



Red Hat OpenShift Serverless 1.28

Serverless 설치

Serverless Operator, Knative CLI, Knative Serving, Knative Eventing 설치

Red Hat OpenShift Serverless 1.28 Serverless 설치

Serverless Operator, Knative CLI, Knative Serving, Knative Eventing 설치

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

이 문서에는 Serverless Operator, Knative CLI, Serving, Eventing 설치에 대한 정보가 포함되어 있습니다. 또한 Apache Kafka용 Knative 구성 및 Serverless Functions 구성에 대한 세부 정보도 제공합니다.

차례

1장. OPENSIFT SERVERLESS 설치 준비	3
1.1. 지원되는 구성	3
1.2. OPENSIFT CONTAINER PLATFORM의 확장성 및 성능	3
1.3. 클러스터 크기 요구 사항 정의	3
1.4. OPENSIFT CONTAINER PLATFORM에서 컴퓨팅 머신 세트를 사용하여 클러스터 스케일링	4
1.5. OPENSIFT CONTAINER PLATFORM 문서의 추가 리소스	4
2장. OPENSIFT SERVERLESS OPERATOR 설치	5
2.1. 웹 콘솔에서 OPENSIFT SERVERLESS OPERATOR 설치	5
2.2. CLI에서 OPENSIFT SERVERLESS OPERATOR 설치	6
2.3. 글로벌 구성	7
2.4. OPENSIFT CONTAINER PLATFORM의 추가 리소스	8
2.5. 다음 단계	8
3장. KNATIVE CLI 설치	9
3.1. OPENSIFT CONTAINER PLATFORM 웹 콘솔을 통한 KNATIVE CLI 설치	9
3.2. RPM 패키지 관리자를 사용하여 LINUX용 KNATIVE CLI 설치	10
3.3. LINUX의 KNATIVE CLI 설치	11
3.4. MACOS용 KNATIVE CLI 설치	12
3.5. WINDOWS용 KNATIVE CLI 설치	12
4장. KNATIVE SERVING 설치	14
4.1. 웹 콘솔을 사용하여 KNATIVE SERVING 설치	14
4.2. YAML을 사용하여 KNATIVE SERVING 설치	16
4.3. 다음 단계	17
5장. KNATIVE EVENTING 설치	18
5.1. 웹 콘솔을 사용하여 KNATIVE EVENTING 설치	18
5.2. YAML을 사용하여 KNATIVE EVENTING 설치	20
5.3. APACHE KAFKA용 KNATIVE 브로커 설치	21
5.4. 다음 단계	24
6장. APACHE KAFKA의 KNATIVE 브로커 구성	25
7장. APACHE KAFKA의 KNATIVE용 KUBE-RBAC-PROXY 구성	26
7.1. APACHE KAFKA용 KNATIVE에 대한 KUBE-RBAC-PROXY 리소스 구성	26
8장. OPENSIFT SERVERLESS FUNCTIONS 구성	27
8.1. 사전 요구 사항	27
8.2. PODMAN 설정	27
8.3. MACOS에서 PODMAN 설정	28
8.4. 다음 단계	29
9장. 서버리스 업그레이드	30
9.1. OPENSIFT SERVERLESS OPERATOR 업그레이드 실패 확인	30

1장. OPENSIFT SERVERLESS 설치 준비

OpenShift Serverless를 설치하기 전에 지원되는 구성 및 사전 요구 사항에 대한 다음 정보를 읽으십시오.

OpenShift Container Platform의 경우:

- OpenShift Serverless는 제한된 네트워크 환경에서 설치할 수 있습니다.
- OpenShift Serverless는 현재 단일 클러스터의 다중 테넌트 구성에 사용할 수 없습니다.

1.1. 지원되는 구성

OpenShift Serverless에 지원되는 일련의 기능, 구성, 통합과 현재 버전 및 이전 버전은 [지원되는 구성 페이지](#)에서 확인할 수 있습니다.

1.2. OPENSIFT CONTAINER PLATFORM의 확장성 및 성능

OpenShift Serverless는 3개의 기본 노드와 작업자 노드 3개로 테스트되었으며 각 노드에는 64개의 CPU, 457GB의 메모리, 각각 394GB의 스토리지가 있습니다.

이 구성을 사용하여 생성할 수 있는 최대 Knative 서비스 수는 3000입니다. 이는 1개의 Knative 서비스가 3개의 Kubernetes 서비스를 생성하므로 [OpenShift Container Platform Kubernetes 서비스 제한 10,000](#)에 해당합니다.

응답 시간의 평균 스케일은 약 3.4초이며 최대 응답 시간은 8초이고 간단한 Quarkus 애플리케이션의 경우 4.5초입니다. 이러한 시간은 애플리케이션 및 애플리케이션 런타임에 따라 다를 수 있습니다.



참고

클러스터 크기 요구 사항을 정의하는 다음 섹션에서는 이러한 배포에 적용됩니다.

- OpenShift Container Platform
- OpenShift Dedicated
- Red Hat OpenShift Service on AWS

1.3. 클러스터 크기 요구 사항 정의

OpenShift Serverless를 설치하고 사용하려면 클러스터의 크기를 올바르게 조정해야 합니다.



참고

다음 요구 사항은 클러스터의 작업자 머신 풀에만 관련이 있습니다. 컨트롤 플레인 노드는 일반 스케줄링에 사용되지 않으며 요구 사항에서 생략됩니다.

OpenShift Serverless를 사용하기 위한 최소 요구 사항은 CPU 10개 및 40GB 메모리가 있는 클러스터입니다. 기본적으로 각 Pod는 ~400m의 CPU를 요청하므로 최소 요구 사항은 이 값을 기반으로 합니다.

OpenShift Serverless를 실행하는 총 크기 요구 사항은 설치된 구성 요소 및 배포된 애플리케이션에 따라 달라지며 배포에 따라 다를 수 있습니다.

1.4. OPENSIFT CONTAINER PLATFORM에서 컴퓨팅 머신 세트를 사용하여 클러스터 스케일링

OpenShift Container Platform **MachineSet** API를 사용하여 클러스터를 원하는 크기로 수동으로 확장할 수 있습니다. 최소 요구 사항은 일반적으로 기본 컴퓨팅 시스템 세트 중 하나를 두 개의 추가 머신으로 확장해야 함을 의미합니다. [컴퓨팅 머신 세트 수동 스케일링](#)을 참조하십시오.

1.4.1. 고급 사용 사례에 대한 추가 요구 사항

OpenShift Container Platform의 로깅 또는 미터링과 같은 고급 사용 사례의 경우 더 많은 리소스를 배포해야 합니다. 이러한 사용 사례에 권장되는 요구 사항은 CPU 24개 및 96GB의 메모리입니다.

클러스터에 HA(고가용성)를 활성화하면 Knative Serving 컨트롤 플레인을 복제할 때마다 0.5~1.5코어 및 200MB~2GB의 메모리가 필요합니다. HA는 기본적으로 일부 Knative Serving 구성 요소에 사용됩니다. "고가용 가용성 복제본 구성" 설명서에 따라 HA를 비활성화할 수 있습니다.

1.5. OPENSIFT CONTAINER PLATFORM 문서의 추가 리소스

- [제한된 네트워크에서 Operator Lifecycle Manager 사용](#)
- [OperatorHub 이해](#)
- [클러스터 기능](#)

2장. OPENSIFT SERVERLESS OPERATOR 설치

OpenShift Serverless Operator를 설치하면 OpenShift Container Platform 클러스터에서 Knative Serving, Knative Eventing 및 Apache Kafka용 Knative 브로커를 설치하고 사용할 수 있습니다. OpenShift Serverless Operator는 클러스터의 Knative CRD(사용자 정의 리소스 정의)를 관리하고 각 구성 요소의 개별 구성 맵을 직접 수정하지 않고도 이를 구성할 수 있습니다.

2.1. 웹 콘솔에서 OPENSIFT SERVERLESS OPERATOR 설치

OpenShift Container Platform 웹 콘솔을 사용하여 OperatorHub에서 OpenShift Serverless Operator를 설치할 수 있습니다. 이 Operator를 설치하면 Knative 구성 요소를 설치하고 사용할 수 있습니다.

사전 요구 사항

- OpenShift Container Platform에 대한 클러스터 관리자 권한이 있거나 AWS 또는 OpenShift Dedicated의 Red Hat OpenShift Service에 대한 클러스터 또는 전용 관리자 권한이 있습니다.
- OpenShift Container Platform의 경우 클러스터에 Marketplace 기능이 활성화되어 있거나 Red Hat Operator 카탈로그 소스가 수동으로 구성되어 있습니다.
- 웹 콘솔에 로그인했습니다.

절차

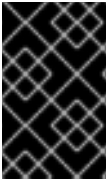
1. 웹 콘솔에서 **Operator → OperatorHub** 페이지로 이동합니다.
2. 스크롤하거나 키워드로 필터링 상자에 키워드 **서버리스**를 입력하여 OpenShift Serverless Operator를 찾습니다.
3. Operator에 대한 정보를 검토하고 **설치**를 클릭합니다.
4. **Operator 설치** 페이지에서 다음을 수행합니다.
 - a. **설치 모드**는 클러스터의 모든 네임스페이스(기본값)입니다. 이 모드에서는 클러스터의 모든 네임스페이스를 감시하고 사용할 수 있도록 기본 **openshift-serverless** 네임스페이스에 Operator를 설치합니다.
 - b. 설치된 네임스페이스는 **openshift-serverless**입니다.
 - c. **stable** 채널을 **업데이트 채널**로 선택합니다. **stable** 채널을 사용하면 안정적인 최신 릴리스의 OpenShift Pipelines Operator를 설치할 수 있습니다.
 - d. **자동** 또는 **수동 승인 전략**을 선택합니다.
5. 이 OpenShift Container Platform 클러스터에서 선택한 네임스페이스에서 Operator를 사용할 수 있도록 하려면 **설치**를 클릭합니다.
6. **카탈로그 → Operator 관리** 페이지에서 OpenShift Serverless Operator 서브스크립션 설치 및 업그레이드 진행 상황을 모니터링할 수 있습니다.
 - a. **수동 승인 전략**을 선택한 경우 설치 계획을 검토하고 승인할 때까지 서브스크립션의 업그레이드 상태가 **업그레이드 중**으로 유지됩니다. **설치 계획** 페이지에서 승인한 후 subscription 업그레이드 상태가 **최신**으로 이동합니다.
 - b. **자동 승인 전략**을 선택한 경우 업그레이드 상태가 개입 없이 **최신** 상태로 확인되어야 합니다.

검증

서브스크립션의 업데이트 상태가 최신인 경우 **카탈로그** → **설치된 Operator**를 선택하여 OpenShift Serverless Operator가 표시되고 관련 네임스페이스에서 **상태**가 최종적으로 **InstallSucceeded**로 표시되는지 확인합니다.

그러지 않은 경우 다음을 수행합니다.

1. **카탈로그** → **Operator 관리** 페이지로 전환한 후 **Operator 서브스크립션 및 설치 계획 탭의 상태**에 장애나 오류가 있는지 검사합니다.
2. **워크로드** → **Pod** 페이지의 **openshift-serverless** 프로젝트에서 문제를 보고하는 Pod의 로그를 확인하여 추가로 문제를 해결합니다.



중요

OpenShift **Serverless**에서 **Red Hat OpenShift distributed tracing**을 사용하려면 Knative Serving 또는 Knative Eventing을 설치하기 전에 Red Hat OpenShift distributed tracing을 설치하고 구성해야 합니다.

2.2. CLI에서 OPENSIFT SERVERLESS OPERATOR 설치

CLI를 사용하여 OperatorHub에서 OpenShift Serverless Operator를 설치할 수 있습니다. 이 Operator를 설치하면 Knative 구성 요소를 설치하고 사용할 수 있습니다.

사전 요구 사항

- OpenShift Container Platform에 대한 클러스터 관리자 권한이 있거나 AWS 또는 OpenShift Dedicated의 Red Hat OpenShift Service에 대한 클러스터 또는 전용 관리자 권한이 있습니다.
- OpenShift Container Platform의 경우 클러스터에 Marketplace 기능이 활성화되어 있거나 Red Hat Operator 카탈로그 소스가 수동으로 구성되어 있습니다.
- 클러스터에 로그인했습니다.

절차

1. **Namespace, OperatorGroup, Subscription** 오브젝트가 포함된 YAML 파일을 생성하여 네임스페이스를 OpenShift Serverless Operator에 등록합니다. 예를 들어 다음 콘텐츠를 사용하여 **서버리스-subscription.yaml** 파일을 생성합니다.

서브스크립션 예

```
---
apiVersion: v1
kind: Namespace
metadata:
  name: openshift-serverless
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: serverless-operators
  namespace: openshift-serverless
spec: {}
```

```

---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: serverless-operator
  namespace: openshift-serverless
spec:
  channel: stable ❶
  name: serverless-operator ❷
  source: redhat-operators ❸
  sourceNamespace: openshift-marketplace ❹

```

- ❶ Operator의 채널 이름입니다. **stable** 채널을 사용하면 안정적인 최신 버전의 OpenShift Serverless Operator를 설치할 수 있습니다.
- ❷ 등록할 Operator의 이름입니다. OpenShift Serverless Operator의 경우 항상 **서버리스-operator** 입니다.
- ❸ Operator를 제공하는 CatalogSource의 이름입니다. 기본 OperatorHub 카탈로그 소스로 **redhat-operators** 를 사용합니다.
- ❹ CatalogSource의 네임스페이스입니다. 기본 OperatorHub 카탈로그 소스에는 **openshift-marketplace**를 사용합니다.

2. Subscription 오브젝트를 생성합니다.

```
$ oc apply -f serverless-subscription.yaml
```

검증

CSV(클러스터 서비스 버전)가 **Succeeded** 단계에 도달했는지 확인합니다.

명령 예

```
$ oc get csv
```

출력 예

NAME	DISPLAY	VERSION	REPLACES	PHASE
serverless-operator.v1.25.0	Red Hat OpenShift Serverless	1.25.0	serverless-operator.v1.24.0	Succeeded



중요

OpenShift [Serverless](#)에서 [Red Hat OpenShift distributed tracing](#)을 사용하려면 Knative Serving 또는 Knative Eventing을 설치하기 전에 Red Hat OpenShift distributed tracing을 설치하고 구성해야 합니다.

2.3. 글로벌 구성

OpenShift Serverless Operator는 **KnativeServing** 및 **KnativeEventing** 사용자 정의 리소스에서 시스템 구성 맵으로 값 전파를 포함하여 Knative 설치의 글로벌 구성을 관리합니다. Operator에서 수동으로 적용

되는 구성 맵에 대한 업데이트를 덮어씁니다. 그러나 Knative 사용자 정의 리소스를 수정하면 이러한 구성 맵의 값을 설정할 수 있습니다.

Knative에는 접두사 **config-**로 이름이 지정된 여러 구성 맵이 있습니다. 모든 Knative 구성 맵은 적용되는 사용자 정의 리소스와 동일한 네임스페이스에 생성됩니다. 예를 들어 **knative-serving** 네임스페이스에 **KnativeServing** 사용자 정의 리소스가 생성되면 이 네임스페이스에 모든 Knative Serving 구성 맵도 생성됩니다.

Knative 사용자 정의 리소스의 **spec.config**에는 **config-< name >**이라는 각 구성 맵에 대해 하나의 **<name >** 항목이 있으며 구성 맵 데이터에 사용되는 값이 있습니다.

2.4. OPENSIFT CONTAINER PLATFORM의 추가 리소스

- 사용자 정의 리소스 정의에서 리소스 관리
- 영구저장장치 이해
- 사용자 정의 PKI 구성

2.5. 다음 단계

- OpenShift Serverless Operator를 설치한 후 [Knative Serving](#)을 설치하거나 [Knative Eventing](#)을 설치할 수 있습니다.

3장. KNATIVE CLI 설치

Knative(**kn**) CLI에는 자체 로그인 메커니즘이 없습니다. 클러스터에 로그인하려면 OpenShift CLI(**oc**)를 설치하고 **oc login** 명령을 사용해야 합니다. CLI에 대한 설치 옵션은 운영 체제에 따라 다를 수 있습니다.

운영 체제의 OpenShift CLI(**oc**) 설치 및 **oc** 로 로그인하는 방법에 대한 자세한 내용은 [OpenShift CLI 시작 설명서](#)를 참조하십시오.

OpenShift Serverless는 Knative(**kn**) CLI를 사용하여 설치할 수 없습니다. 클러스터 관리자는 OpenShift Serverless Operator 설치 설명서에 설명된 대로 [OpenShift Serverless Operator](#)를 설치하고 Knative 구성 요소를 설정해야 합니다.

중요

최신 OpenShift Serverless 릴리스에서 이전 버전의 Knative(**kn**) CLI를 사용하려고 하면 API를 찾을 수 없으며 오류가 발생합니다.

예를 들어 Knative Serving 및 Knative Eventing API의 1.3 버전을 사용하는 1.24.0 OpenShift Serverless 릴리스와 함께 버전 1.2(**kn**) CLI의 1.23.0 릴리스를 사용하는 경우 CLI는 오래된 1.2 API 버전을 계속 찾기 때문에 CLI가 작동하지 않습니다.

문제를 방지하려면 OpenShift Serverless 릴리스에 최신 Knative(**kn**) CLI 버전을 사용하고 있는지 확인합니다.

3.1. OPENSIFT CONTAINER PLATFORM 웹 콘솔을 통한 KNATIVE CLI 설치

OpenShift Container Platform 웹 콘솔을 사용하면 간소화되고 직관적인 사용자 인터페이스를 통해 Knative(**kn**) CLI를 설치할 수 있습니다. OpenShift Serverless Operator가 설치되면 OpenShift Container Platform 웹 콘솔의 **명령줄 툴** 페이지에서 Linux (amd64, s390x, ppc64le), macOS 또는 Windows용 Knative (**kn**) CLI를 다운로드할 수 있는 링크가 표시됩니다.

사전 요구 사항

- OpenShift Container Platform 웹 콘솔에 로그인했습니다.
- OpenShift Serverless Operator 및 Knative Serving이 OpenShift Container Platform 클러스터에 설치되어 있습니다.


중요

libc를 사용할 수 없는 경우 CLI 명령을 실행할 때 다음과 같은 오류가 표시될 수 있습니다.

```
$ kn: No such file or directory
```

- 이 절차에 대한 확인 단계를 사용하려면 OpenShift (**oc**) CLI를 설치해야 합니다.

절차

1. **명령줄 툴** 페이지에서 Knative(**kn**) CLI를 다운로드합니다. 웹 콘솔의 오른쪽 상단에 있는  아이코를 클릭하고 목록에서 **명령줄 툴**을 선택하여 **명령줄 툴**에 액세스할 수 있습니다.

2. 아카이브의 압축을 풉니다.

```
$ tar -xf <file>
```

3. **kn** 바이너리를 **PATH**의 디렉터리로 이동합니다.

4. **PATH**를 확인하려면 다음을 실행합니다.

```
$ echo $PATH
```

검증

- 다음 명령을 실행하여 올바른 Knative CLI 리소스 및 경로가 생성되었는지 확인합니다.

```
$ oc get ConsoleCLIDownload
```

출력 예

NAME	DISPLAY NAME	AGE
kn	kn - OpenShift Serverless Command Line Interface (CLI)	2022-09-20T08:41:18Z
oc-cli-downloads	oc - OpenShift Command Line Interface (CLI)	2022-09-20T08:00:20Z

```
$ oc get route -n openshift-serverless
```

출력 예

NAME	HOST/PORT	PATH	SERVICES	PORT
kn	kn-openshift-serverless.apps.example.com	edge/Redirect	knative-openshift-metrics-3	http-cli
		None		

3.2. RPM 패키지 관리자를 사용하여 LINUX용 KNATIVE CLI 설치

RHEL(Red Hat Enterprise Linux)의 경우 **yum** 또는 **dnf** 와 같은 패키지 관리자를 사용하여 Knative(**kn**) CLI를 RPM으로 설치할 수 있습니다. 이를 통해 Knative CLI 버전을 시스템에서 자동으로 관리할 수 있습니다. 예를 들어 **dnf**와 같은 명령을 사용하면 새 버전을 사용할 수 있는 경우 **kn** 을 포함한 모든 패키지가 업그레이드 됩니다.

사전 요구 사항

- Red Hat 계정에 활성 OpenShift Container Platform 서브스크립션이 있어야 합니다.

절차

1. Red Hat Subscription Manager에 등록합니다.

```
# subscription-manager register
```

2. 최신 서브스크립션 데이터를 가져옵니다.

```
# subscription-manager refresh
```

3. 등록된 시스템에 서브스크립션을 연결합니다.

```
# subscription-manager attach --pool=<pool_id> 1
```

1 활성 OpenShift Container Platform 서브스크립션의 풀 ID

4. Knative(**kn**) CLI에 필요한 리포지토리를 활성화합니다.

- Linux (x86_64, amd64)

```
# subscription-manager repos --enable="openshift-serverless-1-for-rhel-8-x86_64-rpms"
```

- Linux on IBM zSystems and IBM® LinuxONE(s390x)

```
# subscription-manager repos --enable="openshift-serverless-1-for-rhel-8-s390x-rpms"
```

- Linux on IBM Power (ppc64le)

```
# subscription-manager repos --enable="openshift-serverless-1-for-rhel-8-ppc64le-rpms"
```

5. 패키지 관리자를 사용하여 Knative(**kn**) CLI를 RPM으로 설치합니다.

yum 명령 예

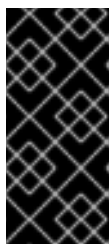
```
# yum install openshift-serverless-clients
```

3.3. LINUX의 KNATIVE CLI 설치

RPM 또는 다른 패키지 관리자가 설치되어 있지 않은 Linux 배포를 사용하는 경우 Knative(**kn**) CLI를 바이너리 파일로 설치할 수 있습니다. 이렇게 하려면 **tar.gz** 아카이브를 다운로드하여 압축을 풀고 **PATH**의 디렉터리에 바이너리를 추가해야 합니다.

사전 요구 사항

- RHEL 또는 Fedora를 사용하지 않는 경우 **libc**가 라이브러리 경로의 디렉터리에 설치되어 있는지 확인합니다.



중요

libc를 사용할 수 없는 경우 CLI 명령을 실행할 때 다음과 같은 오류가 표시될 수 있습니다.

```
$ kn: No such file or directory
```

절차

1. 관련 Knative(**kn**) CLI **tar.gz** 아카이브를 다운로드합니다.

- [Linux \(x86_64, amd64\)](#)

- Linux on IBM zSystems and IBM® LinuxONE(s390x)
- Linux on IBM Power (ppc64le)

Serverless 클라이언트 다운로드 미러에서 해당 버전의 해당 디렉터리로 이동하여 **kn** 버전을 다운로드할 수도 있습니다.

2. 아카이브의 압축을 풉니다.

```
$ tar -xf <filename>
```

3. **kn** 바이너리를 **PATH**의 디렉터리로 이동합니다.
4. **PATH**를 확인하려면 다음을 실행합니다.

```
$ echo $PATH
```

3.4. MACOS용 KNATIVE CLI 설치

macOS를 사용하는 경우 Knative(**kn**) CLI를 바이너리 파일로 설치할 수 있습니다. 이렇게 하려면 **tar.gz** 아카이브를 다운로드하여 압축을 풀고 **PATH**의 디렉터리에 바이너리를 추가해야 합니다.

절차

1. Knative (**kn**) CLI **tar.gz** 아카이브를 다운로드합니다.
Serverless 클라이언트 다운로드 미러에서 해당 버전의 해당 디렉터리로 이동하여 **kn** 버전을 다운로드할 수도 있습니다.
2. 아카이브의 압축을 풀고 추출합니다.
3. **kn** 바이너리를 **PATH**의 디렉터리로 이동합니다.
4. **PATH**를 확인하려면 터미널 창을 열고 다음을 실행합니다.

```
$ echo $PATH
```

3.5. WINDOWS용 KNATIVE CLI 설치

Windows를 사용하는 경우 Knative(**kn**) CLI를 바이너리 파일로 설치할 수 있습니다. 이렇게 하려면 ZIP 아카이브를 다운로드하여 압축을 풀고 **PATH**의 디렉터리에 바이너리를 추가해야 합니다.

절차

1. Knative (**kn**) CLI ZIP 아카이브를 다운로드합니다.
Serverless 클라이언트 다운로드 미러에서 해당 버전의 해당 디렉터리로 이동하여 **kn** 버전을 다운로드할 수도 있습니다.
2. ZIP 프로그램으로 아카이브의 압축을 풉니다.
3. **kn** 바이너리를 **PATH**의 디렉터리로 이동합니다.
4. **PATH**를 확인하려면 명령 프롬프트를 열고 다음 명령을 실행합니다.

```
C:\> path
```


-

4장. KNATIVE SERVING 설치

Knative Serving을 설치하면 클러스터에서 Knative 서비스 및 기능을 생성할 수 있습니다. 또한 애플리케이션의 자동 확장 및 네트워킹 옵션과 같은 추가 기능을 사용할 수 있습니다.

OpenShift Serverless Operator를 설치한 후 기본 설정을 사용하여 **Knative Serving** 을 설치하거나 KnativeServing CR(사용자 정의 리소스)에서 고급 설정을 구성할 수 있습니다. **KnativeServing** CR의 구성 옵션에 대한 자세한 내용은 [글로벌 구성](#) 을 참조하십시오.



중요

OpenShift [Serverless](#)에서 [Red Hat OpenShift distributed tracing](#)을 사용하려면 Knative Serving을 설치하기 전에 Red Hat OpenShift distributed tracing을 설치하고 구성해야 합니다.

4.1. 웹 콘솔을 사용하여 KNATIVE SERVING 설치

OpenShift Serverless Operator를 설치한 후 OpenShift Container Platform 웹 콘솔을 사용하여 Knative Serving을 설치합니다. 기본 설정을 사용하여 **Knative Serving** 을 설치하거나 KnativeServing CR(사용자 정의 리소스)에서 고급 설정을 구성할 수 있습니다.

사전 요구 사항

- OpenShift Container Platform에 대한 클러스터 관리자 권한이 있거나 AWS 또는 OpenShift Dedicated의 Red Hat OpenShift Service에 대한 클러스터 또는 전용 관리자 권한이 있습니다.
- OpenShift Container Platform 웹 콘솔에 로그인했습니다.
- OpenShift Serverless Operator를 설치했습니다.

절차

1. OpenShift Container Platform 웹 콘솔의 관리자 화면에서 **Operator** → 설치된 **Operator**로 이동합니다.
2. 페이지 상단에 있는 프로젝트 드롭다운이 **Project: knative-serving**으로 설정되어 있는지 확인합니다.
3. OpenShift Serverless Operator의 제공되는 API 목록에서 **Knative Serving**을 클릭하여 **Knative Serving** 탭으로 이동합니다.
4. **Knative Serving** 생성을 클릭합니다.
5. **Knative Serving** 생성 페이지에서 생성을 클릭하면 기본 설정을 사용하여 Knative Serving을 설치할 수 있습니다.
제공된 양식을 사용하여 **KnativeServing** 오브젝트를 편집하거나 YAML을 편집하여 Knative Serving 설치에 대한 설정을 수정할 수도 있습니다.
 - 해당 양식은 **KnativeServing** 오브젝트 생성을 완전히 제어할 필요가 없는 단순한 구성에 사용하는 것이 좋습니다.
 - **KnativeServing** 오브젝트 생성을 완전히 제어해야 하는 복잡한 구성의 경우 YAML을 편집하는 것이 좋습니다. **Knative Serving** 생성 페이지의 오른쪽 상단에 있는 **YAML 편집** 링크를 클릭하여 YAML에 액세스할 수 있습니다.
양식을 작성하거나 YAML을 수정한 후 생성을 클릭합니다.



참고

KnativeServing 사용자 정의 리소스 정의의 구성 옵션에 대한 자세한 내용은 [고급 설치 구성 옵션 설명서](#)를 참조하십시오.

- Knative Serving이 설치되면 **KnativeServing** 오브젝트가 생성되고 **Knative Serving** 탭으로 자동으로 이동합니다. 리소스 목록에 **knative-serving** 사용자 정의 리소스가 표시됩니다.

검증

- Knative Serving 탭에서 **knative-serving** 사용자 정의 리소스를 클릭합니다.
- 그러면 Knative Serving 개요 페이지로 자동으로 이동합니다.

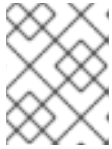
The screenshot shows the Knative Serving Overview page. The left sidebar contains navigation options like Home, Dashboards, Projects, Search, Explore, Events, Operators, Installed Operators, Workloads, Pods, Deployments, Deployment Configs, Stateful Sets, Secrets, Config Maps, and Cron Jobs. The main content area displays the following information:

- Name:** knative-serving
- Version:** 0.13.2
- Namespace:** knative-serving
- Labels:** No labels
- Annotations:** 0 Annotations
- Created At:** 3 minutes ago
- Owner:** No owner

- 조건 목록을 보려면 아래로 스크롤합니다.
- 예제 이미지에 표시된 대로 상태가 **True**인 조건 목록이 표시되어야 합니다.

The screenshot shows the Conditions section of the Knative Serving Overview page. The conditions table is as follows:

Type	Status	Updated	Reason	Message
DependenciesInstalled	True	3 minutes ago	-	-
DeploymentsAvailable	True	3 minutes ago	-	-
InstallSucceeded	True	3 minutes ago	-	-
Ready	True	3 minutes ago	-	-



참고

Knative Serving 리소스를 생성하는 데 몇 초가 걸릴 수 있습니다. 리소스 탭에서 해당 상태를 확인할 수 있습니다.

- 조건이 **알 수 없음** 또는 **False** 상태인 경우 몇 분 정도 기다렸다가 리소스가 생성된 것을 확인한 후 다시 확인하십시오.

4.2. YAML 을 사용하여 KNATIVE SERVING 설치

OpenShift Serverless Operator를 설치한 후 기본 설정을 사용하여 **Knative Serving** 을 설치하거나 KnativeServing CR(사용자 정의 리소스)에서 고급 설정을 구성할 수 있습니다. 다음 절차에 따라 YAML 파일 및 **oc** CLI를 사용하여 Knative Serving을 설치할 수 있습니다.

사전 요구 사항

- OpenShift Container Platform에 대한 클러스터 관리자 권한이 있거나 AWS 또는 OpenShift Dedicated의 Red Hat OpenShift Service에 대한 클러스터 또는 전용 관리자 권한이 있습니다.
- OpenShift Serverless Operator를 설치했습니다.
- OpenShift CLI(**oc**)를 설치합니다.

절차

- servicing.yaml**이라는 파일을 생성하고 다음 예제 YAML을 이 파일에 복사합니다.

```
apiVersion: operator.knative.dev/v1beta1
kind: KnativeServing
metadata:
  name: knative-serving
  namespace: knative-serving
```

- servicing.yaml** 파일을 적용합니다.

```
$ oc apply -f servicing.yaml
```

검증

- 설치가 완료되었는지 확인하려면 다음 명령을 입력합니다.

```
$ oc get knativeserving.operator.knative.dev/knative-serving -n knative-serving --
template='{{range .status.conditions}}{{printf "%s=%s\n" .type .status}}{{end}}'
```

출력 예

```
DependenciesInstalled=True
DeploymentsAvailable=True
InstallSucceeded=True
Ready=True
```



참고

Knative Serving 리소스를 생성하는 데 몇 초가 걸릴 수 있습니다.

조건이 **알 수 없음** 또는 **False** 상태인 경우 몇 분 정도 기다렸다가 리소스가 생성된 것을 확인한 후 다시 확인하십시오.

2. Knative Serving 리소스가 생성되었는지 확인합니다.

```
$ oc get pods -n knative-serving
```

출력 예

NAME	READY	STATUS	RESTARTS	AGE
activator-67ddf8c9d7-p7rm5	2/2	Running	0	4m
activator-67ddf8c9d7-q84fz	2/2	Running	0	4m
autoscaler-5d87bc6dbf-6nqc6	2/2	Running	0	3m59s
autoscaler-5d87bc6dbf-h64rl	2/2	Running	0	3m59s
autoscaler-hpa-77f85f5cc4-lrts7	2/2	Running	0	3m57s
autoscaler-hpa-77f85f5cc4-zx7hl	2/2	Running	0	3m56s
controller-5cfc7cb8db-nlcll	2/2	Running	0	3m50s
controller-5cfc7cb8db-rmv7r	2/2	Running	0	3m18s
domain-mapping-86d84bb6b4-r746m	2/2	Running	0	3m58s
domain-mapping-86d84bb6b4-v7nh8	2/2	Running	0	3m58s
domainmapping-webhook-769d679d45-bkcnj	2/2	Running	0	3m58s
domainmapping-webhook-769d679d45-fff68	2/2	Running	0	3m58s
storage-version-migration-serving-serving-0.26.0--1-6qlkb	0/1	Completed	0	3m56s
webhook-5fb774f8d8-6bqrt	2/2	Running	0	3m57s
webhook-5fb774f8d8-b8lt5	2/2	Running	0	3m57s

3. 필요한 네트워킹 구성 요소가 자동으로 생성된 **knative-serving-ingress** 네임스페이스에 설치되었는지 확인합니다.

```
$ oc get pods -n knative-serving-ingress
```

출력 예

NAME	READY	STATUS	RESTARTS	AGE
net-kourier-controller-7d4b6c5d95-62mkf	1/1	Running	0	76s
net-kourier-controller-7d4b6c5d95-qmgm2	1/1	Running	0	76s
3scale-kourier-gateway-6688b49568-987qz	1/1	Running	0	75s
3scale-kourier-gateway-6688b49568-b5tnp	1/1	Running	0	75s

4.3. 다음 단계

- Knative 이벤트 중심 아키텍처를 사용하려면 [Knative Eventing](#)을 설치할 수 있습니다.

5장. KNATIVE EVENTING 설치

클러스터에서 이벤트 중심 아키텍처를 사용하려면 Knative Eventing을 설치합니다. 이벤트 소스, 브로커 및 채널과 같은 Knative 구성 요소를 생성한 다음 이를 사용하여 이벤트를 애플리케이션 또는 외부 시스템으로 보낼 수 있습니다.

OpenShift Serverless Operator를 설치한 후 기본 설정을 사용하여 Knative Eventing을 설치하거나 **KnativeEventing** 사용자 정의 리소스(CR)에서 고급 설정을 구성할 수 있습니다. **KnativeEventing** CR의 구성 옵션에 대한 자세한 내용은 [글로벌 구성](#) 을 참조하십시오.



중요

OpenShift [Serverless](#)에서 [Red Hat OpenShift distributed tracing](#)을 사용하려면 Knative Eventing을 설치하기 전에 Red Hat OpenShift distributed tracing을 설치하고 구성해야 합니다.

5.1. 웹 콘솔을 사용하여 KNATIVE EVENTING 설치

OpenShift Serverless Operator를 설치한 후 OpenShift Container Platform 웹 콘솔을 사용하여 Knative Eventing을 설치합니다. 기본 설정을 사용하여 Knative Eventing을 설치하거나 **KnativeEventing** CR(사용자 정의 리소스)에서 고급 설정을 구성할 수 있습니다.

사전 요구 사항

- OpenShift Container Platform에 대한 클러스터 관리자 권한이 있거나 AWS 또는 OpenShift Dedicated의 Red Hat OpenShift Service에 대한 클러스터 또는 전용 관리자 권한이 있습니다.
- OpenShift Container Platform 웹 콘솔에 로그인했습니다.
- OpenShift Serverless Operator를 설치했습니다.

절차

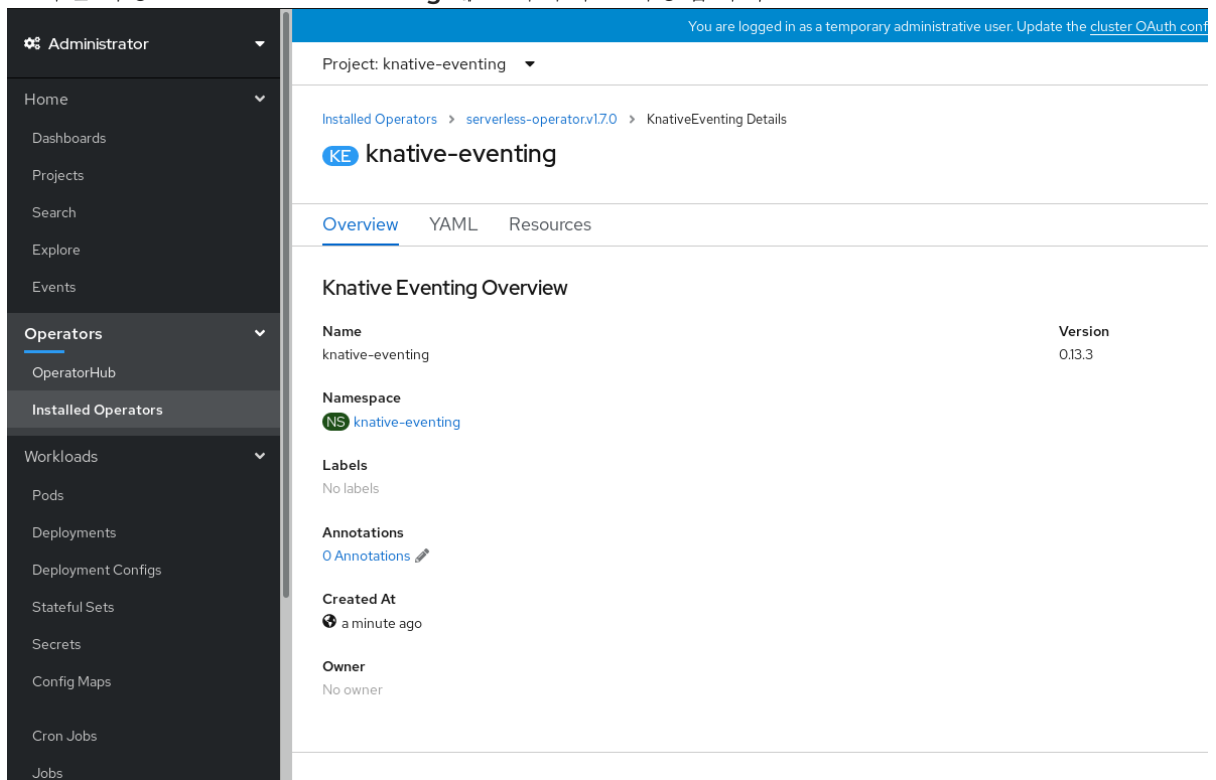
1. OpenShift Container Platform 웹 콘솔의 관리자 화면에서 **Operator** → 설치된 **Operator**로 이동합니다.
2. 페이지 상단에 있는 프로젝트 드롭다운이 **Project: knative-eventing**으로 설정되어 있는지 확인합니다.
3. OpenShift Serverless Operator의 제공되는 API 목록에서 **Knative Eventing**을 클릭하여 **Knative Eventing** 탭으로 이동합니다.
4. **Knative Eventing** 생성을 클릭합니다.
5. **Knative Eventing** 생성 페이지에서 제공된 기본 양식을 사용하거나 YAML을 편집하여 **KnativeEventing** 오브젝트를 구성하도록 선택할 수 있습니다.
 - 해당 양식은 **KnativeEventing** 오브젝트 생성을 완전히 제어할 필요가 없는 단순한 구성에 사용하는 것이 좋습니다. 양식을 사용하여 **KnativeEventing** 오브젝트를 구성하는 경우 Knative Eventing 배포에 구현하려는 변경을 수행합니다.
6. **생성**을 클릭합니다.
 - **KnativeEventing** 오브젝트 생성을 완전히 제어해야 하는 복잡한 구성의 경우 YAML을 편집하는 것이 좋습니다. **Knative Eventing** 생성 페이지의 오른쪽 상단에 있는 **YAML 편집 링크**

를 클릭하여 YAML에 액세스할 수 있습니다.
 선택 사항입니다. YAML을 편집하여 **KnativeEventing** 오브젝트를 구성하는 경우 YAML에 Knative Eventing 배포에 구현하려는 변경을 수행합니다.

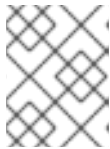
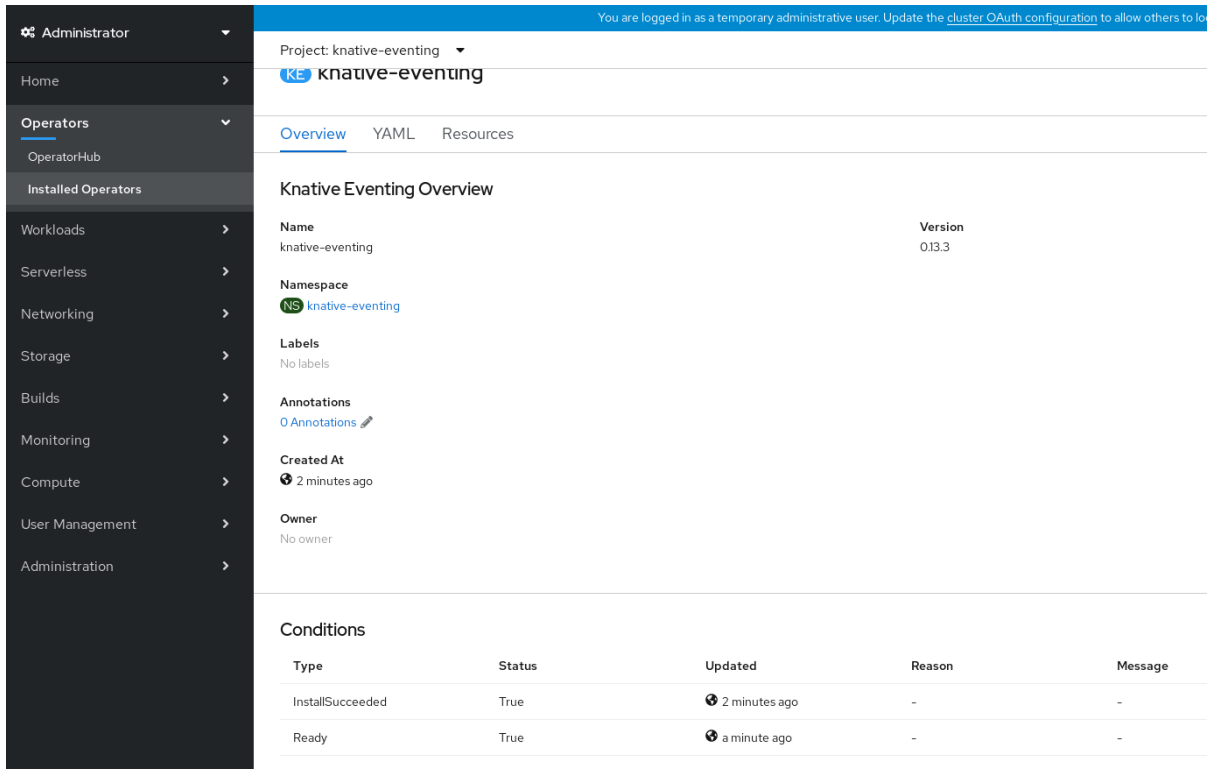
7. **생성**을 클릭합니다.
8. Knative Eventing을 설치하면 **KnativeEventing** 오브젝트가 생성되고 **Knative Eventing** 탭으로 자동으로 이동합니다. 리소스 목록에 **knative-eventing** 사용자 정의 리소스가 표시됩니다.

검증

1. **Knative Eventing** 탭에서 **knative-eventing** 사용자 정의 리소스를 클릭합니다.
2. 그러면 자동으로 **Knative Eventing** 개요 페이지로 이동합니다.



3. **조건** 목록을 보려면 아래로 스크롤합니다.
4. 예제 이미지에 표시된 대로 상태가 **True**인 조건 목록이 표시되어야 합니다.



참고

Knative Eventing 리소스를 생성하는 데 몇 초가 걸릴 수 있습니다. 리소스 탭에서 해당 상태를 확인할 수 있습니다.

- 5. 조건이 **알 수 없음** 또는 **False** 상태인 경우 몇 분 정도 기다렸다가 리소스가 생성된 것을 확인한 후 다시 확인하십시오.

5.2. YAML을 사용하여 KNATIVE EVENTING 설치

OpenShift Serverless Operator를 설치한 후 기본 설정을 사용하여 Knative Eventing을 설치하거나 **KnativeEventing** 사용자 정의 리소스(CR)에서 고급 설정을 구성할 수 있습니다. 다음 절차에 따라 YAML 파일 및 **oc** CLI를 사용하여 Knative Eventing을 설치할 수 있습니다.

사전 요구 사항

- OpenShift Container Platform에 대한 클러스터 관리자 권한이 있거나 AWS 또는 OpenShift Dedicated의 Red Hat OpenShift Service에 대한 클러스터 또는 전용 관리자 권한이 있습니다.
- OpenShift Serverless Operator를 설치했습니다.
- OpenShift CLI(**oc**)를 설치합니다.

절차

1. **eventing.yaml**이라는 파일을 생성합니다.
2. 다음 샘플 YAML을 **eventing.yaml**에 복사합니다.

```
apiVersion: operator.knative.dev/v1beta1
kind: KnativeEventing
metadata:
```



```
name: knative-eventing
namespace: knative-eventing
```

3. 선택 사항입니다. Knative Eventing 배포를 위해 구현하려는 YAML을 변경합니다.
4. 다음을 입력하여 **eventing.yaml** 파일을 적용합니다.

```
$ oc apply -f eventing.yaml
```

검증

1. 다음 명령을 입력하고 출력을 관찰하여 설치가 완료되었는지 확인합니다.

```
$ oc get knativeeventing.operator.knative.dev/knative-eventing \
-n knative-eventing \
--template='{{range .status.conditions}}{{printf "%s=%s\n" .type .status}}{{end}}'
```

출력 예

```
InstallSucceeded=True
Ready=True
```



참고

Knative Eventing 리소스를 생성하는 데 몇 초가 걸릴 수 있습니다.

2. 조건이 **알 수 없음** 또는 **False** 상태인 경우 몇 분 정도 기다렸다가 리소스가 생성된 것을 확인한 후 다시 확인하십시오.
3. 다음을 입력하여 Knative Eventing 리소스가 생성되었는지 확인합니다.

```
$ oc get pods -n knative-eventing
```

출력 예

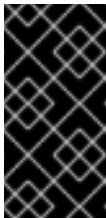
NAME	READY	STATUS	RESTARTS	AGE
broker-controller-58765d9d49-g9zp6	1/1	Running	0	7m21s
eventing-controller-65fdd66b54-jw7bh	1/1	Running	0	7m31s
eventing-webhook-57fd74b5bd-kvhlz	1/1	Running	0	7m31s
imc-controller-5b75d458fc-ptvm2	1/1	Running	0	7m19s
imc-dispatcher-64f6d5fccb-kkc4c	1/1	Running	0	7m18s

5.3. APACHE KAFKA용 KNATIVE 브로커 설치

Apache Kafka의 Knative 브로커 구현에서는 지원되는 Apache Kafka 메시지 스트리밍 플랫폼을 OpenShift Serverless와 함께 사용할 수 있는 통합 옵션을 제공합니다. **KnativeKafka** 사용자 정의 리소스를 설치한 경우 Apache Kafka 기능의 Knative 브로커는 OpenShift Serverless 설치에서 사용할 수 있습니다.

사전 요구 사항

- OpenShift Serverless Operator 및 Knative Eventing을 클러스터에 설치했습니다.
- Red Hat AMQ Streams 클러스터에 액세스할 수 있습니다.
- 확인 단계를 사용하려면 OpenShift CLI(**oc**)를 설치합니다.
- OpenShift Container Platform에 대한 클러스터 관리자 권한이 있거나 AWS 또는 OpenShift Dedicated의 Red Hat OpenShift Service에 대한 클러스터 또는 전용 관리자 권한이 있습니다.
- OpenShift Container Platform 웹 콘솔에 로그인되어 있습니다. Procedure
 1. 관리자 화면에서 **Operator** → 설치된 **Operator**로 이동합니다.
 2. 페이지 상단에 있는 프로젝트 드롭다운이 **Project: knative-eventing**으로 설정되어 있는지 확인합니다.
 3. OpenShift Serverless Operator의 **Provided APIs** 목록에서 **Knative Kafka**를 검색하여 **Create Instance**를 클릭합니다.
 4. **Knative Kafka** 생성 페이지에서 **KnativeKafka** 오브젝트를 구성합니다.



중요

클러스터에서 Kafka 채널, 소스, 브로커 또는 싱크를 사용하려면 사용할 옵션에 대해 **활성화된** 스위치를 **true** 로 전환해야 합니다. 이러한 스위치는 기본적으로 **false**로 설정됩니다. 또한 Kafka 채널, 브로커 또는 싱크를 사용하려면 부트스트랩 서버를 지정해야 합니다.

KnativeKafka 사용자 정의 리소스의 예

```

apiVersion: operator.serverless.openshift.io/v1alpha1
kind: KnativeKafka
metadata:
  name: knative-kafka
  namespace: knative-eventing
spec:
  channel:
    enabled: true 1
    bootstrapServers: <bootstrap_servers> 2
  source:
    enabled: true 3
  broker:
    enabled: true 4
    defaultConfig:
      bootstrapServers: <bootstrap_servers> 5
      numPartitions: <num_partitions> 6
      replicationFactor: <replication_factor> 7
  sink:
    enabled: true 8
    
```

- 1 개발자가 클러스터에서 **KafkaChannel** 채널 유형을 사용할 수 있습니다.
- 2 AMQ Streams 클러스터에 있는 쉽표로 구분된 부트스트랩 서버 목록입니다.

- 3 개발자가 클러스터에서 **KafkaSource** 이벤트 소스 유형을 사용할 수 있습니다.
- 4 개발자가 클러스터에서 Apache Kafka에 대한 Knative 브로커 구현을 사용할 수 있습니다.
- 5 Red Hat AMQ Streams 클러스터의 쉼표로 구분된 부트스트랩 서버 목록입니다.
- 6 **Broker** 오브젝트에서 지원하는 Kafka 주제의 파티션 수를 정의합니다. 기본값은 **10**입니다.
- 7 **Broker** 오브젝트에서 지원하는 Kafka 항목의 복제 요소를 정의합니다. 기본값은 **3**입니다.
- 8 개발자가 클러스터에서 Kafka 싱크를 사용할 수 있습니다.



참고

replicationFactor 값은 Red Hat AMQ Streams 클러스터의 노드 수보다 작거나 같아야 합니다.


- a. 해당 양식은 **KnativeKafka** 오브젝트 생성을 완전히 제어할 필요가 없는 단순한 구성에 사용하는 것이 좋습니다.
 - b. **KnativeKafka** 오브젝트 생성을 완전히 제어해야 하는 복잡한 구성의 경우 YAML을 편집하는 것이 좋습니다. **Knative Kafka** 생성 페이지의 오른쪽 상단에 있는 **YAML 편집 링크**를 클릭하여 YAML에 액세스할 수 있습니다.
5. Kafka에 대한 선택적 구성을 완료한 후 **생성**을 클릭합니다. 그러면 리소스 목록에 **knative-kafka**가 있는 **Knative Kafka** 탭으로 자동으로 이동합니다.

검증


1. **Knative Kafka** 탭에서 **knative-kafka** 리소스를 클릭합니다. 그러면 자동으로 **Knative Kafka** 개요 페이지로 이동합니다.
2. 리소스에 대한 조건 목록을 확인하고 상태가 **True**인지 확인합니다.


Knative Kafka Overview

Name
knative-kafka

Namespace
 knative-eventing




Labels
No labels

Annotations
1 Annotation 

Created At
 Oct 6, 11:29 am

Owner
No owner

Conditions

Type	Status	Updated
DeploymentsAvailable	True	 Oct 6, 11:29 am
InstallSucceeded	True	 Oct 6, 11:29 am
Ready	True	 Oct 6, 11:29 am

조건 상태가 **알 수 없음** 또는 **False**인 경우 몇 분 정도 기다린 후 페이지를 새로 고칩니다.

3. Apache Kafka 리소스의 Knative 브로커가 생성되었는지 확인합니다.

```
$ oc get pods -n knative-eventing
```

출력 예

```
NAME                                READY STATUS RESTARTS AGE
kafka-broker-dispatcher-7769fbbcbb-xgffn  2/2 Running 0      44s
kafka-broker-receiver-5fb56f7656-fhq8d    2/2 Running 0      44s
kafka-channel-dispatcher-84fd6cb7f9-k2tjv  2/2 Running 0      44s
kafka-channel-receiver-9b7f795d5-c76xr    2/2 Running 0      44s
kafka-controller-6f95659bf6-trd6r        2/2 Running 0      44s
kafka-source-dispatcher-6bf98bdfff-8bcsn  2/2 Running 0      44s
kafka-webhook-eventing-68dc95d54b-825xs   2/2 Running 0      44s
```

5.4. 다음 단계

- Knative 서비스를 사용하려면 [Knative Serving](#)을 설치할 수 있습니다.

6장. APACHE KAFKA의 KNATIVE 브로커 구성

Apache Kafka의 Knative 브로커 구현에서는 지원되는 Apache Kafka 메시지 스트리밍 플랫폼을 OpenShift Serverless와 함께 사용할 수 있는 통합 옵션을 제공합니다. Kafka는 이벤트 소스, 채널, 브로커 및 이벤트 싱크 기능에 대한 옵션을 제공합니다.

핵심 OpenShift Serverless 설치의 일부로 제공되는 Knative Eventing 구성 요소 외에도 **KnativeKafka** CR(사용자 정의 리소스)은 다음을 통해 설치할 수 있습니다.

- OpenShift Container Platform의 클러스터 관리자
- AWS의 Red Hat OpenShift Service 또는 OpenShift Dedicated 용 클러스터 또는 전용 관리자입니다.

KnativeKafka CR은 사용자에게 다음과 같은 추가 옵션을 제공합니다.

- Kafka 소스
- Kafka 채널
- Kafka 브로커
- Kafka 싱크

7장. APACHE KAFKA의 KNATIVE용 KUBE-RBAC-PROXY 구성

kube-rbac-proxy 구성 요소는 Apache Kafka에 Knative에 대한 내부 인증 및 권한 부여 기능을 제공합니다.

7.1. APACHE KAFKA용 KNATIVE에 대한 KUBE-RBAC-PROXY 리소스 구성

OpenShift Serverless Operator CR을 사용하여 **kube-rbac-proxy** 컨테이너의 리소스 할당을 전역적으로 덮어쓸 수 있습니다.

You can also override resource allocation for a specific deployment.

다음 구성은 Knative Kafka **kube-rbac-proxy** 최소 및 최대 CPU 및 메모리 할당을 설정합니다.

KnativeKafka CR 예

```
apiVersion: operator.serverless.openshift.io/v1alpha1
kind: KnativeKafka
metadata:
  name: knative-kafka
  namespace: knative-kafka
spec:
  config:
    workload:
      "kube-rbac-proxy-cpu-request": "10m" 1
      "kube-rbac-proxy-memory-request": "20Mi" 2
      "kube-rbac-proxy-cpu-limit": "100m" 3
      "kube-rbac-proxy-memory-limit": "100Mi" 4
```

- 1 최소 CPU 할당을 설정합니다.
- 2 최소 RAM 할당을 설정합니다.
- 3 최대 CPU 할당을 설정합니다.
- 4 최대 RAM 할당을 설정합니다.

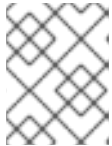
8장. OPENSIFT SERVERLESS FUNCTIONS 구성

애플리케이션 코드 배포 프로세스를 개선하기 위해 OpenShift Serverless를 사용하여 OpenShift Container Platform에 상태 비저장 이벤트 중심 기능을 Knative 서비스로 배포할 수 있습니다. 함수를 개발하려면 설정 단계를 완료해야 합니다.

8.1. 사전 요구 사항

클러스터에서 OpenShift Serverless Functions을 사용하려면 다음 단계를 완료해야 합니다.

- OpenShift Serverless Operator 및 Knative Serving이 클러스터에 설치되어 있습니다.



참고

함수는 Knative 서비스로 배포됩니다. 이벤트 중심 아키텍처를 기능과 함께 사용하려면 Knative Eventing도 설치해야 합니다.

- **oc CLI**가 설치되어 있어야 합니다.
- **Knative(kn) CLI**가 설치되어 있어야 합니다. Knative CLI를 설치하면 함수를 생성하고 관리하는데 사용할 수 있는 **kn func** 명령을 사용할 수 있습니다.
- Docker Container Engine 또는 Podman 버전 3.4.7 이상을 설치했습니다.
- 사용 가능한 이미지 레지스트리(예: OpenShift Container Registry)에 액세스할 수 있습니다.
- **Quay.io**를 이미지 레지스트리로 사용하는 경우 리포지토리가 비공개가 아닌지 확인하거나 **pod**가 다른 보안 레지스트리의 이미지를 참조하도록 허용하는 **OpenShift Container Platform 설명서**를 따라야 합니다.
- OpenShift Container Registry를 사용하는 경우 클러스터 관리자가 **레지스트리를 공개해야** 합니다.

8.2. PODMAN 설정

고급 컨테이너 관리 기능을 사용하려면 OpenShift Serverless Functions에서 Podman을 사용할 수 있습니다. 이렇게 하려면 Podman 서비스를 시작하고 연결하도록 Knative(kn) CLI를 구성해야 합니다.

절차

1. **`\${XDG_RUNTIME_DIR}/podman/podman.sock`**의 UNIX 소켓에서 Docker API를 제공하는 Podman 서비스를 시작합니다.

```
$ systemctl start --user podman.socket
```



참고

대부분의 시스템에서 이 소켓은 **/run/user/\$(id -u)/podman/podman.sock**에 있습니다.

2. 기능을 구축하는 데 사용되는 환경 변수를 설정합니다.

```
$ export DOCKER_HOST="unix://`${XDG_RUNTIME_DIR}/podman/podman.sock"
```

- 3. 함수 프로젝트 디렉터리 내에서 **-v** 플래그를 사용하여 빌드 명령을 실행하여 자세한 출력을 확인합니다. 로컬 UNIX 소켓에 대한 연결이 표시됩니다.

```
$ kn func build -v
```

8.3. MACOS에서 PODMAN 설정

고급 컨테이너 관리 기능을 사용하려면 OpenShift Serverless Functions에서 Podman을 사용할 수 있습니다. macOS에서 이를 수행하려면 Podman 시스템을 시작하고 연결하도록 Knative(**kn**) CLI를 구성해야 합니다.

절차

1. Podman 시스템을 생성합니다.

```
$ podman machine init --memory=8192 --cpus=2 --disk-size=20
```

2. UNIX 소켓에서 Docker API를 제공하는 Podman 시스템을 시작합니다.

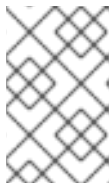
```
$ podman machine start
Starting machine "podman-machine-default"
Waiting for VM ...
Mounting volume... /Users/myuser:/Users/user
```

[...truncated output...]

You can still connect Docker API clients by setting DOCKER_HOST using the following command in your terminal session:

```
export
DOCKER_HOST='unix:///Users/myuser/.local/share/containers/podman/machine/podman-machine-default/podman.sock'
```

```
Machine "podman-machine-default" started successfully
```



참고

대부분의 macOS 시스템에서 이 소켓은 **/Users/myuser/.local/share/containers/podman/podman-machine-machine-default/podman.sock**에 있습니다.

3. 기능을 구축하는 데 사용되는 환경 변수를 설정합니다.

```
$ export
DOCKER_HOST='unix:///Users/myuser/.local/share/containers/podman/machine/podman-machine-default/podman.sock'
```

4. 함수 프로젝트 디렉터리 내에서 **-v** 플래그를 사용하여 빌드 명령을 실행하여 자세한 출력을 확인합니다. 로컬 UNIX 소켓에 대한 연결이 표시됩니다.

```
$ kn func build -v
```


8.4. 다음 단계

- Docker Container Engine 또는 Podman에 대한 자세한 내용은 컨테이너 [빌드 툴 옵션](#)을 참조하십시오.
- [함수 시작하기](#)를 참조하십시오.

9장. 서버리스 업그레이드

릴리스 버전을 건너뛰지 않고 OpenShift Serverless를 업그레이드해야 합니다. 이 섹션에서는 업그레이드 문제를 해결하는 방법을 설명합니다.

9.1. OPENSIFT SERVERLESS OPERATOR 업그레이드 실패 확인

예를 들어 수동으로 설치 제거하고 다시 설치할 때 OpenShift Serverless Operator를 업그레이드할 때 오류가 발생할 수 있습니다. 오류가 발생하면 OpenShift Serverless Operator를 수동으로 다시 설치해야 합니다.

절차

1. OpenShift Serverless 릴리스 노트에서 검색하여 원래 설치된 OpenShift Serverless Operator 버전을 확인합니다.
예를 들어 업그레이드 시도 중 오류 메시지에 다음 문자열이 포함될 수 있습니다.

```
The installed KnativeServing version is v1.5.0.
```

이 예에서 KnativeServing **MAJOR.MINOR** 버전은 **1.5**입니다. 이 버전은 OpenShift Serverless 1.26: *OpenShift Serverless의 릴리스 노트에서 현재 Knative Serving 1.5를 사용 합니다.*

2. OpenShift Serverless Operator 및 모든 설치 계획을 설치 제거합니다.
3. 첫 번째 단계에서 검색한 OpenShift Serverless Operator 버전을 수동으로 설치합니다. 설치하려면 다음 예와 같이 먼저 **서버리스-subscription.yaml** 파일을 생성합니다.

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: serverless-operator
  namespace: openshift-serverless
spec:
  channel: stable
  name: serverless-operator
  source: redhat-operators
  sourceNamespace: openshift-marketplace
  installPlanApproval: Manual
  startingCSV: serverless-operator.v1.26.0
```

4. 그런 다음 다음 명령을 실행하여 서브스크립션을 설치합니다.

```
$ oc apply -f serverless-subscription.yaml
```

5. 표시되는 대로 업그레이드 설치 계획을 수동으로 승인합니다.

추가 리소스

- [OpenShift Serverless 릴리스 정보](#)
- [웹 콘솔을 사용하여 클러스터에서 Operator 삭제](#)
- [웹 콘솔에서 OpenShift Serverless Operator 설치](#)

