



Red Hat Fuse 7.11

OpenShift에서 Fuse 관리

Fuse 콘솔을 사용하여 Fuse 애플리케이션 관리

Red Hat Fuse 7.11 OpenShift에서 Fuse 관리

Fuse 콘솔을 사용하여 Fuse 애플리케이션 관리

법적 공지

Copyright © 2023 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.

초록

Fuse 애플리케이션을 배포할 때 Fuse Console을 사용하여 Red Hat Fuse 통합을 모니터링하고 상호 작용할 수 있습니다.

차례

머리말	3
보다 포괄적 수용을 위한 오픈 소스 용어 교체	4
1장. FUSE 콘솔 정보	5
2장. OPENSIFT 4.X에서 FUSE 콘솔 설정	6
2.1. OPERATORHUB를 사용하여 OPENSIFT 4.X에 FUSE CONSOLE 설치 및 배포	6
2.2. 명령줄을 사용하여 OPENSIFT 4.X에 FUSE CONSOLE 설치 및 배포	8
2.3. OPENSIFT 4.X에서 FUSE CONSOLE에 대한 역할 기반 액세스 제어	12
2.4. OPENSIFT 4.X에서 FUSE CONSOLE 업그레이드	15
2.5. OPENSIFT 4.X 서버에서 FUSE 이미지 스트림 및 템플릿 업그레이드	16
2.6. OPENSIFT 4.X에서 FUSE CONSOLE의 성능 튜닝	18
3장. OPENSIFT 3.11에서 FUSE 콘솔 설정	23
3.1. OPENSIFT 3.11에 FUSE 콘솔 배포	23
3.2. OPENSIFT 3.11의 FUSE CONSOLE에서 단일 FUSE POD 모니터링	25
4장. 컨테이너 및 애플리케이션 보기	27
5장. APACHE CAMEL 애플리케이션 보기 및 관리	28
5.1. 컨텍스트 시작, 일시 중지 또는 삭제	28
5.2. CAMEL 애플리케이션 세부 정보 보기	28
5.3. CAMEL 경로 목록 보기 및 상호 작용	29
5.4. 경로 디버깅	30
6장. AMQ 브로커 보기	32
7장. CRYOSTAT 도메인 및 CRYOSTAT 보기 및 관리	33
8장. QUARTZ 일정 보기 및 관리	34
9장. 진단 보기	35
10장. 스레드 보기	36
11장. FUSE CONSOLE에 데이터가 올바르게 표시되도록 합니다.	37
부록 A. FUSE 콘솔 구성 속성	38

머리말

Red Hat Fuse는 Fuse 통합 보기 및 관리를 위한 두 가지 엔터프라이즈 모니터링 툴을 제공합니다.

- Fuse 콘솔은 실행 중인 Fuse 컨테이너를 모니터링하고 관리하기 위해 브라우저에서 액세스할 수 있는 웹 기반 콘솔입니다. Fuse 콘솔은 Hawtio 오픈 소스 소프트웨어(<https://hawt.io/>)를 기반으로 합니다. 이 가이드에서는 Fuse Console 사용 방법을 설명합니다.
- Prometheus는 Fuse 배포에 대한 시스템 및 통합 수준 지표를 저장합니다. Grafana와 같은 그래픽 분석 인터페이스를 사용하여 저장된 기록 데이터를 보고 분석할 수 있습니다. Prometheus 사용에 대한 자세한 내용은 다음 설명서를 참조하십시오.
 - [Prometheus 문서](#)
 - [OpenShift 기반 Fuse 가이드](#)
 - [OpenShift Container Platform에 온라인 설치 및 운영 체제](#)

이 가이드의 대상은 Red Hat Fuse on JBoss EAP 관리자입니다. 이 가이드에서는 Red Hat Fuse 플랫폼, Apache Camel 및 조직의 처리 요구 사항을 잘 알고 있다고 가정합니다.

보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 [CTO Chris Wright의 메시지](#)에서 참조하십시오.

1장. FUSE 콘솔 정보

Red Hat Fuse Console은 Hawtio 오픈 소스 소프트웨어를 기반으로 하는 웹 콘솔입니다. 지원되는 브라우저 목록은 [지원되는 구성으로 이동합니다](#).

Fuse Console은 배포된 하나 이상의 Fuse 컨테이너의 세부 정보를 검사하고 관리하는 중앙 인터페이스를 제공합니다. 또한 Red Hat Fuse 및 시스템 리소스를 모니터링하고 업데이트를 수행하며 서비스를 시작하거나 중지할 수 있습니다.

Red Hat Fuse 독립 실행형을 설치하거나 OpenShift에 Fuse를 사용하는 경우 Fuse Console을 사용할 수 있습니다. Fuse 콘솔에서 보고 관리할 수 있는 통합은 실행 중인 플러그인에 따라 다릅니다. 가능한 플러그인은 다음과 같습니다.

- Camel
- JMX
- OSGI
- 런타임
- 로그

2장. OPENSIFT 4.X에서 FUSE 콘솔 설정

OpenShift 4.x에서 Fuse 콘솔을 설정하려면 이를 설치하고 배포해야 합니다. Fuse Console을 설치하고 배포하기 위한 다음과 같은 옵션이 있습니다.

- 2.1절. "OperatorHub를 사용하여 OpenShift 4.x에 Fuse Console 설치 및 배포"
Fuse Console Operator를 사용하여 특정 네임스페이스의 Fuse 애플리케이션에 액세스할 수 있도록 Fuse Console을 설치하고 배포할 수 있습니다. Operator는 Fuse Console 보안을 처리합니다.
- 2.2절. "명령줄을 사용하여 OpenShift 4.x에 Fuse Console 설치 및 배포"
명령줄과 Fuse Console 템플릿 중 하나를 사용하여 Fuse Console을 설치하고 배포하여 OpenShift 클러스터 또는 특정 네임스페이스에 있는 여러 네임스페이스의 Fuse 애플리케이션에 액세스할 수 있습니다. 클라이언트 인증서를 배포하기 전에 클라이언트 인증서를 생성하여 Fuse Console을 보호해야 합니다.

필요한 경우 2.3절. "OpenShift 4.x에서 Fuse Console에 대한 역할 기반 액세스 제어"에 설명된 대로 Fuse Console에 대한 역할 기반 액세스 제어(RBAC)를 사용자 지정할 수 있습니다.

2.1. OPERATORHUB를 사용하여 OPENSIFT 4.X에 FUSE CONSOLE 설치 및 배포

OpenShift 4.x에 Fuse Console을 설치하려면 OpenShift OperatorHub에 제공된 Fuse Console Operator를 사용할 수 있습니다. Fuse 콘솔을 배포하려면 설치된 Operator의 인스턴스를 만듭니다.

사전 요구 사항

- 컨테이너 이미지의 registry.redhat.io 로 인증 인증에 설명된 대로 registry.redhat.io 를 사용하여 인증을 구성했습니다.
- Fuse 콘솔에 대한 RBAC(역할 기반 액세스 제어)를 사용자 지정하려면 Fuse Console Operator를 설치하는 동일한 OpenShift 네임스페이스에 RBAC 구성 맵 파일이 있어야 합니다. [OpenShift 4.x의 Fuse Console에 대한 역할 기반 액세스 제어](#)에 설명된 대로 기본 RBAC 동작을 사용하려면 구성 맵 파일을 제공할 필요가 없습니다.

절차


Fuse 콘솔을 설치하고 배포하려면 다음을 수행합니다.

1. 웹 브라우저에서 클러스터 관리자 액세스 권한이 있는 사용자로 OpenShift 콘솔에 로그인합니다.
2. **Operators** 를 클릭한 다음 **OperatorHub** 를 클릭합니다.
3. 검색 필드 창에서 **Fuse Console** 을 입력하여 Operator 목록을 필터링합니다.
4. **Fuse Console Operator** 를 클릭합니다.
5. Fuse Console Operator 설치 창에서 **설치**를 클릭합니다.
Create Operator Subscription 양식이 열립니다.
 - **Update Channel**에서는 **7.11.x** 를 선택합니다.
 - **설치 모드**의 경우 기본값(클러스터의 특정 네임스페이스)을 수락합니다.
Operator를 설치한 후 Fuse Console을 배포할 때 클러스터의 모든 네임스페이스에서 애플리케이션을 모니터링하거나 Fuse Console Operator가 설치된 네임스페이스에서만 애플리케이션을 모니터링하도록 선택할 수 있습니다.

- 설치된 네임스페이스의 경우 Fuse Console Operator를 설치할 네임스페이스를 선택합니다.
 - 업데이트 승인의 경우 자동 또는 수동 을 선택하여 OpenShift에서 Fuse Console Operator에 대한 업데이트를 처리하는 방법을 구성할 수 있습니다.
 - 자동 업데이트를 선택하면 새 버전의 Fuse Console Operator가 사용 가능할 때 OLM(Operator Lifecycle Manager)은 개입 없이 Fuse Console의 실행 중인 인스턴스를 자동으로 업그레이드합니다.
 - 수동 업데이트를 선택하면 최신 버전의 Operator가 사용 가능할 때 OLM에서 업데이트 요청을 생성합니다. 클러스터 관리자는 Fuse Console Operator가 새 버전으로 업데이트 되도록 해당 업데이트 요청을 수동으로 승인해야 합니다.
6. 설치를 클릭합니다.
OpenShift는 현재 네임스페이스에 Fuse Console Operator를 설치합니다.
 7. 설치를 확인하려면 Operator를 클릭한 다음 **Installed Operators** 를 클릭합니다. Operator 목록에서 Fuse Console을 볼 수 있습니다.
 8. OpenShift 웹 콘솔을 사용하여 Fuse 콘솔을 배포하려면 다음을 수행합니다.
 - a. 설치된 Operator 목록에서 이름 열에서 **Fuse Console** 을 클릭합니다.
 - b. 제공된 API 아래에 있는 **Operator 세부 정보** 페이지에서 **인스턴스 생성**을 클릭합니다. 구성 기본값을 수락하거나 선택적으로 편집합니다.

복제본의 경우 Fuse Console 성능(예: 고가용성 환경에서)을 늘리려면 Fuse Console에 할당된 Pod 수를 늘릴 수 있습니다.

Rbac (역할 기반 액세스 제어)의 경우 기본 RBAC 동작을 사용자 정의하고 ConfigMap 파일이 Fuse Console Operator를 설치한 네임스페이스에 이미 존재하는 경우에만 **구성 맵 필드**에 값을 지정합니다. RBAC에 대한 자세한 내용은 [OpenShift 4.x의 Fuse Console에 대한 역할 기반 액세스 제어를 참조하십시오](#).
 - Nginx**의 경우 [Fuse Console Operator 설치에 대한 성능 튜닝](#) 을 참조하십시오.
 - c. 생성을 클릭합니다.
Fuse Console Operator 세부 정보 페이지가 열리고 배포 상태가 표시됩니다.
 9. Fuse 콘솔을 열려면 다음을 수행합니다.
 - a. 네임스페이스 배포: OpenShift 웹 콘솔에서 Fuse Console Operator를 설치한 프로젝트를 열고 **개요** 를 선택합니다. **프로젝트 개요** 페이지에서 **시작** 관리자 섹션까지 아래로 스크롤하고 Fuse Console 링크를 클릭합니다.

클러스터 배포의 경우 OpenShift 웹 콘솔의 제목 표시줄에서 그리드 아이콘()을 클릭합니다. 팝업 메뉴의 **Red Hat 애플리케이션** 에서 Fuse Console URL 링크를 클릭합니다.
 - b. Fuse 콘솔에 로그인합니다.
필요한 권한이 나열된 브라우저에서 권한 **부여** 페이지가 열립니다.
 - c. 선택한 권한 허용을 클릭합니다.
브라우저에서 Fuse Console이 열리고 액세스할 수 있는 권한이 있는 Fuse 애플리케이션 포드가 표시됩니다.
 10. 확인할 애플리케이션에 대한 **연결**을 클릭합니다.
Fuse Console에 애플리케이션이 표시되는 새 브라우저 창이 열립니다.

2.2. 명령줄을 사용하여 OPENSIFT 4.X에 FUSE CONSOLE 설치 및 배포

OpenShift 4.x에서는 명령줄에서 Fuse 콘솔을 설치하고 배포할 다음 배포 옵션 중 하나를 선택할 수 있습니다.

- **클러스터** - Fuse Console은 OpenShift 클러스터의 여러 네임스페이스(프로젝트)에 배포된 Fuse 애플리케이션을 검색하고 연결할 수 있습니다. 이 템플릿을 배포하려면 OpenShift 클러스터에 대한 관리자 역할이 있어야 합니다.
- **역할 기반 액세스 제어**가 있는 클러스터- 구성 가능한 역할 기반 액세스 제어(RBAC)가 있는 클러스터 템플릿입니다. 자세한 내용은 [OpenShift 4.x의 Fuse Console에 대한 역할 기반 액세스 제어를 참조하십시오](#).
- **네임스페이스** - Fuse Console은 특정 OpenShift 프로젝트(네임스페이스)에 액세스할 수 있습니다. 이 템플릿을 배포하려면 OpenShift 프로젝트에 대한 관리자 역할이 있어야 합니다.
- **역할 기반 액세스 제어**가 있는 네임스페이스- 구성 가능한 RBAC가 있는 네임스페이스 템플릿입니다. 자세한 내용은 [OpenShift 4.x의 Fuse Console에 대한 역할 기반 액세스 제어를 참조하십시오](#).

Fuse Console 템플릿의 매개변수 목록을 보려면 다음 OpenShift 명령을 실행합니다.

```
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
```

사전 요구 사항

- Fuse 콘솔을 설치하고 배포하기 전에 [OpenShift 4.x에서 Fuse Console을 보호하기 위해 인증서 생성에 설명된 대로 서비스 서명 인증 기관으로 서명된 클라이언트 인증서를 생성해야 합니다](#).
- OpenShift 클러스터에 대한 **클러스터 관리자** 역할이 있어야 합니다.
- 컨테이너 이미지의 [registry.redhat.io](#) 로 인증 인증에 설명된 대로 [registry.redhat.io](#) 를 사용하여 인증을 구성했습니다.
- OpenShift 4.x 서버에 Fuse 이미지 스트림 설치 및 템플릿에 설명된 대로 Fuse 콘솔 이미지 스트림(다른 Fuse 이미지 스트림과 함께)이 설치됩니다.

절차

1. 다음 명령을 사용하여 모든 템플릿 목록을 검색하여 Fuse Console 이미지 스트림이 설치되었는지 확인합니다.

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

2. 선택적으로 이미 설치된 이미지 스트림을 새 릴리스 태그로 업데이트하려면 다음 명령을 사용하여 Fuse Console 이미지를 **openshift** 네임스페이스로 가져옵니다.

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

3. 다음 명령을 실행하여 Fuse Console **APP_NAME** 값을 가져옵니다.

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

여기서 **TEMPLATE-FILENAME** 은 다음 템플릿 중 하나입니다.

- 클러스터 템플릿:

```
`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`
```

- 구성 가능한 RBAC가 있는 클러스터 템플릿:

```
`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`
```

- 네임스페이스 템플릿:

```
`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`
```

- 구성 가능한 RBAC가 있는 네임스페이스 템플릿:

```
`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`
```

예를 들어 구성 가능한 RBAC가 있는 클러스터 템플릿의 경우 다음 명령을 실행합니다.

```
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. [OpenShift 4.x에서 Fuse Console 보안에서 생성한 인증서에서](#) 다음 명령을 사용하여 시크릿을 생성하고 Fuse Console에 마운트합니다(여기서 **APP_NAME** 은 Fuse Console 애플리케이션의 이름입니다).

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

5. 다음 명령을 실행하여 Fuse Console 템플릿의 로컬 복사본을 기반으로 새 애플리케이션을 생성합니다. 여기서 **myproject** 는 OpenShift 프로젝트의 이름, **mytemp** 는 Fuse Console 템플릿이 포함된 로컬 디렉터리의 경로입니다. **myhost** 는 Fuse Console에 액세스할 수 있는 호스트 이름입니다.

- 클러스터 템플릿의 경우:

```
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 템플릿이 있는 클러스터의 경우:

```
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
```

- 네임스페이스 템플릿의 경우:

```
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 템플릿이 있는 네임스페이스의 경우:

```
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 웹 콘솔을 열 수 있도록 Fuse Console을 구성하려면 다음 명령을 실행하여 **OPENSHIFT_WEB_CONSOLE_URL** 환경 변수를 설정합니다.

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

7. 다음 명령을 실행하여 Fuse Console 배포의 상태 및 URL을 가져옵니다.

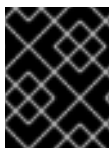
```
oc status
```

8. 브라우저에서 Fuse 콘솔에 액세스하려면 7단계로 반환되는 URL(예: <https://fuse-console.192.168.64.12.nip.io>)을 사용합니다.

2.2.1. OpenShift 4.x에서 Fuse Console을 보호하기 위한 인증서 생성

OpenShift 4.x에서 Fuse Console 프록시와 Jolokia 에이전트 보안 간의 연결을 유지하려면 Fuse Console을 배포하기 전에 클라이언트 인증서를 생성해야 합니다. 클라이언트 인증서에 서명하려면 서비스 서명 인증 기관 개인 키를 사용해야 합니다.

명령줄을 사용하여 Fuse 콘솔을 설치하고 배포하는 **경우에만** 다음 절차를 따라야 합니다. Fuse Console Operator를 사용하는 경우 이 작업을 처리합니다.



중요

각 OpenShift 클러스터에 대해 별도의 클라이언트 인증서를 생성하고 서명해야 합니다. 두 개 이상의 클러스터에 동일한 인증서를 사용하지 마십시오.

사전 요구 사항

- OpenShift 클러스터에 대한 **클러스터 관리자** 액세스 권한이 있어야 합니다.
- 두 개 이상의 OpenShift 클러스터에 대한 인증서를 생성하고 이전에 현재 디렉터리에 다른 클러스터에 대한 인증서를 생성한 경우 다음 중 하나를 수행하여 현재 클러스터에 대한 다른 인증서를 생성하십시오.
 - 현재 디렉터리에서 기존 인증서 파일(예: **ca.crt, ca.key, ca.srl**)을 삭제합니다.
 - 다른 작업 디렉터리로 변경합니다. 예를 들어 현재 작업 디렉터리 이름이 **cluster1** 인 경우 새 **cluster2** 디렉터리를 생성하고 작업 디렉터리를 해당 디렉터리로 변경합니다.

```
mkdir ../cluster2
```

```
cd ../cluster2
```

절차

1. 클러스터 관리자 액세스 권한이 있는 사용자로 OpenShift에 로그인합니다.

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

2. 다음 명령을 실행하여 서비스 서명 인증 기관 키를 검색합니다.

- 인증서를 검색하려면 다음을 수행합니다.

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

- 개인 키를 검색하려면 다음을 수행합니다.

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

3. **easysrsa**, **openssl** 또는 **cfssl** 을 사용하여 [Kubernetes 인증서 관리](#)에 설명된 대로 클라이언트 인증서를 생성합니다.

다음은 openssl을 사용하는 예제 명령입니다.

- a. 개인 키를 생성합니다.

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

- b. CSR 구성 파일을 작성합니다.

```
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
```

여기에서 **CN** 매개변수의 값은 애플리케이션 이름과 애플리케이션에서 사용하는 네임스페이스를 나타냅니다.

- c. CSR을 생성합니다.

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

- d. 서명된 인증서를 발급합니다.

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

다음 단계

명령줄을 사용하여 OpenShift 4.x에 Fuse Console 설치 및 배포에 설명된 대로 Fuse Console의 시크릿을 생성하려면 이 인증서가 필요합니다.

2.3. OPENSIFT 4.X에서 FUSE CONSOLE에 대한 역할 기반 액세스 제어

Fuse 콘솔은 OpenShift에서 제공하는 사용자 권한에 따라 액세스를 유추하는 RBAC(역할 기반 액세스 제어)를 제공합니다. Fuse 콘솔에서 RBAC는 Pod에서 Cryostat 작업을 수행할 수 있는 사용자의 기능을 결정합니다.

OpenShift 권한 부여에 대한 자세한 내용은 [Using RBAC to define and apply permissions](#) section of the OpenShift documentation에서 참조하십시오.

Operator를 사용하여 OpenShift에 Fuse Console을 설치할 때 역할 기반 액세스는 기본적으로 활성화됩니다.

템플릿으로 설치하여 Fuse Console에 대한 역할 기반 액세스를 구현하려면 [명령줄을 사용하여 Fuse Console 설치 및 배포에 설명된 대로 RBAC\(fuse-console-cluster-rbac.yml 또는 fuse-console-namespace-rbac.yml\)](#)로 구성할 수 있는 템플릿 중 하나를 사용해야 합니다.

Fuse Console RBAC는 OpenShift의 Pod 리소스에 대한 사용자 동사 액세스를 활용하여 Fuse Console에서 Pod의 작업에 대한 사용자의 액세스 권한을 결정합니다. 기본적으로 Fuse Console의 사용자 역할은 다음 두 가지가 있습니다.

- **admin**
사용자가 OpenShift에서 포드를 업데이트할 수 있는 경우 사용자는 Fuse Console의 **admin** 역할을 유추합니다. 사용자는 Pod에 대해 Fuse Console에서 **write** Cryostat 작업을 수행할 수 있습니다.
- **뷰어**
사용자가 OpenShift에서 포드를 가져올 수 있는 경우 사용자는 Fuse Console의 **뷰어** 역할을 유추합니다. 사용자는 Pod에 대해 Fuse Console에서 **읽기 전용** 작업을 수행할 수 있습니다.



참고

비 RBAC 템플릿을 사용하여 Fuse 콘솔을 설치한 경우 Pod 리소스에 **update** 동사가 부여된 OpenShift 사용자만 Fuse Console Cryostats 작업을 수행할 수 있습니다. Pod 리소스에 **get** 동사가 부여된 사용자는 포드를 볼 수 있지만 Fuse Console 작업을 수행할 수 없습니다.

추가 리소스

- [OpenShift 4.x에서 Fuse Console에 대한 액세스 역할 확인](#)
- [OpenShift 4.x의 Fuse Console에 대한 역할 기반 액세스 사용자 정의](#)
- [OpenShift 4.x에서 Fuse Console에 대한 역할 기반 액세스 제어 비활성화](#)

2.3.1. OpenShift 4.x에서 Fuse Console에 대한 액세스 역할 확인

Fuse Console 역할 기반 액세스 제어는 Pod에 대한 사용자의 OpenShift 권한에서 유추됩니다. 특정 사용자에게 부여된 Fuse Console 액세스 역할을 확인하려면 Pod의 사용자에게 부여된 OpenShift 권한을 받으십시오.

사전 요구 사항

- 사용자 이름을 알고 있습니다.
- Pod의 이름을 알고 있습니다.

절차

- 사용자에게 Pod에 대한 Fuse Console admin 역할이 있는지 확인하려면 다음 명령을 실행하여 OpenShift에서 Pod를 업데이트할 수 있는지 확인합니다.

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

응답이 **yes** 인 경우 사용자에게 포드에 대한 Fuse Console **admin** 역할이 있습니다. 사용자는 Pod에 대해 Fuse Console에서 **write** Cryostat 작업을 수행할 수 있습니다.

- 사용자에게 Pod에 대한 Fuse Console **뷰어** 역할이 있는지 확인하려면 다음 명령을 실행하여 OpenShift에서 Pod를 가져올 수 있는지 확인합니다.

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

응답이 **yes** 인 경우 사용자에게 포드에 대한 Fuse Console **viewer** 역할이 있습니다. 사용자는 Pod에 대해 Fuse Console에서 **읽기 전용** 작업을 수행할 수 있습니다. 상황에 따라 Fuse Console은 옵션을 비활성화하거나 사용자가 **쓰기**를 시도할 때 "operation not allowed for this user" 메시지를 표시하여 **뷰어** 역할의 사용자가 **쓰기**를 수행하지 못하도록 합니다.

응답이 **없**는 경우 사용자는 Fuse Console 역할에 바인딩되지 않으며 사용자는 Fuse Console의 포드를 볼 수 없습니다.

추가 리소스

- [OpenShift 4.x에서 Fuse Console에 대한 역할 기반 액세스 제어](#)
- [OpenShift 4.x의 Fuse Console에 대한 역할 기반 액세스 사용자 정의](#)
- [OpenShift 4.x에서 Fuse Console에 대한 역할 기반 액세스 제어 비활성화](#)

2.3.2. OpenShift 4.x의 Fuse Console에 대한 역할 기반 액세스 사용자 정의

OperatorHub를 사용하여 Fuse 콘솔을 설치하는 경우 [OpenShift 4.x의 Fuse Console에 대한 역할 기반 액세스 제어에 설명된 대로 기본적으로 역할 기반 액세스 제어 \(RBAC\)](#)가 활성화됩니다. Fuse Console을 배포하기 전에 Fuse Console RBAC 동작을 사용자 지정하려면 ConfigMap 파일을 제공해야 합니다(사용자 정의 RBAC 동작을 정의함). Fuse Console Operator를 설치한 동일한 네임스페이스에 사용자 정의 ConfigMap 파일을 배치해야 합니다.

명령줄 템플릿을 사용하여 Fuse 콘솔을 설치하는 경우 **deployment-cluster-rbac.yml** 및 **deployment-namespace-rbac.yml** 템플릿은 구성 파일(ACLs.yml)이 포함된 **ConfigMap**을 생성합니다. 구성 파일은 Cryostat 작업에 허용되는 역할을 정의합니다.

사전 요구 사항

- OperatorHub를 사용하거나 Fuse Console RBAC 템플릿 중 하나를 사용하여 Fuse 콘솔을 설치했습니다(**deployment-cluster-rbac.yml** 또는 **deployment-namespace-rbac.yml**)

절차

Fuse Console RBAC 역할을 사용자 지정하려면 다음을 수행합니다.

1. 명령줄을 사용하여 Fuse 콘솔을 설치한 경우 설치 템플릿에 기본 ConfigMap 파일이 포함되어 있으므로 다음 단계로 건너뛸 수 있습니다.

OperatorHub를 사용하여 Fuse 콘솔을 설치한 경우 Fuse 콘솔을 배포하기 전에 RBAC ConfigMap을 생성합니다.

- a. 현재 OpenShift 프로젝트가 Fuse Console을 설치할 프로젝트인지 확인합니다. 예를 들어 **fusetest** 프로젝트에 Fuse Console을 설치하려면 다음 명령을 실행합니다.

```
oc project fusetest
```

- b. 템플릿에서 Fuse Console RBAC ConfigMap 파일을 생성하려면 다음 명령을 실행합니다.

```
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. 다음 명령을 실행하여 편집기에서 ConfigMap을 엽니다.

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

예를 들면 다음과 같습니다.

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

3. 파일을 편집합니다.
4. 파일을 저장하여 변경 사항을 적용합니다. OpenShift는 Fuse Console 포드를 자동으로 다시 시작합니다.

추가 리소스

- [OpenShift 4.x에서 Fuse Console에 대한 역할 기반 액세스 제어](#)
- [OpenShift 4.x에서 Fuse Console에 대한 액세스 역할 확인](#)
- [OpenShift 4.x에서 Fuse Console에 대한 역할 기반 액세스 제어 비활성화](#)

2.3.3. OpenShift 4.x에서 Fuse Console에 대한 역할 기반 액세스 제어 비활성화

명령줄을 사용하여 Fuse 콘솔을 설치하고 Fuse Console RBAC 템플릿 중 하나를 지정한 경우 Fuse 콘솔의 **HAWTIO_ONLINE_RBAC_ACL** 환경 변수는 OpenShift 서버에 역할 기반 액세스 제어(RBAC) ConfigMap 구성 파일 경로를 전달합니다. **HAWTIO_ONLINE_RBAC_ACL** 환경 변수가 지정되지 않은 경우 RBAC 지원이 비활성화되고 Pod 리소스(OpenShift의)에 대한 업데이트 동사가 부여된 사용자만 Fuse Console의 Pod에서 Cryostat 작업을 호출할 수 있습니다.

OperatorHub를 사용하여 Fuse Console을 설치할 때 역할 기반 액세스는 기본적으로 활성화되어 있으며 **HAWTIO_ONLINE_RBAC_ACL** 환경 변수가 적용되지 않습니다.

사전 요구 사항

명령줄을 사용하여 Fuse Console을 설치하고 Fuse Console RBAC 템플릿(**deployment-cluster-rbac.yml** 또는 **deployment-namespace-rbac.yml**) 중 하나를 지정했습니다.

절차

Fuse Console에 대한 역할 기반 액세스를 비활성화하려면 다음을 수행합니다.

1. OpenShift에서 Fuse Console의 **Deployment Config** 리소스를 편집합니다.
2. 전체 **HAWTIO_ONLINE_RBAC_ACL** 환경 변수 정의를 삭제합니다.
(값만 지우는 것만으로는 충분하지 않습니다.)
3. 파일을 저장하여 변경 사항을 적용합니다. OpenShift는 Fuse Console 포드를 자동으로 다시 시작합니다.

추가 리소스

- [OpenShift 4.x에서 Fuse Console에 대한 역할 기반 액세스 제어](#)
- [OpenShift 4.x에서 Fuse Console에 대한 액세스 역할 확인](#)
- [OpenShift 4.x의 Fuse Console에 대한 역할 기반 액세스 사용자 정의](#)

2.4. OPENSIFT 4.X에서 FUSE CONSOLE 업그레이드

Red Hat OpenShift 4.x는 Red Hat Fuse Operator를 포함하여 Operator에 대한 업데이트를 처리합니다. 자세한 내용은 [Operator OpenShift 설명서](#)를 참조하십시오.

Operator 업데이트는 애플리케이션 구성 방법에 따라 애플리케이션 업그레이드를 트리거할 수 있습니다.

Fuse Console 애플리케이션의 경우 애플리케이션 사용자 정의 리소스 정의의 **.spec.version** 필드를 편집하여 애플리케이션으로의 업그레이드를 트리거할 수도 있습니다.

사전 요구 사항

- OpenShift 클러스터 관리자 권한이 있어야 합니다.

절차

Fuse Console 애플리케이션을 업그레이드하려면 다음을 수행합니다.

1. 터미널 창에서 다음 명령을 사용하여 애플리케이션 사용자 정의 리소스 정의의 **.spec.version** 필드를 변경합니다.

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

예를 들면 다음과 같습니다.

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

2. 애플리케이션 상태가 업데이트되었는지 확인합니다.

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

응답에는 버전 번호를 포함하여 애플리케이션에 대한 정보가 표시됩니다.

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

.spec.version 필드의 값을 변경하면 OpenShift에서 애플리케이션을 자동으로 재배포합니다.

- 버전 변경으로 트리거되는 재배포 상태를 확인하려면 다음을 수행합니다.

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

성공적인 배포에는 다음 응답이 표시됩니다.

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

2.5. OPENSIFT 4.X 서버에서 FUSE 이미지 스트림 및 템플릿 업그레이드

OpenShift Container Platform 4.x는 OpenShift 네임스페이스에서 작동하는 Samples Operator를 사용하며 RHEL(Red Hat Enterprise Linux) 기반 OpenShift Container Platform 이미지 스트림 및 템플릿을 업그레이드하고 업데이트합니다.

OpenShift 이미지 스트림 및 템플릿에서 Fuse를 업그레이드하려면 다음을 수행합니다.

- Samples Operator 재구성
- Fuse 이미지 스트림과 템플릿을 **Skipped Imagestreams** 및 **Skipped Templates** 필드에 추가합니다.
 - 건너편 이미지 스트림: Samples Operator 인벤토리에 있지만 클러스터 관리자가 Operator에서 무시하거나 관리하지 않도록 하려는 이미지 스트림입니다.
 - 건너뛰기된 템플릿: Samples Operator 인벤토리에 있지만 클러스터 관리자가 Operator에서 무시하거나 관리하지 않도록 하려는 템플릿입니다.

사전 요구 사항

- OpenShift Server에 액세스할 수 있습니다.
- **registry.redhat.io** 에 대한 인증을 구성했습니다.

절차

1. OpenShift 4 Server를 시작합니다.
2. 관리자로 OpenShift 서버에 로그인합니다.

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

3. docker-registry 시크릿을 생성한 프로젝트를 사용하고 있는지 확인합니다.

```
oc project openshift
```

4. Samples Operator의 현재 구성을 확인합니다.

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

5. 추가된 fuse 템플릿 및 이미지 스트림을 무시하도록 Samples Operator를 구성합니다.

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

6. Fuse 이미지 스트림 Skipped Imagestreams 섹션을 추가하고 Fuse 및 Spring Boot 2 템플릿을 Skipped Templates 섹션에 추가합니다.

```
[...]
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. OpenShift 이미지 스트림에서 Fuse를 업그레이드합니다.

```
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. OpenShift 빠른 시작 템플릿에서 Fuse를 업그레이드:

```
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. Upgrade Spring Boot 2 빠른 시작 템플릿:

```
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. (선택 사항) OpenShift 템플릿에서 업그레이드된 Fuse를 확인합니다.

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

2.6. OPENSIFT 4.X에서 FUSE CONSOLE의 성능 튜닝

기본적으로 Fuse Console은 다음 Nginx 설정을 사용합니다.

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

참고: 이러한 설정에 대한 설명은 Nginx 문서를 참조하십시오. <http://nginx.org/en/docs/dirindex.html>

Fuse Console의 성능을 조정하려면 **clientBodyBufferSize**, **proxyBuffers**, **subrequestOutputBufferSize** 환경 변수를 설정할 수 있습니다. 예를 들어 Fuse Console을 사용하여 수 많은 Pod 및 경로를 모니터링하는 경우(예: 총 100개 경로) Fuse Console의 **하위 requestOutputBufferSize** 환경 변수를 **60m** 에서 **100m** 사이로 설정하여 로드 시간 초과 문제를 해결할 수 있습니다.

이러한 환경 변수를 설정하는 방법은 Openshift 4.x에 Fuse 콘솔을 설치하는 방법에 따라 다릅니다.

- Fuse Console Operator 사용
- Fuse Console 템플릿 사용

2.6.1. Fuse Console Operator 설치를 위한 성능 튜닝

Openshift 4.x에서는 Fuse 콘솔을 배포하기 전이나 후에 Nginx 성능 튜닝 환경 변수를 설정할 수 있습니다. 나중에 이렇게 하면 OpenShift에서 Fuse Console을 재배포합니다.

사전 요구 사항

- OpenShift 클러스터에 대한 클러스터 관리자 액세스 권한이 있어야 합니다.
- [OperatorHub](#)를 사용하여 OpenShift 4.x에 Fuse 콘솔 설치 및 배포에 설명된 대로 [Fuse Console Operator](#)를 설치했습니다.

절차

Fuse 콘솔을 배포하기 전이나 후에 환경 변수를 설정할 수 있습니다.

- Fuse 콘솔을 배포하기 전에 환경 변수를 설정하려면 다음을 수행합니다.
 1. OpenShift 웹 콘솔의 Fuse Console Operator가 설치된 프로젝트에서 Operator > **Installed Operators** > **Red Hat Integration - Fuse Console** 을 선택합니다.
 2. **Hawtio** 탭을 클릭한 다음 **Create Hawtio** 를 클릭합니다.
 3. **Create Hawtio** 페이지, **양식 보기** 의 구성 > **Nginx** 섹션까지 아래로 스크롤합니다.
 4. **Nginx** 섹션을 확장한 다음 환경 변수를 설정합니다. 예를 들면 다음과 같습니다.
 - **clientBodyBufferSize**: 256k
 - **proxyBuffers**: 16 128k
 - **subrequestOutputBufferSize**: 100m
 5. 설정을 저장합니다.
 6. **생성** 을 클릭하여 Fuse 콘솔을 배포합니다.
 7. 배포가 완료되면 **Deployments** > **fuse-console** 페이지를 열고 **환경** 을 클릭하여 환경 변수가 목록에 있는지 확인합니다.
- Fuse 콘솔을 배포한 후 환경 변수를 설정하려면 다음을 수행합니다.
 1. OpenShift 웹 콘솔에서 Fuse Console이 배포된 프로젝트를 엽니다.
 2. **Operators** > 설치된 **Operators** > **Red Hat Integration - Fuse Console** 을 선택합니다.

3. **Hawtio** 탭을 클릭한 다음 **fuse-console** 을 클릭합니다.
4. 작업>**Edit Hawtio** 를 선택합니다.
5. 편집기 창에서 사양 섹션까지 아래로 스크롤합니다.
6. **spec** 섹션에서 새 **nginx** 섹션을 추가하고 하나 이상의 환경 변수를 지정합니다. 예를 들면 다음과 같습니다.

```

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

```

7. 저장을 클릭합니다.
OpenShift는 **Fuse Console**을 재배포합니다.
8. 재배포가 완료되면 워크로드>배포>**fuse-console** 페이지를 연 다음 환경을 클릭하여 목록의 환경 변수를 확인합니다.

2.6.2. Fuse Console 템플릿 설치를 위한 성능 튜닝

OpenShift 4.x에서는 Fuse 콘솔을 배포하기 전이나 후에 Nginx 성능 튜닝 환경 변수를 설정할 수 있습니다. 나중에 이렇게 하면 OpenShift에서 Fuse Console을 재배포합니다.

사전 요구 사항

- OpenShift 클러스터에 대한 클러스터 관리자 액세스 권한이 있어야 합니다.
- OpenShift 4.x 서버에 Fuse 이미지 스트림 설치 및 템플릿에 설명된 대로 OpenShift에 Fuse Console 템플릿을 설치했습니다.

절차

Fuse 콘솔을 배포하기 전이나 후에 환경 변수를 설정할 수 있습니다.

- Fuse 콘솔을 배포하기 전에 환경 변수를 설정하려면 다음을 수행합니다.
 1. 사용할 Fuse Console 템플릿을 확인합니다.
 - 클러스터 템플릿(**fuse-console-cluster-os4.json**)
 - 구성 가능한 RBAC가 있는 클러스터 템플릿(**fuse-console-cluster-rbac.yml**)
 - 네임스페이스 템플릿(**fuse-console-namespace-os4.json**)
 - 구성 가능한 RBAC가 있는 네임스페이스 템플릿(**fuse-console-namespace-rbac.yml**)

- 다음 예와 같이 `NGINX_CLIENT_BODY_BUFFER_SIZE`, `NGINX_PROXY_BUFFERS` 및/또는 `NGINX_SUBREQUEST_OUTPUT_BUFFER_SIZE` 환경 변수를 포함하도록 Fuse Console 템플릿의 로컬 사본을 편집합니다.

```

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

```

- 변경 사항을 저장하십시오.
 - [OpenShift 4.x에서 Fuse Console 설정에 설명된 대로 Fuse Console을 설치하고](#) 배포하는 단계를 따르십시오.
- Fuse 콘솔을 배포한 후 환경 변수를 설정하려면 다음을 수행합니다.

- 터미널 창에서 OpenShift 클러스터에 로그인합니다.
- Fuse Console이 배포된 프로젝트를 엽니다. 예를 들어 Fuse Console이 `myfuse` 프로젝트에 배포된 경우 다음 명령을 사용합니다.

```
oc project myfuse
```

- Fuse Console 배포의 이름을 가져옵니다.

```
oc get deployments
```

이 명령은 현재 프로젝트에서 실행 중인 배포 목록을 반환합니다. 예를 들면 다음과 같습니다.

```

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

```

- 다음 명령 중 하나 이상을 실행하여 Fuse Console 배포의 환경 변수를 설정합니다.

```
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"
```

OpenShift는 Fuse Console을 재배포합니다.

- 재배포가 완료되면 환경 변수 설정을 확인합니다.
 - Fuse Console Pod 이름을 가져옵니다.

```
oc get pods
```

- b. 다음 명령을 실행하여 환경 설정 보기

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

예를 들어 Pod 이름이 **fuse-console-6646cbbd4c-9rplg** 이면 다음 명령을 실행합니다.

```
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 Console에서 애플리케이션을 볼 수 있는 성능 튜닝

Fuse 콘솔의 성능 튜닝 기능을 통해 많은 수의 **Cryostat**로 애플리케이션을 볼 수 있습니다. 이 기능을 사용하려면 다음 단계를 수행합니다.

사전 요구 사항

- OpenShift 클러스터에 대한 클러스터 관리자 액세스 권한이 있어야 합니다.
- [OperatorHub](#)를 사용하여 OpenShift 4.x에 Fuse 콘솔 설치 및 배포에 설명된 대로 **Fuse Console Operator**를 설치했습니다.

절차

1. 애플리케이션의 메모리 제한을 늘립니다.

Fuse 콘솔에 도달하기 전에 애플리케이션이 OOM 오류와 충돌하지 않도록 메모리 제한을 256Mi에서 512Mi로 늘려야 합니다. Fuse quickstart의 경우 애플리케이션의 **src/main/kube/deployment.yml** 파일을 편집합니다.

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

2. **Fuse Console Deployment** 또는 **DeploymentConfig**에 충분한 메모리 제한이 있는지 확인합니다. 충분하지 않은 경우 제한을 200Mi에서 512Mi로 늘립니다.
3. **nginx** 로그에 "too big subrequest 응답이" 오류가 표시되면 [2.6.1절. "Fuse Console Operator 설치를 위한 성능 튜닝"](#) 섹션에 언급된 솔루션을 적용합니다.

3장. OPENSIFT 3.11에서 FUSE 콘솔 설정

OpenShift 3.11에서는 Fuse Console에 액세스할 수 있습니다.

- 프로젝트에서 실행 중인 모든 Fuse 컨테이너를 모니터링할 수 있도록 OpenShift 프로젝트에 Fuse Console을 추가하면 됩니다.
- 클러스터의 모든 프로젝트에서 실행 중인 모든 Fuse 컨테이너를 모니터링할 수 있도록 OpenShift 클러스터에 Fuse 콘솔을 추가하면 됩니다.
- 실행 중인 단일 Fuse 컨테이너를 모니터링할 수 있도록 특정 Fuse Pod에서 파일을 엽니다.

명령줄에서 Fuse Console 템플릿을 배포합니다.



참고

Minishift 또는 CDK 기반 환경에 Fuse 콘솔을 설치하려면 아래 KCS 문서에 설명된 단계를 따르십시오.

- Minishift 또는 CDK 기반 환경에 Fuse Console을 설치하려면 [KCS 4998441](#)을 참조하십시오.
- Jolokia 인증을 비활성화해야 하는 경우 [KCS 3988671](#)에 설명된 해결방법을 참조하십시오.

사전 요구 사항

- OpenShift의 Fuse에 설명된 대로 OpenShift 이미지 스트림에 Fuse 및 [Fuse 콘솔 템플릿을 설치합니다](#).



참고

- Fuse Console의 사용자 관리는 OpenShift에서 처리합니다.
- 배포 후 Fuse Console에 액세스하는 사용자의 경우 OpenShift 3.11의 Fuse에서 역할 기반 액세스 제어를 사용할 수 없습니다.

[3.1절. "OpenShift 3.11에 Fuse 콘솔 배포"](#)

[3.2절. "OpenShift 3.11의 Fuse Console에서 단일 Fuse Pod 모니터링"](#)

3.1. OPENSIFT 3.11에 FUSE 콘솔 배포

표 3.1. "Fuse 콘솔 템플릿" Fuse 애플리케이션 배포 유형에 따라 명령줄에서 Fuse Console을 배포하는 데 사용할 수 있는 OpenShift 3.11 템플릿에 대해 설명합니다.

표 3.1. Fuse 콘솔 템플릿

유형	설명
----	----

유형	설명
fis-console-cluster-template.json	Fuse Console은 여러 네임스페이스 또는 프로젝트에 배포된 Fuse 애플리케이션을 검색하고 연결할 수 있습니다. 이 템플릿을 배포하려면 OpenShift cluster-admin 역할이 있어야 합니다.
fis-console-namespace-template.json	이 템플릿은 현재 OpenShift 프로젝트(네임스페이스)에 대한 Fuse Console 액세스를 제한하므로 단일 테넌트 배포 역할을 합니다. 이 템플릿을 배포하려면 현재 OpenShift 프로젝트에 대한 admin 역할이 있어야 합니다.

선택적으로 다음 명령을 실행하여 모든 템플릿에 대한 매개변수 목록을 볼 수 있습니다.

```
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
```



참고

Fuse Console 템플릿은 브라우저에서 클러스터 내 서비스로의 보안 엔드 투 엔드 요청이 되도록 기본적으로 엔드 투 엔드 암호화를 구성합니다.

사전 요구 사항

- OpenShift 3.11의 클러스터 모드인 경우 클러스터 관리자 역할 및 클러스터 모드 템플릿이 필요합니다. 다음 명령을 실행합니다.

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

절차

명령줄에서 Fuse 콘솔을 배포하려면 다음을 수행합니다.

1. 다음 명령 중 하나를 실행하여 Fuse Console 템플릿을 기반으로 새 애플리케이션을 생성합니다(여기서 myproject는 프로젝트 이름임).

- Fuse Console 클러스터 템플릿의 경우 myhost는 Fuse Console에 액세스할 수 있는 호스트 이름입니다.

```
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 네임스페이스 템플릿의 경우:

```
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
```



참고

OpenShift가 자동으로 생성되므로 네임스페이스 템플릿의 `route_hostname` 매개변수를 생략할 수 있습니다.

2. 다음 명령을 실행하여 Fuse Console 배포의 상태 및 URL을 가져옵니다.

```
oc status
```

3. 브라우저에서 Fuse Console에 액세스하려면 제공된 URL을 사용합니다.

예제:

[+https://fuse-console.192.168.64.12.nip.io](https://fuse-console.192.168.64.12.nip.io).

3.2. OPENSIFT 3.11의 FUSE CONSOLE에서 단일 FUSE POD 모니터링

OpenShift 3.11에서 실행되는 Fuse Pod의 Fuse Console을 열 수 있습니다.

사전 요구 사항

- 포트 뷰에 Fuse Console에 대한 링크를 표시하도록 OpenShift를 구성하려면 OpenShift 이미지에서 Fuse를 실행하는 Pod에서 `jolokia` 로 설정된 name 속성 내에서 TCP 포트를 선언해야 합니다.

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

절차

1. OpenShift 프로젝트의 애플리케이션 → 포트 보기에서 Pod이름을 클릭하여 실행 중인 Fuse Pod의 세부 정보를 확인합니다. 이 페이지 오른쪽에는 컨테이너 템플릿에 대한 요약이 표시됩니다.

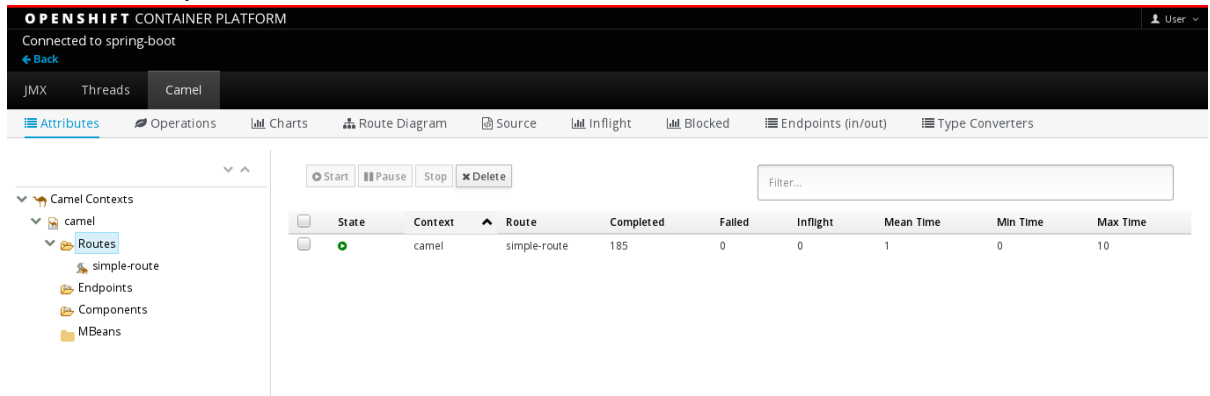
Template

Containers

CONTAINER: SPRING-BOOT

-  **Image:** [test/fuse70-spring-boot](#) eda527f 193.1 MiB
 -  **Build:** [fuse70-spring-boot-s2i, #2](#)
 -  **Source:** Binary
 -  **Ports:** 8080/TCP (http), 8778/TCP (jolokia), 9779/TCP (prometheus)
 -  **Mount:** default-token-p4zsn → /var/run/secrets/kubernetes.io/serviceaccount
read-only
 -  **CPU:** 200 millicores to 1 core
 -  **Readiness Probe:** GET /health on port 8081 (HTTP) 10s delay, 1s timeout
 -  **Liveness Probe:** GET /health on port 8081 (HTTP) 180s delay, 1s timeout
-  [Open Java Console](#)

2. 이 보기에서 **Open Java Console** 링크를 클릭하여 Fuse 콘솔을 엽니다.



The screenshot shows the OpenShift Container Platform console interface. The top navigation bar includes 'Attributes', 'Operations', 'Charts', 'Route Diagram', 'Source', 'Inflight', 'Blocked', 'Endpoints (in/out)', and 'Type Converters'. The left sidebar shows a tree view of Camel contexts, with 'Routes' expanded to show 'simple-route'. The main content area displays a table of routes with the following data:

State	Context	Route	Completed	Failed	Inflight	Mean Time	Min Time	Max Time
●	camel	simple-route	185	0	0	1	0	10

4장. 컨테이너 및 애플리케이션 보기

OpenShift용 Fuse Console에 로그인하면 Fuse Console 홈 페이지에 사용 가능한 컨테이너가 표시됩니다.

절차

- 컨테이너를 관리(생성, 편집 또는 삭제)하려면 OpenShift 콘솔을 사용합니다.
- OpenShift 클러스터에서 Fuse 애플리케이션 및 AMQ 브로커(해당되는 경우)를 보려면 온라인 탭을 클릭합니다.

5장. APACHE CAMEL 애플리케이션 보기 및 관리

Fuse 콘솔의 Camel 탭에서 Apache Camel 컨텍스트, 경로 및 종속성을 보고 관리합니다.

다음 세부 정보를 볼 수 있습니다.

- 실행 중인 모든 Camel 컨텍스트 목록
- Camel 버전 번호 및 런타임 정적과 같은 각 Camel 컨텍스트에 대한 자세한 정보
- 각 Camel 애플리케이션의 모든 경로 목록 및 런타임 통계
- 실행 중인 경로의 그래픽 표현과 실시간 메트릭

다음을 통해 Camel 애플리케이션과 상호 작용할 수도 있습니다.

- 컨텍스트 시작 및 일시 중단
- 모든 Camel 애플리케이션 및 해당 경로의 라이프사이클을 관리하여 재시작, 중지, 일시 중지, 재개 등을 수행할 수 있습니다.
- 실행 중인 경로의 실시간 추적 및 디버깅
- Camel 엔드포인트에 메시지 검색 및 전송

사전 요구 사항

Camel 탭은 하나 이상의 Camel 경로를 사용하는 컨테이너에 연결할 때만 사용할 수 있습니다.

5.1. 컨텍스트 시작, 일시 중지 또는 삭제

1. Camel 탭의 트리 보기에서 Camel Contexts 를 클릭합니다.
2. 목록에서 하나 이상의 컨텍스트 옆에 있는 확인란을 선택합니다.
3. 시작 또는 일시 중지 를 클릭합니다.
4. 컨텍스트를 삭제하려면 다음을 수행합니다.
 - a. 컨텍스트를 중지합니다.
 - b. 아이콘을 클릭한 다음 드롭다운 메뉴에서 삭제 를 선택합니다.



참고

컨텍스트를 삭제하면 배포된 애플리케이션에서 해당 컨텍스트를 제거합니다.

5.2. CAMEL 애플리케이션 세부 정보 보기

1. Camel 탭의 트리 뷰에서 Camel 애플리케이션을 클릭합니다.
2. 애플리케이션 특성 및 값 목록을 보려면 속성을 클릭합니다.
3. 애플리케이션 특성의 그래픽 표시를 보려면 차트 를 클릭한 다음 편집을 클릭하여 차트에서 볼 속성을 선택합니다.

4. 진행 중 및 차단된 교환을 보려면 교환을 클릭합니다.
5. 애플리케이션 엔드포인트를 보려면 끝점을 클릭합니다. URL, 경로 ID 및 방향으로 목록을 필터링할 수 있습니다.
6. 메시지 본문 및 메시지 헤더를 다른 유형으로 변환하는 데 사용되는 Camel 기본 제공 유형 변환 메커니즘과 관련된 통계를 확인, 활성화 및 비활성화하려면 유형 다운로드를 클릭합니다.
7. XML에서 경로 추가 또는 업데이트 또는 classpath에서 사용 가능한 모든 Camel 구성 요소를 찾는 등 Cystat 작업을 보고 실행하려면 Operations 를 클릭합니다.

5.3. CAMEL 경로 목록 보기 및 상호 작용

1. 경로 목록을 보려면 다음을 수행합니다.
 - a. Camel 탭을 클릭합니다.
 - b. 트리 뷰에서 애플리케이션의 경로 폴더를 클릭합니다.

Routes

Start Stop ⋮		
<input type="checkbox"/>	Name ^	State
<input type="checkbox"/>	_route1	Started
<input type="checkbox"/>	_route2	Started

2. 하나 이상의 경로를 시작, 중지 또는 삭제하려면 다음을 수행합니다.
 - a. 목록에서 하나 이상의 경로 옆에 있는 확인란을 선택합니다.
 - b. 시작 또는 중지를 클릭합니다.
 - c. 경로를 삭제하려면 먼저 중지해야 합니다. 그런 다음 아이콘을 클릭하고 드롭다운 메뉴에서 삭제를 선택합니다.

Routes

Start Stop ⋮		
<input checked="" type="checkbox"/>	Name ^	Delete
<input checked="" type="checkbox"/>		



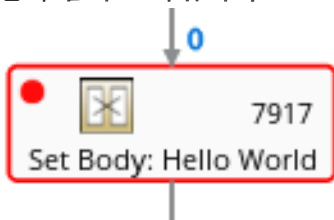
참고

- 경로를 삭제하면 배포된 애플리케이션에서 해당 경로를 제거합니다.
- 트리 보기에서 특정 경로를 선택한 다음 오른쪽 상단 메뉴를 클릭하여 시작, 중지 또는 삭제할 수도 있습니다.

3. 경로의 그래픽 다이어그램을 보려면 **Route Diagram** 을 클릭합니다.
4. 진행 중 및 차단된 교환을 보려면 **교환**을 클릭합니다.
5. 엔드포인트를 보려면 끝점을 클릭합니다. URL, 경로 ID 및 방향별로 목록을 필터링할 수 있습니다.
6. 메시지 본문 및 메시지 헤더를 다른 유형으로 변환하는 데 사용되는 Camel 기본 제공 유형 변환 메커니즘과 관련된 통계를 확인, 활성화 및 비활성화하려면 **Type Cryostat**를 클릭합니다.
7. 특정 경로와 상호 작용하려면 다음을 수행합니다.
 - a. Camel 탭의 트리 보기에서 경로를 선택합니다.
 - b. 경로 속성 및 값 목록을 보려면 속성을 클릭합니다.
 - c. 경로 속성의 그래픽 표시를 보려면 차트를 클릭합니다. 편집을 클릭하여 차트에서 표시할 속성을 선택할 수 있습니다.
 - d. 진행 중 및 차단된 교환을 보려면 **교환**을 클릭합니다.
 - e. 작업에서 경로를 XML로 덤프하거나 경로의 Camel ID 값을 가져오는 등 경로에서 **Cryostat** 작업을 보고 실행합니다.
8. 경로를 통해 메시지를 추적하려면 다음을 수행합니다.
 - a. Camel 탭의 트리 보기에서 경로를 선택합니다.
 - b. 추적 을 선택한 다음추적 시작을 클릭합니다.
9. 경로에 메시지를 보내려면 다음을 수행합니다.
 - a. Camel 탭의 트리 뷰에서 컨텍스트의 엔드포인트 폴더를 연 다음 엔드포인트를 선택합니다.
 - b. **Send** 를 클릭합니다.
 - c. 메시지를 JSON 또는 XML 형식으로 구성합니다.
 - d. 전송을 클릭합니다.
 - e. 경로의 추적 탭으로 돌아가 경로를 통한 메시지 흐름을 확인합니다.

5.4. 경로 디버깅

1. Camel 탭의 트리 보기에서 경로를 선택합니다.
2. 디버그를 선택한 다음디버깅 시작을 클릭합니다.
3. **Cryostat**를 추가하려면 다이어그램에서 노드를 선택한 다음 **Add Cryostat**를 클릭합니다. 노드에 빨간색 점이 표시됩니다.



노드가 Cryostat 목록에 추가됩니다.

Breakpoints	
setBody1	×
log1	×

4. 아래쪽 화살표를 클릭하여 다음 노드 또는 재생 버튼으로 이동하여 경로 실행을 다시 시작합니다.
5. 일시 중지 버튼을 클릭하여 경로에 대한 모든 스레드를 일시 중지합니다.
6. 완료되면 디버깅 중지를 클릭합니다. 모든 Cryostat가 지워집니다.

6장. AMQ 브로커 보기

OpenShift 클러스터에 배포된 모든 AMQ 브로커를 확인하도록 Fuse 콘솔을 구성할 수 있습니다.

사전 요구 사항

Fuse Console에서 보고 싶은 각 AMQ 브로커 이미지는 다음과 같아야 합니다.

- Fuse Console이 설치된 동일한 OpenShift 클러스터에 설치됩니다.
- AMQ Broker 설명서의 Fuse Console의 [Artemis 플러그인 활성화에 대한 섹션에 설명된 대로 Fuse Console](#)이 이를 인식하고 연결할 수 있도록 구성되었습니다.

절차

- Artemis를 클릭하여 AMQ 관리 콘솔을 보고 AMQ Broker의 상태를 모니터링합니다. (AMQ Broker는 [Apache ActiveMQ Artemis](#)를 기반으로 합니다.)

AMQ 관리 콘솔 사용에 대한 자세한 내용은 [AMQ Broker 관리 가이드의 AMQ 관리 콘솔 사용](#)을 참조하십시오.

7장. CRYOSTAT 도메인 및 CRYOSTAT 보기 및 관리

JMX(Java Management Extensions)는 런타임 시 리소스(서비스, 장치 및 애플리케이션)를 동적으로 관리할 수 있는 Java 기술입니다. 리소스는 Cryostats라는 오브젝트로 표시됩니다(관리됨의 경우). 리소스를 생성, 구현 또는 설치하는 즉시 리소스를 관리하고 모니터링할 수 있습니다.

Fuse Console에서 Cryostat 플러그인을 사용하면 Cryostat 도메인 및 Cryostat를 보고 관리할 수 있습니다. Cryostat 특성을 보고, 명령을 실행하고, 통계를 표시하는 차트를 만들 수 있습니다.

Cryostat 탭에서는 폴더에 구성된 활성 Cryostat 도메인 및 Cryostat의 트리 뷰를 제공합니다. Cryostat에서 세부 정보를 보고 명령을 실행할 수 있습니다.

절차

1. Cryostat 특성을 보고 편집하려면 다음을 수행합니다.
 - a. 트리 보기에서 na를 선택합니다. In the tree view, select an value.
 - b. 특성 탭을 클릭합니다.
 - c. 속성을 클릭하여 세부 정보를 확인합니다.
2. 작업을 수행하려면 다음을 수행합니다.
 - a. 트리 보기에서 na를 선택합니다. In the tree view, select an value.
 - b. Operations 탭을 클릭하고 나열된 작업 중 하나를 확장합니다.
 - c. 실행을 클릭하여 작업을 실행합니다.
3. 차트를 보려면 다음을 수행합니다.
 - a. 트리 뷰에서 항목을 선택합니다.
 - b. 차트 탭을 클릭합니다.

8장. QUARTZ 일정 보기 및 관리

Quartz(<http://www.quartz-scheduler.org/>)는 대부분의 Java 애플리케이션 내에서 통합할 수 있는 풍부한 기능을 갖춘 오픈 소스 작업 스케줄링 라이브러리입니다. Quartz를 사용하여 작업 실행을 위한 단순 또는 복잡한 스케줄을 생성할 수 있습니다. 작업은 프로그래밍할 거의 모든 작업을 실행할 수 있는 표준 Java 구성 요소로 정의됩니다.

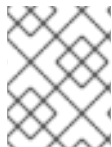
Camel 경로가 `camel-quartz2` 구성 요소를 배포하는 경우 Fuse Console은 Quartz 탭을 표시합니다. Cryostat 트리 보기를 통해 Quartz mbeans에 대체할 수 있습니다.

절차

1. Fuse 콘솔에서 Quartz 탭을 클릭합니다.
Quartz 페이지에는 Quartz 스케줄러 및 스케줄러 트리거 및 작업 탭의 트리 뷰가 포함되어 있습니다.
2. 스케줄러를 일시 중지하거나 시작하려면 스케줄러 탭에서 버튼을 클릭합니다.
3. 트리거 탭을 클릭하여 작업이 실행될 시기를 결정하는 트리거를 확인합니다. 예를 들어, 트리거는 지정된 요일에 (마이크로초) 특정 시간에 작업을 시작하도록 지정하거나 지정된 일수 또는 특정 시간에 반복하도록 지정할 수 있습니다.
 - 트리거 목록을 필터링하려면 드롭다운 목록에서 State, Group, Name 또는 Type 을 선택합니다. 그런 다음 fill-on 필드를 선택하거나 입력하여 목록을 추가로 필터링할 수 있습니다.
 - 트리거를 일시 중지, 재개, 업데이트 또는 수동으로 실행하려면 Action 열에서 옵션을 클릭합니다.
4. 작업 탭을 클릭하여 실행 중인 작업 목록을 확인합니다. 표의 열로 목록을 정렬할 수 있습니다. Group, Name, Durable, Recover, Job ClassName, description .

9장. 진단 보기

JVM Diagnostic Command 및 CryostatDiagnostic 인터페이스를 통해 JVM에 대한 진단 정보를 보려면 Cryostat 탭을 사용합니다.



참고

기능은 jmc(Java Mission Control) 또는 명령줄 도구 jcmd의 진단 명령 보기와 유사합니다. 플러그인은 일부 시나리오에서 해당 jcmd 명령을 제공합니다.

절차

1. 로드된 클래스의 인스턴스 수와 필요한 바이트 수를 검색하려면 클래스 histogram을 클릭합니다. 작업이 반복되면 탭에 마지막 실행 이후의 차이점이 표시됩니다.
2. JVM 진단 플래그 설정을 보려면 JVM 플래그를 클릭합니다.
3. 실행 중인 JVM의 경우 플래그 설정도 수정할 수 있습니다.

추가 리소스

지원되는 JVM은 플랫폼에 따라 다릅니다. 자세한 내용은 다음 소스 중 하나로 이동합니다.

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

10장. 스레드 보기

스레드 상태를 보고 모니터링할 수 있습니다.

절차

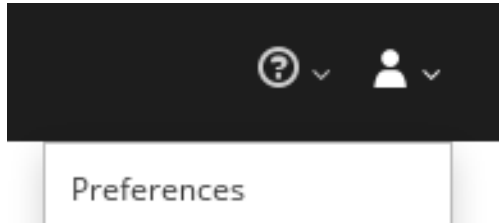
1. **Runtime** 탭을 클릭한 다음 **Threads** 를 클릭합니다. **Threads** 페이지에는 각 스레드에 대한 활성 스레드 및 스택 추적 세부 정보가 나열됩니다. 기본적으로 스레드 목록은 모든 스레드를 내림차순으로 표시합니다.
2. ID를 눌러 목록을 정렬하려면 ID 열 레이블을 클릭합니다.
3. 선택적으로 스레드 상태(예: **Blocked**) 또는 스레드 이름으로 목록을 필터링합니다.
4. 해당 스레드의 잠금 클래스 이름 및 전체 스택 추적과 같은 특정 스레드에 대한 자세한 정보를 드릴 다운하려면 **Actions** 열에서 **More** 를 클릭합니다.

11장. FUSE CONSOLE에 데이터가 올바르게 표시되도록 합니다.

Fuse Console의 대기열 및 연결 표시에 큐가 없거나 연결이 누락되었거나 일관성 없는 아이콘을 표시하는 경우 Jolokia 컬렉션 크기 매개변수를 조정하여 Jolokia marshals를 응답에 마샬링하는 배열의 최대 요소 수를 지정합니다.

절차

1. Fuse Console의 오른쪽 상단에서 사용자 아이콘을 클릭한 다음 기본 설정을 클릭합니다.



2. 최대 컬렉션 크기 옵션의 값을 늘립니다(기본값은 50,000).
3. 단기를 클릭합니다.

부록 A. FUSE 콘솔 구성 속성

기본적으로 Fuse Console 구성은 `hawtconfig.json` 파일에 정의됩니다. 제목, 로고 및 로그인 페이지 정보와 같은 Fuse Console 구성 정보를 사용자 지정할 수 있습니다.

표 A.1. "Fuse 콘솔 구성 속성" 각 속성에 값이 필요한지 여부에 관계없이 속성 및 목록에 대한 설명을 제공합니다.

표 A.1. Fuse 콘솔 구성 속성

섹션	속성 이름	기본값	설명	필수 여부
정보	제목	Red Hat Fuse Management Console	Fuse Console의 정보 페이지에 표시되는 제목입니다.	필수 항목
	productInfo	빈 값	Fuse 콘솔의 정보 페이지에 표시되는 제품 정보입니다.	선택 사항
	additionalInfo	빈 값	Fuse Console의 정보 페이지에 표시되는 추가 정보입니다.	선택 사항
	저작권	빈 값	Fuse Console의 정보 페이지에 표시되는 저작권 정보.	선택 사항
	imgSrc	img/Logo-RedHat-A-Reverse- RGB.png	Fuse 콘솔의 정보 페이지에 표시되는 이미지입니다.	필수 항목
브랜딩	appName	Red Hat Fuse Management Console	애플리케이션 이름입니다. 이 이름은 Fuse Console의 제목 표시줄에 표시됩니다.	필수 항목
	appLogoUrl	img/Logo-Red_Hat-Fuse- A-Reverse- RGB.png	Fuse 콘솔 {navigation 표시줄에 표시되는 애플리케이션 로고 이미지 파일의 경로입니다. 값은 Hawtio 상태 URL 또는 절대 URL을 기준으로 하는 경로일 수 있습니다.	필수 항목

섹션	속성 이름	기본값	설명	필수 여부
	CSS		애플리케이션의 스타일을 지정하는 데 사용할 수 있는 외부 CSS 스타일시트의 URL입니다. Hawtio 상태 URL을 기준으로 하는 경로이거나 절대 URL일 수 있습니다.	선택 사항
	companyLogoUrl	img/Logo-RedHat-A-Reverse- RGB.png	회사 로고 이미지 파일의 경로입니다.	필수 항목
	favicon		일반적으로 웹 브라우저 탭에 표시되는 favicon의 URL입니다. Hawtio 상태 URL을 기준으로 하는 경로이거나 절대 URL일 수 있습니다.	선택 사항
login	description	<i>빈 값</i>	Fuse 콘솔 로그인 페이지에 표시되는 설명 텍스트(예: http://localhost:8181/hawtio).	선택 사항
	링크	[]	"url" 및 "text" 쌍의 배열을 지정하여 사용자가 더 많은 정보 또는 도움말을 가져올 수 있는 페이지에 대한 추가 링크를 제공합니다.	선택 사항
disabledRoutes	<i>none</i>	[]	콘솔에서 특정 경로(예: 플러그인)를 비활성화합니다. 이 섹션을 변경하지 마십시오. OpenShift 이외의 배포에서는 변경 사항이 지원되지 않습니다.	선택 사항

