Red Hat Managed Integration 2

Administering Red Hat Managed Integration 2

For Red Hat Managed Integration 2

Last Updated: 2020-09-17

Red Hat Managed Integration 2 Administering Red Hat Managed Integration 2

For Red Hat Managed Integration 2

Legal Notice

Abstract

This document provides instructions for configuring and managing Red Hat Managed Integration 2.

Table of Contents

PREFACE

This document provides instructions for Managed Integration administrators to:

- Understand the Managed Integration cluster environment
- Configure Managed Integration services for production scenarios
- Customize the Managed Integration cluster

CHAPTER 1. INTRODUCTION TO USER MANAGEMENT

In Managed Integration, all users have a *role* and belong to a particular *group*. You should understand these roles and groups so that you can manage users, and provide them with the appropriate permissions.

All Managed Integration users belong to the **rhmi-developers** group. Within this group, users can have either **dedicated-admin** or **developer** roles. The **dedicated-admin** role has elevated permissions and is one of the core OpenShift Dedicated roles.

1.1. MANAGED INTEGRATION ADMINISTRATORS

Managed Integration administrators have the **dedicated-admin** role. This role enables you to administer the Managed Integration cluster, and provide developers with the necessary permissions to use managed services.

As an administrator with the **dedicated-admin** role, you can do the following:

Red Hat Openshift

- Work with namespaces
- View all namespaces
- Edit and delete your own namespaces
- Edit and delete non-managed namespaces

Single Sign-On

- Manage users and permissions in the master realm
- Create realms
- Administer user-created realms

Red Hat 3scale API Management

- Elevate user permissions to an administrator level
- Edit routes
- View Pod logs in the 3scale namespace
- Create a product in the 3scale console

Red Hat Fuse Online

- "View" role in the Fuse namespace

Red Hat AMQ Online

- Create, read, update, delete, and list custom resources

1.2. MANAGED INTEGRATION DEVELOPERS

With the **developer** role, you can create namespaces in which to develop your cloud-native, integrated applications, and use the Customer Application SSO instance instance to secure your applications.

As a **developer**, you can do the following:

Red Hat Openshift

- Create, edit, and delete own namespaces

Single Sign-On

- Access the console
- Create realms

Red Hat Fuse Online

- "View" role in the Fuse namespace

1.3. GRANTING 3SCALE ADMIN PRIVILEGES TO USERS

As an administrator who is a member of the **dedicated-admins** group, you have administrator privileges in 3scale. However, developers do not. You must explicitly grant 3scale administrator permissions to Managed Integration developers.

Procedure

1. Log in to the Solution Explorer.
2. Navigate to the 3scale console from the **Manage APIs** item.
3. Grant administrator permissions to developers .

CHAPTER 2. CONFIGURING MANAGED INTEGRATION SERVICES

Managed Integration includes pre-installed component applications that developers can use to develop cloud-native, integrated applications. As an administrator, you can configure these component applications so that developers can use them for production scenarios.

2.1. ENABLING DEVELOPERS TO SECURE CUSTOMER APPLICATIONS

Managed Integration includes an instance of Red Hat Single Sign-On to enable you to protect the applications that you deploy on your cluster.

Red Hat manages this instance; however, all administrators have admin-level privileges to configure this instance to perform tasks such as:

- Creating users in the **master** realm
- Creating new realms
- Managing realms
  By default, developers can create new realms. Administrators can administer any realms that developers create, and can disable realm creation.

IMPORTANT

A user named **admin** manages the Red Hat Single Sign-On instance. Do not delete this user.

2.1.1. Disabling realm creation

By default, a developer can create a realm in the Customer Application SSO instance. This section describes how to disable this permission. You might want to disable this permission in a production cluster.

Prerequisites

- You are an administrator who is a member of the **dedicated-admins** group.

Procedure

1. Log in to the Solution Explorer.
2. Navigate to the Customer Application SSO instance from the **Protect customer applications** item in the Solution Explorer.
3. When prompted, choose the **Administration Console**.
4. Choose **Groups** from the menu for the Master realm.
5. Select the **rhmi-developers** group.
6. Click **Edit** from the **User Groups** menu.
7. Choose the **Role Mappings** tab.

8. Select **create-realm** in the **Assigned Roles** panel.
9. Click **Remove selected** to remove that role from the **rhmi-developers** group.

Verification

1. To verify the change, log in to the Customer Application SSO instance as a developer.
2. Make sure you cannot create a realm.

2.2. CONFIGURING AMQ ONLINE FOR PRODUCTION

If you plan to use AMQ Online in Managed Integration, you need to create an AMQ Online configuration for production. You must edit and apply YAML files using the command line (CLI) tools to configure AMQ Online.

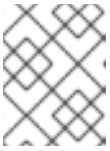
This section provides general guidance on configuring AMQ Online for production usage in Managed Integration.

Prerequisites

- You have the OpenShift CLI (**oc**) installed locally.

Procedure

1. Configure AMQ Online in Managed Integration as described in Configuring AMQ Online.
   - The name of the AMQ Online project in your cluster is **redhat-rhmi-amq-online**.
   - You must log in as an administrator.
2. Configure your address space to use an external authentication service as shown in the address space example using an external authentication service.



NOTE

The authentication services are configured by the AMQ Online service operator and are specified when creating an address space.

Additional resources

- For more information on Configuring AMQ Online, see the Installing and Managing AMQ Online documentation.

CHAPTER 3. MANAGING SOLUTION PATTERNS

As an administrator, you can customize the list of Solution Patterns that developers can access in Solution Explorer.

3.1. SUBSCRIBING TO SOLUTION PATTERN CONTENT

You can display a list of the Solution Patterns you are subscribed to on the Solution Explorer home page. Any administrator who is a member of the **dedicated-admins** group can add Solution Patterns to your cluster. Use the following procedure to list the Solution Patterns from the Git repositories you are subscribed to, on the Solution Explorer homepage.

Prerequisite

- You are an administrator who is a member of the **dedicated-admins** group.

Procedure

1. Log in to Solution Explorer.
2. Click the gear icon in the top right.
3. Click the **Solution Pattern content** tab.
4. Enter the URLs of the Solution Pattern Git repositories you want to add to your cluster using the following syntax:

   ```
   https://github.com/<org>/<repo>.git
   ```

   where **<org>** is the name of your GitHub organization and **<repo>** is the name of your repository.

   a. List URLs in the order you want them to appear in the Solution Explorer.

   b. Enter one URL per line.

   c. To include a specific branch, append #<branch-name> to the URL. For example:

      ```
      https://github.com/<org>/<repo>.git#version-one
      ```

      NOTE

      To remove a Solution Pattern, delete the corresponding URL from the list.

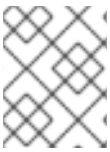
5. Click **Save**. This triggers an automatic refresh of the Solution Explorer.
6. When the deployment is complete, refresh your browser. You should now see new Solution Patterns available from the dashboard.
7. The Solution Explorer does not automatically update when changes are made to the Git repository.
   Refresh the Solution Explorer to view the changes:

   a. Click the gear icon in the top right to display the **Application settings** screen.

b. Click **Save** to trigger a refresh of the Solution Explorer.

c. Refresh the browser when the Solution Explorer refresh is complete.

d. Navigate to the **Solution Patterns** tab to see the updated content.



NOTE

You can access the Git repository that contains the Solution Pattern source code by clicking the **Repository** link in the **All Solutions Patterns** tab in the Solution Explorer.

Additional resources

- To learn more about creating Solution Patterns, see the Creating Solution Patterns documentation.

CHAPTER 4. SETTING NETWORK POLICIES

Network policies specify how groups are allowed to communicate with each other and other network endpoints. In your Managed Integration cluster, you define network policies to control how namespaces, the Managed Integration services, and customer applications communicate with each other.

To set a network policy, you must create and apply a **NetworkPolicy** Custom Resource object in the Managed Integration cluster. You can create **NetworkPolicy** objects to set the following types of network policies in your Managed Integration cluster:

NOTE

Network policies are additive. If multiple policies are created, the network policies are restricted to what is allowed by the union of those policies' ingress and egress rules.

- Section 4.1, “Enabling communication between managed services and customer applications”
- Section 4.2, “Enabling communication between customer applications”
- Section 4.3, “Enabling communication between managed services and projects”
- Section 4.4, “Disabling communication from a managed service to a project”

4.1. ENABLING COMMUNICATION BETWEEN MANAGED SERVICES AND CUSTOMER APPLICATIONS

You can create **NetworkPolicy** objects to define granular rules describing the Ingress network traffic that is allowed for projects in your cluster. By default, when you create projects in a cluster, communication between the projects is disabled.

This procedure describes how to enable communication for a project so that Managed Integration services (such as 3scale), can access customer applications.

Prerequisites

- You have installed the OpenShift command-line interface (CLI), commonly known as **oc**.

Procedure

1. Log in to the cluster using the **oc** login command.
2. Use the following command to change the project:

   ```
   $ oc project <project_name>
   ```

   where **<project_name>** is the name of a project that you want to accept communications from other projects.
3. Create a **NetworkPolicy** object:
   a. Create a **allow-from-middleware-namespaces.yaml** file.
   b. Define a policy in the file you just created, such as in the following example:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-from-middleware-namespaces
spec:
  podSelector:
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              integreatly-middleware-service: 'true'
```

c. Run the following command to create the policy object:

```
$ oc create -f allow-from-middleware-namespaces.yaml -n <project>

networkpolicy "allow-from-middleware-namespaces" created
```

Additional resources

- For more information on Networking in a Managed Integration cluster, see the Understanding Networking documentation.
- For more information on deleting a NetworkPolicy object, see the Deleting a NetworkPolicy object documentation.

4.2. ENABLING COMMUNICATION BETWEEN CUSTOMER APPLICATIONS

You can enable communication between user applications.

Prerequisites

- You have installed the OpenShift command-line interface (CLI), commonly known as **oc**.

Procedure

1. Log in to the cluster using the **oc** login command.
2. Use the following command to change the project:

   ```
   $ oc project <project_name>
   ```

   where **<project_name>** is the name of a project that you want to accept communications from other projects with the label **project=myproject**.
3. Create a NetworkPolicy object:
   a. Create a **allow-from-myproject-namespaces.yaml** file.
   b. Define a policy in the file you just created, such as in the following example.
   This policy enables incoming communication for a specific project (**myproject**):

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-from-myproject-namespace
spec:
  podSelector:
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              project: myproject
```

4. Run the following commands to create the policy object:

```
$ oc create -f allow-from-myproject-namespace.yaml -n <project>
networkpolicy "allow-from-myproject-namespace" created
```

Additional resources

- For more information on Networking in a Managed Integration cluster, see the Understanding Networking documentation.
- For more information on deleting a NetworkPolicy object, see the Deleting a NetworkPolicy object documentation.

4.3. ENABLING COMMUNICATION BETWEEN MANAGED SERVICES AND PROJECTS

By default, when you create projects in a cluster, communication between the projects is disabled. This procedure describes how to enable communication for a project.

Prerequisites

- You have installed the OpenShift command-line interface (CLI), commonly known as **oc**.

Procedure

1. Log in to the cluster using the **oc** login command.
2. Use the following command to change the project:

```
$ oc project <project_name>
```

   where **<project_name>** is the name of a project that you want to accept communications from other projects.

3. Create a NetworkPolicy object:
   a. Create a **NetworkPolicy.yaml** file.
   b. Define a policy in the file you just created, such as in the following example. This policy enables incoming communication for all projects in the cluster:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: allow-all
spec:
  podSelector:
  ingress:
  - {}
```

NOTE

This policy configuration enables this project to communicate with all projects in the cluster.

c. Run the following command to create the policy object:

```
$ oc create -f <policy-name>.yaml -n <project>
```

Additional resources

- For more information on Networking in a Managed Integration cluster, see the Understanding Networking documentation.
- For more information on deleting a NetworkPolicy object, see the Deleting a NetworkPolicy object documentation.

4.4. DISABLING COMMUNICATION FROM A MANAGED SERVICE TO A PROJECT

By default, your projects are created with a template that allows communication from a managed service. For example, Red Hat 3scale API Management can communicate with all of your projects by default. You can disable that communication.

Prerequisites

- You have a project you want to isolate from the managed services.

Procedure

1. Log in to the cluster using the **oc** login command.
2. Use the following command to change the project:

```
$ oc project <project_name>
```

where **<project_name>** is the name of a project that you want to isolate from the managed services.

3. Create a NetworkPolicy object:
   a. Create a **deny-all.yaml** file.
   b. Define a policy in the file you just created, such as in the following example:

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: deny-all
spec:
  podSelector: {}
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              integreatly-middleware-service: 'true'
```

c. Run the following command to create the policy object:

```
$ oc create -f <policy-name>.yaml -n <project>
```

Additional resources

- For more information on Networking in a Managed Integration cluster, see the Understanding Networking documentation.
- For more information on deleting a NetworkPolicy object, see the Deleting a NetworkPolicy object documentation.

Revised on 2020-09-17 09:14:11 UTC