the following packages for working with SELinux:

- **policies:** `selinux-policy-targeted`, `selinux-policy-mls`
- **tools:** `policycoreutils`, `policycoreutils-gui`, `libselinux-utils`, `policycoreutils-python-utils`, `setools-console`, `checkpolicy`

### 17.1.5. SELinux states and modes

SELinux can run in one of three modes: enforcing, permissive, or disabled.

- Enforcing mode is the default, and recommended, mode of operation; in enforcing mode SELinux operates normally, enforcing the loaded security policy on the entire system.

- In permissive mode, the system acts as if SELinux is enforcing the loaded security policy, including labeling objects and emitting access denial entries in the logs, but it does not actually deny any operations. While not recommended for production systems, permissive mode can be helpful for SELinux policy development and debugging.

- Disabled mode is strongly discouraged; not only does the system avoid enforcing the SELinux policy, it also avoids labeling any persistent objects such as files, making it difficult to enable SELinux in the future.

Use the `setenforce` utility to change between enforcing and permissive mode. Changes made with `setenforce` do not persist across reboots. To change to enforcing mode, enter the `setenforce 1` command as the Linux root user. To change to permissive mode, enter the `setenforce 0` command. Use the `getenforce` utility to view the current SELinux mode:

```
# getenforce
Enforcing
```
In Red Hat Enterprise Linux, you can set individual domains to permissive mode while the system runs in enforcing mode. For example, to make the httpd_t domain permissive:

```
# semanage permissive -a httpd_t
```

Note that permissive domains are a powerful tool that can compromise security of your system. Red Hat recommends to use permissive domains with caution, for example, when debugging a specific scenario.

### 17.2. CHANGING SELINUX STATES AND MODES

When enabled, SELinux can run in one of two modes: enforcing or permissive. The following sections show how to permanently change into these modes.

#### 17.2.1. Permanent changes in SELinux states and modes

As discussed in SELinux states and modes, SELinux can be enabled or disabled. When enabled, SELinux has two modes: enforcing and permissive.

Use the `getenforce` or `sestatus` commands to check in which mode SELinux is running. The `getenforce` command returns `Enforcing`, `Permissive`, or `Disabled`.

The `sestatus` command returns the SELinux status and the SELinux policy being used:

```
$ sestatus
SELinux status: enabled
SELinuxfs mount: /sys/fs/selinux
SELinux root directory: /etc/selinux
Loaded policy name: targeted
Current mode: enforcing
Mode from config file: enforcing
Policy MLS status: enabled
Policy deny_unknown status: allowed
Memory protection checking: actual (secure)
Max kernel policy version: 31
```
17.2.2. Changing to permissive mode

Use the following procedure to permanently change SELinux mode to permissive. When SELinux is running in permissive mode, SELinux policy is not enforced. The system remains operational and SELinux does not deny any operations but only logs AVC messages, which can be then used for troubleshooting, debugging, and SELinux policy improvements. Each AVC is logged only once in this case.

Prerequisites

- The `selinux-policy-targeted`, `libselinux-utils`, and `policycoreutils` packages are installed on your system.
- The `selinux=0` or `enforcing=0` kernel parameters are not used.

Procedure

1. Open the `/etc/selinux/config` file in a text editor of your choice, for example:

   ```bash
   # vi /etc/selinux/config
   ```

2. Configure the `SELINUX=permissive` option:

   ```bash
   # This file controls the state of SELinux on the system.
   # SELINUX= can take one of these three values:
   # enforcing - SELinux security policy is enforced.
   # permissive - SELinux prints warnings instead of enforcing.
   # disabled - No SELinux policy is loaded.
   SELINUX=permissive
   # SELINUXTYPE= can take one of these two values:
   ```
3. Restart the system:

```
# reboot
```

Verification

1. After the system restarts, confirm that the `getenforce` command returns **Permissive**:

```
$ getenforce
Permissive
```

17.2.3. Changing to enforcing mode

Use the following procedure to switch SELinux to enforcing mode. When SELinux is running in enforcing mode, it enforces the SELinux policy and denies access based on SELinux policy rules. In RHEL, enforcing mode is enabled by default when the system was initially installed with SELinux.

Prerequisites

- The `selinux-policy-targeted`, `libselinux-utils`, and `policycoreutils` packages are installed on your system.
- The `selinux=0` or `enforcing=0` kernel parameters are not used.

Procedure

1. Open the `/etc/selinux/config` file in a text editor of your choice, for example:

```
# vi /etc/selinux/config
```

2. Configure the `SELINUX=enforcing` option:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

3. Save the change, and restart the system:

```
# reboot
```

On the next boot, SELinux relabels all the files and directories within the system and adds SELinux context for files and directories that were created when SELinux was disabled.
CHAPTER 17. USING SELINUX

Verification

1. After the system restarts, confirm that the `getenforce` command returns `Enforcing`:

$ getenforce
Enforcing

NOTE

After changing to enforcing mode, SELinux may deny some actions because of incorrect or missing SELinux policy rules. To view what actions SELinux denies, enter the following command as root:

# ausearch -m AVC,USER_AVC,SE Linux_ERR,USER_SELINUX_ERR -ts today

Alternatively, with the `setroubleshoot-server` package installed, enter:

# grep "SELinux is preventing" /var/log/messages

If SELinux is active and the Audit daemon (`auditd`) is not running on your system, then search for certain SELinux messages in the output of the `dmesg` command:

# dmesg | grep -i -e type=1300 -e type=1400

See Troubleshooting problems related to SELinux for more information.

17.2.4. Enabling SELinux on systems that previously had it disabled

To avoid problems, such as systems unable to boot or process failures, follow this procedure when enabling SELinux on systems that previously had it disabled.

WARNING

When systems run SELinux in permissive mode, users and processes might label various file-system objects incorrectly. File-system objects created while SELinux is disabled are not labeled at all. This behavior causes problems when changing to enforcing mode because SELinux relies on correct labels of file-system objects.

To prevent incorrectly labeled and unlabeled files from causing problems, SELinux automatically relabels file systems when changing from the disabled state to permissive or enforcing mode.

Before rebooting the system for relabeling, make sure the system will boot in permissive mode, for example by using the `enforcing=0` kernel option. This prevents the system from failing to boot in case the system contains unlabeled files required by `systemd` before launching the `selinux-autorelabel` service. For more information, see RHBZ#2021835.

Procedure
1. Enable SELinux in permissive mode. For more information, see Changing to permissive mode.

2. Restart your system:

   ```
   # reboot
   ```

3. Check for SELinux denial messages. For more information, see Identifying SELinux denials.

4. Ensure that files are relabeled upon the next reboot:

   ```
   # fixfiles -F onboot
   ```

   This creates the `.autorelabel` file containing the `-F` option.

   **WARNING**
   
   Always switch to permissive mode before entering the `fixfiles -F onboot` command. This prevents the system from failing to boot in case the system contains unlabeled files. For more information, see RHBZ#2021835.

5. If there are no denials, switch to enforcing mode. For more information, see Changing SELinux modes at boot time.

**Verification**

1. After the system restarts, confirm that the `getenforce` command returns **Enforcing**:

   ```
   $ getenforce
   Enforcing
   ```

   **NOTE**

   To run custom applications with SELinux in enforcing mode, choose one of the following scenarios:

   - Run your application in the `unconfined_service_t` domain.
   - Write a new policy for your application. See the Writing a custom SELinux policy section for more information.

**Additional resources**

- SELinux states and modes section covers temporary changes in modes.

### 17.2.5. Disabling SELinux

Use the following procedure to permanently disable SELinux.
IMPORTANT

When SELinux is disabled, SELinux policy is not loaded at all; it is not enforced and AVC messages are not logged. Therefore, all benefits of running SELinux are lost.

Red Hat strongly recommends to use permissive mode instead of permanently disabling SELinux. See Changing to permissive mode for more information about permissive mode.

WARNING

Disabling SELinux using the SELINUX=disabled option in the /etc/selinux/config results in a process in which the kernel boots with SELinux enabled and switches to disabled mode later in the boot process. Because memory leaks and race conditions causing kernel panics can occur, prefer disabling SELinux by adding the selinux=0 parameter to the kernel command line as described in Changing SELinux modes at boot time if your scenario really requires to completely disable SELinux.

Procedure

1. Open the /etc/selinux/config file in a text editor of your choice, for example:

   # vi /etc/selinux/config

2. Configure the SELINUX=disabled option:

   # This file controls the state of SELinux on the system.
   # SELINUX= can take one of these three values:
   #     enforcing - SELinux security policy is enforced.
   #     permissive - SELinux prints warnings instead of enforcing.
   #     disabled - No SELinux policy is loaded.
   SELINUX=disabled
   # SELINUXTYPE= can take one of these two values:
   #     targeted - Targeted processes are protected,
   #     mls - Multi Level Security protection.
   SELINUXTYPE=targeted

3. Save the change, and restart your system:

   # reboot

Verification

1. After reboot, confirm that the getenforce command returns Disabled:

   $ getenforce
   Disabled

17.2.6. Changing SELinux modes at boot time
On boot, you can set several kernel parameters to change the way SELinux runs:

enforcing=0

Setting this parameter causes the system to start in permissive mode, which is useful when troubleshooting issues. Using permissive mode might be the only option to detect a problem if your file system is too corrupted. Moreover, in permissive mode, the system continues to create the labels correctly. The AVC messages that are created in this mode can be different than in enforcing mode. In permissive mode, only the first denial from a series of the same denials is reported. However, in enforcing mode, you might get a denial related to reading a directory, and an application stops. In permissive mode, you get the same AVC message, but the application continues reading files in the directory and you get an AVC for each denial in addition.

selinux=0

This parameter causes the kernel to not load any part of the SELinux infrastructure. The init scripts notice that the system booted with the selinux=0 parameter and touch the .autorelabel file. This causes the system to automatically relabel the next time you boot with SELinux enabled.

IMPORTANT

Red Hat does not recommend using the selinux=0 parameter. To debug your system, prefer using permissive mode.

autorelabel=1

This parameter forces the system to relabel similarly to the following commands:

```
# touch /.autorelabel
# reboot
```

If a file system contains a large amount of mislabeled objects, start the system in permissive mode to make the autorelabel process successful.

Additional resources

- For additional SELinux-related kernel boot parameters, such as checkreqprot, see the /usr/share/doc/kernel-doc-<KERNEL_VER>/Documentation/admin-guide/kernel-parameters.txt file installed with the kernel-doc package. Replace the <KERNEL_VER> string with the version number of the installed kernel, for example:

```
# yum install kernel-doc
$ less /usr/share/doc/kernel-doc-4.18.0/Documentation/admin-guide/kernel-parameters.txt
```

17.3. TROUBLESHOOTING PROBLEMS RELATED TO SELINUX

If you plan to enable SELinux on systems where it has been previously disabled or if you run a service in a non-standard configuration, you might need to troubleshoot situations potentially blocked by SELinux. Note that in most cases, SELinux denials are signs of misconfiguration.

17.3.1. Identifying SELinux denials

Follow only the necessary steps from this procedure; in most cases, you need to perform just step 1.
Procedure

1. When your scenario is blocked by SELinux, the /var/log/audit/audit.log file is the first place to check for more information about a denial. To query Audit logs, use the ausearch tool. Because the SELinux decisions, such as allowing or disallowing access, are cached and this cache is known as the Access Vector Cache (AVC), use the AVC and USER_AVC values for the message type parameter, for example:

   ```
   # ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR -ts recent
   ```

   If there are no matches, check if the Audit daemon is running. If it does not, repeat the denied scenario after you start `auditd` and check the Audit log again.

2. In case `auditd` is running, but there are no matches in the output of `ausearch`, check messages provided by the `systemd` Journal:

   ```
   # journalctl -t setroubleshoot
   ```

3. If SELinux is active and the Audit daemon is not running on your system, then search for certain SELinux messages in the output of the `dmesg` command:

   ```
   # dmesg | grep -i -e type=1300 -e type=1400
   ```

4. Even after the previous three checks, it is still possible that you have not found anything. In this case, AVC denials can be silenced because of dontaudit rules. To temporarily disable dontaudit rules, allowing all denials to be logged:

   ```
   # semodule -DB
   ```

   After re-running your denied scenario and finding denial messages using the previous steps, the following command enables dontaudit rules in the policy again:

   ```
   # semodule -B
   ```

5. If you apply all four previous steps, and the problem still remains unidentified, consider if SELinux really blocks your scenario:

   - Switch to permissive mode:
     ```
     # setenforce 0
     $ getenforce
     Permissive
     ```

   - Repeat your scenario.

   If the problem still occurs, something different than SELinux is blocking your scenario.

17.3.2. Analyzing SELinux denial messages

After identifying that SELinux is blocking your scenario, you might need to analyze the root cause before you choose a fix.

Prerequisites
The `policycoreutils-python-utils` and `setroubleshoot-server` packages are installed on your system.

**Procedure**

1. List more details about a logged denial using the `sealert` command, for example:

   ```bash
   $ sealert -l "*
   SELinux is preventing /usr/bin/passwd from write access on the file /root/test.

   *****  Plugin leaks (86.2 confidence) suggests *****************************

   If you want to ignore passwd trying to write access the test file, because you believe it should not need this access. Then you should report this as a bug.

   You can generate a local policy module to dontaudit this access.
   Do
   # ausearch -x /usr/bin/passwd --raw | audit2allow -D -M my-passwd
   # semodule -X 300 -i my-passwd.pp

   *****  Plugin catchall (14.7 confidence) suggests ****************************

   Raw Audit Messages
   type=AVC msg=audit(1553609555.619:127): avc: denied { write } for
   pid=4097 comm="passwd" path="/root/test" dev="dm-0" ino=17142697
   scontext=unconfined_u:unconfined_r:passwd_t:s0-s0:c0.c1023
   tcontext=unconfined_u:object_r:admin_home_t:s0
   tclass=file permissive=0

   Hash: passwd,passwd_t,admin_home_t,file,write
   
2. If the output obtained in the previous step does not contain clear suggestions:

   - Enable full-path auditing to see full paths to accessed objects and to make additional Linux Audit event fields visible:
     
     ```bash
     # auditctl -w /etc/shadow -p w -k shadow-write
     
     # Clear the `setroubleshoot` cache:
     
     ```bash
     # rm -f /var/lib/setroubleshoot/setroubleshoot.xml
     
     # Reproduce the problem.
     
     # Repeat step 1.
     After you finish the process, disable full-path auditing:
     
     ```bash
     # auditctl -W /etc/shadow -p w -k shadow-write
     ```
3. If `sealert` returns only catchall suggestions or suggests adding a new rule using the `audit2allow` tool, match your problem with examples listed and explained in SE Linux denials in the Audit log.

Additional resources

- `sealert(8)` man page

17.3.3. Fixing analyzed SE Linux denials

In most cases, suggestions provided by the `sealert` tool give you the right guidance about how to fix problems related to the SE Linux policy. See Analyzing SE Linux denial messages for information how to use `sealert` to analyze SE Linux denials.

Be careful when the tool suggests using the `audit2allow` tool for configuration changes. You should not use `audit2allow` to generate a local policy module as your first option when you see an SE Linux denial. Troubleshooting should start with a check if there is a labeling problem. The second most often case is that you have changed a process configuration, and you forgot to tell SE Linux about it.

Labeling problems

A common cause of labeling problems is when a non-standard directory is used for a service. For example, instead of using `/var/www/html` for a website, an administrator might want to use `/srv/myweb/`. On Red Hat Enterprise Linux, the `/srv` directory is labeled with the `var_t` type. Files and directories created in `/srv` inherit this type. Also, newly-created objects in top-level directories, such as `/myserver`, can be labeled with the `default_t` type. SE Linux prevents the Apache HTTP Server (`httpd`) from accessing both of these types. To allow access, SE Linux must know that the files in `/srv/myweb/` are to be accessible by `httpd`:

```
# semanage fcontext -a -t httpd_sys_content_t "/srv/myweb/(.*)"
```

This `semanage` command adds the context for the `/srv/myweb/` directory and all files and directories under it to the SE Linux file-context configuration. The `semanage` utility does not change the context. As root, use the `restorecon` utility to apply the changes:

```
# restorecon -R -v /srv/myweb
```

Incorrect context

The `matchpathcon` utility checks the context of a file path and compares it to the default label for that path. The following example demonstrates the use of `matchpathcon` on a directory that contains incorrectly labeled files:

```
$ matchpathcon -V /var/www/html/*
/var/www/html/index.html has context unconfined_u:object_r:user_home_t:s0, should be system_u:object_r:httpd_sys_content_t:s0
/var/www/html/page1.html has context unconfined_u:object_r:user_home_t:s0, should be system_u:object_r:httpd_sys_content_t:s0
```

In this example, the `index.html` and `page1.html` files are labeled with the `user_home_t` type. This type is used for files in user home directories. Using the `mv` command to move files from your home directory may result in files being labeled with the `user_home_t` type. This type should not exist outside of home directories. Use the `restorecon` utility to restore such files to their correct type:
To restore the context for all files under a directory, use the `-R` option:

```bash
# restorecon -R -v /var/www/html/
restorecon reset /var/www/html/page1.html context unconfined_u:object_r:samba_share_t:s0->system_u:object_r:httpd_sys_content_t:s0
restorecon reset /var/www/html/index.html context unconfined_u:object_r:samba_share_t:s0->system_u:object_r:httpd_sys_content_t:s0
```

Confined applications configured in non-standard ways

Services can be run in a variety of ways. To account for that, you need to specify how you run your services. You can achieve this through SELinux booleans that allow parts of SELinux policy to be changed at runtime. This enables changes, such as allowing services access to NFS volumes, without reloading or recompiling SELinux policy. Also, running services on non-default port numbers requires policy configuration to be updated using the `semanage` command.

For example, to allow the Apache HTTP Server to communicate with MariaDB, enable the `httpd_can_network_connect_db` boolean:

```bash
# setsebool -P httpd_can_network_connect_db on
```

Note that the `-P` option makes the setting persistent across reboots of the system.

If access is denied for a particular service, use the `getsebool` and `grep` utilities to see if any booleans are available to allow access. For example, use the `getsebool -a | grep ftp` command to search for FTP related booleans:

```bash
$ getsebool -a | grep ftp
ftp_anon_write --> off
ftp_full_access --> off
ftp_use_cifs --> off
ftp_use_nfs --> off
ftp_connect_db --> off
http_enable_ftp_server --> off
tftp_anon_write --> off
```

To get a list of booleans and to find out if they are enabled or disabled, use the `getsebool -a` command. To get a list of booleans including their meaning, and to find out if they are enabled or disabled, install the `selinux-policy-devel` package and use the `semanage boolean -l` command as root.

Port numbers

Depending on policy configuration, services can only be allowed to run on certain port numbers. Attempting to change the port a service runs on without changing policy may result in the service failing to start. For example, run the `semanage port -l | grep http` command as root to list `http` related ports:

```bash
# semanage port -l | grep http
http_cache_port_t tcp 3128, 8080, 8118
http_cache_port_t udp 3130
```
The `http_port_t` port type defines the ports Apache HTTP Server can listen on, which in this case, are TCP ports 80, 443, 488, 8008, 8009, and 8443. If an administrator configures `httpd.conf` so that `httpd` listens on port 9876 (Listen 9876), but policy is not updated to reflect this, the following command fails:

```
# systemctl start httpd.service
```

```
Job for httpd.service failed. See 'systemctl status httpd.service' and 'journalctl -xn' for details.
```

```
# systemctl status httpd.service
```

```
httpd.service - The Apache HTTP Server
 Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled)
 Active: failed (Result: exit-code) since Thu 2013-08-15 09:57:05 CEST; 59s ago
 Process: 16874 ExecStop=/usr/sbin/httpd $OPTIONS -k graceful-stop (code=exited, status=0/SUCCESS)
 Process: 16870 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=1/FAILURE)
```

An SELinux denial message similar to the following is logged to `/var/log/audit/audit.log`:

```
type=AVC msg=audit(1225948455.061:294): avc: denied { name_bind } for pid=4997
comm="httpd" src=9876 scontext=unconfined_u:system_r:httpd_t:s0
tcontext=system_u:object_r:port_t:s0 tclass=tcp_socket
```

To allow `httpd` to listen on a port that is not listed for the `http_port_t` port type, use the `semanage port` command to assign a different label to the port:

```
# semanage port -a -t http_port_t -p tcp 9876
```

The `-a` option adds a new record; the `-t` option defines a type; and the `-p` option defines a protocol. The last argument is the port number to add.

**Corner cases, evolving or broken applications, and compromised systems**

Applications may contain bugs, causing SELinux to deny access. Also, SELinux rules are evolving – SELinux may not have seen an application running in a certain way, possibly causing it to deny access, even though the application is working as expected. For example, if a new version of PostgreSQL is released, it may perform actions the current policy does not account for, causing access to be denied, even though access should be allowed.

For these situations, after access is denied, use the `audit2allow` utility to create a custom policy module to allow access. You can report missing rules in the SELinux policy in Red Hat Bugzilla. For Red Hat Enterprise Linux 8, create bugs against the Red Hat Enterprise Linux 8 product, and select the `selinux-policy` component. Include the output of the `audit2allow -w -a` and `audit2allow -a` commands in such bug reports.

If an application asks for major security privileges, it could be a signal that the application is compromised. Use intrusion detection tools to inspect such suspicious behavior.

The **Solution Engine** on the Red Hat Customer Portal can also provide guidance in the form of an article containing a possible solution for the same or very similar problem you have. Select the relevant product and version and use SELinux-related keywords, such as `selinux` or `avc`, together with the name of your blocked service or application, for example: `selinux samba`. 
17.3.4. SELinux denials in the Audit log

The Linux Audit system stores log entries in the `/var/log/audit/audit.log` file by default.

To list only SELinux-related records, use the `ausearch` command with the message type parameter set to `AVC` and `AVC_USER` at a minimum, for example:

```
# ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR
```

An SELinux denial entry in the Audit log file can look as follows:

```
type=AVC msg=audit(1395177286.929:1638): avc: denied { read } for pid=6591 comm="httpd" name="webpages" dev="0:37" ino=2112 scontext=system_u:system_r:httpd_t:s0 tcontext=system_u:object_r:nfs_t:s0 tclass=dir
```

The most important parts of this entry are:

- **avc: denied** - the action performed by SELinux and recorded in Access Vector Cache (AVC)
- **{ read }** - the denied action
- **pid=6591** - the process identifier of the subject that tried to perform the denied action
- **comm="httpd"** - the name of the command that was used to invoke the analyzed process
- **httpd_t** - the SELinux type of the process
- **nfs_t** - the SELinux type of the object affected by the process action
- **tclass=dir** - the target object class

The previous log entry can be translated to:

**SELinux denied the `httpd` process with PID 6591 and the `httpd_t` type to read from a directory with the `nfs_t` type.**

The following SELinux denial message occurs when the Apache HTTP Server attempts to access a directory labeled with a type for the Samba suite:

```
type=AVC msg=audit(1226874073.147:96): avc: denied { getattr } for pid=2465 comm="httpd" path="/var/www/html/file1" dev=dm-0 ino=284133 scontext=unconfined_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=file
```

- **{ getattr }** - the `getattr` entry indicates the source process was trying to read the target file’s status information. This occurs before reading files. SELinux denies this action because the process accesses the file and it does not have an appropriate label. Commonly seen permissions include `getattr`, `read`, and `write`.
- **path="/var/www/html/file1"** - the path to the object (target) the process attempted to access.
- **scontext="unconfined_u:system_r:httpd_t:s0"** - the SELinux context of the process (source) that attempted the denied action. In this case, it is the SELinux context of the Apache HTTP Server, which is running with the `httpd_t` type.
- **tcontext="unconfined_u:object_r:samba_share_t:s0"** - the SELinux context of the object (target) the process attempted to access. In this case, it is the SELinux context of `file1`.
This SELinux denial can be translated to:

SELinux denied the **httpd** process with PID 2465 to access the `/var/www/html/file1` file with the **samba_share_t** type, which is not accessible to processes running in the **httpd_t** domain unless configured otherwise.

Additional resources

- **auditd**(8) and **ausearch**(8) man pages

17.3.5. Additional resources

- Basic SELinux Troubleshooting in CLI
- What is SELinux trying to tell me? The 4 key causes of SELinux errors

---

[2] Text files that include information, such as host name to IP address mappings, that are used by DNS servers.
PART III. DESIGN OF NETWORK
CHAPTER 18. USING NETCONSOLE TO LOG KERNEL MESSAGES OVER A NETWORK

Using the netconsole kernel module and the same-named service, you can log kernel messages over a network to debug the kernel when logging to disk fails or when using a serial console is not possible.

18.1. CONFIGURING THE NETCONSOLE SERVICE TO LOG KERNEL MESSAGES TO A REMOTE HOST

Using the netconsole kernel module, you can log kernel messages to a remote system log service.

**Prerequisites**

- A system log service, such as rsyslog is installed on the remote host.
- The remote system log service is configured to receive incoming log entries from this host.

**Procedure**

1. Install the netconsole-service package:
   
   ```bash
   # yum install netconsole-service
   ```

2. Edit the `/etc/sysconfig/netconsole` file and set the SYSLOGADDR parameter to the IP address of the remote host:
   
   ```bash
   # SYSLOGADDR=192.0.2.1
   ```

3. Enable and start the netconsole service:
   
   ```bash
   # systemctl enable --now netconsole
   ```

**Verification steps**

- Display the `/var/log/messages` file on the remote system log server.

**Additional resources**

- Configuring a remote logging solution
CHAPTER 19. GETTING STARTED WITH NETWORKMANAGER

By default, RHEL uses NetworkManager to manage the network configuration and connections.

19.1. BENEFITS OF USING NETWORKMANAGER

The main benefits of using NetworkManager are:

- Offering an API through D-Bus which allows to query and control network configuration and state. In this way, networking can be checked and configured by multiple applications ensuring a synced and up-to-date networking status. For example, the RHEL web console, which monitors and configures servers through a web browser, uses the NetworkManager D-BUS interface to configure networking, as well as the Gnome GUI, the nmcli and the nm-connection-editor tools. Each change made in one of these tools is detected by all the others.

- Making Network management easier: NetworkManager ensures that network connectivity works. When it detects that there is no network configuration in a system but there are network devices, NetworkManager creates temporary connections to provide connectivity.

- Providing easy setup of connection to the user: NetworkManager offers management through different tools – GUI, nmtui, nmcli.

- Supporting configuration flexibility. For example, configuring a WiFi interface, NetworkManager scans and shows the available wifi networks. You can select an interface, and NetworkManager displays the required credentials providing automatic connection after the reboot process. NetworkManager can configure network aliases, IP addresses, static routes, DNS information, and VPN connections, as well as many connection-specific parameters. You can modify the configuration options to reflect your needs.

- Maintaining the state of devices after the reboot process and taking over interfaces which are set into managed mode during restart.

- Handling devices which are not explicitly set unmanaged but controlled manually by the user or another network service.

Additional resources

- Managing systems using the RHEL 8 web console.

19.2. AN OVERVIEW OF UTILITIES AND APPLICATIONS YOU CAN USE TO MANAGE NETWORKMANAGER CONNECTIONS

You can use the following utilities and applications to manage NetworkManager connections:

- **nmcli**: A command-line utility to manage connections.

- **nmtui**: A curses-based text user interface (TUI). To use this application, install the NetworkManager-tui package.

- **nm-connection-editor**: A graphical user interface (GUI) for NetworkManager-related tasks. To start this application, enter nm-connection-editor in a terminal of a GNOME session.

- **control-center**: A GUI provided by the GNOME shell for desktop users. Note that this application supports less features than nm-connection-editor.
The network connection icon in the GNOME shell: This icon represents network connection states and serves as visual indicator for the type of connection you are using.

Additional resources

- Using nmtui to manage network connections using a text-based interface
- Getting started with nmcli

19.3. LOADING MANUALLY-CREATED IFCFG FILES INTO NETWORKMANAGER

In Red Hat Enterprise Linux, if you edit an ifcfg file, NetworkManager is not automatically aware of the change and has to be prompted to notice the change. If you use one of the tools to update NetworkManager profile settings, NetworkManager does not implement those changes until you reconnect using that profile. For example, if configuration files have been changed using an editor, NetworkManager must read the configuration files again.

IMPORTANT

NetworkManager supports profiles stored in the key file format. However, by default, NetworkManager uses the ifcfg format when you use the NetworkManager API to create or update profiles.

In a future major RHEL release, the key file format will be default. Consider using the key file format if you want to manually create and manage configuration files. For details, see Manually creating NetworkManager profiles in key file format.

The /etc/sysconfig/ directory is a location for configuration files and scripts. Most network configuration information is stored there, with the exception of VPN, mobile broadband and PPPoE configuration, which are stored in the /etc/NetworkManager/ subdirectories. For example, interface-specific information is stored in the ifcfg files in the /etc/sysconfig/network-scripts/ directory.

Information for VPNs, mobile broadband and PPPoE connections is stored in /etc/NetworkManager/system-connections/.

Procedure

1. To load a new configuration file:

   # nmcli connection load /etc/sysconfig/network-scripts/ifcfg-connection_name

2. If you updated a connection file that has already been loaded into NetworkManager, enter:

   # nmcli connection up connection_name

Additional resources

- NetworkManager(8) man page
- NetworkManager.conf(5) man page
- /usr/share/doc/initscripts/sysconfig.txt
• `ifcfg(8)` man page
CHAPTER 20. USING NMTUI TO MANAGE NETWORK CONNECTIONS USING A TEXT-BASED INTERFACE

The `nmtui` application is a text user interface (TUI) for NetworkManager. The following section provides how you can configure a network interface using `nmtui`.

NOTE

The `nmtui` application does not support all connection types. In particular, you cannot add or modify VPN connections or Ethernet connections that require 802.1X authentication.

20.1. STARTING THE NMTUI UTILITY

This procedure describes how to start the NetworkManager text user interface, `nmtui`.

Prerequisites

- The `NetworkManager-tui` package is installed.

Procedure

1. To start `nmtui`, enter:

   ```
   # nmtui
   ```

2. To navigate:

   - Use the cursors or press `Tab` to step forwards and press `Shift+Tab` to step back through the options.
   - Use `Enter` to select an option.
   - Use the `Space` bar to toggle the status of check boxes.

20.2. ADDING A CONNECTION PROFILE USING NMTUI
The `nmtui` application provides a text user interface to NetworkManager. This procedure describes how to add a new connection profile.

**Prerequisites**

- The `NetworkManager-tui` package is installed.

**Procedure**

1. Start the NetworkManager text user interface utility:
   
   ```
   # nmtui
   ```

2. Select the **Edit a connection** menu entry, and press **Enter**.

3. Select the **Add** button, and press **Enter**.

4. Select **Ethernet**, and press **Enter**.

5. Fill the fields with the connection details.
6. Select **OK** to save the changes.

7. Select **Back** to return to the main menu.

8. Select **Activate a connection**, and press **Enter**.

9. Select the new connection entry, and press **Enter** to activate the connection.

10. Select **Back** to return to the main menu.

11. Select **Quit**.

**Verification steps**

1. Display the status of the devices and connections:
# nmcli device status

```
DEVICE       TYPE      STATE      CONNECTION
enp1s0       ethernet  connected  Example-Connection
```

2. To display all settings of the connection profile:

```
# nmcli connection show Example-Connection
connection.id:      Example-Connection
connection.uuid:    b6cdfa1c-e4ad-46e5-af8b-a75f06b79f76
connection.stable-id:  --
connection.type:     802-3-ethernet
connection.interface-name: enp1s0
... 
```

If the configuration on the disk does not match the configuration on the device, starting or restarting NetworkManager creates an in-memory connection that reflects the configuration of the device. For further details and how to avoid this problem, see NetworkManager duplicates a connection after restart of NetworkManager service.

**Additional resources**

- Testing basic network settings
- `nmtui(1)` man page

## 20.3. APPLYING CHANGES TO A MODIFIED CONNECTION USING NMTUI

After you modified a connection in `nmtui`, you must reactivate the connection. Note that reactivating a connection in `nmtui` temporarily deactivates the connection.

**Prerequisites**

- The connection profile does not have the auto-connect setting enabled.

**Procedure**

1. In the main menu, select the **Activate a connection** menu entry:
2. Select the modified connection.

3. On the right, select the **Deactivate** button, and press **Enter**:

4. Select the connection again.

5. On the right, select the **Activate** button, and press **Enter**.
CHAPTER 21. GETTING STARTED WITH NMCLI

This section describes general information about the `nmcli` utility.

21.1. THE DIFFERENT OUTPUT FORMATS OF NMCLI

The `nmcli` utility supports different options to modify the output of `nmcli` commands. Using these options, you can display only the required information. This simplifies processing the output in scripts.

By default, the `nmcli` utility displays its output in a table-like format:

```bash
# nmcli device
DEVICE  TYPE      STATE      CONNECTION
enp1s0  ethernet  connected  enp1s0
lo      loopback  unmanaged  --
```

Using the `-f` option, you can display specific columns in a custom order. For example, to display only the `DEVICE` and `STATE` column, enter:

```bash
# nmcli -f DEVICE,STATE device
DEVICE  STATE
enp1s0  connected
lo      unmanaged
```

The `-t` option enables you to display the individual fields of the output in a colon-separated format:

```bash
# nmcli -t device
enp1s0:ethernet:connected:enp1s0
lo:loopback:unmanaged:
```

Combining the `-f` and `-t` to display only specific fields in colon-separated format can be helpful when you process the output in scripts:

```bash
# nmcli -f DEVICE,STATE -t device
enp1s0:connected
lo:unmanaged
```

21.2. USING TAB COMPLETION IN NMCLI

If the `bash-completion` package is installed on your host, the `nmcli` utility supports tab completion. This enables you to auto-complete option names and to identify possible options and values.

For example, if you type `nmcli con` and press Tab, then the shell automatically completes the command to `nmcli connection`.

For the completion, the options or value you have typed must be unique. If it is not unique, then `nmcli` displays all possibilities. For example, if you type `nmcli connection d` and press Tab, then the command shows command `delete` and `down` as possible options.

You can also use tab completion to display all properties you can set in a connection profile. For example, if you type `nmcli connection modify connection_name` and press Tab, the command shows the full list of available properties.
21.3. FREQUENT NMCLI COMMANDS

The following is an overview about frequently-used nmcli commands.

- To display the list connection profiles, enter:

```
# nmcli connection show
NAME    UUID                                  TYPE      DEVICE
enp1s0  45224a39-606f-4bf7-b3dc-d088236c15ee  ethernet  enp1s0
```

- To display the settings of a specific connection profile, enter:

```
# nmcli connection show connection_name
connection.id:             enp1s0
connection.uuid:           45224a39-606f-4bf7-b3dc-d088236c15ee
connection.stable-id:      --
connection.type:           802-3-ethernet
...
```

- To modify properties of a connection, enter:

```
# nmcli connection modify connection_name property value
```

You can modify multiple properties using a single command if you pass multiple `property value` combinations to the command.

- To display the list of network devices, their state, and which connection profiles use the device, enter:

```
# nmcli device
DEVICE  TYPE      STATE         CONNECTION
enp1s0  ethernet  connected     enp1s0
enp8s0  ethernet  disconnected  --
enp7s0  ethernet  unmanaged     --
...
```

- To activate a connection, enter:

```
# nmcli connection up connection_name
```

- To deactivate a connection, enter:

```
# nmcli connection down connection_name
```
CHAPTER 22. GETTING STARTED WITH CONFIGURING NETWORKING USING THE GNOME GUI

You can manage and configure network connections using the following ways on GNOME:

- the GNOME Shell network connection icon on the top right of the desktop
- the GNOME control-center application
- the GNOME nm-connection-editor application

22.1. CONNECTING TO A NETWORK USING THE GNOME SHELL NETWORK CONNECTION ICON

If you use the GNOME GUI, you can use the GNOME Shell network connection icon to connect to a network.

Prerequisites

- The GNOME package group is installed.
- You are logged in to GNOME.
- If the network requires a specific configuration, such as a static IP address or an 802.1x configuration, a connection profile has already been created.

Procedure

1. Click the network connection icon in the top right corner of your desktop.

2. Depending on the connection type, select the Wired or Wi-Fi entry.
• For a wired connection, select **Connect** to connect to the network.

• For a Wi-Fi connection, click **Select network**, select the network to which you want to connect, and enter the password.
CHAPTER 23. CONFIGURING IP NETWORKING WITH IFCFG FILES

This section describes how to configure a network interface manually by editing the ifcfg files.

IMPORTANT

NetworkManager supports profiles stored in the key file format. However, by default, NetworkManager uses the ifcfg format when you use the NetworkManager API to create or update profiles.

In a future major RHEL release, the key file format will be default. Consider using the key file format if you want to manually create and manage configuration files. For details, see Manually creating NetworkManager profiles in key file format.

Interface configuration (ifcfg) files control the software interfaces for individual network devices. As the system boots, it uses these files to determine what interfaces to bring up and how to configure them. These files are usually named ifcfg-name, where the suffix name refers to the name of the device that the configuration file controls. By convention, the ifcfg file’s suffix is the same as the string given by the DEVICE directive in the configuration file itself.

23.1. CONFIGURING AN INTERFACE WITH STATIC NETWORK SETTINGS USING IFCFG FILES

This procedure describes how to configure a network interface using ifcfg files.

Procedure

- To configure an interface with static network settings using ifcfg files, for an interface with the name enp1s0, create a file with the name ifcfg-enp1s0 in the /etc/sysconfig/network-scripts/ directory that contains:
  
  For IPv4 configuration:
  
  ```
  DEVICE=enp1s0
  BOOTPROTO=none
  ONBOOT=yes
  PREFIX=24
  IPADDR=10.0.1.27
  GATEWAY=10.0.1.1
  ```

  For IPv6 configuration:
  
  ```
  DEVICE=enp1s0
  BOOTPROTO=none
  ONBOOT=yes
  IPV6INIT=yes
  IPV6ADDR=2001:db8:1::2/64
  ```

Additional resources

- Testing basic network settings
23.2. CONFIGURING AN INTERFACE WITH DYNAMIC NETWORK SETTINGS USING IFCFG FILES

This procedure describes how to configure a network interface with dynamic network settings using ifcfg files.

Procedure

1. To configure an interface named em1 with dynamic network settings using ifcfg files, create a file with the name ifcfg-em1 in the /etc/sysconfig/network-scripts/ directory that contains:

   ```
   DEVICE=em1
   BOOTPROTO= dhcp
   ONBOOT=yes
   ```

2. To configure an interface to send:
   - A different host name to the DHCP server, add the following line to the ifcfg file:
     ```
     DHCP_HOSTNAME=hostname
     ```
   - A different fully qualified domain name (FQDN) to the DHCP server, add the following line to the ifcfg file:
     ```
     DHCP_FQDN=fully.qualified.domain.name
     ```

   **NOTE**
   You can use only one of these settings. If you specify both DHCP_HOSTNAME and DHCP_FQDN, only DHCP_FQDN is used.

3. To configure an interface to use particular DNS servers, add the following lines to the ifcfg file:

   ```
   PEERDNS=no
   DNS1=ip-address
   DNS2=ip-address
   ```

   where ip-address is the address of a DNS server. This will cause the network service to update /etc/resolv.conf with the specified DNS servers specified. Only one DNS server address is necessary, the other is optional.

23.3. MANAGING SYSTEM-WIDE AND PRIVATE CONNECTION PROFILES WITH IFCFG FILES

This procedure describes how to configure ifcfg files to manage the system-wide and private connection profiles.

Procedure
The permissions correspond to the **USERS** directive in the *ifcfg* files. If the **USERS** directive is not present, the network profile will be available to all users.

- As an example, modify the *ifcfg* file with the following row, which will make the connection available only to the users listed:

```
USERS="joe bob alice"
```
CHAPTER 24. GETTING STARTED WITH IPVLAN

This document describes the IPVLAN driver.

24.1. IPVLAN OVERVIEW

IPVLAN is a driver for a virtual network device that can be used in container environment to access the host network. IPVLAN exposes a single MAC address to the external network regardless the number of IPVLAN device created inside the host network. This means that a user can have multiple IPVLAN devices in multiple containers and the corresponding switch reads a single MAC address. IPVLAN driver is useful when the local switch imposes constraints on the total number of MAC addresses that it can manage.

24.2. IPVLAN MODES

The following modes are available for IPVLAN:

- **L2 mode**
  In IPVLAN L2 mode, virtual devices receive and respond to address resolution protocol (ARP) requests. The netfilter framework runs only inside the container that owns the virtual device. No netfilter chains are executed in the default namespace on the containerized traffic. Using L2 mode provides good performance, but less control on the network traffic.

- **L3 mode**
  In L3 mode, virtual devices process only L3 traffic and above. Virtual devices do not respond to ARP request and users must configure the neighbour entries for the IPVLAN IP addresses on the relevant peers manually. The egress traffic of a relevant container is landed on the netfilter POSTROUTING and OUTPUT chains in the default namespace while the ingress traffic is threaded in the same way as L2 mode. Using L3 mode provides good control but decreases the network traffic performance.

- **L3S mode**
  In L3S mode, virtual devices process the same way as in L3 mode, except that both egress and ingress traffics of a relevant container are landed on netfilter chain in the default namespace. L3S mode behaves in a similar way to L3 mode but provides greater control of the network.

**NOTE**

The IPVLAN virtual device does not receive broadcast and multicast traffic in case of L3 and L3S modes.

24.3. OVERVIEW OF MACVLAN

The MACVLAN driver allows to create multiple virtual network devices on top of a single NIC, each of them identified by its own unique MAC address. Packets which land on the physical NIC are demultiplexed towards the relevant MACVLAN device via MAC address of the destination. MACVLAN devices do not add any level of encapsulation.

24.4. COMPARISON OF IPVLAN AND MACVLAN

The following table shows the major differences between MACVLAN and IPVLAN.
### MACVLAN vs. IPVLAN

<table>
<thead>
<tr>
<th>MACVLAN</th>
<th>IPVLAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uses MAC address for each MACVLAN device. The overlimit of MAC addresses of MAC table in switch might cause loosing the connectivity.</td>
<td>Uses single MAC address which does not limit the number of IPVLAN devices.</td>
</tr>
<tr>
<td>Netfilter rules for global namespace cannot affect traffic to or from MACVLAN device in a child namespace.</td>
<td>It is possible to control traffic to or from IPVLAN device in <strong>L3 mode</strong> and <strong>L3S mode</strong>.</td>
</tr>
</tbody>
</table>

Note that both IPVLAN and MACVLAN do not require any level of encapsulation.

### 24.5. CREATING AND CONFIGURING THE IPVLAN DEVICE USING iproute2

This procedure shows how to set up the IPVLAN device using *iproute2*.

**Procedure**

1. To create an IPVLAN device, enter the following command:

   ```
   # ip link add link real_NIC_device name IPVLAN_device type ipvlan mode l2
   ```

   Note that network interface controller (NIC) is a hardware component which connects a computer to a network.

   **Example 24.1. Creating an IPVLAN device**

   ```
   # ip link add link enp0s31f6 name my_ipvlan type ipvlan mode l2
   # ip link
   47: my_ipvlan@enp0s31f6: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000 link/ether e8:6a:6e:8a:a2:44 brd ff:ff:ff:ff:ff:ff
   ```

2. To assign an **IPv4** or **IPv6** address to the interface, enter the following command:

   ```
   # ip addr add dev IPVLAN_device IP_address/subnet_mask_prefix
   ```

3. In case of configuring an IPVLAN device in **L3 mode** or **L3S mode**, make the following setups:

   a. Configure the neighbor setup for the remote peer on the remote host:

   ```
   # ip neigh add dev peer_device IPVLAN_device_IP_address lladdr MAC_address
   ```

   where **MAC_address** is the MAC address of the real NIC on which an IPVLAN device is based on.

   b. Configure an IPVLAN device for **L3 mode** with the following command:

   ```
   # ip route add dev <real_NIC_device> <peer_IP_address/32>
   ```
For L3S mode:

```bash
# ip route add dev real_NIC_device peer_IP_address/32
```

where IP-address represents the address of the remote peer.

4. To set an IPVLAN device active, enter the following command:

```bash
# ip link set dev IPVLAN_device up
```

5. To check if the IPVLAN device is active, execute the following command on the remote host:

```bash
# ping IP_address
```

where the IP_address uses the IP address of the IPVLAN device.
CHAPTER 25. REUSING THE SAME IP ADDRESS ON DIFFERENT INTERFACES

With Virtual routing and forwarding (VRF), administrators can use multiple routing tables simultaneously on the same host. For that, VRF partitions a network at layer 3. This enables the administrator to isolate traffic using separate and independent route tables per VRF domain. This technique is similar to virtual LANs (VLAN), which partitions a network at layer 2, where the operating system uses different VLAN tags to isolate traffic sharing the same physical medium.

One benefit of VRF over partitioning on layer 2 is that routing scales better considering the number of peers involved.

Red Hat Enterprise Linux uses a virtual vrt device for each VRF domain and adds routes to a VRF domain by adding existing network devices to a VRF device. Addresses and routes previously attached to the original device will be moved inside the VRF domain.

Note that each VRF domain is isolated from each other.

25.1. PERMANENTLY REUSING THE SAME IP ADDRESS ON DIFFERENT INTERFACES

This procedure describes how to permanently use the same IP address on different interfaces in one server by using the VRF feature.

IMPORTANT

To enable remote peers to contact both VRF interfaces while reusing the same IP address, the network interfaces must belong to different broadcasting domains. A broadcast domain in a network is a set of nodes, which receive broadcast traffic sent by any of them. In most configurations, all nodes connected to the same switch belong to the same broadcasting domain.

Prerequisites

- You are logged in as the root user.
- The network interfaces are not configured.

Procedure

1. Create and configure the first VRF device:
   a. Create a connection for the VRF device and assign it to a routing table. For example, to create a VRF device named vrf0 that is assigned to the 1001 routing table:

      ```
      # nmcli connection add type vrf ifname vrf0 con-name vrf0 table 1001 ipv4.method disabled ipv6.method disabled
      ```

   b. Enable the vrf0 device:

      ```
      # nmcli connection up vrf0
      ```
c. Assign a network device to the VRF just created. For example, to add the enp1s0 Ethernet device to the vrf0 VRF device and assign an IP address and the subnet mask to enp1s0, enter:

```
# nmcli connection add type ethernet con-name vrf.enp1s0 ifname enp1s0 master vrf0 ipv4.method manual ipv4.address 192.0.2.1/24
```

d. Activate the vrf.enp1s0 connection:

```
# nmcli connection up vrf.enp1s0
```

2. Create and configure the next VRF device:

a. Create the VRF device and assign it to a routing table. For example, to create a VRF device named vrf1 that is assigned to the 1002 routing table, enter:

```
# nmcli connection add type vrf ifname vrf1 con-name vrf1 table 1002 ipv4.method disabled ipv6.method disabled
```

b. Activate the vrf1 device:

```
# nmcli connection up vrf1
```

c. Assign a network device to the VRF just created. For example, to add the enp7s0 Ethernet device to the vrf1 VRF device and assign an IP address and the subnet mask to enp7s0, enter:

```
# nmcli connection add type ethernet con-name vrf.enp7s0 ifname enp7s0 master vrf1 ipv4.method manual ipv4.address 192.0.2.1/24
```

d. Activate the vrf.enp7s0 device:

```
# nmcli connection up vrf.enp7s0
```

### 25.2. TEMPORARILY REUSING THE SAME IP ADDRESS ON DIFFERENT INTERFACES

The procedure in this section describes how to temporarily use the same IP address on different interfaces in one server by using the virtual routing and forwarding (VRF) feature. Use this procedure only for testing purposes, because the configuration is temporary and lost after you reboot the system.

**IMPORTANT**

To enable remote peers to contact both VRF interfaces while reusing the same IP address, the network interfaces must belong to different broadcasting domains. A broadcast domain in a network is a set of nodes which receive broadcast traffic sent by any of them. In most configurations, all nodes connected to the same switch belong to the same broadcasting domain.

**Prerequisites**

- You are logged in as the root user.
The network interfaces are not configured.

Procedure

1. Create and configure the first VRF device:
   a. Create the VRF device and assign it to a routing table. For example, to create a VRF device named blue that is assigned to the 1001 routing table:
      
      ```bash
      # ip link add dev blue type vrf table 1001
      ```
   b. Enable the blue device:
      
      ```bash
      # ip link set dev blue up
      ```
   c. Assign a network device to the VRF device. For example, to add the enp1s0 Ethernet device to the blue VRF device:
      
      ```bash
      # ip link set dev enp1s0 master blue
      ```
   d. Enable the enp1s0 device:
      
      ```bash
      # ip link set dev enp1s0 up
      ```
   e. Assign an IP address and subnet mask to the enp1s0 device. For example, to set it to 192.0.2.1/24:
      
      ```bash
      # ip addr add dev enp1s0 192.0.2.1/24
      ```

2. Create and configure the next VRF device:
   a. Create the VRF device and assign it to a routing table. For example, to create a VRF device named red that is assigned to the 1002 routing table:
      
      ```bash
      # ip link add dev red type vrf table 1002
      ```
   b. Enable the red device:
      
      ```bash
      # ip link set dev red up
      ```
   c. Assign a network device to the VRF device. For example, to add the enp7s0 Ethernet device to the red VRF device:
      
      ```bash
      # ip link set dev enp7s0 master red
      ```
   d. Enable the enp7s0 device:
      
      ```bash
      # ip link set dev enp7s0 up
      ```
   e. Assign the same IP address and subnet mask to the enp7s0 device as you used for enp1s0 in the blue VRF domain:
3. Optionally, create further VRF devices as described above.

### 25.3. ADDITIONAL RESOURCES

- `/usr/share/doc/kernel-doc-<kernel_version>/Documentation/networking/vrf.txt` from the `kernel-doc` package
CHAPTER 26. SECURING NETWORKS

26.1. USING SECURE COMMUNICATIONS BETWEEN TWO SYSTEMS WITH OPENSSH

SSH (Secure Shell) is a protocol which provides secure communications between two systems using a client-server architecture and allows users to log in to server host systems remotely. Unlike other remote communication protocols, such as FTP or Telnet, SSH encrypts the login session, which prevents intruders to collect unencrypted passwords from the connection.

Red Hat Enterprise Linux includes the basic OpenSSH packages: the general openssh package, the openssh-server package and the openssh-clients package. Note that the OpenSSH packages require the OpenSSL package openssl-libs, which installs several important cryptographic libraries that enable OpenSSH to provide encrypted communications.

26.1.1. SSH and OpenSSH

SSH (Secure Shell) is a program for logging into a remote machine and executing commands on that machine. The SSH protocol provides secure encrypted communications between two untrusted hosts over an insecure network. You can also forward X11 connections and arbitrary TCP/IP ports over the secure channel.

The SSH protocol mitigates security threats, such as interception of communication between two systems and impersonation of a particular host, when you use it for remote shell login or file copying. This is because the SSH client and server use digital signatures to verify their identities. Additionally, all communication between the client and server systems is encrypted.

A host key authenticates hosts in the SSH protocol. Host keys are cryptographic keys that are generated automatically when OpenSSH is first installed, or when the host boots for the first time.

OpenSSH is an implementation of the SSH protocol supported by a number of Linux, UNIX, and similar operating systems. It includes the core files necessary for both the OpenSSH client and server. The OpenSSH suite consists of the following user-space tools:

- `ssh` is a remote login program (SSH client)
- `sshd` is an OpenSSH SSH daemon
- `scp` is a secure remote file copy program
- `sftp` is a secure file transfer program
- `ssh-agent` is an authentication agent for caching private keys
- `ssh-add` adds private key identities to `ssh-agent`
- `ssh-keygen` generates, manages, and converts authentication keys for `ssh`
- `ssh-copy-id` is a script that adds local public keys to the `authorized_keys` file on a remote SSH server
- `ssh-keyscan` - gathers SSH public host keys
Two versions of SSH currently exist: version 1, and the newer version 2. The OpenSSH suite in RHEL supports only SSH version 2, which has an enhanced key-exchange algorithm not vulnerable to known exploits in version 1.

OpenSSH, as one of the RHEL core cryptographic subsystems uses system-wide crypto policies. This ensures that weak cipher suites and cryptographic algorithms are disabled in the default configuration. To adjust the policy, the administrator must either use the update-crypto-policies command to make settings stricter or looser or manually opt-out of the system-wide crypto policies.

The OpenSSH suite uses two different sets of configuration files: those for client programs (that is, ssh, scp, and sftp), and those for the server (the sshd daemon). System-wide SSH configuration information is stored in the /etc/ssh/ directory. User-specific SSH configuration information is stored in ~/.ssh/ in the user’s home directory. For a detailed list of OpenSSH configuration files, see the FILES section in the sshd(8) man page.

Additional resources

- Man pages listed by using the man -k ssh command
- Using system-wide cryptographic policies

26.1.2. Configuring and starting an OpenSSH server

Use the following procedure for a basic configuration that might be required for your environment and for starting an OpenSSH server. Note that after the default RHEL installation, the sshd daemon is already started and server host keys are automatically created.

Prerequisites

- The openssh-server package is installed.

Procedure

1. Start the sshd daemon in the current session and set it to start automatically at boot time:

   ```bash
   # systemctl start sshd
   # systemctl enable sshd
   ```

2. To specify different addresses than the default 0.0.0.0 (IPv4) or :: (IPv6) for the ListenAddress directive in the /etc/ssh/sshd_config configuration file and to use a slower dynamic network configuration, add the dependency on the network-online.target target unit to the sshd.service unit file. To achieve this, create the /etc/systemd/system/sshd.service.d/local.conf file with the following content:

   ```ini
   [Unit]
   Wants=network-online.target
   After=network-online.target
   ```

3. Review if OpenSSH server settings in the /etc/ssh/sshd_config configuration file meet the requirements of your scenario.

4. Optionally, change the welcome message that your OpenSSH server displays before a client authenticates by editing the /etc/issue file, for example:
Welcome to ssh-server.example.com
Warning: By accessing this server, you agree to the referenced terms and conditions.

Ensure that the **Banner** option is not commented out in `/etc/ssh/sshd_config` and its value contains `/etc/issue`:

```
# less /etc/ssh/sshd_config | grep Banner
Banner /etc/issue
```

Note that to change the message displayed after a successful login you have to edit the `/etc/motd` file on the server. See the `pam_motd` man page for more information.

5. Reload the **systemd** configuration and restart **sshd** to apply the changes:

```
# systemctl daemon-reload
# systemctl restart sshd
```

**Verification**

1. Check that the **sshd** daemon is running:

```
# systemctl status sshd
● sshd.service - OpenSSH server daemon
    Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
    Active: active (running) since Mon 2019-11-18 14:59:58 CET; 6min ago
      Docs: man:sshd(8)
      man:sshd_config(5)
    Main PID: 1149 (sshd)
    Tasks: 1 (limit: 11491)
    Memory: 1.9M
    CGroup: /system.slice/sshd.service
        ⬇
          1149 /usr/sbin/sshd -D -oCiphers=aes128-ctr,aes256-ctr,aes128-cbc,aes256-cbc -oMACs=hmac-sha2-256,>
    Nov 18 14:59:58 ssh-server-example.com systemd[1]: Starting OpenSSH server daemon...
    Nov 18 14:59:58 ssh-server-example.com sshd[1149]: Server listening on 0.0.0.0 port 22.
    Nov 18 14:59:58 ssh-server-example.com sshd[1149]: Server listening on :: port 22.
    Nov 18 14:59:58 ssh-server-example.com systemd[1]: Started OpenSSH server daemon.
```

2. Connect to the SSH server with an SSH client.

```
# ssh user@ssh-server-example.com
ECDSA key fingerprint is SHA256:dXbaS0RG/UzITTk8GtXSz0S1++IPEGSy31v3L/FAEc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ssh-server-example.com' (ECDSA) to the list of known hosts.
user@ssh-server-example.com's password:
```

**Additional resources**

- `sshd(8)` and `sshd_config(5)` man pages.

### 26.1.3. Setting an OpenSSH server for key-based authentication
To improve system security, enforce key-based authentication by disabling password authentication on your OpenSSH server.

**Prerequisites**

- The `openssh-server` package is installed.
- The `sshd` daemon is running on the server.

**Procedure**

1. Open the `/etc/ssh/sshd_config` configuration in a text editor, for example:
   ```
   # vi /etc/ssh/sshd_config
   ```
2. Change the `PasswordAuthentication` option to **no**:
   ```
   PasswordAuthentication no
   ```
   On a system other than a new default installation, check that `PubkeyAuthentication no` has not been set and the `ChallengeResponseAuthentication` directive is set to **no**. If you are connected remotely, not using console or out-of-band access, test the key-based login process before disabling password authentication.
3. To use key-based authentication with NFS-mounted home directories, enable the `use_nfs_home_dirs` SELinux boolean:
   ```
   # setsebool -P use_nfs_home_dirs 1
   ```
4. Reload the `sshd` daemon to apply the changes:
   ```
   # systemctl reload sshd
   ```

**Additional resources**

- `sshd(8)`, `sshd_config(5)`, and `setsebool(8)` man pages.

**26.1.4. Generating SSH key pairs**

Use this procedure to generate an SSH key pair on a local system and to copy the generated public key to an OpenSSH server. If the server is configured accordingly, you can log in to the OpenSSH server without providing any password.

**IMPORTANT**

If you complete the following steps as **root**, only **root** is able to use the keys.

**Procedure**

1. To generate an ECDSA key pair for version 2 of the SSH protocol:
   ```
   $ ssh-keygen -t ecdsa
   Generating public/private ecdsa key pair.
   ```
Enter file in which to save the key (/home/joesec/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/joesec/.ssh/id_ecdsa.
Your public key has been saved in /home/joesec/.ssh/id_ecdsa.pub.
The key fingerprint is:
SHA256:Q/x+qms4j7PCQ0qFd09iZEFHA+SqwBKRNaU72oZfaCl
joesec@localhost.example.com
The key's randomart image is:
+----[ECDSA 256]----+
  |.oo..o=++        |
  |.. .00 .       |
  |.. .0 .        |
  |....O.+...     |
  |o.0o.o +S .    |
  |.=+. .0       |
  |E.*+. . . .  |
  |.=.+ +. . 0    |
  | . . oo*+o.   |
+----[SHA256]-----+

You can also generate an RSA key pair by using the -t rsa option with the ssh-keygen command or an Ed25519 key pair by entering the ssh-keygen -t ed25519 command.

2. To copy the public key to a remote machine:

$ ssh-copy-id joesec@ssh-server-example.com
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
joesec@ssh-server-example.com's password:
...
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'joesec@ssh-server-example.com'" and check to make sure that only the key(s) you wanted were added.

If you do not use the ssh-agent program in your session, the previous command copies the most recently modified ~/.ssh/id*.pub public key if it is not yet installed. To specify another public-key file or to prioritize keys in files over keys cached in memory by ssh-agent, use the ssh-copy-id command with the -i option.

NOTE
If you reinstall your system and want to keep previously generated key pairs, back up the ~/.ssh/ directory. After reinstalling, copy it back to your home directory. You can do this for all users on your system, including root.

Verification

1. Log in to the OpenSSH server without providing any password:

$ ssh joesec@ssh-server-example.com
Welcome message.
...
Last login: Mon Nov 18 18:28:42 2019 from ::1
Additional resources

- `ssh-keygen(1)` and `ssh-copy-id(1)` man pages.

26.1.5. Using SSH keys stored on a smart card

Red Hat Enterprise Linux enables you to use RSA and ECDSA keys stored on a smart card on OpenSSH clients. Use this procedure to enable authentication using a smart card instead of using a password.

Prerequisites

- On the client side, the `opensc` package is installed and the `pcscd` service is running.

Procedure

1. List all keys provided by the OpenSC PKCS #11 module including their PKCS #11 URIs and save the output to the `keys.pub` file:

   ```bash
   $ ssh-keygen -D pkcs11: > keys.pub
   $ ssh-keygen -D pkcs11:
   ssh-rsa AAAAB3NzaC1yc2E...KKZMzcQZzx
   pkcs11:id=%02;object=SIGN%20pubkey;token=SSH%20key;manufacturer=piv_II?module-path=/usr/lib64/pkcs11/opensc-pkcs11.so
   ecdsa-sha2-nistp256 AAA...J0hkYnnsM=
   pkcs11:id=%01;object=PIV%20AUTH%20pubkey;token=SSH%20key;manufacturer=piv_II?
   module-path=/usr/lib64/pkcs11/opensc-pkcs11.so
   ```

2. To enable authentication using a smart card on a remote server (example.com), transfer the public key to the remote server. Use the `ssh-copy-id` command with `keys.pub` created in the previous step:

   ```bash
   $ ssh-copy-id -f -i keys.pub username@example.com
   ```

3. To connect to example.com using the ECDSA key from the output of the `ssh-keygen -D` command in step 1, you can use just a subset of the URI, which uniquely references your key, for example:

   ```bash
   $ ssh -i "pkcs11:id=%01?module-path=/usr/lib64/pkcs11/opensc-pkcs11.so" example.com
   Enter PIN for 'SSH key':
   [example.com] $
   ```

4. You can use the same URI string in the `~/.ssh/config` file to make the configuration permanent:

   ```bash
   $ cat ~/.ssh/config
   IdentityFile "pkcs11:id=%01?module-path=/usr/lib64/pkcs11/opensc-pkcs11.so"
   $ ssh example.com
   Enter PIN for 'SSH key':
   [example.com] $
   ```

Because OpenSSH uses the `p11-kit-proxy` wrapper and the OpenSC PKCS #11 module is registered to PKCS#11 Kit, you can simplify the previous commands:
$ ssh -i "pkcs11:id=%01" example.com
Enter PIN for ‘SSH key’:
[example.com] $

If you skip the id= part of a PKCS #11 URI, OpenSSH loads all keys that are available in the proxy module. This can reduce the amount of typing required:

$ ssh -i pkcs11: example.com
Enter PIN for ‘SSH key’:
[example.com] $

Additional resources

- Fedora 28: Better smart card support in OpenSSH
- p11-kit(8), opensc.conf(5), pcscd(8), ssh(1), and ssh-keygen(1) man pages

26.1.6. Making OpenSSH more secure

The following tips help you to increase security when using OpenSSH. Note that changes in the /etc/ssh/sshd_config OpenSSH configuration file require reloading the sshd daemon to take effect:

# systemctl reload sshd

IMPORTANT

The majority of security hardening configuration changes reduce compatibility with clients that do not support up-to-date algorithms or cipher suites.

Disabling insecure connection protocols

- To make SSH truly effective, prevent the use of insecure connection protocols that are replaced by the OpenSSH suite. Otherwise, a user’s password might be protected using SSH for one session only to be captured later when logging in using Telnet. For this reason, consider disabling insecure protocols, such as telnet, rsh, rlogin, and ftp.

Enabling key-based authentication and disabling password-based authentication

- Disabling passwords for authentication and allowing only key pairs reduces the attack surface and it also might save users’ time. On clients, generate key pairs using the ssh-keygen tool and use the ssh-copy-id utility to copy public keys from clients on the OpenSSH server. To disable password-based authentication on your OpenSSH server, edit /etc/ssh/sshd_config and change the PasswordAuthentication option to no:

PasswordAuthentication no

Key types

- Although the ssh-keygen command generates a pair of RSA keys by default, you can instruct it to generate ECDSA or Ed25519 keys by using the -t option. The ECDSA (Elliptic Curve Digital Signature Algorithm) offers better performance than RSA at the equivalent symmetric key
strength. It also generates shorter keys. The Ed25519 public-key algorithm is an implementation of twisted Edwards curves that is more secure and also faster than RSA, DSA, and ECDSA. OpenSSH creates RSA, ECDSA, and Ed25519 server host keys automatically if they are missing. To configure the host key creation in RHEL, use the `sshd-keygen@.service` instantiated service. For example, to disable the automatic creation of the RSA key type:

```bash
# systemctl mask sshd-keygen@rsa.service
```

- To exclude particular key types for SSH connections, comment out the relevant lines in `/etc/ssh/sshd_config`, and reload the `sshd` service. For example, to allow only Ed25519 host keys:

```bash
# HostKey /etc/ssh/ssh_host_rsa_key
# HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
```

### Non-default port

- By default, the `sshd` daemon listens on TCP port 22. Changing the port reduces the exposure of the system to attacks based on automated network scanning and thus increase security through obscurity. You can specify the port using the `Port` directive in the `/etc/ssh/sshd_config` configuration file.

You also have to update the default SELinux policy to allow the use of a non-default port. To do so, use the `semanage` tool from the `policycoreutils-python-utils` package:

```bash
# semanage port -a -t ssh_port_t -p tcp port_number
```

Furthermore, update `firewalld` configuration:

```bash
# firewall-cmd --add-port port_number/tcp
# firewall-cmd --runtime-to-permanent
```

In the previous commands, replace `port_number` with the new port number specified using the `Port` directive.

### No root login

- If your particular use case does not require the possibility of logging in as the root user, you should consider setting the `PermitRootLogin` configuration directive to `no` in the `/etc/ssh/sshd_config` file. By disabling the possibility of logging in as the root user, the administrator can audit which users run what privileged commands after they log in as regular users and then gain root rights.

Alternatively, set `PermitRootLogin` to `prohibit-password`:

```bash
PermitRootLogin prohibit-password
```

This enforces the use of key-based authentication instead of the use of passwords for logging in as root and reduces risks by preventing brute-force attacks.

### Using the X Security extension

- The X server in Red Hat Enterprise Linux clients does not provide the X Security extension. Therefore, clients cannot request another security layer when connecting to untrusted SSH
servers with X11 forwarding. Most applications are not able to run with this extension enabled anyway.

By default, the `ForwardX11Trusted` option in the `/etc/ssh/ssh_config.d/05-redhat.conf` file is set to `yes`, and there is no difference between the `ssh -X remote_machine` (untrusted host) and `ssh -Y remote_machine` (trusted host) command.

If your scenario does not require the X11 forwarding feature at all, set the `X11Forwarding` directive in the `/etc/ssh/sshd_config` configuration file to `no`.

Restricting access to specific users, groups, or domains

- The `AllowUsers` and `AllowGroups` directives in the `/etc/ssh/sshd_config` configuration file server enable you to permit only certain users, domains, or groups to connect to your OpenSSH server. You can combine `AllowUsers` and `AllowGroups` to restrict access more precisely, for example:

  ```
  AllowUsers *@192.168.1.*,*@10.0.0.*,!*@192.168.1.2
  AllowGroups example-group
  ```

  The previous configuration lines accept connections from all users from systems in 192.168.1.* and 10.0.0.* subnets except from the system with the 192.168.1.2 address. All users must be in the `example-group` group. The OpenSSH server denies all other connections.

  Note that using allowlists (directives starting with Allow) is more secure than using blocklists (options starting with Deny) because allowlists block also new unauthorized users or groups.

Changing system-wide cryptographic policies

- OpenSSH uses RHEL system-wide cryptographic policies, and the default system-wide cryptographic policy level offers secure settings for current threat models. To make your cryptographic settings more strict, change the current policy level:

  ```
  # update-crypto-policies --set FUTURE
  Setting system policy to FUTURE
  ```

- To opt-out of the system-wide crypto policies for your OpenSSH server, uncomment the line with the `CRYPTO_POLICY=` variable in the `/etc/sysconfig/sshd` file. After this change, values that you specify in the `Ciphers`, `MACs`, `KexAlgorithms`, and `GSSAPIKexAlgorithms` sections in the `/etc/ssh/sshd_config` file are not overridden. Note that this task requires deep expertise in configuring cryptographic options.

- See Using system-wide cryptographic policies in the Security hardening title for more information.

Additional resources

- `sshd_config(5)`, `ssh-keygen(1)`, `crypto-policies(7)`, and `update-crypto-policies(8)` man pages.

26.1.7. Connecting to a remote server using an SSH jump host

Use this procedure for connecting your local system to a remote server through an intermediary server, also called jump host.

Prerequisites
• A jump host accepts SSH connections from your local system.
• A remote server accepts SSH connections only from the jump host.

Procedure

1. Define the jump host by editing the ~/.ssh/config file on your local system, for example:

   ```
   Host jump-server1
   HostName jump1.example.com
   ```

   • The Host parameter defines a name or alias for the host you can use in ssh commands. The value can match the real host name, but can also be any string.

   • The HostName parameter sets the actual host name or IP address of the jump host.

2. Add the remote server jump configuration with the ProxyJump directive to ~/.ssh/config file on your local system, for example:

   ```
   Host remote-server
   HostName remote1.example.com
   ProxyJump jump-server1
   ```

3. Use your local system to connect to the remote server through the jump server:

   ```
   $ ssh remote-server
   ```

   The previous command is equivalent to the ssh -J jump-server1 remote-server command if you omit the configuration steps 1 and 2.

   **NOTE**

   You can specify more jump servers and you can also skip adding host definitions to the configurations file when you provide their complete host names, for example:

   ```
   $ ssh -J jump1.example.com,jump2.example.com,jump3.example.com
   remote1.example.com
   ```

   Change the host name-only notation in the previous command if the user names or SSH ports on the jump servers differ from the names and ports on the remote server, for example:

   ```
   $ ssh -J
   johndoe@jump1.example.com:75,johndoe@jump2.example.com:75,johndoe@jump3.example.com:75
   joesec@remote1.example.com:220
   ```

   Additional resources

   • ssh_config(5) and ssh(1) man pages.

26.1.8. Connecting to remote machines with SSH keys using ssh-agent
To avoid entering a passphrase each time you initiate an SSH connection, you can use the `ssh-agent` utility to cache the private SSH key. The private key and the passphrase remain secure.

**Prerequisites**
- You have a remote host with SSH daemon running and reachable through the network.
- You know the IP address or hostname and credentials to log in to the remote host.
- You have generated an SSH key pair with a passphrase and transferred the public key to the remote machine.

**Procedure**

1. Optional: Verify you can use the key to authenticate to the remote host:
   
   a. Connect to the remote host using SSH:
      
      ```
      $ ssh example.user1@198.51.100.1 hostname
      ```
   
   b. Enter the passphrase you set while creating the key to grant access to the private key.
      
      ```
      $ ssh example.user1@198.51.100.1 hostname host.example.com
      ```

2. Start the `ssh-agent`.
   
   ```
   $ eval $(ssh-agent)
   Agent pid 20062
   ```

3. Add the key to `ssh-agent`.
   
   ```
   $ ssh-add ~/.ssh/id_rsa
   Enter passphrase for ~/.ssh/id_rsa:
   Identity added: ~/.ssh/id_rsa (example.user0@198.51.100.12)
   ```

**Verification**

- Optional: Log in to the host machine using SSH.
  
  ```
  $ ssh example.user1@198.51.100.1
  Last login: Mon Sep 14 12:56:37 2020
  ```

  Note that you did not have to enter the passphrase.

**26.1.9. Additional resources**

- `sshd(8)`, `ssh(1)`, `scp(1)`, `sftp(1)`, `ssh-keygen(1)`, `ssh-copy-id(1)`, `ssh_config(5)`, `sshd_config(5)`, `update-crypto-policies(8)`, and `crypto-policies(7)` man pages.
- OpenSSH Home Page
- Configuring SELinux for applications and services with non-standard configurations
### 26.2. PLANNING AND IMPLEMENTING TLS

TLS (Transport Layer Security) is a cryptographic protocol used to secure network communications. When hardening system security settings by configuring preferred key-exchange protocols, authentication methods, and encryption algorithms, it is necessary to bear in mind that the broader the range of supported clients, the lower the resulting security. Conversely, strict security settings lead to limited compatibility with clients, which can result in some users being locked out of the system. Be sure to target the strictest available configuration and only relax it when it is required for compatibility reasons.

#### 26.2.1. SSL and TLS protocols

The Secure Sockets Layer (SSL) protocol was originally developed by Netscape Corporation to provide a mechanism for secure communication over the Internet. Subsequently, the protocol was adopted by the Internet Engineering Task Force (IETF) and renamed to Transport Layer Security (TLS).

The TLS protocol sits between an application protocol layer and a reliable transport layer, such as TCP/IP. It is independent of the application protocol and can thus be layered underneath many different protocols, for example: HTTP, FTP, SMTP, and so on.

<table>
<thead>
<tr>
<th>Protocol version</th>
<th>Usage recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL v2</td>
<td>Do not use. Has serious security vulnerabilities. Removed from the core crypto libraries since RHEL 7.</td>
</tr>
<tr>
<td>SSL v3</td>
<td>Do not use. Has serious security vulnerabilities. Removed from the core crypto libraries since RHEL 8.</td>
</tr>
<tr>
<td>TLS 1.0</td>
<td>Not recommended to use. Has known issues that cannot be mitigated in a way that guarantees interoperability, and does not support modern cipher suites. Enabled only in the LEGACY system-wide cryptographic policy profile.</td>
</tr>
<tr>
<td>TLS 1.1</td>
<td>Use for interoperability purposes where needed. Does not support modern cipher suites. Enabled only in the LEGACY policy.</td>
</tr>
<tr>
<td>TLS 1.2</td>
<td>Supports the modern AEAD cipher suites. This version is enabled in all system-wide crypto policies, but optional parts of this protocol contain vulnerabilities and TLS 1.2 also allows outdated algorithms.</td>
</tr>
<tr>
<td>TLS 1.3</td>
<td>Recommended version. TLS 1.3 removes known problematic options, provides additional privacy by encrypting more of the negotiation handshake and can be faster thanks usage of more efficient modern cryptographic algorithms. TLS 1.3 is also enabled in all system-wide crypto policies.</td>
</tr>
</tbody>
</table>

Additional resources


#### 26.2.2. Security considerations for TLS in RHEL 8
In RHEL 8, cryptography-related considerations are significantly simplified thanks to the system-wide crypto policies. The **DEFAULT** crypto policy allows only TLS 1.2 and 1.3. To allow your system to negotiate connections using the earlier versions of TLS, you need to either opt out from following crypto policies in an application or switch to the **LEGACY** policy with the `update-crypto-policies` command. See [Using system-wide cryptographic policies](#) for more information.

The default settings provided by libraries included in RHEL 8 are secure enough for most deployments. The TLS implementations use secure algorithms where possible while not preventing connections from or to legacy clients or servers. Apply hardened settings in environments with strict security requirements where legacy clients or servers that do not support secure algorithms or protocols are not expected or allowed to connect.

The most straightforward way to harden your TLS configuration is switching the system-wide cryptographic policy level to **FUTURE** using the `update-crypto-policies --set FUTURE` command.

---

**WARNING**

Algorithms disabled for the **LEGACY** cryptographic policy do not conform to Red Hat’s vision of RHEL 8 security, and their security properties are not reliable. Consider moving away from using these algorithms instead of re-enabling them. If you do decide to re-enable them, for example for interoperability with old hardware, treat them as insecure and apply extra protection measures, such as isolating their network interactions to separate network segments. Do not use them across public networks.

---

If you decide to not follow RHEL system-wide crypto policies or create custom cryptographic policies tailored to your setup, use the following recommendations for preferred protocols, cipher suites, and key lengths on your custom configuration:

### 26.2.2.1. Protocols

The latest version of TLS provides the best security mechanism. Unless you have a compelling reason to include support for older versions of TLS, allow your systems to negotiate connections using at least TLS version 1.2.

Note that even though RHEL 8 supports TLS version 1.3, not all features of this protocol are fully supported by RHEL 8 components. For example, the 0-RTT (Zero Round Trip Time) feature, which reduces connection latency, is not yet fully supported by the Apache web server.

### 26.2.2.2. Cipher suites

Modern, more secure cipher suites should be preferred to old, insecure ones. Always disable the use of eNULL and aNULL cipher suites, which do not offer any encryption or authentication at all. If at all possible, ciphers suites based on RC4 or HMAC-MD5, which have serious shortcomings, should also be disabled. The same applies to the so-called export cipher suites, which have been intentionally made weaker, and thus are easy to break.

While not immediately insecure, cipher suites that offer less than 128 bits of security should not be considered for their short useful life. Algorithms that use 128 bits of security or more can be expected to be unbreakable for at least several years, and are thus strongly recommended. Note that while 3DES ciphers advertise the use of 168 bits, they actually offer 112 bits of security.
Always prefer cipher suites that support (perfect) forward secrecy (PFS), which ensures the confidentiality of encrypted data even in case the server key is compromised. This rules out the fast RSA key exchange, but allows for the use of ECDHE and DHE. Of the two, ECDHE is the faster and therefore the preferred choice.

You should also prefer AEAD ciphers, such as AES-GCM, over CBC-mode ciphers as they are not vulnerable to padding oracle attacks. Additionally, in many cases, AES-GCM is faster than AES in CBC mode, especially when the hardware has cryptographic accelerators for AES.

Note also that when using the ECDHE key exchange with ECDSA certificates, the transaction is even faster than a pure RSA key exchange. To provide support for legacy clients, you can install two pairs of certificates and keys on a server: one with ECDSA keys (for new clients) and one with RSA keys (for legacy ones).

26.2.2.3. Public key length

When using RSA keys, always prefer key lengths of at least 3072 bits signed by at least SHA-256, which is sufficiently large for true 128 bits of security.

**WARNING**

The security of your system is only as strong as the weakest link in the chain. For example, a strong cipher alone does not guarantee good security. The keys and the certificates are just as important, as well as the hash functions and keys used by the Certification Authority (CA) to sign your keys.

Additional resources

- System-wide crypto policies in RHEL 8.
- update-crypto-policies(8) man page.

26.2.3. Hardening TLS configuration in applications

In RHEL, system-wide crypto policies provide a convenient way to ensure that your applications using cryptographic libraries do not allow known insecure protocols, ciphers, or algorithms.

If you want to harden your TLS-related configuration with your customized cryptographic settings, you can use the cryptographic configuration options described in this section, and override the system-wide crypto policies just in the minimum required amount.

Regardless of the configuration you choose to use, always make sure to mandate that your server application enforces server-side cipher order, so that the cipher suite to be used is determined by the order you configure.

26.2.3.1. Configuring the Apache HTTP server

The Apache HTTP Server can use both OpenSSL and NSS libraries for its TLS needs. Red Hat Enterprise Linux 8 provides the mod_ssl functionality through eponymous packages:
yum install mod_ssl

The `mod_ssl` package installs the `/etc/httpd/conf.d/ssl.conf` configuration file, which can be used to modify the TLS-related settings of the Apache HTTP Server.

Install the `httpd-manual` package to obtain complete documentation for the Apache HTTP Server, including TLS configuration. The directives available in the `/etc/httpd/conf.d/ssl.conf` configuration file are described in detail in `/usr/share/httpd/manual/mod/mod_ssl.html`. Examples of various settings are in `/usr/share/httpd/manual/ssl/ssl_howto.html`.

When modifying the settings in the `/etc/httpd/conf.d/ssl.conf` configuration file, be sure to consider the following three directives at the minimum:

**SSLProtocol**
- Use this directive to specify the version of TLS or SSL you want to allow.

**SSLCipherSuite**
- Use this directive to specify your preferred cipher suite or disable the ones you want to disallow.

**SSLHonorCipherOrder**
- Uncomment and set this directive to `on` to ensure that the connecting clients adhere to the order of ciphers you specified.

For example, to use only the TLS 1.2 and 1.3 protocol:

```bash
SSLProtocol all -SSLv3 -TLSv1 -TLSv1.1
```

See the Configuring TLS encryption on an Apache HTTP Server chapter in the Deploying different types of servers document for more information.

### 26.2.3.2. Configuring the Nginx HTTP and proxy server

To enable TLS 1.3 support in Nginx, add the `TLSv1.3` value to the `ssl_protocols` option in the `server` section of the `/etc/nginx/nginx.conf` configuration file:

```bash
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    ....
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers ....
}
```

See the Adding TLS encryption to an Nginx web server chapter in the Deploying different types of servers document for more information.

### 26.2.3.3. Configuring the Dovecot mail server

To configure your installation of the Dovecot mail server to use TLS, modify the `/etc/dovecot/conf.d/10-ssl.conf` configuration file. You can find an explanation of some of the basic configuration directives available in that file in the `/usr/share/doc/dovecot/wiki/SSL.DovecotConfiguration.txt` file, which is installed along with the standard installation of Dovecot.
When modifying the settings in the `/etc/dovecot/conf.d/10-ssl.conf` configuration file, be sure to consider the following three directives at the minimum:

**ssl_protocols**

Use this directive to specify the version of TLS or SSL you want to allow or disable.

**ssl_cipher_list**

Use this directive to specify your preferred cipher suites or disable the ones you want to disallow.

**ssl_prefer_server_ciphers**

Uncomment and set this directive to `yes` to ensure that the connecting clients adhere to the order of ciphers you specified.

For example, the following line in `/etc/dovecot/conf.d/10-ssl.conf` allows only TLS 1.1 and later:

```
ssl_protocols = !SSLv2 !SSLv3 !TLSv1
```

Additional resources

- Deploying different types of servers on RHEL 8
- `config(5)` and `ciphers(1)` man pages.
- Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS).
- Mozilla SSL Configuration Generator.
- SSL Server Test.

### 26.3. CONFIGURING A VPN WITH IPSEC

In RHEL 8, a virtual private network (VPN) can be configured using the IPsec protocol, which is supported by the Libreswan application.

#### 26.3.1. Libreswan as an IPsec VPN implementation

In RHEL, a Virtual Private Network (VPN) can be configured using the IPsec protocol, which is supported by the Libreswan application. Libreswan is a continuation of the Openswan application, and many examples from the Openswan documentation are interchangeable with Libreswan.

The IPsec protocol for a VPN is configured using the Internet Key Exchange (IKE) protocol. The terms IPsec and IKE are used interchangeably. An IPsec VPN is also called an IKE VPN, IKEv2 VPN, XAUTH VPN, Cisco VPN or IKE/IPsec VPN. A variant of an IPsec VPN that also uses the Level 2 Tunneling Protocol (L2TP) is usually called an L2TP/IPsec VPN, which requires the Optional channel `xl2tpd` application.

Libreswan is an open-source, user-space IKE implementation. IKE v1 and v2 are implemented as a user-level daemon. The IKE protocol is also encrypted. The IPsec protocol is implemented by the Linux kernel, and Libreswan configures the kernel to add and remove VPN tunnel configurations.

The IKE protocol uses UDP port 500 and 4500. The IPsec protocol consists of two protocols:

- Encapsulated Security Payload (ESP), which has protocol number 50.
Authenticated Header (AH), which has protocol number 51.

The AH protocol is not recommended for use. Users of AH are recommended to migrate to ESP with null encryption.

The IPsec protocol provides two modes of operation:

- Tunnel Mode (the default)
- Transport Mode.

You can configure the kernel with IPsec without IKE. This is called Manual Keying. You can also configure manual keying using the ip xfrm commands, however, this is strongly discouraged for security reasons. Libreswan interfaces with the Linux kernel using netlink. Packet encryption and decryption happen in the Linux kernel.

Libreswan uses the Network Security Services (NSS) cryptographic library. Both Libreswan and NSS are certified for use with the Federal Information Processing Standard (FIPS) Publication 140-2.

**IMPORTANT**

IKE/IPsec VPNs, implemented by Libreswan and the Linux kernel, is the only VPN technology recommended for use in RHEL. Do not use any other VPN technology without understanding the risks of doing so.

In RHEL, Libreswan follows system-wide cryptographic policies by default. This ensures that Libreswan uses secure settings for current threat models including IKEv2 as a default protocol. See Using system-wide crypto policies for more information.

Libreswan does not use the terms "source" and "destination" or "server" and "client" because IKE/IPsec are peer to peer protocols. Instead, it uses the terms "left" and "right" to refer to end points (the hosts). This also allows you to use the same configuration on both end points in most cases. However, administrators usually choose to always use "left" for the local host and "right" for the remote host.

The leftid and rightid options serve as identification of the respective hosts in the authentication process. See the ipsec.conf(5) man page for more information.

### 26.3.2. Authentication methods in Libreswan

Libreswan supports several authentication methods, each of which fits a different scenario.

**Pre-Shared key (PSK)**

Pre-Shared Key (PSK) is the simplest authentication method. For security reasons, do not use PSKs shorter than 64 random characters. In FIPS mode, PSKs must comply with a minimum-strength requirement depending on the integrity algorithm used. You can set PSK by using the authby=secret connection.

**Raw RSA keys**

Raw RSA keys are commonly used for static host-to-host or subnet-to-subnet IPsec configurations. Each host is manually configured with the public RSA keys of all other hosts, and Libreswan sets up an IPsec tunnel between each pair of hosts. This method does not scale well for large numbers of hosts.

You can generate a raw RSA key on a host using the ipsec newhostkey command. You can list generated keys by using the ipsec showhostkey command. The leftrsasigkey= line is required for
connection configurations that use CKA ID keys. Use the \texttt{authby=rsasig} connection option for raw RSA keys.

**X.509 certificates**

X.509 certificates are commonly used for large-scale deployments with hosts that connect to a common IPsec gateway. A central certificate authority (CA) signs RSA certificates for hosts or users. This central CA is responsible for relaying trust, including the revocations of individual hosts or users.

For example, you can generate X.509 certificates using the \texttt{openssl} command and the NSS \texttt{certutil} command. Because Libreswan reads user certificates from the NSS database using the certificates' nickname in the \texttt{leftcert=} configuration option, provide a nickname when you create a certificate.

If you use a custom CA certificate, you must import it to the Network Security Services (NSS) database. You can import any certificate in the PKCS #12 format to the Libreswan NSS database by using the \texttt{ipsec import} command.

\begin{warning}
Libreswan requires an Internet Key Exchange (IKE) peer ID as a subject alternative name (SAN) for every peer certificate as described in section 3.1 of RFC 4945. Disabling this check by changing the \texttt{require-id-on-certificated=} option can make the system vulnerable to man-in-the-middle attacks.
\end{warning}

Use the \texttt{authby=rsasig} connection option for authentication based on X.509 certificates using RSA with SHA-1 and SHA-2. You can further limit it for ECDSA digital signatures using SHA-2 by setting \texttt{authby=} to \texttt{ecdsa} and RSA Probabilistic Signature Scheme (RSASSA-PSS) digital signatures based authentication with SHA-2 through \texttt{authby=} \texttt{rsa-sha2}. The default value is \texttt{authby=} \texttt{rsasig,ecdsa}.

The certificates and the \texttt{authby=} signature methods should match. This increases interoperability and preserves authentication in one digital-signature system.

**NULL authentication**

NULL authentication is used to gain mesh encryption without authentication. It protects against passive attacks but not against active attacks. However, because IKEv2 allows asymmetric authentication methods, NULL authentication can also be used for internet-scale opportunistic IPsec. In this model, clients authenticate the server, but servers do not authenticate the client. This model is similar to secure websites using TLS. Use \texttt{authby=} \texttt{null} for NULL authentication.

**Protection against quantum computers**

In addition to the previously mentioned authentication methods, you can use the \textit{Post-quantum Pre-shared Key} (PPK) method to protect against possible attacks by quantum computers. Individual clients or groups of clients can use their own PPK by specifying a PPK ID that corresponds to an out-of-band configured pre-shared key.

Using IKEv1 with pre-shared keys provides protection against quantum attackers. The redesign of IKEv2 does not offer this protection natively. Libreswan offers the use of \textit{Post-quantum Pre-shared Key} (PPK) to protect IKEv2 connections against quantum attacks.

To enable optional PPK support, add \texttt{ppk=} \texttt{yes} to the connection definition. To require PPK, add...
ppk=insist. Then, each client can be given a PPK ID with a secret value that is communicated out-of-band (and preferably quantum safe). The PPK’s should be very strong in randomness and not based on dictionary words. The PPK ID and PPK data are stored in `ipsec.secrets`, for example:

```
@west @east : PPKS "user1" "thestringismeanttobearandomstr"
```

The `PPKS` option refers to static PPKs. This experimental function uses one-time-pad-based Dynamic PPKs. Upon each connection, a new part of the one-time pad is used as the PPK. When used, that part of the dynamic PPK inside the file is overwritten with zeros to prevent re-use. If there is no more one-time-pad material left, the connection fails. See the `ipsec.secrets(5)` man page for more information.

**WARNING**

The implementation of dynamic PPKs is provided as an unsupported Technology Preview. Use with caution.

### 26.3.3. Installing Libreswan

This procedure describes the steps for installing and starting the Libreswan IPsec/IKE VPN implementation.

**Prerequisites**

- The AppStream repository is enabled.

**Procedure**

1. Install the `libreswan` packages:

   ```
   # yum install libreswan
   ```

2. If you are re-installing Libreswan, remove its old database files and create a new database:

   ```
   # systemctl stop ipsec
   # rm /etc/ipsec.d/*db
   # ipsec initnss
   ```

3. Start the `ipsec` service, and enable the service to be started automatically on boot:

   ```
   # systemctl enable ipsec --now
   ```

4. Configure the firewall to allow 500 and 4500/UDP ports for the IKE, ESP, and AH protocols by adding the `ipsec` service:

   ```
   # firewall-cmd --add-service="ipsec"
   # firewall-cmd --runtime-to-permanent
   ```

### 26.3.4. Creating a host-to-host VPN
To configure **Libreswan** to create a host-to-host *IPsec* VPN between two hosts referred to as *left* and *right* using authentication by raw RSA keys, enter the following commands on both of the hosts:

**Prerequisites**

- **Libreswan** is installed and the *ipsec* service is started on each node.

**Procedure**

1. Generate a raw RSA key pair on each host:

   ```
   # ipsec newhostkey
   ```

2. The previous step returned the generated key’s *ckaid*. Use that *ckaid* with the following command on *left*, for example:

   ```
   # ipsec showhostkey --left --ckaid 2d3ea57b61c9419dfd6cf43a1eb6cb306c0e857d
   ```

   The output of the previous command generated the `leftrsasigkey=` line required for the configuration. Do the same on the second host (*right*):

   ```
   # ipsec showhostkey --right --ckaid a9e1f6ce9ecd3608c24e8f701318383f41798f03
   ```

3. In the `/etc/ipsec.d/` directory, create a new *my_host-to-host.conf* file. Write the RSA host keys from the output of the `ipsec showhostkey` commands in the previous step to the new file. For example:

   ```
   conn mytunnel
   leftid=@west
   left=192.1.2.23
   leftrsasigkey=0sAQOrlo+hOafUZDICQmXFrje/oZm [...] W2n417C/4urYHQkCvuIQ==
   rightid=@east
   right=192.1.2.45
   rightrasigkey=0sAQO3fwC6nSSGt64DWiYZzuHbc4 [...] D/v8t5YTQ==
   authby=rsasig
   ```

4. After importing keys, restart the *ipsec* service:

   ```
   # systemctl restart ipsec
   ```

5. Load the connection:

   ```
   # ipsec auto --add mytunnel
   ```

6. Establish the tunnel:

   ```
   # ipsec auto --up mytunnel
   ```

7. To automatically start the tunnel when the *ipsec* service is started, add the following line to the connection definition:

   ```
   auto=start
   ```
26.3.5. Configuring a site-to-site VPN

To create a site-to-site IPsec VPN, by joining two networks, an IPsec tunnel between the two hosts, is created. The hosts thus act as the end points, which are configured to permit traffic from one or more subnets to pass through. Therefore you can think of the host as gateways to the remote portion of the network.

The configuration of the site-to-site VPN only differs from the host-to-host VPN in that one or more networks or subnets must be specified in the configuration file.

**Prerequisites**

- A host-to-host VPN is already configured.

**Procedure**

1. Copy the file with the configuration of your host-to-host VPN to a new file, for example:

   ```
   # cp /etc/ipsec.d/my_host-to-host.conf /etc/ipsec.d/my_site-to-site.conf
   ```

2. Add the subnet configuration to the file created in the previous step, for example:

   ```
   conn mysubnet
   also=mytunnel
   leftsubnet=192.0.1.0/24
   rightsubnet=192.0.2.0/24
   auto=start
   
   conn mysubnet6
   also=mytunnel
   leftsubnet=2001:db8:0:1::/64
   rightsubnet=2001:db8:0:2::/64
   auto=start
   
   # the following part of the configuration file is the same for both host-to-host and site-to-site connections:

   conn mytunnel
   leftid=@west
   left=192.1.2.23
   lefrtrsasigkey=0sAQOrlo+hOafUZDlCQmXFrje/oZm [...] W2n417C/4urYHQkCvuIQ==
   rightid=@east
   right=192.1.2.45
   rightrsasigkey=0sAQO3fwC6nSSGgt64DWiYZzuHbc4 [...] D/v8t5YTQ==
   authby=rasig
   ```

26.3.6. Configuring a remote access VPN

Road warriors are traveling users with mobile clients and a dynamically assigned IP address. The mobile clients authenticate using X.509 certificates.

The following example shows configuration for IKEv2, and it avoids using the IKEv1 XAUTH protocol.

On the server:
conn roadwarriors
  ikev2=insist
  # support (roaming) MOBIKE clients (RFC 4555)
  mobike=yes
  fragmentation=yes
  left=1.2.3.4
  # if access to the LAN is given, enable this, otherwise use 0.0.0.0/0
  # leftsubnet=10.10.0.0/16
  leftsubnet=0.0.0.0/0
  leftcert=gw.example.com
  leftid=%fromcert
  leftxauthserver=yes
  leftmodecfgserver=yes
  right=%any
  # trust our own Certificate Agency
  rightca=%same
  # pick an IP address pool to assign to remote users
  # 100.64.0.0/16 prevents RFC1918 clashes when remote users are behind NAT
  rightaddresspool=100.64.13.100-100.64.13.254
  # if you want remote clients to use some local DNS zones and servers
  modecfgdns="1.2.3.4, 5.6.7.8"
  modecfgdomains="internal.company.com, corp"
  rightxauthclient=yes
  rightmodecfgclient=yes
  authby=rsasig
  # optionally, run the client X.509 ID through pam to allow or deny client
  # pam-authorize=yes
  # load connection, do not initiate
  auto=add
  # kill vanished roadwarriors
  dpddelay=1m
  dpdtimeout=5m
  dpdaction=clear

On the mobile client, the road warrior’s device, use a slight variation of the previous configuration:

conn to-vpn-server
  ikev2=insist
  # pick up our dynamic IP
  left=%defaultroute
  leftsubnet=0.0.0.0/0
  leftcert=mynname.example.com
  leftid=%fromcert
  leftmodecfgclient=yes
  # right can also be a DNS hostname
  right=1.2.3.4
  # if access to the remote LAN is required, enable this, otherwise use 0.0.0.0/0
  # rightsubnet=10.10.0.0/16
  rightsubnet=0.0.0.0/0
  fragmentation=yes
  # trust our own Certificate Agency
  rightca=%same
  authby=rsasig
  # allow narrowing to the server’s suggested assigned IP and remote subnet
  narrowing=yes
  # support (roaming) MOBIKE clients (RFC 4555)
26.3.7. Configuring a mesh VPN

A mesh VPN network, which is also known as an any-to-any VPN, is a network where all nodes communicate using IPsec. The configuration allows for exceptions for nodes that cannot use IPsec. The mesh VPN network can be configured in two ways:

- To require IPsec.
- To prefer IPsec but allow a fallback to clear-text communication.

Authentication between the nodes can be based on X.509 certificates or on DNS Security Extensions (DNSSEC).

The following procedure uses X.509 certificates. These certificates can be generated using any kind of Certificate Authority (CA) management system, such as the Dogtag Certificate System. Dogtag assumes that the certificates for each node are available in the PKCS #12 format (.p12 files), which contain the private key, the node certificate, and the Root CA certificate used to validate other nodes’ X.509 certificates.

Each node has an identical configuration with the exception of its X.509 certificate. This allows for adding new nodes without reconfiguring any of the existing nodes in the network. The PKCS #12 files require a "friendly name", for which we use the name “node” so that the configuration files referencing the friendly name can be identical for all nodes.

Prerequisites

- Libreswan is installed, and the ipsec service is started on each node.

Procedure

1. On each node, import PKCS #12 files. This step requires the password used to generate the PKCS #12 files:

```
# ipsec import nodeXXX.p12
```

2. Create the following three connection definitions for the IPsec required (private), IPsec optional (private-or-clear), and No IPsec (clear) profiles:

```
# cat /etc/ipsec.d/mesh.conf
conn clear
 auto=ondemand
type=passthrough
 authby=never
 left=%defaultroute
 right=%group

conn private
 auto=ondemand
 type=transport
 authby=rasasig
 failureshunt=drop
```
3. Add the IP address of the network in the proper category. For example, if all nodes reside in the 10.15.0.0/16 network, and all nodes should mandate IPsec encryption:

```
# echo "10.15.0.0/16" >> /etc/ipsec.d/policies/private
```

4. To allow certain nodes, for example, 10.15.34.0/24, to work with and without IPsec, add those nodes to the private-or-clear group using:

```
# echo "10.15.34.0/24" >> /etc/ipsec.d/policies/private-or-clear
```

5. To define a host, for example, 10.15.1.2, that is not capable of IPsec into the clear group, use:

```
# echo "10.15.1.2/32" >> /etc/ipsec.d/policies/clear
```

The files in the /etc/ipsec.d/policies directory can be created from a template for each new node, or can be provisioned using Puppet or Ansible.

Note that every node has the same list of exceptions or different traffic flow expectations. Two nodes, therefore, might not be able to communicate because one requires IPsec and the other cannot use IPsec.

6. Restart the node to add it to the configured mesh:

```
# systemctl restart ipsec
```

7. Once you finish with the addition of nodes, a ping command is sufficient to open an IPsec tunnel. To see which tunnels a node has opened:

```
-
# ipsec trafficstatus

## 26.3.8. Deploying a FIPS-compliant IPsec VPN

Use this procedure to deploy a FIPS-compliant IPsec VPN solution based on Libreswan. The following steps also enable you to identify which cryptographic algorithms are available and which are disabled for Libreswan in FIPS mode.

**Prerequisites**

- The **AppStream** repository is enabled.

**Procedure**

1. Install the **libreswan** packages:
   ```bash
   # yum install libreswan
   ```

2. If you are re-installing **Libreswan**, remove its old NSS database:
   ```bash
   # systemctl stop ipsec
   # rm /etc/ipsec.d/*db
   ```

3. Start the **ipsec** service, and enable the service to be started automatically on boot:
   ```bash
   # systemctl enable ipsec --now
   ```

4. Configure the firewall to allow 500 and 4500/UDP ports for the IKE, ESP, and AH protocols by adding the **ipsec** service:
   ```bash
   # firewall-cmd --add-service="ipsec"
   # firewall-cmd --runtime-to-permanent
   ```

5. Switch the system to FIPS mode:
   ```bash
   # fips-mode-setup --enable
   ```

6. Restart your system to allow the kernel to switch to FIPS mode:
   ```bash
   # reboot
   ```

**Verification**

1. To confirm Libreswan is running in FIPS mode:
   ```bash
   # ipsec whack --fipsstatus
   000 FIPS mode enabled
   ```

2. Alternatively, check entries for the **ipsec** unit in the **systemd** journal:
   ```bash
   $ journalctl -u ipsec
   ```
3. To see the available algorithms in FIPS mode:

```bash
# ipsec pluto --selftest 2>&1 | head -11
FIPS Product: YES
FIPS Kernel: YES
FIPS Mode: YES
NSS DB directory: sql:/etc/ipsec.d
Initializing NSS
Opening NSS database "sql:/etc/ipsec.d" read-only
NSS initialized
NSS crypto library initialized
FIPS HMAC integrity support [enabled]
FIPS mode enabled for pluto daemon
NSS library is running in FIPS mode
FIPS HMAC integrity verification self-test passed
```

4. To query disabled algorithms in FIPS mode:

```bash
# ipsec pluto --selftest 2>&1 | grep disabled
Encryption algorithm CAMELLIA_CTR disabled; not FIPS compliant
Encryption algorithm CAMELLIA_CBC disabled; not FIPS compliant
Encryption algorithm SERPENT_CBC disabled; not FIPS compliant
Encryption algorithm TWOFISH_CBC disabled; not FIPS compliant
Encryption algorithm TWOFISH_SSH disabled; not FIPS compliant
Encryption algorithm NULL disabled; not FIPS compliant
Encryption algorithm CHACHA20_POLY1305 disabled; not FIPS compliant
Hash algorithm MD5 disabled; not FIPS compliant
PRF algorithm HMAC_MD5 disabled; not FIPS compliant
PRF algorithm AES_XCBC disabled; not FIPS compliant
Integrity algorithm HMAC_MD5_96 disabled; not FIPS compliant
Integrity algorithm HMAC_SHA2_256_TRUNCBUG disabled; not FIPS compliant
Integrity algorithm AES_XCBC_96 disabled; not FIPS compliant
DH algorithm MODP1024 disabled; not FIPS compliant
DH algorithm MODP1536 disabled; not FIPS compliant
DH algorithm DH31 disabled; not FIPS compliant
```

5. To list all allowed algorithms and ciphers in FIPS mode:

```bash
# ipsec pluto --selftest 2>&1 | grep ESP | grep FIPS | sed "s/^.*FIPS//"
{256,192,*128}  aes_ccm, aes_ccm_c
{256,192,*128}  aes_ccm_b
{256,192,*128}  aes_ccm_a
[*192]  3des
{256,192,*128}  aes_gcm, aes_gcm_c
{256,192,*128}  aes_gcm_b
{256,192,*128}  aes_gcm_a
{256,192,*128}  aesctr
{256,192,*128}  aes
{256,192,*128}  aes_gmac
sha, sha1, sha1_96, hmac_sha1
```
| sha512, sha2_512, sha2_512_256, hmac_sha2_512  |
| sha384, sha2_384, sha2_384_192, hmac_sha2_384  |
| sha2, sha256, sha2_256, sha2_256_128, hmac_sha2_256 |
| aes_cmac                                      |
| null                                          |
| null, dh0                                     |
| dh14                                          |
| dh15                                          |
| dh16                                          |
| dh17                                          |
| dh18                                          |
| ecp_256, ecp256                                |
| ecp_384, ecp384                                |
| ecp_521, ecp521                                |

Additional resources

- [Using system-wide cryptographic policies](#)

### 26.3.9. Protecting the IPsec NSS database by a password

By default, the IPsec service creates its Network Security Services (NSS) database with an empty password during the first start. Add password protection by using the following steps.

**NOTE**

In the previous releases of RHEL up to version 6.6, you had to protect the IPsec NSS database with a password to meet the FIPS 140-2 requirements because the NSS cryptographic libraries were certified for the FIPS 140-2 Level 2 standard. In RHEL 8, NIST certified NSS to Level 1 of this standard, and this status does not require password protection for the database.

**Prerequisites**

- The `/etc/ipsec.d/` directory contains NSS database files.

**Procedure**

1. Enable password protection for the NSS database for Libreswan:

   ```bash
   # certutil -N -d sql:/etc/ipsec.d
   Enter Password or Pin for "NSS Certificate DB":
   Enter a password which will be used to encrypt your keys. The password should be at least 8 characters long, and should contain at least one non-alphabetic character.
   Enter new password:
   ```

2. Create the `/etc/ipsec.d/nsspassword` file containing the password you have set in the previous step, for example:

   ```bash
   # cat /etc/ipsec.d/nsspassword
   NSS Certificate DB:MyStrongPasswordHere
   ```
Note that the `nsspassword` file uses the following syntax:

```
token_1_name:the_password
token_2_name:the_password
```

The default NSS software token is **NSS Certificate DB**. If your system is running in FIPS mode, the name of the token is **NSS FIPS 140-2 Certificate DB**.

3. Depending on your scenario, either start or restart the `ipsec` service after you finish the `nsspassword` file:

```
# systemctl restart ipsec
```

**Verification**

1. Check that the `ipsec` service is running after you have added a non-empty password to its NSS database:

```
# systemctl status ipsec
```

```
● ipsec.service - Internet Key Exchange (IKE) Protocol Daemon for IPsec
   Loaded: loaded (/usr/lib/systemd/system/ipsec.service; enabled; vendor preset: disable>
   Active: active (running)...
```

2. Optionally, check that the **Journal** log contains entries confirming a successful initialization:

```
# journalctl -u ipsec
```

```
pluto[23001]: NSS DB directory: sql:/etc/ipsec.d
pluto[23001]: Initializing NSS
pluto[23001]: Opening NSS database "sql:/etc/ipsec.d" read-only
pluto[23001]: NSS Password from file "/etc/ipsec.d/nsspassword" for token "NSS Certificate DB" with length 20 passed to NSS
pluto[23001]: NSS crypto library initialized
```

**Additional resources**

- `certutil(1)` man page.
- Government Standards Knowledgebase article.

### 26.3.10. Configuring an IPsec VPN to use TCP

Libreswan supports TCP encapsulation of IKE and IPsec packets as described in RFC 8229. With this feature, you can establish IPsec VPNs on networks that prevent traffic transmitted via UDP and Encapsulating Security Payload (ESP). You can configure VPN servers and clients to use TCP either as a fallback or as the main VPN transport protocol. Because TCP encapsulation has bigger performance costs, use TCP as the main VPN protocol only if UDP is permanently blocked in your scenario.

**Prerequisites**

- A **remote-access VPN** is already configured.

**Procedure**
Procedure

1. Add the following option to the `/etc/ipsec.conf` file in the `config setup` section:

   ```
   listen-tcp=yes
   ```

2. To use TCP encapsulation as a fallback option when the first attempt over UDP fails, add the following two options to the client’s connection definition:

   ```
   enable-tcp=fallback
tcp-remoteport=4500
   ```

   Alternatively, if you know that UDP is permanently blocked, use the following options in the client’s connection configuration:

   ```
   enable-tcp=yes
tcp-remoteport=4500
   ```

Additional resources

- IETF RFC 8229: TCP Encapsulation of IKE and IPsec Packets.

26.3.11. Configuring automatic detection and usage of ESP hardware offload to accelerate an IPsec connection

Offloading Encapsulating Security Payload (ESP) to the hardware accelerates IPsec connections over Ethernet. By default, Libreswan detects if hardware supports this feature and, as a result, enables ESP hardware offload. This procedure describes how to enable the automatic detection in case that the feature was disabled or explicitly enabled.

Prerequisites

- The network card supports ESP hardware offload.
- The network driver supports ESP hardware offload.
- The IPsec connection is configured and works.

Procedure

1. Edit the Libreswan configuration file in the `/etc/ipsec.d/` directory of the connection that should use automatic detection of ESP hardware offload support.

2. Ensure the `nic-offload` parameter is not set in the connection’s settings.

3. If you removed `nic-offload`, restart the `ipsec` service:

   ```
   # systemctl restart ipsec
   ```

Verification

If the network card supports ESP hardware offload support, following these steps to verify the result:

1. Display the `tx_ipsec` and `rx_ipsec` counters of the Ethernet device the IPsec connection uses:
ethtool enp1s0 | egrep "_ipsec"
  tx_ipsec: 10
  rx_ipsec: 10

2. Send traffic through the IPsec tunnel. For example, ping a remote IP address:

  # ping -c 5 remote_ip_address

3. Display the tx_ipsec and rx_ipsec counters of the Ethernet device again:

  # ethtool enp1s0 | egrep "_ipsec"
  tx_ipsec: 15
  rx_ipsec: 15

If the counter values have increased, ESP hardware offload works.

Additional resources

- Configuring a VPN with IPsec

26.3.12. Configuring ESP hardware offload on a bond to accelerate an IPsec connection

Offloading Encapsulating Security Payload (ESP) to the hardware accelerates IPsec connections. If you use a network bond for fail-over reasons, the requirements and the procedure to configure ESP hardware offload are different from those using a regular Ethernet device. For example, in this scenario, you enable the offload support on the bond, and the kernel applies the settings to the ports of the bond.

Prerequisites

- All network cards in the bond support ESP hardware offload.
- The network driver supports ESP hardware offload on a bond device. In RHEL, only the ixgbe driver supports this feature.
- The bond is configured and works.
- The bond uses the active-backup mode. The bonding driver does not support any other modes for this feature.
- The IPsec connection is configured and works.

Procedure

1. Enable ESP hardware offload support on the network bond:

   # nmcli connection modify bond0 ethtool.feature-esp-hw-offload on

   This command enables ESP hardware offload support on the bond0 connection.

2. Reactivate the bond0 connection:

   # nmcli connection up bond0
3. Edit the Libreswan configuration file in the `/etc/ipsec.d/` directory of the connection that should use ESP hardware offload, and append the `nic-offload=yes` statement to the connection entry:

```
conn example
  ...
  nic-offload=yes
```

4. Restart the `ipsec` service:

```
# systemctl restart ipsec
```

**Verification**

1. Display the active port of the bond:

```
# grep "Currently Active Slave" /proc/net/bonding/bond0
Currently Active Slave: enp1s0
```

2. Display the `tx_ipsec` and `rx_ipsec` counters of the active port:

```
# ethtool enp1s0 | egrep "_ipsec"
  tx_ipsec: 10
  rx_ipsec: 10
```

3. Send traffic through the IPsec tunnel. For example, ping a remote IP address:

```
# ping -c 5 remote_ip_address
```

4. Display the `tx_ipsec` and `rx_ipsec` counters of the active port again:

```
# ethtool enp1s0 | egrep "_ipsec"
  tx_ipsec: 15
  rx_ipsec: 15
```

If the counter values have increased, ESP hardware offload works.

**Additional resources**

- Configuring network bonding
- The Configuring a VPN with IPsec section in the **Securing networks** documentation
- Configuring a VPN with IPsec chapter in the **Securing networks** document.

26.3.13. Configuring IPsec connections that opt out of the system-wide crypto policies

**Overriding system-wide crypto-policies for a connection**

The RHEL system-wide cryptographic policies create a special connection called `%default`. This connection contains the default values for the `ikev2`, `esp`, and `ike` options. However, you can override the default values by specifying the mentioned option in the connection configuration file.
For example, the following configuration allows connections that use IKEv1 with AES and SHA-1 or SHA-2, and IPsec (ESP) with either AES-GCM or AES-CBC:

```
conn MyExample
...
ikev2=never
ike=aes-sha2,aes-sha1;modp2048
esp=aes_gcm,aes-sha2,aes-sha1
...
```

Note that AES-GCM is available for IPsec (ESP) and for IKEv2, but not for IKEv1.

Disabling system-wide crypto policies for all connections

To disable system-wide crypto policies for all IPsec connections, comment out the following line in the `/etc/ipsec.conf` file:

```
include /etc/crypto-policies/back-ends/libreswan.config
```

Then add the `ikev2=never` option to your connection configuration file.

Additional resources

- [Using system-wide cryptographic policies](#)

26.3.14. Troubleshooting IPsec VPN configurations

Problems related to IPsec VPN configurations most commonly occur due to several main reasons. If you are encountering such problems, you can check if the cause of the problem corresponds to any of the following scenarios, and apply the corresponding solution.

Basic connection troubleshooting

Most problems with VPN connections occur in new deployments, where administrators configured endpoints with mismatched configuration options. Also, a working configuration can suddenly stop working, often due to newly introduced incompatible values. This could be the result of an administrator changing the configuration. Alternatively, an administrator may have installed a firmware update or a package update with different default values for certain options, such as encryption algorithms.

To confirm that an IPsec VPN connection is established:

```
# ipsec trafficstatus
006 #8: "vpn.example.com"[1] 192.0.2.1, type=ESP, add_time=1595296930, inBytes=5999,
    outBytes=3231, id=@vpn.example.com, lease=100.64.13.5/32
```

If the output is empty or does not show an entry with the connection name, the tunnel is broken.

To check that the problem is in the connection:

1. Reload the `vpn.example.com` connection:

```
# ipsec auto --add vpn.example.com
002 added connection description "vpn.example.com"
```

2. Next, initiate the VPN connection:
Firewall-related problems

The most common problem is that a firewall on one of the IPsec endpoints or on a router between the endpoints is dropping all Internet Key Exchange (IKE) packets.

- For IKEv2, an output similar to the following example indicates a problem with a firewall:

  ```
  # ipsec auto --up vpn.example.com
  181 "vpn.example.com"[1] 192.0.2.2 #15: initiating IKEv2 IKE SA
  181 "vpn.example.com"[1] 192.0.2.2 #15: STATE_PARENT_I1: sent v2I1, expected v2R1
  010 "vpn.example.com"[1] 192.0.2.2 #15: STATE_PARENT_I1: retransmission; will wait 0.5 seconds for response
  010 "vpn.example.com"[1] 192.0.2.2 #15: STATE_PARENT_I1: retransmission; will wait 1 seconds for response
  010 "vpn.example.com"[1] 192.0.2.2 #15: STATE_PARENT_I1: retransmission; will wait 2 seconds for response...
  ```

- For IKEv1, the output of the initiating command looks like:

  ```
  # ipsec auto --up vpn.example.com
  002 "vpn.example.com" #9: initiating Main Mode
  102 "vpn.example.com" #9: STATE_MAIN_I1: sent MI1, expecting MR1
  010 "vpn.example.com" #9: STATE_MAIN_I1: retransmission; will wait 0.5 seconds for response
  010 "vpn.example.com" #9: STATE_MAIN_I1: retransmission; will wait 1 seconds for response
  010 "vpn.example.com" #9: STATE_MAIN_I1: retransmission; will wait 2 seconds for response...
  ```

Because the IKE protocol, which is used to set up IPsec, is encrypted, you can troubleshoot only a limited subset of problems using the `tcpdump` tool. If a firewall is dropping IKE or IPsec packets, you can try to find the cause using the `tcpdump` utility. However, `tcpdump` cannot diagnose other problems with IPsec VPN connections.

- To capture the negotiation of the VPN and all encrypted data on the `eth0` interface:

  ```
  # tcpdump -i eth0 -n -n esp or udp port 500 or udp port 4500 or tcp port 4500
  ```

Mismatched algorithms, protocols, and policies

VPN connections require that the endpoints have matching IKE algorithms, IPsec algorithms, and IP address ranges. If a mismatch occurs, the connection fails. If you identify a mismatch by using one of the following methods, fix it by aligning algorithms, protocols, or policies.

- If the remote endpoint is not running IKE/IPsec, you can see an ICMP packet indicating it. For example:

  ```
  # ipsec auto --up vpn.example.com
  000 "vpn.example.com"[1] 192.0.2.2 #16: ERROR: asynchronous network error report on
  ```
Example of mismatched IKE algorithms:

```bash
# ipsec auto --up vpn.example.com
...
003 "vpn.example.com"[1] 193.110.157.148 #3: dropping unexpected IKE_SA_INIT message containing NO_PROPOSAL_CHOSEN notification; message payloads: N; missing payloads: SA,KE,Ni
```

Example of mismatched IPsec algorithms:

```bash
# ipsec auto --up vpn.example.com
...
182 "vpn.example.com"[1] 193.110.157.148 #5: STATE_PARENT_I2: sent v2I2, expected v2R2 {auth=IKEv2 cipher=AES_GCM_16_256 integ=n/a prf=HMAC_SHA2_256 group=MODP2048}
002 "vpn.example.com"[1] 193.110.157.148 #6: IKE_AUTH response contained the error notification NO_PROPOSAL_CHOSEN
```

A mismatched IKE version could also result in the remote endpoint dropping the request without a response. This looks identical to a firewall dropping all IKE packets.

Example of mismatched IP address ranges for IKEv2 (called Traffic Selectors - TS):

```bash
# ipsec auto --up vpn.example.com
...
1v2 "vpn.example.com" #1: STATE_PARENT_I2: sent v2I2, expected v2R2 {auth=IKEv2 cipher=AES_GCM_16_256 integ=n/a prf=HMAC_SHA2_512 group=MODP2048}
002 "vpn.example.com" #2: IKE_AUTH response contained the error notification TS_UNACCEPTABLE
```

Example of mismatched IP address ranges for IKEv1:

```bash
# ipsec auto --up vpn.example.com
...
031 "vpn.example.com" #2: STATE_QUICK_I1: 60 second timeout exceeded after 0 retransmits. No acceptable response to our first Quick Mode message: perhaps peer likes no proposal
```

When using PreSharedKeys (PSK) in IKEv1, if both sides do not put in the same PSK, the entire IKE message becomes unreadable:

```bash
# ipsec auto --up vpn.example.com
...
003 "vpn.example.com" #1: received Hash Payload does not match computed value
223 "vpn.example.com" #1: sending notification INVALID_HASH_INFORMATION to 192.0.2.23:500
```

In IKEv2, the mismatched-PSK error results in an AUTHENTICATION_FAILED message:

```bash
# ipsec auto --up vpn.example.com
```
Maximum transmission unit

Other than firewalls blocking IKE or IPsec packets, the most common cause of networking problems relates to an increased packet size of encrypted packets. Network hardware fragments packets larger than the maximum transmission unit (MTU), for example, 1500 bytes. Often, the fragments are lost and the packets fail to re-assemble. This leads to intermittent failures, when a ping test, which uses small-sized packets, works but other traffic fails. In this case, you can establish an SSH session but the terminal freezes as soon as you use it, for example, by entering the `ls -al /usr` command on the remote host.

To work around the problem, reduce MTU size by adding the `mtu=1400` option to the tunnel configuration file.

Alternatively, for TCP connections, enable an iptables rule that changes the MSS value:

```
# iptables -I FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu
```

If the previous command does not solve the problem in your scenario, directly specify a lower size in the `set-mss` parameter:

```
# iptables -I FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --set-mss 1380
```

Network address translation (NAT)

When an IPsec host also serves as a NAT router, it could accidentally remap packets. The following example configuration demonstrates the problem:

```
conn myvpn
  left=172.16.0.1
  leftsubnet=10.0.2.0/24
  right=172.16.0.2
  rightsubnet=192.168.0.0/16
...
```

The system with address 172.16.0.1 have a NAT rule:

```
iptables -t nat -I POSTROUTING -o eth0 -j MASQUERADE
```

If the system on address 10.0.2.33 sends a packet to 192.168.0.1, then the router translates the source 10.0.2.33 to 172.16.0.1 before it applies the IPsec encryption.

Then, the packet with the source address 10.0.2.33 no longer matches the `conn myvpn` configuration, and IPsec does not encrypt this packet.

To solve this problem, insert rules that exclude NAT for target IPsec subnet ranges on the router, in this example:

```
iptables -t nat -I POSTROUTING -s 10.0.2.0/24 -d 192.168.0.0/16 -j RETURN
```

Kernel IPsec subsystem bugs
The kernel IPsec subsystem might fail, for example, when a bug causes a desynchronizing of the IKE user space and the IPsec kernel. To check for such problems:

```
$ cat /proc/net/xfrm_stat
XfrmInError                 0
XfrmInBufferError           0
...
```

Any non-zero value in the output of the previous command indicates a problem. If you encounter this problem, open a new support case, and attach the output of the previous command along with the corresponding IKE logs.

**Libreswan logs**

Libreswan logs using the *syslog* protocol by default. You can use the `journalctl` command to find log entries related to IPsec. Because the corresponding entries to the log are sent by the *pluto* IKE daemon, search for the “pluto” keyword, for example:

```
$ journalctl -b | grep pluto
```

To show a live log for the *ipsec* service:

```
$ journalctl -f -u ipsec
```

If the default level of logging does not reveal your configuration problem, enable debug logs by adding the `plutodebug=all` option to the *config setup* section in the `/etc/ipsec.conf` file.

Note that debug logging produces a lot of entries, and it is possible that either the *journald* or *syslogd* service rate-limits the *syslog* messages. To ensure you have complete logs, redirect the logging to a file. Edit the `/etc/ipsec.conf`, and add the `logfile=/var/log/pluto.log` in the *config setup* section.

**Additional resources**

- Troubleshooting problems using log files.
- `tcpdump(8)` and `ipsec.conf(5)` man pages.
- Using and configuring firewalld

### 26.3.15. Additional resources

- `ipsec(8)`, `ipsec.conf(5)`, `ipsec.secrets(5)`, `ipsec_auto(8)`, and `ipsec_rsasigkey(8)` man pages.
- `/usr/share/doc/libreswan-version/` directory.
- The website of the upstream project.
- The Libreswan Project Wiki.
- All Libreswan man pages.
- NIST Special Publication 800-77: Guide to IPsec VPNs.
26.4. USING MACSEC TO ENCRYPT LAYER-2 TRAFFIC IN THE SAME PHYSICAL NETWORK

This section describes how to configure MACsec for secure communication for all traffic on Ethernet links.

Media Access Control security (MACsec) is a layer 2 protocol that secures different traffic types over the Ethernet links including:

- dynamic host configuration protocol (DHCP)
- address resolution protocol (ARP)
- Internet Protocol version 4 / 6 (IPv4 / IPv6) and
- any traffic over IP such as TCP or UDP

MACsec encrypts and authenticates all traffic in LANs, by default with the GCM-AES-128 algorithm, and uses a pre-shared key to establish the connection between the participant hosts. If you want to change the pre-shared key, you need to update the NM configuration on all hosts in the network that uses MACsec.

A MACsec connection uses an Ethernet device, such as an Ethernet network card, VLAN, or tunnel device, as parent. You can either set an IP configuration only on the MACsec device to communicate with other hosts only using the encrypted connection, or you can also set an IP configuration on the parent device. In the latter case, you can use the parent device to communicate with other hosts using an unencrypted connection and the MACsec device for encrypted connections.

MACsec does not require any special hardware. For example, you can use any switch, except if you want to encrypt traffic only between a host and a switch. In this scenario, the switch must also support MACsec.

In other words, there are 2 common methods to configure MACsec;

- host to host and
- host to switch then switch to other host(s)

**IMPORTANT**

You can use MACsec only between hosts that are in the same (physical or virtual) LAN.

The following example shows how to configure MACsec between 2 hosts using a pre-shared key.

26.4.1. Configuring a MACsec connection using nmcli

You can configure Ethernet interfaces to use MACsec using the nmcli tool. This procedure describes how to create a MACsec connection that uses an Ethernet interface to encrypt the network traffic.

Run this procedure on all the hosts that should communicate in this MACsec-protected network.

**Procedure**

On Host A:
On the first host on which you configure MACsec, create the connectivity association key (CAK) and connectivity-association key name (CKN) for the pre-shared key:

a. Create 16-byte hexadecimal CAK:

```
dd if=/dev/urandom count=16 bs=1 2> /dev/null | hexdump -e '1/2 "%04x"
```

50b71a8ef0bd5751ea76de6d6c98c03a

b. Create 32-byte hexadecimal CKN:

```
dd if=/dev/urandom count=32 bs=1 2> /dev/null | hexdump -e '1/2 "%04x"
```

f2b4297d39da7330910a74abc0449feb45b5c0b9fc23df1430e1898fcf1c4550

On Host A and B:

1. Create the MACsec connection:

```
# nmcli connection add type macsec con-name macsec0 ifname macsec0
connection.autoconnect yes macsec.parent enp1s0 macsec.mode psk macsec.mka-cak
50b71a8ef0bd5751ea76de6d6c98c03a
macsec.mka-ckn f2b4297d39da7330910a74abc0449feb45b5c0b9fc23df1430e1898fcf1c4550
```

Use the CAK and CKN generated in the previous step in the `macsec.mka-cak` and `macsec.mka-ckn` parameters. The values must be the same on every host in the MACsec-protected network.

2. Configure the IP settings on the MACsec connection.

a. Configure the IPv4 settings. For example, to set a static IPv4 address, network mask, default gateway, and DNS server to the `macsec0` connection, enter:

```
# nmcli connection modify macsec0 ipv4.method manual ipv4.addresses
'192.0.2.1/24' ipv4.gateway '192.0.2.254' ipv4.dns '192.0.2.253'
```

b. Configure the IPv6 settings. For example, to set a static IPv6 address, network mask, default gateway, and DNS server to the `macsec0` connection, enter:

```
# nmcli connection modify macsec0 ipv6.method manual ipv6.addresses
```

3. Activate the connection:

```
# nmcli connection up macsec0
```

Verification steps

1. To verify the traffic is encrypted, enter:

```
tcpdump -nn -i enp1s0
```

2. To view the unencrypted traffic, enter:

```
tcpdump -nn -i macsec0
```
3. To display MACsec statistics:

```
# ip macsec show
```

4. To display individual counters for each type of protection: integrity-only (encrypt off) and encryption (encrypt on)

```
# ip -s macsec show
```

### 26.4.2. Additional resources

- [MACsec: a different solution to encrypt network traffic blog.](#)

### 26.5. USING AND CONFIGURING FIREWALLD

A **firewall** is a way to protect machines from any unwanted traffic from outside. It enables users to control incoming network traffic on host machines by defining a set of **firewall rules**. These rules are used to sort the incoming traffic and either block it or allow through.

**firewalld** is a firewall service daemon that provides a dynamic customizable host-based firewall with a D-Bus interface. Being dynamic, it enables creating, changing, and deleting the rules without the necessity to restart the firewall daemon each time the rules are changed.

**firewalld** uses the concepts of zones and services, that simplify the traffic management. Zones are predefined sets of rules. Network interfaces and sources can be assigned to a zone. The traffic allowed depends on the network your computer is connected to and the security level this network is assigned. Firewall services are predefined rules that cover all necessary settings to allow incoming traffic for a specific service and they apply within a zone.

Services use one or more ports or addresses for network communication. Firewalls filter communication based on ports. To allow network traffic for a service, its ports must be open. **firewalld** blocks all traffic on ports that are not explicitly set as open. Some zones, such as trusted, allow all traffic by default.

Note that **firewalld** with **nftables** backend does not support passing custom **nftables** rules to **firewalld**, using the `--direct` option.

### 26.5.1. Getting started with firewalld

This section provides information about **firewalld**.

#### 26.5.1.1. When to use firewalld, nftables, or iptables

The following is a brief overview in which scenario you should use one of the following utilities:

- **firewalld**: Use the **firewalld** utility for simple firewall use cases. The utility is easy to use and covers the typical use cases for these scenarios.

- **nftables**: Use the **nftables** utility to set up complex and performance critical firewalls, such as for a whole network.

- **iptables**: The **iptables** utility on Red Hat Enterprise Linux uses the **nf_tables** kernel API instead of the **legacy** back end. The **nf_tables** API provides backward compatibility so that scripts that use **iptables** commands still work on Red Hat Enterprise Linux. For new firewall scripts, Red Hat recommends to use **nftables**.
IMPORTNT

To avoid that the different firewall services influence each other, run only one of them on a RHEL host, and disable the other services.

26.5.1.2. Zones

firewalld can be used to separate networks into different zones according to the level of trust that the user has decided to place on the interfaces and traffic within that network. A connection can only be part of one zone, but a zone can be used for many network connections.

NetworkManager notifies firewalld of the zone of an interface. You can assign zones to interfaces with:

- NetworkManager
- firewall-config tool
- firewall-cmd command-line tool
- The RHEL web console

The latter three can only edit the appropriate NetworkManager configuration files. If you change the zone of the interface using the web console, firewall-cmd or firewall-config, the request is forwarded to NetworkManager and is not handled by firewalld.

The predefined zones are stored in the /usr/lib/firewalld/zones/ directory and can be instantly applied to any available network interface. These files are copied to the /etc/firewalld/zones/ directory only after they are modified. The default settings of the predefined zones are as follows:

block
Any incoming network connections are rejected with an icmp-host-prohibited message for IPv4 and icmp6-adm-prohibited for IPv6. Only network connections initiated from within the system are possible.

dmz
For computers in your demilitarized zone that are publicly-accessible with limited access to your internal network. Only selected incoming connections are accepted.

drop
Any incoming network packets are dropped without any notification. Only outgoing network connections are possible.

external
For use on external networks with masquerading enabled, especially for routers. You do not trust the other computers on the network to not harm your computer. Only selected incoming connections are accepted.

home
For use at home when you mostly trust the other computers on the network. Only selected incoming connections are accepted.

internal
For use on internal networks when you mostly trust the other computers on the network. Only selected incoming connections are accepted.

public
For use in public areas where you do not trust other computers on the network. Only selected incoming connections are accepted.
trusted

All network connections are accepted.

work

For use at work where you mostly trust the other computers on the network. Only selected incoming connections are accepted.

One of these zones is set as the default zone. When interface connections are added to NetworkManager, they are assigned to the default zone. On installation, the default zone in firewalld is set to be the public zone. The default zone can be changed.

NOTE

The network zone names should be self-explanatory and to allow users to quickly make a reasonable decision. To avoid any security problems, review the default zone configuration and disable any unnecessary services according to your needs and risk assessments.

Additional resources

- The firewalld.zone(5) man page.

26.5.1.3. Predefined services

A service can be a list of local ports, protocols, source ports, and destinations, as well as a list of firewall helper modules automatically loaded if a service is enabled. Using services saves users time because they can achieve several tasks, such as opening ports, defining protocols, enabling packet forwarding and more, in a single step, rather than setting up everything one after another.

Service configuration options and generic file information are described in the firewalld.service(5) man page. The services are specified by means of individual XML configuration files, which are named in the following format: service-name.xml. Protocol names are preferred over service or application names in firewalld.

Services can be added and removed using the graphical firewall-config tool, firewall-cmd, and firewall-offline-cmd.

Alternatively, you can edit the XML files in the /etc/firewalld/services/ directory. If a service is not added or changed by the user, then no corresponding XML file is found in /etc/firewalld/services/. The files in the /usr/lib/firewalld/services/ directory can be used as templates if you want to add or change a service.

Additional resources

- The firewalld.service(5) man page

26.5.1.4. Starting firewalld

Procedure

1. To start firewalld, enter the following command as root:

   ```
   # systemctl unmask firewalld
   # systemctl start firewalld
   ```
2. To ensure **firewalld** starts automatically at system start, enter the following command as **root**:

```
# systemctl enable firewalld
```

### 26.5.1.5. Stopping firewalld

**Procedure**

1. To stop **firewalld**, enter the following command as **root**:

```
# systemctl stop firewalld
```

2. To prevent **firewalld** from starting automatically at system start:

```
# systemctl disable firewalld
```

3. To make sure firewalld is not started by accessing the **firewalld D-Bus** interface and also if other services require **firewalld**:

```
# systemctl mask firewalld
```

### 26.5.1.6. Verifying the permanent firewalld configuration

In certain situations, for example after manually editing **firewalld** configuration files, administrators want to verify that the changes are correct. This section describes how to verify the permanent configuration of the **firewalld** service.

**Prerequisites**

- The **firewalld** service is running.

**Procedure**

1. Verify the permanent configuration of the **firewalld** service:

```
# firewall-cmd --check-config
success
```

If the permanent configuration is valid, the command returns **success**. In other cases, the command returns an error with further details, such as the following:

```
# firewall-cmd --check-config
Error: INVALID_PROTOCOL: 'public.xml': 'tcpx' not from {'tcp','udp','sctp','dccp'}
```

### 26.5.2. Viewing the current status and settings of firewalld

This section covers information about viewing current status, allowed services, and current settings of **firewalld**.

#### 26.5.2.1. Viewing the current status of firewalld


The firewall service, `firewalld`, is installed on the system by default. Use the `firewalld` CLI interface to check that the service is running.

Procedure

1. To see the status of the service:
   ```
   # firewall-cmd --state
   ```

2. For more information about the service status, use the `systemctl status` sub-command:
   ```
   # systemctl status firewalld
   firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor provided)
   Active: active (running) since Mon 2017-12-18 16:05:15 CET; 50min ago
   Docs: man:firewalld(1)
   Main PID: 705 (firewalld)
   Tasks: 2 (limit: 4915)
   CGroup: /system.slice/firewalld.service
   └─ 705 /usr/bin/python3 -Es /usr/sbin/firewalld --nofork --noid
   ```

26.5.2.2. Viewing allowed services using GUI

To view the list of services using the graphical `firewall-config` tool, press the `Super` key to enter the Activities Overview, type `firewall`, and press `Enter`. The `firewall-config` tool appears. You can now view the list of services under the `Services` tab.

You can start the graphical firewall configuration tool using the command-line.

Prerequisites

- You installed the `firewall-config` package.

Procedure

- To start the graphical firewall configuration tool using the command-line:
  ```
  $ firewall-config
  ```

The `Firewall Configuration` window opens. Note that this command can be run as a normal user, but you are prompted for an administrator password occasionally.

26.5.2.3. Viewing `firewalld` settings using CLI

With the CLI client, it is possible to get different views of the current firewall settings. The `--list-all` option shows a complete overview of the `firewalld` settings.

`firewalld` uses zones to manage the traffic. If a zone is not specified by the `--zone` option, the command is effective in the default zone assigned to the active network interface and connection.

Procedure

- To list all the relevant information for the default zone:
To specify the zone for which to display the settings, add the `--zone=zone-name` argument to the `firewall-cmd --list-all` command, for example:

```
# firewall-cmd --list-all --zone=home
home
    target: default
    icmp-block-inversion: no
    interfaces:
    sources:
    services: ssh mdns samba-client dhcpv6-client
    ... [trimmed for clarity]
```

To see the settings for particular information, such as services or ports, use a specific option. See the `firewalld` manual pages or get a list of the options using the command `help`:

```
# firewall-cmd --help
```

To see which services are allowed in the current zone:

```
# firewall-cmd --list-services
ssh dhcpv6-client
```

**NOTE**

Listing the settings for a certain subpart using the CLI tool can sometimes be difficult to interpret. For example, you allow the SSH service and `firewalld` opens the necessary port (22) for the service. Later, if you list the allowed services, the list shows the SSH service, but if you list open ports, it does not show any. Therefore, it is recommended to use the `--list-all` option to make sure you receive a complete information.

### 26.5.3. Controlling network traffic using `firewalld`

This section covers information about controlling network traffic using `firewalld`.

### 26.5.3.1. Disabling all traffic in case of emergency using CLI

In an emergency situation, such as a system attack, it is possible to disable all network traffic and cut off the attacker.
**Procedure**

1. To immediately disable networking traffic, switch panic mode on:

   ```
   # firewall-cmd --panic-on
   ```

   **IMPORTANT**

   Enabling panic mode stops all networking traffic. For this reason, it should be used only when you have the physical access to the machine or if you are logged in using a serial console.

2. Switching off panic mode reverts the firewall to its permanent settings. To switch panic mode off, enter:

   ```
   # firewall-cmd --panic-off
   ```

**Verification**

- To see whether panic mode is switched on or off, use:

```
# firewall-cmd --query-panic
```

**26.5.3.2. Controlling traffic with predefined services using CLI**

The most straightforward method to control traffic is to add a predefined service to `firewalld`. This opens all necessary ports and modifies other settings according to the service definition file.

**Procedure**

1. Check that the service is not already allowed:

   ```
   # firewall-cmd --list-services
   ssh dhcpv6-client
   ```

2. List all predefined services:

   ```
   # firewall-cmd --get-services
   RH-Satellite-6 amanda-client amanda-k5-client bacula bacula-client bitcoin bitcoin-rpc
   bitcoin-testnet bitcoin-testnet-rpc ceph ceph-mon cfengine condor-collector ctdb dhcp dhcpv6
   dhcpv6-client dns docker-registry ...
   [trimmed for clarity]
   ```

3. Add the service to the allowed services:

   ```
   # firewall-cmd --add-service=<service-name>
   ```

4. Make the new settings persistent:

   ```
   # firewall-cmd --runtime-to-permanent
   ```
26.5.3.3. Controlling traffic with predefined services using GUI

This procedure describes how to control the network traffic with predefined services using graphical user interface.

**Prerequisites**
- You installed the `firewall-config` package

**Procedure**

1. To enable or disable a predefined or custom service:
   a. Start the `firewall-config` tool and select the network zone whose services are to be configured.
   b. Select the **Zones** tab and then the **Services** tab below.
   c. Select the check box for each type of service you want to trust or clear the check box to block a service in the selected zone.

2. To edit a service:
   a. Start the `firewall-config` tool.
   b. Select **Permanent** from the menu labeled **Configuration**. Additional icons and menu buttons appear at the bottom of the **Services** window.
   c. Select the service you want to configure.

The **Ports**, **Protocols**, and **Source Port** tabs enable adding, changing, and removing of ports, protocols, and source port for the selected service. The modules tab is for configuring **Netfilter** helper modules. The **Destination** tab enables limiting traffic to a particular destination address and Internet Protocol (IPv4 or IPv6).

**NOTE**

It is not possible to alter service settings in the **Runtime** mode.

26.5.3.4. Adding new services

Services can be added and removed using the graphical `firewall-config` tool, `firewall-cmd`, and `firewall-offline-cmd`. Alternatively, you can edit the XML files in `/etc/firewalld/services/`. If a service is not added or changed by the user, then no corresponding XML file are found in `/etc/firewalld/services/`. The files `/usr/lib/firewalld/services/` can be used as templates if you want to add or change a service.

**NOTE**

Service names must be alphanumeric and can, additionally, include only _ (underscore) and - (dash) characters.

**Procedure**
To add a new service in a terminal, use `firewall-cmd`, or `firewall-offline-cmd` in case of not active firewalld.
1. Enter the following command to add a new and empty service:

```bash
$ firewall-cmd --new-service=service-name --permanent
```

2. To add a new service using a local file, use the following command:

```bash
$ firewall-cmd --new-service-from-file=service-name.xml --permanent
```

   You can change the service name with the additional `--name=service-name` option.

3. As soon as service settings are changed, an updated copy of the service is placed into `/etc/firewalld/services/`. As root, you can enter the following command to copy a service manually:

```bash
# cp /usr/lib/firewalld/services/service-name.xml /etc/firewalld/services/service-name.xml
```

`firewalld` loads files from `/usr/lib/firewalld/services` in the first place. If files are placed in `/etc/firewalld/services` and they are valid, then these will override the matching files from `/usr/lib/firewalld/services`. The overridden files in `/usr/lib/firewalld/services` are used as soon as the matching files in `/etc/firewalld/services` have been removed or if `firewalld` has been asked to load the defaults of the services. This applies to the permanent environment only. A reload is needed to get these fallbacks also in the runtime environment.

### 26.5.3.5. Opening ports using GUI

To permit traffic through the firewall to a certain port, you can open the port in the GUI.

**Prerequisites**

- You installed the `firewall-config` package

**Procedure**

1. Start the `firewall-config` tool and select the network zone whose settings you want to change.

2. Select the **Ports** tab and click the **Add** button on the right-hand side. The **Port and Protocol** window opens.

3. Enter the port number or range of ports to permit.

4. Select **tcp** or **udp** from the list.

### 26.5.3.6. Controlling traffic with protocols using GUI

To permit traffic through the firewall using a certain protocol, you can use the GUI.

**Prerequisites**

- You installed the `firewall-config` package

**Procedure**

1. Start the `firewall-config` tool and select the network zone whose settings you want to change.
2. Select the **Protocols** tab and click the **Add** button on the right-hand side. The **Protocol** window opens.

3. Either select a protocol from the list or select the **Other Protocol** check box and enter the protocol in the field.

### 26.5.3.7. Opening source ports using GUI

To permit traffic through the firewall from a certain port, you can use the GUI.

**Prerequisites**

- You installed the *firewall-config* package

**Procedure**

1. Start the firewall-config tool and select the network zone whose settings you want to change.

2. Select the **Source Port** tab and click the **Add** button on the right-hand side. The **Source Port** window opens.

3. Enter the port number or range of ports to permit. Select **tcp** or **udp** from the list.

### 26.5.4. Controlling ports using CLI

Ports are logical devices that enable an operating system to receive and distinguish network traffic and forward it accordingly to system services. These are usually represented by a daemon that listens on the port, that is it waits for any traffic coming to this port.

Normally, system services listen on standard ports that are reserved for them. The *httpd* daemon, for example, listens on port 80. However, system administrators by default configure daemons to listen on different ports to enhance security or for other reasons.

#### 26.5.4.1. Opening a port

Through open ports, the system is accessible from the outside, which represents a security risk. Generally, keep ports closed and only open them if they are required for certain services.

**Procedure**

To get a list of open ports in the current zone:

1. List all allowed ports:

   ```
   # firewall-cmd --list-ports
   ```

2. Add a port to the allowed ports to open it for incoming traffic:

   ```
   # firewall-cmd --add-port=port-number/port-type
   ```

   The port types are either **tcp**, **udp**, **sctp**, or **dccp**. The type must match the type of network communication.

3. Make the new settings persistent:
# firewall-cmd --runtime-to-permanent

The port types are either tcp, udp, sctp, or dccp. The type must match the type of network communication.

26.5.4.2. Closing a port

When an open port is no longer needed, close that port in firewalld. It is highly recommended to close all unnecessary ports as soon as they are not used because leaving a port open represents a security risk.

Procedure

To close a port, remove it from the list of allowed ports:

1. List all allowed ports:

   ```
   # firewall-cmd --list-ports
   ```

   **WARNING**

   This command will only give you a list of ports that have been opened as ports. You will not be able to see any open ports that have been opened as a service. Therefore, you should consider using the `--list-all` option instead of `--list-ports`.

2. Remove the port from the allowed ports to close it for the incoming traffic:

   ```
   # firewall-cmd --remove-port=port-number/port-type
   ```

3. Make the new settings persistent:

   ```
   # firewall-cmd --runtime-to-permanent
   ```

26.5.5. Working with firewalld zones

Zones represent a concept to manage incoming traffic more transparently. The zones are connected to networking interfaces or assigned a range of source addresses. You manage firewall rules for each zone independently, which enables you to define complex firewall settings and apply them to the traffic.

26.5.5.1. Listing zones

This procedure describes how to list zones using the command line.

Procedure

1. To see which zones are available on your system:

   ```
   # firewall-cmd --get-zones
   ```
The `firewall-cmd --get-zones` command displays all zones that are available on the system, but it does not show any details for particular zones.

2. To see detailed information for all zones:
   ```
   # firewall-cmd --list-all-zones
   ```

3. To see detailed information for a specific zone:
   ```
   # firewall-cmd --zone=zone-name --list-all
   ```

### 26.5.5.2. Modifying firewalld settings for a certain zone

The Controlling traffic with predefined services using cli and Controlling ports using cli explain how to add services or modify ports in the scope of the current working zone. Sometimes, it is required to set up rules in a different zone.

**Procedure**

- To work in a different zone, use the `--zone=zone-name` option. For example, to allow the `SSH` service in the zone `public`:
  ```
  # firewall-cmd --add-service=ssh --zone=public
  ```

### 26.5.5.3. Changing the default zone

System administrators assign a zone to a networking interface in its configuration files. If an interface is not assigned to a specific zone, it is assigned to the default zone. After each restart of the `firewalld` service, `firewalld` loads the settings for the default zone and makes it active.

**Procedure**

To set up the default zone:

1. Display the current default zone:
   ```
   # firewall-cmd --get-default-zone
   ```

2. Set the new default zone:
   ```
   # firewall-cmd --set-default-zone zone-name
   ```

**NOTE**

Following this procedure, the setting is a permanent setting, even without the `--permanent` option.

### 26.5.5.4. Assigning a network interface to a zone

It is possible to define different sets of rules for different zones and then change the settings quickly by changing the zone for the interface that is being used. With multiple interfaces, a specific zone can be set for each of them to distinguish traffic that is coming through them.
Procedure

To assign the zone to a specific interface:

1. List the active zones and the interfaces assigned to them:

   ```
   # firewall-cmd --get-active-zones
   ```

2. Assign the interface to a different zone:

   ```
   # firewall-cmd --zone=zone_name --change-interface=interface_name --permanent
   ```

26.5.5.5. Assigning a zone to a connection using nmcli

This procedure describes how to add a `firewalld` zone to a `NetworkManager` connection using the `nmcli` utility.

Procedure

1. Assign the zone to the `NetworkManager` connection profile:

   ```
   # nmcli connection modify profile connection.zone zone_name
   ```

2. Activate the connection:

   ```
   # nmcli connection up profile
   ```

26.5.5.6. Manually assigning a zone to a network connection in an ifcfg file

When the connection is managed by `NetworkManager`, it must be aware of a zone that it uses. For every network connection, a zone can be specified, which provides the flexibility of various firewall settings according to the location of the computer with portable devices. Thus, zones and settings can be specified for different locations, such as company or home.

Procedure

- To set a zone for a connection, edit the `/etc/sysconfig/network-scripts/ifcfg-connection_name` file and add a line that assigns a zone to this connection:

  ```
  ZONE=zone_name
  ```

26.5.5.7. Creating a new zone

To use custom zones, create a new zone and use it just like a predefined zone. New zones require the `--permanent` option, otherwise the command does not work.

Procedure

1. Create a new zone:

   ```
   # firewall-cmd --permanent --new-zone=zone-name
   ```
2. Check if the new zone is added to your permanent settings:

```bash
# firewall-cmd --get-zones
```

3. Make the new settings persistent:

```bash
# firewall-cmd --runtime-to-permanent
```

### 26.5.5.8. Zone configuration files

Zones can also be created using a zone configuration file. This approach can be helpful when you need to create a new zone, but want to reuse the settings from a different zone and only alter them a little.

A `firewalld` zone configuration file contains the information for a zone. These are the zone description, services, ports, protocols, icmp-blocks, masquerade, forward-ports and rich language rules in an XML file format. The file name has to be `zone-name.xml` where the length of `zone-name` is currently limited to 17 chars. The zone configuration files are located in the `/usr/lib/firewalld/zones/` and `/etc/firewalld/zones/` directories.

The following example shows a configuration that allows one service (SSH) and one port range, for both the TCP and UDP protocols:

```xml
<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>My Zone</short>
  <description>Here you can describe the characteristic features of the zone.</description>
  <service name="ssh"/>
  <port protocol="udp" port="1025-65535"/>
  <port protocol="tcp" port="1025-65535"/>
</zone>
```

To change settings for that zone, add or remove sections to add ports, forward ports, services, and so on.

**Additional resources**

- `firewalld.zone` manual page

### 26.5.5.9. Using zone targets to set default behavior for incoming traffic

For every zone, you can set a default behavior that handles incoming traffic that is not further specified. Such behavior is defined by setting the target of the zone. There are four options:

- **ACCEPT**: Accepts all incoming packets except those disallowed by specific rules.

- **REJECT**: Rejects all incoming packets except those allowed by specific rules. When `firewalld` rejects packets, the source machine is informed about the rejection.

- **DROP**: Drops all incoming packets except those allowed by specific rules. When `firewalld` drops packets, the source machine is not informed about the packet drop.

- **default**: Similar behavior as for REJECT, but with special meanings in certain scenarios. For details, see the Options to Adapt and Query Zones and Policies section in the `firewall-cmd(1)` man page.
Procedure

To set a target for a zone:

1. List the information for the specific zone to see the default target:

   # firewall-cmd --zone=zone-name --list-all

2. Set a new target in the zone:

   # firewall-cmd --permanent --zone=zone-name --set-target=<default|ACCEPT|REJECT|DROP>

Additional resources

- firewall-cmd(1) man page

26.5.6. Using zones to manage incoming traffic depending on a source

You can use zones to manage incoming traffic based on its source. That enables you to sort incoming traffic and route it through different zones to allow or disallow services that can be reached by that traffic.

If you add a source to a zone, the zone becomes active and any incoming traffic from that source will be directed through it. You can specify different settings for each zone, which is applied to the traffic from the given sources accordingly. You can use more zones even if you only have one network interface.

26.5.6.1. Adding a source

To route incoming traffic into a specific zone, add the source to that zone. The source can be an IP address or an IP mask in the classless inter-domain routing (CIDR) notation.

**NOTE**

In case you add multiple zones with an overlapping network range, they are ordered alphanumerically by zone name and only the first one is considered.

- To set the source in the current zone:

  # firewall-cmd --add-source=<source>

- To set the source IP address for a specific zone:

  # firewall-cmd --zone=zone-name --add-source=<source>

The following procedure allows all incoming traffic from 192.168.2.15 in the trusted zone:

Procedure

1. List all available zones:

   # firewall-cmd --get-zones
2. Add the source IP to the trusted zone in the permanent mode:

```bash
# firewall-cmd --zone=trusted --add-source=192.168.2.15
```

3. Make the new settings persistent:

```bash
# firewall-cmd --runtime-to-permanent
```

### 26.5.6.2. Removing a source

Removing a source from the zone cuts off the traffic coming from it.

**Procedure**

1. List allowed sources for the required zone:

```bash
# firewall-cmd --zone=zone-name --list-sources
```

2. Remove the source from the zone permanently:

```bash
# firewall-cmd --zone=zone-name --remove-source=<source>
```

3. Make the new settings persistent:

```bash
# firewall-cmd --runtime-to-permanent
```

### 26.5.6.3. Adding a source port

To enable sorting the traffic based on a port of origin, specify a source port using the

```
--add-source-port
```

option. You can also combine this with the

```
--add-source
```

option to limit the traffic to a certain IP address or IP range.

**Procedure**

- To add a source port:

```bash
# firewall-cmd --zone=zone-name --add-source-port=<port-name>/<tcp|udp|sctp|dccp>
```

### 26.5.6.4. Removing a source port

By removing a source port you disable sorting the traffic based on a port of origin.

**Procedure**

- To remove a source port:

```bash
# firewall-cmd --zone=zone-name --remove-source-port=<port-name>/<tcp|udp|sctp|dccp>
```

### 26.5.6.5. Using zones and sources to allow a service for only a specific domain
To allow traffic from a specific network to use a service on a machine, use zones and source. The following procedure allows only HTTP traffic from the 192.0.2.0/24 network while any other traffic is blocked.

**WARNING**

When you configure this scenario, use a zone that has the default target. Using a zone that has the target set to ACCEPT is a security risk, because for traffic from 192.0.2.0/24, all network connections would be accepted.

**Procedure**

1. List all available zones:

   ```
   # firewall-cmd --get-zones
   block dmz drop external home internal public trusted work
   ```

2. Add the IP range to the internal zone to route the traffic originating from the source through the zone:

   ```
   # firewall-cmd --zone=internal --add-source=192.0.2.0/24
   ```

3. Add the http service to the internal zone:

   ```
   # firewall-cmd --zone=internal --add-service=http
   ```

4. Make the new settings persistent:

   ```
   # firewall-cmd --runtime-to-permanent
   ```

**Verification**

- Check that the internal zone is active and that the service is allowed in it:

   ```
   # firewall-cmd --zone=internal --list-all
   internal (active)
   target: default
   icmp-block-inversion: no
   interfaces:
   sources: 192.0.2.0/24
   services: cockpit dhcpcv6-client mdns samba-client ssh http ...
   ```

**Additional resources**

- `firewalld.zones(5)` man page

26.5.7. Filtering forwarded traffic between zones
With a policy object, users can group different identities that require similar permissions in the policy. You can apply policies depending on the direction of the traffic.

The policy objects feature provides forward and output filtering in firewalld. The following describes the usage of firewalld to filter traffic between different zones to allow access to locally hosted VMs to connect the host.

26.5.7.1. The relationship between policy objects and zones

Policy objects allow the user to attach firewalld’s primitives’ such as services, ports, and rich rules to the policy. You can apply the policy objects to traffic that passes between zones in a stateful and unidirectional manner.

```
# firewall-cmd --permanent --new-policy myOutputPolicy
# firewall-cmd --permanent --policy myOutputPolicy --add-ingress-zone HOST
# firewall-cmd --permanent --policy myOutputPolicy --add-egress-zone ANY
```

HOST and ANY are the symbolic zones used in the ingress and egress zone lists.

- The HOST symbolic zone allows policies for the traffic originating from or has a destination to the host running firewalld.
- The ANY symbolic zone applies policy to all the current and future zones. ANY symbolic zone acts as a wildcard for all zones.

26.5.7.2. Using priorities to sort policies

Multiple policies can apply to the same set of traffic, therefore, priorities should be used to create an order of precedence for the policies that may be applied.

To set a priority to sort the policies:

```
# firewall-cmd --permanent --policy mypolicy --set-priority -500
```

In the above example -500 is a lower priority value but has higher precedence. Thus, -500 will execute before -100. Higher priority values have precedence over lower values.

The following rules apply to policy priorities:

- Policies with negative priorities apply before rules in zones.
- Policies with positive priorities apply after rules in zones.
- Priority 0 is reserved and hence is unusable.

26.5.7.3. Using policy objects to filter traffic between locally hosted Containers and a network physically connected to the host

The policy objects feature allows users to filter their container and virtual machine traffic.

Procedure

1. Create a new policy.
# firewall-cmd --permanent --new-policy podmanToHost

2. Block all traffic.

```bash
# firewall-cmd --permanent --policy podmanToHost --set-target REJECT
# firewall-cmd --permanent --policy podmanToHost --add-service dhcp
# firewall-cmd --permanent --policy podmanToHost --add-service dns
```

**NOTE**

Red Hat recommends that you block all traffic to the host by default and then selectively open the services you need for the host.

3. Define the ingress zone to use with the policy.

```bash
# firewall-cmd --permanent --policy podmanToHost --add-ingress-zone podman
```

4. Define the egress zone to use with the policy.

```bash
# firewall-cmd --permanent --policy podmanToHost --add-egress-zone ANY
```

**Verification**

- Verify information about the policy.

```bash
# firewall-cmd --info-policy podmanToHost
```

26.5.7.4. **Setting the default target of policy objects**

You can specify --set-target options for policies. The following targets are available:

- **ACCEPT** - accepts the packet
- **DROP** - drops the unwanted packets
- **REJECT** - rejects unwanted packets with an ICMP reply
- **CONTINUE (default)** - packets will be subject to rules in following policies and zones.

```bash
# firewall-cmd --permanent --policy mypolicy --set-target CONTINUE
```

**Verification**

- Verify information about the policy

```bash
# firewall-cmd --info-policy mypolicy
```

26.5.8. **Configuring NAT using firewalld**
With `firewalld`, you can configure the following network address translation (NAT) types:

- Masquerading
- Source NAT (SNAT)
- Destination NAT (DNAT)
- Redirect

### 26.5.8.1. The different NAT types: masquerading, source NAT, destination NAT, and redirect

These are the different network address translation (NAT) types:

**Masquerading and source NAT (SNAT)**

Use one of these NAT types to change the source IP address of packets. For example, Internet Service Providers do not route private IP ranges, such as \texttt{10.0.0.0/8}. If you use private IP ranges in your network and users should be able to reach servers on the Internet, map the source IP address of packets from these ranges to a public IP address.

Both masquerading and SNAT are very similar. The differences are:

- Masquerading automatically uses the IP address of the outgoing interface. Therefore, use masquerading if the outgoing interface uses a dynamic IP address.
- SNAT sets the source IP address of packets to a specified IP and does not dynamically look up the IP of the outgoing interface. Therefore, SNAT is faster than masquerading. Use SNAT if the outgoing interface uses a fixed IP address.

**Destination NAT (DNAT)**

Use this NAT type to rewrite the destination address and port of incoming packets. For example, if your web server uses an IP address from a private IP range and is, therefore, not directly accessible from the Internet, you can set a DNAT rule on the router to redirect incoming traffic to this server.

**Redirect**

This type is a special case of DNAT that redirects packets to the local machine depending on the chain hook. For example, if a service runs on a different port than its standard port, you can redirect incoming traffic from the standard port to this specific port.

### 26.5.8.2. Configuring IP address masquerading

The following procedure describes how to enable IP masquerading on your system. IP masquerading hides individual machines behind a gateway when accessing the Internet.

**Procedure**

1. To check if IP masquerading is enabled (for example, for the \texttt{external} zone), enter the following command as \texttt{root}:

   ```
   # firewall-cmd --zone=external --query-masquerade
   ```

   The command prints \texttt{yes} with exit status \texttt{0} if enabled. It prints \texttt{no} with exit status \texttt{1} otherwise. If \texttt{zone} is omitted, the default zone will be used.

2. To enable IP masquerading, enter the following command as \texttt{root}: 

   ```
   # firewall-cmd --zone=external --add-port=80/tcp
   ```
# firewall-cmd --zone=external --add-masquerade

3. To make this setting persistent, pass the **--permanent** option to the command.

4. To disable IP masquerading, enter the following command as **root**:

```
# firewall-cmd --zone=external --remove-masquerade
```

To make this setting permanent, pass the **--permanent** option to the command.

### 26.5.9. Port forwarding

Redirecting ports using this method only works for IPv4-based traffic. For IPv6 redirecting setup, you must use rich rules.

To redirect to an external system, it is necessary to enable masquerading. For more information, see Configuring IP address masquerading.

#### 26.5.9.1. Adding a port to redirect

Using **firewalld**, you can set up port redirection so that any incoming traffic that reaches a certain port on your system is delivered to another internal port of your choice or to an external port on another machine.

**Prerequisites**

- Before you redirect traffic from one port to another port, or another address, you have to know three things: which port the packets arrive at, what protocol is used, and where you want to redirect them.

**Procedure**

1. To redirect a port to another port:

```
# firewall-cmd --add-forward-port=port=port-number:proto=tcp|udp|sctp|dccp:toport=port-number
```

2. To redirect a port to another port at a different IP address:

   a. Add the port to be forwarded:

```
# firewall-cmd --add-forward-port=port=port-number:proto=tcp|udp:toport=port-number:toaddr=IP
```

   b. Enable masquerade:

```
# firewall-cmd --add-masquerade
```

#### 26.5.9.2. Redirecting TCP port 80 to port 88 on the same machine

Follow the steps to redirect the TCP port 80 to port 88.
Procedure

1. Redirect the port 80 to port 88 for TCP traffic:

   ```
   # firewall-cmd --add-forward-port=port=80:proto=tcp:toport=88
   ```

2. Make the new settings persistent:

   ```
   # firewall-cmd --runtime-to-permanent
   ```

3. Check that the port is redirected:

   ```
   # firewall-cmd --list-all
   ```

26.5.9.3. Removing a redirected port

This procedure describes how to remove the redirected port.

Procedure

1. To remove a redirected port:

   ```
   # firewall-cmd --remove-forward-port=port=port-number:proto=<tcp|udp>:toport=port-number:toaddr=<IP>
   ```

2. To remove a forwarded port redirected to a different address:
   a. Remove the forwarded port:

      ```
      # firewall-cmd --remove-forward-port=port=port-number:proto=<tcp|udp>:toport=port-number:toaddr=<IP>
      ```

   b. Disable masquerade:

      ```
      # firewall-cmd --remove-masquerade
      ```

26.5.9.4. Removing TCP port 80 forwarded to port 88 on the same machine

This procedure describes how to remove the port redirection.

Procedure

1. List redirected ports:

   ```
   ~]# firewall-cmd --list-forward-ports
   port=80:proto=tcp:toport=88:toaddr=
   ```

2. Remove the redirected port from the firewall:

   ```
   ~]# firewall-cmd --remove-forward-port=port=80:proto=tcp:toport=88:toaddr=
   ```

3. Make the new settings persistent:
26.5.10. Managing ICMP requests

The Internet Control Message Protocol (ICMP) is a supporting protocol that is used by various network devices to send error messages and operational information indicating a connection problem, for example, that a requested service is not available. ICMP differs from transport protocols such as TCP and UDP because it is not used to exchange data between systems.

Unfortunately, it is possible to use the ICMP messages, especially echo-request and echo-reply, to reveal information about your network and misuse such information for various kinds of fraudulent activities. Therefore, firewalld enables blocking the ICMP requests to protect your network information.

26.5.10.1. Listing and blocking ICMP requests

Listing ICMP requests

The ICMP requests are described in individual XML files that are located in the /usr/lib/firewalld/icmptypes/ directory. You can read these files to see a description of the request. The firewall-cmd command controls the ICMP requests manipulation.

- To list all available ICMP types:
  ```
  # firewall-cmd --get-icmptypes
  ```
- The ICMP request can be used by IPv4, IPv6, or by both protocols. To see for which protocol the ICMP request has used:
  ```
  # firewall-cmd --info-icmptype=<icmptype>
  ```
- The status of an ICMP request shows yes if the request is currently blocked or no if it is not. To see if an ICMP request is currently blocked:
  ```
  # firewall-cmd --query-icmp-block=<icmptype>
  ```

Blocking or unblocking ICMP requests

When your server blocks ICMP requests, it does not provide the information that it normally would. However, that does not mean that no information is given at all. The clients receive information that the particular ICMP request is being blocked (rejected). Blocking the ICMP requests should be considered carefully, because it can cause communication problems, especially with IPv6 traffic.

- To see if an ICMP request is currently blocked:
  ```
  # firewall-cmd --query-icmp-block=<icmptype>
  ```
- To block an ICMP request:
  ```
  # firewall-cmd --add-icmp-block=<icmptype>
  ```
- To remove the block for an ICMP request:
  ```
  # firewall-cmd --remove-icmp-block=<icmptype>
  ```
Blocking ICMP requests without providing any information at all

Normally, if you block ICMP requests, clients know that you are blocking it. So, a potential attacker who is sniffing for live IP addresses is still able to see that your IP address is online. To hide this information completely, you have to drop all ICMP requests.

- To block and drop all ICMP requests:

  Set the target of your zone to **DROP**:

  ```
  # firewall-cmd --permanent --set-target=DROP
  ```

  Now, all traffic, including ICMP requests, is dropped, except traffic which you have explicitly allowed.

To block and drop certain ICMP requests and allow others:

1. Set the target of your zone to **DROP**:

   ```
   # firewall-cmd --permanent --set-target=DROP
   ```

2. Add the ICMP block inversion to block all ICMP requests at once:

   ```
   # firewall-cmd --add-icmp-block-inversion
   ```

3. Add the ICMP block for those ICMP requests that you want to allow:

   ```
   # firewall-cmd --add-icmp-block=<icmptype>
   ```

4. Make the new settings persistent:

   ```
   # firewall-cmd --runtime-to-permanent
   ```

The block inversion inverts the setting of the ICMP requests blocks, so all requests, that were not previously blocked, are blocked because of the target of your zone changes to **DROP**. The requests that were blocked are not blocked. This means that if you want to unblock a request, you must use the blocking command.

To revert the block inversion to a fully permissive setting:

1. Set the target of your zone to **default** or **ACCEPT**:

   ```
   # firewall-cmd --permanent --set-target=default
   ```

2. Remove all added blocks for ICMP requests:

   ```
   # firewall-cmd --remove-icmp-block=<icmptype>
   ```

3. Remove the ICMP block inversion:

   ```
   # firewall-cmd --remove-icmp-block-inversion
   ```

4. Make the new settings persistent:
# firewall-cmd --runtime-to-permanent

### 26.5.10.2. Configuring the ICMP filter using GUI

- To enable or disable an **ICMP** filter, start the **firewall-config** tool and select the network zone whose messages are to be filtered. Select the **ICMP Filter** tab and select the check box for each type of **ICMP** message you want to filter. Clear the check box to disable a filter. This setting is per direction and the default allows everything.

- To enable inverting the **ICMP Filter**, click the **Invert Filter** check box on the right. Only marked **ICMP** types are now accepted, all other are rejected. In a zone using the DROP target, they are dropped.

### 26.5.11. Setting and controlling IP sets using firewalld

To see the list of IP set types supported by **firewalld**, enter the following command as root.

```
~]# firewall-cmd --get-ipset-types
```

```
hash:net,net hash:net,port hash:net,net,port
```

### 26.5.11.1. Configuring IP set options using CLI

**IP sets** can be used in **firewalld** zones as sources and also as sources in rich rules. In Red Hat Enterprise Linux, the preferred method is to use the IP sets created with **firewalld** in a direct rule.

- To list the IP sets known to **firewalld** in the permanent environment, use the following command as **root**:

  ```
  # firewall-cmd --permanent --get-ipsets
  ```

- To add a new IP set, use the following command using the permanent environment as **root**:

  ```
  # firewall-cmd --permanent --new-ipset=test --type=hash:net
  success
  ```

  The previous command creates a new IP set with the name **test** and the **hash:net** type for **IPv4**. To create an IP set for use with **IPv6**, add the **--option=family=inet6** option. To make the new setting effective in the runtime environment, reload **firewalld**.

- List the new IP set with the following command as **root**:

  ```
  # firewall-cmd --permanent --get-ipsets
  test
  ```

- To get more information about the IP set, use the following command as **root**:

  ```
  # firewall-cmd --permanent --info-ipset=test
  test
  type: hash:net
  options:
  entries:
  ```
Note that the IP set does not have any entries at the moment.

- To add an entry to the test IP set, use the following command as root:

  # firewall-cmd --permanent --ipset=test --add-entry=192.168.0.1
  success

  The previous command adds the IP address 192.168.0.1 to the IP set.

- To get the list of current entries in the IP set, use the following command as root:

  # firewall-cmd --permanent --ipset=test --get-entries
  192.168.0.1

- Generate a file containing a list of IP addresses, for example:

  # cat > iplist.txt <<EOL
  192.168.0.2
  192.168.0.3
  192.168.1.0/24
  192.168.2.254
  EOL

  The file with the list of IP addresses for an IP set should contain an entry per line. Lines starting with a hash, a semi-colon, or empty lines are ignored.

- To add the addresses from the iplist.txt file, use the following command as root:

  # firewall-cmd --permanent --ipset=test --add-entries-from-file=iplist.txt
  success

- To see the extended entries list of the IP set, use the following command as root:

  # firewall-cmd --permanent --ipset=test --get-entries
  192.168.0.1
  192.168.0.2
  192.168.0.3
  192.168.1.0/24
  192.168.2.254

- To remove the addresses from the IP set and to check the updated entries list, use the following commands as root:

  # firewall-cmd --permanent --ipset=test --remove-entries-from-file=iplist.txt
  success
  # firewall-cmd --permanent --ipset=test --get-entries
  192.168.0.1

- You can add the IP set as a source to a zone to handle all traffic coming in from any of the addresses listed in the IP set with a zone. For example, to add the test IP set as a source to the drop zone to drop all packets coming from all entries listed in the test IP set, use the following command as root:
Only the creation and removal of IP sets is limited to the permanent environment, all other IP set options can be used also in the runtime environment without the `--permanent` option.

**WARNING**

Red Hat does not recommend using IP sets that are not managed through `firewalld`. To use such IP sets, a permanent direct rule is required to reference the set, and a custom service must be added to create these IP sets. This service needs to be started before `firewalld` starts, otherwise `firewalld` is not able to add the direct rules using these sets. You can add permanent direct rules with the `/etc/firewalld/direct.xml` file.

### 26.5.12. Prioritizing rich rules

By default, rich rules are organized based on their rule action. For example, `deny` rules have precedence over `allow` rules. The `priority` parameter in rich rules provides administrators fine-grained control over rich rules and their execution order.

#### 26.5.12.1. How the priority parameter organizes rules into different chains

You can set the `priority` parameter in a rich rule to any number between `-32768` and `32767`, and lower values have higher precedence.

The `firewalld` service organizes rules based on their priority value into different chains:

- Priority lower than 0: the rule is redirected into a chain with the `_pre` suffix.
- Priority higher than 0: the rule is redirected into a chain with the `_post` suffix.
- Priority equals 0: based on the action, the rule is redirected into a chain with the `_log`, `_deny`, or `_allow` the action.

Inside these sub-chains, `firewalld` sorts the rules based on their priority value.

#### 26.5.12.2. Setting the priority of a rich rule

The procedure describes an example of how to create a rich rule that uses the `priority` parameter to log all traffic that is not allowed or denied by other rules. You can use this rule to flag unexpected traffic.

**Procedure**

1. Add a rich rule with a very low precedence to log all traffic that has not been matched by other rules:
2. Optionally, display the nftables rule that the command in the previous step created:

```bash
# nft list chain inet firewalld filter_IN_public_post
table inet firewalld {
  chain filter_IN_public_post {
    log prefix "UNEXPECTED: " limit rate 5/minute
  }
}
```

26.5.13. Configuring firewall lockdown

Local applications or services are able to change the firewall configuration if they are running as root (for example, libvirt). With this feature, the administrator can lock the firewall configuration so that either no applications or only applications that are added to the lockdown allow list are able to request firewall changes. The lockdown settings default to disabled. If enabled, the user can be sure that there are no unwanted configuration changes made to the firewall by local applications or services.

26.5.13.1. Configuring lockdown using CLI

This procedure describes how to enable or disable lockdown using the command line.

- To query whether lockdown is enabled, use the following command as root:

  ```bash
  # firewall-cmd --query-lockdown
  ```

  The command prints yes with exit status 0 if lockdown is enabled. It prints no with exit status 1 otherwise.

- To enable lockdown, enter the following command as root:

  ```bash
  # firewall-cmd --lockdown-on
  ```

- To disable lockdown, use the following command as root:

  ```bash
  # firewall-cmd --lockdown-off
  ```

26.5.13.2. Configuring lockdown allowlist options using CLI

The lockdown allowlist can contain commands, security contexts, users and user IDs. If a command entry on the allowlist ends with an asterisk "+", then all command lines starting with that command will match. If the "*" is not there then the absolute command including arguments must match.

- The context is the security (SELinux) context of a running application or service. To get the context of a running application use the following command:

  ```bash
  $ ps -e --context
  ```
That command returns all running applications. Pipe the output through the `grep` tool to get the application of interest. For example:

```
$ ps -e --context | grep example_program
```

- To list all command lines that are in the allowlist, enter the following command as `root`:
  
  ```
  # firewall-cmd --list-lockdown-whitelist-commands
  ```

- To add a command `command` to the allowlist, enter the following command as `root`:
  
  ```
  # firewall-cmd --add-lockdown-whitelist-command='/usr/bin/python3 -Es /usr/bin/command'
  ```

- To remove a command `command` from the allowlist, enter the following command as `root`:
  
  ```
  # firewall-cmd --remove-lockdown-whitelist-command='/usr/bin/python3 -Es /usr/bin/command'
  ```

- To query whether the command `command` is in the allowlist, enter the following command as `root`:
  
  ```
  # firewall-cmd --query-lockdown-whitelist-command='/usr/bin/python3 -Es /usr/bin/command'
  ```

  The command prints `yes` with exit status `0` if true. It prints `no` with exit status `1` otherwise.

- To list all security contexts that are in the allowlist, enter the following command as `root`:
  
  ```
  # firewall-cmd --list-lockdown-whitelist-contexts
  ```

- To add a context `context` to the allowlist, enter the following command as `root`:
  
  ```
  # firewall-cmd --add-lockdown-whitelist-context=context
  ```

- To remove a context `context` from the allowlist, enter the following command as `root`:
  
  ```
  # firewall-cmd --remove-lockdown-whitelist-context=context
  ```

- To query whether the context `context` is in the allowlist, enter the following command as `root`:
  
  ```
  # firewall-cmd --query-lockdown-whitelist-context=context
  ```

  Prints `yes` with exit status `0`, if true, prints `no` with exit status `1` otherwise.

- To list all user IDs that are in the allowlist, enter the following command as `root`:
  
  ```
  # firewall-cmd --list-lockdown-whitelist-uids
  ```

- To add a user ID `uid` to the allowlist, enter the following command as `root`:
  
  ```
  # firewall-cmd --add-lockdown-whitelist-uid=uid
  ```

- To remove a user ID `uid` from the allowlist, enter the following command as `root`:
To query whether the user ID `uid` is in the allowlist, enter the following command:

```bash
$ firewall-cmd --query-lockdown-whitelist-uid=uid
```

Prints `yes` with exit status `0`, if true, prints `no` with exit status `1` otherwise.

To list all user names that are in the allowlist, enter the following command as `root`:

```bash
# firewall-cmd --list-lockdown-whitelist-users
```

To add a user name `user` to the allowlist, enter the following command as `root`:

```bash
# firewall-cmd --add-lockdown-whitelist-user=user
```

To remove a user name `user` from the allowlist, enter the following command as `root`:

```bash
# firewall-cmd --remove-lockdown-whitelist-user=user
```

To query whether the user name `user` is in the allowlist, enter the following command:

```bash
$ firewall-cmd --query-lockdown-whitelist-user=user
```

Prints `yes` with exit status `0`, if true, prints `no` with exit status `1` otherwise.

### 26.5.13.3. Configuring lockdown allowlist options using configuration files

The default allowlist configuration file contains the `NetworkManager` context and the default context of `libvirt`. The user ID 0 is also on the list.

+ The allowlist configuration files are stored in the `/etc/firewalld/` directory.

```xml
<?xml version="1.0" encoding="utf-8"?>
<whitelist>
  <selinux context="system_u:system_r:NetworkManager_t:s0"/>
  <selinux context="system_u:system_r:virtd_t:s0-s0:c0.c1023"/>
  <user id="0"/>
</whitelist>
```

Following is an example allowlist configuration file enabling all commands for the `firewall-cmd` utility, for a user called `user` whose user ID is `815`:

```xml
<?xml version="1.0" encoding="utf-8"?>
<whitelist>
  <command name="/usr/libexec/platform-python -s /bin/firewall-cmd**"/>
  <selinux context="system_u:system_r:NetworkManager_t:s0"/>
  <user id="815"/>
  <user name="user"/>
</whitelist>
```

This example shows both `user id` and `user name`, but only one option is required. Python is the interpreter and is prepended to the command line. You can also use a specific command, for example:
/usr/bin/python3 /bin/firewall-cmd --lockdown-on

In that example, only the --lockdown-on command is allowed.

In Red Hat Enterprise Linux, all utilities are placed in the /usr/bin/ directory and the /bin/ directory is sym-linked to the /usr/bin/ directory. In other words, although the path for firewall-cmd when entered as root might resolve to /bin/firewall-cmd, /usr/bin/firewall-cmd can now be used. All new scripts should use the new location. But be aware that if scripts that run as root are written to use the /bin/firewall-cmd path, then that command path must be added in the allowlist in addition to the /usr/bin/firewall-cmd path traditionally used only for non-root users.

The * at the end of the name attribute of a command means that all commands that start with this string match. If the * is not there then the absolute command including arguments must match.

### 26.5.14. Enabling traffic forwarding between different interfaces or sources within a firewalld zone

Intra-zone forwarding is a firewalld feature that enables traffic forwarding between interfaces or sources within a firewalld zone.

#### 26.5.14.1. The difference between intra-zone forwarding and zones with the default target set to ACCEPT

When intra-zone forwarding is enabled, the traffic within a single firewalld zone can flow from one interface or source to another interface or source. The zone specifies the trust level of interfaces and sources. If the trust level is the same, communication between interfaces or sources is possible.

Note that, if you enable intra-zone forwarding in the default zone of firewalld, it applies only to the interfaces and sources added to the current default zone.

The trusted zone of firewalld uses a default target set to ACCEPT. This zone accepts all forwarded traffic, and intra-zone forwarding is not applicable for it.

As for other default target values, forwarded traffic is dropped by default, which applies to all standard zones except the trusted zone.

#### 26.5.14.2. Using intra-zone forwarding to forward traffic between an Ethernet and Wi-Fi network

You can use intra-zone forwarding to forward traffic between interfaces and sources within the same firewalld zone. For example, use this feature to forward traffic between an Ethernet network connected to enp1s0 and a Wi-Fi network connected to wlp0s20.

**Procedure**

1. Enable packet forwarding in the kernel:

   ```
   # echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/95-IPv4-forwarding.conf
   # sysctl -p /etc/sysctl.d/95-IPv4-forwarding.conf
   ```

2. Ensure that interfaces between which you want to enable intra-zone forwarding are not assigned to a zone different than the internal zone:
# firewall-cmd --get-active-zones

3. If the interface is currently assigned to a zone other than `internal`, reassign it:

   ```bash
   # firewall-cmd --zone=internal --change-interface=interface_name --permanent
   ```

4. Add the `enp1s0` and `wlp0s20` interfaces to the `internal` zone:

   ```bash
   # firewall-cmd --zone=internal --add-interface=enp1s0 --add-interface=wlp0s20
   ```

5. Enable intra-zone forwarding:

   ```bash
   # firewall-cmd --zone=internal --add-forward
   ```

**Verification**

The following verification steps require that the `nmap-ncat` package is installed on both hosts.

1. Log in to a host that is in the same network as the `enp1s0` interface of the host you enabled zone forwarding on.

2. Start an echo service with `ncat` to test connectivity:

   ```bash
   # ncat -e /usr/bin/cat -l 12345
   ```

3. Log in to a host that is in the same network as the `wlp0s20` interface.

4. Connect to the echo server running on the host that is in the same network as the `enp1s0`:

   ```bash
   # ncat <other host> 12345
   ```

5. Type something and press **Enter**, and verify the text is sent back.

**Additional resources**

- `firewalld.zones(5)` man page
- `firewalld(1)` man page
- `firewalld.conf(5)` man page
- `firewall-cmd(1)` man page
- `firewall-config(1)` man page
- `firewall-offline-cmd(1)` man page
- `firewalld.icmptype(5)` man page
- `firewalld.ipset(5)` man page
- `firewalld.service(5)` man page
26.6. GETTING STARTED WITH NFTABLES

The **nftables** framework provides packet classification facilities. The most notable features are:

- built-in lookup tables instead of linear processing
- a single framework for both the IPv4 and IPv6 protocols
- rules all applied atomically instead of fetching, updating, and storing a complete rule set
- support for debugging and tracing in the rule set (**nftrace**) and monitoring trace events (in the **nft** tool)
- more consistent and compact syntax, no protocol-specific extensions
- a Netlink API for third-party applications

The **nftables** framework uses tables to store chains. The chains contain individual rules for performing actions. The **libnftnl** library can be used for low-level interaction with **nftables** Netlink API over the **libmnl** library.

To display the effect of rule set changes, use the **nft list ruleset** command. Since these tools add tables, chains, rules, sets, and other objects to the **nftables** rule set, be aware that **nftables** rule-set operations, such as the **nft flush ruleset** command, might affect rule sets installed using the formerly separate legacy commands.

26.6.1. Migrating from iptables to nftables

If your firewall configuration still uses **iptables** rules, you can migrate your **iptables** rules to **nftables**.

26.6.1.1. When to use firewalld, nftables, or iptables

The following is a brief overview in which scenario you should use one of the following utilities:

- **firewalld**: Use the **firewalld** utility for simple firewall use cases. The utility is easy to use and covers the typical use cases for these scenarios.

- **nftables**: Use the **nftables** utility to set up complex and performance critical firewalls, such as for a whole network.

- **iptables**: The **iptables** utility on Red Hat Enterprise Linux uses the **nf_tables** kernel API instead of the legacy back end. The **nf_tables** API provides backward compatibility so that scripts that use **iptables** commands still work on Red Hat Enterprise Linux. For new firewall scripts, Red Hat recommends to use **nftables**.
IMPORTANT

To avoid that the different firewall services influence each other, run only one of them on a RHEL host, and disable the other services.

26.6.1.2. Converting iptables rules to nftables rules

Red Hat Enterprise Linux provides the `iptables-translate` and `ip6tables-translate` tools to convert existing `iptables` or `ip6tables` rules into the equivalent ones for `nftables`.

Note that some extensions lack translation support. If such an extension exists, the tool prints the untranslated rule prefixed with the `#` sign. For example:

```
# iptables-translate -A INPUT -j CHECKSUM --checksum-fill
nft # -A INPUT -j CHECKSUM --checksum-fill
```

Additionally, users can use the `iptables-restore-translate` and `ip6tables-restore-translate` tools to translate a dump of rules. Note that before that, users can use the `iptables-save` or `ip6tables-save` commands to print a dump of current rules. For example:

```
# iptables-save >/tmp/iptables.dump
# iptables-restore-translate -f /tmp/iptables.dump

# Translated by iptables-restore-translate v1.8.0 on Wed Oct 17 17:00:13 2018
add table ip nat
...
```

For more information and a list of possible options and values, enter the `iptables-translate --help` command.

26.6.1.3. Comparison of common iptables and nftables commands

The following is a comparison of common `iptables` and `nftables` commands:

- Listing all rules:

<table>
<thead>
<tr>
<th><code>iptables</code></th>
<th><code>nftables</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>iptables-save</code></td>
<td><code>nft list ruleset</code></td>
</tr>
</tbody>
</table>

- Listing a certain table and chain:

<table>
<thead>
<tr>
<th><code>iptables</code></th>
<th><code>nftables</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>iptables -L</code></td>
<td><code>nft list table ip filter</code></td>
</tr>
<tr>
<td><code>iptables -L INPUT</code></td>
<td><code>nft list chain ip filter INPUT</code></td>
</tr>
<tr>
<td><code>iptables -t nat -L PREROUTING</code></td>
<td><code>nft list chain ip nat PREROUTING</code></td>
</tr>
</tbody>
</table>

The `nft` command does not pre-create tables and chains. They exist only if a user created them manually.
Example: Listing rules generated by firewalld

```
# nft list table inet firewalld
# nft list table ip firewalld
# nft list table ip6 firewalld
```

### 26.6.2. Writing and executing nftables scripts

The **nftables** framework provides a native scripting environment that brings a major benefit over using shell scripts to maintain firewall rules: the execution of scripts is atomic. This means that the system either applies the whole script or prevents the execution if an error occurs. This guarantees that the firewall is always in a consistent state.

Additionally, the **nftables** script environment enables administrators to:

- add comments
- define variables
- include other rule set files

This section explains how to use these features, as well as creating and executing **nftables** scripts.

When you install the **nftables** package, Red Hat Enterprise Linux automatically creates *.nft* scripts in the `/etc/nftables/` directory. These scripts contain commands that create tables and empty chains for different purposes.

#### 26.6.2.1. Supported nftables script formats

The **nftables** scripting environment supports scripts in the following formats:

- You can write a script in the same format as the **nft list ruleset** command displays the rule set:

  ```
  #!/usr/sbin/nft -f
  # Flush the rule set
  flush ruleset

  table inet example_table {
    chain example_chain {
      # Chain for incoming packets that drops all packets that
      # are not explicitly allowed by any rule in this chain
      type filter hook input priority 0; policy drop;

      # Accept connections to port 22 (ssh)
      tcp dport ssh accept
    }
  }
  ```

- You can use the same syntax for commands as in **nft** commands:

  ```
  #!/usr/sbin/nft -f
  # Flush the rule set
  flush ruleset
  ```
# Create a table
add table inet example_table

# Create a chain for incoming packets that drops all packets that are not explicitly allowed by any rule in this chain
add chain inet example_table example_chain { type filter hook input priority 0 ; policy drop ; }

# Add a rule that accepts connections to port 22 (ssh)
add rule inet example_table example_chain tcp dport ssh accept

26.6.2.2. Running nftables scripts

You can run nftables script either by passing it to the nft utility or execute the script directly.

Prerequisites

- The procedure of this section assumes that you stored an nftables script in the /etc/nftables/example_firewall.nft file.

Procedure

- To run an nftables script by passing it to the nft utility, enter:

```
# nft -f /etc/nftables/example_firewall.nft
```

- To run an nftables script directly:
  a. Steps that are required only once:
     i. Ensure that the script starts with the following shebang sequence:

```
#!/usr/sbin/nft -f
```

**IMPORTANT**

If you omit the -f parameter, the nft utility does not read the script and displays: *Error: syntax error, unexpected newline, expecting string.*

   ii. Optional: Set the owner of the script to root:

```
# chown root /etc/nftables/example_firewall.nft
```

   iii. Make the script executable for the owner:

```
# chmod u+x /etc/nftables/example_firewall.nft
```

b. Run the script:

```
#/etc/nftables/example_firewall.nft
```

If no output is displayed, the system executed the script successfully.
IMPORTANT

Even if `nft` executes the script successfully, incorrectly placed rules, missing parameters, or other problems in the script can cause that the firewall behaves not as expected.

Additional resources

- `chown(1)` man page
- `chmod(1)` man page
- Automatically loading `nftables` rules when the system boots

26.6.2.3. Using comments in `nftables` scripts

The `nftables` scripting environment interprets everything to the right of a `#` character as a comment.

Example 26.1. Comments in an `nftables` script

Comments can start at the beginning of a line, as well as next to a command:

```
...  # Flush the rule set
flush ruleset

add table inet example_table  # Create a table
...
```

26.6.2.4. Using variables in an `nftables` script

To define a variable in an `nftables` script, use the `define` keyword. You can store single values and anonymous sets in a variable. For more complex scenarios, use sets or verdict maps.

Variables with a single value

The following example defines a variable named `INET_DEV` with the value `enp1s0`:

```
define INET_DEV = enp1s0
```

You can use the variable in the script by writing the `$` sign followed by the variable name:

```
...  add rule inet example_table example_chain iifname $INET_DEV tcp dport ssh accept ...
```

Variables that contain an anonymous set

The following example defines a variable that contains an anonymous set:

```
define DNS_SERVERS = { 192.0.2.1, 192.0.2.2 }
```

You can use the variable in the script by writing the `$` sign followed by the variable name:
add rule inet example_table example_chain ip daddr $DNS_SERVERS accept

NOTE
Note that curly braces have special semantics when you use them in a rule because they indicate that the variable represents a set.

Additional resources
- Using sets in nftables commands
- Using verdict maps in nftables commands

26.6.2.5. Including files in an nftables script
The nftables scripting environment enables administrators to include other scripts by using the `include` statement.

If you specify only a file name without an absolute or relative path, nftables includes files from the default search path, which is set to `/etc` on Red Hat Enterprise Linux.

Example 26.2. Including files from the default search directory
To include a file from the default search directory:
```
include "example.nft"
```

Example 26.3. Including all *.nft files from a directory
To include all files ending with `*.nft` that are stored in the `/etc/nftables/rulesets/` directory:
```
include "/etc/nftables/rulesets/*.nft"
```

Note that the `include` statement does not match files beginning with a dot.

Additional resources
- The Include files section in the nft(8) man page

26.6.2.6. Automatically loading nftables rules when the system boots
The nftables systemd service loads firewall scripts that are included in the `/etc/sysconfig/nftables.conf` file. This section explains how to load firewall rules when the system boots.

Prerequisites
- The nftables scripts are stored in the `/etc/nftables/` directory.
Procedure

1. Edit the `/etc/sysconfig/nftables.conf` file.
   - If you enhance `.nft` scripts created in `/etc/nftables/` when you installed the `nftables` package, uncomment the `include` statement for these scripts.
   - If you write scripts from scratch, add `include` statements to include these scripts. For example, to load the `/etc/nftables/example.nft` script when the `nftables` service starts, add:
     
     ```
     include "/etc/nftables/example.nft"
     ```

2. Optionally, start the `nftables` service to load the firewall rules without rebooting the system:
   
   ```
   # systemctl start nftables
   ```

3. Enable the `nftables` service.
   
   ```
   # systemctl enable nftables
   ```

Additional resources

- Supported nftables script formats

26.6.3. Creating and managing nftables tables, chains, and rules

This section explains how to display `nftables` rule sets, and how to manage them.

26.6.3.1. Standard chain priority values and textual names

When you create a chain, the `priority` you can either set an integer value or a standard name that specifies the order in which chains with the same `hook` value traverse.

The names and values are defined based on what priorities are used by `xtables` when registering their default chains.

**NOTE**

The `nft list chains` command displays textual priority values by default. You can view the numeric value by passing the `-y` option to the command.

Example 26.4. Using a textual value to set the priority

The following command creates a chain named `example_chain` in `example_table` using the standard priority value `50`:

```
# nft add chain inet example_table example_chain { type filter hook input priority 50 \; policy accept \; }
```

Because the priority is a standard value, you can alternatively use the textual value:

```
# nft add chain inet example_table example_chain { type filter hook input priority security \; policy accept \; }
```
Table 26.1. Standard priority names, family, and hook compatibility matrix

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Families</th>
<th>Hooks</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>-300</td>
<td>ip, ip6, inet</td>
<td>all</td>
</tr>
<tr>
<td>mangle</td>
<td>-150</td>
<td>ip, ip6, inet</td>
<td>all</td>
</tr>
<tr>
<td>dstnat</td>
<td>-100</td>
<td>ip, ip6, inet</td>
<td>prerouting</td>
</tr>
<tr>
<td>filter</td>
<td>0</td>
<td>ip, ip6, inet, arp, netdev</td>
<td>all</td>
</tr>
<tr>
<td>security</td>
<td>50</td>
<td>ip, ip6, inet</td>
<td>all</td>
</tr>
<tr>
<td>srcnat</td>
<td>100</td>
<td>ip, ip6, inet</td>
<td>postrouting</td>
</tr>
</tbody>
</table>

All families use the same values, but the bridge family uses following values:

Table 26.2. Standard priority names, and hook compatibility for the bridge family

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Hooks</th>
</tr>
</thead>
<tbody>
<tr>
<td>dstnat</td>
<td>-300</td>
<td>prerouting</td>
</tr>
<tr>
<td>filter</td>
<td>-200</td>
<td>all</td>
</tr>
<tr>
<td>out</td>
<td>100</td>
<td>output</td>
</tr>
<tr>
<td>srcnat</td>
<td>300</td>
<td>postrouting</td>
</tr>
</tbody>
</table>

Additional resources

- The Chains section in the nft(8) man page

26.6.3.2. Displaying the nftables rule set

The rule sets of nftables contain tables, chains, and rules. This section explains how to display the rule set.

Procedure

- To display the rule set, enter:

```
# nft list ruleset
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
tcp dport http accept
  }
} `
NOTE

By default, `nftables` does not pre-create tables. As a consequence, displaying the rule set on a host without any tables, the `nft list ruleset` command shows no output.

26.6.3.3. Creating an nftables table

A table in `nftables` is a name space that contains a collection of chains, rules, sets, and other objects. This section explains how to create a table.

Each table must have an address family defined. The address family of a table defines what address types the table processes. You can set one of the following address families when you create a table:

- **ip**: Matches only IPv4 packets. This is the default if you do not specify an address family.
- **ip6**: Matches only IPv6 packets.
- **inet**: Matches both IPv4 and IPv6 packets.
- **arp**: Matches IPv4 address resolution protocol (ARP) packets.
- **bridge**: Matches packets that traverse a bridge device.
- **netdev**: Matches packets from ingress.

Procedure

1. Use the `nft add table` command to create a new table. For example, to create a table named `example_table` that processes IPv4 and IPv6 packets:

   ```
   # nft add table inet example_table
   ```

2. Optionally, list all tables in the rule set:

   ```
   # nft list tables
   table inet example_table
   ```

Additional resources

- The [Address families](nft(8)) section in the `nft(8)` man page
- The [Tables](nft(8)) section in the `nft(8)` man page

26.6.3.4. Creating an nftables chain

Chains are containers for rules. The following two rule types exists:

- Base chain: You can use base chains as an entry point for packets from the networking stack.
Regular chain: You can use regular chains as a jump target and to better organize rules.

The procedure describes how to add a base chain to an existing table.

**Prerequisites**

- The table to which you want to add the new chain exists.

**Procedure**

1. Use the `nft add chain` command to create a new chain. For example, to create a chain named `example_chain` in `example_table`:

   ```
   # nft add chain inet example_table example_chain { type filter hook input priority 0; policy accept; }
   ```

   **IMPORTANT**

   To avoid that the shell interprets the semicolons as the end of the command, prepend the semicolons the \ escape character.

   This chain filters incoming packets. The `priority` parameter specifies the order in which `nftables` processes chains with the same hook value. A lower priority value has precedence over higher ones. The `policy` parameter sets the default action for rules in this chain. Note that if you are logged in to the server remotely and you set the default policy to `drop`, you are disconnected immediately if no other rule allows the remote access.

2. Optionally, display all chains:

   ```
   # nft list chains
   table inet example_table { chain example_chain { type filter hook input priority filter; policy accept; } }
   ```

**Additional resources**

- The [Address families](#) section in the `nft(8)` man page
- The [Chains](#) section in the `nft(8)` man page

### 26.6.3.5. Appending a rule to the end of an nftables chain

This section explains how to append a rule to the end of an existing `nftables` chain.

**Prerequisites**

- The chain to which you want to add the rule exists.

**Procedure**

...
1. To add a new rule, use the `nft add rule` command. For example, to add a rule to the `example_chain` in the `example_table` that allows TCP traffic on port 22:

   ```
   # nft add rule inet example_table example_chain tcp dport 22 accept
   ```

   Instead of the port number, you can alternatively specify the name of the service. In the example, you could use `ssh` instead of the port number `22`. Note that a service name is resolved to a port number based on its entry in the `/etc/services` file.

2. Optionally, display all chains and their rules in `example_table`:

   ```
   # nft list table inet example_table
   table inet example_table {
       chain example_chain {
           type filter hook input priority filter; policy accept;
           tcp dport ssh accept
       }
   }
   ```

Additional resources

- The **Address families** section in the `nft(8)` man page
- The **Rules** section in the `nft(8)` man page

### 26.6.3.6. Inserting a rule at the beginning of an nftables chain

This section explains how to insert a rule at the beginning of an existing `nftables` chain.

**Prerequisites**

- The chain to which you want to add the rule exists.

**Procedure**

1. To insert a new rule, use the `nft insert rule` command. For example, to insert a rule to the `example_chain` in the `example_table` that allows TCP traffic on port 22:

   ```
   # nft insert rule inet example_table example_chain tcp dport 22 accept
   ```

   You can alternatively specify the name of the service instead of the port number. In the example, you could use `ssh` instead of the port number `22`. Note that a service name is resolved to a port number based on its entry in the `/etc/services` file.

2. Optionally, display all chains and their rules in `example_table`:

   ```
   # nft list table inet example_table
   table inet example_table {
       chain example_chain {
           type filter hook input priority filter; policy accept;
           tcp dport ssh accept
       }
   }
   ```
Additional resources

- The **Address families** section in the **nft(8)** man page
- The **Rules** section in the **nft(8)** man page

### 26.6.3.7. Inserting a rule at a specific position of an nftables chain

This section explains how to insert rules before and after an existing rule in an **nftables** chain. This way you can place new rules at the right position.

**Prerequisites**

- The chain to which you want to add the rules exists.

**Procedure**

1. Use the **nft -a list ruleset** command to display all chains and their rules in the **example_table** including their handle:

```bash
# nft -a list table inet example_table
table inet example_table { # handle 1
  chain example_chain { # handle 1
    type filter hook input priority filter; policy accept;
    tcp dport 22 accept # handle 2
    tcp dport 443 accept # handle 3
    tcp dport 389 accept # handle 4
  }
}
```

Using the **-a** displays the handles. You require this information to position the new rules in the next steps.

2. Insert the new rules to the **example_chain** chain in the **example_table**:

   - To insert a rule that allows TCP traffic on port **636** before handle **3**, enter:
     ```bash
     # nft insert rule inet example_table example_chain position 3 tcp dport 636 accept
     ```

   - To add a rule that allows TCP traffic on port **80** after handle **3**, enter:
     ```bash
     # nft add rule inet example_table example_chain position 3 tcp dport 80 accept
     ```

3. Optionally, display all chains and their rules in **example_table**:

```bash
# nft -a list table inet example_table
table inet example_table { # handle 1
  chain example_chain { # handle 1
    type filter hook input priority filter; policy accept;
    tcp dport 22 accept # handle 2
  }
}
```
Additional resources

- The Address families section in the nft(8) man page
- The Rules section in the nft(8) man page

26.6.4. Configuring NAT using nftables

With nftables, you can configure the following network address translation (NAT) types:

- Masquerading
- Source NAT (SNAT)
- Destination NAT (DNAT)
- Redirect

IMPORTANT

You can only use real interface names in iifname and oifname parameters, and alternative names (altname) are not supported.

26.6.4.1. The different NAT types: masquerading, source NAT, destination NAT, and redirect

These are the different network address translation (NAT) types:

Masquerading and source NAT (SNAT)

Use one of these NAT types to change the source IP address of packets. For example, Internet Service Providers do not route private IP ranges, such as 10.0.0.0/8. If you use private IP ranges in your network and users should be able to reach servers on the Internet, map the source IP address of packets from these ranges to a public IP address.

Both masquerading and SNAT are very similar. The differences are:

- Masquerading automatically uses the IP address of the outgoing interface. Therefore, use masquerading if the outgoing interface uses a dynamic IP address.

- SNAT sets the source IP address of packets to a specified IP and does not dynamically look up the IP of the outgoing interface. Therefore, SNAT is faster than masquerading. Use SNAT if the outgoing interface uses a fixed IP address.

Destination NAT (DNAT)

Use this NAT type to rewrite the destination address and port of incoming packets. For example, if your web server uses an IP address from a private IP range and is, therefore, not directly accessible from the Internet, you can set a DNAT rule on the router to redirect incoming traffic to this server.

Redirect
This type is a special case of DNAT that redirects packets to the local machine depending on the chain hook. For example, if a service runs on a different port than its standard port, you can redirect incoming traffic from the standard port to this specific port.

### 26.6.4.2. Configuring masquerading using nftables

Masquerading enables a router to dynamically change the source IP of packets sent through an interface to the IP address of the interface. This means that if the interface gets a new IP assigned, nftables automatically uses the new IP when replacing the source IP.

The following procedure describes how to replace the source IP of packets leaving the host through the ens3 interface to the IP set on ens3.

**Procedure**

1. Create a table:
   ```
   # nft add table nat
   ```

2. Add the **prerouting** and **postrouting** chains to the table:
   ```
   # nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }
   # nft add chain nat postrouting { type nat hook postrouting priority 100 \; }
   ```

   **IMPORTANT**
   
   Even if you do not add a rule to the **prerouting** chain, the nftables framework requires this chain to match incoming packet replies.

   Note that you must pass the -- option to the nft command to avoid that the shell interprets the negative priority value as an option of the nft command.

3. Add a rule to the **postrouting** chain that matches outgoing packets on the ens3 interface:
   ```
   # nft add rule nat postrouting oifname "ens3" masquerade
   ```

### 26.6.4.3. Configuring source NAT using nftables

On a router, Source NAT (SNAT) enables you to change the IP of packets sent through an interface to a specific IP address.

The following procedure describes how to replace the source IP of packets leaving the router through the ens3 interface to 192.0.2.1.

**Procedure**

1. Create a table:
   ```
   # nft add table nat
   ```

2. Add the **prerouting** and **postrouting** chains to the table:
# nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }  
# nft add chain nat postrouting { type nat hook postrouting priority 100 \; }

**IMPORTANT**

Even if you do not add a rule to the **postrouting** chain, the **nftables** framework requires this chain to match outgoing packet replies.

Note that you must pass the `--` option to the **nft** command to avoid that the shell interprets the negative priority value as an option of the **nft** command.

3. Add a rule to the **postrouting** chain that replaces the source IP of outgoing packets through **ens3** with **192.0.2.1**:

```bash
# nft add rule nat postrouting ofname "ens3" snat to 192.0.2.1
```

Additional resources

- Forwarding incoming packets on a specific local port to a different host

### 26.6.4.4. Configuring destination NAT using nftables

Destination NAT enables you to redirect traffic on a router to a host that is not directly accessible from the Internet.

The following procedure describes how to redirect incoming traffic sent to port **80** and **443** of the router to the host with the **192.0.2.1** IP address.

**Procedure**

1. Create a table:

   ```bash
   # nft add table nat
   ```

2. Add the **prerouting** and **postrouting** chains to the table:

   ```bash
   # nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }  
   # nft add chain nat postrouting { type nat hook postrouting priority 100 \; }
   ```

**IMPORTANT**

Even if you do not add a rule to the **postrouting** chain, the **nftables** framework requires this chain to match outgoing packet replies.

Note that you must pass the `--` option to the **nft** command to avoid that the shell interprets the negative priority value as an option of the **nft** command.

3. Add a rule to the **prerouting** chain that redirects incoming traffic on the **ens3** interface sent to port **80** and **443** to the host with the **192.0.2.1** IP:

   ```bash
   # nft add rule nat prerouting ifname ens3 tcp dport { 80, 443 } dnat to 192.0.2.1
   ```
4. Depending on your environment, add either a SNAT or masquerading rule to change the source address:

   a. If the `ens3` interface used dynamic IP addresses, add a masquerading rule:

   ```
   # nft add rule nat postrouting ofname "ens3" masquerade
   ```

   b. If the `ens3` interface uses a static IP address, add a SNAT rule. For example, if the `ens3` uses the `198.51.100.1` IP address:

   ```
   # nft add rule nat postrouting ofname "ens3" snat to 198.51.100.1
   ```

Additional resources

- The different NAT types: masquerading, source NAT, destination NAT, and redirect

26.6.4.5. Configuring a redirect using nftables

The `redirect` feature is a special case of destination network address translation (DNAT) that redirects packets to the local machine depending on the chain hook.

The following procedure describes how to redirect incoming and forwarded traffic sent to port 22 of the local host to port 2222.

Procedure

1. Create a table:

   ```
   # nft add table nat
   ```

2. Add the `prerouting` chain to the table:

   ```
   # nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }
   ```

   Note that you must pass the `--` option to the `nft` command to avoid that the shell interprets the negative priority value as an option of the `nft` command.

3. Add a rule to the `prerouting` chain that redirects incoming traffic on port 22 to port 2222:

   ```
   # nft add rule nat prerouting tcp dport 22 redirect to 2222
   ```

Additional resources

- The different NAT types: masquerading, source NAT, destination NAT, and redirect

26.6.5. Using sets in nftables commands

The `nftables` framework natively supports sets. You can use sets, for example, if a rule should match multiple IP addresses, port numbers, interfaces, or any other match criteria.

26.6.5.1. Using anonymous sets in nftables

- 

---

CHAPTER 26. SECURING NETWORKS
An anonymous set contain comma-separated values enclosed in curly brackets, such as \{22, 80, 443\}, that you use directly in a rule. You can also use anonymous sets also for IP addresses or any other match criteria.

The drawback of anonymous sets is that if you want to change the set, you must replace the rule. For a dynamic solution, use named sets as described in Using named sets in nftables.

**Prerequisites**
- The `example_chain` chain and the `example_table` table in the `inet` family exists.

**Procedure**
1. For example, to add a rule to `example_chain` in `example_table` that allows incoming traffic to port 22, 80, and 443:
   ```bash
   # nft add rule inet example_table example_chain tcp dport \{22, 80, 443\} accept
   ```
2. Optionally, display all chains and their rules in `example_table`:
   ```bash
   # nft list table inet example_table
   table inet example_table {
     chain example_chain {
       type filter hook input priority filter; policy accept;
       tcp dport \{ssh, http, https\} accept
     }
   }
   ```

**26.6.5.2. Using named sets in nftables**

The nftables framework supports mutable named sets. A named set is a list or range of elements that you can use in multiple rules within a table. Another benefit over anonymous sets is that you can update a named set without replacing the rules that use the set.

When you create a named set, you must specify the type of elements the set contains. You can set the following types:

- `ipv4_addr` for a set that contains IPv4 addresses or ranges, such as `192.0.2.1` or `192.0.2.0/24`.
- `ipv6_addr` for a set that contains IPv6 addresses or ranges, such as `2001:db8:1::1` or `2001:db8:1::1/64`.
- `ether_addr` for a set that contains a list of media access control (MAC) addresses, such as `52:54:00:6b:66:42`.
- `inet_proto` for a set that contains a list of Internet protocol types, such as `tcp`.
- `inet_service` for a set that contains a list of Internet services, such as `ssh`.
- `mark` for a set that contains a list of packet marks. Packet marks can be any positive 32-bit integer value (0 to 2147483647).

**Prerequisites**
- The `example_chain` chain and the `example_table` table exists.
Procedure

1. Create an empty set. The following examples create a set for IPv4 addresses:

   - To create a set that can store multiple individual IPv4 addresses:
     
     ```
     # nft add set inet example_table example_set { type ipv4_addr ; }
     ```

   - To create a set that can store IPv4 address ranges:
     
     ```
     # nft add set inet example_table example_set { type ipv4_addr ; flags interval ; }
     ```

   **IMPORTANT**
   
   To avoid that the shell interprets the semicolons as the end of the command, you must escape the semicolons with a backslash.

2. Optionally, create rules that use the set. For example, the following command adds a rule to the `example_chain` in the `example_table` that will drop all packets from IPv4 addresses in `example_set`:

   ```
   # nft add rule inet example_table example_chain ip saddr @example_set drop
   ```

   Because `example_set` is still empty, the rule has currently no effect.

3. Add IPv4 addresses to `example_set`:

   - If you create a set that stores individual IPv4 addresses, enter:
     
     ```
     # nft add element inet example_table example_set { 192.0.2.1, 192.0.2.2 }
     ```

   - If you create a set that stores IPv4 ranges, enter:
     
     ```
     # nft add element inet example_table example_set { 192.0.2.0-192.0.2.255 }
     ```

     When you specify an IP address range, you can alternatively use the Classless Inter-Domain Routing (CIDR) notation, such as `192.0.2.0/24` in the above example.

### 26.6.5.3. Additional resources

- The **Sets** section in the `nft(8)` man page

### 26.6.6. Using verdict maps in nftables commands

Verdict maps, which are also known as dictionaries, enable `nft` to perform an action based on packet information by mapping match criteria to an action.

#### 26.6.6.1. Using anonymous maps in nftables

An anonymous map is a `{ **match_criteria** : **action** }` statement that you use directly in a rule. The statement can contain multiple comma-separated mappings.
The drawback of an anonymous map is that if you want to change the map, you must replace the rule. For a dynamic solution, use named maps as described in Using named maps in nftables.

The example describes how to use an anonymous map to route both TCP and UDP packets of the IPv4 and IPv6 protocol to different chains to count incoming TCP and UDP packets separately.

Procedure

1. Create the example_table:
   
   ```
   # nft add table inet example_table
   ```

2. Create the tcp_packets chain in example_table:
   
   ```
   # nft add chain inet example_table tcp_packets
   ```

3. Add a rule to tcp_packets that counts the traffic in this chain:
   
   ```
   # nft add rule inet example_table tcp_packets counter
   ```

4. Create the udp_packets chain in example_table
   
   ```
   # nft add chain inet example_table udp_packets
   ```

5. Add a rule to udp_packets that counts the traffic in this chain:
   
   ```
   # nft add rule inet example_table udp_packets counter
   ```

6. Create a chain for incoming traffic. For example, to create a chain named incoming_traffic in example_table that filters incoming traffic:
   
   ```
   # nft add chain inet example_table incoming_traffic { type filter hook input priority 0 ; }
   ```

7. Add a rule with an anonymous map to incoming_traffic:
   
   ```
   # nft add rule inet example_table incoming_traffic ip protocol vmap { tcp : jump tcp_packets, udp : jump udp_packets }
   ```

   The anonymous map distinguishes the packets and sends them to the different counter chains based on their protocol.

8. To list the traffic counters, display example_table:
   
   ```
   # nft list table inet example_table
   table inet example_table {
   chain tcp_packets {
       counter packets 36379 bytes 2103816
   }
   chain udp_packets {
       counter packets 10 bytes 1559
   }
   ```
chain incoming_traffic {
  type filter hook input priority filter; policy accept;
  ip protocol vmap { tcp : jump tcp_packets, udp : jump udp_packets }
}

The counters in the tcp_packets and udp_packets chain display both the number of received packets and bytes.

26.6.6.2. Using named maps in nftables

The nftables framework supports named maps. You can use these maps in multiple rules within a table. Another benefit over anonymous maps is that you can update a named map without replacing the rules that use it.

When you create a named map, you must specify the type of elements:

- **ipv4_addr** for a map whose match part contains an IPv4 address, such as 192.0.2.1.
- **ipv6_addr** for a map whose match part contains an IPv6 address, such as 2001:db8:1::1.
- **ether_addr** for a map whose match part contains a media access control (MAC) address, such as 52:54:00:6b:66:42.
- **inet_proto** for a map whose match part contains an Internet protocol type, such as tcp.
- **inet_service** for a map whose match part contains an Internet services name port number, such as ssh or 22.
- **mark** for a map whose match part contains a packet mark. A packet mark can be any positive 32-bit integer value (0 to 2147483647).
- **counter** for a map whose match part contains a counter value. The counter value can be any positive 64-bit integer value.
- **quota** for a map whose match part contains a quota value. The quota value can be any positive 64-bit integer value.

The example describes how to allow or drop incoming packets based on their source IP address. Using a named map, you require only a single rule to configure this scenario while the IP addresses and actions are dynamically stored in the map. The procedure also describes how to add and remove entries from the map.

**Procedure**

1. Create a table. For example, to create a table named example_table that processes IPv4 packets:

```
# nft add table ip example_table
```

2. Create a chain. For example, to create a chain named example_chain in example_table:

```
# nft add chain ip example_table example_chain { type filter hook input priority 0 \; }
```
To avoid that the shell interprets the semicolons as the end of the command, you must escape the semicolons with a backslash.

3. Create an empty map. For example, to create a map for IPv4 addresses:

```
# nft add map ip example_table example_map { type ipv4_addr : verdict ; }
```

4. Create rules that use the map. For example, the following command adds a rule to `example_chain` in `example_table` that applies actions to IPv4 addresses which are both defined in `example_map`:

```
# nft add rule example_table example_chain ip saddr vmap @example_map
```

5. Add IPv4 addresses and corresponding actions to `example_map`:

```
# nft add element ip example_table example_map { 192.0.2.1 : accept, 192.0.2.2 : drop }
```

This example defines the mappings of IPv4 addresses to actions. In combination with the rule created above, the firewall accepts packet from 192.0.2.1 and drops packets from 192.0.2.2.

6. Optionally, enhance the map by adding another IP address and action statement:

```
# nft add element ip example_table example_map { 192.0.2.3 : accept }
```

7. Optionally, remove an entry from the map:

```
# nft delete element ip example_table example_map { 192.0.2.1 }
```

8. Optionally, display the rule set:

```
# nft list ruleset

table ip example_table {
    map example_map {
        type ipv4_addr : verdict
        elements = { 192.0.2.2 : drop, 192.0.2.3 : accept }
    }

    chain example_chain {
        type filter hook input priority filter; policy accept;
        ip saddr vmap @example_map
    }
}
```

### 26.6.6.3. Additional resources

- The Maps section in the `nft(8)` man page

### 26.6.7. Configuring port forwarding using nftables

Port forwarding enables administrators to forward packets sent to a specific destination port to a different local or remote port.
For example, if your web server does not have a public IP address, you can set a port forwarding rule on your firewall that forwards incoming packets on port 80 and 443 on the firewall to the web server. With this firewall rule, users on the internet can access the web server using the IP or host name of the firewall.

26.6.7.1. Forwarding incoming packets to a different local port

This section describes an example of how to forward incoming IPv4 packets on port 8022 to port 22 on the local system.

Procedure

1. Create a table named nat with the ip address family:

   ```bash
   # nft add table ip nat
   ```

2. Add the prerouting and postrouting chains to the table:

   ```bash
   # nft -- add chain ip nat prerouting { type nat hook prerouting priority -100 \; }
   ```

   **NOTE**
   Pass the -- option to the nft command to avoid that the shell interprets the negative priority value as an option of the nft command.

3. Add a rule to the prerouting chain that redirects incoming packets on port 8022 to the local port 22:

   ```bash
   # nft add rule ip nat prerouting tcp dport 8022 redirect to :22
   ```

26.6.7.2. Forwarding incoming packets on a specific local port to a different host

You can use a destination network address translation (DNAT) rule to forward incoming packets on a local port to a remote host. This enables users on the Internet to access a service that runs on a host with a private IP address.

The procedure describes how to forward incoming IPv4 packets on the local port 443 to the same port number on the remote system with the 192.0.2.1 IP address.

Prerequisites

- You are logged in as the root user on the system that should forward the packets.

Procedure

1. Create a table named nat with the ip address family:

   ```bash
   # nft add table ip nat
   ```

2. Add the prerouting and postrouting chains to the table:
3. Add a rule to the **prerouting** chain that redirects incoming packets on port **443** to the same port on **192.0.2.1**:

```
# nft add rule ip nat prerouting tcp dport 443 dnat to 192.0.2.1
```

4. Add a rule to the **postrouting** chain to masquerade outgoing traffic:

```
# nft add rule ip nat postrouting daddr 192.0.2.1 masquerade
```

5. Enable packet forwarding:

```
# echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/95-IPv4-forwarding.conf
# sysctl -p /etc/sysctl.d/95-IPv4-forwarding.conf
```

### 26.6.8. Using nftables to limit the amount of connections

You can use **nftables** to limit the number of connections or to block IP addresses that attempt to establish a given amount of connections to prevent them from using too many system resources.

#### 26.6.8.1. Limiting the number of connections using nftables

The **ct count** parameter of the **nft** utility enables administrators to limit the number of connections. The procedure describes a basic example of how to limit incoming connections.

**Prerequisites**

- The base **example_chain** in **example_table** exists.

**Procedure**

1. Create a dynamic set for IPv4 addresses:

```
# nft add set inet example_table example_meter { type ipv4_addr; flags dynamic ;}
```

2. Add a rule that allows only two simultaneous connections to the SSH port (22) from an IPv4 address and rejects all further connections from the same IP:

```
# nft add rule ip example_table example_chain tcp dport ssh meter example_meter { ip saddr ct count over 2 } counter reject
```

3. Optionally, display the set created in the previous step:

```
# nft list set inet example_table example_meter
```
The `elements` entry displays addresses that currently match the rule. In this example, `elements` lists IP addresses that have active connections to the SSH port. Note that the output does not display the number of active connections or if connections were rejected.

### 26.6.8.2. Blocking IP addresses that attempt more than ten new incoming TCP connections within one minute

This section explains how you temporarily block hosts that are establishing more than ten IPv4 TCP connections within one minute.

**Procedure**

1. Create the `filter` table with the `ip` address family:
   ```bash
   # nft add table ip filter
   ```

2. Add the `input` chain to the `filter` table:
   ```bash
   # nft add chain ip filter input { type filter hook input priority 0 \; }
   ```

3. Add a rule that drops all packets from source addresses that attempt to establish more than ten TCP connections within one minute:
   ```bash
   # nft add rule ip filter input ip protocol tcp ct state new, untracked meter ratemeter { ip saddr timeout 5m limit rate over 10/minute } drop
   ```

   The `timeout 5m` parameter defines that `nftables` automatically removes entries after five minutes to prevent that the meter fills up with stale entries.

**Verification**

- To display the meter’s content, enter:
  ```bash
  # nft list meter ip filter ratemeter
  ```

### 26.6.9. Debugging nftables rules
The `nftables` framework provides different options for administrators to debug rules and if packets match them. This section describes these options.

### 26.6.9.1. Creating a rule with a counter

To identify if a rule is matched, you can use a counter. This section describes how to create a new rule with a counter.

- For more information on a procedure that adds a counter to an existing rule, see [Adding a counter to an existing rule](#) in Configuring and managing networking

**Prerequisites**

- The chain to which you want to add the rule exists.

**Procedure**

1. Add a new rule with the `counter` parameter to the chain. The following example adds a rule with a counter that allows TCP traffic on port 22 and counts the packets and traffic that match this rule:

   ```
   # nft add rule inet example_table example_chain tcp dport 22 counter accept
   ```

2. To display the counter values:

   ```
   # nft list ruleset
   table inet example_table {
     chain example_chain {
       type filter hook input priority filter; policy accept;
       tcp dport ssh counter packets 6872 bytes 105448565 accept
     }
   }
   ```

### 26.6.9.2. Adding a counter to an existing rule

To identify if a rule is matched, you can use a counter. This section describes how to add a counter to an existing rule.

- For more information on a procedure that adds a new rule with a counter, see [Creating a rule with the counter](#) in Configuring and managing networking

**Prerequisites**

- The rule to which you want to add the counter exists.

**Procedure**

1. Display the rules in the chain including their handles:

   ```
   # nft --handle list chain inet example_table example_chain
   table inet example_table {
     chain example_chain { # handle 1
       type filter hook input priority filter; policy accept;
     }
   }
   ```
2. Add the counter by replacing the rule but with the `counter` parameter. The following example replaces the rule displayed in the previous step and adds a counter:

```bash
# nft replace rule inet example_table example_chain handle 4 tcp dport 22 counter accept
```

3. To display the counter values:

```bash
# nft list ruleset
table inet example_table {
    chain example_chain {
        type filter hook input priority filter; policy accept;
tcp dport ssh counter packets 6872 bytes 105448565 accept
    }
}
```

### 26.6.9.3. Monitoring packets that match an existing rule

The tracing feature in `nftables` in combination with the `nft monitor` command enables administrators to display packets that match a rule. The procedure describes how to enable tracing for a rule as well as monitoring packets that match this rule.

**Prerequisites**

- The rule to which you want to add the counter exists.

**Procedure**

1. Display the rules in the chain including their handles:

```bash
# nft --handle list chain inet example_table example_chain
table inet example_table {
    chain example_chain { # handle 1
type filter hook input priority filter; policy accept;
tcp dport ssh accept # handle 4
    }
}
```

2. Add the tracing feature by replacing the rule but with the `meta nftrace set 1` parameters. The following example replaces the rule displayed in the previous step and enables tracing:

```bash
# nft replace rule inet example_table example_chain handle 4 tcp dport 22 meta nftrace set 1 accept
```

3. Use the `nft monitor` command to display the tracing. The following example filters the output of the command to display only entries that contain `inet example_table example_chain`:

```bash
# nft monitor | grep "inet example_table example_chain"
```

trace id 3c5e15e inet example_table example_chain packet: iif "enp1s0" ether saddr 52:54:00:17:ff:e4 ether daddr 52:54:00:72:2f:6e ip saddr 192.0.2.1 ip daddr 192.0.2.2 ip dscp
WARNING

Depending on the number of rules with tracing enabled and the amount of matching traffic, the `nft monitor` command can display a lot of output. Use `grep` or other utilities to filter the output.

26.6.10. Backing up and restoring the `nftables` rule set

This section describes how to backup `nftables` rules to a file, as well as restoring rules from a file. Administrators can use a file with the rules to, for example, transfer the rules to a different server.

26.6.10.1. Backing up the `nftables` rule set to a file

This section describes how to back up the `nftables` rule set to a file.

Procedure

- To backup `nftables` rules:
  - In a format produced by `nft list ruleset` format:

    ```
    # nft list ruleset > file.nft
    ```
  - In JSON format:

    ```
    # nft -j list ruleset > file.json
    ```

26.6.10.2. Restoring the `nftables` rule set from a file

This section describes how to restore the `nftables` rule set.

Procedure

- To restore `nftables` rules:
  - If the file to restore is in the format produced by `nft list ruleset` or contains `nft` commands directly:

    ```
    # nft -f file.nft
    ```
  - If the file to restore is in JSON format:
26.6.11. Additional resources

- Using nftables in Red Hat Enterprise Linux 8
- What comes after iptables? Its successor, of course: nftables
- Firewalld: The Future is nftables
PART IV. DESIGN OF HARD DISK
Choosing the file system that is appropriate for your application is an important decision due to the large number of options available and the trade-offs involved. This chapter describes some of the file systems that ship with Red Hat Enterprise Linux 8 and provides historical background and recommendations on the right file system to suit your application.

### 27.1. TYPES OF FILE SYSTEMS

Red Hat Enterprise Linux 8 supports a variety of file systems (FS). Different types of file systems solve different kinds of problems, and their usage is application specific. At the most general level, available file systems can be grouped into the following major types:

<table>
<thead>
<tr>
<th>Type</th>
<th>File system</th>
<th>Attributes and use cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk or local FS</td>
<td>XFS</td>
<td>XFS is the default file system in RHEL. Because it lays out files as extents, it is less vulnerable to fragmentation than ext4. Red Hat recommends deploying XFS as your local file system unless there are specific reasons to do otherwise: for example, compatibility or corner cases around performance.</td>
</tr>
<tr>
<td></td>
<td>ext4</td>
<td>ext4 has the benefit of longevity in Linux. Therefore, it is supported by almost all Linux applications. In most cases, it rivals XFS on performance. ext4 is commonly used for home directories.</td>
</tr>
<tr>
<td>Network or client-and-server FS</td>
<td>NFS</td>
<td>Use NFS to share files between multiple systems on the same network.</td>
</tr>
<tr>
<td></td>
<td>SMB</td>
<td>Use SMB for file sharing with Microsoft Windows systems.</td>
</tr>
<tr>
<td>Shared storage or shared disk FS</td>
<td>GFS2</td>
<td>GFS2 provides shared write access to members of a compute cluster. The emphasis is on stability and reliability, with the functional experience of a local file system as possible. SAS Grid, Tibco MQ, IBM Websphere MQ, and Red Hat Active MQ have been deployed successfully on GFS2.</td>
</tr>
<tr>
<td>Volume-managing FS</td>
<td>Stratis (Technology Preview)</td>
<td>Stratis is a volume manager built on a combination of XFS and LVM. The purpose of Stratis is to emulate capabilities offered by volume-managing file systems like Btrfs and ZFS. It is possible to build this stack manually, but Stratis reduces configuration complexity, implements best practices, and consolidates error information.</td>
</tr>
</tbody>
</table>

### 27.2. LOCAL FILE SYSTEMS
Local file systems are file systems that run on a single, local server and are directly attached to storage.

For example, a local file system is the only choice for internal SATA or SAS disks, and is used when your server has internal hardware RAID controllers with local drives. Local file systems are also the most common file systems used on SAN attached storage when the device exported on the SAN is not shared.

All local file systems are POSIX-compliant and are fully compatible with all supported Red Hat Enterprise Linux releases. POSIX-compliant file systems provide support for a well-defined set of system calls, such as `read()`, `write()`, and `seek()`.

From the application programmer’s point of view, there are relatively few differences between local file systems. The most notable differences from a user’s perspective are related to scalability and performance. When considering a file system choice, consider how large the file system needs to be, what unique features it should have, and how it performs under your workload.

Available local file systems

- XFS
- ext4

### 27.3. THE XFS FILE SYSTEM

XFS is a highly scalable, high-performance, robust, and mature 64-bit journaling file system that supports very large files and file systems on a single host. It is the default file system in Red Hat Enterprise Linux 8. XFS was originally developed in the early 1990s by SGI and has a long history of running on extremely large servers and storage arrays.

The features of XFS include:

#### Reliability

- Metadata journaling, which ensures file system integrity after a system crash by keeping a record of file system operations that can be replayed when the system is restarted and the file system remounted
- Extensive run-time metadata consistency checking
- Scalable and fast repair utilities
- Quota journaling. This avoids the need for lengthy quota consistency checks after a crash.

#### Scalability and performance

- Supported file system size up to 1024 TiB
- Ability to support a large number of concurrent operations
- B-tree indexing for scalability of free space management
- Sophisticated metadata read-ahead algorithms
- Optimizations for streaming video workloads

#### Allocation schemes
- Extent-based allocation
- Stripe-aware allocation policies
- Delayed allocation
- Space pre-allocation
- Dynamically allocated inodes

**Other features**

- Reflink-based file copies
- Tightly integrated backup and restore utilities
- Online defragmentation
- Online file system growing
- Comprehensive diagnostics capabilities
  - Extended attributes (xattr). This allows the system to associate several additional name/value pairs per file.
  - Project or directory quotas. This allows quota restrictions over a directory tree.
  - Subsecond timestamps

**Performance characteristics**

XFS has a high performance on large systems with enterprise workloads. A large system is one with a relatively high number of CPUs, multiple HBAs, and connections to external disk arrays. XFS also performs well on smaller systems that have a multi-threaded, parallel I/O workload.

XFS has a relatively low performance for single threaded, metadata-intensive workloads: for example, a workload that creates or deletes large numbers of small files in a single thread.

### 27.4. THE EXT4 FILE SYSTEM

The ext4 file system is the fourth generation of the ext file system family. It was the default file system in Red Hat Enterprise Linux 6.

The ext4 driver can read and write to ext2 and ext3 file systems, but the ext4 file system format is not compatible with ext2 and ext3 drivers.

ext4 adds several new and improved features, such as:

- Supported file system size up to 50 TiB
- Extent-based metadata
- Delayed allocation
- Journal checksumming
Large storage support

The extent-based metadata and the delayed allocation features provide a more compact and efficient way to track utilized space in a file system. These features improve file system performance and reduce the space consumed by metadata. Delayed allocation allows the file system to postpone selection of the permanent location for newly written user data until the data is flushed to disk. This enables higher performance since it can allow for larger, more contiguous allocations, allowing the file system to make decisions with much better information.

File system repair time using the `fsck` utility in ext4 is much faster than in ext2 and ext3. Some file system repairs have demonstrated up to a six-fold increase in performance.

### 27.5. COMPARISON OF XFS AND EXT4

XFS is the default file system in RHEL. This section compares the usage and features of XFS and ext4.

**Metadata error behavior**

In ext4, you can configure the behavior when the file system encounters metadata errors. The default behavior is to simply continue the operation. When XFS encounters an unrecoverable metadata error, it shuts down the file system and returns the `EFSCORRUPTED` error.

**Quotas**

In ext4, you can enable quotas when creating the file system or later on an existing file system. You can then configure the quota enforcement using a mount option. XFS quotas are not a remountable option. You must activate quotas on the initial mount.

Running the `quotacheck` command on an XFS file system has no effect. The first time you turn on quota accounting, XFS checks quotas automatically.

**File system resize**

XFS has no utility to reduce the size of a file system. You can only increase the size of an XFS file system. In comparison, ext4 supports both extending and reducing the size of a file system.

**Inode numbers**

The ext4 file system does not support more than $2^{32}$ inodes. XFS dynamically allocates inodes. An XFS file system cannot run out of inodes as long as there is free space on the file system.

Certain applications cannot properly handle inode numbers larger than $2^{32}$ on an XFS file system. These applications might cause the failure of 32-bit stat calls with the `EOVERFLOW` return value. Inode number exceed $2^{32}$ under the following conditions:

- The file system is larger than 1 TiB with 256-byte inodes.
- The file system is larger than 2 TiB with 512-byte inodes.

If your application fails with large inode numbers, mount the XFS file system with the `-o inode32` option to enforce inode numbers below $2^{32}$. Note that using `inode32` does not affect inodes that are already allocated with 64-bit numbers.
Do not use the `inode32` option unless a specific environment requires it. The `inode32` option changes allocation behavior. As a consequence, the `ENOSPC` error might occur if no space is available to allocate inodes in the lower disk blocks.

27.6. CHOOSING A LOCAL FILE SYSTEM

To choose a file system that meets your application requirements, you need to understand the target system on which you are going to deploy the file system. You can use the following questions to inform your decision:

- Do you have a large server?
- Do you have large storage requirements or have a local, slow SATA drive?
- What kind of I/O workload do you expect your application to present?
- What are your throughput and latency requirements?
- How stable is your server and storage hardware?
- What is the typical size of your files and data set?
- If the system fails, how much downtime can you suffer?

If both your server and your storage device are large, XFS is the best choice. Even with smaller storage arrays, XFS performs very well when the average file sizes are large (for example, hundreds of megabytes in size).

If your existing workload has performed well with ext4, staying with ext4 should provide you and your applications with a very familiar environment.

The ext4 file system tends to perform better on systems that have limited I/O capability. It performs better on limited bandwidth (less than 200MB/s) and up to around 1000 IOPS capability. For anything with higher capability, XFS tends to be faster.

XFS consumes about twice the CPU-per-metadata operation compared to ext4, so if you have a CPU-bound workload with little concurrency, then ext4 will be faster. In general, ext4 is better if an application uses a single read/write thread and small files, while XFS shines when an application uses multiple read/write threads and bigger files.

You cannot shrink an XFS file system. If you need to be able to shrink the file system, consider using ext4, which supports offline shrinking.

In general, Red Hat recommends that you use XFS unless you have a specific use case for ext4. You should also measure the performance of your specific application on your target server and storage system to make sure that you choose the appropriate type of file system.

Table 27.2. Summary of local file system recommendations

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Recommended file system</th>
</tr>
</thead>
<tbody>
<tr>
<td>No special use case</td>
<td>XFS</td>
</tr>
</tbody>
</table>
### 27.7. NETWORK FILE SYSTEMS

Network file systems, also referred to as client/server file systems, enable client systems to access files that are stored on a shared server. This makes it possible for multiple users on multiple systems to share files and storage resources.

Such file systems are built from one or more servers that export a set of file systems to one or more clients. The client nodes do not have access to the underlying block storage, but rather interact with the storage using a protocol that allows for better access control.

**Available network file systems**

- The most common client/server file system for RHEL customers is the NFS file system. RHEL provides both an NFS server component to export a local file system over the network and an NFS client to import these file systems.

- RHEL also includes a CIFS client that supports the popular Microsoft SMB file servers for Windows interoperability. The userspace Samba server provides Windows clients with a Microsoft SMB service from a RHEL server.

### 27.8. SHARED STORAGE FILE SYSTEMS

Shared storage file systems, sometimes referred to as cluster file systems, give each server in the cluster direct access to a shared block device over a local storage area network (SAN).

**Comparison with network file systems**

Like client/server file systems, shared storage file systems work on a set of servers that are all members of a cluster. Unlike NFS, however, no single server provides access to data or metadata to other members: each member of the cluster has direct access to the same storage device (the

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Recommended file system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Large server</td>
<td>XFS</td>
</tr>
<tr>
<td>Large storage devices</td>
<td>XFS</td>
</tr>
<tr>
<td>Large files</td>
<td>XFS</td>
</tr>
<tr>
<td>Multi-threaded I/O</td>
<td>XFS</td>
</tr>
<tr>
<td>Single-threaded I/O</td>
<td>ext4</td>
</tr>
<tr>
<td>Limited I/O capability (under 1000 IOPS)</td>
<td>ext4</td>
</tr>
<tr>
<td>Limited bandwidth (under 200MB/s)</td>
<td>ext4</td>
</tr>
<tr>
<td>CPU-bound workload</td>
<td>ext4</td>
</tr>
<tr>
<td>Support for offline shrinking</td>
<td>ext4</td>
</tr>
</tbody>
</table>
shared storage), and all cluster member nodes access the same set of files.

Concurrency

Cache coherency is key in a clustered file system to ensure data consistency and integrity. There must be a single version of all files in a cluster visible to all nodes within a cluster. The file system must prevent members of the cluster from updating the same storage block at the same time and causing data corruption. In order to do that, shared storage file systems use a cluster wide-locking mechanism to arbitrate access to the storage as a concurrency control mechanism. For example, before creating a new file or writing to a file that is opened on multiple servers, the file system component on the server must obtain the correct lock.

The requirement of cluster file systems is to provide a highly available service like an Apache web server. Any member of the cluster will see a fully coherent view of the data stored in their shared disk file system, and all updates will be arbitrated correctly by the locking mechanisms.

Performance characteristics

Shared disk file systems do not always perform as well as local file systems running on the same system due to the computational cost of the locking overhead. Shared disk file systems perform well with workloads where each node writes almost exclusively to a particular set of files that are not shared with other nodes or where a set of files is shared in an almost exclusively read-only manner across a set of nodes. This results in a minimum of cross-node cache invalidation and can maximize performance.

Setting up a shared disk file system is complex, and tuning an application to perform well on a shared disk file system can be challenging.

Available shared storage file systems

- Red Hat Enterprise Linux provides the GFS2 file system. GFS2 comes tightly integrated with the Red Hat Enterprise Linux High Availability Add-On and the Resilient Storage Add-On.

Red Hat Enterprise Linux supports GFS2 on clusters that range in size from 2 to 16 nodes.

27.9. CHOOSING BETWEEN NETWORK AND SHARED STORAGE FILE SYSTEMS

When choosing between network and shared storage file systems, consider the following points:

- NFS-based network file systems are an extremely common and popular choice for environments that provide NFS servers.

- Network file systems can be deployed using very high-performance networking technologies like Infiniband or 10 Gigabit Ethernet. This means that you should not turn to shared storage file systems just to get raw bandwidth to your storage. If the speed of access is of prime importance, then use NFS to export a local file system like XFS.

- Shared storage file systems are not easy to set up or to maintain, so you should deploy them only when you cannot provide your required availability with either local or network file systems.

- A shared storage file system in a clustered environment helps reduce downtime by eliminating the steps needed for unmounting and mounting that need to be done during a typical fail-over scenario involving the relocation of a high-availability service.

Red Hat recommends that you use network file systems unless you have a specific use case for shared storage file systems. Use shared storage file systems primarily for deployments that need to provide high-availability services with minimum downtime and have stringent service-level requirements.
27.10. VOLUME-MANAGING FILE SYSTEMS

Volume-managing file systems integrate the entire storage stack for the purposes of simplicity and in-stack optimization.

Available volume-managing file systems

- Red Hat Enterprise Linux 8 provides the Stratis volume manager as a Technology Preview. Stratis uses XFS for the file system layer and integrates it with LVM, Device Mapper, and other components.

Stratis was first released in Red Hat Enterprise Linux 8.0. It is conceived to fill the gap created when Red Hat deprecated Btrfs. Stratis 1.0 is an intuitive, command line-based volume manager that can perform significant storage management operations while hiding the complexity from the user:

- Volume management
- Pool creation
- Thin storage pools
- Snapshots
- Automated read cache

Stratis offers powerful features, but currently lacks certain capabilities of other offerings that it might be compared to, such as Btrfs or ZFS. Most notably, it does not support CRCs with self healing.
CHAPTER 28. MOUNTING NFS SHARES

As a system administrator, you can mount remote NFS shares on your system to access shared data.

28.1. INTRODUCTION TO NFS

This section explains the basic concepts of the NFS service.

A Network File System (NFS) allows remote hosts to mount file systems over a network and interact with those file systems as though they are mounted locally. This enables you to consolidate resources onto centralized servers on the network.

The NFS server refers to the /etc/exports configuration file to determine whether the client is allowed to access any exported file systems. Once verified, all file and directory operations are available to the user.

28.2. SUPPORTED NFS VERSIONS

This section lists versions of NFS supported in Red Hat Enterprise Linux and their features.

Currently, Red Hat Enterprise Linux 8 supports the following major versions of NFS:

- NFS version 3 (NFSv3) supports safe asynchronous writes and is more robust at error handling than the previous NFSv2; it also supports 64-bit file sizes and offsets, allowing clients to access more than 2 GB of file data.
- NFS version 4 (NFSv4) works through firewalls and on the Internet, no longer requires an rpcbind service, supports Access Control Lists (ACLs), and utilizes stateful operations.
- NFS version 2 (NFSv2) is no longer supported by Red Hat.

Default NFS version
The default NFS version in Red Hat Enterprise Linux 8 is 4.2. NFS clients attempt to mount using NFSv4.2 by default, and fall back to NFSv4.1 when the server does not support NFSv4.2. The mount later falls back to NFSv4.0 and then to NFSv3.

Features of minor NFS versions
Following are the features of NFSv4.2 in Red Hat Enterprise Linux 8:

Server-side copy
- Enables the NFS client to efficiently copy data without wasting network resources using the copy_file_range() system call.

Sparse files
- Enables files to have one or more holes, which are unallocated or uninitialized data blocks consisting only of zeroes. The lseek() operation in NFSv4.2 supports seek_hole() and seek_data(), which enables applications to map out the location of holes in the sparse file.

Space reservation
- Permits storage servers to reserve free space, which prohibits servers to run out of space. NFSv4.2 supports the allocate() operation to reserve space, the deallocate() operation to unreserve space, and the fallocate() operation to preallocate or deallocate space in a file.

Labeled NFS
- Enforces data access rights and enables SELinux labels between a client and a server for individual files on an NFS file system.
Layout enhancements

Provides the `layoutstats()` operation, which enables some Parallel NFS (pNFS) servers to collect better performance statistics.

Following are the features of NFSv4.1:

- Enhances performance and security of network, and also includes client-side support for pNFS.
- No longer requires a separate TCP connection for callbacks, which allows an NFS server to grant delegations even when it cannot contact the client: for example, when NAT or a firewall interferes.
- Provides exactly once semantics (except for reboot operations), preventing a previous issue whereby certain operations sometimes returned an inaccurate result if a reply was lost and the operation was sent twice.

### 28.3. SERVICES REQUIRED BY NFS

This section lists system services that are required for running an NFS server or mounting NFS shares. Red Hat Enterprise Linux starts these services automatically.

Red Hat Enterprise Linux uses a combination of kernel-level support and service processes to provide NFS file sharing. All NFS versions rely on Remote Procedure Calls (RPC) between clients and servers. To share or mount NFS file systems, the following services work together depending on which version of NFS is implemented:

- **nfsd**
  - The NFS server kernel module that services requests for shared NFS file systems.

- **rpcbind**
  - Accepts port reservations from local RPC services. These ports are then made available (or advertised) so the corresponding remote RPC services can access them. The `rpcbind` service responds to requests for RPC services and sets up connections to the requested RPC service. This is not used with NFSv4.

- **rpc.mountd**
  - This process is used by an NFS server to process `MOUNT` requests from NFSv3 clients. It checks that the requested NFS share is currently exported by the NFS server, and that the client is allowed to access it. If the mount request is allowed, the `nfs-mountd` service replies with a Success status and provides the File-Handle for this NFS share back to the NFS client.

- **rpc.nfsd**
  - This process enables explicit NFS versions and protocols the server advertises to be defined. It works with the Linux kernel to meet the dynamic demands of NFS clients, such as providing server threads each time an NFS client connects. This process corresponds to the `nfs-server` service.

- **lockd**
  - This is a kernel thread that runs on both clients and servers. It implements the Network Lock Manager (NLM) protocol, which enables NFSv3 clients to lock files on the server. It is started automatically whenever the NFS server is run and whenever an NFS file system is mounted.

- **rpc.statd**
  - This process implements the Network Status Monitor (NSM) RPC protocol, which notifies NFS clients when an NFS server is restarted without being gracefully brought down. The `rpc-statd` service is started automatically by the `nfs-server` service, and does not require user configuration. This is not used with NFSv4.
**rpc.rquotad**

This process provides user quota information for remote users. The rpc-rquotad service, which is provided by the quota-rpc package, has to be started by user when the nfs-server is started.

**rpc.idmapd**

This process provides NFSv4 client and server upcalls, which map between on-the-wire NFSv4 names (strings in the form of `user@domain`) and local UIDs and GIDs. For idmapd to function with NFSv4, the `/etc/idmapd.conf` file must be configured. At a minimum, the Domain parameter should be specified, which defines the NFSv4 mapping domain. If the NFSv4 mapping domain is the same as the DNS domain name, this parameter can be skipped. The client and server must agree on the NFSv4 mapping domain for ID mapping to function properly.

Only the NFSv4 server uses rpc.idmapd, which is started by the nfs-idmapd service. The NFSv4 client uses the keyring-based nfsidmap utility, which is called by the kernel on-demand to perform ID mapping. If there is a problem with nfsidmap, the client falls back to using rpc.idmapd.

---

**The RPC services with NFSv4**

The mounting and locking protocols have been incorporated into the NFSv4 protocol. The server also listens on the well-known TCP port 2049. As such, NFSv4 does not need to interact with rpcbind, lockd, and rpc-statd services. The nfs-mountd service is still required on the NFS server to set up the exports, but is not involved in any over-the-wire operations.

**Additional resources**

- Configuring an NFSv4 only server without rpcbind.

---

**28.4. NFS HOST NAME FORMATS**

This section describes different formats that you can use to specify a host when mounting or exporting an NFS share.

You can specify the host in the following formats:

**Single machine**

Either of the following:

- A fully-qualified domain name (that can be resolved by the server)
- Host name (that can be resolved by the server)
- An IP address.

**IP networks**

Either of the following formats is valid:

- `a.b.c.d/z`, where `a.b.c.d` is the network and `z` is the number of bits in the netmask; for example, `192.168.0.0/24`.
- `a.b.c.d/netmask`, where `a.b.c.d` is the network and `netmask` is the netmask; for example, `192.168.100.8/255.255.255.0`.

**Netgroups**

The `@group-name` format, where `group-name` is the NIS netgroup name.
28.5. INSTALLING NFS

This procedure installs all packages necessary to mount or export NFS shares.

Procedure

- Install the `nfs-utils` package:
  
  ```
  # yum install nfs-utils
  ```

28.6. DISCOVERING NFS EXPORTS

This procedure discovers which file systems a given NFSv3 or NFSv4 server exports.

Procedure

- With any server that supports NFSv3, use the `showmount` utility:
  
  ```
  $ showmount --exports my-server
  
  Export list for `my-server`
  /exports/foo
  /exports/bar
  ```

- With any server that supports NFSv4, mount the root directory and look around:
  
  ```
  # mount my-server:/ /mnt/
  # ls /mnt/
  
  exports
  
  # ls /mnt/exports/
  
  foo
  bar
  ```

On servers that support both NFSv4 and NFSv3, both methods work and give the same results.

Additional resources

- `showmount(8)` man page

28.7. MOUNTING AN NFS SHARE WITH MOUNT

This procedure mounts an NFS share exported from a server using the `mount` utility.

Procedure

- To mount an NFS share, use the following command:
  
  ```
  # mount -t nfs -o options host:/remote/export /local/directory
  ```
This command uses the following variables:

**options**
A comma-delimited list of mount options.

**host**
The host name, IP address, or fully qualified domain name of the server exporting the file system you wish to mount.

**/remote/export**
The file system or directory being exported from the server, that is, the directory you wish to mount.

**/local/directory**
The client location where /remote/export is mounted.

Additional resources

- Common NFS mount options.
- NFS host name formats.
- Mounting a file system with mount.
- `mount(8)` man page
- `exports(5)` man page

### 28.8. COMMON NFS MOUNT OPTIONS

This section lists options commonly used when mounting NFS shares. These options can be used with manual mount commands, `/etc/fstab` settings, and `autofs`.

**lookupcache=mode**
Specifies how the kernel should manage its cache of directory entries for a given mount point. Valid arguments for `mode` are all, none, or positive.

**nfsvers=version**
Specifies which version of the NFS protocol to use, where version is 3, 4, 4.0, 4.1, or 4.2. This is useful for hosts that run multiple NFS servers, or to disable retrying a mount with lower versions. If no version is specified, NFS uses the highest version supported by the kernel and the `mount` utility. The option `vers` is identical to `nfsvers`, and is included in this release for compatibility reasons.

**noacl**
Turns off all ACL processing. This may be needed when interfacing with older versions of Red Hat Enterprise Linux, Red Hat Linux, or Solaris, because the most recent ACL technology is not compatible with older systems.

**nolock**
Disables file locking. This setting is sometimes required when connecting to very old NFS servers.

**noexec**
Prevents execution of binaries on mounted file systems. This is useful if the system is mounting a non-Linux file system containing incompatible binaries.

**nosuid**
Disables the **set-user-identifier** and **set-group-identifier** bits. This prevents remote users from gaining higher privileges by running a **setuid** program.

**port=num**

Specifies the numeric value of the NFS server port. If *num* is 0 (the default value), then **mount** queries the **rpcbind** service on the remote host for the port number to use. If the NFS service on the remote host is not registered with its **rpcbind** service, the standard NFS port number of TCP 2049 is used instead.

**rsize=num and wsize=num**

These options set the maximum number of bytes to be transferred in a single NFS read or write operation.

There is no fixed default value for *rsize* and *wsize*. By default, NFS uses the largest possible value that both the server and the client support. In Red Hat Enterprise Linux 8, the client and server maximum is 1,048,576 bytes. For more details, see the What are the default and maximum values for rsize and wsize with NFS mounts? KBase article.

**sec=flavors**

Security flavors to use for accessing files on the mounted export. The *flavors* value is a colon-separated list of one or more security flavors.

By default, the client attempts to find a security flavor that both the client and the server support. If the server does not support any of the selected flavors, the mount operation fails.

Available flavors:

- **sec=sys** uses local UNIX UIDs and GIDs. These use **AUTH_SYS** to authenticate NFS operations.
- **sec=krb5** uses Kerberos V5 instead of local UNIX UIDs and GIDs to authenticate users.
- **sec=krb5i** uses Kerberos V5 for user authentication and performs integrity checking of NFS operations using secure checksums to prevent data tampering.
- **sec=krb5p** uses Kerberos V5 for user authentication, integrity checking, and encrypts NFS traffic to prevent traffic sniffing. This is the most secure setting, but it also involves the most performance overhead.

**tcp**

Instructs the NFS mount to use the TCP protocol.

---

### 28.9. ADDITIONAL RESOURCES

- The Linux NFS wiki
- Mounting NFS shares persistently
- Mounting NFS shares on demand
CHAPTER 29. EXPORTING NFS SHARES

As a system administrator, you can use the NFS server to share a directory on your system over network.

29.1. INTRODUCTION TO NFS

This section explains the basic concepts of the NFS service.

A Network File System (NFS) allows remote hosts to mount file systems over a network and interact with those file systems as though they are mounted locally. This enables you to consolidate resources onto centralized servers on the network.

The NFS server refers to the `/etc/exports` configuration file to determine whether the client is allowed to access any exported file systems. Once verified, all file and directory operations are available to the user.

29.2. SUPPORTED NFS VERSIONS

This section lists versions of NFS supported in Red Hat Enterprise Linux and their features.

Currently, Red Hat Enterprise Linux 8 supports the following major versions of NFS:

- NFS version 3 (NFSv3) supports safe asynchronous writes and is more robust at error handling than the previous NFSv2; it also supports 64-bit file sizes and offsets, allowing clients to access more than 2 GB of file data.

- NFS version 4 (NFSv4) works through firewalls and on the Internet, no longer requires an `rpcbind` service, supports Access Control Lists (ACLs), and utilizes stateful operations.

NFS version 2 (NFSv2) is no longer supported by Red Hat.

Default NFS version

The default NFS version in Red Hat Enterprise Linux 8 is 4.2. NFS clients attempt to mount using NFSv4.2 by default, and fall back to NFSv4.1 when the server does not support NFSv4.2. The mount later falls back to NFSv4.0 and then to NFSv3.

Features of minor NFS versions

Following are the features of NFSv4.2 in Red Hat Enterprise Linux 8:

Server-side copy

- Enables the NFS client to efficiently copy data without wasting network resources using the `copy_file_range()` system call.

Sparse files

- Enables files to have one or more holes, which are unallocated or uninitialized data blocks consisting only of zeroes. The `lseek()` operation in NFSv4.2 supports `seek_hole()` and `seek_data()`, which enables applications to map out the location of holes in the sparse file.

Space reservation

- Permits storage servers to reserve free space, which prohibits servers to run out of space. NFSv4.2 supports the `allocate()` operation to reserve space, the `deallocate()` operation to unreserve space, and the `fallocate()` operation to preallocate or deallocate space in a file.

Labeled NFS

- Enforces data access rights and enables SELinux labels between a client and a server for individual files on an NFS file system.
Layout enhancements

Provides the `layoutstats()` operation, which enables some Parallel NFS (pNFS) servers to collect better performance statistics.

Following are the features of NFSv4.1:

- Enhances performance and security of network, and also includes client-side support for pNFS.
- No longer requires a separate TCP connection for callbacks, which allows an NFS server to grant delegations even when it cannot contact the client: for example, when NAT or a firewall interferes.
- Provides exactly once semantics (except for reboot operations), preventing a previous issue whereby certain operations sometimes returned an inaccurate result if a reply was lost and the operation was sent twice.

29.3. THE TCP AND UDP PROTOCOLS IN NFSV3 AND NFSV4

NFSv4 requires the Transmission Control Protocol (TCP) running over an IP network.

NFSv3 could also use the User Datagram Protocol (UDP) in earlier Red Hat Enterprise Linux versions. In Red Hat Enterprise Linux 8, NFS over UDP is no longer supported. By default, UDP is disabled in the NFS server.

29.4. SERVICES REQUIRED BY NFS

This section lists system services that are required for running an NFS server or mounting NFS shares. Red Hat Enterprise Linux starts these services automatically.

Red Hat Enterprise Linux uses a combination of kernel-level support and service processes to provide NFS file sharing. All NFS versions rely on Remote Procedure Calls (RPC) between clients and servers. To share or mount NFS file systems, the following services work together depending on which version of NFS is implemented:

- **nfsd**
  
The NFS server kernel module that services requests for shared NFS file systems.

- **rpcbind**
  
  Accepts port reservations from local RPC services. These ports are then made available (or advertised) so the corresponding remote RPC services can access them. The `rpcbind` service responds to requests for RPC services and sets up connections to the requested RPC service. This is not used with NFSv4.

- **rpc.mountd**
  
  This process is used by an NFS server to process `MOUNT` requests from NFSv3 clients. It checks that the requested NFS share is currently exported by the NFS server, and that the client is allowed to access it. If the mount request is allowed, the `nfs-mountd` service replies with a Success status and provides the File-Handle for this NFS share back to the NFS client.

- **rpc.nfsd**
  
  This process enables explicit NFS versions and protocols the server advertises to be defined. It works with the Linux kernel to meet the dynamic demands of NFS clients, such as providing server threads each time an NFS client connects. This process corresponds to the `nfs-server` service.

- **lockd**
  
  This is a kernel thread that runs on both clients and servers. It implements the Network Lock
This is a kernel thread that runs on both clients and servers. It implements the Network Lock Manager (NLM) protocol, which enables NFSv3 clients to lock files on the server. It is started automatically whenever the NFS server is run and whenever an NFS file system is mounted.

**rpc.statd**

This process implements the Network Status Monitor (NSM) RPC protocol, which notifies NFS clients when an NFS server is restarted without being gracefully brought down. The `rpc.statd` service is started automatically by the `nfs-server` service, and does not require user configuration. This is not used with NFSv4.

**rpc.rquotad**

This process provides user quota information for remote users. The `rpc.rquotad` service, which is provided by the `quota-rpc` package, has to be started by user when the `nfs-server` is started.

**rpc.idmapd**

This process provides NFSv4 client and server upcalls, which map between on-the-wire NFSv4 names (strings in the form of `user@domain`) and local UIDs and GIDs. For `idmapd` to function with NFSv4, the `/etc/idmapd.conf` file must be configured. At a minimum, the Domain parameter should be specified, which defines the NFSv4 mapping domain. If the NFSv4 mapping domain is the same as the DNS domain name, this parameter can be skipped. The client and server must agree on the NFSv4 mapping domain for ID mapping to function properly.

Only the NFSv4 server uses `rpc.idmapd`, which is started by the `nfs-idmapd` service. The NFSv4 client uses the keyring-based `nfsidmap` utility, which is called by the kernel on-demand to perform ID mapping. If there is a problem with `nfsidmap`, the client falls back to using `rpc.idmapd`.

The RPC services with NFSv4

The mounting and locking protocols have been incorporated into the NFSv4 protocol. The server also listens on the well-known TCP port 2049. As such, NFSv4 does not need to interact with `rpcbind`, `lockd`, and `rpc.statd` services. The `nfs-mountd` service is still required on the NFS server to set up the exports, but is not involved in any over-the-wire operations.

Additional resources

- Configuring an NFSv4 only server without `rpcbind`.

29.5. NFS HOST NAME FORMATS

This section describes different formats that you can use to specify a host when mounting or exporting an NFS share.

You can specify the host in the following formats:

**Single machine**

Either of the following:

- A fully-qualified domain name (that can be resolved by the server)
- Host name (that can be resolved by the server)
- An IP address.

**IP networks**

Either of the following formats is valid:

- `a.b.c.d/z`, where `a.b.c.d` is the network and `z` is the number of bits in the netmask; for
example 192.168.0.0/24.

- \(a.b.c.d/netmask\), where \(a.b.c.d\) is the network and \(netmask\) is the netmask; for example, 192.168.100.8/255.255.255.0.

Netgroups

The \(@\textit{group-name}\) format, where \textit{group-name} is the NIS netgroup name.

29.6. NFS SERVER CONFIGURATION

This section describes the syntax and options of two ways to configure exports on an NFS server:

- Manually editing the `/etc/exports` configuration file
- Using the `exportfs` utility on the command line

29.6.1. The `/etc/exports` configuration file

The `/etc/exports` file controls which file systems are exported to remote hosts and specifies options. It follows the following syntax rules:

- Blank lines are ignored.
- To add a comment, start a line with the hash mark (`#`).
- You can wrap long lines with a backslash (`\`).
- Each exported file system should be on its own individual line.
- Any lists of authorized hosts placed after an exported file system must be separated by space characters.
- Options for each of the hosts must be placed in parentheses directly after the host identifier, without any spaces separating the host and the first parenthesis.

Export entry

Each entry for an exported file system has the following structure:

```
export host(options)
```

It is also possible to specify multiple hosts, along with specific options for each host. To do so, list them on the same line as a space-delimited list, with each host name followed by its respective options (in parentheses), as in:

```
export host1(options1) host2(options2) host3(options3)
```

In this structure:

- `export`: The directory being exported
- `host`: The host or network to which the export is being shared
- `options`:
The options to be used for host

Example 29.1. A simple /etc/exports file

In its simplest form, the /etc/exports file only specifies the exported directory and the hosts permitted to access it:

```
/exported/directory bob.example.com
```

Here, bob.example.com can mount /exported/directory/ from the NFS server. Because no options are specified in this example, NFS uses default options.

**IMPORTANT**

The format of the /etc/exports file is very precise, particularly in regards to use of the space character. Remember to always separate exported file systems from hosts and hosts from one another with a space character. However, there should be no other space characters in the file except on comment lines.

For example, the following two lines do not mean the same thing:

```
/home bob.example.com(rw)
/home bob.example.com (rw)
```

The first line allows only users from bob.example.com read and write access to the /home directory. The second line allows users from bob.example.com to mount the directory as read-only (the default), while the rest of the world can mount it read/write.

**Default options**

The default options for an export entry are:

- **ro**
  The exported file system is read-only. Remote hosts cannot change the data shared on the file system. To allow hosts to make changes to the file system (that is, read and write), specify the rw option.

- **sync**
  The NFS server will not reply to requests before changes made by previous requests are written to disk. To enable asynchronous writes instead, specify the option async.

- **wdelay**
  The NFS server will delay writing to the disk if it suspects another write request is imminent. This can improve performance as it reduces the number of times the disk must be accessed by separate write commands, thereby reducing write overhead. To disable this, specify the no_wdelay option, which is available only if the default sync option is also specified.

- **root_squash**
  This prevents root users connected remotely (as opposed to locally) from having root privileges; instead, the NFS server assigns them the user ID nobody. This effectively "squashes" the power of the remote root user to the lowest local user, preventing possible unauthorized writes on the remote server. To disable root squashing, specify the no_root_squash option.
  To squash every remote user (including root), use the all_squash option. To specify the user and group IDs that the NFS server should assign to remote users from a particular host, use the anonuid and anongid options, respectively, as in:
export host(anonuid=uid,anongid=gid)

Here, uid and gid are user ID number and group ID number, respectively. The anonuid and anongid options enable you to create a special user and group account for remote NFS users to share.

By default, access control lists (ACLs) are supported by NFS under Red Hat Enterprise Linux. To disable this feature, specify the no_acl option when exporting the file system.

Default and overridden options
Each default for every exported file system must be explicitly overridden. For example, if the rw option is not specified, then the exported file system is shared as read-only. The following is a sample line from /etc/exports which overrides two default options:

/another/exported/directory 192.168.0.3(rw,async)

In this example, 192.168.0.3 can mount /another/exported/directory/ read and write, and all writes to disk are asynchronous.

29.6.2. The exportfs utility
The exportfs utility enables the root user to selectively export or unexport directories without restarting the NFS service. When given the proper options, the exportfs utility writes the exported file systems to /var/lib/nfs/xtab. Because the nfs-mountd service refers to the xtab file when deciding access privileges to a file system, changes to the list of exported file systems take effect immediately.

Common exportfs options
The following is a list of commonly-used options available for exportfs:

-r
Causes all directories listed in /etc/exports to be exported by constructing a new export list in /var/lib/nfs/etab. This option effectively refreshes the export list with any changes made to /etc/exports.

-a
Causes all directories to be exported or unexported, depending on what other options are passed to exportfs. If no other options are specified, exportfs exports all file systems specified in /etc/exports.

-o file-systems
Specifies directories to be exported that are not listed in /etc/exports. Replace file-systems with additional file systems to be exported. These file systems must be formatted in the same way they are specified in /etc/exports. This option is often used to test an exported file system before adding it permanently to the list of exported file systems.

-i
Ignores /etc/exports; only options given from the command line are used to define exported file systems.

-u
Unexports all shared directories. The command exportfs -ua suspends NFS file sharing while keeping all NFS services up. To re-enable NFS sharing, use exportfs -r.

-v
Verbose operation, where the file systems being exported or unexported are displayed in greater detail when the exportfs command is executed.
If no options are passed to the `exportfs` utility, it displays a list of currently exported file systems.

Additional resources

- NFS host name formats.

29.7. NFS AND RPCBIND

This section explains the purpose of the `rpcbind` service, which is required by NFSv3.

The `rpcbind` service maps Remote Procedure Call (RPC) services to the ports on which they listen. RPC processes notify `rpcbind` when they start, registering the ports they are listening on and the RPC program numbers they expect to serve. The client system then contacts `rpcbind` on the server with a particular RPC program number. The `rpcbind` service redirects the client to the proper port number so it can communicate with the requested service.

Because RPC-based services rely on `rpcbind` to make all connections with incoming client requests, `rpcbind` must be available before any of these services start.

Access control rules for `rpcbind` affect all RPC-based services. Alternatively, it is possible to specify access control rules for each of the NFS RPC daemons.

Additional resources

- `rpc.mountd(8)` man page.
- `rpc.statd(8)` man page.

29.8. INSTALLING NFS

This procedure installs all packages necessary to mount or export NFS shares.

Procedure

- Install the `nfs-utils` package:

  ```bash
  # yum install nfs-utils
  ```

29.9. STARTING THE NFS SERVER

This procedure describes how to start the NFS server, which is required to export NFS shares.

Prerequisites

- For servers that support NFSv3 connections, the `rpcbind` service must be running. To verify that `rpcbind` is active, use the following command:

  ```bash
  $ systemctl status rpcbind
  ```

  If the service is stopped, start and enable it:

  ```bash
  $ systemctl enable --now rpcbind
  ```
Procedure

- To start the NFS server and enable it to start automatically at boot, use the following command:

  # systemctl enable --now nfs-server

Additional resources

- Configuring an NFSv4-only server.

### 29.10. TROUBLESHOOTING NFS AND RPCBIND

Because the `rpcbind` service provides coordination between RPC services and the port numbers used to communicate with them, it is useful to view the status of current RPC services using `rpcbind` when troubleshooting. The `rpcinfo` utility shows each RPC-based service with port numbers, an RPC program number, a version number, and an IP protocol type (TCP or UDP).

**Procedure**

1. To make sure the proper NFS RPC-based services are enabled for `rpcbind`, use the following command:

   # rpcinfo -p

**Example 29.2. rpcinfo -p command output**

The following is sample output from this command:

```
program vers proto   port  service
  100000    4   tcp    111  portmapper
  100000    3   tcp    111  portmapper
  100000    2   tcp    111  portmapper
  100000    4   udp    111  portmapper
  100000    3   udp    111  portmapper
  100000    2   udp    111  portmapper
  100005    1   udp  20048  mountd
  100005    1   tcp  20048  mountd
  100005    2   udp  20048  mountd
  100005    2   tcp  20048  mountd
  100005    3   udp  20048  mountd
  100005    3   tcp  20048  mountd
  100024    1   udp  37769  status
  100024    1   tcp  49349  status
  100003    3   tcp   2049  nfs
  100003    4   tcp   2049  nfs
  100227    3   tcp  2049  nfs_acl
  100021    1   udp  56691  nlockmgr
  100021    3   udp  56691  nlockmgr
  100021    4   udp  56691  nlockmgr
  100021    1   tcp  46193  nlockmgr
  100021    3   tcp  46193  nlockmgr
  100021    4   tcp  46193  nlockmgr
```
If one of the NFS services does not start up correctly, `rpcbind` will be unable to map RPC requests from clients for that service to the correct port.

2. In many cases, if NFS is not present in `rpcinfo` output, restarting NFS causes the service to correctly register with `rpcbind` and begin working:

```
# systemctl restart nfs-server
```

Additional resources

- Configuring an NFSv4-only server.

### 29.11. CONFIGURING THE NFS SERVER TO RUN BEHIND A FIREWALL

NFS requires the `rpcbind` service, which dynamically assigns ports for RPC services and can cause issues for configuring firewall rules. This procedure describes how to configure the NFS server to work behind a firewall.

**Procedure**

1. To allow clients to access NFS shares behind a firewall, set which ports the RPC services run on in the `[mountd]` section of the `/etc/nfs.conf` file:

```
[mountd]
port=port-number
```

   This adds the `-p port-number` option to the `rpc.mount` command line: `rpc.mount -p port-number`.

2. To allow clients to access NFS shares behind a firewall, configure the firewall by running the following commands on the NFS server:

```
firewall-cmd --permanent --add-service mountd
firewall-cmd --permanent --add-service rpc-bind
firewall-cmd --permanent --add-service nfs
firewall-cmd --permanent --add-port=<mountd-port>/tcp
firewall-cmd --permanent --add-port=<mountd-port>/udp
firewall-cmd --reload
```

   In the commands, replace `<mountd-port>` with the intended port or a port range. When specifying a port range, use the `--add-port=<mountd-port>-<mountd-port>/udp` syntax.

3. To allow NFSv4.0 callbacks to pass through firewalls, set `/proc/sys/fs/nfs/nfs_callback_tcpport` and allow the server to connect to that port on the client.

   This step is not needed for NFSv4.1 or higher, and the other ports for `mountd`, `statd`, and `lockd` are not required in a pure NFSv4 environment.

4. To specify the ports to be used by the RPC service `nlockmgr`, set the port number for the `nlm_tcpport` and `nlm_udpport` options in the `/etc/modprobe.d/lockd.conf` file.

5. Restart the NFS server:
If NFS fails to start, check `/var/log/messages`. Commonly, NFS fails to start if you specify a port number that is already in use.

6. Confirm the changes have taken effect:

```bash
# rpcinfo -p
```

Additional resources

- Configuring an NFSv4-only server.

### 29.12. EXPORTING RPC QUOTA THROUGH A FIREWALL

If you export a file system that uses disk quotas, you can use the quota Remote Procedure Call (RPC) service to provide disk quota data to NFS clients.

**Procedure**

1. Enable and start the `rpc-rquotad` service:

   ```bash
   # systemctl enable --now rpc-rquotad
   ``

   **NOTE**

   The `rpc-rquotad` service is, if enabled, started automatically after starting the `nfs-server` service.

2. To make the quota RPC service accessible behind a firewall, the TCP (or UDP, if UDP is enabled) port 875 need to be open. The default port number is defined in the `/etc/services` file. You can override the default port number by appending `-p port-number` to the `RPCRQUOTADOPTS` variable in the `/etc/sysconfig/rpc-rquotad` file.

3. By default, remote hosts can only read quotas. If you want to allow clients to set quotas, append the `-S` option to the `RPCRQUOTADOPTS` variable in the `/etc/sysconfig/rpc-rquotad` file.

4. Restart `rpc-rquotad` for the changes in the `/etc/sysconfig/rpc-rquotad` file to take effect:

   ```bash
   # systemctl restart rpc-rquotad
   ``

### 29.13. ENABLING NFS OVER RDMA (NFSORDMA)

The remote direct memory access (RDMA) service works automatically in Red Hat Enterprise Linux 8 if there is RDMA-capable hardware present.

**Procedure**

1. Install the `rdma-core` package:

   ```bash
   # yum install rdma-core
   ```
2. To enable automatic loading of NFSoRDMA server modules, add the `SVCRDMA_LOAD=yes` option on a new line in the `/etc/rdma/rdma.conf` configuration file. The `rdma=20049` option in the `[nfsd]` section of the `/etc/nfs.conf` file specifies the port number on which the NFSoRDMA service listens for clients. The RFC 5667 standard specifies that servers must listen on port 20049 when providing NFSv4 services over RDMA.

The `/etc/rdma/rdma.conf` file contains a line that sets the `XPRTRDMA_LOAD=yes` option by default, which requests the `rdma` service to load the NFSoRDMA client module.

3. Restart the `nfs-server` service:

```
# systemctl restart nfs-server
```

**Additional resources**

- The RFC 5667 standard.

**29.14. ADDITIONAL RESOURCES**

- The Linux NFS wiki
CHAPTER 30. MOUNTING AN SMB SHARE ON RED HAT ENTERPRISE LINUX

The Server Message Block (SMB) protocol implements an application-layer network protocol used to access resources on a server, such as file shares and shared printers.

NOTE

In the context of SMB, you can find mentions about the Common Internet File System (CIFS) protocol, which is a dialect of SMB. Both the SMB and CIFS protocol are supported, and the kernel module and utilities involved in mounting SMB and CIFS shares both use the name `cifs`.

This section describes how to mount shares from an SMB server. For details about setting up an SMB server on Red Hat Enterprise Linux using Samba, see Using Samba as a server.

Prerequisites

On Microsoft Windows, SMB is implemented by default. On Red Hat Enterprise Linux, the `cifs.ko` file system module of the kernel provides support for mounting SMB shares. Therefore, install the `cifs-utils` package:

```
# yum install cifs-utils
```

The `cifs-utils` package provides utilities to:

- Mount SMB and CIFS shares
- Manage NT Lan Manager (NTLM) credentials in the kernel’s keyring
- Set and display Access Control Lists (ACL) in a security descriptor on SMB and CIFS shares

30.1. SUPPORTED SMB PROTOCOL VERSIONS

The `cifs.ko` kernel module supports the following SMB protocol versions:

- SMB 1
- SMB 2.0
- SMB 2.1

WARNING

The SMB1 protocol is deprecated due to known security issues, and is only safe to use on a private network. The main reason that SMB1 is still provided as a supported option is that currently it is the only SMB protocol version that supports UNIX extensions. If you do not need to use UNIX extensions on SMB, Red Hat strongly recommends using SMB2 or later.
NOTE
Depending on the protocol version, not all SMB features are implemented.

30.2. UNIX EXTENSIONS SUPPORT

Samba uses the CAP_UNIX capability bit in the SMB protocol to provide the UNIX extensions feature. These extensions are also supported by the cifs.ko kernel module. However, both Samba and the kernel module support UNIX extensions only in the SMB 1 protocol.

To use UNIX extensions:

1. Set the `server min protocol` parameter in the `[global]` section in the `/etc/samba/smb.conf` file to `NT1`.

2. Mount the share using the SMB 1 protocol by providing the `-o vers=1.0` option to the `mount` command. For example:

   ```bash
   # mount -t cifs -o vers=1.0,username=user_name //server_name/share_name /mnt/
   ```

   By default, the kernel module uses SMB 2 or the highest later protocol version supported by the server. Passing the `-o vers=1.0` option to the `mount` command forces that the kernel module uses the SMB 1 protocol that is required for using UNIX extensions.

To verify if UNIX extensions are enabled, display the options of the mounted share:

```bash
# mount
... 
//server/share on /mnt type cifs (...)unix,...)
```

If the `unix` entry is displayed in the list of mount options, UNIX extensions are enabled.

30.3. MANUALLY MOUNTING AN SMB SHARE

If you only require an SMB share to be temporary mounted, you can mount it manually using the `mount` utility.

NOTE
Manually mounted shares are not mounted automatically again when you reboot the system. To configure that Red Hat Enterprise Linux automatically mounts the share when the system boots, see Mounting an SMB share automatically when the system boots.

Prerequisites

- The cifs-utils package is installed.

Procedure

To manually mount an SMB share, use the `mount` utility with the `-t cifs` parameter:
In the `-o` parameter, you can specify options that are used to mount the share. For details, see the OPTIONS section in the mount.cifs(8) man page and Frequently used mount options.

Example 30.1. Mounting a share using an encrypted SMB 3.0 connection

To mount the `\server\example\` share as the `DOMAIN\Administrator` user over an encrypted SMB 3.0 connection into the `/mnt/` directory:

```
# mount -t cifs -o username=DOMAIN\Administrator,seal,vers=3.0 //server/example /mnt/
Password for DOMAIN\Administrator@//server_name/share_name:
```

30.4. MOUNTING AN SMB SHARE AUTOMATICALLY WHEN THE SYSTEM BOOTS

If access to a mounted SMB share is permanently required on a server, mount the share automatically at boot time.

Prerequisites

- The cifs-utils package is installed.

Procedure

To mount an SMB share automatically when the system boots, add an entry for the share to the `/etc/fstab` file. For example:

```
//server_name/share_name /mnt cifs credentials=/root/smb.cred 0 0
```

IMPORTANT

To enable the system to mount a share automatically, you must store the user name, password, and domain name in a credentials file. For details, see Authenticating to an SMB share using a credentials file.

In the fourth field of the row in the `/etc/fstab`, specify mount options, such as the path to the credentials file. For details, see the OPTIONS section in the mount.cifs(8) man page and Frequently used mount options.

To verify that the share mounts successfully, enter:

```
# mount /mnt/
```

30.5. AUTHENTICATING TO AN SMB SHARE USING A CREDENTIALS FILE

In certain situations, such as when mounting a share automatically at boot time, a share should be mounted without entering the user name and password. To implement this, create a credentials file.
Prerequisites

- The cifs-utils package is installed.

Procedure

1. Create a file, such as /root/smb.cred, and specify the user name, password, and domain name that file:

   ```
   username=user_name
   password=password
   domain=domain_name
   ```

2. Set the permissions to only allow the owner to access the file:

   ```
   # chown user_name /root/smb.cred
   # chmod 600 /root/smb.cred
   ```

You can now pass the credentials=file_name mount option to the mount utility or use it in the /etc/fstab file to mount the share without being prompted for the user name and password.

30.6. FREQUENTLY USED MOUNT OPTIONS

When you mount an SMB share, the mount options determine:

- How the connection will be established with the server. For example, which SMB protocol version is used when connecting to the server.

- How the share will be mounted into the local file system. For example, if the system overrides the remote file and directory permissions to enable multiple local users to access the content on the server.

To set multiple options in the fourth field of the /etc/fstab file or in the -o parameter of a mount command, separate them with commas. For example, see Mounting a share with the multiuser option.

The following list gives frequently used mount options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>credentials=file_name</td>
<td>Sets the path to the credentials file. See Authenticating to an SMB share using a credentials file.</td>
</tr>
<tr>
<td>dir_mode=mode</td>
<td>Sets the directory mode if the server does not support CIFS UNIX extensions.</td>
</tr>
<tr>
<td>file_mode=mode</td>
<td>Sets the file mode if the server does not support CIFS UNIX extensions.</td>
</tr>
<tr>
<td>password=password</td>
<td>Sets the password used to authenticate to the SMB server. Alternatively, specify a credentials file using the credentials option.</td>
</tr>
<tr>
<td>seal</td>
<td>Enables encryption support for connections using SMB 3.0 or a later protocol version. Therefore, use seal together with the vers mount option set to 3.0 or later. See the example in Manually mounting an SMB share.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>sec=security_mode</td>
<td>Sets the security mode, such as <code>ntlmsspi</code>, to enable NTLMv2 password hashing and enabled packet signing. For a list of supported values, see the option’s description in the <code>mount.cifs(8)</code> man page. If the server does not support the <code>ntlmv2</code> security mode, use <code>sec=ntlmssp</code>, which is the default. For security reasons, do not use the insecure <code>ntlm</code> security mode.</td>
</tr>
<tr>
<td>username=user_name</td>
<td>Sets the user name used to authenticate to the SMB server. Alternatively, specify a credentials file using the <code>credentials</code> option.</td>
</tr>
<tr>
<td>vers=SMB_protocol_version</td>
<td>Sets the SMB protocol version used for the communication with the server.</td>
</tr>
</tbody>
</table>

For a complete list, see the OPTIONS section in the `mount.cifs(8)` man page.
CHAPTER 31. OVERVIEW OF PERSISTENT NAMING ATTRIBUTES

As a system administrator, you need to refer to storage volumes using persistent naming attributes to build storage setups that are reliable over multiple system boots.

31.1. DISADVANTAGES OF NON-PERSISTENT NAMING ATTRIBUTES

Red Hat Enterprise Linux provides a number of ways to identify storage devices. It is important to use the correct option to identify each device when used in order to avoid inadvertently accessing the wrong device, particularly when installing to or reformatting drives.

Traditionally, non-persistent names in the form of `/dev/sd(major number)(minor number)` are used on Linux to refer to storage devices. The major and minor number range and associated `sd` names are allocated for each device when it is detected. This means that the association between the major and minor number range and associated `sd` names can change if the order of device detection changes.

Such a change in the ordering might occur in the following situations:

- The parallelization of the system boot process detects storage devices in a different order with each system boot.

- A disk fails to power up or respond to the SCSI controller. This results in it not being detected by the normal device probe. The disk is not accessible to the system and subsequent devices will have their major and minor number range, including the associated `sd` names shifted down. For example, if a disk normally referred to as `sdb` is not detected, a disk that is normally referred to as `sdc` would instead appear as `sdb`.

- A SCSI controller (host bus adapter, or HBA) fails to initialize, causing all disks connected to that HBA to not be detected. Any disks connected to subsequently probed HBAs are assigned different major and minor number ranges, and different associated `sd` names.

- The order of driver initialization changes if different types of HBAs are present in the system. This causes the disks connected to those HBAs to be detected in a different order. This might also occur if HBAs are moved to different PCI slots on the system.

- Disks connected to the system with Fibre Channel, iSCSI, or FCoE adapters might be inaccessible at the time the storage devices are probed, due to a storage array or intervening switch being powered off, for example. This might occur when a system reboots after a power failure, if the storage array takes longer to come online than the system takes to boot. Although some Fibre Channel drivers support a mechanism to specify a persistent SCSI target ID to WWPN mapping, this does not cause the major and minor number ranges, and the associated `sd` names to be reserved; it only provides consistent SCSI target ID numbers.

These reasons make it undesirable to use the major and minor number range or the associated `sd` names when referring to devices, such as in the `/etc/fstab` file. There is the possibility that the wrong device will be mounted and data corruption might result.

Occasionally, however, it is still necessary to refer to the `sd` names even when another mechanism is used, such as when errors are reported by a device. This is because the Linux kernel uses `sd` names (and also SCSI host/channel/target/LUN tuples) in kernel messages regarding the device.

31.2. FILE SYSTEM AND DEVICE IDENTIFIERS
This sections explains the difference between persistent attributes identifying file systems and block devices.

File system identifiers
File system identifiers are tied to a particular file system created on a block device. The identifier is also stored as part of the file system. If you copy the file system to a different device, it still carries the same file system identifier. On the other hand, if you rewrite the device, such as by formatting it with the `mkfs` utility, the device loses the attribute.

File system identifiers include:

- Unique identifier (UUID)
- Label

Device identifiers
Device identifiers are tied to a block device: for example, a disk or a partition. If you rewrite the device, such as by formatting it with the `mkfs` utility, the device keeps the attribute, because it is not stored in the file system.

Device identifiers include:

- World Wide Identifier (WWID)
- Partition UUID
- Serial number

Recommendations

- Some file systems, such as logical volumes, span multiple devices. Red Hat recommends accessing these file systems using file system identifiers rather than device identifiers.

31.3. DEVICE NAMES MANAGED BY THE UDEV MECHANISM IN /DEV/DISK/

This section lists different kinds of persistent naming attributes that the `udev` service provides in the `/dev/disk/` directory.

The `udev` mechanism is used for all types of devices in Linux, not just for storage devices. In the case of storage devices, Red Hat Enterprise Linux contains `udev` rules that create symbolic links in the `/dev/disk/` directory. This enables you to refer to storage devices by:

- Their content
- A unique identifier
- Their serial number.

Although `udev` naming attributes are persistent, in that they do not change on their own across system reboots, some are also configurable.

31.3.1. File system identifiers

The UUID attribute in `/dev/disk/by-uuid/`
Entries in this directory provide a symbolic name that refers to the storage device by a unique identifier (UUID) in the content (that is, the data) stored on the device. For example:

```
/dev/disk/by-uuid/3e6be9de-8139-11d1-9106-a43f08d823a6
```

You can use the UUID to refer to the device in the `/etc/fstab` file using the following syntax:

```
UUID=3e6be9de-8139-11d1-9106-a43f08d823a6
```

You can configure the UUID attribute when creating a file system, and you can also change it later on.

The *Label* attribute in `/dev/disk/by-label/`

Entries in this directory provide a symbolic name that refers to the storage device by a label in the content (that is, the data) stored on the device. For example:

```
/dev/disk/by-label/Boot
```

You can use the label to refer to the device in the `/etc/fstab` file using the following syntax:

```
LABEL=Boot
```

You can configure the Label attribute when creating a file system, and you can also change it later on.

### 31.3.2. Device identifiers

**The WWID attribute in `/dev/disk/by-id/`**

The World Wide Identifier (WWID) is a persistent, system-independent identifier that the SCSI Standard requires from all SCSI devices. The WWID identifier is guaranteed to be unique for every storage device, and independent of the path that is used to access the device. The identifier is a property of the device but is not stored in the content (that is, the data) on the devices.

This identifier can be obtained by issuing a SCSI Inquiry to retrieve the Device Identification Vital Product Data (page `0x83`) or Unit Serial Number (page `0x80`).

Red Hat Enterprise Linux automatically maintains the proper mapping from the WWID-based device name to a current `/dev/sd` name on that system. Applications can use the `/dev/disk/by-id/` name to reference the data on the disk, even if the path to the device changes, and even when accessing the device from different systems.

#### Example 31.1. WWID mappings

<table>
<thead>
<tr>
<th>WWID symlink</th>
<th>Non-persistent device</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/dev/disk/by-id/scsi-3600508b400105e210000900000490000</code></td>
<td><code>/dev/sda</code></td>
<td>A device with a page <code>0x83</code> identifier</td>
</tr>
<tr>
<td><code>/dev/disk/by-id/scsi-SSEAGATE_ST373453LW_3HW1RHM6</code></td>
<td><code>/dev/sdb</code></td>
<td>A device with a page <code>0x80</code> identifier</td>
</tr>
</tbody>
</table>
In addition to these persistent names provided by the system, you can also use udev rules to implement persistent names of your own, mapped to the WWID of the storage.

**The Partition UUID attribute in /dev/disk/by-partuuid**
The Partition UUID (PARTUUID) attribute identifies partitions as defined by GPT partition table.

### Example 31.2. Partition UUID mappings

<table>
<thead>
<tr>
<th>PARTUUID symlink</th>
<th>Non-persistent device</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/disk/by-partuuid/4cd1448a-01</td>
<td>/dev/sda1</td>
</tr>
<tr>
<td>/dev/disk/by-partuuid/4cd1448a-02</td>
<td>/dev/sda2</td>
</tr>
<tr>
<td>/dev/disk/by-partuuid/4cd1448a-03</td>
<td>/dev/sda3</td>
</tr>
</tbody>
</table>

**The Path attribute in /dev/disk/by-path/**
This attribute provides a symbolic name that refers to the storage device by the hardware path used to access the device.

The Path attribute fails if any part of the hardware path (for example, the PCI ID, target port, or LUN number) changes. The Path attribute is therefore unreliable. However, the Path attribute may be useful in one of the following scenarios:

- You need to identify a disk that you are planning to replace later.
- You plan to install a storage service on a disk in a specific location.

### 31.4. THE WORLD WIDE IDENTIFIER WITH DM MULTIPATH

This section describes the mapping between the World Wide Identifier (WWID) and non-persistent device names in a Device Mapper Multipath configuration.

If there are multiple paths from a system to a device, DM Multipath uses the WWID to detect this. DM Multipath then presents a single “pseudo-device” in the /dev/mapper/wwid directory, such as /dev/mapper/3600508b400105df70000e000000000ac0000.

The command `multipath -l` shows the mapping to the non-persistent identifiers:

- **Host:Channel:Target:LUN**
DM Multipath automatically maintains the proper mapping of each WWID-based device name to its corresponding /dev/sd name on the system. These names are persistent across path changes, and they are consistent when accessing the device from different systems.

When the user_friendly_names feature of DM Multipath is used, the WWID is mapped to a name of the form /dev/mapper/mpathN. By default, this mapping is maintained in the file /etc/multipath/bindings. These mpathN names are persistent as long as that file is maintained.

**IMPORTANT**

If you use user_friendly_names, then additional steps are required to obtain consistent names in a cluster.

### 31.5. LIMITATIONS OF THE UDEV DEVICE NAMING CONVENTION

The following are some limitations of the udev naming convention:

- It is possible that the device might not be accessible at the time the query is performed because the udev mechanism might rely on the ability to query the storage device when the udev rules are processed for a udev event. This is more likely to occur with Fibre Channel, iSCSI or FCoE storage devices when the device is not located in the server chassis.

- The kernel might send udev events at any time, causing the rules to be processed and possibly causing the /dev/disk/by-* links to be removed if the device is not accessible.

- There might be a delay between when the udev event is generated and when it is processed, such as when a large number of devices are detected and the user-space udevd service takes some amount of time to process the rules for each one. This might cause a delay between when the kernel detects the device and when the /dev/disk/by-* names are available.

- External programs such as blkid invoked by the rules might open the device for a brief period of time, making the device inaccessible for other uses.

- The device names managed by the udev mechanism in /dev/disk/ may change between major releases, requiring you to update the links.
31.6. LISTING PERSISTENT NAMING ATTRIBUTES

This procedure describes how to find out the persistent naming attributes of non-persistent storage devices.

Procedure

- To list the UUID and Label attributes, use the `lsblk` utility:

  ```
  $ lsblk --fs storage-device
  ```

  For example:

  **Example 31.4. Viewing the UUID and Label of a file system**

  ```
  $ lsblk --fs /dev/sda1
  NAME FSTYPE LABEL UUID MOUNTPOINT
  sda1 xfs Boot afa5d5e3-9050-48c3-acc1-bb30095f3dc4 /boot
  ```

- To list the PARTUUID attribute, use the `lsblk` utility with the `--output +PARTUUID` option:

  ```
  $ lsblk --output +PARTUUID
  ```

  For example:

  **Example 31.5. Viewing the PARTUUID attribute of a partition**

  ```
  $ lsblk --output +PARTUUID /dev/sda1
  NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT PARTUUID
  sda1  8:1    0  512M  0 part /boot 4cd1448a-01
  ```

- To list the WWID attribute, examine the targets of symbolic links in the `/dev/disk/by-id/` directory. For example:

  **Example 31.6. Viewing the WWID of all storage devices on the system**

  ```
  $ file /dev/disk/by-id/*
  /dev/disk/by-id/ata-QEMU_HARDDISK_QM00001
  symbolic link to ../../../sda
  /dev/disk/by-id/ata-QEMU_HARDDISK_QM00001-part1
  symbolic link to ../../../sda1
  /dev/disk/by-id/ata-QEMU_HARDDISK_QM00001-part2
  symbolic link to ../../../sda2
  /dev/disk/by-id/dm-name-rhel_rhel8-root
  symbolic link to ../../../dm-0
  /dev/disk/by-id/dm-name-rhel_rhel8-swap
  symbolic link to ../../../dm-1
  /dev/disk/by-id/dm-uuid-LVM-
  QIWtEHtXGobe5bewllUDivKOz5ofkgFhP0RMFsNyySVihqEl2cWWbR7MjXJoI6g
  ```
31.7. MODIFYING PERSISTENT NAMING ATTRIBUTES

This procedure describes how to change the UUID or Label persistent naming attribute of a file system.

NOTE
Changing udev attributes happens in the background and might take a long time. The udevadm settle command waits until the change is fully registered, which ensures that your next command will be able to utilize the new attribute correctly.

In the following commands:

- Replace new-uuid with the UUID you want to set; for example, `1cdfbc07-1c90-4984-b5ec-f61943f5ea50`. You can generate a UUID using the `uuidgen` command.
- Replace new-label with a label; for example, `backup_data`.

Prerequisites
- If you are modifying the attributes of an XFS file system, unmount it first.

Procedure
- To change the UUID or Label attributes of an XFS file system, use the xfs_admin utility:
  ```
  # xfs_admin -U new-uuid -L new-label storage-device
  # udevadm settle
  ```

- To change the UUID or Label attributes of an ext4, ext3, or ext2 file system, use the tune2fs utility:
  ```
  # tune2fs -U new-uuid -L new-label storage-device
  # udevadm settle
  ```

- To change the UUID or Label attributes of a swap volume, use the swaplabel utility:
  ```
  # swaplabel --uuid new-uuid --label new-label swap-device
  # udevadm settle
  ```
CHAPTER 32. GETTING STARTED WITH PARTITIONS

As a system administrator, you can use the following procedures to create, delete, and modify various types of disk partitions.

For an overview of the advantages and disadvantages to using partitions on block devices, see the following KBase article: https://access.redhat.com/solutions/163853.

32.1. VIEWING THE PARTITION TABLE

As a system administrator, you can display the partition table of a block device to see the partition layout and details about individual partitions. For an overview of the advantages and disadvantages to using partitions on block devices, see the following KBase article: https://access.redhat.com/solutions/163853.

32.1.1. Viewing the partition table with parted

This procedure describes how to view the partition table on a block device using the parted utility.

Procedure

1. Start the interactive parted shell:

   ```bash
   # parted block-device
   ```

   • Replace block-device with the path to the device you want to examine: for example, /dev/sda.

2. View the partition table:

   ```bash
   (parted) print
   ```

3. Optionally, use the following command to switch to another device you want to examine next:

   ```bash
   (parted) select block-device
   ```

Additional resources

• parted(8) man page.

32.1.2. Example output of parted print

This section provides an example output of the print command in the parted shell and describes fields in the output.

Example 32.1. Output of the print command

```
Model: ATA SAMSUNG MZNLN256 (scsi)
Disk /dev/sda: 256GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
```
Number | Start | End | Size    | Type     | File system | Flags |
-------|-------|-----|---------|----------|-------------|-------|
1      | 1049kB| 269MB| 268MB   | primary  | xfs         | boot  |
2      | 269MB | 34.6GB| 34.4GB  | primary  |             |       |
3      | 34.6GB| 45.4GB| 10.7GB  | primary  |             |       |
4      | 45.4GB| 256GB | 211GB   | extended |             |       |
5      | 45.4GB| 256GB | 211GB   | logical  |             |       |

Following is a description of the fields:

**Model:** ATA SAMSUNG MZNLN256 (scsi)

The disk type, manufacturer, model number, and interface.

**Disk /dev/sda: 256GB**

The file path to the block device and the storage capacity.

**Partition Table: msdos**

The disk label type.

**Number**

The partition number. For example, the partition with minor number 1 corresponds to /dev/sda1.

**Start and End**

The location on the device where the partition starts and ends.

**Type**

Valid types are metadata, free, primary, extended, or logical.

**File system**

The file system type. If the File system field of a device shows no value, this means that its file system type is unknown. The parted utility cannot recognize the file system on encrypted devices.

**Flags**

Lists the flags set for the partition. Available flags are boot, root, swap, hidden, raid, lvm, or lba.

### 32.2. CREATING A PARTITION TABLE ON A DISK

As a system administrator, you can format a block device with different types of partition tables to enable using partitions on the device.

**WARNING**

Formatting a block device with a partition table deletes all data stored on the device.

### 32.2.1. Considerations before modifying partitions on a disk

This section lists key points to consider before creating, removing, or resizing partitions.
NOTE

This section does not cover the DASD partition table, which is specific to the IBM Z architecture. For information on DASD, see:

- Configuring a Linux instance on IBM Z
- The What you should know about DASD article at the IBM Knowledge Center

The maximum number of partitions

The number of partitions on a device is limited by the type of the partition table:

- On a device formatted with the Master Boot Record (MBR) partition table, you can have either:
  - Up to four primary partitions, or
  - Up to three primary partitions, one extended partition, and multiple logical partitions within the extended.

- On a device formatted with the GUID Partition Table (GPT) the maximum number of partitions is 128. While the GPT specification allows for more partitions by growing the area reserved for the partition table, common practice used by the parted utility is to limit it to enough area for 128 partitions.

NOTE

Red Hat recommends that, unless you have a reason for doing otherwise, you should at least create the following partitions: swap, /boot/, and / (root).

The maximum size of a partition

The size of a partition on a device is limited by the type of the partition table:

- On a device formatted with the Master Boot Record (MBR) partition table, the maximum size is 2TiB.

- On a device formatted with the GUID Partition Table (GPT) the maximum size is 8ZiB.

If you want to create a partition larger than 2TiB, the disk must be formatted with GPT.

Size alignment

The parted utility enables you to specify partition size using multiple different suffixes:

MiB, GiB, or TiB

Size expressed in powers of 2.

- The starting point of the partition is aligned to the exact sector specified by size.
- The ending point is aligned to the specified size minus 1 sector.

MB, GB, or TB

Size expressed in powers of 10.

The starting and ending point is aligned within one half of the specified unit: for example, ±500KB when using the MB suffix.
32.2.2. Comparison of partition table types

This section compares the properties of different types of partition tables that you can create on a block device.

Table 32.1. Partition table types

<table>
<thead>
<tr>
<th>Partition table</th>
<th>Maximum number of partitions</th>
<th>Maximum partition size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master Boot Record (MBR)</td>
<td>4 primary, or 3 primary and 12 logical inside an extended partition</td>
<td>2TiB</td>
</tr>
<tr>
<td>GUID Partition Table (GPT)</td>
<td>128</td>
<td>8ZiB</td>
</tr>
</tbody>
</table>

32.2.3. MBR disk partitions

The diagrams in this chapter show the partition table as being separate from the actual disk. However, this is not entirely accurate. In reality, the partition table is stored at the very start of the disk, before any file system or user data, but for clarity, they are separate in the following diagrams.

Figure 32.1. Disk with MBR partition table

As the previous diagram shows, the partition table is divided into four sections of four primary partitions. A primary partition is a partition on a hard drive that can contain only one logical drive (or section). Each section can hold the information necessary to define a single partition, meaning that the partition table can define no more than four partitions.

Each partition table entry contains several important characteristics of the partition:

- The points on the disk where the partition starts and ends.
- Whether the partition is active. Only one partition can be flagged as active.
- The partition’s type.

The starting and ending points define the partition’s size and location on the disk. The "active" flag is used by some operating systems boot loaders. In other words, the operating system in the partition that is marked "active" is booted, in this case.

The type is a number that identifies the partition’s anticipated usage. Some operating systems use the
partition type to denote a specific file system type, to flag the partition as being associated with a particular operating system, to indicate that the partition contains a bootable operating system, or some combination of the three.

The following diagram shows an example of a drive with single partition:

**Figure 32.2. Disk with a single partition**

![Disk (MBR table)](image)

The single partition in this example is labeled as **DOS**. This label shows the partition type, with **DOS** being one of the most common ones.

### 32.2.4. Extended MBR partitions

In case four partitions are insufficient for your needs, you can use extended partitions to create up additional partitions. You can do this by setting the type of partition to "Extended".

An extended partition is like a disk drive in its own right - it has its own partition table, which points to one or more partitions (now called logical partitions, as opposed to the four primary partitions), contained entirely within the extended partition itself. The following diagram shows a disk drive with two primary partitions and one extended partition containing two logical partitions (along with some unpartitioned free space):

**Figure 32.3. Disk with both a primary and an extended MBR partition**

![Disk (MBR table)](image)

As this figure implies, there is a difference between primary and logical partitions - there can be only up
to four primary and extended partitions, but there is no fixed limit to the number of logical partitions that can exist. However, due to the way in which partitions are accessed in Linux, no more than 15 logical partitions can be defined on a single disk drive.

### 32.2.5. MBR partition types

The table below shows a list of some of the commonly used MBR partition types and hexadecimal numbers used to represent them.

<table>
<thead>
<tr>
<th>MBR partition type</th>
<th>Value</th>
<th>MBR partition type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empty</td>
<td>00</td>
<td>Novell Netware 386</td>
<td>65</td>
</tr>
<tr>
<td>DOS 12-bit FAT</td>
<td>01</td>
<td>PIC/IX</td>
<td>75</td>
</tr>
<tr>
<td>XENIX root</td>
<td>02</td>
<td>Old MINIX</td>
<td>80</td>
</tr>
<tr>
<td>XENIX usr</td>
<td>03</td>
<td>Linux/MINUX</td>
<td>81</td>
</tr>
<tr>
<td>DOS 16-bit &lt;=32M</td>
<td>04</td>
<td>Linux swap</td>
<td>82</td>
</tr>
<tr>
<td>Extended</td>
<td>05</td>
<td>Linux native</td>
<td>83</td>
</tr>
<tr>
<td>DOS 16-bit &gt;32</td>
<td>06</td>
<td>Linux extended</td>
<td>85</td>
</tr>
<tr>
<td>OS/2 HPFS</td>
<td>07</td>
<td>Amoeba</td>
<td>93</td>
</tr>
<tr>
<td>AIX</td>
<td>08</td>
<td>Amoeba BBT</td>
<td>94</td>
</tr>
<tr>
<td>AIX bootable</td>
<td>09</td>
<td>BSD/386</td>
<td>a5</td>
</tr>
<tr>
<td>OS/2 Boot Manager</td>
<td>0a</td>
<td>OpenBSD</td>
<td>a6</td>
</tr>
<tr>
<td>Win95 FAT32</td>
<td>0b</td>
<td>NEXTSTEP</td>
<td>a7</td>
</tr>
<tr>
<td>Win95 FAT32 (LBA)</td>
<td>0c</td>
<td>BSDI fs</td>
<td>b7</td>
</tr>
<tr>
<td>Win95 FAT16 (LBA)</td>
<td>0e</td>
<td>BSDI swap</td>
<td>b8</td>
</tr>
<tr>
<td>Win95 Extended (LBA)</td>
<td>0f</td>
<td>Syrinx</td>
<td>c7</td>
</tr>
<tr>
<td>Venix 80286</td>
<td>40</td>
<td>CP/M</td>
<td>db</td>
</tr>
<tr>
<td>Novell</td>
<td>51</td>
<td>DOS access</td>
<td>e1</td>
</tr>
<tr>
<td>PRep Boot</td>
<td>41</td>
<td>DOS R/O</td>
<td>e3</td>
</tr>
</tbody>
</table>
32.2.6. GUID Partition Table

The GUID Partition Table (GPT) is a partitioning scheme based on using Globally Unique Identifier (GUID). GPT was developed to cope with limitations of the MBR partition table, especially with the limited maximum addressable storage space of a disk. Unlike MBR, which is unable to address storage larger than 2 TiB (equivalent to approximately 2.2 TB), GPT is used with hard disks larger than this; the maximum addressable disk size is 2.2 ZiB. In addition, GPT, by default, supports creating up to 128 primary partitions. This number could be extended by allocating more space to the partition table.

NOTE

A GPT has partition types based on GUIDs. Note that certain partitions require a specific GUID. For example, the system partition for EFI boot loaders require GUID C12A7328-F81F-11D2-BA4B-00A0C93EC93B.

GPT disks use logical block addressing (LBA) and the partition layout is as follows:

- To preserve backward compatibility with MBR disks, the first sector (LBA 0) of GPT is reserved for MBR data and it is called "protective MBR".

- The primary GPT header begins on the second logical block (LBA 1) of the device. The header contains the disk GUID, the location of the primary partition table, the location of the secondary GPT header, and CRC32 checksums of itself, and the primary partition table. It also specifies the number of partition entries on the table.

- The primary GPT includes, by default 128 partition entries, each with an entry size of 128 bytes, its partition type GUID and unique partition GUID.

- The secondary GPT is identical to the primary GPT. It is used mainly as a backup table for recovery in case the primary partition table is corrupted.

- The secondary GPT header is located on the last logical sector of the disk and it can be used to recover GPT information in case the primary header is corrupted. It contains the disk GUID, the location of the secondary partition table and the primary GPT header, CRC32 checksums of itself and the secondary partition table, and the number of possible partition entries.
32.2.7. Creating a partition table on a disk with parted

This procedure describes how to format a block device with a partition table using the *parted* utility.

**Procedure**

1. Start the interactive *parted* shell:

   ```shell
   # parted block-device
   ```
   
   - Replace `block-device` with the path to the device where you want to create a partition table: for example, `/dev/sda`.

2. Determine if there already is a partition table on the device:

   ```shell
   (parted) print
   ```

   If the device already contains partitions, they will be deleted in the next steps.

3. Create the new partition table:

   ```shell
   (parted) mklabel table-type
   ```

   - Replace `table-type` with with the intended partition table type:
     - *msdos* for MBR
     - *gpt* for GPT

**Example 32.2. Creating a GPT table**

For example, to create a GPT table on the disk, use:
The changes start taking place as soon as you enter this command, so review it before executing it.

4. View the partition table to confirm that the partition table exists:

   (parted) print

5. Exit the parted shell:

   (parted) quit

Additional resources

- parted(8) man page.

32.3. CREATING A PARTITION

As a system administrator, you can create new partitions on a disk.

32.3.1. Considerations before modifying partitions on a disk

This section lists key points to consider before creating, removing, or resizing partitions.

NOTE

This section does not cover the DASD partition table, which is specific to the IBM Z architecture. For information on DASD, see:

- Configuring a Linux instance on IBM Z
- The What you should know about DASD article at the IBM Knowledge Center

The maximum number of partitions

The number of partitions on a device is limited by the type of the partition table:

- On a device formatted with the Master Boot Record (MBR) partition table, you can have either:
  - Up to four primary partitions, or
  - Up to three primary partitions, one extended partition, and multiple logical partitions within the extended.

- On a device formatted with the GUID Partition Table (GPT) the maximum number of partitions is 128. While the GPT specification allows for more partitions by growing the area reserved for the partition table, common practice used by the parted utility is to limit it to enough area for 128 partitions.
Red Hat recommends that, unless you have a reason for doing otherwise, you should at least create the following partitions: swap, /boot/, and / (root).

The maximum size of a partition
The size of a partition on a device is limited by the type of the partition table:

- On a device formatted with the Master Boot Record (MBR) partition table, the maximum size is 2TiB.
- On a device formatted with the GUID Partition Table (GPT) the maximum size is 8ZiB.

If you want to create a partition larger than 2TiB, the disk must be formatted with GPT.

Size alignment
The parted utility enables you to specify partition size using multiple different suffixes:

MiB, GiB, or TiB
Size expressed in powers of 2.
- The starting point of the partition is aligned to the exact sector specified by size.
- The ending point is aligned to the specified size minus 1 sector.

MB, GB, or TB
Size expressed in powers of 10.
- The starting and ending point is aligned within one half of the specified unit: for example, ±500KB when using the MB suffix.

32.3.2. Partition types
This section describes different attributes that specify the type of a partition.

Partition types or flags
The partition type, or flag, is used by a running system only rarely. However, the partition type matters to on-the-fly generators, such as systemd-gpt-auto-generator, which use the partition type to, for example, automatically identify and mount devices.

- The parted utility provides some control of partition types by mapping the partition type to flags. The parted utility can handle only certain partition types: for example LVM, swap, or RAID.
- The fdisk utility supports the full range of partition types by specifying hexadecimal codes.

Partition file system type
The parted utility optionally accepts a file system type argument when creating a partition. The value is used to:

- Set the partition flags on MBR, or
- Set the partition UUID type on GPT. For example, the swap, fat, or hfs file system types set different GUIDs. The default value is the Linux Data GUID.

The argument does not modify the file system on the partition in any way. It only differentiates between the supported flags or GUIDs.
The following file system types are supported:

- xfs
- ext2
- ext3
- ext4
- fat16
- fat32
- hfs
- hfs+
- linux-swap
- ntfs
- reiserfs

**NOTE**

The only supported local file systems in RHEL 8 are **ext4** and **xfs**.

### 32.3.3. Partition naming scheme

Red Hat Enterprise Linux uses a file-based naming scheme, with file names in the form of `/dev/xxYN`.

Device and partition names consist of the following structure:

**/dev/**

This is the name of the directory in which all device files are located. Because partitions are placed on hard disks, and hard disks are devices, the files representing all possible partitions are located in `/dev`.

**xx**

The first two letters of the partitions name indicate the type of device on which is the partition located, usually `sd`.

**y**

This letter indicates which device the partition is on. For example, `/dev/sda` for the first hard disk, `/dev/sdb` for the second, and so on. In systems with more than 26 drives, you can use more letters. For example, `/dev/sdaa1`.

**N**

The final letter indicates the number that represents the partition. The first four (primary or extended) partitions are numbered 1 through 4. Logical partitions start at 5. For example, `/dev/sda3` is the third primary or extended partition on the first hard disk, and `/dev/sdb6` is the second logical partition on the second hard disk. Drive partition numbering applies only to MBR partition tables. Note that **N** does not always mean partition.
NOTE

Even if Red Hat Enterprise Linux can identify and refer to all types of disk partitions, it might not be able to read the file system and therefore access stored data on every partition type. However, in many cases, it is possible to successfully access data on a partition dedicated to another operating system.

32.3.4. Mount points and disk partitions

In Red Hat Enterprise Linux, each partition is used to form part of the storage necessary to support a single set of files and directories. This is done using the process known as mounting, which associates a partition with a directory. Mounting a partition makes its storage available starting at the specified directory, known as a mount point.

For example, if partition /dev/sda5 is mounted on /usr/, that would mean that all files and directories under /usr/ physically reside on /dev/sda5. So the file /usr/share/doc/FAQ/txt/Linux-FAQ would be stored on /dev/sda5, while the file /etc/gdm/custom.conf would not.

Continuing the example, it is also possible that one or more directories below /usr/ would be mount points for other partitions. For instance, a partition /dev/sda7 could be mounted on /usr/local, meaning that /usr/local/man/whatatis would then reside on /dev/sda7 rather than /dev/sda5.

32.3.5. Creating a partition with parted

This procedure describes how to create a new partition on a block device using the parted utility.

Prerequisites

- There is a partition table on the disk. For details on how to format the disk, see Creating a partition table on a disk.
- If the partition you want to create is larger than 2TiB, the disk must be formatted with the GUID Partition Table (GPT).

Procedure

1. Start the interactive parted shell:

   ```bash
   # parted block-device
   ```

   - Replace block-device with the path to the device where you want to create a partition: for example, /dev/sda.

2. View the current partition table to determine if there is enough free space:

   ```bash
   (parted) print
   ```

   - If there is not enough free space, you can resize an existing partition. For more information, see Resizing a partition.

   - From the partition table, determine:
     - The start and end points of the new partition
     - On MBR, what partition type it should be.
3. Create the new partition:

    (parted) mkpart part-type name fs-type start end

- Replace `part-type` with with `primary`, `logical`, or `extended` based on what you decided from the partition table. This applies only to the MBR partition table.

- Replace `name` with an arbitrary partition name. This is required for GPT partition tables.

- Replace `fs-type` with any one of `xfs`, `ext2`, `ext3`, `ext4`, `fat16`, `fat32`, `hfs`, `hfs+`, `linux-swap`, `ntfs`, or `reiserfs`. The `fs-type` parameter is optional. Note that `parted` does not create the file system on the partition.

- Replace `start` and `end` with the sizes that determine the starting and ending points of the partition, counting from the beginning of the disk. You can use size suffixes, such as `512MiB`, `20GiB`, or `1.5TiB`. The default size megabytes.

**Example 32.3. Creating a small primary partition**

For example, to create a primary partition from 1024MiB until 2048MiB on an MBR table, use:

    (parted) mkpart primary 1024MiB 2048MiB

The changes start taking place as soon as you enter this command, so review it before executing it.

4. View the partition table to confirm that the created partition is in the partition table with the correct partition type, file system type, and size:

    (parted) print

5. Exit the `parted` shell:

    (parted) quit

6. Use the following command to wait for the system to register the new device node:

    # udevadm settle

7. Verify that the kernel recognizes the new partition:

    # cat /proc/partitions

Additional resources

- `parted(8)` man page.

32.3.6. Setting a partition type with fdisk

This procedure describes how to set a partition type, or flag, using the `fdisk` utility.
Prerequisites

- There is a partition on the disk.

Procedure

1. Start the interactive `fdisk` shell:
   
   ```
   # fdisk block-device
   ```
   
   - Replace `block-device` with the path to the device where you want to set a partition type: for example, `/dev/sda`.

2. View the current partition table to determine the minor partition number:
   
   ```
   Command (m for help): print
   ```
   
   You can see the current partition type in the Type column and its corresponding type ID in the Id column.

3. Enter the partition type command and select a partition using its minor number:
   
   ```
   Command (m for help): type
   Partition number (1,2,3 default 3): 2
   ```

4. Optionally, list the available hexadecimal codes:
   
   ```
   Hex code (type L to list all codes): L
   ```

5. Set the partition type:
   
   ```
   Hex code (type L to list all codes): 8e
   ```

6. Write your changes and exit the `fdisk` shell:
   
   ```
   Command (m for help): write
   The partition table has been altered.
   Syncing disks.
   ```

7. Verify your changes:
   
   ```
   # fdisk --list block-device
   ```

32.4. REMOVING A PARTITION

As a system administrator, you can remove a disk partition that is no longer used to free up disk space.
32.4.1. Considerations before modifying partitions on a disk

This section lists key points to consider before creating, removing, or resizing partitions.

**NOTE**

This section does not cover the DASD partition table, which is specific to the IBM Z architecture. For information on DASD, see:

- Configuring a Linux instance on IBM Z
- The What you should know about DASD article at the IBM Knowledge Center

### The maximum number of partitions

The number of partitions on a device is limited by the type of the partition table:

- On a device formatted with the **Master Boot Record (MBR)** partition table, you can have either:
  - Up to four primary partitions, or
  - Up to three primary partitions, one extended partition, and multiple logical partitions within the extended.

- On a device formatted with the **GUID Partition Table (GPT)** the maximum number of partitions is 128. While the GPT specification allows for more partitions by growing the area reserved for the partition table, common practice used by the `parted` utility is to limit it to enough area for 128 partitions.

**NOTE**

Red Hat recommends that, unless you have a reason for doing otherwise, you should at least create the following partitions: `swap`, `/boot`, and `/` (root).

### The maximum size of a partition

The size of a partition on a device is limited by the type of the partition table:

- On a device formatted with the **Master Boot Record (MBR)** partition table, the maximum size is 2TiB.

- On a device formatted with the **GUID Partition Table (GPT)** the maximum size is 8ZiB.

If you want to create a partition larger than 2TiB, the disk must be formatted with GPT.

**Size alignment**

The `parted` utility enables you to specify partition size using multiple different suffixes:

- MiB, GiB, or TiB
Size expressed in powers of 2.

- The starting point of the partition is aligned to the exact sector specified by size.
- The ending point is aligned to the specified size minus 1 sector.

**MB, GB, or TB**

Size expressed in powers of 10.
The starting and ending point is aligned within one half of the specified unit: for example, ±500KB when using the MB suffix.

### 32.4.2. Removing a partition with parted

This procedure describes how to remove a disk partition using the **parted** utility.

**Procedure**

1. Start the interactive **parted** shell:

   ```bash
   # parted block-device
   ```

   - Replace `block-device` with the path to the device where you want to remove a partition: for example, `/dev/sda`.

2. View the current partition table to determine the minor number of the partition to remove:

   ```bash
   (parted) print
   ```

3. Remove the partition:

   ```bash
   (parted) rm minor-number
   ```

   - Replace `minor-number` with the minor number of the partition you want to remove: for example, 3.

   The changes start taking place as soon as you enter this command, so review it before executing it.

4. Confirm that the partition is removed from the partition table:

   ```bash
   (parted) print
   ```

5. Exit the **parted** shell:

   ```bash
   (parted) quit
   ```

6. Verify that the kernel knows the partition is removed:

   ```bash
   # cat /proc/partitions
   ```

7. Remove the partition from the `/etc/fstab` file if it is present. Find the line that declares the removed partition, and remove it from the file.
8. Regenerate mount units so that your system registers the new `/etc/fstab` configuration:

```
# systemctl daemon-reload
```

9. If you have deleted a swap partition or removed pieces of LVM, remove all references to the partition from the kernel command line in the `/etc/default/grub` file and regenerate GRUB configuration:

- On a BIOS-based system:
  
  ```
  # grub2-mkconfig --output=/etc/grub2.cfg
  ```

- On a UEFI-based system:
  
  ```
  # grub2-mkconfig --output=/etc/grub2-efi.cfg
  ```

10. To register the changes in the early boot system, rebuild the `initramfs` file system:

    ```
    # dracut --force --verbose
    ```

Additional resources

- `parted(8)` man page

### 32.5. RESIZING A PARTITION

As a system administrator, you can extend a partition to utilize unused disk space, or shrink a partition to use its capacity for different purposes.

#### 32.5.1. Considerations before modifying partitions on a disk

This section lists key points to consider before creating, removing, or resizing partitions.

**NOTE**

This section does not cover the DASD partition table, which is specific to the IBM Z architecture. For information on DASD, see:

- Configuring a Linux instance on IBM Z
- The [What you should know about DASD](https://www.ibm.com/support/knowledgcenter/SSS7HR_2.2.0/com.ibm.zos.bms.doc/whATABS/whatshou...#DASD) article at the IBM Knowledge Center

**The maximum number of partitions**

The number of partitions on a device is limited by the type of the partition table:

- On a device formatted with the **Master Boot Record (MBR)** partition table, you can have either:
  - Up to four primary partitions, or
  - Up to three primary partitions, one extended partition, and multiple logical partitions within the extended.

- On a device formatted with the **GUID Partition Table (GPT)** the maximum number of
partitions is 128. While the GPT specification allows for more partitions by growing the area reserved for the partition table, common practice used by the parted utility is to limit it to enough area for 128 partitions.

**NOTE**

Red Hat recommends that, unless you have a reason for doing otherwise, you should at least create the following partitions: swap, /boot/, and / (root).

The maximum size of a partition
The size of a partition on a device is limited by the type of the partition table:

- On a device formatted with the **Master Boot Record (MBR)** partition table, the maximum size is 2TiB.
- On a device formatted with the **GUID Partition Table (GPT)** the maximum size is 8ZiB.

If you want to create a partition larger than 2TiB, the disk must be formatted with GPT.

Size alignment
The parted utility enables you to specify partition size using multiple different suffixes:

**MiB, GiB, or TiB**
Size expressed in powers of 2.
- The starting point of the partition is aligned to the exact sector specified by size.
- The ending point is aligned to the specified size minus 1 sector.

**MB, GB, or TB**
Size expressed in powers of 10.
The starting and ending point is aligned within one half of the specified unit: for example, ±500KB when using the MB suffix.

32.5.2. Resizing a partition with parted
This procedure resizes a disk partition using the parted utility.

**Prerequisites**
- If you want to shrink a partition, back up the data that are stored on it.

**WARNING**
Shrinking a partition might result in data loss on the partition.
If you want to resize a partition to be larger than 2TiB, the disk must be formatted with the GUID Partition Table (GPT). For details on how to format the disk, see Creating a partition table on a disk.

Procedure

1. If you want to shrink the partition, shrink the file system on it first so that it is not larger than the resized partition. Note that XFS does not support shrinking.

2. Start the interactive parted shell:

   ```
   # parted block-device
   ```

   Replace `block-device` with the path to the device where you want to resize a partition: for example, `/dev/sda`.

3. View the current partition table:

   ```
   (parted) print
   ```

   From the partition table, determine:

   - The minor number of the partition
   - The location of the existing partition and its new ending point after resizing

4. Resize the partition:

   ```
   (parted) resizepart minor-number new-end
   ```

   Replace `minor-number` with the minor number of the partition that you are resizing: for example, `3`.

   Replace `new-end` with the size that determines the new ending point of the resized partition, counting from the beginning of the disk. You can use size suffixes, such as `512MiB`, `20GiB`, or `1.5TiB`. The default size megabytes.

   **Example 32.4. Extending a partition**

   For example, to extend a partition located at the beginning of the disk to be 2GiB in size, use:

   ```
   (parted) resizepart 1 2GiB
   ```

   The changes start taking place as soon as you enter this command, so review it before executing it.

5. View the partition table to confirm that the resized partition is in the partition table with the correct size:

   ```
   (parted) print
   ```

6. Exit the parted shell:
7. Verify that the kernel recognizes the new partition:

```bash
# cat /proc/partitions
```

8. If you extended the partition, extend the file system on it as well. See (reference) for details.

Additional resources

- `parted(8)` man page.

### 32.6. STRATEGIES FOR REPARTITIONING A DISK

There are several different ways to repartition a disk. This section discusses the following possible approaches:

- Unpartitioned free space is available
- An unused partition is available
- Free space in an actively used partition is available

Note that this section discusses the previously mentioned concepts only theoretically and it does not include any procedural steps on how to perform disk repartitioning step-by-step.

#### NOTE

The following illustrations are simplified in the interest of clarity and do not reflect the exact partition layout that you encounter when actually installing Red Hat Enterprise Linux.

#### 32.6.1. Using unpartitioned free space

In this situation, the partitions that are already defined do not span the entire hard disk, leaving unallocated space that is not part of any defined partition. The following diagram shows what this might look like:

**Figure 32.5. Disk with unpartitioned free space**
In the previous example, the first diagram represents a disk with one primary partition and an undefined partition with unallocated space, and the second diagram represents a disk with two defined partitions with allocated space.

An unused hard disk also falls into this category. The only difference is that all the space is not part of any defined partition.

In any case, you can create the necessary partitions from the unused space. This scenario is mostly likely for a new disk. Most preinstalled operating systems are configured to take up all available space on a disk drive.

32.6.2. Using space from an unused partition

In this case, you can have one or more partitions that you no longer use. The following diagram illustrated such a situation.

Figure 32.6. Disk with an unused partition

In the previous example, the first diagram represents a disk with an unused partition, and the second diagram represents reallocating an unused partition for Linux.

In this situation, you can use the space allocated to the unused partition. You must delete the partition and then create the appropriate Linux partition(s) in its place. You can delete the unused partition and manually create new partitions during the installation process.

32.6.3. Using free space from an active partition

This is the most common situation. It is also the hardest to handle, because even if you have enough free space, it is presently allocated to a partition that is already in use. If you purchased a computer with preinstalled software, the hard disk most likely has one massive partition holding the operating system and data.

Aside from adding a new hard drive to your system, you can choose from destructive and non-destructive repartitioning.

32.6.3.1. Destructive repartitioning
This deletes the partition and creates several smaller ones instead. You must make a complete backup because any data in the original partition is destroyed. Create two backups, use verification (if available in your backup software), and try to read data from the backup before deleting the partition.

**WARNING**

If an operating system was installed on that partition, it must be reinstalled if you want to use that system as well. Be aware that some computers sold with pre-installed operating systems might not include the installation media to reinstall the original operating system. You should check whether this applies to your system before you destroy your original partition and its operating system installation.

After creating a smaller partition for your existing operating system, you can reinstall software, restore your data, and start your Red Hat Enterprise Linux installation.

**Figure 32.7. Destructive repartitioning action on disk**

![Diagram of disk partitioning](image)

**WARNING**

Any data previously present in the original partition is lost.

### 32.6.3.2. Non-destructive repartitioning

With non-destructive repartitioning you execute a program that makes a big partition smaller without losing any of the files stored in that partition. This method is usually reliable, but can be very time-consuming on large drives.

The non-destructive repartitioning process is straightforward and consist of three steps:

1. Compress and backup existing data
2. Resize the existing partition
3. Create new partition(s)

Each step is described further in more detail.

32.6.3.2.1. Compressing existing data

The first step is to compress the data in your existing partition. The reason for doing this is to rearrange the data to maximize the available free space at the “end” of the partition.

Figure 32.8. Compression on disk

![Disk (before) and Disk (after)](image)

In the previous example, the first diagram represents disk before compression, and the second diagram after compression.

This step is crucial. Without it, the location of the data could prevent the partition from being resized to the desired extent. Note that some data cannot be moved. In this case, it severely restricts the size of your new partitions, and you might be forced to destructively repartition your disk.

32.6.3.2.2. Resizing the existing partition

The following figure shows the actual resizing process. While the actual result of the resizing operation varies, depending on the software used, in most cases the newly freed space is used to create an unformatted partition of the same type as the original partition.
In the previous example, the first diagram represents partition before resizing, and the second diagram after resizing.

It is important to understand what the resizing software you use does with the newly freed space, so that you can take the appropriate steps. In the case illustrated here, it would be best to delete the new DOS partition and create the appropriate Linux partition or partitions.

### 32.6.3.2.3. Creating new partitions

As mentioned in the previous example, it might or might not be necessary to create new partitions. However, unless your resizing software supports systems with Linux installed, it is likely that you must delete the partition that was created during the resizing process.

In the previous example, the first diagram represents disk before configuration, and the second diagram after configuration.
CHAPTER 33. GETTING STARTED WITH XFS

This is an overview of how to create and maintain XFS file systems.

33.1. THE XFS FILE SYSTEM

XFS is a highly scalable, high-performance, robust, and mature 64-bit journaling file system that supports very large files and file systems on a single host. It is the default file system in Red Hat Enterprise Linux 8. XFS was originally developed in the early 1990s by SGI and has a long history of running on extremely large servers and storage arrays.

The features of XFS include:

Reliability

- Metadata journaling, which ensures file system integrity after a system crash by keeping a record of file system operations that can be replayed when the system is restarted and the file system remounted
- Extensive run-time metadata consistency checking
- Scalable and fast repair utilities
- Quota journaling. This avoids the need for lengthy quota consistency checks after a crash.

Scalability and performance

- Supported file system size up to 1024 TiB
- Ability to support a large number of concurrent operations
- B-tree indexing for scalability of free space management
- Sophisticated metadata read-ahead algorithms
- Optimizations for streaming video workloads

Allocation schemes

- Extent-based allocation
- Stripe-aware allocation policies
- Delayed allocation
- Space pre-allocation
- Dynamically allocated inodes

Other features

- Reflink-based file copies
- Tightly integrated backup and restore utilities
- Online defragmentation
• Online file system growing
• Comprehensive diagnostics capabilities
• Extended attributes (*xattr*). This allows the system to associate several additional name/value pairs per file.
• Project or directory quotas. This allows quota restrictions over a directory tree.
• Subsecond timestamps

**Performance characteristics**

XFS has a high performance on large systems with enterprise workloads. A large system is one with a relatively high number of CPUs, multiple HBAs, and connections to external disk arrays. XFS also performs well on smaller systems that have a multi-threaded, parallel I/O workload.

XFS has a relatively low performance for single threaded, metadata-intensive workloads: for example, a workload that creates or deletes large numbers of small files in a single thread.

### 33.2. COMPARISON OF TOOLS USED WITH EXT4 AND XFS

This section compares which tools to use to accomplish common tasks on the ext4 and XFS file systems.

<table>
<thead>
<tr>
<th>Task</th>
<th>ext4</th>
<th>XFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a file system</td>
<td>mkfs.ext4</td>
<td>mkfs.xfs</td>
</tr>
<tr>
<td>File system check</td>
<td>e2fsck</td>
<td>xfs_repair</td>
</tr>
<tr>
<td>Resize a file system</td>
<td>resize2fs</td>
<td>xfs_growfs</td>
</tr>
<tr>
<td>Save an image of a file system</td>
<td>e2image</td>
<td>xfs_metadump and xfs_mdrestore</td>
</tr>
<tr>
<td>Label or tune a file system</td>
<td>tune2fs</td>
<td>xfs_admin</td>
</tr>
<tr>
<td>Back up a file system</td>
<td>dump and restore</td>
<td>xfsdump and xfsrestore</td>
</tr>
<tr>
<td>Quota management</td>
<td>quota</td>
<td>xfs_quota</td>
</tr>
<tr>
<td>File mapping</td>
<td>filefrag</td>
<td>xfs_bmap</td>
</tr>
</tbody>
</table>
CHAPTER 34. MOUNTING FILE SYSTEMS

As a system administrator, you can mount file systems on your system to access data on them.

34.1. THE LINUX MOUNT MECHANISM

This section explains basic concepts of mounting file systems on Linux.

On Linux, UNIX, and similar operating systems, file systems on different partitions and removable devices (CDs, DVDs, or USB flash drives for example) can be attached to a certain point (the mount point) in the directory tree, and then detached again. While a file system is mounted on a directory, the original content of the directory is not accessible.

Note that Linux does not prevent you from mounting a file system to a directory with a file system already attached to it.

When mounting, you can identify the device by:

- a universally unique identifier (UUID): for example, UUID=34795a28-ca6d-4fd8-a347-73671d0c19cb
- a volume label: for example, LABEL=home
- a full path to a non-persistent block device: for example, /dev/sda3

When you mount a file system using the `mount` command without all required information, that is without the device name, the target directory, or the file system type, the `mount` utility reads the content of the `/etc/fstab` file to check if the given file system is listed there. The `/etc/fstab` file contains a list of device names and the directories in which the selected file systems are set to be mounted as well as the file system type and mount options. Therefore, when mounting a file system that is specified in `/etc/fstab`, the following command syntax is sufficient:

- Mounting by the mount point:
  ```
  # mount directory
  ```
- Mounting by the block device:
  ```
  # mount device
  ```

Additional resources

- `mount(8)` man page
- How to list persistent naming attributes such as the UUID.

34.2. LISTING CURRENTLY MOUNTED FILE SYSTEMS

This procedure describes how to list all currently mounted file systems on the command line.

Procedure

- To list all mounted file systems, use the `findmnt` utility:
$ findmnt

- To limit the listed file systems only to a certain file system type, add the `--types` option:

```
$ findmnt --types fs-type
```

For example:

```
Example 34.1. Listing only XFS file systems
```

```
$ findmnt --types xfs
```

```
TARGET  SOURCE                                                FSTYPE OPTIONS
-----------  --------  -------------------------------  ─────  ───────────────────  ───────────────  ───────────────
/          /dev/mapper/luks-5564ed00-6aac-4406-bfb4-c59bf5de48b5  xfs    rw,relatime
 /boot  /dev/sda1                                             xfs    rw,relatime
 /home /dev/mapper/luks-9d185660-7537-414d-b727-d92ea036051e  xfs    rw,relatime
```

Additional resources

- `findmnt(8)` man page

### 34.3. MOUNTING A FILE SYSTEM WITH MOUNT

This procedure describes how to mount a file system using the `mount` utility.

**Prerequisites**

- Make sure that no file system is already mounted on your chosen mount point:

  ```
  $ findmnt mount-point
  ```

**Procedure**

1. To attach a certain file system, use the `mount` utility:

  ```
  # mount device mount-point
  ```

  **Example 34.2. Mounting an XFS file system**

  For example, to mount a local XFS file system identified by UUID:

  ```
  # mount UUID=ea74bbed-536d-490c-b8d9-5b40b6d754b /mnt/data
  ```

2. If `mount` cannot recognize the file system type automatically, specify it using the `--types` option:

  ```
  # mount --types type device mount-point
  ```

  **Example 34.3. Mounting an NFS file system**
For example, to mount a remote NFS file system:

```bash
# mount --types nfs4 host:/remote-export /mnt/nfs
```

Additional resources

- `mount(8)` man page

### 34.4. MOVING A MOUNT POINT

This procedure describes how to change the mount point of a mounted file system to a different directory.

**Procedure**

1. To change the directory in which a file system is mounted:

   ```bash
   # mount --move old-directory new-directory
   ```

   **Example 34.4. Moving a home file system**

   For example, to move the file system mounted in the `/mnt/userdirs/` directory to the `/home/` mount point:

   ```bash
   # mount --move /mnt/userdirs /home
   ```

2. Verify that the file system has been moved as expected:

   ```bash
   $ findmnt
   $ ls old-directory
   $ ls new-directory
   ```

Additional resources

- `mount(8)` man page

### 34.5. UNMOUNTING A FILE SYSTEM WITH UMOUNT

This procedure describes how to unmount a file system using the `umount` utility.

**Procedure**

1. Try unmounting the file system using either of the following commands:

   - By mount point:
     
     ```bash
     # umount mount-point
     ```

   - By device:
# umount device

If the command fails with an error similar to the following, it means that the file system is in use because of a process is using resources on it:

```
rmount: /run/media/user/FlashDrive: target is busy.
```

2. If the file system is in use, use the `fuser` utility to determine which processes are accessing it. For example:

```
$ fuser --mount /run/media/user/FlashDrive
/run/media/user/FlashDrive: 18351
```

Afterwards, terminate the processes using the file system and try unmounting it again.

## 34.6. COMMON MOUNT OPTIONS

This section lists some commonly used options of the `mount` utility.

You can use these options in the following syntax:

```
# mount --options option1,option2,option3 device mount-point
```

Table 34.1. Common mount options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Enables asynchronous input and output operations on the file system.</td>
</tr>
<tr>
<td>auto</td>
<td>Enables the file system to be mounted automatically using the <code>mount -a</code> command.</td>
</tr>
<tr>
<td>defaults</td>
<td>Provides an alias for the <code>async,auto,dev,exec,nouser,rw,suid</code> options.</td>
</tr>
<tr>
<td>exec</td>
<td>Allows the execution of binary files on the particular file system.</td>
</tr>
<tr>
<td>loop</td>
<td>Mounts an image as a loop device.</td>
</tr>
<tr>
<td>noauto</td>
<td>Default behavior disables the automatic mount of the file system using the <code>mount -a</code> command.</td>
</tr>
<tr>
<td>noexec</td>
<td>Disallows the execution of binary files on the particular file system.</td>
</tr>
<tr>
<td>nouser</td>
<td>Disallows an ordinary user (that is, other than root) to mount and unmount the file system.</td>
</tr>
<tr>
<td>remount</td>
<td>Remounts the file system in case it is already mounted.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>ro</td>
<td>Mounts the file system for reading only.</td>
</tr>
<tr>
<td>rw</td>
<td>Mounts the file system for both reading and writing.</td>
</tr>
<tr>
<td>user</td>
<td>Allows an ordinary user (that is, other than root) to mount and unmount the file system.</td>
</tr>
</tbody>
</table>
CHAPTER 35. SHARING A MOUNT ON MULTIPLE MOUNT POINTS

As a system administrator, you can duplicate mount points to make the file systems accessible from multiple directories.

35.1. TYPES OF SHARED MOUNTS

There are multiple types of shared mounts that you can use. The difference between them is what happens when you mount another file system under one of the shared mount points. The shared mounts are implemented using the *shared subtrees* functionality.

The following mount types are available:

**private**

This type does not receive or forward any propagation events. When you mount another file system under either the duplicate or the original mount point, it is not reflected in the other.

**shared**

This type creates an exact replica of a given mount point. When a mount point is marked as a *shared* mount, any mount within the original mount point is reflected in it, and vice versa.

This is the default mount type of the root file system.

**slave**

This type creates a limited duplicate of a given mount point. When a mount point is marked as a *slave* mount, any mount within the original mount point is reflected in it, but no mount within a *slave* mount is reflected in its original.

**unbindable**

This type prevents the given mount point from being duplicated whatsoever.

Additional resources

- The *Shared subtrees* article on Linux Weekly News.

35.2. CREATING A PRIVATE MOUNT POINT DUPLICATE

This procedure duplicates a mount point as a private mount. File systems that you later mount under the duplicate or the original mount point are not reflected in the other.

Procedure

1. Create a virtual file system (VFS) node from the original mount point:

   ```bash
   # mount --bind original-dir original-dir
   ```

2. Mark the original mount point as private:
Alternatively, to change the mount type for the selected mount point and all mount points under it, use the `--make-private` option instead of `--make-private`.

### 3. Create the duplicate:

```bash
# mount --bind original-dir duplicate-dir
```

#### Example 35.1. Duplicating /media into /mnt as a private mount point

1. Create a VFS node from the `/media` directory:
   ```bash
   # mount --bind /media /media
   ```

2. Mark the `/media` directory as private:
   ```bash
   # mount --make-private /media
   ```

3. Create its duplicate in `/mnt`:
   ```bash
   # mount --bind /media /mnt
   ```

4. It is now possible to verify that `/media` and `/mnt` share content but none of the mounts within `/media` appear in `/mnt`. For example, if the CD-ROM drive contains non-empty media and the `/media/cdrom` directory exists, use:
   ```bash
   # mount /dev/cdrom /media/cdrom
   # ls /media/cdrom
   # ls /mnt/cdrom
   #
   ```

5. It is also possible to verify that file systems mounted in the `/mnt` directory are not reflected in `/media`. For instance, if a non-empty USB flash drive that uses the `/dev/sdc1` device is plugged in and the `/mnt/flashdisk` directory is present, use:
   ```bash
   # mount /dev/sdc1 /mnt/flashdisk
   # ls /media/flashdisk
   # ls /mnt/flashdisk
   # ls /mnt/flashdisk
   en-US  publican.cfg
   #
   ```

### Additional resources

- `mount(8)` man page

### 35.3. Creating a Shared Mount Point Duplicate

This procedure duplicates a mount point as a shared mount. File systems that you later mount under the original directory or the duplicate are always reflected in the other.
Procedure

1. Create a virtual file system (VFS) node from the original mount point:
   ```
   # mount --bind original-dir original-dir
   ```

2. Mark the original mount point as shared:
   ```
   # mount --make-shared original-dir
   ```
   Alternatively, to change the mount type for the selected mount point and all mount points under it, use the `--make-rshared` option instead of `--make-shared`.

3. Create the duplicate:
   ```
   # mount --bind original-dir duplicate-dir
   ```

Example 35.2. Duplicating /media into /mnt as a shared mount point

To make the `/media` and `/mnt` directories share the same content:

1. Create a VFS node from the `/media` directory:
   ```
   # mount --bind /media /media
   ```

2. Mark the `/media` directory as shared:
   ```
   # mount --make-shared /media
   ```

3. Create its duplicate in `/mnt`:
   ```
   # mount --bind /media /mnt
   ```

4. It is now possible to verify that a mount within `/media` also appears in `/mnt`. For example, if the CD-ROM drive contains non-empty media and the `/media/cdrom/` directory exists, use:
   ```
   # mount /dev/cdrom /media/cdrom
   # ls /media/cdrom
   EFI  GPL  isolinux  LiveOS
   # ls /mnt/cdrom
   EFI  GPL  isolinux  LiveOS
   ```

5. Similarly, it is possible to verify that any file system mounted in the `/mnt` directory is reflected in `/media`. For instance, if a non-empty USB flash drive that uses the `/dev/sdc1` device is plugged in and the `/mnt/flashdisk/` directory is present, use:
   ```
   # mount /dev/sdc1 /mnt/flashdisk
   # ls /media/flashdisk
   en-US  publican.cfg
   # ls /mnt/flashdisk
   en-US  publican.cfg
   ```
**35.4. CREATING A SLAVE MOUNT POINT DUPLICATE**

This procedure duplicates a mount point as a slave mount type. File systems that you later mount under the original mount point are reflected in the duplicate but not the other way around.

**Procedure**

1. Create a virtual file system (VFS) node from the original mount point:

   ```
   # mount --bind original-dir original-dir
   ```

2. Mark the original mount point as shared:

   ```
   # mount --make-shared original-dir
   ```

   Alternatively, to change the mount type for the selected mount point and all mount points under it, use the `--make-rshared` option instead of `--make-shared`.

3. Create the duplicate and mark it as the slave type:

   ```
   # mount --bind original-dir duplicate-dir
   # mount --make-slave duplicate-dir
   ```

**Example 35.3. Duplicating /media into /mnt as a slave mount point**

This example shows how to get the content of the `/media` directory to appear in `/mnt` as well, but without any mounts in the `/mnt` directory to be reflected in `/media`.

1. Create a VFS node from the `/media` directory:

   ```
   # mount --bind /media /media
   ```

2. Mark the `/media` directory as shared:

   ```
   # mount --make-shared /media
   ```

3. Create its duplicate in `/mnt` and mark it as slave:

   ```
   # mount --bind /media /mnt
   # mount --make-slave /mnt
   ```

4. Verify that a mount within `/media` also appears in `/mnt`. For example, if the CD-ROM drive contains non-empty media and the `/media/cdrom/` directory exists, use:

   ```
   # mount /dev/cdrom /media/cdrom
   # ls /media/cdrom
   EFI GPL isolinux LiveOS
   # ls /mnt/cdrom
   EFI GPL isolinux LiveOS
   ```
5. Also verify that file systems mounted in the /mnt directory are not reflected in /media. For instance, if a non-empty USB flash drive that uses the /dev/sdc1 device is plugged in and the /mnt/flashdisk/ directory is present, use:

```
# mount /dev/sdc1 /mnt/flashdisk
# ls /media/flashdisk
# ls /mnt/flashdisk
```

en-US  publican.cfg

Additional resources

- `mount(8)` man page

### 35.5. PREVENTING A MOUNT POINT FROM BEING DUPLICATED

This procedure marks a mount point as unbindable so that it is not possible to duplicate it in another mount point.

**Procedure**

- To change the type of a mount point to an unbindable mount, use:

```
# mount --bind mount-point mount-point
# mount --make-unbindable mount-point
```

Alternatively, to change the mount type for the selected mount point and all mount points under it, use the `--make-runbindable` option instead of `--make-unbindable`.

Any subsequent attempt to make a duplicate of this mount fails with the following error:

```
# mount --bind mount-point duplicate-dir
```

mount: wrong fs type, bad option, bad superblock on mount-point,
missing codepage or helper program, or other error
In some cases useful info is found in syslog - try
dmesg | tail or so

**Example 35.4. Preventing /media from being duplicated**

- To prevent the /media directory from being shared, use:

```
# mount --bind /media /media
# mount --make-unbindable /media
```

Additional resources

- `mount(8)` man page
CHAPTER 36. PERSISTENTLY MOUNTING FILE SYSTEMS

As a system administrator, you can persistently mount file systems to configure non-removable storage.

36.1. THE /ETC/FSTAB FILE

This section describes the /etc/fstab configuration file, which controls persistent mount points of file systems. Using /etc/fstab is the recommended way to persistently mount file systems.

Each line in the /etc/fstab file defines a mount point of a file system. It includes six fields separated by white space:

1. The block device identified by a persistent attribute or a path in the /dev directory.
2. The directory where the device will be mounted.
3. The file system on the device.
4. Mount options for the file system. The option defaults means that the partition is mounted at boot time with default options. This section also recognizes systemd mount unit options in the x-systemd.option format.
5. Backup option for the dump utility.
6. Check order for the fsck utility.

Example 36.1. The /boot file system in /etc/fstab

<table>
<thead>
<tr>
<th>Block device</th>
<th>Mount point</th>
<th>File system</th>
<th>Options</th>
<th>Backup</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>UUID=ea74bbec-536d-490c-b8d9-5b40bbd7545b</td>
<td>/boot</td>
<td>xfs</td>
<td>defaults</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The systemd service automatically generates mount units from entries in /etc/fstab.

Additional resources
- fstab(5) man page
- systemd.mount(5) man page

36.2. ADDING A FILE SYSTEM TO /ETC/FSTAB

This procedure describes how to configure persistent mount point for a file system in the /etc/fstab configuration file.

Procedure

1. Find out the UUID attribute of the file system:
$ lsblk --fs storage-device

For example:

### Example 36.2. Viewing the UUID of a partition

```
$ lsblk --fs /dev/sda1
NAME  FSTYPE  LABEL        UUID                        MOUNTPOINT
sda1  xfs    Boot          ea74bbec-536d-490c-b8d9-5b40bbd7545b /boot
```

2. If the mount point directory does not exist, create it:

```
# mkdir --parents mount-point
```

3. As root, edit the `/etc/fstab` file and add a line for the file system, identified by the UUID. For example:

### Example 36.3. The /boot mount point in /etc/fstab

```
UUID=ea74bbec-536d-490c-b8d9-5b40bbd7545b /boot xfs defaults 0 0
```

4. Regenerate mount units so that your system registers the new configuration:

```
# systemctl daemon-reload
```

5. Try mounting the file system to verify that the configuration works:

```
# mount mount-point
```

**Additional resources**

- [Overview of persistent naming attributes](#)
CHAPTER 37. PERSISTENTLY MOUNTING A FILE SYSTEM USING RHEL SYSTEM ROLES

This section describes how to persistently mount a file system using the `storage` role.

Prerequisites

- An Ansible playbook that uses the `storage` role exists. For information on how to apply such a playbook, see Applying a role.

37.1. EXAMPLE ANSIBLE PLAYBOOK TO PERSISTENTLY MOUNT A FILE SYSTEM

This section provides an example Ansible playbook. This playbook applies the `storage` role to immediately and persistently mount an XFS file system.

Example 37.1. A playbook that mounts a file system on `/dev/sdb` to `/mnt/data`

```yaml
---
- hosts: all
  vars:
    storage_volumes:
      - name: barefs
        type: disk
        disks:
          - sdb
        fs_type: xfs
        mount_point: /mnt/data
    roles:
      - rhel-system-roles.storage

  - hosts: all
    vars:
      storage_volumes:
        - name: barefs
          type: disk
          disks:
            - sdb
          fs_type: xfs
          mount_point: /mnt/data
      roles:
        - rhel-system-roles.storage

  - hosts: all
    vars:
      storage_volumes:
        - name: barefs
          type: disk
          disks:
            - sdb
          fs_type: xfs
          mount_point: /mnt/data
      roles:
        - rhel-system-roles.storage
```

- This playbook adds the file system to the `/etc/fstab` file, and mounts the file system immediately.
- If the file system on the `/dev/sdb` device or the mount point directory do not exist, the playbook creates them.

Additional resources

- The `/usr/share/ansible/roles/rhel-system-roles.storage/README.md` file.
CHAPTER 38. MOUNTING FILE SYSTEMS ON DEMAND

As a system administrator, you can configure file systems, such as NFS, to mount automatically on demand.

38.1. THE AUTOFS SERVICE

This section explains the benefits and basic concepts of the autofs service, used to mount file systems on demand.

One drawback of permanent mounting using the /etc/fstab configuration is that, regardless of how infrequently a user accesses the mounted file system, the system must dedicate resources to keep the mounted file system in place. This might affect system performance when, for example, the system is maintaining NFS mounts to many systems at one time.

An alternative to /etc/fstab is to use the kernel-based autofs service. It consists of the following components:

- A kernel module that implements a file system, and
- A user-space service that performs all of the other functions.

The autofs service can mount and unmount file systems automatically (on-demand), therefore saving system resources. It can be used to mount file systems such as NFS, AFS, SMBFS, CIFS, and local file systems.

Additional resources

- The autofs(8) man page.

38.2. THE AUTOFS CONFIGURATION FILES

This section describes the usage and syntax of configuration files used by the autofs service.

The master map file

The autofs service uses /etc/auto.master (master map) as its default primary configuration file. This can be changed to use another supported network source and name using the autofs configuration in the /etc/autofs.conf configuration file in conjunction with the Name Service Switch (NSS) mechanism.

All on-demand mount points must be configured in the master map. Mount point, host name, exported directory, and options can all be specified in a set of files (or other supported network sources) rather than configuring them manually for each host.

The master map file lists mount points controlled by autofs, and their corresponding configuration files or network sources known as automount maps. The format of the master map is as follows:

```
mount-point map-name options
```

The variables used in this format are:

- `mount-point`  
  The autofs mount point; for example, /mnt/data.

- `map-file`
The map source file, which contains a list of mount points and the file system location from which those mount points should be mounted.

options
If supplied, these apply to all entries in the given map, if they do not themselves have options specified.

Example 38.1. The /etc/auto.master file
The following is a sample line from /etc/auto.master file:

```
/mnt/data /etc/auto.data
```

Map files
Map files configure the properties of individual on-demand mount points.

The automounter creates the directories if they do not exist. If the directories exist before the automounter was started, the automounter will not remove them when it exits. If a timeout is specified, the directory is automatically unmounted if the directory is not accessed for the timeout period.

The general format of maps is similar to the master map. However, the options field appears between the mount point and the location instead of at the end of the entry as in the master map:

```
mount-point options location
```

The variables used in this format are:

mount-point
This refers to the autofs mount point. This can be a single directory name for an indirect mount or the full path of the mount point for direct mounts. Each direct and indirect map entry key (mount-point) can be followed by a space separated list of offset directories (subdirectory names each beginning with /) making them what is known as a multi-mount entry.

options
When supplied, these options are appended to the master map entry options, if any, or used instead of the master map options if the configuration entry append_options is set to no.

location
This refers to the file system location such as a local file system path (preceded with the Sun map format escape character : for map names beginning with /), an NFS file system or other valid file system location.

Example 38.2. A map file
The following is a sample from a map file; for example, /etc/auto.misc:

```
payroll -fstype=nfs4 personnel:/exports/payroll
sales -fstype=xfs :/dev/hda4
```

The first column in the map file indicates the autofs mount point: sales and payroll from the server called personnel. The second column indicates the options for the autofs mount. The third column indicates the source of the mount.
Following the given configuration, the `autofs` mount points will be `/home/payroll` and `/home/sales`. The `-fstype=` option is often omitted and is not needed if the file system is NFS, including mounts for NFSv4 if the system default is NFSv4 for NFS mounts.

Using the given configuration, if a process requires access to an `autofs` unmounted directory such as `/home/payroll/2006/July.sxc`, the `autofs` service automatically mounts the directory.

The amd map format

The `autofs` service recognizes map configuration in the `amd` format as well. This is useful if you want to reuse existing automounter configuration written for the `am-utils` service, which has been removed from Red Hat Enterprise Linux.

However, Red Hat recommends using the simpler `autofs` format described in the previous sections.

Additional resources

- `autofs(5)` man page
- `autofs.conf(5)` man page
- `auto.master(5)` man page
- `/usr/share/doc/autofs/README.amd-maps` file

38.3. CONFIGURING AUTOFS MOUNT POINTS

This procedure describes how to configure on-demand mount points using the `autofs` service.

Prerequisites

- Install the `autofs` package:
  
  ```
  # yum install autofs
  ```

- Start and enable the `autofs` service:
  
  ```
  # systemctl enable --now autofs
  ```

Procedure

1. Create a map file for the on-demand mount point, located at `/etc/auto.identifier`. Replace `identifier` with a name that identifies the mount point.

2. In the map file, fill in the mount point, options, and location fields as described in The autofs configuration files section.

3. Register the map file in the master map file, as described in The autofs configuration files section.

4. Allow the service to re-read the configuration, so it can manage the newly configured `autofs` mount:

  ```
  # systemctl reload autofs.service
  ```
5. Try accessing content in the on-demand directory:

```bash
# Is automouted-directory
```

### 38.4. AUTOMOUNTING NFS SERVER USER HOME DIRECTORIES WITH AUTOFS SERVICE

This procedure describes how to configure the `autofs` service to mount user home directories automatically.

**Prerequisites**

- The `autofs` package is installed.
- The `autofs` service is enabled and running.

**Procedure**

1. Specify the mount point and location of the map file by editing the `/etc/auto.master` file on a server on which you need to mount user home directories. To do so, add the following line into the `/etc/auto.master` file:

   ```
   /home /etc/auto.home
   ```

2. Create a map file with the name of `/etc/auto.home` on a server on which you need to mount user home directories, and edit the file with the following parameters:

   ```
   * -fstype=nfs,rw,sync host.example.com:/home/
   ```

   You can skip `fstype` parameter, as it is `nfs` by default. For more information, see `autofs(5)` man page.

3. Reload the `autofs` service:

   ```
   # systemctl reload autofs
   ```

### 38.5. OVERRIDING OR AUGMENTING AUTOFS SITE CONFIGURATION FILES

It is sometimes useful to override site defaults for a specific mount point on a client system.

**Example 38.3. Initial conditions**

For example, consider the following conditions:

- Automounter maps are stored in NIS and the `/etc/nsswitch.conf` file has the following directive:

  ```
  automount:   files nis
  ```

- The `auto.master` file contains:
The NIS auto.master map file contains:

/home auto.home

The NIS auto.home map contains:

- beth  fileserver.example.com:/export/home/beth
- joe  fileserver.example.com:/export/home/joe
- * fileserver.example.com:/export/home/

The autos configuration option BROWSE_MODE is set to yes:

BROWSE_MODE="yes"

The file map /etc/auto.home does not exist.

Procedure

This section describes the examples of mounting home directories from a different server and augmenting auto.home with only selected entries.

Example 38.4. Mounting home directories from a different server

Given the preceding conditions, let’s assume that the client system needs to override the NIS map auto.home and mount home directories from a different server.

- In this case, the client needs to use the following /etc/auto.master map:

  /home /etc/auto.home
  +auto.master

- The /etc/auto.home map contains the entry:

  * host.example.com:/export/home/

Because the automounter only processes the first occurrence of a mount point, the /home directory contains the content of /etc/auto.home instead of the NIS auto.home map.

Example 38.5. Augmenting auto.home with only selected entries

Alternatively, to augment the site-wide auto.home map with just a few entries:

1. Create an /etc/auto.home file map, and in it put the new entries. At the end, include the NIS auto.home map. Then the /etc/auto.home file map looks similar to:

   mydir someserver:/export/mydir
   +auto.home
2. With these NIS auto.home map conditions, listing the content of the /home directory outputs:

   $ ls /home
   beth joe mydir

   This last example works as expected because autofs does not include the contents of a file map of the same name as the one it is reading. As such, autofs moves on to the next map source in the nsswitch configuration.

38.6. USING LDAP TO STORE AUTOMONTER MAPS

This procedure configures autofs to store automounter maps in LDAP configuration rather than in autofs map files.

Prerequisites

- LDAP client libraries must be installed on all systems configured to retrieve automounter maps from LDAP. On Red Hat Enterprise Linux, the openldap package should be installed automatically as a dependency of the autofs package.

Procedure

1. To configure LDAP access, modify the /etc/openldap/ldap.conf file. Ensure that the BASE, URI, and schema options are set appropriately for your site.

2. The most recently established schema for storing automount maps in LDAP is described by the rfc2307bis draft. To use this schema, set it in the /etc/autofs.conf configuration file by removing the comment characters from the schema definition. For example:

   Example 38.6. Setting autofs configuration

   ```
   DEFAULT_MAP_OBJECT_CLASS="automountMap"
   DEFAULT_ENTRY_OBJECT_CLASS="automount"
   DEFAULT_MAP_ATTRIBUTE="automountMapName"
   DEFAULT_ENTRY_ATTRIBUTE="automountKey"
   DEFAULT_VALUE_ATTRIBUTE="automountInformation"
   ```

3. Ensure that all other schema entries are commented in the configuration. The automountKey attribute of the rfc2307bis schema replaces the cn attribute of the rfc2307 schema. Following is an example of an LDAP Data Interchange Format (LDIF) configuration:

   Example 38.7. LDIF Configuration

   ```
   # auto.master, example.com
   dn: automountMapName=auto.master,dc=example,dc=com
   objectClass: top
   objectClass: automountMap
   automountMapName: auto.master

   # /home, auto.master, example.com
   dn: automountMapName=auto.master,dc=example,dc=com
   ```
Additional resources

- The rfc2307bis draft

38.7. USING SYSTEMD.AUTOMOUNT TO MOUNT A FILE SYSTEM ON DEMAND WITH /ETC/FSTAB

This procedure shows how to mount a file system on demand using the automount systemd units when mount point is defined in /etc/fstab. You have to add an automount unit for each mount and enable it.

Procedure

1. Add desired fstab entry as documented in Chapter 30. Persistently mounting file systems. For example:

   ```
   /dev/disk/by-id/da875760-edb9-4b82-99dc-5f4b1ff2e5f4  /mount/point  xfs  defaults  0 0
   ```

2. Add `x-systemd.automount` to the options field of entry created in the previous step.

3. Load newly created units so that your system registers the new configuration:

   ```
   # systemctl daemon-reload
   ```

4. Start the automount unit:

   ```
   # systemctl start mount-point.automount
   ```

Verification

1. Check that `mount-point.automount` is running:
38.8. USING SYSTEMD.AUTOMOUNT TO MOUNT A FILE SYSTEM ON DEMAND WITH A MOUNT UNIT

This procedure shows how to mount a file system on demand using the automount systemd units when mount point is defined by a mount unit. You have to add an automount unit for each mount and enable it.

Procedure

1. Create a mount unit. For example:
   ```
   [Mount]
   What=/dev/disk/by-uuid/f5755511-a714-44c1-a123-cfde0e4ac688
   Where=/mount/point
   Type=xfs
   ```

2. Create a unit file with the same name as the mount unit, but with extension .automount.

3. Open the file and create an [Automount] section. Set the Where= option to the mount path:
   ```
   [Automount]
   Where=/mount/point
   [Install]
   WantedBy=multi-user.target
   ```

4. Load newly created units so that your system registers the new configuration:
   ```
   # systemctl daemon-reload
   ```

5. Enable and start the automount unit instead:
   ```
   # systemctl enable --now mount-point.automount
   ```

Verification

1. Check that mount-point.automount is running:
# systemctl status mount-point.automount

2. Check that automounted directory has desired content:

   # ls /mount/point

Additional resources

- `systemd.automount(5)` man page.
- `systemd.mount(5)` man page.
- Introduction to systemd.
CHAPTER 39. USING SSSD COMPONENT FROM IDM TO CACHE THE AUTOFS MAPS

The System Security Services Daemon (SSSD) is a system service to access remote service directories and authentication mechanisms. The data caching is useful in case of the slow network connection. To configure the SSSD service to cache the autofs map, follow the procedures below in this section.

39.1. CONFIGURING AUTOFS MANUALLY TO USE IDM SERVER AS AN LDAP SERVER

This procedure shows how to configure autofs to use IdM server as an LDAP server.

Procedure

1. Edit the /etc/autofs.conf file to specify the schema attributes that autofs searches for:

   ```
   # Other common LDAP naming
   map_object_class = "automountMap"
   entry_object_class = "automount"
   map_attribute = "automountMapName"
   entry_attribute = "automountKey"
   value_attribute = "automountInformation"
   ```

   **NOTE**
   
   User can write the attributes in both lower and upper cases in the /etc/autofs.conf file.

2. Optionally, specify the LDAP configuration. There are two ways to do this. The simplest is to let the automount service discover the LDAP server and locations on its own:

   ```
   ldap_uri = "ldap:///dc=example,dc=com"
   ```

   This option requires DNS to contain SRV records for the discoverable servers.

   Alternatively, explicitly set which LDAP server to use and the base DN for LDAP searches:

   ```
   ldap_uri = "ldap://ipa.example.com"
   search_base = "cn=location,cn=automount,dc=example,dc=com"
   ```

3. Edit the /etc/autofs_ldap_auth.conf file so that autofs allows client authentication with the IdM LDAP server.

   - Change `authrequired` to yes.
   - Set the principal to the Kerberos host principal for the IdM LDAP server, `host/fqdn@REALM`. The principal name is used to connect to the IdM directory as part of GSS client authentication.

   ```
   <autofs_ldap_sasl_conf
   ```

Red Hat Enterprise Linux 8 System Design Guide

670
39.2. CONFIGURING SSSD TO CACHE AUTOFS MAPS

The SSSD service can be used to cache autofs maps stored on an IdM server without having to configure autofs to use the IdM server at all.

Prerequisites

- The sssd package is installed.

Procedure

1. Open the SSSD configuration file:

   ```
   # vim /etc/sssd/sssd.conf
   ```

2. Add the autofs service to the list of services handled by SSSD.

   ```
   [sssd]
   domains = ldap
   services = nss,pam,autofs
   ```

3. Create a new [autofs] section. You can leave this blank, because the default settings for an autofs service work with most infrastructures.

   ```
   [nss]
   [pam]
   [sudo]
   [autofs]
   [ssh]
   [pac]
   ```

   For more information, see the sssd.conf man page.

4. Optionally, set a search base for the autofs entries. By default, this is the LDAP search base, but a subtree can be specified in the ldap_autofs_search_base parameter.

   ```
   [domain/EXAMPLE]
   ```

For more information about host principal, see Using canonicalized DNS host names in IdM.
ldap_search_base = "dc=example,dc=com"
ldap_autofs_search_base = "ou=automount,dc=example,dc=com"

5. Restart SSSD service:

   # systemctl restart sssd.service

6. Check the /etc/nsswitch.conf file, so that SSSD is listed as a source for automount configuration:

   automount: sss files

7. Restart autos service:

   # systemctl restart autofs.service

8. Test the configuration by listing a user’s /home directory, assuming there is a master map entry for /home:

   # ls /home/username

If this does not mount the remote file system, check the /var/log/messages file for errors. If necessary, increase the debug level in the /etc/sysconfig/autofs file by setting the logging parameter to debug.
CHAPTER 40. SETTING READ-ONLY PERMISSIONS FOR THE ROOT FILE SYSTEM

Sometimes, you need to mount the root file system (/) with read-only permissions. Example use cases include enhancing security or ensuring data integrity after an unexpected system power-off.

40.1. FILES AND DIRECTORIES THAT ALWAYS RETAIN WRITE PERMISSIONS

For the system to function properly, some files and directories need to retain write permissions. When the root file system is mounted in read-only mode, these files are mounted in RAM using the tmpfs temporary file system.

The default set of such files and directories is read from the /etc/rwtab file. Note that the readonly-root package is required to have this file present in your system.

```
dirs /var/cache/man
dirs /var/gdm
<content truncated>
empty /tmp
empty /var/cache/foomatic
<content truncated>
files /etc/adjtime
files /etc/ntp.conf
<content truncated>
```

Entries in the /etc/rwtab file follow this format:

```
copy-method  path
```

In this syntax:

- Replace `copy-method` with one of the keywords specifying how the file or directory is copied to tmpfs.
- Replace `path` with the path to the file or directory.

The /etc/rwtab file recognizes the following ways in which a file or directory can be copied to tmpfs:

- **empty**
  
  An empty path is copied to tmpfs. For example:
  
  ```
  empty /tmp
  ```

- **dirs**
  
  A directory tree is copied to tmpfs, empty. For example:
  
  ```
  dirs /var/run
  ```

- **files**

673
A file or a directory tree is copied to **tmpfs** intact. For example:

```
files /etc/resolv.conf
```

The same format applies when adding custom paths to `/etc/rwtab.d/`.

### 40.2. Configuring the Root File System to Mount with Read-Only Permissions on Boot

With this procedure, the root file system is mounted read-only on all following boots.

**Procedure**

1. In the `/etc/sysconfig/readonly-root` file, set the **READONLY** option to **yes**:

   ```
   # Set to 'yes' to mount the file systems as read-only.
   READONLY=yes
   ```

2. Add the **ro** option in the root entry (/) in the `/etc/fstab` file:

   ```
   /dev/mapper/luks-c376919e...  /  xfs  x-systemd.device-timeout=0,ro 1 1
   ```

3. Add the **ro** option to the **GRUB_CMDLINE_LINUX** directive in the `/etc/default/grub` file and ensure that the directive does not contain **rw**:

   ```
   GRUB_CMDLINE_LINUX="rhgb quiet... ro"
   ```

4. Recreate the GRUB2 configuration file:

   ```
   # grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

5. If you need to add files and directories to be mounted with write permissions in the **tmpfs** file system, create a text file in the `/etc/rwtab.d/` directory and put the configuration there. For example, to mount the `/etc/example/file` file with write permissions, add this line to the `/etc/rwtab.d/example` file:

   ```
   files /etc/example/file
   ```

   **IMPORTANT**

   Changes made to files and directories in **tmpfs** do not persist across boots.

6. Reboot the system to apply the changes.

**Troubleshooting**

- If you mount the root file system with read-only permissions by mistake, you can remount it with read-and-write permissions again using the following command:

  ```
  # mount -o remount,rw /
  ```
CHAPTER 41. MANAGING STORAGE DEVICES

41.1. SETTING UP STRATIS FILE SYSTEMS

Stratis runs as a service to manage pools of physical storage devices, simplifying local storage management with ease of use while helping you set up and manage complex storage configurations.

IMPORTANT

Stratis is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see https://access.redhat.com/support/offerings/techpreview.

41.1.1. What is Stratis

Stratis is a local storage-management solution for Linux. It is focused on simplicity and ease of use, and gives you access to advanced storage features.

Stratis makes the following activities easier:

- Initial configuration of storage
- Making changes later
- Using advanced storage features

Stratis is a hybrid user-and-kernel local storage management system that supports advanced storage features. The central concept of Stratis is a storage pool. This pool is created from one or more local disks or partitions, and volumes are created from the pool.

The pool enables many useful features, such as:

- File system snapshots
- Thin provisioning
- Tiering

Additional resources

- Stratis website

41.1.2. Components of a Stratis volume

Learn about the components that comprise a Stratis volume.

Externally, Stratis presents the following volume components in the command-line interface and the API:

blockdev
Block devices, such as a disk or a disk partition.

**pool**

Composed of one or more block devices.

A pool has a fixed total size, equal to the size of the block devices.

The pool contains most Stratis layers, such as the non-volatile data cache using the `dm-cache` target.

Stratis creates a `/dev/stratis/my-pool` directory for each pool. This directory contains links to devices that represent Stratis file systems in the pool.

**filesystem**

Each pool can contain one or more file systems, which store files.

File systems are thinly provisioned and do not have a fixed total size. The actual size of a file system grows with the data stored on it. If the size of the data approaches the virtual size of the file system, Stratis grows the thin volume and the file system automatically.

The file systems are formatted with XFS.

**IMPORTANT**

Stratis tracks information about file systems created using Stratis that XFS is not aware of, and changes made using XFS do not automatically create updates in Stratis. Users must not reformat or reconfigure XFS file systems that are managed by Stratis.

Stratis creates links to file systems at the `/dev/stratis/my-pool/my-fs` path.

**NOTE**

Stratis uses many Device Mapper devices, which show up in `dmsetup` listings and the `/proc/partitions` file. Similarly, the `lsblk` command output reflects the internal workings and layers of Stratis.

### 41.1.3. Block devices usable with Stratis

Storage devices that can be used with Stratis.

**Supported devices**

Stratis pools have been tested to work on these types of block devices:

- LUKS
- LVM logical volumes
- MD RAID
- DM Multipath
- iSCSI
- HDDs and SSDs
NVMe devices

Unsupported devices
Because Stratis contains a thin-provisioning layer, Red Hat does not recommend placing a Stratis pool on block devices that are already thinly-provisioned.

41.1.4. Installing Stratis
Install the required packages for Stratis.

Procedure
1. Install packages that provide the Stratis service and command-line utilities:

   # yum install stratisd stratis-cli

2. Make sure that the stratisd service is enabled:

   # systemctl enable --now stratisd

41.1.5. Creating an unencrypted Stratis pool
You can create an unencrypted Stratis pool from one or more block devices.

Prerequisites

- Stratis is installed. For more information, see Installing Stratis.
- The stratisd service is running.
- The block devices on which you are creating a Stratis pool are not in use and are not mounted.
- Each block device on which you are creating a Stratis pool is at least 1 GB.
- On the IBM Z architecture, the /dev/dasd* block devices must be partitioned. Use the partition in the Stratis pool.

For information on partitioning DASD devices, see Configuring a Linux instance on IBM Z.

NOTE
You cannot encrypt an unencrypted Stratis pool.

Procedure
1. Erase any file system, partition table, or RAID signatures that exist on each block device that you want to use in the Stratis pool:

   # wipefs --all block-device

   where block-device is the path to the block device; for example, /dev/sdb.

2. Create the new unencrypted Stratis pool on the selected block device:
# stratis pool create my-pool block-device

where `block-device` is the path to an empty or wiped block device.

**NOTE**

Specify multiple block devices on a single line:

```bash
# stratis pool create my-pool block-device-1 block-device-2
```

3. Verify that the new Stratis pool was created:

```bash
# stratis pool list
```

## 41.1.6. Creating an encrypted Stratis pool

To secure your data, you can create an encrypted Stratis pool from one or more block devices.

When you create an encrypted Stratis pool, the kernel keyring is used as the primary encryption mechanism. After subsequent system reboots this kernel keyring is used to unlock the encrypted Stratis pool.

When creating an encrypted Stratis pool from one or more block devices, note the following:

- Each block device is encrypted using the `cryptsetup` library and implements the `LUKS2` format.
- Each Stratis pool can either have a unique key or share the same key with other pools. These keys are stored in the kernel keyring.
- The block devices that comprise a Stratis pool must be either all encrypted or all unencrypted. It is not possible to have both encrypted and unencrypted block devices in the same Stratis pool.
- Block devices added to the data tier of an encrypted Stratis pool are automatically encrypted.

### Prerequisites

- Stratis v2.1.0 or later is installed. For more information, see [Installing Stratis](#).
- The `stratisd` service is running.
- The block devices on which you are creating a Stratis pool are not in use and are not mounted.
- The block devices on which you are creating a Stratis pool are at least 1GB in size each.
- On the IBM Z architecture, the `/dev/dasd*` block devices must be partitioned. Use the partition in the Stratis pool.

For information on partitioning DASD devices, see [Configuring a Linux instance on IBM Z](#).

### Procedure

1. Erase any file system, partition table, or RAID signatures that exist on each block device that you want to use in the Stratis pool:
# wipefs --all `block-device`

where `block-device` is the path to the block device; for example, `/dev/sdb`.

2. If you have not created a key set already, run the following command and follow the prompts to create a key set to use for the encryption.

```shell
# stratis key set --capture-key `key-description`
```

where `key-description` is a reference to the key that gets created in the kernel keyring.

3. Create the encrypted Stratis pool and specify the key description to use for the encryption. You can also specify the key path using the `--keyfile-path` option instead of using the `key-description` option.

```shell
# stratis pool create --key-desc `key-description` my-pool `block-device`
```

where

`key-description` References the key that exists in the kernel keyring, which you created in the previous step.

`my-pool` Specifies the name of the new Stratis pool.

`block-device` Specifies the path to an empty or wiped block device.

**NOTE**

Specify multiple block devices on a single line:

```shell
# stratis pool create --key-desc `key-description` my-pool `block-device-1` `block-device-2`
```

4. Verify that the new Stratis pool was created:

```shell
# stratis pool list
```

### 41.1.7. Binding a Stratis pool to NBDE

Binding an encrypted Stratis pool to Network Bound Disk Encryption (NBDE) requires a Tang server. When a system containing the Stratis pool reboots, it connects with the Tang server to automatically unlock the encrypted pool without you having to provide the kernel keyring description.

**NOTE**

Binding a Stratis pool to a supplementary Clevis encryption mechanism does not remove the primary kernel keyring encryption.

Prerequisites
- Stratis v2.3.0 or later is installed. For more information, see Installing Stratis.

- The **stratisd** service is running.

- You have created an encrypted Stratis pool, and you have the key description of the key that was used for the encryption. For more information, see Creating an encrypted Stratis pool.

- You can connect to the Tang server. For more information, see Deploying a Tang server with SELinux in enforcing mode.

**Procedure**

- Bind an encrypted Stratis pool to NBDE:

  ```
  # stratis pool bind nbde my-pool key-description tang-server
  ```

  where

  **my-pool**
  
  Specifies the name of the encrypted Stratis pool.

  **key-description**
  
  References the key that exists in the kernel keyring, which was generated when you created the encrypted Stratis pool.

  **tang-server**
  
  Specifies the IP address or URL of the Tang server.

**Additional resources**

- Configuring automated unlocking of encrypted volumes using policy-based decryption

41.1.8. Binding a Stratis pool to TPM

When you bind an encrypted Stratis pool to the Trusted Platform Module (TPM) 2.0, when the system containing the pool reboots, the pool is automatically unlocked without you having to provide the kernel keyring description.

**Prerequisites**

- Stratis v2.3.0 or later is installed. For more information, see Installing Stratis.

- The **stratisd** service is running.

- You have created an encrypted Stratis pool. For more information, see Creating an encrypted Stratis pool.

**Procedure**

- Bind an encrypted Stratis pool to TPM:

  ```
  # stratis pool bind tpm my-pool key-description
  ```

  where
my-pool

Specifies the name of the encrypted Stratis pool.

key-description

References the key that exists in the kernel keyring, which was generated when you created the encrypted Stratis pool.

41.1.9. Unlocking an encrypted Stratis pool with kernel keyring

After a system reboot, your encrypted Stratis pool or the block devices that comprise it might not be visible. You can unlock the pool using the kernel keyring that was used to encrypt the pool.

Prerequisites

- Stratis v2.1.0 is installed. For more information, see Installing Stratis.
- The stratisd service is running.
- You have created an encrypted Stratis pool. For more information, see Creating an encrypted Stratis pool.

Procedure

1. Re-create the key set using the same key description that was used previously:

```
# stratis key set --capture-key key-description
```

   where `key-description` references the key that exists in the kernel keyring, which was generated when you created the encrypted Stratis pool.

2. Unlock the Stratis pool and the block device that comprise it:

```
# stratis pool unlock keyring
```

3. Verify that the Stratis pool is visible:

```
# stratis pool list
```

41.1.10. Unlocking an encrypted Stratis pool with Clevis

After a system reboot, your encrypted Stratis pool or the block devices that comprise it might not be visible. You can unlock an encrypted Stratis pool with the supplementary encryption mechanism that the pool is bound to.

Prerequisites

- Stratis v2.3.0 or later is installed. For more information, see Installing Stratis.
- The stratisd service is running.
- You have created an encrypted Stratis pool. For more information, see Creating an encrypted Stratis pool.
The encrypted Stratis pool is bound to a supported, supplementary encryption mechanism. For more information, see Binding an encrypted Stratis pool to NBDE or Binding an encrypted Stratis pool to TPM.

Procedure

1. Unlock the Stratis pool and the block devices that comprise it:

   # stratis pool unlock clevis

2. Verify that the Stratis pool is visible:

   # stratis pool list

41.1.11. Unbinding a Stratis pool from supplementary encryption

When you unbind an encrypted Stratis pool from a supported supplementary encryption mechanism, the primary kernel keyring encryption remains in place.

Prerequisites

- Stratis v2.3.0 or later is installed on your system. For more information, see Installing Stratis.
- You have created an encrypted Stratis pool. For more information, see Creating an encrypted Stratis pool.
- The encrypted Stratis pool is bound to a supported supplementary encryption mechanism.

Procedure

- Unbind an encrypted Stratis pool from a supplementary encryption mechanism:

  # stratis pool unbind clevis my-pool

  where

  _my-pool_ specifies the name of the Stratis pool you want to unbind.

Additional resources

- Binding an encrypted Stratis pool to NBDE
- Binding an encrypted Stratis pool to TPM

41.1.12. Creating a Stratis file system

Create a Stratis file system on an existing Stratis pool.

Prerequisites

- Stratis is installed. For more information, see Installing Stratis.
- The _stratisd_ service is running.
• You have created a Stratis pool. See Creating an unencrypted Stratis pool or Creating an encrypted Stratis pool.

Procedure

1. To create a Stratis file system on a pool, use:

   ```bash
   # stratis fs create my-pool my-fs
   ```

   where

   **my-pool**
   
   Specifies the name of the Stratis pool.

   **my-fs**
   
   Specifies an arbitrary name for the file system.

2. To verify, list file systems within the pool:

   ```bash
   # stratis fs list my-pool
   ```

Additional resources

• Mounting a Stratis file system.

41.1.13. Mounting a Stratis file system

Mount an existing Stratis file system to access the content.

Prerequisites

• Stratis is installed. For more information, see Installing Stratis.

• The stratisd service is running.

• You have created a Stratis file system. For more information, see Creating a Stratis filesystem.

Procedure

• To mount the file system, use the entries that Stratis maintains in the /dev/stratis/ directory:

   ```bash
   # mount /dev/stratis/my-pool/my-fs mount-point
   ```

The file system is now mounted on the mount-point directory and ready to use.

Additional resources

• Creating a Stratis file system.

41.1.14. Persistently mounting a Stratis file system

This procedure persistently mounts a Stratis file system so that it is available automatically after booting the system.
Prerequisites

- Stratis is installed. See Installing Stratis.
- The `stratisd` service is running.
- You have created a Stratis file system. See Creating a Stratis filesystem.

Procedure

1. Determine the UUID attribute of the file system:

   ```bash
   $ lsblk --output=UUID /stratis/my-pool/my-fs
   
   For example:
   
   Example 41.1. Viewing the UUID of Stratis file system
   
   $ lsblk --output=UUID /stratis/my-pool/fs1
   
   UUID
   
   a1f0b64a-4ebb-4d4e-9543-b1d79f600283
   
2. If the mount point directory does not exist, create it:

   ```bash
   # mkdir --parents mount-point
   
   For example:
   
   Example 41.2. The /fs1 mount point in /etc/fstab
   
   UUID=a1f0b64a-4ebb-4d4e-9543-b1d79f600283 /fs1 xfs defaults,x-systemd.requires=stratisd.service 0 0
   
3. As root, edit the `/etc/fstab` file and add a line for the file system, identified by the UUID. Use `xfs` as the file system type and add the `x-systemd.requires=stratisd.service` option.

   For example:

   Example 41.2. The /fs1 mount point in /etc/fstab

   UUID=a1f0b64a-4ebb-4d4e-9543-b1d79f600283 /fs1 xfs defaults,x-systemd.requires=stratisd.service 0 0

4. Regenerate mount units so that your system registers the new configuration:

   ```bash
   # systemctl daemon-reload
   
   5. Try mounting the file system to verify that the configuration works:

   ```bash
   # mount mount-point

Additional resources

- Persistently mounting file systems.
41.2. EXTENDING A STRATIS VOLUME WITH ADDITIONAL BLOCK DEVICES

You can attach additional block devices to a Stratis pool to provide more storage capacity for Stratis file systems.

**IMPORTANT**

Stratis is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see https://access.redhat.com/support/offerings/techpreview.

41.2.1. Components of a Stratis volume

Learn about the components that comprise a Stratis volume.

Externally, Stratis presents the following volume components in the command-line interface and the API:

- `blockdev`
  - Block devices, such as a disk or a disk partition.

- `pool`
  - Composed of one or more block devices.
  - A pool has a fixed total size, equal to the size of the block devices.

  The pool contains most Stratis layers, such as the non-volatile data cache using the `dm-cache` target.

  Stratis creates a `/dev/stratis/my-pool` directory for each pool. This directory contains links to devices that represent Stratis file systems in the pool.

- `filesystem`
  - Each pool can contain one or more file systems, which store files.
  - File systems are thinly provisioned and do not have a fixed total size. The actual size of a file system grows with the data stored on it. If the size of the data approaches the virtual size of the file system, Stratis grows the thin volume and the file system automatically.

  The file systems are formatted with XFS.

  **IMPORTANT**

  Stratis tracks information about file systems created using Stratis that XFS is not aware of, and changes made using XFS do not automatically create updates in Stratis. Users must not reformat or reconfigure XFS file systems that are managed by Stratis.

  Stratis creates links to file systems at the `/dev/stratis/my-pool/my-fs` path.
NOTE

Stratis uses many Device Mapper devices, which show up in `dmsetup` listings and the `/proc/partitions` file. Similarly, the `lsblk` command output reflects the internal workings and layers of Stratis.

41.2.2. Adding block devices to a Stratis pool

This procedure adds one or more block devices to a Stratis pool to be usable by Stratis file systems.

Prerequisites

- Stratis is installed. See `Installing Stratis`.
- The `stratisd` service is running.
- The block devices that you are adding to the Stratis pool are not in use and not mounted.
- The block devices that you are adding to the Stratis pool are at least 1 GiB in size each.

Procedure

- To add one or more block devices to the pool, use:

  ```
  # stratis pool add-data my-pool device-1 device-2 device-n
  ```

Additional resources

- `stratis(8)` man page

41.2.3. Additional resources

- The Stratis Storage website

41.3. MONITORING STRATIS FILE SYSTEMS

As a Stratis user, you can view information about Stratis volumes on your system to monitor their state and free space.

IMPORTANT

Stratis is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see https://access.redhat.com/support/offerings/techpreview.

41.3.1. Stratis sizes reported by different utilities

This section explains the difference between Stratis sizes reported by standard utilities such as `df` and the `stratis` utility.
Standard Linux utilities such as `df` report the size of the XFS file system layer on Stratis, which is 1 TiB. This is not useful information, because the actual storage usage of Stratis is less due to thin provisioning, and also because Stratis automatically grows the file system when the XFS layer is close to full.

**IMPORTANT**

Regularly monitor the amount of data written to your Stratis file systems, which is reported as the *Total Physical Used* value. Make sure it does not exceed the *Total Physical Size* value.

Additional resources

- `stratis(8)` man page.

### 41.3.2. Displaying information about Stratis volumes

This procedure lists statistics about your Stratis volumes, such as the total, used, and free size or file systems and block devices belonging to a pool.

**Prerequisites**

- Stratis is installed. See [Installing Stratis](#).
- The `stratisd` service is running.

**Procedure**

- To display information about all block devices used for Stratis on your system:

  ```
  # stratis blockdev
  ```

<table>
<thead>
<tr>
<th>Pool Name</th>
<th>Device Node</th>
<th>Physical Size</th>
<th>State</th>
<th>Tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>my-pool</td>
<td>/dev/sdb</td>
<td>9.10 TiB</td>
<td>In-use</td>
<td>Data</td>
</tr>
</tbody>
</table>

- To display information about all Stratis pools on your system:

  ```
  # stratis pool
  ```

<table>
<thead>
<tr>
<th>Name</th>
<th>Total Physical Size</th>
<th>Total Physical Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>my-pool</td>
<td>9.10 TiB</td>
<td>598 MiB</td>
</tr>
</tbody>
</table>

- To display information about all Stratis file systems on your system:

  ```
  # stratis filesystem
  ```

<table>
<thead>
<tr>
<th>Pool Name</th>
<th>Name</th>
<th>Used</th>
<th>Created</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>my-pool</td>
<td>my-fs</td>
<td>546 MiB</td>
<td>Nov 08 2018 08:03</td>
<td>/dev/stratis/my-pool/my-fs</td>
</tr>
</tbody>
</table>

**Additional resources**

- `stratis(8)` man page.

### 41.3.3. Additional resources
41.4. USING SNAPSHOTS ON STRATIS FILE SYSTEMS

You can use snapshots on Stratis file systems to capture file system state at arbitrary times and restore it in the future.

IMPORTANT

Stratis is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see https://access.redhat.com/support/offerings/techpreview.

41.4.1. Characteristics of Stratis snapshots

This section describes the properties and limitations of file system snapshots on Stratis.

In Stratis, a snapshot is a regular Stratis file system created as a copy of another Stratis file system. The snapshot initially contains the same file content as the original file system, but can change as the snapshot is modified. Whatever changes you make to the snapshot will not be reflected in the original file system.

The current snapshot implementation in Stratis is characterized by the following:

- A snapshot of a file system is another file system.
- A snapshot and its origin are not linked in lifetime. A snapshotted file system can live longer than the file system it was created from.
- A file system does not have to be mounted to create a snapshot from it.
- Each snapshot uses around half a gigabyte of actual backing storage, which is needed for the XFS log.

41.4.2. Creating a Stratis snapshot

This procedure creates a Stratis file system as a snapshot of an existing Stratis file system.

Prerequisites

- Stratis is installed. See Installing Stratis.
- The stratisd service is running.
- You have created a Stratis file system. See Creating a Stratis filesystem.

Procedure

- To create a Stratis snapshot, use:
Additional resources

- [stratis](8) man page.

41.4.3. Accessing the content of a Stratis snapshot

This procedure mounts a snapshot of a Stratis file system to make it accessible for read and write operations.

Prerequisites

- Stratis is installed. See Installing Stratis.
- The `stratisd` service is running.
- You have created a Stratis snapshot. See Creating a Stratis file system.

Procedure

1. To access the snapshot, mount it as a regular file system from the `/dev/stratis/my-pool` directory:

   ```
   # mount /dev/stratis/my-pool/my-fs-snapshot mount-point
   ```

Additional resources

- Mounting a Stratis file system.
- [mount](8) man page.

41.4.4. Reverting a Stratis file system to a previous snapshot

This procedure reverts the content of a Stratis file system to the state captured in a Stratis snapshot.

Prerequisites

- Stratis is installed. See Installing Stratis.
- The `stratisd` service is running.
- You have created a Stratis snapshot. See Creating a Stratis snapshot.

Procedure

1. Optionally, back up the current state of the file system to be able to access it later:

   ```
   # stratis filesystem snapshot my-pool my-fs my-fs-backup
   ```

2. Unmount and remove the original file system:
3. Create a copy of the snapshot under the name of the original file system:

   # stratis filesystem snapshot my-pool my-fs-snapshot my-fs

4. Mount the snapshot, which is now accessible with the same name as the original file system:

   # mount /dev/stratis/my-pool/my-fs-snapshot mount-point

The content of the file system named my-fs is now identical to the snapshot my-fs-snapshot.

Additional resources

- stratis(8) man page.

41.4.5. Removing a Stratis snapshot

This procedure removes a Stratis snapshot from a pool. Data on the snapshot are lost.

Prerequisites

- Stratis is installed. See Installing Stratis.
- The stratisd service is running.
- You have created a Stratis snapshot. See Creating a Stratis snapshot.

Procedure

1. Unmount the snapshot:

   # umount /dev/stratis/my-pool/my-fs-snapshot

2. Destroy the snapshot:

   # stratis filesystem destroy my-pool my-fs-snapshot

Additional resources

- stratis(8) man page.

41.4.6. Additional resources

- The Stratis Storage website.

41.5. REMOVING STRATIS FILE SYSTEMS

You can remove an existing Stratis file system or a Stratis pool, destroying data on them.
41.5.1. Components of a Stratis volume

Learn about the components that comprise a Stratis volume.

Externally, Stratis presents the following volume components in the command-line interface and the API:

**blockdev**
- Block devices, such as a disk or a disk partition.

**pool**
- Composed of one or more block devices.
- A pool has a fixed total size, equal to the size of the block devices.

The pool contains most Stratis layers, such as the non-volatile data cache using the dm-cache target.

Stratis creates a `/dev/stratis/my-pool/` directory for each pool. This directory contains links to devices that represent Stratis file systems in the pool.

**filesystem**
- Each pool can contain one or more file systems, which store files.
- File systems are thinly provisioned and do not have a fixed total size. The actual size of a file system grows with the data stored on it. If the size of the data approaches the virtual size of the file system, Stratis grows the thin volume and the file system automatically.

The file systems are formatted with XFS.

**IMPORTANT**
- Stratis tracks information about file systems created using Stratis that XFS is not aware of, and changes made using XFS do not automatically create updates in Stratis. Users must not reformat or reconfigure XFS file systems that are managed by Stratis.

Stratis creates links to file systems at the `/dev/stratis/my-pool/my-fs` path.

**NOTE**
- Stratis uses many Device Mapper devices, which show up in `dmsetup` listings and the `/proc/partitions` file. Similarly, the `lsblk` command output reflects the internal workings and layers of Stratis.
41.5.2. Removing a Stratis file system

This procedure removes an existing Stratis file system. Data stored on it are lost.

Prerequisites

- Stratis is installed. See Installing Stratis.
- The stratisd service is running.
- You have created a Stratis file system. See Creating a Stratis filesystem.

Procedure

1.Unmount the file system:
   ```
   # umount /dev/stratis/my-pool/my-fs
   ```

2. Destroy the file system:
   ```
   # stratis filesystem destroy my-pool my-fs
   ```

3. Verify that the file system no longer exists:
   ```
   # stratis filesystem list my-pool
   ```

Additional resources

- stratis(8) man page.

41.5.3. Removing a Stratis pool

This procedure removes an existing Stratis pool. Data stored on it are lost.

Prerequisites

- Stratis is installed. See Installing Stratis.
- The stratisd service is running.
- You have created a Stratis pool:
  - To create an unencrypted pool, see Creating an unencrypted Stratis pool
  - To create an encrypted pool, see Creating an encrypted Stratis pool.

Procedure

1. List file systems on the pool:
   ```
   # stratis filesystem list my-pool
   ```

2. Unmount all file systems on the pool:
# umount /dev/stratis/my-pool/my-fs-1 \\
# umount /dev/stratis/my-pool/my-fs-2 \\
# umount /dev/stratis/my-pool/my-fs-n

3. Destroy the file systems:

   # stratis filesystem destroy my-pool my-fs-1 my-fs-2

4. Destroy the pool:

   # stratis pool destroy my-pool

5. Verify that the pool no longer exists:

   # stratis pool list

Additional resources

- stratis(8) man page.

41.5.4. Additional resources

- The Stratis Storage website.

41.6. GETTING STARTED WITH SWAP

This section describes swap space, and how to add and remove it.

41.6.1. Overview of swap space

Swap space in Linux is used when the amount of physical memory (RAM) is full. If the system needs more memory resources and the RAM is full, inactive pages in memory are moved to the swap space. While swap space can help machines with a small amount of RAM, it should not be considered a replacement for more RAM.

Swap space is located on hard drives, which have a slower access time than physical memory. Swap space can be a dedicated swap partition (recommended), a swap file, or a combination of swap partitions and swap files.

In years past, the recommended amount of swap space increased linearly with the amount of RAM in the system. However, modern systems often include hundreds of gigabytes of RAM. As a consequence, recommended swap space is considered a function of system memory workload, not system memory.

Adding swap space

The following are the different ways to add a swap space:

- Extending swap on an LVM2 logical volume
- Creating an LVM2 logical volume for swap
- Creating a swap file

For example, you may upgrade the amount of RAM in your system from 1 GB to 2 GB, but there is only 2 GB of swap space. It might be advantageous to increase the amount of swap...
space to 4 GB if you perform memory-intense operations or run applications that require a large amount of memory.

Removing swap space

The following are the different ways to remove a swap space:

- Reducing swap on an LVM2 logical volume
- Removing an LVM2 logical volume for swap
- Removing a swap file

For example, you have downgraded the amount of RAM in your system from 1 GB to 512 MB, but there is 2 GB of swap space still assigned. It might be advantageous to reduce the amount of swap space to 1 GB, since the larger 2 GB could be wasting disk space.

41.6.2. Recommended system swap space

This section describes the recommended size of a swap partition depending on the amount of RAM in your system and whether you want sufficient memory for your system to hibernate. The recommended swap partition size is established automatically during installation. To allow for hibernation, however, you need to edit the swap space in the custom partitioning stage.

The following recommendation are especially important on systems with low memory such as 1 GB and less. Failure to allocate sufficient swap space on these systems can cause issues such as instability or even render the installed system unbootable.

Table 41.1. Recommended swap space

<table>
<thead>
<tr>
<th>Amount of RAM in the system</th>
<th>Recommended swap space</th>
<th>Recommended swap space if allowing for hibernation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \leq 2 \text{ GB} )</td>
<td>2 times the amount of RAM</td>
<td>3 times the amount of RAM</td>
</tr>
<tr>
<td>( &gt; 2 \text{ GB} ) – 8 GB</td>
<td>Equal to the amount of RAM</td>
<td>2 times the amount of RAM</td>
</tr>
<tr>
<td>( &gt; 8 \text{ GB} ) – 64 GB</td>
<td>At least 4 GB</td>
<td>1.5 times the amount of RAM</td>
</tr>
<tr>
<td>( &gt; 64 \text{ GB} )</td>
<td>At least 4 GB</td>
<td>Hibernation not recommended</td>
</tr>
</tbody>
</table>

At the border between each range listed in this table, for example a system with 2 GB, 8 GB, or 64 GB of system RAM, discretion can be exercised with regard to chosen swap space and hibernation support. If your system resources allow for it, increasing the swap space may lead to better performance.

Note that distributing swap space over multiple storage devices also improves swap space performance, particularly on systems with fast drives, controllers, and interfaces.
**IMPORTANT**

File systems and LVM2 volumes assigned as swap space should not be in use when being modified. Any attempts to modify swap fail if a system process or the kernel is using swap space. Use the `free` and `cat /proc/swaps` commands to verify how much and where swap is in use.

Resizing swap space requires temporarily removing the swap space from the system. This can be problematic if running applications rely on the additional swap space and might run into low-memory situations. Preferably, perform swap resizing from rescue mode, see Debug boot options in the Performing an advanced RHEL installation. When prompted to mount the file system, select Skip.

### 41.6.3. Extending swap on an LVM2 logical volume

This procedure describes how to extend swap space on an existing LVM2 logical volume. Assuming `/dev/VolGroup00/LogVol01` is the volume you want to extend by 2 GB.

**Prerequisites**

- You have sufficient disk space.

**Procedure**

1. Disable swapping for the associated logical volume:
   
   ```
   # swapoff -v /dev/VolGroup00/LogVol01
   ```

2. Resize the LVM2 logical volume by 2 GB:
   
   ```
   # lvresize /dev/VolGroup00/LogVol01 -L +2G
   ```

3. Format the new swap space:
   
   ```
   # mkswap /dev/VolGroup00/LogVol01
   ```

4. Enable the extended logical volume:
   
   ```
   # swapon -v /dev/VolGroup00/LogVol01
   ```

**Verification**

- To test if the swap logical volume was successfully extended and activated, inspect active swap space by using the following command:
  
  ```
  $ cat /proc/swaps
  $ free -h
  ```

### 41.6.4. Creating an LVM2 logical volume for swap

This procedure describes how to create an LVM2 logical volume for swap. Assuming `/dev/VolGroup00/LogVol02` is the swap volume you want to add.
Prerequisites

- You have sufficient disk space.

Procedure

1. Create the LVM2 logical volume of size 2 GB:
   ```
   # lvcreate VolGroup00 -n LogVol02 -L 2G
   ```

2. Format the new swap space:
   ```
   # mkswap /dev/VolGroup00/LogVol02
   ```

3. Add the following entry to the `/etc/fstab` file:
   ```
   /dev/VolGroup00/LogVol02 swap swap defaults 0 0
   ```

4. Regenerate mount units so that your system registers the new configuration:
   ```
   # systemctl daemon-reload
   ```

5. Activate swap on the logical volume:
   ```
   # swapon -v /dev/VolGroup00/LogVol02
   ```

Verification

- To test if the swap logical volume was successfully created and activated, inspect active swap space by using the following command:
  ```
  $ cat /proc/swaps
  $ free -h
  ```

41.6.5. Creating a swap file

This procedure describes how to create a swap file.

Prerequisites

- You have sufficient disk space.

Procedure

1. Determine the size of the new swap file in megabytes and multiply by 1024 to determine the number of blocks. For example, the block size of a 64 MB swap file is 65536.

2. Create an empty file:
   ```
   # dd if=/dev/zero of=/swapfile bs=1024 count=65536
   ```
   Replace 65536 with the value equal to the desired block size.
3. Set up the swap file with the command:

```
# mkswap /swapfile
```

4. Change the security of the swap file so it is not world readable.

```
# chmod 0600 /swapfile
```

5. Edit the `/etc/fstab` file with the following entries to enable the swap file at boot time:

```
/swapfile swap swap defaults 0 0
```

The next time the system boots, it activates the new swap file.

6. Regenerate mount units so that your system registers the new `/etc/fstab` configuration:

```
# systemctl daemon-reload
```

7. Activate the swap file immediately:

```
# swapon /swapfile
```

Verification

- To test if the new swap file was successfully created and activated, inspect active swap space by using the following command:

```
$ cat /proc/swaps
$ free -h
```

41.6.6. Reducing swap on an LVM2 logical volume

This procedure describes how to reduce swap on an LVM2 logical volume. Assuming `/dev/VolGroup00/LogVol01` is the volume you want to reduce.

Procedure

1. Disable swapping for the associated logical volume:

```
# swapoff -v /dev/VolGroup00/LogVol01
```

2. Reduce the LVM2 logical volume by 512 MB:

```
# lvreduce /dev/VolGroup00/LogVol01 -L -512M
```

3. Format the new swap space:

```
# mkswap /dev/VolGroup00/LogVol01
```

4. Activate swap on the logical volume:
41.6.7. Removing an LVM2 logical volume for swap

This procedure describes how to remove an LVM2 logical volume for swap. Assuming /dev/VolGroup00/LogVol02 is the swap volume you want to remove.

Procedure

1. Disable swapping for the associated logical volume:

   # swapoff -v /dev/VolGroup00/LogVol02

2. Remove the LVM2 logical volume:

   # lvremove /dev/VolGroup00/LogVol02

3. Remove the following associated entry from the /etc/fstab file:

   /dev/VolGroup00/LogVol02 swap swap defaults 0 0

4. Regenerate mount units so that your system registers the new configuration:

   # systemctl daemon-reload

Verification

- To test if the logical volume was successfully removed, inspect active swap space by using the following command:

   $ cat /proc/swaps
   $ free -h

41.6.8. Removing a swap file

This procedure describes how to remove a swap file.

Procedure

1. At a shell prompt, execute the following command to disable the swap file, where /swapfile is the swap file:

   # swapoff -v /swapfile

Verification

- To test if the swap file was successfully removed, inspect active swap space by using the following command:

   $ cat /proc/swaps
   $ free -h
2. Remove its entry from the `/etc/fstab` file accordingly.

3. Regenerate mount units so that your system registers the new configuration:

   ```
   # systemctl daemon-reload
   ```

4. Remove the actual file:

   ```
   # rm /swapfile
   ```
CHAPTER 42. DEDUPLICATING AND COMPRESSING STORAGE

42.1. DEPLOYING VDO

As a system administrator, you can use VDO to create deduplicated and compressed storage pools.

42.1.1. Introduction to VDO

Virtual Data Optimizer (VDO) provides inline data reduction for Linux in the form of deduplication, compression, and thin provisioning. When you set up a VDO volume, you specify a block device on which to construct your VDO volume and the amount of logical storage you plan to present.

- When hosting active VMs or containers, Red Hat recommends provisioning storage at a 10:1 logical to physical ratio: that is, if you are utilizing 1 TB of physical storage, you would present it as 10 TB of logical storage.

- For object storage, such as the type provided by Ceph, Red Hat recommends using a 3:1 logical to physical ratio: that is, 1 TB of physical storage would present as 3 TB logical storage.

In either case, you can simply put a file system on top of the logical device presented by VDO and then use it directly or as part of a distributed cloud storage architecture.

Because VDO is thinly provisioned, the file system and applications only see the logical space in use and are not aware of the actual physical space available. Use scripting to monitor the actual available space and generate an alert if use exceeds a threshold: for example, when the VDO volume is 80% full.

42.1.2. VDO deployment scenarios

You can deploy VDO in a variety of ways to provide deduplicated storage for:

- both block and file access
- both local and remote storage

Because VDO exposes its deduplicated storage as a standard Linux block device, you can use it with standard file systems, iSCSI and FC target drivers, or as unified storage.

NOTE

Deployment of VDO volumes on top of Ceph RADOS Block Device (RBD) is currently supported. However, the deployment of Red Hat Ceph Storage cluster components on top of VDO volumes is currently not supported.

KVM

You can deploy VDO on a KVM server configured with Direct Attached Storage.
File systems
You can create file systems on top of VDO and expose them to NFS or CIFS users with the NFS server or Samba.

Placement of VDO on iSCSI
You can export the entirety of the VDO storage target as an iSCSI target to remote iSCSI initiators.

When creating a VDO volume on iSCSI, you can place the VDO volume above or below the iSCSI layer. Although there are many considerations to be made, some guidelines are provided here to help you select the method that best suits your environment.

When placing the VDO volume on the iSCSI server (target) below the iSCSI layer:

- The VDO volume is transparent to the initiator, similar to other iSCSI LUNs. Hiding the thin provisioning and space savings from the client makes the appearance of the LUN easier to monitor and maintain.
- There is decreased network traffic because there are no VDO metadata reads or writes, and read verification for the dedupe advice does not occur across the network.

- The memory and CPU resources being used on the iSCSI target can result in better performance. For example, the ability to host an increased number of hypervisors because the volume reduction is happening on the iSCSI target.

- If the client implements encryption on the initiator and there is a VDO volume below the target, you will not realize any space savings.

When placing the VDO volume on the iSCSI client (initiator) above the iSCSI layer:

- There is a potential for lower network traffic across the network in ASYNC mode if achieving high rates of space savings.

- You can directly view and control the space savings and monitor usage.

- If you want to encrypt the data, for example, using `dm-crypt`, you can implement VDO on top of the crypt and take advantage of space efficiency.

**LVM**

On more feature-rich systems, you can use LVM to provide multiple logical unit numbers (LUNs) that are all backed by the same deduplicated storage pool.

In the following diagram, the VDO target is registered as a physical volume so that it can be managed by LVM. Multiple logical volumes (LV1 to LV4) are created out of the deduplicated storage pool. In this way, VDO can support multiprotocol unified block or file access to the underlying deduplicated storage pool.

Deduplicated unified storage design enables for multiple file systems to collectively use the same deduplication domain through the LVM tools. Also, file systems can take advantage of LVM snapshot, copy-on-write, and shrink or grow features, all on top of VDO.

**Encryption**

Device Mapper (DM) mechanisms such as DM Crypt are compatible with VDO. Encrypting VDO volumes helps ensure data security, and any file systems above VDO are still deduplicated.
IMPORTANT

Applying the encryption layer above VDO results in little if any data deduplication. Encryption makes duplicate blocks different before VDO can deduplicate them.

Always place the encryption layer below VDO.

When creating a VDO volume on iSCSI, you can place the VDO volume above or below the iSCSI layer. Although there are many considerations to be made, some guidelines are provided here to help you select the method that best suits your environment.

42.1.3. Components of a VDO volume

VDO uses a block device as a backing store, which can include an aggregation of physical storage consisting of one or more disks, partitions, or even flat files. When a storage management tool creates a VDO volume, VDO reserves volume space for the UDS index and VDO volume. The UDS index and the VDO volume interact together to provide deduplicated block storage.

Figure 42.1. VDO disk organization

The VDO solution consists of the following components:

kvdo

A kernel module that loads into the Linux Device Mapper layer provides a deduplicated, compressed, and thinly provisioned block storage volume.

The kvdo module exposes a block device. You can access this block device directly for block storage or present it through a Linux file system, such as XFS or ext4.

When kvdo receives a request to read a logical block of data from a VDO volume, it maps the requested logical block to the underlying physical block and then reads and returns the requested data.
When `kvdo` receives a request to write a block of data to a VDO volume, it first checks whether the request is a DISCARD or TRIM request or whether the data is uniformly zero. If either of these conditions is true, `kvdo` updates its block map and acknowledges the request. Otherwise, VDO processes and optimizes the data.

**uds**

A kernel module that communicates with the Universal Deduplication Service (UDS) index on the volume and analyzes data for duplicates. For each new piece of data, UDS quickly determines if that piece is identical to any previously stored piece of data. If the index finds a match, the storage system can then internally reference the existing item to avoid storing the same information more than once. The UDS index runs inside the kernel as the `uds` kernel module.

**Command line tools**

For configuring and managing optimized storage.

### 42.1.4. The physical and logical size of a VDO volume

This section describes the physical size, available physical size, and logical size that VDO can utilize:

**Physical size**

This is the same size as the underlying block device. VDO uses this storage for:

- User data, which might be deduplicated and compressed
- VDO metadata, such as the UDS index

**Available physical size**

This is the portion of the physical size that VDO is able to use for user data. It is equivalent to the physical size minus the size of the metadata, minus the remainder after dividing the volume into slabs by the given slab size.

**Logical size**

This is the provisioned size that the VDO volume presents to applications. It is usually larger than the available physical size. If the `--vdoLogicalSize` option is not specified, then the provisioning of the logical volume is now provisioned to a 1:1 ratio. For example, if a VDO volume is put on top of a 20 GB block device, then 2.5 GB is reserved for the UDS index (if the default index size is used). The remaining 17.5 GB is provided for the VDO metadata and user data. As a result, the available storage to consume is not more than 17.5 GB, and can be less due to metadata that makes up the actual VDO volume.

VDO currently supports any logical size up to 254 times the size of the physical volume with an absolute maximum logical size of 4PB.

**Figure 42.2. VDO disk organization**

<table>
<thead>
<tr>
<th>UDS Index</th>
<th>VDO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Block Device</td>
<td></td>
</tr>
</tbody>
</table>
In this figure, the VDO deduplicated storage target sits completely on top of the block device, meaning the physical size of the VDO volume is the same size as the underlying block device.

**Additional resources**

- For more information on how much storage VDO metadata requires on block devices of different sizes, see Section 42.1.6.4, “Examples of VDO requirements by physical size”.

### 42.1.5. Slab size in VDO

The physical storage of the VDO volume is divided into a number of slabs. Each slab is a contiguous region of the physical space. All of the slabs for a given volume have the same size, which can be any power of 2 multiple of 128 MB up to 32 GB.

The default slab size is 2 GB to facilitate evaluating VDO on smaller test systems. A single VDO volume can have up to 8192 slabs. Therefore, in the default configuration with 2 GB slabs, the maximum allowed physical storage is 16 TB. When using 32 GB slabs, the maximum allowed physical storage is 256 TB. VDO always reserves at least one entire slab for metadata, and therefore, the reserved slab cannot be used for storing user data.

Slab size has no effect on the performance of the VDO volume.

**Table 42.1. Recommended VDO slab sizes by physical volume size**

<table>
<thead>
<tr>
<th>Physical volume size</th>
<th>Recommended slab size</th>
</tr>
</thead>
<tbody>
<tr>
<td>10–99 GB</td>
<td>1 GB</td>
</tr>
<tr>
<td>100 GB – 1 TB</td>
<td>2 GB</td>
</tr>
<tr>
<td>2–256 TB</td>
<td>32 GB</td>
</tr>
</tbody>
</table>

**NOTE**

The minimal disk usage for a VDO volume using default settings of 2 GB slab size and 0.25 dense index, requires approx 4.7 GB. This provides slightly less than 2 GB of physical data to write at 0% deduplication or compression.

Here, the minimal disk usage is the sum of the default slab size and dense index.

You can control the slab size by providing the `--config 'allocation/vdo_slab_size_mb=MB_SIZE'` option to the `lvcreate` command.

### 42.1.6. VDO requirements

VDO has certain requirements on its placement and your system resources.

#### 42.1.6.1. VDO memory requirements

Each VDO volume has two distinct memory requirements:

The VDO module
VDO requires a fixed 38 MB of RAM and several variable amounts:

- 1.15 MB of RAM for each 1 MB of configured block map cache size. The block map cache requires a minimum of 150 MB RAM.
- 1.6 MB of RAM for each 1 TB of logical space.
- 268 MB of RAM for each 1 TB of physical storage managed by the volume.

**The UDS index**

The Universal Deduplication Service (UDS) requires a minimum of 250 MB of RAM, which is also the default amount that deduplication uses. You can configure the value when formatting a VDO volume, because the value also affects the amount of storage that the index needs. The memory required for the UDS index is determined by the index type and the required size of the deduplication window:

<table>
<thead>
<tr>
<th>Index type</th>
<th>Deduplication window</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dense</td>
<td>1 TB per 1 GB of RAM</td>
<td>A 1 GB dense index is generally sufficient for up to 4 TB of physical storage.</td>
</tr>
<tr>
<td>Sparse</td>
<td>10 TB per 1 GB of RAM</td>
<td>A 1 GB sparse index is generally sufficient for up to 40 TB of physical storage.</td>
</tr>
</tbody>
</table>

**NOTE**

The minimal disk usage for a VDO volume using default settings of 2 GB slab size and 0.25 dense index, requires approx 4.7 GB. This provides slightly less than 2 GB of physical data to write at 0% deduplication or compression.

Here, the minimal disk usage is the sum of the default slab size and dense index.

The UDS Sparse Indexing feature is the recommended mode for VDO. It relies on the temporal locality of data and attempts to retain only the most relevant index entries in memory. With the sparse index, UDS can maintain a deduplication window that is ten times larger than with dense, while using the same amount of memory.

Although the sparse index provides the greatest coverage, the dense index provides more deduplication advice. For most workloads, given the same amount of memory, the difference in deduplication rates between dense and sparse indexes is negligible.

**Additional resources**

- Examples of VDO requirements by physical size

**42.1.6.2. VDO storage space requirements**

You can configure a VDO volume to use up to 256 TB of physical storage. Only a certain part of the physical storage is usable to store data. This section provides the calculations to determine the usable size of a VDO-managed volume.

VDO requires storage for two types of VDO metadata and for the UDS index:
● The first type of VDO metadata uses approximately 1 MB for each 4 GB of physical storage plus an additional 1 MB per slab.

● The second type of VDO metadata consumes approximately 1.25 MB for each 1 GB of logical storage, rounded up to the nearest slab.

● The amount of storage required for the UDS index depends on the type of index and the amount of RAM allocated to the index. For each 1 GB of RAM, a dense UDS index uses 17 GB of storage, and a sparse UDS index will use 170 GB of storage.

Additional resources

● Examples of VDO requirements by physical size

● Slab size in VDO

42.1.6.3. Placement of VDO in the storage stack

You should place certain storage layers under VDO and others above VDO.

In this section, above means that when layer A is above layer B, A is either stored directly on device B, or indirectly on a layer that is stored on B. Similarly, A under B means that B is stored on A.

A VDO volume is a thinly provisioned block device. To prevent running out of physical space, place the volume above a storage layer that you can expand at a later time. Examples of such expandable storage are LVM volumes or MD RAID arrays.

You can place thick-provisioned layers above VDO, but you cannot rely on the guarantees of thick provisioning in that case. Because the VDO layer is thin-provisioned, the effects of thin provisioning apply to all layers above it. If you do not monitor the VDO device, you might run out of physical space on thick-provisioned volumes above VDO.

Supported configurations

● Layers that you can place only under VDO:
  ○ DM Multipath
  ○ DM Crypt
  ○ Software RAID (LVM or MD RAID)

● Layers that you can place only above VDO:
  ○ LVM cache
  ○ LVM snapshots
  ○ LVM thin provisioning

Unsupported configurations

● VDO above other VDO volumes
● VDO above LVM snapshots
● VDO above LVM cache
- VDO above a loopback device
- VDO above LVM thin provisioning
- Encrypted volumes above VDO
- Partitions on a VDO volume
- RAID, such as LVM RAID, MD RAID, or any other type, above a VDO volume

Additional resources
- For more information on stacking VDO with LVM layers, see the Stacking LVM volumes article.

42.1.6.4. Examples of VDO requirements by physical size

The following tables provide approximate system requirements of VDO based on the physical size of the underlying volume. Each table lists requirements appropriate to the intended deployment, such as primary storage or backup storage.

The exact numbers depend on your configuration of the VDO volume.

Primary storage deployment

In the primary storage case, the UDS index is between 0.01% to 25% the size of the physical size.

Table 42.2. Storage and memory requirements for primary storage

<table>
<thead>
<tr>
<th>Physical size</th>
<th>RAM usage: UDS</th>
<th>RAM usage: VDO</th>
<th>Disk usage</th>
<th>Index type</th>
</tr>
</thead>
<tbody>
<tr>
<td>10GB–1TB</td>
<td>250MB</td>
<td>472MB</td>
<td>2.5GB</td>
<td>Dense</td>
</tr>
<tr>
<td>2–10TB</td>
<td>1GB</td>
<td>3GB</td>
<td>10GB</td>
<td>Dense</td>
</tr>
<tr>
<td></td>
<td>250MB</td>
<td></td>
<td>22GB</td>
<td>Sparse</td>
</tr>
<tr>
<td>11–50TB</td>
<td>2GB</td>
<td>14GB</td>
<td>170GB</td>
<td>Sparse</td>
</tr>
<tr>
<td>51–100TB</td>
<td>3GB</td>
<td>27GB</td>
<td>255GB</td>
<td>Sparse</td>
</tr>
<tr>
<td>101–256TB</td>
<td>12GB</td>
<td>69GB</td>
<td>1020GB</td>
<td>Sparse</td>
</tr>
</tbody>
</table>

Backup storage deployment

In the backup storage case, the UDS index covers the size of the backup set but is not bigger than the physical size. If you expect the backup set or the physical size to grow in the future, factor this into the index size.

Table 42.3. Storage and memory requirements for backup storage

<table>
<thead>
<tr>
<th>Physical size</th>
<th>RAM usage: UDS</th>
<th>RAM usage: VDO</th>
<th>Disk usage</th>
<th>Index type</th>
</tr>
</thead>
<tbody>
<tr>
<td>10GB–1TB</td>
<td>250MB</td>
<td>472MB</td>
<td>2.5 GB</td>
<td>Dense</td>
</tr>
</tbody>
</table>
### 42.1.7. Installing VDO

This procedure installs software necessary to create, mount, and manage VDO volumes.

**Procedure**

- Install the `vdo` and `kmod-kvdo` packages:

  ```
  # yum install vdo kmod-kvdo
  ```

### 42.1.8. Creating a VDO volume

This procedure creates a VDO volume on a block device.

**Prerequisites**

- Install the VDO software. See Section 42.1.7, “Installing VDO”.

- Use expandable storage as the backing block device. For more information, see Section 42.1.6.3, “Placement of VDO in the storage stack”.

**Procedure**

In all the following steps, replace `vdo-name` with the identifier you want to use for your VDO volume; for example, `vdo1`. You must use a different name and device for each instance of VDO on the system.

1. Find a persistent name for the block device where you want to create the VDO volume. For more information on persistent names, see Chapter 31, *Overview of persistent naming attributes*. If you use a non-persistent device name, then VDO might fail to start properly in the future if the device name changes.

2. Create the VDO volume:

   ```
   # vdo create
   --name=vdo-name
   --device=block-device
   --vdoLogicalSize=logical-size
   ```

---

<table>
<thead>
<tr>
<th>Physical size</th>
<th>RAM usage: UDS</th>
<th>RAM usage: VDO</th>
<th>Disk usage</th>
<th>Index type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2–10TB</td>
<td>2GB</td>
<td>3GB</td>
<td>170GB</td>
<td>Sparse</td>
</tr>
<tr>
<td>11–50TB</td>
<td>10GB</td>
<td>14GB</td>
<td>850GB</td>
<td>Sparse</td>
</tr>
<tr>
<td>51–100TB</td>
<td>20GB</td>
<td>27GB</td>
<td>1700GB</td>
<td>Sparse</td>
</tr>
<tr>
<td>101–256TB</td>
<td>26GB</td>
<td>69GB</td>
<td>3400GB</td>
<td>Sparse</td>
</tr>
</tbody>
</table>
Replace `block-device` with the persistent name of the block device where you want to create the VDO volume. For example, `/dev/disk/by-id/scsi-3600508b1001c264ad2af21e903ad031f`.

Replace `logical-size` with the amount of logical storage that the VDO volume should present:

- For active VMs or container storage, use logical size that is ten times the physical size of your block device. For example, if your block device is 1TB in size, use 10T here.
- For object storage, use logical size that is three times the physical size of your block device. For example, if your block device is 1TB in size, use 3T here.

If the physical block device is larger than 16TiB, add the `--vdoSlabSize=32G` option to increase the slab size on the volume to 32GiB. Using the default slab size of 2GiB on block devices larger than 16TiB results in the `vdo create` command failing with the following error:

```
vdo: ERROR - vdoformat: formatVDO failed on '/dev/device': VDO Status: Exceeds maximum number of slabs supported
```

Example 42.1. Creating VDO for container storage

For example, to create a VDO volume for container storage on a 1TB block device, you might use:

```
# vdo create \
    --name=vdo1 \
    --device=/dev/disk/by-id/scsi-3600508b1001c264ad2af21e903ad031f \
    --vdoLogicalSize=10T
```

IMPORTANT

If a failure occurs when creating the VDO volume, remove the volume to clean up. See Removing an unsuccessfully created VDO volume for details.

3. Create a file system on top of the VDO volume:

- For the XFS file system:
  
  ```
  # mkfs.xfs -K /dev/mapper/vdo-name
  ```

- For the ext4 file system:
  
  ```
  # mkfs.ext4 -E nodiscard /dev/mapper/vdo-name
  ```

4. Use the following command to wait for the system to register the new device node:

```
# udevadm settle
```

Next steps
1. Mount the file system. See Section 42.1.9, “Mounting a VDO volume” for details.

2. Enable the discard feature for the file system on your VDO device. See Section 42.1.10, “Enabling periodic block discard” for details.

Additional resources

- The vdo(8) man page

### 42.1.9. Mounting a VDO volume

This procedure mounts a file system on a VDO volume, either manually or persistently.

**Prerequisites**

- A VDO volume has been created on your system. For instructions, see Section 42.1.8, “Creating a VDO volume”.

**Procedure**

- To mount the file system on the VDO volume manually, use:
  
  ```
  # mount /dev/mapper/vdo-name mount-point
  ```

- To configure the file system to mount automatically at boot, add a line to the /etc/fstab file:
  
  For the XFS file system:
  
  ```
  /dev/mapper/vdo-name mount-point xfs defaults,x-systemd.device-timeout=0,x-systemd.requires=vdo.service 0 0
  ```

  For the ext4 file system:
  
  ```
  /dev/mapper/vdo-name mount-point ext4 defaults,x-systemd.device-timeout=0,x-systemd.requires=vdo.service 0 0
  ```

  If the VDO volume is located on a block device that requires network, such as iSCSI, add the _netdev mount option.

**Additional resources**

- The vdo(8) man page.

- For iSCSI and other block devices requiring network, see the systemd.mount(5) man page for information on the _netdev mount option.

### 42.1.10. Enabling periodic block discard

This procedure enables a systemd timer that regularly discards unused blocks on all supported file systems.

**Procedure**
Enable and start the `systemd` timer:

```
# systemctl enable --now fstrim.timer
```

### 42.1.11. Monitoring VDO

This procedure describes how to obtain usage and efficiency information from a VDO volume.

**Prerequisites**

- Install the VDO software. See [Installing VDO](#).

**Procedure**

- Use the `vdostats` utility to get information about a VDO volume:

```
# vdostats --human-readable
```

<table>
<thead>
<tr>
<th>Device</th>
<th>1K-blocks</th>
<th>Used</th>
<th>Available</th>
<th>Use%</th>
<th>Space saving%</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/mapper/node1osd1</td>
<td>926.5G</td>
<td>21.0G</td>
<td>905.5G</td>
<td>2%</td>
<td>73%</td>
</tr>
<tr>
<td>/dev/mapper/node1osd2</td>
<td>926.5G</td>
<td>28.2G</td>
<td>898.3G</td>
<td>3%</td>
<td>64%</td>
</tr>
</tbody>
</table>

**Additional resources**

- The [vdostats(8)](https://redhat.github.io/vdo/vdostats(8)) man page.

### 42.2. MAINTAINING VDO

After deploying a VDO volume, you can perform certain tasks to maintain or optimize it. Some of the following tasks are required for the correct functioning of VDO volumes.

**Prerequisites**

- VDO is installed and deployed. See [Section 42.1, “Deploying VDO”](#).

#### 42.2.1. Managing free space on VDO volumes

VDO is a thinly provisioned block storage target. Because of that, you must actively monitor and manage space usage on VDO volumes.

##### 42.2.1.1. The physical and logical size of a VDO volume

This section describes the physical size, available physical size, and logical size that VDO can utilize:

**Physical size**

This is the same size as the underlying block device. VDO uses this storage for:

- User data, which might be deduplicated and compressed
- VDO metadata, such as the UDS index

**Available physical size**
This is the portion of the physical size that VDO is able to use for user data. It is equivalent to the physical size minus the size of the metadata, minus the remainder after dividing the volume into slabs by the given slab size.

**Logical Size**

This is the provisioned size that the VDO volume presents to applications. It is usually larger than the available physical size. If the `--vdoLogicalSize` option is not specified, then the provisioning of the logical volume is now provisioned to a 1:1 ratio. For example, if a VDO volume is put on top of a 20 GB block device, then 2.5 GB is reserved for the UDS index (if the default index size is used). The remaining 17.5 GB is provided for the VDO metadata and user data. As a result, the available storage to consume is not more than 17.5 GB, and can be less due to metadata that makes up the actual VDO volume.

VDO currently supports any logical size up to 254 times the size of the physical volume with an absolute maximum logical size of 4PB.

**Figure 42.3. VDO disk organization**

In this figure, the VDO deduplicated storage target sits completely on top of the block device, meaning the physical size of the VDO volume is the same size as the underlying block device.

**Additional resources**

- For more information on how much storage VDO metadata requires on block devices of different sizes, see Section 42.1.6.4, "Examples of VDO requirements by physical size".

**42.2.1.2. Thin provisioning in VDO**

VDO is a thinly provisioned block storage target. The amount of physical space that a VDO volume uses might differ from the size of the volume that is presented to users of the storage. You can make use of this disparity to save on storage costs.

**Out-of-space conditions**

Take care to avoid unexpectedly running out of storage space, if the data written does not achieve the expected rate of optimization.

Whenever the number of logical blocks (virtual storage) exceeds the number of physical blocks (actual storage), it becomes possible for file systems and applications to unexpectedly run out of space. For that reason, storage systems using VDO must provide you with a way of monitoring the size of the free pool on the VDO volume.

You can determine the size of this free pool by using the `vdostats` utility. The default output of this utility lists information for all running VDO volumes in a format similar to the Linux `df` utility. For example:

```
<table>
<thead>
<tr>
<th>Device</th>
<th>1K-blocks</th>
<th>Used</th>
<th>Available</th>
<th>Use%</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/mapper/vdo-name</td>
<td>211812352</td>
<td>105906176</td>
<td>105906176</td>
<td>50%</td>
</tr>
</tbody>
</table>
```
When the physical storage capacity of a VDO volume is almost full, VDO reports a warning in the system log, similar to the following:

```
Oct 2 17:27:39 system lvm[13863]: WARNING: VDO pool vdo-name is now 80.69% full.
Oct 2 17:28:19 system lvm[13863]: WARNING: VDO pool vdo-name is now 85.25% full.
Oct 2 17:29:39 system lvm[13863]: WARNING: VDO pool vdo-name is now 90.64% full.
Oct 2 17:30:29 system lvm[13863]: WARNING: VDO pool vdo-name is now 96.07% full.
```

**NOTE**

These warning messages appear only when the `lvm2-monitor` service is running. It is enabled by default.

**How to prevent out-of-space conditions**

If the size of free pool drops below a certain level, you can take action by:

- Deleting data. This reclaims space whenever the deleted data is not duplicated. Deleting data frees the space only after discards are issued.

- Adding physical storage

**IMPORTANT**

Monitor physical space on your VDO volumes to prevent out-of-space situations. Running out of physical blocks might result in losing recently written, unacknowledged data on the VDO volume.

**Thin provisioning and the TRIM and DISCARD commands**

To benefit from the storage savings of thin provisioning, the physical storage layer needs to know when data is deleted. File systems that work with thinly provisioned storage send TRIM or DISCARD commands to inform the storage system when a logical block is no longer required.

Several methods of sending the TRIM or DISCARD commands are available:

- With the `discard` mount option, the file systems can send these commands whenever a block is deleted.

- You can send the commands in a controlled manner by using utilities such as `fstrim`. These utilities tell the file system to detect which logical blocks are unused and send the information to the storage system in the form of a TRIM or DISCARD command.

The need to use TRIM or DISCARD on unused blocks is not unique to VDO. Any thinly provisioned storage system has the same challenge.

**42.2.1.3. Monitoring VDO**

This procedure describes how to obtain usage and efficiency information from a VDO volume.

**Prerequisites**

- Install the VDO software. See Installing VDO.
Procedure

- Use the `vdostats` utility to get information about a VDO volume:

  ```
  # vdostats --human-readable
  Device                 1K-blocks   Used     Available    Use%    Space saving%
  /dev/mapper/node1osd1  926.5G      21.0G    905.5G       2%      73%
  /dev/mapper/node1osd2  926.5G      28.2G    898.3G       3%      64%
  ```

Additional resources

- The `vdostats(8)` man page.

42.2.1.4. Reclaiming space for VDO on file systems

This procedure reclaims storage space on a VDO volume that hosts a file system.

VDO cannot reclaim space unless file systems communicate that blocks are free using the `DISCARD`, `TRIM`, or `UNMAP` commands.

Procedure

- If the file system on your VDO volume supports discard operations, enable them. See Discarding unused blocks.

- For file systems that do not use `DISCARD`, `TRIM`, or `UNMAP`, you can manually reclaim free space. Store a file consisting of binary zeros to fill the free space and then delete that file.

42.2.1.5. Reclaiming space for VDO without a file system

This procedure reclaims storage space on a VDO volume that is used as a block storage target without a file system.

Procedure

- Use the `blkdiscard` utility.

  For example, a single VDO volume can be carved up into multiple subvolumes by deploying LVM on top of it. Before deprovisioning a logical volume, use the `blkdiscard` utility to free the space previously used by that logical volume.

  LVM supports the `REQ_DISCARD` command and forwards the requests to VDO at the appropriate logical block addresses in order to free the space. If you use other volume managers, they also need to support `REQ_DISCARD`, or equivalently, `UNMAP` for SCSI devices or `TRIM` for ATA devices.

Additional resources

- The `blkdiscard(8)` man page

42.2.1.6. Reclaiming space for VDO on Fibre Channel or Ethernet network

---

CHAPTER 42. DEDUPLICATING AND COMPRESSING STORAGE

715
This procedure reclaims storage space on VDO volumes (or portions of volumes) that are provisioned to hosts on a Fibre Channel storage fabric or an Ethernet network using SCSI target frameworks such as LIO or SCST.

**Procedure**

- SCSI initiators can use the **UNMAP** command to free space on thinly provisioned storage targets, but the SCSI target framework needs to be configured to advertise support for this command. This is typically done by enabling thin provisioning on these volumes.

Verify support for **UNMAP** on Linux-based SCSI initiators by running the following command:

```
# sg_vpd --page=0xb0 /dev/device
```

In the output, verify that the *Maximum unmap LBA count* value is greater than zero.

### 42.2.2. Starting or stopping VDO volumes

You can start or stop a given VDO volume, or all VDO volumes, and their associated UDS indexes.

**42.2.2.1. Started and activated VDO volumes**

During the system boot, the **vdo systemd** unit automatically starts all VDO devices that are configured as activated.

The **vdo systemd** unit is installed and enabled by default when the **vdo** package is installed. This unit automatically runs the **vdo start --all** command at system startup to bring up all activated VDO volumes.

You can also create a VDO volume that does not start automatically by adding the **--activate=disabled** option to the **vdo create** command.

**The starting order**

Some systems might place LVM volumes both above VDO volumes and below them. On these systems, it is necessary to start services in the right order:

1. The lower layer of LVM must start first. In most systems, starting this layer is configured automatically when the LVM package is installed.
2. The **vdo systemd** unit must start then.
3. Finally, additional scripts must run in order to start LVM volumes or other services on top of the running VDO volumes.

**How long it takes to stop a volume**

Stopping a VDO volume takes time based on the speed of your storage device and the amount of data that the volume needs to write:

- The volume always writes around 1GiB for every 1GiB of the UDS index.
- The volume additionally writes the amount of data equal to the block map cache size plus up to 8MiB per slab.
- The volume must finish processing all outstanding IO requests.
42.2.2.2. Starting a VDO volume

This procedure starts a given VDO volume or all VDO volumes on your system.

Procedure

- To start a given VDO volume, use:
  
  # vdo start --name=my-vdo

- To start all VDO volumes, use:
  
  # vdo start --all

Additional resources

- The vdo(8) man page

42.2.2.3. Stopping a VDO volume

This procedure stops a given VDO volume or all VDO volumes on your system.

Procedure

1. Stop the volume.
   - To stop a given VDO volume, use:
     
     # vdo stop --name=my-vdo
   - To stop all VDO volumes, use:
     
     # vdo stop --all

2. Wait for the volume to finish writing data to the disk.

Additional resources

- The vdo(8) man page

42.2.2.4. Additional resources

- If restarted after an unclean shutdown, VDO performs a rebuild to verify the consistency of its metadata and repairs it if necessary. See Section 42.2.5, “Recovering a VDO volume after an unclean shutdown” for more information on the rebuild process.

42.2.3. Automatically starting VDO volumes at system boot

You can configure VDO volumes so that they start automatically at system boot. You can also disable the automatic start.

42.2.3.1. Started and activated VDO volumes
During the system boot, the `vdo systemd` unit automatically starts all VDO devices that are configured as activated.

The `vdo systemd` unit is installed and enabled by default when the `vdo` package is installed. This unit automatically runs the `vdo start --all` command at system startup to bring up all activated VDO volumes.

You can also create a VDO volume that does not start automatically by adding the `--activate=disabled` option to the `vdo create` command.

**The starting order**

Some systems might place LVM volumes both above VDO volumes and below them. On these systems, it is necessary to start services in the right order:

1. The lower layer of LVM must start first. In most systems, starting this layer is configured automatically when the LVM package is installed.
2. The `vdo systemd` unit must start then.
3. Finally, additional scripts must run in order to start LVM volumes or other services on top of the running VDO volumes.

**How long it takes to stop a volume**

Stopping a VDO volume takes time based on the speed of your storage device and the amount of data that the volume needs to write:

- The volume always writes around 1GiB for every 1GiB of the UDS index.
- The volume additionally writes the amount of data equal to the block map cache size plus up to 8MiB per slab.
- The volume must finish processing all outstanding IO requests.

### 42.2.3.2. Activating a VDO volume

This procedure activates a VDO volume to enable it to start automatically.

**Procedure**

- To activate a specific volume:
  
  ```
  # vdo activate --name=my-vdo
  ```

- To activate all volumes:
  
  ```
  # vdo activate --all
  ```

**Additional resources**

- The `vdo(8)` man page

### 42.2.3.3. Deactivating a VDO volume

This procedure deactivates a VDO volume to prevent it from starting automatically.
Procedure

- To deactivate a specific volume:
  ```
  # vdo deactivate --name=my-vdo
  ```
- To deactivate all volumes:
  ```
  # vdo deactivate --all
  ```

Additional resources

- The `vdo(8)` man page

42.2.4. Selecting a VDO write mode

You can configure write mode for a VDO volume, based on what the underlying block device requires. By default, VDO selects write mode automatically.

42.2.4.1. VDO write modes

VDO supports the following write modes:

**sync**

When VDO is in `sync` mode, the layers above it assume that a write command writes data to persistent storage. As a result, it is not necessary for the file system or application, for example, to issue FLUSH or force unit access (FUA) requests to cause the data to become persistent at critical points.

VDO must be set to `sync` mode only when the underlying storage guarantees that data is written to persistent storage when the write command completes. That is, the storage must either have no volatile write cache, or have a write through cache.

**async**

When VDO is in `async` mode, VDO does not guarantee that the data is written to persistent storage when a write command is acknowledged. The file system or application must issue FLUSH or FUA requests to ensure data persistence at critical points in each transaction.

VDO must be set to `async` mode if the underlying storage does not guarantee that data is written to persistent storage when the write command completes; that is, when the storage has a volatile write back cache.

**async-unsafe**

This mode has the same properties as `async` but it is not compliant with Atomicity, Consistency, Isolation, Durability (ACID). Compared to `async`, `async-unsafe` has a better performance.

⚠️ **WARNING**

When an application or a file system that assumes ACID compliance operates on top of the VDO volume, `async-unsafe` mode might cause unexpected data loss.
auto

The auto mode automatically selects sync or async based on the characteristics of each device. This is the default option.

42.2.4.2. The internal processing of VDO write modes

This section provides details on how the sync and async VDO write modes operate.

If the kvdo module is operating in synchronous mode:

1. It temporarily writes the data in the request to the allocated block and then acknowledges the request.

2. Once the acknowledgment is complete, an attempt is made to deduplicate the block by computing a MurmurHash-3 signature of the block data, which is sent to the VDO index.

3. If the VDO index contains an entry for a block with the same signature, kvdo reads the indicated block and does a byte-by-byte comparison of the two blocks to verify that they are identical.

4. If they are indeed identical, then kvdo updates its block map so that the logical block points to the corresponding physical block and releases the allocated physical block.

5. If the VDO index did not contain an entry for the signature of the block being written, or the indicated block does not actually contain the same data, kvdo updates its block map to make the temporary physical block permanent.

If kvdo is operating in asynchronous mode:

1. Instead of writing the data, it will immediately acknowledge the request.

2. It will then attempt to deduplicate the block in same manner as described above.

3. If the block turns out to be a duplicate, kvdo updates its block map and releases the allocated block. Otherwise, it writes the data in the request to the allocated block and updates the block map to make the physical block permanent.

42.2.4.3. Checking the write mode on a VDO volume

This procedure lists the active write mode on a selected VDO volume.

Procedure

- Use the following command to see the write mode used by a VDO volume:

  ```
  # vdo status --name=my-vdo
  ```

  The output lists:

  - The configured write policy, which is the option selected from sync, async, or auto
  - The write policy, which is the particular write mode that VDO applied, that is either sync or async

42.2.4.4. Checking for a volatile cache
This procedure determines if a block device has a volatile cache or not. You can use the information to choose between the sync and async VDO write modes.

**Procedure**

1. Use either of the following methods to determine if a device has a writeback cache:
   - Read the `/sys/block/block-device/device/scsi_disk/identifier/cache_type` sysfs file. For example:
     ```bash
     $ cat '/sys/block/sda/device/scsi_disk/7:0:0:0/cache_type'
     write back
     
     $ cat '/sys/block/sdb/device/scsi_disk/1:2:0:0/cache_type'
     None
     ```
   - Alternatively, you can find whether the above mentioned devices have a write cache or not in the kernel boot log:
     ```bash
     sd 7:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support DPO or FUA
     sd 1:2:0:0: [sdb] Write cache: disabled, read cache: disabled, supports DPO and FUA
     ```

2. In the previous examples:
   - Device sda indicates that it has a writeback cache. Use async mode for it.
   - Device sdb indicates that it does not have a writeback cache. Use sync mode for it.

You should configure VDO to use the sync write mode if the cache_type value is None or write through.

### 42.2.4.5. Setting a VDO write mode

This procedure sets a write mode for a VDO volume, either for an existing one or when creating a new volume.

**IMPORTANT**

Using an incorrect write mode might result in data loss after a power failure, a system crash, or any unexpected loss of contact with the disk.

**Prerequisites**

- Determine which write mode is correct for your device. See Section 42.2.4.4, “Checking for a volatile cache”.

**Procedure**

- You can set a write mode either on an existing VDO volume or when creating a new volume:
  - To modify an existing VDO volume, use:
To specify a write mode when creating a VDO volume, add the --writePolicy=sync|async|async-unsafe|auto option to the vdo create command.

42.2.5. Recovering a VDO volume after an unclean shutdown

You can recover a VDO volume after an unclean shutdown to enable it to continue operating. The task is mostly automated. Additionally, you can clean up after a VDO volume was unsuccessfully created because of a failure in the process.

42.2.5.1. VDO write modes

VDO supports the following write modes:

**sync**

When VDO is in sync mode, the layers above it assume that a write command writes data to persistent storage. As a result, it is not necessary for the file system or application, for example, to issue FLUSH or force unit access (FUA) requests to cause the data to become persistent at critical points. VDO must be set to sync mode only when the underlying storage guarantees that data is written to persistent storage when the write command completes. That is, the storage must either have no volatile write cache, or have a write through cache.

**async**

When VDO is in async mode, VDO does not guarantee that the data is written to persistent storage when a write command is acknowledged. The file system or application must issue FLUSH or FUA requests to ensure data persistence at critical points in each transaction. VDO must be set to async mode if the underlying storage does not guarantee that data is written to persistent storage when the write command completes; that is, when the storage has a volatile write back cache.

**async-unsafe**

This mode has the same properties as async but it is not compliant with Atomicity, Consistency, Isolation, Durability (ACID). Compared to async, async-unsafe has a better performance.

**WARNING**

When an application or a file system that assumes ACID compliance operates on top of the VDO volume, async-unsafe mode might cause unexpected data loss.

**auto**

The auto mode automatically selects sync or async based on the characteristics of each device. This is the default option.

42.2.5.2. VDO volume recovery
When a VDO volume restarts after an unclean shutdown, VDO performs the following actions:

- Verifies the consistency of the metadata on the volume.
- Rebuilds a portion of the metadata to repair it if necessary.

Rebuilds are automatic and do not require user intervention.

VDO might rebuild different writes depending on the active write mode:

**sync**

If VDO was running on synchronous storage and write policy was set to **sync**, all data written to the volume are fully recovered.

**async**

If the write policy was **async**, some writes might not be recovered if they were not made durable. This is done by sending VDO a **FLUSH** command or a write I/O tagged with the FUA (force unit access) flag. You can accomplish this from user mode by invoking a data integrity operation like **fsync**, **fdatasync**, **sync**, or **umount**.

In either mode, some writes that were either unacknowledged or not followed by a flush might also be rebuilt.

**Automatic and manual recovery**

When a VDO volume enters **recovering** operating mode, VDO automatically rebuilds the unclean VDO volume after it comes back online. This is called **online recovery**.

If VDO cannot recover a VDO volume successfully, it places the volume in **read-only** operating mode that persists across volume restarts. You need to fix the problem manually by forcing a rebuild.

**Additional resources**

- For more information on automatic and manual recovery and VDO operating modes, see **Section 42.2.5.3, “VDO operating modes”**.

**42.2.5.3. VDO operating modes**

This section describes the modes that indicate whether a VDO volume is operating normally or is recovering from an error.

You can display the current operating mode of a VDO volume using the **vdostats --verbose device** command. See the **Operating mode** attribute in the output.

**normal**

This is the default operating mode. VDO volumes are always in **normal** mode, unless either of the following states forces a different mode. A newly created VDO volume starts in **normal** mode.

**recovering**

When a VDO volume does not save all of its metadata before shutting down, it automatically enters **recovering** mode the next time it starts up. The typical reasons for entering this mode are sudden power loss or a problem from the underlying storage device. In **recovering** mode, VDO is fixing the references counts for each physical block of data on the device. Recovery usually does not take very long. The time depends on how large the VDO volume is, how fast the underlying storage device is, and how many other requests VDO is handling simultaneously. The VDO volume functions normally with the following exceptions:
- Initially, the amount of space available for write requests on the volume might be limited. As more of the metadata is recovered, more free space becomes available.

- Data written while the VDO volume is recovering might fail to deduplicate against data written before the crash if that data is in a portion of the volume that has not yet been recovered. VDO can compress data while recovering the volume. You can still read or overwrite compressed blocks.

- During an online recovery, certain statistics are unavailable: for example, blocks in use and blocks free. These statistics become available when the rebuild is complete.

- Response times for reads and writes might be slower than usual due to the ongoing recovery work.

You can safely shut down the VDO volume in recovering mode. If the recovery does not finish before shutting down, the device enters recovering mode again the next time that it starts up.

The VDO volume automatically exits recovering mode and moves to normal mode when it has fixed all the reference counts. No administrator action is necessary. For details, see Section 42.2.5.4, “Recovering a VDO volume online”.

read-only

When a VDO volume encounters a fatal internal error, it enters read-only mode. Events that might cause read-only mode include metadata corruption or the backing storage device becoming read-only. This mode is an error state.

In read-only mode, data reads work normally but data writes always fail. The VDO volume stays in read-only mode until an administrator fixes the problem.

You can safely shut down a VDO volume in read-only mode. The mode usually persists after the VDO volume is restarted. In rare cases, the VDO volume is not able to record the read-only state to the backing storage device. In these cases, VDO attempts to do a recovery instead.

Once a volume is in read-only mode, there is no guarantee that data on the volume has not been lost or corrupted. In such cases, Red Hat recommends copying the data out of the read-only volume and possibly restoring the volume from backup.

If the risk of data corruption is acceptable, it is possible to force an offline rebuild of the VDO volume metadata so the volume can be brought back online and made available. The integrity of the rebuilt data cannot be guaranteed. For details, see Section 42.2.5.5, “Forcing an offline rebuild of a VDO volume metadata”.

42.2.5.4. Recovering a VDO volume online

This procedure performs an online recovery on a VDO volume to recover metadata after an unclean shutdown.

Procedure

1. If the VDO volume is not already started, start it:

   ```
   # vdo start --name=my-vdo
   ```

   No additional steps are necessary. The recovery runs in the background.
2. If you rely on volume statistics like blocks in use and blocks free, wait until they are available.

42.2.5.5. Forcing an offline rebuild of a VDO volume metadata

This procedure performs a forced offline rebuild of a VDO volume metadata to recover after an unclean shutdown.

**WARNING**

This procedure might cause data loss on the volume.

**Prerequisites**

- The VDO volume is started.

**Procedure**

1. Check if the volume is in read-only mode. See the operating mode attribute in the command output:

   ```
   # vdo status --name=my-vdo
   ```

   If the volume is not in read-only mode, it is not necessary to force an offline rebuild. Perform an online recovery as described in Section 42.2.5.4, “Recovering a VDO volume online”.

2. Stop the volume if it is running:

   ```
   # vdo stop --name=my-vdo
   ```

3. Restart the volume with the **--forceRebuild** option:

   ```
   # vdo start --name=my-vdo --forceRebuild
   ```

42.2.5.6. Removing an unsuccessfully created VDO volume

This procedure cleans up a VDO volume in an intermediate state. A volume is left in an intermediate state if a failure occurs when creating the volume. This might happen when, for example:

- The system crashes
- Power fails
- The administrator interrupts a running vdo create command

**Procedure**

- To clean up, remove the unsuccessfully created volume with the **--force** option:

  ```
  # vdo remove --force --name=my-vdo
  ```
The `--force` option is required because the administrator might have caused a conflict by changing the system configuration since the volume was unsuccessfully created.

Without the `--force` option, the `vdo remove` command fails with the following message:

```
[...]
A previous operation failed.
Recovery from the failure either failed or was interrupted.
Add '--force' to 'remove' to perform the following cleanup.
Steps to clean up VDO my-vdo:
  umount -f /dev/mapper/my-vdo
  udevadm settle
  dmsetup remove my-vdo
vdo: ERROR - VDO volume my-vdo previous operation (create) is incomplete
```

42.2.6. Optimizing the UDS index

You can configure certain settings of the UDS index to optimize it on your system.

**IMPORTANT**

You cannot change the properties of the UDS index *after* creating the VDO volume.

42.2.6.1. Components of a VDO volume

VDO uses a block device as a backing store, which can include an aggregation of physical storage consisting of one or more disks, partitions, or even flat files. When a storage management tool creates a VDO volume, VDO reserves volume space for the UDS index and VDO volume. The UDS index and the VDO volume interact together to provide deduplicated block storage.

**Figure 42.4. VDO disk organization**

The VDO solution consists of the following components:

**kvdo**

A kernel module that loads into the Linux Device Mapper layer provides a deduplicated, compressed, and thinly provisioned block storage volume. The `kvdo` module exposes a block device. You can access this block device directly for block storage or present it through a Linux file system, such as XFS or ext4.

When `kvdo` receives a request to read a logical block of data from a VDO volume, it maps the requested logical block to the underlying physical block and then reads and returns the requested data.

When `kvdo` receives a request to write a block of data to a VDO volume, it first checks whether the request is a DISCARD or TRIM request or whether the data is uniformly zero. If either of these conditions is true, `kvdo` updates its block map and acknowledges the request. Otherwise, VDO
processes and optimizes the data.

**uds**
A kernel module that communicates with the Universal Deduplication Service (UDS) index on the volume and analyzes data for duplicates. For each new piece of data, UDS quickly determines if that piece is identical to any previously stored piece of data. If the index finds a match, the storage system can then internally reference the existing item to avoid storing the same information more than once. The UDS index runs inside the kernel as the **uds** kernel module.

**Command line tools**
For configuring and managing optimized storage.

**42.2.6.2. The UDS index**
VDO uses a high-performance deduplication index called UDS to detect duplicate blocks of data as they are being stored.

The UDS index provides the foundation of the VDO product. For each new piece of data, it quickly determines if that piece is identical to any previously stored piece of data. If the index finds a match, the storage system can then internally reference the existing item to avoid storing the same information more than once.

The UDS index runs inside the kernel as the **uds** kernel module.

The deduplication window is the number of previously written blocks that the index remembers. The size of the deduplication window is configurable. For a given window size, the index requires a specific amount of RAM and a specific amount of disk space. The size of the window is usually determined by specifying the size of the index memory using the **--indexMem=size** option. VDO then determines the amount of disk space to use automatically.

The UDS index consists of two parts:

- A compact representation is used in memory that contains at most one entry per unique block.
- An on-disk component that records the associated block names presented to the index as they occur, in order.

UDS uses an average of 4 bytes per entry in memory, including cache.

The on-disk component maintains a bounded history of data passed to UDS. UDS provides deduplication advice for data that falls within this deduplication window, containing the names of the most recently seen blocks. The deduplication window allows UDS to index data as efficiently as possible while limiting the amount of memory required to index large data repositories. Despite the bounded nature of the deduplication window, most datasets which have high levels of deduplication also exhibit a high degree of temporal locality — in other words, most deduplication occurs among sets of blocks that were written at about the same time. Furthermore, in general, data being written is more likely to duplicate data that was recently written than data that was written a long time ago. Therefore, for a given workload over a given time interval, deduplication rates will often be the same whether UDS indexes only the most recent data or all the data.

Because duplicate data tends to exhibit temporal locality, it is rarely necessary to index every block in the storage system. Were this not so, the cost of index memory would outstrip the savings of reduced storage costs from deduplication. Index size requirements are more closely related to the rate of data
ingestion. For example, consider a storage system with 100 TB of total capacity but with an ingestion rate of 1 TB per week. With a deduplication window of 4 TB, UDS can detect most redundancy among the data written within the last month.

42.2.6.3. Recommended UDS index configuration

This section describes the recommended options to use with the UDS index, based on your intended use case.

In general, Red Hat recommends using a \textit{sparse} UDS index for all production use cases. This is an extremely efficient indexing data structure, requiring approximately one-tenth of a byte of RAM per block in its deduplication window. On disk, it requires approximately 72 bytes of disk space per block. The minimum configuration of this index uses 256 MB of RAM and approximately 25 GB of space on disk.

To use this configuration, specify the \texttt{--sparseIndex=enabled --indexMem=0.25} options to the \texttt{vdo create} command. This configuration results in a deduplication window of 2.5 TB (meaning it will remember a history of 2.5 TB). For most use cases, a deduplication window of 2.5 TB is appropriate for deduplicating storage pools that are up to 10 TB in size.

The default configuration of the index, however, is to use a \textit{dense} index. This index is considerably less efficient (by a factor of 10) in RAM, but it has much lower (also by a factor of 10) minimum required disk space, making it more convenient for evaluation in constrained environments.

In general, a deduplication window that is one quarter of the physical size of a VDO volume is a recommended configuration. However, this is not an actual requirement. Even small deduplication windows (compared to the amount of physical storage) can find significant amounts of duplicate data in many use cases. Larger windows may also be used, but it in most cases, there will be little additional benefit to doing so.

Additional resources

- Speak with your Red Hat Technical Account Manager representative for additional guidelines on tuning this important system parameter.

42.2.7. Enabling or disabling deduplication in VDO

In some instances, you might want to temporarily disable deduplication of data being written to a VDO volume while still retaining the ability to read to and write from the volume. Disabling deduplication prevents subsequent writes from being deduplicated, but the data that was already deduplicated remains so.

42.2.7.1. Deduplication in VDO

Deduplication is a technique for reducing the consumption of storage resources by eliminating multiple copies of duplicate blocks.

Instead of writing the same data more than once, VDO detects each duplicate block and records it as a reference to the original block. VDO maintains a mapping from logical block addresses, which are used by the storage layer above VDO, to physical block addresses, which are used by the storage layer under VDO.

After deduplication, multiple logical block addresses can be mapped to the same physical block address. These are called shared blocks. Block sharing is invisible to users of the storage, who read and write blocks as they would if VDO were not present.
When a shared block is overwritten, VDO allocates a new physical block for storing the new block data to ensure that other logical block addresses that are mapped to the shared physical block are not modified.

42.2.7.2. Enabling deduplication on a VDO volume

This procedure restarts the associated UDS index and informs the VDO volume that deduplication is active again.

**NOTE**

Deduplication is enabled by default.

**Procedure**

- To restart deduplication on a VDO volume, use the following command:

  ```
  # vdo enableDeduplication --name=my-vdo
  ```

42.2.7.3. Disabling deduplication on a VDO volume

This procedure stops the associated UDS index and informs the VDO volume that deduplication is no longer active.

**Procedure**

- To stop deduplication on a VDO volume, use the following command:

  ```
  # vdo disableDeduplication --name=my-vdo
  ```

- You can also disable deduplication when creating a new VDO volume by adding the `--deduplication=disabled` option to the `vdo create` command.

42.2.8. Enabling or disabling compression in VDO

VDO provides data compression. You can disable it to maximize performance or to speed processing of data that is unlikely to compress, or re-enable it to increase space savings.

42.2.8.1. Compression in VDO

In addition to block-level deduplication, VDO also provides inline block-level compression using the HIOPS Compression™ technology.

VDO volume compression is on by default.

While deduplication is the optimal solution for virtual machine environments and backup applications, compression works very well with structured and unstructured file formats that do not typically exhibit block-level redundancy, such as log files and databases.

Compression operates on blocks that have not been identified as duplicates. When VDO sees unique data for the first time, it compresses the data. Subsequent copies of data that have already been stored are deduplicated without requiring an additional compression step.

The compression feature is based on a parallelized packaging algorithm that enables it to handle many compression operations at once. After first storing the block and responding to the requestor, a best-fit
packing algorithm finds multiple blocks that, when compressed, can fit into a single physical block. After it is determined that a particular physical block is unlikely to hold additional compressed blocks, it is written to storage and the uncompressed blocks are freed and reused.

By performing the compression and packaging operations after having already responded to the requestor, using compression imposes a minimal latency penalty.

### 42.2.8.2. Enabling compression on a VDO volume

This procedure enables compression on a VDO volume to increase space savings.

**NOTE**

Compression is enabled by default.

**Procedure**

- To start it again, use the following command:

  ```
  # vdo enableCompression --name=my-vdo
  ```

### 42.2.8.3. Disabling compression on a VDO volume

This procedure stops compression on a VDO volume to maximize performance or to speed processing of data that is unlikely to compress.

**Procedure**

- To stop compression on an existing VDO volume, use the following command:

  ```
  # vdo disableCompression --name=my-vdo
  ```

- Alternatively, you can disable compression by adding the `--compression=disabled` option to the `vdo create` command when creating a new volume.

### 42.2.9. Increasing the size of a VDO volume

You can increase the physical size of a VDO volume to utilize more underlying storage capacity, or the logical size to provide more capacity on the volume.

#### 42.2.9.1. The physical and logical size of a VDO volume

This section describes the physical size, available physical size, and logical size that VDO can utilize:

**Physical size**

This is the same size as the underlying block device. VDO uses this storage for:

- User data, which might be deduplicated and compressed
- VDO metadata, such as the UDS index

**Available physical size**

This is the portion of the physical size that VDO is able to use for user data
It is equivalent to the physical size minus the size of the metadata, minus the remainder after dividing the volume into slabs by the given slab size.

**Logical Size**

This is the provisioned size that the VDO volume presents to applications. It is usually larger than the available physical size. If the `--vdoLogicalSize` option is not specified, then the provisioning of the logical volume is now provisioned to a **1:1** ratio. For example, if a VDO volume is put on top of a 20 GB block device, then 2.5 GB is reserved for the UDS index (if the default index size is used). The remaining 17.5 GB is provided for the VDO metadata and user data. As a result, the available storage to consume is not more than 17.5 GB, and can be less due to metadata that makes up the actual VDO volume.

VDO currently supports any logical size up to 254 times the size of the physical volume with an absolute maximum logical size of 4PB.

**Figure 42.5. VDO disk organization**

In this figure, the VDO deduplicated storage target sits completely on top of the block device, meaning the physical size of the VDO volume is the same size as the underlying block device.

**Additional resources**

- For more information on how much storage VDO metadata requires on block devices of different sizes, see **Section 42.1.6.4, “Examples of VDO requirements by physical size”**.

**42.2.9.2. Thin provisioning in VDO**

VDO is a thinly provisioned block storage target. The amount of physical space that a VDO volume uses might differ from the size of the volume that is presented to users of the storage. You can make use of this disparity to save on storage costs.

**Out-of-space conditions**

Take care to avoid unexpectedly running out of storage space, if the data written does not achieve the expected rate of optimization.

Whenever the number of logical blocks (virtual storage) exceeds the number of physical blocks (actual storage), it becomes possible for file systems and applications to unexpectedly run out of space. For that reason, storage systems using VDO must provide you with a way of monitoring the size of the free pool on the VDO volume.

You can determine the size of this free pool by using the `vdostats` utility. The default output of this utility lists information for all running VDO volumes in a format similar to the Linux `df` utility. For example:

```
Device   1K-blocks Used  Available  Use%
/dev/mapper/vdo-name 211812352 105906176 105906176  50%
```
When the physical storage capacity of a VDO volume is almost full, VDO reports a warning in the system log, similar to the following:

```
Oct  2 17:27:39 system lvm[13863]: WARNING: VDO pool vdo-name is now 80.69% full.
Oct  2 17:28:19 system lvm[13863]: WARNING: VDO pool vdo-name is now 85.25% full.
Oct  2 17:29:39 system lvm[13863]: WARNING: VDO pool vdo-name is now 90.64% full.
Oct  2 17:30:29 system lvm[13863]: WARNING: VDO pool vdo-name is now 96.07% full.
```

**NOTE**

These warning messages appear only when the `lvm2-monitor` service is running. It is enabled by default.

**How to prevent out-of-space conditions**

If the size of free pool drops below a certain level, you can take action by:

- Deleting data. This reclaims space whenever the deleted data is not duplicated. Deleting data frees the space only after discards are issued.
- Adding physical storage

**IMPORTANT**

Monitor physical space on your VDO volumes to prevent out-of-space situations. Running out of physical blocks might result in losing recently written, unacknowledged data on the VDO volume.

**Thin provisioning and the TRIM and DISCARD commands**

To benefit from the storage savings of thin provisioning, the physical storage layer needs to know when data is deleted. File systems that work with thinly provisioned storage send **TRIM** or **DISCARD** commands to inform the storage system when a logical block is no longer required.

Several methods of sending the **TRIM** or **DISCARD** commands are available:

- With the `discard` mount option, the file systems can send these commands whenever a block is deleted.
- You can send the commands in a controlled manner by using utilities such as `fstrim`. These utilities tell the file system to detect which logical blocks are unused and send the information to the storage system in the form of a **TRIM** or **DISCARD** command.

The need to use **TRIM** or **DISCARD** on unused blocks is not unique to VDO. Any thinly provisioned storage system has the same challenge.

**42.2.9.3. Increasing the logical size of a VDO volume**

This procedure increases the logical size of a given VDO volume. It enables you to initially create VDO volumes that have a logical size small enough to be safe from running out of space. After some period of time, you can evaluate the actual rate of data reduction, and if sufficient, you can grow the logical size of the VDO volume to take advantage of the space savings.

It is not possible to decrease the logical size of a VDO volume.
Procedure

- To grow the logical size, use:

  ```
  # vdo growLogical --name=my-vdo
  --vdoLogicalSize=new-logical-size
  ```

  When the logical size increases, VDO informs any devices or file systems on top of the volume of the new size.

42.2.9.4. Increasing the physical size of a VDO volume

This procedure increases the amount of physical storage available to a VDO volume. It is not possible to shrink a VDO volume in this way.

Prerequisites

- The underlying block device has a larger capacity than the current physical size of the VDO volume.
  
  If it does not, you can attempt to increase the size of the device. The exact procedure depends on the type of the device. For example, to resize an MBR or GPT partition, see the Resizing a partition section in the Managing storage devices guide.

Procedure

- Add the new physical storage space to the VDO volume:

  ```
  # vdo growPhysical --name=my-vdo
  ```

42.2.10. Removing VDO volumes

You can remove an existing VDO volume on your system.

42.2.10.1. Removing a working VDO volume

This procedure removes a VDO volume and its associated UDS index.

Procedure

1. Unmount the file systems and stop the applications that are using the storage on the VDO volume.

2. To remove the VDO volume from your system, use:

  ```
  # vdo remove --name=my-vdo
  ```

42.2.10.2. Removing an unsuccessfully created VDO volume

This procedure cleans up a VDO volume in an intermediate state. A volume is left in an intermediate state if a failure occurs when creating the volume. This might happen when, for example:

- The system crashes
Power fails

The administrator interrupts a running `vdo create` command

**Procedure**

- To clean up, remove the unsuccessfully created volume with the **--force** option:

  ```
  # vdo remove --force --name=my-vdo
  ```

  The **--force** option is required because the administrator might have caused a conflict by changing the system configuration since the volume was unsuccessfully created.

  Without the **--force** option, the `vdo remove` command fails with the following message:

  ```
  [...] 
  A previous operation failed.
  Recovery from the failure either failed or was interrupted.
  Add '--force' to 'remove' to perform the following cleanup.
  Steps to clean up VDO my-vdo:
  umount -f /dev/mapper/my-vdo
  udevadm settle
  dmsetup remove my-vdo
  vdo: ERROR - VDO volume my-vdo previous operation (create) is incomplete
  ```

42.2.11. Additional resources

- You can use the **Ansible** tool to automate VDO deployment and administration. For details, see:
  - Ansible documentation: https://docs.ansible.com/
  - VDO Ansible module documentation: https://docs.ansible.com/ansible/latest/modules/vdo_module.html

42.3. DISCARDING UNUSED BLOCKS

You can perform or schedule discard operations on block devices that support them.

42.3.1. Block discard operations

Block discard operations discard blocks that are no longer in use by a mounted file system. They are useful on:

- Solid-state drives (SSDs)
- Thinly-provisioned storage

**Requirements**

The block device underlying the file system must support physical discard operations.

Physical discard operations are supported if the value in the `/sys/block/device/queue/discard_max_bytes` file is not zero.

42.3.2. Types of block discard operations
You can run discard operations using different methods:

**Batch discard**
Are run explicitly by the user. They discard all unused blocks in the selected file systems.

**Online discard**
Are specified at mount time. They run in real time without user intervention. Online discard operations discard only the blocks that are transitioning from used to free.

**Periodic discard**
Are batch operations that are run regularly by a `systemd` service.

All types are supported by the XFS and ext4 file systems and by VDO.

**Recommendations**
Red Hat recommends that you use batch or periodic discard.

Use online discard only if:

- the system’s workload is such that batch discard is not feasible, or
- online discard operations are necessary to maintain performance.

### 42.3.3. Performing batch block discard

This procedure performs a batch block discard operation to discard unused blocks on a mounted file system.

**Prerequisites**

- The file system is mounted.
- The block device underlying the file system supports physical discard operations.

**Procedure**

- Use the `fstrim` utility:
  - To perform discard only on a selected file system, use:
    ```
    # fstrim mount-point
    ```
  - To perform discard on all mounted file systems, use:
    ```
    # fstrim --all
    ```

If you execute the `fstrim` command on:

- a device that does not support discard operations, or
- a logical device (LVM or MD) composed of multiple devices, where any one of the device does not support discard operations,

the following message displays:
```sh
# fstrim /mnt/non_discard

fstrim: /mnt/non_discard: the discard operation is not supported
```

Additional resources

- fstrim(8) man page.

### 42.3.4. Enabling online block discard

This procedure enables online block discard operations that automatically discard unused blocks on all supported file systems.

**Procedure**

- Enable online discard at mount time:
  - When mounting a file system manually, add the `-o discard` mount option:
    ```sh
    # mount -o discard device mount-point
    ```
  - When mounting a file system persistently, add the `discard` option to the mount entry in the `/etc/fstab` file.

Additional resources

- mount(8) man page.
- fstab(5) man page.

### 42.3.5. Enabling periodic block discard

This procedure enables a systemd timer that regularly discards unused blocks on all supported file systems.

**Procedure**

- Enable and start the systemd timer:
  ```sh
  # systemctl enable --now fstrim.timer
  ```

### 42.4. USING THE WEB CONSOLE FOR MANAGING VIRTUAL DATA OPTIMIZER VOLUMES

Configure the Virtual Data Optimizer (VDO) using the RHEL 8 web console.

You will learn how to:

- Create VDO volumes
- Format VDO volumes
- Extend VDO volumes
Prerequisites

- The RHEL 8 web console is installed and accessible. For details, see Installing the web console.
- The cockpit-storaged package is installed on your system.

42.4.1. VDO volumes in the web console

Red Hat Enterprise Linux 8 supports Virtual Data Optimizer (VDO).

VDO is a block virtualization technology that combines:

Compression
For details, see Enabling or disabling compression in VDO.

Deduplication
For details, see Enabling or disabling deduplication in VDO.

Thin provisioning
For details, see Thinly-provisioned logical volumes (thin volumes).

Using these technologies, VDO:

- Saves storage space inline
- Compresses files
- Eliminates duplications
- Enables you to allocate more virtual space than how much the physical or logical storage provides
- Enables you to extend the virtual storage by growing

VDO can be created on top of many types of storage. In the RHEL 8 web console, you can configure VDO on top of:

- LVM
- Physical volume
- Software RAID

For details about placement of VDO in the Storage Stack, see System Requirements.

Additional resources
- For details about VDO, see Deduplicating and compressing storage.

42.4.2. Creating VDO volumes in the web console
Create a VDO volume in the RHEL web console.

Prerequisites

- Physical drives, LVMs, or RAID from which you want to create VDO.

Procedure

1. Log in to the RHEL 8 web console. For details, see Logging in to the web console.
2. Click Storage.
3. Click the + icon in the VDO Devices box.
4. In the Name field, enter a name of a VDO volume without spaces.
5. Select the drive that you want to use.
6. In the Logical Size bar, set up the size of the VDO volume. You can extend it more than ten times, but consider for what purpose you are creating the VDO volume:
   - For active VMs or container storage, use logical size that is ten times the physical size of the volume.
   - For object storage, use logical size that is three times the physical size of the volume.
   For details, see Deploying VDO.
7. In the Index Memory bar, allocate memory for the VDO volume. For details about VDO system requirements, see System Requirements.
8. Select the Compression option. This option can efficiently reduce various file formats. For details, see Enabling or disabling compression in VDO.
9. Select the Deduplication option. This option reduces the consumption of storage resources by eliminating multiple copies of duplicate blocks. For details, see Enabling or disabling deduplication in VDO.
10. [Optional] If you want to use the VDO volume with applications that need a 512 bytes block size, select Use 512 Byte emulation. This reduces the performance of the VDO volume, but should be very rarely needed. If in doubt, leave it off.
11. Click Create.
If the process of creating the VDO volume succeeds, you can see the new VDO volume in the Storage section and format it with a file system.

42.4.3. Formatting VDO volumes in the web console

VDO volumes act as physical drives. To use them, you need to format them with a file system.

**WARNING**

Formatting VDO will erase all data on the volume.

The following steps describe the procedure to format VDO volumes.

**Prerequisites**

- A VDO volume is created. For details, see Creating VDO volumes in the web console.

**Procedure**
1. Log in to the RHEL 8 web console. For details, see Logging in to the web console.

2. Click Storage.

3. Click the VDO volume.

4. Click on the Unrecognized Data tab.

5. Click Format.

6. In the Erase drop down menu, select:
   - Don’t overwrite existing data
     The RHEL web console rewrites only the disk header. The advantage of this option is the speed of formatting.
   - Overwrite existing data with zeros
     The RHEL web console rewrites the whole disk with zeros. This option is slower because the program has to go through the whole disk. Use this option if the disk includes any data and you need to rewrite them.

7. In the Type drop down menu, select a filesystem:
   - The XFS file system supports large logical volumes, switching physical drives online without outage, and growing. Leave this file system selected if you do not have a different strong preference.
     XFS does not support shrinking volumes. Therefore, you will not be able to reduce volume formatted with XFS.
   - The ext4 file system supports logical volumes, switching physical drives online without outage, growing, and shrinking.

   You can also select a version with the LUKS (Linux Unified Key Setup) encryption, which allows you to encrypt the volume with a passphrase.

8. In the Name field, enter the logical volume name.

9. In the Mounting drop down menu, select Custom. The Default option does not ensure that the file system will be mounted on the next boot.

10. In the Mount Point field, add the mount path.

11. Select Mount at boot.
12. Click **Format**.
Formatting can take several minutes depending on the used formatting options and the volume size.

After a successful finish, you can see the details of the formatted VDO volume on the **Filesystem** tab.

13. To use the VDO volume, click **Mount**.

At this point, the system uses the mounted and formatted VDO volume.

### 42.4.4. Extending VDO volumes in the web console

Extend VDO volumes in the RHEL 8 web console.

**Prerequisites**

- The **cockpit-storaged** package is installed on your system.
- The VDO volume created.
Procedure

1. Log in to the RHEL 8 web console.  
   For details, see Logging in to the web console.

2. Click Storage.

3. Click your VDO volume in the VDO Devices box.

4. In the VDO volume details, click the Grow button.

5. In the Grow logical size of VDO dialog box, extend the logical size of the VDO volume.

   Original size of the logical volume from the screenshot was 6 GB. As you can see, the RHEL web console enables you to grow the volume to more than ten times the size and it works correctly because of the compression and deduplication.

6. Click Grow.

   If the process of growing VDO succeeds, you can see the new size in the VDO volume details.
<table>
<thead>
<tr>
<th>Device File</th>
<th>/dev/mapper/myvirtualdataoptimizer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backing Device</td>
<td>/dev/md/127</td>
</tr>
<tr>
<td>Physical</td>
<td>1.11 MIB data + 3.72 GiB overhead used of 5.72 GiB (65%)</td>
</tr>
<tr>
<td>Logical</td>
<td>11.7 MIB used of 60 GiB (50% saved)</td>
</tr>
<tr>
<td>Index Memory</td>
<td>256 MIB</td>
</tr>
<tr>
<td>Compression</td>
<td>ON</td>
</tr>
<tr>
<td>Deduplication</td>
<td>ON</td>
</tr>
</tbody>
</table>
PART V. DESIGN OF LOG FILE
CHAPTER 43. AUDITING THE SYSTEM

Audit does not provide additional security to your system; rather, it can be used to discover violations of security policies used on your system. These violations can further be prevented by additional security measures such as SELinux.

43.1. LINUX AUDIT

The Linux Audit system provides a way to track security-relevant information on your system. Based on pre-configured rules, Audit generates log entries to record as much information about the events that are happening on your system as possible. This information is crucial for mission-critical environments to determine the violator of the security policy and the actions they performed.

The following list summarizes some of the information that Audit is capable of recording in its log files:

- Date and time, type, and outcome of an event.
- Sensitivity labels of subjects and objects.
- Association of an event with the identity of the user who triggered the event.
- All modifications to Audit configuration and attempts to access Audit log files.
- All uses of authentication mechanisms, such as SSH, Kerberos, and others.
- Changes to any trusted database, such as `/etc/passwd`.
- Attempts to import or export information into or from the system.
- Include or exclude events based on user identity, subject and object labels, and other attributes.

The use of the Audit system is also a requirement for a number of security-related certifications. Audit is designed to meet or exceed the requirements of the following certifications or compliance guides:

- Controlled Access Protection Profile (CAPP)
- Labeled Security Protection Profile (LSPP)
- Rule Set Base Access Control (RSBAC)
- National Industrial Security Program Operating Manual (NISPOM)
- Federal Information Security Management Act (FISMA)
- Payment Card Industry – Data Security Standard (PCI-DSS)
- Security Technical Implementation Guides (STIG)

Audit has also been:

- Evaluated by National Information Assurance Partnership (NIAP) and Best Security Industries (BSI).
- Certified to LSPP/CAPP/RSBAC/EAL4+ on Red Hat Enterprise Linux 5.
Use Cases

Watching file access
Audit can track whether a file or a directory has been accessed, modified, executed, or the file's attributes have been changed. This is useful, for example, to detect access to important files and have an Audit trail available in case one of these files is corrupted.

Monitoring system calls
Audit can be configured to generate a log entry every time a particular system call is used. This can be used, for example, to track changes to the system time by monitoring the `settimeofday`, `clock_adjtime`, and other time-related system calls.

Recording commands run by a user
Audit can track whether a file has been executed, so rules can be defined to record every execution of a particular command. For example, a rule can be defined for every executable in the `/bin` directory. The resulting log entries can then be searched by user ID to generate an audit trail of executed commands per user.

Recording execution of system pathnames
Aside from watching file access which translates a path to an inode at rule invocation, Audit can now watch the execution of a path even if it does not exist at rule invocation, or if the file is replaced after rule invocation. This allows rules to continue to work after upgrading a program executable or before it is even installed.

Recording security events
The `pam_faillock` authentication module is capable of recording failed login attempts. Audit can be set up to record failed login attempts as well and provides additional information about the user who attempted to log in.

Searching for events
Audit provides the `ausearch` utility, which can be used to filter the log entries and provide a complete audit trail based on several conditions.

Running summary reports
The `aureport` utility can be used to generate, among other things, daily reports of recorded events. A system administrator can then analyze these reports and investigate suspicious activity further.

Monitoring network access
The `nftables`, `iptables`, and `ebtables` utilities can be configured to trigger Audit events, allowing system administrators to monitor network access.

NOTE
System performance may be affected depending on the amount of information that is collected by Audit.

43.2. AUDIT SYSTEM ARCHITECTURE

The Audit system consists of two main parts: the user-space applications and utilities, and the kernel-side system call processing. The kernel component receives system calls from user-space applications and filters them through one of the following filters: `user`, `task`, `fstype`, or `exit`.

Once a system call passes the `exclude` filter, it is sent through one of the aforementioned filters, which, based on the Audit rule configuration, sends it to the Audit daemon for further processing.
The user-space Audit daemon collects the information from the kernel and creates entries in a log file. Other Audit user-space utilities interact with the Audit daemon, the kernel Audit component, or the Audit log files:

- `auditctl` — the Audit control utility interacts with the kernel Audit component to manage rules and to control many settings and parameters of the event generation process.

- The remaining Audit utilities take the contents of the Audit log files as input and generate output based on user’s requirements. For example, the `aureport` utility generates a report of all recorded events.

In RHEL 8, the Audit dispatcher daemon (`audisp`) functionality is integrated in the Audit daemon (`auditd`). Configuration files of plugins for the interaction of real-time analytical programs with Audit events are located in the `/etc/audit/plugins.d/` directory by default.

### 43.3. CONFIGURING AUDITD FOR A SECURE ENVIRONMENT

The default `auditd` configuration should be suitable for most environments. However, if your environment has to meet strict security policies, the following settings are suggested for the Audit daemon configuration in the `/etc/audit/auditd.conf` file:

**log_file**

The directory that holds the Audit log files (usually `/var/log/audit/`) should reside on a separate mount point. This prevents other processes from consuming space in this directory and provides accurate detection of the remaining space for the Audit daemon.

**max_log_file**

Specifies the maximum size of a single Audit log file, must be set to make full use of the available space on the partition that holds the Audit log files.

**max_log_file_action**

Decides what action is taken once the limit set in `max_log_file` is reached, should be set to `keep_logs` to prevent Audit log files from being overwritten.

**space_left**

Specifies the amount of free space left on the disk for which an action that is set in the `space_left_action` parameter is triggered. Must be set to a number that gives the administrator enough time to respond and free up disk space. The `space_left` value depends on the rate at which the Audit log files are generated.

**space_left_action**

It is recommended to set the `space_left_action` parameter to `email` or `exec` with an appropriate notification method.

**admin_space_left**

Specifies the absolute minimum amount of free space for which an action that is set in the `admin_space_left_action` parameter is triggered, must be set to a value that leaves enough space to log actions performed by the administrator.

**admin_space_left_action**

Should be set to `single` to put the system into single-user mode and allow the administrator to free up some disk space.

**disk_full_action**

Specifies an action that is triggered when no free space is available on the partition that holds the Audit log files, must be set to `halt` or `single`. This ensures that the system is either shut down or operating in single-user mode when Audit can no longer log events.
**disk_error_action**

Specifies an action that is triggered in case an error is detected on the partition that holds the Audit log files, must be set to **syslog**, **single**, or **halt**, depending on your local security policies regarding the handling of hardware malfunctions.

**flush**

Should be set to **incremental_async**. It works in combination with the **freq** parameter, which determines how many records can be sent to the disk before forcing a hard synchronization with the hard drive. The **freq** parameter should be set to **100**. These parameters assure that Audit event data is synchronized with the log files on the disk while keeping good performance for bursts of activity.

The remaining configuration options should be set according to your local security policy.

### 43.4. STARTING AND CONTROLLING AUDITD

After **auditd** is configured, start the service to collect Audit information and store it in the log files. Use the following command as the root user to start **auditd**:

```bash
# service auditd start
```

To configure **auditd** to start at boot time:

```bash
# systemctl enable auditd
```

You can temporarily disable **auditd** with the `# auditctl -e 0` command and re-enable it with `# auditctl -e 1`.

A number of other actions can be performed on **auditd** using the `service auditd action` command, where **action** can be one of the following:

- **stop**
  - Stops **auditd**.
- **restart**
  - Restarts **auditd**.
- **reload** or **force-reload**
  - Reloads the configuration of **auditd** from the `/etc/audit/auditd.conf` file.
- **rotate**
  - Rotates the log files in the `/var/log/audit` directory.
- **resume**
  - Resumes logging of Audit events after it has been previously suspended, for example, when there is not enough free space on the disk partition that holds the Audit log files.
- **condrestart** or **try-restart**
  - Restarts **auditd** only if it is already running.
- **status**
  - Displays the running status of **auditd**.
NOTE

The **service** command is the only way to correctly interact with the **auditd** daemon. You need to use the **service** command so that the **auid** value is properly recorded. You can use the **systemctl** command only for two actions: **enable** and **status**.

### 43.5. UNDERSTANDING AUDIT LOG FILES

By default, the Audit system stores log entries in the `/var/log/audit/audit.log` file; if log rotation is enabled, rotated **audit.log** files are stored in the same directory.

Add the following Audit rule to log every attempt to read or modify the `/etc/ssh/sshd_config` file:

```bash
# auditctl -w /etc/ssh/sshd_config -p warx -k sshd_config
```

If the **auditd** daemon is running, for example, using the following command creates a new event in the Audit log file:

```bash
$ cat /etc/ssh/sshd_config
```

This event in the **audit.log** file looks as follows:

```
# type=SYSCALL msg=audit(1364481363.243:24287): arch=c000003e syscall=2 success=no exit=-13
# a0=7fffd19c5592 a1=0 a2=7fffd19c4b50 a3=a items=1 ppid=2686 pid=3538 auid=1000
# gid=1000 euid=1000 fsuid=1000 seuser=1000 sgid=1000 fsgid=1000 tty=pts0 ses=1
# comm="cat" exe="/bin/cat" subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
# key="sshd_config"
# type=CWD msg=audit(1364481363.243:24287): cwd="/home/shadowman"
# type=PATH msg=audit(1364481363.243:24287): item=0 name="/etc/ssh/sshd_config"
# ino=409248 dev=fd:00 mode=0100600 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:etc_t:s0
# nametype=NORMAL cap_fp=none cap_fi=none cap_fe=0 cap_fver=0
# type=PROCTITLE msg=audit(1364481363.243:24287) proctitle=636174002F6574632F7373682F737368645F636F6E666967
```

The above event consists of four records, which share the same time stamp and serial number. Records always start with the **type=** keyword. Each record consists of several **name=value** pairs separated by a white space or a comma. A detailed analysis of the above event follows:

**First Record**

**type=SYSCALL**

The **type** field contains the type of the record. In this example, the **SYSCALL** value specifies that this record was triggered by a system call to the kernel.

**msg=audit(1364481363.243:24287):**

The **msg** field records:

- a time stamp and a unique ID of the record in the form `audit(time_stamp:ID)`. Multiple records can share the same time stamp and ID if they were generated as part of the same Audit event. The time stamp is using the Unix time format - seconds since 00:00:00 UTC on 1 January 1970.

- various event-specific **name=value** pairs provided by the kernel or user-space applications.
The `arch` field contains information about the CPU architecture of the system. The value, `c000003e`, is encoded in hexadecimal notation. When searching Audit records with the `ausearch` command, use the `-i` or `--interpret` option to automatically convert hexadecimal values into their human-readable equivalents. The `c000003e` value is interpreted as `x86_64`.

The `syscall` field records the type of the system call that was sent to the kernel. The value, `2`, can be matched with its human-readable equivalent in the `/usr/include/asm/unistd_64.h` file. In this case, `2` is the `open` system call. Note that the `ausyscall` utility allows you to convert system call numbers to their human-readable equivalents. Use the `ausyscall --dump` command to display a listing of all system calls along with their numbers. For more information, see the `ausyscall(8)` man page.

The `success` field records whether the system call recorded in that particular event succeeded or failed. In this case, the call did not succeed.

The `exit` field contains a value that specifies the exit code returned by the system call. This value varies for a different system call. You can interpret the value to its human-readable equivalent with the following command:

```
# ausearch --interpret --exit -13
```

Note that the previous example assumes that your Audit log contains an event that failed with exit code `-13`.

The `a0` to `a3` fields record the first four arguments, encoded in hexadecimal notation, of the system call in this event. These arguments depend on the system call that is used; they can be interpreted by the `ausearch` utility.

The `items` field contains the number of PATH auxiliary records that follow the syscall record.

The `ppid` field records the Parent Process ID (PPID). In this case, `2686` was the PPID of the parent process such as `bash`.

The `pid` field records the Process ID (PID). In this case, `3538` was the PID of the `cat` process.

The `auid` field records the Audit user ID, that is the loginuid. This ID is assigned to a user upon login and is inherited by every process even when the user’s identity changes, for example, by switching user accounts with the `su - john` command.

The `uid` field records the user ID of the user who started the analyzed process. The user ID can be interpreted into user names with the following command: `ausearch -i --uid UID`.

The `gid` field records the group ID of the user who started the analyzed process.

The `euid` field records the effective user ID of the user who started the analyzed process.

The `suid` field records the effective user ID of the user who started the analyzed process.
The **suid** field records the set user ID of the user who started the analyzed process.

**fsuid=1000**

The **fsuid** field records the file system user ID of the user who started the analyzed process.

**egid=1000**

The **egid** field records the effective group ID of the user who started the analyzed process.

**sgid=1000**

The **sgid** field records the set group ID of the user who started the analyzed process.

**fsgid=1000**

The **fsgid** field records the file system group ID of the user who started the analyzed process.

**tty=pts0**

The **tty** field records the terminal from which the analyzed process was invoked.

**ses=1**

The **ses** field records the session ID of the session from which the analyzed process was invoked.

**comm="cat"**

The **comm** field records the command-line name of the command that was used to invoke the analyzed process. In this case, the **cat** command was used to trigger this Audit event.

**exe="/bin/cat"**

The **exe** field records the path to the executable that was used to invoke the analyzed process.

**subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023**

The **subj** field records the SELinux context with which the analyzed process was labeled at the time of execution.

**key="sshd_config"**

The **key** field records the administrator-defined string associated with the rule that generated this event in the Audit log.

---

**Second Record**

**type=CWD**

In the second record, the **type** field value is **CWD** – current working directory. This type is used to record the working directory from which the process that invoked the system call specified in the first record was executed.

The purpose of this record is to record the current process’s location in case a relative path winds up being captured in the associated PATH record. This way the absolute path can be reconstructed.

**msg=audit(1364481363.243:24287)**

The **msg** field holds the same time stamp and ID value as the value in the first record. The time stamp is using the Unix time format - seconds since 00:00:00 UTC on 1 January 1970.

**cwd="/home/user_name"**

The **cwd** field contains the path to the directory in which the system call was invoked.

---

**Third Record**

**type=PATH**

In the third record, the **type** field value is **PATH**. An Audit event contains a **PATH**-type record for every path that is passed to the system call as an argument. In this Audit event, only one path (/etc/ssh/sshd_config) was used as an argument.
msg=audit(1364481363.243:24287):
The msg field holds the same time stamp and ID value as the value in the first and second record.

item=0
The item field indicates which item, of the total number of items referenced in the SYSCALL type record, the current record is. This number is zero-based; a value of 0 means it is the first item.

name="/etc/ssh/sshd_config"
The name field records the path of the file or directory that was passed to the system call as an argument. In this case, it was the /etc/ssh/sshd_config file.

inode=409248
The inode field contains the inode number associated with the file or directory recorded in this event. The following command displays the file or directory that is associated with the 409248 inode number:

```
# find / -inum 409248 -print
/etc/ssh/sshd_config
```

dev=fd:00
The dev field specifies the minor and major ID of the device that contains the file or directory recorded in this event. In this case, the value represents the /dev/fd/0 device.

mode=0100600
The mode field records the file or directory permissions, encoded in numerical notation as returned by the stat command in the st_mode field. See the stat(2) man page for more information. In this case, 0100600 can be interpreted as -rw------, meaning that only the root user has read and write permissions to the /etc/ssh/sshd_config file.

ouid=0
The ouid field records the object owner’s user ID.

ogid=0
The ogid field records the object owner’s group ID.

rdev=00:00
The rdev field contains a recorded device identifier for special files only. In this case, it is not used as the recorded file is a regular file.

obj=system_u:object_r:etc_t:s0
The obj field records the SELinux context with which the recorded file or directory was labeled at the time of execution.

nametype=NORMAL
The nametype field records the intent of each path record’s operation in the context of a given syscall.

cap_fp=none
The cap_fp field records data related to the setting of a permitted file system-based capability of the file or directory object.

cap_fi=none
The cap_fi field records data related to the setting of an inherited file system-based capability of the file or directory object.

cap_fe=0
The cap_fe field records the setting of the effective bit of the file system-based capability of the file or directory object.
The `cap_fver` field records the version of the file system-based capability of the file or directory object.

**Fourth Record**

type=PROCTITLE

The `type` field contains the type of the record. In this example, the `PROCTITLE` value specifies that this record gives the full command-line that triggered this Audit event, triggered by a system call to the kernel.

proctitle=636174002F6574632F7373682F737368645F636F6E666967

The `proctitle` field records the full command-line of the command that was used to invoke the analyzed process. The field is encoded in hexadecimal notation to not allow the user to influence the Audit log parser. The text decodes to the command that triggered this Audit event. When searching Audit records with the `ausearch` command, use the `-i` or `--interpret` option to automatically convert hexadecimal values into their human-readable equivalents. The `636174002F6574632F7373682F737368645F636F6E666967` value is interpreted as `cat /etc/ssh/sshd_config`.

### 43.6. USING AUDITCTL FOR DEFINING AND EXECUTING AUDIT RULES

The Audit system operates on a set of rules that define what is captured in the log files. Audit rules can be set either on the command line using the `auditctl` utility or in the `/etc/audit/rules.d/` directory.

The `auditctl` command enables you to control the basic functionality of the Audit system and to define rules that decide which Audit events are logged.

**File-system rules examples**

1. To define a rule that logs all write access to, and every attribute change of, the `/etc/passwd` file:
   ```
   # auditctl -w /etc/passwd -p wa -k passwd_changes
   ```

2. To define a rule that logs all write access to, and every attribute change of, all the files in the `/etc/selinux/` directory:
   ```
   # auditctl -w /etc/selinux/ -p wa -k selinux_changes
   ```

**System-call rules examples**

1. To define a rule that creates a log entry every time the `adjtimex` or `settimeofday` system calls are used by a program, and the system uses the 64-bit architecture:
   ```
   # auditctl -a always,exit -F arch=b64 -S adjtimex -S settimeofday -k time_change
   ```

2. To define a rule that creates a log entry every time a file is deleted or renamed by a system user whose ID is 1000 or larger:
   ```
   # auditctl -a always,exit -S unlink -S unlinkat -S rename -S renameat -F auid>=1000 -F auid!=4294967295 -k delete
   ```

   Note that the `-F auid!=4294967295` option is used to exclude users whose login UID is not set.
Executable-file rules

To define a rule that logs all execution of the /bin/id program, execute the following command:

```bash
# auditctl -a always,exit -F exe=/bin/id -F arch=b64 -S execve -k execution_bin_id
```

Additional resources

- [auditctl(8) man page.](#)

43.7. DEFINING PERSISTENT AUDIT RULES

To define Audit rules that are persistent across reboots, you must either directly include them in the `/etc/audit/rules.d/audit.rules` file or use the `augenrules` program that reads rules located in the `/etc/audit/rules.d/` directory.

Note that the `/etc/audit/audit.rules` file is generated whenever the `auditd` service starts. Files in `/etc/audit/rules.d/` use the same `auditctl` command-line syntax to specify the rules. Empty lines and text following a hash sign (#) are ignored.

Furthermore, you can use the `auditctl` command to read rules from a specified file using the `-R` option, for example:

```bash
# auditctl -R /usr/share/audit/sample-rules/30-stig.rules
```

43.8. USING PRE-CONFIGURED RULES FILES

In the `/usr/share/audit/sample-rules` directory, the `audit` package provides a set of pre-configured rules files according to various certification standards:

- **30-nispom.rules**
  Audit rule configuration that meets the requirements specified in the Information System Security chapter of the National Industrial Security Program Operating Manual.

- **30-ospp-v42*.rules**
  Audit rule configuration that meets the requirements defined in the OSPP (Protection Profile for General Purpose Operating Systems) profile version 4.2.

- **30-pci-dss-v31.rules**
  Audit rule configuration that meets the requirements set by Payment Card Industry Data Security Standard (PCI DSS) v3.1.

- **30-stig.rules**
  Audit rule configuration that meets the requirements set by Security Technical Implementation Guides (STIG).

To use these configuration files, copy them to the `/etc/audit/rules.d/` directory and use the `augenrules --load` command, for example:

```bash
# cd /usr/share/audit/sample-rules/
# cp 10-base-config.rules 30-stig.rules 31-privileged.rules 99-finalize.rules /etc/audit/rules.d/
# augenrules --load
```
You can order Audit rules using a numbering scheme. See the `/usr/share/audit/sample-rules/README-rules` file for more information.

Additional resources

- `audit.rules(7)` man page.

### 43.9. USING AUGENRULES TO DEFINE PERSISTENT RULES

The `augenrules` script reads rules located in the `/etc/audit/rules.d/` directory and compiles them into an `audit.rules` file. This script processes all files that end with `.rules` in a specific order based on their natural sort order. The files in this directory are organized into groups with the following meanings:

- 10 - Kernel and auditctl configuration
- 20 - Rules that could match general rules but you want a different match
- 30 - Main rules
- 40 - Optional rules
- 50 - Server-specific rules
- 70 - System local rules
- 90 - Finalize (immutable)

The rules are not meant to be used all at once. They are pieces of a policy that should be thought out and individual files copied to `/etc/audit/rules.d/`. For example, to set a system up in the STIG configuration, copy rules `10-base-config`, `30-stig`, `31-privileged`, and `99-finalize`.

Once you have the rules in the `/etc/audit/rules.d/` directory, load them by running the `augenrules` script with the `--load` directive:

```
# augenrules --load
/sbin/augenrules: No change
No rules
enabled 1
failure 1
pid 742
rate_limit 0
... 
```

Additional resources

- `audit.rules(8)` and `augenrules(8)` man pages.

### 43.10. DISABLING AUGENRULES

Use the following steps to disable the `augenrules` utility. This switches Audit to use rules defined in the `/etc/audit/audit.rules` file.

**Procedure**
1. Copy the `/usr/lib/systemd/system/auditd.service` file to the `/etc/systemd/system/` directory:

   ```
   # cp -f /usr/lib/systemd/system/auditd.service /etc/systemd/system/
   ```

2. Edit the `/etc/systemd/system/auditd.service` file in a text editor of your choice, for example:

   ```
   # vi /etc/systemd/system/auditd.service
   ```

3. Comment out the line containing `augenrules`, and uncomment the line containing the `auditctl -R` command:

   ```
   #ExecStartPost=-/sbin/augenrules --load
   ExecStartPost=-/sbin/auditctl -R /etc/audit/audit.rules
   ```

4. Reload the `systemd` daemon to fetch changes in the `auditd.service` file:

   ```
   # systemctl daemon-reload
   ```

5. Restart the `auditd` service:

   ```
   # service auditd restart
   ```

Additional resources

- `augenrules(8)` and `audit.rules(8)` man pages.
- `Auditd service restart overrides changes made to /etc/audit/audit.rules`.

43.11. SETTING UP AUDIT TO MONITOR SOFTWARE UPDATES

In RHEL 8.6 and later versions, you can use the pre-configured rule `44-installers.rules` to configure Audit to monitor the following utilities that install software:

- `dnf`

  **NOTE**

  Because `dnf` is a symlink in RHEL, the path in the `dnf` Audit rule must include the target of the symlink. To receive correct Audit events, modify the `44-installers.rules` file by changing the `path=/usr/bin/dnf` path to `/usr/bin/dnf-3`.

- `yum`
- `pip`
- `npm`
- `cpan`
- `gem`
- `luarocks`
By default, `rpm` already provides audit `SOFTWARE_UPDATE` events when it installs or updates a package. You can list them by entering `ausearch -m SOFTWARE_UPDATE` on the command line.

In RHEL 8.5 and earlier versions, you can manually add rules to monitor utilities that install software into a `.rules` file within the `/etc/audit/rules.d/` directory.

**NOTE**

Pre-configured rule files cannot be used on systems with the `ppc64le` and `aarch64` architectures.

Prerequisites

- `auditd` is configured in accordance with the settings provided in [Configuring auditd for a secure environment](#).

Procedure

1. On RHEL 8.6 and later, copy the pre-configured rule file `44-installers.rules` from the `/usr/share/audit/sample-rules/` directory to the `/etc/audit/rules.d/` directory:

   ```
   # cp /usr/share/audit/sample-rules/44-installers.rules /etc/audit/rules.d/
   ```

   On RHEL 8.5 and earlier, create a new file in the `/etc/audit/rules.d/` directory named `44-installers.rules`, and insert the following rules:

   ```
   -a always,exit -F perm=x -F path=/usr/bin/dnf -F key=software-installer
   -a always,exit -F perm=x -F path=/usr/bin/yum -F
   ```

   You can add additional rules for other utilities that install software, for example `pip` and `npm`, using the same syntax.

2. Load the audit rules:

   ```
   # augenrules --load
   ```

Verification

1. List the loaded rules:

   ```
   # auditctl -l
   -p x-w /usr/bin/dnf-3 -k software-installer
   -p x-w /usr/bin/yum -k software-installer
   -p x-w /usr/bin/pip -k software-installer
   -p x-w /usr/bin/npm -k software-installer
   -p x-w /usr/bin/cpan -k software-installer
   -p x-w /usr/bin/gem -k software-installer
   -p x-w /usr/bin/luarocks -k software-installer
   ```

2. Perform an installation, for example:

   ```
   # dnf reinstall -y vim-enhanced
   ```
3. Search the Audit log for recent installation events, for example:

```
# ausearch -ts recent -k software-installer
```

```
time->Thu Dec 16 10:33:46 2021
```
You can use the `-sv yes` option to filter out successful login attempts and `-sv no` for unsuccessful login attempts.

- Pipe the raw output of the `ausearch` command into the `aulast` utility, which displays the output in a format similar to the output of the `last` command. For example:

```
# ausearch --raw | aulast --stdin
root  ssh  10.37.128.108  Mon Nov 22 07:33 - 07:33  (00:00)
root  ssh  10.37.128.108  Mon Nov 22 07:33 - 07:33  (00:00)
root  ssh  10.22.16.106   Mon Nov 22 07:40 - 07:40  (00:00)
reboot  system boot  4.18.0-348.6.el8  Mon Nov 22 07:33
```

- Display the list of login events by using the `aureport` command with the `--login -i` options.

```
# aureport --login -i

Login Report
============================================
# date time auid host term exe success event
============================================
1. 11/16/2021 13:11:30 root 10.40.192.190 ssh /usr/sbin/sshd yes 6920
2. 11/16/2021 13:11:31 root 10.40.192.190 ssh /usr/sbin/sshd yes 6925
5. 11/16/2021 13:11:33 root 10.40.192.190 ssh /usr/sbin/sshd yes 6940
6. 11/16/2021 13:11:33 root 10.40.192.190 /dev/pts/0 /usr/sbin/sshd yes 6945
```

Additional resources
- The `ausearch(8)` man page.
- The `aulast(8)` man page.
- The `aureport(8)` man page.

43.13. ADDITIONAL RESOURCES
- The RHEL Audit System Reference Knowledgebase article.
- The Auditd execution options in a container Knowledgebase article.
- The Linux Audit Documentation Project page.
- The `audit` package provides documentation in the `/usr/share/doc/audit/` directory.
- `auditd(8), auditctl(8), ausearch(8), audit.rules(7), audispd.conf(5), audispd(8), auditd.conf(5), ausearch-expression(5), aulast(8), aulastlog(8), aureport(8), ausyscall(8), autrace(8), and auvirt(8)` man pages.
PART VI. DESIGN OF KERNEL
CHAPTER 44. THE LINUX KERNEL RPM

The following sections describe the Linux kernel RPM package provided and maintained by Red Hat.

44.1. WHAT AN RPM IS

An RPM package is a file containing other files and their metadata (information about the files that are needed by the system).

Specifically, an RPM package consists of the cpio archive.

The cpio archive contains:

- Files
- RPM header (package metadata)
  The rpm package manager uses this metadata to determine dependencies, where to install files, and other information.

Types of RPM packages

There are two types of RPM packages. Both types share the file format and tooling, but have different contents and serve different purposes:

- Source RPM (SRPM)
  An SRPM contains source code and a SPEC file, which describes how to build the source code into a binary RPM. Optionally, the patches to source code are included as well.

- Binary RPM
  A binary RPM contains the binaries built from the sources and patches.

44.2. THE LINUX KERNEL RPM PACKAGE OVERVIEW

The kernel RPM is a meta package that does not contain any files, but rather ensures that the following required sub-packages are properly installed:

- kernel-core - contains the binary image of the kernel, all initramfs-related objects to bootstrap the system, and a minimal number of kernel modules to ensure core functionality. This sub-package alone could be used in virtualized and cloud environments to provide a Red Hat Enterprise Linux 8 kernel with a quick boot time and a small disk size footprint.

- kernel-modules - contains the remaining kernel modules that are not present in kernel-core.

The small set of kernel sub-packages above aims to provide a reduced maintenance surface to system administrators especially in virtualized and cloud environments.

Optional kernel packages are for example:

- kernel-modules-extra - contains kernel modules for rare hardware and modules which loading is disabled by default.

- kernel-debug – contains a kernel with numerous debugging options enabled for kernel diagnosis, at the expense of reduced performance.

- kernel-tools – contains tools for manipulating the Linux kernel and supporting documentation.
- **kernel-devel** – contains the kernel headers and makefiles sufficient to build modules against the kernel package.

- **kernel-abi-whitelists** – contains information pertaining to the RHEL kernel ABI, including a list of kernel symbols that are needed by external Linux kernel modules and a `yum` plug-in to aid enforcement.

- **kernel-headers** – includes the C header files that specify the interface between the Linux kernel and user-space libraries and programs. The header files define structures and constants that are needed for building most standard programs.

### Additional resources

- *What are the kernel-core, kernel-modules, and kernel-modules-extras packages?*

### 44.3. DISPLAYING CONTENTS OF THE KERNEL PACKAGE

The following procedure describes how to view the contents of the kernel package and its sub-packages without installing them using the `rpm` command.

#### Prerequisites

- Obtained **kernel**, **kernel-core**, **kernel-modules**, **kernel-modules-extra** RPM packages for your CPU architecture

#### Procedure

- List modules for **kernel**:

  ```bash
  $ rpm -qlp <kernel_rpm>
  (contains no files)
  ...
  ``

- List modules for **kernel-core**:

  ```bash
  $ rpm -qlp <kernel-core_rpm>
  ...
  /lib/modules/4.18.0-80.el8.x86_64/kernel/fs/udf/udf.ko.xz
  /lib/modules/4.18.0-80.el8.x86_64/kernel/fs/xfs
  /lib/modules/4.18.0-80.el8.x86_64/kernel/fs/xfs/xfs.ko.xz
  /lib/modules/4.18.0-80.el8.x86_64/kernel/kernel
  /lib/modules/4.18.0-80.el8.x86_64/kernel/kernel/trace
  /lib/modules/4.18.0-80.el8.x86_64/kernel/kernel/trace/ring_buffer_benchmark.ko.xz
  /lib/modules/4.18.0-80.el8.x86_64/kernel/lib
  /lib/modules/4.18.0-80.el8.x86_64/kernel/lib/cordic.ko.xz
  ...
  ``

- List modules for **kernel-modules**:

  ```bash
  $ rpm -qlp <kernel-modules_rpm>
  ...
  /lib/modules/4.18.0-80.el8.x86_64/kernel/drivers/infiniband/hw/mlx4/mlx4_ib.ko.xz
  /lib/modules/4.18.0-80.el8.x86_64/kernel/drivers/infiniband/hw/mlx5/mlx5_ib.ko.xz
  /lib/modules/4.18.0-80.el8.x86_64/kernel/drivers/infiniband/hw/qedr/qedr.ko.xz
  ```
List modules for `kernel-modules-extra`:

```
$ rpm -qlp <kernel-modules-extra_rpm>
```

...  
```
/lib/modules/4.18.0-80.el8.x86_64/extra/net/sched/sch_cbq.ko.xz
/lib/modules/4.18.0-80.el8.x86_64/extra/net/sched/sch_choke.ko.xz
/lib/modules/4.18.0-80.el8.x86_64/extra/net/sched/sch_drr.ko.xz
/lib/modules/4.18.0-80.el8.x86_64/extra/net/sched/sch_dsmark.ko.xz
/lib/modules/4.18.0-80.el8.x86_64/extra/net/sched/sch_gred.ko.xz
...```

Additional resources

- `rpm(8)` manual page
- `RPM packages`
- `RPM packages`
- `RPM packages`
CHAPTER 45. UPDATING KERNEL WITH YUM

The following sections bring information about the Linux kernel provided and maintained by Red Hat (Red Hat kernel), and how to keep the Red Hat kernel updated. As a consequence, the operating system will have all the latest bug fixes, performance enhancements, and patches ensuring compatibility with new hardware.

45.1. WHAT IS THE KERNEL

The kernel is a core part of a Linux operating system, which manages the system resources, and provides interface between hardware and software applications. The Red Hat kernel is a custom-built kernel based on the upstream Linux mainline kernel that Red Hat engineers further develop and harden with a focus on stability and compatibility with the latest technologies and hardware.

Before Red Hat releases a new kernel version, the kernel needs to pass a set of rigorous quality assurance tests.

The Red Hat kernels are packaged in the RPM format so that they are easy to upgrade and verify by the yum package manager.

WARNING

Kernels that have not been compiled by Red Hat are not supported by Red Hat.

45.2. WHAT IS YUM

This section refers to description of the yum package manager.

Additional resources

- Configuring basic system settings in RHEL

45.3. UPDATING THE KERNEL

The following procedure describes how to update the kernel using the yum package manager.

Procedure

1. To update the kernel, use the following:

```
# yum update kernel
```

This command updates the kernel along with all dependencies to the latest available version.

2. Reboot your system for the changes to take effect.
NOTE
When upgrading from RHEL 7 to RHEL 8, follow relevant sections of the *Upgrading from RHEL 7 to RHEL 8* document.

45.4. INSTALLING THE KERNEL

The following procedure describes how to install new kernels using the `yum` package manager.

**Procedure**

- To install a specific kernel version, use the following:

```
# yum install kernel-{version}
```

**Additional resources**

- *Red Hat Code Browser*
- *Red Hat Enterprise Linux Release Dates*
CHAPTER 46. CONFIGURING KERNEL COMMAND-LINE PARAMETERS

Kernel command-line parameters are a way to change the behavior of certain aspects of the Red Hat Enterprise Linux kernel at boot time. As a system administrator, you have full control over what options get set at boot. Certain kernel behaviors are only able to be set at boot time, so understanding how to make these changes is a key administration skill.

IMPORTANT

Opting to change the behavior of the system by modifying kernel command-line parameters may have negative effects on your system. You should therefore test changes prior to deploying them in production. For further guidance, contact Red Hat Support.

46.1. UNDERSTANDING KERNEL COMMAND-LINE PARAMETERS

Kernel command-line parameters are used for boot time configuration of:

- The Red Hat Enterprise Linux kernel
- The initial RAM disk
- The user space features

Kernel boot time parameters are often used to overwrite default values and for setting specific hardware settings.

By default, the kernel command-line parameters for systems using the GRUB2 bootloader are defined in the kernelopts variable of the /boot/grub2/grubenv file for each kernel boot entry.

NOTE

For IBM Z, the kernel command-line parameters are stored in the boot entry configuration file because the zipl bootloader does not support environment variables. Thus, the kernelopts environment variable cannot be used.

Additional resources

- kernel-command-line(7), bootparam(7) and dracut.cmdline(7) manual pages
- How to install and boot custom kernels in Red Hat Enterprise Linux 8

46.2. WHAT GRUBBY IS

grubby is a utility for manipulating bootloader-specific configuration files.

You can use grubby also for changing the default boot entry, and for adding/removing arguments from a GRUB2 menu entry.

For more details see the grubby(8) manual page.

46.3. WHAT BOOT ENTRIES ARE
A boot entry is a collection of options which are stored in a configuration file and tied to a particular kernel version. In practice, you have at least as many boot entries as your system has installed kernels. The boot entry configuration file is located in the `/boot/loader/entries/` directory and can look like this:

```
6f9cc9cb7d7845d49698c9537337cedc-4.18.0-5.el8.x86_64.conf
```

The file name above consists of a machine ID stored in the `/etc/machine-id` file, and a kernel version.

The boot entry configuration file contains information about the kernel version, the initial ramdisk image, and the `kernelopts` environment variable, which contains the kernel command-line parameters. The example contents of a boot entry config can be seen below:

```
title Red Hat Enterprise Linux (4.18.0-74.el8.x86_64) 8.0 (Ootpa)
version 4.18.0-74.el8.x86_64
linux /vmlinuz-4.18.0-74.el8.x86_64
initrd /initramfs-4.18.0-74.el8.x86_64.img $tuned_initrd
options $kernelopts $tuned_params
id rhel-20190227183418-4.18.0-74.el8.x86_64
grub_users $grub_users
grub_arg --unrestricted
grub_class kernel
```

The `kernelopts` environment variable is defined in the `/boot/grub2/grubenv` file.

Additional resources

- How to install and boot custom kernels in Red Hat Enterprise Linux 8

### 46.4. CHANGING KERNEL COMMAND-LINE PARAMETERS FOR ALL BOOT ENTRIES

This procedure describes how to change kernel command-line parameters for all boot entries on your system.

**Prerequisites**

- Verify that the `grubby` utility is installed on your system.
- Verify that the `zipl` utility is installed on your IBM Z system.

**IMPORTANT**

Newly installed kernels do not inherit the kernel command-line parameters from your previously configured kernels. However, the kernel command-line parameters are stored in the `/etc/default/grub` file. As a result, you can run the `grub2-mkconfig` command on the newly installed kernel to get the needed parameters propagated to your new kernel.

**Procedure**

- To add a parameter:

  ```
  # grubby --update-kernel=ALL --args="<NEW_PARAMETER>"
  ```
For systems that use the GRUB2 bootloader, the command updates the /boot/grub2/grubenv file by adding a new kernel parameter to the kernelopts variable in that file.

On IBM Z that use the zIPL bootloader, the command adds a new kernel parameter to each /boot/loader/entries/<ENTRY>.conf file.

- On IBM Z, execute the zipl command with no options to update the boot menu.
- To remove a parameter:

  # grubby --update-kernel=ALL --remove-args="<PARAMETER_TO_REMOVE>"

  - On IBM Z, execute the zipl command with no options to update the boot menu.

**Additional resources**

- Understanding kernel command-line parameters
- grubby(8) and zipl(8) manual pages
- grubby tool

**46.5. CHANGING KERNEL COMMAND-LINE PARAMETERS FOR A SINGLE BOOT ENTRY**

This procedure describes how to change kernel command-line parameters for a single boot entry on your system.

**Prerequisites**

- Verify that the grubby and zipl utilities are installed on your system.

**Procedure**

- To add a parameter:

  # grubby --update-kernel=/boot/vmlinuz-$(uname -r) --args="<NEW_PARAMETER>"

  - On IBM Z, execute the zipl command with no options to update the boot menu.

- To remove a parameter use the following:

  # grubby --update-kernel=/boot/vmlinuz-$(uname -r) --remove-args="<PARAMETER_TO_REMOVE>"

  - On IBM Z, execute the zipl command with no options to update the boot menu.

**NOTE**

On systems that use the grub.cfg file, there is, by default, the options parameter for each kernel boot entry, which is set to the kernelopts variable. This variable is defined in the /boot/grub2/grubenv configuration file.
IMPORTANT

On GRUB2 systems:

- If the kernel command-line parameters are modified for all boot entries, the `grubby` utility updates the `kernelopts` variable in the `/boot/grub2/grubenv` file.

- If kernel command-line parameters are modified for a single boot entry, the `kernelopts` variable is expanded, the kernel parameters are modified, and the resulting value is stored in the respective boot entry's `/boot/loader/entries/<RELEVANTKERNEL_BOOT_ENTRY.conf>` file.

On zIPL systems:

- `grubby` modifies and stores the kernel command-line parameters of an individual kernel boot entry in the `/boot/loader/entries/<ENTRY>.conf` file.

Additional resources

- Understanding kernel command-line parameters
- `grubby(8)` and `zipl(8)` manual pages
- `grubby tool`

46.6. CHANGING KERNEL COMMAND-LINE PARAMETERS TEMPORARILY AT BOOT TIME

The following procedure allows you to make temporary changes to a Kernel Menu Entry by changing the kernel parameters only during a single boot process.

Procedure

1. Select the kernel you want to start when the GRUB 2 boot menu appears and press the `e` key to edit the kernel parameters.

2. Find the kernel command line by moving the cursor down. The kernel command line starts with `linux` on 64-Bit IBM Power Series and x86-64 BIOS-based systems, or `linuxefi` on UEFI systems.

3. Move the cursor to the end of the line.

   **NOTE**

   Press `Ctrl+a` to jump to the start of the line and `Ctrl+e` to jump to the end of the line. On some systems, `Home` and `End` keys might also work.

4. Edit the kernel parameters as required. For example, to run the system in emergency mode, add the `emergency` parameter at the end of the `linux` line:

   ```
   linux ($root)/vmlinuz-4.18.0-348.12.2.el8_5.x86_64 root=/dev/mapper/rhel-root ro crashkernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet emergency
   ```
To enable the system messages, remove the `rhgb` and `quiet` parameters.

5. Press `Ctrl+x` to boot with the selected kernel and the modified command line parameters.

**IMPORTANT**

Press `Esc` key to leave command line editing and it will drop all the user made changes.

**NOTE**

This procedure applies only for a single boot and does not persistently make the changes.
CHAPTER 47. CONFIGURING KERNEL PARAMETERS AT RUNTIME

As a system administrator, you can modify many facets of the Red Hat Enterprise Linux kernel’s behavior at runtime. This section describes how to configure kernel parameters at runtime by using the `sysctl` command and by modifying the configuration files in the `/etc/sysctl.d/` and `/proc/sys/` directories.

47.1. WHAT ARE KERNEL PARAMETERS

Kernel parameters are tunable values which you can adjust while the system is running. There is no requirement to reboot or recompile the kernel for changes to take effect.

It is possible to address the kernel parameters through:

- The `sysctl` command
- The virtual file system mounted at the `/proc/sys/` directory
- The configuration files in the `/etc/sysctl.d/` directory

Tunables are divided into classes by the kernel subsystem. Red Hat Enterprise Linux has the following tunable classes:

<table>
<thead>
<tr>
<th>Tunable class</th>
<th>Subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>abi</td>
<td>Execution domains and personalities</td>
</tr>
<tr>
<td>crypto</td>
<td>Cryptographic interfaces</td>
</tr>
<tr>
<td>debug</td>
<td>Kernel debugging interfaces</td>
</tr>
<tr>
<td>dev</td>
<td>Device-specific information</td>
</tr>
<tr>
<td>fs</td>
<td>Global and specific file system tunables</td>
</tr>
<tr>
<td>kernel</td>
<td>Global kernel tunables</td>
</tr>
<tr>
<td>net</td>
<td>Network tunables</td>
</tr>
<tr>
<td>sunrpc</td>
<td>Sun Remote Procedure Call (NFS)</td>
</tr>
<tr>
<td>user</td>
<td>User Namespace limits</td>
</tr>
<tr>
<td>vm</td>
<td>Tuning and management of memory, buffers, and cache</td>
</tr>
</tbody>
</table>
**IMPORTANT**

Configuring kernel parameters on a production system requires careful planning. Unplanned changes may render the kernel unstable, requiring a system reboot. Verify that you are using valid options before changing any kernel values.

Additional resources

- `sysctl(8)`, and `sysctl.d(5)` manual pages

### 47.2. CONFIGURING KERNEL PARAMETERS TEMPORARILY WITH SYSCTL

The following procedure describes how to use the `sysctl` command to temporarily set kernel parameters at runtime. The command is also useful for listing and filtering tunables.

**Prerequisites**

- Root permissions

**Procedure**

1. To list all parameters and their values, use the following:

   ```bash
   # sysctl -a
   ```

   **NOTE**

   The `# sysctl -a` command displays kernel parameters, which can be adjusted at runtime and at boot time.

2. To configure a parameter temporarily, use the command as in the following example:

   ```bash
   # sysctl <TUNABLE_CLASS>.<PARAMETER>=<TARGET_VALUE>
   ```

   The sample command above changes the parameter value while the system is running. The changes take effect immediately, without a need for restart.

   **NOTE**

   The changes return back to default after your system reboots.

**Additional resources**

- `sysctl(8)` manual page
- Configuring kernel parameters permanently with `sysctl`
- Using configuration files in `/etc/sysctl.d/` to adjust kernel parameters

### 47.3. CONFIGURING KERNEL PARAMETERS PERMANENTLY WITH SYSCTL
The following procedure describes how to use the `sysctl` command to permanently set kernel parameters.

**Prerequisites**

- Root permissions

**Procedure**

1. To list all parameters, use the following:

   ```bash
   # sysctl -a
   ```

   The command displays all kernel parameters that can be configured at runtime.

2. To configure a parameter permanently:

   ```bash
   # sysctl -w <TUNABLE_CLASS>.<PARAMETER>=<TARGET_VALUE> >> /etc/sysctl.conf
   ```

   The sample command changes the tunable value and writes it to the `/etc/sysctl.conf` file, which overrides the default values of kernel parameters. The changes take effect immediately and persistently, without a need for restart.

**NOTE**

To permanently modify kernel parameters you can also make manual changes to the configuration files in the `/etc/sysctl.d/` directory.

**Additional resources**

- `sysctl(8)` and `sysctl.conf(5)` manual pages
- [Using configuration files in `/etc/sysctl.d/` to adjust kernel parameters](#)
3. Save the configuration file.

4. Reboot the machine for the changes to take effect.
   - Alternatively, to apply changes without rebooting, execute:

   ```bash
   # sysctl -p /etc/sysctl.d/<some_file.conf>
   ``

   The command enables you to read values from the configuration file, which you created earlier.

---

**47.5. CONFIGURING KERNEL PARAMETERS TEMPORARILY THROUGH /PROC/SYS/**

The following procedure describes how to set kernel parameters temporarily through the files in the virtual file system /proc/sys/ directory.

**Prerequisites**

- Root permissions

**Procedure**

1. Identify a kernel parameter you want to configure:

   ```bash
   # ls -l /proc/sys/<TUNABLE_CLASS>/
   ``

   The writable files returned by the command can be used to configure the kernel. The files with read-only permissions provide feedback on the current settings.

2. Assign a target value to the kernel parameter:

   ```bash
   # echo <TARGET_VALUE> > /proc/sys/<TUNABLE_CLASS>/<PARAMETER>
   ``

   The command makes configuration changes that will disappear once the system is restarted.

3. Optionally, verify the value of the newly set kernel parameter:

   ```bash
   # cat /proc/sys/<TUNABLE_CLASS>/<PARAMETER>
   ``

**Additional resources**

- Configuring kernel parameters permanently with sysctl
- Using configuration files in /etc/sysctl.d/ to adjust kernel parameters
CHAPTER 48. INSTALLING AND CONFIGURING KDUMP

48.1. INSTALLING KDUMP

The `kdump` service is installed and activated by default on the new Red Hat Enterprise Linux installations. The following sections explain what `kdump` is and how to install `kdump` when it is not enabled by default.

48.1.1. What is kdump

`kdump` is a service which provides a crash dumping mechanism. The service enables you to save the contents of the system memory for analysis. `kdump` uses the `kexec` system call to boot into the second kernel (a `capture kernel`) without rebooting; and then captures the contents of the crashed kernel’s memory (a `crash dump` or a `vmcore`) and saves it into a file. The second kernel resides in a reserved part of the system memory.

**IMPORTANT**

A kernel crash dump can be the only information available in the event of a system failure (a critical bug). Therefore, operational `kdump` is important in mission-critical environments. Red Hat advise that system administrators regularly update and test `kexec-tools` in your normal kernel update cycle. This is especially important when new kernel features are implemented.

You can enable `kdump` for all installed kernels on a machine or only for specified kernels. This is useful when there are multiple kernels used on a machine, some of which are stable enough that there is no concern that they could crash.

When `kdump` is installed, a default `/etc/kdump.conf` file is created. The file includes the default minimum `kdump` configuration. You can edit this file to customize the `kdump` configuration, but it is not required.

48.1.2. Installing kdump using Anaconda

The Anaconda installer provides a graphical interface screen for `kdump` configuration during an interactive installation. However, only limited configuration is allowed.

**Procedure**

1. Go to the `Kdump` field.
2. Enable `kdump`.
3. Define how much memory should be reserved for `kdump`. 
48.1.3. Installing kdump on the command line

Some installation options, such as custom Kickstart installations, in some cases do not install or enable kdump by default. If this is your case, follow the procedure below.

Prerequisites

- An active RHEL subscription
- The kexec-tools package
- Fulfilled requirements for kdump configurations and targets. For details, see Supported kdump configurations and targets.

Procedure

1. Check whether kdump is installed on your system:

   $ rpm -q kexec-tools

   Output if the package is installed:

   kexec-tools-2.0.17-11.el8.x86_64

   Output if the package is not installed:

   package kexec-tools is not installed

2. Install kdump and other necessary packages by:

   # yum install kexec-tools

   IMPORTANT

   Starting with kernel-3.10.0-693.el7 the Intel IOMMU driver is supported with kdump. For prior versions, kernel-3.10.0-514[.XYZ].el7 and earlier, it is advised that Intel IOMMU support is disabled, otherwise the capture kernel is likely to become unresponsive.
48.2. CONFIGURING KDUMP ON THE COMMAND LINE

The following sections explain how to plan and build your `kdump` environment.

48.2.1. Estimating the kdump size

When planning and building your `kdump` environment, it is important to know how much space the crash dump file requires.

The `makedumpfile --mem-usage` command estimates how much space the crash dump file requires. It generates a memory usage report. The report helps you determine the dump level and which pages are safe to be excluded.

Procedure

- Execute the following command to generate a memory usage report:

  ```bash
  # makedumpfile --mem-usage /proc/kcore
  ```

<table>
<thead>
<tr>
<th>TYPE</th>
<th>PAGES</th>
<th>EXCLUDABLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZERO</td>
<td>501635</td>
<td>yes</td>
<td>Pages filled with zero</td>
</tr>
<tr>
<td>CACHE</td>
<td>51657</td>
<td>yes</td>
<td>Cache pages</td>
</tr>
<tr>
<td>CACHE_PRIVATE</td>
<td>5442</td>
<td>yes</td>
<td>Cache pages + private</td>
</tr>
<tr>
<td>USER</td>
<td>16301</td>
<td>yes</td>
<td>User process pages</td>
</tr>
<tr>
<td>FREE</td>
<td>77738211</td>
<td>yes</td>
<td>Free pages</td>
</tr>
<tr>
<td>KERN_DATA</td>
<td>1333192</td>
<td>no</td>
<td>Dumpable kernel data</td>
</tr>
</tbody>
</table>

**IMPORTANT**

The `makedumpfile --mem-usage` command reports required memory in pages. This means that you must calculate the size of memory in use against the kernel page size.

48.2.2. Configuring kdump memory usage

The memory for `kdump` is reserved during the system boot. The memory size is configured in the system Grand Unified Bootloader (GRUB) 2 configuration file. The memory size depends on the value of the `crashkernel=` option specified in the configuration file and the size of the system physical memory.

The `crashkernel=` option can be defined in multiple ways. You can either specify the `crashkernel=` value or configure the `auto` option. The `crashkernel=auto` parameter reserves memory automatically, based on the total amount of physical memory in the system. When configured, the kernel will automatically reserve an appropriate amount of required memory for the capture kernel. This helps to prevent Out-of-Memory (OOM) errors.

**NOTE**

The automatic memory allocation for `kdump` varies based on system hardware architecture and available memory size.

If the system has less than the minimum memory threshold for automatic allocation, you can configure the amount of reserved memory manually.
Prerequisites

- Root permissions.
- Fulfilled requirements for kdump configurations and targets. For details, see Supported kdump configurations and targets.

Procedure

1. Edit the /etc/default/grub file.

2. Set the crashkernel= option.
   For example, to reserve 128 MB of memory, use the following:

   ```
   crashkernel=128M
   ```

   Alternatively, you can set the amount of reserved memory to a variable depending on the total amount of installed memory. The syntax for memory reservation into a variable is crashkernel=range1:size1,range2:size2. For example:

   ```
   crashkernel=512M-2G:64M,2G-:128M
   ```

   The above example reserves 64 MB of memory if the total amount of system memory is between 512 MB and 2 GB. If the total amount of memory is more than 2 GB, 128 MB is reserved.

3. Offset the reserved memory.
   Some systems require to reserve memory with a certain fixed offset since crashkernel reservation is very early, and it wants to reserve some area for special usage. If the offset is set, the reserved memory begins there. To offset the reserved memory, use the following syntax:

   ```
   crashkernel=128M@16M
   ```

   In the example above kdump reserves 128 MB of memory starting at 16 MB (physical address 0x01000000). If the offset parameter is set to 0 or omitted entirely, kdump offsets the reserved memory automatically. You can also use this syntax when setting a variable memory reservation. In that case, the offset is always specified last (for example, crashkernel=512M-2G:64M,2G-:128M@16M).

3. Use the following command to update the GRUB2 configuration file:

   ```
   # grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

   **NOTE**

   The alternative way to configure memory for kdump is to append the crashkernel= SOME_VALUE parameter to the kernelopts variable with the grub2-editenv command, which will update all of your boot entries. Or you can use the grubby utility to update one boot entry, more boot entries, or all of your boot entries.

Additional resources

- Memory requirements for kdump
48.2.3. Configuring the kdump target

The crash dump is usually stored as a file in a local file system, written directly to a device. Alternatively, you can set up for the crash dump to be sent over a network using the **NFS** or **SSH** protocols. Only one of these options to preserve a crash dump file can be set at a time. The default behavior is to store it in the `/var/crash/` directory of the local file system.

**Prerequisites**

- **Root** permissions.
- Fulfilled requirements for **kdump** configurations and targets. For details, see [Supported kdump configurations and targets](#).

**Procedure**

- To store the crash dump file in `/var/crash/` directory of the local file system, edit the `/etc/kdump.conf` file and specify the path:

  ```
  path /var/crash
  ```

  The option `path /var/crash` represents the path to the file system in which **kdump** saves the crash dump file. When you specify a dump target in the `/etc/kdump.conf` file, then the `path` is relative to the specified dump target.

  If you do not specify a dump target in the `/etc/kdump.conf` file, then the `path` represents the absolute path from the root directory. Depending on what is mounted in the current system, the dump target and the adjusted dump path are taken automatically.

  **kdump** saves the crash dump file in `/var/crash/var/crash` directory, when the dump target is mounted at `/var/crash` and the option `path` is also set as `/var/crash` in the `/etc/kdump.conf` file. For example, in the following instance, the `ext4` file system is already mounted at `/var/crash` and the `path` are set as `/var/crash`:

  ```
  grep -v ^# etc/kdump.conf | grep -v ^$
  ext4 /dev/mapper/vg00-varcrashvol
  path /var/crash
  core_collector makedumpfile -c --message-level 1 -d 31
  ```

  This results in the `/var/crash/var/crash` path. To solve this problem, use the option `path /` instead of `path /var/crash`

  - To change the local directory in which the crash dump is to be saved, as **root**, edit the `/etc/kdump.conf` configuration file as described below.
1. Remove the hash sign ("#") from the beginning of the \texttt{#path /var/crash} line.
2. Replace the value with the intended directory path. For example:
   \begin{verbatim}
   path /usr/local/cores
   \end{verbatim}

\textbf{IMPORTANT}

In RHEL 8, the directory defined as the kdump target using the \texttt{path} directive must exist when the \texttt{kdump} systemd service is started - otherwise the service fails. This behavior is different from earlier releases of RHEL, where the directory was being created automatically if it did not exist when starting the service.

- To write the file to a different partition, as \texttt{root}, edit the \texttt{/etc/kdump.conf} configuration file as described below.
  1. Remove the hash sign ("#") from the beginning of the \texttt{#ext4} line, depending on your choice.
     \begin{itemize}
     \item device name (the \texttt{#ext4 /dev/vg/lv_kdump} line)
     \item file system label (the \texttt{#ext4 LABEL=/boot} line)
     \item UUID (the \texttt{#ext4 UUID=03138356-5e61-4ab3-b58e-27507ac41937} line)
     \end{itemize}
  2. Change the file system type as well as the device name, label or UUID to the desired values. For example:
     \begin{verbatim}
     ext4 UUID=03138356-5e61-4ab3-b58e-27507ac41937
     \end{verbatim}

\textbf{IMPORTANT}

It is recommended to specify storage devices using a \texttt{LABEL=} or \texttt{UUID=}.
Disk device names such as \texttt{/dev/sda3} are not guaranteed to be consistent across reboot.

- To write the crash dump directly to a device, edit the \texttt{/etc/kdump.conf} configuration file:
  1. Remove the hash sign ("#") from the beginning of the \texttt{#raw /dev/vg/lv_kdump} line.
  2. Replace the value with the intended device name. For example:
     \begin{verbatim}
     raw /dev/sdb1
     \end{verbatim}

- To store the crash dump to a remote machine using the \texttt{NFS} protocol, edit the \texttt{/etc/kdump.conf} configuration file:
  1. Remove the hash sign ("#") from the beginning of the \texttt{#nfs my.server.com:/export/tmp} line.
  2. Replace the value with a valid hostname and directory path. For example:
     \begin{verbatim}
     nfs penguin.example.com:/export/cores
     \end{verbatim}
To store the crash dump to a remote machine using the SSH protocol, edit the /etc/kdump.conf configuration file:

1. Remove the hash sign (“#”) from the beginning of the #ssh user@my.server.com line.
2. Replace the value with a valid username and hostname.
3. Include your SSH key in the configuration.
   - Remove the hash sign from the beginning of the #sshkey /root/.ssh/kdump_id_rsa line.
   - Change the value to the location of a key valid on the server you are trying to dump to. For example:

   ```
   ssh john@penguin.example.com
   sshkey /root/.ssh/mykey
   ```

48.2.4. Configuring the kdump core collector

The kdump service uses a core_collector program to capture the crash dump image. In RHEL, the makedumpfile utility is the default core collector. It helps shrink the dump file by:

- Compressing the size of a crash dump file and copying only necessary pages using various dump levels
- Excluding unnecessary crash dump pages
- Filtering the page types to be included in the crash dump.

Syntax

```bash
core_collector makedumpfile -l --message-level 1 -d 31
```

Options

- `-c`, `-l` or `-p`: specify compress dump file format by each page using either, **zlib** for `-c` option, **lzo** for `-l` option or **snappy** for `-p` option.
- `-d (dump_level)`: excludes pages so that they are not copied to the dump file.
- `--message-level`: specify the message types. You can restrict outputs printed by specifying `message_level` with this option. For example, specifying 7 as `message_level` prints common messages and error messages. The maximum value of `message_level` is 31

Prerequisites

- **Root** permissions
- Fulfilled requirements for kdump configurations and targets. For details, see Supported kdump configurations and targets.

Procedure
1. As root, edit the /etc/kdump.conf configuration file and remove the hash sign ("#") from the beginning of the #core_collector makedumpfile -l --message-level 1 -d 31.

2. To enable crash dump file compression, execute:

```
core_collector makedumpfile -l --message-level 1 -d 31
```

The -l option specifies the dump compressed file format. The -d option specifies dump level as 31. The --message-level option specifies message level as 1.

Also, consider following examples with the -c and -p options:

- To compress a crash dump file using -c:

```
core_collector makedumpfile -c -d 31 --message-level 1
```

- To compress a crash dump file using -p:

```
core_collector makedumpfile -p -d 31 --message-level 1
```

Additional resources
- makedumpfile(8) manual page
- The kdump configuration file

### 48.2.5. Configuring the kdump default failure responses

By default, when kdump fails to create a crash dump file at the configured target location, the system reboots and the dump is lost in the process. To change this behavior, follow the procedure below.

**Prerequisites**
- Root permissions.
- Fulfilled requirements for kdump configurations and targets. For details, see Supported kdump configurations and targets.

**Procedure**

1. As root, remove the hash sign ("#") from the beginning of the #failure_action line in the /etc/kdump.conf configuration file.

```
failure_action poweroff
```

**Additional resources**
- Configuring the kdump target

### 48.2.6. Testing the kdump configuration
You can test that the crash dump process works and is valid before the machine enters production.

**WARNING**

The commands below cause the kernel to crash. Use caution when following these steps, and never carelessly use them on active production system.

**Procedure**

1. Reboot the system with `kdump` enabled.
2. Make sure that `kdump` is running:
   ```bash
   ~]# systemctl is-active kdump
   active
   ``
3. Force the Linux kernel to crash:
   ```bash
   echo 1 > /proc/sys/kernel/sysrq
   echo c > /proc/sysrq-trigger
   ```

**WARNING**

The command above crashes the kernel, and a reboot is required.

Once booted again, the `address-YYYY-MM-DD-HH:MM:SS/vmcore` file is created at the location you have specified in the `/etc/kdump.conf` file (by default to `/var/crash/`).

**NOTE**

This action confirms the validity of the configuration. Also it is possible to use this action to record how long it takes for a crash dump to complete with a representative work-load.

**Additional resources**

- Configuring the kdump target

### 48.3. ENABLING KDUMP

This section provides the information and procedures necessary to enable and start the `kdump` service for all installed kernels or for a specific kernel.
48.4. CONFIGURING KDUMP IN THE WEB CONSOLE

Setup and test the **kdump** configuration in the RHEL 8 web console.

The web console is part of a default installation of RHEL 8 and enables or disables the **kdump** service at boot time. Further, the web console conveniently enables you to configure the reserved memory for **kdump**; or to select the **vmcore** saving location in an uncompressed or compressed format.

48.4.1. Additional resources

- [Getting started using the RHEL web console](#)

48.4.2. Configuring kdump memory usage and target location in web console

The procedure below shows you how to use the **Kernel Dump** tab in the RHEL web console interface to configure the amount of memory that is reserved for the **kdump** kernel. The procedure also describes how to specify the target location of the **vmcore** dump file and how to test your configuration.

**Procedure**

1. Open the **Kernel Dump** tab and start the **kdump** service.

2. Configure the **kdump** memory usage using the command line.

3. Click the link next to the **Crash dump location** option.

4. Select the **Local Filesystem** option from the drop-down and specify the directory you want to save the dump in.
Alternatively, select the **Remote over SSH** option from the drop-down to send the vmcore to a remote machine using the SSH protocol. Fill the **Server**, **ssh key**, and **Directory** fields with the remote machine address, ssh key location, and a target directory.

- Another choice is to select the **Remote over NFS** option from the drop-down and fill the **Mount** field to send the vmcore to a remote machine using the NFS protocol.

**NOTE**

Tick the **Compression** check box to reduce the size of the vmcore file.

5. Test your configuration by crashing the kernel.

**WARNING**

This step disrupts execution of the kernel and results in a system crash and loss of data.
48.5. SUPPORTED KDUMP CONFIGURATIONS AND TARGETS

48.5.1. Memory requirements for kdump

In order for kdump to be able to capture a kernel crash dump and save it for further analysis, a part of the system memory has to be permanently reserved for the capture kernel. When reserved, this part of the system memory is not available to the main kernel.

The memory requirements vary based on certain system parameters. One of the major factors is the system’s hardware architecture. To find out the exact machine architecture (such as Intel 64 and AMD64, also known as x86_64) and print it to standard output, use the following command:

```
$ uname -m
```

Table 48.1 lists the minimum memory requirements to automatically reserve a memory size for kdump on the latest available versions. The size changes according to the system’s architecture and total available physical memory.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Available Memory</th>
<th>Minimum Reserved Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD64 and Intel 64 (%x86_64)</td>
<td>1 GB to 4 GB</td>
<td>160 MB of RAM.</td>
</tr>
<tr>
<td></td>
<td>4 GB to 64 GB</td>
<td>192 MB of RAM.</td>
</tr>
<tr>
<td></td>
<td>64 GB to 1 TB</td>
<td>256 MB of RAM.</td>
</tr>
<tr>
<td></td>
<td>1 TB and more</td>
<td>512 MB of RAM.</td>
</tr>
<tr>
<td>64-bit ARM architecture (%arm64)</td>
<td>2 GB and more</td>
<td>448 MB of RAM.</td>
</tr>
<tr>
<td>IBM Power Systems (%ppc64le)</td>
<td>2 GB to 4 GB</td>
<td>384 MB of RAM.</td>
</tr>
<tr>
<td></td>
<td>4 GB to 16 GB</td>
<td>512 MB of RAM.</td>
</tr>
<tr>
<td></td>
<td>16 GB to 64 GB</td>
<td>1 GB of RAM.</td>
</tr>
<tr>
<td></td>
<td>64 GB to 128 GB</td>
<td>2 GB of RAM.</td>
</tr>
<tr>
<td></td>
<td>128 GB and more</td>
<td>4 GB of RAM.</td>
</tr>
<tr>
<td>IBM Z (%s390x)</td>
<td>1 GB to 4 GB</td>
<td>160 MB of RAM.</td>
</tr>
</tbody>
</table>
On many systems, `kdump` is able to estimate the amount of required memory and reserve it automatically. This behavior is enabled by default, but only works on systems that have more than a certain amount of total available memory, which varies based on the system architecture.

**IMPORTANT**

The automatic configuration of reserved memory based on the total amount of memory in the system is a best effort estimation. The actual required memory may vary due to other factors such as I/O devices. Using not enough of memory might cause that a debug kernel is not able to boot as a capture kernel in case of a kernel panic. To avoid this problem, sufficiently increase the crash kernel memory.

Additional resources

- How has the crashkernel parameter changed between RHEL8 minor releases?
- Technology capabilities and limits tables
- Configuring kdump memory usage
- Configuring kdump memory usage and target location in web console
- Minimum threshold for automatic memory reservation

### 48.5.2. Minimum threshold for automatic memory reservation

On some systems, it is possible to allocate memory for `kdump` automatically, either by using the `crashkernel=auto` parameter in the boot loader configuration file, or by enabling this option in the graphical configuration utility. For this automatic reservation to work, however, a certain amount of total memory needs to be available in the system. The amount differs based on the system's architecture.

The table below lists the threshold values for automatic memory allocation. If the system has memory less than the specified threshold value, you must configure the memory manually.

#### Table 48.2. Minimum Amount of Memory Required for Automatic Memory Reservation

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Required Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD64 and Intel 64 (x86_64)</td>
<td>2 GB</td>
</tr>
<tr>
<td>IBM Power Systems (ppc64le)</td>
<td>2 GB</td>
</tr>
<tr>
<td>IBM Z (s390x)</td>
<td>4 GB</td>
</tr>
</tbody>
</table>
### Additional resources

- Configuring kdump memory usage
- Configuring kdump memory usage and target location in web console

#### 48.5.3. Supported kdump targets

When a kernel crash is captured, the vmcore dump file can be either written directly to a device, stored as a file on a local file system, or sent over a network. The table below contains a complete list of dump targets that are currently supported or explicitly unsupported by kdump.

<table>
<thead>
<tr>
<th>Type</th>
<th>Supported Targets</th>
<th>Unsupported Targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw device</td>
<td>All locally attached raw disks and partitions.</td>
<td>Any local file system not explicitly listed as supported in this table, including the auto type (automatic file system detection).</td>
</tr>
<tr>
<td>Local file system</td>
<td>ext2, ext3, ext4, and xfs file systems on directly attached disk drives, hardware RAID logical drives, LVM devices, and mdraid arrays.</td>
<td></td>
</tr>
<tr>
<td>Remote directories accessed using the iSCSI protocol over both hardware and software initiators.</td>
<td>Remote directories accessed using the iSCSI protocol on be2iscsi hardware.</td>
<td>Multipath-based storages.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Remote directories accessed using the SMB or CIFS protocol.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Remote directories accessed using the FCoE (Fibre Channel over Ethernet) protocol.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Remote directories accessed using wireless network interfaces.</td>
</tr>
</tbody>
</table>
IMPORTANT

Utilizing firmware assisted dump (fadump) to capture a vmcore and store it to a remote machine using SSH or NFS protocol causes renaming of the network interface to kdump-<interface-name>. The renaming happens if the <interface-name> is generic, for example *eth#, net#, and so on. This problem occurs because the vmcore capture scripts in the initial RAM disk (initrd) add the kdump- prefix to the network interface name to secure persistent naming. Since the same initrd is used also for a regular boot, the interface name is changed for the production kernel too.

Additional resources

- Configuring the kdump target
- Configuring kdump memory usage and target location in web console

48.5.4. Supported kdump filtering levels

To reduce the size of the dump file, kdump uses the makedumpfile core collector to compress the data and optionally to omit unwanted information. The table below contains a complete list of filtering levels that are currently supported by the makedumpfile utility.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Zero pages</td>
</tr>
<tr>
<td>2</td>
<td>Cache pages</td>
</tr>
<tr>
<td>4</td>
<td>Cache private</td>
</tr>
<tr>
<td>8</td>
<td>User pages</td>
</tr>
<tr>
<td>16</td>
<td>Free pages</td>
</tr>
</tbody>
</table>

**NOTE**

The makedumpfile command supports removal of transparent huge pages and hugetlbfs pages. Consider both these types of hugepages User Pages and remove them using the -8 level.

Additional resources

- Configuring the core collector

48.5.5. Supported default failure responses

By default, when kdump fails to create a core dump, the operating system reboots. You can, however, configure kdump to perform a different operation in case it fails to save the core dump to the primary target. The table below lists all default actions that are currently supported.
## Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dump_to_rootfs</code></td>
<td>Attempt to save the core dump to the root file system. This option is especially useful in combination with a network target: if the network target is unreachable, this option configures kdump to save the core dump locally. The system is rebooted afterwards.</td>
</tr>
<tr>
<td><code>reboot</code></td>
<td>Reboot the system, losing the core dump in the process.</td>
</tr>
<tr>
<td><code>halt</code></td>
<td>Halt the system, losing the core dump in the process.</td>
</tr>
<tr>
<td><code>poweroff</code></td>
<td>Power off the system, losing the core dump in the process.</td>
</tr>
<tr>
<td><code>shell</code></td>
<td>Run a shell session from within the initramfs, allowing the user to record the core dump manually.</td>
</tr>
<tr>
<td><code>final_action</code></td>
<td>Enable additional operations such as <code>reboot</code>, <code>halt</code>, and <code>poweroff</code> actions after a successful <code>kdump</code> or when a shell or <code>dump_to_rootfs</code> failure action completes. The default <code>final_action</code> option is <code>reboot</code>.</td>
</tr>
</tbody>
</table>

### Additional resources

- Configuring the kdump default failure responses

#### 48.5.6. Using `final_action` parameter

The `final_action` parameter enables you to use certain additional operations such as `reboot`, `halt`, and `poweroff` actions after a successful `kdump` or when an invoked `failure_response` mechanism using `shell` or `dump_to_rootfs` completes. If the `final_action` option is not specified, it defaults to `reboot`.

### Procedure

1. Edit the `/etc/kdump.conf` file and add the `final_action` parameter.
   ```
   final_action <reboot | halt | poweroff>
   ```
2. Restart the `kdump` service:
   ```
   kdumpctl restart
   ```

#### 48.6. TESTING THE KDUMP CONFIGURATION

You can test that the crash dump process works and is valid before the machine enters production.
WARNING
The commands below cause the kernel to crash. Use caution when following these steps, and never carelessly use them on active production system.

Procedure

1. Reboot the system with **kdump** enabled.

2. Make sure that **kdump** is running:

   ```
   ~]# systemctl is-active kdump
   active
   ```

3. Force the Linux kernel to crash:

   ```
   echo 1 > /proc/sys/kernel/sysrq
   echo c > /proc/sysrq-trigger
   ```

   **WARNING**
   The command above crashes the kernel, and a reboot is required.

Once booted again, the **address-YYYY-MM-DD-HH:MM:SS/vmcore** file is created at the location you have specified in the **/etc/kdump.conf** file (by default to **/var/crash/**).

**NOTE**
This action confirms the validity of the configuration. Also it is possible to use this action to record how long it takes for a crash dump to complete with a representative work-load.

Additional resources

- Configuring the kdump target

**48.7. USING KEXEC TO BOOT INTO A DIFFERENT KERNEL**

The **kexec** system call enables loading and booting into another kernel from the currently running kernel, thus performing a function of a boot loader from within the kernel.

The **kexec** utility loads the kernel and the **initramfs** image for the **kexec** system call to boot into another kernel.
The following procedure describes how to manually invoke the *kexec* system call when using the *kexec* utility to reboot into another kernel.

**Procedure**

1. Execute the *kexec* utility:

```sh
# kexec -l /boot/vmlinuz-3.10.0-1040.el7.x86_64 --initrd=/boot/initramfs-3.10.0-1040.el7.x86_64.img --reuse-cmdline
```

The command manually loads the kernel and the initramfs image for the *kexec* system call.

2. Reboot the system:

```sh
# reboot
```

The command detects the kernel, shuts down all services and then calls the *kexec* system call to reboot into the kernel you provided in the previous step.

---

**WARNING**

When you use the *kexec -e* command to reboot your machine into a different kernel, the system does not go through the standard shutdown sequence before starting the next kernel. This can cause data loss or an unresponsive system.

---

### 48.8. PREVENTING KERNEL DRIVERS FROM LOADING FOR KDUMP

This section explains how to prevent the capture kernel from loading certain kernel drivers using the `/etc/sysconfig/kdump` configuration file. You can prevent the *kdump* initramfs from loading the specified kernel module. To achieve this, you need to put the `KDUMP_COMMANDLINE_APPEND=` variable in the `/etc/sysconfig/kdump` file. This helps to prevent the out-of-memory (oom) killer or other crash kernel failures.

You can append the `KDUMP_COMMANDLINE_APPEND=` variable using one of the following configuration options:

- `rd.driver.blacklist=<modules>`
- `modprobe.blacklist=<modules>`

**Procedure**

1. Select a kernel module that you intend to block from loading.

```sh
$ lsmod
Module                  Size  Used by
fuse                  126976  3
xt_CHECKSUM            16384  1
```

---

792
The `lsmod` command displays a list of modules that are loaded to the currently running kernel.

2. Update the `KDUMP_COMMANDLINE_APPEND` variable in the `/etc/sysconfig/kdump` file.

```
KDUMP_COMMANDLINE_APPEND="rd.driver.blacklist=hv_vmbus,hv_storvsc,hv_utils,hv_netvsc,hid-hyperv"
```

Also, consider the following example using the `modprobe.blacklist=<modules>` configuration option.

```
KDUMP_COMMANDLINE_APPEND="modprobe.blacklist=emcp modprobe.blacklist=bnx2fc modprobe.blacklist=libfcoe modprobe.blacklist=fcce"
```

3. Restart the kdump service.

```
# systemctl restart kdump
```

Additional resources

- `dracut.cmdline` manual page

### 48.9. RUNNING KDUMP ON SYSTEMS WITH ENCRYPTED DISK

When you run a LUKS encrypted partition, systems require certain amount of available memory. If the system has less than the required amount of available memory, the `cryptsetup` utility fails to mount the partition. As a result, capturing the `vmcore` file to an encrypted target location fails in the second kernel (capture kernel).

The `kdumpctl estimate` command helps you estimate the amount of memory you need for kdump. `kdumpctl estimate` prints the recommended `crashkernel` value, which is the most suitable memory size required for kdump.

The recommended `crashkernel` value is calculated based on the current kernel size, kernel module, initramfs, and the LUKS encrypted target memory requirement.

In case you are using the custom `crashkernel=` option, `kdumpctl estimate` prints the LUKS required `size` value. The value is the memory size required for LUKS encrypted target.

**Procedure**

1. Print the estimate `crashkernel=` value:

```
# kdumpctl estimate
```

Encrypted kdump target requires extra memory, assuming using the keyslot with minimum memory requirement

- Reserved crashkernel: 256M
- Recommended crashkernel: 652M

- Kernel image size: 47M
Kernel modules size: 8M  
Initramfs size: 20M  
Runtime reservation: 64M  
LUKS required size: 512M  
Large modules: <none>  
WARNING: Current crashkernel size is lower than recommended size 652M.

2. Configure the amount of required memory by increasing crashkernel= to the desired value.

3. Reboot the system.

NOTE
If the kdump service still fails to save the dump file to the encrypted target, increase the crashkernel= value as required.

48.10. FIRMWARE ASSISTED DUMP MECHANISMS

Firmware assisted dump (fadump) is a dump capturing mechanism, provided as an alternative to the kdump mechanism on IBM POWER systems. The kexec and kdump mechanisms are useful for capturing core dumps on AMD64 and Intel 64 systems. However, some hardware such as mini systems and mainframe computers, leverage the onboard firmware to isolate regions of memory and prevent any accidental overwriting of data that is important to the crash analysis. This section covers fadump mechanisms and how they integrate with RHEL. The fadump utility is optimized for these expanded dumping features on IBM POWER systems.

48.10.1. Firmware assisted dump on IBM PowerPC hardware

The fadump utility captures the vmcore file from a fully-reset system with PCI and I/O devices. This mechanism uses firmware to preserve memory regions during a crash and then reuses the kdump userspace scripts to save the vmcore file. The memory regions consist of all system memory contents, except the boot memory, system registers, and hardware Page Table Entries (PTEs).

The fadump mechanism offers improved reliability over the traditional dump type, by rebooting the partition and using a new kernel to dump the data from the previous kernel crash. The fadump requires an IBM POWER6 processor-based or later version hardware platform.

For further details about the fadump mechanism, including PowerPC specific methods of resetting hardware, see the /usr/share/doc/kexec-tools/fadump-howto.txt file.

NOTE
The area of memory that is not preserved, known as boot memory, is the amount of RAM required to successfully boot the kernel after a crash event. By default, the boot memory size is 256MB or 5% of total system RAM, whichever is larger.

Unlike kexec-initiated event, the fadump mechanism uses the production kernel to recover a crash dump. When booting after a crash, PowerPC hardware makes the device node /proc/device-tree/rtas/ibm.kernel-dump available to the proc filesystem (procfs). The fadump-aware kdump scripts, check for the stored vmcore, and then complete the system reboot cleanly.

48.10.2. Enabling firmware assisted dump mechanism
The crash dumping capabilities of IBM POWER systems can be enhanced by enabling the firmware assisted dump (fadump) mechanism.

**Procedure**

1. Install and configure **kdump**.

2. Add `fadump=on` to the `GRUB_CMDLINE_LINUX` line in `/etc/default/grub` file:

   ```
   GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root rhgb quiet fadump=on"
   ```

3. (Optional) If you want to specify reserved boot memory instead of using the defaults, configure `crashkernel=xxM` to `GRUB_CMDLINE_LINUX` in `/etc/default/grub`, where `xx` is the amount of the memory required in megabytes:

   ```
   GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap crashkernel=xxM rd.lvm.lv=rhel/root rhgb quiet fadump=on"
   ```

   **IMPORTANT**

   Red Hat recommends to test all boot configuration options before you execute them. If you observe Out of Memory (OOM) errors when booting from the crash kernel, increase the value specified in `crashkernel=` argument until the crash kernel can boot cleanly. Some trial and error may be required in this case.

**48.10.3. Firmware assisted dump mechanisms on IBM Z hardware**

IBM Z systems support the following firmware assisted dump mechanisms:

- **Stand-alone dump (sadump)**
- **VMDUMP**

The **kdump** infrastructure is supported and utilized on IBM Z systems. However, using one of the firmware assisted dump (fadump) methods for IBM Z can provide various benefits:

- The **sadump** mechanism is initiated and controlled from the system console, and is stored on an IPL bootable device.

- The **VMDUMP** mechanism is similar to **sadump**. This tool is also initiated from the system console, but retrieves the resulting dump from hardware and copies it to the system for analysis.

- These methods (similarly to other hardware based dump mechanisms) have the ability to capture the state of a machine in the early boot phase, before the **kdump** service starts.

- Although **VMDUMP** contains a mechanism to receive the dump file into a Red Hat Enterprise Linux system, the configuration and control of **VMDUMP** is managed from the IBM Z Hardware console.

IBM discusses **sadump** in detail in the **Stand-alone dump program** article and **VMDUMP** in **Creating dumps on z/VM with VMDUMP** article.

IBM also has a documentation set for using the dump tools on Red Hat Enterprise Linux 7 in the **Using the Dump Tools on Red Hat Enterprise Linux 7.4** article.
Additional resources

- Stand-alone dump program
- Creating dumps on z/VM with VMDUMP
- Using the Dump Tools on Red Hat Enterprise Linux 7.4

48.10.4. Using sadump on Fujitsu PRIMEQUEST systems

The Fujitsu sadump mechanism is designed to provide a fallback dump capture in an event when kdump is unable to complete successfully. The sadump mechanism is invoked manually from the system Management Board (MMB) interface. Using MMB, configure kdump like for an Intel 64 or AMD 64 server and then perform the following additional steps to enable sadump.

Procedure

1. Add or edit the following lines in the /etc/sysctl.conf file to ensure that kdump starts as expected for sadump:

   ```
   kernel.panic=0
   kernel.unknown_nmi_panic=1
   ```

   **WARNING**

   In particular, ensure that after kdump, the system does not reboot. If the system reboots after kdump has fails to save the vmcore file, then it is not possible to invoke the sadump.

2. Set the failure_action parameter in /etc/kdump.conf appropriately as halt or shell.

   ```
   failure_action shell
   ```

Additional resources


48.11. ANALYZING A CORE DUMP

To determine the cause of the system crash, you can use the crash utility, which provides an interactive prompt very similar to the GNU Debugger (GDB). This utility allows you to interactively analyze a core dump created by kdump, netdump, diskdump or xendump as well as a running Linux system. Alternatively, you have the option to use Kernel Oops Analyzer or the Kdump Helper tool.

48.11.1. Installing the crash utility

The following procedure describes how to install the crash analyzing tool.
Procedure

1. Enable the relevant repositories:

   # subscription-manager repos --enable baseos repository
   # subscription-manager repos --enable appstream repository
   # subscription-manager repos --enable rhel-8-for-x86_64-baseos-debug-rpms

2. Install the crash package:

   # yum install crash

3. Install the kernel-debuginfo package:

   # yum install kernel-debuginfo

   The package corresponds to your running kernel and provides the data necessary for the dump analysis.

Additional resources

- Configuring basic system settings

48.11.2. Running and exiting the crash utility

The following procedure describes how to start the crash utility for analyzing the cause of the system crash.

Prerequisites

- Identify the currently running kernel (for example 4.18.0-5.el8.x86_64).

Procedure

1. To start the crash utility, two necessary parameters need to be passed to the command:

   - The debug-info (a decompressed vmlinuz image), for example /usr/lib/debug/lib/modules/4.18.0-5.el8.x86_64/vmlinux provided through a specific kernel-debuginfo package.

   - The actual vmcore file, for example /var/crash/127.0.0.1-2018-10-06-14:05:33/vmcore

   The resulting crash command then looks like this:

     # crash /usr/lib/debug/lib/modules/4.18.0-5.el8.x86_64/vmlinux /var/crash/127.0.0.1-2018-10-06-14:05:33/vmcore

   Use the same <kernel> version that was captured by kdump.

     Example 48.1. Running the crash utility
The following example shows analyzing a core dump created on October 6 2018 at 14:05 PM, using the 4.18.0-5.el8.x86_64 kernel.

... WARNING: kernel relocated [202MB]: patching 90160 gdb minimal_symbol values
KERNEL: /usr/lib/debug/lib/modules/4.18.0-5.el8.x86_64/vmlinux
DUMPFILE: /var/crash/127.0.0.1-2018-10-06-14:05:33/vmcore [PARTIAL DUMP]
CPUS: 2
DATE: Sat Oct 6 14:05:16 2018
UPTIME: 01:03:57
LOAD AVERAGE: 0.00, 0.00, 0.00
TASKS: 586
NODENAME: localhost.localdomain
RELEASE: 4.18.0-5.el8.x86_64
VERSION: #1 SMP Wed Aug 29 11:51:55 UTC 2018
MACHINE: x86_64 (2904 Mhz)
MEMORY: 2.9 GB
PANIC: "sysrq: SysRq : Trigger a crash"
PID: 10635
COMMAND: "bash"
TASK: ffff8d6c84271800 [THREAD_INFO: ffff8d6c84271800]
CPU: 1
STATE: TASK_RUNNING (SYSRQ)

```
crash>
```

2. To exit the interactive prompt and terminate crash, type exit or q.

Example 48.2. Exiting the crash utility
```
crash> exit
~]
```

NOTE

The crash command can also be used as a powerful tool for debugging a live system. However use it with caution so as not to break your system.

Additional resources
- A Guide to Unexpected System Restarts

48.11.3. Displaying various indicators in the crash utility

The following procedures describe how to use the crash utility and display various indicators, such as a kernel message buffer, a backtrace, a process status, virtual memory information and open files.

Displaying the message buffer
- To display the kernel message buffer, type the log command at the interactive prompt as displayed in the example below:
crash> log
... several lines omitted ...
EIP: 0060::<c068124f> EFLAGS: 00010096 CPU: 2
EIP is at sysrq_handle_crash+0xf/0x20
EAX: 00000063 EBX: 00000063 ECX: c09e1c8c EDX: 00000000
ESI: c0a09ca0 EDI: 00000286 EBP: 00000000 ESP: ef4dbf24
DS: 007b ES: 007b FS: 00d8 GS: 00e0 SS: 0068
Process bash (pid: 5591, ti=ef4da000 task=f196d560 task.ti=ef4da000)
Stack:
c068146b c0960891 c0968653 00000003 00000000 00000002 efade5c0 c06814d0
<0> fffffffb c068150f b7776000 f2600c40 c0569ec4 ef4dbf9c 000000002 b7776000
<0> efade5c0 000000002 b7776000 c0569e60 c051de50 ef4dbf9c f196d560 ef4dbf64
Call Trace:
[-c068146b>] ? __handle_sysrq+0xfb/0x160
[-c06814d0>] ? write_sysrq_trigger+0x0/0x50
[-c068150f>] ? write_sysrq_trigger+0x3f/0x50
[-c0569ec4>] ? proc_reg_write+0x64/0xa0
[-c0569e60>] ? proc_reg_write+0x0/0xa0
[-c051de50>] ? vfs_write+0x50/0xa0
[-c051e8d1>] ? sys_write+0x41/0x70
[-c0409adc>] ? syscall_call+0x7/0xb
Code: a0 c0 01 0f b6 41 03 19 d2 f7 d2 83 e2 03 83 e0 cf c1 e2 04 09 d0 88 41 03 f3 c3 90 c7 05
c8 1b 9e c0 01 00 00 00 0f ae f8 89 f6 <c6> 05 00 00 00 00 01 c3 89 f6 8d bc 27 00 00 00 00 8d 50
d0 83
EIP: [-c068124f>] sysrq_handle_crash+0xf/0x20 SS:ESP 0068:ef4dbf24
CR2: 0000000000000000

Type help log for more information on the command usage.

NOTE

The kernel message buffer includes the most essential information about the system crash and, as such, it is always dumped first in to the vmcore-dmesg.txt file. This is useful when an attempt to get the full vmcore file failed, for example because of lack of space on the target location. By default, vmcore-dmesg.txt is located in the /var/crash/ directory.

Displaying a backtrace

- To display the kernel stack trace, use the bt command.
Type `bt <pid>` to display the backtrace of a specific process or type `help bt` for more information on `bt` usage.

Displaying a process status

- To display the status of processes in the system, use the `ps` command.

```
crash> ps
    PID   PPID  CPU TASK ST  %MEM VSZ RSS COMM
  > 0    0    0  c09dc560 RU  0.0    0    0  [swapper]
  > 0    0    1  f7072030 RU  0.0    0    0  [swapper]
     0    0    2  f1963a90 RU  0.0    0    0  [swapper]
  > 0    0    3  f70ac560 RU  0.0    0    0  [swapper]
     1    0    1  f705ba90 IN  0.0  2828   1424  init
... several lines omitted ...
  5566   1   1  f2592560 IN  0.0  12876   784  auditd
  5567   1   2  ef427560 IN  0.0  12876   784  auditd
  5587  5132   0  f196d030 IN  0.0  11064   3184  sshd
 >  5591 5587   2  f196d560 RU  0.0   5084   1648  bash
```

Use `ps <pid>` to display the status of a single specific process. Use `help ps` for more information on `ps` usage.

Displaying virtual memory information

- To display basic virtual memory information, type the `vm` command at the interactive prompt.

```
crash> vm
    PID: 5591  TASK: f196d560  CPU: 2  COMMAND: "bash"
  MM   PGD   RSS TOTAL_VM
f19b5900 ef9c6000  1648k    5084k
  VMA   START   END   FLAGS  FILE
f1bb0310    242000    260000 8000875  /lib/ld-2.12.so
f26af0b8    260000    261000 8100871  /lib/ld-2.12.so
efbc275c    261000    262000 8100873  /lib/ld-2.12.so
efbc23d8    268000    3ed000 8000075  /lib/libc-2.12.so
efbc23d8    3ed000    3ee000 8100071  /lib/libc-2.12.so
efbc23d8    3ee000    3f0000 8100073  /lib/libc-2.12.so
efbc23d8    3f0000    3f1000 8100073  /lib/libc-2.12.so
efbc23d8    3f1000    3f4000 100073
efbc28ec    3f4000    3f9000 8000075  /lib/libdl-2.12.so
efbc28ec    3f9000    3fa000 8100071  /lib/libdl-2.12.so
efbc28ec    3fa000    3fb000 8100073  /lib/libdl-2.12.so
f26af888    7e6000    7fc000 8000075  /lib/libtinfo.so.5.7
f26af888    7fc000    7ff000 8100073  /lib/libtinfo.so.5.7
```
Use `vm <pid>` to display information on a single specific process, or use `help vm` for more information on `vm` usage.

**Displaying open files**

- To display information about open files, use the `files` command.

```plaintext
crash> files
PID: 5591 TASK: f196d560 CPU: 2 COMMAND: "bash"
ROOT: / CWD: /root
FD FILE DENTRY INODE TYPE PATH
  0 f734f640 eedc2c6c eecd6048 CHR /pts/0
  1 efade5c0 eee14090 f00431d4 REG /proc/sysrq-trigger
  2 f734f640 eedc2c6c eecd6048 CHR /pts/0
 10 f734f640 eedc2c6c eecd6048 CHR /pts/0
255 f734f640 eedc2c6c eecd6048 CHR /pts/0
```

Use `files <pid>` to display files opened by only one selected process, or use `help files` for more information on `files` usage.

### 48.11.4. Using KernelOops Analyzer

The Kernel Oops Analyzer tool analyzes the crash dump by comparing the oops messages with known issues in the knowledge base.

**Prerequisites**

- Secure an oops message to feed the KernelOops Analyzer.

**Procedure**

1. Access the KernelOops Analyzer tool.

2. To diagnose a kernel crash issue, upload a kernel oops log generated in `vmcore`.
   - Alternatively you can also diagnose a kernel crash issue by providing a text message or a `vmcore-dmesg.txt` as an input.
3. Click **DETECT** to compare the oops message based on information from the **makedumpfile** against known solutions.

Additional resources

- The Kernel Oops Analyzer article
- A Guide to Unexpected System Restarts

### 48.11.5. The Kdump Helper tool

The Kdump Helper tool helps to set up the **kdump** using the provided information. Kdump Helper generates a configuration script based on your preferences. Initiating and running the script on your server sets up the **kdump** service.

Additional resources

- Kdump Helper

### 48.12. USING EARLY KDUMP TO CAPTURE BOOT TIME CRASHES

As a system administrator, you can utilize the **early kdump** support of the **kdump** service to capture a vmcore file of the crashing kernel during the early stages of the booting process. This section describes what **early kdump** is, how to configure it, and how to check the status of this mechanism.

#### 48.12.1. What is early kdump

Kernel crashes during the booting phase occur when the **kdump** service is not yet started, and cannot facilitate capturing and saving the contents of the crashed kernel’s memory. Therefore, the vital information for troubleshooting is lost.

To address this problem, RHEL 8 introduced the **early kdump** feature as a part of the **kdump** service.

#### 48.12.2. Enabling early kdump

This section describes how to enable the **early kdump** feature to eliminate the risk of losing information about the early boot kernel crashes.

**Prerequisites**

- An active RHEL subscription.
A repository containing the kexec-tools package for your system CPU architecture

Fulfilled kdump configuration and targets requirements.

Procedure

1. Verify that the kdump service is enabled and active:

   ```bash
   # systemctl is-enabled kdump.service && systemctl is-active kdump.service
   ```

   If kdump is not enabled and running, set all required configurations and verify that kdump service is enabled.

2. Rebuild the initramfs image of the booting kernel with the early kdump functionality:

   ```bash
   dracut -f --add earlykdump
   ```

3. Add the rd.earlykdump kernel command line parameter:

   ```bash
   grubby --update-kernel=/boot/vmlinuz-$(uname -r) --args="rd.earlykdump"
   ```

4. Reboot the system to reflect the changes

   ```bash
   reboot
   ```

5. Optionally, verify that rd.earlykdump was successfully added and early kdump feature was enabled:

   ```bash
   # cat /proc/cmdline
   BOOT_IMAGE=(hd0,msdos1)/vmlinuz-4.18.0-187.el8.x86_64 root=/dev/mapper/rhel-root ro
   crashkernel=auto resume=/dev/mapper/rhel-swap rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap
   rhgb quiet rd.earlykdump
   
   # journalctl -x | grep early-kdump
   Mar 20 15:44:41 redhat dracut-cmdline[304]: early-kdump is enabled.
   Mar 20 15:44:42 redhat dracut-cmdline[304]: kexec: loaded early-kdump kernel
   ```

Additional resources

- `/usr/share/doc/kexec-tools/early-kdump-howto.txt` file
- What is early kdump support and how do I configure it?