



## Red Hat Quay 3.11

# Red Hat Quay のアップグレード

Red Hat Quay のアップグレード



# Red Hat Quay 3.11 Red Hat Quay のアップグレード

---

Red Hat Quay のアップグレード

## 法律上の通知

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 概要

Red Hat Quay のアップグレード

---

## 目次

<b>第1章 アップグレードの概要</b> .....	<b>3</b>
<b>第2章 RED HAT QUAY OPERATOR のアップグレードの概要</b> .....	<b>4</b>
2.1. OPERATOR LIFECYCLE MANAGER	4
2.2. RED HAT QUAY OPERATOR のアップグレード	4
2.3. QUAYREGISTRY リソースのアップグレード	6
2.4. QUAYECOSYSTEM のアップグレード	6
<b>第3章 スタンドアロンアップグレード</b> .....	<b>9</b>
3.1. イメージへのアクセス	10
3.2. 3.10.Z から 3.11.Z へのアップグレード	10
3.3. 3.9.Z から 3.11.Z へのアップグレード	10
3.4. 3.9.Z から 3.10.Z へのアップグレード	10
3.5. 3.8.Z から 3.10.Z へのアップグレード	10
3.6. 3.7.Z から 3.10.Z へのアップグレード	11
3.7. OPENSIFT CONTAINER PLATFORM 上の RED HAT QUAY の GEO レプリケーションデプロイメントのアップグレード	11
<b>第4章 QUAY BRIDGE OPERATOR のアップグレード</b> .....	<b>15</b>
4.1. OPENSIFT CONTAINER PLATFORM 上の RED HAT QUAY の GEO レプリケーションデプロイメントのアップグレード	15
<b>第5章 RED HAT QUAY のダウングレード</b> .....	<b>18</b>



## 第1章 アップグレードの概要

Red Hat Quay のアップグレード手順は、使用しているインストールの種類によって異なります。

Red Hat Quay Operator は、Red Hat Quay クラスターをデプロイし、管理する簡単な方法を提供します。これは、Red Hat Quay の OpenShift へのデプロイで推奨の手順です。

Quay Operator を使用した Quay のアップグレードの項で説明されているように、Red Hat Quay Operator のアップグレードは、[Operator Lifecycle Manager \(OLM\)](#) を使用する必要があります。

Red Hat Quay および Clair の概念実証または高可用性のインストールをアップグレードする手順は、スタンドアロンでのアップグレードに記載されています。

## 第2章 RED HAT QUAY OPERATOR のアップグレードの概要



### 注記

現在、Red Hat Quay Operator のアップグレードは、IBM Power および IBM Z ではサポートされていません。

Red Hat Quay Operator は、**シンクロナイズドバージョンニング** スキームに従います。つまり、Red Hat Operator の各バージョンは Quay とその管理するコンポーネントに関連付けられます。**QuayRegistry** カスタムリソースには、Red Hat Quay が **deploy** するバージョンを設定するフィールドはありません。Operator は、すべてのコンポーネントを1つのバージョンのみデプロイできます。このスキームは、すべてのコンポーネントが適切に連携し、Kubernetes で Red Hat Quay の多数に渡るバージョンのライフサイクルを管理する方法を把握する必要がある Operator の複雑性を軽減するために、選択されました。

### 2.1. OPERATOR LIFECYCLE MANAGER

Red Hat Quay Operator は [Operator Lifecycle Manager \(OLM\)](#) を使用してインストールし、アップグレードする必要があります。デフォルトの **approvalStrategy: Automatic** で **Subscription** を作成する場合、OLM は新規バージョンが利用可能になると常に Red Hat Quay Operator を自動的にアップグレードします。



### 警告

Red Hat Quay Operator が Operator Lifecycle Manager によってインストールされている場合、自動または手動のアップグレードをサポートするように設定されることがあります。このオプションは、インストール時に Red Hat Quay Operator の **OperatorHub** ページに表示されます。これは、Red Hat Quay Operator **Subscription** オブジェクトの **ApprovalStrategy** フィールドでも確認できます。**Automatic** を選択すると、新規 Operator バージョンがリリースされるたびに Red Hat Quay Operator が自動的にアップグレードされます。これが望ましくない場合は、**Manual** 承認ストラテジーを選択する必要があります。

### 2.2. RED HAT QUAY OPERATOR のアップグレード

インストールされた Operator を OpenShift Container Platform にアップグレードする一般的な方法については、[インストールされた Operator のアップグレード](#) を参照してください。

一般的に、Red Hat Quay は以前の (N-1) マイナーバージョンからのアップグレードのみをサポートしています。たとえば、Red Hat Quay 3.0.5 から最新バージョンの 3.5 への直接アップグレードはサポートされていません。代わりに、次のようにアップグレードする必要があります。

1. 3.0.5 → 3.1.3
2. 3.1.3 → 3.2.2
3. 3.2.2 → 3.3.4
4. 3.3.4 → 3.4.z



## 5. 3.4.z → 3.5.z

この作業は、必要なデータベースの移行が正しく実行され、適切な順序でアップグレードが行われるようにするために必要です。

場合によっては、Red Hat Quay は、以前の (N-2、N-3) マイナーバージョンからの直接のシングルステップアップグレードをサポートします。これにより、古いリリースを使用している顧客のアップグレード手順が簡素化されます。Red Hat Quay 3.11 では、次のアップグレードパスがサポートされています。

- 3.9.z → 3.11.z
- 3.10.z → 3.11.z



## 注記

3.8.z から 3.11 へのアップグレードはサポートされていません。ユーザーはまず 3.9 または 3.10 にアップグレードし、次に 3.11 にアップグレードする必要があります。

Red Hat Quay のスタンドアロンデプロイメントを使用しているユーザーが 3.11 にアップグレードしたい場合は、[スタンドアロンアップグレード](#) ガイドを参照してください。

## 2.2.1. Red Hat Quay のアップグレード

Red Hat Quay をあるマイナーバージョンから次のマイナーバージョン (たとえば、3.10 → 3.11) に更新するには、Red Hat Quay Operator の更新チャンネルを変更する必要があります。

**z** ストリームのアップグレード (3.10.1 → 3.10.2 など) の場合、更新は、ユーザーがインストール時に最初に選択したメジャー/マイナーチャンネルでリリースされます。**z** ストリームのアップグレードを実行する手順は、上記のように **approvalStrategy** によって異なります。承認ストラテジーが **Automatic** に設定されている場合、Red Hat Quay Operator は自動的に最新の **z** ストリームにアップグレードします。この場合、ダウンタイムがほとんどない (またはまったくない) 新しい **z** ストリームへの Red Hat Quay の自動ローリング更新が行われます。それ以外の場合は、インストールを開始する前に更新を手動で承認する必要があります。

## 2.2.2. Red Hat Quay Operator の更新チャンネルの変更

インストールされた Operator のサブスクリプションは、Operator の更新を追跡して受け取るために使用される更新チャンネルを指定します。Red Hat Quay Operator をアップグレードして新規チャンネルからの更新の追跡および受信を開始するには、インストールされた Red Hat Quay Operator の **Subscription** タブで更新チャンネルを変更します。**Automatic** 承認ストラテジーのあるサブスクリプションの場合、アップグレードは自動的に開始し、インストールされた Operator をリスト表示したページでモニターできます。

## 2.2.3. 保留中の Operator アップグレードの手動による承認

インストールされた Operator のサブスクリプションの承認ストラテジーが **Manual** に設定されている場合は、新規の更新が現在の更新チャンネルにリリースされると、インストールを開始する前に更新を手動で承認する必要があります。Red Hat Quay Operator に保留中のアップグレードがある場合、このステータスはインストールされた Operator のリストに表示されます。Red Hat Quay Operator の **Subscription** タブで、インストール計画をプレビューし、アップグレードに利用可能なリソースとして一覧表示されるリソースを確認できます。問題がなければ、**Approve** をクリックし、Installed Operators を一覧表示したページに戻り、アップグレードの進捗を監視します。

以下のイメージには、更新 **Channel**、**Approval** ストラテジー、**Upgrade status** および **InstallPlan** などの UI の **Subscription** タブが表示されています。

Installed Operator のリストは、現在の Quay インストールの概要を提供します。

## 2.3. QUAYREGISTRY リソースのアップグレード

Red Hat Quay Operator を起動すると、監視するように設定されている namespace にある **QuayRegistries** をすぐに検索します。見つかった場合は、次のロジックが使用されます。

- **status.currentVersion** が設定されていない場合は、通常通り調整を行います。
- **status.currentVersion** が Operator のバージョンと等しい場合は、通常通り調整を行います。
- **status.currentVersion** が Operator のバージョンと一致しない場合は、アップグレードできるかどうかを確認します。可能な場合は、アップグレードタスクを実行し、完了後に **status.currentVersion** を Operator のバージョンに設定します。アップグレードできない場合は、エラーを返し、**QuayRegistry** とそのデプロイされた Kubernetes オブジェクトのみを残します。

## 2.4. QUAYECOSYSTEM のアップグレード

アップグレードは、**QuayEcosystem** API を使用して限られた設定を行っていた旧バージョンの Operator からサポートされています。移行が予期せず行われるようにするには、移行を行うために特別なラベルを **QuayEcosystem** に適用する必要があります。Operator が管理するための新しい

**QuayRegistry** が作成されますが、古い **QuayEcosystem** は手動で削除されるまで残り、何か問題が発生した場合にロールバックして Quay にアクセスできるようになります。既存の **QuayEcosystem** を新しい **QuayRegistry** に移行するには、次の手順を実行します。

## 手順

1. **"quay-operator/migrate": "true"** を **QuayEcosystem** の **metadata.labels** に追加します。

```
$ oc edit quayecosystem <quayecosystemname>
```

```
metadata:
  labels:
    quay-operator/migrate: "true"
```

2. **QuayRegistry** が **QuayEcosystem** と同じ **metadata.name** で作成されるまで待機します。**QuayEcosystem** にはラベル **"quay-operator/migration-complete": "true"** のマークが付けられます。
3. 新しい **QuayRegistry** の **status.registryEndpoint** が設定されたら、Red Hat Quay にアクセスし、すべてのデータと設定が正常に移行されたことを確認します。
4. すべてが正常に動作する場合は **QuayEcosystem** を削除でき、Kubernetes ガベージコレクションによって古いリソースがすべてクリーンアップされます。

### 2.4.1. QuayEcosystem アップグレードを元に戻す

**QuayEcosystem** から **QuayRegistry** への自動アップグレード時に問題が発生した場合は、以下の手順を実行して **QuayEcosystem** の使用に戻します。

## 手順

1. UI または **kubectl** のいずれかを使用して **QuayRegistry** を削除します。

```
$ kubectl delete -n <namespace> quayregistry <quayecosystem-name>
```

2. **Route** を使用して外部アクセスを提供していた場合は、UI や **kubectl** を使用して元の **Service** を指すように **Route** を変更します。



## 注記

**QuayEcosystem** が PostgreSQL データベースを管理している場合、アップグレードプロセスにより、アップグレードされた Operator が管理する新しい Postgres データベースにデータが以降されます。古いデータベースは変更または削除されませんが、移行が完了すると Quay はこのデータベースを使用しなくなります。データの移行中に問題が発生した場合は、アップグレードプロセスを終了し、データベースをマネージド外コンポーネントとして継続して使用することが推奨されます。

### 2.4.2. アップグレードでサポートされる QuayEcosystem 設定

**QuayEcosystem** コンポーネントの移行が失敗するかサポートされていない場合、Red Hat Quay Operator はログと **status.conditions** でエラーを報告します。アンマネージドコンポーネントを移行する場合、Kubernetes リソースを導入する必要がなく、必要な値はすべて Red Hat Quay の **config.yaml** で指定されているため、正常に移行できるはずです。

## Database

一時データベースはサポートされません (**volumeSize** フィールドを設定する必要があります)。

## Redis

特別な設定は必要ありません。

## External Access

パススルー **Route** アクセスのみが自動移行でサポートされます。他の方法には手動移行が必要です。

- ホスト名のない **LoadBalancer: QuayEcosystem** にラベル "**quay-operator/migration-complete**": "**true**" が付けられた後、Kubernetes が **Service** をガベージコレクションしてロードバランサーを削除するのを防ぐため、**QuayEcosystem** を削除する **前** に、既存の **Service** から **metadata.ownerReferences** フィールドを削除します。新規 **Service** は **metadata.name** 形式の **<QuayEcosystem-name>-quay-app** で作成されます。既存の **Service** の **spec.selector** を新しい **Service** の **spec.selector** に合わせて編集することで、古いロードバランサーのエンドポイントへのトラフィックが新しい Pod に誘導されるようになります。これで古い **Service** を管理します。Quay Operator はこれを管理しません。
- カスタムホスト名を持つ **LoadBalancer/NodePort/Ingress**: タイプ **LoadBalancer** の新規 **Service** は **metadata.name** 形式の **<QuayEcosystem-name>-quay-app** で作成されます。新しい **Service** が提供する **status.loadBalancer** エンドポイントを指すように、DNS 設定を変更します。

## Clair

特別な設定は必要ありません。

## オブジェクトストレージ

**QuayEcosystem** には管理オブジェクトストレージコンポーネントがないため、オブジェクトストレージには常に管理外のマークが付けられます。ローカルストレージはサポートされません。

## リポジトリのミラーリング

特別な設定は必要ありません。

## 第3章 スタンドアロンアップグレード

一般的に、Red Hat Quay は以前の (N-1) マイナーバージョンからのアップグレードのみをサポートしています。たとえば、Red Hat Quay 3.0.5 から最新バージョンの 3.5 への直接アップグレードはサポートされていません。代わりに、次のようにアップグレードする必要があります。

1. 3.0.5 → 3.1.3
2. 3.1.3 → 3.2.2
3. 3.2.2 → 3.3.4
4. 3.3.4 → 3.4.z
5. 3.4.z → 3.5.z

この作業は、必要なデータベースの移行が正しく実行され、適切な順序でアップグレードが行われるようにするために必要です。

場合によっては、Red Hat Quay は、以前の (N-2、N-3) マイナーバージョンからの直接のシングルステップアップグレードをサポートします。以前のマイナーバージョンのみをアップグレードする通常のアップグレードに対するこの例外により、古いリリースを使用している顧客のアップグレード手順が簡素化されます。Red Hat Quay 3.11 では、次のアップグレードパスがサポートされています。

- 3.9.z → 3.11.z
- 3.10.z → 3.11.z



### 注記

3.8.z から 3.11 へのアップグレードはサポートされていません。ユーザーはまず 3.9 または 3.10 にアップグレードし、次に 3.11 にアップグレードする必要があります。

Red Hat Quay Operator をアップグレードするユーザーは、[Red Hat Quay Operator のアップグレードの概要](#) を参照してください。

このドキュメントでは、各アップグレードに必要な手順を説明します。現在のバージョンを決定し、現在のバージョンから順に、目標とするバージョンへとステップを踏んで進めていきます。

- [3.10.z から 3.11.z へのアップグレード](#)
- [3.9.z から 3.11.z へのアップグレード](#)
- [3.9.z から 3.10.z へのアップグレード](#)
- [3.8.z から 3.10.z へのアップグレード](#)
- [3.7.z から 3.10.z へのアップグレード](#)
- [3.8.z から 3.9.z へのアップグレード](#)

個々のリリースの機能に関する情報は、[Red Hat Quay リリースノート](#) を参照してください。

手動アップグレードの一般的な手順は、以下のとおりです。

1. Quay および Clair コンテナを停止する

2. データベースとイメージストレージをバックアップする (任意ではあるが推奨)
3. 新バージョンのイメージを使用して Clair を起動する
4. Clair が接続を受け入れる準備ができるまで待ってから、新しいバージョンの Quay を起動する

## 3.1. イメージへのアクセス

バージョン 3.4.0 以降の Red Hat Quay イメージは、[registry.redhat.io](https://registry.redhat.io) および [registry.access.redhat.com](https://registry.access.redhat.com) から入手でき、認証は [Red Hat Container Registry Authentication](#) で説明されているようにセットアップされます。

## 3.2. 3.10.Z から 3.11.Z へのアップグレード

### 3.2.1. ターゲットイメージ

- **Quay:** [registry.redhat.io/quay/quay-rhel8:v3.11.0](https://registry.redhat.io/quay/quay-rhel8:v3.11.0)
- **Clair:** [registry.redhat.io/quay/clair-rhel8:v3.11.0](https://registry.redhat.io/quay/clair-rhel8:v3.11.0)
- **PostgreSQL:** [registry.redhat.io/rhel8/postgresql-13:1-109](https://registry.redhat.io/rhel8/postgresql-13:1-109)
- **Redis:** [registry.redhat.io/rhel8/redis-6:1-110](https://registry.redhat.io/rhel8/redis-6:1-110)

## 3.3. 3.9.Z から 3.11.Z へのアップグレード

### 3.3.1. ターゲットイメージ

- **Quay:** [registry.redhat.io/quay/quay-rhel8:v3.11.0](https://registry.redhat.io/quay/quay-rhel8:v3.11.0)
- **Clair:** [registry.redhat.io/quay/clair-rhel8:v3.11.0](https://registry.redhat.io/quay/clair-rhel8:v3.11.0)
- **PostgreSQL:** [registry.redhat.io/rhel8/postgresql-13:1-109](https://registry.redhat.io/rhel8/postgresql-13:1-109)
- **Redis:** [registry.redhat.io/rhel8/redis-6:1-110](https://registry.redhat.io/rhel8/redis-6:1-110)

## 3.4. 3.9.Z から 3.10.Z へのアップグレード

### 3.4.1. ターゲットイメージ

- **Quay:** [registry.redhat.io/quay/quay-rhel8:v3.10.0](https://registry.redhat.io/quay/quay-rhel8:v3.10.0)
- **Clair:** [registry.redhat.io/quay/clair-rhel8:v3.10.0](https://registry.redhat.io/quay/clair-rhel8:v3.10.0)
- **PostgreSQL:** [registry.redhat.io/rhel8/postgresql-13:1-109](https://registry.redhat.io/rhel8/postgresql-13:1-109)
- **Redis:** [registry.redhat.io/rhel8/redis-6:1-110](https://registry.redhat.io/rhel8/redis-6:1-110)

## 3.5. 3.8.Z から 3.10.Z へのアップグレード

### 3.5.1. ターゲットイメージ

- **Quay:** registry.redhat.io/quay/quay-rhel8:v3.10.0
- **Clair:** registry.redhat.io/quay/clair-rhel8:v3.10.0
- **PostgreSQL:** registry.redhat.io/rhel8/postgresql-13:1-109
- **Redis:** registry.redhat.io/rhel8/redis-6:1-110

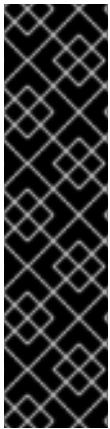
### 3.6. 3.7.Z から 3.10.Z へのアップグレード

#### 3.6.1. ターゲットイメージ

- **Quay:** registry.redhat.io/quay/quay-rhel8:v3.10.0
- **Clair:** registry.redhat.io/quay/clair-rhel8:v3.10.0
- **PostgreSQL:** registry.redhat.io/rhel8/postgresql-13:1-109
- **Redis:** registry.redhat.io/rhel8/redis-6:1-110

## 3.7. OPENSIFT CONTAINER PLATFORM 上の RED HAT QUAY の GEO レプリケーションデプロイメントのアップグレード

以下の手順を実行して、Red Hat Quay の Geo レプリケーションデプロイメントをアップグレードします。



#### 重要

- Red Hat Quay の Geo レプリケーションデプロイメントを次の y-stream リリース (例: Red Hat Quay 3.7 → Red Hat Quay 3.8) または Geo レプリケーションデプロイメントにアップグレードする場合は、アップグレードを実行する前に操作を停止する必要があります。
- y-stream リリースを次のリリースにアップグレードする場合は、アップグレード中にダウンタイムが断続的に発生します。
- アップグレードする前に、Red Hat Quay デプロイメントをバックアップすることが強く推奨されます。

#### 前提条件

- **registry.redhat.io** にログインしている。



#### 手順

この手順では、3 つ以上のシステムで Red Hat Quay サービスを実行していることを前提としています。詳細は、[Red Hat Quay の高可用性の準備](#) を参照してください。

1. Red Hat Quay インスタンスを実行している各システムですべての Red Hat Quay インスタンスのリストを取得します。
  - a. システム A で以下のコマンドを入力して、Red Hat Quay インスタンスを表示します。

```
$ sudo podman ps
```

### 出力例

```
CONTAINER ID IMAGE COMMAND CREATED
STATUS PORTS NAMES
ec16ece208c0 registry.redhat.io/quay/quay-rhel8:v3.7.0 registry 6 minutes ago Up
6 minutes ago 0.0.0.0:80->8080/tcp, 0.0.0.0:443->8443/tcp quay01
```

- b. System B で以下のコマンドを入力して、Red Hat Quay インスタンスを表示します。

```
$ sudo podman ps
```

### 出力例

```
CONTAINER ID IMAGE COMMAND CREATED
STATUS PORTS NAMES
7ae0c9a8b37d registry.redhat.io/quay/quay-rhel8:v3.7.0 registry 5 minutes ago Up
2 seconds ago 0.0.0.0:82->8080/tcp, 0.0.0.0:445->8443/tcp quay02
```

- c. System C で以下のコマンドを入力して、Red Hat Quay インスタンスを表示します。

```
$ sudo podman ps
```

### 出力例

```
CONTAINER ID IMAGE COMMAND CREATED
STATUS PORTS NAMES
e75c4aebfee9 registry.redhat.io/quay/quay-rhel8:v3.7.0 registry 4 seconds ago Up
4 seconds ago 0.0.0.0:84->8080/tcp, 0.0.0.0:447->8443/tcp quay03
```

2. 各システムの Red Hat Quay インスタンスをすべて一時的にシャットダウンします。
- a. システム A で以下のコマンドを入力して、Red Hat Quay インスタンスをシャットダウンします。

```
$ sudo podman stop ec16ece208c0
```

- b. System B で以下のコマンドを入力して、Red Hat Quay インスタンスをシャットダウンします。

```
$ sudo podman stop 7ae0c9a8b37d
```

- c. System C で以下のコマンドを入力して、Red Hat Quay インスタンスをシャットダウンします。

```
$ sudo podman stop e75c4aebfee9
```

3. 各システムで、最新の Red Hat Quay バージョン (Red Hat Quay 3.11 など) を入手します。
- a. システム A で以下のコマンドを入力して、最新の Red Hat Quay バージョンを取得します。



```
$ sudo podman pull registry.redhat.io/quay/quay-rhel8:v3.8.0
```

- b. システム B で以下のコマンドを入力して、最新の Red Hat Quay バージョンを取得します。

```
$ sudo podman pull registry.redhat.io/quay/quay-rhel8:v3.8.0
```

- c. システム C で以下のコマンドを入力して、最新の Red Hat Quay バージョンを取得します。

```
$ sudo podman pull registry.redhat.io/quay/quay-rhel8:v3.8.0
```

4. 高可用性 Red Hat Quay デプロイメントのシステム A で、新しいイメージバージョン (例: Red Hat Quay 3.11) を実行します。

```
# sudo podman run --restart=always -p 443:8443 -p 80:8080 \
--sysctl net.core.somaxconn=4096 \
--name=quay01 \
-v /mnt/quay/config:/conf/stack:Z \
-v /mnt/quay/storage:/datastorage:Z \
-d registry.redhat.io/quay/quay-rhel8:v3.8.0
```

5. 新しい Red Hat Quay コンテナがシステム A で完全に動作可能になるまで待ちます。コンテナのステータスは、次のコマンドを実行すると確認できます。

```
$ sudo podman ps
```

## 出力例

```
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
70b9f38c3fb4 registry.redhat.io/quay/quay-rhel8:v3.8.0 registry 2 seconds ago Up 2
seconds ago 0.0.0.0:82->8080/tcp, 0.0.0.0:445->8443/tcp quay01
```

6. オプション: Red Hat Quay UI に移動して、Red Hat Quay が完全に動作していることを確認します。
7. システム A 上の Red Hat Quay が完全に動作可能であることを確認したら、System B および System C で新しいイメージバージョンを実行します。
- a. 高可用性 Red Hat Quay デプロイメントのシステム B で、新しいイメージバージョン (例: Red Hat Quay 3.11) を実行します。

```
# sudo podman run --restart=always -p 443:8443 -p 80:8080 \
--sysctl net.core.somaxconn=4096 \
--name=quay02 \
-v /mnt/quay/config:/conf/stack:Z \
-v /mnt/quay/storage:/datastorage:Z \
-d registry.redhat.io/quay/quay-rhel8:v3.8.0
```

- b. 高可用性 Red Hat Quay デプロイメントの System C で、新しいイメージバージョン (例: Red Hat Quay 3.11) を実行します。

```
# sudo podman run --restart=always -p 443:8443 -p 80:8080 \  
--sysctl net.core.somaxconn=4096 \  
--name=quay03 \  
-v /mnt/quay/config:/conf/stack:Z \  
-v /mnt/quay/storage:/datastorage:Z \  
-d registry.redhat.io/quay/quay-rhel8:v3.8.0
```

8. 次のコマンドを入力して、システム B およびシステム C のコンテナのステータスを確認できます。

```
$ sudo podman ps
```

## 第4章 QUAY BRIDGE OPERATOR のアップグレード

Quay Bridge Operator (QBO) をアップグレードするには、Subscription タブの Channel Subscription 更新チャンネルを目的のチャンネルに変更します。

QBO をバージョン 3.5 から 3.7 にアップグレードする場合は、いくつかの追加の手順が必要です。

1. 新しい **QuayIntegration** カスタムリソースを作成する必要があります。これは、Web コンソールまたはコマンドラインから実行できます。

### upgrade-quay-integration.yaml

```
- apiVersion: quay.redhat.com/v1
  kind: QuayIntegration
  metadata:
    name: example-quayintegration-new
  spec:
    clusterID: openshift 1
    credentialsSecret:
      name: quay-integration
      namespace: openshift-operators
    insecureRegistry: false
    quayHostname: https://registry-quay-quay35.router-default.apps.cluster.openshift.com
```

- 1** **clusterID** が既存の **QuayIntegration** リソースの値と一致することを確認してください。

2. 新しい **QuayIntegration** カスタムリソースを作成します。

```
$ oc create -f upgrade-quay-integration.yaml
```

3. 古い **QuayIntegration** カスタムリソースを削除します。
4. 古い **mutatingwebhookconfigurations** を削除します。

```
$ oc delete mutatingwebhookconfigurations.admissionregistration.k8s.io quay-bridge-operator
```

### 4.1. OPENSIFT CONTAINER PLATFORM 上の RED HAT QUAY の GEO レプリケーションデプロイメントのアップグレード

OpenShift Container Platform デプロイメント上の geo レプリケートされた Red Hat Quay をアップグレードするには、次の手順を使用します。



## 重要

- OpenShift Container Platform デプロイメント上の geo レプリケートされた Red Hat Quay を次の y-stream リリース (例: Red Hat Quay 3.7 → Red Hat Quay 3.8) にアップグレードする場合は、アップグレードする前に操作を停止する必要があります。
- y-stream リリースを次のリリースにアップグレードする場合は、アップグレード中にダウンタイムが断続的に発生します。
- アップグレードする前に、OpenShift Container Platform デプロイメント上の Red Hat Quay をバックアップすることを強く推奨します。



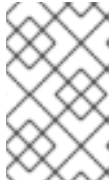
## 手順

この手順は、3 つ以上のシステムで Red Hat Quay レジストリーを実行していることを前提としています。この手順では、**System A**、**System B**、および **System C** という名前の 3 つのシステムを想定します。**System A** は、Red Hat Quay Operator がデプロイされるプライマリーシステムとして機能します。

1. システム B およびシステム C で、Red Hat Quay レジストリーをスケールダウンします。これを行うには、自動スケーリングを無効にし、Red Hat Quay、ミラーワーカー、および Clair (マネージドの場合) のレプリカ数をオーバーライドします。次の **quayregistry.yaml** ファイルを参照として使用します。

```
apiVersion: quay.redhat.com/v1
kind: QuayRegistry
metadata:
  name: registry
  namespace: ns
spec:
  components:
    ...
    - kind: horizontalpodautoscaler
      managed: false ①
    - kind: quay
      managed: true
      overrides: ②
        replicas: 0
    - kind: clair
      managed: true
      overrides:
        replicas: 0
    - kind: mirror
      managed: true
      overrides:
        replicas: 0
    ...
```

- ① **Quay**、**Clair**、および **Mirroring** ワーカーの自動スケーリングを無効にします。
- ② データベースおよびオブジェクトストレージにアクセスするコンポーネントのレプリカ数を 0 に設定します。



### 注記

Red Hat Quay レジストリーがシステム A で実行されている状態を維持する必要があります。システム A の **quayregistry.yaml** ファイルは更新しないでください。

2. **registry-quay-app**、**registry-quay-mirror**、および **registry-clair-app** Pod が消えるまで待機します。以下のコマンドを入力してステータスを確認します。

```
oc get pods -n <quay-namespace>
```

### 出力例

```
quay-operator.v3.7.1-6f9d859bd-p5ftc      1/1   Running   0          12m
quayregistry-clair-postgres-7487f5bd86-xnxpr  1/1   Running   1 (12m ago) 12m
quayregistry-quay-app-upgrade-xq2v6       0/1   Completed 0          12m
quayregistry-quay-redis-84f888776f-hhgms   1/1   Running   0          12m
```

3. システム A で、最新の y-stream バージョンへの Red Hat Quay のアップグレードを開始します。これは手動プロセスです。インストールされた Operator のアップグレードの詳細は、[インストールされた Operator のアップグレード](#) を参照してください。Red Hat Quay アップグレードパスの詳細は、[Red Hat Quay Operator のアップグレード](#) を参照してください。
4. 新しい Red Hat Quay レジストリーがインストールされると、クラスター上で必要なアップグレードが自動的に完了します。その後、新しい Red Hat Quay Pod は、最新の y-stream バージョンで起動します。さらに、新しい **Quay** Pod がスケジューラされ、起動します。
5. Red Hat Quay UI に移動して、更新が適切に機能していることを確認します。
  - a. **OpenShift** コンソールで **Operators** → **Installed Operators** に移動し、**Registry Endpoint** リンクをクリックします。



### 重要

Red Hat Quay UI が利用可能になるまで、次の手順を実行しないでください。システム A で UI が利用可能になるまで、システム B およびシステム C で Red Hat Quay レジストリーをアップグレードしないでください。

6. システム A で更新が適切に機能したことを確認した後、システム B とシステム C で Red Hat Quay のアップグレードを開始します。Operator のアップグレードにより、Red Hat Quay インストールがアップグレードされ、Pod が再起動します。



### 注記

データベーススキーマが新しい y-stream インストールに適したものであるため、システム B とシステム C の新しい Pod がすぐに起動します。

## 第5章 RED HAT QUAY のダウングレード

Red Hat Quay は、以前の z-stream バージョン (3.7.2 → 3.7.1 など) へのロールバックまたはダウングレードのみをサポートします。以前の y-stream バージョン (3.7.0 → 3.6.0) へのロールバックはサポートされていません。これは、Red Hat Quay の更新に、Red Hat Quay の新しいバージョンにアップグレードするときに適用されるデータベーススキーマのアップグレードが含まれている可能性があるためです。データベーススキーマのアップグレードでは下位互換性は保証されていません。



### 重要

以前の z-stream へのダウングレードは、Operator ベースのデプロイメントでも仮想マシンベースのデプロイメントでも推奨もサポートもされていません。ダウングレードは、非常事態でのみ行う必要があります。Red Hat Quay サポートおよび開発チームと協力して Red Hat Quay デプロイメントをロールバックするかどうかを決定する必要があります。詳細は、Red Hat Quay サポートにお問い合わせください。