



# Red Hat Quay 3.7

## 升级 Red Hat Quay

升级 Red Hat Quay



# Red Hat Quay 3.7 升级 Red Hat Quay

---

升级 Red Hat Quay

## 法律通告

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<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.

## 摘要

升级 Red Hat Quay

---

# 目录

<b>第1章 升级概述</b> .....	<b>3</b>
<b>第2章 升级 QUAY OPERATOR 概述</b> .....	<b>4</b>
2.1. OPERATOR LIFECYCLE MANAGER	4
2.2. 升级 QUAY OPERATOR	4
2.3. 升级 QUAYREGISTRY	7
2.4. 在 QUAY 3.7 中启用功能	7
2.5. 在 QUAY 3.6 中启用功能	8
2.6. 升级 QUAYECO 系统	8
<b>第3章 独立升级</b> .....	<b>11</b>
3.1. 访问镜像	12
3.2. 从 3.6.Z 升级到 3.7.Z	12
3.3. 从 3.5.Z 升级到 3.7.Z	12
3.4. 从 3.4.Z 升级到 3.7.Z	12
3.5. 从 3.3.Z 升级到 3.7.Z	13
3.6. 从 3.5.Z 升级到 3.6.Z	13
3.7. 从 3.4.Z 升级到 3.6.Z	13
3.8. 从 3.3.Z 升级到 3.6.Z	13
3.9. 从 3.4.Z 升级到 3.5.7	14
3.10. 从 3.3.Z 升级到 3.4.6	14
3.11. 从 3.2.Z 升级到 3.3.4	15
3.12. 从 3.1.Z 升级到 3.2.2	15
3.13. 从 3.0.Z 升级到 3.1.3	16
3.14. 从 2.9.5 升级到 3.0.5	16
<b>第4章 升级 QUAY BRIDGE OPERATOR</b> .....	<b>20</b>



## 第 1 章 升级概述

Red Hat Quay 的升级步骤取决于您使用的安装类型。

Red Hat Quay Operator 提供了部署和管理 Red Hat Quay 集群的简单方法。这是在 OpenShift 中部署 Red Hat Quay 的首选步骤。

Red Hat Quay Operator 使用 [Operator Lifecycle Manager\(OLM\)](#) 升级，如“使用 Quay Operator 升级 Quay”一节中所述。

升级概念验证或高可用安装 Red Hat Quay 和 Clair 的步骤包括在 "Standalone upgrade" 部分。



### 重要

Red Hat Quay 只支持回滚到以前的 z-stream 版本，如 3.7.2 → 3.7.1。不支持回滚到以前的 y-stream 版本(3.7.0 → 3.6.0)。这是因为 Red Hat Quay 更新可能包含升级到 Red Hat Quay 的新版本时应用的数据库架构升级。数据库架构升级不被视为向后兼容。

## 第 2 章 升级 QUAY OPERATOR 概述

Quay Operator 遵循一个 *同步的版本方案*，这意味着每个 Operator 版本都绑定到 Quay 版本及其管理的组件。**QuayRegistry** 自定义资源中没有设置要部署的 Quay 版本的字段。Operator 只了解如何部署所有组件的单一版本。选择此方案以确保所有组件都良好工作，并降低 Operator 如何管理 Kubernetes 上许多不同版本的 Quay 的生命周期的复杂性。

### 2.1. OPERATOR LIFECYCLE MANAGER

应使用 [Operator Lifecycle Manager\(OLM\)](#) 来安装和升级 Quay Operator。当使用默认 **approvalStrategy: Automatic** 创建订阅时，每当有新版本可用时，OLM 将自动升级 Quay Operator。



#### 警告

当 Quay Operator 通过 Operator Lifecycle Manager 安装时，可能会将其配置为支持自动或手动升级。这个选项显示在安装过程中 Quay Operator 的 **Operator Hub** 页面中。它还可以通过 **approvalStrategy** 字段在 Quay Operator **Subscription** 对象中找到。选择 **Automatic** 意味着每当发布新 Operator 版本时自动升级您的 Quay Operator。如果这不是理想的选择，则应选择 **手动批准策略**。

### 2.2. 升级 QUAY OPERATOR

在 OpenShift 上升级已安装的 Operator 的标准方法包括在 [升级已安装的 Operator](#) 中。

通常，Red Hat Quay 支持从以前的(N-1)次版本进行升级。例如，不支持直接从 Red Hat Quay 3.0.5 升级到最新版本的 3.5。相反，用户需要如下升级：

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

这需要确保在升级过程中正确且按正确顺序执行任何必要的数据库迁移。

在某些情况下，Red Hat Quay 支持从以前的 (N-2、N-3) 次版本直接进行单步升级。这个例外是只在次版本前的次版本中，对旧版本的用户进行升级流程。支持以下升级路径：

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

对于 Quay 独立部署的用户，想升级到 3.7，请参阅 [单机升级指南](#)。



## 2.2.1. 升级 Quay

要将 Quay 从一个次要版本升级到下一个次要版本，如 3.4 → 3.5，您需要更改 Quay Operator 的更新频道。

对于 **z** 流升级，例如 3.4.2 → 3.4.3，会在用户在安装过程中初始选择的主次频道中发布更新。执行 **z** 流升级的步骤取决于以上所述的批准 **Strategy**。如果批准策略被设置为 **Automatic**，Quay Operator 会自动升级到最新的 **z** 流。这会导致自动的、滚动式的到较新的 **z** 流的 Quay 更新，几乎没有或只有非常短的停机时间。否则，必须手动批准更新，然后才能开始安装。

## 2.2.2. 有关直接从 3.3.z 或 3.4.z 升级到 3.6 的备注

### 2.2.2.1. 启用边缘路由升级

- 在以前的版本中，当运行启用了边缘路由的 3.3.z 版本时，用户无法升级到 Red Hat Quay 的 3.4.z 版本。这个问题已通过 Red Hat Quay 3.6 的发行版本解决。
- 当从 3.3.z 升级到 3.6 时，如果在 Red Hat Quay 3.3.z 部署中将 **tls.termination** 设置为 **none**，它将改为使用 TLS 边缘终止的 HTTPS，并使用默认集群通配符证书。例如：

```
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:
    ...
```

### 2.2.2.2. 使用自定义 TLS 证书/密钥对升级，无需主题备用名称

当直接从 Red Hat Quay 3.3.4 升级到 Red Hat Quay 3.6 时，用户可以使用自己的 TLS 证书/密钥对而无需 Subject Alternative Name(SAN)的用户有问题。在升级到 Red Hat Quay 3.6 时，部署会被阻止，来自 Quay Operator pod 日志的错误消息表示 Quay TLS 证书必须具有 SAN。

如果可能，您应该使用 SAN 中的正确主机名重新生成 TLS 证书。一个可能的解决方法涉及在 **quay-app**、**quay-upgrade** 和 **quay-config-editor** pod 中定义环境变量来启用 **CommonName** 匹配：

```
GODEBUG=x509ignoreCN=0
```

**GODEBUG=x509ignoreCN=0** 标志可在没有 SANs 时将 X.509 证书上的 **CommonName** 字段视为主机名的传统行为。但是，我们不推荐使用这个临时解决方案，因为它不会在重新部署之间保留。

### 2.2.2.3. 使用 Quay Operator 从 3.3.z 或 3.4.z 升级到 3.6 时配置 Clair v4

要在 OpenShift 上的新 Red Hat Quay 部署上设置 Clair v4，强烈建议使用 Quay Operator。默认情况下，Quay Operator 将安装或升级 Clair 部署以及 Red Hat Quay 部署，并自动配置 Clair 安全扫描。

有关在 OpenShift 上设置 Clair v4 的说明，请参阅在 [Red Hat Quay OpenShift 部署 上设置 Clair](#)。

### 2.2.3. 从 3.3.z 升级到 3.6 时 Swift 配置

当从 Red Hat Quay 3.3.z 升级到 3.6.z 时，一些用户可能会收到以下错误：**Switch auth v3 需要 os\_options 中的 tenant\_id (字符串)**。作为临时解决方案，您可以手动更新 **DISTRIBUTED\_STORAGE\_CONFIG** 以添加 **os\_options** 和 **tenant\_id** 参数：

```
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: *****
```

### 2.2.4. 更改 Operator 的更新频道

已安装的 Operator 的订阅指定一个更新频道，用于跟踪和接收 Operator 的更新。要升级 Quay Operator 以开始跟踪并从更新频道接收更新，请更改已安装 Quay Operator 的 **Subscription** 选项卡中的更新频道。对于带有 **自动批准策略** 的订阅，升级会自动开始，并可在列出 Installed Operators 的页面上进行监控。

### 2.2.5. 手动批准待处理的 Operator 升级

如果已安装的 Operator 的订阅被设置为 **Manual**，则当其当前更新频道中发布新更新时，在开始安装前必须手动批准更新。如果 Quay Operator 有待处理的升级，则此状态将显示在 Installed Operators 列表中。在 Quay Operator 的 **Subscription** 选项卡中，您可以预览安装计划，并查看列出可用于升级的资源。如果满意，点 **Approve** 并返回到列出 Installed Operators 的页面，以监控升级的进度。

下图显示了 UI 中的 **Subscription** 选项卡，包括 **更新频道**、**批准策略**、**Upgrade 状态** 和 **InstallPlan**：

The screenshot shows the 'Subscription details' page for the 'quay-operator' operator. The left sidebar contains navigation options like Home, Operators, Workloads, etc. The main content area displays the following information:

- Channel:** quay-v3.4
- Approval:** Automatic
- Upgrade status:** Up to date (1 installed, 0 installing)
- 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-operatorv3.4.3
- Starting version:** quay-operatorv3.4.3
- CatalogSource:** redhat-operators (Healthy)
- InstallPlan:** install-wf26n

Installed Operators 列表提供当前 Quay 安装的高级别概述：

The screenshot shows the 'Installed Operators' list page. It displays a table with the following columns: Name, Managed Namespaces, Status, Last updated, and Provided APIs. The table contains one entry:

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. 升级 QUAYREGISTRY

当 Quay Operator 启动时，它会立即查找它配置为监视的命名空间中的任何 **QuayRegistries**。当找到时，会使用以下逻辑：

- 如果 **status.currentVersion** 没有设置，则正常协调。
- 如果 **status.currentVersion** 等于 Operator 版本，请正常协调。
- 如果 **status.currentVersion** 没有等于 Operator 版本，请检查是否升级它。如果可以，执行升级任务，并在完成后将 **status.currentVersion** 设置为 Operator 的版本。如果无法升级，返回错误并只保留 **QuayRegistry** 及其部署的 Kubernetes 对象。

## 2.4. 在 QUAY 3.7 中启用功能

### 2.4.1. 配额管理配置

现在，配额管理在 **FEATURE\_QUOTA\_MANAGEMENT** 属性下被支持，并默认关闭。要启用配额管理，请将 **config.yaml** 中的功能标记设置为 **true**：

```
FEATURE_QUOTA_MANAGEMENT: true
```

## 2.4.2. 使用 Red Hat Quay 代理远程机构配置

现在，在 **FEATURE\_PROXY\_CACHE** 属性下支持使用 Red Hat Quay 代理远程机构。要启用代理缓存，将 **config.yaml** 中的功能标记设置为 **true**：

```
FEATURE_PROXY_CACHE: true
```

## 2.4.3. Red Hat Quay 构建增强

构建可以在虚拟平台上运行。还提供向后兼容以运行之前的构建配置。要启用虚拟构建，请将 **config.yaml** 中的功能标记设置为 **true**：

```
FEATURE_BUILD_SUPPORT: true
```

## 2.4.4. 使用 Red Hat Quay Operator 进行异地复制

Operator 部署现在支持使用 geo-replication 的 Red Hat Quay 部署。要启用 geo-replication，请将 **config.yaml** 中的功能标记设置为 **true**：

```
FEATURE_STORAGE_REPLICATION: true
```

## 2.5. 在 QUAY 3.6 中启用功能

### 2.5.1. 控制台监控和警报

在 OpenShift 控制台中对监控 Quay 3.6 的支持要求在所有命名空间中安装 Operator。如果您之前在特定命名空间中安装了 Operator，请删除 Operator 本身，并在升级完成后为所有命名空间重新安装它。

### 2.5.2. OCI 和 Helm 支持

现在，Red Hat Quay 3.6 中默认启用对 Helm 和一些 OCI 工件的支持。如果要显式启用该功能，例如，如果您是从没有默认启用的版本升级，则需要重新配置 Quay 部署，以使用以下属性启用 OCI 工件：

```
FEATURE_GENERAL_OCI_SUPPORT: true
```

## 2.6. 升级 QUAYECO 系统

以前的 Operator 版本支持升级，这些 Operator 将 **QuayEcosystem** API 用于一组有限的配置。为确保迁移不会意外发生，需要将一个特殊的标签应用到 **QuayEco** 系统供迁移。为 Operator 创建一个新的 **QuayRegistry**，但旧的 **QuayEcosystem** 将保留到手动删除，以确保您可以回滚并仍能访问 Quay，以出现问题。要将现有的 **QuayEcosystem** 迁移到新的 **QuayRegistry** 中，请按照以下步骤操作：

1. 将 **"quay-operator/migrate": "true"** 添加到 **QuayEcosystem** 的 **metadata.labels**。

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

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

2. 等待 **QuayEcosystem** 使用相同的 **metadata.name** 创建 **QuayRegistry**。 **QuayEcosystem** 将标记为标签 **"quay-operator/migration-complete": "true"**。
3. 设置了新的 **QuayRegistry** 的 **status.registryEndpoint** 后，访问 Quay 并确认所有数据和设置已成功迁移。
4. 当您确定一切正常工作时，您可以删除 **QuayEcosystem** 和 Kubernetes 垃圾回收会清理所有旧资源。

### 2.6.1. 恢复 QuayEco 系统升级

如果在从 **QuayEcosystem** 自动升级到 **QuayRegistry** 时出现问题，请按照以下步骤恢复至使用 **QuayEcosystem**：

1. 使用 UI 或 **kubectl** 来删除 **QuayRegistry**：

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

2. 如果使用 **Route** 提供外部访问，将 **Route** 更改为指回原始的 **Service**（使用 UI 或 **kubectl**）。



#### 注意

如果您的 **QuayEcosystem** 管理 Postgres 数据库，升级过程会将数据迁移到由升级 Operator 管理的新 Postgres 数据库。旧数据库不会更改或删除，但 Quay 在迁移完成后将不再使用它。如果数据迁移过程中出现问题，升级过程将退出，建议您将数据库作为非受管组件继续进行。

### 2.6.2. 升级支持的 QuayEcosystem 配置

如果迁移 **QuayEcosystem** 组件失败或不受支持，Quay Operator 会在日志中报告错误，并且 **status.conditions**。所有非受管组件都应成功迁移，因为 Quay 的 **config.yaml** 中还没有提供任何 Kubernetes 资源，且所有必要的值都已在 Quay 的 **config.yaml** 中提供。

#### 数据库

不支持临时数据库（必须设置 **volumeSize** 字段）。

#### Redis

不需要任何特殊操作。

#### 外部访问

自动迁移只支持 passthrough **Route** 访问。其他方法需要手动迁移。

- 没有自定义主机名的 **LoadBalancer**：当 **QuayEcosystem** 使用标签 **"quay-operator/migration-complete": "true"** 标记后，在删除 **QuayEcosystem** 前需要从现存的 **Service** 中删除 **metadata.ownerReferences** 字段，这可以防止 Kubernetes 对 **Service** 进行垃圾回收并删除负载均衡器。一个新的 **Service** 将被创建，带有 **metadata.name** 格式 **<QuayEcosystem-name>-quay-app**。编辑现有 **Service** 的 **spec.selector**，使其与新 **Service** 的 **spec.selector** 匹配，以便进入旧负载均衡器端点的流量现在会被定向到新的 pod。旧的 **Service** 将会被您管理；Quay Operator 将不再管理它。

- 带有自定义主机名的 **LoadBalancer/NodePort/Ingress** : 将创建一个类型为 **LoadBalancer** 的新的 **Service**, 带有 **metadata.name** 格式 **<QuayEcosystem-name>-quay-app**。将您的 DNS 设置更改为指向由新 **Service** 提供的 **status.loadBalancer** 端点。

### Clair

不需要任何特殊操作。

### Object Storage

**QuayEcosystem** 没有受管对象存储组件, 因此对象存储始终标记为非受管状态。不支持本地存储。

### 仓库镜像

不需要任何特殊操作。

## 第 3 章 独立升级

通常，Red Hat Quay 支持从以前的(N-1)次版本进行升级。例如，不支持直接从 Red Hat Quay 3.0.5 升级到最新版本的 3.5。相反，用户需要如下升级：

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

这需要确保在升级过程中正确且按正确顺序执行任何必要的数据库迁移。

在某些情况下，Red Hat Quay 支持从以前的（N-2、N-3）次版本直接进行单步升级。这个例外是只在次版本前的次版本中，对旧版本的用户进行升级流程。支持以下升级路径：

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

对于希望通过 Quay Operator 升级的用户，[请参阅升级 Quay Operator](#)。

本文档描述了执行每个单独升级所需的步骤。确定您的当前版本，然后按照顺序执行步骤，从您的当前版本开始，并处理所需目标版本。

- [从 3.6.z 升级到 3.7.z](#)
- [从 3.5.z 升级到 3.7.z](#)
- [从 3.4.z 升级到 3.7.z](#)
- [从 3.3.z 升级到 3.7.z](#)
- [从 3.5.z 升级到 3.6.z](#)
- [从 3.4.z 升级到 3.6.z](#)
- [从 3.3.z 升级到 3.6.z](#)
- [从 3.4.z 升级到 3.5.z](#)
- [从 3.3.4 升级到 3.4.z](#)
- [从 3.2.2 升级到 3.3.4](#)
- [从 3.1.3 升级到 3.2.2](#)
- [从 3.0.5 升级到 3.1.3](#)
- [从 2.9.5 升级到 3.0.5](#)

有关各个版本的功能信息，请参阅 [Red Hat Quay 发行注记](#)。

手动升级的一般步骤包括以下步骤：

1. 停止 Quay 和 Clair 容器。
2. 备份数据库和镜像存储（可选但推荐使用）。
3. 使用镜像的新版本启动 Clair。
4. 等待 Clair 准备好接受连接，然后启动新版本的 Quay。

### 3.1. 访问镜像

Quay 3.4.0 及更新版本的镜像可从 [registry.redhat.io](#) 和 [registry.access.redhat.com](#) 获取，并设置了身份验证设置，如 [Red Hat Container Registry Authentication](#) 所述。

Quay 3.3.4 及更早的版本的镜像可从 [quay.io](#) 获取，并设置了身份验证设置，如在 [没有 CoreOS 登录的情况下访问 Red Hat Quay](#) 所述。

## 3.2. 从 3.6.Z 升级到 3.7.Z

### 3.2.1. 目标镜像

- `quay` : [registry.redhat.io/quay/quay-rhel8:v3.7.13](#)
- **Clair**: [registry.redhat.io/quay/clair-rhel8:v3.7.13](#)
- **postgresql**: [registry.redhat.io/rhel8/postgresql-10](#)
- `redis` : [registry.redhat.io/rhel8/redis-6](#)

## 3.3. 从 3.5.Z 升级到 3.7.Z

### 3.3.1. 目标镜像

- `quay` : [registry.redhat.io/quay/quay-rhel8:v3.7.13](#)
- **Clair**: [registry.redhat.io/quay/clair-rhel8:v3.7.13](#)
- **postgresql**: [registry.redhat.io/rhel8/postgresql-10](#)
- `redis` : [registry.redhat.io/rhel8/redis-6](#)

## 3.4. 从 3.4.Z 升级到 3.7.Z

### 3.4.1. 目标镜像

- `quay` : [registry.redhat.io/quay/quay-rhel8:v3.7.13](#)
- **Clair**: [registry.redhat.io/quay/clair-rhel8:v3.7.13](#)
- **postgresql**: [registry.redhat.io/rhel8/postgresql-10](#)



- `redis`: registry.redhat.io/rhel8/redis-6

## 3.5. 从 3.3.Z 升级到 3.7.Z

不支持从 3.3 升级到 Red Hat Quay 3.7。用户必须首先从 3.3 升级到 3.6，然后升级到 3.7。如需更多信息，请参阅[从 3.3.z 升级到 3.6.z](#)。

## 3.6. 从 3.5.Z 升级到 3.6.Z

### 3.6.1. 目标镜像

- `quay`: registry.redhat.io/quay/quay-rhel8:v3.6.0
- **Clair**: registry.redhat.io/quay/clair-rhel8:v.3.60
- **postgresql**: registry.redhat.io/rhel8/postgresql-10
- `redis`: registry.redhat.io/rhel8/redis-6

## 3.7. 从 3.4.Z 升级到 3.6.Z

+



### 注意

Red Hat Quay 3.6 支持从 3.4.z 直接进行单步升级。这个例外是只在次版本前的次版本中，对旧版本的用户进行升级流程。

从 3.4.z 升级到 Red Hat Quay 3.6 需要有一个数据库迁移，它不支持降级到以前的 Red Hat Quay 版本。在执行此迁移前，请备份您的数据库。

在从 3.4.z 升级时，用户还需要配置全新的 Clair v4 实例，以替换旧的 Clair v2。有关配置 Clair v4 的说明，请参阅[在非 OpenShift Red Hat Quay 部署上设置 Clair](#)。

### 3.7.1. 目标镜像

- `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-10
- `redis`: registry.redhat.io/rhel8/redis-6

## 3.8. 从 3.3.Z 升级到 3.6.Z

+



### 注意

Red Hat Quay 3.6 支持从 3.3.z 进行直接的、单步升级。这个例外是只在次版本前的次版本中，对旧版本的用户进行升级流程。

从 3.3.z 升级到 Red Hat Quay 3.6.z 需要有一个数据库迁移，它不支持降级到以前的 Red Hat Quay 版本。在执行此迁移前，请备份您的数据库。

从 3.3.z 升级时，用户还需要配置全新的 Clair v4 实例来取代旧的 Clair v2。有关配置 Clair v4 的说明，请参阅 [在非 OpenShift Red Hat Quay 部署上设置 Clair](#)。

### 3.8.1. 目标镜像

- 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-10
- redis : registry.redhat.io/rhel8/redis-6

### 3.8.2. 从 3.3.z 升级到 3.6 时 Swift 配置

当从 Red Hat Quay 3.3.z 升级到 3.6.z 时，一些用户可能会收到以下错误：**Switch auth v3 需要 os\_options 中的 tenant\_id (字符串)**。作为临时解决方案，您可以手动更新 **DISTRIBUTED\_STORAGE\_CONFIG** 以添加 **os\_options** 和 **tenant\_id** 参数：

```
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: *****
```

## 3.9. 从 3.4.Z 升级到 3.5.7

### 3.9.1. 目标镜像

- quay : registry.redhat.io/quay/quay-rhel8:v3.5.7
- Clair: registry.redhat.io/quay/clair-rhel8:v3.5.7
- postgresql : registry.redhat.io/rhel8/postgresql-10:1
- redis: registry.redhat.io/rhel8/redis-5

## 3.10. 从 3.3.Z 升级到 3.4.6

升级到 Quay 3.4 需要数据库迁移，它不支持降级到之前的 Quay 版本。在执行此迁移前，请备份您的数据库。

### 3.10.1. 目标镜像

- `quay` : registry.redhat.io/quay/quay-rhel8:v3.4.6
- `Clair`: registry.redhat.io/quay/clair-rhel8:v3.4.6
- `postgresql`: registry.redhat.io/rhel8/postgresql-10
- `redis`: registry.redhat.io/rhel8/redis-5

## 3.11. 从 3.2.Z 升级到 3.3.4

### 3.11.1. 目标镜像

- `quay` : quay.io/redhat/quay:v3.3.4
- `clair` : registry.redhat.io/quay/clair-rhel8:v3.4
- `postgresql`: rhscl/postgresql-96-rhel7
- `Redis` : registry.access.redhat.com/rhscl/redis-32-rhel7

## 3.12. 从 3.1.Z 升级到 3.2.2

在集群运行任何 Red Hat Quay 3.1.z 版本后，要把集群升级到 3.2.2，您必须关闭整个集群，并在使用 3.2.2 版本重新启用前对配置进行小的更改。



#### 警告

在此流程中设置 `DATABASE_SECRET_KEY` 的值后，请勿更改它。如果您这样做，则现有机器人帐户、API 令牌等无法再使用。您必须创建新的机器人帐户和 API 令牌，才能与 Quay 搭配使用。

1. 从服务中使用 Red Hat Quay 集群中的所有主机。
2. 生成一些随机数据，以用作数据库 secret 密钥。例如：

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

3. 向 `config.yaml` 文件添加新的 `DATABASE_SECRET_KEY` 字段。例如：

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



#### 注意

对于 OpenShift 安装，`config.yaml` 文件存储为 secret。

4. 启动一个 **Quay** 容器以完成从 3.2.2 的迁移。
5. 迁移完成后，确保所有节点上都具有相同的 **config.yaml**，并在这些节点上启动新的 quay 3.2.2 服务。
6. 启动 quay-builder 和 Clair 的 3.0.z 版本，以替换您要返回到集群的那些容器的任何实例。

### 3.12.1. 目标镜像

- quay : quay.io/redhat/quay:v3.2.2
- Clair: registry.redhat.io/quay/clair-rhel8:v3.7.13
- postgresql: rhscl/postgresql-96-rhel7
- Redis : registry.access.redhat.com/rhscl/redis-32-rhel7

## 3.13. 从 3.0.Z 升级到 3.1.3

### 3.13.1. 目标镜像

- quay : quay.io/redhat/quay:v3.1.3
- Clair: registry.redhat.io/quay/clair-rhel8:v3.7.13
- postgresql: rhscl/postgresql-96-rhel7
- Redis : registry.access.redhat.com/rhscl/redis-32-rhel7

## 3.14. 从 2.9.5 升级到 3.0.5

对于 2.9.5 到 3.0.5 升级，您可以使用 Red Hat Quay down（同步升级）进行整个升级，或者只让 Red Hat Quay 运行进行大量升级（background 升级）。

根据需要处理多少个标签，后台升级可能需要更长的时间。但是，总停机时间较少。后台升级的缺点是，在升级完成后您将无法访问最新的功能。集群以 v2 兼容性模式运行 Quay v3 容器，直到升级完成为止。

### 3.14.1. 升级概述

如果您是从 Red Hat Quay 2.y.z 集群开始，请按照以下步骤操作。升级到最新的 Red Hat Quay 3.x 版本前，您必须首先将该集群迁移到 3.0.5，如[这里](#)所述。集群正在运行 3.0.5 后，您可以通过按顺序升级到每个次要版本来升级到最新的 3.x 版本。例如：

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

在开始 Red Hat Quay 2.y.z 升级到 3.0 之前，请注意以下几点：

- **同步升级**：进行同步升级，小型安装的总停机时间少于一小时。考虑小安装包含几千容器镜像标签或更少。对于该大小的安装，您可能只需几小时的计划停机时间即可获得。整个 Red Hat Quay

服务在持续时间内停机，因此如果您要在带有数百万标签的 registry 上进行同步升级，您可能在数天内停机。

- **后台升级**：进行后台升级（也称为兼容性模式升级），在短暂关闭 Red Hat Quay 集群升级在后台之后。对于大型 Red Hat Quay registry，这可能要花费数周才能完成，但集群在升级期间在 v2 模式中继续操作。作为一个参考点，一个 Red Hat Quay v3 升级需要 4 天时间来处理跨六个机器的约 3,000 万个标签。
- **完成完全的功能**：在可以访问与 Docker 版本 2 相关的功能前，模式 2 更改（如对不同架构的容器支持），整个迁移必须完成。在切换时，其他 v3 功能会立即可用。
- **升级已完成**：升级完成后，您需要在 Red Hat Quay `config.yaml` 文件中设置 `V3_UPGRADE_MODE: complete` 以供新功能使用。所有新的 Red Hat Quay v3 安装都会自动设置。

### 3.14.2. 先决条件

为确保最佳结果，我们建议以下先决条件：

- 在开始升级前备份您的 Red Hat Quay 数据库（进行常规备份是一个一般的最佳做法）。在关闭 Red Hat Quay 集群进行升级后，最好这样做。
- 备份您的存储（通用最佳实践）。
- 在开始 v3 升级前，将当前的 Red Hat Quay 2.y.z setup 升级到最新的 2.9.z 版本（当前为 2.9.5）。为此，请执行以下操作：
  - 当 Red Hat Quay 集群仍在运行时，需要一个一个节点，并将该系统上的 **Quay** 容器改为运行最新的 2.9.z 版本的 **Quay** 容器。
  - 等待所有数据库迁移运行，使数据库进入最新的 2.9.z 版本。这只需要几分钟时间才会变为半小时。
  - 完成后，将所有现有节点上的 **Quay** 容器替换为相同的最新 2.9.z 版本。通过新版本上的整个 Red Hat Quay 集群，您可以继续进行 v3 升级。

### 3.14.3. 选择升级类型

选择同步升级（在停机后完成升级）和背景升级（在 Red Hat Quay 仍在运行时完成升级）。这两个主要版本升级都需要在短时间内关闭 Red Hat Quay 集群。

无论您选择哪个升级类型，在 Red Hat Quay 集群停机期间，如果您使用 builder 和 Clair 镜像，您还需要升级到这些新镜像：

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

这些镜像都在 `registry.redhat.io/quay` 存储库中获得。

### 3.14.4. 运行同步升级

要运行同步升级，在整个集群期间为整个升级关闭，请执行以下操作：

1. 关闭整个 Red Hat Quay 集群，包括任何 `quay-builder` 和 `Clair` 容器。

- 将以下设置添加到所有节点上的 **config.yaml** 文件中：
 

```
V3_UPGRADE_MODE: complete
```
- 在单个节点上拉取并启动 v3 容器，并等待升级需要很长时间（需要几分钟）。使用以下容器或更高版本：
  - **Quay**: quay.io/redhat/quay:v3.0.5  
 请注意，**Quay** 容器位于 Red Hat Quay 3 的端口 8080 和 8443，而不是 80 和 443，因为它们用于 Red Hat Quay 2。因此，我们建议将 8080 和 8443 重新映射到 80 和 443，如下例所示：

```
# 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
```

- 升级完成后，在所有其他节点上启动 Red Hat Quay 3 容器。
- 启动 quay-builder 和 Clair 的 3.0.z 版本，以替换您要返回到集群的那些容器的任何实例。
- 验证 Red Hat Quay 正常工作，包括推送和拉取与 Docker 版本 2，schema 2 兼容的容器。这包括不同计算机架构的 Windows 容器镜像和镜像（arm、ppc 等）。

### 3.14.5. 运行后台升级

要进行后台升级，您需要在两个 occasions 短时间内关闭集群。当第一次停机后，quay v3 容器会在 v2 兼容性模式下运行时，它会回填数据库。此后台进程可能需要几小时甚至数天才能完成。对于大型安装，建议进行后台升级，因为停机会超过几个小时。

对于这种类型的升级，您可以将 Red Hat Quay 置于兼容性模式，其中运行 **Quay** 3 容器，但在升级完成时在旧数据模型上运行。以下是您做什么：

- 将 Red Hat Quay 3 容器拉取到所有节点。使用以下容器或更高版本：
 

```
quay.io/redhat/quay:v3.0.5
```
- 关闭整个 Red Hat Quay 集群，包括任何 quay-builder 和 Clair 容器。
- 编辑每个节点上的 **config.yaml** 文件，并将升级模式设置为后台，如下所示：
 

```
V3_UPGRADE_MODE: background
```
- 在单一节点上使 Red Hat Quay 3 容器启动，并等待迁移完成（最多几分钟）。下面是一个该命令的示例：
 

请注意，**Quay** 容器位于 Red Hat Quay 3 的端口 8080 和 8443，而不是 80 和 443，因为它们用于 Red Hat Quay 2。因此，我们建议将 8080 和 8443 重新映射到 80 和 443，如下例所示：

```
# 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
```

- 在所有其他节点上启动 Red Hat Quay 3 容器。

6. 监控 `/upgradeprogress` API 端点，直到报告足够完成以移动到下一步（状态会到达 99%）。例如，查看 <https://myquay.example.com/upgradeprogress> 或使用一些其他工具查询 API。
7. 当后台进程足够时，您必须调度另一个维护窗口。
8. 在计划的维护期间，关闭整个 Red Hat Quay 集群。
9. 编辑每个节点上的 `config.yaml` 文件，并将升级模式设置为 **完整**，如下所示：

```
V3_UPGRADE_MODE: complete
```

10. 启动 Red Hat Quay 在一个节点上进行最终检查。
11. 完成最后的检查后，将 Red Hat Quay v3 放回所有其他节点上。
12. 启动 `quay-builder` 和 `Clair` 的 3.0.z 版本，以替换您要返回到集群的那些容器的任何实例。
13. 验证 Quay 是否正常工作，包括推送和拉取与 Docker 版本 2 兼容的容器，即架构 2。这包括不同计算机架构的 Windows 容器镜像和镜像（arm、ppc 等）。

### 3.14.6. 目标镜像

- **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`

## 第 4 章 升级 QUAY BRIDGE OPERATOR

要升级 Quay Bridge Operator(QBO)，将 Subscription 选项卡中的 Channel Subscription 更新频道改为所需的频道。

当将 QBO 从 3.5 升级到 3.7 时，需要执行几个额外的步骤：

1. 您需要创建一个新的 **QuayIntegration** 自定义资源。这可以在 Web 控制台中或命令行完成。

### 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** 确保 **clusterID** 与现有 **QuayIntegration** 资源的值匹配。

2. 创建新的 **QuayIntegration** 自定义资源：

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

3. 删除旧的 **QuayIntegration** 自定义资源。

4. 删除旧的 **变异webhook 配置**：

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