Composing, Installing, and Managing RHEL for Edge Images on Red Hat Enterprise Linux 8
Legal Notice

Copyright © 2021 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at http://creativecommons.org/licenses/by-sa/3.0/.

In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation’s permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document is for users who want to compose customized RHEL (rpm-ostree) images using Image Builder, and then remotely install and manage the images on Edge servers.
Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE .......................................................... 4

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION .................................... 5

CHAPTER 1. INTRODUCING RHEL FOR EDGE IMAGES ....................................... 6
  1.1. RHEL FOR EDGE—SUPPORTED ARCHITECTURE ......................................... 6
  1.2. HOW TO COMPOSE AND DEPLOY A RHEL FOR EDGE IMAGE ..................... 6
  1.3. DIFFERENCE BETWEEN RHEL RPM IMAGES AND RHEL FOR EDGE IMAGES .... 7

CHAPTER 2. SETTING UP IMAGE BUILDER ....................................................... 9
  2.1. IMAGE BUILDER SYSTEM REQUIREMENTS ........................................... 9
  2.2. INSTALLING IMAGE BUILDER ................................................................ 9

CHAPTER 3. COMPOSING A RHEL FOR EDGE IMAGE USING IMAGE BUILDER IN RHEL WEB CONSOLE .............................................................. 11
  3.1. ACCESSING IMAGE BUILDER IN THE RHEL WEB CONSOLE ...................... 11
  3.2. CREATING A RHEL FOR EDGE IMAGE BLUEPRINT USING IMAGE BUILDER IN RHEL WEB CONSOLE .............................................................. 12
  3.3. CREATING A RHEL FOR EDGE IMAGE USING IMAGE BUILDER IN RHEL WEB CONSOLE .............................................................. 12
  3.4. DOWNLOADING A RHEL FOR EDGE IMAGE ........................................... 13
  3.5. ADDING A SOURCE TO RHEL FOR EDGE IMAGE BLUEPRINT ................ 13
  3.6. CREATING AN ADMINISTRATIVE USER ACCOUNT FOR A RHEL FOR EDGE IMAGE BLUEPRINT .............................................................. 14

CHAPTER 4. COMPOSING A RHEL FOR EDGE IMAGE USING IMAGE BUILDER COMMAND-LINE ............... 16
  4.1. CREATING A RHEL FOR EDGE IMAGE BLUEPRINT USING IMAGE BUILDER COMMAND-LINE INTERFACE .............................................................. 16
  4.2. CREATING A RHEL FOR EDGE IMAGE USING IMAGE BUILDER COMMAND-LINE INTERFACE .............................................................. 17
  4.3. DOWNLOADING A RHEL FOR EDGE IMAGE USING THE IMAGE BUILDER COMMAND-LINE INTERFACE .............................................................. 18

CHAPTER 5. DEPLOYING A RHEL FOR EDGE IMAGE ........................................ 20
  5.1. EXTRACTING RHEL FOR EDGE IMAGE COMMIT ...................................... 20
  5.2. SETTING UP A WEB SERVER TO INSTALL RHEL FOR EDGE IMAGE .......... 21
  5.3. DOWNLOADING RHEL BOOT.IMG IMAGE ............................................... 22
  5.4. INSTALLING THE RHEL FOR EDGE IMAGE USING A KICKSTART FILE .... 23

CHAPTER 6. MANAGING RHEL FOR EDGE IMAGES ........................................ 25
  6.1. EDITING A RHEL FOR EDGE IMAGE BLUEPRINT USING IMAGE BUILDER IN RHEL WEB CONSOLE .............................................................. 25
    6.1.1. Adding a component to RHEL for Edge image blueprint using Image Builder in RHEL web console .............................................................. 25
    6.1.2. Changing the version of an existing component in a RHEL for Edge image blueprint using the RHEL web console .............................................................. 26
    6.1.3. Removing a component from RHEL for Edge image blueprint using Image Builder in RHEL web console .............................................................. 26
  6.2. EDITING A RHEL FOR EDGE IMAGE BLUEPRINT USING COMMAND-LINE INTERFACE .............................................................. 27
  6.3. UPDATING RHEL FOR EDGE IMAGES ................................................ 28
    6.3.1. How are RHEL for Edge image updates deployed ............................... 28
    6.3.2. Deploying RHEL for Edge image updates manually ........................... 29
    6.3.3. Deploying RHEL for Edge automatic image updates ......................... 31
    6.3.4. Updating the RHEL for Edge image update policy ........................... 31
    6.3.5. Enabling RHEL for Edge automatic download and staging of updates ..... 32
  6.4. ROLLING BACK RHEL FOR EDGE IMAGES ........................................ 33
    6.4.1. How are RHEL for Edge images rolled back ..................................... 33
    6.4.2. Rolling back RHEL for Edge images manually ................................... 34
    6.4.3. Rolling back RHEL for Edge images using an automated process ....... 34

APPENDIX A. TERMINOLOGY AND COMMANDS ............................................. 36
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1. OSTREE AND RPM-OSTREE TERMINOLOGY</td>
<td>36</td>
</tr>
<tr>
<td>A.2. OSTREE COMMANDS</td>
<td>36</td>
</tr>
<tr>
<td>A.3. RPM-OSTREE COMMANDS</td>
<td>37</td>
</tr>
</tbody>
</table>
MAKING OPEN SOURCE MORE INCLUSIVE

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

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

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

- For submitting more complex feedback, create a Bugzilla ticket:
  1. Go to the **Bugzilla** website.
  2. As the Component, use **Documentation**.
  3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
  4. Click **Submit Bug**.
CHAPTER 1. INTRODUCING RHEL FOR EDGE IMAGES

A RHEL for Edge image is an rpm-ostree image that includes system packages to remotely install RHEL on Edge servers.

The system packages include:

- Base OS package
- Podman as the container engine
- Additional RPM content

You can customize the image to configure the OS content as per your requirements, and can deploy them on physical and virtual machines, in an on-premise, cloud, or container environment.

You can deploy a RHEL for Edge image on Bare Metal, Appliance, and Edge servers.

With a RHEL for Edge image, you can achieve the following:

1.1. RHEL FOR EDGE—SUPPORTED ARCHITECTURE

Currently, you can deploy RHEL for Edge images on AMD and Intel 64-bit systems.

1.2. HOW TO COMPOSE AND DEPLOY A RHEL FOR EDGE IMAGE

Use Image Builder to create flexible RHEL rpm-ostree images to suit your requirements, and then use Anaconda to deploy them in your environment. Image Builder automatically identifies the details of your deployment setup and generates the image output as a `rhel-edge-commit` as a `.tar` file.

You can access Image Builder through a command-line interface in the `composer-cli` tool, or use a graphical user interface in the RHEL 8 Web Console.

You can compose and deploy the RHEL for Edge image by performing the following high-level steps:

1. Install and register a RHEL system
2. Install Image Builder
3. Using Image Builder, create a blueprint for RHEL for Edge image
4. Import the RHEL for Edge blueprint in Image Builder
5. Create a RHEL for Edge image
6. Download the RHEL for Edge image file
7. Set up a web server
8. Download a RHEL boot.iso image
9. Using the web server, install the RHEL for Edge image on Edge devices

The following diagram represents the RHEL for Edge image deployment workflow:

Additional resources
For more information about installing and registering a RHEL system, see Performing a Standard RHEL Installation.

1.3. DIFFERENCE BETWEEN RHEL RPM IMAGES AND RHEL FOR EDGE IMAGES

You can create RHEL system images in traditional package-based RPM format and also as RHEL for Edge (rpm-ostree) images.

You can use the traditional package-based RPMs to deploy RHEL on traditional data centers. However, with RHEL for Edge images you can deploy RHEL on servers other than traditional datacenter. These servers include systems where processing of large amounts of data is done closest to the source where data is generated—Edge servers.

Refer to the following table to know how RHEL for Edge images differ from the package-based RHEL RPM images.
<table>
<thead>
<tr>
<th>Key attributes</th>
<th>RHEL RPM image</th>
<th>RHEL for Edge image</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS assembly</td>
<td>You can assemble the packages locally to form an image.</td>
<td>The packages are assembled in an ostree which you can install on a system.</td>
</tr>
<tr>
<td>OS updates</td>
<td>You can use <code>yum update</code> to apply the available updates from the enabled repositories.</td>
<td>You can use <code>rpm-ostree upgrade</code> to stage an update if any new commit is available in the ostree remote at <code>/etc/ostree/remotes.d/</code>. The update takes effect on system reboot.</td>
</tr>
<tr>
<td>Repository</td>
<td>The package contains Yum repositories</td>
<td>The package contains Ostree remote repository</td>
</tr>
<tr>
<td>User access permissions</td>
<td>Read write</td>
<td>Read-only (<code>/usr</code>)</td>
</tr>
<tr>
<td>Data persistence</td>
<td>You can mount the image to any non tmpfs mount point</td>
<td><code>/etc</code> &amp; <code>/var</code> are read-write enabled and include persisting data.</td>
</tr>
</tbody>
</table>
CHAPTER 2. SETTING UP IMAGE BUILDER

Use Image Builder to create your customized RHEL for Edge images. After you install Image Builder on a RHEL system, Image Builder is available as an application in RHEL web console. You can also access Image Builder through a command line interface in the composer-cli tool.

NOTE

It is recommended to install Image Builder on a virtual machine.

In the environment where you want to install Image Builder, ensure that you first meet the system requirements and then install it.

2.1. IMAGE BUILDER SYSTEM REQUIREMENTS

The environment where Image Builder runs, for example a virtual machine, must meet the requirements that are listed in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Minimal Required Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>System type</td>
<td>A dedicated virtual machine</td>
</tr>
<tr>
<td>Processor</td>
<td>2 cores</td>
</tr>
<tr>
<td>Memory</td>
<td>4 GiB</td>
</tr>
<tr>
<td>Disk space</td>
<td>20 GiB</td>
</tr>
<tr>
<td>Access privileges</td>
<td>Administrator level (root)</td>
</tr>
<tr>
<td>Network</td>
<td>Connectivity to Internet</td>
</tr>
</tbody>
</table>

2.2. INSTALLING IMAGE BUILDER

To install Image Builder on a dedicated virtual machine, follow these steps:

Prerequisites

- The virtual machine is created and is powered On.
- You have installed RHEL and you have subscribed to RHSM or Redhat Satellite.

Procedure

1. Install the following packages on the virtual machine.
   - osbuild-composer
   - composer-cli
• cockpit-composer
• Bash-completion

# yum install osbuild-composer composer-cli cockpit-composer bash-completion

Image Builder is installed as an application in RHEL web console.

2. Reboot the virtual machine

3. Enable Image Builder.

   # systemctl enable osbuild-composer.socket --now
   # systemctl enable cockpit.socket --now

   The osbuild-composer and cockpit services start automatically on first access.

4. Configure the system firewall to allow access to the web console:

   # firewall-cmd --add-service=cockpit && firewall-cmd --add-service=cockpit --permanent

5. Load the shell configuration script so that the autocomplete feature for the composer-cli command starts working immediately without reboot:

   $ source /etc/bash_completion.d/composer-cli
CHAPTER 3. COMPOSING A RHEL FOR EDGE IMAGE USING IMAGE BUILDER IN RHEL WEB CONSOLE

Use Image Builder to create a custom RHEL for Edge image (OSTree commit).

To access Image Builder and to create your custom RHEL for Edge image, you can either use the RHEL web console interface or the command-line interface.

This section provides information about accessing Image Builder and creating RHEL for Edge images using the RHEL web console. To access Image Builder and to create RHEL for Edge images using CLI, see, Chapter 4, Composing a RHEL for Edge image using Image Builder command-line.

You can compose RHEL for Edge images using Image Builder in RHEL web console by performing the following high-level steps:

1. Access Image Builder in RHEL web console
2. Create a blueprint for RHEL for Edge image
3. Create a RHEL for Edge image
4. Download the RHEL for Edge image
5. Create a user account for the RHEL for Edge image blueprint

3.1. ACCESSING IMAGE BUILDER IN THE RHEL WEB CONSOLE

To access Image Builder in RHEL web console, ensure that you have met the following prerequisites and then follow the procedure.

Prerequisites

- You have installed a RHEL system.
- You have administrative rights on the system.
- You have subscribed the RHEL system to Redhat Subscription Manager (RHSM) or to Redhat Satellite server.
- The system is powered on and accessible over network.
- You have installed Image Builder on the system.

Procedure

2. For more information about how to remotely access Image Builder, see Managing systems using the RHEL 8 web console document.
3. Log in to the web console using an administrative user account.
4. On the web console, in the left hand menu, click Apps.
5. Click Image Builder.
6. The Image Builder dashboard appears in the right pane.

7. You can now proceed to create a blueprint for the RHEL for Edge images.

3.2. CREATING A RHEL FOR EDGE IMAGE BLUEPRINT USING IMAGE BUILDER IN RHEL WEB CONSOLE

To create a RHEL for Edge image blueprint using Image Builder in RHEL web console, ensure that you have met the following prerequisites and then follow the procedure.

Prerequisites

- On a RHEL system, you have opened the Image Builder dashboard.

Procedure

2. Specify a name and description for the blueprint that you want to create.
3. Click Create. The dashboard displays a list of available components.
4. From the Available Components, click + for the component that you want to include in the blueprint. To search for a specific component, enter the component name in the Filter By Name text box, and then press Enter. The Component Details pane displays the component details and its dependent components. You can select the required version. The web console selects the latest version by default.
   
   NOTE: To remove a component from the blueprint, in the Available Components pane, click - against the component name.
5. In the Components Details pane, click Add.
6. Click Commit to save the blueprint. A dialog box with the blueprint summary appears.
7. Click Commit.
8. From the breadcrumbs, click Back to Blueprints. The Image Builder dashboard lists the blueprints that you created.

3.3. CREATING A RHEL FOR EDGE IMAGE USING IMAGE BUILDER IN RHEL WEB CONSOLE

To create a RHEL for Edge image using Image Builder in RHEL web console, ensure that you have met the following prerequisites and then follow the procedure.

Prerequisites

- On a RHEL system, you have accessed the Image Builder dashboard.
Procedure

1. On the Image Builder dashboard, for the blueprint that you have created for RHEL for Edge image, click **Create Image**. To search for a specific blueprint, enter the blueprint name in the **Filter By Name** text box, and then press **Enter**.

2. On the **Create Image** window, perform the following steps:
   a. From the **Type** dropdown list, select “RHEL for Edge Commit (.tar)”.
   b. In the **Parent commit** textbox, specify a previous commit, or leave it empty; if you do not have a commit at this time.
   c. In the **Ref textbox**, specify a name for your commit. By default, the web console specifies `rhel/8/arch_name/edge`.
   d. Click **Create**.

   **NOTE**

   If you have made any earlier edits to the blueprint, but have not committed them, then click **Commit** and create.

   Image Builder creates a RHEL for Edge image for the blueprint that you created.

   To view the RHEL for Edge image creation progress, click the blueprint name from the breadcrumbs, and then click the **Images** tab.

   **NOTE**

   The image creation process takes a few minutes to complete. To abort the image creation process, click **Stop** from the More Options menu.

3.4. DOWNLOADING A RHEL FOR EDGE IMAGE

After the Image Builder successfully creates the RHEL for Edge image, download the image on the local host.

**Procedure**

To download an image:

1. Click **Download** from the **More Options** menu.

   The Image Builder downloads the file at your default download location.

The downloaded file consists of a `.tar` file with an OSTree repository. This repository contains the commit and a `json` file which contains information metadata about the repository content.

3.5. ADDING A SOURCE TO RHEL FOR EDGE IMAGE BLUEPRINT

The sources defined in Image Builder provide the contents that you can add to blueprints. These sources are global and therefore available to all blueprints. The **System sources** are repositories that are set up locally on your computer and cannot be removed from Image Builder. You can add additional
custom sources and thus be able to access other contents than the System sources available on your system. The following steps describe how to add a Source to your local system.

**Prerequisites**

- You have opened the Image Builder interface of the RHEL 8 web console in a browser.

**Procedure**

1. Click the **Manage Sources** button in the upper right corner. A pop-up window appears with the available sources, their names and descriptions.
2. On the right side of the pop-up window, click the **Add Source** button.
3. Add the desired **Source name**, the **Source path**, and the **Source Type**. The Security field is optional.
4. Click **Add Source** button. The screen shows the available sources window and lists the source you have added.

As a result, the new System source is available and ready to be used or edited.

### 3.6. CREATING AN ADMINISTRATIVE USER ACCOUNT FOR A RHEL FOR EDGE IMAGE BLUEPRINT

To build and deploy the RHEL for Edge image that you have created, you must have administrative user privileges on the image blueprint. Use Image Builder to create an administrative user account for the image blueprint. You can then use this user account to deploy the image that you have created. For the administrative user account you can either have a password-based access or an SSH-key based access.

**Prerequisites**

- You have created an SSH key that you can use for the user account to be created.
- You have accessed the Image Builder dashboard in the RHEL web console.
- You have created a blueprint for RHEL for Edge image.

**Procedure**

1. On the Image Builder dashboard, find the blueprint for RHEL for Edge image. To search a required blueprint, specify the blueprint name in the Filter by Name text box and then press **Enter**.
2. Click the blueprint name. Image Builder displays the blueprint details.
3. On the Customizations tab, click **Create User Account**.
4. On the Create User Account dialogue box, specify the required details and a password for the user account. For password-based access, specify a password for the user account. For SSH-based access, specify an SSH key for the user account.
NOTE

Ensure that you select the **Server administrator** check box, if you want to provide administrators rights to the user account you are creating.

Image Builder creates a specified user account and displays the details.

You can create additional user accounts, if required.
CHAPTER 4. COMPOSING A RHEL FOR EDGE IMAGE USING IMAGE BUILDER COMMAND-LINE

Use Image Builder to create a custom RHEL for Edge image (OSTree commit).

To access Image Builder and to create your custom RHEL for Edge image, you can either use the RHEL web console interface or the command-line interface.

This chapter provides information about creating RHEL for Edge images using the CLI. To create RHEL for Edge images using RHEL web console, see, Chapter 3, Composing a RHEL for Edge image using Image Builder in RHEL web console.

The workflow to compose RHEL for Edge images using the CLI, involves the following high-level steps:

1. Create a blueprint for RHEL for Edge image
2. Create a RHEL for Edge image
3. Download the RHEL for Edge image

To perform the steps, use the composer-cli package.

NOTE
To run the composer-cli commands as non-root, you must be part of the weldr group or you must have administrator access to the system.

4.1. CREATING A RHEL FOR EDGE IMAGE BLUEPRINT USING IMAGE BUILDER COMMAND-LINE INTERFACE

To create a blueprint for RHEL for Edge image, perform the following steps:

Procedure

1. Create a plain text file in the Tom’s Obvious, Minimal Language (TOML) format, with the following content:

   ```toml
   [blueprint]
   name = "blueprint-name"
   description = "blueprint-text-description"
   version = "0.0.1"
   modules = []
   groups = []
   ```

   Where,
   - `blueprint-name` is the name and blueprint-text-description is the description for your blueprint.
   - `0.0.1` is the version number according to the Semantic Versioning scheme.
   - Modules describe the package name and matching version glob to be installed into the image, for example, the package name = "tmux" and the matching version glob is version = "2.9a".
     + Notice that currently there are no differences between packages and modules.
Groups are packages grouped to be installed into the image, for example the group package anaconda-tools. At this time, if you do not know the modules and groups, leave them empty.

2. Include the required packages and customize the other details in the blueprint to suit your requirements.

For more information about the packages that you can include and customize in a blueprint, see Supported Image Customizations.

For every package that you want to include in the blueprint, add the following lines to the file:

```
[packages]
name = "package-name"
version = "package-version"
```

Where,

- package-name is the name of the package, such as httpd, gdb-doc, or coreutils.
- package-version is the version number of the package that you want to use. The package-version supports the following dnf version specifications:
  - For a specific version, use the exact version number such as 8.3.0.
  - For the latest available version, use the asterisk *.
  - For the latest minor version, use formats such as 8.*.

3. Push (import) the blueprint to the Image Builder server:

```
# composer-cli blueprints push blueprint-name.toml
```

4. List the existing blueprints to check whether the created blueprint is successfully pushed and exists.

```
# composer-cli blueprints show BLUEPRINT-NAME
```

5. Check whether the components and versions listed in the blueprint and their dependencies are valid:

```
# composer-cli blueprints depsolve blueprint-name
```

### 4.2. CREATING A RHEL FOR EDGE IMAGE USING IMAGE BUILDER COMMAND-LINE INTERFACE

To create a RHEL for Edge image using Image Builder command line interface, ensure that you have met the following prerequisites and then follow the procedure.

**Prerequisites**

- You have created a blueprint for RHEL for Edge image.

**Procedure**
1. Begin to create the RHEL for Edge image.

```
# composer-cli compose start blueprint-name image-type
```

Where, `blueprint-name` is the RHEL for Edge blueprint name. `image-type` is `rhel-edge-commit`.

A confirmation that the composer process has been added to the queue appears. It also shows a Universally Unique Identifier (UUID) number for the image created. Use the UUID number to track your build. Also keep the UUID number handy for further tasks.

For more information about supported image types, refer to the Creating a system image with Image Builder in the command-line interface.

2. Check the image compose status.

```
# composer-cli compose status
```

The output displays the status in the following format:

```
<UUID> RUNNING date blueprint-name _image-type
```

**NOTE**

The image creation process takes a few minutes to complete.

To interrupt the image creation process, run:

```
# composer-cli compose cancel <UUID>
```

4.3. DOWNLOADING A RHEL FOR EDGE IMAGE USING THE IMAGE BUILDER COMMAND-LINE INTERFACE

To download a RHEL for Edge image using Image Builder command line interface, ensure that you have met the following prerequisites and then follow the procedure.

**Prerequisites**

- You have created a RHEL for Edge image.

**Procedure**

1. Review the RHEL for Edge image status.

```
# composer-cli compose status
```

The output must display the following:

```
$ <UUID> FINISHED date blueprint-name image-type
```

2. Download the image.
# composer-cli compose image <UUID>

Image Builder downloads the image as a tar file at `<var/lib/osbuild-composer/composer/results/<UUID>/compose/>`.

The UUID number and the image size is displayed alongside.

$ <UUID>-commit.tar: size MB

The image contains a commit and a `json` file that contains information metadata about the repository content.
CHAPTER 5. DEPLOYING A RHEL FOR EDGE IMAGE

You can deploy a RHEL for Edge image using the RHEL installer graphical user interface or a Kickstart file.

To deploy the images on bare metal, use a Kickstart file.

Deploying a RHEL for Edge image involves the following high-level steps:

1. Extract the image file contents. See Section 3.4, “Downloading a RHEL for Edge image” for information on how to download the RHEL for Edge image
2. Set up a web server
3. Install the image

5.1. EXTRACTING RHEL FOR EDGE IMAGE COMMIT

After you download the commit, extract the .tar file and note the ref name and the commit ID.

The downloaded commit file consists of a .tar file with an OSTree repository. The OSTree repository contains a commit and a compose.json file.

The compose.json file contains information metadata about the commit with information such as the “Ref” (reference ID) and the commit ID. The commit contains the rpm packages.

To extract the package contents, perform the following the steps:

Prerequisites

1. Create a kickstart file or use an existing one.

Procedure

1. Extract the downloaded image tar file:
   
   ```
   # tar xvf <UUID>-commit.tar
   ```

2. Navigate to the directory where you have extracted the .tar file.
   It contains a compose.json file and a OSTree directory. The compose.json file contains the commit number and the OSTree directory contains the rpm packages.

3. Open the compose.json file and note the commit ID number. You need this number handy when you proceed to set up a web server.

4. List the rpm packages in the commit.
   
   ```
   # rpm-ostree db list rhel/8/x86_64/edge --repo=repo
   ```

5. Use a Kickstart file to run the RHEL installer. You can either use any existing file or can create one using the Kickstart Generator tool.
   In the Kickstart file, ensure that you include the details about how to provision the file system, create a user, and how to fetch and deploy the RHEL for Edge image. The RHEL installer uses this information during the installation process.
The following is a Kickstart file example:

```plaintext
lang en_US.UTF-8
keyboard us
timezone Etc/UTC --isUtc
text
zerombr
clearpart --all --initlabel
autopart
reboot
user --name=core --group=wheel
sshkey --username=core "ssh-rsa AAAA3Nza...."

ostreesetup --nogpg --osname=rhel --remote=edge
--url=https://mirror.example.com/repo/
--ref=rhel/8/x86_64/edge
```

Additional resources

For more details on how to create a Kickstart file, see:

- Performing an Advanced RHEL Installation

## 5.2. SETTING UP A WEB SERVER TO INSTALL RHEL FOR EDGE IMAGE

After you have extracted the RHEL for Edge image contents, set up a web server to provide the image commit details to the RHEL installer via HTTP.

The following example provides the steps to set up a web server using a container.

### Prerequisites

- You have installed Podman on your system. See How do I install podman in RHEL 8

### Procedure

1. Create a dockerfile.
   
   On the command prompt, run the following commands:

   ```
   FROM registry.access.redhat.com/ubi8/ubi
   RUN yum -y install httpd && yum clean all
   ADD kickstart.ks /var/www/html/
   ARG commit=commit.tar
   ADD $commit /var/www/html/
   EXPOSE 80
   CMD ["/usr/sbin/httpd", 
   "-D", 
   "FOREGROUND"]
   ```

   Where,

   - `kickstart.ks` is the name of the Kickstart file from the RHEL for Edge image. The kickstart file includes directive information. To help you manage the images later, it is advisable to include the checks and settings for Greenboot checks. To do so, you can update the Kickstart file to include the following settings:
#!/bin/bash

DNS_SERVER=$(grep nameserver /etc/resolv.conf | cut -f2 -d"")
COUNT=0

# check DNS server is available
ping -c1 $DNS_SERVER
while [ $? != '0' ] && [ $COUNT -lt 10 ]; do
  ((COUNT++))
  echo "Checking for DNS: Attempt $COUNT ."
  sleep 10
  ping -c 1 $DNS_SERVER
done

- *Commit.tar* is the image commit number

- `~/var/www/html` is the location where the commit is saved. You can choose a different location to save the commit.
  The ostree repo can be hosted from any HTTP server/service, and that the example, using a container, is just an option for how to do this. The docker file performs the following tasks:

  a. Uses the latest Universal Base Image (ubi)
  b. Installs the web server (httpd)
  c. Adds the kickstart file to the server
  d. Adds the RHEL for Edge image commit to the server

  The line *ARG commit=commit.tar* allows you to specify future commits using the command-line.

2. Build a docker container

```bash
# podman build -t name-of-container-image --build-arg commit=uuid-commit.tar
```

3. Run the container

```bash
# podman run --rm -d -p port:80 localhost/name-of-container-image
```

As a result, the server is set up and ready to launch the RHEL installer using the *commit.tar* repository and the kickstart file.

### 5.3. DOWNLOADING RHEL BOOT.ISO IMAGE

You can download a Red Hat Boot ISO image from the Red Hat Customer Portal. The Red Hat Boot ISO image is used to launch the RHEL installer. The installer fetches the Kickstart file that you provide for installing RHEL for Edge images.

**Prerequisites**

- You have an active Red Hat subscription.
You are logged in to the Product Downloads section of the Red Hat Customer Portal at https://access.redhat.com/downloads.

Procedure

1. Open a browser and access https://access.redhat.com/downloads.

2. Click the Red Hat Enterprise Linux 8.

3. Click the button Download Now for the option “Red Hat Enterprise Linux 8.2 Boot ISO”

For more information about downloading Boot.iso image, see Downloading the installation ISO image.

5.4. INSTALLING THE RHEL FOR EDGE IMAGE USING A KICKSTART FILE

To install the RHEL for Edge image using a Kickstart file, use the web server. The web server uses the RHEL for Edge image commit.tar repository and the kickstart file to launch the RHEL installer.

Prerequisites

- The server to fetch the commit in the Installer is available and running.
- A disk image to install the commit you created.

Procedure

1. Run the Anaconda Installer using the libvirt virt-install:

   ```
   virt-install --name rhel-edge-test-1 --memory 2048 --vcpus 2 --disk
   path=prepared_disk_image.qcow2,format=qcow2 --os-variant rhel8.0 --cdrom
   /home/username/Downloads/rhel8-edge.iso
   ```

2. On the installation screen, press TAB to specify an additional kernel parameter:

   ```
   inst.ks=http://edge_device_ip:port/kickstart.ks
   ```

   The kernel parameter specifies that you want to install RHEL using the Kickstart file and not the RHEL image contained in the RHEL Installer.

   The RHEL Installer starts, fetches the Kickstart file from the server (HTTP) endpoint and executes the commands, including the command to install the RHEL for Edge image commit from the HTTP endpoint. Once the installation is complete, the installer prompts for login details.

3. On the Login screen, specify your user account credentials and click Enter.

4. Verify whether the RHEL for Edge image is successfully installed.

   ```
   $ rpm-ostree status
   ```

   The command output provides the image commit ID and shows that the installation is successful.
Following is a sample output:

State: idle
Deployments:
  * ostree://edge:rhel/8/x86_64/edge
    Timestamp: 2020-09-18T20:06:54Z
    Commit: 836e637095554e0b634a0a48ea05c75280519dd6576a392635e6fa7d4d5e96
CHAPTER 6. MANAGING RHEL FOR EDGE IMAGES

To manage the RHEL for Edge images, you can perform any of the following administrative tasks:

- Edit the RHEL for Edge image blueprint using Image Builder in RHEL web console
- Edit the RHEL for Edge image blueprint using Image Builder command-line
- Update the RHEL for Edge images
- Configure rpm-ostree remotes on nodes/Update node policy
- Restore RHEL for Edge images manually or automatically using a Greenboot

6.1. EDITING A RHEL FOR EDGE IMAGE BLUEPRINT USING IMAGE BUILDER IN RHEL WEB CONSOLE

You can edit the RHEL for Edge image blueprint to:

- add additional components that you may require
- modify the version of any existing component
- remove any existing component

6.1.1. Adding a component to RHEL for Edge image blueprint using Image Builder in RHEL web console

To add a component to a RHEL for Edge image blueprint, ensure that you have met the following prerequisites and then follow the procedure to edit the corresponding blueprint.

Prerequisites

- On a RHEL system, you have accessed the Image Builder dashboard.
- You have created a blueprint for RHEL for Edge image.

Procedure

1. On the Image Builder dashboard, click the RHEL for Edge image blueprint that you want to edit. To search for a specific blueprint, enter the blueprint name in the Filter By Name text box, and then press Enter.

2. On the upper right side of the blueprint, click Edit Packages. The view changes to the Edit Packages mode.

3. Enter the component name that you want to add in the Filter By Name text box, and then press Enter. A list with the component name appears.

4. Click the + sign adjacent to the component. The component is added to the Blueprint.

5. Click Commit.
The blueprint updates are saved, and a message with the pending commit appears.

6. On the summary dialogue box, review the changes and then click **Commit**.
   A message confirming the successful commit appears.

   As a result, a new version of the blueprint is created and the right pane lists the latest components.

### 6.1.2. Changing the version of an existing component in a RHEL for Edge image blueprint using the RHEL web console

You had selected a default (latest) version or had chosen a version for the components that you included in the blueprint. If required, you can now change the version for any component that you might want to.

To do so, ensure that you have met the following prerequisites and then follow the procedure to change the version of the component in the corresponding blueprint.

**Prerequisites**

- On a RHEL system, you have accessed the Image Builder dashboard.
- You have created a blueprint for RHEL for Edge image.
- You have added at least one component to the RHEL for Edge blueprint.

**Procedure**

1. On the Image Builder dashboard, click the blueprint that you want to edit. To search for a specific blueprint, enter the blueprint name in the Filter By Name text box, and then press **Enter**.

2. On the upper right side of the blueprint, click **Edit Packages**. The view changes to the Edit Packages mode, and the right panel lists the component names that are currently committed to the blueprint.

3. Click the component name.

4. Select the desired version from the Component Options Version dropdown list.

5. Click **Apply Changes**. The change is saved and the right pane lists the latest changes.

6. Click **Commit**. The new version is saved in the blueprint. A message with pending commits appears.

7. On the summary dialogue box, review the changes and then click **Commit**. A message confirming the successful commit appears.

   As a result a new version of the blueprint is created and the right pane lists the latest components.

### 6.1.3. Removing a component from RHEL for Edge image blueprint using Image Builder in RHEL web console
To remove one or more unwanted components from a RHEL for Edge image blueprint that you created, ensure that you have met the following prerequisites and then follow the procedure.

**Prerequisites**

- On a RHEL system, you have accessed the Image Builder dashboard.
- You have created a blueprint for RHEL for Edge image.
- You have added at least one component to the RHEL for Edge blueprint.

**Procedure**

1. On the Image Builder dashboard, click the blueprint that you want to edit. To search for a specific blueprint, enter the blueprint name in the Filter By Name text box, and then press Enter.
2. On the upper right side of the blueprint, click **Edit Packages**. The view changes to the Edit Packages mode. The right panel lists the component names that are currently committed to the blueprint.
3. From the More Options menu, click **Remove**. Optionally, click the component name and then click **Remove**.
4. Click **Commit**. A message with pending commits appears.
5. Review your changes and then click **Commit**. A message confirming the successful commit appears.

As a result, a new version of the blueprint is created and the right pane lists the latest components.

### 6.2. EDITING A RHEL FOR EDGE IMAGE BLUEPRINT USING COMMAND-LINE INTERFACE

You can change the specifications for your RHEL for Edge image blueprint using Image Builder command-line. To do so, ensure that you have met the following prerequisites and then follow the procedure to edit the corresponding blueprint.

**Prerequisites**

- You have access to the Image Builder command-line.
- You have created a RHEL for Edge image blueprint.

**Procedure**

1. Save (export) the blueprint to a local text file:

   ```
   # composer-cli blueprints save BLUEPRINT-NAME
   ```

2. Edit the **BLUEPRINT-NAME.toml** file with a text editor of your choice and make your changes. Before finishing with the edits, make sure the file is a valid blueprint:
3. Increase the version number. Ensure that you use a Semantic Versioning scheme.

**NOTE**

if you do not change the version, the patch component of the version is increased automatically.

4. Check if the contents are valid TOML specifications. See the TOML documentation for more information.

**NOTE**

TOML documentation is a community product and is not supported by Red Hat. You can report any issues with the tool at [https://github.com/toml-lang/toml/issues](https://github.com/toml-lang/toml/issues).

5. Save the file and close the editor.

6. Push (import) the blueprint back into Image Builder command-line:

   ```bash
   # composer-cli blueprints push BLUEPRINT-NAME.toml
   ```

   **NOTE**

   When pushing the blueprint back into the Image Builder command-line, provide the file name including the `.toml` extension.

7. Verify that the contents uploaded to Image Builder match your edits:

   ```bash
   # composer-cli blueprints show BLUEPRINT-NAME
   ```

8. Check whether the components and versions listed in the blueprint and their dependencies are valid:

   ```bash
   # composer-cli blueprints depsolve BLUEPRINT-NAME
   ```

### 6.3. UPDATING RHEL FOR EDGE IMAGES

#### 6.3.1. How are RHEL for Edge image updates deployed

With RHEL for Edge images, you can either deploy the updates manually or can automate the deployment process. The updates are applied in an atomic manner, where the state of each update is known, and the updates are staged and applied only upon reboot. Because no changes are seen until you reboot the device, you can schedule a reboot to ensure the highest possible uptime.

During the image update, because only the updated OS content is transferred over the network, the deployment process is more efficient compared to transferring the entire image. The OS binaries and libraries (/usr) are read-only, and the read-write state is maintained in /var and /etc directories.

With the delta transfer, you can deploy the updates even in case of intermittent and low-bandwidth
connections, or to disconnected devices by using local media. Additionally, you can also create static-deltas to further reduce the network usage. The static-deltas pull all individual updates into a single file archive, and significantly reduces the TCP network overhead of transferring the OS updates over a single TCP connection as opposed to transferring each update individually over multiple connections.

The following diagram illustrates the RHEL for Edge image update deployment process:

![Diagram of RHEL for Edge image update deployment process]

6.3.2. Deploying RHEL for Edge image updates manually

After you have edited a RHEL for Edge blueprint, you can update the image commit. Image Builder generates a new commit for the updated RHEL for Edge image. Use this new commit to deploy the image with latest package versions or with additional packages.

To deploy RHEL for Edge images updates, ensure that you meet the prerequisites and then follow the procedure.

Prerequisites

- On a RHEL system, you have accessed the Image Builder dashboard.
- You have created a RHEL for Edge image blueprint.
- You have edited the RHEL for Edge image blueprint. See Section 6.1, “Editing a RHEL for Edge image blueprint using Image Builder in RHEL web console”.

Procedure

1. On the Image Builder dashboard, for the blueprint that you have edited, click **Create Image**.
2. On the Create Image window, perform the following steps:
   a. From the **Type** dropdown list, select “RHEL for Edge Commit (.tar)”.

CHAPTER 6. MANAGING RHEL FOR EDGE IMAGES

29
b. In the **Parent commit** textbox, specify the parent commit ID that was previously generated. See Section 5.1, “Extracting RHEL for Edge image commit”.

c. In the Ref textbox, you can either specify a name for your commit or leave it empty. By default, the web console specifies the Ref as `rhel/8/arch_name/edge`.

d. Click **Create**. Image Builder creates a RHEL for Edge image for the updated blueprint. To view the RHEL for Edge image creation progress, click the blueprint name from the breadcrumbs, and then click the **Images** tab.

**NOTE**

The image creation process takes a few minutes to complete.

The resulting image includes the latest packages that you have added, if any, and have the original **commit ID** as a parent.

3. Download the resulting RHEL for Edge image. For more information about downloading a RHEL for Edge image, see Section 3.4, “Downloading a RHEL for Edge image”.

4. Extract the OSTree commit. For more information about extracting an OSTree commit, see Section 5.1, “Extracting RHEL for Edge image commit”.

5. Build a docker container, serving the child commit ID this time.

   ```
   # podman build -t <_name-of-server_> --build-arg commit=<uuid>-child_commit.tar
   ```

6. Run the container.

   ```
   # podman run --rm -p 8000:80 <_name-of-server_>
   ```

7. On the RHEL system provisioned, from the original edge image, verify the current status.

   ```
   $ rpm-ostree status
   ```

   If there is no new commit ID, run the following command to verify if there is any upgrade available:

   ```
   $ rpm-ostree upgrade --check
   ```

   The command output provides the current active OSTree commit ID.

8. Update OSTree to make the new OSTree commit ID available.

   ```
   $ rpm-ostree upgrade
   ```

   OSTree verifies if there is an update on the repository. If yes, it fetches the update and requests you to reboot your system so that you can activate the deployment of this new commit update.

9. Check the current status again:

   ```
   $ rpm-ostree status
   ```

   You can now see that there are 2 commits available:
• The active parent commit.
• A new commit that is not active and contains 1 added difference.

10. To activate the new deployment and to make the new commit active, reboot your system.

```
# systemctl reboot
```

The Anaconda Installer reboots into the new deployment. On the login screen, you can see a new deployment available for you to boot.

11. If you want to boot into the newest commit/deployment, the rpm-ostree upgrade command automatically orders the boot entries so that the new deployment is first in the list. Optionally, you can use the arrow key on your keyboard to select the GRUB menu entry and press Enter.

12. Provide your login user account credentials.

13. Verify the OSTree status:

```
$ rpm-ostree status
```

The command output provides the active commit ID.

14. To view the changed packages, if any, run a diff between the parent commit and the new commit:

```
$ rpm-ostree db diff parent_commit new_commit
```

The update shows that the package you have installed is available and ready for use.

### 6.3.3. Deploying RHEL for Edge automatic image updates

After you install a RHEL for Edge image on an Edge device, you can check for image updates available, if any, and can auto-apply them.

The `rpm-ostreed-automatic.service` (systemd service) and `rpm-ostreed-automatic.timer` (systemd timer) control the frequency of checks and upgrades. The available updates, if any, appear as staged deployments.

Deploying automatic image updates involves the following high-level steps:

• Update the image update policy
• Enable automatic download and staging of updates

### 6.3.4. Updating the RHEL for Edge image update policy

To update the image update policy, use the `AutomaticUpdatePolicy` and an `IdleExitTimeout` setting from the `rpm-ostreed.conf` file at `/etc/rpm-ostreed.conf` location on an Edge device.

The `AutomaticUpdatePolicy` setting has the following options:

• none: Disables automatic updates. By default, the `AutomaticUpdatePolicy` setting is set to `none`. 
- check: Downloads enough metadata to display available updates with rpm-ostree status.
- stage: Downloads and unpacks the updates that are applied on a reboot.

To enable automatic updates, perform the following steps:

Procedure

1. In the `/etc/rpm-ostreed.conf` file, update the following:
   - Change the value of `AutomaticUpdatePolicy` to `check`.
   - To run the update checks, specify a value in seconds for `IdleExitTimeout`.

2. Reload the `rpm-ostreed` service and enable the `systemd` timer.

```
# systemctl reload rpm-ostreed
# systemctl enable rpm-ostreed-automatic.timer --now
```

3. Verify the rpm-ostree status to ensure the automatic update policy is configured and time is active.

```
# rpm-ostree status
```

The command output shows the following:

State: idle; auto updates enabled (check; last run <minutes> ago)

Additionally, the output also displays information about the available updates.

6.3.5. Enabling RHEL for Edge automatic download and staging of updates

After you update the image update policy to check for image updates, the updates if any are displayed along with the update details. If you decide to apply the updates, enable the policy to automatically download and stage the updates. The available image updates are then downloaded and staged for deployment. The updates are applied and take effect when you reboot the Edge device.

To enable the policy for automatic download and staging of updates, perform the following updates:

Procedure

1. In the `/etc/rpm-ostreed.conf` file, update `AutomaticUpdatePolicy` to `stage`.

2. Reload the rpm-ostreed service.

```
# systemctl enable rpm-ostreed-automatic.timer --now
```

3. Verify the rpm-ostree status

```
# rpm-ostree status
```

The command output shows the following:
4. To initiate the updates, you can either wait for the timer to initiate the updates, or can manually start the service.

```bash
# systemctl start rpm-ostreed-automatic.service
```

After the updates are initiated, the `rpm-ostree status` shows the following:

```bash
# rpm-ostree status
State: busy
AutomaticUpdates: stage; rpm-ostreed-automatic.service: running
Transaction: automatic (stage)
```

When the update is complete, a new deployment is staged in the list of deployments, and the original booted deployment is left untouched. You can decide if you want to boot the system using the new deployment or can wait for the next update.

To view the list of deployments, run the `rpm-ostree status` command.

Following is a sample output.

```
# rpm-ostree status
State: idle
AutomaticUpdates: stage; rpm-ostreed-automatic.timer: last run <time> ago
```

To view the list of deployments with the updated package details, run the `rpm-ostree status -v` command.

### 6.4. ROLLING BACK RHEL FOR EDGE IMAGES

You can verify if the updated image is successfully deployed or not. If the deployment is unsuccessful, you can roll back to a previous version (commit). To roll back to a previous functional state, you can either perform the steps manually or can use an automated process.

#### 6.4.1. How are RHEL for Edge images rolled back

With RHEL for Edge images, only transactional updates are applied to the operating system. With the transactional updates, you can easily rollback the unsuccessful updates to the last known good state, preventing system failure during updates.

You can use intelligent rollbacks with Greenboot to eliminate the issues of choosing between application stability and application of security updates.

Greenboot leverages `rpm-ostree` and runs custom health checks that run on system startup. In the event of an issue, the system rolls back the changes and preserves the last working state.

The following diagram illustrates the RHEL for Edge image roll back process.
6.4.2. Rolling back RHEL for Edge images manually

If the deployment for RHEL for Edge image update fails or if the update fails to work successfully, then you can manually roll back to a previous deployment version.

To roll back to a previous version, perform the following steps:

Procedure

1. Run the **rollback** command:

   ```bash
   # rpm-ostree rollback
   ```

   The command output provides details about the commit ID that is being moved and indicates a completed transaction with the details of the package being removed.

2. Reboot the system.

   ```bash
   # systemctl reboot
   ```

   The command activates the previous commit with the stable content. The changes are applied and the previous version is restored.

6.4.3. Rolling back RHEL for Edge images using an automated process

Greenboot checks provides a framework that is integrated into the boot process and can trigger **rpm-ostree** rollbacks when a health check fails. For the health checks, you can create a custom script that indicates whether a health check passed or failed. Based on the result, you can decide when a rollback should be triggered.

To create a health check script, perform the following steps:

Procedure

1. Create a script that returns a standard exit code **0**.
   For example, the following script ensures that the configured DNS server is available:

   ```bash
   #!/bin/bash
   DNS_SERVER=$(grep ^nameserver /etc/resolv.conf | head -n 1 | cut -f2 -d" ")
   COUNT=0
   ```
# check DNS server is available
ping -c 1 $DNS_SERVER
while [ $? != '0' ] && [ $COUNT -lt 10 ]; do
  ((COUNT++))
  echo "Checking for DNS: Attempt $COUNT ."
  sleep 10
  ping -c 1 $DNS_SERVER
done

2. Include an executable file for the health checks at `/etc/greenboot/check/required.d/`.

  chmod +x check-dns.sh

During the next reboot, the script is executed as part of the boot process, before the system enters the boot-complete.target. If the health checks are successful, no action is taken. If the health checks fail, the system is rebooted several times, before marking the update as failed and rolling back to the previous update.

Verification steps

To check if the default gateway is reachable, run the following health check script:

1. Create a script that returns a standard exit code 0.

   ```bash
   #!/bin/bash
   DEF_GW=$(ip r | awk '/^default/ {print $3}')
   SCRIPT=$(basename $0)
   count=10
   connected=0
   ping_timeout=5
   interval=5

   while [ $count -gt 0 -a $connected -eq 0 ]; do
     echo "$SCRIPT: Pinging default gateway $DEF_GW"
     ping -c 1 -q -W $ping_timeout $DEF_GW > /dev/null 2>&1 && connected=1 || sleep $interval
     ((--count))
   done

   if [ $connected -eq 1 ]; then
     echo "$SCRIPT: Default gateway $DEF_GW is reachable."
     exit 0
   else
     echo "$SCRIPT: Failed to ping default gateway $DEF_GW!" 1>&2
     exit 1
   fi
   ```

2. Include an executable file for the health checks at `/etc/greenboot/check/required.d/` directory.

  chmod +x check-gw.sh
APPENDIX A. TERMINOLOGY AND COMMANDS

This section provides the rpmostree terminology and commands.

A.1. OSTREE AND RPM-OSTREE TERMINOLOGY

Following are some helpful terms that are used in context to OSTree and rpm-ostree images.

Table A.1. OSTree and rpm-ostree terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSTree</td>
<td>A tool used for managing Linux-based operating system versions. The OSTree tree view is similar to Git and is based on similar concepts.</td>
</tr>
<tr>
<td>rpm-ostree</td>
<td>A hybrid image or system package that hosts operating system updates.</td>
</tr>
<tr>
<td>Commit</td>
<td>A release or image version of the operating system. Image Builder generates an ostree commit for RHEL for Edge images. You can use these images to install or update RHEL on Edge servers.</td>
</tr>
<tr>
<td>Refs</td>
<td>Represents a branch in ostree. Refs always resolve to the latest commit. For example, rhel/8/x86_64/edge.</td>
</tr>
<tr>
<td>Revision (Rev)</td>
<td>SHA-256 for a specific commit.</td>
</tr>
<tr>
<td>Remote</td>
<td>The http or https endpoint that hosts the ostree content. This is analogous to the baseurl for a yum repository.</td>
</tr>
<tr>
<td>static-delta</td>
<td>Updates to ostree images are always delta updates. In case of RHEL for Edge images, the TCP overhead can be higher than expected due to the updates to number of files. To avoid TCP overhead, you can generate static-delta between specific commits, and send the update in a single connection. This optimization helps large deployments with constrained connectivity.</td>
</tr>
</tbody>
</table>

A.2. OSTREE COMMANDS

This section provides a few ostree commands that you can use when installing or managing ostree images.

Table A.2. ostree commands
<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ostree pull</code></td>
<td><code>ostree pull-local --repo [path] src</code></td>
</tr>
<tr>
<td></td>
<td><code>ostree pull-local &lt;path&gt; &lt;rev&gt; --repo=&lt;repo-path&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>ostree pull &lt;URL&gt; &lt;rev&gt; --repo=&lt;repo-path&gt;</code></td>
</tr>
<tr>
<td><code>ostree summary</code></td>
<td><code>ostree summary -u --repo=&lt;repo-path&gt;</code></td>
</tr>
<tr>
<td>View refs</td>
<td><code>ostree refs --repo ~/Code/src/osbuild-iot/build/repo/ --list</code></td>
</tr>
<tr>
<td>View commits in repo</td>
<td><code>ostree log --repo=/home/gicmo/Code/src/osbuild-iot/build/repo/ &lt;REV&gt;</code></td>
</tr>
<tr>
<td>Inspect a commit</td>
<td><code>ostree show --repo build/repo &lt;REV&gt;</code></td>
</tr>
<tr>
<td>List remotes of a repo</td>
<td><code>ostree remote list --repo &lt;repo-path&gt;</code></td>
</tr>
<tr>
<td>Resolve a REV</td>
<td><code>ostree rev-parse --repo ~/Code/src/osbuild-iot/build/repo fedora/x86_64/osbuild-demo</code></td>
</tr>
<tr>
<td></td>
<td><code>ostree rev-parse --repo ~/Code/src/osbuild-iot/build/repo b3a008eceeddd0cfd</code></td>
</tr>
<tr>
<td>Create static-delta</td>
<td><code>ostree static-delta generate --repo=[path] --from=REV --to=REV</code></td>
</tr>
<tr>
<td>Sign an existing ostree commit with a GPG key</td>
<td><code>ostree gpg-sign --repo=&lt;repo-path&gt; --gpg-homedir &lt;gpg_home&gt; COMMIT KEY-ID...</code></td>
</tr>
</tbody>
</table>

A.3. RPM-OSTREE COMMANDS

This section provides a few `rpm-ostree` commands that you can use when installing or managing ostree images.

Table A.3. rpm-ostree commands

<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>rpm-ostree --repo=/home/gicmo/Code/src/osbuild-iot/build/repo/ db list &lt;REV&gt;</code></td>
<td>This command lists the packages existing in the <code>&lt;REV&gt;</code> commit into the repository.</td>
</tr>
</tbody>
</table>
**Commands** | **Description**
--- | ---
**rpm-ostree rollback** | OSTree manages an ordered list of bootloader entries, called **deployments**. The entry at index 0 is the default bootloader entry. Each entry has a separate **/etc** directory, but all the entries share a single **/var** directory. You can use the bootloader to choose between entries by pressing Tab to interrupt startup. This rolls back to the previous state, i.e. the default deployment changes places with the non-default one.

**rpm-ostree status** | This command gives information about the current deployment in use. Lists the names and refspects of all possible deployments in order, such that the first deployment in the list is the default upon boot. The deployment marked with * is the current booted deployment, and marking with ‘r’ indicates the most recent upgrade.

**rpm-ostree db list** | Use this command to see which packages are within the commit or commits. You must specify at least one commit, but more than one or a range of commits also work.

**rpm-ostree db diff** | Use this command to show how the packages are different between the trees in two revs (revisions). If no revs are provided, the booted commit is compared to the pending commit. If only a single rev is provided, the booted commit is compared to that rev.

**rpm-ostree upgrade** | This command downloads the latest version of the current tree, and deploys it, setting up the current tree as the default for the next boot. This has no effect on your running filesystem tree. You must reboot for any changes to take effect.

**Additional resources**
- For more information, see the **rpm-ostree** man page.