



Red Hat Quay 3

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 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 © 2023 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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] 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[®] 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

第1章 アップグレードの概要	3
第2章 RED HAT QUAY OPERATOR のアップグレードの概要	4
2.1. OPERATOR LIFECYCLE MANAGER	4
2.2. QUAY OPERATOR のアップグレード	4
2.3. QUAYREGISTRY のアップグレード	9
2.4. QUAYECOSYSTEM のアップグレード	10
第3章 スタンドアロンアップグレード	12
3.1. イメージへのアクセス	13
3.2. 3.8.Z から 3.9.Z へのアップグレード	13
3.3. 3.8.Z から 3.9.Z へのアップグレード	15
3.4. 3.7.Z から 3.8.Z へのアップグレード	15
3.5. 3.6.Z から 3.7.Z へのアップグレード	16
3.6. 3.5.Z から 3.7.Z へのアップグレード	16
3.7. 3.4.Z から 3.7.Z へのアップグレード	16
3.8. 3.3.Z から 3.7.Z へのアップグレード	16
3.9. 3.5.Z から 3.6.Z へのアップグレード	16
3.10. 3.4.Z から 3.6.Z へのアップグレード	17
3.11. 3.3.Z から 3.6.Z へのアップグレード	17
3.12. 3.4.Z から 3.5.7 へのアップグレード	18
3.13. 3.3.Z から 3.4.6 へのアップグレード	18
3.14. 3.2.Z から 3.3.4 へのアップグレード	19
3.15. 3.1.Z から 3.2.2 へのアップグレード	19
3.16. 3.0.Z から 3.1.3 へのアップグレード	20
3.17. 2.9.5 から 3.0.5 へのアップグレード	20
3.18. RED HAT QUAY の GEO-REPLICATION デプロイメントのアップグレード	24
第4章 QUAY BRIDGE OPERATOR のアップグレード	27
4.1. RED HAT QUAY OPERATOR 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	27
第5章 RED HAT QUAY のダウングレード	30

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。

Red Hat Quay Operator は、Quay Operator を使用した Quay のアップグレードセクションで説明されているように、[Operator Lifecycle Manager \(OLM\)](#) を使用してアップグレード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

Red Hat Quay Operator は、**シンクロナイズドバージョンスキーム**に従います。つまり、Red Hat Quay 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および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 **Operator Hub** 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. QUAY OPERATOR のアップグレード

OpenShift Container Platform にインストールされている Operator をアップグレードするための標準的なアプローチは、[インストールされている Operator のアップグレード](#) に記載されています。

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。次のアップグレードパスがサポートされています。

1. 3.3.z → 3.6.z
2. 3.4.z → 3.6.z
3. 3.4.z → 3.7.z
4. 3.5.z → 3.7.z
5. 3.7.z → 3.8.z
6. 3.6.z → 3.9.z
7. 3.7.z → 3.9.z
8. 3.8.z → 3.9.z

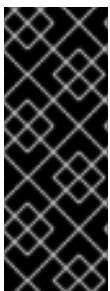
Red Hat Quay のスタンドアロンデプロイメントを 3.9 にアップグレードする場合には、[スタンドアロンアップグレード](#) ガイドを参照してください。

2.2.1. 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、3.4 → 3.5) 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。

3.4.2 → 3.4.3 などの **z** ストリームのアップグレードの場合、更新は、ユーザーが最初にインストール時に選択した major-minor 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、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。これにより、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 Quay の自動更新が行われます。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 を 3.8 → 3.9 に更新する



重要

Red Hat Quay デプロイメントを 1 つの y ストリームから次の y ストリームにアップグレードする場合 (たとえば、3.8.10 → 3.8.11 に)、アップグレードチャンネルを **stable-3.8** から **stable-3.9** に切り替えないでください。y-stream アップグレードの途中でアップグレードチャンネルを変更すると、Red Hat Quay を 3.9 にアップグレードできなくなります。これは既知の問題であり、Red Hat Quay の今後のバージョンで修正される予定です。

Red Hat Quay 3.8 → 3.9 に更新すると、Operator は Clair および Red Hat Quay の既存の PostgreSQL データベースをバージョン 10 からバージョン 13 に自動的にアップグレードします。



重要

- このアップグレードは元に戻すことができません。PostgreSQL 13 にアップグレードすることが強く推奨されます。PostgreSQL 10 は 2022 年 11 月 10 日に最終リリースとなり、サポートされなくなりました。詳細は、[PostgreSQL のバージョン管理ポリシー](#) を参照してください。
- デフォルトでは、Red Hat Quay は PostgreSQL 10 から古い Persistent Volume Claim (PVC) を削除するように設定されています。この設定を無効にして古い PVC をバックアップするには、**quay-operator** サブスクリプションオブジェクトで **POSTGRES_UPGRADE_RETAIN_BACKUP** を **True** に設定する必要があります。

前提条件

- OpenShift Container Platform に Red Hat Quay 3.8 がインストールされている。
- 100 GB の空き容量、追加のストレージ。
アップグレードプロセス中に、移行されたデータを保存するために追加の永続ボリュームクレーム (PVC) がプロビジョニングされる。これにより、ユーザーデータに対する破壊的な操作を防ぐことが可能。アップグレードプロセスでは、Red Hat Quay データベースのアップグレードと Clair データベースのアップグレードの両方で、50 GB の PVC をロールアウト。

手順

1. オプション: **quay-operator** サブスクリプションオブジェクトの **POSTGRES_UPGRADE_RETAIN_BACKUP** を **True** に設定して、PostgreSQL 10 から古い PVC をバックアップします。以下に例を示します。

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: quay-operator
  namespace: quay-enterprise
spec:
  channel: stable-3.8
  name: quay-operator
  source: redhat-operators
  sourceNamespace: openshift-marketplace
config:
  env:
    - name: POSTGRES_UPGRADE_RETAIN_BACKUP
      value: "true"
```

2. OpenShift Container Platform Web コンソールで、**Operators → Installed Operators** に移動します。
3. Red Hat Quay Operator をクリックします。
4. **Subscription** タブに移動します。
5. **Subscription details** で **Update channel** をクリックします。
6. **stable-3.9** を選択して変更を保存します。

7. **Upgrade status** で新規インストールの進行状況を確認します。アップグレードのステータスが **1 installed** に変わるまで待ってから続行してください。
8. OpenShift Container Platform クラスターで、**Workloads** → **Pod** に移動します。既存の Pod は終了するか、または終了中である必要があります。
9. データベースのアップグレードと既存データのアレンジック移行を担当する Pod **clair-postgres-upgrade**、**quay-postgres-upgrade**、および **quay-app-upgrade** が起動するまで待ちます。
10. **clair-postgres-upgrade**、**quay-postgres-upgrade**、および **quay-app-upgrade** Pod が **Completed** としてマークされると、Red Hat Quay デプロイメントの残りの Pod が起動します。これには約 10 分かかります。
11. **quay-database** および **clair-postgres** Pod が **postgresql-13** イメージを使用していることを確認します。
12. **quay-app** Pod が **Running** としてマークされると、Red Hat Quay レジストリーにアクセスできるようになります。

2.2.3. 3.3.z または 3.4.z から 3.6 への直接アップグレード

次のセクションでは、Red Hat Quay 3.3.z または 3.4.z から 3.6 にアップグレードする際の重要な情報を説明しています。

2.2.3.1. エッジルーティングを有効にした状態でのアップグレード

- 以前は、エッジルーティングを有効にして 3.3.z バージョンの Red Hat Quay を実行している場合、ユーザーは 3.4.z バージョンの Red Hat Quay にアップグレードできませんでした。これは、Red Hat Quay 3.6 のリリースで解決されました。
- 3.3.z から 3.6 にアップグレードするときに、Red Hat Quay 3.3.z デプロイメントで **tls.termination** が **none** に設定されている場合は、TLS エッジ終端を使用して HTTPS に変更され、デフォルトのクラスターワイルドカード証明書が使用されます。以下に例を示します。

```
apiVersion: redhatcop.redhat.io/v1alpha1
kind: QuayEcosystem
metadata:
  name: quay33
spec:
  quay:
    imagePullSecretName: redhat-pull-secret
    enableRepoMirroring: true
    image: quay.io/quay/quay:v3.3.4-2
    ...
  externalAccess:
    hostname: quayv33.apps.devcluster.openshift.com
    tls:
      termination: none
  database:
    ...
```

2.2.3.2. サブジェクト別名のないカスタム SSL/TLS 証明書/キーペアを使用したアップグレード

Red Hat Quay 3.3.4 から Red Hat Quay 3.6 に直接アップグレードするときに、サブジェクト代替名 (SAN) なしで独自の SSL/TLS 証明書/キーペアを使用しているお客様には問題があります。Red Hat Quay 3.6 へのアップグレード中に、デプロイメントがブロックされ、Red Hat Quay Operator Pod ログから Red Hat Quay SSL/TLS 証明書に SAN が必要であることを示すエラーメッセージが表示されます。

可能であれば、SAN 内の正しいホスト名を使用して SSL/TLS 証明書を再生成する必要があります。考えられる回避策には、アップグレード後に **quay-app**、**quay-upgrade**、**quay-config-editor** Pod で環境変数を定義して、CommonName のマッチングを有効にすることが含まれます。

```
GODEBUG=x509ignoreCN=0
```

GODEBUG=x509ignoreCN=0 フラグは、SAN が存在しない場合に、X.509 証明書の CommonName フィールドをホスト名として扱うという従来の動作を有効にします。ただし、この回避策は再デプロイメント後も持続しないため、お勧めしません。

2.2.3.3. Red Hat Quay Operator を使用して 3.3.z または 3.4.z から 3.6 にアップグレードする場合の Clair v4 の設定

OpenShift Container Platform 上の新しい Red Hat Quay デプロイメントに Clair v4 をセットアップする場合は、Red Hat Quay Operator を使用することが強く推奨されます。デフォルトでは、Red Hat Quay Operator は、Clair デプロイメントを Red Hat Quay デプロイメントとともにインストールまたはアップグレードし、Clair を自動的に設定します。

接続されていない OpenShift Container Platform クラスターで Clair v4 をセットアップする手順については、[Red Hat Quay OpenShift デプロイメントでの Clair のセットアップ](#) を参照してください。

2.2.4. 3.3.z から 3.6 にアップグレードする際の Swift 設定

Red Hat Quay 3.3.z から 3.6.z にアップグレードすると、ユーザーが **Switch auth v3 requires tenant_id (string) in os_options** エラーを受け取る場合があります。回避策として、**DISTRIBUTED_STORAGE_CONFIG** を手動で更新して、**os_options** パラメーターおよび **tenant_id** パラメーターを追加できます。

```
DISTRIBUTED_STORAGE_CONFIG:
  brscale:
  - SwiftStorage
  - auth_url: http://****/v3
    auth_version: "3"
  os_options:
    tenant_id: ****
    project_name: ocp-base
    user_domain_name: Default
  storage_path: /datastorage/registry
  swift_container: ocp-svc-quay-ha
  swift_password: *****
  swift_user: *****
```

2.2.5. Red Hat Quay Operator の更新チャネルの変更

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 を一覧表示したページでモニターできます。

2.2.6. 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。

The screenshot shows the 'Subscription details' page for the 'quay-operator' installed version 3.4.3. The page is divided into two main sections: 'Subscription details' and 'Metadata'. The 'Subscription details' section includes a table with the following information:

Channel	Approval	Upgrade status
quay-v3.4	Automatic	Up to date

The 'Metadata' section includes the following information:

- Name:** quay-operator
- Namespace:** quay-enterprise
- Labels:** operators.coreos.com/quay-operator.quay-enterprise
- Created at:** Mar 25, 12:17 pm
- Owner:** No owner
- Installed version:** quay-operator.v3.4.3
- Starting version:** quay-operator.v3.4.3
- CatalogSource:** redhat-operators (Healthy)
- InstallPlan:** install-wf26n

Installed Operator の一覧は、現在の Quay インストールの概要を提供します。

The screenshot shows the 'Installed Operators' page for the 'quay-enterprise' project. It displays a table with the following information:

Name	Managed Namespaces	Status	Last updated	Provided APIs
Red Hat Quay 3.4.3 provided by Red Hat	quay-enterprise	Succeeded Up to date	Mar 25, 12:18 pm	Quay Registry

2.3. QUAYREGISTRY 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. すべての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、データベースを管理対象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。

データベース

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。次のアップグレードパスがサポートされています。

1. 3.3.z → 3.6.z
2. 3.4.z → 3.6.z
3. 3.4.z → 3.7.z
4. 3.5.z → 3.7.z
5. 3.7.z → 3.9.z

Red Hat Quay Operator をアップグレードするユーザーは、[Upgrading the Red Hat Quay Operator Overview](#) 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。

- [3.8.z から 3.9.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](#)
- [3.7.z から 3.8.z へのアップグレード](#)
- [3.6.z から 3.7.z へのアップグレード](#)
- [3.5.z から 3.7.z へのアップグレード](#)
- [3.4.z から 3.7.z へのアップグレード](#)
- [3.3.z から 3.7.z へのアップグレード](#)
- [3.5.z から 3.6.z へのアップグレード](#)
- [3.4.z から 3.6.z へのアップグレード](#)

- [3.3.z から 3.6.z へのアップグレード](#)
- [3.4.z から 3.5.z へのアップグレード](#)
- [3.3.4 から 3.4.z へのアップグレード](#)
- [3.2.2 から 3.3.4 へのアップグレード](#)
- [3.1.3 から 3.2.2 へのアップグレード](#)
- [3.0.5 から 3.1.3 へのアップグレード](#)
- [2.9.5 から 3.0.5 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

Quay 3.4.0 以降のイメージは [registry.redhat.io](#) および [registry.access.redhat.com](#) から入手でき、[Red Hat コンテナレジストリーの認証](#) で説明されているように認証が設定されています。

Quay 3.3.4 以前のイメージは [quay.io](#) から入手可能で、認証は、[Accessing Red Hat Quay without a CoreOS login](#) で説明されている通りに設定されています。

3.2. 3.8.Z から 3.9.Z へのアップグレード

スタンドアロン Red Hat Quay デプロイメントを 3.8.z → 3.9 にアップグレードする場合は、PostgreSQL をバージョン 10 → 13 にアップグレードすることを強く推奨します。PostgreSQL を 10 → 13 にアップグレードするには、PostgreSQL 10 データベースを停止し、移行スクリプトを実行してプロセスを開始する必要があります。

スタンドアロン Red Hat Quay デプロイメントで PostgreSQL を 10 から 13 にアップグレードするには、次の手順を使用します。

手順

1. 次のコマンドを入力して、Red Hat Quay データベースをスケールダウンします。

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

2. オプション: Clair を使用している場合は、次のコマンドを入力して Clair コンテナを停止します。

```
$ sudo podman stop <clair_container_id>
```

3. SCLOrg の [データ移行](#) プロシージャから Podman プロセスを実行します。これにより、リモート PostgreSQL サーバーからのデータ移行が可能になります。

```
$ sudo podman run -d --name <migration_postgresql_database> ①
-e POSTGRESQL_MIGRATION_REMOTE_HOST=172.17.0.2 \ ②
-e POSTGRESQL_MIGRATION_ADMIN_PASSWORD=remoteAdminP@ssword \
-v </host/data/directory:/var/lib/pgsql/data:Z> ③
[ OPTIONAL_CONFIGURATION_VARIABLES ]
rhel8/postgresql-13
```

- ① PostgreSQL 13 移行データベースの名前。
- ② 現在の Red Hat Quay PostgreSQL 10 データベースコンテナの IP アドレス。次のコマンドを実行して取得できます: **sudo podman inspection -f "{.NetworkSettings.IPAddress}" postgresql-quay**。
- ③ 最初の PostgreSQL 10 デプロイメントのボリュームマウントポイントとは異なるボリュームマウントポイントを指定し、そのディレクトリーのアクセス制御リストを変更する必要があります。以下に例を示します。

```
$ mkdir -p /host/data/directory
```

```
$ setfacl -m u:26:-wx /host/data/directory
```

これにより、新しいコンテナによってデータが上書きされるのを防ぎます。

4. オプション: Clair を使用している場合は、Clair PostgreSQL データベースコンテナに対して前の手順を繰り返します。
5. PostgreSQL 10 コンテナを停止します。

```
$ sudo podman stop <postgresql_container_name>
```

6. PostgreSQL の移行が完了したら、手順 3 の新しいデータボリュームマウント (例: **</host/data/directory:/var/lib/postgresql/data>**) を使用して PostgreSQL 13 コンテナを実行します。

```
$ sudo podman run -d --rm --name postgresql-quay \
-e POSTGRESQL_USER=<username> \
-e POSTGRESQL_PASSWORD=<password> \
-e POSTGRESQL_DATABASE=<quay_database_name> \
-e POSTGRESQL_ADMIN_PASSWORD=<admin_password> \
-p 5432:5432 \
-v </host/data/directory:/var/lib/pgsql/data:Z> \
registry.redhat.io/rhel8/postgresql-13:1-109
```

7. オプション: Clair を使用している場合は、Clair PostgreSQL データベースコンテナに対して前の手順を繰り返します。
8. Red Hat Quay コンテナを再起動します。

```
$ sudo podman run -d --rm -p 80:8080 -p 443:8443 --name=quay \
-v /home/<quay_user>/quay-poc/config:/conf/stack:Z \
```

```
-v /home/<quay_user>/quay-poc/storage:/datastorage:Z \
{productrepo}/{quayimage}:{productminv}
```

9. オプション: たとえば、Clair コンテナを再起動します。

```
$ sudo podman run -d --name clairv4 \
-p 8081:8081 -p 8088:8088 \
-e CLAIR_CONF=/clair/config.yaml \
-e CLAIR_MODE=combo \
registry.redhat.io/quay/clair-rhel8:v3.9.0
```

詳細については、[データ移行](#) を参照してください。

3.2.1. ターゲットイメージ

- **Quay:** registry.redhat.io/quay/quay-rhel8:v3.9.0
- **Clair:** registry.redhat.io/quay/clair-rhel8:4.6.0
- **PostgreSQL:** registry.redhat.io/rhel8/postgresql-13:1-109
- **Redis:** registry.redhat.io/rhel8/redis-6:1-110)

3.3. 3.8.Z から 3.9.Z へのアップグレード

スタンドアロン Red Hat Quay デプロイメントを 3.8.z → 3.9 にアップグレードする場合は、PostgreSQL をバージョン 10 → 13 にアップグレードすることを強く推奨します。PostgreSQL を 10 → 13 にアップグレードするには、PostgreSQL 10 データベースを停止し、移行スクリプトを実行してプロセスを開始する必要があります。



注記

- Red Hat Quay 3.7 から 3.9 にアップグレードする場合、次のエラーが発生する場合があります。**pg_dumpall: error: query failed: ERROR: xlog flush request 1/B446CCD8 is not satisfied --- flushed only to 1/B0013858**この問題の回避策として、OpenShift Container Platform デプロイメント上の **quayregistry-clair-postgres-upgrade** ジョブを削除すると、問題が解決されます。

3.3.1. ターゲットイメージ

- **Quay:** registry.redhat.io/quay/quay-rhel8:v3.9.0
- **Clair:** registry.redhat.io/quay/clair-rhel8:4.6.0
- **PostgreSQL:** registry.redhat.io/rhel8/postgresql-13:1-109
- **Redis:** registry.redhat.io/rhel8/redis-6:1-110)

3.4. 3.7.Z から 3.8.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:v3.8.0

- **Clair:** registry.redhat.io/quay/clair-rhel8:4.6.0
- **PostgreSQL:** registry.redhat.io/rhel8/postgresql-13:1-109
- **Redis:** registry.redhat.io/rhel8/redis-6:1-110)

3.5. 3.6.Z から 3.7.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:v3.7.0
- **Clair:** registry.redhat.io/quay/clair-rhel8:4.6.0
- **PostgreSQL:** registry.redhat.io/rhel8/postgresql-13:1-109
- **Redis:** registry.redhat.io/rhel8/redis-6:1-110)

3.6. 3.5.Z から 3.7.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:v3.7.0
- **Clair:** registry.redhat.io/quay/clair-rhel8:4.6.0
- **PostgreSQL:** registry.redhat.io/rhel8/postgresql-13:1-109
- **Redis:** registry.redhat.io/rhel8/redis-6:1-110)

3.7. 3.4.Z から 3.7.Z へのアップグレード

3.7.1. ターゲットイメージ

- **Quay:** registry.redhat.io/quay/quay-rhel8:v3.7.0
- **Clair:** registry.redhat.io/quay/clair-rhel8:4.6.0
- **PostgreSQL:** registry.redhat.io/rhel8/postgresql-13:1-109
- **Redis:** registry.redhat.io/rhel8/redis-6:1-110)

3.8. 3.3.Z から 3.7.Z へのアップグレード

Red Hat Quay 3.3 から 3.7 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、最初に 3.3 から 3.6 にアップグレードしてから、3.7 にアップグレードする必要があります。詳細は、[3.3.z から 3.6.z へのアップグレード](#) を参照してください。

3.9. 3.5.Z から 3.6.Z へのアップグレード

3.9.1. ターゲットイメージ

- **Quay:** registry.redhat.io/quay/quay-rhel8:v3.6.0
- **Clair:** registry.redhat.io/quay/clair-rhel8:4.6.0
- **PostgreSQL:** registry.redhat.io/rhel8/postgresql-13:1-109
- **Redis:** registry.redhat.io/rhel8/redis-6:1-110)

3.10. 3.4.Z から 3.6.Z へのアップグレード



注記

Red Hat Quay 3.6 は、3.4.z 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。

3.4.z から Red Hat Quay 3.6 にアップグレードするには、以前のバージョンの Red Hat Quay へのダウングレードをサポートしないデータベースの移行が必要です。この移行を行う前に、データベースをバックアップしてください。

また、ユーザーは、3.4.z からアップグレードするときに、古い Clair v2 を置き換えるために完全に新しい Clairv4 インスタンスを設定する必要があります。Clair v4 の設定手順については、[OpenShift 以外の Red Hat Quay デプロイメントでの Clair のセットアップ](#)を参照してください。

3.10.1. ターゲットイメージ

- **Quay:** registry.redhat.io/quay/quay-rhel8:v3.6.0
- **Clair:** registry.redhat.io/quay/clair-rhel8:v3.6.0
- **PostgreSQL:** registry.redhat.io/rhel8/postgresql-13:1-109
- **Redis:** registry.redhat.io/rhel8/redis-6:1-110)

3.11. 3.3.Z から 3.6.Z へのアップグレード



注記

Red Hat Quay 3.6 は、3.3.z 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。

3.3.z から Red Hat Quay 3.6.z にアップグレードするには、以前のバージョンの Red Hat Quay へのダウングレードをサポートしないデータベースの移行が必要です。この移行を行う前に、データベースをバックアップしてください。

また、ユーザーは、3.3.z からアップグレードするときに、古い Clair v2 を置き換えるために完全に新しい Clairv4 インスタンスを設定する必要があります。Clair v4 の設定手順については、[OpenShift 以外の Red Hat Quay デプロイメントでの Clair のセットアップ](#)を参照してください。

3.11.1. ターゲットイメージ

- **Quay:** registry.redhat.io/quay/quay-rhel8:v3.6.0
- **Clair:** registry.redhat.io/quay/clair-rhel8:v3.6.0
- **PostgreSQL:** registry.redhat.io/rhel8/postgresql-13:1-109
- **Redis:** registry.redhat.io/rhel8/redis-6:1-110)

3.11.2. 3.3.z から 3.6 にアップグレードする際の Swift 設定

Red Hat Quay 3.3.z から 3.6.z にアップグレードすると、ユーザーが **Switch auth v3 requires tenant_id (string) in os_options** エラーを受け取る場合があります。回避策として、**DISTRIBUTED_STORAGE_CONFIG** を手動で更新して、**os_options** パラメーターおよび **tenant_id** パラメーターを追加できます。

```
DISTRIBUTED_STORAGE_CONFIG:
  brscale:
  - SwiftStorage
  - auth_url: http://****/v3
    auth_version: "3"
  os_options:
    tenant_id: ****
    project_name: ocp-base
    user_domain_name: Default
  storage_path: /datastorage/registry
  swift_container: ocp-svc-quay-ha
  swift_password: *****
  swift_user: *****
```

3.12. 3.4.Z から 3.5.7 へのアップグレード

3.12.1. ターゲットイメージ

- **Quay:** registry.redhat.io/quay/quay-rhel8:v3.5.7
- **Clair:** registry.redhat.io/quay/clair-rhel8:v3.9.0
- **PostgreSQL:** registry.redhat.io/rhel8/postgresql-13:1-109
- **Redis:** registry.redhat.io/rhel8/redis-6:1-110)

3.13. 3.3.Z から 3.4.6 へのアップグレード

Quay 3.4 にアップグレードするには、データベースの移行が必要ですが、データベースを移行すると、以前のバージョンの Quay にダウングレードできません。この移行を行う前に、データベースをバックアップしてください。

3.13.1. ターゲットイメージ

- **Quay:** registry.redhat.io/quay/quay-rhel8:v3.4.6
- **Clair:** registry.redhat.io/quay/clair-rhel8:v3.9.0
- **PostgreSQL:** registry.redhat.io/rhel8/postgresql-13:1-109

- **Redis:** registry.redhat.io/rhel8/redis-6:1-110)

3.14. 3.2.Z から 3.3.4 へのアップグレード

3.14.1. ターゲットイメージ

- **Quay:** quay.io/redhat/quay:v3.3.4
- **Clair:** registry.redhat.io/quay/clair-rhel8:v3.9.0
- **PostgreSQL:** rhsc/postgresql-96-rhel7
- **Redis:** registry.access.redhat.com/rhsc/redis-32-rhel7

3.15. 3.1.Z から 3.2.2 へのアップグレード

クラスターで Red Hat Quay 3.1.z が稼働しており、クラスターを 3.2.2 にアップグレードするには、クラスター全体をダウンさせ、設定を少し変更してから 3.2.2 バージョンで、起動し直す必要があります。



警告

この手順で DATABASE_SECRET_KEY の値を設定したら、絶対に変更しないでください。変更すると、既存のロボットアカウントや API トークンなどは使用できなくなります。Quay で使用するためには、新しいロボットアカウントと API トークンを作成する必要があります。

1. Red Hat Quay クラスターの全ホストのサービスを停止します。
2. データベースの秘密鍵として使用するランダムなデータを生成します。以下に例を示します。

```
$ openssl rand -hex 48
2d023adb9c477305348490aa0fd9c
```

3. **config.yaml** ファイルに新しい DATABASE_SECRET_KEY フィールドを追加します。以下に例を示します。

```
DATABASE_SECRET_KEY: "2d023adb9c477305348490aa0fd9c"
```



注記

OpenShift のインストールでは、**config.yaml** ファイルはシークレットとして保存されます。

4. **Quay** コンテナを 1 つ起動し、3.2.2 への移行を完了します。
5. 移行が完了したら、すべてのノードで同じ **config.yaml** が利用可能であることを確認し、それらのノードで新しい quay 3.2.2 サービスを起動します。

6. quay-builder と Clair の 3.0.z バージョンを起動して、クラスターに戻すコンテナのインスタンスを置き換えます。

3.15.1. ターゲットイメージ

- **Quay:** quay.io/redhat/quay:v3.2.2
- **Clair:** registry.redhat.io/quay/clair-rhel8:v3.9.0
- **PostgreSQL:** rhsc/postgresql-96-rhel7
- **Redis:** registry.access.redhat.com/rhsc/redis-32-rhel7

3.16. 3.0.Z から 3.1.3 へのアップグレード

3.16.1. ターゲットイメージ

- **Quay:** quay.io/redhat/quay:v3.1.3
- **Clair:** registry.redhat.io/quay/clair-rhel8:v3.9.0
- **PostgreSQL:** rhsc/postgresql-96-rhel7
- **Redis:** registry.access.redhat.com/rhsc/redis-32-rhel7

3.17. 2.9.5 から 3.0.5 へのアップグレード

2.9.5 から 3.0.5 へのアップグレードでは、Red Hat Quay がダウンしている状態で全アップグレードを行う (同時アップグレード) か、Red Hat Quay を数分間だけダウンさせて、アップグレードの大部分は Red Hat Quay が稼働している状態で行う (バックグラウンドアップグレード) かを選択できます。

処理する必要があるタグの数によっては、バックグラウンドアップグレードの実行に時間がかかる場合があります。ただし、合計ダウンタイムは短くなります。バックグラウンドアップグレードの欠点は、アップグレードが完了するまで最新の機能にアクセスできないことです。クラスターは、アップグレードが完了するまで、Quay v3 コンテナから v2 互換モードで実行されます。

3.17.1. アップグレードの概要

Red Hat Quay 2.y.z クラスターで作業を開始する場合は、以下の手順に従います。最新の Red Hat Quay 3.x バージョンにアップグレードする前に、[ここ](#) で説明されているように、まずそのクラスターを 3.0.5 に移行する必要があります。クラスターで 3.0.5 が実行されたら、各マイナーバージョンに順番にアップグレードすることで、最新の 3.x バージョンにアップグレードできます。以下に例を示します。

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

Red Hat Quay 2.y.z から 3.0 へのアップグレードを開始する前に、次の点に注意してください。

- **同期アップグレード:** 同期アップグレードの場合は、小規模なインストールであれば、ダウンタ

イムの想定時間は合計1時間未満となっています。小規模なインストールの場合は、コンテナイメージのタグが数千以下であると想定してください。このサイズのインストールでは、計画ダウンタイムが2時間程度に抑えられるはずです。Red Hat Quay サービス全体がこの期間、停止しているので、何百万ものタグが含まれるレジストリーで同期アップグレードを試行する場合はダウンタイムが数日間に及ぶ場合があります。

- **バックグラウンドアップグレード:** バックグラウンドアップグレード (互換性モードのアップグレードとも呼ばれる) の場合は、シャットダウンが短時間で行われた後に Red Hat Quay クラスターのアップグレードがバックグラウンドで実行されます。大規模な Red Hat Quay レジストリーの場合、これには数週間の時間がかかる可能性があります。アップグレード中には、クラスターは引き続き v2 モードで動作します。参考までに、ある Red Hat Quay v3 のアップグレードでは、6 台のマシンで約 3000 万個のタグを処理するのに 4 日かかりました。
- **完了時に完全な機能:** Docker バージョン 2、スキーマ 2 の変更に伴う機能 (異なるアーキテクチャーのコンテナのサポートなど) にアクセスするには、移行がすべて完了している必要があります。その他の v3 機能は、切り替え後すぐに利用できます。
- **アップグレードの完了:** アップグレードが完了したら、新機能が利用可能になるように Red Hat Quay **config.yaml** ファイルで **V3_UPGRADE_MODE: complete** を設定する必要があります。すべての新しい Red Hat Quay v3 のインストールには、自動的にこの設定がされています。

3.17.2. 前提条件

最善の結果が得られるように、以下の前提条件を満たすことを推奨します。

- アップグレードを開始する前に Red Hat Quay データベースをバックアップしておきます (定期的なバックアップを実行するのが一般的なベストプラクティスです)。バックアップは、アップグレードを行うために Red Hat Quay クラスターを停止した直後が適切なタイミングです。
- ストレージをバックアップします (こちらも一般的なベストプラクティス)。
- V3 のアップグレードを開始する前に、現在の Red Hat Quay 2.y.z 設定を最新の 2.9.z バージョン (現時点では 2.9.5) にアップグレードします。これを実行するには、以下を行います。
 - Red Hat Quay クラスターがまだ実行中の間に、ノード1つを取り、そのシステムの **Quay** コンテナを最新の 2.9.z バージョンを実行している **Quay** コンテナに変更します。
 - すべてのデータベース移行の実行を待機し、データベースを最新の 2.9.z バージョンにします。これには数分から1時間かかります。
 - 完了したら、すべての既存ノードの **Quay** コンテナを、同じ最新の 2.9.z バージョンに置き換えます。新規バージョンの Red Hat Quay クラスター全体で、v3 アップグレードに進むことができます。

3.17.3. アップグレードタイプの選択

同期アップグレード (ダウンタイムでアップグレードを完了) か、バックグラウンドアップグレード (Red Hat Quay の実行中にアップグレードを完了) のいずれかを選択します。これらのメジャーリリースのアップグレードでは、Red Hat Quay クラスターを少なくとも短期間停止する必要があります。

選択したアップグレードのタイプを問わず、ビルダーおよび clair イメージを使用している場合は、Red Hat Quay クラスターが停止している間に、ビルダーおよび Clair を新規イメージにアップグレードする必要があります。

- **Builder:** quay.io/redhat/quay-builder:v3.0.5

- **Clair:** quay.io/redhat/clair-jwt:v3.0.5

これらのイメージはいずれも registry.redhat.io/quay リポジトリから入手できます。

3.17.4. 同期アップグレードの実行

同期アップグレード (アップグレード中にクラスター全体が停止する) を実行するには、以下を実行します。

1. Quay-builder および Clair コンテナなど Red Hat Quay クラスター全体で停止します。
2. 以下の設定を全ノードの **config.yaml** ファイルに追加します。
V3_UPGRADE_MODE: complete
3. 単一ノードで v3 コンテナをプルおよび起動して、アップグレードが完了するのを待ちます (数分で完了するはずです)。以下のコンテナのバージョンより新しいものを使用します。

- **Quay:** quay.io/redhat/quay:v3.0.5

Quay コンテナは、Red Hat Quay 2 の場合のように 80 および 443 ではなく、Red Hat Quay 3 のポート 8080 および 8443 で起動することに注意してください。したがって、以下の例のように 8080 および 8443 をそれぞれ 80 および 443 に再マッピングすることを推奨します。

```
# docker run --restart=always -p 80:8080 -p 443:8443 \
--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:v3.0.5
```

4. アップグレードが完了したら、他のすべてのノードで Red Hat Quay 3 コンテナを起動します。
5. quay-builder と Clair の 3.0.z バージョンを起動して、クラスターに戻すコンテナのインスタンスを置き換えます。
6. Docker バージョン 2、スキーマ 2 と互換性のあるコンテナのプッシュおよびプルなど、Red Hat Quay が機能していることを確認します。これには、Windows コンテナイメージおよび異なるコンピューターアーキテクチャーのイメージ (arm、ppc など) が含まれます。

3.17.5. バックグラウンドアップグレードの実行

バックグラウンドアップグレードは、2 回ほどクラスターを短時間ダウンさせるだけで実行できます。最初のダウンタイム後にクラスターを再起動すると、データベースをバックフィルするため、quay v3 コンテナは v2 互換性モードで実行します。このバックグラウンドプロセスが完了するまでに時間または数日かかる場合があります。数時間を超えるダウンタイムが問題となるような大規模なインストールを行う場合は、バックグラウンドアップグレードが推奨されます。

このタイプのアップグレードでは、Red Hat Quay を互換性モードにします。互換性モードでは、**Quay** 3 コンテナが実行しますが、アップグレードが完了するまで以前のデータモデルで実行します。手順は以下のとおりです。

1. Red Hat Quay 3 コンテナをすべてのノードにプルします。以下のコンテナのバージョンより新しいものを使用します。
quay.io/redhat/quay:v3.0.5

2. Quay-builder および Clair コンテナなど Red Hat Quay クラスター全体で停止します。
3. 各ノードで **config.yaml** ファイルを編集し、以下のようにアップグレードモードを background に設定します。
V3_UPGRADE_MODE: background

4. Red Hat Quay 3 コンテナを単一ノードで起動し、移行が完了するまで待機します (最大で数分かかります)。以下はコマンドの例です。

Quay コンテナは、Red Hat Quay 2 の場合のように 80 および 443 ではなく、Red Hat Quay 3 のポート 8080 および 8443 で起動することに注意してください。したがって、以下の例のように 8080 および 8443 をそれぞれ 80 および 443 に再マッピングすることを推奨します。

```
# docker run --restart=always -p 80:8080 -p 443:8443 \
--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:v3.0.5
```

5. その他のすべてのノードで Red Hat Quay 3 コンテナを起動します。
6. 次の手順に進むのに十分なレポーティングがされるまで (ステータスが 99% に到達するまで)、**/upgradeprogress** API エンドポイントを監視します。たとえば <https://myquay.example.com/upgradeprogress> を表示するか、他のツールを使用して API をクエリーします。
7. バックグラウンドプロセスが十分に終了したら、別のメンテナンス期間をスケジュールする必要があります。
8. 定期メンテナンス時に、Red Hat Quay クラスター全体を停止します。
9. 各ノードで **config.yaml** ファイルを編集し、以下のように、アップグレードモードを **complete** に設定します。

```
V3_UPGRADE_MODE: complete
```

10. Red Hat Quay を 1 つのノードで再び起動し、最終チェックを実行できるようにします。
11. 最終チェックが完了したら、Red Hat Quay v3 を他のすべてのノードでも起動します。
12. quay-builder と Clair の 3.0.z バージョンを起動して、クラスターに戻すコンテナのインスタンスを置き換えます。
13. Docker バージョン 2、スキーマ 2 と互換性のあるコンテナのプッシュおよびプルなど、Quay が機能していることを確認します。これには、Windows コンテナイメージおよび異なるコンピューターアーキテクチャのイメージ (arm、ppc など) が含まれます。

3.17.6. ターゲットイメージ

- **Quay:** quay.io/redhat/quay:v3.0.5
- **Clair:** quay.io/redhat/clair-jwt:v3.0.5
- **Redis:** registry.access.redhat.com/rhsc/redis-32-rhel7

- **PostgreSQL:** rhsc/postgresql-96-rhel7
- **Builder:** quay.io/redhat/quay-builder:v3.0.5

3.18. RED HAT QUAY の GEO-REPLICATION 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、geo-replication Red Hat Quay デプロイメントをアップグレードします。



重要

- geo-replication Red Hat Quay 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-replication 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、Red Hat Quay 3 などの最新の Red Hat Quay バージョンを取得します。

- a. システム A で以下のコマンドを入力して、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 System A で、Red Hat Quay 3 などの新しいイメージバージョンを実行します。

```
# 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
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 System B で、Red Hat Quay 3 などの新しいイメージバージョンを実行します。

```
# 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、Red Hat Quay 3 などの新しいイメージバージョンを実行します。

```
# 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。

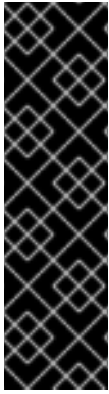
```
$ oc create -f upgrade-quay-integration.yaml
```

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. RED HAT QUAY OPERATOR 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

geo レプリケーションされた Red Hat Quay Operator をアップグレードするには、次の手順を使用します。



重要

- geo レプリケーションされた Red Hat Quay Operator 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) にアップグレードする場合は、アップグレードを行う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。
- アップグレードを行う前に、Red Hat Quay Operator 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。



手順

この手順では、Red Hat Quay Operator を 3 つ (以上) のシステムで実行していることを前提とします。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. System B および System C で、Red Hat Quay Operator デプロイメント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 ❶
    - kind: quay
      managed: true
      overrides: ❷
        replicas: 0
    - kind: clair
      managed: true
      overrides:
        replicas: 0
    - kind: mirror
      managed: true
      overrides:
        replicas: 0
    ...
```

❶ Quay、Clair、ミラーリングワーカーの自動スケーリングの無効化

❷ 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

**注記**

Red Hat Quay Operator 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-config-editor-6dfdcfc44f-hlvwm 1/1   Running   0          73s
quayregistry-quay-redis-84f888776f-hhgms    1/1   Running   0          12m
```

3. システム A で、Red Hat Quay Operator を最新の y-stream バージョンにアップグレードします。これは手動プロセスです。インストールされている Operator のアップグレードの詳細については、[インストールされている Operator のアップグレード](#) を参照してください。Red Hat Quay のアップグレードパスの詳細については、[Red Hat Quay Operator のアップグレード](#) を参照してください。
4. 新規の Red Hat Quay Operator のインストール後に、クラスターに必要なアップグレードは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 がスケジュールされ、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 Operator をアップグレードしないでください。

6. 更新が System A で適切に機能していることを確認した後に、System B および System C で Red Hat Quay Operator 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 が再起動されます。

**注記**

データベーススキーマは新しい y-stream インストールに適しているため、System B および System 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。