



Red Hat 3scale API Management 2.4

Service Discovery

Add services from an underlying OpenShift cluster.

Red Hat 3scale API Management 2.4 Service Discovery

Add services from an underlying OpenShift cluster.

Legal Notice

Copyright © 2019 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, 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 Software Collections 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.

Abstract

This guide provides the information to recognize and import services running on OpenShift.

Table of Contents

PREFACE	3
CHAPTER 1. ABOUT SERVICE DISCOVERY	4
CHAPTER 2. USING SERVICE DISCOVERY	5
2.1. PREREQUISITES	5
2.2. DISCOVERING SERVICES	5
2.3. UPDATING THE SERVICE	5
CHAPTER 3. CONFIGURING SERVICE DISCOVERY	6
3.1. PREREQUISITES	6
3.2. CONFIGURING WITH SSO	6
3.2.1. Using OpenShift OAuth server	6
3.2.2. Using Red Hat SSO server (Keycloak)	7
3.3. CONFIGURING WITHOUT SSO	8
3.4. MAKING A SERVICE DISCOVERABLE	8

PREFACE

This guide will help you to import services from OpenShift.

CHAPTER 1. ABOUT SERVICE DISCOVERY

Service Discovery allows adding APIs for management by recognizing the [discoverable](#) running services in the OpenShift cluster. It has the following features:

- It uses the cluster API to query for services properly annotated for discovery.
- It configures 3scale to hit the service using an internal endpoint inside the cluster.
- It imports Open API Specification (Swagger) up to version 2.0 as 3scale Active Docs.
- It allows you to update the API integration and the Open API Specification at any time (resynchronize with the cluster).
- It supports OpenShift and Red Hat SSO authorization flows.
- It works with Fuse from version 7.2.

Additional notes

- For on premise installations, the API provider may have its own namespace and services.
- With Fuse, services will be deployed to Fuse production namespace.

CHAPTER 2. USING SERVICE DISCOVERY

Here you see the instructions about how to create or discover new services, and to update or refresh existing ones.

2.1. PREREQUISITES

You have created `<3scale-project>`.

2.2. DISCOVERING SERVICES

This section contains instructions to discover a new API service corresponding to an Open API Specification (OAS, also known as Swagger specification), if applicable; which is discovered from the cluster, for management with 3scale.

1. Log in to 3scale Tenant Admin Portal.
2. From the Dashboard, click **New API**.
3. Choose **Import from OpenShift**.
4. In the **Namespace** field, specify the OpenShift project.
5. In the **Name** field, type the the name of an OpenShift service within that given namespace.
6. Click **Create Service**.
7. Wait for the new API service to be asynchronously imported into 3scale. You will see a message in the upper right section of the Dashboard: **The service will be imported shortly. You will receive a notification when it is done.**

2.3. UPDATING THE SERVICE

This section explains how to update (refresh) an existing API service in 3scale, which was previously imported/discovered from the cluster, with the current definitions for the service in the cluster.

1. Log in to 3scale Administration Portal.
2. Navigate to the Overview page of the API service.
3. Click the *Refresh* link, next to Source: OpenSource.
4. Wait for the new API service to be asynchronously imported into 3scale.

CHAPTER 3. CONFIGURING SERVICE DISCOVERY

This guide helps you to configure Service Discovery with the following alternatives:

- [Section 3.2, “Configuring with SSO”](#).
- [Section 3.3, “Configuring without SSO”](#).

Additionally, you can find instructions to [make an API discoverable](#).

3.1. PREREQUISITES

- You need to deploy 3scale 2.4 to an OpenShift cluster (from version 3.10).
- To deploy 3scale to Openshift, you need to use [3scale-amp-openshift-templates](#).

3.2. CONFIGURING WITH SSO

When configuring with SSO, consider the following:

- Users will be redirected to the OAuth Server.
- If users are not logged into the OAuth Server, they will see a request to enter the credentials to log in.
- If it is the first time they use 3scale Service Discovery feature with SSO, the OAuth server will request them to authorize 3scale to perform the relevant actions.
- They will be redirected back to 3scale.

To configure Service Discovery feature with SSO, you have the following options:

1. [Section 3.2.1, “Using OpenShift OAuth server”](#).
2. [Section 3.2.2, “Using Red Hat SSO server \(Keycloak\)”](#).

3.2.1. Using OpenShift OAuth server

As a system administrator, you want users to individually authenticate and authorize 3scale to discover APIs on their behalf using OpenShift builtin OAuth server. This means that 3scale tenant admin users who are willing to use Service Discovery are required to be also OpenShift users, possibly with different access levels over the cluster among them.

1. Create an OpenShift OAuth client for 3scale. For more details about OpenShift authentication, see [OAuth Clients](#).

```
$ oc project default
$ cat <<-EOF | oc create -f -
kind: OAuthClient
apiVersion: v1
metadata:
  name: 3scale
secret: "<choose-a-client-secret>"
redirectURIs:
```

```
- "<3scale-master-domain-route>"
grantMethod: prompt
EOF
```

2. Edit 3scale Service Discovery settings.

```
$ oc project <3scale-project>
$ oc edit configmap system
```

You need to configure the following settings:

```
service_discovery.yml:
  production:
    enabled: true
    authentication_method: oauth
    oauth_server_type: builtin
    client_id: '3scale'
    client_secret: '<choose-a-client-secret>'
```

3. Ensure users have proper permissions to view cluster projects containing discoverable services.

Additional notes:

- After modifying **configmap**, you need to redeploy the **system-app** and **system-sidekiq** pods to apply the changes.
- Token lifespan: By default, OpenShift OAuth session tokens expire after 24 hours, as indicated in [OpenShift Token Options](#).

3.2.2. Using Red Hat SSO server (Keycloak)

As a system administrator, you can allow users to individually authenticate and authorize 3scale to discover services using [Red Hat SSO for OpenShift](#).

1. Create an OAuth client for 3scale in Red Hat OAuth server (Keycloak).

```
IMPORTANT: In the client configuration, make sure that the user's
username maps to `preferred_username`, so OpenShift can link
accounts.
```

2. Edit 3scale Service Discovery settings.

```
$ oc project <3scale-project>
$ oc edit configmap system
```

3. Make sure these settings are configured.

```
service_discovery.yml:
  production:
    enabled: true
    authentication_method: oauth
    oauth_server_type: rh_sso
    client_id: '3scale'
    client_secret: '<choose-a-client-secret>'
```

-
- 4. Make sure users have proper permissions to view cluster projects containing discoverable services

Additional notes:

- After modifying **configmap**, you need to redeploy the **system-app** and **system-sidekiq** pods to apply the changes.
- Token lifespan: By default, session tokens expire after 1 minute, as indicated in [Keycloak - Session and Token Timeouts](#). However, it is a good recommendation to set the timeout to an acceptable value of 1 day.

3.3. CONFIGURING WITHOUT SSO

To configure Service Discovery feature without SSO, you can use 3scale Single Service Account to authenticate to OpenShift API service. In this case, there is a seamless authentication to the cluster for the service discovery feature without an authorization layer at the user level, meaning all 3scale tenant administration users will have the same access level to the cluster while discovering API services through 3scale.

Procedure

1. Edit 3scale Service Discovery settings.

```
$ oc project <3scale-project>
$ oc edit configmap system
```

2. Make sure these settings are configured.

```
service_discovery.yml:
  production:
    enabled: <%= cluster_token_file_exists = File.exists?
(cluster_token_file_path =
'/var/run/secrets/kubernetes.io/serviceaccount/token') %>
    bearer_token: "<%= File.read(cluster_token_file_path) if
cluster_token_file_exists %>"
    authentication_method: service_account
```

3. Provide 3scale deployment **default** service account with the relevant permissions to view projects containing discoverable services, by following one of these options:

- Grant 3scale deployment **default** service account with **view** cluster level permission.

```
-----
  oc adm policy add-cluster-role-to-user view
system:serviceaccount:<3scale-project>:default
-----
```

- Apply a more restrictive policy. For more information, see [OpenShift - Service Accounts](#).

3.4. MAKING A SERVICE DISCOVERABLE

An API service, deployed or to be deployed to the OpenShift cluster, can be discoverable by 3scale. To achieve this, configure the OpenShift Service Object YAML definition to include the following metadata:

```
metadata:
  labels:
    discovery.3scale.net: "true"
  annotations:
    discovery.3scale.net/scheme: <http | https>
    discovery.3scale.net/port: '<port number of the service within the
cluster>'
    discovery.3scale.net/path: <relative base path of the API backend
from the root>
    discovery.3scale.net/description-path: <relative or absolute path
to the Open API Specification for the API>
```

From the described metadata, **port** and **scheme** are mandatory annotations.

Additionally, Service Discovery will only import the specification if the **Content - Type** header is one of the following values:

- **application/swagger+json**
- **application/vnd.oai.openapi+json**
- **application/json**



NOTE

Make sure the user or service account (depending on the configured authentication mode) has the right privileges to view the namespace and service.