



# Red Hat 3scale API Management 2.14

## 迁移 Red Hat 3scale API Management

迁移或升级 3scale API 管理及其组件



# Red Hat 3scale API Management 2.14 迁移 Red Hat 3scale API Management

---

迁移或升级 3scale API 管理及其组件

## 法律通告

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.

## 摘要

通过 3scale operator 将 Red Hat 3scale API Management 升级到最新版本，并在基于 Operator 的部署中找到要升级 APIcast 的信息。

---

## 目录

前言 .....	3
对红帽文档提供反馈 .....	4
第 1 章 3SCALE API 管理基于 OPERATOR 的升级指南：从 2.13 升级到 2.14 .....	5
1.1. 执行升级的先决条件 .....	5
1.2. 在基于 OPERATOR 的安装中，从 2.13 升级到 2.14 .....	5
1.3. 在使用外部 ORACLE 数据库的基于 OPERATOR 的安装中，从 2.13 升级到 2.14 .....	6
第 2 章 基于 APICAST OPERATOR 的升级指南：从 2.13 升级到 2.14 .....	7
2.1. 执行升级的先决条件 .....	7
2.2. 在基于 OPERATOR 的安装中，将 APICAST 从 2.13 升级到 2.14 .....	7
第 3 章 从嵌入式 POSTGRESQL 10 数据库迁移到外部 POSTGRESQL 10 数据库 .....	9



---

## 前言

本指南提供了通过 3scale 操作器将 Red Hat 3scale API Management 升级到最新版本的信息。您将找到将 3scale 安装从 2.13 升级到 2.14 所需的详细信息，以及在基于 Operator 的部署中升级 APIcast 的步骤。

要将 3scale On-premise 部署从 2.13 升级到 2.14，请参阅以下指南：

- [3scale 基于 operator 的升级指南](#)

要在基于 Operator 的部署中升级 APIcast，请参阅以下指南：

- [APIcast 升级指南](#)

## 对红帽文档提供反馈

我们感谢您对我们文档的反馈。

要改进，创建一个 JIRA 问题并描述您推荐的更改。提供尽可能多的详细信息，以便我们快速解决您的请求。

### 前提条件

- 您有红帽客户门户网站帐户。此帐户可让您登录到 Red Hat Jira Software 实例。如果您没有帐户，系统会提示您创建一个帐户。

### 流程

1. 单击以下链接：[创建问题](#)。
2. 在 **Summary** 文本框中输入问题的简短描述。
3. 在 **Description** 文本框中提供以下信息：
  - 找到此问题的页面的 URL。
  - 有关此问题的详细描述。  
您可以将信息保留在任何其他字段中的默认值。
4. 点 **Create** 将 JIRA 问题提交到文档团队。

感谢您花时间来提供反馈。

# 第 1 章 3SCALE API 管理基于 OPERATOR 的升级指南：从 2.13 升级到 2.14



## 警告

在某些情况下，将 3scale API Management 从 2.13 升级到 2.14 可能无法完成。要防止这种情况，请在尝试升级前按照红帽客户门户网站中的经过测试和批准的步骤操作。<https://access.redhat.com/solutions/7049647> 该解决方案包含有关造成问题的根本原因和条件的详细信息。

在基于 operator 的安装中，将 Red Hat 3scale API Management 从 2.13 升级到 2.14，以在 OpenShift 4.x 上管理 3scale。

要自动获得 3scale 的微版本，请确保自动更新已经启动。如果您使用 Oracle 外部数据库，请不要设置自动更新。要查看这一点，请参阅[配置微版本的自动应用程序](#)。



## 重要

为了了解所需的条件和程序，请在应用列出的步骤前阅读整个升级指南。升级过程会破坏服务的调配，直到过程完成为止。因为这个过程需要涉及到系统中断，请确保计划有一个维护窗口进行。

## 1.1. 执行升级的先决条件

本节介绍了在基于 Operator 的安装中，将 3scale 从 2.13 升级到 2.14 所需的配置。

- 具有管理员访问权限的 OpenShift Container Platform (OCP) 4.12、4.13 或 4.14 集群。确保您的 OCP 环境至少升级到 4.12 版本，这是进行 3scale 更新的最低要求。
- 3scale 2.13 以前通过 3scale Operator 部署。
- 确保正在使用 **threescale-2.13** 频道的最新 CSV。检查它：
  - 如果订阅的批准设置是 *自动的*，您应该已处于该频道的最新 CSV 版本。
  - 如果订阅的批准设置是 *manual*，请确保批准所有待处理的 *InstallPlans*，并具有最新的 CSV 版本。
  - 请记住，如果有一个待处理的安装计划，可能还有更多待处理的安装计划，只有在安装了现有待定计划后才会显示这些计划。

## 1.2. 在基于 OPERATOR 的安装中，从 2.13 升级到 2.14

在基于 operator 的部署中，将 3scale 从 2.13 升级到 2.14：

1. 使用具有管理员特权的帐户登录 OCP 控制台。
2. 选择部署了 *3scale-operator* 的项目。

3. 点 **Operators > Installed Operators**.
4. 选择 **Red Hat Integration - 3scale > Subscription > Channel**
5. 选择 *threescale-2.14* 并保存更改，以编辑订阅的频道。  
这将开始升级过程。
6. 查询项目中 pod 的状态，直到您看到所有新版本都在运行并就绪且没有错误：

```
$ oc get pods -n <3scale_namespace>
```



### 注意

- pod 在升级过程中可能会出现临时错误。
- 升级 pod 所需的时间可能从 5 到 10 分钟。

7. 新 pod 版本运行后，通过登录 3scale 管理门户并检查它是否按预期工作，确认升级是否成功。
8. 运行以下命令，检查 *APIManager* 对象的状态并获取 YAML 内容。<myapimanager> 代表 *APIManager* 的名称：

```
$ oc get apimanager <myapimanager> -n <3scale_namespace> -o yaml
```

- 带有值的新注解应如下所示：

```
apps.3scale.net/apimanager-threescale-version: "2.14"
apps.3scale.net/threescale-operator-version: "0.11.x"
```

执行所有步骤后，基于 Operator 的部署中的 3scale 从 2.13 升级到 2.14 已完成。

## 1.3. 在使用外部 ORACLE 数据库的基于 OPERATOR 的安装中，从 2.13 升级到 2.14

按照以下步骤，使用外部 Oracle 数据库更新基于 Operator 的 3scale 安装。

### 流程

1. 按照以下步骤 [安装 Red Hat 3scale API 管理指南](#)，以创建新的 **system-oracle-3scale-2.14.0-1** 镜像。
2. 按照 [在基于 Operator 的安装中从 2.13 升级到 2.14 中的步骤升级](#) 3scale Operator。
3. 升级完成后，使用此流程的第一步中创建的新镜像更新 APIManager 自定义资源，如使用 [operator 安装 3scale API 管理](#) 中所述。

## 第 2 章 基于 APICAST OPERATOR 的升级指南：从 2.13 升级到 2.14

在基于 Operator 的安装中，将 APICast 从 2.13 升级到 2.14，可帮助您使用 APICast API 网关将您的内部和外部应用程序编程接口(API)服务与 3scale 集成。



### 重要

为了了解所需的条件和程序，请在应用列出的步骤前阅读整个升级指南。升级过程会破坏服务的调配，直到过程完成为止。因为这个过程需要涉及到系统中断，请确保计划有一个维护窗口进行。

### 2.1. 执行升级的先决条件

要在基于 operator 的安装过程中执行 APICast 从 2.13 升级到 2.14，需要满足以下先决条件：

- 具有管理员访问权限的 OpenShift Container Platform (OCP) 4.12、4.13 或 4.14 集群。确保您的 OCP 环境至少升级到 4.12 版本，这是继续 APICast 更新的最低要求。
- APICast 2.13 之前通过 APICast operator 部署。
- 确保正在使用 **threescale-2.13** 频道的最新 CSV。检查它：
  - 如果订阅的批准设置是 *自动的*，您应该已处于该频道的最新 CSV 版本。
  - 如果订阅的批准设置是 *manual*，请确保批准所有待处理的 *InstallPlans*，并具有最新的 CSV 版本。
  - 请记住，如果有一个待处理的安装计划，可能还有更多待处理的安装计划，只有在安装了现有待定计划后才会显示这些计划。

### 2.2. 在基于 OPERATOR 的安装中，将 APICAST 从 2.13 升级到 2.14

在基于 Operator 的安装中，将 APICast 从 2.13 升级到 2.14，以便 APICast 可以在 3scale 安装中作为 API 网关运行。

#### 流程

1. 使用具有管理员特权的帐户登录 OCP 控制台。
2. 选择部署了 *APICast Operator* 的项目。
3. 点 **Operators > Installed Operators**.
4. 在 **Subscription > Channel** 中，选择 *Red Hat Integration - 3scale APICast gateway*。
5. 选择 *threescale-2.14* 频道并保存更改，以编辑订阅的频道。这将开始升级过程。
6. 查询项目中 pod 的状态，直到您看到所有新版本都在运行并就绪且没有错误：

```
$ oc get pods -n <apicast_namespace>
```

**注意**

- pod 在升级过程中可能会出现临时错误。
- 升级 pod 所需的时间可能从 5 到 10 分钟。

7. 运行以下命令，检查 *APIcast* 对象的状态，并获取 *YAML* 内容：

```
$ oc get apicast <myapicast> -n <apicast_namespace> -o yaml
```

执行所有步骤后，基于 Operator 的部署中的 *APIcast* 从 2.13 升级到 2.14 已完成。

## 第 3 章 从嵌入式 PostgreSQL 10 数据库迁移到外部 PostgreSQL 10 数据库

### 重要

- 在将 **system.appSpec.replicas** 扩展到 1 之前，数据库应升级到受支持的版本，当前为 PostgreSQL 13。请参阅 [Red Hat 3scale API Management 支持的配置](#)
- 本文档关于从嵌入式 PostgreSQL 10 数据库迁移到外部 PostgreSQL 10 数据库。要从外部 PostgreSQL 10 数据库升级到外部 PostgreSQL 13 数据库，您必须遵循官方 [PostgreSQL 文档](#)。
- *免责声明：此处包含的外部网络链接仅为方便用户而提供。红帽没有审阅链接的内容，并不对其内容负责。包含任何指向外部网站的链接并不表示红帽认可该网站或其实体、产品或服务。您同意红帽对因您使用（或依赖）外部网站或内容而导致的任何损失或费用不承担任何责任。*

在同一 DB 版本中应该发生从嵌入式 PostgreSQL 数据库迁移到和外部 PostgreSQL 数据库的过程。在此迁移指南中，它应该是 PostgreSQL 10。您应该在生产环境中使用外部数据库。

如果您使用 PostgreSQL 作为 **system-database**，请使用 3scale [安装的外部数据库](#) 支持版本。

### 重要

- 这些步骤是常规准则。具体步骤可能会因您的操作系统、PostgreSQL 版本和数据库的特定要求而异。
- 在升级前，仔细阅读 [PostgreSQL 文档](#) 和发行注记。
- 在将此步骤应用到生产部署之前，在非生产环境中测试此步骤。
- 此过程会破坏服务的调配，直到过程完成为止。因为这个过程需要涉及到系统中断，请确保计划有一个维护窗口进行。

### 流程

1. 使用 APIManager 客户资源(CR)缩减 **system-app** DeploymentConfig (DC) :

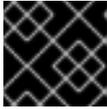
```
apiVersion: apps.3scale.net/v1alpha1
kind: APIManager
metadata:
  name: <apimanager_sample>
spec:
  system:
    appSpec:
      replicas: 0
  wildcardDomain: <example.com>
```

2. 验证 pod 是否已缩减 :

```
$ oc get deploymentconfig system-app -o jsonpath='{.status.availableReplicas}'
0
```

- a. 等待所有 3scale pod 的状态为 **Terminated**, 然后继续 PostgreSQL 迁移。
3. 备份现有的 PostgreSQL 数据库, 包括所有数据、配置和用户帐户 :

```
$ DB_USER=$(oc get secret system-database -o jsonpath="{.data.DB_USER}" | base64 --
decode)
$ DATABASE_NAME=$(oc get secret system-database -o jsonpath="{.data.URL}" | base64 -
-decode | cut -d '/' -f4)
```



### 重要

不要管道到 **stdout**。二进制文件已损坏。

4. 带有自定义格式的转储 :

```
$ oc rsh $(oc get pods -l 'deploymentConfig=system-postgresql' -o json | jq -r
'.items[0].metadata.name') bash -c "pg_dump -U $DB_USER -F c $DATABASE_NAME -f
/tmp/<backupfilename>.backup"
```

5. 下载备份 :

```
$ oc cp $(oc get pods -l 'deploymentConfig=system-postgresql' -o json | jq -r
'.items[0].metadata.name'):/tmp/<backupfilename>.backup <backupfilename>.backup
```

6. 在目标外部系统中安装部署在 3scale 上的同一 PostgreSQL 10 版本。按照安装说明, 从 PostgreSQL 网站下载安装软件包。
7. 复制并恢复您现有的 PostgreSQL 数据库的备份, 包括所有数据、配置和用户帐户到目标外部系统。
8. 在 PostgreSQL 中创建新数据库 :

```
$ createdb -U <username> <databasename>
```

9. 将备份文件中的数据导入到新的 PostgreSQL 数据库。

- 使用自定义格式恢复 :

```
$ pg_restore [--host <databasehostname>] -U <username> -d <databasename> --
verbose -F c <backupfilename>.backup
```

10. 通过连接到数据库并运行查询来验证数据是否已成功导入到新的 PostgreSQL 数据库 :

```
postgresql://<username>:<password>@<databasehostname>/<databasename>
```

11. 更新 **system-database** secret :

```
$ oc apply -f - <<EOF
---
apiVersion: v1
kind: Secret
metadata:
  name: system-database
```

```
stringData:  
  DB_PASSWORD: <password>  
  DB_USER: <username>  
  URL: "postgresql://<username>:<password>@<databasehostname>:  
<databaseport>/<databasename>"  
type: Opaque  
EOF
```

12. 更新 APImanager CR，以启用外部数据库并扩展系统：

```
$ oc patch apimanager <apimanager_sample> --type=merge --patch '{"spec": {"system":  
{"database": null, "appSpec": {"replicas": 1}}, "externalComponents": {"system": {"database":  
true}}}]'
```

13. 删除本地 postgresql 部署：

```
$ oc delete service system-postgresql  
$ oc delete deploymentconfig system-postgresql  
$ oc delete pvc postgresql-data
```

14. 验证 pod 是否已扩展：

```
$ oc wait --for=condition=available apimanager/<apimanager_sample> --timeout=-1s
```

## 其他资源

- [PostgreSQL 文档](#)
- [PostgreSQL Downloads](#)