



Red Hat Enterprise Linux 8.0 Beta

Installing and deploying RHEL

Installation documentation for Red Hat Enterprise Linux 8.0 Beta

Red Hat Enterprise Linux 8.0 Beta Installing and deploying RHEL

Installation documentation for Red Hat Enterprise Linux 8.0 Beta

Legal Notice

Copyright © 2019 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, 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 Software Collections 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 describes installation of Red Hat Enterprise Linux 8.0 Beta on a variety of hardware systems.

Table of Contents

THIS IS A BETA VERSION!	7
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	8
CHAPTER 1. PREPARING FOR YOUR INSTALLATION	9
1.1. BETA INSTALLATION IMAGE TYPES	9
1.2. DOWNLOADING BETA INSTALLATION IMAGES	9
1.2.1. Processing your Subscription Request	10
Prerequisites	10
Procedure	10
1.2.2. Downloading a Beta ISO Image	10
Prerequisites	10
Procedure	11
1.2.3. Registering your Beta Subscription	11
1.2.4. UEFI Secure Boot	12
1.2.4.1. Adding a Custom Private Key for UEFI Secure Boot	12
Prerequisites	12
Procedure	12
1.2.4.2. Removing the Beta public key	13
CHAPTER 2. PREPARING TO BOOT THE INSTALLER	14
2.1. CREATING MEDIA	14
2.1.1. Creating an Installation CD or DVD	14
2.1.2. Creating a Bootable USB Drive on Linux	14
Prerequisites	14
Procedure	14
CHAPTER 3. BOOTING THE INSTALLER	16
3.1. BOOTING THE INSTALLER FROM LOCAL MEDIA	16
Additional Resources	16
3.2. ANACONDA BOOT OPTIONS REFERENCE	16
Additional resources	16
CHAPTER 4. GRAPHICAL INSTALLATION	17
4.1. INTRODUCTION TO ANACONDA	17
4.2. CONFIGURING LANGUAGE AND LOCATION SETTINGS	17
Additional resources	18
4.3. INSTALLATION SUMMARY	18
Additional resources	20
4.4. LOCALIZATION SETTINGS	20
4.4.1. Configuring Keyboard, Language, and Time and Date Settings	20
4.5. SOFTWARE SETTINGS	22
4.5.1. Configuring Installation Source	22
Prerequisites	22
Procedure	22
4.5.2. Configuring Software Selection	24
Prerequisites	24
4.6. SYSTEM SETTINGS	25
4.6.1. Configuring Installation Destination	25
Prerequisites	26
Procedure	26
4.6.1.1. Configuring Boot Loader	29
4.6.2. Configuring Kdump	30

Procedure	30
4.6.3. Network and Host Name	30
4.6.3.1. Configuring Network and Host Name	31
4.6.3.2. Adding a Virtual Network Interface	31
4.6.3.3. Editing Network Interface Configuration	32
4.6.3.4. Enabling or Disabling the Interface Connection	32
4.6.3.5. Setting up Static IPv4 or IPv6 Settings	33
4.6.3.6. Configuring Routes	34
4.6.4. Security Policy	34
4.6.4.1. About Security Policy	34
4.6.4.2. Configuring a Security Policy	35
Prerequisites	35
Procedure	35
4.6.4.3. Related information	35
4.6.5. System Purpose	35
4.6.5.1. Configuring System Purpose using Anaconda	36
4.6.5.2. Configuring System Purpose using Kickstart	36
4.7. STORAGE DEVICES	38
4.7.1. Storage Device Selection	38
4.7.2. Filtering Storage Devices	39
Procedure	39
4.7.3. Advanced Storage Options	39
4.7.3.1. Discovering and Starting an iSCSI Session	40
4.7.3.2. Configuring FCoE Parameters	41
4.7.3.3. Configuring DASD Storage Devices	42
4.7.3.4. Configuring FCP Devices	42
4.7.4. Installing to an NVDIMM Device	43
4.7.4.1. Criteria for using an NVDIMM Device as an Installation Target	43
4.7.4.2. Configuring an NVDIMM Device using Anaconda	44
4.7.4.3. Configuring an NVDIMM Device using Kickstart	45
4.8. MANUAL PARTITIONING	46
4.8.1. Starting Manual Partitioning	46
4.8.2. Adding a Mount Point File System	48
4.8.3. Configuring a Mount Point File System	48
4.8.4. Customizing a Partition or Volume	49
4.8.4.1. Creating Software RAID	51
Prerequisites	51
Procedure	51
4.8.4.2. Creating an LVM Logical Volume	52
Procedure	52
4.8.4.3. Configuring an LVM Logical Volume	52
Procedure	52
4.8.5. Recommended Partitioning Scheme	53
4.9. STARTING THE INSTALLATION PROGRAM	56
4.9.1. Beginning Installation	56
Prerequisites	56
4.9.2. Configuring a Root Password	56
4.9.3. Creating a User Account	57
4.9.3.1. Configuring Advanced User Settings	58
4.9.4. Installation Complete	59
CHAPTER 5. KICKSTART INSTALLATION	60
5.1. CREATING A KICKSTART FILE	60

5.2. MAKING A KICKSTART FILE AVAILABLE	61
5.3. PERFORMING A KICKSTART INSTALLATION	61
Procedure	61
Additional Resources	61
5.4. KICKSTART SYNTAX REFERENCE	61
5.5. INSTALLING PACKAGE MODULES IN KICKSTART SCRIPTS	61
CHAPTER 6. BUILDING CUSTOM SYSTEM IMAGES WITH COMPOSER	64
6.1. INTRODUCTION TO COMPOSER	64
Composer Output Formats	64
Composer User Interfaces	64
Composer Blueprints	65
6.2. COMPOSER SYSTEM REQUIREMENTS	65
6.3. PREPARING A REPOSITORY MIRROR FOR COMPOSER	65
Prerequisites	65
Procedure	66
6.4. INSTALLING COMPOSER	66
Prerequisites	66
Procedure	67
6.5. ACCESSING COMPOSER GUI IN COCKPIT	68
Prerequisites	68
Procedure	68
6.6. CREATING A COMPOSER BLUEPRINT	68
Prerequisites	69
Procedure	69
6.7. EDITING A COMPOSER BLUEPRINT	69
Prerequisites	70
Procedure	70
6.8. CREATING A SYSTEM IMAGE WITH COMPOSER	71
Prerequisites	71
Procedure	71
6.9. ADDITIONAL RESOURCES	72
CHAPTER 7. CREATING CLOUD IMAGES WITH COMPOSER	73
7.1. PREPARING COMPOSER FOR CREATING AWS AMI IMAGES	73
7.2. PREPARING COMPOSER FOR CREATING AZURE VHD IMAGES	74
7.3. CREATING CLOUD IMAGES USING COMPOSER	75
7.4. USING AN AMI IMAGE ON AWS	75
7.5. USING AN VHD IMAGE ON AZURE	77
7.6. USING AN VMDK IMAGE ON VSPHERE	78
7.7. USING AN QCOW2 IMAGE ON OPENSTACK	80
CHAPTER 8. PARTITIONING REFERENCE	83
8.1. SUPPORTED DEVICE TYPES	83
8.2. SUPPORTED FILE SYSTEMS	83
8.3. SUPPORTED RAID TYPES	84
8.4. RECOMMENDED PARTITIONING SCHEME	84
8.5. ADVICE ON PARTITIONS	87
CHAPTER 9. TROUBLESHOOTING INSTALLATION PROBLEMS	89
9.1. CONSOLES AND LOGGING DURING INSTALLATION	89
9.2. SAVING SCREENSHOTS	90
PART I. INSTALLING RED HAT ENTERPRISE LINUX ON IBM POWER	91

CHAPTER 10. BOOTING THE INSTALLATION ON IBM POWER SYSTEMS	92
10.1. THE BOOT MENU	92
10.2. INSTALLING FROM A DIFFERENT SOURCE	94
10.3. BOOTING FROM THE NETWORK USING AN INSTALLATION SERVER	94
CHAPTER 11. PLANNING FOR INSTALLATION ON IBM POWER SYSTEMS	96
11.1. IS YOUR HARDWARE COMPATIBLE?	96
11.2. IBM INSTALLATION TOOLS	96
11.2.1. Preparation for IBM Power Systems Servers	96
11.2.2. Supported Installation Targets	97
11.2.3. System Specifications List	98
11.2.4. Disk Space and Memory Requirements	99
11.2.5. RAID and Other Disk Devices	99
11.2.5.1. Hardware RAID	99
11.2.5.2. Software RAID	99
11.2.5.3. USB Disks	100
11.2.6. Choose an Installation Boot Method	100
11.2.7. Automating the Installation with Kickstart	100
CHAPTER 12. UPDATING DRIVERS DURING INSTALLATION ON IBM POWER SYSTEMS	101
12.1. PREPARING FOR A DRIVER UPDATE DURING INSTALLATION	101
12.1.1. Preparing to Use a Driver Update Image File on Local Storage	102
12.1.2. Preparing a Driver Disc	102
12.1.3. Performing a Driver Update During Installation	103
12.1.3.1. Automatic Driver Update	103
12.1.3.2. Assisted Driver Update	104
12.1.3.3. Manual Driver Update	105
12.1.3.4. Blacklisting a Driver	106
PART II. INSTALLING RED HAT ENTERPRISE LINUX ON IBM Z	107
CHAPTER 13. PLANNING FOR INSTALLATION ON IBM Z	108
13.1. PRE-INSTALLATION	108
13.2. OVERVIEW OF THE SYSTEM Z INSTALLATION PROCEDURE	108
13.2.1. Booting the Installation	109
13.2.2. Installation using Anaconda	109
CHAPTER 14. BOOTING THE INSTALLATION ON IBM Z	111
14.1. CUSTOMIZING BOOT PARAMETERS	111
14.2. CONSIDERATIONS FOR HARD DRIVE INSTALLATION ON IBM Z	112
14.3. INSTALLING UNDER Z/VM	113
14.3.1. Using the z/VM Reader	114
14.3.2. Using a Prepared DASD	115
14.3.3. Using a Prepared FCP-attached SCSI Disk	115
14.3.4. Using an FCP-attached SCSI DVD Drive	116
14.4. INSTALLING IN AN LPAR	116
14.4.1. Using an FTP Server	117
14.4.2. Using a Prepared DASD	117
14.4.3. Using a Prepared FCP-attached SCSI Disk	118
14.4.4. Using an FCP-attached SCSI DVD Drive	118
CHAPTER 15. CONFIGURING AN INSTALLED LINUX ON IBM Z INSTANCE	119
15.1. ADDING DASDS	119
15.1.1. Dynamically Setting DASDs Online	119
15.1.2. Preparing a New DASD with Low-level Formatting	120

15.1.3. Persistently Setting DASDs Online	121
15.1.3.1. DASDs That Are Part of the Root File System	121
15.1.3.2. DASDs That Are Not Part of the Root File System	123
15.2. ADDING FCP-ATTACHED LOGICAL UNITS (LUNS)	124
15.2.1. Dynamically Activating an FCP LUN	124
15.2.2. Persistently activating FCP LUNs	125
15.2.2.1. FCP LUNs That Are Part of the Root File System	125
15.2.2.2. FCP LUNs That Are Not Part of the Root File System	127
15.3. ADDING A NETWORK DEVICE	128
15.3.1. Adding a qeth Device	128
15.3.1.1. Dynamically Adding a qeth Device	129
15.3.1.2. Dynamically Removing a qeth Device	131
15.3.1.3. Persistently Adding a qeth Device	132
15.3.2. Adding an LCS Device	134
15.3.2.1. Dynamically Adding an LCS Device	135
15.3.2.2. Persistently Adding an LCS Device	135
15.3.3. Configuring an IBM z Network Device for Network Root File System	136
CHAPTER 16. PARAMETER AND CONFIGURATION FILES ON IBM Z	138
16.1. REQUIRED PARAMETERS	138
16.2. THE Z/VM CONFIGURATION FILE	138
16.3. INSTALLATION NETWORK PARAMETERS	139
16.4. PARAMETERS FOR KICKSTART INSTALLATIONS	142
16.5. MISCELLANEOUS PARAMETERS	143
16.6. SAMPLE PARAMETER FILE AND CMS CONFIGURATION FILE	144
CHAPTER 17. IBM Z REFERENCES	145
17.1. IBM Z PUBLICATIONS	145
17.2. IBM REDBOOKS PUBLICATIONS FOR SYSTEM Z	145
17.3. ONLINE RESOURCES	145

THIS IS A BETA VERSION!

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

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

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

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

- For simple comments on specific passages, make sure you are viewing the documentation in the Multi-page HTML format. Highlight the part of text that you want to comment on. Then, click the **Add Feedback** pop-up that appears below the highlighted text, and follow the displayed instructions.
- For submitting more complex feedback, create a Bugzilla ticket:
 1. Go to the [Bugzilla](#) website.
 2. As the Component, use **Documentation**.
 3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
 4. Click **Submit Bug**.

CHAPTER 1. PREPARING FOR YOUR INSTALLATION

You must have at least one active Red Hat subscription to download Red Hat Enterprise Linux 8.0 Beta ISO image files from the Red Hat Customer Portal. You can purchase Red Hat Enterprise Linux subscriptions from the Red Hat Store at <https://www.redhat.com/store>. You can evaluate Red Hat Enterprise Linux in your environment using time-limited evaluation entitlements at <https://www.redhat.com>.

1.1. BETA INSTALLATION IMAGE TYPES

Two types of Red Hat Enterprise Linux 8.0 Beta installation images are available for the AMD64 and Intel 64, ARM, IBM Power, and IBM Z.

Binary DVD ISO image file

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



IMPORTANT

- It is **recommended** that the Binary DVD ISO image file is used to install Red Hat Enterprise Linux 8.0 Beta.
- Binary DVDs for IBM Z can be used to boot the installation program using a SCSI DVD drive or as installation sources.

Boot ISO image file

A minimal boot image to boot the installation program. The Boot ISO image requires additional repositories.

The following table lists the types of boot and installation images available for the supported architectures.

Table 1.1. Boot and Installation Images

Architecture	Installation DVD	Boot DVD
AMD64 and Intel 64	x86_64 Binary DVD ISO image file	x86_64 Boot ISO image file
ARM 64	AArch64 Binary DVD ISO image file	AArch64 Boot ISO image file
IBM POWER	ppc64le Binary DVD ISO image file	ppc64le Boot ISO image file
IBM Z	s390x Binary DVD ISO image file	s390x Boot ISO image file

1.2. DOWNLOADING BETA INSTALLATION IMAGES

This section provides instructions on how to download the installation images to install Red Hat Enterprise Linux 8.0 Beta.

1.2.1. Processing your Subscription Request

Your Red Hat Enterprise Linux subscription status is processed before you can download Red Hat Enterprise Linux 8.0 Beta.

Prerequisites

- You have an active, non-evaluation Red Hat Enterprise Linux subscription.
- You have not previously received a Red Hat Enterprise Linux 8.0 Beta subscription.
- You are logged in to the Red Hat Customer Portal at <https://access.redhat.com/home>.

Procedure

1. Visit <https://access.redhat.com/products/red-hat-enterprise-linux/beta>.
2. Click **Get Started**.
A notification message displays your subscription status and next steps.
3. Depending on your subscription status, complete the required steps in the following table.

Table 1.2. Subscription Status

Subscription Status	Notification Message Contains	Next Steps
You do not have a Red Hat Enterprise Linux subscription	Instructions on how to purchase a Red Hat Enterprise Linux subscription.	Complete the subscription purchase instructions and then complete the steps in Section 1.2.1, “Processing your Subscription Request” .
You have previously purchased a Red Hat Enterprise Linux subscription but it is no longer active	Information that your request is being processed. When complete, a message with further details is sent to your email address.	Complete the subscription purchase or renew instructions and then complete the steps in Section 1.2.1, “Processing your Subscription Request” .
You have an active Red Hat Enterprise Linux subscription	Information that your request is being processed. When complete, a message with further details is sent to your email address.	Your subscription is verified. Proceed to Section 1.2.2, “Downloading a Beta ISO Image” to download a Red Hat Enterprise Linux 8.0 Beta ISO image.

1.2.2. Downloading a Beta ISO Image

This section provides instructions on how to download a Red Hat Enterprise Linux 8.0 Beta ISO image.

Prerequisites

- Your Red Hat subscription status is verified in [Section 1.2.1, “Processing your Subscription Request”](#).
- You are logged in to the Red Hat Customer Portal at <https://access.redhat.com/home>.

Procedure

1. Click **DOWNLOADS**.
2. Select the **By Category** tab.
3. Click the **Red Hat Enterprise Linux** link.
The **Download Red Hat Enterprise Linux** web page opens.
4. From the **Product Variant** drop-down menu, select the required variant.



NOTE

If you are unsure of the variant for your requirements, see <http://www.redhat.com/en/technologies/linux-platforms/enterprise-linux>.

5. From the **Version** drop-down menu, select **8.0 Beta**.
6. From the **Architecture** drop-down menu, select the required architecture.
A **Beta Binary DVD** image and a **Beta Boot ISO** image are displayed under the **Product Software** tab. Additional images may be available, for example, preconfigured virtual machine images, but they are beyond the scope of this document.



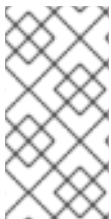
NOTE

The **Beta Binary DVD** is the **recommended** ISO image for installing Red Hat Enterprise Linux 8.0 Beta as it contains all the required repositories and does not require additional configuration.

7. Click **Download Now** beside the required image.

1.2.3. Registering your Beta Subscription

This section provides instructions on how to register your Red Hat Enterprise Linux 8.0 Beta subscription.



NOTE

When auto-attaching a system, the subscription service checks if the system is physical or virtual, as well as how many sockets are on the system. A physical system usually consumes two entitlements, a virtual system usually consumes one. One entitlement is consumed per two sockets on a system.

Prerequisites

- You have an active, non-evaluation Red Hat Enterprise Linux subscription.
- Your Red Hat subscription status is verified in [Section 1.2.1, “Processing your Subscription Request”](#).
- You have not previously received a Red Hat Enterprise Linux 8.0 Beta subscription.

- You have activated your subscription before attempting to download entitlements from the Customer Portal. You need an entitlement for each instance you plan to use. Red Hat Customer Service is available if you need help activating your subscription.
- You have successfully installed a Red Hat Enterprise Linux 8.0 Beta system and logged into it.

Procedure

1. Open a terminal window and type the following command to check for available subscriptions:

```
# subscription-manager list --available
```

2. Use the Pool ID printed by this command to attach the Red Hat Enterprise Linux entitlement:

```
# subscription-manager attach --pool=Pool-ID  
Successfully attached a subscription for: <Subscription-Name>
```

3. List the enabled repositories to verify that your system is correctly subscribed:

```
$ yum repolist enabled
```



NOTE

You can also register Red Hat Enterprise Linux 8.0 Beta in a graphical interface by logging into the system with the **root** user and using the Subscription Manager GUI.

1.2.4. UEFI Secure Boot

UEFI Secure Boot requires that the operating system kernel is signed with a recognized private key. For Red Hat Enterprise Linux 8.0 Beta, the kernel is signed with a Red Hat Beta-specific private key, which is different from the standard Red Hat key used in a General Availability release.

Red Hat Enterprise Linux 8.0 Beta cannot boot if your hardware does not recognize the Beta private key. To use UEFI Secure Boot with the Beta release, add the Red Hat Beta public key to your system using the Machine Owner Key (MOK) facility.

1.2.4.1. Adding a Custom Private Key for UEFI Secure Boot

This section provides instructions on how to add a private key for UEFI Secure Boot on Red Hat Enterprise Linux 8.0 Beta.

Prerequisites

Disable UEFI Secure Boot on the system and install Red Hat Enterprise Linux 8.0 Beta.

Procedure

1. Install the **kernel-doc** package:

```
# yum install kernel-doc
```

The package contains the Red Hat CA public Beta key, located in **/usr/share/doc/kernel-*keys/kernel-version/kernel-signing-ca.cer***, where *kernel-version* is the string without the platform architecture suffix - for example, **3.10.0-686.el7**.

2. Execute the following commands to enroll the public key in the system's Machine Owner Key (MOK) list:

```
# kr=$(uname -r)
# mokutil --import /usr/share/doc/kernel-keys/${kr%.$(uname -
p)}/kernel-signing-ca.cer
```

3. Enter a password.
4. Reboot the system.
5. During system startup, you are prompted to confirm the key request. Select **Yes** and enter the password.
The system reboots and the key is imported into the system's firmware.
6. Enable Secure Boot on the system.



WARNING

Remove the imported Beta public key if you install a General Availability (GA) release of Red Hat Enterprise Linux 8.0 Beta or if you install a different operating system.

1.2.4.2. Removing the Beta public key

1. Execute following command to reset the MOK and remove the Beta public key:

```
# mokutil --reset
```

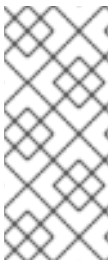
2. Reboot the system and enter the password used when importing the key.
The key is removed from the MOK and the system reverts to its original state.

CHAPTER 2. PREPARING TO BOOT THE INSTALLER

The following section explains how to prepare to boot the installation from local media such as a CD, DVD, or USB drive.

2.1. CREATING MEDIA

This section describes how to use ISO image files downloaded in [Section 1.2, “Downloading Beta Installation Images”](#) to create bootable physical media, such as a DVD or a USB flash drive. You can then use the media to boot the installation program and start the installation. These steps only apply if you plan to install Red Hat Enterprise Linux 8.0 Beta on a 64-bit AMD, Intel, or ARM system, or an IBM Power Systems server using physical boot media. For information on installing on an IBM Z server, see [Planning for Installation on IBM Z](#).



NOTE

By default, the `inst.stage2=` boot option is used on the installation media and set to a specific label (for example, `inst.stage2=hd:LABEL=RHEL7\x20Server.x86_64`). If you modify the default label of the file system containing the runtime image, or if using a customized procedure to boot the installation system, you must ensure this option is set to the correct value.

2.1.1. Creating an Installation CD or DVD

You can create an installation CD or DVD using burning software on your computer and a CD/DVD burner. The exact series of steps to create CD or DVD from an ISO image file varies, depending on the operating system and disc burning software. Consult your burning software’s documentation for the exact steps needed to burn a CD or DVD from an ISO image file.



WARNING

The full installation ISO image may be larger than 4.7 GB and as a result, it may not fit on a single-layer DVD. A dual-layer DVD or USB key is recommended.

2.1.2. Creating a Bootable USB Drive on Linux

Prerequisites

- An installation ISO image downloaded after following the instructions in [Section 1.2, “Downloading Beta Installation Images”](#).
- A USB flash drive that is large enough to hold the ISO image.

Procedure

1. Open a terminal window and insert the USB drive.
2. Find the *device node* assigned to the drive. In the example below, the drive is given `sdd`.

```
$ dmesg|tail
```

```
[288954.686557] usb 2-1.8: New USB device strings: Mfr=0, Product=1,
SerialNumber=2
[288954.686559] usb 2-1.8: Product: USB Storage
[288954.686562] usb 2-1.8: SerialNumber: 000000009225
[288954.712590] usb-storage 2-1.8:1.0: USB Mass Storage device
detected
[288954.712687] scsi host6: usb-storage 2-1.8:1.0
[288954.712809] usbcore: registered new interface driver usb-storage
[288954.716682] usbcore: registered new interface driver uas
[288955.717140] scsi 6:0:0:0: Direct-Access      Generic  STORAGE
DEVICE    9228 PQ: 0 ANSI: 0
[288955.717745] sd 6:0:0:0: Attached scsi generic sg4 type 0
[288961.876382] sd 6:0:0:0: sdd Attached SCSI removable disk
```

3. Use the **dd** utility to write the image. Verify you have the correct drive.

```
# dd if=/path/to/RHEL-8-Alpha.iso of=/dev/sdd
```

CHAPTER 3. BOOTING THE INSTALLER

The following section explains how to boot the installer for AMD64, Intel 64, and ARM 64 architectures.

3.1. BOOTING THE INSTALLER FROM LOCAL MEDIA

After you have made a bootable USB flash drive or a CD or DVD, you are ready to boot the installation. Note that the steps described below are generic. The exact steps depend on your system. Consult your hardware manufacturer's documentation for specific instructions for your system.

When you successfully boot the installer from local media, the first screen is the boot menu. You have several options to choose from. These options are defined by sets of **boot options** - commands that tell the system how to proceed. You can use one of the pre-set menu entries, or you can highlight an entry using the arrow keys, and press **E** to enter edit mode to change the predefined command line. This is useful if you want to load a Kickstart file during the installation. In that case, add the **inst.ks=** option and enter the location of the Kickstart file.

Additional Resources

- [Section 3.2, “Anaconda Boot Options Reference”](#) provides a list of available boot options you can use on the boot command line.
- [Chapter 5, *Kickstart Installation*](#) describes what a Kickstart file is and how to create one.

3.2. ANACONDA BOOT OPTIONS REFERENCE

The installer boot option reference content is not available in the Beta release of this document. Use the [upstream version](#) instead.

Additional resources

[Chapter 2, *Preparing to Boot the Installer*](#) explains how to create a bootable DVD or a USB drive.

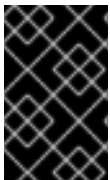
CHAPTER 4. GRAPHICAL INSTALLATION

This section provides instructions on how to install Red Hat Enterprise Linux 8.0 Beta using the Graphical User Interface (GUI) of the Anaconda installer.



NOTE

The instructions provided in this section are for AMD64 and Intel 64 (x86_64), ARM (AArch64), IBM Power Systems, and IBM Z architectures. The instructions do not cover all elements of the GUI, only those that are required to complete a task.



IMPORTANT

This content is not final. There may be some discrepancy between the online help content and the content that is published on the Customer Portal. For the latest Beta content, see the installation content on the Customer Portal.

4.1. INTRODUCTION TO ANACONDA

The Graphical User Interface (GUI) of the Anaconda installer is the preferred method of installing Red Hat Enterprise Linux 8.0 Beta when you boot the system from local media (CD, DVD, or a USB flash drive). Once Anaconda starts you must select the language and location. You can configure the remaining aspects of the installation in the **Installation Summary** window.

Anaconda allows full control over all available settings, including custom partitioning and advanced storage configuration.



NOTE

This does not apply to all parts of the installation program. For example, when you install the program from a network location, you must configure the network before you can select the packages you want to install.

Some installation windows are automatically configured depending on the hardware and type of media used to start the installation; you can change these settings during the installation.

Yelp is the default help viewer for GNOME desktop. The **Help** button opens the **Yelp** help viewer and displays the Red Hat Enterprise Linux 8.0 Beta graphical installation help content.



IMPORTANT

Due to an open issue with **Yelp**, some of the hyperlinks in the built-in help table of contents do not render correctly.

4.2. CONFIGURING LANGUAGE AND LOCATION SETTINGS

From the **Welcome to Red Hat Enterprise Linux** window, configure the language and location parameters. The selected language is used during the installation program and on the installed system. During the installation program you can change the language and location settings for the installed system using the **Installation Summary** window.

Procedure

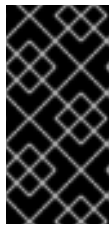
1. From the left-hand pane, select a language of your choice. Alternatively, type your preferred language into the **Search** field.



NOTE

When the **Welcome to Red Hat Enterprise Linux** window opens, a language is pre-selected by default. If network access is configured, that is, if you booted from a network server instead of local media, the pre-selected language is determined by the automatic location detection feature of the **GeoIP** module. If you used the **inst.lang=** option on the boot command line or in your PXE server configuration, then this language is selected.

2. From the right-hand pane, select a location specific to your region.
3. Click **Continue** to proceed to the [Section 4.3, “Installation Summary”](#) window.



IMPORTANT

- If you are installing a pre-release version of Red Hat Enterprise Linux, a warning message is displayed about the pre-release status of the installation media. Click **I want to proceed** to continue with the installation, or **I want to exit** to quit the installation and reboot the system.

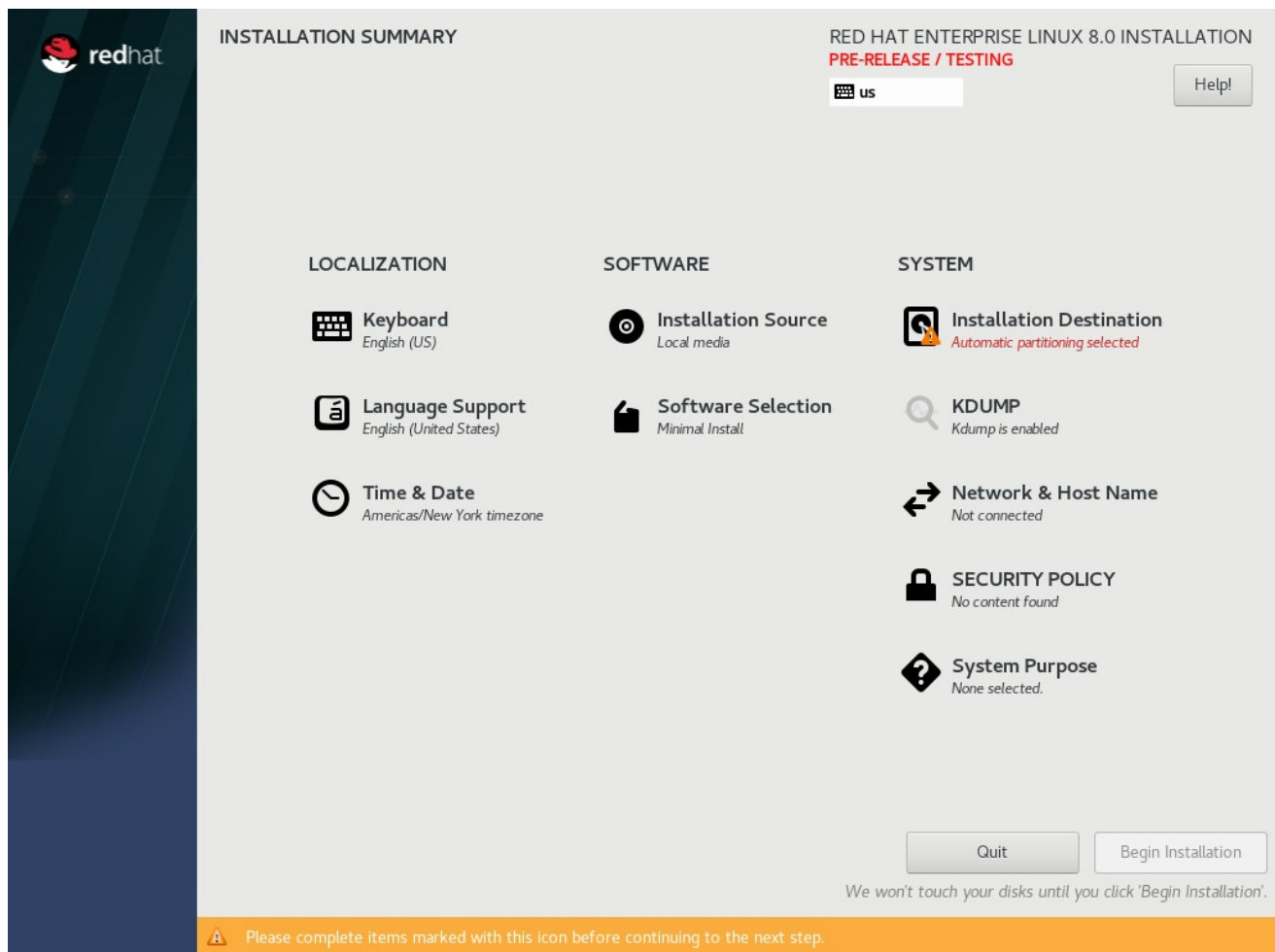
Additional resources

For information on how to change language and location settings during the installation program, see: [Section 4.4, “Localization Settings”](#)

4.3. INSTALLATION SUMMARY

The **Installation Summary** window is the central location for the Red Hat Enterprise Linux 8.0 Beta installation program.

Figure 4.1. Installation Summary

**NOTE**

If you used a Kickstart option or a boot option to specify an installation repository on a network, but no network is available at the start of the installation, the installation program displays the **Network Configuration** window to set up a network connection prior to displaying the **Installation Summary** window.

The **Installation Summary** window contains three categories:

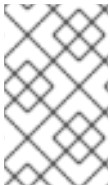
- **LOCALIZATION** enables you to configure Keyboard, Language Support, and Time and Date.
- **SOFTWARE** enables you to configure Installation Source and Software Selection.
- **SYSTEM** enables you to configure Installation Destination, KDUMP, Network and Host Name, Security Policy, and System Purpose.

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

Table 4.1. Installation Summary category status

Category status	Status	Description
-----------------	--------	-------------

Category status	Status	Description
Warning symbol type 1	Yellow triangle with an exclamation mark and red text	Requires attention before installation. For example, Installation Destination requires attention as you have to confirm the default automatic partitioning variant.
Warning symbol type 2	Greyed out and with a warning symbol (yellow triangle with an exclamation mark)	The installation program is configuring a category and you must wait for it to finish before accessing the window.

**NOTE**

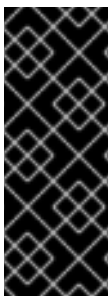
A warning message is displayed at the bottom of the **Installation Summary** window and the **Begin Installation** button is disabled until you configure all of the required categories.

Additional resources

- For information on how to configure Localization settings, see: [Section 4.4, “Localization Settings”](#)
- For information on how to configure Software settings, see: [Section 4.5, “Software Settings”](#)
- For information on how to configure System settings, see: [Section 4.6, “System Settings”](#)

4.4. LOCALIZATION SETTINGS

This section describes how to configure your keyboard, language support, and time and date settings. It does not detail all aspects of the GUI, only those that are required to complete the task.

**IMPORTANT**

If you use a layout that cannot accept Latin characters, such as **Russian**, you are advised to also add the **English (United States)** layout and configure a keyboard combination to switch between the two layouts. If you only select a layout that does not have Latin characters, you may be unable to enter a valid **root** password and user credentials later in the installation process. This can prevent you from completing the installation.

4.4.1. Configuring Keyboard, Language, and Time and Date Settings

**NOTE**

Keyboard, Language, and Time and Date Settings were configured as part of [Section 4.2, “Configuring Language and Location Settings”](#). If you want to change any of the settings complete the following steps, otherwise proceed to [Section 4.5, “Software Settings”](#).

Procedure

1. From the **Installation Summary** window, click **Keyboard**. The default layout depends on the option selected in [Section 4.2, “Configuring Language and Location Settings”](#).
 - a. Click **+** to open the **Add a Keyboard Layout** window and change to a different layout.
 - b. Select a layout by browsing the list or use the **Search** field.
 - c. Select the required layout and click **Add**. The new layout appears under the default layout.
 - d. Click **Options** to optionally configure a keyboard switch that can be used to cycle between available layouts. The **Layout Switching Options** window opens.
 - e. To configure key combinations for switching, select one or more key combinations and click **OK** to confirm your selection.



NOTE

Once you select a layout, clicking the **Keyboard** button opens a new dialog box that displays a visual representation of the selected layout.

- a. Click **Done** to apply the settings and return to [Section 4.3, “Installation Summary”](#).
2. From the **Installation Summary** window, click **Language Support**. The **Language Support** window opens. The left pane lists the available language groups. If at least one language from a group is configured, a check mark is displayed and the supported language is highlighted.
 - a. From the left pane, click a group to select additional languages, and from the right pane, select regional options. Repeat this process for all languages.
 - b. Click **Done** to apply the changes and return to [Section 4.3, “Installation Summary”](#).
3. From the **Installation Summary** window, click **Time & Date**. The **Time & Date** window opens.



NOTE

The **Time & Date** window is automatically configured based on the settings you selected in [Section 4.2, “Configuring Language and Location Settings”](#).

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

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



NOTE

You can select a time zone relative to Greenwich Mean Time (GMT) without setting your location to a specific region. To do so, select **Etc** as your region.

- b. From the **City** drop-down menu, select the city, or the city closest to your location in the same time zone.
- c. Toggle the **Network Time** switch to enable or disable network time synchronization using the Network Time Protocol (NTP).

**NOTE**

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

- d. Click **Done** to apply the changes and return to [Section 4.3, “Installation Summary”](#).

**NOTE**

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

4.5. SOFTWARE SETTINGS

This section describes how to configure your installation source and software selection settings. It does not detail all aspects of the GUI, only those that are required to complete the task.

4.5.1. Configuring Installation Source

This section details how to install and configure the full installation image, which is the **recommended** method of installing Red Hat Enterprise Linux 8.0 Beta.

Prerequisites

- You have downloaded the full installation image as detailed in [Section 1.2, “Downloading Beta Installation Images”](#).
- You have created a bootable physical media as detailed in [Section 2.1.2, “Creating a Bootable USB Drive on Linux”](#).
- The **Installation Summary** window is open.

**NOTE**

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

Procedure

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

- a. Review the **Auto-detected installation** section to verify the details. This option is selected by default if you started the installation program from media containing an installation source, for example, a DVD.
- b. Click **Verify** to check the media integrity.
- c. Review the **Additional repositories** section and note that the **Appstream** checkbox is selected by default.



IMPORTANT

- No additional configuration is necessary as the **BaseOS** and **Appstream** repositories are installed as part of the full installation image.
- Do not disable the **Appstream** repository check box if you want a full Red Hat Enterprise Linux 8.0 Beta installation.

2. Select the **On the network** option to download and install packages from a network location instead of local media.



NOTE

- This option is only available when a network connection is active. See [Section 4.6.3, “Network and Host Name”](#) for information on how to set up network connections in the GUI.
- If you do not want to download and install additional repositories from a network location, proceed to [Section 4.5.2, “Configuring Software Selection”](#).

- a. Select the **On the network** drop-down menu to specify the protocol for downloading packages. This setting depends on the server you want to use.



WARNING

The Appstream repository check box is disabled if you select **On the network** and then decide to revert to **Auto-detected installation**. You must select the Appstream check box to enable the Appstream repository.

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

**IMPORTANT**

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

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

**NOTE**

The following steps are optional and are only required if a proxy is used for network access.

- c. Click **Proxy setup...** to configure a proxy for a HTTP or HTTPS source.
- d. Select the **Enable HTTP proxy** check box and type the URL into the **Proxy Host** field.
- e. Select the **Use Authentication** check box if the proxy server requires authentication.
- f. Type in your user name and password.
- g. Click **OK** to finish the configuration and exit the **Proxy Setup...** dialog box.

**NOTE**

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

3. Click **+** to add a repository.
4. Click **-** to delete a repository.
5. Click the arrow icon to replace the current entries with those that were present at the time you opened the **Installation Source** window.
6. To activate or deactivate a repository, click the check box in the **Enabled** column at each entry in the list.

**NOTE**

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

7. Click **Done** to apply the settings and return to [Section 4.3, “Installation Summary”](#).

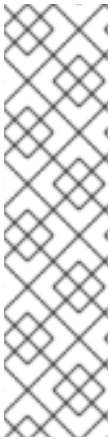
4.5.2. Configuring Software Selection

Use the **Software Selection** window to select a base environment and one or more add-ons. These options determine which software packages are installed on your system during the installation process.

Prerequisites

- You have configured the installation source.

- The installation program downloaded package metadata.
- You selected environments and add-ons.
- The Installation Summary window is open.



NOTE

- It is not possible to select specific packages during a manual installation. You can only select pre-defined environments and add-ons. If you need to control exactly which packages are installed, you must use a Kickstart file and define the packages in the **%packages** section. See [Chapter 5, Kickstart Installation](#) for information on Kickstart installations.
- Environments and add-ons are defined using a **comps.xml** file in your installation source. For example, in the **repodata/** directory on the full installation DVD. Review this file to see which packages are installed as part of a certain environment or add-on.

Procedure

1. From the **Installation Summary** window, click **Software Selection**. The **Software Selection** window opens.
2. From the **Base Environment** pane, select a base environment. Only one environment can be selected.
3. From the **Add-ons for Selected Environment** pane, select one or more add-ons.



NOTE

The list of add-ons is divided into two parts by a horizontal line. Add-ons above this line are defined as part of your chosen environment; if you select a different environment, the available add-ons change. The add-ons displayed below the horizontal line are not specific to your chosen environment.

4. Click **Done** to apply the settings and return to [Section 4.3, “Installation Summary”](#).

4.6. SYSTEM SETTINGS

This section describes how to configure Installation Destination, KDUMP, Network and Host Name, Security Policy, and System Purpose. It does not detail all aspects of the GUI, only those that are required to complete the tasks.

4.6.1. Configuring Installation Destination

Use the **Installation Destination** window to configure the storage options, for example, the disks used as the installation target for your Red Hat Enterprise Linux 8.0 Beta installation. At least one disk must always be selected for the installation to proceed. For information about the theory and concepts behind disk partitioning in Linux, see Partitioning Reference.



WARNING

Back up your data if you plan to use a disk that already contains data. For example, if you want to shrink an existing Microsoft Windows partition and install Red Hat Enterprise Linux 8.0 Beta as a second system, or if you are upgrading a previous release of Red Hat Enterprise Linux. Manipulating partitions always carries a risk. For example, if the process is interrupted or fails for any reason data on the disk can be lost.



IMPORTANT

- Some BIOS types do not support booting from a RAID card. In these instances, the **/boot** partition must be created on a partition outside of the RAID array, such as on a separate hard drive. It is necessary to use an internal hard drive for partition creation with problematic RAID cards. A **/boot** partition is also necessary for software RAID setups. If you have chosen to automatically partition your system, you should manually edit your **/boot** partition.
- To configure the Red Hat Enterprise Linux 8.0 Beta boot loader to *chain load* from a different boot loader, you must specify the boot drive manually by clicking the **Full disk summary and bootloader** link from the **Installation Destination** window.
- When you install Red Hat Enterprise Linux 8.0 Beta on a system with both multipath and non-multipath storage devices, the automatic partitioning layout in the installation program creates volume groups that contain a mix of multipath and non-multipath devices. This defeats the purpose of multipath storage. It is recommended that you select either multipath or non-multipath devices on the **Installation Destination** window. Alternatively, proceed to manual partitioning.

Prerequisites

- You have completed the steps in [Section 4.5, “Software Settings”](#)
- The **Installation Summary** window is open.

Procedure

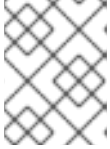
1. From the **Installation Summary** window, click **Installation Destination**. The **Installation Destination** window opens.



NOTE

All locally available storage devices (SATA, IDE and SCSI hard drives, and USB flash drives) are displayed in the **Local Standard Disks** section when the installation program starts. Any storage devices connected after the installation program has started are not detected.

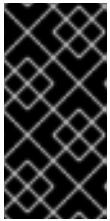
2. From the **Local Standard Disks** section, select the storage device. All storage devices that are used to install Red Hat Enterprise Linux 8.0 Beta are denoted by a white check mark. Disks that do not have a white check mark are not to be used during the installation, they are ignored if you choose automatic partitioning and they are not available in manual partitioning.
3. Click **Refresh** if you want to configure additional local storage devices to connect new hard drives.
4. The **Rescan Disks** dialog box opens.



NOTE

All storage changes made during the installation program are lost once you click **Rescan Disks**.

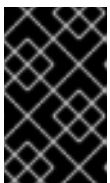
- a. Click **Rescan Disks** and wait until the scanning process completes.
- b. Click **OK** to return to the **Installation Destination** window. All detected disks including any new ones are displayed in the **Local Standard Disks** section.
5. Select one or more storage device(s) that you want to make available during the installation.



IMPORTANT

USB storage devices such as flash drives and external disks are also listed under the **Local Standard Disks** section and are available for selection the same way internal hard drives are. If you use a removable drive to install Red Hat Enterprise Linux 8.0 Beta, your system is unusable if you remove the device.

6. To add a specialized storage device, click **Add a disk...**
The **Storage Device Selection** window opens and lists all storage devices that the installation program has access to. See [Section 4.7.3, “Advanced Storage Options”](#) for information on adding a specialized disk.
7. Under **Storage Configuration**, select the **Automatic** radio button.



IMPORTANT

Automatic partitioning is the **recommended** method of partitioning your storage. You can configure custom partitioning, for more details see [Section 4.8, “Manual Partitioning”](#)

8. To reclaim space from an existing partitioning layout, select the **I would like to make additional space available** check box. For example, if a disk you want to use already contains a different operating system and you want to make this system’s partitions smaller to allow more room for Red Hat Enterprise Linux 8.0 Beta.
9. Select **Encrypt my data** to encrypt all partitions except the ones needed to boot the system (such as **/boot**) using *Linux Unified Key Setup* (LUKS). Encrypting your hard drive is recommended.
10. Click the **Full disk summary and bootloader** link to select which storage device contains the boot loader. For more information, see [Section 4.6.1.1, “Configuring Boot Loader”](#).

**NOTE**

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

11. Click **Done.**

- a. If you selected **Encrypt my data** the **Disk Encryption Passphrase** dialog box opens.
 - i. Type your passphrase into the **Passphrase** and **Confirm** fields.
 - ii. Click **Save Passphrase** to complete disk encryption.

**WARNING**

If you lose the LUKS passphrase, any encrypted partitions and the data on them will become completely inaccessible. There is no way to recover a lost passphrase. However, if you perform a Kickstart installation, you can save encryption passphrases and create backup encryption passphrases during the installation.

- b. If you selected automatic partitioning and the **I would like to make additional space available**, or if there is not enough free space on your selected hard drives to install Red Hat Enterprise Linux 8.0 Beta, the **Reclaim Disk Space** dialog box opens and lists all configured disk devices and all partitions on those devices. The dialog box displays information about how much space the system needs for a minimal installation and how much space you have reclaimed.

**WARNING**

If you used the **Reclaim Space** dialog to **delete** a partition, all data on that partition is lost. If you want to preserve your data, use the **Shrink** option, not the **Delete** option.

- c. Review the displayed list of available storage devices. The **Reclaimable Space** column shows how much space can be reclaimed from each entry.
- d. To reclaim space, select a disk or partition, and click either the **Delete** button to delete that partition, or all partitions on a selected disk, or **Shrink** to use free space on a partition while preserving existing data.

**NOTE**

Alternatively, you can click **Delete all**, this deletes all existing partitions on all disks and makes this space available to Red Hat Enterprise Linux 8.0 Beta. Existing data on all disks is lost.

12. Click **Reclaim space** to apply the changes and return to [Section 4.3, “Installation Summary”](#).

**IMPORTANT**

No changes to any disks are made until you click **Begin Installation** on the **Installation Summary** window. The **Reclaim Space** dialog only marks partitions for resizing or deletion, but no such action is performed immediately.

4.6.1.1. Configuring Boot Loader

Red Hat Enterprise Linux 8.0 Beta uses GRand Unified Bootloader version 2 (**GRUB2**) as its boot loader for AMD64 and Intel 64, IBM Power Systems, and ARM. For IBM Z, the **zipl** boot loader is used.

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

**WARNING**

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

If an operating system is already installed, the Red Hat Enterprise Linux 8.0 Beta installation program attempts to automatically detect and configure the boot loader to start them. If they are not detected, you can manually configure any additional operating systems after you finish the installation.

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

Procedure

1. From the **Installation Destination** window, click the **Full disk summary and bootloader** link. The **Selected Disks** dialog box opens.
The boot loader is installed on the device of your choice, or on a UEFI system; the **EFI system partition** is created on that device during guided partitioning.
2. To change the boot device, select a device from the list and click **Set as Boot Device**. Only one device can be set as the boot device.
3. To disable a new boot loader installation, select the device currently marked for boot and click **Do not install boot loader**. This ensures **GRUB2** is not installed on any device.

**WARNING**

If you choose not to install a boot loader, you cannot boot the system directly and you must use another boot method, such as a stand-alone commercial boot loader application. Use this option only if you have another way to boot your system.

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

4.6.2. Configuring Kdump

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

Procedure

1. From the **Installation Summary** window, click **Kdump**. The **Kdump** window opens.
2. Select the **Enabled kdump** check box.
3. Select either the **Automatic** or **Manual** memory reservation setting.
 - a. If you select **Manual** enter the amount of memory (in megabytes) to be reserved in the **Memory to be reserved** field using the + and - buttons. The **Usable System Memory** readout below the reservation input field shows how much memory is accessible to your main system once your selected amount of RAM is reserved.
4. Click **Done** to apply the settings and return to [Section 4.3, “Installation Summary”](#).

**NOTE**

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

**IMPORTANT**

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

4.6.3. Network and Host Name

The **Network and Host name** window is used to configure network interfaces. Options selected here are available both during the installation for tasks such as downloading packages from a remote location, and on the installed system.

4.6.3.1. Configuring Network and Host Name

This section provides information on configuring network and host name.

Procedure

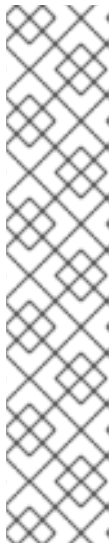
1. From the **Installation Summary** window, click **Network and Host Name**.
2. From the list in the left-hand pane, select an interface. The details are displayed in the right-hand pane.
3. Toggle the **ON/OFF** switch to enable or disable the selected interface.



NOTE

Locally accessible interfaces are automatically detected by the installation program and cannot be manually added or deleted.

4. Click **+** to add a virtual network interface, which can be either: Team, Bond, Bridge, or VLAN.
5. Click **-** to remove a virtual interface.
6. Click **Configure** to change settings such as IP addresses, DNS servers, or routing configuration for an existing interface (both virtual and physical).
7. Type a host name for your system in the **Host Name** field.



NOTE

- There are several types of network device naming standards used to identify network devices with persistent names such as **em1** or **wl3sp0**.
- The host name can be either a fully-qualified domain name (FQDN) in the format `hostname.domainname`, or a short host name with no domain name. Many networks have a Dynamic Host Configuration Protocol (DHCP) service that automatically supplies connected systems with a domain name. To allow the DHCP service to assign the domain name to this machine, only specify the short host name. The value **localhost.localdomain** means that no specific static host name for the target system is configured, and the actual host name of the installed system is configured during the processing of the network configuration, for example, by NetworkManager using DHCP or DNS.

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

4.6.3.2. Adding a Virtual Network Interface

Procedure

1. From the **Network & Host name** window, click the **+** button to add a virtual network interface. The **Add a device** dialog opens.
2. Select one of the four available types of virtual interfaces:

Team, Bond, Bridge, or VLAN

- a. **Bond**: NIC (*Network Interface Controller*) Bonding, a method to bind multiple physical network interfaces together into a single bonded channel.
 - b. **Bridge**: Represents NIC Bridging, a method to connect multiple separate networks into one aggregate network.
 - c. **Team**: NIC Teaming, a new implementation to aggregate links, designed to provide a small kernel driver to implement the fast handling of packet flows, and various applications to do everything else in user space.
 - d. **Vlan** (*Virtual LAN*): A method to create multiple distinct broadcast domains which are mutually isolated.
3. Select the interface type and click **Add**. An editing interface dialog box opens, allowing you to edit any available settings for your chosen interface type. For more information see [Section 4.6.3.3, “Editing Network Interface Configuration”](#).
 4. Click **Save** to confirm the virtual interface settings and return to the **Network & Host name** window.

**NOTE**

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

4.6.3.3. Editing Network Interface Configuration

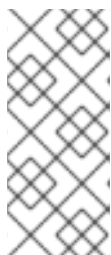
This section contains information on the most important settings for a typical wired connection used during installation. Many of the available options do not have to be changed and are not carried over to the installed system. Configuration of other types of networks is broadly similar, although the specific configuration parameters may be different.

**NOTE**

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

Procedure

1. To configure a network connection manually, select the interface from the **Network & Host name** window and click **Configure**.
An editing dialog specific to the selected interface opens.

**NOTE**

The options presented depend on the connection type - the available options are slightly different depending on whether it is a physical interface (wired or wireless network interface controller) or a virtual interface (Bond, Bridge, Team, or Vlan) that was previously configured in [Section 4.6.3.2, “Adding a Virtual Network Interface”](#).

The three most common and useful options in the editing dialog are:

4.6.3.4. Enabling or Disabling the Interface Connection

Procedure

1. Click the **General** tab.
2. Select the **Automatically connect to this network when it is available** check box to enable connection by default.



NOTE

- When enabled on a wired connection, the system typically connects during startup (unless you unplug the network cable). On a wireless connection, the interface attempts to connect to any known wireless networks in range.
- You can enable or disable all users on the system from connecting to this network using the **All users may connect to this network** option. If you disable this option, only **root** will be able to connect to this network.
- It is not possible to only allow a specific user other than **root** to use this interface, because no other users are created at this point during the installation. If you need a connection for a different user, you must configure it after the installation.

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

4.6.3.5. Setting up Static IPv4 or IPv6 Settings

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

Procedure

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



NOTE

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

2. Click **Add** and enter your address settings.
3. Type the IP addresses in the **Additional DNS servers** field; it accepts one or more IP addresses of DNS servers, for example, **10.0.0.1,10.0.0.8**.
4. Select the **Require IPvX addressing for this connection to complete** check box.

**NOTE**

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

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

4.6.3.6. Configuring Routes

Procedure

1. In the **IPv4 Settings** and **IPv6 Settings** tabs, click **Routes** to configure routing settings for a specific IP protocol on an interface. An editing routes dialog specific to the interface opens.
2. Click **Add** to add a route.
3. Select the **Ignore automatically obtained routes** check box to configure at least one static route and want to disable all routes not specifically configured.
4. Select the **Use this connection only for resources on its network** check box to prevent the connection from becoming the default route.

**NOTE**

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

5. Click **OK** to save your settings and return to the editing routes dialog specific to the interface.
6. Click **Save** to apply the settings and return to the **Network and Host Name** window.

4.6.4. Security Policy

This section provides information on the Red Hat Enterprise Linux 8.0 Beta security policy add-on and how to configure it for use on your system.

4.6.4.1. About Security Policy

The Red Hat Enterprise Linux 8.0 Beta security policy adheres to restrictions and recommendations (compliance policies) defined by the Security Content Automation Protocol (SCAP) standard. The packages are automatically installed. However, by default, no policies are enforced and therefore no checks are performed during or after installation unless specifically configured.

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

compliance and vulnerability scanning. After the installation finishes, the system is automatically scanned to verify compliance. The results of this scan are saved to the **/root/openscap_data** directory on the installed system. You can also load additional profiles from a HTTP, HTTPS, or FTP server.

4.6.4.2. Configuring a Security Policy

This section provides information on how to configure a security policy.

Prerequisites

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

Procedure

1. From the **Installation Summary** window, click **Security Policy**. The **Security Policy** window opens.
2. To enable security policies on the system, toggle the **Apply security policy** switch to **ON**.
3. Select one of the profiles listed in the top pane.
4. Click **Select profile**.
Profile changes that you must apply before installation appear in the bottom pane.



NOTE

The default profiles do not require changes before installation. However, loading a custom profile can require pre-installation tasks.

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



NOTE

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

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

4.6.4.3. Related information

- Pre-defined policies are provided by **SCAP Security Guide**. See the OpenSCAP website for information.

4.6.5. System Purpose

System administrators use System Purpose to record the intended use of a Red Hat Enterprise Linux 8.0 Beta system by the organization. When a system's purpose is recorded, entitlements are automatically attached.

You can enter System Purpose data in the following ways:

- During Composer image creation.
- During Installation using the Anaconda GUI and Kickstart automation scripts.
- Using Runtime operations, for example command line and Cockpit.

You can configure the following components:

- **Role:** Server, Desktop, Workstation, or HPC compute node systems.
- **Service Level Agreement:** Premium, Standard, or Self-Support service level.
- **Usage:** Production or Disaster Recovery environments.

Benefits include:

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

4.6.5.1. Configuring System Purpose using Anaconda



NOTE

While it is strongly recommended that you configure System Purpose, it is an optional feature of the Red Hat Enterprise Linux 8.0 Beta installation program. You can select the **Not Specified** option if you do not want to configure System Purpose. If you want to enable System Purpose after the installation completes, you can do so using the **syspurpose** Kickstart command.

Procedure

1. From the **Installation Summary** window, click **System Purpose**.
2. Select the required system role from the **Role** pane.
3. Select the required service level agreement from the **Red Hat Service Level Agreement** pane.
4. Select the required usage from the **Usage** pane.
5. Click **Done** to apply the settings and return to the **Installation Summary** window.

The System Purpose data is now available for Subscription Manager to auto-attach to the system. See System Manager documentation for further System Purpose information.

4.6.5.2. Configuring System Purpose using Kickstart

Use the **syspurpose** command to configure System Purpose in a Kickstart configuration file.

The following actions are available:

role

Set the intended role of the system. This action uses the following format:

```
syspurpose --role=
```

The role assigned can be:

- **Desktop**
- **Server**
- **Workstation**
- **Computenode**
- **Not Specified**

sla

Set the intended sla of the system. This action uses the following format:

```
syspurpose --sla=
```

The sla assigned can be:

- **Premium**
- **Standard**
- **Self Support**
- **Not Specified**

usage

Set the intended usage of the system. This action uses the following format:

```
syspurpose --usage=
```

The usage assigned can be:

- **Production**
- **Disaster Recovery**
- **Not Specified**

addon

Set additional layered products or features. To add multiple items specify **--addon** multiple times, once per layered product/feature:

```
syspurpose --addon=
```

4.7. STORAGE DEVICES

You can install Red Hat Enterprise Linux 8.0 Beta on a large variety of storage devices. You can configure basic, locally accessible, storage devices in the **Installation Destination** window. Basic storage devices directly connected to the local system, such as hard disk drives and solid-state drives, are displayed in the **Local Standard Disks** section of the window. On System z, this contains activated Direct Access Storage Devices (DASDs).



WARNING

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

4.7.1. Storage Device Selection

The storage device selection window lists all storage devices to which Anaconda has access. Depending on your system and available hardware, some tabs might not be displayed. The devices are grouped under the following tabs:

Multipath Devices

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



IMPORTANT

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

Other SAN Devices

Devices available on a Storage Area Network (SAN).

Firmware RAID

Storage devices attached to a firmware RAID controller.

NVDIMM Devices

Under specific circumstances, Red Hat Enterprise Linux 8.0 Beta can boot and run from (NVDIMM) devices in sector mode on the Intel 64 and AMD64 architectures.

System z Devices

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

4.7.2. Filtering Storage Devices

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

Procedure

1. From the **Installation Summary** window, click **Installation Destination**. The **Installation Destination** window opens, listing all available drives.
2. Under the **Specialized & Network Disks** section, click **Add a disk....** The storage devices selection window opens.
3. Click the **Search by** tab to search by port, target, LUN, or WWID. Searching by WWID or LUN requires additional values in the corresponding input text fields.
4. Click the required option from the **Search** drop-down menu.
5. Click **Find** to start the search. Each device is presented on a separate row with a corresponding check box.
6. Select the check box to make the device available during the installation process. Later in the installation process you can choose to install Red Hat Enterprise Linux 8.0 Beta on any of the selected devices and you can choose to automatically mount any of the other selected devices as part of the installed system.



NOTE

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

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



IMPORTANT

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

4.7.3. Advanced Storage Options

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

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

**NOTE**

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

4.7.3.1. Discovering and Starting an iSCSI Session

This section describes how to discover and start an iSCSI session.

Procedure

1. From the **Installation Summary** window, click **Installation Destination**. The Installation Destination window opens, listing all available drives.
2. Under the **Specialized & Network Disks** section, click **Add a disk...** The storage devices selection window opens.
3. Click **Add iSCSI target...** The **Add iSCSI Storage Target** window opens.
4. Enter the IP address of the iSCSI target in the **Target IP Address** field.
5. Type a name in the **iSCSI Initiator Name** field for the iSCSI initiator in iSCSI qualified name (IQN) format. A valid IQN entry contains the following information:
 - The string **iqn.** (note the period).
 - A date code that specifies the year and month in which your organization's Internet domain or subdomain name was registered, represented as four digits for the year, a dash, and two digits for the month, followed by a period. For example, represent September 2010 as **2010-09.**
 - Your organization's Internet domain or subdomain name, presented in reverse order with the top-level domain first. For example, represent the subdomain **storage.example.com** as **com.example.storage.**
 - A colon followed by a string that uniquely identifies this particular iSCSI initiator within your domain or subdomain. For example, **:diskarrays-sn-a8675309.**

A complete IQN is as follows: **iqn.2010-09.storage.example.com:diskarrays-sn-a8675309**. **Anaconda** prepopulates the **iSCSI Initiator Name** field with a name in this format to help you with the structure.

For more information on IQNs, see 3.2.6. *iSCSI Names* in *RFC 3720 - Internet Small Computer Systems Interface (iSCSI)* available from tools.ietf.org and 1. *iSCSI Names and Addresses* in *RFC 3721 - Internet Small Computer Systems Interface (iSCSI) Naming and Discovery* available from tools.ietf.org.
6. Select the **Discovery Authentication Type** drop-down menu to specify the type of authentication to use for iSCSI discovery. The following options are available:
 - No credentials
 - CHAP pair
 - CHAP pair and a reverse pair

7. a. If you selected **CHAP pair** as the authentication type, provide the user name and password for the iSCSI target in the **CHAP Username** and **CHAP Password** fields.
- b. If you selected **CHAP pair and a reverse pair** as the authentication type, provide the user name and password for the iSCSI target in the **CHAP Username** and **CHAP Password** field and the user name and password for the iSCSI initiator in the **Reverse CHAP Username** and **Reverse CHAP Password** fields.
8. Optionally, select the **Bind targets to network interfaces** check box.
9. Click **Start Discovery**.
Anaconda attempts to discover an iSCSI target based on the information provided. If discovery succeeds, the **Add iSCSI Storage Target** window displays a list of all iSCSI nodes discovered on the target.
10. Select the required node check boxes to use for installation.



NOTE

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

11. Click the additional **Use the credentials from discovery** drop-down menu. When the proper credentials have been provided, the **Log In** button becomes available.
12. Click **Log In** to initiate an iSCSI session.

4.7.3.2. Configuring FCoE Parameters

This section describes how to configure FCoE parameters.

Procedure

1. From the **Installation Summary** window, click **Installation Destination**. The **Installation Destination** window opens, listing all available drives.
2. Under the **Specialized & Network Disks** section, click **Add a disk...**. The storage devices selection window opens.
3. Click **Add FCoE SAN...**. A dialog box opens for you to configure network interfaces for discovering FCoE storage devices.
4. Select a network interface that is connected to an FCoE switch in the **NIC** drop-down menu.
5. Click **Add FCoE disk(s)** to scan the network for SAN devices.
6. Select the required check boxes:
 - Use DCB: *Data Center Bridging* (DCB) is a set of enhancements to the Ethernet protocols designed to increase the efficiency of Ethernet connections in storage networks and clusters. Select the check box to enable or disable the installation program's awareness of

DCB. This option should only be enabled for network interfaces that require a host-based DCBX client. Configurations on interfaces that implement a hardware DCBX client should disable the check box.

- Use auto vlan: *Auto VLAN* indicates whether VLAN discovery should be performed. If this check box is enabled, then the FIP (FCoE Initiation Protocol) VLAN discovery protocol will run on the Ethernet interface once the link configuration has been validated. If they are not already configured, network interfaces for any discovered FCoE VLANs will be automatically created and FCoE instances will be created on the VLAN interfaces. This option is enabled by default.
7. Discovered FCoE devices are displayed under the **Other SAN Devices** tab in the **Installation Destination** window.

4.7.3.3. Configuring DASD Storage Devices

This section describes how to configure DASD storage devices.

Procedure

1. From the **Installation Summary** window, click **Installation Destination**. The Installation Destination window opens, listing all available drives.
2. Under the **Specialized & Network Disks** section, click **Add a disk...** The storage devices selection window opens.
3. Click **Add DASD**. The **Add DASD Storage Target** dialog box opens and allows you to attach additional DASDs that were not detected when the installation started.
The **Add DASD Storage Target** dialog box prompts you to specify a device number, such as **0.0.0204**.
4. Type the device number of the DASD you want to attach in the **Device number** field.
5. Click **Start Discovery**.



NOTE

- If a DASD with the specified device number is found and if it is not already attached, the dialog box closes and the newly-discovered drives appear in the list of drives. You can then select the check boxes to select the drives that should be made available. After you do so, click **Done**. The new DASDs are available for selection (marked as **DASD device 0.0.xxxx**) in the **Local Standard Disks** section of the **Installation Destination** window.
- If you entered an invalid device number, or if the DASD with the specified device number is already attached to the system, an error message appears in the dialog box, explaining the error and prompting you to try again with a different device number.

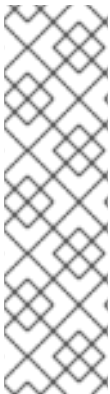
4.7.3.4. Configuring FCP Devices

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

Procedure

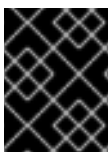
1. From the **Installation Summary** window, click **Installation Destination**. The Installation Destination window opens, listing all available drives.
2. Under the **Specialized & Network Disks** section, click **Add a disk....** The storage devices selection window opens.
3. Click **Add ZFCP LUN**. The **Add zFCP Storage Target** dialog box opens allowing you to add a FCP (Fibre Channel Protocol) storage device.
IBM Z requires that any FCP device is entered manually for the installation program to activate FCP LUNs. This can be done either in **Anaconda** interactively, or specified as a unique parameter entry in the parameter or CMS configuration file. The values entered here are unique to each site in which they are set up.
4. Type the 4 digit hexadecimal device number in the **Device number** field.
5. Type the 16 digit hexadecimal World Wide Port Number (WWPN) in the **WWPN** field.
6. Type the 16 digit hexadecimal FCP LUN identifier in the **LUN** field.
7. Click **Start Discovery** to connect to the FCP device.

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



NOTE

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



IMPORTANT

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

4.7.4. Installing to an NVDIMM Device

Non-Volatile Dual In-line Memory Module (NVDIMM) devices combine the performance of RAM with disk-like data persistence when no power is supplied. Under specific circumstances, Red Hat Enterprise Linux 8.0 Beta can boot and run from (NVDIMM) devices.

4.7.4.1. Criteria for using an NVDIMM Device as an Installation Target

Red Hat Enterprise Linux 8.0 Beta can be installed to Non-Volatile Dual In-line Memory Module (NVDIMM) devices in sector mode on the Intel 64 and AMD64 architectures, supported by the **nd_pmem** driver.

Using an NVDIMM Device as Storage

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

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

Booting from an NVDIMM Device

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

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

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

4.7.4.2. Configuring an NVDIMM Device using Anaconda

A Non-Volatile Dual In-line Memory Module (NVDIMM) device must be properly configured for use by Red Hat Enterprise Linux 8.0 Beta and **Anaconda** as an installation target.



WARNING

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

Prerequisites

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

Procedure

1. From the **Installation Summary** window, click **Installation Destination**. The **Installation Destination** window opens, listing all available drives.
2. Under the **Specialized & Network Disks** section, click **Add a disk...**. The storage devices selection window opens.

3. Click the **NVDIMM Devices** tab.
4. To reconfigure a device select it from the list.
If a device is not listed, it is not in sector mode.
5. Click **Reconfigure NVDIMM....** A reconfiguration dialog opens.
6. Enter the required sector size and click **Start Reconfiguration**.
The supported sector sizes are 512 and 4096 bytes.
7. Once reconfiguration completes click **OK**.
8. Select the device check box.
9. Click **Done** to return to the **Installation Destination** window.
The NVDIMM device you reconfigured is displayed in the **Specialized & Network Disks** section.
10. Click **Done** to return to the **Installation Summary** window.

The NVDIMM device is now available for selection as an installation target. Additionally, if the device meets the requirements for booting, it can be set as such.

4.7.4.3. Configuring an NVDIMM Device using Kickstart

By default, all Non-Volatile Dual In-line Memory Module (NVDIMM) devices are ignored by the installation program. To manipulate an NVDIMM device in a Kickstart configuration file, use the **nvdimm** command:

```
nvdimm action [options]
```

The following actions are available:

reconfigure

Reconfigure a specific NVDIMM device to a given mode. Additionally, the specified device is implicitly marked in use, so a subsequent **nvdimm use** command for the same device is redundant. This action uses the following format:

```
nvdimm reconfigure [--namespace=NAMESPACE] [--mode=MODE] [--  
sectorsize=SECTORSIZE]
```

- **--namespace=** - The device specification by namespace. For example:

```
nvdimm reconfigure --namespace=namespace0.0 --mode=sector --  
sectorsize=512
```

- **--mode=** - The mode specification. Currently, only the value **sector** is available.
- **--sectorsize=** - Size of a sector for sector mode. For example:

```
nvdimm reconfigure --namespace=namespace0.0 --mode=sector --  
sectorsize=512
```

The supported sector sizes are 512 and 4096 bytes.

use

Specify a NVDIMM device as a target for installation. The device must be already configured in the sector mode. This action uses the following format:

```
nvdimm use [--namespace=NAMESPACE|--blockdevs=DEVICES]
```

- **--namespace=** - Specifies the device by namespace. For example:

```
nvdimm use --namespace=namespace0.0
```

- **--blockdevs=** - Specifies a comma-separated list of block devices corresponding to the NVDIMM devices to be used. The asterisk `*` wildcard is supported. For example:

```
nvdimm use --blockdevs=pmem0s,pmem1s
nvdimm use --blockdevs=pmem*
```

4.8. MANUAL PARTITIONING

Manual Partitioning allows you to configure your disk partitions and mount points. This defines the file system that Red Hat Enterprise Linux 8.0 Beta is installed on.

An installation of Red Hat Enterprise Linux 8.0 Beta requires a minimum of one partition but Red Hat recommends using at least the following partitions or volumes: `/`, `/home`, `/boot`, and `swap`. You can also create additional partitions and volumes as you require. See [Section 4.8.5, “Recommended Partitioning Scheme”](#) for more information.



WARNING

It is recommended that you back up data before proceeding. If you are upgrading or creating a dual-boot system, you should back up any data you want to keep on your storage devices. Unforeseen circumstances can result in data loss.

4.8.1. Starting Manual Partitioning

Prerequisites

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

Procedure

1. Select disks for installation:
 - a. Click **Installation Destination** to open the **Installation Destination** window.

- b. Select the required disks for installation by clicking the corresponding icon. A selected disk has a check-mark displayed on it.
 - c. Under **Storage Configuration**, select the **Custom** radio-button.
 - d. Optional: To enable storage encryption with LUKS, select the **Encrypt my data** check box.
 - e. Click **Done**.
2. If you selected to encrypt the storage, a dialog box for entering a disk encryption passphrase opens. Type in the passphrase:
 - a. Enter the passphrase into the two text fields. To switch keyboard layout, use the keyboard icon.

**WARNING**

In the dialog box for entering the passphrase, you are not able to change keyboard layout. Select the English keyboard layout to enter the passphrase in the installation program.

- b. The dialog box provides an assessment of the passphrase strength. Change your passphrase if necessary.
 - c. Click **Save Passphrase** to save the passphrase.
The Manual Partitioning window opens.
3. Mount points that the installation program has detected are listed in the left-hand pane. The mount points are organized by detected operating system installations. As a result, some file systems may be displayed multiple times if a partition is shared among several installations.
 - a. Select the mount points in the left pane; the customizable options are displayed in the right pane.
 - b. If your system contains existing file systems, ensure that enough space is available for the installation. To remove any partitions that you do not want to be present, select them in the list and click the - button.
The dialog has a check box to remove all other partitions used by the system to which the deleted partition belongs.
 - c. If there are no existing partitions and you want to create the recommended set of partitions as a starting point, select your preferred partitioning scheme from the left pane (default for Red Hat Enterprise Linux is LVM) and click the **Click here to create them automatically** link.
A **/boot** partition, a **/** (root) volume, and a **swap** volume proportionate to the size of the available storage are created and listed in the left pane. These are the recommended file systems for a typical installation, but you can add additional file systems and mount points.
 - d. Continue with [adding mount points](#), [configuring the individual mount points](#), and [configuring the underlying partitions or volumes](#).

4.8.2. Adding a Mount Point File System

You can add multiple mount point file systems.

Prerequisites

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

Procedure

1. Click **+** to create a new mount point file system. The **Add a New Mount Point** dialog opens.
2. Select one of the preset paths from the **Mount Point** drop-down menu or type your own; for example, select **/** for the root partition or **/boot** for the boot partition.
3. Type the size of the file system in the **Desired Capacity** field; for example, **2GiB**.



WARNING

If you leave the **Desired Capacity** field empty or if you specify a size bigger than available space, then all remaining free space is used.

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

4.8.3. Configuring a Mount Point File System

You can set the partitioning scheme for each mount point that was created manually. The available options are **Standard Partition**, **LVM**, and **LVM Thin Provisioning**.



NOTE

- BTRFS support has been deprecated in Red Hat Enterprise Linux 8.0 Beta.
- The **/boot** partition is always located on a standard partition, regardless of the value selected.

Procedure

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

2. Under the **Device(s)** heading, click **Modify...** The **Configure Mount Point** dialog opens.
3. Select one or more devices and click **Select** to confirm your selection and return to the **Manual Partitioning** window.
4. Click **Update Settings** to apply the changes.



NOTE

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

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

4.8.4. Customizing a Partition or Volume

Customizing a partition or volume is available if you want to set specific settings.

Procedure

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

Figure 4.2. Customizing Partitions

The screenshot shows the 'MANUAL PARTITIONING' window for 'RED HAT ENTERPRISE LINUX 8.0 INSTALLATION'. The left pane lists partitions: /boot (1024 MiB), / (rhel-root, 17 GiB), and swap (2 GiB). The 'rhel-root' partition is selected. The right pane shows configuration options for 'rhel-root': Mount Point (/), Device(s) (ATA QEMU HARDDISK (sda)), Desired Capacity (17 GiB), Device Type (LVM), File System (xfs), Volume Group (rhel), and Name (root). There are buttons for 'Done', 'Help!', 'Update Settings', and 'Reset All'. A note at the bottom states: 'Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.'

2. You can customize the following options in the right pane:

- a. Enter the file system's mount point into the **Mount Point** field. For example, if a file system is the root file system, enter `/`; enter `/boot` for the `/boot` file system, and so on. For a swap file system, the mount point should not be set as setting the file system type to **swap** is sufficient.
- b. Enter the size of the file system in the **Desired Capacity** field. You can use common size units such as KiB or GiB. The default is MiB if no other unit is specified.
- c. Select the required device type from the drop-down **Device Type** menu: **Standard Partition**, **LVM**, or **LVM Thin Provisioning**.

**NOTE**

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

- d. Select the **Encrypt** check box to encrypt the partition or volume. You are required to set a password later in the installation program. The **LUKS Version** drop-down menu is displayed.
- e. Select the required LUKS version from the drop-down menu.
- f. Select the appropriate file system type for this partition or volume from the **File system** drop-down menu.
- g. Select the **Reformat** check box to format an existing partition, or deselect it to retain your data. The newly-created partitions and volumes must be reformatted, and the check box cannot be deselected.
- h. Type a label for the partition in the **Label** field. Labels are used for you to easily recognize and address individual partitions.
- i. Type a name in the **Name** field.

**NOTE**

Note that standard partitions are named automatically when they are created and their name cannot be edited, such as `/boot` being assigned the name **sda1**.

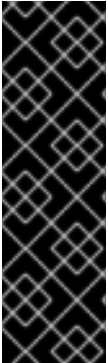
3. Click **Update Settings** to apply your changes and if required, select another partition to customize. Changes are not applied until you start the installation from the **Installation Summary** window.

**NOTE**

Click **Reset All** to discard partition changes and start over.

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

5. Click **Cancel & Return to Custom Partitioning** to return to the **Manual Partitioning** window and make additional changes.
6. Click **Accept Changes** to apply the changes and return to the **Installation Summary** window.



IMPORTANT

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

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

4.8.4.1. Creating Software RAID

This section describes how to create a RAID device. Redundant Arrays of Independent Disks (RAIDs) are constructed from multiple storage devices that are arranged to provide increased performance and, in some configurations, greater fault tolerance.

A RAID device is created in one step and disks are added or removed as necessary. One RAID partition per physical disk is allowed for each device, so the number of disks available to the installation program determines the levels of RAID device available. For example, if your system has two hard drives, the installation program does not allow you to create a RAID10 device, as it requires 4 separate partitions.



NOTE

On IBM Z, the storage subsystem uses RAID transparently. There is no need to set up a software RAID manually.

Prerequisites

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

Procedure

1. From the left pane of the **Manual Partitioning** window, select the required partition.
2. Under the **Device(s)** section, click **Modify**. The **Configure Mount Point** dialog box opens.
3. Select the disks that are to be included in the RAID device and click **Select**.
4. Click the **Device Type** drop-down menu and select **RAID**.
5. Click the **File System** drop-down menu and select your preferred file system type.

6. Click the **RAID Level** drop-down menu and select your preferred level of RAID.
7. Click **Update Settings** to save your changes.
8. Click **Done** to apply the settings and return to the **Installation Summary** window.

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

4.8.4.2. Creating an LVM Logical Volume

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



NOTE

Note that LVM configuration is only available in the graphical installation program.



IMPORTANT

During text-mode installation, LVM configuration is not available. To create an LVM configuration, press **Ctrl+Alt+F2** to use a different virtual console, and run the **lvm** command. To return to the text-mode installation, press **Ctrl+Alt+F1**.

Procedure

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



NOTE

The configuration dialog does not allow you to specify the size of the volume group's physical extents. The size is always be set to the default value of 4 MiB. If you want to create a volume group with different physical extents, create it manually by switching to an interactive shell and using the **vgcreate** command, or use a Kickstart file with the **volgroup --pesize=size** command.

4.8.4.3. Configuring an LVM Logical Volume

This section describes how to configure a newly-created LVM logical volume.

Procedure

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



NOTE

The configuration dialog does not allow you to specify the size of the volume group's physical extents. The size is always set to the default value of 4 MiB. If you want to create a volume group with different physical extents, create it manually by switching to an interactive shell and using the **vgcreate** command, or use a Kickstart file with the **volgroup --pesize=size** command.

4. From the **RAID Level** drop-down menu, select the required RAID level.
The available RAID levels are the same as with actual RAID devices.
5. Select the **Encrypt** check box to mark the volume group for encryption.
6. From the **Size policy** drop-down menu, select the size policy for the volume group.
The available policy options are:
 - **Automatic:** The size of the volume group is set automatically so that it is large enough to contain the configured logical volumes. This is optimal if you do not need free space within the volume group.
 - **As large as possible:** The volume group is created with maximum size, regardless of the size of the configured logical volumes it contains. This is optimal if you plan to keep most of your data on LVM and later need to increase the size of some existing logical volumes, or if you need to create additional logical volumes within this group.
 - **Fixed:** You can set an exact size of the volume group. Any configured logical volumes must then fit within this fixed size. This is useful if you know exactly how large you need the volume group to be.
7. Click **Save** to apply the settings and return to the **Manual Partitioning** window.
8. Click **Update Settings** to save your changes
9. Click **Done** to return to the **Installation Summary** window.



WARNING

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

4.8.5. Recommended Partitioning Scheme

Red Hat recommends that you create separate file systems at the following mount points:

- **/boot**
- **/ (root)**
- **/home**
- **swap**

/boot partition - recommended size at least 1 GiB

The partition mounted on **/boot** contains the operating system kernel, which allows your system to boot Red Hat Enterprise Linux 8.0 Beta, along with files used during the bootstrap process. Due to the limitations of most firmwares, creating a small partition to hold these is recommended. In most scenarios, a 1 GiB boot partition is adequate. Unlike other mount points, using an LVM volume for **/boot** is not possible - **/boot** must be located on a separate disk partition.



WARNING

Normally, the **/boot** partition is created automatically by the installation program. However, if the **/** (root) partition is larger than 2 TiB and (U)EFI is used for booting, you need to create a separate **/boot** partition that is smaller than 2 TiB to boot the machine successfully.



NOTE

If you have a RAID card, be aware that some BIOS types do not support booting from the RAID card. In such a case, the **/boot** partition must be created on a partition outside of the RAID array, such as on a separate hard drive.

root - recommended size of 10 GiB

This is where **/**, or the root directory, is located. The root directory is the top-level of the directory structure. By default, all files are written to this file system unless a different file system is mounted in the path being written to, for example, **/boot** or **/home**.

While a 5 GiB root file system allows you to install a minimal installation, it is recommended to allocate at least 10 GiB so that you can install as many package groups as you want.



IMPORTANT

Do not confuse the **/** directory with the **/root** directory. The **/root** directory is the home directory of the root user. The **/root** directory is sometimes referred to as *slash root* to distinguish it from the root directory.

/home - recommended size at least 1 GiB

To store user data separately from system data, create a dedicated file system for the **/home** directory. Base the file system size on the amount of data that is stored locally, number of users, and so on. You can upgrade or reinstall Red Hat Enterprise Linux 8.0 Beta without erasing user data files. If you select automatic partitioning, it is recommended to have at least 55 GiB of disk space available for the installation, to ensure that the **/home** file system is created.

swap partition - recommended size at least 1 GB

Swap file systems support virtual memory; data is written to a swap file system when there is not enough RAM to store the data your system is processing. Swap size is a function of system memory workload, not total system memory and therefore is not equal to the total

system memory size. It is important to analyze what applications a system will be running and the load those applications will serve in order to determine the system memory workload. Application providers and developers can provide guidance.

When the system runs out of swap space, the kernel terminates processes as the system RAM memory is exhausted. Configuring too much swap space results in storage devices being allocated but idle and is a poor use of resources. Too much swap space can also hide memory leaks. The maximum size for a swap partition and other additional information can be found in the **mkswap(8)** manual page.

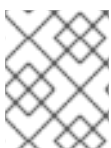
The following table provides the recommended size of a swap partition depending on the amount of RAM in your system and if you want sufficient memory for your system to hibernate. If you let the installation program partition your system automatically, the swap partition size is established using these guidelines. Automatic partitioning setup assumes hibernation is not in use. The maximum size of the swap partition is limited to 10 percent of the total size of the hard drive, and the installation program cannot create swap partitions more than 128 GB in size. To set up enough swap space to allow for hibernation, or if you want to set the swap partition size to more than 10 percent of the system's storage space, or more than 128 GB, you must edit the partitioning layout manually.

Amount of RAM in the system	Recommended swap space	Recommended swap space if allowing for hibernation
Less than 2 GB	2 times the amount of RAM	3 times the amount of RAM
2 GB - 8 GB	Equal to the amount of RAM	2 times the amount of RAM
8 GB - 64 GB	4 GB to 0.5 times the amount of RAM	1.5 times the amount of RAM
More than 64 GB	Workload dependent (at least 4GB)	Hibernation not recommended

At the border between each range, for example, a system with 2 GB, 8 GB, or 64 GB of system RAM, discretion can be exercised with regard to chosen swap space and hibernation support. If your system resources allow for it, increasing the swap space can lead to better performance.

Distributing swap space over multiple storage devices - particularly on systems with fast drives, controllers and interfaces - also improves swap space performance.

Many systems have more partitions and volumes than the minimum required. Choose partitions based on your particular system needs.



NOTE

Only assign storage capacity to those partitions you require immediately. You can allocate free space at any time, to meet needs as they occur.



NOTE

If you are unsure about how to configure partitions, accept the automatic default partition layout provided by the installation program.

4.9. STARTING THE INSTALLATION PROGRAM

Starting the installation program requires the configuration of your root password and user settings.

4.9.1. Beginning Installation

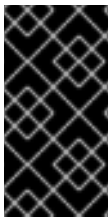
When you have started the installation process, it is not possible to go back to the **Installation Summary** window and change any settings. To change settings, you must wait for the installation process to finish, reboot your system, log in, and change your settings on the installed system.

Prerequisites

- You have completed all configuration steps in [Section 4.3, “Installation Summary”](#).
- The **Installation Summary** window is open.

Procedure

1. From the **Installation Summary** window, click **Begin Installation**. The **Configuration** window opens and the installation process starts.
Two user setting options, **Root Password** (mandatory) and **User Creation** (optional) are available.

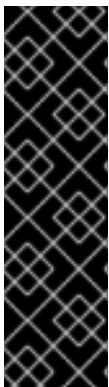


IMPORTANT

Before you finish the installation and reboot, either remove the media (CD, DVD, or a USB drive) used to start the installation, or verify that your system tries to boot from the hard drive before attempting removable media. Otherwise, your system starts the installation program again, instead of the installed system.

4.9.2. Configuring a Root Password

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



IMPORTANT

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

**WARNING**

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

Procedure

1. From the **Configuration** window, click **Root Password**. The **Root Password** window opens.
2. Type your password in the **Root Password** field. For security purposes, the characters are displayed as dots.
 - a. The requirements and recommendations for creating a strong root password are:
 - i. *Must* be at least eight characters long
 - ii. May contain numbers, letters (upper and lower case) and symbols
 - iii. Is case-sensitive
3. Type the same password in the **Confirm** field.
4. Click **Done** to confirm your root password and return to [Section 4.9.1, “Beginning Installation”](#).

**NOTE**

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

4.9.3. Creating a User Account

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

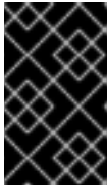
Procedure

1. From the **Configuration** window, click **User Creation**. The **Create User** window opens.
2. Type the user account name into the **Full name** field, for example: John Smith.
3. Type the username into the **User name** field, for example: jsmith.

**NOTE**

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

4. Select the **Make this user administrator** check box if the user requires administrative rights (it is added into the **wheel** group).



IMPORTANT

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

5. Select the **Require a password to use this account** check box.



WARNING

If you give administrator privileges to a user, verify that the account is password protected. Never give a user administrator privileges without assigning a password to the account.

6. Type a password into the **Password** field.
7. Type the same password into the **Confirm password** field.
8. Click **Save Changes** to apply the changes and return to the **Configuration** window.
9. Click **Reboot** to reboot and log in to your Red Hat Enterprise Linux 8.0 Beta system.

4.9.3.1. Configuring Advanced User Settings

You can change the default settings for the user account in the **Advanced User Configuration** dialog box.

Procedure

1. Change the details in the **Home directory** field, if required. The field is populated by default with `/home/username`.
2. In the **User and Groups IDs** section you can:
 - a. Select the **Specify a user ID manually** check box and use the **+** or **-** to enter the required value.



NOTE

The default value is 1000. UIDs 0-999 are reserved by the system so they can not be assigned to a user.

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

**NOTE**

The default group name is the same as the user name, and its default GID is 1000. GIDs 0-999 are reserved by the system so they can not be assigned to a user's group.

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

**NOTE**

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

4. Click **Save Changes** to apply the updates and return to the **Configuration** window.

4.9.4. Installation Complete

Congratulations! Your Red Hat Enterprise Linux 8.0 Beta installation is complete! Remove any installation media if it is not ejected automatically upon reboot.

After your system's normal power-up sequence completes, Red Hat Enterprise Linux 8.0 Beta loads and starts. By default, the start process is hidden behind a graphical screen that displays a progress bar. When complete, a GUI login window (or if the X Window System is not installed, a **login:** prompt) is displayed.

If your system was installed with the X Window System, applications to set up your system are launched the first time you start your Red Hat Enterprise Linux 8.0 Beta system. These applications guide you through initial configuration and you can set your system time and date, register your machine with Red Hat Network, and more.

CHAPTER 5. KICKSTART INSTALLATION

Kickstart installations offer a means to automate the installation process, either partially or fully. Kickstart files contain answers to all questions normally asked by the installation program, such as what time zone do you want the system to use, how should the drives be partitioned or which packages should be installed. Providing a prepared Kickstart file when the installation begins therefore allows you to perform the installation automatically, without need for any intervention from the user. This is especially useful when deploying Red Hat Enterprise Linux on a large number of systems at once.

All Kickstart scripts and the log files of their execution are stored in the `/tmp` directory to assist with debugging installation issues. After the installation is finished, the installed system will contain a Kickstart file at `/root/anaconda-ks.cfg`. If you used a Kickstart file during the installation, the same file will be stored on the installed system; if you installed the system manually, a file generated from the choices you made in the installer will be saved instead. This file can then be used to reproduce the installation in exactly the same way as before. Alternatively you can modify the file as you wish and use it to install a different system.

The following sections will explain the details of performing an installation using a Kickstart file.

5.1. CREATING A KICKSTART FILE

The Kickstart file itself is a plain text file, containing keywords listed in [Section 5.4, “Kickstart Syntax Reference”](#), which serve as directions for the installation. Any text editor able to save files as ASCII text (such as **Gedit** or **vim** on Linux systems or **Notepad** on Windows systems) can be used to create and edit Kickstart files.

The recommended approach to creating Kickstart files is to perform a manual installation on one system first. After the installation completes, all choices made during the installation are saved into a file named **anaconda-ks.cfg**, located in the `/root/` directory on the installed system. You can then copy this file, make any changes you need, and use the resulting configuration file in further installations.

When creating a Kickstart file, keep in mind the following:

- Lines starting with a pound sign (`#`) are treated as comments and are ignored.
- Sections must be specified **in order**. Items within the sections do not have to be in a specific order unless otherwise specified. The correct section order is:
 1. The **command** section which contains actual Kickstart commands and options as listed in [Section 5.4, “Kickstart Syntax Reference”](#). Note that some commands, such as **install**, are mandatory, but most commands are optional.
 2. The **%packages** section which contains a list of packages and package groups to be installed.
 3. The **%pre** and **%post** sections, containing a pre-installation and post-installation scripts. These two sections can be in any order and are not mandatory.



IMPORTANT

The **%packages**, **%pre** and **%post** sections must end with an **%end** command, otherwise the installation program will refuse the Kickstart file. The main command section has no special ending statement.

- Omitting any required item results in the installation program prompting the user for an answer to

the related item, just as the user would be prompted during a typical installation. Once the answer is given, the installation will continue. Note that if the system you are installing has no display, you will not be able to see the prompt, and the installation will appear to have failed.

5.2. MAKING A KICKSTART FILE AVAILABLE

A kickstart file you have created based on the instructions in [Section 5.1, “Creating a Kickstart File”](#) must be loaded from the installation system. This means the file must be available, either on local storage or on a network location (HTTP, FTP, or NFS server).

To load the Kickstart file, you must use the `inst.ks=` boot option and provide the location of the file to the installer. This option is used either in the boot menu if booting from local media, or in the boot loader configuration file if booting from the network. See [Section 3.2, “Anaconda Boot Options Reference”](#) for information about using boot options.

5.3. PERFORMING A KICKSTART INSTALLATION

This section explains how to perform an installation using a Kickstart file from start to finish.

Procedure

1. Create the Kickstart file.
2. Make the file available in a way that will allow the system being installed to access it. You can accomplish this by either making it available on local storage, or on a HTTP(S), FTP, or NFS server.
3. Use the `inst.ks=` boot option either in the boot menu or in the boot loader configuration file to load the Kickstart file and use it during the installation.
4. Let the installation finish. This will happen automatically if the Kickstart file contains all mandatory commands and sections. If one or more of these mandatory parts are missing, or if an error happens, the installation will require manual intervention to finish.

Additional Resources

- [Section 5.1, “Creating a Kickstart File”](#) explains how to create a valid Kickstart file.
- [Section 5.4, “Kickstart Syntax Reference”](#) provides a list of all Kickstart commands and options.
- [Section 3.2, “Anaconda Boot Options Reference”](#) lists all boot options, including the `inst.ks=` option discussed in this section.

5.4. KICKSTART SYNTAX REFERENCE

The Kickstart syntax reference is not available in the Beta release of this document. Use the [upstream version](#) instead.

5.5. INSTALLING PACKAGE MODULES IN KICKSTART SCRIPTS

Modules are a package organization mechanism which enables the user to choose from multiple versions of package sets. Modules combine features of groups and repositories.

The Anaconda installer can enable module streams and install module profiles.

Installing Module Profiles

Install module profiles to enable the module and stream combination and install multiple packages at once. Use the `@module:stream/profile` syntax in place of a package in the **%packages** section.

- When a module has a default stream specified, you can leave it out. When the default stream is not specified, you must specify it.
- When a module stream has a default profile specified, you can leave it out. When the default profile is not specified, you must specify it.
- Installing a module multiple times with different streams is not possible.
- Installing multiple profiles of the same module and stream is possible.

When a module and a package group exist with the same name, the module takes precedence.

The following values are possible in the **%packages** section after introduction of modules:

```
%packages

^an_environment
a_group
module_with_default_stream
module_without_default_stream:stream_name
some_module:some_stream_name/profile_1
some_module:some_stream_name/profile_2
a_package

%end
```

In Red Hat Enterprise Linux 8, modules are present only in the Application Stream repository. To list available modules, use the **yum module list** command on an installed Red Hat Enterprise Linux 8 system.

Enabling Modules and Streams

Alternatively, enable modules and streams with a command, and later install packages provided by these combinations.

To enable a package module stream within kickstart script, use the **module** command:

```
module --name=NAME [--stream=STREAM]
```

- **--name=** - Specifies a name of the module to enable. Replace *NAME* with the actual name.
- **--stream=** - Specifies a name of the module stream to enable. Replace *STREAM* with the actual name.
You do not need to specify this option for modules with a default stream defined. For modules without a default stream, this option is mandatory and leaving it out results in an error. Enabling a module multiple times with different streams is not possible.

This allows you to install packages provided by the enabled module and stream combination, without specifying the module and stream explicitly. Modules must be enabled before package installation. After enabling a module with the **module** command, you can install the packages enabled by this module by listing them in the **%packages** section.

A single **module** command can enable only a single module and stream combination. To enable multiple modules, use multiple **module** commands. Enabling a module multiple times with different streams is not possible.

Additional Resources

- [Using Application Stream](#)

CHAPTER 6. BUILDING CUSTOM SYSTEM IMAGES WITH COMPOSER

Composer is a tool for creating custom system images. The following sections describe how to install it and how to perform basic usage.

6.1. INTRODUCTION TO COMPOSER

Composer is a tool that enables users to create customized system images of Red Hat Enterprise Linux. This includes ability to create system images prepared for deployment on cloud platforms. Composer functionality can be accessed through a graphical user interface in Cockpit, or with a command line interface in the **composer-cli** tool.

On Red Hat Enterprise Linux 8, Composer is available in Application Stream as a Technology Preview in the **lorax-composer** package.



IMPORTANT

Composer is available as a Technology Preview. See the [Technology Preview Features Support Scope](#) for more details.

Customers deploying Composer are encouraged to provide feedback to Red Hat.

Composer Output Formats

Composer can create these output formats:

Description	CLI name	file extension
QEMU QCOW2 Image	qcow2	.qcow2
Ext4 File System Image	ext4-filesystem	.img
Raw Partitioned Disk Image	partitioned-disk	.img
Live Bootable ISO	live-iso	.iso
TAR Archive	tar	.tar
Amazon Machine Image Disk	ami	.ami
Azure Disk Image	vhd	.vhd
VMware Virtual Machine Disk	vmdk	.vmdk

Composer User Interfaces

The Composer back end runs as a system service **lorax-composer**. Users can interact with this service through two front ends:

- GUI available as a Cockpit plugin. This is the preferred method.

- CLI available as the **composer-cli** tool for running commands.

Composer Blueprints

In Composer, a *blueprint* defines a customized system image by listing packages that will be part of the system. Blueprints can be edited and they are versioned.

When a system image is created from a blueprint, the image is associated with the blueprint in Composer Cockpit interface.

6.2. COMPOSER SYSTEM REQUIREMENTS

Use a virtual machine to run Composer, because the underlying **lorax** tool performs a number of potentially insecure and unsafe actions while creating the system images. The environment where Composer runs should meet these requirements:

Parameter	Minimal Required Value
System type	A dedicated virtual machine
Processor	2 cores
Memory	4 GiB
Disk space	20 GiB
Access privileges	Administrator level (root)
SELinux	Off (permissive mode)
Network	Connectivity to Internet and to a system with repository mirrors

Apart from the virtual machine that runs Composer itself, another system is needed to provide mirrors of Red Hat content delivery network (CDN) package repositories.



IMPORTANT

Composer is available as a Technology Preview. See the [Technology Preview Features Support Scope](#) for more details.

Customers deploying Composer are encouraged to provide feedback to Red Hat.

6.3. PREPARING A REPOSITORY MIRROR FOR COMPOSER

For technical reasons, Composer cannot directly use the Red Hat content delivery network (CDN). This procedure describes how to prepare a mirror of such content on a system different from the one running Composer.

Prerequisites

- The file system containing the **/var/www** directory must have at least 50 GiB of free space available. To check this:

```
$ df -h /var/www/
```

- The system must use the same version of Red Hat Enterprise Linux as the system using Composer and be fully subscribed.

Procedure

1. Install the tools for handling packages and repositories, and the Apache web server:

```
# yum install yum-utils createrepo httpd
```

2. List the repositories enabled on this machine and note their identifiers:

```
$ sudo yum repolist
```

3. Create local mirrors of the repositories that you want to use in Composer. For each of these repositories, run:

```
# mkdir -p /var/www/html
# reposync --download-path=/var/www/html --repoid REPO-ID --
downloadcomps --download-metadata
$ cd /var/www/html/REPO-ID
$ createrepo -v /var/www/html/REPO-ID -g comps.xml
```

Replace *REPO-ID* with the identifier you noted in the previous step.

4. Make sure that the repositories have the correct SELinux context so that the **httpd** Apache web server can access the repository mirrors:

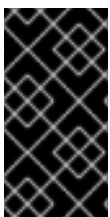
```
# chcon -vR -t httpd_sys_content_t /var/www/html/
```

5. Enable the web server to start after each reboot, configure the system firewall, and start the server for the first time:

```
# systemctl enable httpd
# firewall-cmd --add-service=http --permanent
# firewall-cmd --add-service=http
# systemctl start httpd
```

6.4. INSTALLING COMPOSER

To install Composer on a dedicated virtual machine, follow these steps.



IMPORTANT

Composer is available as a Technology Preview. See the [Technology Preview Features Support Scope](#) for more details.

Customers deploying Composer are encouraged to provide feedback to Red Hat.

Prerequisites

- The virtual machine for Composer must be already installed, meet the [requirements](#), be

subscribed, and be running.

- The [system with repository mirrors](#) must be accessible on network.
- The Composer system must use the same version of Red Hat Enterprise Linux as the system containing repository mirrors.
- You must be connected to the virtual machine and run all the commands there.

Procedure

1. Install the Composer and other necessary packages:

```
# yum install lorax-composer composer-cli cockpit-composer
```



NOTE

If Cockpit is not installed yet, it is implicitly installed as a dependency of the *cockpit-composer* package.

2. List the repositories enabled on this machine and note their identifiers:

```
$ sudo yum repolist
```

3. Create a repository configuration file in the `/etc/yum.repos.d` directory which points to the mirrored repositories. Include the IP address or host name of the virtual machine system. For each of the repository mirrors, run:

```
# cat >> /etc/yum.repos.d/mirror.repo <<EOF
[mirror-REPO-ID]
name=NAME
baseurl=http://IP-ADDR/cdrom/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

EOF
```

Replace *REPO-ID* with the repository identifier, *NAME* with the repository name, and *IP-ADDR* with the IP address or host name. This will present the repository mirrors to the system with an identifier containing the prefix *mirror-*.

4. Verify the repository configuration:

```
# yum clean all
# yum repolist
```

5. Disable the original subscribed repositories from Red Hat content delivery network. For each of the mirrored repositories, run:

```
# yum-config-manager --disable REPO-ID
```

Replace *REPO-ID* with the repository identifier.

6. Enable Composer to start after each reboot and configure the system firewall:

```
# systemctl enable lorax-composer.socket
# systemctl enable cockpit.socket
# firewall-cmd --add-service=cockpit && firewall-cmd --add-
service=cockpit --permanent
```

7. The Composer and Cockpit services are started automatically on each system reboot. For this first session after installation, start these services manually:

```
# systemctl start lorax-composer
# systemctl start cockpit
```

6.5. ACCESSING COMPOSER GUI IN COCKPIT

Using Cockpit is the recommended way of accessing Composer functionality.



IMPORTANT

Composer is available as a Technology Preview. See the [Technology Preview Features Support Scope](#) for more details.

Customers deploying Composer are encouraged to provide feedback to Red Hat.

Prerequisites

- You must have root access to the system.

Procedure

1. Open <https://localhost:9090/> in a web browser on the system with Composer.



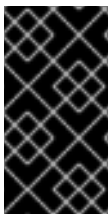
NOTE

Accessing Composer from another system falls under the topic of remote access to Cockpit.

2. Log into Cockpit with credentials for an user account with sufficient privileges on the system.
3. On the left, click the **Image Builder** icon to display the Composer controls.
The Composer view opens, listing existing blueprints.

6.6. CREATING A COMPOSER BLUEPRINT

These are the steps for creating a blueprint.



IMPORTANT

Composer is available as a Technology Preview. See the [Technology Preview Features Support Scope](#) for more details.

Customers deploying Composer are encouraged to provide feedback to Red Hat.

Prerequisites

- You have opened the Composer Cockpit interface in a browser.

Procedure

1. Click **Create Blueprint** in the top right corner.
A pop-up appears with fields for the blueprint name and description.
2. Fill in the name of the blueprint, its description, then click **Create**.
The screen changes to blueprint editing mode.
3. Add components that you want to include in the system image:
 1. On the left, enter component name or a part of it into the field under the heading **Available Components** and press **Enter**.
The search is added to the list of filters under the text entry field, and the list of components below is reduced to these that match the search.

If the list of components is too long, add further search terms in the same way.
 2. The list of components is paged. To move to other result pages, use the arrows and entry field above the component list.
 3. Click on name of the component you intend to use to display its details. The right pane fills with details of the components, such as its version and dependencies.
 4. Select the version you want to use in the **Component Options** box, with the **Version Release** dropdown.
 5. Click **Add** in the top left.
 6. If you added a component by mistake, remove it by clicking the ... button at the far right of its entry in the right pane, and select **Remove** in the menu.



NOTE

If you do not intend to select version for some components, you can skip the component details screen and version selection by clicking the **+** buttons on the right side of the component list.

4. To save the blueprint, click **Commit** in the top left. A dialog with a summary of the changes pops up. Click **Commit**.
A small pop-up on the right informs you of the saving progress and then result.
5. In the top left, click **Back to Blueprints** to exit the editing screen.
The Composer view opens, listing existing blueprints.

6.7. EDITING A COMPOSER BLUEPRINT

These are the steps for editing a blueprint.



IMPORTANT

Composer is available as a Technology Preview. See the [Technology Preview Features Support Scope](#) for more details.

Customers deploying Composer are encouraged to provide feedback to Red Hat.

Prerequisites

- You have opened the Composer Cockpit interface in a browser.
- A blueprint exists.


Procedure

1. Locate the blueprint that you want to edit by entering its name or a part of it into the search box at top left, and press **Enter**.
The search is added to the list of filters under the text entry field, and the list of blueprints below is reduced to these that match the search.

If the list of blueprints is too long, add further search terms in the same way.

2. On the left side of the blueprint, press the **Edit Blueprint** button that belongs to the blueprint.

The view changes to the blueprint editing screen.


3. Remove unwanted components by clicking their  button at the far right of its entry in the right pane, and select **Remove** in the menu.

4. Change version of existing components:

- a. On the left, enter component name or a part of it into the field under the heading **Blueprint Components** and press **Enter**.

The search is added to the list of filters under the text entry field, and the list of components below is reduced to these that match the search.

If the list of components is too long, add further search terms in the same way.

- b. Click the  button at the far right of the component entry, and select **Edit** in the menu.

A component details screen opens in the right pane.

- c. Select the desired version in the **Version Release** drop-down menu and click **Apply Change** in top left.

The change is saved and the right pane returns to listing the blueprint components.


5. Add new components:

1. On the left, enter component name or a part of it into the field under the heading **Available Components** and press **Enter**.

The search is added to the list of filters under the text entry field, and the list of components below is reduced to these that match the search.

If the list of components is too long, add further search terms in the same way.

2. The list of components is paged. To move to other result pages, use the arrows and entry field above the component list.

3. Click on name of the component you intend to use to display its details. The right pane fills with details of the components, such as its version and dependencies.
4. Select the version you want to use in the **Component Options** box, with the **Version Release** drop-down menu.
5. Click **Add** in the top left.
6. If you added a component by mistake, remove it by clicking the  button at the far right of its entry in the right pane, and select **Remove** in the menu.



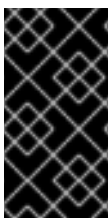
NOTE

If you do not intend to select version for some components, you can skip the component details screen and version selection by clicking the **+** buttons on the right side of the component list.

6. Commit a new version of the blueprint with your changes:
 - a. Click the **Commit** button in top left.
A pop-up window with a summary of your changes appears.
 - b. Review your changes and confirm them by clicking **Commit**.
A small pop-up on the right informs you of the saving progress and then result. A new version of the blueprint is created.
7. In the top left, click **Back to Blueprints** to exit the editing screen.
The Composer view opens, listing existing blueprints.

6.8. CREATING A SYSTEM IMAGE WITH COMPOSER

The following steps below describe creating a system image.



IMPORTANT

Composer is available as a Technology Preview. See the [Technology Preview Features Support Scope](#) for more details.

Customers deploying Composer are encouraged to provide feedback to Red Hat.

Prerequisites

- You have opened the Composer Cockpit interface in a browser.
- A blueprint exists.

Procedure

1. Locate the blueprint that you want to edit by entering its name or a part of it into the search box at top left, and press **Enter**.
The search is added to the list of filters under the text entry field, and the list of blueprints below is reduced to these that match the search.

If the list of blueprints is too long, add further search terms in the same way.

2. On the left side of the blueprint, press the **Create Image** button that belongs to the blueprint.
A pop-up window appears.
3. Select the image type and architecture and press **Create**.
A small pop-up in the top left informs you that the image creation has been added to the queue.
4. Click the name of the blueprint.
A screen with details of the blueprint opens.
5. Click the **Images** tab to switch to it.
6. The image that is being created is listed with the status **Pending**.

**NOTE**

Image creation takes a longer time, measured in minutes. There is no indication of progress while the image is created.

To abort image creation, press its **Stop** button on the right.

7. Once the image is successfully created, the **Stop** button is replaced by a **Download** button.
Click this button to download the image to your system.

6.9. ADDITIONAL RESOURCES

- Upstream Weldr documentation: <https://weldr.io/>

CHAPTER 7. CREATING CLOUD IMAGES WITH COMPOSER

Composer can create custom system images ready for use in clouds of various providers. This chapter describes setting up the Composer virtual machine for this task, and uploading the resulting images to the various types of clouds.

7.1. PREPARING COMPOSER FOR CREATING AWS AMI IMAGES

This describes steps to configure system with Composer for making AWS AMI images.

Prerequisites

- Composer must be installed on a dedicated virtual machine system.
- You must have an Access Key ID configured in the [AWS IAM account manager](#).
- You must have a writable [S3 bucket](#) prepared.

Procedure

1. Install Python 3 and the **pip** tool:

```
# yum install @python36
# yum install python3-pip
```

2. Install the [AWS command line tools](#) with **pip**:

```
# pip3 install awscli
```

3. Configure the AWS command line client according to your AWS access details:

```
$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]:
Default output format [None]:
```

4. Configure the AWS command line client to use your bucket:

```
$ BUCKET=bucketname
$ aws s3 mb s3://$BUCKET
```

Replace *bucketname* with the actual bucket name.

5. Create a **vmimport** S3 Role in IAM and grant it permissions to access S3, if you have not already done so in the past:

```
$ printf '{ "Version": "2012-10-17", "Statement": [ { "Effect":
"Allow", "Principal": { "Service": "vmie.amazonaws.com" }, "Action":
"sts:AssumeRole", "Condition": { "StringEquals":{ "sts:Externalid":
"vmimport" } } } ] }' > trust-policy.json
$ printf '{ "Version": "2012-10-17", "Statement": [ {
"Effect": "Allow", "Action": [ "s3:GetBucketLocation", "s3:GetObject",
```

```
"s3:ListBucket" ], "Resource":[ "arn:aws:s3:::%s",
"arn:aws:s3:::%s/*" ] }, { "Effect":"Allow", "Action":[
"ec2:ModifySnapshotAttribute", "ec2:CopySnapshot",
"ec2:RegisterImage", "ec2:Describe*" ], "Resource":"*" } ] }'
$BUCKET $BUCKET > role-policy.json
$ aws iam create-role --role-name vmimport --assume-role-policy-
document file://trust-policy.json
$ aws iam put-role-policy --role-name vmimport --policy-name
vmimport --policy-document file://role-policy.json
```

Additional Resources

- [Weldr upstream documentation on AWS images](#)

7.2. PREPARING COMPOSER FOR CREATING AZURE VHD IMAGES

This describes steps to configure system with Composer for making Azure images.

Prerequisites

- Composer must be installed on a dedicated virtual machine system.
- You must have a usable Azure resource group and storage account.

Procedure

1. Install the [Azure CLI](#) tooling:

```
# rpm --import https://packages.microsoft.com/keys/microsoft.asc
# sh -c 'echo -e "[azure-cli]\nname=Azure
CLI\nbaseurl=https://packages.microsoft.com/yumrepos/azure-
cli\nenabled=1\npgpcheck=1\npgpkey=https://packages.microsoft.com/ke
ys/microsoft.asc" > /etc/yum.repos.d/azure-cli.repo'
# yum install azure-cli
```

2. Log into the Azure CLI:

```
$ az login
To sign in, use a web browser to open the page
https://microsoft.com/devicelogin and enter the code XXXXXXXXX to
authenticate.
...
```

3. List the keys for the storage account in Azure:

```
$ GROUP=resource-group-name
$ ACCOUNT=storage-account-name
$ az storage account keys list --resource-group $GROUP --account-
name $ACCOUNT
```

Replace *resource-group-name* with name of the Azure resource group and *storage-account-name* with name of the Azure storage account.

4. Make note of value **key1** in the output of the previous command, and assign it to an environment variable:

```
$ KEY1=value
```

5. Create a storage container:

```
$ CONTAINER=storage-account-name
$ az storage container create --account-name $ACCOUNT \
  --account-key $KEY1 --name $CONTAINER
```

Replace *storage-account-name* with name of the storage account.

Additional Resources

- [Weldr upstream documentation on Azure images](#)

7.3. CREATING CLOUD IMAGES USING COMPOSER

To create a cloud image in the Composer Cockpit interface:

1. Prepare your system for creating or uploading images of the particular type.
2. Follow the steps in [Chapter 6, Building Custom System Images with Composer](#) while creating the image.
3. Select a suitable output cloud image format.

7.4. USING AN AMI IMAGE ON AWS

This describes steps to upload an AMI image to AWS.

Prerequisites

- Your system must be set up for creating AWS images.
- You must have an AWS image created by Composer.

Procedure

1. Push the image to S3 and start an EC2 instance:

```
$ AMI=8db1b463-91ee-4fd9-8065-938924398428-disk.ami
$ aws s3 cp $AMI s3://$BUCKET
Completed 24.2 MiB/4.4 GiB (2.5 MiB/s) with 1 file(s) remaining
...
```

2. After the upload to S3 completes, import it as a snapshot into EC2:

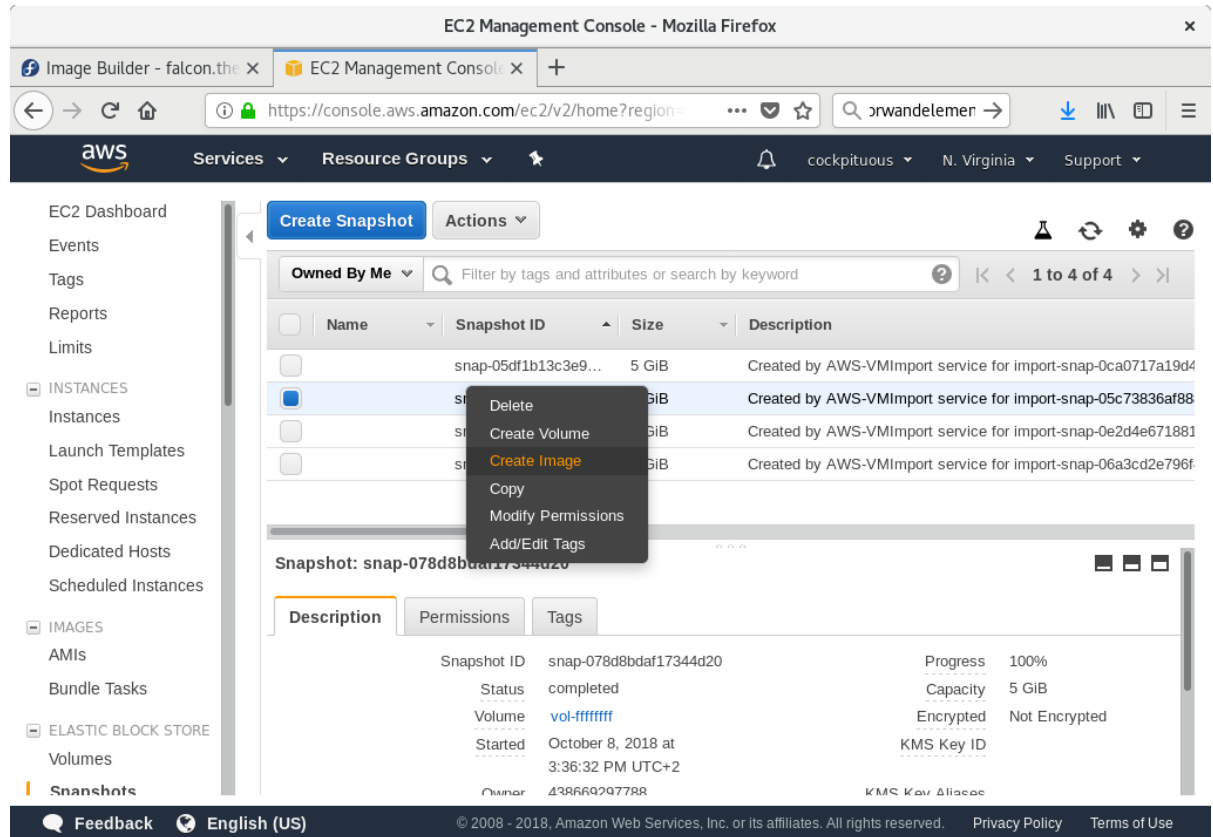
```
$ printf '{ "Description": "my-image", "Format": "raw",
  "UserBucket": { "S3Bucket": "%s", "S3Key": "%s" } }' $BUCKET $AMI >
containers.json
$ aws ec2 import-snapshot --disk-container file://containers.json
```

Replace *my-image* with the name you want for the image.

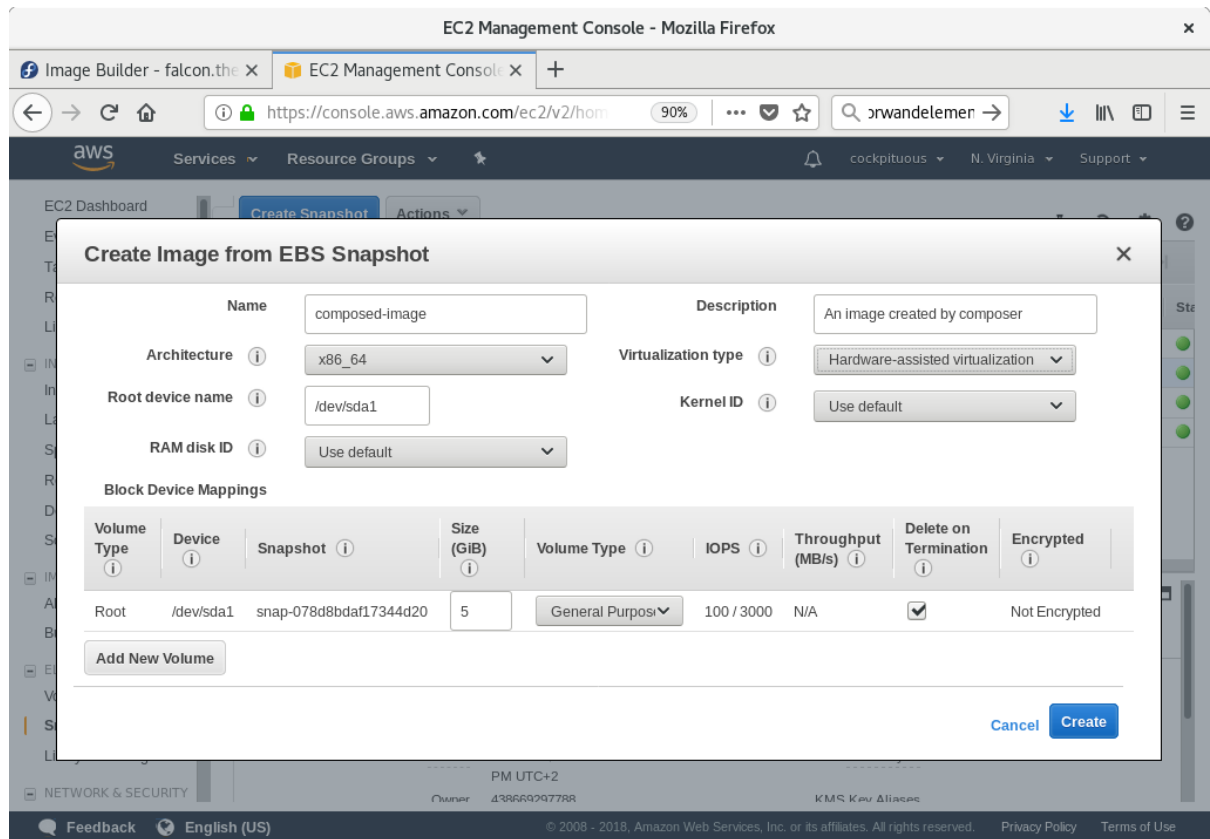
To track progress of the import, run:

```
$ aws ec2 describe-import-snapshot-tasks --filters Name=task-state,Values=active
```

3. Create an image from the uploaded snapshot by selecting the snapshot in the EC2 console, right clicking on it and selecting **Create Image**:



4. Select the **Virtualization** type of **Hardware-assisted virtualization** in the image you create:



- Now you can run an instance using whatever mechanism you like (CLI or AWS Console) from the snapshot. Use your private key via SSH to access the resulting EC2 instance. Log in as **ec2-user**.

Additional Resources

- [Weldr upstream documentation on AWS images](#)

7.5. USING AN VHD IMAGE ON AZURE

This describes steps to upload an VHD image to Azure.

Prerequisites

- Your system must be set up for creating Azure VHD images.
- You must have an Azure VHD image created by Composer.

Procedure

- Push the image to Azure and create an instance from it:

```
$ VHD=25ccb8dd-3872-477f-9e3d-c2970cd4bbaf-disk.vhd
$ az storage blob upload --account-name $ACCOUNT --container-name
$CONTAINER --file $VHD --name $VHD --type page
...
```

- Once the upload to the Azure BLOB completes, create an Azure image from it:

```
$ az image create --resource-group $GROUP --name $VHD --os-type
```

```
linux --location eastus --source
https://$ACCOUNT.blob.core.windows.net/$CONTAINER/$VHD
- Running ...
```

3. Create an instance either with the Azure portal, or a command similar to the following:

```
$ az vm create --resource-group $GROUP --location eastus --name $VHD
--image $VHD --admin-username azure-user --generate-ssh-keys
- Running ...
```

4. Use your private key via SSH to access the resulting instance. Log in as **azure-user**.

Additional Resources

- [Weldr upstream documentation on Azure images](#)

7.6. USING AN VMDK IMAGE ON VSPHERE

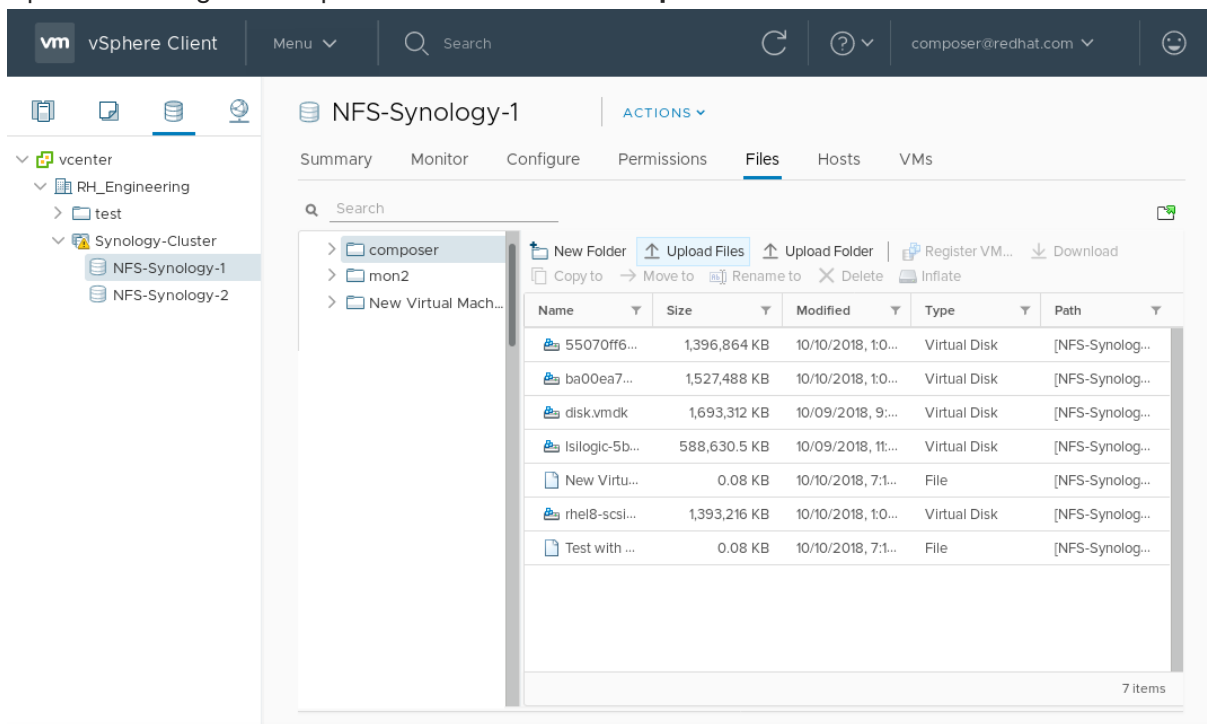
Composer can generate images suitable for uploading to a VMware ESXi or vSphere system. This describes steps to upload an VMDK image to VMware vSphere.

Prerequisites

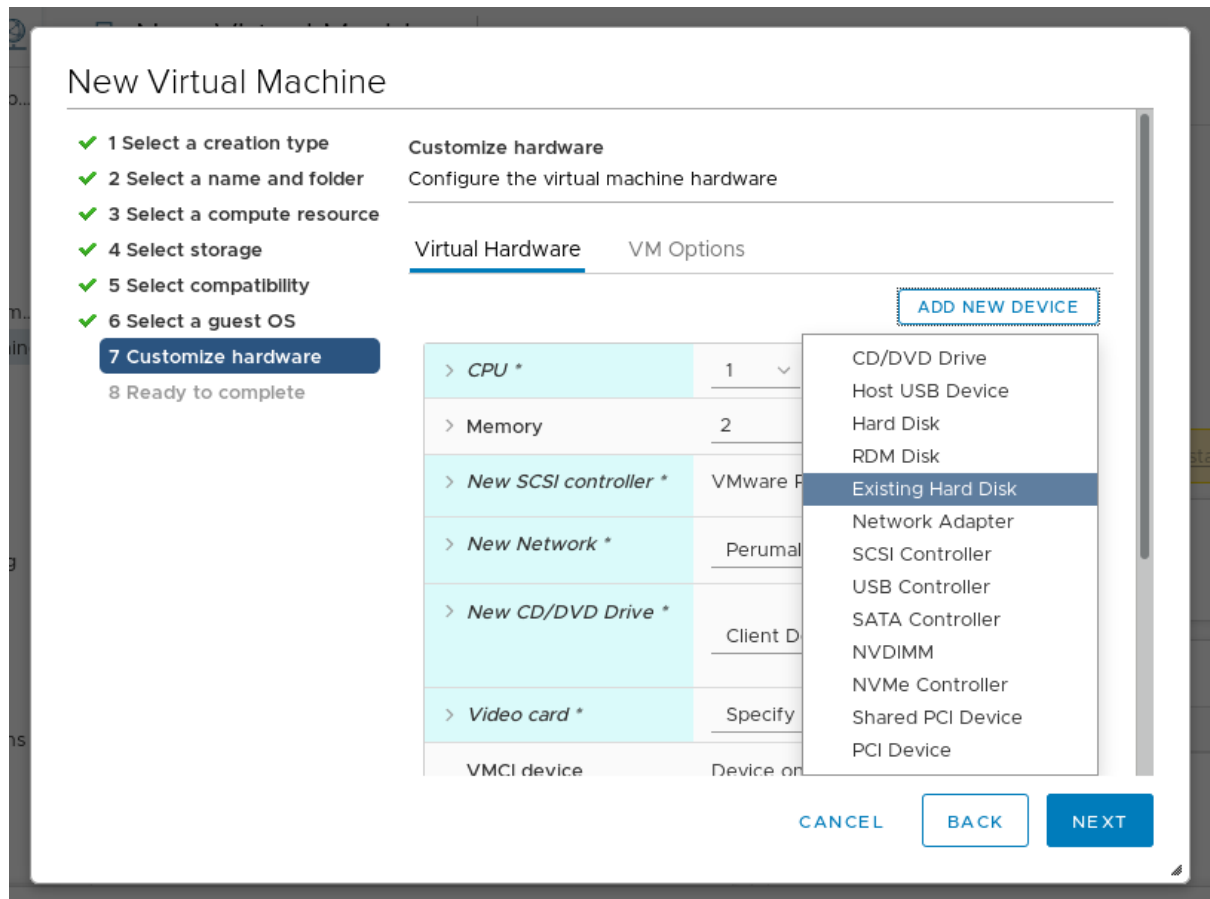
- You must have an VMDK image created by Composer.

Procedure

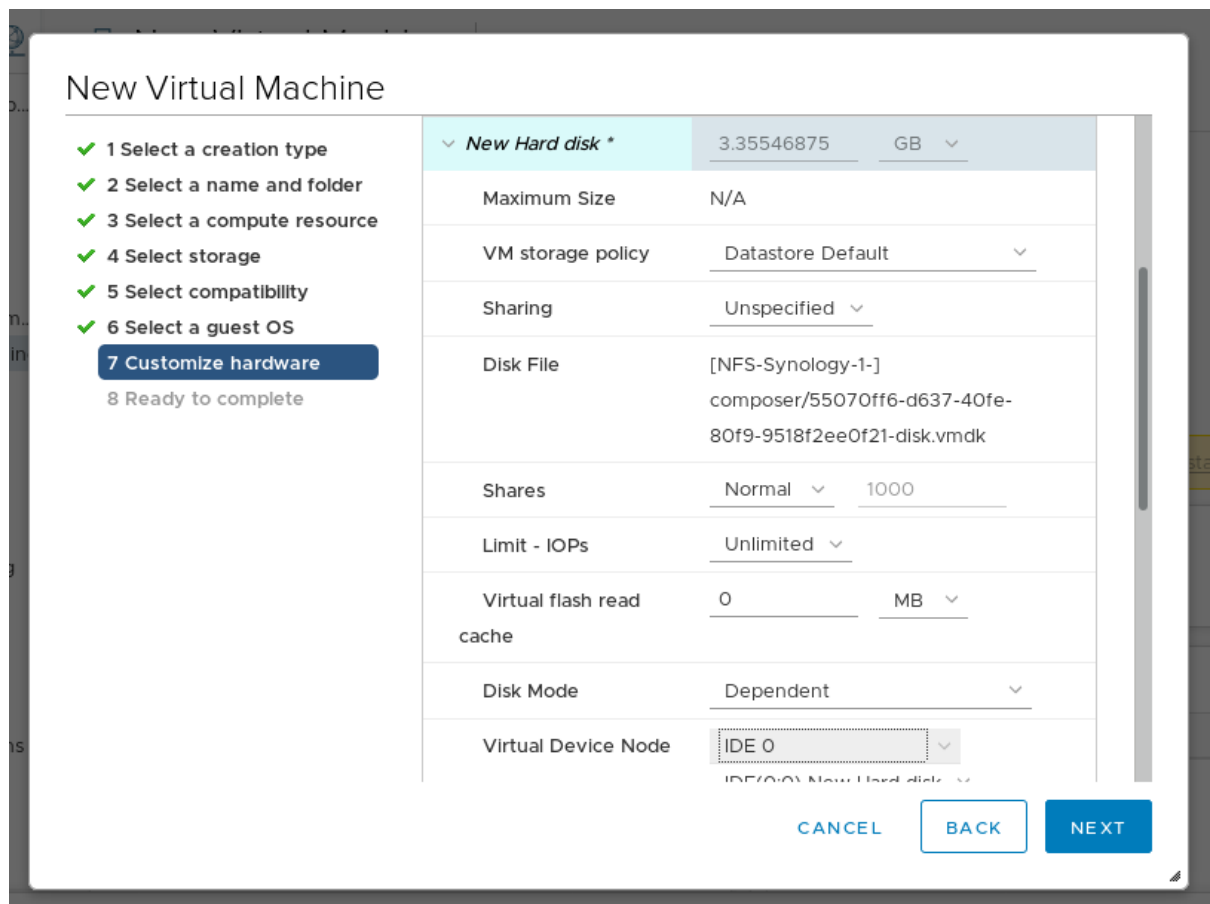
1. Upload the image into vSphere via HTTP. Click on **Upload Files** in the vCenter:



2. When you create a VM, on the **Device Configuration**, delete the default **New Hard Disk** and use the drop-down to select an **Existing Hard Disk** disk image:



3. Make sure you use an **IDE** device as the **Virtual Device Node** for the disk you create. The default value **SCSI** results in an unbootable virtual machine.



- [Weldr upstream documentation on vSphere and VMware images](#)

7.7. USING AN QCOW2 IMAGE ON OPENSTACK

Composer can generate images suitable for uploading to OpenStack cloud deployments, and starting instances there. This describes steps to upload an QCOW2 image to OpenStack.

Prerequisites

- You must have an OpenStack-specific image created by Composer.
In Composer, ensure during the selection process for the output format that "OpenStack Image" has been selected. This is due to several changes in the output format that is specific to OpenStack.

Procedure

1. Upload the image to OpenStack and start an instance from it. Use the **Images** interface to do this:

Create An Image

Name: *

96268ffb-2c71-4e97-a855-7ac25e983a6e-disk.qcow2

Description:

Image Source:

Image File

Image File

Browse...

96268ffb-2c71-4e97-a85...c25e98

Format: *

QCOW2 - QEMU Emulator

Architecture:

x86_64

Minimum Disk (GB):

5

Minimum Ram (MB):

1024

Public:

☒

Protected:

☐

Cancel

Create Image

Description:

Specify an image to upload to the Image Service.

Currently only images available via an HTTP URL are supported. The image location must be accessible to the Image Service. Compressed image binaries are supported (.zip and .tar.gz.)

Please note: The Image Location field MUST be a valid and direct URL to the image binary. URLs that redirect or serve error pages will result in unusable images.

2. Start an instance with that image:

Launch Instance

Details *

Access & Security *

Networking *

Post-Creation

Advanced Options

Availability Zone:

nova

Instance Name: *

my-instance

Flavor: *

m1.small

Some flavors not meeting minimum image requirements have been disabled.

Instance Count: *

1

Instance Boot Source: *

Boot from image

Image Name:

96268ffb-2c71-4e97-a855-7ac25e983a6e-disk.qc

Specify the details for launching an instance.

The chart below shows the resources used by this project in relation to the project's quotas.

Flavor Details

Name	m1.small
VCPUs	1
Root Disk	20 GB
Ephemeral Disk	0 GB
Total Disk	20 GB
RAM	2,048 MB

Project Limits

Number of Instances

4 of 10 Used

Number of VCPUs

17 of 20 Used

Total RAM

34,816 of 51,200 MB Used

Cancel

Launch

- You can run the instance using any mechanism (CLI or AWS Console) from the snapshot. Use your private key via SSH to access the resulting EC2 instance. Log in as **ccloud-user**.

Additional Resources

- [Weldr upstream documentation on OpenStack images](#)

CHAPTER 8. PARTITIONING REFERENCE

8.1. SUPPORTED DEVICE TYPES

This paragraph is the reference module introduction and is only optional. Include it to provide a short overview of the module.

Standard partition

A standard partition can contain a file system or swap space. Standard partitions are most commonly used for **/boot** and the **BIOS Boot** and **EFI System partitions**. LVM logical volumes are recommended for most other uses.

LVM

Choosing **LVM** (or Logical Volume Management) as the Device Type creates an LVM logical volume. If no LVM volume group currently exists, one is automatically created to contain the new volume, if one already exists, the volume is assigned to it. LVM can improve performance when using physical disks and allows for advanced setups such as using multiple physical disks for one mount point, and setting up software RAID for increased performance, reliability, or both.

LVM Thin Provisioning

Using thin provisioning, you can manage a storage pool of free space, known as a thin pool, which can be allocated to an arbitrary number of devices when needed by applications. You can dynamically expand the pool when needed for cost-effective allocation of storage space.

8.2. SUPPORTED FILE SYSTEMS

This section describes the file systems available in the installer.

xfs

XFS is a highly scalable, high-performance file system that supports file systems up to 16 exabytes (approximately 16 million terabytes), files up to 8 exabytes (approximately 8 million terabytes), and directory structures containing tens of millions of entries. XFS also supports metadata journaling, which facilitates quicker crash recovery. The maximum supported size of a single XFS file system is 500 TB. XFS is the default and recommended file system on Red Hat Enterprise Linux.

ext4

The ext4 file system is based on the ext3 file system and features a number of improvements. These include support for larger file systems and larger files, faster and more efficient allocation of disk space, no limit on the number of subdirectories within a directory, faster file system checking, and more robust journaling. The maximum supported size of a single ext4 file system is 50 TB.

ext3

The ext3 file system is based on the ext2 file system and has one main advantage - journaling. Using a journaling file system reduces time spent recovering a file system after a crash, as there is no need to check the file system for metadata consistency by running the fsck utility every time a crash occurs.

ext2

An ext2 file system supports standard Unix file types, including regular files, directories, or symbolic links. It provides the ability to assign long file names, up to 255 characters.

swap

Swap partitions are used to support virtual memory. In other words, data is written to a swap partition when there is not enough RAM to store the data your system is processing.

vfat

The VFAT file system is a Linux file system that is compatible with Microsoft Windows long file names on the FAT file system.

BIOS Boot

A very small partition required for booting from a device with a GUID partition table (GPT) on BIOS systems and UEFI systems in BIOS compatibility mode.

EFI System Partition

A small partition required for booting a device with a GUID partition table (GPT) on a UEFI system.

8.3. SUPPORTED RAID TYPES

RAID stands for Redundant Array of Independent Disks, a technology which allows you to combine multiple physical disks into logical units. Some setups are designed to enhance performance at the cost of reliability, while others will improve reliability at the cost of requiring more disks for the same amount of available space.

This section describes supported software RAID types which you can use with LVM and LVM Thin Provisioning to set up the installed system's storage.

None

No RAID array will be set up.

RAID0

Performance - Distributes data across multiple disks. Level 0 RAID offers increased performance over standard partitions and can be used to pool the storage of multiple disks into one large virtual device. Note that Level 0 RAID offers no redundancy and that the failure of one device in the array destroys data in the entire array. RAID 0 requires at least two disks.

RAID1

Redundancy - Mirrors all data from one partition onto one or more other disks. Additional devices in the array provide increasing levels of redundancy. RAID 1 requires at least two disks.

RAID4

Error checking - Distributes data across multiple disks and uses one disk in the array to store parity information which safeguards the array in case any disk within the array fails. Because all parity information is stored on one disk, access to this disk creates a "bottleneck" in the array's performance. Level 4 RAID requires at least three disks.

RAID5

Distributed error checking - Distributes data and parity information across multiple disks. Level 5 RAID therefore offers the performance advantages of distributing data across multiple disks, but does not share the performance bottleneck of level 4 RAID because the parity information is also distributed through the array. RAID 5 requires at least three disks.

RAID6

Redundant error checking - Level 6 RAID is similar to level 5 RAID, but instead of storing only one set of parity data, they store two sets. RAID 6 requires at least four disks.

RAID10

Performance and redundancy - Level 10 RAID is a nested RAID or hybrid RAID. They are constructed by distributing data over mirrored sets of disks. For example, a level 10 RAID array constructed from four RAID partitions consists of two mirrored pairs of striped partitions. RAID 10 requires at least four disks.

8.4. RECOMMENDED PARTITIONING SCHEME

Red Hat recommends that you create separate file systems at the following mount points:

- **/boot**
- **/** (root)
- **/home**
- **swap**

/boot partition - recommended size at least 1 GiB

The partition mounted on **/boot** contains the operating system kernel, which allows your system to boot Red Hat Enterprise Linux 8.0 Beta, along with files used during the bootstrap process. Due to the limitations of most firmwares, creating a small partition to hold these is recommended. In most scenarios, a 1 GiB boot partition is adequate. Unlike other mount points, using an LVM volume for **/boot** is not possible - **/boot** must be located on a separate disk partition.



WARNING

Normally, the **/boot** partition is created automatically by the installation program. However, if the **/** (root) partition is larger than 2 TiB and (U)EFI is used for booting, you need to create a separate **/boot** partition that is smaller than 2 TiB to boot the machine successfully.



NOTE

If you have a RAID card, be aware that some BIOS types do not support booting from the RAID card. In such a case, the **/boot** partition must be created on a partition outside of the RAID array, such as on a separate hard drive.

root - recommended size of 10 GiB

This is where **/**, or the root directory, is located. The root directory is the top-level of the directory structure. By default, all files are written to this file system unless a different file system is mounted in the path being written to, for example, **/boot** or **/home**.

While a 5 GiB root file system allows you to install a minimal installation, it is recommended to allocate at least 10 GiB so that you can install as many package groups as you want.



IMPORTANT

Do not confuse the **/** directory with the **/root** directory. The **/root** directory is the home directory of the root user. The **/root** directory is sometimes referred to as *slash root* to distinguish it from the root directory.

/home - recommended size at least 1 GiB

To store user data separately from system data, create a dedicated file system for the **/home**

directory. Base the file system size on the amount of data that is stored locally, number of users, and so on. You can upgrade or reinstall Red Hat Enterprise Linux 8.0 Beta without erasing user data files. If you select automatic partitioning, it is recommended to have at least 55 GiB of disk space available for the installation, to ensure that the **/home** file system is created.

swap partition - recommended size at least 1 GB

Swap file systems support virtual memory; data is written to a swap file system when there is not enough RAM to store the data your system is processing. Swap size is a function of system memory workload, not total system memory and therefore is not equal to the total system memory size. It is important to analyze what applications a system will be running and the load those applications will serve in order to determine the system memory workload. Application providers and developers can provide guidance.

When the system runs out of swap space, the kernel terminates processes as the system RAM memory is exhausted. Configuring too much swap space results in storage devices being allocated but idle and is a poor use of resources. Too much swap space can also hide memory leaks. The maximum size for a swap partition and other additional information can be found in the **mkswap(8)** manual page.

The following table provides the recommended size of a swap partition depending on the amount of RAM in your system and if you want sufficient memory for your system to hibernate. If you let the installation program partition your system automatically, the swap partition size is established using these guidelines. Automatic partitioning setup assumes hibernation is not in use. The maximum size of the swap partition is limited to 10 percent of the total size of the hard drive, and the installation program cannot create swap partitions more than 128 GB in size. To set up enough swap space to allow for hibernation, or if you want to set the swap partition size to more than 10 percent of the system's storage space, or more than 128 GB, you must edit the partitioning layout manually.

Amount of RAM in the system	Recommended swap space	Recommended swap space if allowing for hibernation
Less than 2 GB	2 times the amount of RAM	3 times the amount of RAM
2 GB - 8 GB	Equal to the amount of RAM	2 times the amount of RAM
8 GB - 64 GB	4 GB to 0.5 times the amount of RAM	1.5 times the amount of RAM
More than 64 GB	Workload dependent (at least 4GB)	Hibernation not recommended

At the border between each range, for example, a system with 2 GB, 8 GB, or 64 GB of system RAM, discretion can be exercised with regard to chosen swap space and hibernation support. If your system resources allow for it, increasing the swap space can lead to better performance.

Distributing swap space over multiple storage devices - particularly on systems with fast drives, controllers and interfaces - also improves swap space performance.

Many systems have more partitions and volumes than the minimum required. Choose partitions based on your particular system needs.

**NOTE**

Only assign storage capacity to those partitions you require immediately. You can allocate free space at any time, to meet needs as they occur.

**NOTE**

If you are unsure about how to configure partitions, accept the automatic default partition layout provided by the installation program.

8.5. ADVICE ON PARTITIONS

There is no best way to partition every system; the optimal setup depends on how you plan to use the system being installed. However, the following tips may help you find the optimal layout for your needs:

- Create partitions that have specific requirements first, for example, if a particular partition must be on a specific disk.
- Consider encrypting any partitions and volumes which might contain sensitive data. Encryption prevents unauthorized people from accessing the data on the partitions, even if they have access to the physical storage device. In most cases, you should at least encrypt the **/home** partition, which contains user data.
- In some cases, creating separate mount points for directories other than **/**, **/boot** and **/home** may be useful; for example, on a server running a **MySQL** database, having a separate mount point for **/var/lib/mysql** will allow you to preserve the database during a reinstallation without having to restore it from backup afterwards. However, having unnecessary separate mount points will make storage administration more difficult.
- Some special restrictions apply to certain directories with regards on which partitioning layouts can they be placed. Notably, the **/boot** directory must always be on a physical partition (not on an LVM volume).
- If you are new to Linux, consider reviewing the *Linux Filesystem Hierarchy Standard* at http://refspecs.linuxfoundation.org/FHS_2.3/fhs-2.3.html for information about various system directories and their contents.
- Each kernel installed on your system requires approximately 20 MB on the **/boot** partition. The default partition size of 500 MB for **/boot** should suffice for most common uses; increase the size of this partition if you plan to keep many kernels installed at the same time.
- The **/var** directory holds content for a number of applications, including the **Apache** web server, and is used by the **DNF** package manager to temporarily store downloaded package updates. Make sure that the partition or volume containing **/var** has at least 3 GB.
- The contents of the **/var** directory usually change very often. This may cause problems with older solid state drives (SSDs), as they can handle a lower number of read/write cycles before becoming unusable. If your system root is on an SSD, consider creating a separate mount point for **/var** on a classic (platter) HDD.
- The **/usr** directory holds the majority of software on a typical Red Hat Enterprise Linux installation. The partition or volume containing this directory should therefore be at least 5 GB for minimal installations, and at least 10 GB for installations with a graphical environment.
- If **/usr** or **/var** is partitioned separately from the rest of the root volume, the boot process

becomes much more complex because these directories contain boot-critical components. In some situations, such as when these directories are placed on an iSCSI drive or an FCoE location, the system may either be unable to boot, or it may hang with a **Device is busy** error when powering off or rebooting.

This limitation only applies to **/usr** or **/var**, not to directories below them. For example, a separate partition for **/var/www** will work without issues.

- Consider leaving a portion of the space in an LVM volume group unallocated. This unallocated space gives you flexibility if your space requirements change but you do not wish to remove data from other volumes. You can also select the **LVM Thin Provisioning** device type for the partition to have the unused space handled automatically by the volume.
- The size of an XFS file system can not be reduced - if you need to make a partition or volume with this file system smaller, you must back up your data, destroy the file system, and create a new, smaller one in its place. Therefore, if you expect needing to manipulate your partitioning layout later, you should use the ext4 file system instead.
- Use Logical Volume Management (LVM) if you anticipate expanding your storage by adding more hard drives after the installation. With LVM, you can create physical volumes on the new drives, and then assign them to any volume group and logical volume as you see fit - for example, you can easily expand your system's **/home** (or any other directory residing on a logical volume).
- Creating a BIOS Boot partition or an EFI System Partition may be necessary, depending on your system's firmware, boot drive size, and boot drive disk label. See [Section 8.4, "Recommended Partitioning Scheme"](#) for information about these partitions. Note that the graphical installer will not let you create a BIOS Boot or EFI System Partition if your system does **not** require one - in that case, they will be hidden from the menu.
- If you need to make any changes to your storage configuration after the installation, Red Hat Enterprise Linux repositories offer several different tools which can help you do this. If you prefer a command line tool, try **system-storage-manager**.

CHAPTER 9. TROUBLESHOOTING INSTALLATION PROBLEMS

The following sections cover various troubleshooting information which may be helpful when diagnosing installation problems.

9.1. CONSOLES AND LOGGING DURING INSTALLATION

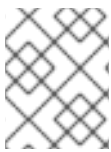
The Red Hat Enterprise Linux installer uses the **tmux** terminal multiplexer to display and control several windows you can use in addition to the main interface. Each of these windows serves a different purpose - they display several different logs, which can be used to troubleshoot any issues during the installation, and one of the windows provides an interactive shell prompt with **root** privileges, unless this prompt was specifically disabled using a boot option or a Kickstart command.



NOTE

In general, there is no reason to leave the default graphical installation environment unless you need to diagnose an installation problem.

The terminal multiplexer is running in virtual console 1. To switch from the actual installation environment to **tmux**, press **Ctrl+Alt+F1**. To go back to the main installation interface which runs in virtual console 6, press **Ctrl+Alt+F6**.



NOTE

If you choose text mode installation, you will start in virtual console 1 (**tmux**), and switching to console 6 will open a shell prompt instead of a graphical interface.

The console running **tmux** has 5 available windows; their contents are described in the table below, along with keyboard shortcuts used to access them. Note that the keyboard shortcuts are two-part: first press **Ctrl+b**, then release both keys, and press the number key for the window you want to use.

You can also use **Ctrl+b n** and **Ctrl+b p** to switch to the next or previous **tmux** window, respectively.

Table 9.1. Available tmux Windows

Shortcut	Contents
Ctrl+b 1	Main installation program window. Contains text-based prompts (during text mode installation or if you use VNC direct mode), and also some debugging information.
Ctrl+b 2	Interactive shell prompt with root privileges.
Ctrl+b 3	Installation log; displays messages stored in /tmp/anaconda.log .
Ctrl+b 4	Storage log; displays messages related storage devices from kernel and system services, stored in /tmp/storage.log .

Shortcut	Contents
ctrl+b 5	Program log; displays messages from other system utilities, stored in /tmp/program.log .

9.2. SAVING SCREENSHOTS

You can press **Shift+Print Screen** at any time during the graphical installation to capture the current screen. These screenshots are saved to **/tmp/anaconda-screenshots**.

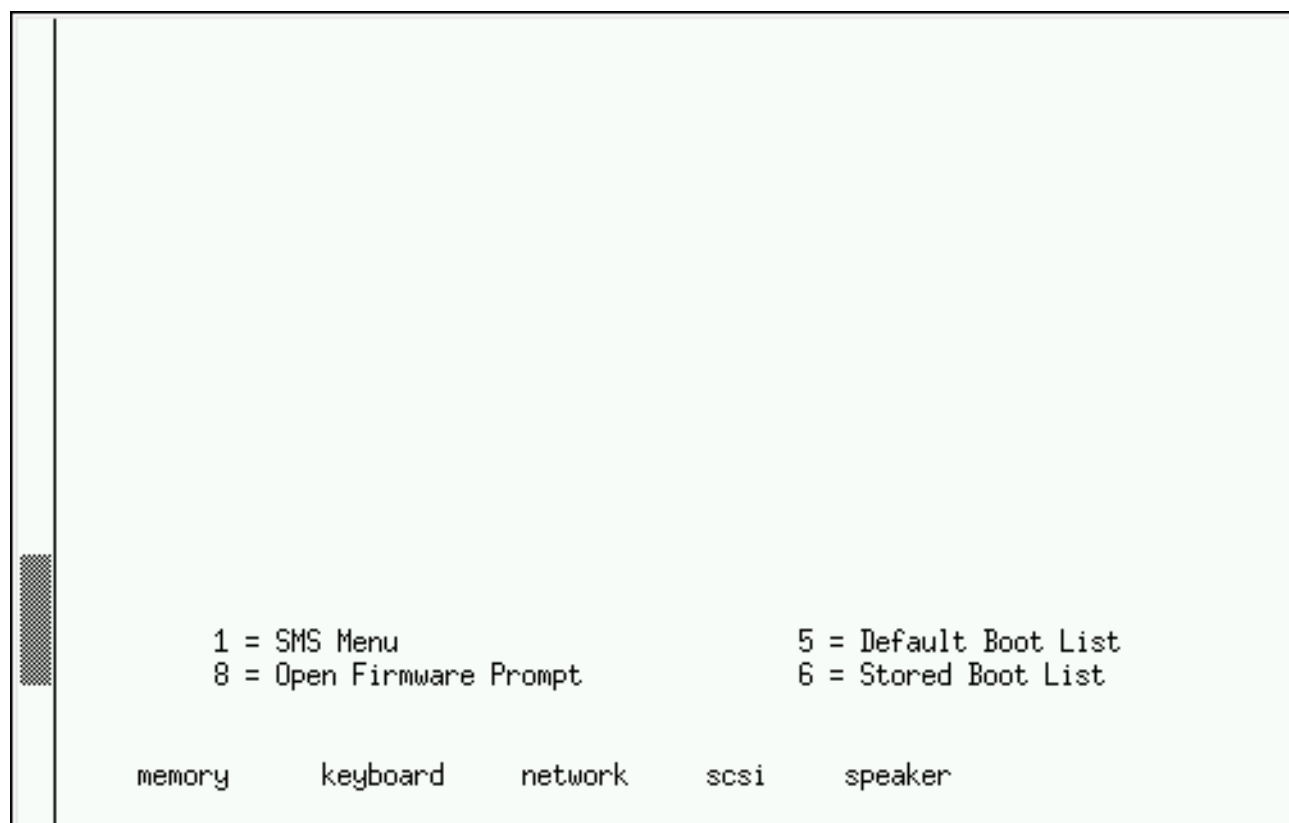
PART I. INSTALLING RED HAT ENTERPRISE LINUX ON IBM POWER

The following text describes how to install Red Hat Enterprise Linux on the IBM Power architecture.

CHAPTER 10. BOOTING THE INSTALLATION ON IBM POWER SYSTEMS

To boot an IBM Power Systems server from a DVD, the install boot device must be specified in the **System Management Services** (SMS) menu.

1. To enter the **System Management Services** console, press the **1** key during the boot process when you hear the chime sound. This brings up a console interface similar to the one described in this section.
2. On a text console, press **1** when the self test displays the banner along with the tested components:



1. Once in the SMS menu, select the **Select Boot Options**.
2. In the Boot Options menu, specify **Select Install or Boot a Device**.
3. There, select **CD/DVD**, and then the bus type (in most cases SCSI). If you are uncertain, you can select to view all devices. This scans all available buses for boot devices, including network adapters and hard drives.
4. Finally, select the device containing the installation DVD. The boot menu will now load.

Because IBM Power Systems servers primarily use text consoles, **Anaconda** will not automatically start a graphical installation. However, the graphical installation program offers more features and customization and is recommended if your system has a graphical display.

10.1. THE BOOT MENU

Once your system has completed loading the boot media, a boot menu is displayed using **GRand Unified Bootloader**, version 2 (**GRUB2**). The boot menu provides several options in addition to

launching the installation program. If no key is pressed within 60 seconds, the default boot option (the one highlighted in white) will be run. To choose the default, either wait for the timer to run out or press **Enter**.

Figure 10.1. The Boot Screen



To select a different option than the default, use the arrow keys on your keyboard, and press **Enter** when the correct option is highlighted.

To customize the boot options for a particular menu entry, press the **e** key and add custom boot options to the command line. When ready press **Ctrl+X** to boot the modified option.

The boot menu options are:

Install Red Hat Enterprise Linux 8.0

Choose this option to install Red Hat Enterprise Linux onto your computer system using the graphical installation program see [Section 4.1, “Introduction to Anaconda”](#).

Test this media & install Red Hat Enterprise Linux 8.0

This option is the default. Prior to starting the installation program, a utility is launched to check the integrity of the installation media.

Troubleshooting >

This item is a separate menu containing options that help resolve various installation issues. When highlighted, press **Enter** to display its contents.

The Troubleshooting Menu

Install Red Hat Enterprise Linux 8.0 in basic graphics mode

This option allows you to install Red Hat Enterprise Linux in graphical mode even if the installation program is unable to load the correct driver for your video card. If your screen appears distorted or goes blank when using the **Install Red Hat Enterprise Linux 8.0** option, restart your computer and try this option instead.

Rescue a Red Hat Enterprise Linux system

Choose this option to repair a problem with your installed Red Hat Enterprise Linux system that prevents you from booting normally. The rescue environment contains utility programs that allow you fix a wide variety of these problems.

Run a memory test, Boot from local drive

This option boots the system from the first installed disk. If you booted this disc accidentally, use this option to boot from the hard disk immediately without starting the installation program.

10.2. INSTALLING FROM A DIFFERENT SOURCE

You can install Red Hat Enterprise Linux from the ISO images stored on hard disk, or from a network using NFS, FTP, HTTP, or HTTPS methods. Experienced users frequently use one of these methods because it is often faster to read data from a hard disk or network server than from a DVD.

The following table summarizes the different boot methods and recommended installation methods to use with each:

Table 10.1. Boot Methods and Installation Sources

Boot method	Installation source
Full installation media (DVD)	The boot media itself
Minimal boot media (CD or DVD)	Full installation DVD ISO image or the installation tree extracted from this image, placed in a network location or on a hard drive
Network boot	Full installation DVD ISO image or the installation tree extracted from this image, placed in a network location

10.3. BOOTING FROM THE NETWORK USING AN INSTALLATION SERVER

Configure the computer to boot from the network interface by selecting **Select Boot Options** in the SMS menu, then **Select Boot/Install Device**. Finally, select your network device from the list of available devices.

Once you properly configure booting from an installation server, the computer can boot the Red Hat Enterprise Linux installation system without any other media.

To boot a computer from a server:

How to Start the Installation Program from the Network

1. Ensure that the network cable is attached. The link indicator light on the network socket should be lit, even if the computer is not switched on.
2. Switch on the computer.
3. Networking setup and diagnostic information usually appears before your computer connects to the server, although this depends on the hardware in use. Then you will see a menu with options specifying how the network boot server is setup. Press the number key that corresponds to the

desired option. In case you are not sure which option you should select, ask your server administrator.

If your system does not boot from the network installation server, ensure that the SMS is configured to boot first from the correct network interface. See your hardware's documentation for more information.



IMPORTANT

Use the **vmlinux** and **initrd.img** images to boot your system over a network.

CHAPTER 11. PLANNING FOR INSTALLATION ON IBM POWER SYSTEMS

This chapter outlines the decisions and preparations you will need to make when deciding how to proceed with the installation.

11.1. IS YOUR HARDWARE COMPATIBLE?

Red Hat Enterprise Linux 8 (little endian) is compatible with IBM Power Systems (ppc64le) and is currently supported on POWER8 and POWER9 processors. It is also supported as a KVM guest on Red Hat Enterprise Virtualization for Power, on PowerVM, and PowerNV (bare metal).



NOTE

Red Hat Enterprise Linux 8 is not being built or delivered in big endian. IBM Power Systems servers which use the POWER6 and POWER7 processor series and older are no longer supported.

11.2. IBM INSTALLATION TOOLS

IBM Installation Toolkit is an optional utility that speeds up the installation of Linux on IBM Power Systems and is especially helpful for those unfamiliar with Linux. You can use the **IBM Installation Toolkit** to: ^[1] * Install and configure Linux on a non-virtualized IBM Power Systems server.

- Install and configure Linux on servers with previously-configured logical partitions (LPARs, also known as virtualized servers).
- Install IBM service and productivity tools on a new or previously installed Linux system. The IBM service and productivity tools include dynamic logical partition (DLPAR) utilities.
- Upgrade system firmware level on IBM Power Systems servers.
- Perform diagnostics or maintenance operations on previously installed systems.
- Migrate a LAMP server (software stack) and application data from a System x to a System p system. A LAMP server is a bundle of open source software. LAMP is an acronym for Linux, **Apache HTTP Server**, **MySQL** relational database, and the PHP (or sometimes Perl, or Python) language.

Documentation for the **IBM Installation Toolkit** for PowerLinux is available in the Linux Information Center at <http://publib.boulder.ibm.com/infocenter/lxinfo/v3r0m0/topic/liaan/powerpack.htm>

PowerLinux service and productivity tools is an optional set of tools that include hardware service diagnostic aids, productivity tools, and installation aids for Linux operating systems on IBM servers based on POWER7, POWER6, POWER5, and POWER4 technology.

Documentation for the service and productivity tools is available in the Linux Information Center at <http://publib.boulder.ibm.com/infocenter/lxinfo/v3r0m0/topic/liaau/liaauraskickoff.htm>

11.2.1. Preparation for IBM Power Systems Servers



IMPORTANT

Ensure that the real-base boot parameter is set to **c00000**, otherwise you might see errors such as:

DEFAULT CATCH!, exception-handler=fff00300

IBM Power Systems servers offer many options for partitioning, virtual or native devices, and consoles.

If you are using a non-partitioned system, you do not need any pre-installation setup. For systems using the HVSI serial console, hook up your console to the T2 serial port.

If using a partitioned system the steps to create the partition and start the installation are largely the same. You should create the partition at the HMC and assign some CPU and memory resources, as well as SCSI and Ethernet resources, which can be either virtual or native. The HMC create partition wizard steps you through the creation.

For more information on creating the partition, see the *Partitioning for Linux with an HMC* PDF in the IBM Systems Hardware Information Center at:

http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/topic/iphbi_p5/iphbibook.pdf

If you are using virtual SCSI resources, rather than native SCSI, you must configure a 'link' to the virtual SCSI serving partition, and then configure the virtual SCSI serving partition itself. You create a 'link' between the virtual SCSI client and server slots using the HMC. You can configure a virtual SCSI server on either Virtual I/O Server (VIOS) or IBM i, depending on which model and options you have.

If you are installing using Intel iSCSI Remote Boot, all attached iSCSI storage devices must be disabled. Otherwise, the installation will succeed but the installed system will not boot.

For more information on using virtual devices, see the IBM Redbooks publication *Virtualizing an Infrastructure with System p and Linux* at: <http://publib-b.boulder.ibm.com/abstracts/sg247499.html>

Once you have your system configured, you need to Activate from the HMC or power it on. Depending on the type of installation, you need to configure SMS to correctly boot the system into the installation program.

11.2.2. Supported Installation Targets

An installation target is a storage device that will store Red Hat Enterprise Linux and boot the system. Red Hat Enterprise Linux supports the following installation targets for AMD64 and Intel 64 systems:

- Storage connected by a standard internal interface, such as SCSI, SATA, or SAS
- Fibre Channel Host Bus Adapters and multipath devices. Some can require vendor-provided drivers.
- Virtualized installation on IBM Power Systems servers is also supported when using Virtual SCSI (vSCSI) adapters in virtual client LPARs

Red Hat does not support installation to USB drives or SD memory cards. For information about the support for third-party virtualization technologies, see the *Red Hat Hardware Compatibility List*, available online at <https://hardware.redhat.com>.



IMPORTANT

On IBM Power Systems servers, the eHEA module fails to initialize if 16GB *huge pages* are assigned to a system or partition and the kernel command line does not contain the huge page parameters. Therefore, when you perform a network installation through an IBM eHEA ethernet adapter, you cannot assign huge pages to the system or partition during the installation. Use *large pages* instead.

11.2.3. System Specifications List

The installation program automatically detects and installs your computer's hardware and you do not usually need to supply the installation program with any specific details about your system. However, when performing certain types of installation, it is important to know specific details about your hardware. For this reason, it is recommended that you record the following system specifications for reference during the installation, depending on your installation type.

- If you plan to use a customized partition layout, record:
 - The model numbers, sizes, types, and interfaces of the hard drives attached to the system. For example, Seagate ST3320613AS 320 GB on SATA0, Western Digital WD7500AAKS 750 GB on SATA1. This will allow you to identify specific hard drives during the partitioning process.
- If you are installing Red Hat Enterprise Linux as an additional operating system on an existing system, record:
 - Information about the partitions used on the system. This information can include file system types, device node names, file system labels, and sizes. This will allow you to identify specific partitions during the partitioning process. Remember that different operating systems identify partitions and drives differently, therefore even if the other operating system is a Unix operating system, the device names can be reported by Red Hat Enterprise Linux differently. This information can usually be found by executing the equivalent of the **mount** command and **blkid** command and in the **/etc/fstab** file.
- If you plan to install from an image on a local hard drive:
 - The hard drive and directory that holds the image.
- If you plan to install from a network location:
 - The make and model numbers of the network adapters on your system. For example, Netgear GA311. This will allow you to identify adapters when manually configuring the network.
 - IP, DHCP, and BOOTP addresses
 - Netmask
 - Gateway IP address
 - One or more name server IP addresses (DNS)
 - The location of the installation source on an FTP server, HTTP (web) server, HTTPS (web) server, or NFS server.
If any of these networking requirements or terms are unfamiliar to you, contact your network administrator for assistance.

- If you plan to install on an iSCSI target:
 - The location of the iSCSI target. Depending on your network, you might also need a CHAP user name and password, and perhaps a reverse CHAP user name and password.
- If your computer is part of a domain:
 - You should verify that the domain name will be supplied by the DHCP server. If not, you will need to input the domain name manually during installation.

11.2.4. Disk Space and Memory Requirements

The disk space used by Red Hat Enterprise Linux must be separate from the disk space used by other operating systems you might have installed on your system.



NOTE

For IBM Power Systems servers, at least three partitions (`/`, **swap** and a **PreP** boot partition) must be dedicated to Red Hat Enterprise Linux.

Red Hat Enterprise Linux requires minimum the following amount of RAM:

Installation type	Minimum required RAM
Local media installation (USB, DVD)	1,280 MiB
NFS network installation	1,280 MiB
HTTP, HTTPS, or FTP network installation	1,664 MiB

Installing Red Hat Enterprise Linux using a Kickstart file has the same minimum RAM requirements as a manual installation. However, if you use a Kickstart file that runs commands which require additional memory or write data to the RAM disk, additional RAM might be necessary.

11.2.5. RAID and Other Disk Devices

Some storage technology requires special consideration when using Red Hat Enterprise Linux. Generally, it is important to understand how these technologies are configured, visible to Red Hat Enterprise Linux, and how support for them might have changed between major versions.

11.2.5.1. Hardware RAID

RAID (Redundant Array of Independent Disks) allows a group, or array, of drives to act as a single device. Configure any RAID functions provided by the mainboard of your computer, or attached controller cards, before you begin the installation process. Each active RAID array appears as one drive within Red Hat Enterprise Linux.

11.2.5.2. Software RAID

**NOTE**

When a pre-existing RAID array's member devices are all unpartitioned disks/drives, the installer will treat the array itself as a disk and will not provide a way to remove the array.

11.2.5.3. USB Disks

You can connect and configure external USB storage after installation. Most such devices are recognized by the kernel and available for use at that time.

Some USB drives might not be recognized by the installation program. If configuration of these disks at installation time is not vital, disconnect them to avoid potential problems.

11.2.6. Choose an Installation Boot Method

You can use several methods to boot the Red Hat Enterprise Linux 8 installation program. The method you choose depends upon your installation media.

**NOTE**

Installation media must remain mounted throughout installation, including during execution of the **%post** section of a kickstart file.

Full installation DVD or USB drive, Minimal boot CD, DVD or USB Flash Drive, PXE Server

11.2.7. Automating the Installation with Kickstart

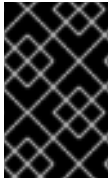
Red Hat Enterprise Linux 8 offers a way to partially or fully automate the installation process using a *Kickstart file*. Kickstart files contain answers to all questions normally asked by the installation program, such as what time zone do you want the system to use, how should the drives be partitioned or which packages should be installed. Providing a prepared Kickstart file at the beginning of the installation therefore allows you to perform the entire installation (or parts of it) automatically, without need for any intervention from the user. This is especially useful when deploying Red Hat Enterprise Linux on a large number of systems at once.

In addition to allowing you to automate the installation, Kickstart files also provide more options regarding software selection. When installing Red Hat Enterprise Linux manually using the graphical installation interface, your software selection is limited to pre-defined environments and add-ons. A Kickstart file allows you to install or remove individual packages as well.

[1] Parts of this section were previously published at IBM's *Linux information for IBM systems* resource at http://publib.boulder.ibm.com/infocenter/lxinfo/v3r0m0/index.jsp?topic=%2Fliaay%2Ftools_overview.htm

CHAPTER 12. UPDATING DRIVERS DURING INSTALLATION ON IBM POWER SYSTEMS

In most cases, Red Hat Enterprise Linux already includes drivers for the devices that make up your system. However, if your system contains hardware that has been released very recently, drivers for this hardware might not yet be included. Sometimes, a driver update that provides support for a new device might be available from Red Hat or your hardware vendor on a *driver disc* that contains *RPM packages*. Typically, the driver disc is available for download as an *ISO image file*.



IMPORTANT

Driver updates should only be performed if a missing driver prevents you to complete the installation successfully. The drivers included in the kernel should always be preferred over drivers provided by other means.

Often, the new hardware does not need to be present during the installation process. For example, if you use a DVD to install to a local hard drive, the installation will succeed even if drivers for your network card are not available. In such a situation, complete the installation and add support for the new hardware afterward.

In other situations, you might want to add drivers for a device during the installation process to support a particular configuration. For example, you might want to install drivers for a network device or a storage adapter card to give the installation program access to the storage devices that your system uses. You can use a driver disc to add this support during installation in one of two ways:

1. Place the ISO image file of the driver disc in a location accessible to the installation program, on a local hard drive, on a USB flash drive, or on a CD or DVD.
2. Create a driver disc by extracting the image file onto a CD or a DVD, or a USB flash drive. See the instructions for making installation discs in for more information on burning ISO image files to a CD or DVD, and for instructions on writing ISO images to USB drives.

If Red Hat, your hardware vendor, or a trusted third party told you that you will require a driver update during the installation process, choose a method to supply the update from the methods described in this chapter and test it before beginning the installation. Conversely, do not perform a driver update during installation unless you are certain that your system requires it. The presence of a driver on a system for which it was not intended can complicate support.



WARNING

Driver update disks sometimes disable conflicting kernel drivers, where necessary. In rare cases, unloading a kernel module in this way can cause installation errors.

12.1. PREPARING FOR A DRIVER UPDATE DURING INSTALLATION

If a driver update is necessary and available for your hardware, Red Hat, your hardware vendor, or another trusted third party will typically provide it in the form of an image file in ISO format. Once you obtain the ISO image, you must decide on the method you want to use to perform the driver update.

The available methods are:

Automatic driver update

When starting the installation, the **Anaconda** installation program will attempt to detect all attached storage devices. If there is a storage device labeled **OEMDRV** present when the installation begins, **Anaconda** will always treat it like a driver update disc and attempt to load drivers present on it.

Assisted driver update

You can specify the **inst.dd** boot option when starting the installation. If you use this option without any parameters, **Anaconda** will display a list of all storage devices connected to the system, and it will prompt you to select a device which contains a driver update.

Manual driver update

You can specify the **inst.dd=location** boot option when starting the installation, where *location* is the path to a driver update disc or ISO image. When you specify this option, **Anaconda** will attempt to load any driver updates it finds at the specified location. With manual driver updates, you can specify either locally available storage devices, or a network location (an **HTTP**, **HTTPS** or **FTP** server).



NOTE

You can also use both **inst.dd=location** and **inst.dd** at the same time. However, what **Anaconda** does in this case depends on the type of *location* that you use. If it is a device, **Anaconda** prompts you to select drivers to update from the specified device and then it offers you additional devices. If *location* is a network location, **Anaconda** first prompts you to select a device containing a driver update and then it lets you update drivers from the specified network location.

If you want to use the automatic driver update method, you must create a storage device labeled **OEMDRV**, and it must be physically connected to the installation system. To use the assisted method, you can use any local storage device any label other than **OEMDRV**. To use the manual method, you can use any local storage with a different label, or a network location accessible from the installation system.

12.1.1. Preparing to Use a Driver Update Image File on Local Storage

If you use a local storage device to provide the ISO file, such as a hard drive or USB flash drive, you can make the installation program to recognize it automatically by properly labeling the device. Only if it is not possible, install the update manually as described below.

- In order for the installation program to automatically recognize the driver disk, the volume label of the storage device must be **OEMDRV**. Also, you will need to extract the contents of the ISO image file to the root directory of the storage device rather than copy the ISO image itself. See Note that installation of a driver from a device labeled **OEMDRV** is always recommended and preferable to the manual installation.
- For manual installation, simply copy the ISO image, as a single file, onto the storage device. You can rename the file if you find it helpful but you must not change the file name extension, which must remain **.iso**, for example **dd.iso**. See to learn how to select the driver update manually during installation.

12.1.2. Preparing a Driver Disc

You can create a driver update disc on a CD or DVD. See to learn more about burning discs from image files.

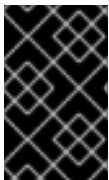
After you burn a driver update disc CD or DVD, verify that the disc was created successfully by inserting it into your system and browsing to it using the file manager. You should see a single file named **rhdd3**, which is a signature file that contains the driver disc's description, and a directory named **rpms**, which contains the RPM packages with the actual drivers for various architectures.

If you see only a single file ending in **.iso**, then you have not created the disc correctly and should try again. Ensure that you choose an option similar to **Burn from Image** if you use a Linux desktop other than **GNOME**, or if you use a different operating system.

12.1.3. Performing a Driver Update During Installation

At the very beginning of the installation process, you can perform a driver update in the following ways:

- Let the installation program automatically find and offer a driver update for installation,
- Let the installation program prompt you to locate a driver update,
- Manually specify a path to a driver update image or an RPM package.



IMPORTANT

Always make sure to put your driver update discs on a standard disk partition. Advanced storage, such as RAID or LVM volumes, might not be accessible during the early stage of the installation when you perform driver updates.

12.1.3.1. Automatic Driver Update

To have the installation program automatically recognize a driver update disc, connect a block device with the **OEMDRV** volume label to your computer before starting the installation process.

When the installation begins, the installation program detects all available storage connected to the system. If it finds a storage device labeled **OEMDRV**, it will treat it as a driver update disc and attempt to load driver updates from this device. You will be prompted to select which drivers to load:

Figure 12.1. Selecting a Driver

```
DD: Checking devices /dev/sr1
DD: Checking device /dev/sr1
DD: Processing DD repo /media/DD//rpms/x86_64 on /dev/sr1

Page 1 of 1
Select drivers to install
  1) [ ] /media/DD//rpms/x86_64/kmod_e10.rpm

# to toggle selection, 'n'-next page, 'p'-previous page or 'c'-continue:
```

Use number keys to toggle selection on individual drivers. When ready, press **c** to install the selected drivers and proceed to the **Anaconda** graphical user interface.

12.1.3.2. Assisted Driver Update

It is always recommended to have a block device with the **OEMDRV** volume label available to install a driver during installation. However, if no such device is detected and the **inst.dd** option was specified at the boot command line, the installation program lets you find the driver disk in interactive mode.

1. First, select a local disk partition from the list for **Anaconda** to scan for ISO files.
2. Then, select one of the detected ISO files.
3. Finally, select one or more available drivers. The image below demonstrates the process in the text user interface with individual steps highlighted.

Figure 12.2. Selecting a Driver Interactively

```

Starting Driver Update Disk UI on tty1...
DD: Checking devices

Page 1 of 1
Driver disk device selection
  DEVICE      TYPE    LABEL      UUID
  1) vda1      ext2    HOME       8c9d0c6e-4fea-4910-9bac-6609bc8ff847
  2) vda2      xfs     DD_PART    9dcc606d-a9ca-41d1-98b5-e9411769e37f
  3) vdb1      ext4    DD_PART    dd69ffa5-c72e-4b61-ae39-0197d6960fc3

# to select, 'n'-next page, 'p'-previous page or 'c'-continue: 3
[ 97.268612] EXT4-fs (vdb1): mounted filesystem without journal. Opts: (null)

Page 1 of 1
Choose driver disk ISO file
  1) dd.iso

# to select, 'n'-next page, 'p'-previous page or 'c'-continue: 1
DD: Checking device /media/DD-search/dd.iso
[ 112.233480] loop: module loaded
DD: Processing DD repo /media/DD/rpms/x86_64 on /media/DD-search/dd.iso

Page 1 of 1
Select drivers to install
  1) [ ] /media/DD/rpms/x86_64/kmod_e10.rpm

# to toggle selection, 'n'-next page, 'p'-previous page or 'c'-continue: 1

Page 1 of 1
Select drivers to install
  1) [x] /media/DD/rpms/x86_64/kmod_e10.rpm

# to toggle selection, 'n'-next page, 'p'-previous page or 'c'-continue: _

```



NOTE

If you extracted your ISO image file and burned it on a CD or DVD but the media does not have the **OEMDRV** volume label, either use the **inst.dd** option with no arguments and use the menu to select the device, or use the following boot option for the installation program to scan the media for drivers:

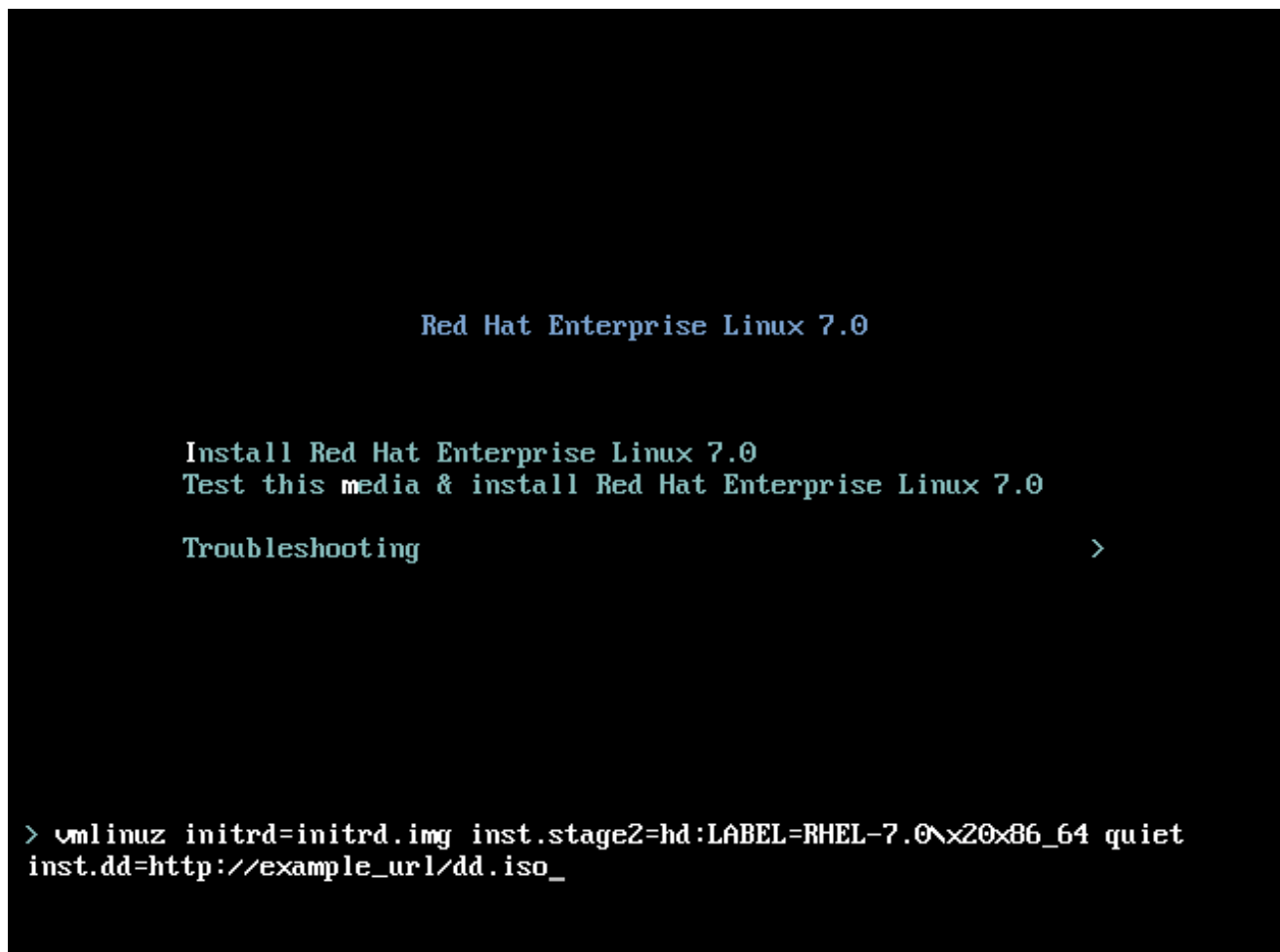
```
inst.dd=/dev/sr0
```

Press the number keys to toggle selection on individual drivers. When ready, press **c** to install the selected drivers and proceed to the **Anaconda** graphical user interface.

12.1.3.3. Manual Driver Update

For manual driver installation, prepare an ISO image file containing your drivers to an accessible location, such a USB flash drive or a web server, and connect it to your computer. At the welcome screen, press **Tab** to display the boot command line and append the **inst.dd=location** to it, where *location* is a path to the driver update disc:

Figure 12.3. Specifying a Path to a Driver Update



Typically, the image file is located on a web server (for example, <http://server.example.com/dd.iso>) or on a USB flash drive (for example, `/dev/sdb1`). It is also possible to specify an RPM package containing the driver update (for example <http://server.example.com/dd.rpm>).

When ready, press **Enter** to execute the boot command. Then, your selected drivers will be loaded and the installation process will proceed normally

12.1.3.4. Blacklisting a Driver

A malfunctioning driver can prevent a system from booting normally during installation. When this happens, you can disable (or blacklist) the driver by customizing the boot command line. At the boot menu, display the boot command line by pressing the **Tab** key. Then, append the **modprobe.blacklist=*driver_name*** option to it. Replace *driver_name* with names of a driver or drivers you want to disable, for example:

```
modprobe.blacklist=ahci
```

Note that the drivers blacklisted during installation using the **modprobe.blacklist=** boot option will remain disabled on the installed system and appear in the **/etc/modprobe.d/anaconda-blacklist.conf** file. See for more information about blacklisting drivers and other boot options.

PART II. INSTALLING RED HAT ENTERPRISE LINUX ON IBM Z

The following text describes how to install Red Hat Enterprise Linux on the IBM Z architecture.

CHAPTER 13. PLANNING FOR INSTALLATION ON IBM Z

13.1. PRE-INSTALLATION

Red Hat Enterprise Linux 8 runs on z 13 or later IBM mainframe systems.

The installation process assumes that you are familiar with the IBM Z and can set up *logical partitions* (LPARs) and z/VM guest virtual machines. For additional information on System z, see <http://www.ibm.com/systems/z>.

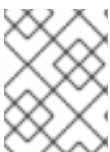
For installation of Red Hat Enterprise Linux on IBM Z, Red Hat supports Direct Access Storage Device (DASD) and Fiber Channel Protocol (FCP) storage devices.

Before you install Red Hat Enterprise Linux, you must decide on the following:

- Decide whether you want to run the operating system on an LPAR or as a z/VM guest operating system.
- Decide if you need swap space and if so, how much. Although it is possible (and recommended) to assign enough memory to a z/VM guest virtual machine and let z/VM do the necessary swapping, there are cases where the amount of required RAM is hard to predict. Such instances should be examined on a case-by-case basis.
- Decide on a network configuration. Red Hat Enterprise Linux 8 for IBM Z supports the following network devices:
 - Real and virtual *Open Systems Adapter* (OSA)
 - Real and virtual HiperSockets
 - *LAN channel station* (LCS) for real OSA

You require the following hardware:

- Disk space. Calculate how much disk space you need and allocate sufficient disk space on DASDs^[2] or SCSI^[3] disks. You require at least 10 GB for a server installation, and 20 GB if you want to install all packages. You also require disk space for any application data. After the installation, you can add or delete more DASD or SCSI disk partitions.
The disk space used by the newly installed Red Hat Enterprise Linux system (the Linux instance) must be separate from the disk space used by other operating systems you have installed on your system.
- RAM. Acquire 1 GB (recommended) for the Linux instance. With some tuning, an instance might run with as little as 512 MB RAM.



NOTE

When initializing swap space on a Fixed Block Architecture (FBA) DASD using the **SWAPGEN** utility, the **FBAPART** option must be used.

13.2. OVERVIEW OF THE SYSTEM Z INSTALLATION PROCEDURE

You can install Red Hat Enterprise Linux on System z interactively or in unattended mode. Installation on System z differs from installation on other architectures in that it is typically performed over a network and not from local media. The installation consists of two phases:

1. Booting the Installation Connect with the mainframe, then perform an initial program load (IPL), or boot, from the medium containing the installation program.
2. Anaconda Use the **Anaconda** installation program to configure network, specify language support, installation source, software packages to be installed, and to perform the rest of the installation.

13.2.1. Booting the Installation

After establishing a connection with the mainframe, you need to perform an initial program load (IPL), or boot, from the medium containing the installation program. This document describes the most common methods of installing Red Hat Enterprise Linux on System z. In general, you can use any method to boot the Linux installation system, which consists of a kernel (**kernel.img**) and initial RAM disk (**initrd.img**) with at least the parameters in the **generic.prm** file. Additionally, a **generic.ins** file is loaded which determines file names and memory addresses for the initrd, kernel and generic.prm.

The Linux installation system is also called the *installation program* in this book.

The control point from where you can start the IPL process depends on the environment where your Linux is to run. If your Linux is to run as a z/VM guest operating system, the control point is the *control program* (CP) of the hosting z/VM. If your Linux is to run in LPAR mode, the control point is the mainframe's *Support Element* (SE) or an attached IBM Z *Hardware Management Console* (HMC).

You can use the following boot media only if Linux is to run as a guest operating system under z/VM:

- z/VM reader -

You can use the following boot media only if Linux is to run in LPAR mode:

- SE or HMC through a remote FTP server -
- SE or HMC DVD -

You can use the following boot media for both z/VM and LPAR:

- DASD -
- SCSI device that is attached through an FCP channel -
- FCP-attached SCSI DVD -

If you use DASD and FCP-attached SCSI devices (except SCSI DVDs) as boot media, you must have a configured **zip1** boot loader.

13.2.2. Installation using Anaconda

In the second installation phase, you will use the **Anaconda** installation program in graphical, text-based, or command-line mode:

Graphical Mode

Graphical installation is done through a VNC client. You can use your mouse and keyboard to navigate through the screens, click buttons, and type into text fields see [Section 4.1, "Introduction to Anaconda"](#).

Text-based Mode

This interface does not offer all interface elements of the GUI and does not support all settings. Use this for interactive installations if you cannot use a VNC client.

Command-line Mode

This is intended for automated and non-interactive installations on System z. Note that if the installation program encounters an invalid or missing kickstart command, the system will reboot.

In Red Hat Enterprise Linux 8 the text-based installation has been reduced to minimize user interaction. Features like installation on FCP-attached SCSI devices, customizing partition layout, or package add-on selection are only available with the graphical user interface installation. Use the graphical installation whenever possible.

[2] *Direct Access Storage Devices* (DASDs) are hard disks that allow a maximum of three partitions per device. For example, **dasda** can have partitions **dasda1**, **dasda2**, and **dasda3**.

[3] Using the SCSI-over-Fibre Channel device driver (the **zfcp** device driver) and a switch, SCSI LUNs can be presented to Linux on System z as if they were locally attached SCSI drives.

CHAPTER 14. BOOTING THE INSTALLATION ON IBM Z

The steps to perform the initial program boot (IPL) of the **Anaconda** installation program depend on the environment (either z/VM or LPAR) in which Red Hat Enterprise Linux will run.

14.1. CUSTOMIZING BOOT PARAMETERS

Before the installation can begin, you must configure some mandatory boot parameters. When installing through z/VM, these parameters must be configured before you boot in the **generic.prm** file. When installing on an LPAR, the **rd.cmdline** parameter is set to **ask** by default, meaning that you will be given a prompt on which you can enter these boot parameters. In both cases, the required parameters are the same.

Unlike Red Hat Enterprise Linux 6, which featured an interactive utility to assist network configuration, all network configuration must now be specified by the use of the following parameters, either by using a parameter file, or at the prompt.

Installation source

An installation source must always be configured. Use the **inst.repo=** option to specify the package source for the installation.

Network devices

Network configuration must be provided if network access will be required during the installation. If you plan to perform an unattended (Kickstart-based) installation using only local media such as a hard drive, network configuration can be omitted.

Use the **ip=** option for basic network configuration, and other options listed in as required.

Also use the **rd.znet=** kernel option, which takes a network protocol type, a comma delimited list of sub-channels, and, optionally, comma delimited **sysfs** parameter and value pairs. This parameter can be specified multiple times to activate multiple network devices.

For example:

```
rd.znet=qeth,0.0.0600,0.0.0601,0.0.0602,layer2=1,portname=foo
rd.znet=ctc,0.0.0600,0.0.0601,protocol=bar
```

Storage devices

At least one storage device must always be configured.

The **rd.dasd=** option takes a Direct Access Storage Device (DASD) adapter device bus identifier and, optionally, comma separated **sysfs** parameter and value pairs, then activates the device. This parameter can be specified multiple times to activate multiple DASDs. Example:

```
rd.dasd=0.0.0200,readonly=0
rd.dasd=0.0.0202,readonly=0
```

The **rd.zfcp=** option takes a SCSI over FCP (zFCP) adapter device bus identifier, a world wide port name (WWPN), and a FCP LUN, then activates the device. This parameter can be specified multiple times to activate multiple zFCP devices. Example:

```
rd.zfcp=0.0.4000,0x5005076300C213e9,0x5022000000000000
```

Kickstart options

If you are using a Kickstart file to perform an automatic installation, you must always specify the location of the Kickstart file using the **inst.ks=** option. For an unattended, fully automatic Kickstart installation, the **inst.cmdline** option is also useful.

An example customized **generic.prm** file containing all mandatory parameters look similar to the following example:

Example 14.1. Customized generic.prm file

```
ro ramdisk_size=40000 cio_ignore=all,!condev
inst.repo=http://example.com/path/to/repository
rd.znet=qeth,0.0.0600,0.0.0601,0.0.0602,layer2=1,portno=0,portname=foo
ip=192.168.17.115::192.168.17.254:24:foobar.systemz.example.com:enccw0.0
.0600:none
nameserver=192.168.17.1
rd.dasd=0.0.0200 rd.dasd=0.0.0202
rd.zfcp=0.0.4000,0x5005076300C213e9,0x5022000000000000
inst.ks=http://example.com/path/to/kickstart
```

Some installation methods also require a file with a mapping of the location of installation data in the file system of the DVD or FTP server and the memory locations where the data is to be copied. The file is typically named **generic.ins**, and contains file names for the initial RAM disk, kernel image, and parameter file (**generic.prm**) and a memory location for each file. An example **generic.ins** will look similar to the following example:

Example 14.2. Sample generic.ins file

```
images/kernel.img 0x00000000
images/initrd.img 0x02000000
images/genericdvd.prm 0x00010480
images/initrd.addrsize 0x00010408
```

A valid **generic.ins** file is provided by Red Hat along with all other files required to boot the installer. Modify this file only if you want to, for example, load a different kernel version than default.

14.2. CONSIDERATIONS FOR HARD DRIVE INSTALLATION ON IBM Z

If you want to boot the installation program from a hard drive, you can optionally install the **zipl** boot loader on the same (or a different) disk. Be aware that **zipl** only supports one boot record per disk. If you have multiple partitions on a disk, they all "share" the disk's single boot record.

To prepare a hard drive to boot the installation program, install the **zipl** boot loader on the hard drive by entering the following command:

```
# zipl -V -t /mnt/ -i /mnt/images/kernel.img -r /mnt/images/initrd.img -p
/mnt/images/generic.prm
```

See [Section 14.1, "Customizing boot parameters"](#) for details on customizing boot parameters in the **generic.prm** configuration file.

14.3. INSTALLING UNDER Z/VM

When installing under z/VM, you can boot from:

- the z/VM virtual reader
- a DASD or an FCP-attached SCSI device prepared with the **zipl** boot loader
- an FCP-attached SCSI DVD drive

Log on to the z/VM guest virtual machine chosen for the Linux installation. You can use the **x3270** or **c3270** terminal emulator, available in the **x3270-text** package in Red Hat Enterprise Linux, to log in to z/VM from other Linux systems. Alternatively, use the IBM 3270 terminal emulator on the IBM Z Hardware Management Console (HMC). If you are working from a machine with a Microsoft Windows operating system, Jolly Giant (<http://www.jollygiant.com/>) offers an SSL-enabled 3270 emulator. A free native Windows port of **c3270** called **wc3270** also exists.



NOTE

If your 3270 connection is interrupted and you cannot log in again because the previous session is still active, you can replace the old session with a new one by entering the following command on the z/VM logon screen:

```
logon user here
```

Replace *user* with the name of the z/VM guest virtual machine. Depending on whether an external security manager, for example RACF, is used, the logon command might vary.

If you are not already running **CMS** (single-user operating system shipped with z/VM) in your guest, boot it now by entering the command:

```
cp ip1 cms
```

Be sure not to use CMS disks such as your A disk (often device number 0191) as installation targets. To find out which disks are in use by CMS, use the following query:

```
query disk
```

You can use the following CP (z/VM Control Program, which is the z/VM hypervisor) query commands to find out about the device configuration of your z/VM guest virtual machine:

- Query the available main memory, which is called *storage* in IBM z terminology. Your guest should have at least 1 GB of main memory.

```
cp query virtual storage
```

- Query available network devices by type:

osa

OSA - CHPID type OSD, real or virtual (VSWITCH or GuestLAN), both in QDIO mode

hsi

HiperSockets - CHPID type IQD, real or virtual (GuestLAN type Hipers)

lcs

LCS - CHPID type OSE

For example, to query all of the network device types mentioned above, run:

```
cp query virtual osa
```

- Query available DASDs. Only those that are flagged **RW** for read-write mode can be used as installation targets:

```
cp query virtual dasd
```

- Query available FCP channels:

```
cp query virtual fcp
```

14.3.1. Using the z/VM Reader

Perform the following steps to boot from the z/VM reader:

1. If necessary, add the device containing the z/VM TCP/IP tools to your CMS disk list. For example:

```
cp link tcpmaint 592 592
acc 592 fm
```

Replace *fm* with any **FILEMODE** letter.

2. Execute the command:

```
ftp host
```

Where *host* is the host name or IP address of the FTP server that hosts the boot images (**kernel.img** and **initrd.img**).

3. Log in and execute the following commands. Use the **(repl** option if you are overwriting existing **kernel.img**, **initrd.img**, **generic.prm**, or **redhat.exec** files:

```
cd /location/of/install-tree/images/
ascii
get generic.prm (repl
get redhat.exec (repl
locsite fix 80
binary
get kernel.img (repl
get initrd.img (repl
quit
```

4. Optionally, check whether the files were transferred correctly by using the CMS command **filelist** to show the received files and their format. It is important that **kernel.img** and **initrd.img** have a fixed record length format denoted by **F** in the Format column and a record length of 80 in the **Lrec1** column. For example:

```
VMUSER FILELIST A0 V 169 Trunc=169 Size=6 Line=1 Col=1 Alt=0
```

```

Cmd Filename Filetype Fm Format Lrecl Records Blocks Date Time
REDHAT EXEC B1 V 22 1 1 4/15/10 9:30:40
GENERIC PRM B1 V 44 1 1 4/15/10 9:30:32
INITRD IMG B1 F 80 118545 2316 4/15/10 9:30:25
KERNEL IMG B1 F 80 74541 912 4/15/10 9:30:17

```

Press **PF3** to quit **filelist** and return to the CMS prompt.

5. Customize boot parameters in **generic.prm** as necessary. See [Section 14.1, “Customizing boot parameters”](#) for details.
Another way to configure storage and network devices is by using a CMS configuration file. In such a case, add the **CMSDASD=** and **CMSCONFFILE=** parameters to **generic.prm**.
6. Finally, execute the REXX script **redhat.exec** to boot the installation program:

```
redhat
```

14.3.2. Using a Prepared DASD

Boot from the prepared DASD and select the **zipl** boot menu entry referring to the Red Hat Enterprise Linux installation program. Use a command of the following form:

```
cp ip1 DASD_device_number loadparm boot_entry_number
```

Replace *DASD_device_number* with the device number of the boot device, and *boot_entry_number* with the **zipl** configuration menu for this device. For example:

```
cp ip1 eb1c loadparm 0
```

14.3.3. Using a Prepared FCP-attached SCSI Disk

Perform the following steps to boot from a prepared FCP-attached SCSI disk:

1. Configure the SCSI boot loader of z/VM to access the prepared SCSI disk in the FCP Storage Area Network. Select the prepared **zipl** boot menu entry referring to the Red Hat Enterprise Linux installation program. Use a command of the following form:

```
cp set loaddev portname WWPN lun LUN bootprog boot_entry_number
```

Replace *WWPN* with the World Wide Port Name of the storage system and *LUN* with the Logical Unit Number of the disk. The 16-digit hexadecimal numbers must be split into two pairs of eight digits each. For example:

```
cp set loaddev portname 50050763 050b073d lun 40204011 00000000
bootprog 0
```

2. Optionally, confirm your settings with the command:

```
query loaddev
```

3. Boot the FCP device connected with the storage system containing the disk with the following command:

```
-
```

```
cp ip1 FCP_device
```

For example:

```
cp ip1 fc00
```

14.3.4. Using an FCP-attached SCSI DVD Drive

This requires a SCSI DVD drive attached to an FCP-to-SCSI bridge which is in turn connected to an FCP adapter in your IBM z. The FCP adapter must be configured and available under z/VM.

1. Insert your Red Hat Enterprise Linux for IBM z DVD into the DVD drive.
2. Configure the SCSI boot loader of z/VM to access the DVD drive in the FCP Storage Area Network and specify **1** for the boot entry on the Red Hat Enterprise Linux for IBM z DVD. Use a command of the following form:

```
cp set loaddev portname WWPN lun FCP_LUN bootprog 1
```

Replace *WWPN* with the WWPN of the FCP-to-SCSI bridge and *FCP_LUN* with the LUN of the DVD drive. The 16-digit hexadecimal numbers must be split into two pairs of eight characters each. For example:

```
cp set loaddev portname 20010060 eb1c0103 lun 00010000 00000000
bootprog 1
```

3. Optionally, confirm your settings with the command:

```
cp query loaddev
```

4. IPL on the FCP device connected with the FCP-to-SCSI bridge.

```
cp ip1 FCP_device
```

For example:

```
cp ip1 fc00
```

14.4. INSTALLING IN AN LPAR

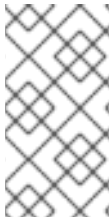
When installing in a *logical partition* (LPAR), you can boot from:

- an FTP server
- a DASD or an FCP-attached SCSI drive prepared with the **zipl** boot loader
- an FCP-attached SCSI DVD drive

Perform these common steps first:

1. Log in on the *IBM Z Hardware Management Console (HMC)* or the *Support Element (SE)* as a user with sufficient privileges to install a new operating system to an LPAR. The **SYSPROG** user is recommended.
2. Select **Images**, then select the LPAR to which you want to install. Use the arrows in the frame on the right side to navigate to the CPC Recovery menu.
3. Double-click **Operating System Messages** to show the text console on which Linux boot messages will appear.

Continue with the procedure for your installation source.



NOTE

Once you finish this procedure and one of the following ones depending on your installation source, the installation will begin. The installer will then prompt you to provide additional boot parameters. Required parameters are described in [Section 14.1](#), “Customizing boot parameters”.

14.4.1. Using an FTP Server

1. Double-click **Load from CD-ROM, DVD, or Server**.
2. In the dialog box that follows, select **FTP Source**, and enter the following information:
 - **Host Computer** - Host name or IP address of the FTP server you want to install from, for example **ftp.redhat.com**
 - **User ID** - Your user name on the FTP server. Or, specify **anonymous**.
 - **Password** - Your password. Use your email address if you are logging in as **anonymous**.
 - **Account (optional)** - Leave this field empty.
 - **File location (optional)** - Directory on the FTP server holding the Red Hat Enterprise Linux for IBM z, for example **/rhel/s390x/**.
3. Click **Continue**.
4. In the dialog that follows, keep the default selection of **generic.ins** and click **Continue**.

14.4.2. Using a Prepared DASD

1. Double-click **Load**.
2. In the dialog box that follows, select **Normal** as the **Load type**.
3. As **Load address**, fill in the device number of the DASD.
4. As **Load parameter**, fill in the number corresponding the **zipl** boot menu entry that you prepared for booting the Red Hat Enterprise Linux installation program.
5. Click the **OK** button.

14.4.3. Using a Prepared FCP-attached SCSI Disk

1. Double-click **Load**.
2. In the dialog box that follows, select **SCSI** as the **Load type**.
3. As **Load address**, fill in the device number of the FCP channel connected with the SCSI disk.
4. As **World wide port name**, fill in the WWPN of the storage system containing the disk as a 16-digit hexadecimal number.
5. As **Logical unit number**, fill in the LUN of the disk as a 16-digit hexadecimal number.
6. As **Boot program selector**, fill in the number corresponding the **zipl** boot menu entry that you prepared for booting the Red Hat Enterprise Linux installation program.
7. Leave the **Boot record logical block address** as **0** and the **Operating system specific load parameters** empty.
8. Click the **OK** button.

14.4.4. Using an FCP-attached SCSI DVD Drive

This requires a SCSI DVD drive attached to an FCP-to-SCSI bridge which is in turn connected to an FCP adapter in your IBM z machine. The FCP adapter must be configured and available in your LPAR.

1. Insert your Red Hat Enterprise Linux for IBM z DVD into the DVD drive.
2. Double-click **Load**.
3. In the dialog box that follows, select **SCSI** as the **Load type**.
4. As **Load address**, fill in the device number of the FCP channel connected with the FCP-to-SCSI bridge.
5. As **World wide port name**, fill in the WWPN of the FCP-to-SCSI bridge as a 16-digit hexadecimal number.
6. As **Logical unit number**, fill in the LUN of the DVD drive as a 16-digit hexadecimal number.
7. As **Boot program selector**, fill in the number **1** to select the boot entry on the Red Hat Enterprise Linux for IBM z DVD.
8. Leave the **Boot record logical block address** as **0** and the **Operating system specific load parameters** empty.
9. Click the **OK** button.

CHAPTER 15. CONFIGURING AN INSTALLED LINUX ON IBM Z INSTANCE

15.1. ADDING DASDS

Direct Access Storage Devices (DASDs) are a type of storage commonly used with IBM Z. Additional information about working with these storage devices can be found at the IBM Knowledge Center at http://www-01.ibm.com/support/knowledgecenter/linuxonibm/com.ibm.linux.z.lgdd/lgdd_t_dasd_wrk.html.

The following is an example of how to set a DASD online, format it, and make the change persistent.

Make sure the device is attached or linked to the Linux system if running under z/VM.

```
CP ATTACH EB1C TO *
```

To link a mini disk to which you have access, issue, for example:

```
CP LINK RHEL7X 4B2E 4B2E MR
DASD 4B2E LINKED R/W
```

See *z/VM: CP Commands and Utilities Reference, SC24-6175* for details about the commands.

15.1.1. Dynamically Setting DASDs Online

To set a DASD online, follow these steps:

1. Use the **cio_ignore** utility to remove the DASD from the list of ignored devices and make it visible to Linux:

```
# cio_ignore -r device_number
```

Replace *device_number* with the device number of the DASD. For example:

```
# cio_ignore -r 4b2e
```

2. Set the device online. Use a command of the following form:

```
# chccwdev -e device_number
```

Replace *device_number* with the device number of the DASD. For example:

```
# chccwdev -e 4b2e
```

As an alternative, you can set the device online using sysfs attributes:

- a. Use the **cd** command to change to the `/sys/` directory that represents that volume:

```
# cd /sys/bus/ccw/drivers/dasd-eckd/0.0.4b2e/
# ls -l
total 0
-r--r--r-- 1 root root 4096 Aug 25 17:04 availability
-rw-r--r-- 1 root root 4096 Aug 25 17:04 cmb_enable
```

```
-r--r--r-- 1 root root 4096 Aug 25 17:04 cutype
-rw-r--r-- 1 root root 4096 Aug 25 17:04 detach_state
-r--r--r-- 1 root root 4096 Aug 25 17:04 devtype
-r--r--r-- 1 root root 4096 Aug 25 17:04 discipline
-rw-r--r-- 1 root root 4096 Aug 25 17:04 online
-rw-r--r-- 1 root root 4096 Aug 25 17:04 readonly
-rw-r--r-- 1 root root 4096 Aug 25 17:04 use_diag
```

b. Check to see if the device is already online:

```
# cat online
0
```

c. If it is not online, enter the following command to bring it online:

```
# echo 1 > online
# cat online
1
```

3. Verify which block devnode it is being accessed as:

```
# ls -l
total 0
-r--r--r-- 1 root root 4096 Aug 25 17:04 availability
lrwxrwxrwx 1 root root    0 Aug 25 17:07 block ->
../../../.././block/dasdb
-rw-r--r-- 1 root root 4096 Aug 25 17:04 cmb_enable
-r--r--r-- 1 root root 4096 Aug 25 17:04 cutype
-rw-r--r-- 1 root root 4096 Aug 25 17:04 detach_state
-r--r--r-- 1 root root 4096 Aug 25 17:04 devtype
-r--r--r-- 1 root root 4096 Aug 25 17:04 discipline
-rw-r--r-- 1 root root    0 Aug 25 17:04 online
-rw-r--r-- 1 root root 4096 Aug 25 17:04 readonly
-rw-r--r-- 1 root root 4096 Aug 25 17:04 use_diag
```

As shown in this example, device 4B2E is being accessed as `/dev/dasdb`.

These instructions set a DASD online for the current session, but this is not persistent across reboots. For instructions on how to set a DASD online persistently, see [Section 15.1.3, “Persistently Setting DASDs Online”](#). When you work with DASDs, use the persistent device symbolic links under `/dev/disk/by-path/`.

15.1.2. Preparing a New DASD with Low-level Formatting

Once the disk is online, change back to the `/root` directory and low-level format the device. This is only required once for a DASD during its entire lifetime:

```
# cd /root
# dasdfmt -b 4096 -d cdl -p /dev/disk/by-path/ccw-0.0.4b2e
Drive Geometry: 10017 Cylinders * 15 Heads = 150255 Tracks
```

I am going to format the device `/dev/disk/by-path/ccw-0.0.4b2e` in the following way:

```
Device number of device : 0x4b2e
```

```

Labelling device      : yes
Disk label            : VOL1
Disk identifier       : 0X4B2E
Extent start (trk no) : 0
Extent end (trk no)   : 150254
Compatible Disk Layout : yes
Blocksize             : 4096

--->> ATTENTION! <---
All data of that device will be lost.
Type "yes" to continue, no will leave the disk untouched: yes
cyl    97 of  3338 |#-----|
2%

```

When the progress bar reaches the end and the format is complete, **dasdfmt** prints the following output:

```

Rereading the partition table...
Exiting...

```

Now, use **fdasd** to partition the DASD. You can create up to three partitions on a DASD. In our example here, we create one partition spanning the whole disk:

```

# fdasd -a /dev/disk/by-path/ccw-0.0.4b2e
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
checking !
wrote NATIVE!
rereading partition table...

```

After a (low-level formatted) DASD is online, it can be used like any other disk under Linux. For instance, you can create file systems, LVM physical volumes, or swap space on its partitions, for example **/dev/disk/by-path/ccw-0.0.4b2e-part1**. Never use the full DASD device (**dev/dasdb**) for anything but the commands **dasdfmt** and **fdasd**. If you want to use the entire DASD, create one partition spanning the entire drive as in the **fdasd** example above.

To add additional disks later without breaking existing disk entries in, for example, **/etc/fstab**, use the persistent device symbolic links under **/dev/disk/by-path/**.

15.1.3. Persistently Setting DASDs Online

The above instructions described how to activate DASDs dynamically in a running system. However, such changes are not persistent and do not survive a reboot. Making changes to the DASD configuration persistent in your Linux system depends on whether the DASDs belong to the root file system. Those DASDs required for the root file system need to be activated very early during the boot process by the **initramfs** to be able to mount the root file system.

The **cio_ignore** commands are handled transparently for persistent device configurations and you do not need to free devices from the ignore list manually.

15.1.3.1. DASDs That Are Part of the Root File System

The only file you have to modify to add DASDs that are part of the root file system is **/etc/zipl.conf**. Then run the **zipl** boot loader tool. There is no need to recreate the **initramfs**.

There is one boot option to activate DASDs early in the boot process: **rd.dasd=**. This option takes a comma-separated list as input. The list contains a device bus ID and optional additional parameters consisting of key-value pairs that correspond to DASD **sysfs** attributes.

Below is an example **zipl.conf** for a system that uses physical volumes on partitions of two DASDs for an LVM volume group **vg_devel1** that contains a logical volume **lv_root** for the root file system.

```
[defaultboot]
default=linux
target=/boot/

[linux]
image=/boot/vmlinuz-2.6.32-19.el7.s390x
ramdisk=/boot/initramfs-2.6.32-19.el7.s390x.img
parameters="root=/dev/mapper/vg_devel1-lv_root
rd.dasd=0.0.0200,use_diag=0,readonly=0,erplog=0,failfast=0
rd.dasd=0.0.0207,use_diag=0,readonly=0,erplog=0,failfast=0
rd_LVM_LV=vg_devel1/lv_root rd_NO_LUKS rd_NO_MD rd_NO_DM LANG=en_US.UTF-8
SYSEFONT=latacyrheb-sun16 KEYTABLE=us cio_ignore=all,!condev"
```

Suppose that you want to add another physical volume on a partition of a third DASD with device bus ID **0.0.202b**. To do this, add **rd.dasd=0.0.202b** to the parameters line of your boot kernel in **zipl.conf**:

```
[defaultboot]
default=linux
target=/boot/

[linux]
image=/boot/vmlinuz-2.6.32-19.el7.s390x
ramdisk=/boot/initramfs-2.6.32-19.el7.s390x.img
parameters="root=/dev/mapper/vg_devel1-lv_root
rd.dasd=0.0.0200,use_diag=0,readonly=0,erplog=0,failfast=0
rd.dasd=0.0.0207,use_diag=0,readonly=0,erplog=0,failfast=0
rd.dasd=0.0.202b rd_LVM_LV=vg_devel1/lv_root rd_NO_LUKS rd_NO_MD rd_NO_DM
LANG=en_US.UTF-8 SYSEFONT=latacyrheb-sun16 KEYTABLE=us
cio_ignore=all,!condev"
```



WARNING

Make sure the length of the kernel command line in **/etc/zipl.conf** does not exceed 896 bytes. Otherwise, the boot loader cannot be saved, and the installation fails.

Run **zipl** to apply the changes of **/etc/zipl.conf** for the next IPL:

```
# zipl -V
Using config file '/etc/zipl.conf'
Target device information
```

```

Device.....: 5e:00
Partition.....: 5e:01
Device name.....: dasda
DASD device number.....: 0201
Type.....: disk partition
Disk layout.....: ECKD/compatible disk layout
Geometry - heads.....: 15
Geometry - sectors.....: 12
Geometry - cylinders.....: 3308
Geometry - start.....: 24
File system block size.....: 4096
Physical block size.....: 4096
Device size in physical blocks..: 595416
Building bootmap in '/boot/'
Building menu 'rh-automatic-menu'
Adding #1: IPL section 'linux' (default)
kernel image.....: /boot/vmlinuz-2.6.32-19.el7.s390x
kernel parmline...: 'root=/dev/mapper/vg_devel1-lv_root
rd.dasd=0.0.0200,use_diag=0,readonly=0,erplog=0,failfast=0
rd.dasd=0.0.0207,use_diag=0,readonly=0,erplog=0,failfast=0
rd.dasd=0.0.202b rd_LVM_LV=vg_devel1/lv_root rd_NO_LUKS rd_NO_MD rd_NO_DM
LANG=en_US.UTF-8 SYSFONT=latarcyrheb-sun16 KEYTABLE=us
cio_ignore=all,!condev'
initial ramdisk...: /boot/initramfs-2.6.32-19.el7.s390x.img
component address:
kernel image....: 0x00010000-0x00a70fff
parmline.....: 0x00001000-0x00001fff
initial ramdisk.: 0x02000000-0x022d2fff
internal loader.: 0x0000a000-0x0000afff
Preparing boot device: dasda (0201).
Preparing boot menu
Interactive prompt.....: enabled
Menu timeout.....: 15 seconds
Default configuration...: 'linux'
Syncing disks...
Done.

```

15.1.3.2. DASDs That Are Not Part of the Root File System

DASDs that are not part of the root file system, that is, *data disks*, are persistently configured in the file `/etc/dasd.conf`. It contains one DASD per line. Each line begins with the device bus ID of a DASD. Optionally, each line can continue with options separated by space or tab characters. Options consist of key-value-pairs, where the key and value are separated by an equals sign.

The key corresponds to any valid **sysfs** attribute a DASD can have. The value will be written to the key's **sysfs** attribute. Entries in `/etc/dasd.conf` are activated and configured by **udev** when a DASD is added to the system. At boot time, all DASDs visible to the system get added and trigger **udev**.

Example content of `/etc/dasd.conf`:

```

0.0.0207
0.0.0200 use_diag=1 readonly=1

```

Modifications of `/etc/dasd.conf` only become effective after a reboot of the system or after the dynamic addition of a new DASD by changing the system's I/O configuration (that is, the DASD is

attached under z/VM). Alternatively, you can trigger the activation of a new entry in `/etc/dasd.conf` for a DASD which was previously not active, by executing the following commands:

1. Use the **cio_ignore** utility to remove the DASD from the list of ignored devices and make it visible to Linux:

```
# cio_ignore -r device_number
```

For example:

```
# cio_ignore -r 021a
```

2. Trigger the activation by writing to the **uevent** attribute of the device:

```
# echo add > /sys/bus/ccw/devices/device-bus-ID/uevent
```

For example:

```
# echo add > /sys/bus/ccw/devices/0.0.021a/uevent
```

15.2. ADDING FCP-ATTACHED LOGICAL UNITS (LUNS)

If running under z/VM, make certain the FCP adapter is attached to the z/VM guest virtual machine. For multipathing in production environments there would be at least two FCP devices on two different physical adapters (CHPIDs). For example:

```
CP ATTACH FC00 TO *
CP ATTACH FCD0 TO *
```

15.2.1. Dynamically Activating an FCP LUN

Follow these steps to activate a LUN:

1. Use the **cio_ignore** utility to remove the FCP adapter from the list of ignored devices and make it visible to Linux:

```
# cio_ignore -r device_number
```

Replace *device_number* with the device number of the FCP adapter. For example:

2. To bring the FCP adapter device online, use the following command:

```
# chccwdev -e fc00
```

3. Verify that the required WWPN was found by the automatic port scanning of the zfcplib device driver:

```
# ls -l /sys/bus/ccw/drivers/zfcplib/0.0.fc00/
drwxr-xr-x. 3 root root 0 Apr 28 18:19 0x500507630040710b
drwxr-xr-x. 3 root root 0 Apr 28 18:19 0x50050763050b073d
drwxr-xr-x. 3 root root 0 Apr 28 18:19 0x500507630e060521
drwxr-xr-x. 3 root root 0 Apr 28 18:19 0x500507630e860521
```



```

-r--r--r--. 1 root root 4096 Apr 28 18:17 availability
-r--r--r--. 1 root root 4096 Apr 28 18:19 card_version
-rw-r--r--. 1 root root 4096 Apr 28 18:17 cmb_enable
-r--r--r--. 1 root root 4096 Apr 28 18:17 cutype
-r--r--r--. 1 root root 4096 Apr 28 18:17 devtype
lrwxrwxrwx. 1 root root    0 Apr 28 18:17 driver ->
../../../../bus/ccw/drivers/zfcp
-rw-r--r--. 1 root root 4096 Apr 28 18:17 failed
-r--r--r--. 1 root root 4096 Apr 28 18:19 hardware_version
drwxr-xr-x. 35 root root    0 Apr 28 18:17 host0
-r--r--r--. 1 root root 4096 Apr 28 18:17 in_recovery
-r--r--r--. 1 root root 4096 Apr 28 18:19 lic_version
-r--r--r--. 1 root root 4096 Apr 28 18:17 modalias
-rw-r--r--. 1 root root 4096 Apr 28 18:17 online
-r--r--r--. 1 root root 4096 Apr 28 18:19 peer_d_id
-r--r--r--. 1 root root 4096 Apr 28 18:19 peer_wwnn
-r--r--r--. 1 root root 4096 Apr 28 18:19 peer_wwpn
--w-----. 1 root root 4096 Apr 28 18:19 port_remove
--w-----. 1 root root 4096 Apr 28 18:19 port_rescan
drwxr-xr-x. 2 root root    0 Apr 28 18:19 power
-r--r--r--. 1 root root 4096 Apr 28 18:19 status
lrwxrwxrwx. 1 root root    0 Apr 28 18:17 subsystem ->
../../../../bus/ccw
-rw-r--r--. 1 root root 4096 Apr 28 18:17 uevent

```

4. Activate the FCP LUN by adding it to the port (WWPN) through which you would like to access the LUN:

```

# echo 0x4020400100000000 >
/sys/bus/ccw/drivers/zfcp/0.0.fc00/0x50050763050b073d/unit_add

```

5. Find out the assigned SCSI device name:

```

# lszfcp -DV
/sys/devices/css0/0.0.0015/0.0.fc00/0x50050763050b073d/0x40204001000
00000
/sys/bus/ccw/drivers/zfcp/0.0.fc00/host0/rport-0:0-
21/target0:0:21/0:0:21:1089355792

```

15.2.2. Persistently activating FCP LUNs

The above instructions described how to activate FCP LUNs dynamically in a running system. However, such changes are not persistent and do not survive a reboot. How you make the changes to the FCP configuration persistent in your Linux system depends on whether the FCP LUNs belong to the root file system. Those required for the root file system need to be activated very early during the boot process by the **initramfs** to be able to mount the root file system. The **cio_ignore** commands are handled transparently for persistent device configurations and you do not need to free devices from the ignore list manually.

15.2.2.1. FCP LUNs That Are Part of the Root File System

The only file you have to modify for adding FCP LUNs that are part of the root file system is **/etc/zipl.conf** followed by a run of the **zipl** boot loader tool. There is no more need to recreate the **initramfs**.

Red Hat Enterprise Linux provides a parameter to activate FCP LUNs early in the boot process: **rd.zfcp=**. The value is a comma-separated list containing the device bus ID, the WWPN as 16 digit hexadecimal number prefixed with **0x**, and the FCP LUN prefixed with **0x** and padded with zeroes to the right to have 16 hexadecimal digits.

The following example **zipl.conf** is for a system that uses physical volumes on partitions of two FCP LUNs for an LVM volume group **vg_devel1** that contains a logical volume **lv_root** for the root file system. For simplicity, the example shows a configuration without multipathing.

```
[defaultboot]
default=linux
target=/boot/

[linux]
image=/boot/vmlinuz-2.6.32-19.el7.s390x
ramdisk=/boot/initramfs-2.6.32-19.el7.s390x.img
parameters="root=/dev/mapper/vg_devel1-lv_root
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a000000000
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a100000000
rd_LVM_LV=vg_devel1/lv_root rd_NO_LUKS rd_NO_MD rd_NO_DM LANG=en_US.UTF-8
SYSEFONT=latarcyrheb-sun16 KEYTABLE=us cio_ignore=all,!condev"
```

To add another physical volume on a partition of a third FCP LUN with device bus ID 0.0.fc00, WWPN 0x5105074308c212e9 and FCP LUN 0x401040a300000000, add **rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a300000000** to the parameters line of your boot kernel in **zipl.conf**. For example:

```
[defaultboot]
default=linux
target=/boot/

[linux]
image=/boot/vmlinuz-2.6.32-19.el7.s390x
ramdisk=/boot/initramfs-2.6.32-19.el7.s390x.img
parameters="root=/dev/mapper/vg_devel1-lv_root
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a000000000
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a100000000
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a300000000
rd_LVM_LV=vg_devel1/lv_root rd_NO_LUKS rd_NO_MD rd_NO_DM LANG=en_US.UTF-8
SYSEFONT=latarcyrheb-sun16 KEYTABLE=us cio_ignore=all,!condev"
```



WARNING

Make sure the length of the kernel command line in **/etc/zipl.conf** does not exceed 896 bytes. Otherwise, the boot loader cannot be saved, and the installation fails.

Run **zipl** to apply the changes of **/etc/zipl.conf** for the next IPL:

```
# zipl -V
Using config file '/etc/zipl.conf'
Target device information
Device.....: 08:00
Partition.....: 08:01
Device name.....: sda
Device driver name.....: sd
Type.....: disk partition
Disk layout.....: SCSI disk layout
Geometry - start.....: 2048
File system block size.....: 4096
Physical block size.....: 512
Device size in physical blocks..: 10074112
Building bootmap in '/boot/'
Building menu 'rh-automatic-menu'
Adding #1: IPL section 'linux' (default)
kernel image.....: /boot/vmlinuz-2.6.32-19.el7.s390x
kernel parmline...: 'root=/dev/mapper/vg_devel1-lv_root
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a000000000
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a100000000
rd.zfcp=0.0.fc00,0x5105074308c212e9,0x401040a300000000
rd_LVM_LV=vg_devel1/lv_root rd_NO_LUKS rd_NO_MD rd_NO_DM LANG=en_US.UTF-8
SYSEFONT=latarcyrheb-sun16 KEYTABLE=us cio_ignore=all,!condev'
initial ramdisk...: /boot/initramfs-2.6.32-19.el7.s390x.img
component address:
kernel image.....: 0x00010000-0x007a21ff
parmline.....: 0x00001000-0x000011ff
initial ramdisk..: 0x02000000-0x028f63ff
internal loader..: 0x0000a000-0x0000a3ff
Preparing boot device: sda.
Detected SCSI PCBIOS disk layout.
Writing SCSI master boot record.
Syncing disks...
Done.
```

15.2.2.2. FCP LUNs That Are Not Part of the Root File System

FCP LUNs that are not part of the root file system, such as data disks, are persistently configured in the file **/etc/zfcp.conf**. It contains one FCP LUN per line. Each line contains the device bus ID of the FCP adapter, the WWPN as 16 digit hexadecimal number prefixed with **0x**, and the FCP LUN prefixed with **0x** and padded with zeroes to the right to have 16 hexadecimal digits, separated by a space or tab. Entries in **/etc/zfcp.conf** are activated and configured by **udev** when an FCP adapter is added to the system. At boot time, all FCP adapters visible to the system are added and trigger **udev**.

Example content of **/etc/zfcp.conf**:

```
0.0.fc00 0x5105074308c212e9 0x401040a000000000
0.0.fc00 0x5105074308c212e9 0x401040a100000000
0.0.fc00 0x5105074308c212e9 0x401040a300000000
0.0.fcd0 0x5105074308c2aee9 0x401040a000000000
0.0.fcd0 0x5105074308c2aee9 0x401040a100000000
0.0.fcd0 0x5105074308c2aee9 0x401040a300000000
```

Modifications of **/etc/zfcp.conf** only become effective after a reboot of the system or after the

dynamic addition of a new FCP channel by changing the system's I/O configuration (for example, a channel is attached under z/VM). Alternatively, you can trigger the activation of a new entry in `/etc/zfcp.conf` for an FCP adapter which was previously not active, by executing the following commands:

1. Use the **cio_ignore** utility to remove the FCP adapter from the list of ignored devices and make it visible to Linux:

```
# cio_ignore -r device_number
```

Replace *device_number* with the device number of the FCP adapter. For example:

```
# cio_ignore -r fcfc
```

2. To trigger the uevent that activates the change, issue:

```
# echo add > /sys/bus/ccw/devices/device-bus-ID/uevent
```

For example:

```
# echo add > /sys/bus/ccw/devices/0.0.fcfc/uevent
```

15.3. ADDING A NETWORK DEVICE

Network device driver modules are loaded automatically by **udev**.

You can add a network interface on IBM Z dynamically or persistently.

- Dynamically
 - a. Load the device driver
 - b. Remove the network devices from the list of ignored devices.
 - c. Create the group device.
 - d. Configure the device.
 - e. Set the device online.
- Persistently
 - a. Create a configuration script.
 - b. Activate the interface.

The following sections provide basic information for each task of each IBM Z network device driver.

[Section 15.3.1, “Adding a qeth Device”](#) describes how to add a qeth device to an existing instance of Red Hat Enterprise Linux. [Section 15.3.2, “Adding an LCS Device”](#) describes how to add an lcs device to an existing instance of Red Hat Enterprise Linux.

15.3.1. Adding a qeth Device

The **qeth** network device driver supports IBM z OSA-Express features in QDIO mode, Hipersockets, z/VM guest LAN, and z/VM VSWITCH.

The **qeth** device driver assigns the same interface name for Ethernet and Hipersockets devices: **enccwbus_ID**. The bus ID is composed of the channel subsystem ID, subchannel set ID, and device number, for example **enccw0.0.0a00**.

15.3.1.1. Dynamically Adding a qeth Device

To add a **qeth** device dynamically, follow these steps:

1. Determine whether the **qeth** device driver modules are loaded. The following example shows loaded **qeth** modules:

```
# lsmod | grep qeth
               qeth_l3                127056  9
               qeth_l2                73008  3
               ipv6                    492872
155ip6t_REJECT,nf_conntrack_ipv6,qeth_l3
               qeth                   115808  2 qeth_l3,qeth_l2
               qdio                   68240  1 qeth
               ccwgroup               12112  2 qeth
```

If the output of the **lsmod** command shows that the **qeth** modules are not loaded, run the **modprobe** command to load them:

```
# modprobe qeth
```

2. Use the **cio_ignore** utility to remove the network channels from the list of ignored devices and make them visible to Linux:

```
# cio_ignore -r
read_device_bus_id,write_device_bus_id,data_device_bus_id
```

Replace *read_device_bus_id*, *write_device_bus_id*, *data_device_bus_id* with the three device bus IDs representing a network device. For example, if the *read_device_bus_id* is **0.0.f500**, the *write_device_bus_id* is **0.0.f501**, and the *data_device_bus_id* is **0.0.f502**:

```
# cio_ignore -r 0.0.f500,0.0.f501,0.0.f502
```

3. Use the **znetconf** utility to sense and list candidate configurations for network devices:

```
# znetconf -u
Scanning for network devices...
Device IDs                Type      Card Type      CHPID Drv.
-----
0.0.f500,0.0.f501,0.0.f502 1731/01  OSA (QDIO)      00 qeth
0.0.f503,0.0.f504,0.0.f505 1731/01  OSA (QDIO)      01 qeth
0.0.0400,0.0.0401,0.0.0402 1731/05  Hipersockets    02 qeth
```

4. Select the configuration you want to work with and use **znetconf** to apply the configuration and to bring the configured group device online as network device.

```
# znetconf -a f500
Scanning for network devices...
Successfully configured device 0.0.f500 (enccw0.0.f500)
```

5. Optionally, you can also pass arguments that are configured on the group device before it is set online:

```
# znetconf -a f500 -o portname=myname
Scanning for network devices...
Successfully configured device 0.0.f500 (enccw0.0.f500)
```

Now you can continue to configure the **enccw0.0.f500** network interface.

Alternatively, you can use **sysfs** attributes to set the device online as follows:

1. Create a **qeth** group device:

```
# echo read_device_bus_id,write_device_bus_id,data_device_bus_id >
/sys/bus/ccwgroup/drivers/qeth/group
```

For example:

```
# echo 0.0.f500,0.0.f501,0.0.f502 >
/sys/bus/ccwgroup/drivers/qeth/group
```

2. Next, verify that the **qeth** group device was created properly by looking for the read channel:

```
# ls /sys/bus/ccwgroup/drivers/qeth/0.0.f500
```

You can optionally set additional parameters and features, depending on the way you are setting up your system and the features you require, such as:

- **portno**
- **layer2**
- **portname**

3. Bring the device online by writing **1** to the online **sysfs** attribute:

```
# echo 1 > /sys/bus/ccwgroup/drivers/qeth/0.0.f500/online
```

4. Then verify the state of the device:

```
# cat /sys/bus/ccwgroup/drivers/qeth/0.0.f500/online
1
```

A return value of **1** indicates that the device is online, while a return value **0** indicates that the device is offline.

5. Find the interface name that was assigned to the device:

```
# cat /sys/bus/ccwgroup/drivers/qeth/0.0.f500/if_name
enccw0.0.f500
```

Now you can continue to configure the **enccw0.0.f500** network interface.

The following command from the **s390utils** package shows the most important settings of your **qeth** device:

```
# lsqeth enccw0.0.f500
Device name                : enccw0.0.f500
-----
card_type                  : OSD_1000
cdev0                     : 0.0.f500
cdev1                     : 0.0.f501
cdev2                     : 0.0.f502
chpid                     : 76
online                    : 1
portname                  : OSAPORT
portno                    : 0
state                     : UP (LAN ONLINE)
priority_queueing         : always queue 0
buffer_count              : 16
layer2                    : 1
isolation                  : none
```

15.3.1.2. Dynamically Removing a qeth Device

To remove a **qeth** device, use the **znetconf** utility. For example:

1. Use the **znetconf** utility to show you all configured network devices:

```
# znetconf -c
Device IDs                Type      Card Type      CHPID Drv. Name
State
-----
0.0.8036,0.0.8037,0.0.8038 1731/05 HiperSockets    FB qeth hsi1
online
0.0.f5f0,0.0.f5f1,0.0.f5f2 1731/01 OSD_1000      76 qeth
enccw0.0.09a0             online
0.0.f500,0.0.f501,0.0.f502 1731/01 GuestLAN QDIO 00 qeth
enccw0.0.f500             online
```

2. Select the network device to be removed and run **znetconf** to set the device offline and ungroup the **ccw>** group device.

```
# znetconf -r f500
Remove network device 0.0.f500 (0.0.f500,0.0.f501,0.0.f502)?
Warning: this may affect network connectivity!
Do you want to continue (y/n)?y
Successfully removed device 0.0.f500 (enccw0.0.f500)
```

3. Verify the success of the removal:

■

```
# znetconf -c
Device IDs                Type      Card Type      CHPID Drv. Name
State
-----
-----
0.0.8036,0.0.8037,0.0.8038 1731/05 HiperSockets      FB qeth hsi1
online
0.0.f5f0,0.0.f5f1,0.0.f5f2 1731/01 OSD_1000      76 qeth
enccw0.0.09a0      online
```

15.3.1.3. Persistently Adding a qeth Device

To make your new **qeth** device persistent, you need to create the configuration file for your new interface. The network interface configuration files are placed in the **/etc/sysconfig/network-scripts/** directory.

The network configuration files use the naming convention **ifcfg-*device***, where *device* is the value found in the **if_name** file in the **qeth** group device that was created earlier, for example **enccw0.0.09a0**. The **cio_ignore** commands are handled transparently for persistent device configurations and you do not need to free devices from the ignore list manually.

If a configuration file for another device of the same type already exists, the simplest way to add the config file is to copy it to the new name and then edit it:

```
# cd /etc/sysconfig/network-scripts
# cp ifcfg-enccw0.0.09a0 ifcfg-enccw0.0.0600
```

To learn IDs of your network devices, use the **lsqeth** utility:

```
# lsqeth -p
devices                CHPID interface      cardtype      port
chksum prio-q'ing rtr4 rtr6 lay'2 cnt
-----
-----
0.0.09a0/0.0.09a1/0.0.09a2 x00  enccw0.0.09a0      Virt.NIC QDIO  0      sw
always_q_2 n/a  n/a  1      64
0.0.0600/0.0.0601/0.0.0602 x00  enccw0.0.0600      Virt.NIC QDIO  0      sw
always_q_2 n/a  n/a  1      64
```

If you do not have a similar device defined, you must create a new file. Use this example of **/etc/sysconfig/network-scripts/ifcfg-0.0.09a0** as a template:

```
# IBM QETH
DEVICE=enccw0.0.09a0
BOOTPROTO=static
IPADDR=10.12.20.136
NETMASK=255.255.255.0
ONBOOT=yes
NETTYPE=qeth
SUBCHANNELS=0.0.09a0,0.0.09a1,0.0.09a2
PORTNAME=OSAPORT
OPTIONS='layer2=1 portno=0'
MACADDR=02:00:00:23:65:1a
TYPE=Ethernet
```


Edit the new **ifcfg-0.0.0600** file as follows:

1. Modify the **DEVICE** statement to reflect the contents of the **if_name** file from your **ccw** group.
2. Modify the **IPADDR** statement to reflect the IP address of your new interface.
3. Modify the **NETMASK** statement as needed.
4. If the new interface is to be activated at boot time, then make sure **ONBOOT** is set to **yes**.
5. Make sure the **SUBCHANNELS** statement matches the hardware addresses for your qeth device.
6. Modify the **PORTNAME** statement or leave it out if it is not necessary in your environment.
7. You can add any valid **sysfs** attribute and its value to the **OPTIONS** parameter. The Red Hat Enterprise Linux installation program currently uses this to configure the layer mode (**layer2**) and the relative port number (**portno**) of **qeth** devices.
The **qeth** device driver default for OSA devices is now layer 2 mode. To continue using old **ifcfg** definitions that rely on the previous default of layer 3 mode, add **layer2=0** to the **OPTIONS** parameter.

/etc/sysconfig/network-scripts/ifcfg-0.0.0600

```
# IBM QETH
DEVICE=enccw0.0.0600
BOOTPROTO=static
IPADDR=192.168.70.87
NETMASK=255.255.255.0
ONBOOT=yes
NETTYPE=qeth
SUBCHANNELS=0.0.0600,0.0.0601,0.0.0602
PORTNAME=OSAPORT
OPTIONS='layer2=1 portno=0'
MACADDR=02:00:00:b3:84:ef
TYPE=Ethernet
```

Changes to an **ifcfg** file only become effective after rebooting the system or after the dynamic addition of new network device channels by changing the system's I/O configuration (for example, attaching under z/VM). Alternatively, you can trigger the activation of a **ifcfg** file for network channels which were previously not active yet, by executing the following commands:

1. Use the **cio_ignore** utility to remove the network channels from the list of ignored devices and make them visible to Linux:

```
# cio_ignore -r
read_device_bus_id,write_device_bus_id,data_device_bus_id
```

Replace *read_device_bus_id*, *write_device_bus_id*, *data_device_bus_id* with the three device bus IDs representing a network device. For example, if the *read_device_bus_id* is **0.0.0600**, the *write_device_bus_id* is **0.0.0601**, and the *data_device_bus_id* is **0.0.0602**:

```
# cio_ignore -r 0.0.0600,0.0.0601,0.0.0602
```

2. To trigger the uevent that activates the change, issue:

```
# echo add > /sys/bus/ccw/devices/read-channel/uevent
```

For example:

```
# echo add > /sys/bus/ccw/devices/0.0.0600/uevent
```

3. Check the status of the network device:

```
# lsqeth
```

4. Now start the new interface:

```
# ifup enccw0.0.0600
```

5. Check the status of the interface:

```
# ip addr show enccw0.0.0600
3: enccw0.0.0600: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP group default qlen 1000
link/ether 3c:97:0e:51:38:17 brd ff:ff:ff:ff:ff:ff
inet 10.85.1.245/24 brd 10.34.3.255 scope global dynamic
enccw0.0.0600
valid_lft 81487sec preferred_lft 81487sec
inet6 1574:12:5:1185:3e97:eff:fe51:3817/64 scope global
noprefixroute dynamic
valid_lft 2591994sec preferred_lft 604794sec
inet6 fe45::a455:eff:d078:3847/64 scope link
valid_lft forever preferred_lft forever
```

6. Check the routing for the new interface:

```
# ip route
default via 10.85.1.245 dev enccw0.0.0600 proto static metric 1024
12.34.4.95/24 dev enp0s25 proto kernel scope link src 12.34.4.201
12.38.4.128 via 12.38.19.254 dev enp0s25 proto dhcp metric 1
192.168.122.0/24 dev virbr0 proto kernel scope link src
192.168.122.1
```

7. Verify your changes by using the **ping** utility to ping the gateway or another host on the subnet of the new device:

```
# ping -c 1 192.168.70.8
PING 192.168.70.8 (192.168.70.8) 56(84) bytes of data.
64 bytes from 192.168.70.8: icmp_seq=0 ttl=63 time=8.07 ms
```

8. If the default route information has changed, you must also update **/etc/sysconfig/network** accordingly.

15.3.2. Adding an LCS Device

The *LAN channel station* (LCS) device driver supports 1000Base-T Ethernet on the OSA-Express2 and OSA-Express 3 features.

The **LCS** device driver assigns the following interface name for OSA-Express Fast Ethernet and Gigabit Ethernet devices: **enccwbus_ID**. The bus ID is composed of the channel subsystem ID, subchannel set ID, and device number, for example **enccw0.0.0a00**.

15.3.2.1. Dynamically Adding an LCS Device

1. Load the device driver:

```
# modprobe lcs
```

2. Use the **cio_ignore** utility to remove the network channels from the list of ignored devices and make them visible to Linux:

```
# cio_ignore -r read_device_bus_id,write_device_bus_id
```

Replace *read_device_bus_id* and *write_device_bus_id* with the two device bus IDs representing a network device. For example:

```
# cio_ignore -r 0.0.09a0,0.0.09a1
```

3. Create the group device:

```
# echo read_device_bus_id,write_device_bus_id >
/sys/bus/ccwgroup/drivers/lcs/group
```

4. Configure the device. OSA cards can provide up to 16 ports for a single CHPID. By default, the LCS group device uses port **0**. To use a different port, issue a command similar to the following:

```
# echo portno > /sys/bus/ccwgroup/drivers/lcs/device_bus_id/portno
```

Replace *portno* with the port number you want to use.

5. Set the device online:

```
# echo 1 > /sys/bus/ccwgroup/drivers/lcs/read_device_bus_id/online
```

6. To find out what network device name has been assigned, enter the command:

```
# ls -l /sys/bus/ccwgroup/drivers/lcs/read_device_bus_ID/net/
drwxr-xr-x 4 root root 0 2010-04-22 16:54 enccw0.0.0600
```

15.3.2.2. Persistently Adding an LCS Device

The **cio_ignore** commands are handled transparently for persistent device configurations and you do not need to free devices from the ignore list manually.

To add an LCS device persistently, follow these steps:

1. Create a configuration script as file in **/etc/sysconfig/network-scripts/** with a name like **ifcfg-device**, where *device* is the value found in the **if_name** file in the **qeth** group device that was created earlier, for example **enccw0.0.09a0**. The file should look similar to the following:

```
# IBM LCS
DEVICE=enccw0.0.09a0
BOOTPROTO=static
IPADDR=10.12.20.136
NETMASK=255.255.255.0
ONBOOT=yes
NETTYPE=lcs
SUBCHANNELS=0.0.09a0,0.0.09a1
PORTNAME=0
OPTIONS=''
TYPE=Ethernet
```

2. Modify the value of **PORTNAME** to reflect the LCS port number (**portno**) you would like to use. You can add any valid lcs sysfs attribute and its value to the optional **OPTIONS** parameter. See [Section 15.3.1.3, “Persistently Adding a qeth Device”](#) for the syntax.
3. Set the **DEVICE** parameter as follows:

```
DEVICE=enccwbus_ID
```

4. Issue an **ifup** command to activate the device:

```
# ifup enccwbus_ID
```

Changes to an **ifcfg** file only become effective after rebooting the system. You can trigger the activation of a **ifcfg** file for network channels by executing the following commands:

1. Use the **cio_ignore** utility to remove the LCS device adapter from the list of ignored devices and make it visible to Linux:

```
# cio_ignore -r read_device_bus_id,write_device_bus_id
```

Replace *read_device_bus_id* and *write_device_bus_id* with the device bus IDs of the LCS device. For example:

```
# cio_ignore -r 0.0.09a0,0.0.09a1
```

2. To trigger the uevent that activates the change, issue:

```
# echo add > /sys/bus/ccw/devices/read-channel/uevent
```

For example:

```
# echo add > /sys/bus/ccw/devices/0.0.09a0/uevent
```

15.3.3. Configuring an IBM z Network Device for Network Root File System

To add a network device that is required to access the root file system, you only have to change the boot options. The boot options can be in a parameter file or part of a **zip1.conf** on a DASD or FCP-attached SCSI LUN prepared with the **zip1** boot loader. There is no need to recreate the initramfs.

Dracut, the **mkinitrd** successor that provides the functionality in the **initramfs** that in turn replaces **initrd**, provides a boot parameter to activate network devices on IBM z early in the boot process: **rd.znet=**.

As input, this parameter takes a comma-separated list of the **NETTYPE** (qeth, lcs, etc), two (lcs, etc) or three (qeth) device bus IDs, and optional additional parameters consisting of key-value pairs corresponding to network device sysfs attributes. This parameter configures and activates the IBM z network hardware. The configuration of IP addresses and other network specifics works the same as for other platforms. See the **dracut** documentation for more details.

The **cio_ignore** commands for the network channels are handled transparently on boot.

Example boot options for a root file system accessed over the network through NFS:

```
root=10.16.105.196:/nfs/nfs_root cio_ignore=all,!condev
rd.znet=qeth,0.0.0a00,0.0.0a01,0.0.0a02,layer2=1,portno=0,portname=OSAPORT
ip=10.16.105.197:10.16.105.196:10.16.111.254:255.255.248.0:nfs-server.subd
omain.domain:enccw0.0.09a0:none rd_NO_LUKS rd_NO_LVM rd_NO_MD rd_NO_DM
LANG=en_US.UTF-8 SYSFONT=latarcyrheb-sun16 KEYTABLE=us
```

CHAPTER 16. PARAMETER AND CONFIGURATION FILES ON IBM Z

The IBM Z architecture can use a customized parameter file to pass boot parameters to the kernel and the installation program. This section describes the contents of this parameter file.

You need only read this section if you intend to change the shipped parameter file. You need to change the parameter file if you want to:

- Install unattended with Kickstart.
- Choose non-default installation settings that are not accessible through the installation program's interactive user interface, such as rescue mode.

The parameter file can be used to set up networking non-interactively before the installation program (loader and **Anaconda**) starts.

The kernel parameter file is limited to 895 characters plus an end-of-line character. The parameter file can be variable or fixed record format. Fixed record format increases the file size by padding each line up to the record length. Should you encounter problems with the installation program not recognizing all specified parameters in LPAR environments, you can try to put all parameters in one single line or start and end each line with a space character.

The parameter file contains kernel parameters, such as **ro**, and parameters for the installation process, such as **vncpassword=test** or **vnc**.

16.1. REQUIRED PARAMETERS

The following parameters are required and must be included in the parameter file. They are also provided in the file **generic.prm** in directory **images/** of the installation DVD:

ro

mounts the root file system, which is a RAM disk, read-only.

ramdisk_size=size

modifies the memory size reserved for the RAM disk to ensure that the Red Hat Enterprise Linux installation program fits within it. For example: **ramdisk_size=40000**.

The **generic.prm** file also contains the additional parameter **cio_ignore=all, !condev**. This setting speeds up boot and device detection on systems with many devices. The installation program transparently handles the activation of ignored devices.



IMPORTANT

To avoid installation problems arising from **cio_ignore** support not being implemented throughout the entire stack, adapt the **cio_ignore=** parameter value to your system or remove the parameter entirely from your parameter file used for booting (IPL) the installation program.

16.2. THE Z/VM CONFIGURATION FILE

This applies only if installing under z/VM. Under z/VM, you can use a configuration file on a CMS-formatted disk. The purpose of the CMS configuration file is to save space in the parameter file by moving the parameters that configure the initial network setup, the DASD, and the FCP specification out

of the parameter file (see [Section 16.3, “Installation Network Parameters”](#)).

Each line of the CMS configuration file contains a single variable and its associated value, in the following shell-style syntax: ***variable=value*** .

You must also add the **CMSDASD** and **CMSCONFFILE** parameters to the parameter file. These parameters point the installation program to the configuration file:

CMSDASD=*cmsdasd_address*

Where *cmsdasd_address* is the device number of a CMS-formatted disk that contains the configuration file. This is usually the CMS user's **A** disk.

For example: **CMSDASD=191**

CMSCONFFILE=*configuration_file*

Where *configuration_file* is the name of the configuration file. This value must be specified in lower case. It is specified in a Linux file name format: ***CMS_file_name.CMS_file_type***.

The CMS file **REDHAT CONF** is specified as **redhat.conf**. The CMS file name and the file type can each be from one to eight characters that follow the CMS conventions.

For example: **CMSCONFFILE=redhat.conf**

16.3. INSTALLATION NETWORK PARAMETERS

The following parameters can be used to set up the preliminary network automatically and can be defined in the CMS configuration file. The parameters in this section are the only parameters that can also be used in a CMS configuration file. All other parameters in other sections must be specified in the parameter file.

NETTYPE="type"

Where *type* must be one of the following: **qeth**, **lcs**, or **ctc**. The default is **qeth**.

Choose **lcs** for:

- OSA-2 Ethernet/Token Ring
 - OSA-Express Fast Ethernet in non-QDIO mode
 - OSA-Express High Speed Token Ring in non-QDIO mode
 - Gigabit Ethernet in non-QDIO mode
- Choose **qeth** for:
- OSA-Express Fast Ethernet
 - Gigabit Ethernet (including 1000Base-T)
 - High Speed Token Ring
 - HiperSockets
 - ATM (running Ethernet LAN emulation)

SUBCHANNELS="device_bus_IDs"

Where *device_bus_IDs* is a comma-separated list of two or three device bus IDs. The IDs must be specified in lowercase.

Provides required device bus IDs for the various network interfaces:

```
qeth:
SUBCHANNELS="read_device_bus_id,write_device_bus_id,data_device_bus_id"
lcs or ctc: SUBCHANNELS="read_device_bus_id,write_device_bus_id"
```

For example (a sample qeth SUBCHANNEL statement):

```
SUBCHANNELS="0.0.f5f0,0.0.f5f1,0.0.f5f2"
```

PORTNAME="osa_portname"PORTNAME="lcs_portnumber"

This variable supports OSA devices operating in qdio mode or in non-qdio mode.

When using qdio mode (**NETTYPE="qeth"**), *osa_portname* is the portname specified on the OSA device when operating in qeth mode.

When using non-qdio mode (**NETTYPE="lcs"**), *lcs_portnumber* is used to pass the relative port number as a decimal integer in the range of 0 through 15.

PORTNO="portnumber"

You can add either **PORTNO="0"** (to use port 0) or **PORTNO="1"** (to use port 1 of OSA features with two ports per CHPID) to the CMS configuration file to avoid being prompted for the mode.

LAYER2="value"

Where *value* can be **0** or **1**.

Use **LAYER2="0"** to operate an OSA or HiperSockets device in layer 3 mode (**NETTYPE="qeth"**).

Use **LAYER2="1"** for layer 2 mode. For virtual network devices under z/VM this setting must match the definition of the GuestLAN or VSWITCH to which the device is coupled.

To use network services that operate on layer 2 (the Data Link Layer or its MAC sublayer) such as DHCP, layer 2 mode is a good choice.

The qeth device driver default for OSA devices is now layer 2 mode. To continue using the previous default of layer 3 mode, set **LAYER2="0"** explicitly.

VSWITCH="value"

Where *value* can be **0** or **1**.

Specify **VSWITCH="1"** when connecting to a z/VM VSWITCH or GuestLAN, or **VSWITCH="0"** (or nothing at all) when using directly attached real OSA or directly attached real HiperSockets.

MACADDR="MAC_address"

If you specify **LAYER2="1"** and **VSWITCH="0"**, you can optionally use this parameter to specify a MAC address. Linux requires six colon-separated octets as pairs lower case hex digits - for example, **MACADDR=62:a3:18:e7:bc:5f**. Note that this is different from the notation used by z/VM.

If you specify **LAYER2="1"** and **VSWITCH="1"**, you must not specify the **MACADDR**, because z/VM assigns a unique MAC address to virtual network devices in layer 2 mode.

CTCPR0T="value"

Where *value* can be **0**, **1**, or **3**.

Specifies the CTC protocol for **NETTYPE="ctc"**. The default is **0**.

HOSTNAME="string"

Where *string* is the host name of the newly-installed Linux instance.

IPADDR="IP"

Where *IP* is the IP address of the new Linux instance.

NETMASK="netmask"

Where *netmask* is the netmask.

The netmask supports the syntax of a prefix integer (from 1 to 32) as specified in IPv4 *classless interdomain routing* (CIDR). For example, you can specify **24** instead of **255 . 255 . 255 . 0**, or **20** instead of **255 . 255 . 240 . 0**.

GATEWAY="gw"

Where *gw* is the gateway IP address for this network device.

MTU="mtu"

Where *mtu* is the *Maximum Transmission Unit* (MTU) for this network device.

DNS="server1:server2:additional_server_terms:serverN"

Where "server1:server2:additional_server_terms:serverN" is a list of DNS servers, separated by colons. For example:

```
DNS="10.1.2.3:10.3.2.1"
```

SEARCHDNS="domain1:domain2:additional_dns_terms:domainN"

Where "domain1:domain2:additional_dns_terms:domainN" is a list of the search domains, separated by colons. For example:

```
SEARCHDNS="subdomain.domain:domain"
```

You only need to specify **SEARCHDNS=** if you specify the **DNS=** parameter.

DASD=

Defines the DASD or range of DASDs to configure for the installation.

The installation program supports a comma-separated list of device bus IDs, or ranges of device bus IDs with the optional attributes **ro**, **diag**, **erplog**, and **failfast**. Optionally, you can abbreviate device bus IDs to device numbers with leading zeros stripped. Any optional attributes should be separated by colons and enclosed in parentheses. Optional attributes follow a device bus ID or a range of device bus IDs.

The only supported global option is **autodetect**. This does not support the specification of non-existent DASDs to reserve kernel device names for later addition of DASDs. Use persistent DASD device names (for example **/dev/disk/by-path/...**) to enable transparent addition of disks later. Other global options such as **probeonly**, **nopav**, or **nofcx** are not supported by the installation program.

Only specify those DASDs that you really need to install your system. All unformatted DASDs specified here must be formatted after a confirmation later on in the installation program. Add any data DASDs that are not needed for the root file system or the **/boot** partition after installation as described in [Section 15.1.3.2, "DASDs That Are Not Part of the Root File System"](#).

For example:

```
DASD="eb1c,0.0.a000-0.0.a003,eb10-eb14(diag),0.0.ab1c(ro:diag)"
```

For FCP-only environments, remove the **DASD=** option from the CMS configuration file to indicate no DASD is present.

FCP_n="device_bus_ID WWPN FCP_LUN"

Where:

- *n* is typically an integer value (for example **FCP_1** or **FCP_2**) but could be any string with alphabetic or numeric characters or underscores.
- *device_bus_ID* specifies the device bus ID of the FCP device representing the *host bus adapter* (HBA) (for example **0.0.fc00** for device fc00).
- *WWPN* is the world wide port name used for routing (often in conjunction with multipathing) and is as a 16-digit hex value (for example **0x50050763050b073d**).
- *FCP_LUN* refers to the storage logical unit identifier and is specified as a 16-digit hexadecimal value padded with zeroes to the right (for example **0x4020400100000000**). These variables can be used on systems with FCP devices to activate FCP LUNs such as SCSI disks. Additional FCP LUNs can be activated during the installation interactively or by means of a Kickstart file. An example value looks similar to the following:

```
FCP_1="0.0.fc00 0x50050763050b073d 0x4020400100000000"
```



IMPORTANT

Each of the values used in the FCP parameters (for example **FCP_1** or **FCP_2**) are site-specific and are normally supplied by the FCP storage administrator.

The installation program prompts you for any required parameters not specified in the parameter or configuration file except for FCP_n.

16.4. PARAMETERS FOR KICKSTART INSTALLATIONS

The following parameters can be defined in a parameter file but do not work in a CMS configuration file.

inst.ks=URL

References a Kickstart file, which usually resides on the network for Linux installations on System z. Replace *URL* with the full path including the file name of the Kickstart file. This parameter activates automatic installation with Kickstart.

RUNKS=value



IMPORTANT

This parameter is deprecated. If you use it in a Kickstart file, it will be ignored. Only the **inst.ks=** parameter is necessary to start a Kickstart installation on IBM Z.

Where *value* is defined as *1* if you want to run the loader automatically on the Linux console without

having to log in over the network with SSH. To use **RUNKS=1**, the console must either support full-screen or the **inst.cmdline** option (below) should be used. The latter applies for the 3270 terminal under z/VM or the operating system messages console for LPAR. We recommend **RUNKS=1** for fully automatic installations with Kickstart. When **RUNKS=1** is set, the installation program automatically continues in case of parameter errors and does not interrupt unattended installations by prompting for user interaction.

Leave out the parameter or specify **RUNKS=0** otherwise.

inst.cmdline

When this option is specified, output on line-mode terminals (such as 3270 under z/VM or operating system messages for LPAR) becomes readable, as the installation program disables escape terminal sequences that are only applicable to UNIX-like consoles. This requires installation with a Kickstart file that answers all questions, because the installation program does not support interactive user input in cmdline mode.

Ensure that your Kickstart file contains all required parameters before you use the **inst.cmdline** option. If a required command is missing, the installation will fail.

16.5. MISCELLANEOUS PARAMETERS

The following parameters can be defined in a parameter file but do not work in a CMS configuration file.

rd.live.check

Turns on testing of an ISO-based installation source; for example, when booted from an FCP-attached DVD or using **inst.repo=** with an ISO on local hard disk or mounted with NFS.

nompah

Disables support for multipath devices.

proxy=[protocol://][username[:password]@]host[:port]

Specify a proxy to use with installation over HTTP, HTTPS, or FTP.

inst.rescue

Boot into a rescue system running from a RAM disk that can be used to fix and restore an installed system.

inst.stage2=URL

Specifies a path to an **install.img** file instead of to an installation source. Otherwise, follows the same syntax as **inst.repo=**. If **inst.stage2** is specified, it typically takes precedence over other methods of finding **install.img**. However, if **Anaconda** finds **install.img** on local media, the **inst.stage2** URL will be ignored.

If **inst.stage2** is not specified and **install.img** cannot be found locally, **Anaconda** looks to the location given by **inst.repo=** or **method=**.

If only **inst.stage2=** is given without **inst.repo=** or **method=**, **Anaconda** uses whatever repos the installed system would have enabled by default for installation.

Use the option multiple times to specify multiple HTTP, HTTPS or FTP sources. The HTTP, HTTPS or FTP paths are then tried sequentially until one succeeds:

```
inst.stage2=host1/install.img inst.stage2=host2/install.img
inst.stage3=host3/install.img
```

inst.syslog=IP/hostname[:port]

Sends log messages to a remote syslog server.

The boot parameters described here are the most useful for installations and trouble shooting on System z, but only a subset of those that influence the installation program.

16.6. SAMPLE PARAMETER FILE AND CMS CONFIGURATION FILE

To change the parameter file, begin by extending the shipped **generic.prm** file.

Example of **generic.prm** file:

```
ro ramdisk_size=40000 cio_ignore=all,!condev
CMSDASD="191" CMSCONFFILE="redhat.conf"
vnc
inst.repo=http://example.com/path/to/repository
```

Example of **redhat.conf** file configuring a QETH network device (pointed to by **CMSCONFFILE** in **generic.prm**):

```
NETTYPE="qeth"
SUBCHANNELS="0.0.0600,0.0.0601,0.0.0602"
PORTNAME="FOOBAR"
PORTNO="0"
LAYER2="1"
MACADDR="02:00:be:3a:01:f3"
HOSTNAME="foobar.systemz.example.com"
IPADDR="192.168.17.115"
NETMASK="255.255.255.0"
GATEWAY="192.168.17.254"
DNS="192.168.17.1"
SEARCHDNS="systemz.example.com:example.com"
DASD="200-203"
```

CHAPTER 17. IBM Z REFERENCES

17.1. IBM Z PUBLICATIONS

Current versions of the Linux on System z publications can be found at http://www.ibm.com/developerworks/linux/linux390/documentation_red_hat.html. They include:

- Linux on System z - How to use FC-attached SCSI devices with Linux on System z9 and zSeries, IBM, 2008, SC33-8413
- Linux on System z - How to Improve Performance with PAV, IBM, 2008, SC33-8414
- z/VM - Getting Started with Linux on System z, IBM, 2009, SC24-6194

17.2. IBM REDBOOKS PUBLICATIONS FOR SYSTEM Z

Current versions of IBM Redbooks publications can be found at <http://www.redbooks.ibm.com/>. They include:

Introductory publications

- Introduction to the New Mainframe: z/VM Basics, IBM Redbooks, 2007, SG24-7316
- Practical Migration to Linux on System z, IBM Redbooks, 2009, SG24-7727

Performance and high availability

- Linux on IBM System z: Performance Measurement and Tuning, IBM Redbooks, 2011, SG24-6926
- Achieving High Availability on Linux for System z with Linux-HA Release 2, IBM Redbooks, 2009, SG24-7711

Security

- Security for Linux on System z, IBM Redbooks, 2013, SG24-7728

Networking

- IBM System z Connectivity Handbook, IBM Redbooks, 2013, SG24-5444
- OSA Express Implementation Guide, IBM Redbooks, 2009, SG24-5948
- HiperSockets Implementation Guide, IBM Redbooks, 2007, SG24-6816
- Fibre Channel Protocol for Linux and z/VM on IBM System z, IBM Redbooks, 2007, SG24-7266

17.3. ONLINE RESOURCES

- For z/VM publications, refer to <http://www.vm.ibm.com/library/>
- For System z I/O connectivity information, refer to <http://www.ibm.com/systems/z/hardware/connectivity/index.html>

- For System z cryptographic coprocessor information, refer to <http://www.ibm.com/security/cryptocards/>
- For System z DASD storage information, refer to http://www-01.ibm.com/support/knowledgecenter/linuxonibm/com.ibm.linux.z.lgdd/lgdd_t_dasd_wrk.html