RBFS and Tools Installation, and RBFS Licensing Guide

Version 21.3.1, 14 April 2021

Registered Address	Support	Sales
26, Kingston Terrace, Princeton, New Jersey 08540, United States		
		+91 80 4850 5445
http://www.rtbrick.com	support@rtbrick.com	sales@rtbrick.com

Table of Contents

1. RBFS and Tools Installation

1.1. Introduction

RtBrick software is delivered via different means: **RBFS** (RtBrick Full Stack) software is delivered as custom RtBrick container images (also called *RBFS container images*) which can used for virtual topologies on x86 servers or as custom RtBrick ONL installer images which be used on supported whitebox 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 to be used on supported Ubuntu Linux distribution (currently the only supported Ubuntu release is 18.04 LTS Bionic Beaver). We usually refer to this category of software as *RtBrick Tools*. The software delivered as Debian packages is composed of a set of CLI tools and/or daemons meant to facilitate working with RBFS containers and the RBFS API.

Debian package repositories containing the packages are published on https://releases.rtbrick.com/ and updated when new package versions are available.

1.2. Managed 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).

In order to gain access to the **restricted** *image stores* and *debian package repositories* on https://releases.rtbrick.com/, perform the following steps:

Step 1: Customer must generate client certificate

To help (facilitate) this step RtBrick provides the rtb-apt tools. For more information, see section The RtBrick APT tool.

Step 2: Customer must send client certificate to RtBrick

Step 3: RtBrick will approve and trust client certificate

Step 4: Customer can now use usual rtb-apt, apt, rtb-image tools to download RtBrick software

rtb-image must be at least version 1.3.0 to correctly work with managed downloads.

1.3. The RtBrick APT tool

The RtBrick APT tool is statically compiled Linux 64bit executable currently verified on Ubuntu 18.04. It is provided through a direct download link.

1.3.1. Installing RtBrick APT Tool

Before you install the RtBrick APT Tool, ensure you have installed the following software:

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

```
admin@rtbrick:~$ sudo apt install gnupg
```

- HTTPS for APT, which is used by APT to access repositories via HTTPS

```
admin@rtbrick:~$ sudo apt install apt-transport-https
```

The example below shows the URL of the RtBrick APT tool package.

```
admin@rtbrick:~$ 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 ugo+x /usr/local/bin/rtb-apt
```

1.3.2. Generating a Certificate by Using the RtBrick APT tool

The example below shows how to generate a certificate by using the RtBrick APT tool.

```
admin@rtbrick:~$ sudo rtb-apt auth generate
A new self-signed TLS client certificate has been generated for this system:

Subject: CN=admin@rtbrick,OU=rtb-apt
Valid until: 2022-02-23 13:52:24 +0000 UTC
<...>

Please send the certificate in the PEM encoded format below to your RtBrick
contact or to
support@rtbrick.com:

-----BEGIN CERTIFICATE-----
MIIGETCCA/mgAwIBAgIQURwHovZ71kn/4DtJa+MF6jANBgkqhkiG9w0BAQsFADAs
MRAwDgYDVQQLEwdydGItYXB0MRgwFgYDVQQDDA9tYXJ0aW5AdmJveC12bTEwHhcN
MjEwMjI0MTM1MjI0WhcNMjIwMjIzMTM1MjI0WjAsMRAwDgYDVQQLEwdydGItYXB0
MRgwFgYDVQQDDA9tYXJ0aW5AdmJveC12bTEwggIiMA0GCSqGSIb3DQEBAQUAA4IC
DwAwggIKAoICAQC2V7PxE6ZjEkSKkmQClS4Wo/1kNsbIJpYae4wRIPp6NHD3EibW
KIJgNt7bIDIoFsDtL02CjgyxCPPU87d/1jBPGBD+zp+QNOAyHtcYvKgURj63EYe7
gpQrYHXTYdMd+gxbHgjSV/Kj6iSubL6C2Xnu5GYfcgRoLhPtEJ5dYscyu2LW3uW4
<...>
-----END CERTIFICATE-----
```

1.3.3. Identifying and Adding Package Repositories

1.3.3.1. Find available repositories

The example below shows how to find the available repositories.

```
admin@rtbrick:~$ rtb-apt repo list
Group Repository Distribution Release Active Restricted
releases/latest rtbrick-tools ubuntu bionic No No
releases/latest rtbrick-canary ubuntu bionic No No
releases/latest rtbrick-rbfs ubuntu bionic No No
releases/latest rtbrick-onl debian stretch No No
```

1.3.3.2. Activate repository

The example below shows how to activate the repositories.

```
admin@rtbrick:~$ sudo rtb-apt repo activate releases/latest/rtbrick-tools
admin@rtbrick:~$ sudo rtb-apt repo activate releases/latest/rtbrick-canary
```

The activated repositories are added to */etc/apt/sources.list.d/rtbrick.list.*

1.3.3.3. Verify active repositories

The example below shows how to verify the active repositories.

```
admin@rtbrick:~$ rtb-apt repo list
Group Repository Distribution Release Active Restricted
releases/latest rtbrick-tools ubuntu bionic Yes No
releases/latest rtbrick-canary ubuntu bionic Yes Yes
releases/latest rtbrick-rbfs ubuntu bionic No No
releases/latest rtbrick-onl debian stretch No No
```

1.3.4. Verifying Authentication for Package Repositories

- Firstly, your certificate needs to be added to trusted list by RtBrick
- Verify access to RtBrick package repositories
  - If your certificate is accepted, you can use APT to download packages

```
admin@rtbrick:~$ sudo rtb-apt auth check
Repository: releases/latest/rtbrick-tools ... not restricted
Repository: releases/latest/rtbrick-canary ... restricted ... TLS client
certificate accepted
```

1.3.5. Installing RtBrick Image Tool

The RtBrick image tool (rtb-image) uses the certificate generated by APT tool. It requires version 1.3.0. It can be installed from (restricted) the tools repository.

You need to activate the tools repository by entering the following command:

```
admin@rtbrick:~$ sudo rtb-apt repo activate releases/latest/rtbrick-tools
```

1.3.6. Install or upgrade RtB image tool

To install the RtB image tool, enter the following command:

```
admin@rtbrick:~$ sudo apt update
admin@rtbrick:~$ sudo apt install rtbrick-imgstore
```

1.3.7. Verifying Authentication for Image Stores

1.3.7.1. View available image stores

The example below shows how to view the available image stores.

```
admin@rtbrick:~$ sudo rtb-image stores list
Index UUID Name RemoteURL Active Restricted
0 af73c0a6-40e7-4775-b7... latest https://releases.rtbrick.com/_/im... Yes No
1 0879fdac-9dd2-4faf-9c... canary https://releases.rtbrick.com/_/im... No Yes
2 20.11.1 https://releases.rtbrick.com/_/im... No No
3 21.1.1 https://releases.rtbrick.com/_/im... No No
4 21.2.1 https://releases.rtbrick.com/_/im... No No
5 21.3.1 https://releases.rtbrick.com/_/im... No No
<...>
```

1.3.7.2. Activate (restricted) image store

The example below shows how to activate (restricted) image store.

```
admin@rtbrick:~$ sudo rtb-image stores activate 1
```

1.3.7.3. Verify access to image stores

If your certificate is accepted, you can use rtb-image to download images.

The example below shows how to verify the access to the image stores.

```
admin@rtbrick:~$ sudo rtb-image auth check
Image store: canary (0879fdac-9dd2-4faf-9c32-310804d9ea57) ... restricted ...
TLS client certificate
accepted
```

1.4. RtBrick Tools and Packages

The RtBrick tools 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*.

1.4.1. rtbrick-toolkit

1.4.1.1. Version 21.3.1

The rtbrick-toolkit package has been updated to version 21.3.1 to match the corresponding RBFS release and has been updated to depend on the following RtBrick tools packages with these exact versions:

- rtbrick-imgstore 1.3.0
- rtbrick-ansible 4.1.0
- rtbrick-apigwd 0.9.10

- rtbrick-ctrld 0.9.13
- rtbrick-lxcd 0.0.2
- rtbrick-robot-infrastructure 1.7.0

1.4.2. rtb-image version 1.3.0

The rtb-image command has been updated to correctly work with restricted *image stores* on https://releases.rtbrick.com through the use of TLS client certificates. rtb-image can generate a new TLS client certificate if none is available or it can use the TLS client certificate generated by rtb-apt.

1.4.3. rtb-ansible version 4.2.0

This new **major** version of rtb-ansible adds support for per-container RBFS CLI2 (Yang-based) configuration files. At the same time it drops support for old-style YAML container app configurations. If you are currently using rtb-ansible with topologies which include YAML container app configurations you need to migrate those to RBFS CLI2 configurations before using rtb-ansible version 4.2.0.

Please consult the *RtBrick Automation Using Ansible Manual* for more details about rtb-ansible version 4.2.0 and the container configuration migration.

1.5. RtBrick Tools Installation

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

The following commands and outputs are validated only for the Ubuntu 18.04 LTS Bionic Beaver release.

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 some the RtBrick tools packages are already installed it might be necessary to remove the currently installed versions:

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

Among other output, you will get the following:

The following packages will be REMOVED:

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

Step 2: Please use rtb-apt to configure debian package repositories

Step 3: Update the local apt package cache

We then have to update the local apt package cache: sudo apt update

Step 5: Install 3rd-party dependencies

Some RtBrick tools packages might have dependencies on 3rd-party software which cannot be delivered thought the RtBrick package repositories.

Currently the rtbrick-ansible package depends on Ansible. For installing Ansible, you can use the official documentation, which can be found at https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#installing-ansible-on-ubuntu.

One very important dependency of *rtbrick-ansible* is Ansible itself. **Make sure you have the latest version of Ansible installed, before trying to install rtbrick-ansible!**

Step 6: Install a specific RtBrick tool package

For example, in order to install the rtbrick-ansible package, if the steps above have been completed successfully is it sufficient to run the following command:

sudo apt install rtbrick-ansible

1.6. RtBrick tools packages

1.6.1. rtbrick-toolkit

The rtbrick-toolkit is a meta package which can be used to install all the tools needed to work with RBFS images (container or ONL installer) and with the RBFS API in one command:

sudo apt install rtbrick-toolkit

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

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

If only part of the functionality is required each package can be installed individually.

1.6.2. rtbrick-ansible

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

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

```
sudo apt install rtbrick-ansible
```

More information about rtb-ansible and how to use it is available in the **RtBrick Automation Using Ansible** manual.

1.6.3. rtbrick-imgstored

This package provides the rtb-image CLI utility which is RtBrick's image store handling tool. An image store (imgstore) is a versioned, checksumed and cryptographically signed store of versioned files. It was 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 files.

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, which can later be used to start LXC containers running RBFS.

```
pinky@tattooine:~$ sudo apt search rtbrick-imgstore
Sorting... Done
Full Text Search... Done
rtbrick-imgstore/bionic,now 0.4.1 amd64 [installed,automatic]
  RtBrick image store handling tool

pinky@tattooine:~$ sudo apt show rtbrick-imgstore
Package: rtbrick-imgstore
Version: 0.4.1
Priority: extra
Section: rtbrick-internal
Maintainer: RtBrick Support <support@rtbrick.com>
Installed-Size: 24.1 MB
Provides: rtbrick-imgstore
Depends: liblxc-common, liblxc1, lxc, zstd
Replaces: rtbrick-imgstore
Download-Size: 8786 kB
APT-Manual-Installed: no
APT-Sources: http://releases.rtbrick.com/_/20.6.1-rc0/ubuntu/rtbrick-tools
bionic/rtbrick-tools amd64 Packages
Description: RtBrick image store handling tool
 rtbrick_package_properties:
 version: 0.4.1
 branch: master
 commit: 1b14aa3e49b5b35a41899e20f73340b9d34b780d
 commit_timestamp: 1584356254
 commit_date: 2020-03-16 10:57:34 UTC
 build_timestamp: 1584356367
 build_date: 2020-03-16 10:59:27 UTC
 build_job_hash: 423be4f25ec9
 git_dependencies:
 - git_dep: gopackages/imgstore @ master > imgstore
 git_dep_branch: master
 git_dep_commit: 7f0eac0104646c4d067d3849513d4f75364455a8
```

The tool (the binary) has in it embedded the GPG public key of support@rtbrick.com , identity which is used to sign all RtBrick images and the image store itself.

1.6.3.1. Common usage of rtb-image

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

- containers list - List all the LXC containers which are created on the **local** system.
- show [<flags>] <UUID> - Show details of 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 must not be already created.
- list [<flags>] <UUID> - List all the images in the store. By default this lists in the images in the local cached copy of the store.

Table 1. 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 and leaf.
-p, --platform=PLATFORM	List only images for a specific platform.
-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 the l newest images.

An important part of rtb-image is that it is used to create a local cache of the remote RtBrick image repo. This is done using the rtb-image update command:

```
sudo rtb-image update
2020/03/16 13:49:54 [DEBUG] GET http://releases.rtbrick.com/_/images/20.6.1-
rc0/index.sha512
2020/03/16 13:49:54 [DEBUG] GET http://releases.rtbrick.com/_/images/20.6.1-
rc0/index.asc
2020/03/16 13:49:54 [DEBUG] GET http://releases.rtbrick.com/_/images/20.6.1-
rc0/index
Local cached copy updated to: Store: /var/cache/rtbrick/imagestore Version:
0.1.4 ValidUntil: 2020-05-17 13:25:24.443775551 +0000 UTC
```

Then we can list the local copies:

```
pinky@tattooine:~$ rtb-image list

Store: /var/cache/rtbrick/imagestore Version: 0.1.4 ValidUntil: 2020-05-17
13:25:24.443775551 +0000 UTC

UUID                                      Version     Filename
Format  Role        Platform    Cached
4838fd65-c4b6-4d05-a372-ac0334f3623b    20.6.1-rc0-rc0  rbfs-cont/rbfs-spine-
virtual-20.6.1-rc0-rc0.tar.zst         lxd     spine       virtual     false
0e2194a9-4cbd-484b-a1a5-4b2c13dc1ccf    20.6.1-rc0-rc0  rbfs-cont/rbfs-
accessleaf-virtual-20.6.1-rc0-rc0.tar.zst    lxd     accessleaf  virtual
false
638a28bb-7ee8-460f-8fe6-9ec8d4337894    20.6.1-rc0-rc0  rbfs-cont/rbfs-spine-
qmx-20.6.1-rc0-rc0.tar.zst             lxd     spine       qmx         false
21ce3b5c-1e18-474a-8456-06e431da158d    20.6.1-rc0-rc0  rbfs-cont/rbfs-
accessleaf-qmx-20.6.1-rc0-rc0.tar.zst        lxd     accessleaf  qmx
false
```

1.7. Image formats and ONL image installation for supported hardware

RtBrick images delivered through the RtBrick image store and the rtb-image utility have 3 main attributes:

- format: This is the file format of 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 mean to be used as containers in a virtualized environment will have format == lxd and platform == virtual.

RtBrick images mean to be installed on supported whitebox switch hardware devices will have 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.

1.7.1. ONL images

ONL images are generally installed using a Zero Touch Provisioning (ZTP) server. 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 download will look as in the following snippet:

```
pinky@tattooine$ sudo rtb-image update
2020/03/17 07:06:41 [DEBUG] GET http://releases.rtbrick.com/_/images/20.6.1-
rc0/index.sha512
2020/03/17 07:06:42 [DEBUG] GET http://releases.rtbrick.com/_/images/20.6.1-
rc0/index.asc
2020/03/17 07:06:42 [DEBUG] GET http://releases.rtbrick.com/_/images/20.6.1-
rc0/index
Local cached copy already up to date: Store: /var/cache/rtbrick/imagestore
Version: 0.1.10 ValidUntil: 2020-05-17 18:27:28.624270218 +0000 UTC

$ rtb-image list --format onl-installer --platform qmx --role spine --ver
-range latest

Store: /var/cache/rtbrick/imagestore Version: 0.1.10 ValidUntil: 2020-05-17
18:27:28.624270218 +0000 UTC

UUID                                   Version     Filename
Format       Role Platform Cached
c23c4095-5b16-4535-9786-16436a0273d3 20.6.1-rc0-rc0.1 rtbrick-onl-
installer/rtbrick-onl-installer-spine-qmx-20... onl-installer spine qmx
false

pinky@tattooine$ sudo rtb-image pull c23c4095-5b16-4535-9786-16436a0273d3
2020/03/17 07:07:09 [DEBUG] GET http://releases.rtbrick.com/_/images/20.6.1-
rc0/index.sha512
2020/03/17 07:07:09 [DEBUG] GET http://releases.rtbrick.com/_/images/20.6.1-
rc0/index.asc
2020/03/17 07:07:09 [DEBUG] GET http://releases.rtbrick.com/_/images/20.6.1-
rc0/index
rtbrick-onl-installer-spine-qmx-20.6.1-rc0-rc0.1.sha512 207 B / 207 B
[==============================================================================
==============] 100.00% 0s
rtbrick-onl-installer-spine-qmx-20.6.1-rc0-rc0.1.asc 833 B / 833 B
[==============================================================================
=================] 100.00% 0s
rtbrick-onl-installer-spine-qmx-20.6.1-rc0-rc0.1 1.53 GiB / 1.53 GiB
[==============================================================================
===============] 100.00% 23s
rtbrick-onl-installer-spine-qmx-20.6.1-rc0-rc0.1: decompressing 100 B / 100 B
[==============================================================================
======] 100.00% 0s

pinky@tattooine$ rtb-image show c23c4095-5b16-4535-9786-16436a0273d3

Store: /var/cache/rtbrick/imagestore Version: 0.1.10 ValidUntil: 2020-05-17
18:27:28.624270218 +0000 UTC

UUID:         c23c4095-5b16-4535-9786-16436a0273d3
Version:       20.6.1-rc0-rc0.1
Filename:     rtbrick-onl-installer/rtbrick-onl-installer-spine-qmx-20.6.1-
rc0-rc0.1
FullPath/URL: /var/cache/rtbrick/imagestore/rtbrick-onl-installer/rtbrick-
```

```
onl-installer-spine-qmx-20.6.1-rc0-rc0.1
SHA512:
d4d7dfa52bfb644914a4e83d40683503cd77076df44316eeee5ed23fe7d72840abff716909ca8
d29b9fbc7dc8defcd95d50d60fd075352a945a56e14dc25d91a
Format:       onl-installer
Role:         spine
Platform:     qmx
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, we 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.

1.8. The rtb-ssh CLI command

rtb-ssh is a script meant to ease connecting into an already running container. It was previously called **rssh** , and it was renamed, as it was causing confusion with Linux's restricted shell *rssh* package which is available in the official Ubuntu apt package repositories.

Besides renaming only minor some 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 create 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 clear the environment before attaching, so no undesired environment variables leak into the container. The variable container=lxc will be 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, etc.

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

2. RBFS Licensing

2.1. 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.

2.2. Obtaining Licenses

To obtain RBFS licenses, get in touch with the Support team by email: support@rtbrick.com

2.3. 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/XiFD
FRzAtXPUOaIoh5GqiXa+kOJBWp3LgDeJooVrl88mpPs2ZRMPC+k5HvoZDXvsreqRrqoFR3vk7S2Pl
qmLxYf0bNBly4d1hrloBwwFkFuJaiU/M+ZGPExgILdVyXumI88VYx8m/Z5SxEj0bFQGUy8UHRUYW/
Ay8fhPfYejWuSgpv3OrIThH9CVjlDmrp/k4yOuHyTz5gLgq4A0h33vB5O99aOIJW5UX4XDKvQqmqX
5kytRlR1SseWuAbWKjUdVOkf2Mk36IbF9/xAKier++LzXESpLMI+MT63AybSDHOBZydoMjLH9C6cP
EfGHzWTIBNtT3679Tokf25EK1Jw==”
```

The following example shows the running configuration.

```
supervisor@rtbrick: cfg> show config system
{
    "rtbrick-config:system": {
        "license": [
            {
                "license-key":
"eyJzdGFydF9kYXRlIjogMTYxNTg3MTE3MCwgImVuZF9kYXRlIjogMTYxNTk1NzU3MH0=.Yx/XiFD
FRzAtXPUOaIoh5GqiXa+kOJBWp3LgDeJooVrl88mpPs2ZRMPC+k5HvoZDXvsreqRrqoFR3vk7S2Pl
qmLxYf0bNBly4d1hrloBwwFkFuJaiU/M+ZGPExgILdVyXumI88VYx8m/Z5SxEj0bFQGUy8UHRUYW/
Ay8fhPfYejWuSgpv3OrIThH9CVjlDmrp/k4yOuHyTz5gLgq4A0h33vB5O99aOIJW5UX4XDKvQqmqX
5kytRlR1SseWuAbWKjUdVOkf2Mk36IbF9/xAKier++LzXESpLMI+MT63AybSDHOBZydoMjLH9C6cP
EfGHzWTIBNtT3679Tokf25EK1Jw=="
            }
        ]
    }
}
```

2.4. 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.

2.5. 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
```

2.6. 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/XiFD
FRzAtXPUOaIoh5GqiXa+kOJBWp3LgDeJooVrl88mpPs2ZRMPC+k5HvoZDXvsreqRrqoFR3vk7S2Pl
qmLxYf0bNBly4d1hrloBwwFkFuJaiU/M+ZGPExgILdVyXumI88VYx8m/Z5SxEj0bFQGUy8UHRUYW/
Ay8fhPfYejWuSgpv3OrIThH9CVjlDmrp/k4yOuHyTz5gLgq4A0h33vB5O99aOIJW5UX4XDKvQqmqX
5kytRlR1SseWuAbWKjUdVOkf2Mk36IbF9/xAKier++LzXESpLMI+MT63AybSDHOBZydoMjLH9C6cP
EfGHzWTIBNtT3679Tokf25EK1Jw==”
```

2.7. License Expiry

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