Red Hat Quay 3.2

Deploy Red Hat Quay - Basic

Deploy Red Hat Quay

Last Updated: 2020-04-30

Red Hat Quay 3.2 Deploy Red Hat Quay - Basic

Deploy Red Hat Quay

Legal Notice

Abstract

Get started with Red Hat Quay

Table of Contents

PREFACE

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 Red Hat Quay include:

- High availability
- Geo-replication
- Repository mirroring (Technology Preview in Red Hat Quay v3.1, supported in v3.2)
- Docker v2, schema 2 (multiarch) support
- Continuous integration
- Security scanning with Clair
- Custom log rotation
- Zero downtime garbage collection
- 24/7 support

Red Hat Quay provides support for:

- Multiple authentication and access methods
- Multiple storage backends
- Custom certificates for Quay, Clair, and storage backends
- Application registries
- Different container image types

CHAPTER 2. ARCHITECTURE

Red Hat Quay is made up of several core components.

- **Database**: Used by Red Hat Quay as its primary metadata storage (not for image storage).

- **Redis (key, value store)**: Stores live builder logs and the Red Hat Quay tutorial.

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

- **Clair**: Scans container images for vulnerabilities and suggests fixes.

For supported deployments, 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.

Do not use "Locally mounted directory" Storage Engine for any production configurations. Mounted NFS volumes are not supported. Local storage is meant for Red Hat Quay test-only installations.

CHAPTER 3. PREPARING FOR 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 SUPPORTING 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:

    > NOTE
    >
    > This procedure was tested on RHEL 7. The **docker** command is not included in RHEL 8, so you would need to use the **podman** command instead. Because the **--restart** option is not supported by podman, instead of using **--restart**, you could set up to use **podman** as a systemd service, as described in Starting containers with systemd.

```
# 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 \
    --enable="rhel-7-server-rpms" \
    --enable="rhel-7-server-extras-rpms"
# yum update -y
```

3. **Add Quay.io 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):

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

5. **Open ports in firewall:** If you have a firewall running on your system, to access the Red Hat Quay config tool (port 8443) and application (ports 80 and 443) outside of the local system, run the following commands (add **--zone=<yourzone>** for each command to open ports on a particular zone):

```
# firewall-cmd --permanent --add-port=8443/tcp
# firewall-cmd --permanent --add-port=80/tcp
# firewall-cmd --permanent --add-port=443/tcp
# firewall-cmd --reload
```

6. **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 (/var/lib/mysql) and setting database values:

```
# mkdir -p /var/lib/mysql
# chmod 777 /var/lib/mysql
# export MYSQL_CONTAINER_NAME=mysql
# export MYSQL_DATABASE=enterpriseregistrydb
# export MYSQL_PASSWORD=JzxCTamgFBmHRhcGFtoPHFkrx1BH2vwQ
# export MYSQL_USER=quayuser
# export MYSQL_ROOT_PASSWORD=L36PrivxRB02bqOB9jtZtWiCcMsApOGn

# 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} \
    --privileged=true \
    --publish 3306:3306 \
    -v /var/lib/mysql:/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:

# 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)

7. **(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:

```
# yum install -y mariadb
# mysql -h 192.168.122.99 -u root --password=L36PrivxRB02bqOB9jtZtWiCcMsApOGn
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
```

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

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

9. **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:

```
# 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.
```

With the supporting services running, you can move on to creating configuration files to use with the Red Hat Quay deployment.

CHAPTER 4. CONFIGURING RED HAT QUAY

Before running the Red Hat Quay service as a container, you need to use that same quay container to create the configuration file (**config.yaml**) needed to deploy Red Hat Quay. To do that, you pass a **config** argument and a password (replace my-secret-password here) to the quay container. Later, you use that password to log into the configuration tool as the user **quayconfig**.

Here's an example of how to do that:

1. **Start quay in config mode** On the first quay node, run the following:

   ```
   # docker run --privileged=true -p 8443:8443 -d quay.io/redhat/quay:v3.2.2 config my-secret-password
   ```

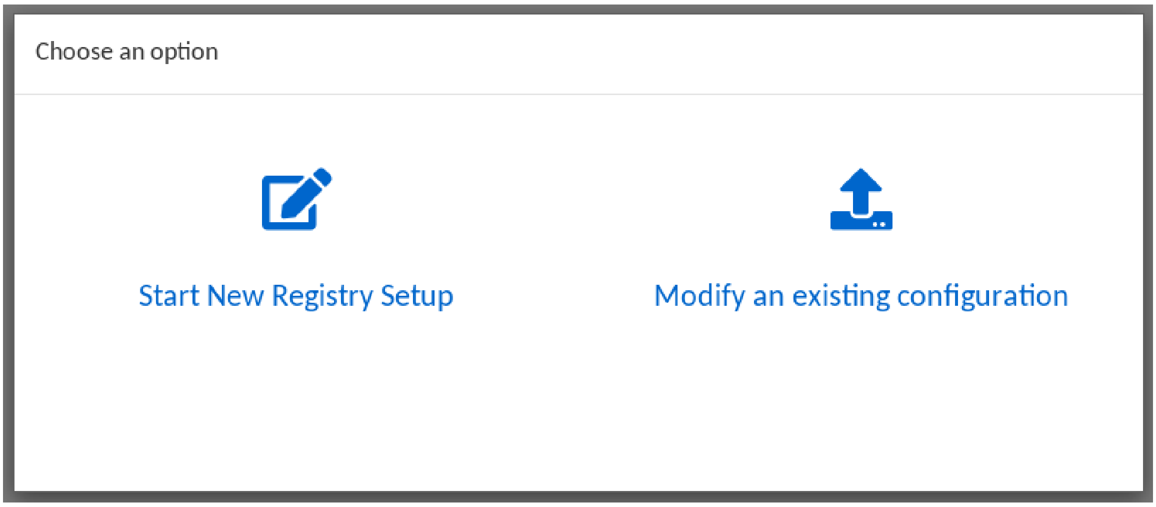
2. **Open browser**: When the quay configuration tool starts up, open a browser to the URL and port 8443 of the system you are running the configuration tool on (for example https://myquay.example.com:8443). You are prompted for a username and password.

3. **Log in as quayconfig** When prompted, enter the **quayconfig** username and password (the one from the **docker run** command line).

4. **Choose configuration mode**: You are prompted to choose to either create a new Red Hat Quay configuration file or edit an existing one in these two modes:

   - **Start New Registry Setup** The result of this selection is the creation of a new configuration file (**config.yaml**) and optional **ssl.cert** and **ssl.key** files. Those files are bundled into a tarball file you can use to actually deploy all your Red Hat Quay nodes.

   - **Modify an existing configuration** With this selection, you are prompted to provide an existing tarball and modify it before you use it to start your Red Hat Quay nodes.
     The following figure shows an example of the resulting **Choose an option** page:

     For an initial setup, you are asked to identify the database type. For a subsequent configuration, you are prompted for a tarball containing the **config.yaml** and credential files (optional). Then you can continue on with the configuration.

5. **Identify the database**: For the initial setup, add the following information about the type and location of the database to be used by Red Hat Quay:

   - **Database Type**: Choose MySQL or PostgreSQL. MySQL will be used in the basic example; PostgreSQL is used with the high availability Red Hat Quay on OpenShift examples.

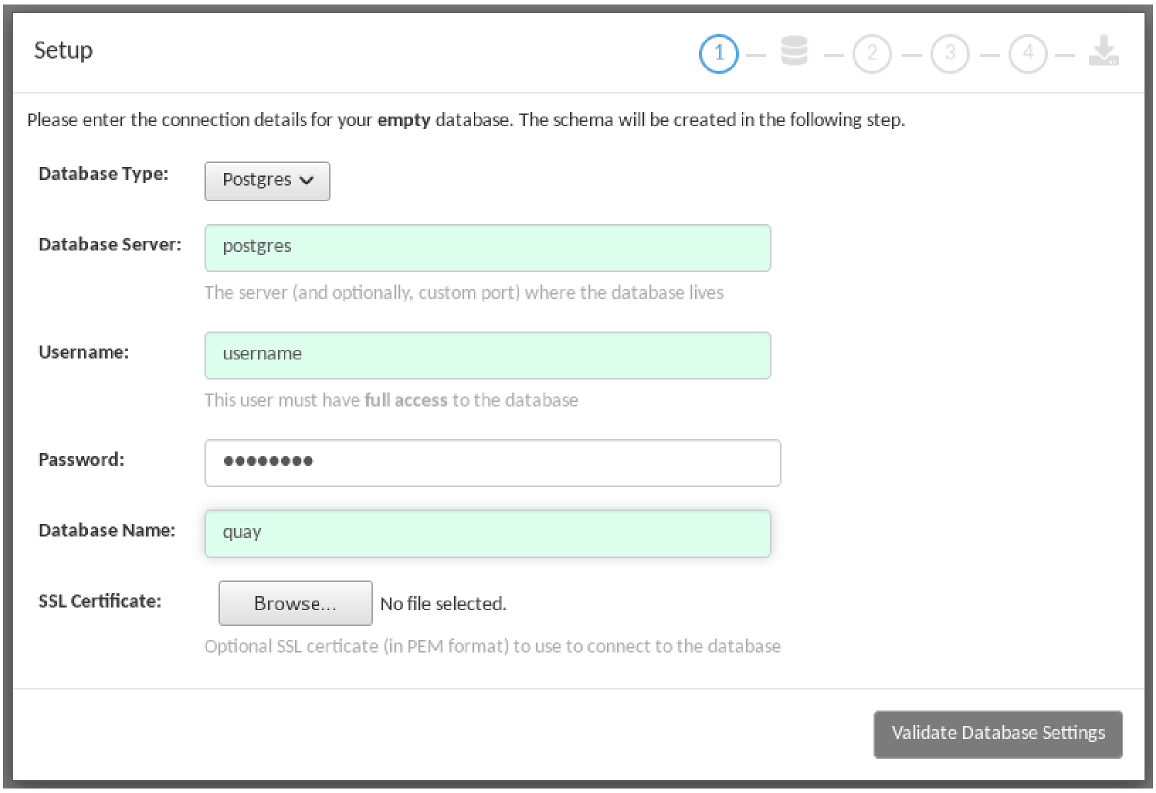
- **Database Server**: Identify the IP address or hostname of the database, along with the port number if it is different from 3306.

- **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.
  The following figure shows an example of the screen for identifying the database used by Red Hat Quay:



Setup

Please enter the connection details for your **empty** database. The schema will be created in the following step.

Database Type: Postgres

Database Server: postgres

The server (and optionally, custom port) where the database lives

Username: username

This user must have **full access** to the database

Password: ••••••••

Database Name: quay

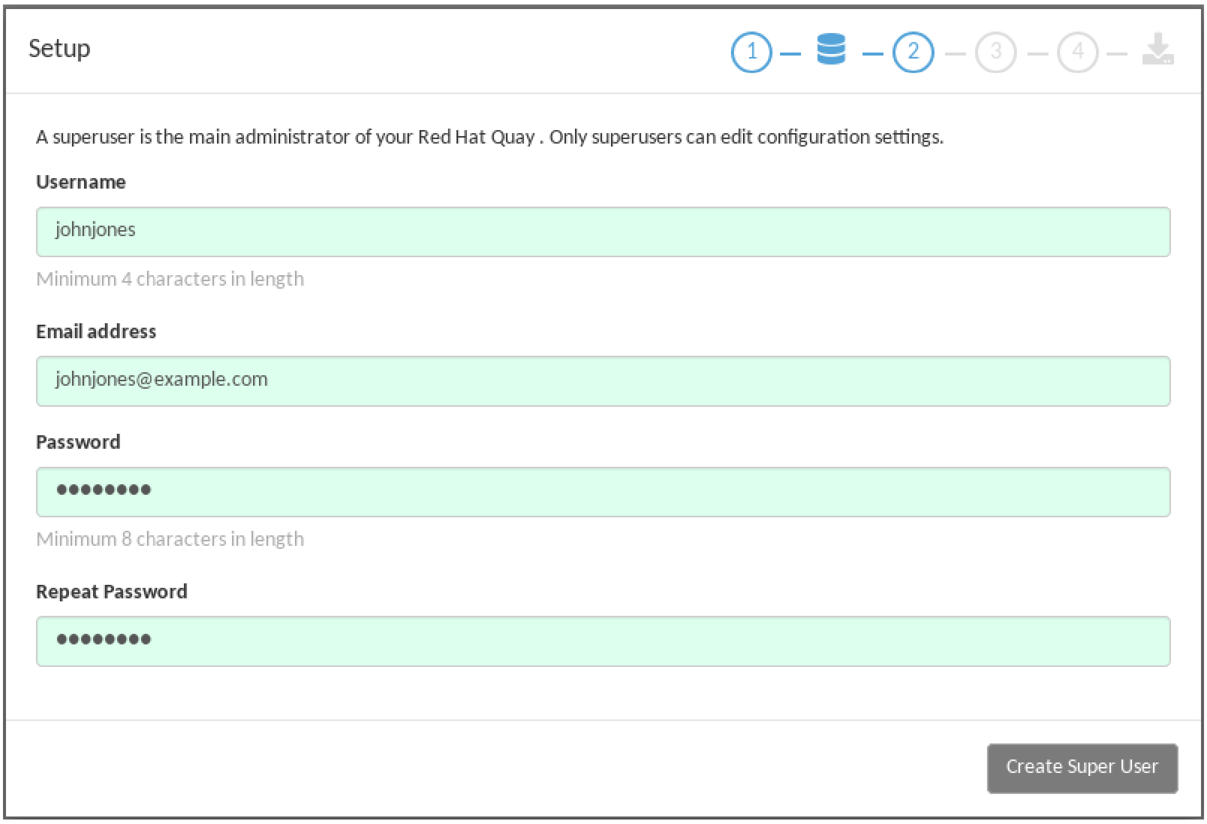
SSL Certificate: Browse... No file selected.

Optional SSL certicate (in PEM format) to use to connect to the database

Validate Database Settings

6. **Validate database**: Select **Validate Database Settings**, and proceed to the next screen.

7. **Create Red Hat Quay superuser** You need to set up an account with superuser privileges to Red Hat Quay, to use for editing Red Hat Quay configuration settings. That information includes a Username, Email address, and Password (entered twice).
   The following figure shows an example of the Red Hat Quay Setup screen for setting up a Red Hat Quay superuser account:



A superuser is the main administrator of your Red Hat Quay . Only superusers can edit configuration settings.

Username

johnjones

Minimum 4 characters in length

Email address

johnjones@example.com

Password

••••••••

Minimum 8 characters in length

Repeat Password

••••••••

Select **Create Super User**, and proceed to the next screen.

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

    - For the basic, test configuration, identifying the Redis Hostname should be all you need to do. However, you can add other features, such as Clair Scanning and Repository Mirroring, as described at the end of this procedure.

    - For the high availability and OpenShift configurations, 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 Red Hat Quay. See Using SSL to protect connections to Red Hat Quay for details. Recommended for high availability.

      IMPORTANT

      Using SSL certificates is recommended for both basic and high availability deployments. If you decide to not use SSL, you must configure your container clients to use your new Red Hat Quay setup as an insecure registry as described in Test an Insecure Registry .

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

    - **Server Configuration**: Hostname or IP address to reach the Red Hat Quay service, along with TLS indication (recommended for production installations). The Server Hostname is required for all Red Hat Quay deployments. TLS termination can be done in two different ways:

  - On the instance itself, with all TLS traffic governed by the nginx server in the quay container (recommended).
  - On the load balancer. This is not recommended. Access to Red Hat Quay could be lost if the TLS setup is not done correctly on the load balancer.

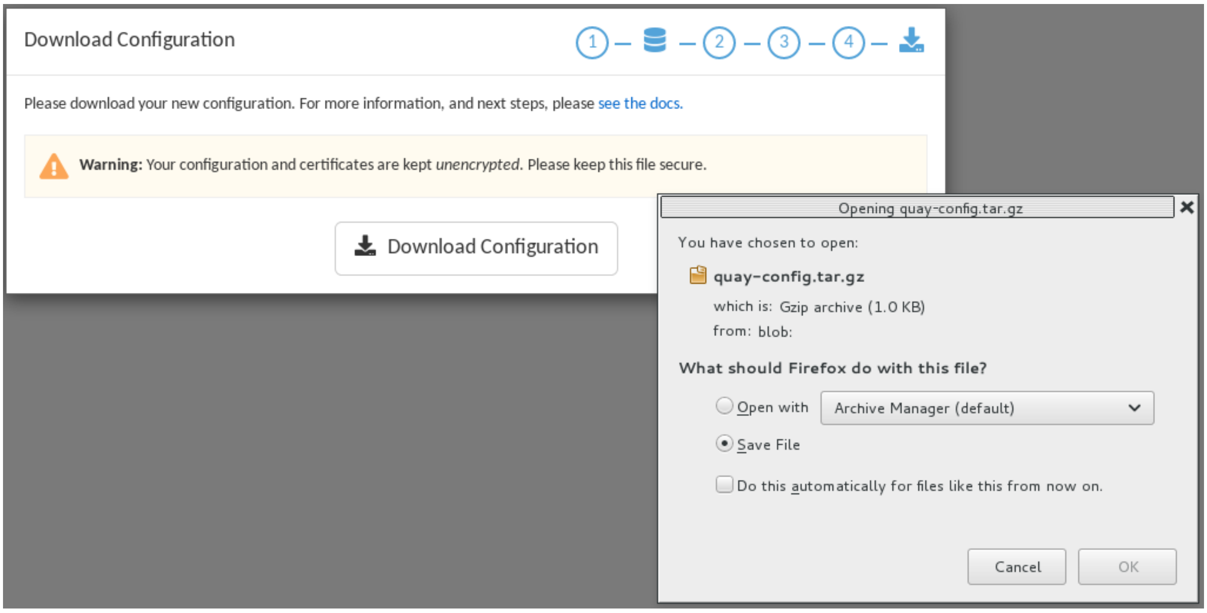
- **Data Consistency Settings**: Select to relax logging consistency guarantees to improve performance and availability.
- **Time Machine**: Allow older image tags to remain in the repository for set periods of time and allow users to select their own tag expiration times.
- **redis**: Identify the hostname or IP address (and optional password) to connect to the redis service used by Red Hat Quay.
- **Repository Mirroring**: Choose the checkbox to Enable Repository Mirroring. With this enabled, you can create repositories in your Red Hat Quay cluster that mirror selected repositories from remote registries. Before you can enable repository mirroring, start the repository mirroring worker as described later in this procedure.
- **Registry Storage**: Identify the location of storage. A variety of cloud and local storage options are available. Remote storage is required for high availability. Identify the Ceph storage location if you are following the example for Red Hat Quay high availability storage. On OpenShift, the example uses Amazon S3 storage.
- **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 the **quayctl** tool).
- **rkt Conversion**: Allow **rkt fetch** to be used to fetch images from Red Hat Quay registry. Public and private GPG2 keys are needed (see Generating signing keys for ACI conversion for details. This field is deprecated.
- **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).

- **Registry Protocol Settings**: Leave the **Restrict V1 Push Support** checkbox enabled to restrict access to Docker V1 protocol pushes. Although Red Hat recommends against enabling Docker V1 push protocol, if you do allow it, you must explicitly whitelist the namespaces for which it is enabled.

- **Dockerfile Build Support** Enable to allow users to submit Dockerfiles to be built and pushed to Red Hat Quay. This is not recommended for multitenant environments.

9. **Save the changes**: Select **Save Configuration Changes**. You are presented with the following Download Configuration screen:



Download Configuration

Please download your new configuration. For more information, and next steps, please see the docs.

Warning: Your configuration and certificates are kept *unencrypted*. Please keep this file secure.

Download Configuration

Opening quay-config.tar.gz

You have chosen to open:

quay-config.tar.gz

which is: Gzip archive (1.0 KB)

from: blob:

What should Firefox do with this file?

○ Open with Archive Manager (default)

● Save File

□ Do this automatically for files like this from now on.

Cancel OK

10. **Download configuration**: Select the **Download Configuration** button and save the tarball (**quay-config.tar.gz**) to a local directory to use later to start Red Hat Quay.

At this point, you can shutdown the Red Hat Quay configuration tool and close your browser. Next, copy the tarball file to the system on which you want to install your first Red Hat Quay node. For a basic install, you might just be running Red Hat Quay on the same system.

CHAPTER 5. DEPLOYING RED HAT QUAY

To deploy the Red Hat Quay service on the nodes in your cluster, you use the same quay container you used to create the configuration file. The differences here are that you:

- Identify directories where the configuration files and data are stored
- Run the command with **--sysctl net.core.somaxconn=4096**
- Don't use the **config** option or password

For a basic setup, you can deploy on a single node; for high availability you probably want three or more nodes (for example, quay01, quay02, and quay03).

> NOTE
>
> The resulting Red Hat Quay service will listen on regular port 8080 and SSL port 8443. This is different from previous releases of Red Hat Quay, which listened on standard ports 80 and 443, respectively. In this document, we map 8080 and 8443 to standard ports 80 and 443 on the host, respectively. Througout the rest of this document, we assume you have mapped the ports in this way.

Here is what you do:

1. **Create directories**: Create two directories to store configuration information and data on the host. For example:

```
# mkdir -p /mnt/quay/config
# #optional: if you don't choose to install an Object Store
# mkdir -p /mnt/quay/storage
```

2. **Copy config files**: Copy the tarball (**quay-config.tar.gz**) to the configuration directory and unpack it. For example:

```
# cp quay-config.tar.gz /mnt/quay/config/
# tar xvf quay-config.tar.gz
config.yaml ssl.cert ssl.key
```

3. **Deploy Red Hat Quay**: Having already authenticated to Quay.io (see Accessing Red Hat Quay) 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.

```
# docker run --restart=always -p 443:8443 -p 80:8080 \
  --sysctl net.core.somaxconn=4096 \
  --privileged=true \
  -v /mnt/quay/config:/conf/stack:Z \
  -v /mnt/quay/storage:/datastorage:Z \
  -d quay.io/redhat/quay:v3.2.2
```

4. **Open browser to UI:** Once the quay container has started, go to your web browser and open the URL, to the node running the quay container.

5. **Log into Red Hat Quay:** Using the superuser account you created during configuration, log in and make sure Red Hat Quay is working properly.

6. **Add more Red Hat Quay nodes** At this point, you have the option of adding more nodes to this Red Hat Quay cluster by simply going to each node, then adding the tarball and starting the quay container as just shown.

7. **Add optional features:** To add more features to your Red Hat Quay cluster, such as Clair images scanning and Repository Mirroring, continue on to the next section.

5.1. ADD CLAIR IMAGE SCANNING TO RED HAT QUAY

Setting up and deploying Clair image scanning for your Red Hat Quay deployment requires the following basic steps:

- Restarting the Red Hat Quay Setup tool
- Creating authentication keys for Clair
- Setting up a database for Clair
- Deploying the Clair container

These steps are described in Red Hat Quay Security Scanning with Clair .

5.2. ADD REPOSITORY MIRRORING RED HAT QUAY

Enabling repository mirroring allows you to create container image repositories on your Red Hat Quay cluster that exactly match the content of a selected external registry, then sync the contents of those repositories on a regular schedule and on demand.

To add the repository mirroring feature to your Red Hat Quay cluster:

- Run the repository mirroring worker. To do this, you start a quay pod with the **repomirror** option.
- Select "Enable Repository Mirroring in the Red Hat Quay Setup tool.
- Log into your Red Hat Quay Web UI and begin creating mirrored repositories as described in Repository Mirroring in Red Hat Quay .

The following procedure assumes you already have a running Red Hat Quay cluster on an OpenShift platform, with the Red Hat Quay Setup container running in your browser:

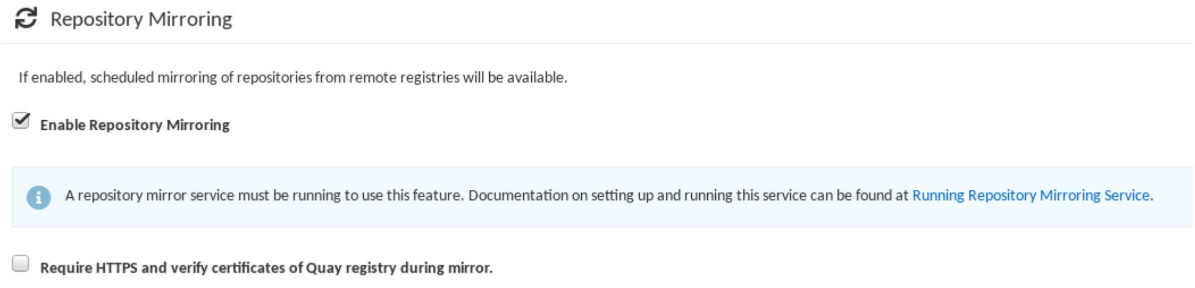
1. **Start the repo mirroring worker:** Start the quay container in **repomirror** mode. This example assumes you have configured TLS communications using a certificate that is currently stored in **/root/ca.crt**. If not, then remove the line that adds **/root/ca.crt** to the container:

   ```
   $ docker run -d --name mirroring-worker \
     -v /mnt/quay/config:/conf/stack \
     -v /root/ca.crt:/etc/pki/ca-trust/source/anchors/ca.crt \
     quay.io/redhat/quay:v3.2.2 repomirror
   ```

2. **Log into config tool:** Log into the Red Hat Quay Setup Web UI (config tool).

3. **Enable repository mirroring**: Scroll down the the Repository Mirroring section and select the Enable Repository Mirroring check box, as shown here:

4. **Select HTTPS and cert verification**: If you want to require HTTPS communications and verify certificates during mirroring, select this check box.



Repository Mirroring

If enabled, scheduled mirroring of repositories from remote registries will be available.

☑ Enable Repository Mirroring

A repository mirror service must be running to use this feature. Documentation on setting up and running this service can be found at Running Repository Mirroring Service.

□ Require HTTPS and verify certificates of Quay registry during mirror.

5. **Save configuration**: Select the Save Configuration Changes button. Repository mirroring should now be enabled on your Red Hat Quay cluster. Refer to Repository Mirroring in Red Hat Quay for details on setting up your own mirrored container image repositories.

CHAPTER 6. STARTING TO USE 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