Red Hat Quay 2.9

Deploy Red Hat Quay - Basic

Deploy Red Hat Quay
Deploy Red Hat Quay
Abstract

Get started with Red Hat Quay
# Table of Contents

- PREFACE ................................................................. 3.
- CHAPTER 1. OVERVIEW .................................................. 4.
- CHAPTER 2. ARCHITECTURE ............................................. 5.
- CHAPTER 3. INSTALLING RED HAT QUAY (BASIC) ..................... 6.
  - 3.1. PREREQUISITES ................................................. 6
  - 3.2. STARTING UP THE SERVICES .................................. 6
- CHAPTER 5. START USING RED HAT QUAY ............................. 14.
  - ADDITIONAL RESOURCES .......................................... 14
Red Hat Quay is an enterprise-quality container registry. Use Quay to build and store containers, then deploy them to the servers across your enterprise.

This procedure describes how to deploy a non-production, test-only Red Hat Quay setup (based on For Testing as a container).
CHAPTER 1. OVERVIEW

Features of Quay include:

- High availability
- Geo-replication
- Continuous integration
- Security scanning
- Custom log rotation
- 24/7 support

Quay provides support for multiple:

- Authentication and access methods
- Storage backends
- SSL certificates
- Application registries
CHAPTER 2. ARCHITECTURE

Quay is made up of three core components for a basic setup. In highly available setups, an additional object storage component is needed. The three core components are:

- **Database (MySQL or PostgreSQL)**: Used by Quay as its primary metadata storage (not for image storage).

- **Redis (key, value store)**: Used for providing real time events and during Quay setup and installation.

- **Quay (container registry)**: Runs Quay as a service, consisting of several components in the pod.

For the high availability installation, you need to use one of the following types of storage:

- **Public cloud storage**: In public cloud environments, you should use the cloud provider’s object storage, such as Amazon S3 (for AWS) or Google Cloud Storage (for Google Cloud).

- **Private cloud storage**: In private clouds, an S3 or Swift compliant Object Store is needed, such as Ceph RADOS, or OpenStack Swift.

Local storage is supported for the Red Hat Quay test-only installation, but not for high-availability.
CHAPTER 3. INSTALLING RED HAT QUAY (BASIC)

NOTE

This installation process is only for POC purposes and is not intended for use as a production install.

3.1. PREREQUISITES

For a Red Hat Quay Registry installation (appropriate for non-production purposes), you need one system (physical or virtual machine) that has the following attributes:

- **Red Hat Enterprise Linux (RHEL):** Obtain the latest Red Hat Enterprise Linux server media from the Downloads page and follow instructions from the Red Hat Enterprise Linux 7 Installation Guide.

- **Valid Red Hat Subscription:** Obtain a valid Red Hat Enterprise Linux server subscription.

- **CPUs:** Two or more virtual CPUs

- **RAM:** 4GB or more

- **Disk space:** (dependant on storage needs for registry)
  - About 30GB of disk space should be enough for a test system (broken down in the following manner):
    - At least 10GB of disk space for the operating system (Red Hat Enterprise Linux Server).
    - At least 10GB of disk space for docker storage (to run 3 containers)
    - At least 10GB of disk space for Quay local storage (CEPH or other local storage might require more memory)

3.2. STARTING UP THE SERVICES

Follow these steps to install Red Hat Quay on a single system (VM or bare metal).

1. **Install Red Hat Enterprise Linux server:** Install the latest RHEL server. You can do a Minimal install (shell access only) or Server plus GUI (if you want a desktop).

2. **Register the System:** Register and subscribe your RHEL server system to Red Hat. See How to register and subscribe a system... for details. The following commands register your system and list available subscriptions. Choose an available RHEL server subscription, attach to its poolid, enable rhel-7-server-rpms and rhel-7-server-extras-rpms repositories, and upgrade to the latest software:

   ```bash
   # subscription-manager register --username=<user_name> --password=<password>
   # subscription-manager refresh
   # subscription-manager list --available
   # subscription-manager attach --pool=<pool_id>
   # subscription-manager repos --disable="*"
   # subscription-manager repos \
   ```
3. **Add Quay authentication**: Set up authentication to Quay.io, so you can pull the Quay container, as described in Accessing Red Hat Quay without a CoreOS login.

4. **Setup Docker**: Install, enable, and start the docker service as shown here (see Getting Docker in RHEL 7 for details):

   ```bash
   # yum install docker
   # systemctl enable docker
   # systemctl start docker
   # systemctl is-active docker
   active
   ```

5. **Install / Deploy a Database**: Choose either MySQL or PostgreSQL as a database. This example shows how to deploy the MySQL database container (see the MySQL section of Using Red Hat Software Collections Container Images for details.) To configure the MySQL database, you can use the values shown here or change any of the following for storing MySQL data (/mnt/hostmysql) and setting database values:

   ```bash
   # mkdir -p /mnt/hostmysql
   # chmod 777 /mnt/hostmysql
   # export MYSQL_CONTAINER_NAME=mysql
   # export MYSQL_DATABASE=enterpriseregistrydb
   # export MYSQL_PASSWORD=JzxCTamgFBmHRhcGToPHFkrx1BH2vWQ
   # export MYSQL_USER=quayuser
   # export MYSQL_ROOT_PASSWORD=L36PrivxRB02bq0B9jtZtWiCccMsApOGn

   # docker run \
   --detach \
   --restart=always \
   --env MYSQL_ROOT_PASSWORD=${MYSQL_ROOT_PASSWORD} \
   --env MYSQL_USER=${MYSQL_USER} \
   --env MYSQL_PASSWORD=${MYSQL_PASSWORD} \
   --env MYSQL_DATABASE=${MYSQL_DATABASE} \
   --name ${MYSQL_CONTAINER_NAME} \
   --publish 3306:3306 \
   -v /mnt/hostmysql:/var/lib/mysql/data:Z \
   registry.access.redhat.com/rhscl/mysql-57-rhel7
   ```

   **NOTE**

   To generate passwords for MySQL user accounts, instead of setting them statically, run the following:

   ```bash
   # export MYSQL_PASSWORD=$(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 32 | sed 1q)
   # export MYSQL_ROOT_PASSWORD=$(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 32 | sed 1q)
   ```

6. **(optional) Check database connectivity**: To check connectivity to the database, you can log in
using the mysql command (from the mariadb package). Substitute the hostname (or IP address) of your MySQL service and your password. Type `status` to see information about your MySQL connection:

```bash
# yum install -y mariadb
# mysql -h 192.168.122.99 -u root --password=L36PrivxR802bq0B9jtZtWiCcMsApOGn
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 10184
Server version: 5.7.21 MySQL Community Server (GPL)
Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MySQL [(none)]> status
------------
mysql  Ver 15.1 Distrib 5.7.21-MariaDB, for Linux (x86_64) using readline 5.1
Connection id:  10184
Current database:  
Current user:  root@192.168.122.99
...
Server version:  5.7.21 MySQL Community Server (GPL)
Protocol version: 10
Connection:  192.168.122.99 via TCP/IP
...
MySQL [(none)]> \q
```

7. **Install / Deploy Redis**: Run Redis as a container:

```bash
# mkdir -p /mnt/hostredis
# chmod 777 /mnt/hostredis
# docker run -d --restart=always -p 6379:6379 \
    -v /mnt/hostredis:/var/lib/redis/data:Z \
    registry.access.redhat.com/rhscl/redis-32-rhel7
```

8. **Check redis connectivity**: You can use the `telnet` command to test connectivity to the redis service. Type MONITOR (to begin monitoring the service) and QUIT to exit:

```bash
# yum install telnet -y
# telnet 192.168.122.99 6379
Trying 192.168.122.99...
Connected to 192.168.122.99.
Escape character is '^]'.
MONITOR
+OK
+1525703165.754099 [0 172.17.0.1:43848] "PING"
QUIT
+OK
Connection closed by foreign host.
```

9. **Install / Deploy Quay**: To run Red Hat Quay as a container, create two directories to store data on the host, then run Red Hat Quay as a container, as follows:
NOTE

Add `-e DEBUGLOG=true` to the `docker run` command line for the quay container to enable debug level logging.

```bash
# mkdir -p /mnt/quay/config
# #optional: if you don't choose to install an Object Store
# mkdir -p /mnt/quay/storage
# docker run --restart=always -p 443:443 -p 80:80 \
#   --privileged=true \n#   -v /mnt/quay/config:/conf/stack \n#   -v /mnt/quay/storage:/datastorage \n#   -d quay.io/coreos/quay:v2.9.5
```

Wait several minutes for the Quay service to come up. Then proceed to Completing the Guided Setup.

NOTE

This can take several minutes, and when attempting to access the Guided Setup you might receive a proxy error. If you do please wait a while longer and try again.
CHAPTER 4. COMPLETING THE GUIDED SETUP

Open a browser to the setup page on the system where you just started quay (for example http://hostname/setup) and complete the following steps:

1. **Identify the database**: Add the following information about the type and location of the database to be used by Quay:
   - **Database Type**: Choose MySQL or PostgreSQL. (MySQL is used in the basic example; PostgreSQL is used with the high availability example.)
   - **Database Server**: Identify the IP address or hostname of the database, along with the port number if it is different from 3306.
   - **Username**: Identify a user with full access to the database.
   - **Password**: Enter the password you assigned to the selected user.
   - **Database Name**: Enter the database name you assigned when you started the database server.
   - **SSL Certificate**: For production environments, you should provide an SSL certificate to connect to the database.

   Figure 1 shows an example of the screen for identifying the database used by Red Hat Quay.

   ![Database Identification Screen](image)

   Select “Validate Database Settings”, and proceed to the next section.

   Figure 2 shows an example of the Red Hat Quay Setup screen as the database schema is set up.

   ![Database Schema Setup Screen](image)
NOTE

At this point a restart of the Quay container should happen. If the container does not restart, the docker restart policy may not be working properly, and a manual restart of the container may be required.

2. **Create Quay superuser**: You need to set up an account with superuser privileges to Quay, to use for editing Quay configuration settings. That information includes a Username, Email address, and Password (entered twice).

Figure 3 shows an example of the Red Hat Quay Setup screen for setting up a Quay superuser account:

Select "Create Super User", and proceed to the next section.

3. **Identify the Redis hostname and add other desired settings**: Other setting you can add to complete the setup are as follows. More settings for high availability Quay deployment that for the basic deployment:

   - For the basic, test configuration, identifying the Redis Hostname should be all you need to do.
   - For the high availability configuration, more settings are needed (as noted below) to allow for shared storage, secure communications between systems, and other features.

Here are the settings you need to consider:

- **Custom SSL Certificates**: Upload custom or self-signed SSL certificates for use by Quay. See Using SSL to protect connections to Red Hat Quay for details. Recommended for high availability.

- **Basic Configuration**: Upload a company logo to rebrand your Quay registry.

- **Server Configuration**: Hostname or IP address to reach the Quay service, along with TLS indication (recommended for production installations). Recommended for high availability.

- **Data Consistency Settings**: Select to relax logging consistency guarantees to improve performance and availability.

- **Time Machine**: Allow older image tags to remain in the repository for set periods of time and allow users to select their own tag expiration times.
• **redis**: Identify the hostname or IP address (and optional password) to connect to the redis service used by Quay.

• **Registry Storage**: Identify the location of storage. A variety of cloud and local storage options are available. Remote storage is required for high availability.

• **Action Log Rotation and Archiving**: Select to enable log rotation, which moves logs older than 30 days into storage, then indicate storage area.

• **Security Scanner**: Enable security scanning by selecting a security scanner endpoint and authentication key. To setup Clair to do image scanning, refer to Clair Setup and Configuring Clair. Recommended for high availability.

• **Application Registry**: Enable an additional application registry that includes things like Kubernetes manifests or Helm charts (see the App Registry specification).

• **BitTorrent-based download**: Allow all registry images to be downloaded using BitTorrent protocol (using quayctl tool).

• **rkt Conversion**: Allow `rkt fetch` to be used to fetch images from Quay registry. Public and private GPG2 keys are needed (see Generating signing keys for ACI conversion for details).

• **E-mail**: Enable e-mail to use for notifications and user password resets.

• **Internal Authentication**: Change default authentication for the registry from Local Database to LDAP, Keystone (OpenStack), JWT Custom Authentication, or External Application Token.

• **External Authorization (OAuth)**: Enable to allow GitHub or GitHub Enterprise to authenticate to the registry.

• **Google Authentication**: Enable to allow Google to authenticate to the registry.

• **Access settings**: Basic username/password authentication is enabled by default. Other authentication types that can be enabled include: external application tokens (user-generated tokens used with docker or rkt commands), anonymous access (enable for public access to anyone who can get to the registry), user creation (let users create their own accounts), encrypted client password (require command-line user access to include encrypted passwords), and prefix username autocompletion (disable to require exact username matches on autocompletion).

• **Dockerfile Build Support**: Enable to allow users to submit Dockerfiles to be built and pushed to Quay. Select “Save Configuration Changes”, then “Save Configuration.

4. **Restart Quay**: When prompted, select “Restart Container” to restart Quay. Figure 4 shows that screen that appears as you wait for Quay to restart.
NOTE

At this point a restart of the Quay container should happen. If the container does not
restart, the docker restart policy may not be working properly, and a manual restart of the
container may be required.
CHAPTER 5. START USING RED HAT QUAY

With Red Hat Quay now running, you can:

- Select Tutorial from the Quay home page to try the 15-minute tutorial. In the tutorial, you learn to log into Quay, start a container, create images, push repositories, view repositories, and change repository permissions with Quay.

- Refer to the Use Red Hat Quay for information on working with Red Hat Quay repositories.

ADDITIONAL RESOURCES