



RtBrick Tools Installation Guide

Version 20.8.1, 10 August 2020

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

©Copyright 2020 RtBrick, Inc. All rights reserved. The information contained herein is subject to change without notice. The trademarks, logos and service marks ("Marks") displayed in this documentation are the property of RtBrick in the United States and other countries. Use of the Marks are subject to RtBrick's Term of Use Policy, available at <https://www.rtbrick.com/privacy>. Use of marks belonging to other parties is for informational purposes only.

Table of Contents

1. RtBrick Tools Installation Guide	3
1.1. Introduction	3
1.2. What's New in Tools and Packages	4
1.3. Installation	8
1.4. RtBrick tools packages	10
1.4.1. rtbrick-toolkit	10
1.4.2. rtbrick-ansible	10
1.4.3. rtbrick-imgstored	11
1.4.3.1. Common usage of rtb-image	12
1.5. Image formats and ONL image installation for supported hardware	14
1.5.1. ONL images	14
1.6. The rtb-ssh CLI command	16

1. RtBrick Tools Installation Guide

Version History

Version	Date	Description
1.0	16.03.2020	Initial version
20.4.1-rc0	01.04.2020	The following changes were made: <ul style="list-style-type: none"> • Updated <code>configuration.yaml</code> section, as a result of the release of <code>rtb-ansible 3.0.0</code>. • Added the <i>Connecting a container to outside world</i> section, with examples
20.5.1-rc0	25.05.2020	Updated the document with the latest version of the image, which is <code>20.5.1-rc0</code>
20.6.1-rc0	15.06.2020	Updated the document with the latest version of the image, which is <code>20.6.1-rc0</code>
20.7.1	15.07.2020	Updated the document with details corresponding to the <code>20.7.1</code> RBFS release and details about new and updated RtBrick tools packages.
20.7.1.2	20.07.2020	Version update for the <code>rtbrick-toolkit</code> package to match the corresponding RBFS release.
20.7.2	31.07.2020	Version updates for the <code>rtbrick-toolkit</code> and <code>rtbrick-imgstore</code> packages.
20.8.1	10.08.2020	Version updates for the <code>rtbrick-toolkit</code> and <code>rtbrick-imgstore</code> packages.

1.1. Introduction

RtBrick software is delivered via different means: RBFS (RtBrick Full Stack) software is delivered as custom RtBrick container images (to be used for virtual topologies on x86 servers) or as custom RtBrick ONL installer images (to be used on supported whitebox switches).

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). 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.

1.2. What's New in 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.

rtbrick-toolkit

Version 20.8.1

What's new in **rtbrick-toolkit** version 20.8.1

The **rtbrick-toolkit** package has been updated to version **20.8.1** to match the corresponding RBFS release and has been updated to depend on the following RtBrick tools packages with these exact versions:

- rtbrick-imgstore **0.6.3**
- rtbrick-ansible **3.1.0**
- rtbrick-apigwd **0.9.7**
- rtbrick-ctrlld **0.9.9**

Version 20.7.2

What's new in **rtbrick-toolkit** version 20.7.2

The **rtbrick-toolkit** package has been updated to version **20.7.2** to match the corresponding RBFS release and has been updated to depend on the following RtBrick tools packages with these exact versions:

- rtbrick-imgstore **0.6.3**
- rtbrick-ansible **3.1.0**
- rtbrick-apigwd **0.9.7**
- rtbrick-ctrlld **0.9.9**

Version 20.7.1.3

Version update for the **rtbrick-toolkit** package to match the corresponding RBFS release. No changes to the package dependencies as compared to **20.7.1.2**.

What's new in **rtbrick-toolkit** version 20.7.1.3

Version 20.7.1.2

What's new in **rtbrick-toolkit** version 20.7.1.2

The **rtbrick-toolkit** package has been updated to version **20.7.1.2** to match the corresponding RBFS release and has been updated to depend on the following RtBrick tools packages with these exact versions:

- rtbrick-imgstore **0.6.2**
- rtbrick-ansible **3.1.0**
- rtbrick-apigwd **0.9.7**
- rtbrick-ctrlld **0.9.9**



Please check the information about the **rtbrick-imgstore** package version **0.6.2**.

Version 20.7.1

The **rtbrick-toolkit** package was created with a version of **20.7.1** to match the corresponding RBFS release. The **rtbrick-toolkit** package depends on the following RtBrick tools packages with these exact versions:

- rtbrick-imgstore **0.6.0**
- rtbrick-ansible **3.1.0**
- rtbrick-apigwd **0.9.7**
- rtbrick-ctrlld **0.9.9**

More information about the **rtbrick-toolkit** package can be found in the *RtBrick Tools Installation Manual*.

rtbrick-imgstore

Version 0.6.3

What's new in **rtb-image** version 0.6.3

The **rtb-image** tool contained in the **rtbrick-imgstore** package has been updated with an improvement for handling the situation when a download (**pull**) operation is requested multiple times concurrently for the same image. In such a situation the downloaded image might become corrupted. Starting with version **0.6.3** a lock file is used to prevent multiple concurrent downloads of the same image.

Version 0.6.2

What's new in **rtb-image** version 0.6.2

The **rtb-image** tool contained in the **rtbrick-imgstore** package has been updated to be able to work with image stores which contain images with *post-release* versions.

Post-release versions (sometimes called *bugfix* versions or *maintenance* versions) are composed of 4 numbers separated by dots. For example for the final-release of **20.7.1** the *bugfix* releases will be versioned **20.7.1.2**, and so on.

If an image store contains images with *post-release* versions older versions of **rtb-image** will report an error and will not be able to use any images from that particular image store therefore upgrading **rtb-image** (upgrading the **rtbrick-imgstore** package) to version **0.6.2** is mandatory.

Version 0.6.0

What's new in **rtb-image** version 0.6.0

The **rtb-image** tool contained in the **rtbrick-imgstore** package has been updated to work with multiple image stores. Additional CLI sub-commands have been added under the **rtb-image stores** hierarchy to facilitate this. Please consult the tool's help function with **rtb-image --help-long** to learn more about this.

rtbrick-ansible

Version 3.1.0

What's new in **rtb-ansible** version 3.1.0

Several changes and improvements have been made to the **rtb-ansible** CLI options and to the **configuration.yaml** now expanded and renamed to **topology.yaml**:

- Major changes in **topology.yaml** (old **configuration.yaml**) configuration options (like **linux_connections**, **rbfs_connections**), and used files.
- Added support for running on remote hosts.
- Added support for totally cleaning a topology (**destroy-topology** option).
- Removed dependency on then older **rtbrick-lxc-tools** package.
- Added 2 example virtual RBFS container topologies within the installation package
- Changed the **rtb-ansible** CLI output for running tasks: cleaned and in normal mode the output is more terse and with less debug information.
- Removed old/unmaintained/unused options (like **lazy-reboot-containers**).
- Removed support for unmaintained tasks (like static route programming).

For further details please see the *RtBrick Automation using Ansible* manual.

Version 3.0.0

What's new in **rtb-ansible** version 3.0.0

Starting with version 3.0.0, **rtb-ansible** moves well away from the 2.0.x versions, and substantially modifies the per-container **extra_veth** section of **configuration.yaml** file. The following are modified/added:

- Support for physical host interfaces, inside a container, without using DPDK
- Updated keys (options) in the per-container **extra_veth**



These updates **are not** backwards compatible!

Support for physical host interfaces inside a container

This changes addresses usecases when the user wants traffic generated by a container to be passed to some other host inside the physical network (e.g. for traffic captures).

Updated keys (options) in the per-container **extra_veth**

Old options that could be passed to **extra_veth** were unintuitive, so more user-friendly values were chosen. The changes are the following:

Table 1. New **extra_veth** keys

Usage	Old option	Replaced by:
interface name inside the container	name:	cont_intf:
corresponding interface name on the host (outside the container)	vethname:	host_intf:
the bridge interface to which the container (via host_intf) is connected (thus this needs to be the same for each 2 containers that you want to connect)	link:	bridge_intf:

Version check and upgrading to version 3.0.0

To see the current **rtb-ansible** version, run the following:


```
pinky@Tattooine:$ sudo apt list --installed | grep rtbrick-ansible

WARNING: apt does not have a stable CLI interface. Use with caution in
scripts.

rtbrick-ansible/bionic,now 2.0.0-
private.20200301171753+Bswitchtortbimgstore.C026ebdc-bionic amd64
[installed,upgradable to: 3.0.0-rc0-bionic]
```



rtb-ansible command comes with the **rtbrick-ansible** package.

To upgrade to the version 3.0.0 use the following commands:

```
sudo apt update
sudo -H apt install rtbrick-ansible
```

The same command set should be used for updating to versions newer than 3.0.0 too.

Limitations and restrictions

There are no known limitations and restrictions.

1.3. Installation

The installation of RtBrick tools is broken 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: Add the new RtBrick GPG signing key

The RtBrick Debian (apt) package repositories are always signed with the **RtBrick Support (Package and Image Signing Key)** <support@rtbrick.com> GPG key. If not already present it is required to add this public key to the local apt configuration.



Notice the URL called in the command: it should always be <https://releases.rtbrick.com/>

```
curl -fsSL https://releases.rtbrick.com/security/RtBrick-Support.pubkey.asc | sudo apt-key add -
```

Step 3: Adding the correct RtBrick repositories

Remember that in Step 1 we removed the old packages; the same action might be necessary for the apt repository URLs, if the local configuration includes old entires. Existing RtBrick apt repository URLs can be deleted with the command below, as well as adding the new ones in the same step:

```
echo 'deb [ arch=amd64 ] https://releases.rtbrick.com/_/latest/ubuntu/rtbrick-tools bionic rtbrick-tools' | sudo tee /etc/apt/sources.list.d/rtbrick.list
```

The RtBrick tools packages are delivered in the apt repository named **rtbrick-tools**, however this repository is offered multiple times, at different HTTPS locations:

- https://releases.rtbrick.com/_/latest/ubuntu/rtbrick-tools - this version of the repository will be continuously updated as new package versions are released.
- https://releases.rtbrick.com/_/20.6.1/ubuntu/rtbrick-tools - a version of the repository tied to a specific RBFS release for situation when it is desirable to continue using the RtBrick tools package versions releases together with that RBFS release. The packages in such a repository will only receive bugfix and security updates. The example above is the **rtbrick-tools** repository tied to the **20.6.1** RBFS release.
- Other examples are:
 - https://releases.rtbrick.com/_/20.7.1/ubuntu/rtbrick-tools - tied to **20.7.1** RBFS release
 - https://releases.rtbrick.com/_/20.7.2/ubuntu/rtbrick-tools - tied to **20.7.2** RBFS release
 - https://releases.rtbrick.com/_/20.8.1/ubuntu/rtbrick-tools - tied to **20.8.1** RBFS release

Step 4: 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 through 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 it is sufficient to run the following command:

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

1.4. RtBrick tools packages

1.4.1. 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-ctrlld`

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

1.4.2. 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.

1.4.3. 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, checksummed 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.

1.4.3.1. 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 2. *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.rtbbrick.com/_/images/20.6.1-rc0/index.sha512
2020/03/16 13:49:54 [DEBUG] GET http://releases.rtbbrick.com/_/images/20.6.1-rc0/index.asc
2020/03/16 13:49:54 [DEBUG] GET http://releases.rtbbrick.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	Format	Role	Platform	Cached	Version	Filename
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-ala5-4b2c13dc1ccf					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		

1.5. 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.

1.5.1. 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.rtbbrick.com/_/images/20.6.1-rc0/index.sha512
2020/03/17 07:06:42 [DEBUG] GET http://releases.rtbbrick.com/_/images/20.6.1-rc0/index.asc
2020/03/17 07:06:42 [DEBUG] GET http://releases.rtbbrick.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.rtbbrick.com/_/images/20.6.1-rc0/index.sha512
2020/03/17 07:07:09 [DEBUG] GET http://releases.rtbbrick.com/_/images/20.6.1-rc0/index.asc
2020/03/17 07:07:09 [DEBUG] GET http://releases.rtbbrick.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.

1.6. 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** .