Red Hat Quay 3.15

Proof of Concept - Deploying Red Hat Quay

Deploying Red Hat Quay

Last Updated: 2025-10-14

Red Hat Quay 3.15 Proof of Concept - Deploying Red Hat Quay

Deploying Red Hat Quay

Abstract

Getting started with Red Hat Quay

Table of Contents

PREFACE

IMPORTANT

The following *proof of concept* deployment method is unsupported for production purposes. This deployment type uses local storage. Local storage is not guaranteed to provide the required read-after-write consistency and data integrity guarantees during parallel access that a storage registry like Red Hat Quay requires. Do not use this deployment type for production purposes. Use it for testing purposes only.

Red Hat Quay is an enterprise-quality registry for building, securing and serving container images. The documents in this section detail how to deploy Red Hat Quay for *proof of concept*, or non-production, purposes. The primary objectives of this document includes the following:

- How to deploy Red Hat Quay for basic non-production purposes.
- Asses Red Hat Quay's container image management, including how to push, pull, tag, and organize images.
- Explore availability and scalability.
- How to deploy an advanced Red Hat Quay proof of concept deployment using SSL/TLS certificates.

Beyond the primary objectives of this document, a proof of concept deployment can be used to test various features offered by Red Hat Quay, such as establishing superusers, setting repository quota limitations, enabling Splunk for action log storage, enabling Clair for vulnerability reporting, and more. See the "Next steps" section for a list of some of the features available after you have followed this guide.

This proof of concept deployment procedure can be followed on a single machine, either physical or virtual.

CHAPTER 1. PREREQUISITES

- Red Hat Enterprise Linux (RHEL) 9
  - To obtain the latest version of Red Hat Enterprise Linux (RHEL) 9, see Download Red Hat Enterprise Linux.
  - For installation instructions, see the Product Documentation for Red Hat Enterprise Linux 9 .
- An active subscription to Red Hat
- Two or more virtual CPUs
- 4 GB or more of RAM
- Approximately 30 GB of disk space on your test system, which can be broken down as follows:
  - Approximately 10 GB of disk space for the Red Hat Enterprise Linux (RHEL) operating system.
  - Approximately 10 GB of disk space for Docker storage for running three containers.
  - Approximately 10 GB of disk space for Red Hat Quay local storage.

    NOTE

    CEPH or other local storage might require more memory.

    More information on sizing can be found at Quay 3.x Sizing Guidelines.
- The following architectures are supported for Red Hat Quay:
  - amd64/x86_64
  - s390x
  - ppc64le

1.1. INSTALLING PODMAN

This document uses Podman for creating and deploying containers.

For more information on Podman and related technologies, see Building, running, and managing Linux containers on Red Hat Enterprise Linux 9.

IMPORTANT

If you do not have Podman installed on your system, the use of equivalent Docker commands might be possible, however this is not recommended. Docker has not been tested with Red Hat Quay 3.15, and will be deprecated in a future release. Podman is recommended for highly available, production quality deployments of Red Hat Quay 3.15.

Use the following procedure to install Podman.

Procedure

- Enter the following command to install Podman:

  ```
  $ sudo yum install -y podman
  ```

- Alternatively, you can install the **container-tools** module, which pulls in the full set of container software packages:

  ```
  $ sudo yum module install -y container-tools
  ```

CHAPTER 2. PREPARING RED HAT ENTERPRISE LINUX FOR A RED HAT QUAY PROOF OF CONCEPT DEPLOYMENT

Use the following procedures to configure Red Hat Enterprise Linux (RHEL) for a Red Hat Quay proof of concept deployment.

2.1. INSTALL AND REGISTER THE RHEL SERVER

Use the following procedure to configure the Red Hat Enterprise Linux (RHEL) server for a Red Hat Quay proof of concept deployment.

Procedure

1. Install the latest RHEL 9 server. You can do a minimal, shell-access only install, or Server plus GUI if you want a desktop.

2. Register and subscribe your RHEL server system as described in How to register and subscribe a RHEL system to the Red Hat Customer Portal using Red Hat Subscription-Manager

3. Enter the following commands to register your system and list available subscriptions. Choose an available RHEL server subscription, attach to its pool ID, and upgrade to the latest software:

```
# subscription-manager register --username=<user_name> --password=<password>
# subscription-manager refresh
# subscription-manager list --available
# subscription-manager attach --pool=<pool_id>
# yum update -y
```

2.2. REGISTRY AUTHENTICATION

Use the following procedure to authenticate your registry for a Red Hat Quay proof of concept.

Procedure

1. Set up authentication to **registry.redhat.io** by following the Red Hat Container Registry Authentication procedure. Setting up authentication allows you to pull the **Quay** container.

   NOTE

   This differs from earlier versions of Red Hat Quay, when the images were hosted on Quay.io.

2. Enter the following command to log in to the registry:

```
$ sudo podman login registry.redhat.io
```

   You are prompted to enter your **username** and **password**.

2.3. FIREWALL CONFIGURATION

If you have a firewall running on your system, you might have to add rules that allow access to Red Hat Quay. Use the following procedure to configure your firewall for a proof of concept deployment.

Procedure

- The commands required depend on the ports that you have mapped on your system, for example:

```
# firewall-cmd --permanent --add-port=80/tcp \
&& firewall-cmd --permanent --add-port=443/tcp \
&& firewall-cmd --permanent --add-port=5432/tcp \
&& firewall-cmd --permanent --add-port=5433/tcp \
&& firewall-cmd --permanent --add-port=6379/tcp \
&& firewall-cmd --reload
```

2.4. IP ADDRESSING AND NAMING SERVICES

There are several ways to configure the component containers in Red Hat Quay so that they can communicate with each other, for example:

- **Using a naming service**. If you want your deployment to survive container restarts, which typically result in changed IP addresses, you can implement a naming service. For example, the dnsname plugin is used to allow containers to resolve each other by name.
- **Using the host network**. You can use the **podman run** command with the **--net=host** option and then use container ports on the host when specifying the addresses in the configuration. This option is susceptible to port conflicts when two containers want to use the same port. This method is not recommended.
- **Configuring port mapping**. You can use port mappings to expose ports on the host and then use these ports in combination with the host IP address or host name.

This document uses port mapping and assumes a static IP address for your host system.

Table 2.1. Sample proof of concept port mapping

Component	Port mapping	Address
Quay	**-p 80:8080 -p 443:8443**	http://quay-server.example.com
Postgres for Quay	**-p 5432:5432**	quay-server.example.com:5432
Redis	**-p 6379:6379**	quay-server.example.com:6379
Postgres for Clair V4	**-p 5433:5432**	quay-server.example.com:5433

Component	Port mapping	Address
Clair V4	**-p 8081:8080**	http://quay-server.example.com:8081

CHAPTER 3. PREPARING YOUR SYSTEM TO DEPLOY RED HAT QUAY

For a proof of concept Red Hat Quay deployment, you must configure port mapping, a database, and Redis prior to deploying the registry. Use the following procedures to prepare your system to deploy Red Hat Quay.

3.1. CONFIGURING PORT MAPPING FOR RED HAT QUAY

You can use port mappings to expose ports on the host and then use these ports in combination with the host IP address or host name to navigate to the Red Hat Quay endpoint.

Procedure

1. Enter the following command to obtain your static IP address for your host system:

   ```
   $ ip a
   ```

   Example output

   ```
   ---
       link/ether 6c:6a:77:eb:09:f1 brd ff:ff:ff:ff:ff:ff
       inet 192.168.1.132/24 brd 192.168.1.255 scope global dynamic noprefixroute wlp82s0
   ---
   ```

2. Add the IP address and a local hostname, for example, **quay-server.example.com** to your **/etc/hosts** file that will be used to reach the Red Hat Quay endpoint. You can confirm that the IP address and hostname have been added to the **/etc/hosts** file by entering the following command:

   ```
   $ cat /etc/hosts
   ```

   Example output

   ```
   192.168.1.132 quay-server.example.com
   ```

3.2. CONFIGURING THE DATABASE

Red Hat Quay requires a database for storing metadata. PostgreSQL is used throughout this document. For this deployment, a directory on the local file system to persist database data is used.

Use the following procedure to set up a PostgreSQL database.

Procedure

1. In the installation folder, denoted here by the **$QUAY** variable, create a directory for the database data by entering the following command:

   ```
   $ mkdir -p $QUAY/postgres-quay
   ```

2. Set the appropriate permissions by entering the following command:

```
$ setfacl -m u:26:-wx $QUAY/postgres-quay
```

3. Start the **Postgres** container, specifying the username, password, and database name and port, with the volume definition for database data:

```
$ sudo podman run -d --rm --name postgresql-quay \
  -e POSTGRESQL_USER=quayuser \
  -e POSTGRESQL_PASSWORD=quaypass \
  -e POSTGRESQL_DATABASE=quay \
  -e POSTGRESQL_ADMIN_PASSWORD=adminpass \
  -p 5432:5432 \
  -v $QUAY/postgres-quay:/var/lib/pgsql/data:Z \
  registry.redhat.io/rhel8/postgresql-13
```

4. Ensure that the Postgres **pg_trgm** module is installed by running the following command:

```
$ sudo podman exec -it postgresql-quay /bin/bash -c 'echo "CREATE EXTENSION IF NOT
EXISTS pg_trgm" | psql -d quay -U postgres'
```

NOTE

The **pg_trgm** module is required for the **Quay** container.

3.3. CONFIGURING REDIS

Redis is a key-value store that is used by Red Hat Quay for live builder logs.

Use the following procedure to deploy the **Redis** container for the Red Hat Quay proof of concept.

Procedure

- Start the **Redis** container, specifying the port and password, by entering the following command:

```
$ sudo podman run -d --rm --name redis \
  -p 6379:6379 \
  -e REDIS_PASSWORD=strongpassword \
  registry.redhat.io/rhel8/redis-6:1-110
```

CHAPTER 4. DEPLOYING RED HAT QUAY

After you have configured your Red Hat Quay deployment, you can deploy it using the following procedures.

Prerequisites

- The Red Hat Quay database is running.
- The Redis server is running.

4.1. CREATING THE YAML CONFIGURATION FILE

Use the following procedure to deploy Red Hat Quay locally.

Procedure

1. Enter the following command to create a minimal **config.yaml** file that is used to deploy the Red Hat Quay container:

   ```
   $ touch config.yaml
   ```

2. Copy and paste the following YAML configuration into the **config.yaml** file:

   ```
   BUILDLOGS_REDIS:
       host: quay-server.example.com
       password: strongpassword
       port: 6379
   CREATE_NAMESPACE_ON_PUSH: true
   DATABASE_SECRET_KEY: a8c2744b-7004-4af2-bcee-e417e7bdd235
   DB_URI: postgresql://quayuser:quaypass@quay-server.example.com:5432/quay
   DISTRIBUTED_STORAGE_CONFIG:
       default:
           - LocalStorage
           - storage_path: /datastorage/registry
   DISTRIBUTED_STORAGE_DEFAULT_LOCATIONS: []
   DISTRIBUTED_STORAGE_PREFERENCE:
       - default
   FEATURE_MAILING: false
   SECRET_KEY: e9bd34f4-900c-436a-979e-7530e5d74ac8
   SERVER_HOSTNAME: quay-server.example.com
   SETUP_COMPLETE: true
   USER_EVENTS_REDIS:
       host: quay-server.example.com
       password: strongpassword
       port: 6379
   ```

3. Create a directory to copy the Red Hat Quay configuration bundle to:

   ```
   $ mkdir $QUAY/config
   ```

4. Copy the Red Hat Quay configuration file to the directory:

```
$ cp -v config.yaml $QUAY/config
```

4.1.1. Configuring a Red Hat Quay superuser

You can optionally add a superuser by editing the **config.yaml** file to add the necessary configuration fields. The list of superuser accounts is stored as an array in the field **SUPER_USERS**. Superusers have the following capabilities:

- User management
- Organization management
- Service key management
- Change log transparency
- Usage log management
- Globally-visible user message creation

Procedure

1. Add the **SUPER_USERS** array to the **config.yaml** file:

   ```
   SERVER_HOSTNAME: quay-server.example.com
   SETUP_COMPLETE: true
   SUPER_USERS:
     - quayadmin (1)
   ...
   ```

   (1) If following this guide, use **quayadmin**.

4.2. PREPARE LOCAL STORAGE FOR IMAGE DATA

Use the following procedure to set your local file system to store registry images.

Procedure

1. Create a local directory that will store registry images by entering the following command:

   ```
   $ mkdir $QUAY/storage
   ```

2. Set the directory to store registry images:

   ```
   $ setfacl -m u:1001:-wx $QUAY/storage
   ```

4.3. DEPLOY THE RED HAT QUAY REGISTRY

Use the following procedure to deploy the **Quay** registry container.

Procedure

- Enter the following command to start the **Quay** registry container, specifying the appropriate volumes for configuration data and local storage for image data:

```
$ sudo podman run -d --rm -p 80:8080 -p 443:8443  \
   --name=quay \
   -v $QUAY/config:/conf/stack:Z \
   -v $QUAY/storage:/datastorage:Z \
   registry.redhat.io/quay/quay-rhel8:v3.15.2
```

CHAPTER 5. USING RED HAT QUAY

The following steps show you how to use the interface to create new organizations and repositories, and to search and browse existing repositories. Following step 3, you can use the command line interface to interact with the registry and to push and pull images.

Procedure

1. Use your browser to access the user interface for the Red Hat Quay registry at **http://quay-server.example.com**, assuming you have configured **quay-server.example.com** as your hostname in your **/etc/hosts** file and in your **config.yaml** file.

2. Click **Create Account** and add a user, for example, **quayadmin** with a password **password**.

3. From the command line, log in to the registry:

```
$ sudo podman login --tls-verify=false quay-server.example.com
```

Example output

```
Username: quayadmin
Password: password
Login Succeeded!
```

5.1. PUSHING AND PULLING IMAGES ON RED HAT QUAY

Use the following procedure to push and pull images to your Red Hat Quay registry.

Procedure

1. To test pushing and pulling images from the Red Hat Quay registry, first pull a sample image from an external registry:

```
$ sudo podman pull busybox
```

Example output

```
Trying to pull docker.io/library/busybox...
Getting image source signatures
Copying blob 4c892f00285e done
Copying config 22667f5368 done
Writing manifest to image destination
Storing signatures
22667f53682a2920948d19c7133ab1c9c3f745805c14125859d20cede07f11f9
```

2. Enter the following command to see the local copy of the image:

```
$ sudo podman images
```

Example output

```
REPOSITORY                    TAG      IMAGE ID       CREATED        SIZE
docker.io/library/busybox     latest   22667f53682a   14 hours ago   1.45 MB
```

3. Enter the following command to tag this image, which prepares the image for pushing it to the registry:

```
$ sudo podman tag docker.io/library/busybox quay-
server.example.com/quayadmin/busybox:test
```

4. Push the image to your registry. Following this step, you can use your browser to see the tagged image in your repository.

```
$ sudo podman push --tls-verify=false quay-server.example.com/quayadmin/busybox:test
```

Example output

```
Getting image source signatures
Copying blob 6b245f040973 done
Copying config 22667f5368 done
Writing manifest to image destination
Storing signatures
```

5. To test access to the image from the command line, first delete the local copy of the image:

```
$ sudo podman rmi quay-server.example.com/quayadmin/busybox:test
```

Example output

```
Untagged: quay-server.example.com/quayadmin/busybox:test
```

6. Pull the image again, this time from your Red Hat Quay registry:

```
$ sudo podman pull --tls-verify=false quay-server.example.com/quayadmin/busybox:test
```

Example output

```
Trying to pull quay-server.example.com/quayadmin/busybox:test...
Getting image source signatures
Copying blob 6ef22a7134ba [--------------------------------------] 0.0b / 0.0b
Copying config 22667f5368 done
Writing manifest to image destination
Storing signatures
22667f53682a2920948d19c7133ab1c9c3f745805c14125859d20cede07f11f9
```

5.2. ACCESSING THE SUPERUSER ADMIN PANEL

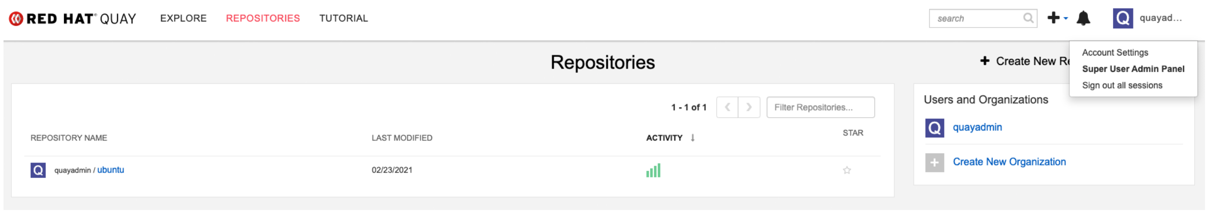
If you added a superuser to your **config.yaml** file, you can access the **Superuser Admin Panel** on the Red Hat Quay UI by using the following procedure.

Prerequisites

- You have configured a superuser.

Procedure

1. Access the **Superuser Admin Panel** on the Red Hat Quay UI by clicking on the current user's name or avatar in the navigation pane of the UI. Then, click **Superuser Admin Panel**.



On this page, you can manage users, your organization, service keys, view change logs, view usage logs, and create global messages for your organization.

CHAPTER 6. PROOF OF CONCEPT DEPLOYMENT USING SSL/TLS CERTIFICATES

Use the following sections to configure a proof of concept Red Hat Quay deployment with SSL/TLS certificates.

6.1. USING SSL/TLS

Documentation for *Using SSL/TLS* has been revised and moved to Securing Red Hat Quay. This chapter will be removed in a future version of Red Hat Quay.

6.1.1. Creating a Certificate Authority

Use the following procedure to set up your own CA and use it to issue a server certificate for your domain. This allows you to secure communications with SSL/TLS using your own certificates.

Procedure

1. Generate the root CA key by entering the following command:

```
$ openssl genrsa -out rootCA.key 2048
```

2. Generate the root CA certificate by entering the following command:

```
$ openssl req -x509 -new -nodes -key rootCA.key -sha256 -days 1024 -out rootCA.pem
```

3. Enter the information that will be incorporated into your certificate request, including the server hostname, for example:

```
Country Name (2 letter code) [XX]:IE
State or Province Name (full name) []:GALWAY
Locality Name (eg, city) [Default City]:GALWAY
Organization Name (eg, company) [Default Company Ltd]:QUAY
Organizational Unit Name (eg, section) []:DOCS
Common Name (eg, your name or your server's hostname) []:quay-server.example.com
```

4. Generate the server key by entering the following command:

```
$ openssl genrsa -out ssl.key 2048
```

5. Generate a signing request by entering the following command:

```
$ openssl req -new -key ssl.key -out ssl.csr
```

6. Enter the information that will be incorporated into your certificate request, including the server hostname, for example:

```
Country Name (2 letter code) [XX]:IE
State or Province Name (full name) []:GALWAY
Locality Name (eg, city) [Default City]:GALWAY
Organization Name (eg, company) [Default Company Ltd]:QUAY
```

```
Organizational Unit Name (eg, section) []:DOCS
Common Name (eg, your name or your server's hostname) []:quay-server.example.com
Email Address []:
```

7. Create a configuration file **openssl.cnf**, specifying the server hostname, for example:

Example openssl.cnf file

```
[req]
req_extensions = v3_req
distinguished_name = req_distinguished_name
[req_distinguished_name]
[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
subjectAltName = @alt_names
[alt_names]
DNS.1 = <quay-server.example.com>
IP.1 = 192.168.1.112
```

8. Use the configuration file to generate the certificate **ssl.cert**:

```
$ openssl x509 -req -in ssl.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out
ssl.cert -days 356 -extensions v3_req -extfile openssl.cnf
```

9. Confirm your created certificates and files by entering the following command:

```
$ ls /path/to/certificates
```

Example output

```
rootCA.key ssl-bundle.cert ssl.key custom-ssl-config-bundle-secret.yaml rootCA.pem ssl.cert
openssl.cnf rootCA.srl  ssl.csr
```

6.2. CONFIGURING SSL/TLS FOR STANDALONE RED HAT QUAY DEPLOYMENTS

For standalone Red Hat Quay deployments, SSL/TLS certificates must be configured by using the command-line interface and by updating your **config.yaml** file manually.

6.2.1. Configuring custom SSL/TLS certificates by using the command line interface

SSL/TLS must be configured by using the command-line interface (CLI) and updating your **config.yaml** file manually.

Prerequisites

- You have created a certificate authority and signed the certificate.

Procedure

1. Copy the certificate file and primary key file to your configuration directory, ensuring they are named **ssl.cert** and **ssl.key** respectively:

```
cp ~/ssl.cert ~/ssl.key /path/to/configuration_directory
```

2. Navigate to the configuration directory by entering the following command:

```
$ cd /path/to/configuration_directory
```

3. Edit the **config.yaml** file and specify that you want Red Hat Quay to handle SSL/TLS:

Example config.yaml file

```
# ...
SERVER_HOSTNAME: <quay-server.example.com>
...
PREFERRED_URL_SCHEME: https
# ...
```

4. Optional: Append the contents of the **rootCA.pem** file to the end of the **ssl.cert** file by entering the following command:

```
$ cat rootCA.pem >> ssl.cert
```

5. Stop the **Quay** container by entering the following command:

```
$ sudo podman stop <quay_container_name>
```

6. Restart the registry by entering the following command:

```
$ sudo podman run -d --rm -p 80:8080 -p 443:8443 \
  --name=quay \
  -v $QUAY/config:/conf/stack:Z \
  -v $QUAY/storage:/datastorage:Z \
  registry.redhat.io/quay/quay-rhel8:v3.15.2
```

6.3. TESTING THE SSL/TLS CONFIGURATION

Your SSL/TLS configuration can be tested by using the command-line interface (CLI). Use the following procedure to test your SSL/TLS configuration.

6.3.1. Testing the SSL/TLS configuration using the CLI

Your SSL/TLS configuration can be tested by using the command-line interface (CLI). Use the following procedure to test your SSL/TLS configuration.

Use the following procedure to test your SSL/TLS configuration using the CLI.

Procedure

1. Enter the following command to attempt to log in to the Red Hat Quay registry with SSL/TLS enabled:

```
$ sudo podman login quay-server.example.com
```

Example output

```
Error: error authenticating creds for "quay-server.example.com": error pinging docker registry
quay-server.example.com: Get "https://quay-server.example.com/v2/": x509: certificate
signed by unknown authority
```

2. Because Podman does not trust self-signed certificates, you must use the **--tls-verify=false** option:

```
$ sudo podman login --tls-verify=false quay-server.example.com
```

Example output

```
Login Succeeded!
```

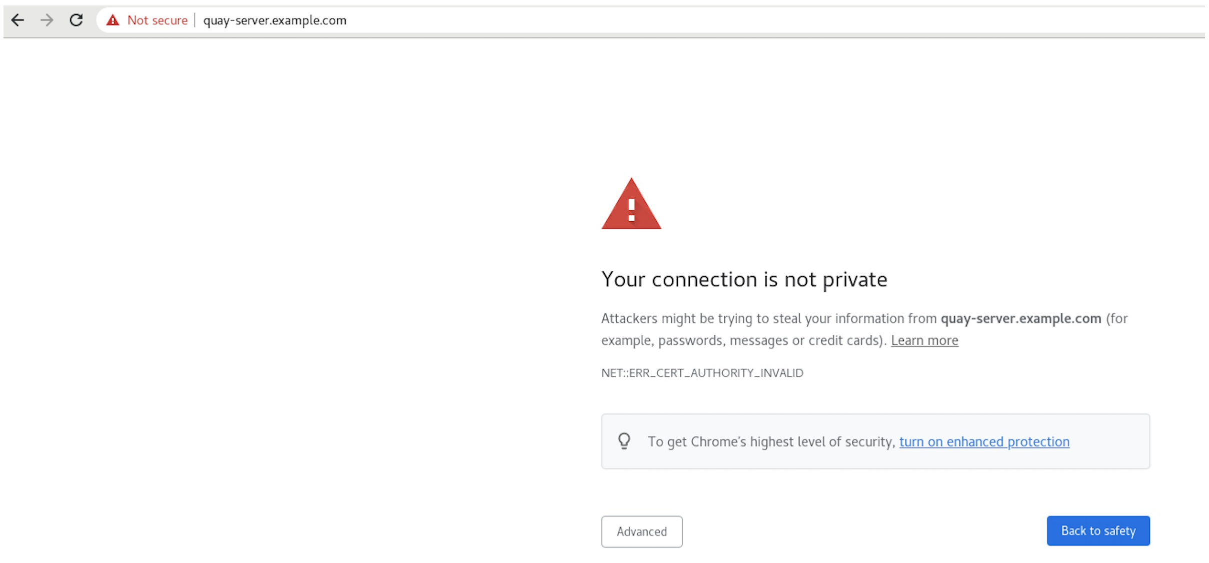
In a subsequent section, you will configure Podman to trust the root Certificate Authority.

6.3.2. Testing the SSL/TLS configuration using a browser

Use the following procedure to test your SSL/TLS configuration using a browser.

Procedure

1. Navigate to your Red Hat Quay registry endpoint, for example, https://quay-server.example.com. If configured correctly, the browser warns of the potential risk:



Not secure | quay-server.example.com

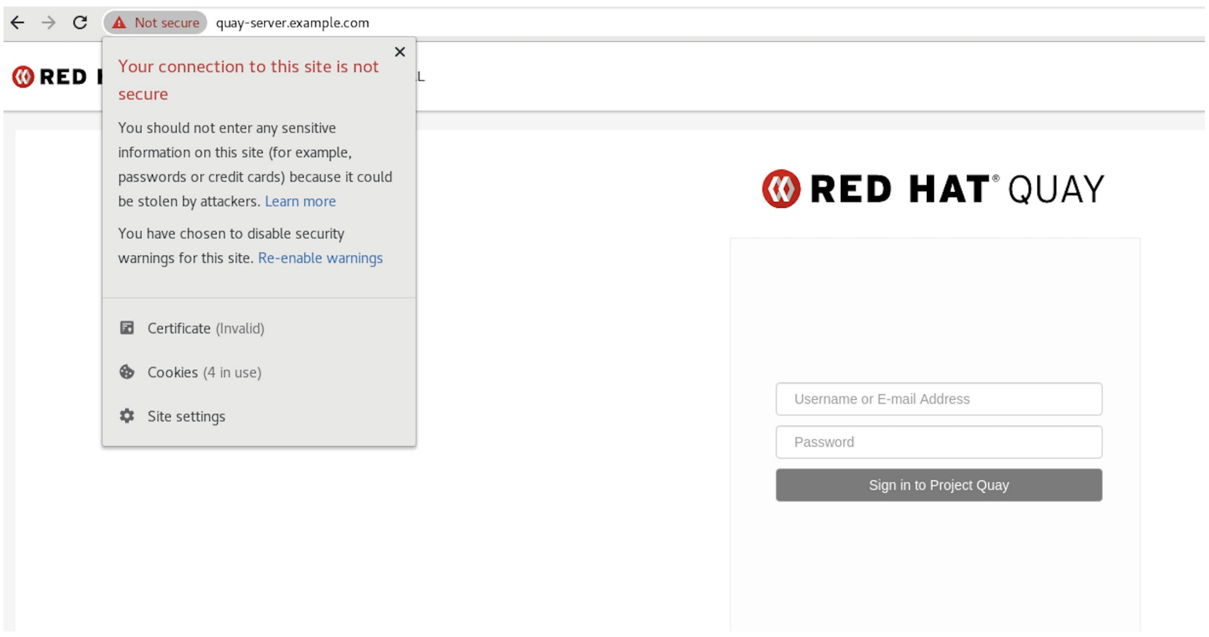
Your connection is not private

Attackers might be trying to steal your information from **quay-server.example.com** (for example, passwords, messages or credit cards). Learn more

NET::ERR_CERT_AUTHORITY_INVALID

To get Chrome's highest level of security, turn on enhanced protection

2. Proceed to the log in screen. The browser notifies you that the connection is not secure. For example:



Your connection to this site is not secure

You should not enter any sensitive information on this site (for example, passwords or credit cards) because it could be stolen by attackers. Learn more

You have chosen to disable security warnings for this site. Re-enable warnings

In the following section, you will configure Podman to trust the root Certificate Authority.

6.4. CONFIGURING PODMAN TO TRUST THE CERTIFICATE AUTHORITY

Podman uses two paths to locate the Certificate Authority (CA) file: **/etc/containers/certs.d/** and **/etc/docker/certs.d/**. Use the following procedure to configure Podman to trust the CA.

Procedure

1. Copy the root CA file to one of **/etc/containers/certs.d/** or **/etc/docker/certs.d/**. Use the exact path determined by the server hostname, and name the file **ca.crt**:

```
$ sudo cp rootCA.pem /etc/containers/certs.d/quay-server.example.com/ca.crt
```

2. Verify that you no longer need to use the **--tls-verify=false** option when logging in to your Red Hat Quay registry:

```
$ sudo podman login quay-server.example.com
```

Example output

```
Login Succeeded!
```

6.5. CONFIGURING THE SYSTEM TO TRUST THE CERTIFICATE AUTHORITY

Use the following procedure to configure your system to trust the certificate authority.

Procedure

1. Enter the following command to copy the **rootCA.pem** file to the consolidated system-wide trust store:

```
$ sudo cp rootCA.pem /etc/pki/ca-trust/source/anchors/
```

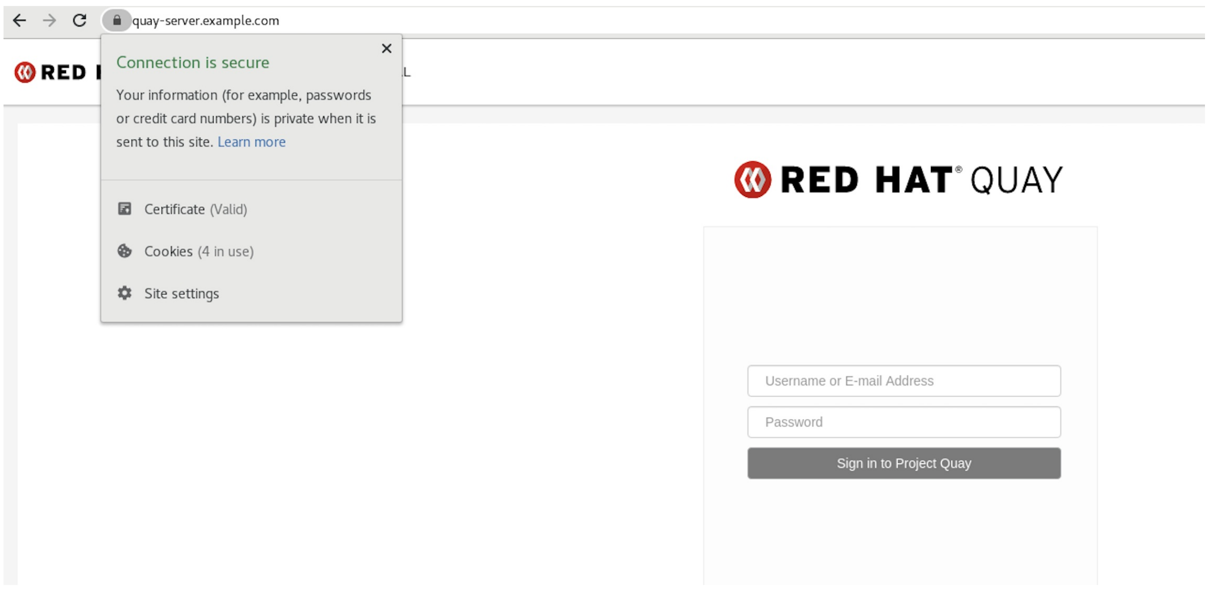
2. Enter the following command to update the system-wide trust store configuration:

```
$ sudo update-ca-trust extract
```

3. Optional. You can use the **trust list** command to ensure that the **Quay** server has been configured:

```
$ trust list | grep quay
    label: quay-server.example.com
```

Now, when you browse to the registry at **https://quay-server.example.com**, the lock icon shows that the connection is secure:



Your information (for example, passwords or credit card numbers) is private when it is sent to this site. Learn more

4. To remove the **rootCA.pem** file from system-wide trust, delete the file and update the configuration:

```
$ sudo rm /etc/pki/ca-trust/source/anchors/rootCA.pem
```

```
$ sudo update-ca-trust extract
```

```
$ trust list | grep quay
```

More information can be found in the RHEL 9 documentation in the chapter Using shared system certificates.

CHAPTER 7. CONFIGURING THE PODMAN CNI TO USE IPV6

In some cases, you might want to run a local instance of Red Hat Quay to use IPv6. This setup is common for development or testing purposes.

By default, the Podman network for a root user does not use IPv6, and instead defaults to use IPv4. You can configure the Podman Container Network Interface (CNI) to use both IPv4 and IPv6, which allows for a local instance of Red Hat Quay using IPv6.

Additional resources

- How to configure the default podman container network for the root user to use both IPv4 and IPv6.

7.1. CREATING A DUAL-STACK CONTAINER NETWORK

Use the following procedure to create a new container network that is dual-stack.

Procedure

- Unless there is a requirement that the *default* container network use both IPv4 and IPv6, it is suggested that a *new* container network is created that is dual-stack. As a root user, create a new container network that is dual-stack by running the following command:

  ```
  # podman network create ip-dual-stack --ipv6
  ```

  With this command, new containers use this network are a natively dual-stack.

7.2. DEPLOYING A LOCAL INSTANCE OF RED HAT QUAY IN IPV6

After you have created a new dual-stack container network, you can deploy a local instance of Red Hat Quay.

Prerequisites

- You have configured Redis, your PostgreSQL database, local image storage, and creating a **config.yaml** file with the desired settings.

Procedure

1. In your Red Hat Quay **config.yaml** file, set the **FEATURE_LISTEN_IP_VERSION** field to **IPv6**. For example:

   ```
   # ...
   FEATURE_LISTEN_IP_VERSION: dual-stack
   # ...
   ```

2. Enter the following command to start the **Quay** registry container, specifying the appropriate volumes for configuration data and local storage for image data. Note that if you are using dual-stack, you must specify explicit IPv6 port mapping when starting the container.

   ```
   $ sudo podman run -d --rm -p "[::]:80:8080" -p "[::]:443:8443"  \
      --name=quay \
   ```

```
-v $QUAY/config:/conf/stack:Z \
-v $QUAY/storage:/datastorage:Z \
registry.redhat.io/quay/quay-rhel8:v3.15.2
```

CHAPTER 8. NEXT STEPS

The following sections might be useful after deploying a proof of concept version of Red Hat Quay. Many of these procedures can be used on a proof of concept deployment, offering insights to Red Hat Quay's features.

- Using Red Hat Quay . The content in this guide explains the following concepts:
  - Adding users and repositories
  - Using image tags
  - Building Dockerfiles with build workers
  - Setting up build triggers
  - Adding notifications for repository events
  - and more
- Managing Red Hat Quay. The content in this guide explains the following concepts:
  - Using SSL/TLS
  - Configuring action log storage
  - Configuring Clair security scanner
  - Repository mirroring
  - IPv6 and dual-stack deployments
  - Configuring OIDC for Red Hat Quay
  - Geo-replication
  - and more