



## Red Hat AMQ 2021.Q2

# OpenShift を使用したサービスネットワークの作成

AMQ Interconnect 2.0 テクノロジープレビューでの使用



# Red Hat AMQ 2021.Q2 OpenShift を使用したサービスネットワークの作成

---

AMQ Interconnect 2.0 テクノロジープレビューでの使用

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## 法律上の通知

Copyright © 2021 | You need to change the HOLDER entity in the en-US/Creating\_a\_service\_network\_with\_OpenShift.ent file |.

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.

## 概要

このチュートリアルでは、OpenShift で AMQ Interconnect サイトを作成し、サービスネットワークを構築する方法を説明します。

---

## 目次

前書き .....	3
第1章 OPENSIFT を使用したサービスネットワークの作成 .....	4
1.1. AMQ INTERCONNECT 2.0 の概要	4
1.2. SKUPPER CLI のインストール	5
1.3. 端末セッションの設定	5
1.4. 両方のクラスターへのサービスネットワークルーターのインストール	6
1.5. サービスネットワーク作成用の NAMESPACE の接続	7
1.6. フロントエンドサービスの作成	7
1.7. バックエンドサービスを作成してそのサービスをサービスネットワーク上で利用可能にする手順	8
1.8. サービスネットワークの終了	9



## 前書き

### 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。これは大規模な取り組みであるため、これらの変更は今後の複数のリリースで段階的に実施されます。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#)をご覧ください。



#### 重要

AMQ Interconnect 2.0 テクノロジープレビューの機能は、Red Hat の本番環境のサービスレベルアグリーメント (SLA) ではサポートされず、機能的に完全ではないことがあります。Red Hat では、これらについて実稼働環境での使用を推奨していません。

テクノロジープレビューの機能は、最新の技術をいち早く提供して、開発段階で機能のテストやフィードバックの収集を可能にするために提供されます。Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、<https://access.redhat.com/ja/support/offerings/techpreview> を参照してください。

## 第1章 OPENSIFT を使用したサービスネットワークの作成

このチュートリアルでは、**skupper** コマンドラインインターフェース (CLI) を使用して、OpenShift クラスタにバックエンドサービスがある OpenShift クラスタでフロントエンドサービスを接続する方法を説明します。

### 前提条件

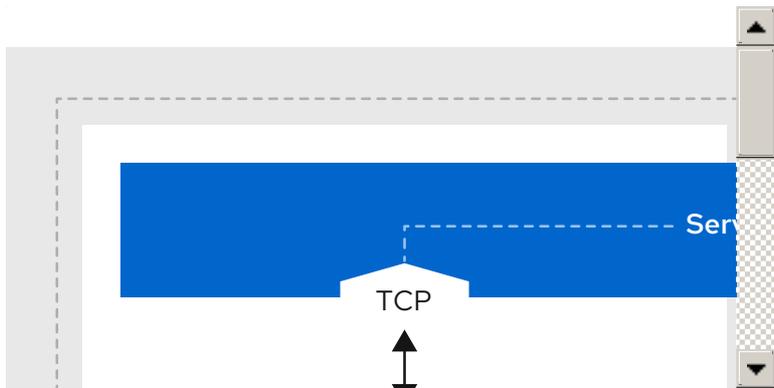
- 2つの OpenShift クラスタのプロジェクトに対するアクセス権。**cluster-admin** アクセスは必要ありません。
- OpenShift クラスタの1つが他のクラスタからアドレス指定できる。

このチュートリアルでは、以下の namespace に接続する方法について説明します。

- **west** - フロントエンドサービスを実行します。これは通常パブリッククラスタです。
- **east** - バックエンドサービスを実行します。

### 1.1. AMQ INTERCONNECT 2.0 の概要

Interconnect 2.0 では、ハイブリッドクラウド全体でサービスをリンクするサービスネットワークが導入されました。サービスネットワークは、異なるネットワークの場所で実行中のサービス間で通信を可能にします。これにより、地理的に分散したサービスが同じサイトですべて実行されているかのように接続できます。



たとえば、フロントエンドをパブリック OpenShift クラスタに、バックエンドをプライベート OpenShift クラスタにデプロイして、サービスネットワークに接続できます。

サービスネットワークは、以下の機能を提供します。

- セキュリティー (デフォルト)。すべてのサイトトラフィックは、プライベートかつ専用の認証局 (CA) を使用して相互 TLS によって保護されます。
- プライベートクラスタの場合でも、OpenShift クラスタ間の接続が容易になる。
- サービスネットワークが変更なしで既存の TCP ベースのアプリケーションをサポートする。
- サービスネットワークコンソールを使用して、複数の OpenShift クラスタに分散されるアプリケーショントラフィックを監視する。

**skupper** CLI を使用してサービスネットワークをデプロイし、管理します。

## 1.2. SKUPPER CLI のインストール

**skupper** コマンドラインインターフェース (CLI) をインストールすると、AMQ Interconnect を簡単に使用開始できます。

### 手順

1. サブスクリプションがアクティベートされ、システムが登録されていることを確認します。
2. 必要なりポジトリにサブスクライブします。

#### Red Hat Enterprise Linux 7

```
$ sudo subscription-manager repos --enable=interconnect-2-for-rhel-7-server-rpms
```

#### Red Hat Enterprise Linux 8

```
$ sudo subscription-manager repos --enable=interconnect-2-for-rhel-8-x86_64-rpms
```

3. **yum** または **dnf** コマンドを使用して **skupper** パッケージをインストールします。

```
$ sudo yum install skupper
```

4. インストールを確認します。

```
$ skupper version
client version 0.6.0-redhat-interconnect-2.0.0
```

## 1.3. 端末セッションの設定

この手順では、AMQ Interconnect を別のクラスターに設定する時に問題を回避するような設定を端末セッションが使用するよう設定する方法を説明します。

以下の表は、ターミナルセッションの設定方法を示しています。

表1.1 端末セッション

west 端末セッション	east 端末セッション
<pre>\$ oc project west</pre>	<pre>\$ oc project east</pre>

### 前提条件

- OpenShift CLI がインストールされている。**oc** のインストール方法については、[OpenShift CLI のドキュメント](#) を参照してください。



## 注記

OpenShift 4.6 以降では、[Web ターミナル](#) のドキュメントで説明されているように、Web ターミナルを使用して以下の手順を実行します。

## 手順

1. **west** namespace で作業するように端末セッションを開始し、**KUBECONFIG** 環境変数を設定します。

```
$ export KUBECONFIG=$HOME/.kube/config-west
```

このセッションは、後で **west** 端末セッションと呼ばれます。

2. **east** namespace で作業するように端末セッションを開始し、**KUBECONFIG** 環境変数を設定します。

```
$ export KUBECONFIG=$HOME/.kube/config-east
```

このセッションは、後で **east** 端末セッションと呼ばれます。

3. 各ターミナルセッションで、以下のように OpenShift クラスタにログインします。

```
$ oc login
```

## 1.4. 両方のクラスタへのサービスネットワークルーターのインストール

1. **west** 端末セッションで以下を行います。
  - a. **west** プロジェクト (namespace) を作成します。

```
$ oc new-project west
```

- b. サービスネットワークルーターを作成します。

```
$ skupper init
```

- c. サイトのステータスを確認します。

```
$ skupper status
```

出力は以下のようになります。

```
Skupper enabled for namespace 'west'. It is not connected to any other sites.
```

2. **east** 端末セッションで以下を行います。
  - a. **east** プロジェクト (namespace) を作成します。

```
$ oc new-project east
```

- b. サービスネットワークルーターを作成します。

```
$ skupper init
```

- c. サイトのステータスを確認します。

```
$ skupper status
```

出力は以下のようになります。

```
Skupper enabled for namespace 'east'. It is not connected to any other sites.
```

## 1.5. サービスネットワーク作成用の NAMESPACE の接続

サービスネットワークルーターがインストールされている場合には、そのルーターを相互に接続し、サービスネットワーク全体でサービスを共有できます。

### 手順

1. west 端末セッションで、west namespace への接続を許可する接続トークンを作成します。

```
$ skupper token create $HOME/secret.yaml
```

このコマンドは、ホームディレクトリーに **secret.yaml** ファイルを作成し、セキュアな接続の作成に使用できます。

2. east 端末セッションで、トークンを使用して west namespace への接続を作成します。

```
$ skupper link create $HOME/secret.yaml
```

3. west 端末セッションからサイトのステータスを確認します。

```
$ skupper status
```

出力は以下のようになります。

```
Skupper is enabled for namespace "west" in interior mode. It is connected to 1 other site. It has no exposed services.
The site console url is: https://<skupper-url>
The credentials for internal console-auth mode are held in secret: 'skupper-console-users'
```

## 1.6. フロントエンドサービスの作成

フロントエンドサービスは、バックエンドアプリケーションのメッセージから表示する単純な Python アプリケーションです。

### 手順

west 端末セッションですべてのタスクを実行します。

1. フロントエンドサービスをデプロイします。

```
$ oc create deployment hello-world-frontend --image quay.io/skupper/hello-world-frontend
```

2. フロントエンドデプロイメントをクラスターサービスとして公開します。

```
$ oc expose deployment hello-world-frontend --port 8080 --type LoadBalancer
```

3. フロントエンドのルートを作成します。

```
$ oc expose svc/hello-world-frontend
```

4. フロントエンドのルートを確認します。

- a. ルートの詳細を取得します。

```
$ oc get routes
```

出力は以下のようになります。

```
NAME          HOST/PORT
hello-world-frontend <frontend-url>
```

- b. ブラウザーで **<frontend-url>** の値に移動すると、フロントエンドがバックエンドと通信できないため、以下のようなメッセージが表示されます。

```
Trouble! HTTPConnectionPool(host='hello-world-backend', port=8080): Max retries
exceeded with url: /api/hello (Caused by
NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7fbfcdf0d1d0>:
Failed to establish a new connection: [Errno -2] Name or service not known'))
```

この状況を解決するには、バックエンドサービスを作成し、サービスネットワーク上で利用可能にする必要があります。

## 1.7. バックエンドサービスを作成してそのサービスをサービスネットワーク上で利用可能にする手順

バックエンドサービスは **east** namespace で実行され、デフォルトではサービスネットワークで利用できません。 **skupper** コマンドを使用して、サービスネットワークのすべての namespace にサービスを公開します。バックエンドアプリケーションは、メッセージをフロントエンドアプリケーションに渡す単純な Python アプリケーションです。

### 手順

1. バックエンドサービスを east 端末セッションにデプロイします。

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

2. サービスネットワーク上で、east 端末セッションからバックエンドサービスを公開します。

```
$ skupper expose deployment hello-world-backend --port 8080 --protocol tcp
```

3. west 端末セッションからサイトのステータスを確認します。

```
$ skupper status
```

出力は以下のようになります。

```
Skupper is enabled for namespace "west" in interior mode. It is connected to 1 other site. It has 1 exposed service.
```

サービスは **east** namespace から公開されます。

4. west 端末セッションでフロントエンドルートを確認します。

a. ルートの詳細を取得します。

```
$ oc get routes
```

出力は以下のようになります。

```
NAME          HOST/PORT
hello-world-frontend <frontend-url>
```

b. ブラウザーで **<frontend-url>** の値に移動すると、以下のようなメッセージが表示されません。

```
I am the frontend. The backend says 'Hello from hello-world-backend-78cd4d7d8c-plr9 (1)'.
```

これは、フロントエンドが、別の OpenShift クラスターからサービスネットワーク上でバックエンドサービスを呼び出す方法を示しています。

## その他のリソース

- [コンソールを使用した AMQ Interconnect サイトのモニタリング](#)
- [CLI を使用した AMQ Interconnect サイトの設定](#)

## 1.8. サービスネットワークの終了

この手順では、作成したサービスネットワークを削除する方法を説明します。

1. west 端末セッションから **west** namespace を削除します。

```
$ oc delete project west
```

2. **east** namespace を east 端末セッションから削除します。

```
$ oc delete project east
```

改訂日時: 2021-08-22 20:23:33 +1000