Configuring AMQ Interconnect sites using the CLI

For Use with AMQ Interconnect 2.0 TECHNOLOGY PREVIEW
Legal Notice

Copyright © 2021 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 ® 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 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 describes how to install, configure, and manage AMQ Interconnect sites to build a service network.
# Table of Contents

PREFACE ........................................................................................................... 3

CHAPTER 1. USING THE AMQ INTERCONNECT CLI ........................................... 4
  1.1. INSTALLING THE SKUPPER CLI .................................................................. 4
  1.2. CREATING A SITE USING THE CLI ............................................................ 4
  1.3. CUSTOM SITES ........................................................................................... 5
  1.4. LINKING SITES .......................................................................................... 5
  1.5. EXPOSING SERVICES ON THE SERVICE NETWORK ................................. 6
    1.5.1. Exposing simple services on the service network .................... 7
    1.5.2. Exposing complex services on the service network .................. 7
  1.6. CLI OPTIONS FOR WORKING WITH DIFFERENT CLUSTERS .................. 8
Making open source more inclusive

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.

IMPORTANT

AMQ Interconnect 2.0 Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production.

These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see https://access.redhat.com/support/offerings/techpreview.
CHAPTER 1. USING THE AMQ INTERCONNECT CLI

Using the **skupper** command-line interface (CLI) allows you to create and manage AMQ Interconnect sites from the context of the current namespace.

A typical workflow is to create a site, link sites together, and expose services to the service network.

### 1.1. INSTALLING THE **skupper** CLI

Installing the **skupper** command-line interface (CLI) provides a simple method to get started with AMQ Interconnect.

#### Procedure

1. Ensure your subscription has been activated and your system is registered.

2. Subscribe to the required repositories:
   - **Red Hat Enterprise Linux 7**
     ```bash
     $ sudo subscription-manager repos --enable=interconnect-2-for-rhel-7-server-rpms
     ```
   - **Red Hat Enterprise Linux 8**
     ```bash
     $ sudo subscription-manager repos --enable=interconnect-2-for-rhel-8-x86_64-rpms
     ```

3. Use the **yum** or **dnf** command to install the **skupper** package:
   ```bash
   $ sudo yum install skupper
   ```

4. Verify the installation.
   ```bash
   $ skupper version
   client version 0.6.0-redhat-interconnect-2.0.0
   ```

### 1.2. CREATING A SITE USING THE CLI

A service network consists of AMQ Interconnect sites. This section describes how to create a site using the default settings.

#### Prerequisites

- The **skupper** CLI is installed.
- You are logged into the cluster.
- The services you want to expose on the service network are in the active namespace.

#### Procedure

1. Create a default site:
$ skupper init

2. Check the site:

$ skupper status

Skupper is enabled for namespace "west" in interior mode. It is not connected to any other sites.

The default settings include:

- **console** - The Skupper console is provisioned with a single user. The password for the admin user is stored in the `skupper-console-users` secret. For more information on the console, see [Monitoring AMQ Interconnect sites using the console](#).

- **site name** - The site name defaults to the namespace name, for example, west.

### 1.3. CUSTOM SITES

The default `skupper init` creates sites that satisfy typical requirements.

If you require a custom configuration, note the following options:

- Creating a site without a console:

  $ skupper init --enable-console false

- Configuring console authentication. There are a number of `skupper` options regarding authentication for the console:

  **--console-auth <authentication-mode>**

  Set the authentication mode for the console:

  - **openshift** - Use OpenShift authentication, so that users who have permission to log into OpenShift and view the Project (namespace) can view the console.

  - **internal** - Use AMQ Interconnect authentication, see the `console-user` and `console-password` options.

  - **unsecured** - No authentication, anyone with the URL can view the console.

  **--console-user <username>**

  Username for the console user when authentication mode is set to `internal`. Defaults to admin.

  **--console-password <password>**

  Password for the console user when authentication mode is set to `internal`. If not specified, a random passwords is generated.

### 1.4. LINKING SITES

A service network consists of AMQ Interconnect sites. This section describes how to link sites to form a service network.
Linking two sites requires a single initial directional connection. However:

- Communication between the two sites is bidirectional, only the initial linking is directional.

- The choice of direction for linking is typically determined by accessibility. For example, if you are linking an OpenShift Dedicated cluster with a CodeReady Containers cluster, you must link from the CodeReady Containers cluster to the OpenShift Dedicated cluster because that route is accessible.

**Procedure**

1. Determine the direction of the link. If both clusters are publicly addressable, then the direction is not significant. If one of the clusters is addressable from the other cluster, perform step 2 below on the addressable cluster.

2. Generate a token on the cluster that you want to link to:

   
   ```
   $ skupper token create <filename>
   
   where `<filename>` is the name of a YAML file that is saved on your local filesystem.
   
   This file contains a key and the location of the site that created it.
   ```

   **NOTE**

   Access to this file provides access to the service network. Protect it appropriately.

3. Use a token on the cluster that you want to connect from:

   a. Create a link to the service network:

   ```
   $ skupper link create <filename> [-name <link-name>]
   
   where `<filename>` is the name of a YAML file generated from the `skupper token create` command and `<link-name>` is the name of the link.
   ```

   b. Check the link:

   ```
   $ skupper link status
   Connection for conn1 not active
   
   In this example no `<link-name>` was specified, the name defaulted to `conn1`.
   ```

4. If you want to delete a link:

   ```
   $ skupper link delete <link-name>
   
   where `<link-name>` is the name of the link specified during creation.
   ```

### 1.5. EXPOSING SERVICES ON THE SERVICE NETWORK

After creating a service network, exposed services can communicate across that network.
The `skupper` CLI has two options for exposing services:

- **expose** supports simple use cases, for example, a deployment with a single service. See Section 1.5.1, “Exposing simple services on the service network” for instructions.

- **service create** and **service bind** is a more flexible method of exposing services, for example, if you have multiple services for a deployment. See Section 1.5.2, “Exposing complex services on the service network” for instructions.

### 1.5.1. Exposing simple services on the service network

This section describes how services can be enabled for a service network for simple use cases.

#### Procedure

1. Create a deployment, some pods, or a service in one of your sites, for example to create the backend service for the tutorial:

   ```
   $ kubectl create deployment hello-world-backend --image quay.io/skupper/hello-world-backend
   ``

   This step is not AMQ Interconnect-specific, that is, this process is unchanged from standard processes for your cluster.

2. Create a service that can communicate on the service network:

   ```
   $ skupper expose [deployment <name>|pods <selector>|statefulset <statefulsetname>|service <name>]
   ``

   where

   - **<name>** is the name of your deployment
   - **<selector>** is a pod selector
   - **<statefulsetname>** is the name of a statefulset

   For the example deployment in step 1, you create a service using the following command:

   ```
   $ skupper expose deployment/hello-world-backend --port 8080
   ``

   Options for this command include:

   ```
   --port <port-number>
   ```

   Specify the port number that this service is available on the service network.

   ```
   --target-port <port-number>
   ```

   Specify the port number of pods that you want to expose.

   **NOTE**

   If you do not specify ports, `skupper` uses the `containerPort` value of the deployment.

### 1.5.2. Exposing complex services on the service network
This section describes how services can be enabled for a service network for more complex use cases.

### Procedure

1. Create a deployment, some pods, or a service in one of your sites, for example to create the backend service for the tutorial:

   ```
   $ kubectl create deployment hello-world-backend --image quay.io/skupper/hello-world-backend
   ```

   This step is not AMQ Interconnect-specific, that is, this process is unchanged from standard processes for your cluster.

2. Create a service that can communicate on the service network:

   ```
   $ skupper service create <name> <port>
   ```

   where

   - `<name>` is the name of the service you want to create
   - `<port>` is the port the service uses

   For the example deployment in step 1, you create a service using the following command:

   ```
   $ skupper service create hello-world-backend 8080
   ```

3. Bind the service to a cluster service:

   ```
   $ skupper service bind <service-name> <target-type> <target-name>
   ```

   where

   - `<service-name>` is the name of the service on the service network
   - `<target-type>` is the object you want to expose, `deployment`, `statefulset`, `pods`, or `service`.
   - `<target-name>` is the name of the cluster service

   For the example deployment in step 1, you bind the service using the following command:

   ```
   $ skupper service bind hello-world-backend deployment hello-world-backend
   ```

### 1.6. CLI OPTIONS FOR WORKING WITH DIFFERENT CLUSTERS

By default, all skupper commands apply to the cluster you are logged into and the current namespace. The following skupper options allow you to override that behavior and apply to all commands:

```
--namespace <namespace-name>
```

Apply command to `<namespace-name>`. For example, if you are currently working on `frontend` namespace and want to initialize a site in the `backend` namespace:

```
$ skupper init --namespace backend
```
**--kubeconfig <kubeconfig-path>**

Path to the kubeconfig file - This allows you run multiple sessions to a cluster from the same client. An alternative is to set the `KUBECONFIG` environment variable. See the tutorial for an example of using kubeconfig files.

**--context <context-name>**

The kubeconfig file can contain defined contexts, and this option allows you to use those contexts.

*Revised on 2021-06-17 10:50:31 UTC*