Red Hat Enterprise Linux Atomic Host 7

Installation and Configuration Guide

Installation and Configuration Guide

Last Updated: 2019-10-15

Red Hat Enterprise Linux Atomic Host 7 Installation and Configuration Guide

Installation and Configuration Guide

Legal Notice

Abstract

Install using various methods, upgrade or downgrade and create example applications

Table of Contents

CHAPTER 1. INTRODUCTION TO ATOMIC HOST

1.1. OPERATING SYSTEM CONTENT

Red Hat Enterprise Linux Atomic Host is a variation of Red Hat Enterprise Linux 7 optimized to run Linux containers. It has been built to be light-weight and efficient, making it a particularly optimal operating system to use as a container run-time system for cloud environments. RHEL Atomic Host comes with many tools for running containers preinstalled - **docker**, **atomic**, **etcd**, **flannel**. All-in-one **kubernetes** installs are still supported, but Red Hat no longer supports Kubernetes clusters.

Red Hat Enterprise Linux Atomic Host uses rpm-OSTree, an open source tool, to manage bootable, immutable, versioned file system trees made of RPM content. These trees are composed by Red Hat, from packages and the **rpm-ostree** tool replicates the trees atomically. This results in a strategy for upgrade and maintenance that centers around atomic updates. The use of **rpm-ostree** instead of **Yum** to upgrade and maintain software means that Red Hat Enterprise Linux Atomic Host is managed differently than other Red Hat Enterprise Linux 7 variants.

Specifically, when using Red Hat Enterprise Linux Atomic Host, the operating system content is mounted in read-only mode. There are only two writable directories for local system configuration: **/etc/** and **/var/**. Updates work in the following way: a new bootable file system tree is generated, which shares storage with the current file system tree. When you download the new system tree, the old one is retained in parallel with it. This means that the first, pre-upgrade, version of the file system tree can be atomically restored when needed.

User files that are intended to persist across upgrades, including containers and data, should be placed in the **/var/** directory. The operating system itself is stored in the **/usr/** directory and is read-only. If you perform a long file listing in the root directory using the command **ls -l** /, you will discover that many of the traditional root-level directories are symbolic links to one of these two locations. For example, the **/home/** directory is a symbolic link to the **/var/home/** directory. This directory will therefore persist across upgrades.

Starting with Red Hat Enterprise Linux Atomic Host 7.4.2, you can configure the /**var** directory to be a mount point. This allows placing /**var** into a separate partition, which prevents other mount points from getting full if /**var** gets full.

The default partitioning dedicates most of the available space for the containers, using direct LVM as the storage backend instead of the default loopback as it is on Red Hat Enterprise Linux. Storage is managed by the **docker-storage-setup** daemon, which creates two Logical Volumes during installation, **root** for the file system content, and **docker-pool** for the images and containers.

Red Hat Enterprise Linux Atomic Host uses SELinux to provide strong safeguards in multi-tenant environments. The iptables services are available as firewall, iptables is turned off by default.

NOTE

In some RHEL Atomic Host versions you can run either **docker** or **docker-latest**. However, Red Hat does not support running both **docker** and **docker-latest** on the same machine simultaneously.

1.2. SYSTEM REQUIREMENTS

Red Hat Enterprise Linux Atomic Host should be compatible with most hardware in systems that were factory built within the last two years. Hardware compatibility is a particularly important concern if you have an older or custom-built system. Because hardware specifications change almost daily, it is

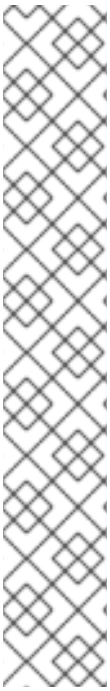
recommended that all systems be checked for compatibility. The most recent list of supported hardware can be found in the Red Hat Hardware Compatibility List . Also see Red Hat Enterprise Linux technology capabilities and limits for general information about system requirements.

Red Hat Enterprise Linux Atomic Host has the same runtime requirements as Red Hat Enterprise Linux. However, for Anaconda based installations (interactive or Kickstart) and PXE installations on bare metal or in virtual environments, a minimum 2GB of memory is required.

CHAPTER 2. TYPES OF INSTALLATION

Red Hat Enterprise Linux Atomic Host is distributed in multiple formats and able to be installed on bare-metal, in multiple virtual environments and in public and private cloud infrastructures.

You can find the installation media on the Red Hat Enterprise Linux Atomic Host Product Page when you click the **Download** button under **Installation Media**. Complete installation instructions can be found in the Red Hat Enterprise Linux Installation Guide .



NOTE

Not every version of RHEL Atomic Host has an **.iso** image available. For example, **rhel-atomic-installer-7.3.3-1.x86_64.iso** is available for Atomic Host 7.3.3, but no **.iso** is available for versions 7.3.4 to 7.3.6.

To install the latest Atomic Host from an **.iso**:

1. Download the latest available **.iso**.
2. Install it.
3. Register it.
4. Attach the subscription.
5. Run this command:

```
# atomic host upgrade
```

2.1. DEVELOPER MODE

Developer Mode provides a way to try out Atomic Host without actually going through an installation. It is available as an option in the GRUB boot menu on cloud images (but not on the bare-metal ISO) and this way you also avoid setting up the *meta-data* and *user-data* files and configuring cloud-init.

When your Atomic Host machine boots up, choose the "Developer Mode" selection in the GRUB boot menu to enter Developer Mode.

Developer Mode provides cloud-init with a local data source that automatically provides the following:

- a randomly-generated root password
- autologin of the root account into a tmux session
- pulling and starting of the **rhel7/cockpit-ws** container

2.2. PHYSICAL MACHINE INSTALLATIONS

2.2.1. Manual Partitioning

While physical machine installation of RHEL Atomic Host and RHEL is usually similar, there are some important differences. One such difference is which custom partitioning schemes are available.

In RHEL Atomic Host, unlike in RHEL, the /**var** directory is the only writeable directory (apart from the

small **/etc** directory). Hence, most writeable subdirectories of the root directory are actually stored in **/var**, which usually makes **/var** the biggest directory. Therefore, you might want to configure **/var** to be a mount point. It would allow you to place **/var** into a separate partition, which prevents other mount points from getting full if **/var** gets full.

Starting with RHEL Atomic Host 7.4.2, you can do this. If you decide to do manual partitioning, consider these points:

- Containers and their data are stored in **/var**. System containers are normally pulled to **/ostree** and hardlinked to **/var**, but if **/var** is on a separate partition, system containers are pulled to **/var** only.
  This means that **/var** is big. Make sure to dedicate a large enough partition to it.

- If for storage you use LVM thin-pool and devicemapper (default on RHEL Atomic Host), make sure to leave enough space in the volume group to allow for the thin-pool logical volume to be created and used. For instructions on this, see How to Leave Space in the Volume Group Backing Root During Installation.

- With extra precautions, you can even use a more advanced scheme, where subdirectories of **/var** are put on different partitions, for example:

  - **/var/lib/docker/** - for images for **docker** or **cri-o** containers (largest space, usually)

  - **/var/lib/containers/atomic/** - for system containers and images

  - **/var/lib/docker/volumes/** - for data from running containers

2.2.2. Anaconda Installation

You can find the procedure for installing RHEL Atomic Host with Anaconda in the Installing with Anaconda chapter of the Red Hat Enterprise Linux Installation Guide.

An important difference between installing RHEL Atomic Host and RHEL is which custom partitioning schemes are available. Generally, RHEL Atomic Host supports fewer partitioning schemes. Beginning RHEL Atomic Host 7.4.2, the **/var** directory can be configured to be a mount point. This allows placing **/var** into a separate partition, which prevents other mount points from getting full if **/var** gets full. For full manual partitioning instructions see the Manual Partitioning section of the Red Hat Enterprise Linux Installation Guide.

2.2.3. Kickstart Installation

To prepare for a Kickstart installation, you can follow the instructions in the Kickstart Installations chapter from the Red Hat Enterprise Linux Installation Guide. Kickstart installations of Red Hat Enterprise Linux Atomic Host do not differ much from Red Hat Enterprise Linux installations except for a few specific considerations.

Red Hat Enterprise Linux Atomic Host uses the **rpm-ostree** technology for package management and updates. Therefore, the **%packages** section is not used in the Kickstart file. Instead, the file *must* contain a command for including the **interactive-defaults.ks** file from the installation media. This file contains Kickstart commands that point to an OSTree repository on the media and also disable the cloud-init service.

Following is an example Kickstart file for Atomic Host which can be used as a reference:

```
lang en_US.UTF-8
keyboard us
```

```
timezone America/Chicago
#rootpw --iscrypted password_hash
rootpw --plaintext atomic
auth --enableshadow --passalgo=sha512
ostreesetup --nogpg --osname=rhel-atomic-host --remote=rhel-atomic-host --url=file:///install/ostree --
ref=rhel-atomic-host/7/x86_64/standard
services --disabled cloud-init,cloud-config,cloud-final,cloud-init-local
clearpart --all --initlabel
zerombr
autopart
#%include /usr/share/anaconda/interactive-defaults.ks
%post --erroronfail
fn=/etc/ostree/remotes.d/rhel-atomic-host.conf; if test -f ${fn} && grep -q -e '^url=file:///install/ostree'
${fn}$; then rm ${fn}; fi
%end
%post --erroronfail
rm -f /etc/ostree/remotes.d/*.conf
echo 'unconfigured-state=This system is not registered to Red Hat Subscription Management. You
can use subscription-manager to register.' >> $(ostree admin --print-current-dir).origin
%end"
```

Here is what the commands in that kickstart file do:

- The **rootpw** command tells the installer to set the root user's password using the plain text argument that follows (in this case, the password is set to **atomic**). You could you the **--iscrypted** option instead if you have a password hash you created previously.
- The **auth** command uses **--enableshadow** to tell the installer to store user passwords in the **/etc/shadow** file and **--passalgo=sha512** says to encrypt those passwords using the SHA512 algorithm.
- The **ostreesetup** command tells the installer how to get and setup the ostree file system.
- The **services** command disables some services that are inappropriate to an Atomic host.
- The **clearpart --all --initlabel** command erases all disks that can be reached by the installer, including any attached network storage.
- Using **zerombr** prevents Anaconda from prompting for confirmation which allows for an unattended installation.
- The **autopart** command sets up the partitioning automatically (more on that later).
- The **%include** command points to the file which contains commands that point to an OSTree repository and disables the cloud-init service. This command is mandatory for RHEL Atomic Host.
- The **%post** section at the end of the file runs several commands to further configure the system after installation is completed.

By default, partitioning for Red Hat Enterprise Linux Atomic Host is done automatically with the **autopart** command, to configure Logical Volume Management (LVM) style partitioning. Although **autopart** partitioning is preferred, you have the option of partitioning yourself to set such things as the names of physical volumes, volume groups, and logical volumes, along with the amount of disk space associated with those entities. Here is an example of how you might set partitioning manually, to replace the **autopart** entry shown in the kickstart example above:

```
zerombr
part /boot --ondisk=sda --asprimary --fstype="xfs"   --size=512
part pv.01 --ondisk=sda --asprimary --fstype="lvmpv" --grow
volgroup vg.atomic --pesize=16384 pv.01
logvol swap --fstype="swap" --name=lv.swap --vgname=vg.atomic --size=4096
logvol / --fstype="xfs" --name=lv.root --vgname=vg.atomic --grow
```

This example sets a 512MB primary partition with an xfs file system type on disk **/dev/sda** that is assigned to the /boot partition. The rest of the disk is assigned to an LVM physical volume (lvmpv) named pv.01. That physical volume is assigned to a volume group named vg.atomic. Two logical volumes are created from that volume group: a 4G swap partition and a root partition (/) with an XFS file system that consumes the rest of the remaining space from the volume group.

2.3. VIRTUAL MACHINE INSTALLATIONS

This chapter explains how to install Red Hat Enterprise Linux Atomic Host in several different virtualization environments and public cloud services. Before you start following the procedures below, download the appropriate ISO image for your environment as described in Downloading Red Hat Enterprise Linux from the *Red Hat Enterprise Linux 7 Installation Guide* .

2.3.1. Linux Hypervisor Installation with qcow2 Media

The following sections describe the installation of Red Hat Enterprise Linux Atomic Host using a **qcow2** disk image in a Linux hypervisor environment on a Red Hat Enterprise Linux 7 system.

Overview

Red Hat Enterprise Linux Atomic Host is available as a fully-configured disk image ready to be used with a Linux hypervisor. This variant is distributed as a compressed **gzip** archive. Decompress it using the following command:

```
# gzip -d rhel-atomic-host-7.qcow2.gz
```

The resulting uncompressed **qcow2** image can be used to create an instance of Red Hat Enterprise Linux Atomic Host. This means that the file will be written to once you start the virtual machine; after you use it to start one instance, you can not reuse it to start another one or reconfigure it using **cloud-init**. Therefore, you should back up the original **qcow2** file before starting the first instance. You can use the **qemu-img** command to create a **snapshot** of the unmodified file:

```
# qemu-img create -f qcow2 -o backing_file=rhel-atomic-host-standard.qcow2 atomic-beta-instance-0.qcow2
```

This command creates a snapshot named **rhel-atomic-host-standard.qcow2** which is the original, unmodified image, and a new file called **atomic-beta-instance-0.qcow2**, which can be used for the actual virtual machine.

2.3.1.1. Preparing for Installation

The installation configuration options are set with a pair of cloud-init configuration files:

- **meta-data**
  A plain text file which provides information that identifies the instance of Red Hat Enterprise Linux Atomic Host being installed. Its contents should be similar to the following example:

```
instance-id: Atomic0
local-hostname: atomic-00
```

The **instance-id** can be any identifying name and the **local-hostname** should be a host name that follows your site standards.

- **user-data**
  A plain text file which provides information about users on the system. This information will be used to enable access to the Red Hat Enterprise Linux Atomic Host instance. by default, the **root** user is password-locked; therefore, if you do not create the **user-data** file, you will not be able to log in.

  An example of a **user-data** file is below:

```
#cloud-config
password: atomic
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
- ssh-rsa AAA...SDvz user1@yourdomain.com
- ssh-rsa AAB...QTuo user2@yourdomain.com
```

NOTE

The first line of the example (**#cloud-config**) is not a comment or a command example - it is a mandatory line in the configuration file.

This example enables the **cloud-user** user to log in either with a password or an **SSH** key. The use of both methods is possible, but not required. An initial password is set on the **password** line; when the user logs in for the first time on this instance, they will be prompted to change their password as defined on the **chpasswd** line. Forcing the user to change their password after the first login is recommended because initially the password is stored in plain text.

The final four lines in the example configure remote login using **SSH**. The **ssh_pwauth: True** line enables **SSH** using a password, and the **ssh_authorized_keys** starts a block of one or more authorized public keys. Keys described in this file will be added to the **~/.ssh/authorized_keys** file. Each authorized key must be on a separate line and start with two spaces followed by a hyphen (**-**) and another space.

For additional information about these files, see the "Creating a cloud-init ISO File" section.

Once you have created both of the files described above, you must package them into the ISO image. This image will then be used as a virtual configuration CD on the virtual machine. To package the files into an image, use the following command:

```
# genisoimage -output atomic0-cidata.iso -volid cidata -joliet -rock user-data meta-data
```

This will create a new ISO image file named **atomic0-cidata.iso**.

2.3.1.2. Starting Red Hat Enterprise Linux Atomic Host for the First Time

After you unpacked the distributed **qcow2** image and created a configuration image as described in the previous section, you can create the virtual machine and begin the installation process. This section will describe creating an instance using the **virt-install** command; it is also possible to use the **virt-manager**

graphical interface. Both are documented in the Red Hat Enterprise Linux 7 Virtualization Deployment and Administration Guide. See also the Red Hat Enterprise Linux 7 Virtualization Getting Started Guide for introduction to virtualization on Red Hat Enterprise Linux 7.

The following command will create a new virtual machine using the **qcow2** image distributed by Red Hat and the configuration image you have created earlier:

```
# virt-install --import --name Atomic0 --ram 4096 --vcpus 2 --disk path=/path/to/rhel-atomic-host-standard.qcow2,format=qcow2,bus=virtio --disk path=/path/to/atomic0-cidata.iso,device=cdrom --network bridge=virbr0 --graphics vnc
```

The two **--disk-path=** options specify locations of the image files and device types which should be created (a **virtio** device for the main image and a virtual CD drive for the configuration image). It also assigns 4 GB of RAM (**--ram 4096**) and 2 virtual CPUs (**--vcpus 2**) to the virtual machine, sets up a VNC graphical interface (**--graphics vnc**) and a network bridge (**--network bridge=virbr0**). You can change these settings to suit your needs, but you must always use both of the image files.

NOTE

Currently, **DHCP** is the preferred network configuration method for use with Red Hat Enterprise Linux Atomic Host. Network settings can be changed by editing configuration files in the /**etc** directory after the initial boot.

NOTE

If you want to have your virtual machine accessible outside of the host machine. You should use a direct network interface. For example, you can replace **--network bridge=virbr0** with **--network type=direct,source=em1**, where **em1** is the name of an active network interface on the host system.

At this point, you can log into the Red Hat Enterprise Linux Atomic Host virtual machine using the credentials you set up in your **user-data** file. To access a **root** shell, use the **sudo -i** command. To connect to the virtual machine's console from the host system, use the following command:

```
# virsh console Atomic0
```

Replace **Atomic0** with the name of the virtual machine - the **--name** option of the **virt-install** command.

For information about working with your new Red Hat Enterprise Linux Atomic Host instance, see the Red Hat Enterprise Linux Atomic Host 7 Getting Started Guide .

2.3.2. Red Hat Enterprise Virtualization Environment Installation

The following sections explain how to use Red Hat Enterprise Virtualization (RHEV) to create virtual machines that run RHEL Atomic Host with **.ova** files and an ISO files.

.ova-based Installation

NOTE

RHEV OVA images of Atomic Host currently cannot be imported into RHEV.

See this Bugzilla for details.

The **.ova** based installation method allows for rapid deployment of a Red Hat Enterprise Linux Atomic Host installation, but permits less customization than does the ISO-based installation described in the subsequent section.

1. Acquire the RHEL Atomic Host **.ova** media from Download Red Hat Enterprise Linux .
2. Copy the **.ova** file to the Red Hat Enterprise Virtualization Manager.
3. Use the **engine-image-uploader** command to upload the **.ova** file to the Export storage domain.
4. Create instances of Red Hat Enterprise Linux from the **.ova** files uploaded to your Red Hat Enterprise Virtualization instance.

ISO-based Installation

The **.iso** based installation method allows for greater customization of the installation than does the **.ova** based installation method, but requires the configuration of the virtual machine hosting the Atomic environment.

1. Acquire the Red Hat Enterprise Linux Atomic Host installation media from Download Red Hat Enterprise Linux. and copy it to the Red Hat Enterprise Virtualization Manager's file system.
2. Use **engine-image-uploader** to add the ISO image to the storage domain of your Red Hat Enterprise Virtualization environment.
3. Attach the uploaded Red Hat Enterprise Linux Atomic Host ISO image to a new virtual machine and install Red Hat Enterprise Linux Atomic Host on that virtual machine.
4. Use the newly-created Red Hat Enterprise Linux Atomic Host virtual machine.

For more details, see the documentation set for Red Hat Enterprise Virtualization .

2.3.2.1. Installing Red Hat Enterprise Linux Atomic Host from an .ova File

The following section explains how to install Red Hat Enterprise Linux Atomic Host in Red Hat Enterprise Virtualization, from an **.ova** (Open Virtualization Appliance) source. This operation consists of a procedure in three stages. The first stage describes how to unpack the **.ova** file in the export storage domain of your Red Hat Enterprise Virtualization environment and how to set permissions so that Red Hat Enterprise Virtualization has ownership of the unpacked files. The second stage describes how to import the virtual machine template from the export domain into the Red Hat Enterprise Virtualization environment. The third stage describes how to create a virtual machine from the imported template.

Importing the .ova File with engine-image-uploader

This procedure explains how to use **rhevnm-image-uploader** to upload the virtual machine template of the Red Hat Enterprise Linux Atomic Host to the Export storage domain. Perform the following steps from within the Red Hat Enterprise Virtualization Manager environment.

1. Transfer the **.ova** file to the Red Hat Enterprise Virtualization Manager.

```
[a computer that is not the RHEV Manager]# scp filename.ova root@rhevm.hostname.com:/
```

2. Log in to the Red Hat Enterprise Virtualization Manager machine as root.

```
[a computer that is not the RHEV Manager]# ssh root@rhevm.hostname.com
```

3. Move to the directory to which you transferred the **.ova** file. In this example we assume that the directory is root (/):

```
[RHEVM]# cd /
```

4. Use the following command to upload the **.ova** file to the Export storage domain:

```
[RHEVM]# engine-image-uploader -N imagename -e Export upload filename.ova
```

Include **-N imagename** to give the image a human-readable file name. Otherwise, the name of the image will be a long alphanumeric string. Also substitute the name of your export domain for "Export" and the name of the .ova file for "filename.ova".

5. Provide the REST API password for the admin@internal oVirt engine user when prompted. The upload may take some time, depending on the size of the uploaded file. The upload succeeds silently, returning you to a command prompt when it is complete.

Importing the Virtual Machine Template into Red Hat Enterprise Virtualization

After the **.ova** file has been unpacked and the virtual machine template that it contained has its permissions set so that Red Hat Enterprise Virtualization can operate on it, you must import the virtual machine template into the Red Hat Enterprise Virtualization environment through the Administration Portal user interface. When this procedure is complete, it will be possible to create virtual machines from the imported template.

1. Sign in to the Red Hat Enterprise Virtualization Manager Administrator Portal as **admin**.

2. In the Red Hat Enterprise Virtualization Manager User Interface, click the **Storage** tab in the Navigation Pane (the pane at the top of the interface).

3. In the Red Hat Enterprise Virtualization Manager User Interface, click the name of the Export Domain in the Navigation Pane.

4. In the Red Hat Enterprise Virtualization Manager User Interface, click the **Template Import** tab in the Details Pane (the pane at the bottom of the interface).

5. In the Red Hat Enterprise Virtualization Manager User Interface, in the **Details Pane** (the pane at the bottom of the interface), click the name of the file you plan to import.

6. In the Red Hat Enterprise Virtualization Manager User Interface, click **Import** at the top left of the Details Pane.

7. In the **Import Template** window, click the name of the virtual machine you plan to import.

8. In the **Import Template** window, click **OK** in the bottom right corner.

Adding a cloud-init ISO to the ISO Domain

1. Create a cloud-init ISO by following the instructions in the "Creating a cloud-init ISO File" section.

2. From a machine remote to the RHEV Manager machine in your Red Hat Enterprise Virtualization environment, use **scp** to copy the cloud-init ISO to the file system of the RHEV Manager machine in the Red Hat Enterprise Virtualization Environment.

```
[a computer that is not the RHEV Manager]# scp atomic-cloud.iso root@rhevm.hostname.com:/
```

1. Log in to the Red Hat Enterprise Virtualization Manager machine as **root**.

```
[a computer that is not the RHEV Manager]# ssh root@rhevm.hostname.com
```

1. Move to the directory to which you uploaded the **atomic-cloud.iso**:

```
[RHEVM]# cd /
```

1. Use **rhevm-iso-uploader** to upload the cloud-init ISO to the ISO domain.

```
[RHEVM]# rhevm-iso-uploader --iso-domain=domain_name upload atomic-cloud.iso
```

1. Sign in to the Red Hat Enterprise Virtualization Manager Administrator Portal as **admin**.
2. In the Red Hat Enterprise Virtualization Manager User Interface, select the **Storage** tab in the **Navigation** pane.
3. In the **Details** pane (the pane at the bottom of the interface), select the **Images** tab.
4. Confirm that the **.iso** file is present in the ISO domain (it will appear in a list in the **Images** subtab of the **Details** pane if it is present).

Creating a Virtual Machine from the Imported Template

Now that your Red Hat Enterprise Linux Atomic Host virtual machine template has been unpacked and imported to your Red Hat Enterprise Virtualization environment and your cloud-init ISO file is present in the Red Hat Enterprise Virtualization ISO domain, you can create Red Hat Enterprise Linux Atomic Host virtual machines using the following procedure.

1. Log in to the Red Hat Enterprise Virtualization Manager user interface.
2. Open the **Virtual Machines** tab in the **Navigation** pane.
3. In the Navigation Pane of the Red Hat Enterprise Virtualization User Interface, click **New VM**.
4. In the **New Virtual Machine** window, in the **Based on Template** drop-down menu, select the name of the Red Hat Enterprise Linux Atomic Host template that you imported earlier.
5. In the **New Virtual Machine** window, fill out the "Name", "Description", and "Comment" fields.
6. In the **Boot Options** tab of the **New Virtual Machine** window, select the "Attach CD" check box, and select the name of the cloud-init ISO that contains the user credentials you want to use on this virtual machine.
7. Click **OK**.

Updating the RHEV Guest Agent in the Atomic Host VM

To allow the RHEV Manager to control an Atomic Host VM, a guest agent must be running on that VM. The ovirt-guest-agent interfaces with the RHEV Manager to supply run-time data and heart beat information, as well as allowing the RHEV Manager to control the operation of the VM (including shutdown and restart).

The latest Atomic Host ova image for RHEV includes the ovirt-guest-agent in the form of a container named rhevm-guest-agent. When you created a virtual machine from the imported ova image (as described previously), the rhevm-guest-agent container image included in the VM is automatically set

to run when the VM starts up.

You can check the status of the rhevm-guest-agent container (and update the container if needed) by logging into the Atomic Host VM on the RHEV environment and running the following commands:

1. List that the rhevm-guest-agent is available and running:

```
# runc list
ID              PID  STATUS  BUNDLE                                          CREATED
rhevm-guest-agent 674  running /var/lib/containers/atomic/rhevm-guest-agent.0   2017-06-...
```

1. Check the status of the rhevm-guest-agent running as a systemd service:

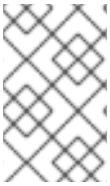
```
# systemctl status rhevm-guest-agent
● rhevm-guest-agent.service - oVirt Guest Agent Container
   Loaded: loaded (/etc/systemd/system/rhevm-guest-agent.service; enabled; vendor preset:
disabled)
   Active: active (running) since Mon 2017-06-19 19:06:58 UTC; 1 weeks 0 days ago
 Main PID: 644 (runc)
   Memory: 5.8M
   CGroup: /system.slice/rhevm-guest-agent.service
           └─644 /bin/runc --systemd-cgroup run rhevm-guest-agent
```

1. Update the rhevm-guest-agent. If an newer version of the rhevm-guest-agent container is available, you can update the container by running the following command (in this example, no new version was available):

```
# atomic containers update rhevm-guest-agent
Latest version already installed.
```

2.3.2.2. Installing Red Hat Enterprise Linux Atomic Host from an ISO Image

Uploading ISO



NOTE

This section pertains only to the procedure describing the installation of a Red Hat Enterprise Linux Atomic Host system from an ISO image. This section does not pertain to the creation of a Red Hat Enterprise Linux Atomic Host system from an **.ova** file.

1. Transfer the ISO file to the file system of the Red Hat Enterprise Virtualization Manager.

```
[a computer that is not the RHEV Manager]# scp filename.iso root@rhevm.hostname.com:/
```

2. Log in to the back end of the Red Hat Enterprise Virtualization Manager as root. Note that this does not mean that you should log in to the Red Hat Enterprise Virtualization Manager Administrator Portal.

```
[a computer that is not the RHEV Manager]# ssh root@rhevm.hostname.com
```

3. Move to the directory to which you transferred the ISO file:

```
[RHEVM]# cd /
```

4. Determine the name of the ISO storage domain on your Red Hat Enterprise Virtualization Manager. In the example here, the name of the ISO storage domain is **ISO_DOMAIN**:

```
# rhevm-iso-uploader list
ISO Storage Domain Name   | Datacenter             | ISO Domain Status
ISO_DOMAIN                | Default                | active
```

5. Use **rhevm-iso-uploader** to upload the Red Hat Enterprise Linux Atomic Host installation ISO image to the Red Hat Enterprise Virtualization storage domain:

```
[RHEVM]# rhevm-iso-uploader upload -i ISO_DOMAIN filename.iso
```

For more information on uploading ISO files to ISO domains in Red Hat Enterprise Virtualization, see the Red Hat Enterprise Virtualization Installation Guide .

Creating a Red Hat Enterprise Linux Atomic Virtual Machine

1. Log in to the Red Hat Enterprise Virtualization Manager.
2. Click the **Virtual Machines** tab.
3. Click the **New VM** button to open the **New Virtual Machine** window.
4. Click the **Show Advanced Options** button in the lower left corner of the **New Virtual Machine** window.
5. On the **General** tab, fill in the Name and Operating System fields. You can accept the default settings for other fields, or change them if required.
6. Click **Boot** Options in the menu on the left of the **New Virtual Machine** window.
7. In the **Boot** Sequence menu, select **CD-ROM** in the **First Device** drop-down menu.
8. In the **Boot Sequence** menu, select **Hard Disk** in the **Second Device** drop-down menu.
9. Select the **Attach CD** check box.
10. In the drop-down menu to the right of the **Attach CD** check box, select the name of the Red Hat Enterprise Linux Atomic Host installation ISO.
11. Click **OK** in the bottom right of the **New Virtual Machine** window.
12. The **New Virtual Machine - Guide Me** window opens, displaying two buttons: **Configure Network Interfaces** and **Configure Virtual Disks**.
13. Click **Configure Network Interfaces**.
14. The **New Network Interface** window opens. The default values in this window are sufficient to create a virtual network interface for the virtual machine.
15. Click **OK** in the bottom right of the **New Network Interface** window.
16. In the **New Virtual Machine - Guide Me** window, click the **Configure Virtual Disks** button.
17. The **New Virtual Disk** window opens. In the **Size (GB)** field, enter the size in gigabytes of your virtual hard drive.

18. Click **OK** in the bottom right of the **New Virtual Disk** window

19. In the **New Virtual Machine - Guide Me** window, click **Configure Later** in the bottom right.

2.3.3. Red Hat Enterprise Linux OpenStack Platform Installation

This section explains how to launch an instance of Red Hat Enterprise Linux Atomic Host on the Red Hat Enterprise Linux OpenStack Platform using a **QCOW2** image. Before you start the procedure, download the **QCOW2** image from here: Download Red Hat Enterprise Linux .

Creating a Red Hat Enterprise Linux Atomic Host Instance from a QCOW2 image

The following procedure assumes you are familiar with Red Hat Enterprise Linux OpenStack Platform. For more information about Red Hat Enterprise Linux OpenStack Platform, see the Red Hat Enterprise Linux OpenStack Platform End User Guide.

1. Create a project.
   a. Log into the Red Hat Enterprise Linux OpenStack Platform Dashboard
   b. Create a project by going to the **Admin Tab** and then clicking on **Projects** under *Identity Panel*.
   c. Click **Create Project** and provide a Project Name that is meets your site requirements. Additional configuration is not required, but should be done to meet your site requirements.
2. Setup networking for your project. This will vary by site configuration. In general the following steps are required:
   a. Create a network and a subnet for the internal networking for the project.
   b. Create a router and assign a gateway and create an interface to configure it to connect the internal network to the external network.
3. Create or upload a key pair to use with instances. The key pair settings can be found in the **Project Tab** under **Manage Compute** in **Access & Security** on the **Keypair Tab**.
4. Load the **QCOW2** image into Red Hat Enterprise Linux OpenStack Platform.
   a. Click **Images & Snapshots** located on the **Project Tab** under *Manage Compute*.
   b. Click **Create Image** and provide the following information:
      - *Name*: A meaningful image name
      - *Image Source*: Choose Image File to allow a file to be uploaded from your local workstation.
      - *Format*: Choose QCOW2
      - *Minimum Disk (GB)*: The minimum amount of disk space this image should be allowed to have. For more information, see Disk Space and Memory Requirements .
      - *Minimum Ram (MB)*: The minimum amount of memory this image should be allowed to have. For more information, see Disk Space and Memory Requirements .
   c. Finally, click **Choose File** and select the **QCOW2** image to upload and then click **Create Image** to start the upload.

5. Set up the instance to be launched, including basic first boot configuration using cloud-init.

    a. Access the *Launch Instance* dialog box by clicking on the **Launch Instance** button found on the **Projects Tab** under *Manage Compute* on the **Instances Screen**.

    b. Provide the following information in the *Launch Instance* dialog box on the **Details Tab**.

    - *Instance Name*: A meaningful instance name
    - *Flavor*: A properly sized instance for your application requirements that meets the minimum requirements for Red Hat Enterprise Linux Atomic Host.
    - *Instance Boot Source*: Choose the image you loaded in the previous step. For more information, see Disk Space and Memory Requirements .

    c. Provide the following information in the *Launch Instance* dialog box on the **Access & Security Tab**.

    - *Keypair*: Select the key pair you wish to use with this instance.

    d. Provide the following information in the *Launch Instance* dialog box on the **Networking Tab**.

    - *Selected Network*: Select the network you wish to use with this instance.

    e. Provide the following information in the *Launch Instance* dialog box on the **Post-Creation Tab**.

    - *Customization Script*: In this field, paste the equivalent of a **user-data** file for cloud-init. A **user-data** is a plain text file which provides information about users and configuration of the system. This information will be used to enable access to the Red Hat Enterprise Linux Atomic Host instance. By default, the **root** user is password protected; therefore, if you do not create the **user-data** file, you will not be able to log in.

    > An example of a `user-data` file is below:

    ```
    #cloud-config
    password: atomic
    chpasswd: {expire: False}
    ssh_pwauth: True
    ssh_authorized_keys:
    - ssh-rsa AAA...SDvz user1@yourdomain.com
    - ssh-rsa AAB...QTuo user2@yourdomain.com
    ```

    NOTE

    The first line of the example (**#cloud-config**) is not a comment or a command example - it is a mandatory line in the configuration file.

    This example enables the **cloud-user** user to log in either with a password or an SSH key. The use of both methods is possible, but not required. An initial password is set on the **password** line; when the user logs in for the first time on this instance, they will be prompted to change their password as defined on the **chpasswd** line. Forcing the user to change their password after the first login is recommended because initially the password is stored in plain text.

    The final four lines in the example configure remote login using SSH. The **ssh_pwauth:**

True line enables SSH using a password, and the **ssh_authorized_keys** starts a block of one or more authorized public keys. Keys described in this file will be added to the ~/**.ssh**/**authorized_keys** file. Each authorized key must be on a separate line and start with two spaces followed by a hyphen (**-**) and another space.

For additional information about this file, see the "Creating a cloud-init ISO File" section.

a. Click the **Launch** button to start your instance.

2.3.4. VMWare Installation

VMware vSphere provides a means of deploying and managing virtual machine resources. This section explains how to run Red Hat Enterprise Linux Atomic Host using the VMware vSphere Client. For the examples in this article, the ISO image was created on a Red Hat Enterprise Linux 7 system and Red Hat Enterprise Linux Atomic Host was run on VMware vSphere that was set up as a single ESXi 5.5 hypervisor and vCenter host running on a Microsoft Windows system.

Getting a Red Hat Enterprise Linux Atomic Host Image

To create a Red Hat Enterprise Linux Atomic Host virtual machine image that you can run on VMware vSphere, first download the Red Hat Enterprise Linux Atomic Host OVA file for VMware from the Download Red Hat Enterprise Linux page.

The vSphere OVA plug-in has a configurable network controller and a configurable SCSI controller.

The configurable parameters are:

vsphere_scsi_controller_type

Valid settings are: **lsilogic** and **VirtualSCSI**

vsphere_network_controller_type

Valid settings are: **E1000** and **VmxNet3**

When these parameters are not explicitly set, they default to the non-paravirtualization settings. The SCSI controller non-paravirtualization setting is **lsilogic**. The network controller non-paravirtualization setting is **E1000**.

Creating a cloud-init ISO File

You need to create a cloud-init ISO image that includes information that is used to configure the Red Hat Enterprise Linux Atomic Host system. This information can include a host name, a user name and password, and other configuration settings. Create the configuration information needed and produce the ISO image as described in the following steps:

1. Create cloud-init **meta-data** file.
   The final installation configuration options are set with a pair of cloud-init configuration files. The first installation configuration file contains the metadata. Create this file with a text editor and call it **meta-data**. This file provides information that identifies the instance of Red Hat Enterprise Linux Atomic Host being installed. The **instance-id** can be any identifying name and the **local-hostname** should be a host name that follows your site standards, for example:

   ```
   instance-id: Atomic0
   local-hostname: atomic-00
   ```

2. Create cloud-init **user-data** file.
   The second installation configuration option file is the user data file. This file provides

information about users on the system. Create it with a text editor and call it **user-data**. This file will be used to enable access to the installation of Red Hat Enterprise Linux Atomic Host. By default, the root user is password locked and it is not possible to log in if this step is skipped. The following is an example of what the **user-data** file will look like:

```
#cloud-config
password: atomic
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
  - ssh-rsa AAA...SDvz user1@yourdomain.com
  - ssh-rsa AAB...QTuo user2@yourdomain.com
```

This **user-data** file enables the default user, **cloud-user**, to log in either with a password or with an SSH key. The use of both methods is possible but not required. Password login is enabled by the **password** and **chpasswd** lines. The password contains the plain-text password for the **cloud-user** user. The **chpasswd** line turns off password expiration to prevent the first login from immediately prompting for a change of password. This is optional. If you set a password, it is recommended that you change it when you first log in because the password has been stored in a plain text file.

SSH login is enabled by the last three lines of the file. The **ssh_pwauth** line enables SSH login. The **ssh_authorized_keys** line begins a block of one or more authorized keys. Each public SSH key listed on the **ssh-rsa** lines will be added to the cloud-user **~/.ssh/authorized_keys** file. In this example, two keys are listed. For this example, the key has been truncated, in a real file the entire public key must be listed. Note that the **ssh-rsa** lines must be preceded by two spaces, followed by a hyphen, followed by another space.

3. Create ISO file.
   Once you have completed your files, they need to be packaged into an ISO image. This ISO image is used as a virtual configuration CD with the virtual machine. This ISO image, called **atomic0-cidata.iso**, is created with the following command on Red Hat Enterprise Linux:

   ```
   # genisoimage -output atomic0-cidata.iso -volid cidata -joliet -rock user-data meta-data
   ```

4. Transfer the newly created ISP image to the host running VMware.

2.3.4.1. Setting up a Red Hat Enterprise Linux Atomic Host Virtual Machine in VMware*

The steps for running a Red Hat Enterprise Linux Atomic Host on a VMware vSphere client include the following:

1. Adding the ISO image you created earlier into your VMware vSphere data store.
2. Deploying your OVA file as an OVF template in vSphere.
3. Attaching the ISO image as a CD/DVD drive to the vSphere template.
4. Run the Red Hat Enterprise Linux Atomic Host virtual machine.

NOTE

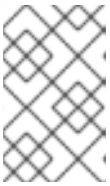
This procedure assumes you are familiar with VMware vSphere and is not written with reference to any specific version of VMware vSphere.

Add image to the Datastore

1. Open the VMware vSphere client.
2. In the left pane, access **Datastores**.
3. Select the target datastore.
4. Select **Browse this datastore**.
5. Select the folder icon and create a new folder. In this example, it is called **atomic01**/.
6. With the new folder **atomic01**/ highlighted, select the GUI option to upload data to the datastore (and to the folder).
7. Browse to the cloud-init ISO file you created earlier (for example, **atomic01-cid.iso**), select it, and upload it to the datastore. If an identically named file already exists in the datastore, you will be asked if you want to overwrite it.
8. Close the Datastore Browser.

Deploy OVF template

1. Select **Home**, then **Inventory**, then the **Hosts and Clusters** option.
2. Select **File and Deploy OVF Template**
3. Browse to the location where you have the OVA file, for example, **rhel-atomic-cloud-7.1-6.x86_64.vsphere.ova**, select it, and click **Open**.
4. Select the **Next** button. You see the OVF template details screen.
5. From the **OVF template details screen**, select **Next** again.
6. Type in the name for your Red Hat Enterprise Linux Atomic Host virtual machine.
7. Select a host or cluster for the virtual machine to run in and click **Next**.
8. Select the **Disk Format** option. You may leave the defaults. Then click **Next**.



NOTE

Be sure not to select the **Power on after deployment** check box. Selecting it will start the virtual machine and it should be started later after the cloud-init ISO has been attached.

9. Click **Finish** to begin deploying the template. This should take no more than two minutes.

Attach ISO image as a CD/DVD to Virtual Machine

1. Right-click on the newly added Red Hat Enterprise Linux Atomic Host template and select **Edit Settings**. (Select the **Virtual Machines** tab or expand the server in the Tree View in order to see the virtual machine.)
2. From the **Virtual Machine Properties** window, select **Add** and then **CD/DVD Drive** and click **Next**.

3. Select the **Use an ISO image** option and click **Next**.

4. Browse to find the ISO image you created earlier (we called ours **atomic0-cidata.iso**), select it, and click **Next**. The ISO can be found in the datastore that you uploaded it to, in the folder that you created.

5. After the **Advanced** options are displayed, click **Next** to continue.

6. When the **Ready to Complete** screen appears, click **Finish** to complete the settings. Now you are ready to run the Red Hat Enterprise Linux Atomic Host virtual machine.

7. Click **OK** to exit the **Properties** screen.

Run the Red Hat Enterprise Linux Atomic Host virtual machine

1. To start up the Red Hat Enterprise Linux Atomic Host virtual machine, click to highlight it, then select the **Power On** button.

2. Select the **Console** tab to watch as the virtual machine boots up.

If you configured Red Hat Enterprise Linux Atomic Host as described here, you should be able to log into the virtual machine with the user name **cloud-user** and password **atomic** that you defined when you created the cloud-init ISO.

2.3.5. Microsoft Hyper-V Installation

This section explains how to use Microsoft Hyper-V to create virtual machines that run Red Hat Enterprise Linux Atomic Host. Before you begin the installation process, make sure to download the installation media from the Download Red Hat Enterprise Linux page. The VHD image provided by Red Hat is a pre-deployed disk image which can be used to rapidly deploy Generation 1 Hyper-V virtual machines; alternatively you can use the Red Hat Enterprise Linux Atomic Host ISO installer, which allows for customized installations.

For full documentation of Microsoft Hyper-V, see the Hyper-V Getting Started section of the Microsoft TechNet Library.

Creating a Virtual Machine in Hyper-V

1. In the **Actions** menu, select **New**. Then, select **Virtual Machine** from the drop-down menu, and click **Next**. A new dialog window titled **New Virtual Machine Wizard** will open.

2. *Before You Begin*. Click **Next**.

3. *Specify Name and Location* . Name the new virtual machine, and click **Next**.

4. *Specify Generation*. Specify Generation 1 if you want to use the VHD disk image provided by Red Hat, or Generation 2 if you need to. (See Section 25.5.3, “Differences Between Generation 1 and Generation 2” for information about Generation 1 and Generation 2 virtual machines.)

5. Click **Next** to continue.

6. *Assign Memory*. Select how much memory should be assigned to the virtual machine, and click **Next**.

7. *Configure Networking*. In the **Connections** drop-down menu, select **external**. Then, click **Next**.

8. *Connect Virtual Hard Disk*. If you are using the VHD disk image provided by Red Hat, choose **Use an existing virtual hard disk** and then specify the location of the VHD file you have downloaded from Red Hat Customer Portal. Click **Next**.

9. **Summary**. Review your selections and click **Finish** to create the virtual machine.

Preparing for Installation

Once you run the Hyper-V image, you will be asked for login credentials. These can be preset using a pair of cloud-init files and you can also use the files to set other installation configuration options. The following is an example procedure:

- **meta-data**
  A plain text file which provides information that identifies the instance of Red Hat Enterprise Linux Atomic Host being installed. Its contents should be similar to the following example:

  ```
  instance-id: Atomic0
  local-hostname: atomic-00
  ```

  The **instance-id** can be any identifying name and the **local-hostname** should be a host name that follows your site standards.

- **user-data**
  A plain text file which provides information about users on the system. This information will be used to enable access to the Red Hat Enterprise Linux Atomic Host instance. By default, the **root** user is password protected; therefore, if you do not create the **user-data** file, you will not be able to log in.

  An example of a **user-data** file is below:

  ```
  #cloud-config
  password: atomic
  chpasswd: {expire: False}
  ssh_pwauth: True
  ssh_authorized_keys:
  - ssh-rsa AAA...SDvz user1@yourdomain.com
  - ssh-rsa AAB...QTuo user2@yourdomain.com
  ```

  NOTE

  The first line of the example (**#cloud-config**) is not a comment or a command example – it is a mandatory line in the configuration file.

  This example enables the **cloud-user** user to log in either with a password or an **SSH** key. The use of both methods is possible, but not required. An initial password is set on the **password** line; when the user logs in for the first time on this instance, they will be prompted to change their password as defined on the **chpasswd** line. Forcing the user to change their password after the first login is recommended because initially the password is stored in plain text.

  The final four lines in the example configure remote login using **SSH**. The **ssh_pwauth: True** line enables **SSH** using a password, and the **ssh_authorized_keys** starts a block of one or more authorized public keys. Keys described in this file will be added to the **~/.ssh/authorized_keys** file. Each authorized key must be on a separate line and start with two spaces followed by a hyphen (**-**) and another space.

Once you have created both of the files described above, you must package them into the ISO image. This image will then be used as a virtual configuration CD on the virtual machine. To package the files into an image, use the following command:

```
# genisoimage -output atomic0-cidata.iso -volid cidata -joliet -rock user-data meta-data
```

This will create a new ISO image file named **atomic0-cidata.iso**.

Differences Between Generation 1 and Generation 2

Microsoft Hyper-V has two different *generations* (also known as *modes*): Generation 1 and Generation 2. The differences between these generations have impact on the installation process of Red Hat Enterprise Linux Atomic Host.

Generation 1 disk images are supported on all Microsoft Hyper-V hosts. Generation 2 disk images are supported only on Microsoft Windows 2012 and Microsoft Windows 8.1.

Images provided by Red Hat fall into the Generation 1 category. These disk images allow for immediate deployment of preconfigured instances of Red Hat Enterprise Linux Atomic Host as described in Section 25.5.1, “Creating a Virtual Machine in Hyper-V”.

Preconfigured Generation 2 disk images are not provided by Red Hat. If you want to deploy Red Hat Enterprise Linux Atomic Host as a Generation 2 virtual machine, you can use the interactive installer ISO image and perform an installation using Anaconda (either manually or automatically using a Kickstart file). This process is described in earlier sections of this guide, starting with Chapter 6, Installing Using Anaconda; Kickstart installations are discussed in Chapter 23, Kickstart Installations.

2.3.6. Microsoft Azure Installation

Use this procedure to upload a RHEL Atomic Host image to the Microsoft Azure Cloud and run that image as a virtual machine. The basic steps to run from a RHEL server system are:

- Get the Azure CLI tool (az command) as described in Install the Azure CLI.
- Get the Red Hat Atomic Cloud (qcow2) image from Red Hat Atomic Host Download page.
- Convert the image to VHD format.
- Get and log into an Azure login account.
- Create the following Azure resources (or use existing ones):
  - Create a storage account
  - Create a container
  - Create a virtual network and subnetwork
- Upload the Atomic VHD image.
- Create a gold custom image (optional).
- Start a RHEL Atomic VM.
- Add the Azure agent to the VM (optional).

Replace the following resource names used in the example below with ones appropriate for your own setup.

Resource name	Examples
Azure group	myazgroup
Azure storage	myatomicstorage
Azure container	myatomiccontainer
Azure virtual network	myazatomicnet01
Azure subnetwork	myazatomicsubnet01
Atomic image	rhel-atomic-cloud-7.4.vhd
Azure region	southcentralus (use a region that suits you)
Atomic image in Azure	atomiccloud-74.vhd
Azure gold image group	myatomicgold

With an Azure account in hand, use the following procedure to create an Atomic virtual machine in Azure with that image.

1. Get the Azure CLI tool: Follow the instructions in Install the Azure CLI to get the **az** command.
2. Get the Red Hat Atomic Cloud (qcow2) image from Red Hat Atomic Host Download page.
3. Convert the image to VHD format as follows:

```
$ qemu-img convert -f qcow2 -o subformat=fixed,force_size -O vpc \
    rhel-atomic-cloud-7.4.4-2.x86_64.qcow2 rhel-atomic-cloud-7.4.vhd
```

NOTE

Azure requires that VHD images be fixed and aligned. The image described here should work properly. If the image fails to upload and run in a later step, check and fix its alignment as described in Convert the RHEL VM Image to VHD .

4. Log into the Azure Cloud:

```
$ az login

To sign in, use a web browser to open the page https://aka.ms/devicelogin and enter the code
ABCDEFGH9 to authenticate.
[
  {
    "cloudName": "AzureCloud",
```

```
...
    "user": {
      "name": "joe@example.com",
      "type": "user"
    }
  }
]
```

After following the instructions from your browser, close the browser window and continue from the command line.

5. Create Azure group resource: If you don't already have an Azure group, create one as follows:

```
$ az group create --name myazgroup --location southcentralus

{
  "id": "/subscriptions/xxxxxxxx-xxxx-xxxx.../resourceGroups/myazgroup",
  "location": "southcentralus",
  "managedBy": null,
  "name": "myazgroup",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

6. Choose an Azure region that is appropriate for you. Refer to Microsoft Azure Regions to see available regions, then type the following to see names you need to identify your region:

```
$ az account list-locations -o table
DisplayName      Latitude    Longitude   Name
----------       ----------  ----------- ----------
...
South Central US 29.4167     -98.5       southcentralus
...
```

7. Create Azure storage account: For your group, create a storage account, replacing southcentralus with your region and selecting a SKU Type:

```
$ az storage account create -l southcentralus -n myatomicstorage \
    -g myazgroup --sku Standard_LRS

{
  "accessTier": null,
  "creationTime": "2018-01-23T16:14:51.478598+00:00",
...
  "id": "/subscriptions/xxxxxxxx-
xxxx.../resourceGroups/myazgroup/providers/Microsoft.Storage/storageAccounts/myatomicstor
age",
  "name": "myatomicstorage",
...
  "provisioningState": "Succeeded",
  "resourceGroup": "myazgroup",
...
```

8. Get the storage account connection string:

```
$ az storage account show-connection-string -n myatomicstorage -g myazgroup

{
  "connectionString":
"DefaultEndpointsProtocol=https;EndpointSuffix=core.windows.net;AccountName=myatomicsto
age;AccountKey=xxxxxxxx/xxxxx+xxx/w=="
}
```

9. Export the connection string: Copy and paste your connection string into the AZURE_STORAGE_CONNECTION_STRING variable:

```
$ export
AZURE_STORAGE_CONNECTION_STRING="DefaultEndpointsProtocol=https;EndpointSuffi
x=core.windows.net;AccountName=myatomicstorage;AccountKey=xxxxxxxx/xxxxx+xxx/w=="
```

10. Create the storage container:

```
$ az storage container create -n myatomiccontainer

{
  "created": true
}
```

11. Create virtual network and subnetwork:

```
$ az network vnet create -g myazgroup -n myazatomicnet01 \
   --subnet-name myazatomicsubnet01

{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
        "10.0.0.0/16"
...
    "id": "/subscriptions/xxxxxxxx-
xxxx.../resourceGroups/myazgroup/providers/Microsoft.Network/virtualNetworks/myazatomicnet
01",
...
```

12. Upload the Atomic VHD image:

```
$ az storage blob upload --account-name myatomicstorage \
    --container-name myatomiccontainer --type page     \
    --file rhel-atomic-cloud-7.4.vhd --name myatomiccloud-74.vhd

Finished[#####################]  100.0000%
{
  "etag": "\"0x8D123456789ABCD\"",
  "lastModified": "2018-01-25T16:30:41+00:00"
}
```

13. Get the URL for the uploaded VHD:

```
$ az storage blob url -c myatomiccontainer -n myatomiccloud-74.vhd

"https://myatomicstorage.blob.core.windows.net/myatomiccontainer/myatomiccloud-74.vhd"
```

14. Create a new resource group for a gold Azure custom Atomic image (optional): This optional step keeps your gold image separate from any non-permanent resources you create. Your new resource group must be created in the same region where you uploaded your vhd file.

```
$ az group create --name myatomicgold --location southcentralus

{
  "id": "/subscriptions/xxxxxxxx-xxxx-.../resourceGroups/myatomicgold",
  "location": "southcentralus",
  "managedBy": null,
  "name": "myatomicgold",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

15. Create the gold Atomic custom image for Azure:

```
$ az image create -n myrhelatomcloud74 -g myatomicgold -l southcentralus \
  --source \
  "https://myatomicstorage.blob.core.windows.net/myatomiccontainer/myatomiccloud-74.vhd"
\
  --os-type linux

{
  "additionalProperties": {},
  "id": "/subscriptions/xxxxxxxx-xxxx-
.../resourceGroups/myatomicgold/providers/.../images/myrhelatomcloud74",
...
      "additionalProperties": {},
      "blobUri":
"https://myatomicstorage.blob.core.windows.net/myatomiccontainer/myatomiccloud-74.vhd",
...
}
```

16. Create a virtual machine: This example creates a running virtual machine named myatomic74vm-1. (NOTE: You could further configure this command line by creating a cloud-init script and adding it to the command line. For example: **--custom-data RHELCloudInit.yml**. See Cloud-Init Support for details.)

```
$ az vm create -g myatomicgold -l southcentralus -n myatomic74vm-1 \
    --size Standard_A2 --os-disk-name vm-1-osdisk    \
    --admin-username clouduser --generate-ssh-keys --image myrhelatomcloud74

{
  "fqdns": "",
  "id": "/subscriptions/xxxxxxxx-xxxx-
.../resourceGroups/myatomicgold/providers/Microsoft.Compute/virtualMachines/myatomic74vm-
1",
  "powerState": "VM running",
```

```
  "privateIpAddress": "10.0.0.5",
  "publicIpAddress": "49.82.154.297",
  "resourceGroup": "myatomicgold",
  "zones": ""
}
```

17. Log into the virtual machine: Note the publicIpAddress (49.82.154.297 in this fake address) and use it to log into the virtual machine:

```
$ ssh clouduser@49.82.154.297
The authenticity of host '49.82.154.297 (49.82.154.297)' can't be established.
ECDSA key fingerprint is bd:fe:12:1b:3c:d3:e2:4c:9f:b5:4a:87:10:48:5d:92.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '49.82.154.297' (ECDSA) to the list of known hosts.
[clouduser@myatomic74vm-1 ~]$
```

18. Subscribe the system: Use your Red Hat subscription account to subscribe the system and update to the latest version of Atomic:

```
$ sudo subscription-manager register
Registering to: subscription.rhsm.redhat.com:443/subscription
Username: yourusername
Password: ********
The system has been registered with ID: e4da51cb-4b89-3c94-30b5-946e5c222e91

$ sudo subscription-manager subscribe --auto
Installed Product Current Status:
Product Name: Red Hat Enterprise Linux Atomic Host
Status:       Subscribed
Product Name: Red Hat Enterprise Linux Server
Status:       Subscribed

$ atomic host upgrade

$ sudo reboot
```

19. Add the Azure agent (optional): If you need more advanced Azure functions and accuracy in the Azure Web portal, consider adding the Azure agent to the Atomic host. To do that, enable needed repositories, add the WALinuxAgent package, reboot, disable provisioning, and set up the Azure agent to start, as shown here:

```
$ ssh clouduser@41.89.184.287
[clouduser@myatomic74vm-1 ~]$

$ sudo subscription-manager repos --disable=*
$ sudo subscription-manager repos --enable rhel-7-server-rpms \
  --enable rhel-7-server-extras-rpms
$ sudo rpm-ostree install WALinuxAgent
$ sudo systemctl reboot
$ sudo vi /etc/waagent.conf
  # Enable instance creation
  Provisioning.Enabled=n
  # Create and use swapfile on resource disk.
  ResourceDisk.EnableSwap=y
  # Size of the swapfile.
```

```
  ResourceDisk.SwapSizeMB=2048    <-- or choose different size

$ sudo systemctl enable waagent
$ sudo systemctl start waagent
```

The Atomic virtual machine should now be running and accessible from your Azure dashboard.

2.3.7. Google Compute Engine Installation

Google Compute Engine (GCE) is a service that provides virtual machines that run on Google infrastructure. This document shows how to run Red Hat Enterprise Linux Atomic Host on GCE.

Red Hat Enterprise Linux Atomic Host has been designed to take advantage of the heritage of powerful technology available in Red Hat Enterprise Linux 7, in a variation of Red Hat Enterprise Linux 7 optimized for Linux containers that run using the Docker engine. Google Compute Engine (GCE) is a service that provides virtual machines (VMs) that run on Google infrastructure. These VMs can be used for running Red Hat Enterprise Linux Atomic Host. This sections explains how to start a virtual machine instance of Red Hat Enterprise Linux Atomic Host on GCE.

If you an interested in more details, refer to The official documentation for Google Compute Engine .

2.3.7.1. Enabling Google Compute Engine

Creating a Project and Setting Up Billing

Perform the following steps to create a project and set up billing for Google Compute Engine:

1. Log into your Google account, go to the Google Developers Console at https://console.developers.google.com/project. The Developers Console provides a list of projects that are available to you.

2. Select the project you wish to enable. If you want to create a new project, click on the red **Create Project** button. You are prompted to select the project name and ID. If your project belongs to a specific domain, your project ID would be in the form **\<domain\>:\<your-chosen-project-id\>**. Then, you are directed to the project dashboard.

3. To activate Google Compute Engine, set up billing by clicking on the **Billing & Settings** menu item on the right bar. Then click on **Enable Billing**. Fill in the form that appears afterwards. Google Compute Engine will prompt you to set up billing before you can use the service. It is not possible to use Google Compute Engine without activating billing. Note that after activating, your account may take about five minutes to be ready.

Downloading and Setting Up GCE tools

To manage your Google Compute Engine resources, first download and install the gcloud command-line tool:

1. Execute the following command to install the Google Cloud SDK:

```
$ curl https://sdk.cloud.google.com | bash
```

2. During the installation, you will be prompted several times to provide necessary information. First, you are asked to specify a destination directory for Google Cloud SDK:

```
Directory to extract under (this will create a directory google-cloud-sdk) (/home/user):
```

3. Then you are asked whether you wish to allow usage reporting to Google so that they can use this data to improve the tool.

4. The Google Cloud SDK is then installed. Afterwards, several prompts for configuring your profile follow. You can specify an **rc** file, change the **$PATH** variable, and enable bash completion. Adding these programs to your **$PATH** variable is good because it allows you to run them without having to provide their full path. Enabling bash completion is also helpful because the command consists of multiple arguments that are easier to type with completion.

5. Restart your terminal to allow changes to your PATH to take affect. For example, you can use:

```
$ source ~/.bash-profile-file
```

6. Replace **bash-profile-file** with a path to your bash profile file. This is typically the **~/.bashrc file**.

Authenticating to GCE

Authenticate to the Google Cloud platform by running:

```
$ gcloud auth login
```

The above command launches a web browser with a sign-up dialog for your Google account. Sign in to proceed. During the sign-in process you will need to allow Google Compute Engine to access some information about your Google Account. It is possible to authenticate without launching the browser by using the **--no-launch-browser** option, see https://cloud.google.com/compute/docs/gcloud-compute/#auth for details.

Setting Up Project Defaults

Using the command template, **gcloud config set default default_value** it is possible to set project defaults so that command options for commonly used flags do not have to be passed to every command. To list the current defaults execute the **gcloud config list** command. The template, **gcloud config unset default** will remove a project default. Execute the following command to set the default project:

```
$ gcloud config set project project_id
```

Where *project_id* stands for the id of the project you created in *Creating a Project and Setting Up Billing* .

Execute the following command to set the default zone:

```
$ gcloud config set compute/zone zone
```

Where **zone** determines a geographical location where your instance should live. See https://developers.google.com/compute/docs/zones#available for a list of available zones.

2.3.7.2. Starting a Red Hat Enterprise Linux Atomic Host Instance

Before the Red Hat Enterprise Linux Atomic Host image can be used in GCE, it needs to be transformed from a qcow2 file into a RAW image. This is done by downloading the qcow2 file and then transforming it into a tar file. This file is uploaded to GCE and then an instance is created.

Creating a Red Hat Enterprise Linux Atomic Host RAW File

Perform the following steps to create a RAW file that can be uploaded to GCE.

1. Download the Red Hat Enterprise Linux Atomic Host qcow2 file from Download Red Hat Enterprise Linux.

2. The qcow2 image has been compressed with **xz**. To decompress the image, enter the following command:

```
$ xz -d rhel-atomic-cloud-7.1-0.x86_64.qcow2.xz
```

1. The qcow2 image must be converted into a RAW disk file in order to used in GCE. This is done with qemu.

```
$ qemu-img convert -S 4096 -f qcow2 -O raw rhel-atomic-cloud-7.1-0.x86_64.qcow2 disk.raw
```

1. The raw disk file needs to be packaged with tar prior to being uploaded to GCE. The raw file has to be named disk.raw.

```
$ tar -Szcf rhel-atomic-cloud-7.1-0.x86_64.tar.gz disk.raw
```

1. The uploaded raw disk file will be stored in a Google Cloud Storage bucket. If you do not already have a bucket created, you can create one using **gsutil**.

```
$ gsutil mb gs://bucketname
```

1. Upload the raw disk file using **gsutil**.

```
$ gsutil cp rhel-atomic-cloud-7.1-0.x86_64.tar.gz gs://bucketname
```

1. Before you can use the raw disk file, it has to be created as a GCE image.

```
$ gcloud compute images create GCE_IMAGE_NAME --source-uri gs://bucketname/rhel-atomic-cloud-7.1-0.x86_64.tar.gz
```

1. You can verify the image is uploaded and available by looking for it in the output of **gcloud compute images list**.

Creating a Red Hat Enterprise Linux Atomic Host Instance

Execute the following command to create an Atomic Host Instance:

```
$ gcloud compute instances create my-atomic-instance --machine-type n1-standard-1 --image GCE_IMAGE_NAME --metadata-from-file startup-script=<your-statup-script>
```

where:

my-atomic-instance is a name of an instance for this example. Instance names can contain only lowercase letters, digits, and dashes (except the last character, which cannot be a dash).

--machine-type sets your desired machine types. A machine type determines the memory, number of virtual cores, and persistent disk limits that your instance will have. Refer to https://developers.google.com/compute/docs/machine-types for details.

--image sets the image to be used. An image contains the operating system and root file system that is necessary for starting an instance. GCE automatically creates a root persistent disk to store the root file system. The GCE_IMAGE_NAME is the image you created in the previous step.

--metadata-from-file specifies the metadata to be made available in the instance environment through the local metadata server. Use this flag to specify a script to be executed automatically when the Red Hat Enterprise Linux Atomic Host instance launches for the first time. See "Executing a Custom Script on Instance Creation" section for more information. Note that the **user-data** key is required and cannot be replaced with a custom key, since the startup scripts for Red Hat Enterprise Linux Atomic Host instance are processed by the **cloud-init** utility and not by the GCE agent.

NOTE

This command blocks until the instance is running. When the instances is first created, it must boot and then self-configure. This takes a few moments and may delay your ability to log in to the instance.

Executing a Custom Script on Instance Creation

As mentioned above, you can use the **--metadata-from-file** option when creating the instance to a specify custom script to be executed in that instance on its first start. You can run any system commands necessary from this script, as these commands are executed with root permissions. For example:

```
--metadata-from-file startup-script=<your-startup-script>
```

Invokes the startup.sh script with the following content:

```
#! /bin/sh
touch newfile
```

This line creates a new file called **newfile**.

NOTE

The startup script for Red Hat Enterprise Linux Atomic Host instance is not processed by the GCE agent, but by the **cloud-init** utility. Therefore, you cannot use custom keys with **--metadata-from-file**. Always use the **user-data** key when configuring custom script for Red Hat Enterprise Linux Atomic Host instance.

As an alternative to locally-stored startup script, you can upload your script to Google Cloud Storage and then access it with the **--metadata** option. This is required if your script exceeds the metadata value length limit of 32,768 bytes. See http://developers.google.com/compute/docs/howtos/startupscript#googlecloudstorage for more details.

2.3.7.3. Logging into a Red Hat Enterprise Linux Atomic Host Instance

The **gcloud** tool has a built-in ssh command that enables you to log into an instance using the instance name.

To log into your instance, execute the following command:

```
$ gcloud compute ssh cloud-user@my-atomic-instance
```

Here, **cloud-user** is the default user name. If you have not yet created an SSH key, you will be prompted to create one. Further information is available in Password Protecting Your SSH Keys.

NOTE

For security reasons, the standard Google images do not provide the ability to connect using SSH directly as root. The instance creator and any users that were added using the **--authorized_ssh_keys** flag or the metadata **sshKeys** value are automatically administrators to the account, with the ability to run **sudo** without requiring a password. Although it is not recommended, advanced users can modify **/etc/ssh/sshd_config** and restart **sshd** to change this policy.

WARNING

GNOME users can occasionally see the message

+

```
Agent admitted failure to sign using the key
```

+ when trying to connect to the GCE instance trough SSH. This is caused by the GNOME keyring management that tries to use a wrong SSH key. It is specific to the rhel-atomic-host-20141111 image for the GCE environment.

+ To work around this problem, enter the following command before executing **gcutil**:

+

```
$ ssh-add ~/.ssh/google_compute_engine
```

Once you have logged in, you can work as you would on other Red Hat Enterprise Linux machines. You have root permissions on your instance and full control over everything. To become root, execute:

```
cloud-user@my-atomic-instance$ sudo -i
```

If you need to log out of your instance, you can execute the following command:

```
cloud-user@my-atomic-instance$ exit
```

Once you have installed Red Hat Enterprise Linux Atomic Host, it is ready to run Linux containers.

Password Protecting Your SSH Keys

The first time you log into an instance with SSH, gcloud creates an ssh public/private key pair on your local machine, and copies the public key to your project. These keys are needed to log into your instance using ssh. When creating these keys for the first time, gcutil will ask you to enter and confirm a passphrase:

```
WARNING: You don't have an ssh key for Google Compute Engine. Creating one now...
Enter passphrase (empty for no passphrase):
```

Although you can leave the passphrase empty, we highly recommend entering a passphrase to protect your SSH keys. You will rarely be asked to enter your passphrase, and if you do not password protect these keys, a malicious user could use them to access your instances as you.

2.3.7.4. Monitoring a Red Hat Enterprise Linux Atomic Host Instance

The Google Cloud SDK provides several ways to monitor parameters of your instances. To view general information about the current gcloud environment, run:

```
$ gcloud info
```

Execute the **describe** command to find detailed information about a specific instance:

```
$ gcloud compute instances describe my-atomic-instance
canIpForward: false
creationTimestamp: '2014-11-11T02:15:58.372-08:00'
disks:
- autoDelete: true
  boot: true
  deviceName: persistent-disk-0
  index: 0
  interface: SCSI
  kind: compute#attachedDisk
  mode: READ_WRITE
  source: https://www.googleapis.com/compute/v1/projects/eighth-saga-761/zones/europe-west1-
b/disks/my-atomic-instance2
  type: PERSISTENT
id: '6632858316955862880'
kind: compute#instance
machineType: https://www.googleapis.com/compute/v1/projects/eighth-saga-761/zones/europe-
west1-b/machineTypes/n1-standard-1
metadata:
  fingerprint: owFsCDPRlkY=
  kind: compute#metadata
name: my-atomic-instance2
networkInterfaces:
- accessConfigs:
  - kind: compute#accessConfig
 name: external-nat
 natIP: 23.251.142.75
 type: ONE_TO_ONE_NAT
  name: nic0
  network: https://www.googleapis.com/compute/v1/projects/eighth-saga-761/global/networks/default
  networkIP: 10.240.184.150
scheduling:
  automaticRestart: true
  onHostMaintenance: MIGRATE
selfLink: https://www.googleapis.com/compute/v1/projects/eighth-saga-761/zones/europe-west1-
b/instances/my-atomic-instance2
serviceAccounts:
- email: 464767924601-compute@developer.gserviceaccount.com
  scopes:
  - https://www.googleapis.com/auth/devstorage.read_only
status: RUNNING
```

```
tags:
  fingerprint: 42WmSpB8rSM=
zone: https://www.googleapis.com/compute/v1/projects/eighth-saga-761/zones/europe-west1-b
```

To get data from the serial port of your Red Hat Enterprise Linux Atomic Host instance, run:

```
$ gcloud compute instances get-serial-port-output my-atomic-instance
```

This command returns the output of the GCE instance's serial port. With this command, you get information about the instance without logging into it, which is useful for diagnostic purposes.

Finding the External IP Address of an Instance

By default, your instance is assigned a new ephemeral external IP address. You can to find this address along with other information in the output of **gcutil getinstance** shown above. Alternatively, you can enter the following command to get addresses of all your instances:

```
$ gcloud compute instances list
NAME              ZONE          MACHINE_TYPE  INTERNAL_IP    EXTERNAL_IP   STATUS
my-atomic-instance  us-central1-a  n1-standard-1 10.240.184.150 23.251.142.75 RUNNING
```

2.3.7.5. Creating a Firewall Rule

By default, Google Compute Engine blocks all connections to and from an instance to the Internet. To open ports for services like **httpd**, you must manually create a firewall rule. Every project comes with three default firewalls:

1. A firewall that allows SSH access to any instance.
2. A firewall that allows all communication between instances in the same network.
3. A firewall that allows ICMP traffic from any source to any instance on the network.

For example, to permit HTTP requests to your instance, create a new firewall using the following **gcutil** command:

```
$ gcloud compute firewall-rules create http-allow --allow tcp:80
```

By executing the above command, you have:

1. Created a new firewall named **http-allow** that allows port 80 tcp traffic.
2. Assigned the firewall to the default network in the project.
3. Allowed all sources inside and outside the network (including over the Internet) to make requests to the server. We did not specify a permitted source for the firewall, so all sources are allowed to make requests to instances assigned to the default network.
4. Applied this firewall rule to all instances on the network. Because we did not specify a target for your firewall, the firewall applies this rule to all instances in the network.

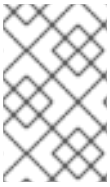
To review information about your firewall, run:

```
$ gcloud compute firewall-rules list
NAME              NETWORK SRC_RANGES    RULES                 SRC_TAGS TARGET_TAGS
```

```
default-allow-icmp     default 0.0.0.0/0     icmp
default-allow-internal default 10.240.0.0/16 tcp:1-65535,udp:1-65535,icmp
default-allow-rdp      default 0.0.0.0/0     tcp:3389
default-allow-ssh      default 0.0.0.0/0     tcp:22
http-allow             default 0.0.0.0/0     tcp:80
```

It is possible to restrict the sources and targets to specific callers and instances using appropriate **addfirewall** flags. To see a complete list of supported flags, run the command **gcutil help addfirewall**, or see https://cloud.google.com/sdk/gcloud/reference/compute/firewall-rules/.

Firewalls only regulate incoming traffic to an instance; they cannot block outgoing packets. Once a connection has been established with an instance, traffic is permitted in both directions over that connection. To prevent an instance from sending outgoing packets, use another technology such as **iptables**.



NOTE

By default, GCE drops TCP connections to instances after 10 minutes of inactivity. To prevent this, configure TCP keep-alives as described in https://developers.google.com/compute/docs/troubleshooting#communicatewithinternet

2.3.7.6. Removing a Red Hat Enterprise Linux Atomic Host Instance

Execute the following command to remove **my-atomic-instance**:

```
$ gcloud compute instances delete my-atomic-instance
```

You are prompted to confirm your decision before the instance is deleted. Deleting the instance may take several seconds time. As a part of the deletion confirmation dialog, gcloud informs you that disks will be deleted unless also used by another instance.

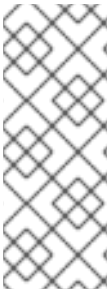
2.3.8. Amazon Web Services Installation

Amazon Web Services (AWS) is a service that provides virtual machines that run on Amazon infrastructure. This section shows how to run Red Hat Enterprise Linux Atomic Host on AWS.

For more details about AWS, refer to Amazon Elastic Compute Cloud Documentation.

Launching a Red Hat Enterprise Linux Atomic Host Instance on Amazon Web Services

The following procedure will guide you through creating a new instance of Red Hat Enterprise Linux Atomic Host on Amazon Web Services. The procedure assumes that you already have a user account on AWS. This procedure assumes some familiarity with AWS.



NOTE

In order for this procedure to work, you must first have moved your subscriptions to Amazon through the Cloud Access Program. To move your subscriptions to Amazon through the Cloud Access Program complete this form: https://engage.redhat.com/forms/cloud-access-registration. The Cloud Access Program is described in greater detail at http://www.redhat.com/en/technologies/cloud-computing/cloud-access.

1. Log in to and open the Amazon EC2 console.

2. In the navigation bar at the top of the screen, the current region is displayed. Select the region in which you wish to launch your instance of Red Hat Enterprise Linux Atomic Host. This choice is important because some Amazon EC2 resources can be shared between regions, while others cannot.

3. From the console dashboard, click **Launch Instance**.

4. Select **My AMIs** and select the **Shared with Me** check box. It is now possible to search for the AMI.
   Choose **Community AMIs** and search for the Red Hat Enterprise Atomic Host AMI instance for your particular region.

WARNING

Make sure that the ID of AMI you choose is listed in the Atomic Host Release Notes. You can also get IDs of AMIs supplied by Red Hat by running this command in the AWS Command Line Interface:

```
aws ec2 describe-images --owners 309956199498
```

The command shows information about AMIs published by account **309956199498**, which is Red Hat's AWS account for publishing AMIs.

For details on searching for AMIs provided by Red Hat, see this Knowledgebase Article.

5. Click the **Select** button next to the AMI.

6. On the **Choose an Instance Type** page, select your Instance Type. The Instance Type should meet the minimum requirements for Red Hat Enterprise Linux Atomic Host. See Disk Space and Memory Requirements for more information.

7. Click **Review** and **Launch**.

NOTE

In some Amazon EC2 regions, for example, US East (N. Virginia), Instance Types that use EBS storage require the creation of a VPC before they can be launched. In those cases, **Review and Launch** is not clickable. Click **Next: Configure Instance Details** instead and proceed to the Instance Details screen. Review the defaults and modify them if necessary for your environment, and click **Review and Launch** when ready to proceed.

8. On the **Review Instance Launch** page, assign a security group by clicking **Edit security groups**. You should select an existing security group or create one that opens the ports you will need for your instance. It is encouraged to leave port 22 open so that SSH will work. AWS accounts can be set up in a manner that restricts the ability of users of that account to create or add security groups. If this occurs, contact the administrator of the AWS account.

9. When you are satisfied with the settings, click **Review and Launch** to go to the **Review Instance Launch** page. Once you are satisfied with all settings, click **Launch** to start your instance.

10. In the **Select an existing key pair or create a new key pai**modal dialog, select an existing key pair or create a new one. A key pair is critical as all access to your launched instance is through private SSH key. The key pair is either one that you have already uploaded or one that you will create at this moment. AWS accounts can be set up in a manner that restricts the ability of users of that account to create or add key pairs. If this occurs, contact the administrator of the AWS account.

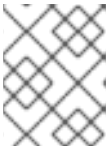
11. Click the **View Instances** button to track the progress of your instances launch.

Logging into a Red Hat Enterprise Linux Atomic Host Instance

Once your instance is listed as **running**, you may connect to it by following the steps below.

1. From your command prompt, connect to the instance using SSH.

```
$ ssh cloud-user@instancedns.compute.amazonaws.com
```



NOTE

You may need to include the **-i /path/key_pair.pem** option to specify the proper private key file.

2. In the **Description** tab at the bottom, locate the **Public DNS** information.

3. On the **Instances** page, select your instance.

4. At this point you are logged into your instance and may continue working with Red Hat Enterprise Linux Atomic Host and run Linux containers.

Verifying authenticity of an Atomic Host instance on AWS

You can verify that an Atomic Host instance is the authentic software provided by Red Hat. To do this, run this command on the Atomic Host instance:

```
ostree show rhel-atomic-host/7/x86_64/standard
```

If the last line of output is this:

```
Good signature from "Red Hat, Inc. <security@redhat.com>"
```

Then your Atomic Host instance has passed the verification.

2.4. PXE INSTALLATION

Configuring a PXE server to boot Red Hat Enterprise Linux Atomic Host from it does not differ from the standard procedure for Red Hat Enterprise Linux. You can use the detailed instructions in the, Preparing for a Network Installation chapter from the Red Hat Enterprise Linux Installation Guide.

Here is an example entry for Atomic for the **/var/lib/tftpboot/pxelinux/pxelinux.cfg/default** file:

```
label Atomic-7.3
```

```
  menu label ^1. RHEL Atomic Host 7.3 kickstart
  kernel atomic7.3/vmlinuz
  append initrd=atomic7.3/initrd.img inst.stage2=http://192.168.122.1/distros/atomic xdriver=vesa
nomodeset quiet ks=http://192.168.122.1/ks/atomic.ks
```

Make sure the kernel, inird image, installation program runtime image (**inst.stage2**), and the Kickstart file are present in the locations that are specified.

CHAPTER 3. SETTING UP CLOUD-INIT

Red Hat Enterprise Linux Atomic Host uses cloud-init to configure the system during installation and first-boot. Cloud-init was initially developed to provide early initialization of cloud instances. In Red Hat Enterprise Linux Atomic Host it can also be used for virtual machine installations.

The files used by cloud-init are YAML formatted files.

NOTE

cloud-init is run only the first time that the machine is booted. If cloud-init fails because of syntax errors in the file or doesn't contain all of the needed directives, such as user credentials, a new instance must be created and launched. Restarting the failed instance with a new cloud-init file will **not** work.

Here are some examples of how to do common tasks with cloud-init.

- How do I create users with cloud-init?
  To create users with cloud-init, you must create two files: **meta-data** and **user-data**, and then package them into an ISO image.

  1. Make a directory and move into it:

     ```
     $ mkdir cloudinitiso
     $ cd cloudinitiso
     ```

  2. Create a file called meta-data. Add the following to the file called meta-data:

     ```
     instance-id: Atomic0
     local-hostname: atomic-00
     ```

  3. Create a file called user-data. Add the following to the file called user-data:

     ```
     #cloud-config
     password: atomic
     chpasswd: {expire: False}
     ssh_pwauth: True
     ssh_authorized_keys:
       - ssh-rsa AAA...SDvZ user1@domain.com
     ```

     Note: The final line of the **user-data** file above is an SSH public key. SSH public keys are found in **~/.ssh/id_rsa.pub**.

  4. Create an ISO image that includes **meta-data** and **user-data**:

     ```
     # genisoimage -output atomic0cidata.iso -volid cidata -joliet -rock user-data meta-data
     ```

  5. A file named **atomic0cidata.iso** is generated. Attach this file to the machine on which you plan to install Red Hat Enterprise Linux Atomic Host, and your username will be "cloud-user" and your password will be "atomic".

- How do I expire the cloud-user's password so that the user must change it during their first login?

To force "cloud-user" to change their password at first login, change the line **chpasswd: {expire: False}** to **chpasswd: {expire: True}** in the **user-data** file.

```
#cloud-config
password: atomic
chpasswd: {expire: True}
ssh_pwauth: True
ssh_authorized_keys:
  - ssh-rsa AAA...SDvz user1@yourdomain.com
  - ssh-rsa AAB...QTuo user2@yourdomain.com
```

This works because the password and chpasswd operate on the default user unless otherwise indicated.

Note: This is a global setting. If you set this to True all users who are created (see below) will have to change their password.

- How do I change the default username?
  To change the default username from cloud-user to something else, add the line **user: username** to the **user-data** file:

```
#cloud-config
user: username
password: atomic
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
  - ssh-rsa AAA...SDvz user1@yourdomain.com
  - ssh-rsa AAB...QTuo user2@yourdomain.com
```

- How do I set the root password?
  To set the root password you must create a user list in the **chpasswd** section of the user-data file. The format of the list is shown below. Whitespace is significant, so do not include any on either side of the colon (**:**) as it will set a password with a space in it. If you use this method to set the user passwords, **all passwords** must be set in this section. This means that the **password:** line must be moved from the top and into this section.

```
#cloud-config
ssh_pwauth: True
ssh_authorized_keys:
  - ssh-rsa AAA...SDvz user1@yourdomain.com
  - ssh-rsa AAB...QTuo user2@yourdomain.com
chpasswd:
  list: |
     root:password
     cloud-user:atomic
  expire: False
```

- How do I manage Red Hat subscriptions with cloud-init?
  The **rh_subscription** directive can be used to perform various operations concerning registering your system (for RHEL Atomic 7.4 and later). Following are a few examples showing different available options:

```
rh_subscription:
```

```
  username: atomic@redhat.com
  password: '<password>'
  auto-attach: True
  service-level: self-support
```

Note that service-level is only used with the auto-attach option. Alternatively, you can use an activation key and org instead of username and password:

```
rh_subscription:
  activation-key: example_key
  org: 12345
  auto-attach: True
```

There is also support for adding pools. The following is the equivalent of the **subscription-manager attach --pool=XYZ01234567** command:

```
rh_subscription:
  username: atomic@redhat.com
  password: '<password>'
  add-pool: XYZ01234567
```

You can set up the server hostname in */etc/rhsm/rhsm.conf* with the following:

```
rh_subscription:
  username: atomic@redhat.com
  password: '<password>'
  server-hostname: atomic.example.com
  auto-attach: True
```

- How do I add more users during initial system configuration? How do I set additional user options?
  Users are created and described in the users section of the user-data file. Adding this section requires that options for the default user be set here as well.

  If the first entry in the *users* section is **default**, the default user, *cloud-user* will be created along with the other users. If the default line is omitted, then *cloud-user* is not created.

```
#cloud-config
users:
  - default
  - name: foobar
    gecos: User N. Ame
    selinux-user: staff_u
    groups: users,wheel
    ssh_pwauth: True
    ssh_authorized_keys:
      - ssh-rsa AA..vz user@domain.com
chpasswd:
  list: |
    root:password
    cloud-user:atomic
    foobar:foobar
  expire: False
```

Note: By default users will be labeled as unconfined_u if there is not an *se-linux-user* value.

Note: This example places the user *foobar* into two groups: **users** and **wheel**. As of cloud-init 0.7.5, no whitespace is supported in the group list: BZ 1126365

- How do I run first boot commands?
  The **runcmd** and **bootcmd** sections of the *user-data* file can be used to execute arbitrary commands during startup and initialization. The **bootcmd** section is run early in the initialization process. The **runcmd** section is executed near the end of the process by init. These commands are **not** saved for future boots and will only be executed during the first initialization-boot.

```
#cloud-config
users:
  - default
  - name: foobar
    gecos: User N. Ame
    groups: users
chpasswd:
  list: |
    root:password
    fedora:atomic
    foobar:foobar
  expire: False
bootcmd:
 - echo New MOTD >> /etc/motd
runcmd:
 - echo New MOTD2 >> /etc/motd
```

- How do I add additional sudoers?
  A user can be configured as a sudoer by adding a sudo and groups entry to the users section of the user-data file, as shown below.

```
#cloud-config
users:
  - default
  - name: foobar
    gecos: User D. Two
    sudo: ["ALL=(ALL) NOPASSWD:ALL"]
    groups: wheel,adm,systemd-journal
    ssh_pwauth: True
    ssh_authorized_keys:
      - ssh-rsa AA...vz user@domain.com
chpasswd:
  list: |
    root:password
    cloud-user:atomic
    foobar:foobar
  expire: False
```

- How do I set up a static networking configuration?
  Add a **network-interfaces** section to the *meta-data* file. This section contains the usual set of networking configuration options.

Because of a current bug in cloud-init, static networking configurations are not automatically started. Instead the default DHCP configuration remains active. A suggested work around is to manually stop and restart the network interface via the **bootcmd** directive.

```
network-interfaces: |
  iface eth0 inet static
  address 192.168.1.10
  network 192.168.1.0
  netmask 255.255.255.0
  broadcast 192.168.1.255
  gateway 192.168.1.254
bootcmd:
  - ifdown eth0
  - ifup eth0
```

- How do I delete cloud-user and just have root and no other users?
  To have only a root user created, create an entry for root in the **users** section of the *user-data* file. This section can be as simple as just a **name** option:

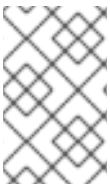
```
users:
  - name: root
chpasswd:
  list: |
    root:password
  expire: False
```

  Optionally, you can set up SSH keys for the root user as follows:

```
users:
  - name: root
    ssh_pwauth: True
    ssh_authorized_keys:
      - ssh-rsa AA..vz user@domain.com
```

- How do I set up storage with container-storage-setup?
  To set up the size of the root logical volume to 6GB for example instead of the default 3GB, use the **write_files** directive in *user-data*:

```
write_files:
  - path: /etc/sysconfig/docker-storage-setup
    permissions: 0644
    owner: root
    content: |
    ROOT_SIZE=6G
```



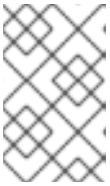
NOTE

Prior to RHEL 7.4, **container-storage-setup** was called **docker-storage-setup**. If you are using OverlayFS for storage, note that as of RHEL 7.4 you can now use that type of filesystem with SELinux in enforcing mode.

- How do I enable the Overlay Graph Driver?

The Overlay Graph Driver is enabled through *container-storage-setup*. Use the **runcmd** directive to change the STORAGE_DRIVER option to "overlay2":

```
runcmd:
  - echo "STORAGE_DRIVER=overlay2" >> /etc/sysconfig/docker-storage-setup
```



NOTE

Note that changing the backend storage driver is a destructive operation. Furthermore, OverlayFS is not POSIX-compliant and it can be used with restrictions. For more information, see RHEL 7.2 Release Notes .

- How do I re-run cloud-init on an instance?
  In most situations it is not possible to re-run cloud-init to change the configuration of a virtual machine that has already been created.

  When cloud-init is used in an environment where the Instance ID can be changed (for instance, from **Atomic0** to **Atomic1**), it is possible to re-configure an existing virtual machine **by changing the Instance ID and rebooting to re-run cloud-init**. This is not recommended practice for production environments because cloud-init is supposed to be set up to create on first boot systems that are fully and properly configured.

  In most IAAS implementations it is not possible to change the Instance ID. If cloud-init must be re-run, the instance should be cloned in order to obtain a new Instance ID.

- Can I put shell scripts in bootcmd and runcmd?
  Yes. If you use a list value for **bootcmd** or **runcmd**, each list item is run in turn using **execve**. If you use a string value, then the entire string is run as a shell script. Alternatively, if you want simply to use cloud-init to run a shell script, you can provide a shell script (complete with shebang (#!)) instead of providing cloud-init with a '.yaml' file.

See this website for examples of how to put shell scripts in **bootcmd** and **runcmd**.

CHAPTER 4. POST INSTALLATION CONFIGURATION

Red Hat Enterprise Linux Atomic Host is configured using the configuration files in the **/etc/** directory. This is similar to the way that Red Hat Enterprise Linux 7 is configured. Because Red Hat Enterprise Linux Atomic Host is a minimal server product that has no desktop, the graphical configuration tools found in the GUI are not available.

4.1. CONFIGURING NETWORKING

If you did not configure networking during the installation you may configure it post-installation using the *nmcli* tool. The following commands create a network connection called **atomic**, set up a host name, and then activate that connection.

```
# nmcli con add type ethernet con-name atomic ifname eth0
# nmcli con modify my-office my-office ipv4.dhcp-hostname atomic ipv6.dhcp-hostname atomic
# nmcli con up atomic
```

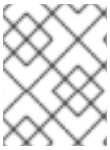
For more details on how to use the *nmcli* tool, see Section 2.3.2. Connecting to a Network Using nmcli in the Red Hat Enterprise Linux 7 Networking Guide.

4.2. REGISTERING RHEL ATOMIC HOST

To enable software updates, you must register your Red Hat Enterprise Linux Atomic Host installation. This is done with the **subscription-manager** command as described below. By default, **subscription-manager** assumes that your system has Internet access with the ability to reach Red Hat software repositories. If you don't have direct Internet access, you have a couple of options:

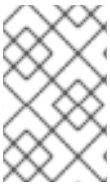
- **HTTP proxy**: If your system is located on a network that requires the use of an HTTP proxy, see the Red Hat Knowledge Base Article on configuring subscription manager to use an HTTP proxy . The **--name=** option may be included if you wish to provide an easy-to-remember name to be used when reviewing subscription records.
- **Internal ostree mirror**: If your location has no Internet access (even via an HTTP proxy), you can mirror the official Red Hat Enterprise Linux Atomic Host ostree repository for use in a way that is local to your environment. A procedure for setting up such a mirror is available in Using Atomic Host as an internal ostree mirror.

```
$ sudo subscription-manager register --username=<username> --auto-attach
```



NOTE

Red Hat Enterprise Linux Atomic Host works only with Red Hat Subscription Manager (RHSM). Red Hat Enterprise Linux Atomic Host does not work with RHN.



NOTE

Red Hat Enterprise Linux Atomic Host registers two product IDs. The first is Product ID 271, Red Hat Enterprise Linux Atomic Host. The second is Product ID 69, Red Hat Enterprise Linux Server. They both use the same entitlement.

A properly registered system will display both IDs as is shown below:

```
$ sudo subscription-manager list
+-------------------------------------------+
   Installed Product Status
+-------------------------------------------+
Product Name:   Red Hat Enterprise Linux Atomic Host
Product ID:     271
Version:        7
Arch:           x86_64
Status:         Subscribed
Status Details:
Starts:         02/27/2015
Ends:           02/26/2016

Product Name:   Red Hat Enterprise Linux Server
Product ID:     69
Version:        7.1
Arch:           x86_64
Status:         Subscribed
Status Details:
Starts:         02/27/2015
Ends:           02/26/2016
```

The **subscription-manager** command is also documented in Registering from the Command Line of the Red Hat Subscription Management guide.

4.3. MANAGING USER ACCOUNTS

Currently, some system users that in Red Hat Enterprise Linux 7 would be listed in the **/etc/passwd** file have been relocated into the read-only **/usr/lib/passwd** file. Because applications on Red Hat Enterprise Linux Atomic Host are run inside Linux containers, this does not affect deployment. The traditional user management tools, such as **useradd**, write locally added users to the **/etc/passwd** file as expected.

CHAPTER 5. UPGRADING AND DOWNGRADING

5.1. SETTING UP AN ATOMIC COMPOSE SERVER

This procedure explains how to set up an Atomic Compose server. It is possible to use an Atomic Compose server to create atomic update trees. The procedure here explains how to set up an Atomic Compose server that creates a local mirror of the upstream OSTree repository.

1. Log into a shell on the host, and run the Atomic Tools container.

```
# atomic run rhel7/rhel-tools
```

2. From inside the tools container, create an unprivileged user.

```
# adduser container
```

3. Acquire the entitlement certificates and use **chown** to make them owned by the unprivileged container user.

```
# cd ~container
# cp /host/etc/pki/entitlement/*.pem .
# chown container: *.pem
# runuser -u container bash
```

4. Log out of the root account.

```
# exit
```

NOTE

We use **/host/var/tmp/repo** so the data is outside of the container. This could be a remote mount point to Ceph/etc.

5. Put the entitlement certificates inside the repo directory.

```
$ cd /host/var/tmp
$ mkdir repo && ostree --repo=repo init --mode=archive-z2
$ mv ~/*.pem repo/
```

6. Copy the remote configuration from the host into the repository:

```
$ cat /host/etc/ostree/remotes.d/redhat.conf >> repo/config
```

7. Change variables
   Edit **repo/config** and change the **tls-client-*** variables to look like the ones below. This tells the command where to find the client certificates that are necessary to access the CDN.

```
tls-client-cert-path = ./repo/123451234512345.pem
tls-client-key-path = ./repo/123451234512345-key.pem
```

8. Final steps

Everything is now set up. The following command will incrementally mirror all of the content. It is possible to run the command from a cron job or systemd timer.

```
$ ostree --repo=repo pull --mirror rhel-atomic-host-ostree
```

For client machines, change **/etc/ostree/remotes.d/redhat.conf** to point to a static web server that is exporting the repo directory.

5.2. UPGRADING TO A NEW VERSION

Unlike Red Hat Enterprise Linux 7 which uses *Yum* and has a traditional package management model, RHEL Atomic Host uses OSTree and is upgraded by preparing a new operating system root, and making it the default for the next boot.

To perform an upgrade, execute the following commands:

```
# atomic host upgrade
# systemctl reboot
```

NOTE

The OSTrees are downloaded securely. However, if you want, you can manually verify the provenance of the OSTree to which you are upgrading. See Manually Verifying OS Trees .

If you are using a system that requires an HTTP proxy, the proxy is configured with an environment variable. To configure the environment variable, use a command similar to the following one:

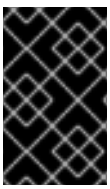
```
# env http_proxy=http://proxy.example.com:port/ atomic host upgrade
```

5.3. ROLLING BACK TO A PREVIOUS VERSION

To revert to a previous installation of Red Hat Enterprise Linux Atomic Host, execute the following commands:

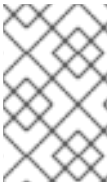
```
# atomic host rollback
# systemctl reboot
```

Two versions of Red Hat Enterprise Linux Atomic Host are available on the system after the initial upgrade. One is the currently running version. The other is either a new version recently installed from an upgrade or the version that was in place prior to the last upgrade.



IMPORTANT

Configuration is preserved across updates, but is only forward-preserved. This means that if you make a configuration change and then later roll back to a previous version, the configuration change you made is reverted.



NOTE

Running the **atomic host upgrade** command will replace the non-running version of Red Hat Enterprise Linux Atomic Host. This version will also be configured to be used during the next boot.

To determine which version of the operating system is running, execute the following command:

```
# atomic host status
```

The output that includes the hash name of the directory in the **/ostree/deploy/rhel-atomic-host/** directory looks like this:

```
# atomic host status
State: idle
Deployments:
* rhel-atomic-host-ostree:rhel-atomic-host/7/x86_64/standard
       Version: 7.3 (2016-09-27 17:53:07)
    BaseCommit: d3fa3283db8c5ee656f78dcfc0fcffe6cd5aa06596dac6ec5e436352208a59cb
        Commit: f5e639ce8186386d74e2558e6a34f55a427d8f59412d47a907793e046875d8dd
        OSName: rhel-atomic-host

  rhel-atomic-host-ostree:rhel-atomic-host/7.2/x86_64/standard
       Version: 7.2.7 (2016-09-15 22:28:54)
    BaseCommit: dbbc8e805f0003d8e55658dc220f1fe1397caf80221cc050eeb1bbf44bef56a1
        Commit: 5cd426fa86bd1652ecd8f7d489f89f13ecb7d36e66003b0d7669721cb79545a8
        OSName: rhel-atomic-host
```

This fictional sample output shows that version 7.3 will be booted into on the next restart. The version to be booted on the next restart is printed first.

This fictional sample also shows that version 7.2.7 is the currently running version. The currently running version is marked with an asterisk (*).

This output was created just after the atomic host upgrade command was executed, and that means that a new version has been staged to be applied at the next restart.

5.4. GENERATING THE INITRAMFS IMAGE ON THE CLIENT

By default, Atomic Host uses a generic **initramfs** image built on the server side. This is distinct from the **yum**-based Red Hat Enterprise Linux, where **initramfs** is generated per installation. However, in some situations, additional configuration or content may need to be added, which requires generating **initramfs** on the client side.

To make an Atomic Host client machine generate **initramfs** on every upgrade, run:

```
$ rpm-ostree initramfs --enable
```

After this, on every upgrade, the client runs the **dracut** program, which builds the new **initramfs**.

To disable generating **initramfs** on the client, run:

```
$ rpm-ostree initramfs --disable
```

CHAPTER 6. MANAGING ATOMIC HOSTS

6.1. ATOMIC HOST

The **atomic** command-line tool can be used to check the status of your Atomic Host system, perform upgrades and rollbacks or deploy a specific operating system tree.

Use **atomic host status** to list the operating system trees downloaded on your system and check which version you are currently running. The asterisk (*) marks the currently running tree.

```
# atomic host status
State: idle
Deployments:
* rhel-atomic-host-ostree:rhel-atomic-host/7/x86_64/standard
       Version: 7.3 (2016-09-27 17:53:07)
    BaseCommit: d3fa3283db8c5ee656f78dcfc0fcffe6cd5aa06596dac6ec5e436352208a59cb
        Commit: f5e639ce8186386d74e2558e6a34f55a427d8f59412d47a907793e046875d8dd
        OSName: rhel-atomic-host

  rhel-atomic-host-ostree:rhel-atomic-host/7.2/x86_64/standard
       Version: 7.2.7 (2016-09-15 22:28:54)
    BaseCommit: dbbc8e805f0003d8e55658dc220f1fe1397caf80221cc050eeb1bbf44bef56a1
        Commit: 5cd426fa86bd1652ecd8f7d489f89f13ecb7d36e66003b0d7669721cb79545a8
        OSName: rhel-atomic-host
```

To upgrade your system, use **atomic host upgrade**. This command will download the latest ostree available and will deploy it after you reboot the system. When you upgrade again, the newly downloaded ostree will replace the oldest one. Upgrading can take a few minutes.

```
# atomic host upgrade
# systemctl reboot
```

To switch back to the other downloaded tree on your system, use **atomic host rollback**. This command is particularly useful when there is a problem after upgrade (for example the new packages break a service that you've configured) because it lets you quickly switch back to the previous state. You can use the **-r** option to initiate a reboot immediately:

```
# atomic host rollback -r
```

To deploy a specific version of an ostree, use **atomic host deploy**. You can specify a version or a commit ID if you know it.

```
# atomic host deploy <version/commit ID>
```

Use the **--preview** option to check the package difference between the specified version and your currently running tree.

```
# atomic host deploy 7.3 --preview
```

For finer tasks you can use the **ostree** tool to manage you Atomic Host. For example, if you are unsure about the version numbering of the trees, you can use the following commands to fetch the ostree logs and list the versions available:

```
# ostree pull --commit-metadata-only --depth -1 rhel-atomic-host-ostree:rhel-atomic-
host/7/x86_64/standard
# ostree log rhel-atomic-host/7/x86_64/standard
```

You can delete an ostree deployment using one of the following commands:

```
# rpm-ostree cleanup -r
# rpm-ostree cleanup -p
```

The **-p** option causes the pending deployment to be removed, while **-r** removes the rollback deployment.

To clear temporary files and cached data, use one of the following commands:

```
# rpm-ostree -m
# rpm-ostree -b
```

The **-m** option deletes cached RPM repository metadata, while **-b** clears temporary files, but leaves deployments intact.

Both the **atomic** and **ostree** tools have built-in detailed **--help** options, to check all commands available on the system, use **atomic host --help** or **ostree --help**.

6.2. PACKAGE LAYERING

Using **rpm-ostree install**, you can add an RPM software packages that is not part of the original OSTree permanently to the system. With **rpm-ostree override**, you can override an existing RPM package in the base system layer with a different version of that package. These features are done using the package layering feature.

Package layering is useful when you need to install a certain package on a single machine, without affecting other machines that share the same OSTree. An example use case of package layering is installing diagnostics tools, such as **strace**. An example of overriding a package in the base system is if you wanted to use a different version of the docker package.

6.2.1. Installing a new RPM package on a RHEL Atomic Host

To install a layered package and its dependencies on RHEL Atomic Host, run the following command:

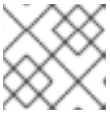
```
# rpm-ostree install <package>
```

To remove a layered package, use the following command:

```
# rpm-ostree uninstall <package>
```

Running the **rpm-ostree install** or **rpm-ostree uninstall** does not immediately install or uninstall the packages. To actually install or uninstall the packages, you have two options:

- Reboot the system.
- Use LiveFS to apply the changes immediately.



NOTE

LiveFS is still a technology preview feature, so do not rely on it in production.

If you are only installing packages, use the **rpm-ostree ex livefs** command.

If you are deleting or upgrading the packages, use the **rpm-ostree ex livefs --replace** command.

You can find out which packages have been manually installed on the system by running **atomic host status**.

The following is an example of installing **strace** on RHEL Atomic Host and how to verify it is part of the OSTree. Just as with installing a package with **yum**, you must register and subscribe your Atomic Host system before installing packages:

1. Check the current status of your Atomic Host's deployments:

```
-bash-4.2# rpm-ostree status
State: idle
Deployments:
● rhelah-7.4:rhel-atomic-host/7/x86_64/standard
      Version: 7.4.0 (2017-07-28 00:26:01)
       Commit: 846fb0e18e65bd9a62fc9d952627413c6467c33c2d726449a1d7ad7690bbb93a

  rhel-atomic-host-ostree:rhel-atomic-host/7/x86_64/standard
      Version: 7.4.0 (2017-07-13 17:46:26)
       Commit: c28cad0d4144d91a3c206574e9341cd5bdf7d34cfaa2acb74dd84c0bf022593a
    GPGSignature: 1 signature
        Signature made Thu 13 Jul 2017 01:54:13 PM EDT using RSA key ID
199E2F91FD431D51
        Good signature from "Red Hat, Inc. <security@redhat.com>"
```

2. Install the strace package as follows:

```
-bash-4.2# rpm-ostree install strace
Checking out tree 846fb0e... done
...
Importing metadata [==========================================] 100%
Resolving dependencies... done
Will download: 1 package (470.0 kB)
  Downloading from rhel-7-server-rpms: [======================] 100%
Importing: [================================================] 100%
Overlaying... done
Writing rpmdb... done
Writing OSTree commit... done
Copying /etc changes: 20 modified, 5 removed, 43 added
Transaction complete; bootconfig swap: yes deployment count change: 0
Freed objects: 388.5 MB
Added:
  strace-4.12-4.el7.x86_64
Run "systemctl reboot" to start a reboot
```

3. Check the status again to see the layered package created by installing strace.

```
-bash-4.2# rpm-ostree  status
State: idle
Deployments:
  rhelah-7.4:rhel-atomic-host/7/x86_64/standard
      Version: 7.4.0 (2017-07-28 00:26:01)
    BaseCommit:
846fb0e18e65bd9a62fc9d952627413c6467c33c2d726449a1d7ad7690bbb93a
    LayeredPackages: strace

● rhelah-7.4:rhel-atomic-host/7/x86_64/standard
      Version: 7.4.0 (2017-07-28 00:26:01)
       Commit: 846fb0e18e65bd9a62fc9d952627413c6467c33c2d726449a1d7ad7690bbb93a
```

4. Note that the strace package does not appear to be installed yet:

```
# rpm -q strace
package strace is not installed
```

5. Consider several issues: Although the package was installed on its own layer, it does not yet appear as being installed on the system. At this point you need to apply the pending deployment by either rebooting or applying them immediately using **rpm-ostree ex livefs**. Before making that decision, however, take into account these notes and limitations:

   - If you run **rpm-ostree install** several times consecutively without rebooting or applying changes live, only the most recent command will take effect. If you install **wget** and **strace** consecutively and reboot afterwards, only **strace** will be present after reboot. To add multiple packages onto a new deployment, specify them all on the same line with the command. For example:

     ```
     # rpm-ostree install wget strace
     ```

   - Installing packages which contain files owned by users other than root is currently not supported. For example, the **httpd** package contains files with a group ownership of **apache**, installing it will fail:

     ```
     # rpm-ostree install httpd
     notice: pkg-add is a preview command and subject to change.

     Downloading metadata: [==========] 100%
     Resolving dependencies... done
     error: Unpacking httpd-2.4.6-40.el7_2.4.x86_64: Non-root ownership currently
     unsupported: path "/run/httpd" marked as root:apache)
     ```

   - After **rpm-ostree install**, do not use **atomic host deploy** or **rpm-ostree deploy** to deploy a specific version OSTree version older than 7.2.6. If you attempt to deploy to such a version after **rpm-ostree install**, the system will be left in a state where you will be unable to use **atomic host upgrade** or **rpm-ostree upgrade** to upgrade to the next release. However, **atomic host rollback** or **rpm-ostree rollback** will still be successful and bring the system back to the previous deployment.

6. Reboot or LiveFS: Either reboot for the deployments to take effect or use the livefs feature, to have them immediately take effect, as follows:

```
# rpm-ostree ex livefs
```

```
notice: "livefs" is an experimental command and subject to change.
Diff Analysis: 846fb0e18e65bd9a62fc9d952627413c6467c33c2d726449a1d7ad7690bbb93a
=> 97f937f3789d0f25b887bcd4fcc03d33b76ee4c87095af48c602b5826519ce1b
Files:
 modified: 0
 removed: 0
 added: 11
Packages:
 modified: 0
 removed: 0
 added: 1
Preparing new rollback matching currently booted deployment
Copying /etc changes: 20 modified, 5 removed, 43 added
Transaction complete; bootconfig swap: yes deployment count change: 1
Overlaying /usr... done
```

7. Check again to see that the strace package is installed and note the status of deployments, including the new LiveCommit:

```
# rpm -q strace
strace-4.12-4.el7.x86_64

# rpm-ostree status
State: idle
Deployments:
  rhelah-7.4:rhel-atomic-host/7/x86_64/standard
                Version: 7.4.0 (2017-07-28 00:26:01)
             BaseCommit:
846fb0e18e65bd9a62fc9d952627413c6467c33c2d726449a1d7ad7690bbb93a
                Commit:
97f937f3789d0f25b887bcd4fcc03d33b76ee4c87095af48c602b5826519ce1b
         LayeredPackages: strace

● rhelah-7.4:rhel-atomic-host/7/x86_64/standard
                Version: 7.4.0 (2017-07-28 00:26:01)
            BootedCommit:
846fb0e18e65bd9a62fc9d952627413c6467c33c2d726449a1d7ad7690bbb93a
              LiveCommit:
97f937f3789d0f25b887bcd4fcc03d33b76ee4c87095af48c602b5826519ce1b

  rhelah-7.4:rhel-atomic-host/7/x86_64/standard
                Version: 7.4.0 (2017-07-28 00:26:01)
                Commit:
846fb0e18e65bd9a62fc9d952627413c6467c33c2d726449a1d7ad7690bbb93a
```

At this point, you can go ahead and start using the installed software. For more information on rpm-ostree and Live updates, see the Project Atomic rpm-ostree

6.2.2. Downloading and caching RPMs for later installation

The **--download-only** and **--cache-only** options allow to separate the **rpm-ostree install** process into two stages:

1. Downloading and caching the layered RPMs.

2. Installing from the cached data.

These options enable you to download the RPMs at one time, and then install them later whenever you choose, even offline.

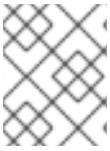
6.2.3. Updating the repository metadata

The **rpm-ostree refresh-md** subcommand downloads and caches the latest repository metadata. It is similar to the **yum makecache** command for the Yum package manager.

6.2.4. Overriding an existing RPM package

To override an RPM package that is in the Atomic base and install a different version, you use the **rpm-ostree override** command. Here's how it works:

- Copy the RPM package you want to use to the Atomic host. Include any dependent packages needed by the RPM as well. The packages can be upgrades or downgrades from the current packages.
- Run the **rpm-ostree override** command.
- Reboot the Atomic host for the change to take effect.



NOTE

See Locking the version of the docker package on RHEL Atomic Host for an example of how to use **rpm-ostree override** to replace the docker runtime in Atomic.

Here's an example of replacing the openssh-server package (and dependent packages) on a RHEL Atomic Host.

1. Get the RPM package (and dependent packages) you want to replace and put them in a directory on the Atomic Host.
2. With the packages in the current directory (in this case, downgrades of openssh-server, openssh-clients, and openssh), type the following to replace those packages:

```
# rpm-ostree override replace \
      openssh-server-6.6.1p1-35.el7_3.x86_64.rpm   \
      openssh-clients-6.6.1p1-35.el7_3.x86_64.rpm  \
      openssh-6.6.1p1-35.el7_3.x86_64.rpm

Checking out tree 5df677d... done
...
Transaction complete; bootconfig swap: yes deployment count change: 1
Downgraded:
  openssh 7.4p1-16.el7 -> 6.6.1p1-35.el7_3
  openssh-clients 7.4p1-16.el7 -> 6.6.1p1-35.el7_3
  openssh-server 7.4p1-16.el7 -> 6.6.1p1-35.el7_3
Run "systemctl reboot" to start a reboot
```

3. Reboot the Atomic Host system:

```
# systemctl reboot
```

4. Check that the packages have been installed and are available:

```
# atomic host status
State: idle
Deployments:
● ostree://rhel-atomic-host-ostree:rhel-atomic-host/7/x86_64/standard
    Version: 7.5.0 (2018-04-05 10:29:00)
  BaseCommit: 5df677dcfef08a87dd0ace55790e184a35716cf11260239216bfeba2eb7c60b0
  ReplacedBasePackages: openssh openssh-server openssh-clients
    7.4p1-16.el7 -> 6.6.1p1-35.el7_3

# rpm -q openssh openssh-clients openssh-server
openssh-6.6.1p1-35.el7_3.x86_64
openssh-clients-6.6.1p1-35.el7_3.x86_64
openssh-server-6.6.1p1-35.el7_3.x86_64
```

If you just want to go back to the previous package versions, you can use **rpm-ostree override reset** to do that. Use **rpm-ostree override reset <packagename>** to remove individual packages or **rpm-ostree override reset --all** to remove all overridden packages.

6.3. "OSTREE ADMIN UNLOCK"

ostree admin unlock unlocks the current ostree deployment and allows packages to be installed temporarily by mounting a writable overlayfs on */usr*. However, the packages installed afterwards will not persist after reboot. **ostree admin unlock** also has certain limitations and known issues with overlayfs and SELinux, so it should be used only for testing. For adding software, **rpm-ostree install** is recommended for long-term use.

6.4. SYSTEM CONTAINERS AND SUPER-PRIVILEGED CONTAINERS (SPCS)

In some cases, containerized services or applications require that they are run from a container image that is built in a different than the default way for Docker-formatted containers. Such special case containers are the Super-Privileged Containers (SPCs), and the system containers. They are necessary in two situations:

- **SPCs**: When a certain container needs more privileges and access to the host.

Super-Privileged Containers are run with special privileges to the host computer, and unlike the default Docker-formatted containers, are able to modify the host. Tools for monitoring and logging are containzerized in SPCs so they can have the access to the host they requires. Such SPCs are **rsyslog**, **sadc**, and the **atomic-tools** container. For detailed information about SPCs, see Running Super-Privileged Containers chapter from the Red Hat Enterprise Linux Atomic Host Managing Containers Guide.

- **System Containers**: A certain container needs to run independently of the docker daemon.

System containers are a way to containerize services which are needed before the docker daemon is running. Such services configure the environment for docker, (for example setting up networking), so they can't be run *by* the docker daemon and because of that, they are not Docker-formatted images. They use **runc** for runtime, **ostree** for storage, **skopeo** for searching and pulling from a registry and systemd for management. A system container is executed from a systemd UNIT file using the **runc**

utility. Additionally, containerizing such services is a way to make the ostree system image smaller. Such System Containers are **etcd** and **flannel**. For detailed information, see Running System Containers chapter from the Red Hat Enterprise Linux Atomic Host Managing Containers Guide.