# Red Hat Fuse 7.11

## Fuse on OpenShift の管理

Fuse Console を使用した Fuse アプリケーションの管理

Last Updated: 2023-12-12

# Red Hat Fuse 7.11 Fuse on OpenShift の管理

Fuse Console を使用した Fuse アプリケーションの管理

## 法律上の通知



## 概要

Fuse アプリケーションをデプロイするときに Fuse Console を使用すると、Red Hat Fuse インテグレーションを監視および操作できます。

# 目次

# はじめに

Red Hat Fuse は、Fuse インテグレーションを表示および管理する、以下の 2 つのエンタープライズ管理ツールを提供します。

- Fuse Console は、ブラウザーからアクセスする Web ベースのコンソールで、実行中の Fuse コンテナーを監視および管理します。Fuse Console は Hawtio オープンソースソフトウェア (https://hawt.io/) をベースにしています。このガイドでは Fuse Console の使用方法を説明します。

- Prometheus は、Fuse ディストリビューションのシステムおよびインテグレーションレベルのメトリックを保存します。Grafana などのグラフィカル分析インターフェイスを使用して、保存された履歴データを表示および分析できます。Prometheus の使用に関する詳細は、以下のドキュメントを参照してください。

  - Prometheus ドキュメント
  - Fuse on OpenShift ガイド
  - OpenShift Container Platform での Fuse Online のインストールと操作

本ガイドの対象者は、Red Hat Fuse on JBoss EAP の管理者です。このガイドは、Red Hat Fuse プラットフォーム、Apache Camel、および所属組織の処理要件に精通していることを前提としています。

# 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 プロパティー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。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、Red Hat CTO である Chris Wright のメッセージ をご覧ください。

# 第1章 FUSE CONSOLE

Red Hat Fuse Console は、HawtIO オープンソースソフトウェアをベースとする Web コンソールです。サポートされるブラウザーのリストは Red Hat Fuse でサポートされる構成 を参照してください。

Fuse Console は、デプロイされた 1 つ以上の Fuse コンテナーの詳細を確認および管理するための中央インターフェイスを提供します。また、Red Hat Fuse およびシステムリソースの監視、更新の実行、およびサービスの起動と停止を行うこともできます。

Fuse Console は、Red Hat Fuse スタンドアロンをインストールするか、Fuse on OpenShift を使用する場合に利用できます。Fuse Console で表示および管理できるインテグレーションは、実行されているプラグインによって異なります。使用できるプラグインには以下が含まれます。

- Camel
- JMX
- OSGI
- Runtime
- Logs

# 第2章 OPENSHIFT 4.X での FUSE CONSOLE の設定

OpenShift 4.x では、Fuse Console の設定に、インストール、およびデプロイが必要になります。Fuse Console のインストールおよびデプロイには、以下のオプションがあります。

- 「OperatorHub を使用した OpenShift 4.x での Fuse Console のインストールおよびデプロイ」
  特定の namespace で Fuse アプリケーションにアクセスするために、Fuse Console Operator を使用して Fuse Console をインストールおよびデプロイします。Operator は、Fuse Console のセキュリティー保護を処理します。

- 「コマンドラインを使用した OpenShift 4.x での Fuse Console のインストールおよびデプロイ」
  OpenShift クラスターの複数の namespace または特定の namespace にある Fuse アプリケーションにアクセスするために、コマンドラインと Fuse Console テンプレートの 1 つを使用して Fuse Console をインストールおよびデプロイします。デプロイ前にクライアント証明書を生成して、Fuse Console をセキュアにする必要があります。

任意で、「OpenShift 4.x 上の Fuse Console のロールベースアクセス制御」 の説明通りに、Fuse Console のロールベースアクセス制御 (RBAC) をカスタマイズすることができます。

## 2.1. OPERATORHUB を使用した OPENSHIFT 4.X での FUSE CONSOLE のインストールおよびデプロイ

OpenShift 4.x に Fuse Console をインストールするには、OpenShift OperatorHub で提供される Fuse Console Operator を使用します。Fuse Console をデプロイするには、インストールされた Operator のインスタンスを作成します。

**前提条件**

- コンテナーイメージの **registry.redhat.io** を使用した認証 で説明されているように、**registry.redhat.io** での認証を設定している。

- Fuse Console のロールベースアクセス制御 (RBAC) をカスタマイズする場合は、Fuse Console Operator のインストール先と同じ OpenShift namespace に RBAC 設定マップファイルが必要になる。OpenShift 4.x 上の Fuse Console のロールベースアクセス制御 の説明にしたがって、デフォルトの RBAC 動作を使用する場合は、設定マップファイルを指定する必要はありません。

**手順**

Fuse Console をインストールおよびデプロイするには、以下を行います。

1. Web ブラウザーで、**cluster admin** 権限を持つユーザーとして OpenShift コンソールにログインします。

2. **Operators** をクリックした後、**OperatorHub** をクリックします。

3. 検索フィールドウインドウに **Fuse Console** と入力し、Operator のリストを絞り込みます。

4. **Fuse Console Operator** をクリックします。

5. Fuse Console Operator インストールウィンドウで **Install** をクリックします。
   **Create Operator Subscription** フォームが表示されます。

   - **Update Channel** には **7.11.x** を選択します。

- **Installation Mode** では、デフォルト (クラスターの特定の namespace) を受け入れます。operator のインストール後に Fuse Console をデプロイする場合、クラスター上のすべての namespace でアプリケーションを監視するか、Fuse Console operator がインストールされた namespace のみでアプリケーションを監視するかを選択できる点に注意してください。

- **Installed Namespace** には、Fuse Console Operator をインストールする namespace を選択します。

- **Update Approval** では、**Automatic** または **Manual** を選択し、Fuse Console の更新が OpenShift によってどのように対処されるかを設定します。

  - **Automatic** (自動) 更新を選択した場合、新しいバージョンの Fuse Console Operator が使用できるようになると、人的な介入なしで OpenShfit Operator Lifecycle Manager (OLM) によって、Fuse Console の稼働中のインスタンスが自動的にアップグレードされます。

  - **Manual** (手動) 更新を選択した場合、Operator の新しいバージョンが使用できるようになると、OLM によって更新リクエストが作成されます。クラスター管理者は、更新リクエストを手動で承認して、Fuse Console Operator を新しいバージョンに更新する必要があります。

6. **Install** をクリックします。
   OpenShift によって、Fuse Console Operator が現在の namespace にインストールされます。

7. インストールを確認するには、**Operators** をクリックした後、**Installed Operators** をクリックします。Operator のリストに Fuse Console が表示されます。

8. OpenShift Web コンソールで Fuse Console をデプロイするには、以下を行います。

   a. **Installed Operators** のリストで、**Name** 列の **Fuse Console** をクリックします。

   b. **Provided APIs** の **Operator Details** ページで、**Create Instance** をクリックします。
   設定のデフォルト値を使用するか、任意でデフォルト値を編集します。

   Fuse Console のパフォーマンスを向上する必要がある場合 (高可用性環境など)、**Replicas** で Fuse Console に割り当てられた Pod 数を増やすことができます。

   **Rbac** (ロールベースアクセス制御) では、デフォルトの RBAC 動作をカスタマイズする場合や、Fuse Console Operator をインストールした namespace に ConfigMap ファイルがすでに存在する場合にのみ、**config Map** フィールドに値を指定します。RBAC の詳細は、OpenShift 4.x 上の Fuse Console のロールベースアクセス制御 を参照してください。

   **Nginx** については、Fuse Console Operator インストールのパフォーマンスチューニング を参照してください。

   c. **Create** をクリックします。
   **Fuse Console Operator Details** ページが開き、デプロイメントの状態が表示されます。

9. Fuse Console を開くには以下を行います。

   a. **namespace** デプロイメントの場合: OpenShift Web コンソールで、Fuse Console の operator をインストールしたプロジェクトを開き、**Overview** を選択します。**Project Overview** ページで **Launcher** セクションまで下方向にスクロールし、Fuse Console のリンクをクリックします。

**cluster** デプロイメントでは、OpenShift Web コンソールのタイトルバーで、グリッドアイコン (  ) をクリックします。ポップアップメニューの **Red Hat アプリケーション** の下にある、Fuse Console の URL リンクをクリックします。

b. Fuse Console にログインします。
ブラウザーに **Authorize Access** ページが表示され、必要なパーミッションが表示されます。

c. **Allow selected permissions** をクリックします。
Fuse Console がブラウザーで開き、アクセス権限のある Fuse アプリケーション Pod が表示されます。

10. 表示するアプリケーションの **Connect** をクリックします。
新しいブラウザーウィンドウが開かれ、Fuse Console にアプリケーションが表示されます。

## 2.2. コマンドラインを使用した OPENSHIFT 4.X での FUSE CONSOLE のインストールおよびデプロイ

OpenShift 4.x では、次のデプロイメントオプションの 1 つを選択して、コマンドラインから Fuse Console をインストールおよびデプロイできます。

- **cluster** - Fuse Console は、OpenShift クラスターで複数の namespace (プロジェクト) 全体にデプロイされた Fuse アプリケーションを検索し、接続することができます。このテンプレートをデプロイするには、OpenShift クラスターの管理者ロールが必要です。
- **ロールベースアクセス制御のあるクラスター** - 設定可能なロールベースアクセス制御 (RBAC) のあるクラスターテンプレート。詳細は、OpenShift 4.x 上の Fuse Console のロールベースアクセス制御 を参照してください。
- **namespace** - Fuse Console は特定の OpenShift プロジェクト (namespace) にアクセスできます。このテンプレートをデプロイするには、OpenShift プロジェクトの管理者ロールが必要です。
- **ロールベースアクセス制御のある namespace** - 設定可能な RBAC のある namespace テンプレート。詳細は、OpenShift 4.x 上の Fuse Console のロールベースアクセス制御 を参照してください。

Fuse Console テンプレートのパラメーターリストを表示するには、以下の OpenShift コマンドを実行します。

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-7_11_1-00016-redhat-00002/fuse-console-namespace-os4.json
```

**前提条件**

- Fuse Console をインストールおよびデプロイする前に、OpenShift 4.x で Fuse Console をセキュア化するための証明書の生成 で説明されているとおりに、サービス署名認証局で署名されたクライアント証明書を生成している。
- OpenShift クラスターの **cluster admin** ロールがある。
- コンテナーイメージの **registry.redhat.io** を使用した認証 で説明されているように、**registry.redhat.io** での認証を設定している。

- OpenShift 4.x サーバーでの Fuse イメージストリームおよびテンプレートのインストール の説明どおりに Fuse Console イメージストリーム (およびその他の Fuse イメージストリーム) がインストールされている。

**手順**

1. 以下のコマンドを実行してすべてのテンプレートのリストを取得し、Fuse Console イメージストリームがインストールされていることを確認します。

```
oc get template -n openshift
```

2. 任意で、すでにインストールされているイメージストリームを新しいリリースタグで更新する場合は、以下のコマンドを使用して **openshift** namespace に Fuse Console イメージをインポートします。

```
oc import-image fuse7/fuse-console-rhel8:1.10 --from=registry.redhat.io/fuse7/fuse-console-rhel8:1.10 --confirm -n openshift
```

3. 以下のコマンドを実行して、Fuse Console の **APP_NAME** の値を取得します。

```
oc process --parameters -f TEMPLATE-FILENAME
```

ここで、**TEMPLATE-FILENAME** は以下のテンプレートのいずれかになります。

- クラスターテンプレート:

```
`https://github.com/jboss-fuse/application-templates/blob/application-templates-2.1.0.fuse-sb2-7_11_1-00016-redhat-00002//fuse-console-cluster-os4.json`
```

- 設定可能な RBAC のあるクラスターテンプレート:

```
`https://github.com/jboss-fuse/application-templates/blob/application-templates-2.1.0.fuse-sb2-7_11_1-00016-redhat-00002//fuse-console-cluster-rbac.yml`
```

- namespace テンプレート:

```
`https://github.com/jboss-fuse/application-templates/blob/application-templates-2.1.0.fuse-sb2-7_11_1-00016-redhat-00002//fuse-console-namespace-os4.json`
```

- 設定可能な RBAC のある namespace テンプレート:

```
`https://github.com/jboss-fuse/application-templates/blob/application-templates-2.1.0.fuse-sb2-7_11_1-00016-redhat-00002//fuse-console-namespace-rbac.yml`
```

たとえば、設定可能な RBAC のあるクラスターテンプレートの場合は、以下のコマンドを実行します。

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-7_11_1-00016-redhat-00002/fuse-console-cluster-rbac.yml
```

4. 以下のコマンドを使用して、OpenShift 4.x での Fuse Console のセキュア化 で生成した証明書からシークレットを作成し、Fuse Console にマウントします。コマンドの **APP_NAME** は、Fuse Console アプリケーションの名前に置き換えます。

```
oc create secret tls APP_NAME-tls-proxying --cert server.crt --key server.key
```

5. 以下のコマンドを実行して、ローカルにコピーされた Fuse Console テンプレートをベースとした新しいアプリケーションを作成します。コマンドの **myproject** は OpenShift プロジェクトの名前、**mytemp** は Fuse Console テンプレートが含まれるローカルディレクトリーへのパス、**myhost** は Fuse Console にアクセスするホストの名前に置き換えます。

- クラスターテンプレートの場合:

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-7_11_1-00016-redhat-00002/fuse-console-cluster-os4.json  -p ROUTE_HOSTNAME=myhost
```

- RBAC テンプレートのあるクラスターの場合:

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-7_11_1-00016-redhat-00002/fuse-console-cluster-rbac.yml -p ROUTE_HOSTNAME=myhost
```

- Namespace テンプレートの場合:

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-7_11_1-00016-redhat-00002/fuse-console-namespace-os4.json
```

- RBAC テンプレートのある namespace の場合:

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-7_11_1-00016-redhat-00002/fuse-console-namespace-rbac.yml
```

6. OpenShift Web コンソールを開くように Fuse Console を設定するには、以下のコマンドを実行して **OPENSHIFT_WEB_CONSOLE_URL** 環境変数を設定します。

```
oc set env dc/${APP_NAME} OPENSHIFT_WEB_CONSOLE_URL=`oc get -n openshift-config-managed cm console-public -o jsonpath={.data.consoleURL}`
```

7. 以下のコマンドを実行して、Fuse Console デプロイメントの状態と URL を取得します。

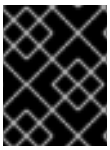
```
oc status
```

8. ブラウザーから Fuse Console にアクセスするには、手順 7. で返される URL (例: https://fuse-console.192.168.64.12.nip.io) を使用します。

### 2.2.1. OpenShift 4.x で Fuse Console をセキュア化するための証明書の生成

OpenShift 4.x で、Fuse Console プロキシーと Jolokia エージェントとの間の接続をセキュアにするには、Fuse Console をデプロイする前にクライアント証明書を生成する必要があります。サービス署名認証局の秘密鍵を使用して、クライアント証明書を署名する必要があります。

この手順は、コマンドラインを使用して Fuse Console をインストールおよびデプロイしている場合に**のみ** 実行する必要があります。Fuse Console Operator を使用している場合は、Fuse Console Operator がこのタスクを処理します。



**重要**

各 OpenShift クラスターに別のクライアント証明書を生成し、署名する必要があります。複数のクラスターに同じ証明書を使用しないでください。

**前提条件**

- OpenShift クラスターにアクセス可能な **cluster admin** 権限がある。
- 複数の OpenShift クラスターの証明書を生成し、現在のディレクトリーに別のクラスターの証明書を生成している場合は、以下のいずれかを実行して、現在のクラスターに別の証明書を生成するようにします。
  - 現在のディレクトリーから既存の証明書ファイル (**ca.crt**、**ca.key**、および **ca.srl**) を削除します。
  - 別の作業ディレクトリーに移動します。たとえば、現在の作業ディレクトリーの名前が **cluster1** の場合、新しい **cluster2** ディレクトリーを作成し、作業ディレクトリーをそのディレクトリーに変更します。
    **mkdir ../cluster2**

    **cd ../cluster2**

**手順**

1. cluster admin 権限を持つユーザーとして OpenShift にログインします。

```
oc login -u <user_with_cluster_admin_role>
```

2. 以下のコマンドを実行して、サービス署名認証局のキーを取得します。
   - 以下を実行して、証明書を取得します。

```
oc get secrets/signing-key -n openshift-service-ca -o "jsonpath={.data['tls\.crt']}" | base64 --decode > ca.crt
```

   - 以下を実行して、秘密鍵を取得します。

```
oc get secrets/signing-key -n openshift-service-ca -o "jsonpath={.data['tls\.key']}" | base64 --decode > ca.key
```

3. Kubernetes certificates administration の説明にしたがって、**easyrsa**、**openssl**、または **cfssl** を使用して、クライアント証明書を生成します。
   以下に、openssl を使用したコマンドの例を示します。
   a. 秘密鍵を生成します。

```
openssl genrsa -out server.key 2048
```

   b. CSR 設定ファイルを作成します。

```
cat <<EOT >> csr.conf
 [ req ]
 default_bits = 2048
 prompt = no
 default_md = sha256
 distinguished_name = dn

 [ dn ]
 CN = fuse-console.fuse.svc

 [ v3_ext ]
 authorityKeyIdentifier=keyid,issuer:always
 keyUsage=keyEncipherment,dataEncipherment,digitalSignature
 extendedKeyUsage=serverAuth,clientAuth
EOT
```

ここでは、**CN** パラメーターの値はアプリケーション名とアプリケーションが使用する namespace を参照します。

c. CSR を生成します。

```
openssl req -new -key server.key -out server.csr -config csr.conf
```

d. 署名済みの証明書を発行します。

```
openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt
-days 10000 -extensions v3_ext -extfile csr.conf
```

**次のステップ**

コマンドラインを使用した OpenShift 4.x での Fuse Console のインストールおよびデプロイ で説明されているように、Fuse Console のシークレットを作成するには、この証明書が必要です。

## 2.3. OPENSHIFT 4.X 上の FUSE CONSOLE のロールベースアクセス制御

Fuse Console は、OpenShift によって提供されるユーザー承認に応じてアクセスを推測する、ロールベースアクセス制御 (RBAC) を提供します。Fuse Console では、RBAC はユーザーが Pod で MBean 操作を実行できるかどうかを判断します。

OpenShift の承認に関する詳細は、OpenShift ドキュメントの RBAC の使用によるパーミッションの定義および適用 を参照してください。

Operator を使用して OpenShift に Fuse Console をインストールした場合、ロールベースのアクセスはデフォルトで有効になります。

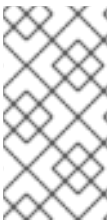
テンプレートを使用したインストールによって Fuse Console のロールベースアクセスを実装する場合は、RBAC で設定可能なテンプレートの 1 つ (**fuse-console-cluster-rbac.yml** または **fuse-console-namespace-rbac.yml**) を使用して、コマンドラインを使用した OpenShift 4.x での Fuse Console のインストールおよびデプロイ の説明どおりに Fuse Console をインストールする必要があります。

Fuse Console RBAC は、OpenShift の Pod リソースでユーザーの **verb (動詞)** アクセスを利用して、Fuse Console の Pod の MBean 操作にユーザーがアクセスできるかどうかを判断します。デフォルトでは、Fuse Console には 2 つのユーザーロールがあります。

- **admin**

ユーザーが OpenShift で Pod を **更新** できる場合、ユーザーには Fuse Console の **admin** ロールが付与されます。ユーザーは、Fuse Console で、その Pod に対して **書き込み** の MBean 操作を実行できます。

- **viewer**
ユーザーが OpenShift で Pod を **取得** できる場合、ユーザーには Fuse Console の **viewer** ロールが付与されます。ユーザーは、Fuse Console で、その Pod に対して **読み取り専用** の MBean 操作を実行できます。



**注記**

RBAC 以外のテンプレートを使用して Fuse Console をインストールした場合、Pod リソースで **update** verb (動詞) が付与された OpenShift ユーザーのみが Fuse Console の MBeans 操作の実行が許可されます。Pod リソースで **get** verb (動詞) を付与されたユーザーは、Pod を **表示** できますが、Fuse Console の操作は実行できません。

**関連情報**

- OpenShift 4.x での Fuse Console のアクセスロールの判断
- OpenShift 4.x での Fuse Console へのロールベースアクセスのカスタマイズ
- OpenShift 4.x での Fuse Console のロールベースアクセス制御の無効化

### 2.3.1. OpenShift 4.x での Fuse Console のアクセスロールの判断

Fuse Console のロールベースアクセス制御は、Pod に対するユーザーの OpenShift パーミッションから推測されます。特定のユーザーに付与された Fuse Console のアクセスロールを判断するには、Pod に対してユーザーに付与された OpenShift パーミッションを取得します。

**前提条件**

- ユーザーの名前を知っている必要があります。
- Pod の名前を知っている必要があります。

**手順**

- ユーザーが Pod に対して Fuse Console の admin ロールを持っているかどうかを確認するには、以下のコマンドを実行してユーザーが OpenShift で Pod を更新できるかどうかを確認します。

```
oc auth can-i update pods/<pod> --as <user>
```

応答が **yes** の場合、ユーザーはその Pod に対して Fuse Console の **admin** ロールを持っています。ユーザーは、Fuse Console で、その Pod に対して **書き込み** の MBean 操作を実行できます。

- ユーザーが Pod に対して Fuse Console の **viewer** ロールを持っているかどうかを確認するには、以下のコマンドを実行してユーザーが OpenShift で Pod を**取得**できるかどうかを確認します。

```
oc auth can-i get pods/<pod> --as <user>
```

応答が **yes** の場合、ユーザーはその Pod に対して Fuse Console の **viewer** ロールを持ってい

ます。ユーザーは、Fuse Console で、その Pod に対して **読み取り専用** の MBean 操作を実行できます。コンテキストによっては、Fuse Console でオプションを無効にしたり、ユーザーによる **書き込み** MBean 操作の試行時にこのユーザーの操作は許可されないという内容のメッセージを表示したりして、**viewer** ロールを持つユーザーによる **書き込み** MBean 操作が実行されないようにします。

応答が **no** の場合、ユーザーは Fuse Console のロールにバインドされず、ユーザーは Fuse Console で Pod を確認できません。

**関連情報**

- OpenShift 4.x 上の Fuse Console のロールベースアクセス制御
- OpenShift 4.x での Fuse Console へのロールベースアクセスのカスタマイズ
- OpenShift 4.x での Fuse Console のロールベースアクセス制御の無効化

### 2.3.2. OpenShift 4.x での Fuse Console へのロールベースアクセスのカスタマイズ

OperatorHub を使用して Fuse Console をインストールした場合、OpenShift 4.x 上の Fuse Console のロールベースアクセス制御 で説明されているように、ロールベースアクセス制御 (RBAC) はデフォルトで有効になります。Fuse Console の RBAC 動作をカスタマイズする場合、Fuse Console をデプロイする前に ConfigMap ファイル (カスタムの RBAC 動作を定義する) を指定する必要があります。カスタム ConfigMap ファイルは、Fuse Console Operator をインストールした namespace に配置する必要があります。

コマンドラインテンプレートを使用して Fuse Console をインストールする場合、**deployment-cluster-rbac.yml** および **deployment-namespace-rbac.yml** テンプレートによって、設定ファイル (**ACL.yml**) が含まれる ConfigMap が作成されます。設定ファイルでは、MBean 操作に許可されるロールが定義されます。

**前提条件**

- OperatorHub または Fuse Console の RBAC テンプレート (**deployment-cluster-rbac.yml** または **deployment-namespace-rbac.yml**) の 1 つを使用して、Fuse Console がインストール済みである必要があります。

**手順**

Fuse Console の RBAC ロールをカスタマイズする場合は、以下を行います。

1. コマンドラインを使用して Fuse Console をインストールした場合、インストールテンプレートにはデフォルトの ConfigMap ファイルが含まれるため、次のステップを省略できます。
   OperatorHub を使用して Fuse Console をインストールした場合、Fuse Console をデプロイする前に RBAC ConfigMap を作成します。

   a. 現在の OpenShift プロジェクトが Fuse Console をインストールするプロジェクトであることを確認します。たとえば、Fuse Console を **fusetest** プロジェクトにインストールするには、以下のコマンドを実行します。

   ```
   oc project fusetest
   ```

   b. テンプレートから Fuse Console RBAC ConfigMap ファイルを作成するには、以下のコマンドを実行します。

```
oc process -f https://raw.githubusercontent.com/jboss-fuse/application-templates/2.1.x.sb2.redhat-7-8-x/fuse-console-operator-rbac.yml -p APP_NAME=fuse-console | oc create -f -
```

2. 以下のコマンドを実行してエディターで ConfigMap を開きます。

```
oc edit cm $APP_NAME-rbac
```

以下に例を示します。

```
oc edit cm fuse-console-rbac
```

3. ファイルを編集します。
4. ファイルを保存して変更を適用します。Fuse Console の Pod は OpenShift によって自動的に再起動されます。

**関連情報**

- OpenShift 4.x 上の Fuse Console のロールベースアクセス制御
- OpenShift 4.x での Fuse Console のアクセスロールの判断
- OpenShift 4.x での Fuse Console のロールベースアクセス制御の無効化

### 2.3.3. OpenShift 4.x での Fuse Console のロールベースアクセス制御の無効化

コマンドラインを使用して Fuse Console をインストール し、Fuse Console の RBAC テンプレートのいずれかを指定した場合、Fuse Console の **HAWTIO_ONLINE_RBAC_ACL** 環境変数は、ロールベースアクセス制御 (RBAC) の ConfigMap 設定ファイルのパスを OpenShift サーバーに渡します。**HAWTIO_ONLINE_RBAC_ACL** 環境変数が指定されていない場合、RBAC のサポートは無効になり、Pod リソース (OpenShift の) で **update** verb (動詞) が付与されたユーザーのみが Fuse Console の Pod で MBean 操作を呼び出すことが承認されます。

OperatorHub を使用して Fuse Console をインストールする場合は、ロールベースのアクセスはデフォルトで有効になり、**HAWTIO_ONLINE_RBAC_ACL** 環境変数は適用されません。

**前提条件**

コマンドラインを使用して Fuse Console がインストール済みで、Fuse Console の RBAC テンプレート (**deployment-cluster-rbac.yml** または **deployment-namespace-rbac.yml**) のいずれかを指定している。

**手順**

Fuse Console のロールベースのアクセスを無効にするには、以下を実行します。

1. OpenShift で、Fuse Console の **Deployment Config** リソースを編集します。
2. **HAWTIO_ONLINE_RBAC_ACL** 環境変数の定義をすべて削除します。
(値を消去するだけでは不十分です)。
3. ファイルを保存して変更を適用します。Fuse Console の Pod は OpenShift によって自動的に再起動されます。

**関連情報**

**関連情報**

- OpenShift 4.x 上の Fuse Console のロールベースアクセス制御
- OpenShift 4.x での Fuse Console のアクセスロールの判断
- OpenShift 4.x での Fuse Console へのロールベースアクセスのカスタマイズ

## 2.4. OPENSHIFT 4.X での FUSE CONSOLE のアップグレード

Red Hat OpenShift 4.x では、Red Hat Fuse Operator などの Operator の更新が処理されます。詳細は、OpenShift ドキュメントの Operator を参照してください。

その後、アプリケーションの設定方法によっては、Operator の更新でアプリケーションのアップグレードをトリガーできるようになります。

Fuse Console アプリケーションでは、アプリケーションカスタムリソース定義の **.spec.version** フィールドを編集して、アプリケーションのアップグレードをトリガーすることもできます。

### 前提条件

- OpenShift クラスターの管理者権限が必要です。

### 手順

Fuse Console アプリケーションをアップグレードするには、以下を行います。

1. ターミナルウィンドウで、以下のコマンドを使用してアプリケーションカスタムリソース定義の **.spec.version** フィールドを変更します。

```
oc patch -n <project-name> <custom-resource-name> --type='merge' -p '{"spec":{"version":"1.7.1"}}'
```

以下に例を示します。

```
oc patch -n myproject hawtio/example-fuseconsole --type='merge' -p '{"spec":{"version":"1.7.1"}}'
```

2. アプリケーションの状態が更新されたことを確認します。

```
oc get -n myproject hawtio/example-fuseconsole
```

応答には、バージョン番号などのアプリケーションに関する情報が表示されます。

```
NAME                  AGE   URL                                    IMAGE
example-fuseconsole   1m    https://fuseconsole.192.168.64.38.nip.io
docker.io/fuseconsole/online:1.7.1
```

**.spec.version** フィールドの値を変更すると、OpenShift によってアプリケーションが自動的に再デプロイされます。

3. バージョンの変更によってトリガーされた再デプロイの状態をチェックするには、以下を実行します。

```
oc rollout status deployment.v1.apps/example-fuseconsole
```

正常にデプロイされた場合は以下の応答が表示されます。

```
deployment "example-fuseconsole" successfully rolled out
```

## 2.5. OPENSHIFT 4.X サーバーでの FUSE イメージストリームおよびテンプレートのアップグレード

OpenShift Container Platform 4.x は、OpenShift namespace で操作する Samples Operator を使用して、Red Hat Enterprise Linux (RHEL) ベースの OpenShift Container Platform イメージストリームおよびテンプレートをアップグレードおよび更新します。

Fuse on OpenShift イメージストリームおよびテンプレートをアップグレードするには以下を実行します。

- Samples Operator の再設定
- Fuse イメージストリームおよびテンプレートを **Skipped Imagestreams and Skipped Templates** フィールドに追加します。
  - Skipped Imagestreams (スキップされるイメージストリーム): Samples Operator のインベントリーにあるが、クラスター管理者の希望により Operator が無視または管理しないようにするイメージストリーム。
  - Skipped Templates (スキップされるテンプレート): Samples Operator のインベントリーにあるが、クラスター管理者の希望により Operator が無視または管理しないようにするテンプレート。

**前提条件**

- OpenShift サーバーにアクセスできる必要があります。
- **registry.redhat.io** への認証が設定されている。

**手順**

1. OpenShift 4 サーバーを起動します。
2. OpenShift サーバーに管理者としてログインします。

```
oc login --user system:admin --token=my-token --server=https://my-cluster.example.com:6443
```

3. docker-registry シークレットを作成したプロジェクトを使用していることを確認します。

```
oc project openshift
```

4. Samples Operator の現在の設定を表示します。

```
oc get configs.samples.operator.openshift.io -n openshift-cluster-samples-operator -o yaml
```

5. Samples Operator が、追加された Fuse テンプレートおよびイメージストリームを無視するよう設定します。

```
oc edit configs.samples.operator.openshift.io -n openshift-cluster-samples-operator
```

6. Fuse imagestreams の Skipped Imagestreams セクションを追加し、Fuse および Spring Boot 2 のテンプレートを Skipped Templates セクションに追加します。

```
[...]
spec:
  architectures:
  - x86_64
  managementState: Managed
  skippedImagestreams:
  - fuse-console-rhel8
  - fuse-eap-openshift-jdk8-rhel7
  - fuse-eap-openshift-jdk11-rhel8
  - fuse-java-openshift-rhel8
  - fuse-java-openshift-jdk11-rhel8
  - fuse-karaf-openshift-rhel8
  - fuse-karaf-openshift-jdk11-rhel8
  - fuse-apicurito-generator-rhel8
  - fuse-apicurito-rhel8
  skippedTemplates:
  - s2i-fuse711-eap-camel-amq
  - s2i-fuse711-eap-camel-cdi
  - s2i-fuse711-eap-camel-cxf-jaxrs
  - s2i-fuse711-eap-camel-cxf-jaxws
  - s2i-fuse711-karaf-camel-amq
  - s2i-fuse711-karaf-camel-log
  - s2i-fuse711-karaf-camel-rest-sql
  - s2i-fuse711-karaf-cxf-rest
  - s2i-fuse711-spring-boot-2-camel-amq
  - s2i-fuse711-spring-boot-2-camel-config
  - s2i-fuse711-spring-boot-2-camel-drools
  - s2i-fuse711-spring-boot-2-camel-infinispan
  - s2i-fuse711-spring-boot-2-camel-rest-3scale
  - s2i-fuse711-spring-boot-2-camel-rest-sql
  - s2i-fuse711-spring-boot-2-camel
  - s2i-fuse711-spring-boot-2-camel-xa
  - s2i-fuse711-spring-boot-2-camel-xml
  - s2i-fuse711-spring-boot-2-cxf-jaxrs
  - s2i-fuse711-spring-boot-2-cxf-jaxws
  - s2i-fuse711-spring-boot-2-cxf-jaxrs-xml
  - s2i-fuse711-spring-boot-2-cxf-jaxws-xml
```

7. Fuse on OpenShift イメージストリームをアップグレードします。

```
BASEURL=https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-7_11_1-00016-redhat-00002

oc replace -n openshift -f ${BASEURL}/fis-image-streams.json
```

8. Fuse on OpenShift クイックスタートテンプレートをアップグレードします。

```
for template in eap-camel-amq-template.json \
 eap-camel-cdi-template.json \
 eap-camel-cxf-jaxrs-template.json \
```

```
eap-camel-cxf-jaxws-template.json \
karaf-camel-amq-template.json \
karaf-camel-log-template.json \
karaf-camel-rest-sql-template.json \
karaf-cxf-rest-template.json ;
do
oc replace -n openshift \
${BASEURL}/quickstarts/${template}
done
```

9. Spring Boot 2 クイックスタートテンプレートをアップグレードします。

```
for template in spring-boot-2-camel-amq-template.json \
 spring-boot-2-camel-config-template.json \
 spring-boot-2-camel-drools-template.json \
 spring-boot-2-camel-infinispan-template.json \
 spring-boot-2-camel-rest-3scale-template.json \
 spring-boot-2-camel-rest-sql-template.json \
 spring-boot-2-camel-template.json \
 spring-boot-2-camel-xa-template.json \
 spring-boot-2-camel-xml-template.json \
 spring-boot-2-cxf-jaxrs-template.json \
 spring-boot-2-cxf-jaxws-template.json \
 spring-boot-2-cxf-jaxrs-xml-template.json \
 spring-boot-2-cxf-jaxws-xml-template.json ;
 do oc replace -n openshift \
 ${BASEURL}/quickstarts/${template}
 done
```

10. **(オプション)** アップグレードされた Fuse on OpenShift テンプレートを表示します。

```
oc get template -n openshift
```

## 2.6. OPENSHIFT 4.X での FUSE CONSOLE のパフォーマンスチューニング

デフォルトでは、Fuse Console は以下の Nginx 設定を使用します。

- **clientBodyBufferSize: 256k**
- **proxyBuffers: 16 128k**
- **subrequestOutputBufferSize: 10m**

**注記:** これらの設定の説明は、Nginx のドキュメント (http://nginx.org/en/docs/dirindex.html) を参照してください。

Fuse Console のパフォーマンスを調整するには、**clientBodyBufferSize**、**proxyBuffers**、および **subrequestOutputBufferSize** 環境変数を設定します。たとえば、Fuse Console を使用して多数の Pod およびルート (合計 100 のルートなど) を監視する場合、Fuse Console の **subrequestOutputBufferSize** 環境変数を **60m** から **100m** の間で設定することで、ロードタイムアウトの問題を解決することができます 。

これらの環境変数の設定方法は、OpenShift 4.x に Fuse Console をインストールする方法によって異なります。

- Fuse Console Operator の使用
- Fuse Console テンプレートの使用

### 2.6.1. Fuse Console Operator インストールのパフォーマンスチューニング

Openshift 4.x では、Fuse Console のデプロイ前後に Nginx パフォーマンスチューニング環境変数を設定できます。これを後で行うと、OpenShift は Fuse Console を再デプロイします。

**前提条件**

- OpenShift クラスターにアクセス可能な **cluster admin** 権限がある。
- OperatorHub を使用した OpenShift 4.x での Fuse Console のインストールおよびデプロイ の説明どおりに、Fuse Console Operator がインストールされている必要があります。

**手順**

Fuse Console のデプロイ前後に環境変数を設定できます。

- **Fuse Console をデプロイする前に環境変数を設定するには、以下を実行します。**
  1. OpenShift Web コンソールから、Fuse Console Operator がインストールされているプロジェクトで、**Operators**> **Installed Operators**> **Red Hat Integration - Fuse Console** を選択します。
  2. **Hawtio** タブをクリックし、**Create Hawtio** をクリックします。
  3. **Create Hawtio** ページで、**Form ビュー** で **Config**> **Nginx** セクションまでスクロールダウンします。
  4. **Nginx** セクションをデプロイメントしてから、環境変数を設定します。以下に例を示します。
     - clientBodyBufferSize: **256k**
     - proxyBuffers: **16 128k**
     - subrequestOutputBufferSize: **100m**
  5. 設定を保存します。
  6. **Create** をクリックして Fuse Console をデプロイします。
  7. デプロイメントが完了したら、**Deployments**> **fuse-console** ページを開いて **Environment** をクリックし、環境変数がリストにあることを確認します。
- **Fuse Console のデプロイ後に環境変数を設定するには、以下を実行します。**
  1. OpenShift Web コンソールで、Fuse Console がデプロイされているプロジェクトを開きます。
  2. **Operators**> **Installed Operators**> **Red Hat Integration - Fuse Console** の順に選択します。

3. **Hawtio** タブをクリックし、**fuse-console** をクリックします。
4. **Actions**> **Edit Hawtio** を選択します。
5. Editor ウィンドウで、**spec** セクションまでスクロールダウンします。
6. 以下のように、**spec** セクションで、新規の **nginx** セクションを追加し、1 つ以上の環境変数を指定します。

```
apiVersion: hawt.io/v1alpha1
kind: Hawtio
metadata:
  name: fuse-console
spec:
  type: Namespace
  nginx:
    clientBodyBufferSize: 256k
    proxyBuffers: 16 128k
    subrequestOutputBufferSize: 100m
.
.
.
```

7. **Save** をクリックします。
   OpenShift は Fuse Console を再デプロイします。
8. 再デプロイメントが完了したら、**Workloads**> **Deployments**> **fuse-console** ページを開いてから **Environment** をクリックし、リストの環境変数を確認します。

### 2.6.2. Fuse Console テンプレートインストールのパフォーマンスチューニング

Openshift 4.x では、Fuse Console のデプロイ前後に Nginx パフォーマンスチューニング環境変数を設定できます。これを後で行うと、OpenShift は Fuse Console を再デプロイします。

**前提条件**

- OpenShift クラスターにアクセス可能な **cluster admin** 権限がある。
- OpenShift 4.x サーバーでの Fuse イメージストリームおよびテンプレートのインストール の説明どおりに、OpenShift に Fuse Console テンプレートがインストールされている必要があります。

**手順**

Fuse Console のデプロイ前後に環境変数を設定できます。

- **Fuse Console をデプロイする前に環境変数を設定するには、以下を実行します。**
  1. 使用する Fuse Console テンプレートを確認します。
     - クラスターテンプレート (**fuse-console-cluster-os4.json**)
     - 設定可能な RBAC が含まれるクラスターテンプレート (**fuse-console-cluster-rbac.yml**)
     - namespace テンプレート (**fuse-console-namespace-os4.json**)

- 設定可能な RBAC が含まれる namespace テンプレート (**fuse-console-namespace-rbac.yml**)

2. 以下の例のように、Fuse Console に使用する Fuse Console テンプレートのローカルコピーを編集し、**NGINX_CLIENT_BODY_BUFFER_SIZE**、**NGINX_PROXY_BUFFERS**、および/または **NGINX_SUBREQUEST_OUTPUT_BUFFER_SIZE** の環境変数を追加します。

```
apiVersion: apps.openshift.io/v1
kind: DeploymentConfig
metadata:
  name: fuse-console
spec:
  template:
    spec:
      containers:
      - env:
        - name: NGINX_CLIENT_BODY_BUFFER_SIZE
          value: 256k
        - name: NGINX_PROXY_BUFFERS
          value: 16 128k
        - name: NGINX_SUBREQUEST_OUTPUT_BUFFER_SIZE
          value: 100m
```

3. 変更を保存します。

4. OpenShift 4.x での Fuse Console の設定 の説明にあるように、Fuse Console をインストールおよびデプロイする手順にしたがいます。

- **Fuse Console のデプロイ後に環境変数を設定するには、以下を実行します。**

1. ターミナルウィンドウで、OpenShift クラスターにログインします。

2. Fuse Console がデプロイされているプロジェクトを開きます。たとえば、Fuse Console が **myfuse** プロジェクトにデプロイされている場合は、以下のコマンドを使用します。
**oc project myfuse**

3. Fuse Console デプロイメントの名前を取得します。
**oc get deployments**

このコマンドは、現在のプロジェクトで実行しているデプロイメントのリストを返します。以下に例を示します。

```
NAME                READY   UP-TO-DATE   AVAILABLE    AGE
fuse-console         1/1     1            1           114m
```

4. 以下のいずれかのコマンドを実行して、Fuse Console デプロイメントの環境変数を設定します。

```
oc set env dc/fuse-console NGINX_CLIENT_BODY_BUFFER_SIZE="256k"

oc set env dc/fuse-console NGINX_PROXY_BUFFERS="16 128k"

oc set env dc/fuse-console NGINX_SUBREQUEST_OUTPUT_BUFFER_SIZE="10m"
```

OpenShift は Fuse Console を再デプロイします。

5. 再デプロイメントが完了したら、環境変数の設定を確認します。

   a. Fuse Console の Pod 名を取得します。

   ```
   oc get pods
   ```

   b. 次のコマンドを実行して、環境設定を表示します。

   ```
   oc exec <fuse-console-podname> -- cat /opt/app-root/etc/nginx.d/nginx-gateway.conf | grep "Performance tuning" -A 3
   ```

   たとえば、Pod 名が **fuse-console-6646cbbd4c-9rplg** である場合は、以下のコマンドを実行します。

   ```
   oc exec fuse-console-6646cbbd4c-9rplg -- cat /opt/app-root/etc/nginx.d/nginx-gateway.conf | grep "Performance tuning" -A 3
   ```

### 2.6.3. Fuse Console でアプリケーションを表示するためのパフォーマンスチューニング

Fuse console の強化されたパフォーマンスチューニング機能により、多数の MBean を持つアプリケーションを表示できます。この機能を使用するには、次の手順を実行します。

**前提条件**

- OpenShift クラスターにアクセス可能な **cluster admin** 権限がある。
- OperatorHub を使用した OpenShift 4.x での Fuse Console のインストールおよびデプロイ の説明どおりに、Fuse Console Operator がインストールされている必要があります。

**手順**

1. アプリケーションのメモリー制限を増やします。
   Fuse コンソールに到達する前にアプリケーションが OOM エラーでクラッシュしないように、メモリー制限を (たとえば 256Mi から 512 Mi に) 増やす必要があります。Fuse クイックスタートの場合、アプリケーションの **src/main/jkube/deployment.yml** ファイルを編集します。

   ```
   spec:
     template:
       spec:
         containers:
           -
             resources:
               [...]
               limits:
                 cpu: "1.0"
                 memory: 512Mi
   ```

2. **Fuse Console Deployment** または **DeploymentConfig** に十分なメモリー制限があることを確認してください。不十分な場合は、たとえば 200Mi から 512Mi に制限を増やします。

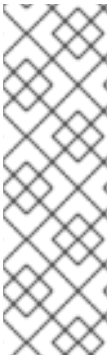
3. nginx ログにクライアントへの送信中に「too big subrequest response while sending to client」エラーが表示される場合は、「Fuse Console Operator インストールのパフォーマンスチューニング」 セクションに記載される解決策を適用します。

# 第3章 OPENSHIFT 3.11 での FUSE CONSOLE の設定

OpenShift 3.11 では、以下のように Fuse Console にアクセスできます。

- プロジェクトで実行しているすべての Fuse コンテナーを監視できるように、Fuse Console を OpenShift プロジェクトに追加する。
- クラスター上のすべてのプロジェクトで稼働中のすべての Fuse コンテナーを監視できるように、Fuse Console を OpenShift クラスターに追加する。
- 実行している単一の Fuse コンテナーを監視できるように、特定の Fuse Pod から Fuse Console を開く。

コマンドラインから Fuse Console テンプレートをデプロイします。



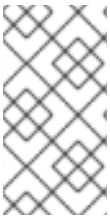
**注記**

Minishift または CDK ベースの環境変数に Fuse Console をインストールするには、以下の KCS の記事で説明されている手順に従います。

- Minishift または CDK ベースの環境に Fuse Console をインストールするには、KCS 4998441 を参照してください。
- Jolokia 認証を無効にする必要がある場合は、KCS 3988671 で説明されている回避策を参照してください。

**前提条件**

- Fuse on OpenShift ガイド の説明にしたがって、Fuse Console の Fuse on OpenShift イメージストリームおよびテンプレートをインストールしている。



**注記**

- Fuse Console のユーザー管理は、OpenShift によって処理されます。
- ロールベースアクセス制御 (デプロイ後に Fuse Console にアクセスするユーザーの場合) は現在 Fuse on OpenShift 3.11 では使用できません。

「OpenShift 3.11 での Fuse Console のデプロイ」

「OpenShift 3.11 の Fuse Console から単一の Fuse Pod を監視」

## 3.1. OPENSHIFT 3.11 での FUSE CONSOLE のデプロイ

表3.1「Fuse Console テンプレート」は、Fuse アプリケーションのデプロイメントのタイプに応じて、コマンドラインから Fuse Console をデプロイするために使用する OpenShift 3.11 テンプレートについて説明しています。

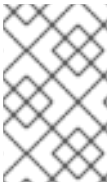
**表3.1 Fuse Console テンプレート**

| タイプ | 説明 |
|---|---|

| タイプ | 説明 |
| --- | --- |
| **fis-console-cluster-template.json** | Fuse Console は、複数の namespace またはプロジェクトにまたがってデプロイされた Fuse アプリケーションを検出し、接続することができます。このテンプレートをデプロイするには、OpenShift の cluster-admin ロールが必要です。 |
| **fis-console-namespace-template.json** | このテンプレートは、Fuse Console の現在の OpenShift プロジェクト (namespace) へのアクセスを制限するため、単一のテナントデプロイメントとして動作します。このテンプレートをデプロイするには、現在の OpenShift プロジェクトの admin ロールが必要です。 |

任意で以下のコマンドを実行すると、すべてのテンプレートのパラメーターのリストを表示できます。

```
oc process --parameters -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-7_11_1-00016-redhat-00002/fis-console-namespace-template.json
```



**注記**

Fuse Console のテンプレートは、デフォルトでエンドツーエンド暗号化を設定するため、Fuse Console のリクエストはブラウザーからクラスター内のサービスまでエンドツーエンドでセキュア化されます。

**前提条件**

- OpenShift 3.11 のクラスターモードでは、cluster admin ロールとクラスターモードテンプレートが必要です。以下のコマンドを実行します。

```
oc adm policy add-cluster-role-to-user cluster-admin system:serviceaccount:openshift-infra:template-instance-controller
```

**手順**

コマンドラインから Fuse Console をデプロイするには、以下を行います。

1. 以下のコマンドの 1 つを実行して、Fuse Console テンプレートをベースとした新しいアプリケーションを作成します。コマンドの **myproject** はプロジェクトの名前に置き換えます。
   - Fuse Console の **cluster** テンプレートの場合は、以下のようになります。**myhost** は Fuse Console にアクセスするホストの名前になります。

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-7_11_1-00016-redhat-00002/fis-console-cluster-template.json -p ROUTE_HOSTNAME=myhost
```

- Fuse Console の **namespace** テンプレートの場合は以下のようになります。

```
oc new-app -n myproject -f https://raw.githubusercontent.com/jboss-fuse/application-templates/application-templates-2.1.0.fuse-sb2-7_11_1-00016-redhat-00002/fis-console-namespace-template.json
```

**注記**

**namespace** テンプレートの route_hostname パラメーターは OpenShift によって自動的に生成されるため、省略することが可能です。

2. 以下のコマンドを実行して、Fuse Console デプロイメントの状態と URL を取得します。

```
oc status
```

3. ブラウザーから Fuse Console にアクセスするには、提供される URL を使用します。

**以下に例を示します。**

+**https://fuse-console.192.168.64.12.nip.io**.

## 3.2. OPENSHIFT 3.11 の FUSE CONSOLE から単一の FUSE POD を監視

OpenShift 3.11 で実行している Fuse Pod の Fuse Console を開きます。

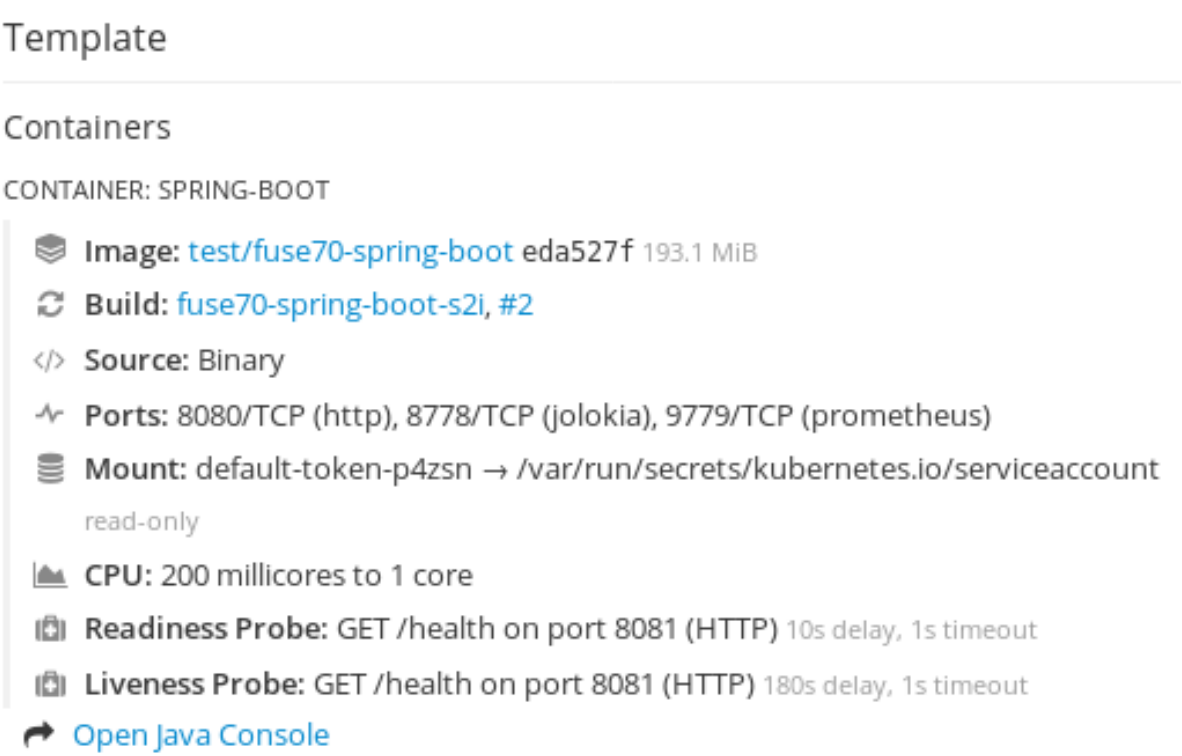
**前提条件**

- Pod ビューで Fuse Console へのリンクを表示するよう OpenShift を設定するには、Fuse on OpenShift イメージを実行している Pod が **jolokia** に設定された name 属性内で TCP ポートを宣言する必要があります。

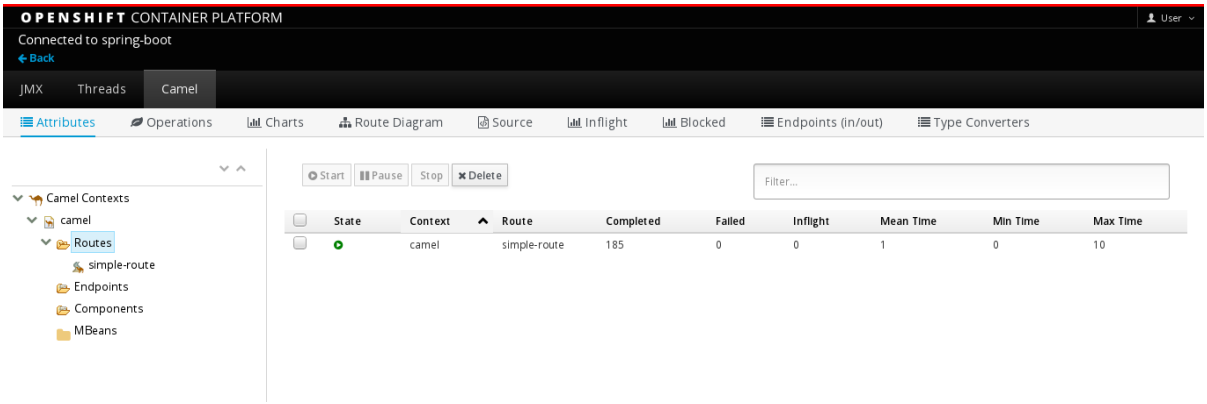
```
{
  "kind": "Pod",
  [...]
  "spec": {
    "containers": [
      {
        [...]
        "ports": [
          {
            "name": "jolokia",
            "containerPort": 8778,
            "protocol": "TCP"
          }
```

**手順**

1. OpenShift プロジェクトの **Applications → Pods** ビューで、Pod 名をクリックし、実行している Fuse Pod の詳細を表示します。このページの右側に、コンテナーテンプレートの概要が表示されます。




2. このビューの **Open Java Console** リンクをクリックし、Fuse Console を開きます。



# 第4章 コンテナーおよびアプリケーションの表示

OpenShift の Fuse Console にログインすると、Fuse Console のホームページに利用可能なコンテナーが表示されます。

**手順**

- コンテナーを管理 (作成、編集、または削除) するには、OpenShift コンソールを使用します。
- OpenShift クラスターで Fuse アプリケーションと AMQ ブローカー (該当する場合) を確認するには、**Online** タブをクリックします。

# 第5章 APACHE CAMEL アプリケーションの表示および管理

Fuse Console の **Camel** タブで Apache Camel のコンテキスト、ルート、および依存関係を表示および管理します。

次の詳細を表示できます。

- 実行中の Camel コンテキストすべてのリスト。
- Camel バージョン番号やランタイム統計など、各 Camel コンテキストの詳細情報。
- 各 Camel アプリケーションの全ルートおよびランタイム統計のリスト。
- 実行中のルートとリアルタイムのメトリクスのグラフィカル表示。

また、以下を行うと Camel アプリケーションと対話もできます。

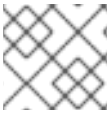
- コンテキストの起動および一時停止。
- 再起動、停止、一時停止、再開などを実行できるよう、すべての Camel アプリケーションとそれらのルートのライフサイクルを管理すること。
- 実行中のルートのライブトレースおよびデバッグ。
- Camel エンドポイントへのメッセージの閲覧および送信。

**前提条件**

**Camel** タブは、1 つ以上の Camel ルートを使用するコンテナーに接続する場合のみ使用できます。

## 5.1. コンテキストの起動、一時停止、または削除

1. **Camel** タブのツリービューで、**Camel Contexts** をクリックします。
2. リストのコンテキストの横にあるボックスにチェックマークを入れます。
3. **Start** または **Suspend** をクリックします。
4. コンテキストを削除するには以下を行います。
   a. コンテキストを停止します。
   b. 楕円のアイコンをクリックし、ドロップダウンメニューで **Delete** を選択します。



**注記**

コンテキストを削除する場合、デプロイされたアプリケーションから削除します。

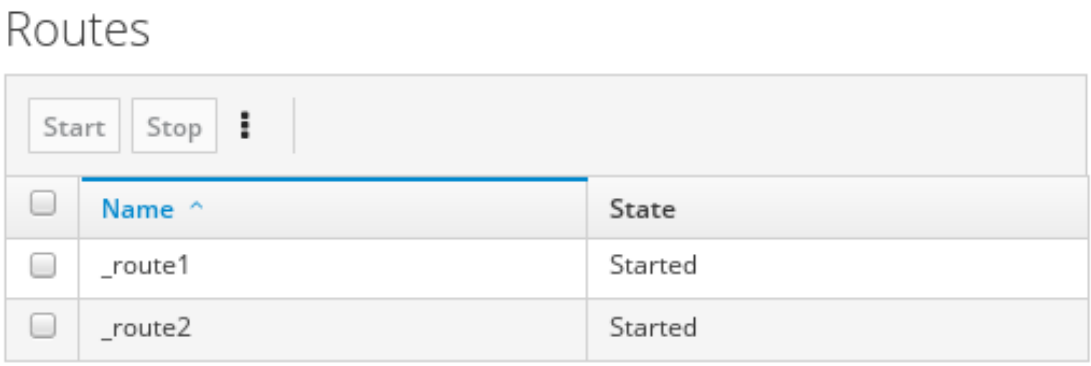
## 5.2. CAMEL アプリケーションの詳細表示

1. **Camel** タブのツリービューで、Camel アプリケーションをクリックします。
2. アプリケーションの属性と値のリストを表示するには、**Attributes** をクリックします。
3. アプリケーション属性をグラフィカルに表示するには、**Chart** をクリックした後、**Edit** をクリックし、チャートに表示する属性を選択します。

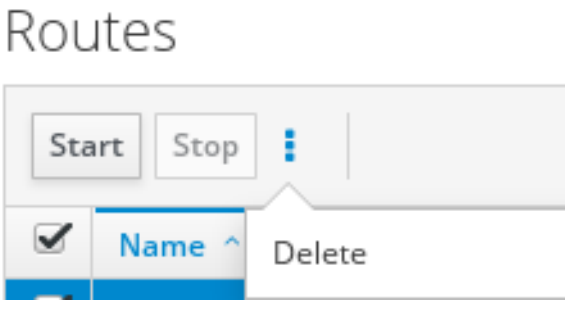
4. inflight exchange および blocked exchange を表示するには、**Exchanges** をクリックします。
5. アプリケーションエンドポイントを表示するには、**Endpoints** をクリックします。リストは **URL**、**Route ID**、および **direction** で絞り込むことができます。
6. メッセージ本文とメッセージヘッダーを別のタイプに変換するために使用される Camel 組み込みタイプ変換メカニズムに関連する統計を表示、有効化、および無効化するには、**Type Converters** をクリックします。
7. JMX 操作 (XML からのルートの追加または更新、クラスパスで利用できる Camel コンポーネントの検索など) を表示および実行するには、**Operations** をクリックします。

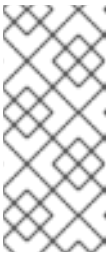
## 5.3. CAMEL ルートリストの表示および CAMEL ルートとの対話

1. ルートのリストを表示するには、以下を行います。
   a. **Camel** タブをクリックします。
   b. ツリービューでアプリケーションの routes フォルダーをクリックします。




2. 1 つまたは複数のルートを起動、停止、または削除するには、以下を行います。
   a. リストのルートの横にあるボックスにチェックマークを入れます。
   b. **Start** または **Stop** をクリックします。
   c. 最初にルートを停止してから削除する必要があります。停止したら楕円のアイコンをクリックし、ドロップダウンメニューで **Delete** を選択します。



**注記**

- ルートを削除する場合、デプロイされたアプリケーションから削除します。
- ツリービューで特定のルートを選択し、右上のメニューをクリックして起動、停止、または削除することもできます。

3. ルートのグラフィカルな図を表示するには、**Route Diagram** をクリックします。

4. inflight exchange および blocked exchange を表示するには、**Exchanges** をクリックします。

5. エンドポイントを表示するには、**Endpoints** をクリックします。URL、Route ID、および方向でリストを絞り込むことができます。

6. **Type Converters** をクリックし、Camel の組み込みタイプ変換メカニズムに関連する統計を表示、有効化、および無効化します。このメカニズムはメッセージ本文とメッセージヘッダーを別のタイプに変換するために使用されます。

7. 特定のルートと対話するには、以下を行います。

    a. **Camel** タブのツリービューで、ルートを選択します。

    b. ルート属性と値のリストを表示するには、**Attributes** をクリックします。

    c. ルート属性をグラフィカルに表示するには、**Chart** をクリックします。**Edit** をクリックすると、チャートに表示する属性を選択できます。

    d. inflight exchange および blocked exchange を表示するには、**Exchanges** をクリックします。

    e. **Operations** をクリックして JMX 操作 (ルートを XML としてダンプ、ルートの Camel ID 値の取得など) を表示および実行できます。

8. ルートを介してメッセージをトレースするには、以下を実行します。

    a. **Camel** タブのツリービューで、ルートを選択します。

    b. **Trace** を選択し、**Start tracing** をクリックします。

9. メッセージをルートに送信するには、以下を行います。

    a. **Camel** タブのツリービューでコンテキストのエンドポイントフォルダーを開き、エンドポイントを選択します。

    b. **Send** サブタブをクリックします。

    c. JSON または XML 形式のメッセージを設定します。

    d. **Send** をクリックします。

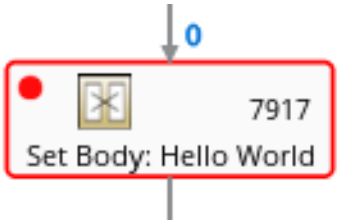
    e. ルートの **Trace** タブに戻り、ルートを介したメッセージのフローを確認します。

## 5.4. ルートのデバッグ

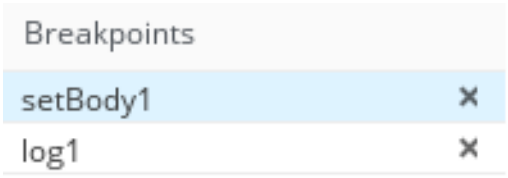
1. **Camel** タブのツリービューで、ルートを選択します。

2. **Debug** を選択し、**Start debugging** をクリックします。

3. ブレークポイントを追加するには、図のノードを選択し、**Add breakpoint** をクリックします。ノードに赤い点が表示されます。




ノードがブレークポイントのリストに追加されます。




4. 下矢印をクリックして次のノードに移動するか、**Play** ボタンをクリックしてルートの実行を再開します。
5. **Pause** ボタンをクリックして、ルートのすべてのスレッドを一時停止します。
6. 終了したら **Stop debugging** をクリックします。すべてのブレークポイントが消去されます。

# 第6章 AMQ ブローカーの表示

Fuse Console を設定して、OpenShift クラスターにデプロイされた AMQ ブローカーを表示できます。

### 前提条件

各 AMQ ブローカーイメージ (Fuse Console で表示する) は以下である必要があります。

- Fuse Console がインストールされているのと同じ OpenShift クラスターにインストールする必要があります。
- Deploying AMQ Broker on OpenShift の Fuse Console で Artemis プラグインを有効にする 説明にあるとおり、Fuse Console が認識および接続するように 各 AMQ ブローカーイメージを設定する必要があります。

### 手順

- **Artemis** をクリックして AMQ 管理コンソールを表示し、AMQ ブローカーの状態を監視します。AMQ ブローカーは Apache ActiveMQ Artemis を基にしています。

AMQ 管理コンソールの使用に関する詳細は、**AMQ Broker の管理**の AMQ 管理コンソールの使用を参照してください。

# 第7章 JMX ドメインおよび MBEAN の表示および管理

JMX (Java Management Extensions) は、実行時にリソース (サービス、デバイス、およびアプリケーション) を動的に管理できる Java 技術です。リソースは MBean (Managed Bean) と呼ばれるオブジェクトで表現されます。リソースは、作成、実装、またはインストール後、すぐに管理および監視できます。

Fuse Console で JMX プラグインを使用すると、JMX ドメインと MBean を表示および管理できます。MBean 属性の表示、コマンドの実行、および MBean の統計を示すチャートの作成を行うことができます。

**JMX** タブは、フォルダーに整理されたアクティブな JMX ドメインと MBean のツリービューを提供します。詳細を確認し、MBean でコマンドを実行できます。

**手順**

1. MBean 属性を表示および編集するには、以下を行います。
   a. ツリービューで MBean を選択します。
   b. **Attributes** タブをクリックします。
   c. 属性をクリックしてその詳細を表示します。
2. 操作を実行するには、以下を行います。
   a. ツリービューで MBean を選択します。
   b. **Operations** タブをクリックし、リストにある操作の 1 つをデプロイメントします。
   c. **Execute** をクリックし、操作を実行します。
3. チャートを表示するには、以下を行います。
   a. ツリービューで項目を選択します。
   b. **Chart** タブをクリックします。

# 第8章 QUARTZ スケジュールの表示および管理

Quartz (http://www.quartz-scheduler.org/) は、ほとんどの Java アプリケーション内で統合できる、機能豊富なオープンソースジョブスケジューリングライブラリーです。Quartz を使用して、ジョブを実行するための単純または複雑なスケジュールを作成できます。ジョブは、標準の Java コンポーネントとして定義され、プログラムで実行するほとんどの処理を実行できます。

Camel ルートが **camel-quartz2** コンポーネントをデプロイすると、Fuse Console には **Quartz** タブが表示されます。JMX ツリービューから Quartz mbeans に交互にアクセスできることに注意してください。

**手順**

1. Fuse Console で **Quartz** タブをクリックします。
   **Quartz** ページには、Quartz スケジューラーのツリービューと、**Scheduler**、**Triggers**、および **Jobs** タブがあります。

2. スケジューラーを一時停止または起動するには、**Scheduler** タブのボタンをクリックします。

3. **Triggers** タブをクリックして、ジョブを実行するタイミングを決定するトリガーを表示します。たとえば、トリガーは、一日の指定の時刻 (ミリ秒単位) に、指定した日数、または指定した回数もしくは特定の回数繰り返してジョブを開始するように指定できます。

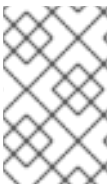
   - トリガーの一覧をフィルタリングするには、ドロップダウンリストから **State**、**Group**、**Name**、または **Type** を選択します。続いて、入力フィールドに選択または入力することで、さらにリストをフィルターできます。

   - トリガーを一時停止、再開、更新、または手動で実行するには、**Action** 列のオプションをクリックします。

4. **Jobs** タブをクリックして実行中のジョブのリストを表示します。テーブルの **Group**、**Name**、**Durable**、**Recover**、**Job ClassName**、および **Description** の列でリストをソートできます。

# 第9章 診断の表示

**Diagnostics** タブを使用して、JVM DiagnosticCommand および HotspotDiangostic インターフェイスから JVM に関する診断情報を表示します。



**注記**

この機能は、Java Mission Control (jmc) の **Diagnostic Commands** ビューや、コマンドラインツールの jcmd と似ています。場合によっては、プラグインが対応する jcmd コマンドを提供します。

**手順**

1. ロードされたクラスのインスタンス数や、これらのインスタンスが使用するバイト数を取得するには、**Class Histogram** をクリックします。操作が繰り返し行われた場合、最後の操作実行との差異がタブに表示されます。
2. JVM 診断のフラグ設定を表示するには、**JVM flags** をクリックします。
3. 稼働中の JVM でもフラグ設定を変更できます。

**関連情報**

サポートされる JVM はプラットフォームによって異なります。詳細は以下を参照してください。

- http://www.oracle.com/technetwork/java/vmoptions-jsp-140102.html
- http://openjdk.java.net/groups/hotspot/docs/RuntimeOverview.html

# 第10章 スレッドの表示

スレッドの状態を表示および監視できます。

**手順**

1. **Runtime** タブをクリックし、**Threads** サブタブをクリックします。**Threads** ページには、アクティブなスレッドと各スレッドのスタックトレースの詳細が表示されます。デフォルトでは、スレッドリストにはすべてのスレッドが ID 値が大きい順に表示されます。

2. ID 値が小さい順に表示するには、**ID** 列ラベルをクリックします。

3. 任意で、スレッドの状態 (例: **Blocked**) やスレッド名でリストを絞り込むことができます。

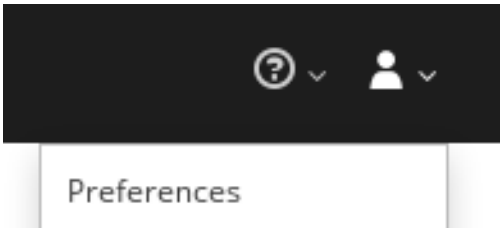
4. ロッククラス名やスレッドのフルスタックトレースなど、特定スレッドの詳細情報を表示するには、**Actions** 列で **More** をクリックします。

# 第11章 FUSE CONSOLE でデータが正しく表示されるよう確認

Fuse Console のキューおよび接続の表示で、不足しているキューや接続があったり、一貫性のないアイコンが表示される場合は、Jolokia コレクションサイズパラメーターを調節します。このパラメーターは、Jolokia が応答でマーシャルするアレイの最大要素数を指定するものです。

**手順**

1. Fuse Console の右上隅にあるユーザーアイコンをクリックして、**Preferences** をクリックします。




2. **Maximum collection size** オプションの値を大きくします (デフォルトは 50,000)。
3. **Close** をクリックします。

# 付録A FUSE CONSOLE 設定プロパティー

デフォルトでは、Fuse Console 設定は **hawtconfig.json** ファイルで定義されます。タイトル、ロゴ、ログインページの情報などの Fuse Console 設定情報をカスタマイズできます。

表A.1「Fuse Console 設定プロパティー」 には、プロパティーの説明と、各プロパティーに値が必要かどうかが示されています。

**表A.1 Fuse Console 設定プロパティー**

| セクション | プロパティー名 | デフォルト値 | 説明 | 必須/オプション |
|---|---|---|---|---|
| about | Title | Red Hat Fuse Management Console | Fuse Console の About ページに表示されるタイトル。 | 必須 |
| | productInfo | **空の値** | Fuse Console の About ページに表示される製品情報。 | オプション |
| | additionalInfo | **空の値** | Fuse Console の About ページに表示される追加情報。 | オプション |
| | copyright | **空の値** | Fuse Console の About ページに表示される著作権情報。 | オプション |
| | imgSrc | **img/Logo-RedHat-A-Reverse-RGB.png** | Fuse Console の About ページに表示されるイメージ。 | 必須 |
| branding | appName | Red Hat Fuse Management Console | アプリケーションの名前。この名前は Fuse Console のタイトルバーに表示されます。 | 必須 |

| セクション | プロパティー名 | デフォルト値 | 説明 | 必須/オプション |
|---|---|---|---|---|
| | appLogoUrl | **img/Logo-Red_Hat-Fuse-A-Reverse-RGB.png** | Fuse Console のナビゲーションバーに表示されるアプリケーションのロゴイメージファイルへのパス。Hawtio ステータス URL への相対パス、または絶対 URL を値として指定可能です。 | 必須 |
| | Css | | アプリケーションのスタイルに使用できる外部 CSS スタイルシートの URL。Hawtio ステータス URL への相対パス、または絶対 URL を指定可能です。 | オプション |
| | companyLogoUrl | **img/Logo-RedHat-A-Reverse-RGB.png** | 会社のロゴイメージファイルへのパス。 | 必須 |
| | Favicon | | 通常、Web ブラウザータブに表示される favicon の URL。Hawtio ステータス URL への相対パス、または絶対 URL を指定可能です。 | オプション |
| login | description | **空の値** | Fuse Console の Login ページに表示される説明テキスト (例: **http://localhost:8181/hawtio**)。 | オプション |

| セクション | プロパティー名 | デフォルト値 | 説明 | 必須/オプション |
|---|---|---|---|---|
| | links | [ ] | **"url"** と **"text"** のペアのアレイを指定し、ユーザーが詳細またはヘルプを取得できるページへの追加リンクを提供します。 | オプション |
| disabledRoutes | **なし** | [ ] | コンソールの特定のパス (プラグインなど) を無効にします。このセクションは変更しないでください。変更は、OpenShift 以外のディストリビューションではサポートされません。 | オプション |