Red Hat Quay 3.11

概念実証 - Red Hat Quay のデプロイ

Red Hat Quay のデプロイ

Last Updated: 2024-04-05

Red Hat Quay 3.11 概念実証 - Red Hat Quay のデプロイ

Red Hat Quay のデプロイ

法律上の通知

概要

Red Hat Quay の使用開始

目次

はじめに

Red Hat Quay は、コンテナーイメージをビルドして、保護し、これを提供するためのエンタープライズ品質のレジストリーです。このドキュメントでは、**概念実証** (非実稼働) 目的で Red Hat Quay をデプロイする方法について詳しく説明します。このドキュメントの主な目的は次のとおりです。

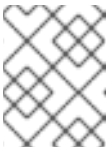
- 基本的な非実稼働目的で Red Hat Quay をデプロイする方法
- イメージのプッシュ、プル、タグ付け、整理の方法など、Red Hat Quay のコンテナーイメージ管理の評価
- 可用性とスケーラビリティーの調査
- SSL/TLS 証明書を使用した高度な Red Hat Quay 概念実証環境をデプロイする方法

このドキュメントの主な目的以外にも、概念実証デプロイメントを使用して、Red Hat Quay が提供するさまざまな機能をテストできます。たとえば、スーパーユーザーの設定、リポジトリークォータ制限の設定、アクションログストレージ用の Splunk の有効化、脆弱性レポート用の Clair の有効化などです。このガイドの手順を実行した後に利用できる機能のリストについては、「次のステップ」セクションを参照してください。

この概念実証のデプロイ手順は、単一の物理または仮想マシン上で実行できます。

第1章 前提条件

- Red Hat Enterprise Linux (RHEL) 9
  - Red Hat Enterprise Linux (RHEL) 9 の最新バージョンを入手するには、Red Hat Enterprise Linux の ダウンロード を参照してください。
  - インストール手順は、Red Hat Enterprise Linux 8 の製品ドキュメント を参照してください。
- Red Hat への有効なサブスクリプション
- 2 つ以上の仮想 CPU
- 4 GB 以上の RAM
- テストシステムに約 30 GB のディスク容量。次のように分類できます。
  - Red Hat Enterprise Linux (RHEL) オペレーティングシステム用に約 10 GB のディスク容量。
  - 3 つのコンテナーを実行するための Docker ストレージ用に約 10 GB のディスク容量。
  - Red Hat Quay ローカルストレージ用に約 10 GB のディスク容量。



注記

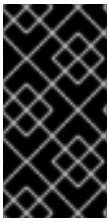
CEPH またはその他のローカルストレージでは、より多くのメモリーが必要になる場合があります。

サイジングについての詳細は Quay 3.x Sizing Guidlines を参照してください。

1.1. PODMAN のインストール

本書では、コンテナーを作成し、デプロイするために Podman を使用します。

Podman および関連技術の詳細は、Red Hat Enterprise Linux 9 のコンテナーの構築、実行、および管理 を参照してください。



重要

システムに Podman がインストールされていない場合は、同等の Docker コマンドを使用できる可能性がありますが、これは、推奨しません。Docker は Red Hat Quay 3.11 でテストされておらず、今後のリリースで非推奨になる予定です。Podman は、Red Hat Quay 3.11 の高可用性と本番環境品質が必要なデプロイメントに推奨されます。

次の手順を使用して、Podman をインストールします。

手順

- 次のコマンドを入力して、Podman をインストールします。

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

- または、コンテナーソフトウェアパッケージの完全なセットをプルする **container-tools** モジュールをインストールできます。

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

第2章 RED HAT QUAY の概念実証デプロイメントのための RED HAT ENTERPRISE LINUX の準備

以下の手順を使用して、Red Hat Quay の概念実証デプロイメント用に Red Hat Enterprise Linux (RHEL) を設定します。

2.1. RHEL サーバーのインストールおよび登録

以下の手順を使用して、Red Hat Quay の概念実証デプロイメント用に Red Hat Enterprise Linux (RHEL) サーバーを設定します。

手順

1. 最新の RHEL 9 サーバーをインストールします。最小インストール (シェルアクセスのみ) を実行するか、Server plus GUI (デスクトップが必要な場合) を実行できます。
2. Red Hat Subscription-Manager を使用して RHEL システムを Red Hat Customer Portal に登録およびサブスクライブする方法 の説明に従って、RHEL サーバーシステムを登録およびサブスクライブします。
3. 以下のコマンドを入力して、システムを登録し、利用可能なサブスクリプションを一覧表示します。利用可能な RHEL サーバーのサブスクリプションを選択し、プール ID に割り当て、最新のソフトウェアにアップグレードします。

```
# 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. レジストリー認証

以下の手順を使用して、Red Hat Quay の概念実証のためにレジストリーを認証します。

手順

1. Red Hat Container Registry の認証 手順に従って、**registry.redhat.io** への認証を設定します。認証を設定すると、**Quay** コンテナーをプルできます。

注記

これは、イメージが Quay.io でホストされていた以前のバージョンの Red Hat Quay とは異なります。

2. 次のコマンドを入力して、レジストリーにログインします。

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

ユーザー名 と **パスワード** を入力するよう求められます。

2.3. ファイアウォールの設定

システムでファイアウォールを実行している場合は、Red Hat Quay へのアクセスが許可されるルールを追加する必要がある場合があります。次の手順を使用して、概念実証のデプロイメントのためにファイアウォールを設定します。

手順

- 必要なコマンドは、システムにマップしたポートによって異なります。次に例を示します。

```
# 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 アドレスおよび命名サービス

相互に通信できるように Red Hat Quay でコンポーネントコンテナーを設定するには、いくつかの方法があります。次に例を示します。

- **コンテナーの IP アドレスの使用。Podman inspect** を使用してコンテナーの IP アドレスを特定し、接続文字列を指定するときに設定ツールで値を使用できます。次に例を示します。

```
$ sudo podman inspect -f "{{.NetworkSettings.IPAddress}}" postgresql-quay
```

  この方法は、コンテナーの IP アドレスが再起動後に変更されるためにホストの再起動の影響を受けます。

- **ネームサービスの使用。**デプロイメントをコンテナーの再起動後も存続させたい場合は、つまり IP アドレスが変更されることとなりますが、ネーミングサービスを実装できます。たとえば、dnsname プラグインは、コンテナーが名前で相互を解決できるように使用します。

- **ホストネットワークの使用。Podman run** コマンドを **--net=host** オプションと共に使用してから、アドレスを設定に指定する際にホストでコンテナーポートを使用できます。このオプションは、2 つのコンテナーが同じポートを使用する必要がある場合に、ポートの競合の影響を受けやすくなります。この方法は、推奨しません。

- **ポートマッピングの設定。**ポートマッピングを使用してホスト上のポートを公開し、これらのポートをホストの IP アドレスまたはホスト名と組み合わせて使用できます。

本書では、ポートマッピングを使用し、ホストシステムの静的 IP アドレスを使用することを前提としています。

表2.1 ポートマッピングの概念実証のサンプル

コンポーネント	ポートマッピング	アドレス
Quay	**-p 80:8080 -p 443:8443**	http://quay-server.example.com
Postgres for Quay	**-p 5432:5432**	quay-server.example.com:5432

コンポーネント	ポートマッピング	アドレス
Redis	**-p 6379:6379**	quay-server.example.com:6379
Postgres for Clair V4	**-p 5433:5432**	quay-server.example.com:5433
Clair V4	**-p 8081:8080**	http://quay-server.example.com:8081

第3章 RED HAT QUAY をデプロイするためのシステムの準備

Red Hat Quay の概念実証環境をデプロイするには、ポートマッピング、データベース、および Redis を設定してからレジストリーをデプロイする必要があります。以下の手順に従って、Red Hat Quay をデプロイするためにシステムを準備します。

3.1. RED HAT QUAY のポートマッピングの設定

ポートマッピングを使用してホスト上のポートを公開し、そのポートをホスト IP アドレスまたはホスト名と組み合わせて使用して、設定ツールのエンドポイントに移動できます。

手順

1. 次のコマンドを入力して、ホストシステムの静的 IP アドレスを取得します。

```
$ ip a
```

出力例

```
---
    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. 設定ツールのエンドポイントに到達するために使用する IP アドレスとローカルホスト名 (例: **quay-server.example.com**) を **/etc/hosts** ファイルに追加します。次のコマンドを入力すると、IP アドレスとホスト名が **/etc/hosts** ファイルに追加されたことを確認できます。

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

出力例

```
192.168.1.138 quay-server.example.com
```

3.2. データベースの設定

Red Hat Quay には、メタデータを保存するためのデータベースが必要です。このドキュメントでは、全体を通して PostgreSQL を使用しています。このデプロイメントでは、ローカルファイルシステム上のディレクトリーを使用してデータベースデータを永続化します。

PostgreSQL データベースをセットアップするには、次の手順を使用します。

手順

1. ここでは **$QUAY** 変数で示されているインストールフォルダーに、次のコマンドを入力して、データベースデータ用のディレクトリーを作成します。

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

2. 次のコマンドを入力して、適切な権限を設定します。

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

3. データベースデータのボリューム定義を使用して、ユーザー名、パスワード、およびデータベース名とポートを指定して、**Postgres** コンテナーを起動します。

```
$ 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-13:1-109
```

4. 次のコマンドを実行して、Postgres **pg_trgm** モジュールがインストールされていることを確認します。

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

注記

Quay コンテナーには **pg_trgm** モジュールが必要です。

3.3. REDIS の設定

Redis は、Red Hat Quay によってライブビルダーログに使用されるキーと値のストアです。

以下の手順を使用して、Red Hat Quay の概念実証用の **Redis** コンテナーをデプロイします。

手順

- 次のコマンドを入力して、ポートとパスワードを指定して **Redis** コンテナーを起動します。

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

第4章 RED HAT QUAY 設定ツールのデプロイ

Red Hat Quay 設定ツールをデプロイするには、次の手順を使用します。その後、レジストリーエンドポイントに移動し、レジストリー設定、データベース、Redis 接続パラメーターなどのすべてのコンポーネントの詳細を記述した設定ファイルを生成できます。

手順

1. 設定ファイルを生成するには、次のコマンドを入力して、**Quay** コンテナーを **設定** モードで実行します。パスワード (文字列 **secret** など) を指定する必要があります。

```
$ sudo podman run --rm -it --name quay_config -p 80:8080 -p 443:8443
registry.redhat.io/quay/quay-rhel8:v3.11.0 config secret
```

2. ブラウザーを使用して、**http://quay-server.example.com** で設定ツールのユーザーインターフェイスにアクセスします。

注記

本書では、**quay-server.example.com** ホスト名を **/etc/hosts** ファイルに設定しています。

3. ユーザー名とパスワードを指定してログイン
4. Red Hat Quay の設定 のステップ 1 で設定したユーザー名とパスワードでログインします。

注記

この手順に従った場合、ユーザー名は **quayconfig** で、パスワードは **secret** です。

4.1. RED HAT QUAY のセットアップ

Red Hat Quay 設定エディターで、次の認証情報を入力する必要があります。

- Basic configuration
- Server configuration
- Database
- Redis

4.1.1. Basic configuration

Basic configuration には、**Registry Title**、**Registry Title Short**、**Enterprise Logo URL**、および **Contact Information** フィールドがあります。

手順

デフォルト値が設定されていればそのデフォルト値を使用できます。

1. **Registry Title** に **Project Quay** と入力します。

2. **Registry Title Short** に、**Project Quay** と入力します。
3. オプション: **Enterprise Logo URL** に URL を入力します。
4. オプション: **URL**、**E-mail**、**IRC**、**Telephone** から 1 つ選択し、連絡先情報を入力します。

4.1.2. Server configuration

Server configuration には、**Server Hostname** フィールドとオプションの **TLS** フィールドがあります。

手順

- このデプロイメントの場合は、**quay-server.example.com** と入力します。

4.1.3. Database

Database セクションで、Red Hat Quay がメタデータを保存するために使用するデータベースの接続の詳細を指定します。

手順

1. **Database Type** に **Postgres** と入力します。
2. **Database Server** に、**quay-server.example.com:5432** と入力します。
3. **Username** に **quayuser** と入力します。
4. **Password** に **quaypass** と入力します。
5. **Database Name** に **quay** と入力します。

4.1.4. Redis

Redis の key-value ストアは、リアルタイムイベントとビルドログを保管するために使用されます。

手順

1. **Redis Hostname** に、**quay-server.example.com** と入力します。
2. **Redis port** に **6379** と入力します。これはデフォルト値です。
3. **Redis password** に、**strongpassword** と入力します。

4.2. 設定の検証およびダウンロード

必須フィールドをすべて設定したら、設定を検証します。

手順

- **Validate Configuration Changes** ボタンをクリックします。エラーが報告される場合は、設定が有効となり、Red Hat Quay がデータベースおよび Redis サーバーに接続できるまで、設定の編集を継続します。

検証後、**Configuration** ファイルをダウンロードします。設定エディターを実行している **Quay** コンテナーを停止します。

第5章 RED HAT QUAY のデプロイ

Red Hat Quay デプロイメントを設定したら、次の手順を使用してデプロイできます。

前提条件

- Red Hat Quay データベースが実行されている。
- Redis サーバーが実行されている。
- 有効な設定ファイルを生成している。
- 設定エディターを実行していた **Quay** コンテナーを停止している。

5.1. 設定フォルダーの準備

以下の手順を使用して、Red Hat Quay 設定フォルダーを準備します。

手順

1. Red Hat Quay 設定バンドルをコピーするディレクトリーを作成します。

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

2. 生成された Red Hat Quay 設定バンドルをディレクトリーにコピーします。

```
$ cp ~/Downloads/quay-config.tar.gz ~/config
```

3. そのディレクトリーに移動します。

```
$ cd $QUAY/config
```

4. Red Hat Quay 設定バンドルを解凍します。

```
$ tar xvf quay-config.tar.gz
```

5.2. イメージデータ用のローカルストレージの準備

次の手順を使用して、レジストリーイメージを保存するローカルファイルシステムを設定します。

手順

1. 次のコマンドを入力して、レジストリーイメージを保存するローカルディレクトリーを作成します。

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

2. レジストリーイメージを保存するディレクトリーを設定します。

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

5.3. RED HAT QUAY レジストリーのデプロイ

次の手順を使用して、**Quay** レジストリーコンテナーをデプロイします。

手順

1. 次のコマンドを入力して、設定データ用の適切なボリュームとイメージデータ用のローカルストレージを指定して、**Quay** レジストリーコンテナーを起動します。

```
$ 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:v3.11.0
```

第6章 RED HAT QUAY の使用

次の手順では、インターフェイスを使用して新しい組織とリポジトリーを作成する方法、および既存のリポジトリーを検索および参照する方法を示します。手順 3 の後に、コマンドラインインターフェイスを使用してレジストリーと対話し、イメージのプルおよびプッシュを実行できます。

1. ブラウザーを使用して、**http://quay-server.example.com** で Red Hat Quay レジストリーのユーザーインターフェイスにアクセスします (**quay-server.example.com** を **/etc/hosts** ファイルのホスト名として設定していることを前提とします)。
2. **Create Account** をクリックし、ユーザーを追加します (例: **quayadmin** とパスワード **password**)。
3. コマンドラインで、レジストリーにログインします。

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

出力例

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

6.1. RED HAT QUAY でのイメージのプッシュとプル

以下の手順を使用して、イメージを Red Hat Quay レジストリーにプッシュおよびプルします。

手順

1. Red Hat Quay レジストリーからイメージのプッシュおよびプルをテストするには、まず外部レジストリーからサンプルイメージをプルします。

```
$ sudo podman pull busybox
```

出力例

```
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. 次のコマンドを入力して、イメージのローカルコピーを表示します。

```
$ sudo podman images
```

出力例

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

3. 次のコマンドを入力してこのイメージにタグを付けます。これにより、イメージをレジストリーにプッシュする準備が整います。

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

4. イメージをレジストリーにプッシュします。この手順の後に、ブラウザーを使用して、リポジトリーでタグ付けされたイメージを確認できます。

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

出力例

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

5. コマンドラインからイメージへのアクセスをテストするには、まずイメージのローカルコピーを削除します。

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

6. 今度は Red Hat Quay レジストリーからイメージを再度プルします。

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

出力例

```
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
```

第7章 SSL/TLS 証明書を使用した概念実証デプロイメント

以下のセクションを使用して、SSL/TLS 証明書を使用した Red Hat Quay 概念実証デプロイメントを設定します。

7.1. SSL/TLS の使用

自己署名証明書を使用して Red Hat Quay を設定するには、認証局 (CA) と **ssl.cert** および **ssl.key** という名前のプライマリーキーファイルを作成する必要があります。

注記

以下の例では、**/etc/hosts** ファイルにエントリーを追加するなど、DNS または別の命名メカニズムを使用してサーバーホスト名 **quay-server.example.com** を設定していることを前提としています。詳細は、「Red Hat Quay のポートマッピングの設定」を参照してください。

7.1.1. 認証局の作成

認証局 (CA) を作成するには、次の手順を使用します。

手順

1. 次のコマンドを入力して、ルート CA キーを生成します。

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

2. 次のコマンドを入力して、ルート CA 証明書を生成します。

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

3. サーバーのホスト名など、証明書の要求に組み込まれる情報を入力します。以下に例を示します。

```
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
```

7.1.1.1. 証明書への署名

証明書に署名するには、次の手順を使用します。

手順

1. 次のコマンドを入力してサーバーキーを生成します。

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

2. 次のコマンドを入力して、署名リクエストを生成します。

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

3. サーバーのホスト名など、証明書の要求に組み込まれる情報を入力します。以下に例を示します。

```
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. 以下のようにサーバーのホスト名を指定して、設定ファイルの **openssl.cnf** を作成します。

openssl.cnf

```
[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
```

5. 設定ファイルを使用して、証明書 **ssl.cert** を生成します。

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

7.2. SSL/TLS の設定

SSL/TLS は、コマンドラインインターフェイス (CLI) または Red Hat Quay レジストリー UI を使用して設定できます。次のいずれかの手順を使用して、SSL/TLS を設定します。

7.2.1. Red Hat Quay UI を使用した SSL/TLS の設定

Red Hat Quay UI を使用して SSL/TLS を設定するには、次の手順を実行します。

コマンドラインインターフェイスを使用して SSL/TLS を設定するには、「コマンドラインインターフェイスを使用した SSL/TLS の設定」を参照してください。

前提条件

- 認証局を作成して証明書に署名している。

手順

1. **Quay** コンテナーを設定モードで起動します。

```
$ sudo podman run --rm -it --name quay_config -p 80:8080 -p 443:8443
registry.redhat.io/quay/quay-rhel8:v3.11.0 config secret
```

2. **Server Configuration** セクションで、**Red Hat Quay handles TLS**を選択します。前に作成した証明書ファイルと秘密鍵ファイルをアップロードし、**Server Hostname** 証明書の作成時に使用された値と一致することを確認します。

3. 更新された設定を検証およびダウンロードします。

4. 次のコマンドを入力して、**Quay** コンテナーを停止し、レジストリーを再起動します。

```
$ sudo podman rm -f quay
$ 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:v3.11.0
```

7.2.2. コマンドラインインターフェイスを使用した SSL/TLS の設定

CLI を使用して SSL/TLS を設定するには、次の手順を実行します。

前提条件

- 認証局を作成して証明書に署名している。

手順

1. 証明書ファイルとプライマリーキーファイルを設定ディレクトリーにコピーして、それぞれ **ssl.cert** と **ssl.key** という名前が付けられていることを確認します。

```
cp ~/ssl.cert ~/ssl.key $QUAY/config
```

2. 次のコマンドを入力して、**$QUAY/config** ディレクトリーに移動します。

```
$ cd $QUAY/config
```

3. **config.yaml** ファイルを編集し、Red Hat Quay が TLS/SSL を処理するように指定します。

config.yaml

```
...
SERVER_HOSTNAME: quay-server.example.com
...
PREFERRED_URL_SCHEME: https
...
```

4. オプション: 次のコマンドを入力して、rootCA.pem ファイルの内容を ssl.cert ファイルの末尾に追加します。

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

5. 次のコマンドを入力して、**Quay** コンテナーを停止します。

```
$ sudo podman stop quay
```

6. 次のコマンドを入力してレジストリーを再起動します。

```
$ 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:v3.11.0
```

7.3. SSL/TLS 設定のテスト

SSL/TLS 設定は、コマンドラインインターフェイス (CLI) または Red Hat Quay レジストリー UI を使用してテストできます。次のいずれかの手順を使用して、SSL/TLS 設定をテストします。

7.3.1. CLI を使用した SSL/TLS 設定のテスト

CLI を使用して SSL/TLS 設定をテストするには、次の手順を実行します。

手順

- 次のコマンドを入力して、SSL/TLS が有効な Red Hat Quay レジストリーへのログインを試行します。

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

出力例

```
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
```

1. Podman は自己署名証明書を信頼しないため、**--tls-verify=false** オプションを使用する必要があります。

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

出力例

```
Login Succeeded!
```

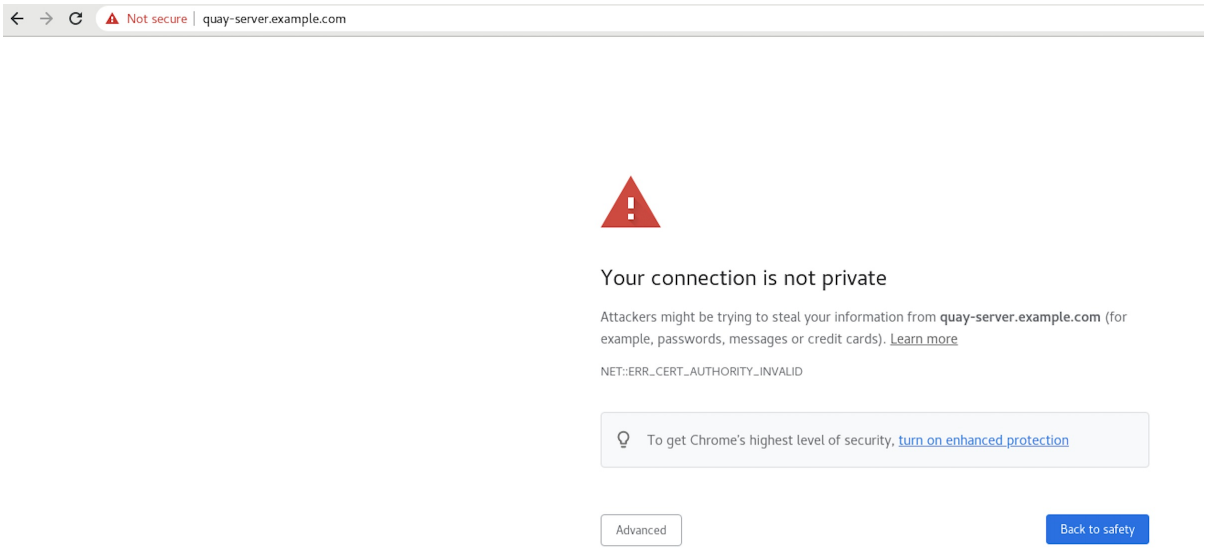
後のセクションで、ルート認証局を信頼するように Podman を設定します。

7.3.2. ブラウザーを使用した SSL/TLS 設定のテスト

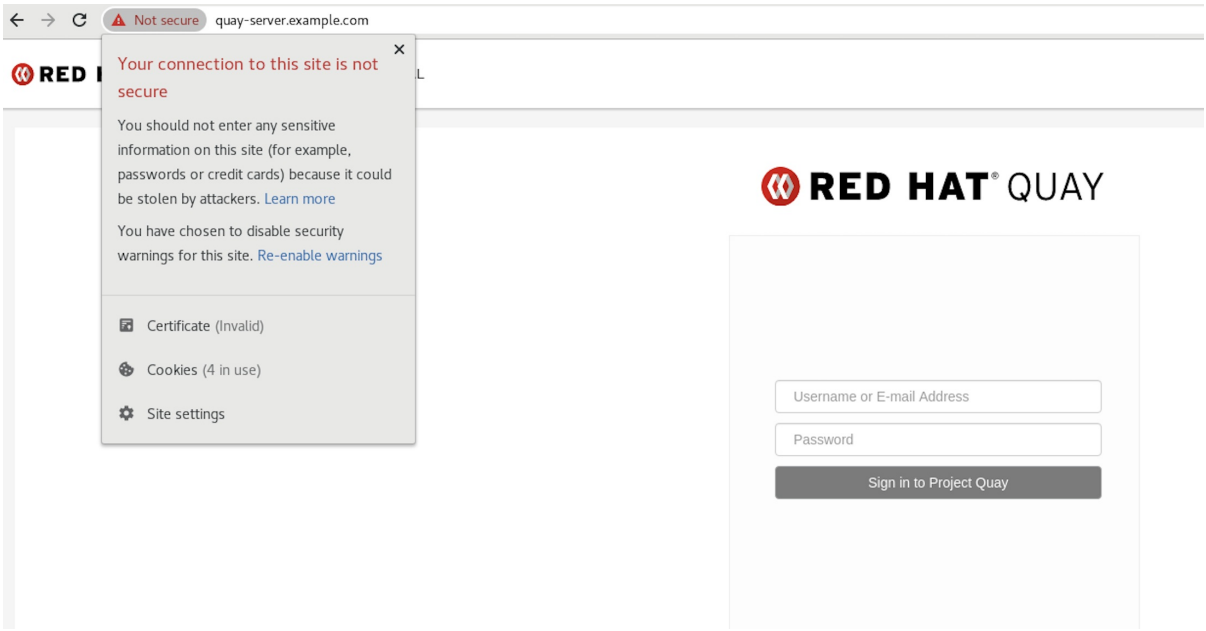
ブラウザーを使用して SSL/TLS 設定をテストするには、次の手順を実行します。

手順

1. Red Hat Quay レジストリーエンドポイント (例: **https://quay-server.example.com)** に移動します。正しく設定されている場合、潜在的なリスクについての警告がブラウザーに表示されます。



2. ログイン画面に進みます。接続が安全ではないという通知がブラウザーに表示されます。以下に例を示します。



次のセクションで、ルート認証局を信頼するように Podman を設定します。

7.4. 認証局を信頼するように PODMAN を設定する

Podman は、**/etc/containers/certs.d/** と **/etc/docker/certs.d/** の 2 つのパスを使用して認証局 (CA) ファイルを検出します。CA を信頼するように Podman を設定するには、次の手順を使用します。

手順

1. ルート CA ファイルを **/etc/containers/certs.d/** または **/etc/docker/certs.d/** のいずれかにコピーします。サーバーのホスト名によって決定される正確なパスを使用し、ファイルに **ca.crt** という名前を付けます。

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

2. Red Hat Quay レジストリーにログインするときに **--tls-verify=false** オプションを使用する必要がなくなったことを確認します。

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

出力例

```
Login Succeeded!
```

7.5. 認証局を信頼するようにシステムを設定

認証局を信頼するようにシステムを設定するには、次の手順を使用します。

手順

1. 次のコマンドを入力して、**rootCA.pem** ファイルをシステム全体の統合トラストストアにコピーします。

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

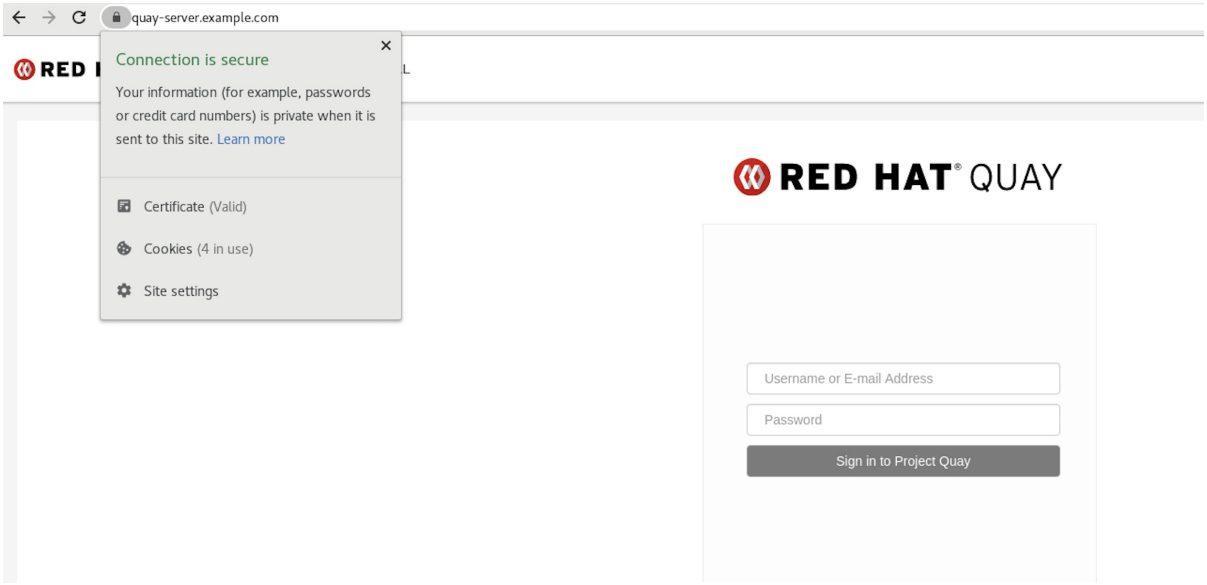
2. 次のコマンドを入力して、システム全体のトラストストア設定を更新します。

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

3. オプション: **trust list** コマンドを使用して、**Quay** サーバーが設定されていることを確認できます。

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

https://quay-server.example.com でレジストリーを参照すると、接続が安全であることを示すロックアイコンが表示されます。



4. **rootCA.pem** ファイルをシステム全体の信頼から削除するには、ファイルを削除して設定を更新します。

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

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

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

詳細は、RHEL 9 のドキュメントの 共有システム証明書の使用 を参照してください。

第8章 次のステップ

以下のセクションは、Red Hat Quay の概念実証バージョンをデプロイした後に役立ちます。以下の手順の多くは、概念実証デプロイメントで使用でき、Red Hat Quay の機能についての詳細情報を提供します。

- Red Hat Quay の使用。このガイドでは、次の概念について説明します。
  - ユーザーとリポジトリーの追加
  - イメージタグの使用
  - ビルドワーカーを使用した Dockerfile のビルド
  - ビルドトリガーのセットアップ
  - リポジトリーイベントの通知の追加
  - その他
- Red Hat Quay の管理。このガイドでは、次の概念について説明します。
  - SSL/TLS の使用
  - アクションログストレージの設定
  - Clair セキュリティースキャナーの設定
  - リポジトリーのミラーリング
  - IPv6 およびデュアルスタックデプロイメント
  - Red Hat Quay の OIDC の設定
  - Geo レプリケーション
  - その他