Red Hat Enterprise Linux 8.0 Beta

Configuring and deploying different types of servers

A guide to configuring and deploying different types of servers on Red Hat Enterprise Linux 8

Last Updated: 2019-02-25

Red Hat Enterprise Linux 8.0 Beta Configuring and deploying different types of servers

A guide to configuring and deploying different types of servers on Red Hat Enterprise Linux 8

Legal Notice

Abstract

This document describes how to configure and run different types of servers on Red Hat Enterprise Linux 8, including Apache HTTP web server, Samba server and available database servers.

Table of Contents

THIS IS A BETA VERSION!

Thank you for your interest in Red Hat Enterprise Linux 8.0 Beta. Be aware that:

- Beta code should not be used with production data or on production systems.
- Beta does not include a guarantee of support.
- Feedback and bug reports are welcome. Discussions with your account representative, partner contact, and Technical Account Manager (TAM) are also welcome.
- Upgrades to or from a Beta are not supported or recommended.

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

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

- For simple comments on specific passages, make sure you are viewing the documentation in the Multi-page HTML format. Highlight the part of text that you want to comment on. Then, click the **Add Feedback** pop-up that appears below the highlighted text, and follow the displayed instructions.

- For submitting more complex feedback, create a Bugzilla ticket:

  1. Go to the Bugzilla 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**.

CHAPTER 1. SETTING UP THE APACHE HTTP WEB SERVER

1.1. INTRODUCTION TO THE APACHE HTTP WEB SERVER

A *web server* is a network service that serves content to a client over the web. This typically means web pages, but any other documents can be served as well. Web servers are also known as HTTP servers, as they use the *hypertext transport protocol* (**HTTP**).

The web servers available in Red Hat Enterprise Linux 8 are:

- **Apache HTTP Server**
- **nginx**

The **Apache HTTP Server**, **`httpd`**, is an open source web server developed by the Apache Software Foundation.

If you are upgrading from a previous release of Red Hat Enterprise Linux, you will need to update the **`httpd`** service configuration accordingly. This section reviews some of the newly added features, and guides you through the update of prior configuration files.

1.2. NOTABLE CHANGES COMPARED TO RED HAT ENTERPRISE LINUX 7

Most notably, the **Apache HTTP Server**, has been updated from version 2.4.6 to version 2.4.33. This updated version includes several new features, but maintains backwards compatibility with Red Hat Enterprise Linux 7 version at configuration and external module Application Binary Interface (ABI) level.

The new features include:

- The multi-processing module (MPM) configured by default with the **Apache HTTP Server** has changed from a multi-process, forked model (known as **`prefork`**) to a high-performance multi-threaded model, **`event`**.
- systemd socket activation is supported (see **`httpd.socket(8)`** man page for more details).
- The minimum UID and GID allowed for users by suexec are now 1000 and 500 respectively (previously 100 and 100).
- **`/etc/sysconfig/httpd`** is no longer a supported interface for setting environment variables for the **`httpd`** service. The **`httpd.service(8)`** man page has been added for systemd service.
- **`HTTP/2`** is now supported through the **`mod_http2`** package.
- The **`mod_file_cache`** module has been removed.
- Multiple new modules have been added:
    - **`mod_proxy_hcheck`** (proxy health-check module)
    - **`mod_proxy_uswsgi`** (Web Server Gateway Interface (WSGI) proxy)
    - **`mod_proxy_fdpass`**

  - **`mod_cache_socache`** (HTTP cache using for example memcache backend)
  - **`mod_md`** (ACME protocol SSL/TLS certificate service)

- Some modules now load by default:
  - **`mod_request`**
  - **`mod_macro`**
  - **`mod_watchdog`**
- New subpackage **`httpd-filesystem`** owning directories.
- Instantiated service support (httpd@.service, described in man page).
- New **`httpd-init.service`** replaces **`%post script`** to create self-signed **`mod_ssl`** keypair.
- Stopping the **`httpd`** service now uses a “graceful stop” by default.
- The **`mod_auth_kerb`** module has been removed, and replaced by the **`mod_auth_gssapi`** module

1.3. UPDATING THE CONFIGURATION

To update the configuration files from the **Apache HTTP Server** version used in Red Hat Enterprise Linux 7, choose one of the following options:

- If **`/etc/sysconfig/httpd`** is used to set environment variables, create a systemd drop-in file instead.
- If any third-party modules are used, ensure they are compatible with a threaded MPM.
- If suexec is used, ensure user and group IDs meet the new minimums.

You can check the configuration for possible errors by using the following command:

```
~]# apachectl configtest
Syntax OK
```

1.4. RUNNING THE HTTPD SERVICE

This section describes how to start, stop, restart, and check the current status of the **Apache HTTP Server**. To be able to use the **`httpd`** service, make sure you have the **`httpd`** package installed:

```
~]# yum install httpd
```

On Red Hat Enterprise Linux 8, the **Apache HTTP Server** can be installed also through the **`httpd`** module, which is available in the Application stream.

To install the **`httpd`** module, run the following command as **`root`**:

```
~]# yum module install httpd
```

Note, that this command will install also the **mod_ssl** module, which provides the SSL/TLS support.

1.4.1. Starting the service

To run the **httpd** service, type the following at a shell prompt as **root**:

```
~]# systemctl start httpd.service
```

If you want the service to start automatically at boot time, use the following command:

```
~]# systemctl enable httpd.service
Created symlink from /etc/systemd/system/multi-
user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
```

NOTE

If running the **Apache HTTP Server** as a secure server, a password is required after the machine boots if using an encrypted private SSL key.

1.4.2. Stopping the service

To stop the running **httpd** service, type the following at a shell prompt as **root**:

```
~]# systemctl stop httpd.service
```

To prevent the service from starting automatically at boot time, type:

```
~]# systemctl disable httpd.service
Removed symlink /etc/systemd/system/multi-user.target.wants/httpd.service.
```

1.4.3. Restarting the service

There are two ways to restart a running **httpd** service:

1. To restart the service completely, enter the following command as **root**:

   ```
   ~]# systemctl restart httpd.service
   ```

   This stops the running **httpd** service and immediately starts it again. Use this command after installing or removing a dynamically loaded module such as PHP.

2. To reload the configuration without affecting active requests, enter the following command as **root**:

   ```
   ~]# systemctl reload httpd.service
   ```

   This causes the running **httpd** service to reload its configuration file. Any requests currently being processed will continue to use the old configuration.

1.4.4. Verifying the Service Status

To verify that the **httpd** service is running, type the following at a shell prompt:

```
~]# systemctl is-active httpd.service
active
```

1.5. EDITING THE CONFIGURATION FILES

When the **httpd** service is started, by default, it reads the configuration from locations that are listed in Table 1.1, “The httpd service configuration files”.

Table 1.1. The httpd service configuration files

Path	Description
/etc/httpd/conf/httpd.conf	The main configuration file.
/etc/httpd/conf.d/	An auxiliary directory for configuration files that are included in the main configuration file.
/etc/httpd/conf.modules.d/	An auxiliary directory for configuration files which load installed dynamic modules packaged in Red Hat Enterprise Linux. In the default configuration, these configuration files are processed first.

Although, the default configuration is suitable for most situations, you can use also other configuration options. For any changes to take effect, restart the web server first. See Section 1.4.3, “Restarting the service” for more information on how to restart the **httpd** service.

To check the configuration for possible errors, type the following at a shell prompt:

```
~]# apachectl configtest
Syntax OK
```

To make the recovery from mistakes easier, make a copy of the original file before editing it.

1.6. WORKING WITH MODULES

Being a modular application, the **httpd** service is distributed along with a number of *Dynamic Shared Objects* (**DSO**s), which can be dynamically loaded or unloaded at runtime as necessary. These modules are located in the **/usr/lib64/httpd/modules/** directory.

1.6.1. Loading a module

To load a particular DSO module, use the **LoadModule** directive. Note that modules provided by a separate package often have their own configuration file in the **/etc/httpd/conf.modules.d/** directory.

Example 1.1. Loading the mod_ssl DSO

```
LoadModule ssl_module modules/mod_ssl.so
```

After loading the module, restart the web server to reload the configuration. See Section 1.4.3, “Restarting the service” for more information on how to restart the **httpd** service.

1.6.2. Writing a module

To create a new DSO module, make sure you have the **httpd-devel** package installed. To do so, enter the following command as **root**:

```
~]# yum install httpd-devel
```

This package contains the include files, the header files, and the **APache eXtenSion** (**apxs**) utility required to compile a module.

Once written, you can build the module with the following command:

```
~]# apxs -i -a -c module_name.c
```

If the build was successful, you should be able to load the module the same way as any other module that is distributed with the **Apache HTTP Server**.

1.7. SETTING UP VIRTUAL HOSTS

The **Apache HTTP Server’s** built in virtual hosting allows the server to provide different information based on which IP address, host name, or port is being requested.

To create a name-based virtual host, copy the example configuration file **/usr/share/doc/httpd/httpd-vhosts.conf** into the **/etc/httpd/conf.d/** directory. Customize the options according to your requirements as shown in Example 1.2, “Example virtual host configuration”.

Example 1.2. Example virtual host configuration

```
<VirtualHost *:80>
    ServerAdmin webmaster@penguin.example.com
    DocumentRoot "/www/docs/penguin.example.com"
    ServerName penguin.example.com
    ServerAlias www.penguin.example.com
    ErrorLog "/var/log/httpd/dummy-host.example.com-error_log"
    CustomLog "/var/log/httpd/dummy-host.example.com-access_log" common
</VirtualHost>
```

Note that **ServerName** must be a valid DNS name assigned to the machine. The **<VirtualHost>** container is highly customizable, and accepts most of the directives available within the main server configuration. Directives that are **not** supported within this container include **User** and **Group**, which were replaced by **SuexecUserGroup**.

NOTE

If you configure a virtual host to listen on a non-default port, make sure you update the **`Listen`** directive in the global settings section of the **`/etc/httpd/conf/httpd.conf`** file accordingly.

To activate a newly created virtual host, restart the web server first. See Section 1.4.3, “Restarting the service” for more information on how to restart the **`httpd`** service.

1.8. SETTING UP AN SSL SERVER

Secure Sockets Layer (**SSL**) is a cryptographic protocol that allows a server and a client to communicate securely. Along with its extended and improved version called *Transport Layer Security* (**TLS**), it ensures both privacy and data integrity. The **Apache HTTP Server** in combination with **`mod_ssl`**, a module that uses the OpenSSL toolkit to provide the SSL/TLS support, is commonly referred to as the *SSL server*.

Unlike an HTTP connection that can be read and possibly modified by anybody who is able to intercept it, the use of SSL/TLS over HTTP, referred to as HTTPS, prevents any inspection or modification of the transmitted content. This section provides basic information on how to enable this module in the **Apache HTTP Server** configuration, and guides you through the process of generating private keys and self-signed certificates.

1.8.1. An overview of certificates and security

Secure communication is based on the use of keys. In conventional or *symmetric cryptography*, both ends of the transaction have the same key they can use to decode each other’s transmissions. On the other hand, in public or *asymmetric cryptography*, two keys co-exist: a *private key* that is kept a secret, and a *public key* that is usually shared with the public. While the data encoded with the public key can only be decoded with the private key, data encoded with the private key can in turn only be decoded with the public key.

To provide secure communications using SSL, an SSL server must use a digital certificate signed by a *Certificate Authority* (**CA**). The certificate lists various attributes of the server (that is, the server host name, the name of the company, its location, etc.), and the signature produced using the CA’s private key. This signature ensures that a particular certificate authority has signed the certificate, and that the certificate has not been modified in any way.

1.8.2. Certificates and security for web servers

When a web browser establishes a new SSL connection, it checks the certificate provided by the web server. If the certificate does not have a signature from a trusted CA, or if the host name listed in the certificate does not match the host name used to establish the connection, it refuses to communicate with the server and usually presents a user with an appropriate error message.

By default, most web browsers are configured to trust a set of widely used certificate authorities. Because of this, an appropriate CA should be chosen when setting up a secure server, so that target users can trust the connection, otherwise they will be presented with an error message, and will have to accept the certificate manually. Since encouraging users to override certificate errors can allow an attacker to intercept the connection, you should use a trusted CA whenever possible. For more information, see Table 1.2, “Information about CA lists used by common web browsers”.

Table 1.2. Information about CA lists used by common web browsers

Web Browser	Link
Mozilla Firefox	Mozilla root CA list.
Opera	Information on root certificates used by Opera .
Internet Explorer	Information on root certificates used by Microsoft Windows.
Chromium	Information on root certificates used by the Chromium project.

When setting up an SSL server, generate a certificate request and a private key, and then send the certificate request, proof of the company's identity, and payment to a certificate authority. Once the CA verifies the certificate request and your identity, it will send you a signed certificate you can use with your server. Alternatively, you can create a self-signed certificate that does not contain a CA signature, and thus should be used for testing purposes only.

1.9. ENABLING THE MOD_SSL MODULE

If you intend to set up an SSL or HTTPS server using **mod_ssl**, you **cannot** have the another application or module configured to use the same port. Port **443** is the default port for HTTPS.

To set up an SSL server using the **mod_ssl** module and the OpenSSL toolkit, install the **mod_ssl** and **openssl** packages. Enter the following command as **root**:

```
~]# yum install mod_ssl openssl
```

This will create the **mod_ssl** configuration file at **/etc/httpd/conf.d/ssl.conf**, which is included in the main **Apache HTTP Server** configuration file by default. For the module to be loaded, restart the **httpd** service as described in Section 1.4.3, "Restarting the service".

IMPORTANT

Due to the vulnerability described in POODLE: SSLv3 vulnerability (CVE-2014-3566), Red Hat recommends disabling **SSL** and using only **TLSv1.1** or **TLSv1.2**. Backwards compatibility can be achieved using **TLSv1.0**. Many products Red Hat supports have the ability to use **SSLv2** or **SSLv3** protocols, or enable them by default. However, the use of **SSLv2** or **SSLv3** is now strongly recommended against.

1.9.1. Enabling and disabling SSL and TLS in mod_ssl

To disable and enable specific versions of the SSL and TLS protocol, either do it globally by adding the **SSLProtocol** directive in the "SSL Global Context" section of the configuration file and removing it everywhere else, or edit the default entry under "SSL Protocol support" in all "VirtualHost" sections. If you do not specify it in the per-domain VirtualHost section then it will inherit the settings from the global section. To make sure that a protocol version is being disabled the administrator should either **only** specify **SSLProtocol** in the "SSL Global Context" section, or specify it in **all** per-domain VirtualHost sections.

Disable SSLv2 and SSLv3

To disable SSL version 2 and SSL version 3, which implies enabling everything except SSL version 2 and SSL version 3, in all VirtualHost sections, proceed as follows:

1. As **`root`**, open the **`/etc/httpd/conf.d/ssl.conf`** file and search for **all** instances of the **`SSLProtocol`** directive. By default, the configuration file contains one section that looks as follows:

   ```
   ~]# vi /etc/httpd/conf.d/ssl.conf
   #   SSL Protocol support:
   # List the enable protocol levels with which clients will be able to
   # connect.  Disable SSLv2 access by default:
   SSLProtocol all -SSLv2
   ```

   This section is within the VirtualHost section.

2. Edit the **`SSLProtocol`** line as follows:

   ```
   #   SSL Protocol support:
   # List the enable protocol levels with which clients will be able to
   # connect.  Disable SSLv2 access by default:
   SSLProtocol all -SSLv2 -SSLv3
   ```

   Repeat this action for all VirtualHost sections. Save and close the file.

3. Verify that all occurrences of the **`SSLProtocol`** directive have been changed as follows:

   ```
   ~]# grep SSLProtocol /etc/httpd/conf.d/ssl.conf
   SSLProtocol all -SSLv2 -SSLv3
   ```

   This step is particularly important if you have more than the one default VirtualHost section.

4. Restart the Apache daemon as follows:

   ```
   ~]# systemctl restart httpd
   ```

   Note that any sessions will be interrupted.

Disable All SSL and TLS Protocols Except TLS 1 and Up

To disable all SSL and TLS protocol versions except TLS version 1 and higher, proceed as follows:

1. As **`root`**, open the **`/etc/httpd/conf.d/ssl.conf`** file and search for **all** instances of **`SSLProtocol`** directive. By default the file contains one section that looks as follows:

   ```
   ~]# vi /etc/httpd/conf.d/ssl.conf
   #   SSL Protocol support:
   # List the enable protocol levels with which clients will be able to
   # connect.  Disable SSLv2 access by default:
   SSLProtocol all -SSLv2
   ```

2. Edit the **`SSLProtocol`** line as follows:

   ```
   #   SSL Protocol support:
   ```

```
# List the enable protocol levels with which clients will be able to
# connect.  Disable SSLv2 access by default:
SSLProtocol -all +TLSv1 +TLSv1.1 +TLSv1.2
```

Save and close the file.

3. Verify the change as follows:

   ```
   ~]# grep SSLProtocol /etc/httpd/conf.d/ssl.conf
   SSLProtocol -all +TLSv1 +TLSv1.1 +TLSv1.2
   ```

4. Restart the Apache daemon as follows:

   ```
   ~]# systemctl restart httpd
   ```

   Note that any sessions will be interrupted.

Testing the Status of SSL and TLS Protocols

To check which versions of SSL and TLS are enabled or disabled, make use of the **`openssl s_client -connect`** command. The command has the following form:

```
openssl s_client -connect hostname:port -protocol
```

Where *port* is the port to test and *protocol* is the protocol version to test for. To test the SSL server running locally, use **`localhost`** as the host name. For example, to test the default port for secure HTTPS connections, port **`443`** to see if SSLv3 is enabled, issue a command as follows:

```
~]# openssl s_client -connect localhost:443 -ssl3
CONNECTED(00000003)
139809943877536:error:14094410:SSL routines:SSL3_READ_BYTES:sslv3 alert
handshake failure:s3_pkt.c:1257:SSL alert number 40
139809943877536:error:1409E0E5:SSL routines:SSL3_WRITE_BYTES:ssl handshake
failure:s3_pkt.c:596:
output omitted
New, (NONE), Cipher is (NONE)
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol  : SSLv3
output truncated
```

The above output indicates that the handshake failed and therefore no cipher was negotiated.

```
~]$ openssl s_client -connect localhost:443 -tls1_2
CONNECTED(00000003)
depth=0 C = --, ST = SomeState, L = SomeCity, O = SomeOrganization, OU =
SomeOrganizationalUnit, CN = localhost.localdomain, emailAddress =
root@localhost.localdomain
output omitted
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
```

```
Expansion: NONE
SSL-Session:
    Protocol  : TLSv1.2
output truncated
```

The above output indicates that no failure of the handshake occurred and a set of ciphers was negotiated.

The **openssl s_client** command options are documented in the **s_client(1)** manual page.

For more information on the SSLv3 vulnerability and how to test for it, see the Red Hat Knowledgebase article POODLE: SSLv3 vulnerability (CVE-2014-3566)_.

1.10. USING AN EXISTING KEY AND CERTIFICATE

If you have a previously created key and certificate, you can configure the SSL server to use these files instead of generating new ones. There are only two situations where this is not possible:

1. **You are changing the IP address or domain name.**
   Certificates are issued for a particular IP address and domain name pair. If one of these values changes, the certificate becomes invalid.

2. **You have a certificate from VeriSign, and you are changing the server software.**
   VeriSign, a widely used certificate authority, issues certificates for a particular software product, IP address, and domain name. Changing the software product renders the certificate invalid.

In either of the above cases, you will need to obtain a new certificate.

If you want to use an existing key and certificate, move the relevant files to the **/etc/pki/tls/private/** and **/etc/pki/tls/certs/** directories by issuing the following commands as **root**:

```
~]# mv key_file.key /etc/pki/tls/private/hostname.key
~]# mv certificate.crt /etc/pki/tls/certs/hostname.crt
```

Then add the following lines to the **/etc/httpd/conf.d/ssl.conf** configuration file:

```
SSLCertificateFile /etc/pki/tls/certs/hostname.crt
SSLCertificateKeyFile /etc/pki/tls/private/hostname.key
```

To load the updated configuration, restart the **httpd** service as described in Section 1.4.3, “Restarting the service”.

1.11. CONFIGURE THE FIREWALL FOR HTTP AND HTTPS USING THE COMMAND LINE

Red Hat Enterprise Linux does not allow **HTTP** and **HTTPS** traffic by default. To enable the system to act as a web server, make use of **firewalld**'s supported services to enable **HTTP** and **HTTPS** traffic to pass through the firewall as required.

To enable **HTTP** using the command line, issue the following command as **root**:

```
~]# firewall-cmd --add-service http
 success
```

To enable **HTTPS** using the command line, issue the following command as **root**:

```
~]# firewall-cmd --add-service https
 success
```

Note that these changes does not persist after the next system start. To make permanent changes to the firewall, repeat the commands adding the **--permanent** option.

1.11.1. Checking network access for incoming HTTPS and HTTPS using the command line

To check what services the firewall is configured to allow, using the command line, issue the following command as **root**:

```
~]# firewall-cmd --list-all
public (default, active)
  interfaces: em1
  sources:
  services: dhcpv6-client ssh
output truncated
```

In this example taken from a default installation, the firewall is enabled but **HTTP** and **HTTPS** have not been allowed to pass through.

Once the **HTTP** and **HTTP** firewall services are enabled, the **services** line will appear similar to the following:

```
services: dhcpv6-client http https ssh
```

1.12. ADDITIONAL RESOURCES

To learn more about the **Apache HTTP Server**, see the following resources.

1.12.1. Installed documentation

- **httpd(8)** — The manual page for the **httpd** service containing the complete list of its command-line options.
- **httpd.service(8)** — The manual page for the **httpd.service** unit file, describing how to customize and enhance the service."
- **apachectl(8)** — The manual page for the **Apache HTTP Server** Control Interface.

1.12.2. Installable documentation

- http://localhost/manual/ — The official documentation for the **Apache HTTP Server** with the full description of its directives and available modules. Note that in order to access this documentation, you must have the **httpd-manual** package installed, and the web server must be running.

Before accessing the documentation, issue the following commands as **root**:

```
~]# yum install httpd-manual
~]# apachectl graceful
```

1.12.3. Online documentation

- http://httpd.apache.org/ — The official website for the **Apache HTTP Server** with documentation on all the directives and default modules.

- http://www.openssl.org/ — The OpenSSL home page containing further documentation, frequently asked questions, links to the mailing lists, and other useful resources.

CHAPTER 2. USING SAMBA AS A SERVER

Samba implements the Server Message Block (SMB) protocol in Red Hat Enterprise Linux. The SMB protocol is used to access resources on a server, such as file shares and shared printers. Additionally, Samba implements the Distributed Computing Environment Remote Procedure Call (DCE RPC) protocol used by Microsoft Windows.

You can run Samba as:

- An Active Directory (AD) or NT4 domain member
- A standalone server
- An NT4 Primary Domain Controller (PDC) or Backup Domain Controller (BDC)

> **NOTE**
>
> Red Hat supports the PDC and BDC modes only in existing installations with Windows versions which support NT4 domains. Red Hat recommends not setting up a new Samba NT4 domain, because Microsoft operating systems later than Windows 7 and Windows Server 2008 R2 do not support NT4 domains.
>
> Red Hat does not support running Samba as an AD domain controller (DC).

Independently of the installation mode, you can optionally share directories and printers. This enables Samba to act as a file and print server.

Prerequisites

- Red Hat Enterprise Linux 8 is installed on the server.

2.1. THE SAMBA SERVICES

Samba provides the following services:

`smbd`

This service provides file sharing and printing services using the SMB protocol. Additionally, the service is responsible for resource locking and for authenticating connecting users. The **`smb systemd`** service starts and stops the **`smbd`** daemon.
To use the **`smbd`** service, install the **`samba`** package.

`nmbd`

This service provides host name and IP resolution using the NetBIOS over IPv4 protocol. Additionally to the name resolution, the **`nmbd`** service enables browsing the SMB network to locate domains, work groups, hosts, file shares, and printers. For this, the service either reports this information directly to the broadcasting client or forwards it to a local or master browser. The **`nmb systemd`** service starts and stops the **`nmbd`** daemon.
Note that modern SMB networks use DNS to resolve clients and IP addresses.

To use the **`nmbd`** service, install the **`samba`** package.

`winbindd`

This service provides an interface for the Name Service Switch (NSS) to use AD or NT4 domain

users and groups on the local system. This enables, for example, domain users to authenticate to services hosted on a Samba server or to other local services. The **`winbind systemd`** service starts and stops the **`winbindd`** daemon.
If you set up Samba as a domain member, **`winbindd`** must be started before the **`smbd`** service. Otherwise, domain users and groups are not available to the local system..

To use the **`winbindd`** service, install the **`samba-winbind`** package.

IMPORTANT

Red Hat only supports running Samba as a server with the **`winbindd`** service to provide domain users and groups to the local system. Due to certain limitations, such as missing Windows access control list (ACL) support and NT LAN Manager (NTLM) fallback, SSSD is not supported.

2.2. VERIFYING THE SMB.CONF FILE BY USING THE TESTPARM UTILITY

The **`testparm`** utility verifies that the Samba configuration in the **`/etc/samba/smb.conf`** file is correct. The utility detects invalid parameters and values, but also incorrect settings, such as for ID mapping. If **`testparm`** reports no problem, the Samba services will successfully load the **`/etc/samba/smb.conf`** file. Note that **`testparm`** cannot verify that the configured services will be available or work as expected.

IMPORTANT

Red Hat recommends that you verify the **`/etc/samba/smb.conf`** file by using **`testparm`** after each modification of this file.

Procedure

1. Run the **`testparm`** utility as the **`root`** user:

```
# testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit
(16384)
Unknown parameter encountered: "log levell"
Processing section "[example_share]"
Loaded services file OK.
ERROR: The idmap range for the domain * (tdb) overlaps with the
range of DOMAIN (ad)!

Server role: ROLE_DOMAIN_MEMBER

Press enter to see a dump of your service definitions

# Global parameters
[global]
 ...

[example_share]
 ...
```

The previous example output reports a non-existent parameter and an incorrect ID mapping configuration.

2. If **testparm** reports incorrect parameters, values, or other errors in the configuration, fix the problem and run the utility again.

2.3. THE SAMBA SECURITY SERVICES

The **security** parameter in the **[global]** section in the **/etc/samba/smb.conf** file manages how Samba authenticates users that are connecting to the service. Depending on the mode you install Samba in, the parameter must be set to different values:

On an AD domain member, set security = ads

In this mode, Samba uses Kerberos to authenticate AD users.
For details about setting up Samba as a domain member, see Section 2.5, “Setting up Samba as a domain member server”.

On a standalone server, set security = user

In this mode, Samba uses a local database to authenticate connecting users.
For details about setting up Samba as a standalone server, see Section 2.4, “Setting up Samba as a standalone server”.

On an NT4 PDC or BDC, set security = user

In this mode, Samba authenticates users to a local or LDAP database.

On an NT4 domain member, set security = domain

In this mode, Samba authenticates connecting users to an NT4 PDC or BDC. You cannot use this mode on AD domain members.
For details about setting up Samba as a domain member, see Section 2.5, “Setting up Samba as a domain member server”.

Additional resources

- See the description of the **security** parameter in the **smb.conf(5)** man page.

2.4. SETTING UP SAMBA AS A STANDALONE SERVER

You can set up Samba as a server that is not a member of a domain. In this installation mode, Samba authenticates users to a local database instead of to a central DC. Additionally, you can enable guest access to allow users to connect to one or multiple services without authentication.

2.4.1. Setting up the server configuration for the standalone server

This section describes how to set up the server configuration for a Samba standalone server.

Procedure

1. Install the **samba** package:

```
# yum install samba
```

2. Edit the **/etc/samba/smb.conf** file and set the following parameters:

```
[global]
 workgroup = Example-WG
 netbios name = Server
 security = user

 log file = /var/log/samba/%m.log
 log level = 1
```

This configuration defines a standalone server named **`Server`** within the **`Example-WG`** work group. Additionally, this configuration enables logging on a minimal level (**`1`**) and log files will be stored in the **`/var/log/samba/`** directory. Samba will expand the **`%m`** macro in the **`log file`** parameter to the NetBIOS name of connecting clients. This enables individual log files for each client.

3. Configure file or printer sharing. See:
   - Section 2.6, “Configuring file shares on a Samba server”
   - Section 2.7, “Setting up Samba as a print server ”
4. Verify the **`/etc/samba/smb.conf`** file:

```
# testparm
```

5. If you set up shares that require authentication, create the user accounts. See Section 2.4.2, “Creating and enabling local user accounts”.
6. Open the required ports and reload the firewall configuration by using the **`firewall-cmd`** utility:

```
# firewall-cmd --permanent --add-port={139/tcp,445/tcp}
# firewall-cmd --reload
```

7. Start the **`smb`** service:

```
# systemctl start smb
```

Optionally, enable the **`smb`** service to start automatically when the system boots:

```
# systemctl enable smb
```

Additional resources

- For further details about the parameters used in the procedure, see the descriptions of the parameters in the **`smb.conf(5)`** man page.
- Section 2.2, “Verifying the **`smb.conf`** file by using the **`testparm`** utility”

2.4.2. Creating and enabling local user accounts

To enable users to authenticate when they connect to a share, you must create the accounts on the Samba host both in the operating system and in the Samba database. Samba requires the operating system account to validate the Access Control Lists (ACL) on file system objects and the Samba account to authenticate connecting users.

If you use the **passdb backend = tdbsam** default setting, Samba stores user accounts in the **/var/lib/samba/private/passdb.tdb** database.

The procedure in this section describes how to create a local Samba user named **example**.

Prerequisites

- Samba is installed configured as a standalone server.

Procedure

1. Create the operating system account:

   ```
   # useradd -M -s /sbin/nologin example
   ```

   This command adds the **example** account without creating a home directory. If the account is only used to authenticate to Samba, assign the **/sbin/nologin** command as shell to prevent the account from logging in locally.

2. Set a password to the operating system account to enable it:

   ```
   # passwd example
   Enter new UNIX password: password
   Retype new UNIX password: password
   passwd: password updated successfully
   ```

   Samba does not use the password set on the operating system account to authenticate. However, you need to set a password to enable the account. If an account is disabled, Samba denies access if this user connects.

3. Add the user to the Samba database and set a password to the account:

   ```
   # smbpasswd -a example
   New SMB password: password
   Retype new SMB password: password
   Added user example.
   ```

   Use this password to authenticate when using this account to connect to a Samba share.

4. Enable the Samba account:

   ```
   # smbpasswd -e example
   Enabled user example.
   ```

2.5. SETTING UP SAMBA AS A DOMAIN MEMBER SERVER

If you are running an AD or NT4 domain, use Samba to add your Red Hat Enterprise Linux server as a member to the domain to gain the following:

- Access domain resources on other domain members
- Authenticate domain users to local services, such as **sshd**
- Share directories and printers hosted on the server to act as a file and print server

2.5.1. Joining Samba to a Domain

This section describes how to join a Red Hat Enterprise Linux system to a domain.

Procedure

1. Install the following packages:

   ```
   # yum install realmd oddjob-mkhomedir oddjob samba-winbind-clients \
          samba-winbind samba-common-tools samba
   ```

2. To share directories or printers on the domain member, install the **samba** package:

   ```
   # yum install samba
   ```

3. If you are joining an AD, additionally install the Winbind Kerberos locator plug-in:

   ```
   # yum install samba-winbind-krb5-locator
   ```

   This plug-in enables Kerberos to locate the Key Distribution Center (KDC) based on AD sites using DNS service records.

4. Optionally, for backup purposes, rename the existing **/etc/samba/smb.conf** Samba configuration file:

   ```
   # mv /etc/samba/smb.conf /etc/samba/smb.conf.old
   ```

5. Join the domain. For example, to join a domain named **ad.example.com**:

   ```
   # realm join --client-software=winbind ad.example.com
   ```

   Using the previous command, the **realm** utility automatically:

   - Creates a **/etc/samba/smb.conf** file for a membership in the **ad.example.com** domain
   - Adds the **winbind** module for user and group lookups to the **/etc/nsswitch.conf** file
   - Updates the Pluggable Authentication Module (PAM) configuration files in the **/etc/pam.d/** directory
   - Starts the **winbind** service and enables the service to start when the system boots

6. Optionally, set an alternative ID mapping back end or customized ID mapping settings in the **/etc/samba/smb.conf** file. For details, see Section 2.5.4, “Samba ID mapping”.

7. Optionally, verify the configuration. See Section 2.5.2, “Verifying that Samba was correctly joined as a domain member”.

8. Verify that the **winbind** service is running:

   ```
   # systemctl status winbind
   ...
      Active: active (running) since Tue 2018-11-06 19:10:40 CET; 15s
   ago
   ```

IMPORTANT

To enable Samba to query domain user and group information, the **winbind** service must be running before you start **smb**.

9. If you installed the **samba** package to share directories and printers, start the **smb** service:

   ```
   # systemctl start smb
   ```

10. Optionally, if you are authenticating local logins to Active Directory, enable the **winbind_krb5_localauth** plug-in. See Section 2.5.3, “Using the local authorization plug-in for MIT Kerberos”.

Additional resources

- For further details about the **realm** utility, see
  - The **realm(8)** man page

2.5.2. Verifying that Samba was correctly joined as a domain member

To verify you added your Red Hat Enterprise Linux correctly to your domain perform the tests described in this section on the domain member.

2.5.2.1. Verifying that the operating system can retrieve domain user accounts and groups

Use the **getent** and **chown** utilities to verify that the operating system can retrieve domain users and groups

Procedure

1. To query the **administrator** account in the **AD** domain:

   ```
   # getent passwd "AD\administrator"
   AD\administrator:*:10000:10000::/home/administrator@AD:/bin/bash
   ```

2. To query the members of the **Domain Users** group in the **AD** domain:

   ```
   # getent group "AD\Domain Users"
   AD\domain users:x:10000:user1,user2
   ```

3. Verify that you can use domain users and groups when you set permissions on files and directories. For example, to set the owner of the **/srv/samba/example.txt** file to **AD\administrator** and the group to **AD\Domain Users**:

   ```
   # chown "AD\administrator":"AD\Domain Users" /srv/samba/example.txt
   ```

2.5.2.2. Verifying if AD domain users can obtain a Kerberos ticket

In an AD environment, users can obtain a Kerberos ticket from the DC.

The procedure in this section describes how to verify if the **administrator** user can obtain a Kerberos ticket.

Prerequisites

- Install the **krb5-workstation** package on the Samba domain member

Procedure

1. On the AD domain member, obtain a ticket for the administrator@AD.EXAMPLE.COM principal:

```
# kinit administrator@AD.EXAMPLE.COM
```

2. Display the cached Kerberos ticket:

```
# klist
Ticket cache: KCM:0
Default principal: administrator@AD.EXAMPLE.COM

Valid starting       Expires              Service principal
01.11.2018 10:00:00  01.11.2018 20:00:00
krbtgt/AD.EXAMPLE.COM@AD.EXAMPLE.COM
        renew until 08.11.2018 05:00:00
```

2.5.2.3. Listing the available domains

Use the **wbinfo** utility to list all domains available through the **winbindd** service.

Procedure

```
# wbinfo --all-domains
```

If Samba was successfully joined as a domain member, the command displays the built-in and local host name, as well as the domain Samba is a member of including trusted domains.

Example 2.1. Displaying the available domains

The following is an example of the **wbinfo --all-domains** command's output:

```
# wbinfo --all-domains
BUILTIN
SAMBA-SERVER
AD
```

2.5.3. Using the local authorization plug-in for MIT Kerberos

The **winbind** service provides Active Directory users to the domain member. In certain situations, administrators want to enable domain users to authenticate to local services, such as an SSH server, which are running on the domain member. When using Kerberos to authenticate the domain users, enable the **winbind_krb5_localauth** plug-in to correctly map Kerberos principals to Active Directory accounts through the **winbind** service.

For example, if the **sAMAccountName** attribute of an Active Directory user is set to **EXAMPLE** and the user tries to log with the user name lowercase, Kerberos returns the user name in upper case. As a consequence, the entries do not match an authentication fails.

Using the **winbind_krb5_localauth** plug-in, the account names are mapped correctly. Note that this only applies to GSSAPI authentication and not for getting the initial ticket granting ticket (TGT).

Prerequisites

- Samba is configured as a member of an Active Directory.
- Red Hat Enterprise Linux authenticates log in attempts against Active Directory.
- The **winbind** service is running.

Procedure
Edit the **/etc/krb5.conf** file and add the following section:

```
[plugins]
localauth = {
     module = winbind:/usr/lib64/samba/krb5/winbind_krb5_localauth.so
     enable_only = winbind
}
```

Additional resources

- See the **winbind_krb5_localauth(8)** man page.

2.5.4. Samba ID mapping

Windows domains distinguish users and groups by unique Security Identifiers (SID). However, Linux requires unique UIDs and GIDs for each user and group. If you run Samba as a domain member, the **winbindd** service is responsible for providing information about domain users and groups to the operating system.

To enable the **winbindd** service to provide unique IDs for users and groups to Linux, you must configure ID mapping in the **/etc/samba/smb.conf** file for:

- The local database (default domain)
- The AD or NT4 domain the Samba server is a member of
- Each trusted domain from which users must be able to access resources on this Samba server

2.5.4.1. Planning Samba ID ranges

Regardless of whether you store the Linux UIDs and GIDs in AD or if you configure Samba to generate them, each domain configuration requires a unique ID range that must not overlap with any of the other domains.

WARNING

If you set overlapping ID ranges, Samba fails to work correctly.

Example 2.2. Unique ID Ranges

The following shows non-overlapping ID mapping ranges for the default (`*`), **`AD-DOM`**, and the **`TRUST-DOM`** domains.

```
[global]
...
idmap config * : backend = tdb
idmap config * : range = 10000-999999

idmap config AD-DOM:backend = rid
idmap config AD-DOM:range = 2000000-2999999

idmap config TRUST-DOM:backend = rid
idmap config TRUST-DOM:range = 4000000-4999999
```

IMPORTANT

You can only assign one range per domain. Therefore, leave enough space between the domains ranges. This enables you to extend the range later if your domain grows.

If you later assign a different range to a domain, the ownership of files and directories previously created by these users and groups will be lost.

2.5.4.2. The `*` default domain

In a domain environment, you add one ID mapping configuration for each of the following:

- The domain the Samba server is a member of
- Each trusted domain that should be able to access the Samba server

However, for all other objects, Samba assigns IDs from the default domain. This includes:

- Local Samba users and groups
- Samba built-in accounts and groups, such as **`BUILTIN\Administrators`**

IMPORTANT

You must configure the default domain as described in this section to enable Samba to operate correctly.

The default domain back end must be writable to permanently store the assigned IDs.

For the default domain, you can use one of the following back ends:

tdb

When you configure the default domain to use the **tdb** back end, set an ID range that is big enough to include objects that will be created in the future and that are not part of a defined domain ID mapping configuration.
For example, set the following in the **[global]** section in the **/etc/samba/smb.conf** file:

```
idmap config * : backend = tdb
idmap config * : range = 10000-999999
```

For further details, see Section 2.5.5.1, "Using the **tdb** ID mapping back end".

autorid

When you configure the default domain to use the **autorid** back end, adding additional ID mapping configurations for domains is optional.
For example, set the following in the **[global]** section in the **/etc/samba/smb.conf** file:

```
idmap config * : backend = autorid
idmap config * : range = 10000-999999
```

For further details, see Section 2.5.5.4, "Using the **autorid** ID mapping back end ".

2.5.5. The different Samba ID mapping back ends

Samba provides different ID mapping back ends for specific configurations. The most frequently used back ends are:

Back end	Use case
tdb	The * default domain only
ad	AD domains only
rid	AD and NT4 domains
autorid	AD, NT4, and the * default domain

The following sections describe the benefits, recommended scenarios where to use the back end, and how to configure it.

2.5.5.1. Using the tdb ID mapping back end

The **winbindd** service uses the writable **tdb** ID mapping back end by default to store Security Identifier (SID), UID, and GID mapping tables. This includes local users, groups, and built-in principals.

Use this back end only for the * default domain. For example:

```
idmap config * : backend = tdb
idmap config * : range = 10000-999999
```

Additional resources

- Section 2.5.4.2, “The `*` default domain”.

2.5.5.2. Using the `ad` ID mapping back end

This section describes how to configure a Samba AD member to use the **`ad`** ID mapping back end.

The **`ad`** ID mapping back end implements a read-only API to read account and group information from AD. This provides the following benefits:

- All user and group settings are stored centrally in AD.
- User and group IDs are consistent on all Samba servers that use this back end.
- The IDs are not stored in a local database which can corrupt, and therefore file ownerships cannot be lost.

NOTE

The **`ad`** ID mapping back end does not support Active Directory domains with one-way trusts. If you configure a domain member in an Active Directory with one-way trusts, use instead one of the following ID mapping back ends: **`tdb`**, **`rid`**, or **`autorid`**.

The ad back end reads the following attributes from AD:

AD attribute name	Object type	Mapped to
`sAMAccountName`	User and group	User or group name, depending on the object
`uidNumber`	User	User ID (UID)
`gidNumber`	Group	Group ID (GID)
`loginShell` [a]	User	Path to the shell of the user
`unixHomeDirectory` [a]	User	Path to the home directory of the user
`primaryGroupID` [b]	User	Primary group ID

[a] Samba only reads this attribute if you set **`idmap config`** ***`DOMAIN`***:**`unix_nss_info = yes`**.

[b] Samba only reads this attribute if you set **`idmap config`** ***`DOMAIN`***:**`unix_primary_group = yes`**.

Prerequisites

To use the **ad** ID mapping back end:

- Both users and groups must have unique IDs set in AD, and the IDs must be within the range configured in the **/etc/samba/smb.conf** file. Objects whose IDs are outside of the range will not be available on the Samba server.

- Users and groups must have all required attributes set in AD. If required attributes are missing, the user or group will not be available on the Samba server. The required attributes depend on your configuration.

Procedure

1. Edit the **[global]** section in the **/etc/samba/smb.conf** file:

   a. Add an ID mapping configuration for the default domain (*****) if it does not exist. For example:

      ```
      idmap config * : backend = tdb
      idmap config * : range = 10000-999999
      ```

   b. Enable the **ad** ID mapping back end for the AD domain:

      ```
      idmap config DOMAIN : backend = ad
      ```

   c. Set the range of IDs that is assigned to users and groups in the AD domain. For example:

      ```
      idmap config DOMAIN : range = 2000000-2999999
      ```

      IMPORTANT

      The range must not overlap with any other domain configuration on this server. Additionally, the range must be set big enough to include all IDs assigned in the future. For further details, see Section 2.5.4.1, “Planning Samba ID ranges”.

   d. Set that Samba uses the RFC 2307 schema when reading attributes from AD:

      ```
      idmap config DOMAIN : schema_mode = rfc2307
      ```

   e. To enable Samba to read the login shell and the path to the users home directory from the corresponding AD attribute, set:

      ```
      idmap config DOMAIN : unix_nss_info = yes
      ```

      Alternatively, you can set a uniform domain-wide home directory path and login shell that is applied to all users. For example:

      ```
      template shell = /bin/bash
      template homedir = /home/%U
      ```

   f. By default, Samba uses the **primaryGroupID** attribute of a user object as the user’s primary group on Linux. Alternatively, you can configure Samba to use the value set in the **gidNumber** attribute instead:

```
idmap config DOMAIN : unix_primary_group = yes
```

2. Verify the **/etc/samba/smb.conf** file:

   ```
   # testparm
   ```

3. Reload the Samba configuration:

   ```
   # smbcontrol all reload-config
   ```

4. Verify that the settings work as expected. See Section 2.5.2.1, “Verifying that the operating system can retrieve domain user accounts and groups”.

Additional resources

- Section 2.5.4.2, “The `*` default domain”
- For further details about the parameters used in the procedure, see the **smb.conf(5)** and **idmap_ad(8)** man pages.
- For details about variable substitution, see the **VARIABLE SUBSTITUTIONS** section in the **smb.conf(5)** man page.
- Section 2.2, “Verifying the **smb.conf** file by using the **testparm** utility”

2.5.5.3. Using the `rid` ID mapping back end

This section describes how to configure a Samba domain member to use the **rid** ID mapping back end.

Samba can use the relative identifier (RID) of a Windows SID to generate an ID on Red Hat Enterprise Linux.

NOTE

The RID is the last part of a SID. For example, if the SID of a user is **S-1-5-21-5421822485-1151247151-421485315-30014**, then **30014** is the corresponding RID.

The **rid** ID mapping back end implements a read-only API to calculate account and group information based on an algorithmic mapping scheme for AD and NT4 domains. When you configure the back end, you must set the lowest and highest RID in the **idmap config *DOMAIN* : *range*** parameter. Samba will not map users or groups with a lower or higher RID than set in this parameter.

IMPORTANT

As a read-only back end, **rid** cannot assign new IDs, such as for **BUILTIN** groups. Therefore, do not use this back end for the **`*`** default domain.

Benefits of using the `rid` back end

- All domain users and groups that have an RID within the configured range are automatically available on the domain member.

- You do not need to manually assign IDs, home directories, and login shells.

Drawbacks of using the rid back end

- All domain users get the same login shell and home directory assigned. However, you can use variables.
- User and group IDs are only the same across Samba domain members if all use the **rid** back end with the same ID range settings.
- You cannot exclude individual users or groups from being available on the domain member. Only users and groups outside of the configured range are excluded.
- Based on the formulas the **winbindd** service uses to calculate the IDs, duplicate IDs can occur in multi-domain environments if objects in different domains have the same RID.

Procedure

1. Edit the **[global]** section in the **/etc/samba/smb.conf** file:

   a. Add an ID mapping configuration for the default domain (*****) if it does not exist. For example:

   ```
   idmap config * : backend = tdb
   idmap config * : range = 10000-999999
   ```

   b. Enable the **rid** ID mapping back end for the domain:

   ```
   idmap config DOMAIN : backend = rid
   ```

   c. Set a range that is big enough to include all RIDs that will be assigned in the future. For example:

   ```
   idmap config DOMAIN : range = 2000000-2999999
   ```

   Samba ignores users and groups whose RIDs in this domain are not within the range.

   IMPORTANT

   The range must not overlap with any other domain configuration on this server. For further details, see Section 2.5.4.1, “Planning Samba ID ranges”.

   d. Set a shell and home directory path that will be assigned to all mapped users. For example:

   ```
   template shell = /bin/bash
   template homedir = /home/%U
   ```

2. Verify the **/etc/samba/smb.conf** file:

   ```
   # testparm
   ```

3. Reload the Samba configuration:

   ```
   # smbcontrol all reload-config
   ```

Verify that the settings work as expected. See Section 2.5.2.1, “Verifying that the operating system can retrieve domain user accounts and groups”.

Additional resources

- Section 2.5.4.2, “The `*` default domain”
- For details about variable substitution, see the **`VARIABLE SUBSTITUTIONS`** section in the **`smb.conf(5)`** man page.
- For details, how Samba calculates the local ID from a RID, see the **`idmap_rid(8)`** man page.
- Section 2.2, “Verifying the **`smb.conf`** file by using the **`testparm`** utility”

2.5.5.4. Using the `autorid` ID mapping back end[1]

This section describes how to configure a Samba domain member to use the **`autorid`** ID mapping back end.

The **`autorid`** back end works similar to the **`rid`** ID mapping back end, but can automatically assign IDs for different domains. This enables you to use the **`autorid`** back end in the following situations:

- Only for the **`*`** default domain
- For the **`*`** default domain and additional domains, without the need to create ID mapping configurations for each of the additional domains
- Only for specific domains

NOTE

If you use **`autorid`** for the default domain, adding additional ID mapping configuration for domains is optional.

Benefits of using the `autorid` back end

- All domain users and groups whose calculated UID and GID is within the configured range are automatically available on the domain member.
- You do not need to manually assign IDs, home directories, and login shells.
- No duplicate IDs, even if multiple objects in a multi-domain environment have the same RID.

Drawbacks

- User and group IDs are not the same across Samba domain members.
- All domain users get the same login shell and home directory assigned. However, you can use variables.
- You cannot exclude individual users or groups from being available on the domain member. Only users and groups whose calculated UID or GID is outside of the configured range are excluded.

Procedure

1. Edit the **`[global]`** section in the **`/etc/samba/smb.conf`** file

1. Edit the **[global]** section in the **/etc/samba/smb.conf** file:

   a. Enable the **autorid** ID mapping back end for the ***** default domain:

   ```
   idmap config * : backend = autorid
   ```

   b. Set a range that is big enough to assign IDs for all existing and future objects. For example:

   ```
   idmap config * : range = 10000-999999
   ```

   Samba ignores users and groups whose calculated IDs in this domain are not within the range.

   > **WARNING**
   >
   > After you set the range and Samba starts using it, you can only increase the upper limit of the range. Any other change to the range can result in new ID assignments, and thus in losing file ownerships.

   c. Optionally, set a range size. For example:

   ```
   idmap config * : rangesize = 200000
   ```

   Samba assigns this number of continuous IDs for each domain's object until all IDs from the range set in the **idmap config * : *range*** parameter are taken.

   d. Set a shell and home directory path that will be assigned to all mapped users. For example:

   ```
   template shell = /bin/bash
   template homedir = /home/%U
   ```

   e. Optionally, add additional ID mapping configuration for domains. If no configuration for an individual domain is available, Samba calculates the ID using the **autorid** back end settings in the previously configured ***** default domain.

   > **IMPORTANT**
   >
   > If you configure additional back ends for individual domains, the ranges for all ID mapping configuration must not overlap. For further details, see Section 2.5.4.1, "Planning Samba ID ranges".

2. Verify the **/etc/samba/smb.conf** file:

   ```
   # testparm
   ```

3. Reload the Samba configuration:

   ```
   # smbcontrol all reload-config
   ```

4. Verify that the settings work as expected. See Section 2.5.2.1, “Verifying that the operating system can retrieve domain user accounts and groups”.

Additional resources

- For details about how the back end calculated IDs, see the **`THE MAPPING FORMULAS`** section in the **`idmap_autorid(8)`** man page.
- For details about using the **`idmap config rangesize`** parameter, see the **`rangesize`** parameter description in the **`idmap_autorid(8)`** man page.
- For details about variable substitution, see the **`VARIABLE SUBSTITUTIONS`** section in the **`smb.conf(5)`** man page.
- Section 2.2, “Verifying the **`smb.conf`** file by using the **`testparm`** utility”

2.6. CONFIGURING FILE SHARES ON A SAMBA SERVER

To use Samba as a file server, add shares to the **`/etc/samba/smb.conf`** file of your standalone server or domain member configuration.

You can add shares that uses either:

- POSIX ACLs. See Section 2.6.1, “Setting up a share that uses POSIX ACLs ”.
- Fine-granular Windows ACLs. See Section 2.6.2, “Setting up a share that uses Windows ACLs ”.

Prerequisites

Samba has been set up in one of the following modes:

- Standalone server
- Domain member

2.6.1. Setting up a share that uses POSIX ACLs [2]

As a Linux service, Samba supports shares with POSIX ACLs. They enable you to manage permissions locally on the Samba server using utilities, such as **`chmod`**. If the share is stored on a file system that supports extended attributes, you can define ACLs with multiple users and groups.

NOTE

If you need to use fine-granular Windows ACLs instead, see Section 2.6.2, “Setting up a share that uses Windows ACLs ”.

2.6.1.1. Adding a share that uses POSIX ACLs

This section describes how to create a share named **`example`** that provides the content of the **`/srv/samba/example/`** directory, and uses POSIX ACLs.

Procedure

1. Create the folder if it does not exist. For example:

```
# mkdir -p /srv/samba/example/
```

2. If you run SELinux in **enforcing** mode, set the **samba_share_t** context on the directory:

   ```
   # semanage fcontext -a -t samba_share_t "/srv/samba/example(/.*)?"
   # restorecon -Rv /srv/samba/example/
   ```

3. Set file system ACLs on the directory. See Section 2.6.1, “Setting up a share that uses POSIX ACLs ”.

4. Add the example share to the **/etc/samba/smb.conf** file. For example, to add the share write-enabled:

   ```
   [example]
    path = /srv/samba/example/
    read only = no
   ```

   NOTE

   Regardless of the file system ACLs; if you do not set **read only = no**, Samba shares the directory in read-only mode.

5. Verify the **/etc/samba/smb.conf** file:

   ```
   # testparm
   ```

6. Open the required ports and reload the firewall configuration using the **firewall-cmd** utility:

   ```
   # firewall-cmd --permanent --add-service=samba
   # firewall-cmd --reload
   ```

7. Restart the **smb** service:

   ```
   # systemctl restart smb
   ```

8. Optionally, enable the smb service to start automatically at boot time:

   ```
   # systemctl enable smb
   ```

Additional resources

- Section 2.2, “Verifying the **smb.conf** file by using the **testparm** utility”

2.6.1.2. Setting ACLs on a share that uses POSIX ACLs

This section describes how to set ACLs on a share that uses POSIX ACLs.

Shares that use POSIX ACLs support:

- Standard Linux ACLs. For details, see Section 2.6.1.2.1, “Setting standard Linux ACLs”.

- Extended ACLs. For details, see Section 2.6.1.2.2, “Setting extended ACLs”.

Prerequisites

- A share with POSIX ACLs has been set up according to Section 2.6.1.1, “Adding a share that uses POSIX ACLs”.

2.6.1.2.1. Setting standard Linux ACLs

The standard ACLs on Linux support setting permissions for one owner, one group, and for all other undefined users. You can use the **chown**, **chgrp**, and **chmod** utility to update the ACLs. If you require precise control, then you use the more complex POSIX ACLs, see Section 2.6.1.2.2, “Setting extended ACLs”.

The following procedure sets the owner of the **/srv/samba/example/** directory to the **root** user, grant read and write permissions to the **Domain Users** group, and deny access to all other users.

Procedure

```
# chown root:"Domain Users" /srv/samba/example/
# chmod 2770 /srv/samba/example/
```

NOTE

Enabling the set-group-ID (SGID) bit on a directory automatically sets the default group for all new files and subdirectories to that of the directory group, instead of the usual behavior of setting it to the primary group of the user who created the new directory entry.

Additional resources

- For further details about permissions, see the **chown(1)** and **chmod(1)** man pages.

2.6.1.2.2. Setting extended ACLs

If the file system the shared directory is stored on supports extended ACLs, you can use them to set complex permissions. Extended ACLs can contain permissions for multiple users and groups.

Extended POSIX ACLs enable you to configure complex ACLs with multiple users and groups. However, you can only set the following permissions:

- No access
- Read access
- Write access
- Full control

If you require the fine-granular Windows permissions, such as **Create folder / append data**, configure the share to use Windows ACLs. See Section 2.6.2, “Setting up a share that uses Windows ACLs ”.

The following procedure shows how to enable extended ACLs on a share. Additionally, it contains an example about setting extended ACLs.

Procedure

1. Enable the following parameter in the share's section in the **/etc/samba/smb.conf** file to enable ACL inheritance of extended ACLs:

   ```
   inherit acls = yes
   ```

   For details, see the parameter description in the **smb.conf(5**) man page.

2. Restart the **smb** service:

   ```
   # systemctl restart smb
   ```

3. Set the ACLs on the directory. For example:

   Example 2.3. Setting Extended ACLs

   The following procedure sets read, write, and execute permissions for the **Domain Admins** group, read, and execute permissions for the **Domain Users** group, and deny access to everyone else on the ***/srv/samba/example/*** directory:

   1. Disable auto-granting permissions to the primary group of user accounts:

      ```
      # setfacl -m group::--- /srv/samba/example/
      # setfacl -m default:group::--- /srv/samba/example/
      ```

      The primary group of the directory is additionally mapped to the dynamic **CREATOR GROUP** principal. When you use extended POSIX ACLs on a Samba share, this principal is automatically added and you cannot remove it.

   2. Set the permissions on the directory:

      a. Grant read, write, and execute permissions to the **Domain Admins** group:

      ```
      # setfacl -m group:"DOMAIN\Domain Admins":rwx
      /srv/samba/example/
      ```

      b. Grant read and execute permissions to the **Domain Users** group:

      ```
      # setfacl -m group:"DOMAIN\Domain Users":r-x
      /srv/samba/example/
      ```

      c. Set permissions for the **other** ACL entry to deny access to users that do not match the other ACL entries:

      ```
      # setfacl -R -m other::--- /srv/samba/example/
      ```

      These settings apply only to this directory. In Windows, these ACLs are mapped to the **This folder only** mode.

   3. To enable the permissions set in the previous step to be inherited by new file system objects created in this directory:

      ```
      # setfacl -m default:group:"DOMAIN\Domain Admins":rwx
      /srv/samba/example/
      # setfacl -m default:group:"DOMAIN\Domain Users":r-x
      ```

```
/srv/samba/example/
# setfacl -m default:other::--- /srv/samba/example/
```

With these settings, the **`This folder only`** mode for the principals is now set to **`This folder, subfolders, and files`**.

Samba maps the permissions set in the procedure to the following Windows ACLs:

Principal	Access	Applies to
Domain\Domain Admins	Full control	This folder, subfolders, and files
Domain\Domain Users	Read & execute	This folder, subfolders, and files
`Everyone` [a]	None	This folder, subfolders, and files
owner (*Unix User\owner*) [b]	Full control	This folder only
primary_group (*Unix User\primary_group*) [c]	None	This folder only
`CREATOR OWNER` [d] [e]	Full control	Subfolders and files only
`CREATOR GROUP` [e] [f]	None	Subfolders and files only

[a] Samba maps the permissions for this principal from the **`other`** ACL entry.

[b] Samba maps the owner of the directory to this entry.

[c] Samba maps the primary group of the directory to this entry.

[d] On new file system objects, the creator inherits automatically the permissions of this principal.

[e] Configuring or removing these principals from the ACLs not supported on shares that use POSIX ACLs.

[f] On new file system objects, the creator's primary group inherits automatically the permissions of this principal.

2.6.1.3. Setting permissions on a share that uses POSIX ACLs

Optionally, to limit or grant access to a Samba share, you can set certain parameters in the share's section in the **`/etc/samba/smb.conf`** file.

NOTE

Share-based permissions manage if a user, group, or host is able to access a share. These settings do not affect file system ACLs.

Use share-based settings to restrict access to shares, for example, to deny access from specific hosts.

Prerequisites

- A share with POSIX ACLs has been set up according to Section 2.6.1.1, “Adding a share that uses POSIX ACLs”.

2.6.1.3.1. Configuring user and group-based share access

User and group-based access control enables you to grant or deny access to a share for certain users and groups.

Procedure

1. For example, to enable all members of the **Domain Users** group to access a share while access is denied for the **user** account, add the following parameters to the share’s configuration:

   ```
   valid users = +DOMAIN\"Domain Users"
   invalid users = DOMAIN\user
   ```

   The **invalid users** parameter has a higher priority than the **valid users** parameter. For example, if the **user** account is a member of the **Domain Users** group, access is denied to this account when you use the previous example.

2. Reload the Samba configuration:

   ```
   # smbcontrol all reload-config
   ```

Additional resources

- For further details, see the parameter descriptions in the **smb.conf(5)** man page.

2.6.1.3.2. Configuring host-based share access

Host-based access control enables you to grant or deny access to a share based on client’s host names, IP addresses, or IP range.

The following procedure explains how to enable the **127.0.0.1** IP address, the **192.0.2.0/24** IP range, and the **client1.example.com** host to access a share, and additionally deny access for the **client2.example.com** host:

Procedure

1. Add the following parameters to the configuration of the share in the **/etc/samba/smb.conf** file:

   ```
   hosts allow = 127.0.0.1 192.0.2.0/24 client1.example.com
   hosts deny = client2.example.com
   ```

   The **hosts deny** parameter has a higher priority than **hosts allow**. For example, if **client1.example.com** resolves to an IP address that is listed in the **hosts allow** parameter, access for this host is denied.

2. Reload the Samba configuration:

   ```
   # smbcontrol all reload-config
   ```

Additional resources

- For further details, see the parameter descriptions in the **`smb.conf(5)`** man page.

2.6.2. Setting up a share that uses Windows ACLs [3]

Samba supports setting Windows ACLs on shares and file system object. This enables you to:

- Use the fine-granular Windows ACLs
- Manage share permissions and file system ACLs using Windows

Alternatively, you can configure a share to use POSIX ACLs. For details, see Section 2.6.1, “Setting up a share that uses POSIX ACLs ”.

2.6.2.1. Granting the SeDiskOperatorPrivilege privilege

Only users and groups having the **`SeDiskOperatorPrivilege`** privilege granted can configure permissions on shares that use Windows ACLs.

Procedure

1. For example, to grant the **`SeDiskOperatorPrivilege`** privilege to the ***`DOMAIN`*****`\Domain Admins`** group:

```
# net rpc rights grant "DOMAIN\Domain Admins"
SeDiskOperatorPrivilege \
       -U "DOMAIN\administrator"
Enter DOMAIN\administrator's password:
Successfully granted rights.
```

NOTE

In a domain environment, grant **`SeDiskOperatorPrivilege`** to a domain group. This enables you to centrally manage the privilege by updating a user’s group membership.

2. To list all users and groups having **`SeDiskOperatorPrivilege`** granted:

```
# net rpc rights list privileges SeDiskOperatorPrivilege \
       -U "DOMAIN\administrator"
Enter administrator's password:
SeDiskOperatorPrivilege:
  BUILTIN\Administrators
  DOMAIN\Domain Admins
```

2.6.2.2. Enabling Windows ACL support

To configure shares that support Windows ACLs, you must enable this feature in Samba.

Prerequisites

- A user share is configured on the Samba server.

Procedure

1. To enable it globally for all shares, add the following settings to the **`[global]`** section of the **`/etc/samba/smb.conf`** file:

   ```
   vfs objects = acl_xattr
   map acl inherit = yes
   store dos attributes = yes
   ```

   Alternatively, you can enable Windows ACL support for individual shares, by adding the same parameters to a share's section instead.

2. Restart the **`smb`** service:

   ```
   # systemctl restart smb
   ```

2.6.2.3. Adding a share that uses Windows ACLs

This section describes how to create a share named **`example`**, that shares the content of the **`/srv/samba/example/`** directory, and uses Windows ACLs.

Procedure

1. Create the folder if it does not exists. For example:

   ```
   # mkdir -p /srv/samba/example/
   ```

2. If you run SELinux in **`enforcing`** mode, set the **`samba_share_t`** context on the directory:

   ```
   # semanage fcontext -a -t samba_share_t "/srv/samba/example(/.*)?"
   # restorecon -Rv /srv/samba/example/
   ```

3. Add the example share to the **`/etc/samba/smb.conf`** file. For example, to add the share write-enabled:

   ```
   [example]
    path = /srv/samba/example/
    read only = no
   ```

   NOTE

   Regardless of the file system ACLs; if you do not set **`read only = no`**, Samba shares the directory in read-only mode.

4. If you have not enabled Windows ACL support in the **`[global]`** section for all shares, add the following parameters to the **`[example]`** section to enable this feature for this share:

   ```
   vfs objects = acl_xattr
   map acl inherit = yes
   store dos attributes = yes
   ```

5. Verify the **`/etc/samba/smb.conf`** file:

```
# testparm
```

6. Open the required ports and reload the firewall configuration using the **firewall-cmd** utility:

```
# firewall-cmd --permanent --add-service=samba
# firewall-cmd --reload
```

7. Restart the **smb** service:

```
# systemctl restart smb
```

Additional resources

- Section 2.2, “Verifying the **smb.conf** file by using the **testparm** utility”

2.6.2.4. Managing share permissions and file system ACLs of a share that uses Windows ACLs

To manage share permissions and file system ACLs on a Samba share that uses Windows ACLs, use a Windows applications, such as **Computer Management**. For details, see the Windows documentation. Alternatively, use the **smbcacls** utility to manage ACLs.

NOTE

To modify the file system permissions from Windows, you must use an account that has the **SeDiskOperatorPrivilege** privilege granted.

Additional resources

- Section 2.6.3, “Managing ACLs on an SMB share using smbcacls”
- Section 2.6.2.1, “Granting the SeDiskOperatorPrivilege privilege”

2.6.3. Managing ACLs on an SMB share using smbcacls

The **smbcacls** utility can list, set, and delete ACLs of files and directories stored on an SMB share. You can use **smbcacls** to manage file system ACLs:

- On a local or remote Samba server that uses advanced Windows ACLs or POSIX ACLs
- On Red Hat Enterprise Linux to remotely manage ACLs on a share hosted on Windows

2.6.3.1. Access control entries

Each ACL entry of a file system object contains Access Control Entries (ACE) in the following format:

```
security_principal:access_right/inheritance_information/permissions
```

Example 2.4. Access control entries

If the **AD\Domain Users** group has **Modify** permissions that apply to **This folder, subfolders, and files** on Windows, the ACL contains the following ACE:

```
AD\Domain Users:ALLOWED/OI|CI/CHANGE
```

An ACE contains the following parts:

Security principal

The security principal is the user, group, or SID the permissions in the ACL are applied to.

Access right

Defines if access to an object is granted or denied. The value can be **`ALLOWED`** or **`DENIED`**.

Inheritance information

The following values exist:

Table 2.1. Inheritance settings

Value	Description	Maps to
`OI`	Object Inherit	This folder and files
`CI`	Container Inherit	This folder and subfolders
`IO`	Inherit Only	The ACE does not apply to the current file or directory
`ID`	Inherited	The ACE was inherited from the parent directory

Additionally, the values can be combined as follows:

Table 2.2. Inheritance settings combinations

Value combinations	Maps to the Windows `Applies to` setting
`OI|CI`	This folder, subfolders, and files
`OI|CI|IO`	Subfolders and files only
`CI|IO`	Subfolders only
`OI|IO`	Files only

Permissions

This value can be either a hex value that represents one or more Windows permissions or an **`smbcacls`** alias:

- A hex value that represents one or more Windows permissions.
  The following table displays the advanced Windows permissions and their corresponding value in hex format:

  Table 2.3. Windows permissions and their corresponding `smbcacls` value in hex format

Windows permissions	Hex values
Full control	`0x001F01FF`
Traverse folder / execute file	`0x00100020`
List folder / read data	`0x00100001`
Read attributes	`0x00100080`
Read extended attributes	`0x00100008`
Create files / write data	`0x00100002`
Create folders / append data	`0x00100004`
Write attributes	`0x00100100`
Write extended attributes	`0x00100010`
Delete subfolders and files	`0x00100040`
Delete	`0x00110000`
Read permissions	`0x00120000`
Change permissions	`0x00140000`
Take ownership	`0x00180000`

Multiple permissions can be combined as a single hex value using the bit-wise **`OR`** operation. For details, see Section 2.6.3.3, “ACE mask calculation”.

- An **`smbcacls`** alias. The following table displays the available aliases:

Table 2.4. Existing `smbcacls` aliases and their corresponding Windows permission

`smbcacls` alias	Maps to Windows permission
`R`	Read
`READ`	Read & execute

`smbcacls` alias	Maps to Windows permission
`W`	Special: ∘ Create files / write data ∘ Create folders / append data ∘ Write attributes ∘ Write extended attributes ∘ Read permissions
`D`	Delete
`P`	Change permissions
`O`	Take ownership
`X`	Traverse / execute
`CHANGE`	Modify
`FULL`	Full control

NOTE

You can combine single-letter aliases when you set permissions. For example, you can set **`RD`** to apply the Windows permission **`Read`** and **`Delete`**. However, you can neither combine multiple non-single-letter aliases nor combine aliases and hex values.

2.6.3.2. Displaying ACLs using smbcacls

To display ACLs on an SMB share, use the **`smbcacls`** utility. If you run **`smbcacls`** without any operation parameter, such as **`--add`**, the utility displays the ACLs of a file system object.

Procedure

For example, to list the ACLs of the root directory of the **`//server/example`** share:

```
# smbcacls //server/example / -U "DOMAIN\administrator"
Enter DOMAIN\administrator's password:
REVISION:1
CONTROL:SR|PD|DI|DP
OWNER:AD\Administrators
GROUP:AD\Domain Users
ACL:AD\Administrator:ALLOWED/OI|CI/FULL
ACL:AD\Domain Users:ALLOWED/OI|CI/CHANGE
ACL:AD\Domain Guests:ALLOWED/OI|CI/0x00100021
```

The output of the command displays:

- **REVISION**: The internal Windows NT ACL revision of the security descriptor
- **CONTROL**: Security descriptor control
- **OWNER**: Name or SID of the security descriptor's owner
- **GROUP**: Name or SID of the security descriptor's group
- **ACL** entries. For details, see Section 2.6.3.1, "Access control entries".

2.6.3.3. ACE mask calculation

In most situations, when you add or update an ACE, you use the **smbcacls** aliases listed in Table 2.4, "Existing **smbcacls** aliases and their corresponding Windows permission".

However, if you want to set advanced Windows permissions as listed in Table 2.3, "Windows permissions and their corresponding **smbcacls** value in hex format", you must use the bit-wise **OR** operation to calculate the correct value. You can use the following shell command to calculate the value:

```
# echo $(printf '0x%X' $(( hex_value_1 | hex_value_2 | ... )))
```

Example 2.5. Calculating an ACE Mask

You want to set the following permissions:

- Traverse folder / execute file (0x00100020)
- List folder / read data (0x00100001)
- Read attributes (0x00100080)

To calculate the hex value for the previous permissions, enter:

```
# echo $(printf '0x%X' $(( 0x00100020 | 0x00100001 | 0x00100080 )))
0x1000A1
```

Use the returned value when you set or update an ACE.

2.6.3.4. Adding, updating, and removing an ACL using smbcacls

Depending on the parameter you pass to the **smbcacls** utility, you can add, update, and remove ACLs from a file or directory.

Adding an ACL

To add an ACL to the root of the **//server/example** share that grants **CHANGE** permissions for **This folder, subfolders, and files** to the **AD\Domain Users** group:

```
# smbcacls //server/example / -U "DOMAIN\administrator \
       --add ACL:"AD\Domain Users":ALLOWED/OI|CI/CHANGE
```

Updating an ACL

Updating an ACL is similar to adding a new ACL. You update an ACL by overriding the ACL using the **--modify** parameter with an existing security principal. If **smbcacls** finds the security principal in the ACL list, the utility updates the permissions. Otherwise the command fails with an error:

```
ACL for SID principal_name not found
```

For example, to update the permissions of the **AD\Domain Users** group and set them to **READ** for **This folder, subfolders, and files**:

```
# smbcacls //server/example / -U "DOMAIN\administrator \
       --modify ACL:"AD\Domain Users":ALLOWED/OI|CI/READ
```

Deleting an ACL
To delete an ACL, pass the **--delete** parameter with the exact ACL to the **smbcacls** utility. For example:

```
# smbcacls //server/example / -U "DOMAIN\administrator \
       --delete ACL:"AD\Domain Users":ALLOWED/OI|CI/READ
```

2.6.4. Enabling users to share directories on a Samba server

On a Samba server, you can configure that users can share directories without root permissions.

2.6.4.1. Enabling the user shares feature

Before users can share directories, the administrator must enable user shares in Samba.

For example, to enable only members of the local **example** group to create user shares.

Procedure

1. Create the local **example** group, if it does not exist:

   ```
   # groupadd example
   ```

2. Prepare the directory for Samba to store the user share definitions and set its permissions properly. For example:

   a. Create the directory:

   ```
   # mkdir -p /var/lib/samba/usershares/
   ```

   b. Set write permissions for the **example** group:

   ```
   # chgrp example /var/lib/samba/usershares/
   # chmod 1770 /var/lib/samba/usershares/
   ```

   c. Set the sticky bit to prevent users to rename or delete files stored by other users in this directory.

3. Edit the **/etc/samba/smb.conf** file and add the following to the **[global]** section:

   a. Set the path to the directory you configured to store the user share definitions. For example:

```
usershare path = /var/lib/samba/usershares/
```

b. Set how many user shares Samba allows to be created on this server. For example:

```
usershare max shares = 100
```

If you use the default of **0** for the **usershare max shares** parameter, user shares are disabled.

c. Optionally, set a list of absolute directory paths. For example, to configure that Samba only allows to share subdirectories of the **/data** and **/srv** directory to be shared, set:

```
usershare prefix allow list = /data /srv
```

For a list of further user share-related parameters you can set, see the **USERSHARES** section in the **smb.conf(5)** man page.

4. Verify the **/etc/samba/smb.conf** file:

```
# testparm
```

5. Reload the Samba configuration:

```
# smbcontrol all reload-config
```

Users are now able to create user shares. For details, see Section 2.6.4.2, “Adding a user share”.

Additional resources

- Section 2.2, “Verifying the **smb.conf** file by using the **testparm** utility”

2.6.4.2. Adding a user share

After you configured Samba according to Section 2.6.4.1, “Enabling the user shares feature”, users can share directories on the Samba server without **root** permissions by running the **net usershare add** command.

Synopsis of the **net usershare add** command:

net usershare add share_name path [[comment] | [ACLs]] [guest_ok=y|n]

IMPORTANT

If you set ACLs when you create a user share, you must specify the comment parameter prior to the ACLs. To set an empty comment, use an empty string in double quotes.

Note that users can only enable guest access on a user share, if the administrator set **usershare allow guests = yes** in the **[global]** section in the **/etc/samba/smb.conf** file.

Example 2.6. Adding a user share

A user wants to share the **/srv/samba/** directory on a Samba server. The share should be named **example**, have no comment set, and should be accessible by guest users. Additionally, the share

> permissions should be set to full access for the **AD\Domain Users** group and read permissions for other users. To add this share, run as the user:
>
> ```
> $ net usershare add example /srv/samba/ "" \
>       "AD\Domain Users":F,Everyone:R guest_ok=yes
> ```

2.6.4.3. Updating settings of a user share

To update settings of a user share, override the share by using the **net usershare add** command with the same share name and the new settings. See Section 2.6.4.2, “Adding a user share”.

2.6.4.4. Displaying information about existing user shares

Users can enter the **net usershare info** command on a Samba server to display user shares and their settings.

Prerequisites

- A user share is configured on the Samba server.

Procedure

1. To display all user shares created by any user:

   ```
   $ net usershare info -l
   [share_1]
   path=/srv/samba/
   comment=
   usershare_acl=Everyone:R,host_name\user:F,
   guest_ok=y
   ...
   ```

   To list only shares created by the user who runs the command, omit the **-l** parameter.

2. To display only the information about specific shares, pass the share name or wild cards to the command. For example, to display the information about shares whose name starts with **share_**:

   ```
   $ net usershare info -l share_*
   ```

2.6.4.5. Listing user shares

If you want to list only the available user shares without their settings on a Samba server, use the **net usershare list** command.

Prerequisites

- A user share is configured on the Samba server.

Procedure

1. To list the shares created by any user:

```
$ net usershare list -l
share_1
share_2
...
```

To list only shares created by the user who runs the command, omit the **-l** parameter.

2. To list only specific shares, pass the share name or wild cards to the command. For example, to list only shares whose name starts with **share_**:

   ```
   $ net usershare list -l share_*
   ```

2.6.4.6. Deleting a user share

To delete a user share, use the command **net usershare delete** command as the user who created the share or as the **root** user.

Prerequisites

- A user share is configured on the Samba server.

Procedure

```
$ net usershare delete share_name
```

2.6.5. Enabling guest access to a share

In certain situations, you want to share a directory to which users can connect without authentication. To configure this, enable guest access on a share.

> **WARNING**
>
> Shares that do not require authentication can be a security risk.

If guest access is enabled on a share, Samba maps guest connections to the operating system account set in the **guest account** parameter. Guest users can access files on this share if at least one of the following conditions is satisfied:

- The account is listed in file system ACLs
- The POSIX permissions for **other** users allow it

> **Example 2.7. Guest share permissions**
>
> If you configured Samba to map the guest account to **nobody**, which is the default, the ACLs in the following example:
>
> - Allow guest users to read **file1.txt**

- Allow guest users to read and modify **file2.txt**
- Prevent guest users to read or modify **file3.txt**

```
-rw-r--r--. 1 root      root      1024 1. Sep 10:00 file1.txt
-rw-r-----. 1 nobody    root      1024 1. Sep 10:00 file2.txt
-rw-r-----. 1 root      root      1024 1. Sep 10:00 file3.txt
```

Procedure

1. Edit the **/etc/samba/smb.conf** file:

   a. If this is the first guest share you set up on this server:

      i. Set **map to guest = Bad User** in the **[global]** section:

      ```
      [global]
              ...
              map to guest = Bad User
      ```

      With this setting, Samba rejects login attempts that use an incorrect password unless the user name does not exist. If the specified user name does not exist and guest access is enabled on a share, Samba treats the connection as a guest log in.

      ii. By default, Samba maps the guest account to the **nobody** account on Red Hat Enterprise Linux. Alternatively, you can set a different account. For example:

      ```
      [global]
              ...
              guest account = user_name
      ```

      The account set in this parameter must exist locally on the Samba server. For security reasons, Red Hat recommends using an account that does not have a valid shell assigned.

   b. Add the **guest ok = yes** setting to the **[example]** share section:

   ```
   [example]
           ...
           guest ok = yes
   ```

2. Verify the **/etc/samba/smb.conf** file:

   ```
   # testparm
   ```

3. Reload the Samba configuration:

   ```
   # smbcontrol all reload-config
   ```

Additional resources

- Section 2.2, “Verifying the **smb.conf** file by using the **testparm** utility”

2.7. SETTING UP SAMBA AS A PRINT SERVER [4]

If you set up Samba as a print server, clients in your network can use Samba to print. Additionally, Windows clients can, if configured, download the driver from the Samba server.

2.7.1. Prerequisites

Samba has been set up in one of the following modes:

- Standalone server
- Domain member

2.7.2. The Samba spoolssd service

The Samba **spoolssd** is a service that is integrated into the **smbd** service. Enable **spoolssd** in the Samba configuration to significantly increase the performance on print servers with a high number of jobs or printers.

Without **spoolssd**, Samba forks the **smbd** process and initializes the **printcap** cache for each print job. In case of a large number of printers, the **smbd** service can become unresponsive for multiple seconds while the cache is initialized. The **spoolssd** service enables you to start pre-forked **smbd** processes that are processing print jobs without any delays. The main **spoolssd smbd** process uses a low amount of memory, and forks and terminates child processes.

The following procedure explains how to enable the **spoolssd** service.

Procedure

1. Edit the **[global]** section in the **/etc/samba/smb.conf** file:

   a. Add the following parameters:

```
rpc_server:spoolss = external
rpc_daemon:spoolssd = fork
```

   b. Optionally, you can set the following parameters:

Parameter	Default	Description
spoolssd:prefork_min_children	5	Minimum number of child processes
spoolssd:prefork_max_children	25	Maximum number of child processes
spoolssd:prefork_spawn_rate	5	Samba forks the number of new child processes set in this parameter, up to the value set in **spoolssd:prefork_max_children**, if a new connection is established

Parameter	Default	Description
`spoolssd:prefork_max_allowed_clients`	100	Number of clients, a child process serves
`spoolssd:prefork_child_min_life`	60	Minimum lifetime of a child process in seconds. 60 seconds is the minimum.

2. Verify the **`/etc/samba/smb.conf`** file:

```
# testparm
```

3. Restart the **`smb`** service:

```
# systemctl restart smb
```

After you restarted the service, Samba automatically starts **`smbd`** child processes:

```
# ps axf
...
30903 smbd
30912  \_ smbd
30913      \_ smbd
30914      \_ smbd
30915      \_ smbd
...
```

Additional resources

- Section 2.2, “Verifying the **`smb.conf`** file by using the **`testparm`** utility”

2.7.3. Enabling print server support in Samba

This section explains how to enable the print server support in Samba.

Procedure

1. On the Samba server, set up CUPS and add the printer to the CUPS back end. For details about configuring printers in CUPS; see the documentation provided in the CUPS web interface (https://*print_server_host_name*:631/help) on the print server.

NOTE

Samba can only forward the print jobs to CUPS if CUPS is installed locally on the Samba print server.

2. Edit the **`/etc/samba/smb.conf`** file:

   a. If you want to enable the **`spoolssd`** service, add the following parameters to the **`[global]`** section:

```
rpc_server:spoolss = external
rpc_daemon:spoolssd = fork
```

b. To configure the printing back end, add the **[printers]** section:

```
[printers]
        comment = All Printers
        path = /var/tmp/
        printable = yes
        create mask = 0600
```

IMPORTANT

The **[printers]** share name is hard-coded and cannot be changed.

3. Verify the **/etc/samba/smb.conf** file:

```
# testparm
```

4. Open the required ports and reload the firewall configuration using the **firewall-cmd** utility:

```
# firewall-cmd --permanent --add-service=samba
# firewall-cmd --reload
```

5. Restart the **smb** service:

```
# systemctl restart smb
```

After restarting the service, Samba automatically shares all printers that are configured in the CUPS back end. If you want to manually share only specific printers, see Section 2.7.4, “Manually sharing specific printers”.

Additional resources

- For further details and **spoolssd** parameters you can set, see Section 2.7.2, “The Samba spoolssd service”
- Section 2.2, “Verifying the **smb.conf** file by using the **testparm** utility”

2.7.4. Manually sharing specific printers

If you configured Samba as a print server, by default, Samba shares all printers that are configured in the CUPS back end. The following procedure explains how to share only specific printers.

Prerequisites

- Samba is set up as a print server

Procedure

1. Edit the **/etc/samba/smb.conf** file:

   a. In the **[global]** section, disable automatic printer sharing by setting:

```
load printers = no
```

b. Add a section for each printer you want to share. For example, to share the printer named **example** in the CUPS back end as **Example-Printer** in Samba, add the following section:

```
[Example-Printer]
        path = /var/tmp/
        printable = yes
        printer name = example
```

You do not need individual spool directories for each printer. You can set the same spool directory in the **path** parameter for the printer as you set in the **[printers]** section.

2. Verify the **/etc/samba/smb.conf** file:

```
# testparm
```

3. Reload the Samba configuration:

```
# smbcontrol all reload-config
```

Additional resources

- Section 2.2, “Verifying the **smb.conf** file by using the **testparm** utility”

2.7.5. Setting up automatic printer driver downloads for Windows clients [5]

If you are running a Samba print server for Windows clients, you can upload drivers and preconfigure printers. If a user connects to a printer, Windows automatically downloads and installs the driver locally on the client. The user does not require local administrator permissions for the installation. Additionally, Windows applies preconfigured driver settings, such as the number of trays.

Prerequisites

- Samba is set up as a print server

2.7.5.1. Basic information about printer drivers

This section provides general information about printer drivers.

Supported driver model version
Samba only supports the printer driver model version 3 which is supported in Windows 2000 and later, and Windows Server 2000 and later. Samba does not support the driver model version 4, introduced in Windows 8 and Windows Server 2012. However, these and later Windows versions also support version 3 drivers.

Package-aware drivers
Samba does not support package-aware drivers.

Preparing a printer driver for being uploaded
Before you can upload a driver to a Samba print server:

- Unpack the driver if it is provided in a compressed format.

- Some drivers require to start a setup application that installs the driver locally on a Windows host. In certain situations, the installer extracts the individual files into the operating system’s temporary folder during the setup runs. To use the driver files for uploading:

    a. Start the installer.

    b. Copy the files from the temporary folder to a new location.

    c. Cancel the installation.

Ask your printer manufacturer for drivers that support uploading to a print server.

Providing 32-bit and 64-bit drivers for a printer to a client
To provide the driver for a printer for both 32-bit and 64-bit Windows clients, you must upload a driver with exactly the same name for both architectures. For example, if you are uploading the 32-bit driver named **`Example PostScript`** and the 64-bit driver named **`Example PostScript (v1.0)`**, the names do not match. Consequently, you can only assign one of the drivers to a printer and the driver will not be available for both architectures.

2.7.5.2. Enabling users to upload and preconfigure drivers

To be able to upload and preconfigure printer drivers, a user or a group needs to have the **`SePrintOperatorPrivilege`** privilege granted. A user must be added into the **`printadmin`** group. Red Hat Enterprise Linux automatically creates this group when you install the **`samba`** package. The **`printadmin`** group gets assigned the lowest available dynamic system GID that is lower than 1000.

Procedure

1. For example, to grant the **`SePrintOperatorPrivilege`** privilege to the **`printadmin`** group:

```
# net rpc rights grant "printadmin" SePrintOperatorPrivilege \
       -U "DOMAIN\administrator"
Enter DOMAIN\administrator's password:
Successfully granted rights.
```

NOTE

In a domain environment, grant **`SePrintOperatorPrivilege`** to a domain group. This enables you to centrally manage the privilege by updating a user’s group membership.

2. To list all users and groups having **`SePrintOperatorPrivilege`** granted:

```
# net rpc rights list privileges SePrintOperatorPrivilege \
       -U "DOMAIN\administrator"
Enter administrator's password:
SePrintOperatorPrivilege:
  BUILTIN\Administrators
  DOMAIN\printadmin
```

2.7.5.3. Setting up the print$ share

Windows operating systems download printer drivers from a share named **`print$`** from a print server. This share name is hard-coded in Windows and cannot be changed.

The following procedure explains how to share the **/var/lib/samba/drivers/** directory as **print$**, and enable members of the local **printadmin** group to upload printer drivers.

Procedure

1. Add the **[print$]** section to the **/etc/samba/smb.conf** file:

   ```
   [print$]
           path = /var/lib/samba/drivers/
           read only = no
           write list = @printadmin
           force group = @printadmin
           create mask = 0664
           directory mask = 2775
   ```

   Using these settings:

   Only members of the **printadmin** group can upload printer drivers to the share.

   - The group of new created files and directories will be set to **printadmin**.
   - The permissions of new files will be set to **664**.
   - The permissions of new directories will be set to **2775**.

2. To upload only 64-bit drivers for a printer, include this setting in the **[global]** section in the **/etc/samba/smb.conf** file:

   ```
   spoolss: architecture = Windows x64
   ```

   Without this setting, Windows only displays drivers for which you have uploaded at least the 32-bit version.

3. Verify the **/etc/samba/smb.conf** file:

   ```
   # testparm
   ```

4. Reload the Samba configuration

   ```
   # smbcontrol all reload-config
   ```

5. Create the **printadmin** group if it does not exists:

   ```
   # groupadd printadmin
   ```

6. Grant the **SePrintOperatorPrivilege** privilege to the **printadmin** group.

   ```
   # net rpc rights grant "printadmin" SePrintOperatorPrivilege \
          -U "DOMAIN\administrator"
   Enter DOMAIN\administrator's password:
   Successfully granted rights.
   ```

7. If you run SELinux in **enforcing** mode, set the **samba_share_t** context on the directory:

```
# semanage fcontext -a -t samba_share_t
"/var/lib/samba/drivers(/.*)?"
# restorecon -Rv /var/lib/samba/drivers/
```

8. Set the permissions on the **`/var/lib/samba/drivers/`** directory:

   - If you use POSIX ACLs, set:

     ```
     # chgrp -R "printadmin" /var/lib/samba/drivers/
     # chmod -R 2775 /var/lib/samba/drivers/
     ```

   - If you use Windows ACLs, set:

Principal	Access	Apply to
`CREATOR OWNER`	Full control	Subfolders and files only
`Authenticated Users`	Read & execute, List folder contents, Read	This folder, subfolders, and files
`printadmin`	Full control	This folder, subfolders, and files

     For details about setting ACLs on Windows, see the Windows documentation.

Additional resources

- Section 2.7.5.2, “Enabling users to upload and preconfigure drivers”.
- Section 2.2, “Verifying the **`smb.conf`** file by using the **`testparm`** utility”

2.7.5.4. Creating a GPO to enable clients to trust the Samba print server

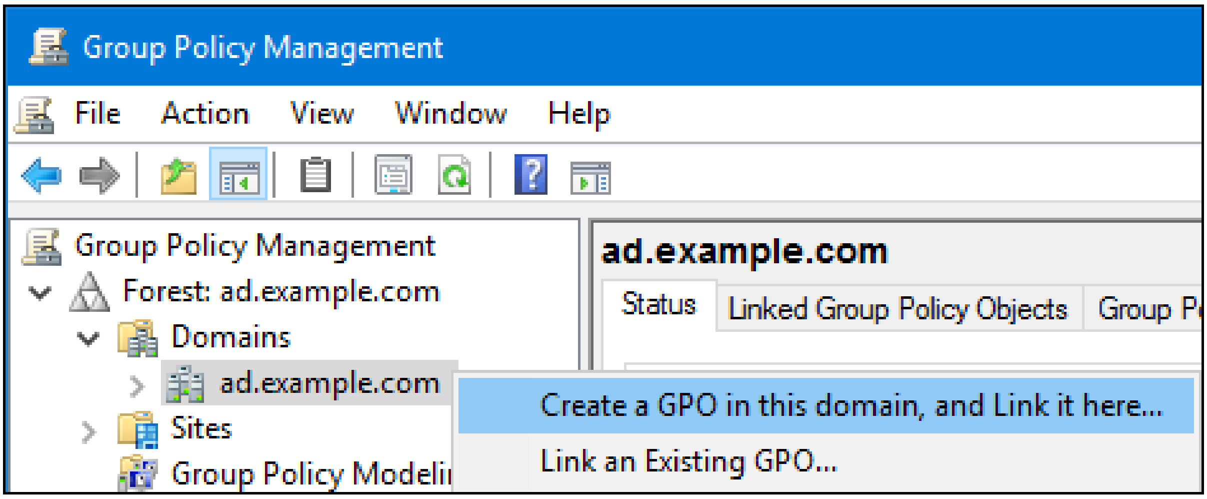
For security reasons, recent Windows operating systems prevent clients from downloading non-package-aware printer drivers from an untrusted server. If your print server is a member in an AD, you can create a Group Policy Object (GPO) in your domain to trust the Samba server.

Prerequisites

- The Samba print server is a member of an AD domain.
- The Windows computer you are using to create the GPO must have the Windows Remote Server Administration Tools (RSAT) installed. For details, see the Windows documentation.

Procedure

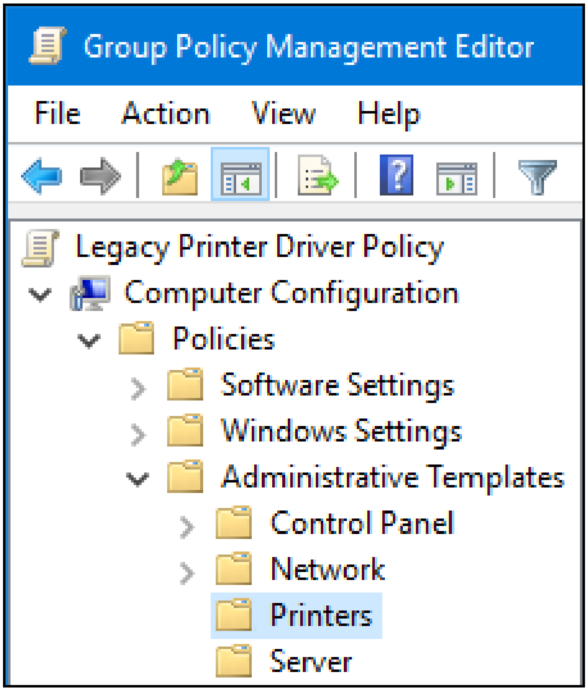
1. Log into a Windows computer using an account that is allowed to edit group policies, such as the AD domain **`Administrator`** user.
2. Open the **`Group Policy Management Console`**.
3. Right-click to your AD domain and select **`Create a GPO in this domain, and Link it here`**.



4. Enter a name for the GPO, such as **`Legacy Printer Driver Policy`** and click **`OK`**. The new GPO will be displayed under the domain entry.

5. Right-click to the newly-created GPO and select **`Edit`** to open the **`Group Policy Management Editor`**.

6. Navigate to **Computer Configuration** → **Policies** → **Administrative Templates** → **Printers**.

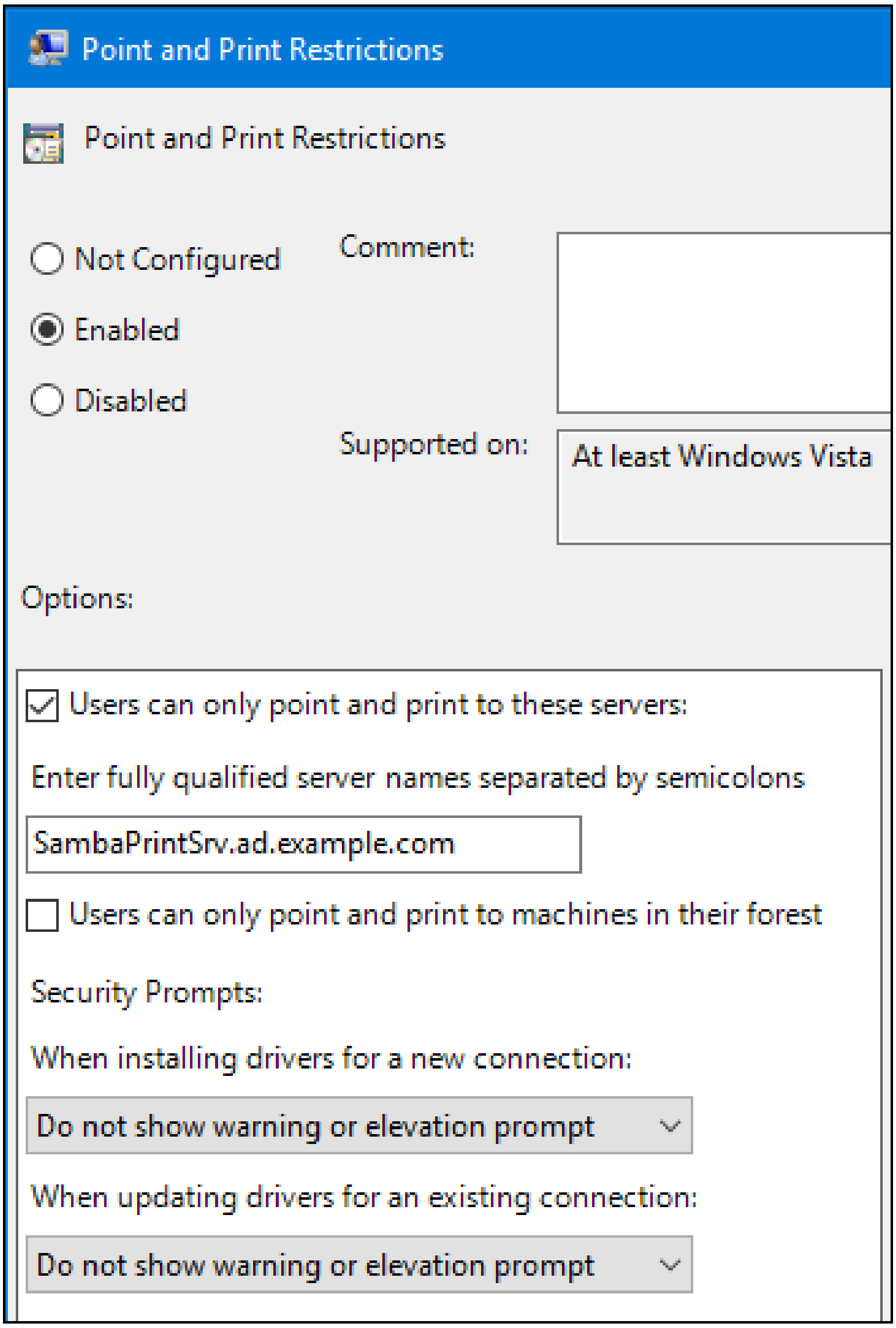


7. On the right side of the window, double-click **`Point and Print Restriction`** to edit the policy:

   a. Enable the policy and set the following options:

      i. Select **`Users can only point and print to these servers`** and enter the fully-qualified domain name (FQDN) of the Samba print server to the field next to this option.

      ii. In both check boxes under **`Security Prompts`**, select **`Do not show warning or elevation prompt`**.



Point and Print Restrictions

○ Not Configured

● Enabled

○ Disabled

Comment:

Supported on: At least Windows Vista

Options:

☑ Users can only point and print to these servers:

Enter fully qualified server names separated by semicolons

SambaPrintSrv.ad.example.com

□ Users can only point and print to machines in their forest

Security Prompts:

When installing drivers for a new connection:

Do not show warning or elevation prompt

When updating drivers for an existing connection:

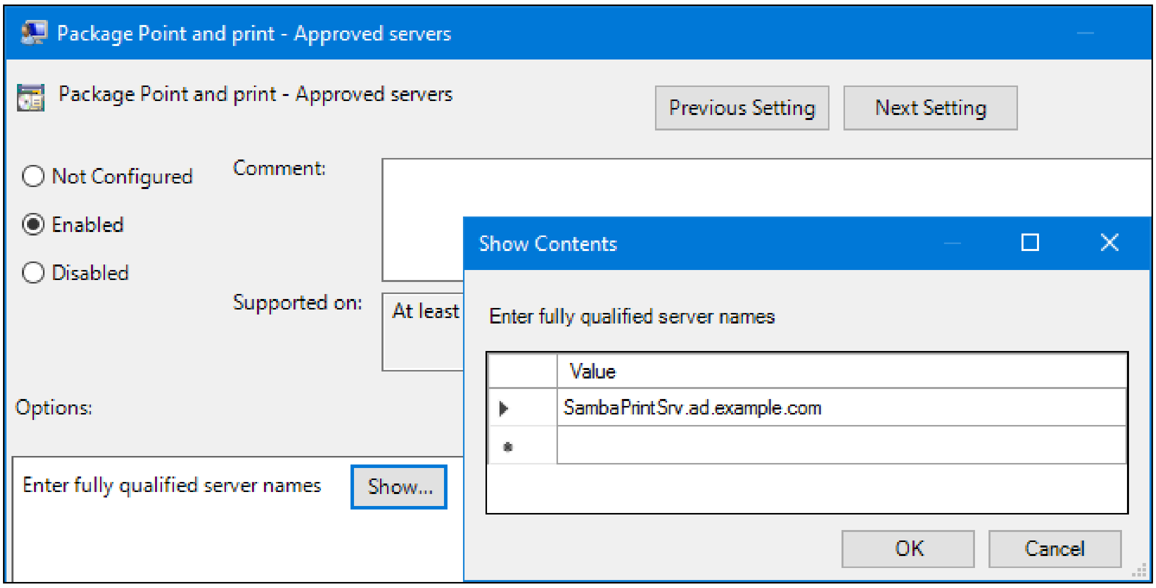
Do not show warning or elevation prompt

b. Click OK.

8. Double-click `Package Point and Print - Approved servers` to edit the policy:

   a. Enable the policy and click the **Show** button.

   b. Enter the FQDN of the Samba print server.



Package Point and print - Approved servers

○ Not Configured
● Enabled
○ Disabled

Enter fully qualified server names

	Value
▶	SambaPrintSrv.ad.example.com
*	

c. Close both the **`Show Contents`** and the policy's properties window by clicking **`OK`**.

9. Close the **`Group Policy Management Editor`**.

10. Close the **`Group Policy Management Console`**.

After the Windows domain members applied the group policy, printer drivers are automatically downloaded from the Samba server when a user connects to a printer.

Additional resources

- For further details about using group policies, see the Windows documentation.

2.7.5.5. Uploading drivers and preconfiguring printers

Use the **`Print Management`** application on a Windows client to upload drivers and preconfigure printers hosted on the Samba print server. For further details, see the Windows documentation.

2.8. TUNING THE PERFORMANCE OF A SAMBA SERVER [6]

This chapter describes what settings can improve the performance of Samba in certain situations, and which settings can have a negative performance impact.

Prerequisites

- Samba is set up as a file or print server

2.8.1. Setting the SMB protocol version

Each new SMB version adds features and improves the performance of the protocol. The recent Windows and Windows Server operating systems always supports the latest protocol version. If Samba also uses the latest protocol version, Windows clients connecting to Samba benefit from the performance improvements. In Samba, the default value of the server max protocol is set to the latest supported stable SMB protocol version.

NOTE

To always have the latest stable SMB protocol version enabled, do not set the **server max protocol** parameter. If you set the parameter manually, you will need to modify the setting with each new version of the SMB protocol, to have the latest protocol version enabled.

The following procedure explains how to use the default value in the **server max protocol** parameter.

Procedure

1. Remove the **server max protocol** parameter from the **[global]** section in the **/etc/samba/smb.conf** file.

2. Reload the Samba configuration

```
# smbcontrol all reload-config
```

2.8.2. Tuning shares with directories that contain a large number of files

Linux supports case-sensitive file names. For this reason, Samba needs to scan directories for uppercase and lowercase file names when searching or accessing a file. You can configure a share to create new files only in lowercase or uppercase, which improves the performance.

Prerequisites

- Samba is configured as a file server

Procedure

1. Rename all files on the share to lowercase.

   NOTE

   Using the settings in this procedure, files with names other than in lowercase will no longer be displayed.

2. Set the following parameters in the share’s section:

```
case sensitive = true
default case = lower
preserve case = no
short preserve case = no
```

   For details about the parameters, see their descriptions in the **smb.conf(5)** man page.

3. Verify the **/etc/samba/smb.conf** file:

```
# testparm
```

4. Reload the Samba configuration:

```
# smbcontrol all reload-config
```

After you applied these settings, the names of all newly created files on this share use lowercase. Because of these settings, Samba no longer needs to scan the directory for uppercase and lowercase, which improves the performance.

Additional resources

- Section 2.2, "Verifying the **smb.conf** file by using the **testparm** utility"

2.8.3. Settings that can have a negative performance impact

By default, the kernel in Red Hat Enterprise Linux is tuned for high network performance. For example, the kernel uses an auto-tuning mechanism for buffer sizes. Setting the **socket options** parameter in the **/etc/samba/smb.conf** file overrides these kernel settings. As a result, setting this parameter decreases the Samba network performance in most cases.

To use the optimized settings from the Kernel, remove the **socket options** parameter from the **[global]** section in the **/etc/samba/smb.conf**.

2.9. FREQUENTLY USED SAMBA COMMAND-LINE UTILITIES

This chapter describes frequently used commands when working with a Samba server.

2.9.1. Using the net utility

The **net** utility enables you to perform several administration tasks on a Samba server. This section describes the most frequently used subcommands of the **net** utility.

Prerequisites

- The **samba-common-tools** package is installed.

2.9.1.1. Using the net ads join and net rpc join commands

Using the **join** subcommand of the **net** utility, you can join Samba to an AD or NT4 domain. To join the domain, you must create the **/etc/samba/smb.conf** file manually, and optionally update additional configurations, such as PAM.

IMPORTANT

Red Hat recommends using the **realm** utility to join a domain. The **realm** utility automatically updates all involved configuration files. For details, see Section 2.5.1, "Joining Samba to a Domain".

Procedure

1. Manually create the **/etc/samba/smb.conf** file with the following settings:

   - For an AD domain member:

     ```
     [global]
     workgroup = domain_name
     security = ads
     passdb backend = tdbsam
     realm = AD_REALM
     ```

- For an NT4 domain member:

  ```
  [global]
  workgroup = domain_name
  security = user
  passdb backend = tdbsam
  ```

2. Add an ID mapping configuration for the `*` default domain and for the domain you want to join to the `[global]` section in the `/etc/samba/smb.conf` file. For details, see Section 2.5.4, “Samba ID mapping”.

3. Verify the `/etc/samba/smb.conf` file:

   ```
   # testparm
   ```

4. Join the domain as the domain administrator:

   - To join an AD domain:

     ```
     # net ads join -U "DOMAIN\administrator"
     ```

   - To join an NT4 domain:

     ```
     # net rpc join -U "DOMAIN\administrator"
     ```

5. Append the `winbind` source to the `passwd` and `group` database entry in the `/etc/nsswitch.conf` file:

   ```
   passwd:     files winbind
   group:      files winbind
   ```

6. Enable and start the `winbind` service:

   ```
   # systemctl enable winbind
   # systemctl start winbind
   ```

7. Optionally, configure PAM using the `authselect` utility.
   For details, see the `authselect(8)` man page.

8. Optionally for AD environments, configure the Kerberos client.
   For details, see the documentation of your Kerberos client.

Additional resources

- Section 2.2, “Verifying the `smb.conf` file by using the `testparm` utility”

2.9.1.2. Using the `net rpc rights` command

In Windows, you can assign privileges to accounts and groups to perform special operations, such as setting ACLs on a share or upload printer drivers. On a Samba server, you can use the `net rpc rights` command to manage privileges.

Listing privileges you can set
To list all available privileges and their owners, use the **`net rpc rights list`** command. For example:

```
net rpc rights list -U "DOMAIN\administrator"
Enter DOMAIN\administrator's password:
     SeMachineAccountPrivilege  Add machines to domain
      SeTakeOwnershipPrivilege  Take ownership of files or other objects
             SeBackupPrivilege  Back up files and directories
            SeRestorePrivilege  Restore files and directories
     SeRemoteShutdownPrivilege  Force shutdown from a remote system
      SePrintOperatorPrivilege  Manage printers
           SeAddUsersPrivilege  Add users and groups to the domain
       SeDiskOperatorPrivilege  Manage disk shares
           SeSecurityPrivilege  System security
```

Granting privileges
To grant a privilege to an account or group, use the **`net rpc rights grant`** command.

For example, grant the **`SePrintOperatorPrivilege`** privilege to the ***`DOMAIN`*****`\printadmin`** group:

```
# net rpc rights grant "DOMAIN\printadmin" SePrintOperatorPrivilege \
       -U "DOMAIN\administrator"
Enter DOMAIN\administrator's password:
Successfully granted rights.
```

Revoking privileges
To revoke a privilege from an account or group, use the **`net rpc rights revoke`** command.

For example, to revoke the **`SePrintOperatorPrivilege`** privilege from the ***`DOMAIN`*****`\printadmin`** group:

```
# net rpc rights remoke "DOMAIN\printadmin" SePrintOperatorPrivilege \
       -U "DOMAIN\administrator"
Enter DOMAIN\administrator's password:
Successfully revoked rights.
```

2.9.1.3. Using the `net rpc share` command

The **`net rpc share`** command provides the capability to list, add, and remove shares on a local or remote Samba or Windows server.

Listing shares
To list the shares on an SMB server, use the **`net rpc share list`** command. Optionally, pass the **`-S`** ***`server_name`*** parameter to the command to list the shares of a remote server. For example:

```
# net rpc share list -U "DOMAIN\administrator" -S server_name
Enter DOMAIN\administrator's password:
IPC$
share_1
share_2
...
```

NOTE

Shares hosted on a Samba server that have **`browseable = no`** set in their section in the **`/etc/samba/smb.conf`** file are not displayed in the output.

Adding a share

The **`net rpc share add`** command enables you to add a share to an SMB server.

For example, to add a share named **`example`** on a remote Windows server that shares the **`C:\example\`** directory:

```
# net rpc share add example="C:\example" -U "DOMAIN\administrator" -S
server_name
```

NOTE

You must omit the trailing backslash in the path when specifying a Windows directory name.

To use the command to add a share to a Samba server:

- The user specified in the **`-U`** parameter must have the **`SeDiskOperatorPrivilege`** privilege granted on the destination server.
- You must write a script that adds a share section to the **`/etc/samba/smb.conf`** file and reloads Samba. The script must be set in the **`add share command`** parameter in the **`[global]`** section in **`/etc/samba/smb.conf`**. For further details, see the **`add share command`** description in the **`smb.conf(5)`** man page.

Removing a share

The **`net rpc share delete`** command enables you to remove a share from an SMB server.

For example, to remove the share named example from a remote Windows server:

```
# net rpc share delete example -U "DOMAIN\administrator" -S server_name
```

To use the command to remove a share from a Samba server:

- The user specified in the **`-U`** parameter must have the **`SeDiskOperatorPrivilege`** privilege granted.
- You must write a script that removes the share’s section from the **`/etc/samba/smb.conf`** file and reloads Samba. The script must be set in the **`delete share command`** parameter in the **`[global]`** section in **`/etc/samba/smb.conf`**. For further details, see the **`delete share command`** description in the **`smb.conf(5)`** man page.

2.9.1.4. Using the `net user` command

The **`net user`** command enables you to perform the following actions on an AD DC or NT4 PDC:

- List all user accounts
- Add users

- Remove Users

> **NOTE**
>
> Specifying a connection method, such as **`ads`** for AD domains or **`rpc`** for NT4 domains, is only required when you list domain user accounts. Other user-related subcommands can auto-detect the connection method.

Pass the **`-U`** ***`user_name`*** parameter to the command to specify a user that is allowed to perform the requested action.

Listing domain user accounts
To list all users in an AD domain:

```
# net ads user -U "DOMAIN\administrator"
```

To list all users in an NT4 domain:

```
# net rpc user -U "DOMAIN\administrator"
```

Adding a user account to the domain
On a Samba domain member, you can use the **`net user add`** command to add a user account to the domain.

For example, add the **`user`** account to the domain:

1. Add the account:

   ```
   # net user add user password -U "DOMAIN\administrator"
   User user added
   ```

2. Optionally, use the remote procedure call (RPC) shell to enable the account on the AD DC or NT4 PDC. For example:

   ```
   # net rpc shell -U DOMAIN\administrator -S DC_or_PDC_name
   Talking to domain DOMAIN (S-1-5-21-1424831554-512457234-5642315751)

   net rpc> user edit disabled user: no
   Set user's disabled flag from [yes] to [no]

   net rpc> exit
   ```

Deleting a user account from the domain
On a Samba domain member, you can use the **`net user delete`** command to remove a user account from the domain.

For example, to remove the **`user`** account from the domain:

```
# net user delete user -U "DOMAIN\administrator"
User user deleted
```

2.9.1.5. Using the `net usershare` command

On a Samba server, you can configure that users can share directories without root permissions.

2.9.1.5.1. Enabling the user shares feature

Before users can share directories, the administrator must enable user shares in Samba.

For example, to enable only members of the local **`example`** group to create user shares.

Procedure

1. Create the local **`example`** group, if it does not exist:

   ```
   # groupadd example
   ```

2. Prepare the directory for Samba to store the user share definitions and set its permissions properly. For example:

   a. Create the directory:

   ```
   # mkdir -p /var/lib/samba/usershares/
   ```

   b. Set write permissions for the **`example`** group:

   ```
   # chgrp example /var/lib/samba/usershares/
   # chmod 1770 /var/lib/samba/usershares/
   ```

   c. Set the sticky bit to prevent users to rename or delete files stored by other users in this directory.

3. Edit the **`/etc/samba/smb.conf`** file and add the following to the **`[global]`** section:

   a. Set the path to the directory you configured to store the user share definitions. For example:

   ```
   usershare path = /var/lib/samba/usershares/
   ```

   b. Set how many user shares Samba allows to be created on this server. For example:

   ```
   usershare max shares = 100
   ```

   If you use the default of **`0`** for the **`usershare max shares`** parameter, user shares are disabled.

   c. Optionally, set a list of absolute directory paths. For example, to configure that Samba only allows to share subdirectories of the **`/data`** and **`/srv`** directory to be shared, set:

   ```
   usershare prefix allow list = /data /srv
   ```

   For a list of further user share-related parameters you can set, see the **USERSHARES** section in the **`smb.conf(5)`** man page.

4. Verify the **`/etc/samba/smb.conf`** file:

   ```
   # testparm
   ```

5. Reload the Samba configuration:

```
# smbcontrol all reload-config
```

Users are now able to create user shares. For details, see Section 2.6.4.2, “Adding a user share”.

Additional resources

- Section 2.2, “Verifying the **smb.conf** file by using the **testparm** utility”

2.9.1.5.2. Adding a user share

After you configured Samba according to Section 2.6.4.1, “Enabling the user shares feature”, users can share directories on the Samba server without **root** permissions by running the **net usershare add** command.

Synopsis of the **net usershare add** command:

net usershare add share_name path [[comment] | [ACLs]] [guest_ok=y|n]

IMPORTANT

If you set ACLs when you create a user share, you must specify the comment parameter prior to the ACLs. To set an empty comment, use an empty string in double quotes.

Note that users can only enable guest access on a user share, if the administrator set **usershare allow guests = yes** in the **[global]** section in the **/etc/samba/smb.conf** file.

Example 2.8. Adding a user share

A user wants to share the **/srv/samba/** directory on a Samba server. The share should be named **example**, have no comment set, and should be accessible by guest users. Additionally, the share permissions should be set to full access for the **AD\Domain Users** group and read permissions for other users. To add this share, run as the user:

```
$ net usershare add example /srv/samba/ "" \
      "AD\Domain Users":F,Everyone:R guest_ok=yes
```

2.9.1.5.3. Updating settings of a user share

To update settings of a user share, override the share by using the **net usershare add** command with the same share name and the new settings. See Section 2.6.4.2, “Adding a user share”.

2.9.1.5.4. Displaying information about existing user shares

Users can enter the **net usershare info** command on a Samba server to display user shares and their settings.

Prerequisites

- A user share is configured on the Samba server.

Procedure

1. To display all user shares created by any user:

   ```
   $ net usershare info -l
   [share_1]
   path=/srv/samba/
   comment=
   usershare_acl=Everyone:R,host_name\user:F,
   guest_ok=y
   ...
   ```

   To list only shares created by the user who runs the command, omit the **`-l`** parameter.

2. To display only the information about specific shares, pass the share name or wild cards to the command. For example, to display the information about shares whose name starts with **`share_`**:

   ```
   $ net usershare info -l share_*
   ```

2.9.1.5.5. Listing user shares

If you want to list only the available user shares without their settings on a Samba server, use the **`net usershare list`** command.

Prerequisites

- A user share is configured on the Samba server.

Procedure

1. To list the shares created by any user:

   ```
   $ net usershare list -l
   share_1
   share_2
   ...
   ```

   To list only shares created by the user who runs the command, omit the **`-l`** parameter.

2. To list only specific shares, pass the share name or wild cards to the command. For example, to list only shares whose name starts with **`share_`**:

   ```
   $ net usershare list -l share_*
   ```

2.9.1.5.6. Deleting a user share

To delete a user share, use the command **`net usershare delete`** command as the user who created the share or as the **`root`** user.

Prerequisites

- A user share is configured on the Samba server.

Procedure

```
$ net usershare delete share_name
```

2.9.2. Using the rpcclient utility

The **rpcclient** utility enables you to manually execute client-side Microsoft Remote Procedure Call (MS-RPC) functions on a local or remote SMB server. However, most of the features are integrated into separate utilities provided by Samba. Use **rpcclient** only for testing MS-PRC functions.

Prerequisites

- The **samba-client** package is installed.

Examples

For example, you can use the **rpcclient** utility to:

- Manage the printer Spool Subsystem (SPOOLSS).

  Example 2.9. Assigning a Driver to a Printer

  ```
  # rpcclient server_name -U "DOMAIN\administrator" \
         -c 'setdriver "printer_name" "driver_name"'
  Enter DOMAIN\administrators password:
  Successfully set printer_name to driver driver_name.
  ```

- Retrieve information about an SMB server.

  Example 2.10. Listing all File Shares and Shared Printers

  ```
  # rpcclient server_name -U "DOMAIN\administrator" -c
  'netshareenum'
  Enter DOMAIN\administrators password:
  netname: Example_Share
   remark:
   path:   C:\srv\samba\example_share\
   password:
  netname: Example_Printer
   remark:
   path:   C:\var\spool\samba\
   password:
  ```

- Perform actions using the Security Account Manager Remote (SAMR) protocol.

  Example 2.11. Listing Users on an SMB Server

  ```
  # rpcclient server_name -U "DOMAIN\administrator" -c
  'enumdomusers'
  Enter DOMAIN\administrators password:
  user:[user1] rid:[0x3e8]
  user:[user2] rid:[0x3e9]
  ```

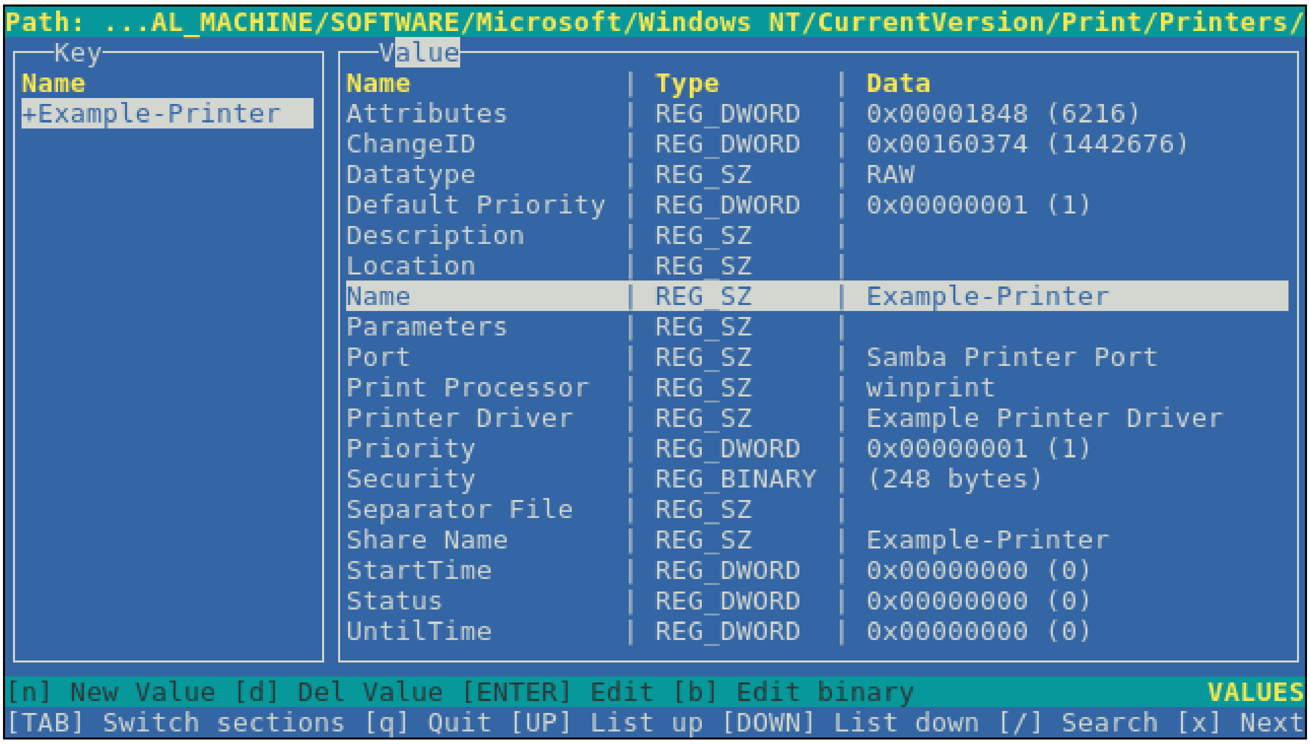
  If you run the command against a standalone server or a domain member, it lists the users in the local database. Running the command against an AD DC or NT4 PDC lists the domain users.

Additional resources

For a complete list of supported subcommands, see the **`COMMANDS`** section in the **`rpcclient(1)`** man page.

2.9.3. Using the samba-regedit application

Certain settings, such as printer configurations, are stored in the registry on the Samba server. You can use the ncurses-based **`samba-regedit`** application to edit the registry of a Samba server.



```
Path: ...AL_MACHINE/SOFTWARE/Microsoft/Windows NT/CurrentVersion/Print/Printers/
 ┌Key──────────────┐┌Value──────────────────────────────────────────────────────┐
 │Name             ││Name              | Type         | Data                   │
 │+Example-Printer ││Attributes        | REG_DWORD    | 0x00001848 (6216)      │
 │                 ││ChangeID          | REG_DWORD    | 0x00160374 (1442676)   │
 │                 ││Datatype          | REG_SZ       | RAW                    │
 │                 ││Default Priority  | REG_DWORD    | 0x00000001 (1)         │
 │                 ││Description       | REG_SZ       |                        │
 │                 ││Location          | REG_SZ       |                        │
 │                 ││Name              | REG_SZ       | Example-Printer        │
 │                 ││Parameters        | REG_SZ       |                        │
 │                 ││Port              | REG_SZ       | Samba Printer Port     │
 │                 ││Print Processor   | REG_SZ       | winprint               │
 │                 ││Printer Driver    | REG_SZ       | Example Printer Driver │
 │                 ││Priority          | REG_DWORD    | 0x00000001 (1)         │
 │                 ││Security          | REG_BINARY   | (248 bytes)            │
 │                 ││Separator File    | REG_SZ       |                        │
 │                 ││Share Name        | REG_SZ       | Example-Printer        │
 │                 ││StartTime         | REG_DWORD    | 0x00000000 (0)         │
 │                 ││Status            | REG_DWORD    | 0x00000000 (0)         │
 │                 ││UntilTime         | REG_DWORD    | 0x00000000 (0)         │
 └─────────────────┘└───────────────────────────────────────────────────────────┘
[n] New Value [d] Del Value [ENTER] Edit [b] Edit binary                  VALUES
[TAB] Switch sections [q] Quit [UP] List up [DOWN] List down [/] Search [x] Next
```

Prerequisites

- The **`samba-client`** package is installed.

Procedure

To start the application, enter:

```
# samba-regedit
```

Use the following keys:

- Cursor up and cursor down: Navigate through the registry tree and the values.

- **`Enter`**: Opens a key or edits a value.

- **`Tab`**: Switches between the **`Key`** and **`Value`** pane.

- **`Ctrl`**+**`C`**: Closes the application.

2.9.4. Using the `smbcacls` utility

The **`smbcacls`** utility can list, set, and delete ACLs of files and directories stored on an SMB share. You can use **`smbcacls`** to manage file system ACLs:

- On a local or remote Samba server that uses advanced Windows ACLs or POSIX ACLs
- On Red Hat Enterprise Linux to remotely manage ACLs on a share hosted on Windows

2.9.4.1. Access control entries

Each ACL entry of a file system object contains Access Control Entries (ACE) in the following format:

```
security_principal:access_right/inheritance_information/permissions
```

Example 2.12. Access control entries

If the **AD\Domain Users** group has **Modify** permissions that apply to **This folder, subfolders, and files** on Windows, the ACL contains the following ACE:

```
AD\Domain Users:ALLOWED/OI|CI/CHANGE
```

An ACE contains the following parts:

Security principal

The security principal is the user, group, or SID the permissions in the ACL are applied to.

Access right

Defines if access to an object is granted or denied. The value can be **ALLOWED** or **DENIED**.

Inheritance information

The following values exist:

Table 2.5. Inheritance settings

Value	Description	Maps to
OI	Object Inherit	This folder and files
CI	Container Inherit	This folder and subfolders
IO	Inherit Only	The ACE does not apply to the current file or directory
ID	Inherited	The ACE was inherited from the parent directory

Additionally, the values can be combined as follows:

Table 2.6. Inheritance settings combinations

Value combinations	Maps to the Windows `Applies to` setting
OI|CI	This folder, subfolders, and files
OI|CI|IO	Subfolders and files only

Value combinations	Maps to the Windows `Applies to` setting
`CI|IO`	Subfolders only
`OI|IO`	Files only

Permissions

This value can be either a hex value that represents one or more Windows permissions or an **`smbcacls`** alias:

- A hex value that represents one or more Windows permissions.
  The following table displays the advanced Windows permissions and their corresponding value in hex format:

Table 2.7. Windows permissions and their corresponding `smbcacls` value in hex format

Windows permissions	Hex values
Full control	**`0x001F01FF`**
Traverse folder / execute file	**`0x00100020`**
List folder / read data	**`0x00100001`**
Read attributes	**`0x00100080`**
Read extended attributes	**`0x00100008`**
Create files / write data	**`0x00100002`**
Create folders / append data	**`0x00100004`**
Write attributes	**`0x00100100`**
Write extended attributes	**`0x00100010`**
Delete subfolders and files	**`0x00100040`**
Delete	**`0x00110000`**
Read permissions	**`0x00120000`**
Change permissions	**`0x00140000`**
Take ownership	**`0x00180000`**

Multiple permissions can be combined as a single hex value using the bit-wise **`OR`** operation. For details, see Section 2.6.3.3, “ACE mask calculation”.

- An **`smbcacls`** alias. The following table displays the available aliases:

Table 2.8. Existing `smbcacls` aliases and their corresponding Windows permission

`smbcacls` alias	Maps to Windows permission
`R`	Read
`READ`	Read & execute
`W`	Special: ◦ Create files / write data ◦ Create folders / append data ◦ Write attributes ◦ Write extended attributes ◦ Read permissions
`D`	Delete
`P`	Change permissions
`O`	Take ownership
`X`	Traverse / execute
`CHANGE`	Modify
`FULL`	Full control

NOTE

You can combine single-letter aliases when you set permissions. For example, you can set **`RD`** to apply the Windows permission **`Read`** and **`Delete`**. However, you can neither combine multiple non-single-letter aliases nor combine aliases and hex values.

2.9.4.2. Displaying ACLs using smbcacls

To display ACLs on an SMB share, use the **`smbcacls`** utility. If you run **`smbcacls`** without any operation parameter, such as **`--add`**, the utility displays the ACLs of a file system object.

Procedure

For example, to list the ACLs of the root directory of the **//server/example** share:

```
# smbcacls //server/example / -U "DOMAIN\administrator"
Enter DOMAIN\administrator's password:
REVISION:1
CONTROL:SR|PD|DI|DP
OWNER:AD\Administrators
GROUP:AD\Domain Users
ACL:AD\Administrator:ALLOWED/OI|CI/FULL
ACL:AD\Domain Users:ALLOWED/OI|CI/CHANGE
ACL:AD\Domain Guests:ALLOWED/OI|CI/0x00100021
```

The output of the command displays:

- **REVISION**: The internal Windows NT ACL revision of the security descriptor
- **CONTROL**: Security descriptor control
- **OWNER**: Name or SID of the security descriptor's owner
- **GROUP**: Name or SID of the security descriptor's group
- **ACL** entries. For details, see Section 2.6.3.1, "Access control entries".

2.9.4.3. ACE mask calculation

In most situations, when you add or update an ACE, you use the **smbcacls** aliases listed in Table 2.4, "Existing **smbcacls** aliases and their corresponding Windows permission".

However, if you want to set advanced Windows permissions as listed in Table 2.3, "Windows permissions and their corresponding **smbcacls** value in hex format", you must use the bit-wise **OR** operation to calculate the correct value. You can use the following shell command to calculate the value:

```
# echo $(printf '0x%X' $(( hex_value_1 | hex_value_2 | ... )))
```

Example 2.13. Calculating an ACE Mask

You want to set the following permissions:

- Traverse folder / execute file (0x00100020)
- List folder / read data (0x00100001)
- Read attributes (0x00100080)

To calculate the hex value for the previous permissions, enter:

```
# echo $(printf '0x%X' $(( 0x00100020 | 0x00100001 | 0x00100080 )))
0x1000A1
```

Use the returned value when you set or update an ACE.

2.9.4.4. Adding, updating, and removing an ACL using smbcacls

Depending on the parameter you pass to the **smbcacls** utility, you can add, update, and remove ACLs from a file or directory.

Adding an ACL

To add an ACL to the root of the **//server/example** share that grants **CHANGE** permissions for **This folder, subfolders, and files** to the **AD\Domain Users** group:

```
# smbcacls //server/example / -U "DOMAIN\administrator \
       --add ACL:"AD\Domain Users":ALLOWED/OI|CI/CHANGE
```

Updating an ACL

Updating an ACL is similar to adding a new ACL. You update an ACL by overriding the ACL using the **--modify** parameter with an existing security principal. If **smbcacls** finds the security principal in the ACL list, the utility updates the permissions. Otherwise the command fails with an error:

```
ACL for SID principal_name not found
```

For example, to update the permissions of the **AD\Domain Users** group and set them to **READ** for **This folder, subfolders, and files**:

```
# smbcacls //server/example / -U "DOMAIN\administrator \
       --modify ACL:"AD\Domain Users":ALLOWED/OI|CI/READ
```

Deleting an ACL

To delete an ACL, pass the **--delete** parameter with the exact ACL to the **smbcacls** utility. For example:

```
# smbcacls //server/example / -U "DOMAIN\administrator \
       --delete ACL:"AD\Domain Users":ALLOWED/OI|CI/READ
```

2.9.5. Using the smbclient utility

The smbclient utility enables you to access file shares on an SMB server, similarly to a command-line FTP client. You can use it, for example, to upload and download files to and from a share.

2.9.5.1. Prerequisites

- The **samba-client** package is installed.

2.9.5.2. How the smbclient interactive mode works

For example, to authenticate to the **example** share hosted on **server** using the ***DOMAIN\user*** account:

```
# smbclient -U "DOMAIN\user" //server/example
Enter domain\user's password:
Try "help" to get a list of possible commands.
smb: \>
```

After **smbclient** connected successfully to the share, the utility enters the interactive mode and shows the following prompt:

```
smb: \>
```

To display all available commands in the interactive shell, enter:

```
smb: \> help
```

To display the help for a specific command, enter:

```
smb: \> help command_name
```

Additional resources

- For further details and descriptions of the commands available in the interactive shell, see the **`smbclient(1)`** man page.

2.9.5.3. Using `smbclient` in interactive mode

If you use **`smbclient`** without the **`-c`** parameter, the utility enters the interactive mode. The following procedure shows how to connect to an SMB share and download a file from a subdirectory.

Procedure

1. Connect to the share:

   ```
   # smbclient -U "DOMAIN\user_name" //server_name/share_name
   ```

2. Change into the **`/example/`** directory:

   ```
   smb: \> cd /example/
   ```

3. List the files in the directory:

   ```
   smb: \example\> ls
     .                    D         0  Thu Nov 1 10:00:00 2018
     ..                   D         0  Thu Nov 1 10:00:00 2018
     example.txt          N   1048576  Thu Nov 1 10:00:00 2018

            9950208 blocks of size 1024. 8247144 blocks available
   ```

4. Download the **`example.txt`** file:

   ```
   smb: \example\> get example.txt
   getting file \directory\subdirectory\example.txt of size 1048576 as
   example.txt (511975,0 KiloBytes/sec) (average 170666,7
   KiloBytes/sec)
   ```

5. Disconnect from the share:

   ```
   smb: \example\> exit
   ```

2.9.5.4. Using `smbclient` in scripting mode

If you pass the **-c** parameter to **smbclient**, you can automatically execute the commands on the remote SMB share. This enables you to use **smbclient** in scripts.

The following procedure shows how to connect to an SMB share and download a file from a subdirectory.

Procedure

```
# smbclient -U DOMAIN\user_name //server_name/share_name \
      -c "cd /example/ ; get example.txt ; exit"
```

2.9.6. Using the smbcontrol utility

The **smbcontrol** utility enables you to send command messages to the **smbd**, **nmbd**, **winbindd**, or all of these services. These control messages instruct the service, for example, to reload its configuration.

The procedure in this section shows how to to reload the configuration of the **smbd**, **nmbd**, **winbindd** services by sending the **reload-config** message type to the **all** destination.

Prerequisites

- The **samba-common-tools** package is installed.

Procedure

```
# smbcontrol all reload-config
```

Additional resources
For further details and a list of available command message types, see the **smbcontrol(1)** man page.

2.9.7. Using the smbpasswd utility

The **smbpasswd** utility manages user accounts and passwords in the local Samba database.

Prerequisites

- The **samba-common-tools** package is installed.

Procedure

1. If you run the command as a user, **smbpasswd** changes the Samba password of the user who run the command. For example:

   ```
   [user@server ~]$ smbpasswd
   New SMB password: password
   Retype new SMB password: password
   ```

2. If you run **smbpasswd** as the **root** user, you can use the utility, for example, to:

   - Create a new user:

     ```
     [root@server ~]# smbpasswd -a user_name
     New SMB password: password
     Retype new SMB password: password
     Added user user_name.
     ```

> **NOTE**
>
> Before you can add a user to the Samba database, you must create the account in the local operating system. See the Adding a new user section in the Configuring and managing system administration guide.

- Enable a Samba user:

  ```
  [root@server ~]# smbpasswd -e user_name
  Enabled user user_name.
  ```

- Disable a Samba user:

  ```
  [root@server ~]# smbpasswd -x user_name
  Disabled user ser_name
  ```

- Delete a user:

  ```
  [root@server ~]# smbpasswd -x user_name
  Deleted user user_name.
  ```

Additional resources
For further details, see the **`smbpasswd(8)`** man page.

2.9.8. Using the `smbstatus` utility

The **`smbstatus`** utility reports on:

- Connections per PID of each **smbd** daemon to the Samba server. This report includes the user name, primary group, SMB protocol version, encryption, and signing information.

- Connections per Samba share. This report includes the PID of the **smbd** daemon, the IP of the connecting machine, the time stamp when the connection was established, encryption, and signing information.

- A list of locked files. The report entries include further details, such as opportunistic lock (oplock) types

Prerequisites

- The **samba** package is installed.

- The **smbd** service is running.

Procedure

```
# smbstatus

Samba version 4.7.1
PID  Username              Group                Machine
Protocol Version  Encryption  Signing
----------------------------------------------------------------------
```

```
-------------------------------------------------------
963  DOMAIN\administrator  DOMAIN\domain users  client-pc
(ipv4:192.0.2.1:57786)  SMB3_02           -           AES-128-CMAC

Service  pid  Machine    Connected at                 Encryption
Signing:
-----------------------------------------------------------------------
---------
example  969  192.0.2.1  Thu Nov  1 10:00:00 2018 CEST  -          AES-
128-CMAC

Locked files:
Pid  Uid    DenyMode   Access    R/W     Oplock      SharePath
Name      Time
-----------------------------------------------------------------------
-------------------------------------
969  10000  DENY_WRITE 0x120089  RDONLY  LEASE(RWH)  /srv/samba/example
file.txt  Thu Nov  1 10:00:00 2018
```

Additional resources
For further details, see the **smbstatus(1)** man page.

2.9.9. Using the smbtar utility

The **smbtar** utility backs up the content of an SMB share or a subdirectory of it and stores the content in a **tar** archive. Alternatively, you can write the content to a tape device.

The procedure in this section describes how to back up the content of the **demo** directory on the ***//server/example/*** share and store the content in the ***/root/example.tar*** archive.

Prerequisites

- The **samba-client** package is installed.

Procedure

```
# smbtar -s server -x example -u user_name -p password -t
/root/example.tar
```

Additional resources
For further details, see the **smbtar(1)** man page.

2.9.10. Using the testparm utility

The **testparm** utility verifies that the Samba configuration in the **/etc/samba/smb.conf** file is correct. The utility detects invalid parameters and values, but also incorrect settings, such as for ID mapping. If **testparm** reports no problem, the Samba services will successfully load the **/etc/samba/smb.conf** file. Note that **testparm** cannot verify that the configured services will be available or work as expected.

IMPORTANT

Red Hat recommends that you verify the **/etc/samba/smb.conf** file by using **testparm** after each modification of this file.

Procedure

1. Run the **`testparm`** utility as the **`root`** user:

    ```
    # testparm
    Load smb config files from /etc/samba/smb.conf
    rlimit_max: increasing rlimit_max (1024) to minimum Windows limit
    (16384)
    Unknown parameter encountered: "log levell"
    Processing section "[example_share]"
    Loaded services file OK.
    ERROR: The idmap range for the domain * (tdb) overlaps with the
    range of DOMAIN (ad)!

    Server role: ROLE_DOMAIN_MEMBER

    Press enter to see a dump of your service definitions

    # Global parameters
    [global]
     ...

    [example_share]
     ...
    ```

    The previous example output reports a non-existent parameter and an incorrect ID mapping configuration.

2. If **`testparm`** reports incorrect parameters, values, or other errors in the configuration, fix the problem and run the utility again.

2.9.11. Using the wbinfo utility

The **`wbinfo`** utility queries and returns information created and used by the **`winbindd`** service.

Prerequisites

- The **`samba-winbind-clients`** package is installed.

Procedure

You can use **`wbinfo`**, for example, to:

- List domain users:

    ```
    # wbinfo -u
    AD\administrator
    AD\guest
    ...
    ```

- List domain groups:

    ```
    # wbinfo -g
    AD\domain computers
    AD\domain admins
    ```

```
AD\domain users
...
```

- Display the SID of a user:

```
# wbinfo --name-to-sid="AD\administrator"
S-1-5-21-1762709870-351891212-3141221786-500 SID_USER (1)
```

- Display information about domains and trusts:

```
# wbinfo --trusted-domains --verbose
Domain Name   DNS Domain             Trust Type  Transitive  In
Out
BUILTIN                              None        Yes         Yes
Yes
server                               None        Yes         Yes
Yes
DOMAIN1       domain1.example.com    None        Yes         Yes
Yes
DOMAIN2       domain2.example.com    External    No          Yes
Yes
```

Additional resources
For further details, see the **wbinfo(1)** man page.

2.10. RELATED INFORMATION

- The Red Hat Samba packages include manual pages for all Samba commands and configuration files the package installs. For example, to display the man page of the **/etc/samba/smb.conf** file that explains all configuration parameters you can set in this file:

```
# man 5 smb.conf
```

- The **/usr/share/docs/samba-version/** directory contains general documentation, example scripts, and LDAP schema files, provided by the Samba project.

- Red Hat Cluster Storage Administration Guide: Provides information about setting up Samba and the Clustered Trivial Database (CDTB) to share directories stored on an GlusterFS volume.

- For details about mounting an SMB share on Red Hat Enterprise Linux, see the corresponding section in the Configuring and managing storage and file systems documentation.

[1] Parts of this section were adopted from the idmap config autorid documentation published in the Samba Wiki. License: CC BY 4.0. Authors and contributors: See the history tab on the Wiki page.

[2] Parts of this section were adopted from the Setting up a Share Using POSIX ACLs documentation published in the Samba Wiki. License: CC BY 4.0. Authors and contributors: See the history tab on the Wiki page.

[3] Parts of this section were adopted from the Setting up a Share Using Windows ACLs documentation published in the Samba Wiki. License: CC BY 4.0. Authors and contributors: See the history tab on the Wiki page.

[4] Parts of this section were adopted from the Setting up Samba as a Print Server documentation published in the Samba Wiki. License: CC BY 4.0. Authors and contributors: See the history tab on the Wiki page.

[5] Parts of this section were adopted from the Setting up Automatic Printer Driver Downloads for Windows Clients documentation published in the Samba Wiki. License: CC BY 4.0. Authors and contributors: See the history tab on the Wiki page.

[6] Parts of this section were adopted from the Performance Tuning documentation published in the Samba Wiki. License: CC BY 4.0. Authors and contributors: See the history tab on the Wiki page.

CHAPTER 3. DATABASE SERVERS ON RED HAT ENTERPRISE LINUX 8

3.1. INTRODUCTION TO DATABASES AND DATABASE SERVERS

Databases are organized collections of data. The data in databases are stored and can be accessed electronically.

The organization of data in a database is typically designed to support:

- Modelling of various aspects of reality
- Querying and filtering data according to their attributes

Red Hat Enterprise Linux 8 provides the following database servers:

- MariaDB 10.3
- MySQL 8.0
- PostgeSQL 10
- PostgreSQL 9.6

3.2. USING MARIADB ON RED HAT ENTERPRISE LINUX 8

3.2.1. Getting started with MariaDB on Red Hat Enterprise Linux 8

The **MariaDB** server is an open source fast and robust database server that is based on MySQL technology.

MariaDB is a relational database which converts data into structured information and provides an SQL interface for accessing data. It includes multiple storage engines and plug-ins, as well as geographic information system (GIS) and JavaScript Object Notation (JSON) features.

This section describes how to install the **MariaDB** server in Installing MariaDB or how to migrate from the Red Hat Enterprise Linux 7 default version, **MariaDB 5.5**, to the Red Hat Enterprise Linux 8 default version, **MariaDB 10.3**, in Migrating to MariaDB 10.3. One of the prerequisites of migration is performing data backup, which is described in Backing up MariaDB data.

3.2.2. Installing MariaDB

To install **MariaDB**, follow this procedure:

1. Ensure that the **`mariadb`** package, available in the AppStream repository, is installed on the required server:

   ```
   ~]# yum install mariadb
   ```

2. Start the **`mariadb`** service:

   ```
   ~]# systemctl start mariadb.service
   ```

3. Enable the **`mariadb`** service to start at boot:

```
~]# systemctl enable mariadb.service
```

> **NOTE**
>
> Note that the **MariaDB** and **MySQL** database servers cannot be installed in parallel in Red Hat Enterprise Linux 8.0 due to conflicting RPM packages. Parallel installation of components is possible in Red Hat Software Collections for Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7. In Red Hat Enterprise Linux 8, different versions of database servers can be used in containers.

3.2.2.1. Improving MariaDB installation security

You can improve security when installing **MariaDB** by running the **`mysql_secure_installation`** command:

```
~]# mysql_secure_installation
```

This command launches a fully interactive script, which prompts for each step in the process.

The script enables to improve security in the following ways:

- Setting a password for root accounts
- Removing anonymous users
- Disallowing remote (outside the local host) root logins

3.2.3. Configuring MariaDB

3.2.3.1. Configuring the MariaDB server for networking

To configure the **MariaDB** server for networking, use the **`[mysqld]`** section of the **`/etc/my.cnf.d/mariadb-server.cnf`** file, where you can set the following configuration directives:

- **`bind-address`**
  Bind-address is the address on which the server will listen.

  Possible options are: a host name, an IPv4 address, or an IPv6 address.

- **`skip-networking`** +fna Possible values are:
  0 - to listen for all clients

  1 - to listen for local clients only

- **`port`**
  The port on which **MariaDB** listens for TCP/IP connections.

3.2.4. Backing up MariaDB data

3.2.4.1. Types of MariaDB backup

There are two main ways to back up data from a MariaDB database:

- Logical backup
- Physical backup

Logical backup consists of the SQL statements necessary to restore the data. This type of backup exports information and records in plain text files.

The main advantage of logical backup over physical backup is portability and flexibility. The data can be restored on other hardware configurations, MariaDB versions or Database Management System (DBMS), which is not possible with physical backups.

Note that logical backup can be performed if the **`mariadb.service`** is running. Logical backup does not include log and configuration files.

Physical backup consists of copies of files and directories that store the content.

Physical backup has the following advantages compared to logical backup:

- Output is more compact.
- Backup is smaller in size.
- Backup and restore are faster.
- Backup includes log and configuration files.

Note that physical backup must be performed when the **`mariadb.service`** is not running or all tables in the database are locked to prevent changes during the backup.

3.2.4.2. MariaDB backup methods

You can use one of the following approaches to back up data from a **MariaDB** database:

- Section 3.2.4.3, “Logical backup with mysqldump”
- Section 3.2.4.4, “Physical online backup using the Mariabackup tool”
- Section 3.2.4.5, “File system backup”
- Section 3.2.4.6, “Replication as a backup solution”

3.2.4.3. Logical backup with mysqldump

The **mysqldump** client is a backup utility, which can can be used to dump a database or a collection of databases for the purpose of a backup or transfer to another database server. The output of **mysqldump** typically consists of SQL statements, which enable to create a table, populate a table, or create a table and populate it. Alternatively, **mysqldump** can also generate files in other formats, including CSV or other delimited text formats, and XML.

For more information on logical backup with **mysqldump**, see the MariaDB Documentation.

To perform the **mysqldump** backup, you can use one of the following options:

- Back up a selected database

- Back up a subset of tables from one database
- Back up multiple databases
- Back up all databases

3.2.4.3.1. Frequently used commands in mysqldump backups

- To back up an entire database, run:

  ```
  ~]# mysqldump [options] db_name > backup-file.sql
  ```

- To back up a subset of tables from one database, add a list of the chosen tables at the end of the command:

  ```
  ~]# mysqldump [options] db_name [tbl_name …]
  ```

- To load the dump file back into a server, run:

  ```
  ~]# mysql db_name < backup-file.sql
  ```

  or

  ```
  ~]# mysql -e "source /path-to-backup/backup-file.sql" db_name
  ```

- To populate databases by copying data from one **MariaDB** server to another, run:

  ```
  ~]# mysqldump --opt db_name | mysql --host=remote_host -C db_name
  ```

- To dump multiple databases at once, run:

  ```
  ~]# mysqldump [options] --databases db_name1 [db_name2 …] >
  my_databases.sql
  ```

- To dump all databases, run:

  ```
  ~]# mysqldump [options]--all-databases > all_databases.sql
  ```

- To see a list of the options that mysqldump supports, run:

  ```
  ~]$ mysqldump --help
  ```

3.2.4.4. Physical online backup using the Mariabackup tool

Mariabackup is a tool based on the Percona XtraBackup technology, which enables performing physical online backups of InnoDB, Aria, and MyISAM tables.

Mariabackup, provided by the **`mariadb-backup`** package from the AppStream repository, supports full backup capability for **MariaDB** server, which includes encrypted and compressed data.

To install **Mariabackup**, run the following command as **`root`**:

```
~]# yum install mariadb-backup
```

To configure **Mariabackup**, set the user name and password by adding the following lines into the **`[xtrabackup]`** or **`[mysqld]`** section of the **`*.cnf`** configuration file that you need to create (for example, **`/etc/my.cnf.d/mariabackup.cnf`**):

```
[xtrabackup]
user=myuser
password=mypassword
```

IMPORTANT

Mariabackup does not read options in the **`[mariadb]`** section of configuration files. If a non-default data directory is specified on a **MariaDB** server, you must specify this directory in the **`[xtrabackup]`** or **`[mysqld]`** sections of configuration files, so that **Mariabackup** is able to find the data directory.

To specify such a data directory, include the following line in the **`[xtrabackup]`** or **`[mysqld]`** sections of **MariaDB** configuration files:

```
datadir=/var/mycustomdatadir
```

NOTE

Users of **Mariabackup** must have the **`RELOAD`**, **`LOCK TABLES`**, and **`REPLICATION CLIENT`** privileges to be able to work with the backup.

To create a backup of a database using **Mariabackup**, run the following command:

```
~]$ mariabackup --backup --target-dir <backup_directory> --user
<backup_user> --password <backup_passwd>
```

The **`target-dir`** option defines the directory where the backup files will be stored.

Note that if you want to perform a full backup, the target directory must be empty or not exist.

For more information on performing backups with **Mariabackup**, see Full Backup and Restore with Mariabackup.

3.2.4.5. File system backup

To perform a file system backup, copy the data files to another location. Make sure that the **`mariadb`** service is not running when copying the data files.

To create a file system backup of **MariaDB** data files, change to the **`root`** user, and use the following procedure:

1. Stop the **`mariadb`** service:

   ```
   ~]# systemctl stop mariadb.service
   ```

2. Copy the data files to the required location:

```
~]# cp -r /var/lib/mysql /backup-location
```

3. Start the **mariadb** service:

   ```
   ~]# systemctl start mariadb.service
   ```

3.2.4.6. Replication as a backup solution

Replication is an alternative backup solution for master servers. If a master server replicates to a slave server, backups can be run on the slave without any impact on the master.

> **WARNING**
>
> Replication is not a sufficient backup solution. Replication protects master servers against hardware failures, but it does not ensure protection against data loss.

For more information on replication as a backup solution, see MariaDB Documentation.

3.2.5. Migrating to MariaDB 10.3

Red Hat Enterprise Linux 7 contains **MariaDB 5.5** as the default implementation of a server from the MySQL databases family. Later versions of the **MariaDB** database server are available as Software Collections for Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7. Red Hat Enterprise Linux 8 provides **MariaDB 10.3** and **MySQL 8.0**.

3.2.5.1. Notable differences between the RHEL 7 and RHEL 8 versions of MariaDB

The most important changes between **MariaDB 5.5** and **MariaDB** 10.3 are as follows:

- **MariaDB** Galera Cluster, a synchronous multi-master cluster, is a standard part of **MariaDB** since 10.1.

- The ARCHIVE storage engine is no longer enabled by default, and the plug-in needs to be specifically enabled.

- The BLACKHOLE storage engine is no longer enabled by default, and the plug-in needs to be specifically enabled.

- InnoDB is used as the default storage engine instead of XtraDB, which was used in **MariaDB 10.1** and earlier versions.
  For more details, see Why does MariaDB 10.2 use InnoDB instead of XtraDB?.

- The new **mariadb-connector-c** packages provide a common client library for MySQL and MariaDB. This library is usable with any version of the **MySQL** and **MariaDB** database servers. As a result, the user is able to connect one build of an application to any of the MySQL and **MariaDB** servers distributed with Red Hat Enterprise Linux 8.

To migrate from **MariaDB 5.5** to **MariaDB 10.3**, you need to perform multiple configuration changes as described in Section 3.2.5.2, “Configuration changes”.

3.2.5.2. Configuration changes

The recommended migration path from **MariaDB 5.5** to **MariaDB 10.3** is to upgrade to **MariaDB 10.0** first, and then upgrade by one version successively.

For more information about configuration changes when migrating from **MariaDB 5.5** to **MariaDB 10.0**, see Migrating to MariaDB 10.0 in Red Hat Software Collections documentation.

The migration to following successive versions of **MariaDB** and the required configuration changes is described in these documents:

- Migrating to MariaDB 10.1 in Red Hat Software Collections documentation.
- Migrating to MariaDB 10.2 in Red Hat Software Collections documentation.
- Upgrading from MariaDB 10.2 to MariaDB 10.3 in upstream documentation.

> **NOTE**
>
> Migration directly from **MariaDB 5.5** to **MariaDB 10.3** is also possible, but you must perform all configuration changes that are required by differences described in the migration documents above.

3.2.5.3. Types of upgrades

Users of **MariaDB** on Red Hat Enterprise Linux 7 can use two migration paths for the database files to Red Hat Enterprise Linux 8:

- In-place upgrade using the mysql_upgrade tool
- Dump and restore upgrade

3.2.5.3.1. In-place upgrade using the mysql_upgrade tool

To perform an in-place upgrade, you need to copy binary data files to the `/var/lib/mysql/` data directory on the Red Hat Enterprise Linux 8 system and use the `mysql_upgrade` tool.

You can use this method for migrating data from:

- The Red Hat Enterprise Linux 7 version of **MariaDB 5.5**
- The Red Hat Software Collections versions of:
  - **MariaDB 5.5** (no longer supported)
  - **MariaDB 10.0** (no longer supported)
  - **MariaDB 10.1**
  - **MariaDB 10.2**

Note that it is recommended to upgrade to **MariaDB 10.2** by one version successively. See the respective Migration chapters in the Release Notes for Red Hat Software Collections.

NOTE

If you are upgrading from the Red Hat Enterprise Linux 7 version of **MariaDB**, the source data is stored in the **/var/lib/mysql/** directory. In case of Red Hat Software Collections versions of **MariaDB**, the source data directory is **/var/opt/rh/<collection_name>/lib/mysql/** (with the exception of the **mariadb55**, which uses the **/opt/rh/mariadb55/root/var/lib/mysql/** data directory).

IMPORTANT

Before performing the upgrade, back up all your data stored in the **MariaDB** databases as described in Section 3.2.4, “Backing up MariaDB data”.

To perform the in-place upgrade, change to the **root** user, and use the following procedure:

1. Ensure that the **mariadb-server** package is installed on the Red Hat Enterprise Linux 8 system:

   ```
   ~]# yum install mariadb-server
   ```

2. Ensure that the mariadb daemon is not running on either of the source and target systems at the time of copying data:

   ```
   ~]# systemctl stop mariadb.service
   ```

3. Copy the data from the source location to the **/var/lib/mysql/** directory on the Red Hat Enterprise Linux 8 target system.

4. Set the appropriate permissions and SELinux context for copied files on the target system:

   ```
   ~]# restorecon -vr /var/lib/mysql
   ```

5. Start the **MariaDB** server on the target system:

   ```
   ~]# systemctl start mariadb.service
   ```

6. Run the **mysql_upgrade** command to check and repair internal tables:

   ```
   ~]# systemctl mysql_upgrade
   ```

7. When the upgrade is complete, make sure that all configuration files within the **/etc/my.cnf.d/** directory include only valid options for **MariaDB 10.3**.

IMPORTANT

The in-place upgrade method is usually faster compared to the method when you perform a full SQL dump of the database server, initialize an empty database and load the SQL dump back.

However, there are certain risks and known problems related to in-place upgrade. For example, some queries might not work or they will be run in different order than before the upgrade. For more information on these risks and problems, and for general information about in-place upgrade, see MariaDB 10.3 Release Notes.

3.2.5.3.2. Dump and restore upgrade

When using the dump and restore upgrade method, you need to dump all databases contents into an SQL file dump file.

You can use this method for migrating data from:

- The Red Hat Enterprise Linux 6 version of **MySQL 5.1**
- The Red Hat Enterprise Linux 7 version of **MariaDB 5.5**
- The Red Hat Enterprise Linux 8 version of **MySQL 8.0**
- The Red Hat Software Collections versions of:
  - **MariaDB 5.5** (no longer supported)
  - **MariaDB 10.0** (no longer supported)
  - **MariaDB 10.1**
  - **MariaDB 10.2**
  - **MySQL 5.5** (no longer supported)
  - **MySQL 5.6** (no longer supported)
  - **MySQL 5.7**
  - **MySQL 8.0**

Note that the dump and restore upgrade is slower than the in-place upgrade method. In addition, it may require significant manual fixing in the generated SQL file and changes to configuration to resolve incompatibilities. See also Section 3.2.5.2, “Configuration changes”.

NOTE

The data files that **Mariabackup** creates in the target directory are not point-in-time consistent, given that the data is taken at different times during the backup operation. If you try to restore from these files, InnoDB notices the discrepancies and terminates to protect your data from corruption.

IMPORTANT

Before performing the upgrade, back up all your data stored in the **MariaDB** databases as described in Section 3.2.4, “Backing up MariaDB data”.

To perform the dump and restore upgrade, change to the **`root`** user, and follow this procedure, which uses the Red Hat Enterprise Linux 7 system version of **MariaDB 5.5** as an example:

1. On your source Red Hat Enterprise Linux 7 system, start the **`mariadb`** daemon:

   ```
   ~]# systemctl start mariadb.service
   ```

2. On your source Red Hat Enterprise Linux 7 system, dump all databases into the **`all_databases.sql`** file:

   ```
   ~]# mysqldump --all-databases > all_databases.sql
   ```

3. Ensure that the **`mariadb-server`** package is installed on the Red Hat Enterprise Linux 8 system:

   ```
   ~]# yum install mariadb-server
   ```

4. Ensure that the **`mariadb`** daemon is not running on the target Red Hat Enterprise Linux 8 system:

   ```
   ~]# systemctl stop mariadb.service
   ```

5. Copy the dumped **`all_databases.sql`** file to the Red Hat Enterprise Linux 8 system.

6. On the Red Hat Enterprise Linux 8 system, start the **`mariadb`** daemon:

   ```
   ~]# systemctl start mariadb.service
   ```

7. On the Red Hat Enterprise Linux 8 system, restore the data:

   ```
   ~]# mysql db_name < all_databases.sql
   ```

3.3. USING POSTGRESQL ON RED HAT ENTERPRISE LINUX 8

3.3.1. Getting started with PostgreSQL on Red Hat Enterprise Linux 8q

The **PostgreSQL** server is an open source robust and highly-extensible database server based on the SQL language. It provides an object-relational database system, which allows to manage extensive datasets and a high number of concurrent users. For these reasons, the PostgreSQL servers can be used in clusters to manage high amounts of data.

The **PostgreSQL** server includes features for ensuring data integrity, building fault-tolerant environments or building applications. It allows to extend a database with user's own data types, custom functions, or code from different programming languages without the need to recompile the database.

This section describes how to install **PostgreSQL** in Installing PostgreSQL or how to migrate from the Red Hat Enterprise Linux 7 default version **PostgreSQL 9.2** to Red Hat Enterprise Linux 8 default version **PostgreSQL 10.0** in Migrating to PostgreSQL 10.0. One of the prerequisites of migration is performing a data backup as described in Backing up PostgreSQL data.

3.3.2. Installing PostgreSQL

In Red Hat Enterprise Linux 8.0 Beta, the **PostgreSQL** server is available in two versions: 10 and 9.6, provided by two streams. This section assumes using **PostgreSQL 10** from the default stream. For information about changing streams, see Using Application Stream.

NOTE

By design, it is impossible to install more than one version (stream) of the same module in parallel. For example, you need to choose only one of the available streams from the **`postgresql`** module, either 10 (default) or 9.6. Parallel installation of components is possible in Red Hat Software Collections for Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7. In Red Hat Enterprise Linux 8, different versions of database servers can be used in containers.

To install **PostgreSQL** follow this procedure:

1. Ensure that the **`postgresql-server`** package, available in the Application Stream repository, is installed on the required server:

   ```
   ~]# yum install postgresql-server
   ```

2. Initialize the data directory

   ```
   postgresql-setup --initdb
   ```

3. Start the **`postgresql`** service:

   ```
   ~]# systemctl start postgresql.service
   ```

4. Enable the **`postgresql`** service to start at boot:

   ```
   ~]# systemctl enable postgresql.service
   ```

3.3.3. Configuring PostgreSQL

To change the **PostgreSQL** configuration, use the **`/var/lib/pgsql/data/postgresql.conf`** file. Afterwards, restart the **`postgresql`** service so that the changes become effective:

```
systemctl restart postgresql.service
```

Apart from **`/var/lib/pgsql/data/postgresql.conf`**, other files to change **PostgreSQL** configuration exist:

- **`postgresql.auto.conf`**
- **`pg_ident.conf`**
- **`pg_hba.conf`**

The **`postgresql.auto.conf`** file holds basic **PostgreSQL** settings similarly to **`/var/lib/pgsql/data/postgresql.conf`**. However, this file is under the server control. It is edited by the **`ALTER SYSTEM`** queries, and cannot be edited manually.

The **pg_ident.conf** file is used for mapping user identities from external authentication mechanisms into the postgresql user identities.

The **pg_hba.conf** file is used for configuring detailed user permissions for **PostgreSQL** databases.

3.3.3.1. Initializing a database cluster

In a **PostgreSQL** database, all data is stored a single directory, which is called a database cluster. You can choose where to store your data but Red Hat recommends to store the data in the default **/var/lib/pgsql/data** directory.

To initialize this data directory, run:

```
postgresql-setup --initdb
```

3.3.4. Backing up PostgreSQL data

To back up **PostgreSQL** data, use one of the following approaches:

- SQL dump
- File system level backup
- Continuous archiving

3.3.4.1. Backing up PostgreSQL data with an SQL dump

3.3.4.1.1. Performing an SQL dump

The SQL dump method is based on generating a file with SQL commands. When this file is uploaded back to the database server, it recreates the database in the same state as it was at the time of the dump. The SQL dump is ensured by the **pg_dump** utility, which is a **PostgreSQL** client application. The basic usage of the **pg_dump** command is such that the command writes its result into the standard output:

```
pg_dump dbname > dumpfile
```

The resulting SQL file can be either in a text format or in other different formats, which allows for parallelism and for more detailed control of object restoration.

You can perform the SQL dump from any remote host that has access to the database. The **pg_dump** utility does not operate with special permissions, but it must have a read access to all tables that you want to back up. To back up the entire database, you must run it as a database superuser.

To specify which database server **pg_dump** will contact, use the following command-line options:

- The **-h** option to define the host.
  The default host is either the local host or what is specified by the **PGHOST** environment variable.
- The **-p** option to define the port.
  The default port is indicated by the **PGPORT** environment variable or the compiled-in default.

NOTE

Note that **pg_dump** dumps only a single database. It does not dump information about roles or tablespaces because such information is cluster-wide.

To back up each database in a given cluster and to preserve cluster-wide data, such as role and tablespace definitions, use the **pg_dumpall** command:

```
pg_dumpall > dumpfile
```

3.3.4.1.2. Restoring database from an SQL dump

To restore a database from an SQL dump:

1. Create a new database (dbname):

   ```
   createdb dbname
   ```

2. Make sure that all users who own objects or were granted permissions on objects in the dumped database already exist.
   If such users do not exist, the restore fails to recreate the objects with the original ownership and permissions.

3. Run the **psql** utility to restore a text file dump created by the **pg_dump** utility:

   ```
   psql dbname < dumpfile
   ```

where **dumpfile** is the output of the **pg_dump** command.

If you want to restore a non-text file dump, use the **pg_restore** utility:

```
pg_restore non-plain-text-file
```

3.3.4.1.2.1. Restoring a database on another server

Dumping a database directly from one server to another is possible because **pg_dump** and **psql** can write to and read from pipes.

To dump a database from one server to another, run:

```
pg_dump -h host1 dbname | psql -h host2 dbname
```

3.3.4.1.2.2. Handling SQL errors during restore

By default, **psql** continues to execute if an SQL error occurs. Consequently, the database is restored only partially.

If you want to change this default behavior, use one of the following approaches:

- Make **psql** exit with an exit status of 3 if an SQL error occurs by setting the**ON_ERROR_STOP** variable:

  ```
  psql --set ON_ERROR_STOP=on dbname < dumpfile
  ```

- Specify that the whole dump is restored as a single transaction so that the restore is either fully completed or canceled by using **`psql`** with one of the following options:

  ```
  psql -1
  ```

  or

  ```
  psql --single-transaction
  ```

  Note that when using this approach, even a minor error can cancel a restore operation that has already run for many hours.

3.3.4.1.3. Advantages and disadvantages of an SQL dump

An SQL dump has the following advantages compared to other **PostgreSQL** backup methods:

- An SQL dump is the only **PostgreSQL** backup method that is not server version-specific. The output of the **pg_dump** utility can be reloaded into later versions of **PostgreSQL**, which is not possible for file system level backups or continuous archiving.
- An SQL dump is the only method that works when transferring a database to a different machine architecture, such as going from a 32-bit to a 64-bit server.
- An SQL dump provides internally consistent dumps. A dump represents a snapshot of the database at the time pg_dump began running.
- The **pg_dump** utility does not block other operations on the database when it is running.

A disadvantage of an SQL dump is that it takes more time compared to file system level backup.

3.3.4.1.4. Additional resources

For more information about the SQL dump, see PostgreSQL 10 Documentation.

3.3.4.2. Backing up PostgreSQL data with a file system level backup

3.3.4.2.1. Performing a file system level backup

To perform a file system level backup, you need to copy the files that **PostgreSQL** uses to store the data in the database to another location:

1. Choose the location of a database cluster and initialize this cluster as described in Section 3.3.3.1, “Initializing a database cluster”.
2. Stop the postgresql service:

   ```
   ~]# systemctl stop postgresql.service
   ```

3. Use any method to make a file system backup, for example:

   ```
   tar -cf backup.tar /var/lib/pgsql/data
   ```

4. Start the postgresql service:

```
~]# systemctl start postgresql.service
```

3.3.4.2.2. Advantages and disadvantages of a file system level backup

A file system level backup has the following advantage compared to other **PostgreSQL** backup methods:

- File system level backup is usually faster than SQL dump.

File system level backup has the following disadvantages compared to other **PostgreSQL** backup methods:

- The backup is architecture-specific and Red Hat Enterprise Linux 7-specific. It can only be used as a backup to return to Red Hat Enterprise Linux 7 if the upgrade was not successful, but it cannot be used with **PostgreSQL 10.0**.
- The database server must be shut down before data backup and before data restore as well.
- Backup and restore of certain individual files or tables is impossible. The file system backups only work for a complete backup and restoration of an entire database cluster.

3.3.4.2.3. Alternative approaches to file system level backup

Alternative approaches to file system backup include:

- A consistent snapshot of the data directory
- The rsync utility

3.3.4.2.4. Additional resources

For more information about the file system level backup, see PostgreSQL 10 Documentation.

3.3.4.3. Backing up PostgreSQL data by continuous archiving

3.3.4.3.1. Introduction to continuous archiving

PostgreSQL records every change made to the database’s data files into a write ahead log (WAL) file that is available in the `pg_wal/` subdirectory of the cluster’s data directory. This log is intended primarily for a crash recovery. After a crash, the log entries made since the last checkpoint can be used for restoring the database to a consistency.

The continuous archiving method, also known as `online backup`, combines the WAL files with a file system level backup. If a database recovery is needed, you can restore the database from the file system backup and then replay log from the backed up WAL files to bring the system to the current state.

For this backup method, you need a continuous sequence of archived WAL files that extends back to the start time of your backup at least.

If you want to start using the continuous archiving backup method, make sure to set up and test your procedure for archiving WAL files before taking your first base backup.

NOTE

You cannot use **pg_dump** and **pg_dumpall** dumps as a part of a continuous archiving backup solution. These dumps produce logical backups, not file system level backups. As such, they do not contain enough information to be used by a WAL replay.

3.3.4.3.2. Performing continuous archiving backup

To perform a database backup and restore using the continuous archiving method, follow these steps:

1. Section 3.3.4.3.2.1, “Making a base backup”
2. Section 3.3.4.3.2.2, “Restoring the database using a continuous archive backup”

3.3.4.3.2.1. Making a base backup

To perform a base backup, use the **pg_basebackup** tool, which can create a base backup in the form of either individual files or a `tar` archive.

To use the base backup, you need to keep all the WAL segment files generated during and after the file system backup. The base backup process creates a backup history file that is stored into the WAL archive area and is named after the first WAL segment file that you need for the file system backup. When you have safely archived the file system backup and the WAL segment files used during the backup, which are specified in the backup history file, you can delete all archived WAL segments with names numerically less because they are no longer needed to recover the file system backup. However, consider keeping several backup sets to be certain that you can recover your data.

The backup history file is a small text file, which contains the label string that you gave to **pg_basebackup**, the starting and ending times, and WAL segments of the backup. If you used the label string to identify the associated dump file, then the archived history file is enough to tell you which dump file to restore.

With the continuous archiving method, you need to keep all the archived WAL files back to your last base backup. Thus the ideal frequency of base backups depends on:

- The storage volume available for archived WAL files.
- The maximum possible duration of data recovery in situations when recovery is necessary.
  In cases with long period since the last backup, the system replays more WAL segments, and the recovery thus takes more time.

For more information about making a base backup, see PostgreSQL 10 Documentation.

3.3.4.3.2.2. Restoring the database using a continuous archive backup

To restore a database using a continuous backup:

1. Stop the server:

   ```
   ~]# systemctl stop postgresql.service
   ```

2. Copy the necessary data to a temporary location.
   Preferably, copy the whole cluster data directory and any tablespaces. Note that this requires enough free space on your system to hold two copies of your existing database.

If you do not have enough space, save the contents of the cluster's **pg_wal** directory, which can contain logs that were not archived before the system went down.

3. Remove all existing files and subdirectories under the cluster data directory and under the root directories of any tablespaces you are using.

4. Restore the database files from your file system backup.
   Make sure that:

   - The files are restored with the correct ownership (the database system user, not **root**)

   - The files are restored with the correct permissions

   - The symbolic links in the **pg_tblspc/** subdirectory were restored correctly

5. Remove any files present in the **pg_wal/** subdirectory
   These files resulted from the file system backup and are therefore obsolete. If you did not archive **pg_wal/**, recreate it with proper permissions.

6. Copy the unarchived WAL segment files that you saved in step 2 into **pg_wal/** if there are such files.

7. Create the **recovery.conf** recovery command file in the cluster data directory.

8. Start the server:

   ```
   ~]# systemctl start postgresql.service
   ```

   The server will enter the recovery mode and proceed to read through the archived WAL files that it needs.

   If the recovery is terminated due to an external error, the server can simply be restarted and it will continue the recovery. When the recovery process is completed, the server renames **recovery.conf** to **recovery.done** to prevent accidental re-entering the recovery mode later, when the server starts normal database operations.

9. Check the contents of the database to make sure that the database has recovered into the required state.
   If the database has not recovered into the required state, return to step 1. If the database has recovered into the required state, allow the users to connect by restoring the **pg_hba.conf** file to normal.

For more information about restoring using the continuous backup, see PostgreSQL 10 Documentation.

3.3.4.3.3. Advantages and disadvantages of continuous archiving

Continuous archiving has the following advantages compared to other **PostgreSQL** backup methods:

- With the continuous backup method, it is possible to use a file system backup that is not entirely consistent because any internal inconsistency in the backup is corrected by the log replay. A file system snapshot is not needed; **tar** or a similar archiving tool is sufficient.

- Continuous backup can be achieved by continuing to archive the WAL files because the sequence of WAL files for the log replay can be indefinitely long. This is particularly valuable for large databases.

- Continuous backup supports point-in-time recovery. It is not necessary to replay the WAL entries to the end. The replay can be stopped at any point and the database can be restored to its state at any time since the base backup was taken.
- If the series of WAL files are continuously available to another machine that has been loaded with the same base backup file, it is possible to restore the other machine with a nearly-current copy of the database at any point.

Continuous archiving has the following disadvantages compared to other **PostgreSQL** backup methods:

- Continuous backup method supports only restoration of an entire database cluster, not a subset.
- Continuous backup requires extensive archival storage.

3.3.4.3.4. Additional resources

For more information on continuous archiving method, see PostgreSQL 10 Documentation.

3.3.5. Migrating to PostgreSQL 10.0

Red Hat Enterprise Linux 7 contains **PostgreSQL 9.2** as the default version of the **PostgreSQL** server. In addition, several versions of **PostgreSQL** are provided as Software Collections for Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7. Red Hat Enterprise Linux 8 provides **PostgreSQL 10.0** and **PostgreSQL 9.6**.

This section assumes migration from the Red Hat Enterprise Linux 7 system version of **Postgreql 9.2** to the Red Hat Enterprise Linux 8 version of **PostgreSQL 10**, which is provided by the default stream.

Users of **PostgreSQL** on Red Hat Enterprise Linux 7 can use two migration paths for the database files to Red Hat Enterprise Linux 8:

- Fast upgrade using the pg_upgrade tool
- Dump and restore upgrade

Use preferably the fast upgrade method, which is faster than the dump and restore process.

However, in certain cases, the fast upgrade does not work, and you can only use the dump and restore process. Such cases include:

- Cross-architecture upgrades
- Systems using the `plpython` or `plpython2` extensions
  Red Hat Enterprise Linux 8 Application stream repository includes only the `postgresql-plpython3` package, not the `postgresql-plpython2` package.
- Fast upgrade is not supported for migration from Red Hat Software Collections versions of **PostgreSQL**.

As a prerequisite for migration from **PostgreSQL 9.2** to **PostgreSQL 10.0**, back up your **PostgreSQL** databases as described in Backing up PostgreSQL data.

Dumping the databases and performing backup of the SQL files is a necessary part of the dump and restore process. However, you are recommended to do so if performing the fast upgrade as well.

3.3.5.1. Fast upgrade using the pg_upgrade tool

During a fast upgrade, you need to copy binary data files to the **/var/lib/pgsql/data/** directory and use the **pg_upgrade** tool. You can use this method for migrating data from the Red Hat Enterprise Linux 7 version of **PostgreSQL 9.2**. By default, all data is stored in the **/var/lib/pgsql/data/** directory on both the Red Hat Enterprise Linux 7 and Red Hat Enterprise Linux 8 systems. For migration from Red Hat Software Collections versions of **PostgreSQL**, use Dump and restore upgrade.

IMPORTANT

Before performing the upgrade, back up all your data stored in the **PostgreSQL** databases as described in Backing up PostgreSQL data.

To perform a fast upgrade, use the following procedure:

1. On the RHEL 8 system, install the **postgresql-server** and **postgresql-upgrade** packages:

   ```
   ~]# yum install postgresql-server postgresql-upgrade
   ```

   Optionally, if you used any **PostgreSQL** server modules on RHEL 7, install them also on the RHEL 8 system in two versions, compiled both against **PostgreSQL 9.2** (installed as the **postgresql-upgrade** package) and **PostgreSQL 10** (installed as the **postgresql-server** package). If you need to compile a third-party **PostgreSQL** server module, build it both against the **postgresql-devel** and **postgresql-upgrade-devel** packages.

2. Check the following items:

   - Basic configuration: On the RHEL 8 system, check whether your server uses the default **/var/lib/pgsql/data** directory and the database is correctly initialized and enabled. In addition, the data files must be stored in the same path as mentioned in the **/usr/lib/systemd/system/postgresql.service** file.

   - **PostgreSQL** servers: Your system can run multiple **PostgreSQL** servers. Make sure that the data directories for all these servers are handled independently.

   - **PostgreSQL** server modules: Ensure that the **PostgreSQL** server modules that you used on {RHE} 7 are installed on your RHEL 8 system as well. Note that plug-ins are installed in the **/usr/lib64/pgsql/** directory (or in the **/usr/lib/pgsql/** directory on 32-bit systems).

3. Ensure that the **postgresql** service is not running on either of the source and target systems at the time of copying data.

   ```
   ~]# systemctl stop postgresql.service
   ```

4. Copy the database files from the source location to the **/var/lib/pgsql/data/** directory on the RHEL 8 system.

5. Perform the upgrade process by running the following command as the **PostgreSQL** user:

   ```
   ~]$ /bin/postgresql-setup --upgrade
   ```

   This launches the **pg_upgrade** process in the background.

In case of failure, **`postgresql-setup`** provides an informative error message.

6. Copy the prior configuration from **`/var/lib/pgsql/data-old`** to the new cluster.
   Note that the fast upgrade does not reuse the prior configuration in the newer data stack and the configuration is generated from scratch. If you want to combine the old and new configurations manually, use the *.conf files in the data directories.

7. Start the new **PostgreSQL 10** server:

   ```
   ~]# systemctl start postgresql.service
   ```

8. Run the **`analyze_new_cluster.sh`** script located in the **PostgreSQL** home directory:

   ```
   su postgres -c '~/analyze_new_cluster.sh'
   ```

9. If you want the **PostgreSQL 10** server to be automatically started on boot, run:

   ```
   ~]# systemctl enable postgresql.service
   ```

3.3.5.2. Dump and restore upgrade

When using the dump and restore upgrade, you need to dump all databases contents into an SQL file dump file.

Note that the dump and restore upgrade is slower than the fast upgrade method and it may require some manual fixing in the generated SQL file.

You can use this method for migrating data from:

- TheRed Hat Enterprise Linux7 version of **PostgreSQL 9.2**
- The Red Hat Software Collections versions of:
  - **PostgreSQL 9.2** (no longer supported)
  - **PostgreSQL 9.4** (no longer supported)
  - **PostgreSQL 9.6**
  - **PostgreSQL 10**

On Red Hat Enterprise Linux 7 and Red Hat Enterprise Linux 8 systems, **PostgreSQL** data is stored in the **`/var/lib/pgsql/data/`** directory by default. In case of Red Hat Software Collections versions of **PostgreSQL**, the default data directory is **`/var/opt/rh/collection_name/lib/pgsql/data/`** (with the exception of **`postgresql92`**, which uses the **`/opt/rh/postgresql92/root/var/lib/pgsql/data/`** directory).

To perform the dump and restore upgrade, change the user to **`root`**, and use the following procedure for migrating from the base Red Hat Enterprise Linux 7 system version of **PostgreSQL 9.2** to the Red Hat Enterprise Linux 8 default version of **PostgreSQL 10**:

1. On your RHEL 7 system, start the **PostgreSQL 9.2** server:

   ```
   ~]# systemctl start postgresql.service
   ```

2. On the RHEL 7 system, dump all databases contents into the **pgdump_file.sql** file:

```
su - postgres -c "pg_dumpall > ~/pgdump_file.sql"
```

3. Make sure that the databases were dumped correctly:

```
su - postgres -c 'less "$HOME/pgdump_file.sql"'
```

   As a result, the path to the dumped sql file is displayed: **/var/lib/pgsql/pgdump_file.sql**.

4. On the RHEL 8 system, install the **postgresql-server** package:

```
~]# yum install postgresql-server
```

   Optionally, if you used any **PostgreSQL** server modules on RHEL 7, install them also on the RHEL 8 system. If you need to compile a third-party **PostgreSQL** server module, build it against the **postgresql-devel** package.

5. On the RHEL 8 system, initialize the data directory for the **PostgreSQL 10.0** server:

```
~]# postgresql-setup --initdb
```

6. On the RHEL 8 system, copy the **pgdump_file.sql** into the **PostgreSQL** home directory, and check that the file was copied correctly:

```
su - postgres -c 'test -e "$HOME/pgdump_file.sql" && echo exists'
```

7. Copy the configuration files from the RHEL 7 system:

```
su - postgres -c 'ls -1 $PGDATA/*.conf'
```

   The configuration files to be copied are:

   - **/var/lib/pgsql/data/pg_hba.conf**
   - **/var/lib/pgsql/data/pg_ident.conf**
   - **/var/lib/pgsql/data/postgresql.conf**

8. On the RHEL 8 system, start the new **PostgreSQL 10** server:

```
~]# systemctl start postgresql.service
```

9. On the RHEL 8 system, import data from the dumped sql file:

```
su - postgres -c 'psql -f ~/pgdump_file.sql postgres'
```

NOTE

When upgrading from a Red Hat Software Collections version of **PostgreSQL**, adjust the commands to include **`scl enable collection_name.`** For example, to dump data from the **`rh-postgresql96`** Software Collection, use the following command:

```
su - postgres -c 'scl enable rh-postgresql96 "pg_dumpall >
~/pgdump_file.sql"'
```