openstack-helm-images Documentation

Release 0.0.1.dev663

OpenStack Developers

Feb 05, 2026

CONTENTS:

This repository is in charge of the image building for openstack-helm repositories.

Please check the documentation of each section for the relevant build instructions.

By default, these images are built on a Ubuntu 18.04 LTS node.

CHAPTER

ONE

SETUP A BUILD NODE

Here are the instructions to setup a build node with Ubuntu 18.04 LTS:

```
apt update
apt install -y docker.io git
```

CHAPTER

TWO

MODIFYING THE BUILD WITH ENVIRONMENT

Unless explicitly written, all the *build.sh* convenience scripts allow to pass arguments to the docker build process: The *build.sh* scripts have a environment variable (*extra_build_args*), which can be used to pass arbitrary data.

Next to the extra arguments, you can modify the *build.sh* behavior by setting the following environment variables:

```
VERSION
DISTRO
REGISTRY_URI=${REGISTRY_URI:-"openstackhelm/"}
```

VERSION is the expected tag version of the image, and defaults to *latest*

DISTRO is used if you want to build an image with a different Dockerfile, for example with another distribution. *Dockerfile.${DISTRO}* must match an existing filename.

REGISTRY_URI is part of the image name, representing the location of the image, used in the image tagging process. For example *REGISTRY_URI* could be *docker.io/openstackhelm/*. In that case, the full name and tag of the *vbmc* image would be:

```
docker.io/openstackhelm/vbmc:latest
```

Please check each section of the documentation for an overview of the build process for each container.

2.1 So You Want to Contribute

For general information on contributing to OpenStack, please check out the contributor guide to get started. It covers all the basics that are common to all OpenStack projects: the accounts you need, the basics of interacting with our Gerrit review system, how we communicate as a community, etc.

Additional information could be found in OpenDev Developers Guide.

Below will cover the more project specific information you need to get started with OpenStack-Helm images.

2.1.1 Communication

- Join us on IRC: #openstack-helm on oftc
- Community IRC Meetings: [Every Tuesday @ 3PM UTC], #openstack-meeting-alt on oftc
- Meeting Agenda Items: Agenda
- Join us on Slack - #openstack-helm

2.1.2 Contacting the Core Team

Projects Core Team could be contacted via IRC or Slack, usually during weekly meetings. List of current Cores could be found on a Members tab of openstack-helm-images-core Gerrit group.

2.1.3 New Feature Planning

New features are planned and implemented trough the process described in Project Specifications section of OpenStack-Helm documents.

2.1.4 Task Tracking

We track our tasks on our StoryBoard.

If youre looking for some smaller, easier work item to pick up and get started on, search for the low-hanging-fruit tag.

Other OpenStack-Helm components tasks could be found on the group Storyboard.

2.1.5 Reporting a Bug

You found an issue and want to make sure we are aware of it? You can do so on our Storyboard.

If issue is on one of other OpenStack-Helm components, report it to the appropriate group Storyboard.

Bugs should be filed as stories in Storyboard, not GitHub.

2.1.6 Getting Your Patch Merged

We require two Code-Review +2s from reviewers, before getting your patch merged with giving Work-force +1. Trivial patches (e.g. typos) could be merged with one Code-Review +2.

Changes affecting code base often require CI tests and documentation to be added in the same patch set.

Pull requests submitted through GitHub will be ignored.

2.1.7 Project Team Lead Duties

All common PTL duties are enumerated in the PTL guide.

2.2 ceph-config-helper container image

This container builds a small image with kubectl and some other utilites for use in the ceph charts or interact with a ceph deployment.

2.2.1 Manual build

Ubuntu Xenial

Here are the instructions for building Xenial image:

```
IMAGE="ceph-config-helper"
VERSION=${VERSION:-latest}
DISTRO=${DISTRO:-ubuntu}
DISTRO_VERSION=${DISTRO_VERSION:-jammy}
REGISTRY_URI=${REGISTRY_URI:-"openstackhelm/"}
EXTRA_TAG_INFO=${EXTRA_TAG_INFO:-""}
```

Alternatively, this step can be performed by running the script directly:

```
./ceph-config-helper/build.sh
```

openSUSE Leap 15

To build an openSUSE leap 15 image, you can export varibles before running the build script:

```
DISTRO=suse_15 ./ceph-config-helper/build.sh
```

2.3 ceph-daemon container image

This container builds a small image with ceph service, kubectl and some other utilities for use in the ceph charts.

2.3.1 Manual build

Ubuntu Xenial

Here are the instructions for building Xenial image:

```
IMAGE="ceph-daemon"
VERSION=${VERSION:-latest}
DISTRO=${DISTRO:-ubuntu}
DISTRO_VERSION=${DISTRO_VERSION:-jammy}
REGISTRY_URI=${REGISTRY_URI:-"openstackhelm/"}
EXTRA_TAG_INFO=${EXTRA_TAG_INFO:-""}
```

Alternatively, this step can be performed by running the script directly:

```
./ceph-daemon/build.sh
```

2.4 `libvirt` container image

This container builds a small image with Libvirt for use with OpenStack-Helm.

If you need to build a `libvirt` image, you can use the `Dockerfile` with the `FROM` build argument set to your source image and the `RELEASE` set to the OpenStack release youre deploying. For example:

```
.. code-block:: shell
```

docker buildx build

build-arg FROM=ubuntu:22.04 build-arg RELEASE=zed libvirt/

You can also use `buildx` to build the image for multiple architectures:

```
.. code-block:: shell
```

docker buildx build

build-arg FROM=ubuntu:22.04 build-arg RELEASE=zed platform linux/amd64,linux/arm64 libvirt/

2.5 MariaDB container image

This image is based on upstream MariaDB image, with extra Kubernetes libraries to work with OpenStack-Helm

2.5.1 Manual build for Ubuntu Xenial

Here are the instructions for building Xenial image:

```
IMAGE="mariadb"
VERSION=${VERSION:-10.5.9-focal}
DISTRO=${DISTRO:-ubuntu}
DISTRO_VERSION=${DISTRO_VERSION:-focal}

REGISTRY_URI=${REGISTRY_URI:-"openstackhelm/"}
```

Alternatively, this step can be performed by running the script directly:

```
./mariadb/build.sh
```

2.6 OpenvSwitch container image

This container builds a small image with OpenvSwitch for use with OpenStack-Helm.

2.6.1 Manual build

Debian

Here are the instructions for building the default Debian image:

```
IMAGE="openvswitch"
VERSION=${VERSION:-latest}
DISTRO=${DISTRO:-ubuntu}
DISTRO_VERSION=${DISTRO_VERSION:-focal}
REGISTRY_URI=${REGISTRY_URI:-"openstackhelm/"}
EXTRA_TAG_INFO=${EXTRA_TAB_INFO:-""}
```

Alternatively, this step can be performed by running the script directly:

```
./openvswitch/build.sh
```

openSUSE Leap 15

To build an openSUSE leap 15 image, you can export varibles before running the build script:

```
DISTRO=suse_15 ./openvswitch/build.sh
```

2.7 Tempest container image

This image is installing tempest with a few tempest plugins from the head of the master branch in OpenStack.

2.7.1 Manual build for Ubuntu Xenial

Here are the instructions for building Xenial image:

```
IMAGE="tempest"
VERSION=${VERSION:-latest}
DISTRO=${DISTRO:-ubuntu_focal}
REGISTRY_URI=${REGISTRY_URI:-"openstackhelm/"}
EXTRA_TAG_INFO=${EXTRA_TAG_INFO:-""}
docker build -f ${IMAGE}/Dockerfile.${DISTRO} --network=host -t ${REGISTRY_URI}${IMAGE}:${VERSION}-${DISTRO}${EXTRA_TAG_INFO} ${extra_build_args} ${IMAGE}
```

Alternatively, this step can be performed by running the script directly:

```
./tempest/build.sh
```

2.8 LOCI based images

OpenStack-Helm requires packages that arent installed in the LOCI images by default.

2.8.1 Mechanism used

Currently, we are passing arguments to the loci build, which is enough to customize the build system.

LOCI build process is a relatively staged process:

1. Build (or re-use) a base image
2. Build a requirements image, building wheels.
3. Build the project image, re-using requirements.

2.8.2 Code and parameters

OpenStack-Helm-Images can build multiple OpenStack images based on LOCI.

By default, OpenStack-Helm-Image has one *build.sh* script, in the *openstack/loci/* folder.

For convenience, default overrides per OpenStack branch are provided in the same folder: *build-newton.sh* builds an OpenStack newton image, *build-ocata.sh* builds an ocata image, and so on.