Installation

Version 24.3.1, 06 May 2024

Table of Contents

1. RBFS Installation, ZTP, and Licensing

RBFS Installation Overview

The RBFS software is available for different roles based on their functions. RBFS C-BNG, Spine, Leaf and L2BSA images are stored in the RBFS image repository from where users can download and install the images on the supported hardware platforms. When new features and software patches become available in the image store, you can upgrade the software to use them. All the latest versions of software images are stored in the RBFS image store.

The RBFS software is delivered as custom RtBrick Linux container images (also called *RBFS container images*) which can be used for virtual topologies on x86 servers or as custom RtBrick ONL installer images that can be used on the supported white box switches.

Image stores containing the container and ONL installer images are published on https://releases.rtbrick.com/ and updated when new image versions are available.

In addition to RBFS, other RtBrick software is delivered in the Debian package format that can be used on Ubuntu. Currently, the only supported Ubuntu release is 22.04 LTS Jammy. This category of software is referred to *RtBrick Tools*. The software delivered as Debian packages is composed of a set of CLI tools and/or daemons intended to facilitate working with RBFS containers and the RBFS API.

Debian package repositories containing the packages are available on https://releases.rtbrick.com/ and updated whenever a new version of the package is available.

Software Download

The RtBrick managed software download functionality enables authenticated users to download and install the RtBrick software (packages or images). Access to *image stores* and *Debian package repositories* on https://releases.rtbrick.com/ is **restricted** through the use of TLS mutual authentication with TLS client certificates (TLS client certificates can be self-signed).

To access the **restricted** *image stores* and *Debian package repositories* on https://releases.rtbrick.com/, perform the following steps:

Step 1: Generate a client certificate

RtBrick provides the rtb-apt tools to generate a client certificate. For more information, see the section The RtBrick APT tool (rtb-apt).

Step 2: Send a client certificate to RtBrick

Step 3: RtBrick approves and trusts the client's certificate

Step 4: Download the RBFS software using the rtb-apt, apt, rtb-image tools

rtb-image version 1.3.0 or later is required to correctly work with managed downloads.

The RtBrick APT tool (rtb-apt)

The rtb-apt tool is an APT (https://wiki.debian.org/Apt)
utility application that provides an easier way for managing the system configuration of RtBrick package repositories (https://wiki.debian.org/DebianRepository)
which can be used with the usual apt commands to install RtBrick software.

Some RtBrick package repositories require authentication via TLS client certificates and the rtb-apt tool provides commands for managing those specific repositories and the required apt authentication configuration.

The rtb-apt tool is a statically compiled Linux 64-bit executable file. Currently, it is verified to run on **Ubuntu 22.04**. It is available through a direct download link.

Install rtb-apt

Before you install rtb-apt, ensure that you have installed the following software:

- GNU Privacy Guard (GPG), which is used by apt to validate package repositories. To install GPG, enter the following command:

```
¬ sudo apt install gnupg
```

- HTTPS support for apt is required to access the package repositories via HTTPS.

```
¬ sudo apt install apt-transport-https ca-certificates
```

The following example shows how to download and install the rtb-apt tool. It shows the URL where the latest version of the rtb-apt tool is available for download:

```
¬ curl -o /tmp/rtb-apt https://releases.rtbrick.com/_/dl/sw/rtb-
apt/latest/linux_amd64/rtb-apt  \
    && sudo mv /tmp/rtb-apt /usr/local/bin/     \
    && sudo chown root:root /usr/local/bin/rtb-apt  \
    && sudo chmod 0755 /usr/local/bin/rtb-apt
```

The following example shows the rtb-apt tool version. The rtb-apt version 2.0.0 or later is required.

```
¬ rtb-apt --version
2.0.0
```

Generate a TLS client certificate

The following example shows how to generate a TLS client certificate using the rtb-apt tool.

```
¬ sudo rtb-apt auth generate
A new self-signed TLS client certificate has been generated for this system:

Subject:     CN=bb59a25d-6b38-4f3c-81e0-065e525c8335,OU=rtb-apt
Valid until: 2024-09-06 10:30:26 +0000 UTC

The following additional auto-generated information is included in the certificate
and can be used to uniquely identify this system:

DNS names:       [hostname.example.net]
Email addresses: [root@hostname.example.net user@hostname.example.net]
< ..................................... >

If you already have a working account on https://portal.rtbrick.com then you can
use the Self-Service section to upload this certificate. If you DO NOT yet have an
account on https://portal.rtbrick.com, send the certificate to your RtBrick
support contact:
```

```
-----BEGIN CERTIFICATE-----
MIIHHzCCBYegAwIBAgIRAJcI5pqSK9O+g6yJGB15i7YwDQYJKoZIhvcNAQELBQAw
QTEQMA4GA1UECxMHcnRiLWFwdDEtMCsGA1UEAxMkYmI1OWEyNWQtNmIzOC00ZjNj
< ..................................... >
NuLIKfmwrcyXmzAOe1bRtlJrRw0zofxX4rFcMmJReNqOV0obP5r7TCtnWtAqkFx/
7JJa
-----END CERTIFICATE-----
```

After generating the TLS Client Certificate, upload it to the **Certificates** section on https://portal.rtbrick.com.

If your domain is registered with https://portal.rtbrick.com, you will be able to log into your account. If not, reach out to your sales/partner contact to initially have your domain registered with the portal.

Identify and add RtBrick debian package repositories

Find available repositories

The following example shows how to find the available repositories:

```
¬ sudo rtb-apt repo list
Group               Repository      Distribution    Release Active  Restricted
releases/latest     rtbrick-tools   ubuntu          jammy   No      No
releases/23.8.1     rtbrick-tools   ubuntu          jammy   No      No
releases/23.9.1     rtbrick-tools   ubuntu          jammy   No      No
releases/23.10.1    rtbrick-tools   ubuntu          jammy   No      No
releases/23.11.1    rtbrick-tools   ubuntu          jammy   No      No
releases/23.12.1    rtbrick-tools   ubuntu          jammy   No      No
< ..................................... >

¬ sudo rtb-apt repo show releases/latest/rtbrick-tools
Group       releases/latest
Description Packages repositories in the releases/latest group are updated with
new package versions for every final or
            bugfix release.

    Repository      rtbrick-tools
    Description     Packages for any RtBrick software not part of RBFS and meant
to run on an standalone Linux system (usually
                    Ubuntu). A standalone Linux system will be any non-RBFS
container and non-ONL Linux system. These packages can
                    be tools like rtb-image or rtb-ansible but also software like
ctrld or apigwd.
    Distribution    ubuntu
    Release         jammy
    Active          No
    Restricted      No
```

Activate a repository

The following example shows how to activate a specific repository. rtb-apt will add the required configuration in /etc/apt/ such so that the repository can then be used with commands such as apt update and apt install:

```
¬ sudo rtb-apt repo activate releases/latest/rtbrick-tools
```

The activated repository is added to */etc/apt/sources.list.d/rtbrick.list.*:

```
¬ cat /etc/apt/sources.list.d/rtbrick.list
deb [arch=amd64 signed-by=/etc/rtbrick/RtBrick-Support.pubkey.asc]
https://releases.rtbrick.com/_/latest/ubuntu/jammy/rtbrick-tools    jammy
rtbrick-tools
```

Verify active repositories

You can verify the active repositories using the command:

```
¬ sudo rtb-apt repo list
Group               Repository      Distribution    Release Active  Restricted
releases/latest     rtbrick-tools   ubuntu          jammy   Yes     Yes  <<<<<<<<
releases/23.8.1     rtbrick-tools   ubuntu          jammy   No      No
releases/23.9.1     rtbrick-tools   ubuntu          jammy   No      No
< ..................................... >
```

Verify access (authentication) for the active package repositories

Some of the RtBrick package repositories are *restricted* meaning that they require the client application (apt in this case) to authenticate with a TLS client certificate. The TLS client certificate for the current system must be trusted by RtBrick. This is achieved either by uploading it in the Self-Service section of https://portal.rtbrick.com (if you already have a valid account on https://portal.rtbrick.com) or by sending your certificate to your RtBrick support contact.

Expected output BEFORE the TLS client certificate is trusted by RtBrick

```
¬ sudo rtb-apt auth check
Repository: releases/latest/rtbrick-tools ... restricted ... TLS client
```

```
certificate NOT accepted
```

Expected output AFTER the TLS client certificate is trusted by RtBrick

```
¬ sudo rtb-apt auth check
Repository: releases/latest/rtbrick-tools ... restricted ... TLS client
certificate accepted
```

Install a package from an RtBrick package repository

Once the TLS client certificate for the current system is trusted by RtBrick and once RtBrick package repositories have been activated with rtb-apt, the apt commands can be used to install the RtBrick software contained in those package repositories.

The following example shows the installation of the rtbrick-imgstore package which provides the rtb-image CLI tool.

```
¬ sudo apt update
Hit:1 https://releases.rtbrick.com/_/latest/ubuntu/jammy/rtbrick-tools jammy
InRelease
Hit:3 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [970 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [979
kB]
< ..................................... >

¬ sudo apt install rtbrick-imgstore
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  rtbrick-imgstore
0 upgraded, 1 newly installed, 0 to remove and 46 not upgraded.
Need to get 7,731 kB of archives.
After this operation, 26.3 MB of additional disk space will be used.
Get:1 https://releases.rtbrick.com/_/latest/ubuntu/jammy/rtbrick-tools
jammy/rtbrick-tools amd64 rtbrick-imgstore amd64 3.3.0 [7,731 kB]
Fetched 7,731 kB in 0s (41.4 MB/s)
Selecting previously unselected package rtbrick-imgstore.
< ..................................... >
```

Verify access (authentication) for image stores

The rtb-image command (CLI tool) provided by the rtbrick-imgstore package is

used to interact with *image stores*. The *image stores* are used for delivery of RBFS container images and RtBrick ONL installer images.

Similarly to package repositories some of the image stores are *restricted* meaning that they require the client application (rtb-image in this case) to authenticate with a TLS client certificate. rtb-image re-uses the TLS client certificate already generated by rtb-apt for the current system.

View available image stores

The following example shows how to view the existing image stores:

```
¬ sudo rtb-image stores list

Index   UUID                                    Name        RemoteURL
Active  Restricted
0       af73c0a6-40e7-4775-b74b-aadafeabe86d    latest
https://releases.rtbrick.com/_/images/latest    Yes     No
1       c4c896b0-52c5-4343-8a21-e2ca3ea440f1    resources
https://releases.rtbrick.com/_/resources        No      No
2                                               22.5.1
https://releases.rtbrick.com/_/images/22.5.1    No      No
3                                               22.6.1
https://releases.rtbrick.com/_/images/22.6.1    No      No
4                                               22.7.1
https://releases.rtbrick.com/_/images/22.7.1    No      No
< ..................................... >
```

Activate a restricted image store

The following example shows how to activate a (possibly restricted) image store:

```
¬ sudo rtb-image stores activate 0
```

Verify access to image stores

If the TLS client certificate for the current system is already trusted by RtBrick, you can use rtb-image to download the images.

The following example shows how to verify the access to the image stores:

```
¬ sudo rtb-image auth check
Image store: latest (af73c0a6-40e7-4775-b74b-aadafeabe86d) ... restricted ... TLS
client certificate accepted
```

The following example shows how to download an RBFS container image for the *virtual* platform and how to run a test container with it:

```
¬ sudo rtb-image update
Local image store cached copy updated to: Store:
/var/cache/rtbrick/imagestores/af73c0a6-40e7-4775-b74b-aadafeabe86d Version:
0.9.102 ValidUntil: 2023-09-23 07:03:28

¬ sudo rtb-image list -p virtual -r spine -v latest

Store: /var/cache/rtbrick/imagestores/af73c0a6-40e7-4775-b74b-aadafeabe86d
Version: 0.9.102 ValidUntil: 2023-09-23 07:03:28

UUID                                   Version Filename
Format  Role    Platform    Cached
af7108e0-95b3-4e25-91a4-a2cee7a63a38    23.6.1  rbfs-cont/rbfs-spine-virtual-
23.6.0-g8daily.202306210...    lxd     spine   virtual     false

¬ sudo rtb-image pull af7108e0-95b3-4e25-91a4-a2cee7a63a38
rbfs-spine-virtual-23.6.0-g8daily.20230621060657+Bmaster.Cf5ebfbd4.tar.zst.sha512
225 B / 225 B [============================================================] 100.00%
0s
rbfs-spine-virtual-23.6.0-g8daily.20230621060657+Bmaster.Cf5ebfbd4.tar.zst.asc 833
B / 833 B [============================================================] 100.00%
0s
rbfs-spine-virtual-23.6.0-g8daily.20230621060657+Bmaster.Cf5ebfbd4.tar.zst 435.06
MiB / 435.06 MiB [============================================================] 100.00%
4s
rbfs-spine-virtual-23.6.0-g8daily.20230621060657+Bmaster.Cf5ebfbd4.tar.zst:
decompressing 100 B / 100 B [==================================================]
100.00% 10s
af7108e0-95b3-4e25-91a4-a2cee7a63a38 downloaded as
/var/cache/rtbrick/imagestores/af73c0a6-40e7-4775-b74b-aadafeabe86d/rbfs-
cont/rbfs-spine-virtual-23.6.0-g8daily.20230621060657+Bmaster.Cf5ebfbd4

¬ sudo rtb-image run -n testRBFScont01 af7108e0-95b3-4e25-91a4-a2cee7a63a38
2023-09-08 15:15:11 UTC INF creating container testRBFScont01 with image version:
23.6.1 image uuid: af7108e0-95b3-4e25-91a4-a2cee7a63a38
2023-09-08 15:15:11 UTC INF Trying to start container container=testRBFScont01
2023-09-08 15:15:11 UTC INF Waiting for container IP addresses
container=testRBFScont01
2023-09-08 15:15:15 UTC INF Updating /etc/hosts entry address=10.0.3.96
container=testRBFScont01
2023-09-08 15:15:15 UTC INF Container was started container=testRBFScont01

¬ rssh testRBFScont01
Logging into testRBFScont01 as supervisor ...

+------------------------------------------------------------------------+
|                                                                        |
    RBFS container testRBFScont01 running on ubuntu2204host:
        Date:       Fri Sep  8 15:15:23 UTC 2023
        Uptime:     up

    Image metadata:
        UUID:       af7108e0-95b3-4e25-91a4-a2cee7a63a38
        Version:    23.6.1
        Role:       spine
        Platform:   virtual
```

```
        Format:     lxd
        Build date: 2023-06-21 06:06:57 UTC
        Based on:   Ubuntu 18.04.6 LTS
|                                                                     |
+---------------------------------------------------------------------+

supervisor@testRBFScont01>ubuntu2204host:~ $
```

RtBrick Tools and Packages

The RtBrick tools are distributed in the Debian (apt) package format in one of the rtbrick-tools debian (apt) package repositories as described in the *RtBrick Tools Installation Guide* section *1.3* step *3*.

rtbrick-toolkit

Version 24.3.1

The rtbrick-toolkit package has been updated to version 24.3.1 to match the corresponding RBFS release and has been updated to depend on the following RtBrick tools packages with these exact versions:

- rtbrick-imgstore 3.7.0
- rtbrick-apt-helper 2.0.1
- rtbrick-ansible 7.0.0
- rtbrick-apigwd 0.11.3
- rtbrick-ctrld 0.24.2
- rtbrick-lxcd 0.2.2
- rtbrick-robot-infrastructure 3.4.0
- rtbrick-fabric-ztp 2.1.0
- rtbrick-rbfs-cli-yang-models 24.3.1

rtbrick-ansible version 5.0.1 is compatible with the new RBFS RESTCONF hierarchy that was introduced with RBFS version 23.4.1. Therefore, rtbrick-ansible version 5.0.1 is backward-incompatible and will not work correctly with RBFS versions older than 23.4.1.

Understanding RBFS Release Versioning

An RBFS release can be defined as a set of software packages (currently, in the Debian package format). However, it is delivered as an image, either a container (LXC/LXD) image or as a complete ONL installation image. The ONL installation image may or may not contain a container image pre-installed in it. An image can be defined as the archived root file system of a Linux OS installation with the needed software packages pre-installed and with a default configuration. In the current context, the terms 'RBFS release' and 'image' are used interchangeably.

RtBrick uses a versioning scheme called rtbver for RBFS release versioning. An rtbver version string is syntactically similar to SEMVER 2.0, but semantically different. For the RBFS releases, the first two numbers of a version is YEAR.MONTH (corresponding to the MAJOR.MINOR of SEMVER). For example, the first RBFS release in a calendar month is 22.4.1. If a second RBFS release occurs in the same calendar month gets version as 22.4.2. The RBFS release in the next calendar month will have a version (for example) 22.5.1 irrespective of how many RBFS releases occurred in the previous calendar month.

The rtbver scheme also supports four-number versioning, such as 22.4.1.1. This four-number version is used for maintenance releases. Maintenance releases are built only when required, based on and for an already existing RBFS release (such as 22.4.1.1 for 22.4.1.).

RtBrick Tools Installation

The installation of RtBrick tools is split into several steps, as follows:

The following commands and outputs are validated only on Ubuntu 22.04.

Step 1: Removing any existing RtBrick tools Debian packages

Some of the RtBrick tools Debian packages have changed and have been upgraded several times. If any of the RtBrick tools packages are already installed, it is essential to remove any existing package.

```
apt list --installed | egrep -i rtbrick | awk -F '/' '{print $1;}' | xargs sudo apt remove -y
```

In the output, you can see the following:

The following packages are removed.

```
rtbrick-ansible rtbrick-imgstore rtbrick-lxc-tools
```

Step 2: Use rtb-apt to configure debian package repositories

Step 3: Update the local apt package cache

Update the local apt package cache using the command: sudo apt update

Step 5: Install third-party dependencies

Some RtBrick tools or packages have dependencies on third-party software such as Ansible. These third-party software packages are not delivered through the RtBrick package repositories.

Currently, the rtbrick-ansible package depends on Ansible software. For information about installing Ansible, see https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#installing-ansible-on-ubuntu.

You can install and run the *rtbrick-ansible* package only after installing Ansible. Ensure that you have installed the latest version of Ansible, before installing rtbrick-ansible.

Step 6: Install a specific RtBrick tool package

Install a particular RtBrick tool package, run the command:

sudo apt install <tool name>

For example, to install the rtbrick-ansible package, run the following command:

sudo apt install rtbrick-ansible

RtBrick Tools Packages

rtbrick-toolkit

The rtbrick-toolkit is a meta package that can be used to install all the tools required to work with RBFS images (container or ONL installer) and with the RBFS APIs. You can install all the RtBrick tools together in one go using the package. Run the following command:

sudo apt install rtbrick-toolkit

The rtbrick-toolkit meta package automatically installs the following packages:

- rtbrick-imgstore
- rtbrick-ansible
- rtbrick-apigwd
- rtbrick-ctrld
- rtbrick-robot-infrastructure

Based on your functionality requirement, you can install only the required packages individually.

rtbrick-ansible

To speed up the process of RBFS container bring-up, the rtbrick-ansible package provides the rtb-ansible tool which is an ansible-based automation solution that is used to create and maintain topologies of RBFS containers and optionally to configure the RtBrick applications in each container.

In the rtbrick-tools repository includes a rtbrick.list file under /etc/apt/sources.list.d/ with the following content:

```
deb [arch=amd64 signed-by=/etc/rtbrick/RtBrick-Support.pubkey.asc]
https://releases.rtbrick.com/_/latest/ubuntu/jammy/rtbrick-tools jammy  rtbrick-
tools
```

Add the Ansible repository using the following command:

sudo apt-add-repository ppa:ansible/ansible

Update the apt cache using the following command:

sudo apt-get update

Install the rtbrick-ansible package using the following command:

sudo apt-get install rtbrick-ansible

For information about rtb-ansible and how to use it, see the *RtBrick Automation Using Ansible* document.

rtbrick-imgstored

This package provides the rtb-image CLI utility which handles RtBrick image store. An image store (imgstore) is a versioned, checksumed, and cryptographically signed store of versioned files. The image store has been developed and optimized with the primary goal of storing and distributing Linux OS and Linux container images. However, it can be used to store any kind of file.

An image store is for images what an *apt* repository is for Debian packages. It also has some similarities with a docker registry (not to be confused with a docker repository).

The rtb-image command is used for interacting with an image store accessible via HTTP(s), making a local cache of that image store that can later be used to start LXC containers running RBFS.

```
rtbrick@access-test:~$ sudo apt search rtbrick-imgstore
Sorting... Done
Full Text Search... Done
rtbrick-imgstore/bionic,now 2.1.1-xdaily.20220531062114+Cc35c1aa0-bionic amd64
[installed]
  RtBrick image store handling tool

rtbrick-imgstore-server/bionic 0.9.0-
internal.20210615185257+Bdevelopment.C506acfa4-focal amd64
  RtBrick Image Store Upload Server

rtbrick@access-test:~$ sudo apt show rtbrick-imgstore
Package: rtbrick-imgstore
Version: 2.1.1-xdaily.20220531062114+Cc35c1aa0-bionic
Priority: extra
Section: rtbrick-tools
Maintainer: RtBrick Support <support@rtbrick.com>
Installed-Size: 29.4 MB
Provides: rtbrick-imgstore
Depends: liblxc-common, liblxc1, lxc, zstd
Replaces: rtbrick-imgstore
Download-Size: 11.1 MB
APT-Manual-Installed: yes
APT-Sources: https://server.example.net/_/latest/ubuntu/rtbrick-tools
```

```
bionic/rtbrick-tools amd64 Packages
Description: RtBrick image store handling tool
 RtBrick package tracker UUID=89989764-69f8-4848-a066-8f8db2360253

N: There are 139 additional records. Use the '-a' switch to see them.
```

The tool is embedded with the GPG public key of support@rtbrick.com's identity. This key is used to sign all RtBrick images and the image store itself.

Common usage of rtb-image

rtb-image has enough versatility, but a few options are commonly used:

- containers list - List all the LXC containers that are created on the 'local' system.
- show [<flags>] <UUID> - Show details of the image identified by UUID. By default, this shows the image in the local cached copy of the store.
- run --name=NAME [<flags>] <UUID> - Run an LXC container using the specified image. The container should not already be created.
- list [<flags>] <UUID> - List all the images in the store. By default, this lists the images in the local cached copy of the store.

rtb-image list flags

Value	Description
-o, --remote	List images directly from the remote store and not from the local cached copy.
-d, --detailed	List detailed information about images.
-f, --format=FORMAT	List only images with a specific format.
-r, --role=ROLE	List only images with a specific role. Currently, roles are 'spine', 'accessleaf', and 'consolidated-bng'.
-p, --platform=PLATFORM	List only images for a specific platform.

Value	Description
-v, --ver-range=VER-RANGE	List only images versions that fall in the provided version range. See the syntax for version ranges at https://godoc.org/github.com/blang/semver#Range . The hardcoded strings 'latest' or 'newest' will always filter down to a single image, the one considered the newest according to the sorting rules for versions.
-l, --limit=LIMIT	Limit the list of returned images to the l newest images.
-m, --model=MODEL	List only images for a specific model. For virtual images, the model will always be empty. For hardware images, the model will be filled with the corresponding model. The value 'combined' indicates that the model contains support for multiple models.

rtb-image is used to create a local cache of the remote RtBrick image repository. The local cache can be created using the rtb-image update command.

```
rtbrick@access-test:~$ sudo rtb-image update
Local image store cached copy updated to: Store:
/var/cache/rtbrick/imagestores/847c6ecd-df58-462e-a447-38c620a12fe1 Version:
0.22.6360 ValidUntil: 2195-05-21 12:27:50.527696657 +0000 UTC
```

To list the local copies:

```
rtbrick@access-test:~$ sudo rtb-image list

Store: /var/cache/rtbrick/imagestores/847c6ecd-df58-462e-a447-38c620a12fe1
Version: 2.2.12237 ValidUntil: 2153-12-19 09:57:40

UUID                                  Version
Format       Role               Platform Model       Cached
c489b066-d4cd-4656-ae68-8438ce7f588c 24.3.0-g8daily.20240310111320+C4ba3e287
onl-installer rtbrick-onl-base   hw                   false
a9c706fe-d4ad-4eda-9dc5-9e9e181373e5 24.3.0-g8daily.20240310111320+Cc5ec66c6
lxd           rtbrick-jammy-base all                 false
d501fe94-40ae-4e63-ad33-14730f6b8e6d 24.3.0-g8daily.20240310103349+Bmaster.Ccc5...
onl-installer onl-base           hw                   false
15311e7e-f815-44cd-88f5-8b9d53939152 24.3.0-g8daily.20240310103345+Cc5ec66c6
lxd           upstream-jammy-base all                 false
```

```
ba614ae5-6348-41b6-a1f2-92ebbd4990f4 24.3.0-g8daily.20240303105945+Cc5ec66c6
lxd           rtbrick-jammy-base all                 false
75c8694a-fff6-43e1-b5b8-0be51e2427dd 24.3.0-g8daily.20240303105945+C4ba3e287
onl-installer rtbrick-onl-base   hw                   false
56711e4a-1b2f-437b-9648-3346e2f0ee4c 24.3.0-g8daily.20240303103433+Bmaster.Ccc5...
onl-installer onl-base           hw                   false
79aac073-72f3-4eb6-847e-22783fd68695 24.3.0-g8daily.20240303103430+Cc5ec66c6
lxd           upstream-jammy-base all                 false
1eb61a74-dfb6-46d6-a382-ca5d3b026054 24.3.0-g6daily.20240311164602+Bdevelopment...
onl-installer l2bsa               qax     combined     false
e30994ce-eb35-4f0c-a777-d5028d7da849 24.3.0-g6daily.20240311164602+Bdevelopment...
onl-installer consolidated-bng   qax     combined     false
31971f0a-fa4a-4266-a593-e5b4a6462772 24.3.0-g6daily.20240311164601+Bdevelopment...
onl-installer consolidated-bng   q2a     as7535-28xb false
ed9d83bb-5f4d-4e7f-a378-b5c7cd8f76db 24.3.0-g6daily.20240311164601+Bdevelopment...
onl-installer l2bsa               qax     s9500-22xst false
653dbe51-172c-429c-96d7-f352822c1117 24.3.0-g6daily.20240311164557+Bdevelopment...
onl-installer consolidated-bng   q2c     as7946-74xkb false
a1a9870c-f967-4f75-afb7-e756c01801fa 24.3.0-g6daily.20240311164557+Bdevelopment...
onl-installer consolidated-bng   q2a     combined     false
ea7137c6-a798-464e-ab15-0e5667b263d0 24.3.0-g6daily.20240311164556+Bdevelopment...
onl-installer consolidated-bng   q2c     s9600-72xc   false
1495068b-66fe-45b4-a0cd-64588738b449 24.3.0-g6daily.20240311164556+Bdevelopment...
onl-installer consolidated-bng   q2a     s9510-28dc   false
f68c358e-52db-4baf-934a-1d7ca773b0d8 24.3.0-g6daily.20240311164552+Bdevelopment...
onl-installer accessleaf         q2c     s9600-102xc false
dc89ef7c-8a56-4608-a058-0ea0eb3200d3 24.3.0-g6daily.20240311164552+Bdevelopment...
onl-installer accessleaf         q2c     s9600-72xc   false
4b402662-3ff3-45f5-ac31-d4a376fe6212 24.3.0-g6daily.20240311164551+Bdevelopment...
onl-installer accessleaf         q2c     as7946-74xkb false
33d3e86a-9bbd-44b2-9172-766de1b4462e 24.3.0-g6daily.20240311164551+Bdevelopment...
onl-installer consolidated-bng   q2c     combined     false
7c659767-008a-4b66-93ee-f510c6d43241 24.3.0-g6daily.20240311164547+Bdevelopment...
onl-installer spine               q2c     s9600-32x    false
6d5f2ff3-669b-4929-b51a-d0a0dcc48da3 24.3.0-g6daily.20240311164547+Bdevelopment...
onl-installer spine               q2c     combined     false
9f782e8d-9dc0-4c33-a9b6-b34f1f955c2f 24.3.0-g6daily.20240311164547+Bdevelopment...
onl-installer spine               q2c     as7946-30xb false
d368a511-ae69-4747-8eff-8b4c63522892 24.3.0-g6daily.20240311164546+Bdevelopment...
onl-installer accessleaf         q2c     combined     false
7c9419bc-17e7-4f9f-9d72-68b90981c960 24.3.0-g6daily.20240311163602+Bdevelopment...
lxd           consolidated-bng   q2a     as7535-28xb false
66ea8833-c21c-4d86-b2ac-a8b11b020888 24.3.0-g6daily.20240311163602+Bdevelopment...
lxd           l2bsa               qax     s9500-22xst false
91184443-648e-408d-a33d-7890e03a67b1 24.3.0-g6daily.20240311163602+Bdevelopment...
lxd           consolidated-bng   qax     combined     false
d505e1aa-ec33-4ef1-bb47-89db1921cf8f 24.3.0-g6daily.20240311163602+Bdevelopment...
lxd           l2bsa               qax     combined     false
283b7c6f-a448-4f46-8940-b4c8fc126d8e 24.3.0-g6daily.20240311163557+Bdevelopment...
lxd           consolidated-bng   q2c     as7946-74xkb false
1e3dc338-96f4-4fe2-aa2b-3c4da64a7322 24.3.0-g6daily.20240311163557+Bdevelopment...
lxd           consolidated-bng   q2a     s9510-28dc   false
29007803-33a6-44d8-99c6-2458f467f7db 24.3.0-g6daily.20240311163557+Bdevelopment...
lxd           consolidated-bng   q2c     s9600-72xc   false
fda5e4fe-a859-412f-bc84-56d64e9a8376 24.3.0-g6daily.20240311163557+Bdevelopment...
lxd           consolidated-bng   q2a     combined     false
0b082a32-87eb-469c-a98c-e3be7de9ebdc 24.3.0-g6daily.20240311163552+Bdevelopment...
lxd           accessleaf         q2c     as7946-74xkb false
fe87e8dd-0474-47a3-a4c9-fa5984105f6c 24.3.0-g6daily.20240311163552+Bdevelopment...
lxd           accessleaf         q2c     s9600-72xc   false
8453cd96-7920-486c-a564-1460e1bbdb2d 24.3.0-g6daily.20240311163552+Bdevelopment...
```

```
lxd           consolidated-bng   q2c     combined     false
f6ad99a5-fc93-4927-a03b-5926d903d274 24.3.0-g6daily.20240311163552+Bdevelopment...
lxd           accessleaf         q2c     s9600-102xc false
fc433fe3-b9c4-4d03-9aaa-464ff0f56cff 24.3.0-g6daily.20240311163547+Bdevelopment...
lxd           spine               q2c     s9600-32x   false
d826e098-ea22-49d8-b241-c4d93ae8af3f 24.3.0-g6daily.20240311163547+Bdevelopment...
lxd           spine               q2c     as7946-30xb false
e246cf8a-6106-4dc3-b677-3dbdafd22949 24.3.0-g6daily.20240311163547+Bdevelopment...
lxd           spine               q2c     combined     false
1f8f649b-a148-44d7-ab97-f586cb7a2907 24.3.0-g6daily.20240311163547+Bdevelopment...
lxd           accessleaf         q2c     combined     false
```

rtbrick-fabric-ztp

This package provides an implementation of a Zero-Touch Provisioning (ZTP) server. It is required for automatic provisioning of switches newly installed in a network, where the ZTP server provides installation or upgrade images for the switches.

The rtbrick-fabric-ztp package can be installed with the following command:

```
sudo apt install rtbrick-fabric-ztp
```

After installation, the server is running as a systemd-daemon by default on port 80 and taking ZTP configuration files from the location /var/rtbrick/ztp/configs/.

For information about Zero Touch Provisioning, see section *Zero-Touch Provisioning*.

Image formats and ONL image installation for supported hardware

RtBrick images delivered through the RtBrick image store and the rtb-image utility have three main attributes:

- format: This is the file format in which the image is packaged and archived.
- role: The role inside a network of the device which will be running the image.
- platform: Identifies the hardware platform or virtualized environment in which the image can run.

RtBrick images intended to be used as containers in a virtualized environment contains format == lxd and platform == virtual.

RtBrick images intended to be installed on supported hardware devices contain format == onl-installer and platform set accordingly to the specific switching hardware.

You can see this using rtb-image list command and looking for the Format column.

ONL images

ONL images are generally installed using the Zero-Touch Provisioning (ZTP) process. The [Installation] section applies for both virtual and hardware installations, with the difference that, when having a physical deployment (One with a ZTP server and switched running ONL images) we can install just the rtbrick-imgstore package on the ZTP server, since it doesn't have Ansible as dependency (Ansible not being a part of the default Ubuntu repositories), and because generally you will not have containers running on the ZTP server itself.

A typical ONL image downloaded looks as in the following snippet:

```
rtbrick@access-test:~$ sudo rtb-image update
Local image store cached copy updated to: Store:
/var/cache/rtbrick/imagestores/847c6ecd-df58-462e-a447-38c620a12fe1 Version:
0.22.6360 ValidUntil: 2195-05-21 12:27:50.527696657 +0000 UTC

rtbrick@access-test:~$ sudo rtb-image list --format onl-installer --platform qmx
--role spine --ver-range latest

Store: /var/cache/rtbrick/imagestores/847c6ecd-df58-462e-a447-38c620a12fe1
Version: 0.22.6360 ValidUntil: 2195-05-21 12:27:50.527696657 +0000 UTC

UUID                                    Version
Filename                                        Format          Role    Platform
Cached
fa52399e-a2d8-48f0-8ad0-5b17b69b826d    22.6.0-
g8daily.20220605220700+Bmaster.C2f0...   rtbrick-onl-installer/rtbrick-onl-
installe...   onl-installer   spine   qmx         false

rtbrick@access-test:~$ sudo rtb-image pull fa52399e-a2d8-48f0-8ad0-5b17b69b826d
rtbrick-onl-installer-spine-qmx-22.6.0-
g8daily.20220605220700+Bmaster.C2f0eae65.d.sha512 244 B / 244 B
[=============================================================] 100.00% 0s
rtbrick-onl-installer-spine-qmx-22.6.0-
g8daily.20220605220700+Bmaster.C2f0eae65.d.asc 833 B / 833 B
[=============================================================] 100.00%
0s
rtbrick-onl-installer-spine-qmx-22.6.0-g8daily.20220605220700+Bmaster.C2f0eae65.d
1.15 GiB / 1.15 GiB
[=============================================================] 100.00% 5s
rtbrick-onl-installer-spine-qmx-22.6.0-g8daily.20220605220700+Bmaster.C2f0eae65.d:
decompressing 100 B / 100 B
[=============================================================] 100.00% 0s
```

```
rtbrick@access-test:~$ sudo rtb-image show fa52399e-a2d8-48f0-8ad0-5b17b69b826d

Store: /var/cache/rtbrick/imagestores/847c6ecd-df58-462e-a447-38c620a12fe1
Version: 0.22.6360 ValidUntil: 2195-05-21 12:27:50.527696657 +0000 UTC

UUID:               fa52399e-a2d8-48f0-8ad0-5b17b69b826d
Version:            22.6.0-g8daily.20220605220700+Bmaster.C2f0eae65
Creation Date:      2022-06-06 03:37:00 +0530 IST (7 hours ago)
Role:               spine
Platform:           qmx
Format:             onl-installer
Architecture:       amd64
Filename:           rtbrick-onl-installer/rtbrick-onl-installer-spine-qmx-22.6.0-
g8daily.20220605220700+Bmaster.C2f0eae65.d
FullPath/URL:       /var/cache/rtbrick/imagestores/847c6ecd-df58-462e-a447-
38c620a12fe1/rtbrick-onl-installer/rtbrick-onl-installer-spine-qmx-22.6.0-
g8daily.20220605220700+Bmaster.C2f0eae65.d
SHA512:
9a6b989edacf8daedc656dce310eb4ad680dc3cd5afd16a9c1783bb57ac78c97b4cc2d30fd0ac91edd
2ef7a86c7d6aea6441ef2ef0fbd16cd1f682f21601ed64
Base Image:         fd52536e-bd16-421d-a883-da263768aeb6
Embedded Packages:  18
Embedded Images:    1
Cached:             true
ExtractedPath:
```

In a design where the download of the image happens on a different server than the ZTP used for the actual installation, you can install the *rtbrick-imgstore* package, and move by some means (rsync, for example) the images from var/cache/rtbrick/imagestore/ of that internet-connected to the ZTP server.

The rtb-ssh command

The rtb-ssh command allows you to connect to a container that is already running.

The command was previously called rssh, and it was renamed as it confused with Linux's restricted shell *rssh* package which is available in the official Ubuntu *apt* package repositories.

Besides renaming, a few changes have been made to the rtb-ssh / rssh script.

The script is installed automatically as part of the rtbrick-imgstore package installation.

The script uses lxc-attach to establish a connection to the container specified as the argument. While doing so, it uses the ubuntu user (currently, the default user inside an RBFS container) to connect to the container, and uses the bash shell after opening the connection.

Before connecting, it clears the environment before attaching, so no undesired environment variables leak into the container. The variable container=lxc is the only environment with which the attached program starts.

It only keeps the TERM variable, to have the same strings the user is currently using for clear screen, move cursor, and so on.

The **rtb-ssh** is installed in the /usr/local/bin/ path (alongside rtb-image, etc.). For convenience and backward compatibility the script is still also installed as **rssh**.

Installing ONL Manually

You can install open network Linux (ONL) manually on a bare-metal switch. Open Network Install Environment (ONIE) should be installed on the switch. The Open Network Install Environment (ONIE) is an open-source utility that provides an installation environment for bare-metal switches. ONIE is used to install different network operating systems (NOS) on a device.

The RtBrick ONL installer images are compatible with ONIE and can be used by ONIE to install an RtBrick ONL (Open Network Linux) on a bare-metal switch.

- When installing ONL, any existing configurations on the switch will be deleted.
- The current RBFS configurations can be retrieved via a REST call from the RESTCONF endpoint. If you have saved the RBFS configuration using this method, you can load it onto the switch through a RESTCONF endpoint. For more information, refer to the following sections of the RtBrick documentation.

  Using the Proxy Endpoint

  RESTCONF API: Use Cases and Examples

Prerequisites

- Ensure that ONIE has been installed on the switch by the vendor of the switch. If ONIE is not unavailable with the switch, contact the switch vendor.
- Ensure that the switch management interface has been provisioned with an IP address either through manual configuration or through DHCP.
- Ensure that you have set up the necessary infrastructure to download RtBrick

ONL installer images on your environment. For information, see section *RtBrick Tools Installation*.

- Ensure that you have set up an HTTP server that will make available the downloaded images for ONIE to use. For more information, see https://opencomputeproject.github.io/onie/user-guide/index.html#installing-over-the-network

Installation Procedure

To install the ONL image, perform the following steps:

On a fresh box, ONL prompt is not available, so skip to ONIE prompt section.

ONL prompt section:

Option 1: Manually select ONIE boot mode

1. Connect to the console port
2. Reboot the device

```
root@bl1-pod1:~# reboot
```

3. Once the selection menu appears as shown in the selection menu below, select "**ONIE**" and press enter.

```
                    GNU GRUB  version 2.02

+----------------------------------------------------------------------------+
| Open Network Linux                                                         |
|*ONIE   <----- Select this one                                              |
+----------------------------------------------------------------------------+

      Use the ^ and v keys to select which entry is highlighted.
      Press enter to boot the selected OS, `e' to edit the commands
      before booting or `c' for a command-line.
```

4. Select "**ONIE: Install OS**" from the next selection menu displayed.

```
                    GNU GRUB  version 2.02
 +----------------------------------------------------------------------------+
 |*ONIE: Install OS   <----- Select this one                                  |
 | ONIE: Rescue                                                               |
```

```
 | ONIE: Uninstall OS                                                          |
 | ONIE: Update ONIE                                                           |
 | ONIE: Embed ONIE                                                            |
 +-----------------------------------------------------------------------------+

      Use the ^ and v keys to select which entry is highlighted.
      Press enter to boot the selected OS, `e' to edit the commands
      before booting or `c' for a command-line.
```

5. Wait for the "**ONIE:/ #**" prompt.

```
NOTICE: ONIE started in NOS install mode.  Install mode persists
NOTICE: until a NOS installer runs successfully.

** Installer Mode Enabled **
ONIE:/ #
ONIE:/ #
ONIE:/ #
```

Provide the URL of the ONL installer image location.

```
ONIE:/ # onie-nos-install http://server.example.net/_/images/latest/rtbrick-onl-
ins
taller/rtbrick-onl-installer-spine-q2c-21.9.1.d
```

Wait until the device displays the "**login:**" prompt after the image upgrade completes. You can then log into the device and verify the image version.

Option 2: Preselect ONIE boot mode

1. Connect to the console port
2. Select ONIE boot mode

```
root@onl>bl1-pod1:~ # onl-onie-boot-mode --help
usage: onl-onie-boot-mode [-h] [--onie-only]
                          {install,rescue,uninstall,update,embed,diag,none}

positional arguments:
  {install,rescue,uninstall,update,embed,diag,none}

optional arguments:
  -h, --help            show this help message and exit
  --onie-only           Do not set ONIE boot menu option.
root@onl>bl1-pod1:~ #

root@onl>bl1-pod1:~ # onl-onie-boot-mode install
The system will boot into ONIE install mode at the next restart.
root@onl>bl1-pod1:~ #
```

3. Reboot switch

```
root@onl>bl1-pod1:~ # reboot
```

ONIE prompt section:

You must update the URL of the ONL installer image location as per your specific HTTP server configuration.

```
ONIE:/ # onie-discovery-stop
NOTICE: The 'onie-discovery-stop' command is deprecated and will be removed in
2019.02.
NOTICE: Use 'onie-stop' instead.
discover: installer mode detected.
Stopping: discover... done.

ONIE:/ # onie-nos-install http://server.example.net/_/images/latest/rtbrick-onl-
ins
taller/rtbrick-onl-installer-spine-q2c-21.9.1.d

discover: installer mode detected.
Stopping: discover... done.

Info: Attempting http://server.example.net/_/images/latest/rtbrick-onl-
installer/rtbrick-onl-installer-spine-q2c-21.9.1.d ...

Connecting to server.example.net (198.51.100.125)
installer 100% |*******************************| 1176M 0:00:00 ETA

ONIE: Executing installer: http://server.example.net/_/images/latest/rtbrick-onl-
installer/rtbrick-onl-installer-spine-q2c-21.9.1.d
```

2. Zero Touch Provisioning

Overview

Zero Touch Provisioning (ZTP) automates the tasks of installing software images. It is a method for setting up and configuring devices automatically. ZTP installs or upgrades the RBFS software image on your hardware platforms without any manual intervention.

ZTP automatically provisions routers newly installed in the network and it is very useful in deploying routers in a large-scale environment as it eliminates much of the manual intervention. ZTP is also used to automate the software upgrade process and help with a high level of network automation.

ZTP Workflow

A new hardware platform comes pre-installed with the ONIE (Open Network Installation Environment). ONIE is an open-source installation environment that acts as an enhanced boot loader utilizing capabilities in a Linux or BusyBox environment. ONIE allows users and channel partners to install the Network Operating System as part of provisioning.

ONIE requires a management LAN to obtain the configuration and software image information through the management interface. ONIE can access only the management interface. It starts a Dynamic Host Configuration Protocol (DHCP) based discovery process to obtain basic configuration information, such as the management IP address and the URL of the image to install on the bare-metal switch.

Then ONIE pulls the image and boots it.

Even after ONIE boots the image, the switch is not configured. This leads to questions about how to configure the switch.

The RtBrick images come with some pre-installed daemons. The pre-installed Control Daemon (CtrlD) is responsible for the management of the switch, and takes over after the image is activated.

The Control Daemon is responsible for configuring the switch. To do this, the

hardware platform must be connected to the DHCP server and the management server through a management LAN.

The management server is responsible for providing the image binaries and the configuration of each device.

In the ZTP, ONIE performs the role of discovering, downloading and activating the image from the image registry.

In essence, the following is the high-level workflow of ZTP process:

ONIE performs the following tasks:

- DHCP discovery
- Image download
- Image activation

Control Daemon performs the following tasks:

- DHCP discovery
- Switch configuration

ONIE allows to automate the firmware update. The image request to the management server is slightly different, and the management server needs to provide the firmware update image that the device vendor provides.

This section provides information about the NOS installation and firmware (FW) update.

ZTP Process

This section provides information about ZTP process. Figure. 1 illustrates the ZTP process at a high level.

The ZTP process is divided into two main parts:

Software Image Discovery and Installation

The ONIE in the device uses information that you have defined on the Dynamic Host Configuration Protocol (DHCP) server to locate the IP address and image download URL.

- ONIE uses different ways to pull the image from the repository for downloading. In the ZTP process, HTTP is used to pull the image because ONIE conveys the serial number as the HTTP header. This serial number allows the image registry to identify the switch and select the appropriate image.

Along with the serial number, ONIE also sends the onie-operation that allows to distinguish between an os-install and onie-update, and select the correct image for either NOS install or firmware upgrade.

- See the ONIE image discovery for further information (/ONIE/)
- CtrlD configuration discovery and application.
- CtrlD sends DHCPINFORM to request all options required for configuration discovery.
- The configurations are downloaded from the management server (httpd) and applied.

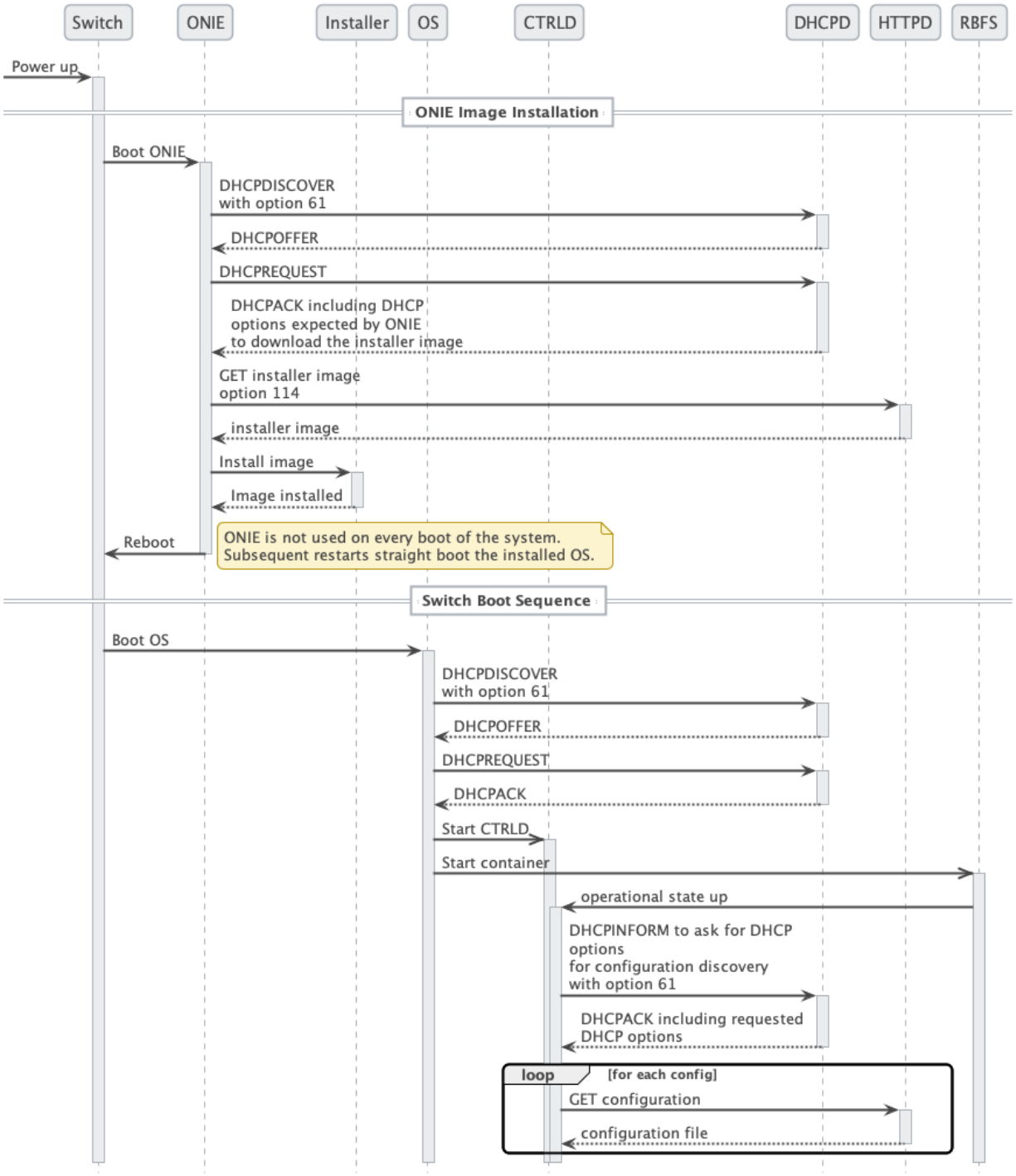

Figure 1. The ZTP Process

Figure 2. depicts the relationship between the fabric, the DHCP server, and the management server.

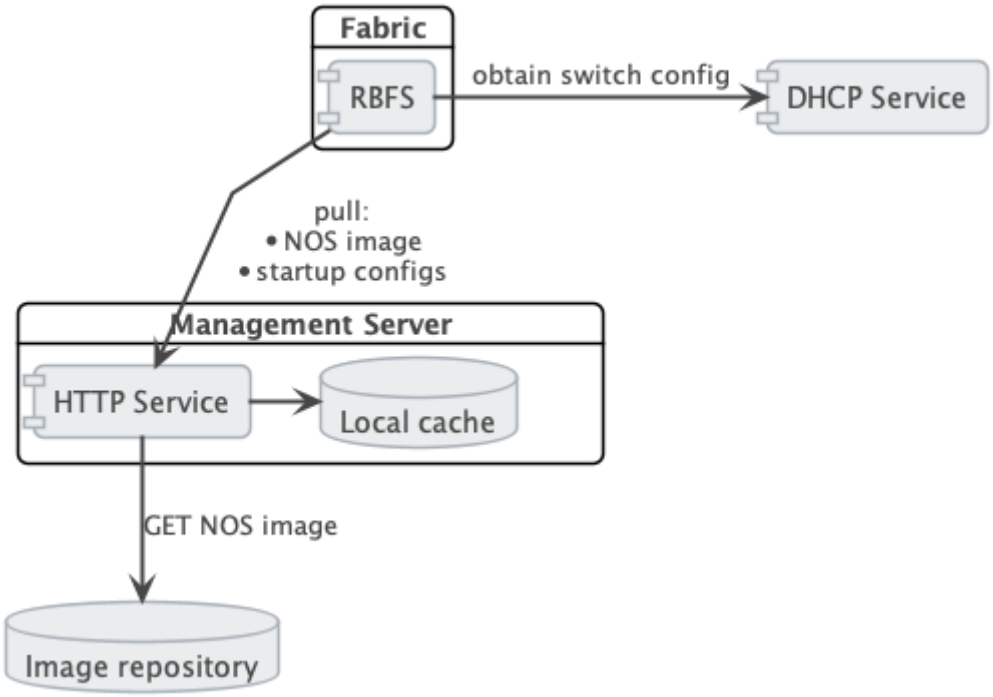

Figure 2. The Management Server Architecture

DHCP Service

Because of its low set of requirements, the default DHCP server shipped with ubuntu, isc-dhcp, is used to run the DHCP service.

The following code shows an example configuration of a DHCP server and hardware box (**dhcp.conf**).

dhcp.conf

```
authoritative;
default-lease-time 600;
max-lease-time 72----

# This is only needed if the version is lower than 4.4
option loader-pathprefix code 210 = text;


subnet 10.0.0.0 netmask 255.255.255.0 {
  range 10.0.0.200 10.0.0.250;
  option routers 10.0.0.138;
  option domain-name-servers 10.0.0.210;
  option domain-name "local";
  host LEAF01 {
    # Identify client by MAC address.
    hardware ethernet 48:65:ee:11:da:85;
    # Identify client by serial number
    option dhcp-client-identifier "\000WLC1C27L00003P2";
    fixed-address 10.0.0.250;
    option host-name LEAF01;
```

```
    # Set DHCP option 114 (default-url) to set the installer image URL.
    # ONIE loads the installer image from the specified URL.
    option default-url "http://managementserver/ztp/image";
    # Set DHCP option 210 (path prefix) to set the configuration base URL.
    # CTRLD loads all configuration files from this base URL.
    option loader-pathprefix "http://managementserver";
  }
}
```

Most of the used options are already predefined in the ISC-DHCP server. You can see the reference under /ISCKB/, the loader-pathprefix is defined since DHCP 4.4, so if you use an older one, define it as described above.

HTTP Service (Management Server)

The HTTP daemon (httpd) is responsible for providing the NOS installer and the configuration files.

Therefore, a self-implemented Golang HTTP server is used, which reads the ONIE_SERIAL_NUMBER and ONIE-OPERATION HTTP header and maps them to the NOS/FW installer image download path, and maps the serial number to the ZTP configuration files. For more details about the configuration files, see the following section.

The ONIE-OPERATION header can have the following values:

- install nos: os-install
- update firmware: onie-update

The following sections provide information about the installation and configuration of the server.

ZTP installation

For the installation, you can choose any one of the following two options:

ZTP Installation with the Debian Package

You must perform the following steps for ZTP installation using the Debian package.

- Ensure that you have added the rtrbick repository to your apt.sources list and updated the cache.

- Ensure that the port 80 is available and not in use on your device.
- Install the package rtbrick-fabric-ztp.
- The package installs a systemd service named rtbrick-fabric-ztp.
- Ensure that the service is running with sudo systemctl status rtbrick-fabric-ztp.
- The default location for the ZTP configuration files is /var/rtbrick/ztp/configs/ where you need to copy your configuration files.

If you want to override server settings, perform the following:

- Edit the service configuration file /etc/systemd/system/rtbrick-fabric-ztp.service and add parameters to the ExecStart command.
- Parameter --addr: the listen address of the server, default is 0.0.0.0:80.
- Parameter --requestTimeout: the request timeout server in seconds, default is 600, must possibly be increased depending on the connection speed and image file sizes.
- Parameter --filePath: the location for the ZTP configuration files, the default location is /var/rtbrick/ztp/configs/.

ZTP Installation as Docker Container

You must perform the following steps for ZTP installation as a docker container.

- Ensure that you have access to the rtbrick docker registry.
- Ensure that the port 80 is available and not in use on your device.
- Create a compose file docker-compose.yml. The following is a sample compose file.

```
version: '3.3'
services:
  ztp:
    image: 'docker.rtbrick.com/rbms-fabric-ztp:latest'
    container_name: rbms-fabric-ztp
    restart: unless-stopped
    ports:
      - '80:80'
    volumes:
      - './configs:/var/rtbrick/ztp/configs'
```

- The compose setup uses a 'bind mound' method for the ZTP configuration

folder. Therefore, the docker-compose.yml must be placed in the same location together with the ./configs folder for the ZTP configurations. To know the details of the configuration files, see the following sections.

- If required, adapt the compose file for a different image version, port binding or different configuration folder location.
- Start the container using the docker-compose up -d command.

ZTP configuration

The HTTP service matches the ONIE-SERIAL-NUMBER header to the configuration files. Therefore, the configuration folder should contain a JSON file for the serial number (<serial_number>.json) for each supported serial number.

This file contains settings for locations of all additional configuration files that have to be delivered for the specific device and settings for the NOS installer image and the firmware update image.

Example sample.json file for a serial number 'sample':

/var/rtbrick/ztp/configs/sample.json

```
{
  "description": "192.168.202.116",
  "ctrld": "ctrld.json",
  "ctrldrbac": "ctrldrbac.json",
  "startup": "sample_startup.json",
  "accessjwks": "sample_accessjwks.json",
  "apigwd": "sample_apigwd.json",
  "tls": "sample_tls.pem",
  "image": "http://pkg.rtbrick.net/_/images/latest/rtbrick-onl-installer/rtbrick-
onl-installer-accessleaf-qmx-20.4.0-g8daily.20200415051734+Bmaster.C059a09ea",
  "update_image": "http://pgk.rtbrick.net/firmwares/onie-firmware-x86_64-
ufispace_s9600_32x_ufispace_s9600_64x-r0_v0.3.0.updater"
}
```

Image Location Configuration

For the configuration entries "image" and "update_image" you have three possibilities:

- Redirect URL: Configuration value must start with http, the server redirects the request to download the image from the URL. For example, "http://pkg.rtbrick.net/_/images/latest/rtbrick-onl-installer/rtbrick-onl-installer-accessleaf-qmx-20.4.0-g8daily.20200415051734+Bmaster.C059a09ea"

- Absolute File Location: config value must start with /, can point to any file on the local disk, example /usr/share/images/rtbrick-onl-installer.img.
- Relative File Location: config value must be a filename and not start with /, points to any file in the <ztppath>/configs/images/ folder, example "rtbrick-onl-installer.img"

ZTP APIs

For information about ZTP REST APIs, refer to the /resources/techdocs/current/api/rbms-apis.html[ZTP Management Server API].

Control Daemon

Once the RBFS image is activated by ONIE, Control Daemon (CtrlD) is responsible for executing the remaining tasks and configuring the switch. CtrlD acts as a post-ZTP demon, it runs after the image is activated.

There are various configuration files that CtrlD can load from a management server and apply to the system.

- **CtrlD config**: This is the base configuration for CtrlD. There the RBMS and Graylog can be specified, but also the authentication and authorization mechanism can be controlled.
- **CtrlD rbac policy**: The Role Based Access Control (RBAC) policy of CtrlD is defined in this configuration file.
- **Startup Config**: This is the file for RBFS switch configuration.
- **TLS pem file**: This file is intended for the API Gateway (ApiGwD). The file is an X509 public/private key file in PEM format defined in the RFC7468.
- **Access JWKS file**: This file is intended for the ApiGwD. The JSON Web Key Set (JKWS) is described in the RFC 7517.

Trigger the ZTP process

The ZTP process in CtrlD is triggered for a specific container (LXC) on the switch. This can be triggered in the following ways.

- By the switch (RBFS Linux container) itself by sending the *operational state up* to CtrlD.

- By sending a REST request to trigger the ZTP process to CtrlD (/api/v1/ctrld/ztp/_run).

If 'load-last-config' option is set to true, ZTP is in the disabled state. ZTP is enabled if load-last-config is false.

By default, 'load-last-config' is false and ZTP is enabled. You must set to 'load-last-config' true to disable ZTP.

Trigger the reinstall

The reinstall of a switch can be triggered by sending a POST request to CtrlD (/api/v1/ctrld/system/_reinstall)

Trigger Firmware Update

The firmware update of a switch can be triggered by sending a POST request to CtrlD (/api/v1/ctrld/system/_update)

Management Server URL Discovery

CtrlD has to discover the management server URL to download the configuration files from the management server. Therefore, a management interface, that allows sending an DHCPINFORM request to the DHCP server, is defined.

The request contains the **DHCP option 61** that conveys the client identifier. The client identifier is either omitted or contains the serial number. The serial number is gathered from the ONIE file system information file /lib/platform-config/current/onl/onie-info.json. If that does not result in a valuable result the following command is executed dmidecode -s system-serial-number (see /RFC2131/ and /RFC2132/ for further information).

There are at least two DHCP options requested, **DHCP option 54** that conveys the IP address of the DHCP server (see /RFC2132/ for further information), and **DHCP option 210** that conveys the path prefix for all configuration files (see /RFC5071/ for further information).

If the DHCP option 210 is not returned, CtrlD attempts to read the configurations from the IP address of the ZTP server. Otherwise, CtrlD attempts to read the configurations from the base URL specified in DHCP option 210.

Request configurations

The request to the Management server contains the following HTTP headers:

- ONIE-SERIAL-NUMBER: This serial number is either the onie serial number or empty string.
- CONTAINER-NAME: Container that triggered the ZTP process.
- ELEMENT-NAME: Element name that triggered the ZTP process.
- HOST-NAME: Host name of the device that triggered the ZTP process.

All this information can be used to select the right configurations for the container. This also allows the use of ZTP Configuration Process for virtual environments.

The requested URL:

- CtrlD Config: <management server url>/ztp/config/ctrld
- CtrlD rbac policy: <management server url>/ztp/config/ctrldrbac
- Startup Config: <management server url>/ztp/config/startup
- TLS pem file: <management server url>/ztp/config/tls
- Access JWKS file: <management server url>/ztp/config/accessjwks

If any of the file is not found, the process still goes forward.

Business Events

During the ZTP Process log messages are sent to the configured ztp graylog endpoint.

For more information, see the switch API documentation.

Overall Process Flow

The following two figures show the CtrlD ZTP process flow.

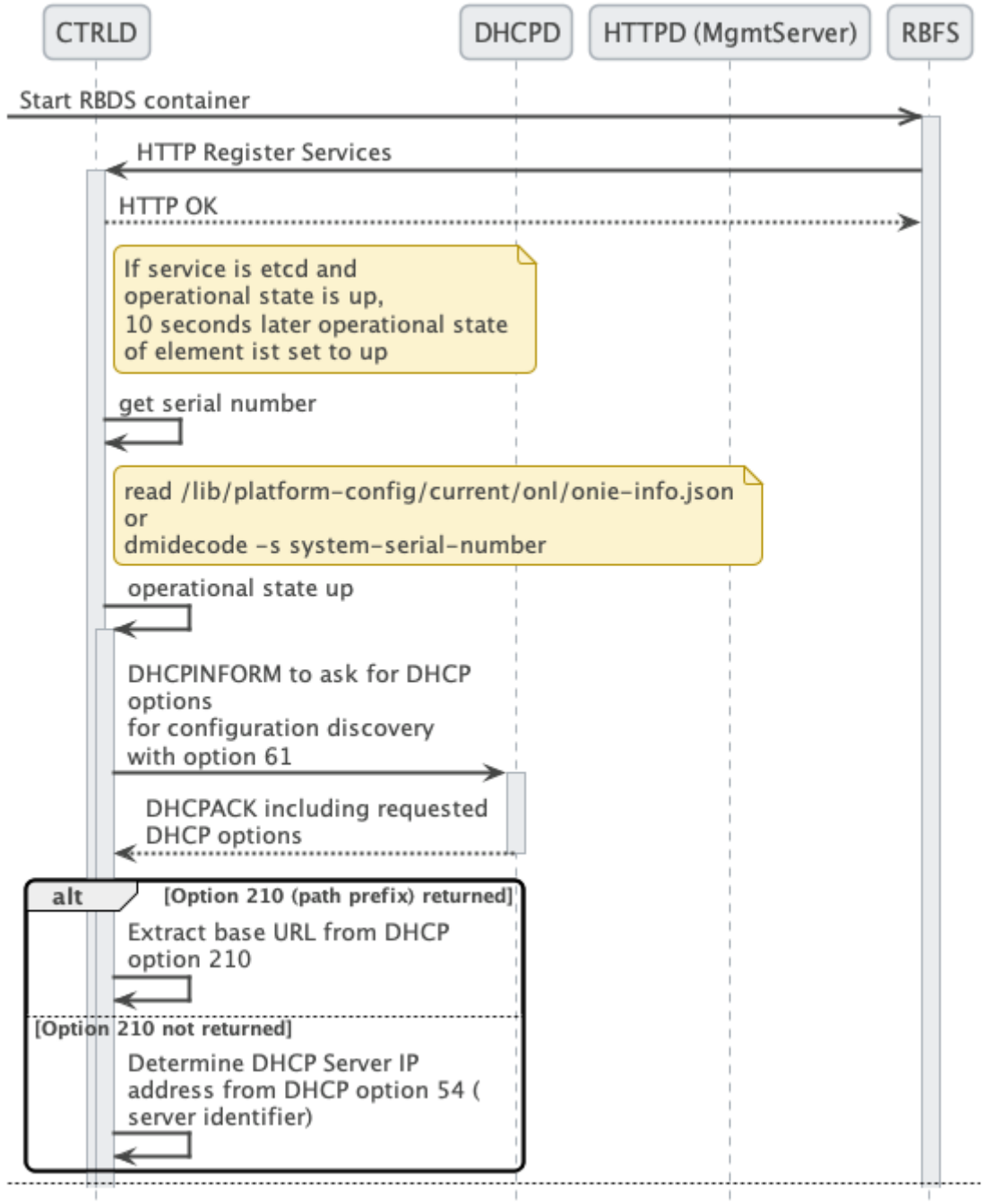

Figure 3. CTRLD ZTP process flow (Part 1/2)

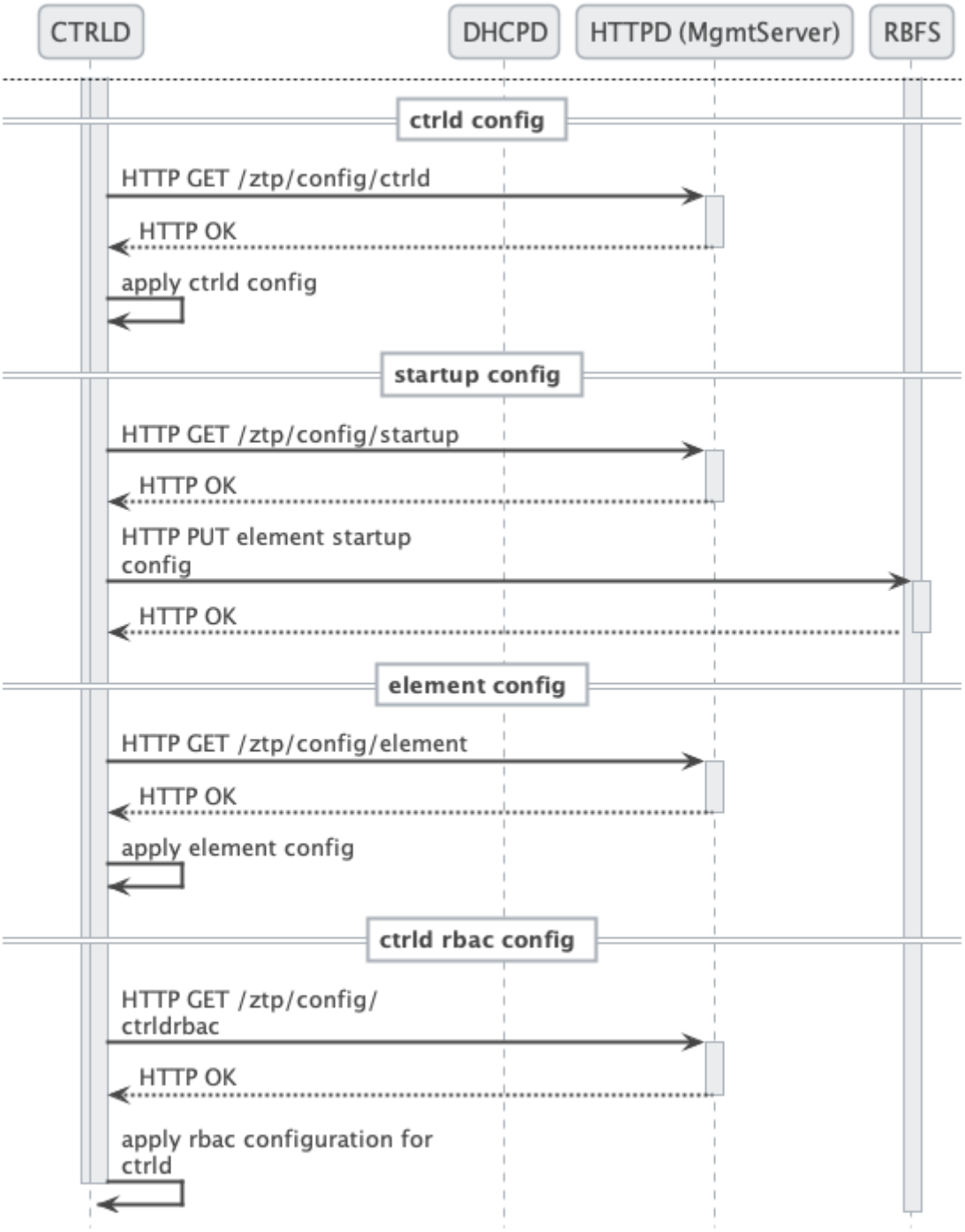

Figure 4. CTRLD ZTP process flow (Part 2/2)

References

References

/ONIE/	Open Network Installation Environment Image Discovery
/RFC2131/	RFC2131 - Dynamic Host Configuration Protocol
/RFC2132/	RFC2132 - DHCP Options and BOOTP Vendor Extensions https://tools.ietf.org/html/rfc2132

/RFC5071/	RFC5071 - Dynamic Host Configuration Protocol Options Used by PXELINUX
/ISCKB/	ISC Default DHCP Options

3. RBFS Licensing

Overview

RBFS Licensing allows you to access the full functionality of your RtBrick FullStack (RBFS) installation. Rtbrick provides a 28-day evaluation license on request. It is not allowed to be used in production. Use a permanent or subscription license that has been purchased through RtBrick Sales. If you want to extend the evaluation period and get additional licenses, contact RtBrick Support.

Without any license installed on your system, you can evaluate RBFS for 7 days. You need to get an evaluation license or purchase an actual license within 7 days to use the full functionality of RBFS.

Obtaining or Extending Licenses

To obtain new RBFS licenses or extend the existing licenses, go to https://portal.rtbrick.com/, click **Licenses** in the left navigation pane, and then select the **Request license** link.

Installing a License

You can install a license by using the RBFS CLI or via the RESTCONF API. You should get a license encrypted string from Rtbrick and configure the same via CLI.

When you upgrade your RBFS installation, the existing license should either get restored via saved configuration or it needs to be installed again.

To install a license, enter the following command:

Syntax

set system license <license_key>

Example

```
supervisor@rtbrick: cfg> set system license
“eyJzdGFydF9kYXRlIjogMTYxNTg3MTE3MCwgImVuZF9kYXRlIjogMTYxNTk1NzU3MH0=.Yx/XiFDFRzAt
XPUOaIoh5GqiXa+kOJBWp3LgDeJooVrl88mpPs2ZRMPC+k5HvoZDXvsreqRrqoFR3vk7S2PlqmLxYf0bNB
ly4d1hrloBwwFkFuJaiU/M+ZGPExgILdVyXumI88VYx8m/Z5SxEj0bFQGUy8UHRUYW/Ay8fhPfYejWuSgp
v3OrIThH9CVjlDmrp/k4yOuHyTz5gLgq4A0h33vB5O99aOIJW5UX4XDKvQqmqX5kytRlR1SseWuAbWKjUd
VOkf2Mk36IbF9/xAKier++LzXESpLMI+MT63AybSDHOBZydoMjLH9C6cPEfGHzWTIBNtT3679Tokf25EK1
Jw==”
```

The following example shows the running configuration.

```
supervisor@rtbrick: cfg> show config system
{
    "rtbrick-config:system": {
        "license": [
            {
                "license-key":
"eyJzdGFydF9kYXRlIjogMTYxNTg3MTE3MCwgImVuZF9kYXRlIjogMTYxNTk1NzU3MH0=.Yx/XiFDFRzAt
XPUOaIoh5GqiXa+kOJBWp3LgDeJooVrl88mpPs2ZRMPC+k5HvoZDXvsreqRrqoFR3vk7S2PlqmLxYf0bNB
ly4d1hrloBwwFkFuJaiU/M+ZGPExgILdVyXumI88VYx8m/Z5SxEj0bFQGUy8UHRUYW/Ay8fhPfYejWuSgp
v3OrIThH9CVjlDmrp/k4yOuHyTz5gLgq4A0h33vB5O99aOIJW5UX4XDKvQqmqX5kytRlR1SseWuAbWKjUd
VOkf2Mk36IbF9/xAKier++LzXESpLMI+MT63AybSDHOBZydoMjLH9C6cPEfGHzWTIBNtT3679Tokf25EK1
Jw==”
            }
        ]
    }
}
```

Installing Multiple Licenses

You can install multiple licenses. Additional licenses can be installed even when you have existing license(s). The license with the maximum evaluation period will be prioritised over others. When you have multiple evaluation licenses installed, the one that expires later takes higher priority compared to the other licenses.

Viewing the installed license

Syntax

show system license

Example

```
root@rtbrick: cfg> show system license
License Validity:
  License 1:
    Start date : Tue Mar 16 05:06:10 GMT +0000 2021
    End date   : Wed Mar 17 05:06:10 GMT +0000 2021
root@rtbrick: cfg>
```

After verifying the validity of the license, the license file will be installed at the following location:

```
/etc/rtbrick/license/rtbrick-license
```

Deleting a License

To delete a license, enter the following command:

Syntax

delete system license <license_key>

Example

```
supervisor@rtbrick: cfg> delete system license
“eyJzdGFydF9kYXRlIjogMTYxNTg3MTE3MCwgImVuZF9kYXRlIjogMTYxNTk1NzU3MH0=.Yx/XiFDFRzAt
XPUOaIoh5GqiXa+kOJBWp3LgDeJooVrl88mpPs2ZRMPC+k5HvoZDXvsreqRrqoFR3vk7S2PlqmLxYf0bNB
ly4d1hrloBwwFkFuJaiU/M+ZGPExgILdVyXumI88VYx8m/Z5SxEj0bFQGUy8UHRUYW/Ay8fhPfYejWuSgp
v3OrIThH9CVjlDmrp/k4yOuHyTz5gLgq4A0h33vB5O99aOIJW5UX4XDKvQqmqX5kytRlR1SseWuAbWKjUd
VOkf2Mk36IbF9/xAKier++LzXESpLMI+MT63AybSDHOBZydoMjLH9C6cPEfGHzWTIBNtT3679Tokf25EK1
Jw==”
```

License Expiry

When a license expires, you will not be able see the operational state of the system via CLI or BDS API.

License Validation

The process of verifying the validity of the software license is known as license validation. If no license is installed, a 7-day evaluation period will be provided.

During this time, there will be no license validation. After the evaluation period ends, the system will check perform license validation every 12 hours. If a valid license is not found, access to the operational state of the system via CLI or BDS API will not be available.

Once a license is installed on the device, it will be validated every 12 hours. If a license is installed within 7 days of evaluation, it is considered the end of the evaluation period, and the license validation will start from that point onward.

Relevant warning or error messages will be generated based on the license validation:

- A warning is generated if the license validity is less than seven days.
- An error message is generated if the license validity is less than one day.
- A critical message is generated if the license has already expired.

Both BDS and file logs are generated for license expiry, and if the Graylog plugin is configured, they are sent to the Graylog. For a list the logs related to license expiry, refer to the section License Log Messages.

To find out the details about the license installed on your system, run the “show system license” command as explained in the section Viewing the installed license.

Registered Address	Support	Sales
40268, Dolerita Avenue Fremont CA 94539		
+1-650-351-2251		+91 80 4850 5445
http://www.rtbrick.com	support@rtbrick.com	sales@rtbrick.com