Red Hat 3scale API Management 2.14

Migrating Red Hat 3scale API Management

Migrate or upgrade 3scale API Management and its components

Last Updated: 2024-06-28

Red Hat 3scale API Management 2.14 Migrating Red Hat 3scale API Management

Migrate or upgrade 3scale API Management and its components

Legal Notice

Abstract

Upgrade Red Hat 3scale API Management to the latest version via the 3scale operator and also find information to upgrade APIcast in an operator-based deployment.

Table of Contents

PREFACE

This guide provides the information to upgrade Red Hat 3scale API Management to the latest version via the 3scale operator. You will find details required to upgrade your 3scale installation from 2.13 to 2.14, as well as the steps to upgrade APIcast in an operator-based deployment.

To upgrade your 3scale On-premises deployment from 2.13 to 2.14, refer to the following guide:

- 3scale operator-based upgrade guide

To upgrade APIcast in an operator-based deployment, refer to the following guide:

- APIcast upgrade guide

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation.

To propose improvements, open a Jira issue and describe your suggested changes. Provide as much detail as possible to enable us to address your request quickly.

Prerequisite

- You have a Red Hat Customer Portal account. This account enables you to log in to the Red Hat Jira Software instance. If you do not have an account, you will be prompted to create one.

Procedure

1. Click the following link: Create issue.
2. In the **Summary** text box, enter a brief description of the issue.
3. In the **Description** text box, provide the following information:
   - The URL of the page where you found the issue.
   - A detailed description of the issue.
     You can leave the information in any other fields at their default values.
4. Click **Create** to submit the Jira issue to the documentation team.

Thank you for taking the time to provide feedback.

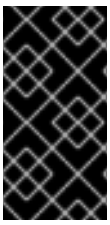
CHAPTER 1. 3SCALE API MANAGEMENT OPERATOR-BASED UPGRADE GUIDE: FROM 2.13 TO 2.14

WARNING

In some cases, upgrading 3scale API Management from 2.13 to 2.14 might fail to complete. To prevent this, follow the tested and approved steps in the solution from the Red Hat Customer Portal **before you attempt the upgrade** The solution includes detailed information on the root cause and conditions under which the issue can occur.

Upgrade Red Hat 3scale API Management from version 2.13 to 2.14, in an operator-based installation to manage 3scale on OpenShift 4.x.

To automatically obtain a micro-release of 3scale, make sure automatic updates is on. Do not set automatic updates if you are using an Oracle external database. To check this, see Configuring automated application of micro releases.



IMPORTANT

In order to understand the required conditions and procedure, read the entire upgrade guide before applying the listed steps. The upgrade process disrupts the provision of the service until the procedure finishes. Due to this disruption, make sure to have a maintenance window.

1.1. PREREQUISITES TO PERFORM THE UPGRADE

This section describes the required configurations to upgrade 3scale from 2.13 to 2.14 in an operator-based installation.

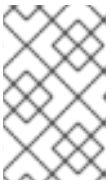
- An OpenShift Container Platform (OCP) 4.12, 4.13, or 4.14 cluster with administrator access. Ensure that your OCP environment is upgraded to at least version 4.12, which is the minimal requirement for proceeding with a 3scale update.
- 3scale 2.13 previously deployed via the 3scale operator.
- Make sure the latest CSV of the **threescale-2.13** channel is in use. To check it:
  - If the approval setting for the subscription is *automatic*, you should already be in the latest CSV version of the channel.
  - If the approval setting for the subscription is *manual*, make sure you approve all pending *InstallPlans* and have the latest CSV version.
  - Keep in mind if there is a pending install plan, there might be more pending install plans, which will only be shown after the existing pending plan has been installed.

1.2. UPGRADING FROM 2.13 TO 2.14 IN AN OPERATOR-BASED INSTALLATION

To upgrade 3scale from version 2.13 to 2.14 in an operator-based deployment:

1. Log in to the OCP console using the account with administrator privileges.
2. Select the project where the *3scale-operator* has been deployed.
3. Click **Operators > Installed Operators**.
4. Select **Red Hat Integration - 3scale > Subscription > Channel**
5. Edit the channel of the subscription by selecting *threescale-2.14* and save the changes. This will start the upgrade process.
6. Query the pods' status on the project until you see all the new versions are running and ready without errors:

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



NOTE

- The pods might have temporary errors during the upgrade process.
- The time required to upgrade pods can vary from 5-10 minutes.

7. After new pod versions are running, confirm a successful upgrade by logging in to the 3scale Admin Portal and checking that it works as expected.
8. Check the status of the *APIManager* objects and get the *YAML* content by running the following command. <myapimanager> represents the name of your *APIManager*:

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

- The new annotations with the values should be as follows:

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

After you have performed all steps, the 3scale upgrade from 2.13 to 2.14 in an operator-based deployment is complete.

1.3. UPGRADING FROM 2.13 TO 2.14 IN AN OPERATOR-BASED INSTALLATION WITH AN EXTERNAL ORACLE DATABASE

Follow this procedure to update your 3scale operator-based installation with an external Oracle database.

Procedure

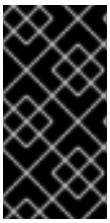
1. Follow these steps in Installing Red Hat 3scale API Management guide to create a new **system-oracle-3scale-2.14.0-1** image.

2. Follow the steps in Upgrading from 2.13 to 2.14 in an operator-based installation to upgrade the 3scale operator.

3. Once the upgrade is completed, update the APIManager custom resource with the new image created in the first step of this procedure as described in Installing 3scale API Management with Oracle using the operator.

CHAPTER 2. APICAST OPERATOR-BASED UPGRADE GUIDE: FROM 2.13 TO 2.14

Upgrading APIcast from 2.13 to 2.14 in an operator-based installation helps you use the APIcast API gateway to integrate your internal and external application programming interfaces (APIs) services with 3scale.



IMPORTANT

In order to understand the required conditions and procedure, read the entire upgrade guide before applying the listed steps. The upgrade process disrupts the provision of the service until the procedure finishes. Due to this disruption, make sure to have a maintenance window.

2.1. PREREQUISITES TO PERFORM THE UPGRADE

To perform the upgrade of APIcast from 2.13 to 2.14 in an operator-based installation, the following required prerequisites must already be in place:

- An OpenShift Container Platform (OCP) 4.12, 4.13, or 4.14 cluster with administrator access. Ensure that your OCP environment is upgraded to at least version 4.12, which is the minimal requirement for proceeding with an APIcast update.
- APIcast 2.13 previously deployed via the APIcast operator.
- Make sure the latest CSV of the **threescale-2.13** channel is in use. To check it:
  - If the approval setting for the subscription is *automatic*, you should already be in the latest CSV version of the channel.
  - If the approval setting for the subscription is *manual*, make sure you approve all pending *InstallPlans* and have the latest CSV version.
  - Keep in mind if there is a pending install plan, there might be more pending install plans, which will only be shown after the existing pending plan has been installed.

2.2. UPGRADING APICAST FROM 2.13 TO 2.14 IN AN OPERATOR-BASED INSTALLATION

Upgrade APIcast from 2.13 to 2.14 in an operator-based installation so that APIcast can function as the API gateway in your 3scale installation.

Procedure

1. Log in to the OCP console using the account with administrator privileges.
2. Select the project where the *APIcast operator* has been deployed.
3. Click **Operators > Installed Operators**.
4. In **Subscription > Channel**, select *Red Hat Integration - 3scale APIcast gateway* .
5. Edit the channel of the subscription by selecting the *threescale-2.14* channel and save the changes.

This will start the upgrade process.

6. Query the pods status on the project until you see all the new versions are running and ready without errors:

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

NOTE

- The pods might have temporary errors during the upgrade process.
- The time required to upgrade pods can vary from 5-10 minutes.

7. Check the status of the *APIcast* objects and get the *YAML* content by running the following command:

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

After you have performed all steps, the APIcast upgrade from 2.13 to 2.14 in an operator-based deployment is complete.

CHAPTER 3. MIGRATING FROM AN EMBEDDED POSTGRESQL 10 DATABASE TO AN EXTERNAL POSTGRESQL 10 DATABASE

IMPORTANT

- Before scaling **system.appSpec.replicas** to **1**, the database should be upgraded to the supported version, which is currently PostgreSQL 13. See Red Hat 3scale API Management Supported Configurations
- This documentation is about migrating from an embedded PostgreSQL 10 database to an external PostgreSQL 10 database. To upgrade from an external PostgreSQL 10 database to an external PostgreSQL 13 database, you must following the official PostgreSQL documentation.
- *Disclaimer: Links contained herein to external website(s) are provided for convenience only. Red Hat has not reviewed the links and is not responsible for the content or its availability. The inclusion of any link to an external website does not imply endorsement by Red Hat of the website or their entities, products or services. You agree that Red Hat is not responsible or liable for any loss or expenses that may result due to your use of (or reliance on) the external site or content.*

The process to move from an embedded PostgreSQL database to and external PostgreSQL database should happen with the same DB version. In this migration guide, it should be PostgreSQL 10. You should use external databases for production environments.

If you are using PostgreSQL as your **system-database**, use the supported version for external database installation with 3scale.

IMPORTANT

- These steps are general guidelines. Exact steps may vary depending on your operating system, version of PostgreSQL, and specific requirements of your database.
- Read the PostgreSQL documentation and release notes carefully before upgrading.
- Test this procedure in a non-production environment before applying it to a production deployment.
- This process disrupts the provision of the service until the procedure finishes. Due to this disruption, be sure to have a maintenance window.

Procedure

1. Use APIManager customer resource (CR) to scale down the **system-app** DeploymentConfig (DC):

```
apiVersion: apps.3scale.net/v1alpha1
kind: APIManager
metadata:
  name: <apimanager_sample>
spec:
```

```
system:
  appSpec:
    replicas: 0
wildcardDomain: <example.com>
```

2. Verify that the pods are scaled down:

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

   a. Wait for all the 3scale pods to have the status of **Terminated** before proceeding with the PostgreSQL migration.

3. Make a backup of the existing PostgreSQL database, including all data, configurations, and user accounts:

```
$ 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)
```

IMPORTANT

Do not pipe to **stdout**. Binary files get corrupted.

4. Dump with custom format:

```
$ 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. Download the backup:

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

6. Install the same version of PostgreSQL 10 that you deployed on 3scale in your target external system. Download the installation package from the PostgreSQL website following the installation instructions.

7. Copy and restore the backup you made of the existing PostgreSQL database, including all data, configurations, and user accounts to the target external system.

8. Create a new database in PostgreSQL:

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

9. Import the data from the backup file into the new PostgreSQL database.

   - Restore with custom format:

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

10. Verify that the data was successfully imported into the new PostgreSQL database by connecting to the database and running queries:

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

11. Update **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. Update APImanager CR to enable external database and scale up system:

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

13. Remove local postgresql deployment:

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

14. Verify that the pods are scaled up:

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

Additional resources

- PostgreSQL documentation
- PostgreSQL Downloads