



Red Hat OpenShift Pipelines 1.15

설치 및 구성

OpenShift Pipelines 설치 및 구성

Red Hat OpenShift Pipelines 1.15 설치 및 구성

OpenShift Pipelines 설치 및 구성

법적 공지

Copyright © 2024 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 문서에서는 OpenShift Pipelines 설치 및 구성에 대한 정보를 제공합니다.

차례

1장. OPENSIFT PIPELINES 설치	3
사전 요구 사항	3
1.1. 웹 콘솔에서 RED HAT OPENSIFT PIPELINES OPERATOR 설치	3
1.2. CLI를 사용하여 OPENSIFT PIPELINES OPERATOR 설치	5
1.3. 제한된 환경의 RED HAT OPENSIFT PIPELINES OPERATOR	6
1.4. 추가 리소스	6
2장. OPENSIFT PIPELINES 설치 제거	8
2.1. OPENSIFT PIPELINES 사용자 정의 리소스 삭제	8
2.2. RED HAT OPENSIFT PIPELINES OPERATOR 설치 제거	9
2.3. OPERATOR.TEKTON.DEV 그룹의 사용자 정의 리소스 정의 삭제	9
3장. TEKTONCONFIG 사용자 정의 리소스에서 구성 사용자 정의	11
3.1. 사전 요구 사항	11
3.2. TEKTONCONFIG CR을 사용한 성능 튜닝	11
3.3. RED HAT OPENSIFT PIPELINES CONTROL PLANE 구성	13
3.4. OPENSIFT PIPELINES의 기본 서비스 계정 변경	16
3.5. OPENSIFT PIPELINES 설치 네임스페이스에 대한 라벨 및 주석 설정	16
3.6. 서비스 모니터 비활성화	17
3.7. 파이프라인 확인기 구성	18
3.8. 클러스터 작업 및 파이프라인 템플릿 비활성화	19
3.9. TEKTON HUB 통합 비활성화	20
3.10. RBAC 리소스의 자동 생성 비활성화	20
3.11. 파이프라인 및 작업의 인라인 사양 비활성화	21
3.12. 작업 실행 및 파이프라인 실행 자동 정리	23
3.13. 추가 리소스	27

1장. OPENSIFT PIPELINES 설치

이 가이드에서는 클러스터 관리자에게 Red Hat OpenShift Pipelines Operator를 OpenShift Container Platform 클러스터에 설치하는 프로세스를 안내합니다.

사전 요구 사항

- **cluster-admin** 권한이 있는 계정을 사용하여 OpenShift Container Platform 클러스터에 액세스할 수 있습니다.
- **oc** CLI를 설치했습니다.
- 로컬 시스템에 **tkn(OpenShift Pipelines) CLI** 를 설치했습니다.
- 클러스터에 **Marketplace** 기능이 활성화되어 있거나 Red Hat Operator 카탈로그 소스가 수동으로 구성되어 있습니다.



참고

Windows 및 Linux 노드가 모두 있는 클러스터에서 Red Hat OpenShift Pipelines는 Linux 노드에서만 실행할 수 있습니다.

1.1. 웹 콘솔에서 RED HAT OPENSIFT PIPELINES OPERATOR 설치

OpenShift Container Platform OperatorHub에 나열된 Operator를 사용하여 Red Hat OpenShift Pipelines를 설치할 수 있습니다. Red Hat OpenShift Pipelines Operator를 설치하면 파이프라인 구성에 필요한 CR(사용자 정의 리소스)이 Operator와 함께 자동으로 설치됩니다.

기본 Operator CRD(사용자 정의 리소스 정의) **config.operator.tekton.dev**가 **tektonconfigs.operator.tekton.dev**로 교체되었습니다. 또한 Operator에서 OpenShift Pipelines 구성 요소를 개별적으로 관리하기 위해 추가 CRD인 **tektonpipelines.operator.tekton.dev**, **tektontriggers.operator.tekton.dev**, **tektonaddons.operator.tekton.dev**를 제공합니다.

OpenShift Pipelines가 클러스터에 이미 설치되어 있는 경우 기존 설치가 원활하게 업그레이드됩니다. Operator는 필요에 따라 클러스터의 **config.operator.tekton.dev** 인스턴스를 **tektonconfigs.operator.tekton.dev** 인스턴스 및 기타 CRD의 추가 오브젝트로 교체합니다.



주의

resource name - cluster 필드를 변경하여 **config.operator.tekton.dev** CRD 인스턴스의 타겟 네임스페이스를 변경하는 등 기존 설치를 수동으로 변경한 경우 업그레이드 경로가 제대로 작동하지 않습니다. 이러한 경우 권장되는 워크플로는 설치를 제거한 후 Red Hat OpenShift Pipelines Operator를 다시 설치하는 것입니다.

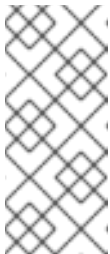
Red Hat OpenShift Pipelines Operator는 이제 **TektonConfig** CR(사용자 정의 리소스)의 일부로 프로필을 지정하여 설치할 구성 요소를 선택할 수 있는 옵션을 제공합니다. Operator가 설치되면 **TektonConfig** CR이 자동으로 설치됩니다. 지원되는 프로필은 다음과 같습니다.

- Lite: Tekton 파이프라인만 설치합니다.

- Basic: Tekton 파이프라인, Tekton 트리거 및 Tekton 체인을 설치합니다.
- 모두: **TektonConfig** CR을 설치할 때 사용하는 기본 프로필입니다. 이 프로필은 Tekton Pipelines, Tekton Triggers, Tekton Chains, Pipelines as Code 및 Tekton Addons를 포함한 모든 Tekton 구성 요소를 설치합니다. Tekton Addons에는 **ClusterTasks, ClusterTriggerBindings, ConsoleCLIDownload, ConsoleQuickStart, ConsoleYAMLSample** 리소스와 **openshift-pipelines** 네임스페이스의 클러스터 확인기를 사용하여 사용할 수 있는 작업이 포함됩니다.

프로세스

1. 웹 콘솔의 관리자 화면에서 **Operator → OperatorHub**로 이동합니다.
2. 키워드로 필터링 박스를 사용하여 카탈로그에서 **Red Hat OpenShift Pipelines Operator**를 검색합니다. **Red Hat OpenShift Pipelines Operator** 타일을 클릭합니다.
3. **Red Hat OpenShift Pipelines Operator** 페이지에서 Operator에 대한 간략한 설명을 확인합니다. 설치를 클릭합니다.
4. **Operator** 설치 페이지에서 다음을 수행합니다.
 - a. **Installation Mode**로 **All namespaces on the cluste(default)**를 선택합니다. 이 모드에서는 기본 **openshift-operators** 네임스페이스에 Operator가 설치되므로 Operator가 클러스터의 모든 네임스페이스를 감시하고 사용 가능하게 만들 수 있습니다.
 - b. **Approval Strategy**으로 **Automatic**을 선택합니다. 그러면 Operator에 향후 지원되는 업그레이드가 OLM(Operator Lifecycle Manager)에 의해 자동으로 처리됩니다. **Manual** 승인 전략을 선택하면 OLM에서 업데이트 요청을 생성합니다. 클러스터 관리자는 Operator를 새 버전으로 업데이트하려면 OLM 업데이트 요청을 수동으로 승인해야 합니다.
 - c. **Update Channel**을 선택합니다.
 - 최신 채널을 사용하면 Red Hat OpenShift Pipelines Operator의 최신 안정적인 버전을 설치할 수 있습니다. 현재 이 채널은 Red Hat OpenShift Pipelines Operator를 설치하는 기본 채널입니다.
 - 특정 버전의 Red Hat OpenShift Pipelines Operator를 설치하기 위해 클러스터 관리자는 해당 **pipelines-<version>** 채널을 사용할 수 있습니다. 예를 들어 Red Hat OpenShift Pipelines Operator 버전 **1.8.x**를 설치하려면 **pipelines-1.8** 채널을 사용할 수 있습니다.



참고

OpenShift Container Platform 4.11부터 Red Hat OpenShift Pipelines Operator를 설치하고 업그레이드하기 위한 **프리뷰** 및 **안정적인** 채널을 사용할 수 없습니다. 그러나 OpenShift Container Platform 4.10 및 이전 버전에서는 **프리뷰** 및 **안정적인** 채널을 사용하여 Operator를 설치 및 업그레이드할 수 있습니다.

5. 설치를 클릭합니다. **Installed Operators** 페이지의 목록에 해당 Operator가 나타납니다.



참고

Operator는 **openshift-operators** 네임스페이스에 자동으로 설치됩니다.

6. Red Hat OpenShift Pipelines Operator가 성공적으로 설치되었는지 확인하려면 **상태가 최신 업데이트 완료**로 설정되어 있는지 확인합니다.



주의

다른 구성 요소의 설치에 진행 중인 경우에도 **성공 상태가 최신 업데이트**로 표시될 수 있습니다. 따라서 터미널에서 수동으로 설치를 확인하는 것이 중요합니다.

7. Red Hat OpenShift Pipelines Operator의 모든 구성 요소가 성공적으로 설치되었는지 확인합니다. 터미널에서 클러스터에 로그인하고 다음 명령을 실행합니다.

```
$ oc get tektonconfig config
```

출력 예

```
NAME   VERSION  READY  REASON
config 1.15.0   True
```

READY 조건이 **True** 이면 Operator 및 해당 구성 요소가 성공적으로 설치됩니다.

Additionally, 다음 명령을 실행하여 구성 요소의 버전을 확인합니다.

```
$ oc get tektonpipeline,tektontrigger,tektonchain,tektonaddon,pac
```

출력 예

```
NAME                                     VERSION  READY  REASON
tektonpipeline.operator.tekton.dev/pipeline  v0.47.0  True
```

```
NAME                                     VERSION  READY  REASON
tektontrigger.operator.tekton.dev/trigger  v0.23.1  True
```

```
NAME                                     VERSION  READY  REASON
tektonchain.operator.tekton.dev/chain      v0.16.0  True
```

```
NAME                                     VERSION  READY  REASON
tektonaddon.operator.tekton.dev/addon      1.11.0   True
```

```
NAME                                     VERSION  READY  REASON
openshiftpipelinesascode.operator.tekton.dev/pipelines-as-code  v0.19.0  True
```

1.2. CLI를 사용하여 OPENSIFT PIPELINES OPERATOR 설치

CLI를 사용하여 OperatorHub에서 Red Hat OpenShift Pipelines Operator를 설치할 수 있습니다.

프로세스

1. 서브스크립션 오브젝트 YAML 파일을 생성하여 Red Hat OpenShift Pipelines Operator에 네임스페이스를 서브스크립션합니다(예: **sub.yaml**).

서브스크립션의 예

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: openshift-pipelines-operator
  namespace: openshift-operators
spec:
  channel: <channel name> 1
  name: openshift-pipelines-operator-rh 2
  source: redhat-operators 3
  sourceNamespace: openshift-marketplace 4
```

- 1 Operator의 채널 이름입니다. **pipelines-<version>** 채널은 기본 채널입니다. 예를 들어 Red Hat OpenShift Pipelines Operator 버전 **1.7**의 기본 채널은 **pipelines-1.7**입니다. **최신** 채널을 사용하면 Red Hat OpenShift Pipelines Operator의 최신 안정적인 버전을 설치할 수 있습니다.
- 2 등록할 Operator의 이름입니다.
- 3 Operator를 제공하는 CatalogSource의 이름입니다.
- 4 CatalogSource의 네임스페이스입니다. 기본 OperatorHub CatalogSources에는 **openshift-marketplace**를 사용합니다.

2. 서브스크립션 오브젝트를 생성합니다.

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

서브스크립션은 Red Hat OpenShift Pipelines Operator를 **openshift-operators** 네임스페이스에 설치합니다. Operator는 OpenShift Pipelines를 기본 **openshift-pipelines** 대상 네임스페이스에 자동으로 설치합니다.

1.3. 제한된 환경의 RED HAT OPENSIFT PIPELINES OPERATOR

Red Hat OpenShift Pipelines Operator는 제한된 네트워크 환경에서 파이프라인 설치를 지원합니다.

Operator는 **cluster** 프록시 오브젝트를 기반으로 tekton-controller에서 생성한 Pod의 컨테이너에 프록시 환경 변수를 설정하는 프록시 Webhook를 설치합니다. 또한 **TektonPipelines, TektonTriggers, Controllers, Webhooks, Operator Proxy Webhook** 리소스에서 프록시 환경 변수를 설정합니다.

기본적으로 프록시 Webhook는 **openshift-pipelines** 네임스페이스에 대해 비활성화되어 있습니다. 다른 네임스페이스에 대해 비활성화하려면 **namespace** 오브젝트에 **operator.tekton.dev/disable-proxy: true** 라벨을 추가하면 됩니다.

1.4. 추가 리소스

- OpenShift Container Platform에 Operator를 설치하는 방법에 대한 자세한 내용은 [클러스터에 Operator 추가](#) 섹션에서 확인할 수 있습니다.

- Red Hat OpenShift Pipelines Operator를 사용하여 Tekton 체인을 설치하려면 [Red Hat OpenShift Pipelines 공급망 보안에 Tekton 체인 사용을 참조하십시오](#).
- in-cluster Tekton Hub를 설치하고 배포하려면 [Red Hat OpenShift Pipelines에서 Tekton Hub 사용을 참조하십시오](#).
- 제한된 환경에서 파이프라인을 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.
 - 제한된 환경에서 파이프라인을 실행하도록 이미지 미러링
 - 제한된 클러스터에 대한 Samples Operator 구성
 - 미러링된 레지스트리로 클러스터 생성

2장. OPENSIFT PIPELINES 설치 제거

클러스터 관리자는 다음 단계를 수행하여 Red Hat OpenShift Pipelines Operator를 설치 제거할 수 있습니다.

1. 선택적 구성 요소인 **TektonHub** 및 **TektonResult** 의 CR(사용자 정의 리소스)을 삭제하고 **TektonConfig** CR을 삭제합니다.

경고

선택적 구성 요소의 CR을 제거하지 않고 Operator를 설치 제거하는 경우 나중에 구성 요소를 제거할 수 없습니다.


2. Red Hat OpenShift Pipelines Operator를 설치 제거합니다.
3. **operator.tekton.dev** 그룹의 CRD(Custom Resource Definitions)를 삭제합니다.

Operator를 설치 제거하는 것만으로 설치 과정에서 기본적으로 생성된 Red Hat OpenShift Pipelines 구성 요소가 제거되지는 않습니다.

2.1. OPENSIFT PIPELINES 사용자 정의 리소스 삭제

선택적 구성 요소의 CR(사용자 정의 리소스)인 **TektonHub** 및 **TektonResult** 가 있는 경우 이러한 CR을 삭제합니다. 그런 다음 **TektonConfig** CR을 삭제합니다.

프로세스

1. 웹 콘솔의 관리자 화면에서 **Administration** → **CustomResourceDefinitions** 로 이동합니다.
2. **Filter by name** 필드에 **TektonHub** 를 입력하여 **TektonHub** CRD(Custom Resource Definition)를 검색합니다.
3. **TektonHub** CRD의 이름을 클릭하여 CRD 세부 정보 페이지를 표시합니다.
4. **Instances** 탭을 클릭합니다.
5. 인스턴스가 표시되면 표시된 인스턴스의 **옵션** 메뉴  를 클릭합니다.
6. **TektonHub** 삭제를 선택합니다.
7. **Delete** 를 클릭하여 CR 삭제를 확인합니다.
8. 이러한 단계를 반복하여 **TektonResult** 를 검색한 다음 **Filter by name** 상자에서 **TektonConfig** 를 검색합니다. 이러한 CRD에 대한 인스턴스가 있는 경우 이러한 인스턴스를 삭제합니다.



참고

CR을 삭제하면 클러스터의 Red Hat OpenShift Pipelines 구성 요소와 모든 작업 및 파이프라인도 삭제됩니다.



중요

TektonHub 및 **TektonResult** CR을 제거하지 않고 Operator를 설치 제거하는 경우 나중에 Tekton Hub 및 Tekton Results 구성 요소를 제거할 수 없습니다.

2.2. RED HAT OPENSIFT PIPELINES OPERATOR 설치 제거

웹 콘솔의 관리자 화면을 사용하여 Red Hat OpenShift Pipelines Operator를 설치 제거할 수 있습니다.

프로세스

1. **Operators** → **OperatorHub** 페이지에서 키워드로 필터링 박스를 사용하여 **Red Hat OpenShift Pipelines Operator**를 검색합니다.
2. **Red Hat OpenShift Pipelines Operator** 타일을 클릭합니다. Operator 타일은 Operator가 설치되었음을 나타냅니다.
3. **Red Hat OpenShift Pipelines Operator** 설명 페이지에서 **설치 제거**를 클릭합니다.
4. **Operator 설치 제거?** 창에서 이 Operator의 모든 피연산자 인스턴스 삭제를 선택한 다음 **제거**를 클릭합니다.




주의

OpenShift Pipelines Operator를 설치 제거할 때 구성된 시크릿을 포함하여 OpenShift Pipelines가 설치된 **openshift-pipelines** 대상 네임스페이스 내의 모든 리소스가 손실됩니다.

2.3. OPERATOR.TEKTON.DEV 그룹의 사용자 정의 리소스 정의 삭제

operator.tekton.dev 그룹의 CRD(Custom Resource Definitions)를 삭제합니다. 이러한 CRD는 Red Hat OpenShift Pipelines Operator를 설치하는 동안 기본적으로 생성됩니다.

프로세스

1. 웹 콘솔의 관리자 화면에서 **Administration** → **CustomResourceDefinitions** 로 이동합니다.
2. 이름으로 필터링 상자에 **operator.tekton.dev** 를 입력하여 **operator.tekton.dev** 그룹에서 CRD를 검색합니다.
3. 표시된 각 CRD를 삭제하려면 다음 단계를 완료합니다.
 - a. 옵션 메뉴  를 클릭합니다.
 - b. **Delete CustomResourceDefinition** 을 선택합니다.
 - c. **Delete** 를 클릭하여 CRD 삭제를 확인합니다.

추가 리소스

- OpenShift Container Platform 에서 [Operator](#)를 설치 제거하는 방법에 대한 자세한 내용은 클러스터에서 [Operator 삭제](#) 섹션에서 확인할 수 있습니다.

3장. TEKTONCONFIG 사용자 정의 리소스에서 구성 사용자 정의

Red Hat OpenShift Pipelines에서는 **TektonConfig** CR(사용자 정의 리소스)을 사용하여 다음 구성을 사용자 지정할 수 있습니다.

- OpenShift Pipelines 컨트롤러의 고가용성 모드를 포함한 OpenShift Pipelines 성능 최적화
- Red Hat OpenShift Pipelines Control Plane 구성
- 기본 서비스 계정 변경
- 서비스 모니터 비활성화
- 파이프라인 확인기 구성
- 클러스터 작업 및 파이프라인 템플릿 비활성화
- Tekton Hub 통합 비활성화
- RBAC 리소스의 자동 생성 비활성화
- 작업 실행 및 파이프라인 실행 정리

3.1. 사전 요구 사항

- Red Hat OpenShift Pipelines Operator를 설치했습니다.

3.2. TEKTONCONFIG CR을 사용한 성능 튜닝

TektonConfig CR(사용자 정의 리소스)의 **.spec.pipeline.performance** 매개변수 아래에 있는 필드를 수정하여 OpenShift Pipelines 컨트롤러의 HA(고가용성) 지원 및 성능 구성을 변경할 수 있습니다.

TektonConfig 성능 필드의 예

```
apiVersion: operator.tekton.dev/v1alpha1
kind: TektonConfig
metadata:
  name: config
spec:
  pipeline:
    performance:
      disable-ha: false
      buckets: 7
      replicas: 5
      threads-per-controller: 2
      kube-api-qps: 5.0
      kube-api-burst: 10
```

모든 필드는 선택 사항입니다. 이를 설정하면 Red Hat OpenShift Pipelines Operator에 대부분의 필드가 **opensehift-pipelines-controller** 컨테이너의 **opensehift-pipelines-controller** 배포에 인수로 포함됩니다. OpenShift Pipelines Operator는 **opensehift-pipelines** 네임스페이스의 **config-leader-election** 구성 맵의 **buckets** 필드도 업데이트합니다.

값을 지정하지 않으면 OpenShift Pipelines Operator에서 해당 필드를 업데이트하지 않고 OpenShift Pipelines 컨트롤러의 기본값을 적용합니다.



참고

성능 필드를 수정하거나 제거하는 경우 OpenShift Pipelines Operator는 **openshift-pipelines-controller** 배포 및 **config-leader-election** 구성 맵(**buckets** 필드가 변경된 경우) 구성 맵을 업데이트하고 **openshift-pipelines-controller** Pod를 다시 생성합니다.

HA(고가용성) 모드는 파이프라인 실행 및 작업 실행 정의를 기반으로 Pod를 생성하고 시작하는 OpenShift Pipelines 컨트롤러에 적용됩니다. HA 모드가 없으면 단일 Pod가 이러한 작업을 실행하여 로드가 길어지고 상당한 지연이 발생할 수 있습니다.

HA 모드에서 OpenShift Pipelines는 여러 Pod(복제본)를 사용하여 이러한 작업을 실행합니다. 처음에는 OpenShift Pipelines에서 모든 컨트롤러 작업을 버킷에 할당합니다. 각 복제본은 하나 이상의 버킷에서 작업을 선택합니다. 두 복제본이 동시에 동일한 작업을 선택할 수 있는 경우 컨트롤러는 내부적으로 이 작업을 실행하는 *리더*를 결정합니다.

HA 모드는 Pod를 생성한 후 작업 실행 실행에 영향을 미치지 않습니다.

표 3.1. OpenShift Pipelines 성능 튜닝을 위한 수정 가능한 필드

이름	설명	OpenShift Pipelines 컨트롤러의 기본값
disable-ha	HA(고가용성) 모드를 활성화하거나 비활성화합니다. 기본적으로 HA 모드는 활성화됩니다.	false
버킷	HA 모드에서 컨트롤러 작업을 처리하는 데 사용되는 버킷 수입니다. 최대값은 10 입니다.	1
replicas	HA 모드에서 컨트롤러 작업을 처리하기 위해 생성된 Pod 수입니다. 이 값을 buckets 값보다 동일하거나 더 낮은 수로 설정합니다.	1
threads-per-controller	OpenShift Pipelines 컨트롤러의 작업 대기열이 처리될 때 사용할 스레드(작업자) 수입니다.	2
kube-api-qps	REST 클라이언트에서 클러스터 마스터에 대한 초당 최대 쿼리(QPS)입니다.	5.0
kube-api-burst	스로틀의 최대 버스트입니다.	10



참고

OpenShift Pipelines Operator는 OpenShift Pipelines 컨트롤러의 복제본 수를 제어하지 않습니다. 배포의 **replicas** 설정에 따라 복제본 수가 결정됩니다. 예를 들어 복제본 수를 3으로 변경하려면 다음 명령을 입력합니다.

```
$ oc --namespace openshift-pipelines scale deployment openshift-pipelines-controller --replicas=3
```



중요

OpenShift Pipelines 컨트롤러에서 **kube-api-qps** 및 **kube-api-burst** 필드에 2를 곱합니다. 예를 들어 **kube-api-qps** 및 **kube-api-burst** 값이 10이면 실제 QPS 및 burst 값은 20이 됩니다.

3.3. RED HAT OPENSIFT PIPELINES CONTROL PLANE 구성

TektonConfig CR(사용자 정의 리소스)에서 구성 필드를 편집하여 OpenShift Pipelines 컨트롤 플레인을 사용자 지정할 수 있습니다. Red Hat OpenShift Pipelines Operator는 OpenShift Pipelines 컨트롤 플레인을 사용할 수 있도록 기본값을 사용하여 구성 필드를 자동으로 추가합니다.

프로세스

1. 웹 콘솔의 관리자 화면에서 **Administration** → **CustomResourceDefinitions** 로 이동합니다.
2. 이름으로 검색 상자를 사용하여 **tektonconfigs.operator.tekton.dev** CRD(사용자 정의 리소스 정의)를 검색합니다. **TektonConfig** 를 클릭하여 CRD 세부 정보 페이지를 확인합니다.
3. **Instances** 탭을 클릭합니다.
4. **config** 인스턴스를 클릭하여 **TektonConfig** CR 세부 정보를 확인합니다.
5. **YAML** 탭을 클릭합니다.
6. 요구 사항에 따라 **TektonConfig** YAML 파일을 편집합니다.

기본값이 있는 TektonConfig CR의 예

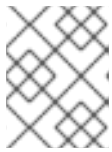
```
apiVersion: operator.tekton.dev/v1alpha1
kind: TektonConfig
metadata:
  name: config
spec:
  pipeline:
    running-in-environment-with-injected-sidecars: true
    metrics.taskrun.duration-type: histogram
    metrics.pipelinerun.duration-type: histogram
    await-sidecar-readiness: true
  params:
    - name: enableMetrics
      value: 'true'
  default-service-account: pipeline
  require-git-ssh-secret-known-hosts: false
  enable-tekton-oci-bundles: false
```

```
metrics.taskrun.level: task
metrics.pipelinerun.level: pipeline
enable-api-fields: stable
enable-provenance-in-status: false
enable-custom-tasks: true
disable-creds-init: false
disable-affinity-assistant: true
```

3.3.1. 기본값을 사용하여 수정 가능한 필드

다음 목록에는 **TektonConfig** CR에 기본값이 있는 모든 수정 가능한 필드가 포함되어 있습니다.

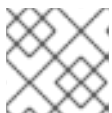
- **running-in-environment-with-injected-sidecars** (기본값: **true**): Istio와 같은 삽입된 사이드카를 사용하지 않는 클러스터에서 파이프라인이 실행되는 경우 이 필드를 **false** 로 설정합니다. **false** 로 설정하면 파이프라인이 작업 실행을 시작하는 데 걸리는 시간이 줄어듭니다.



참고

삽입된 사이드카를 사용하는 클러스터의 경우 이 필드를 **false** 로 설정하면 예기치 않은 동작이 발생할 수 있습니다.

- **await-sidecar-readiness** (기본값: **true**): OpenShift Pipelines가 작동을 시작하기 전에 **TaskRun** 사이드카 컨테이너가 실행될 때까지 대기하지 못하도록 이 필드를 **false** 로 설정합니다. 이를 통해 **DownwardAPI** 볼륨 유형을 지원하지 않는 환경에서 작업을 실행할 수 있습니다.
- **default-service-account** (기본값: **pipeline**): 이 필드에는 **TaskRun** 및 **PipelineRun** 리소스에 사용할 기본 서비스 계정 이름이 포함되어 있습니다.
- **require-git-ssh-secret-known-hosts** (기본값: **false**): 이 필드를 **true** 로 설정하려면 Git SSH 시크릿에 **known_hosts** 필드를 포함해야 합니다.
 - Git SSH 시크릿 구성에 대한 자세한 내용은 *추가 리소스* 섹션에서 *Git에 대한 SSH 인증 구성* 을 참조하십시오.
- **enable-tekton-oci-bundles** (기본값: **false**): Tekton OCI 번들이라는 실험적인 알파 기능을 사용할 수 있도록 이 필드를 **true** 로 설정합니다.
- **enable-api-fields** (기본값: **stable**): 이 필드를 설정하면 활성화된 기능이 결정됩니다. 허용 가능한 값은 **stable**, **beta** 또는 **alpha** 입니다.



참고

Red Hat OpenShift Pipelines는 **알파** 값을 지원하지 않습니다.

- **enable-provenance-in-status** (기본값: **false**): **TaskRun** 및 **PipelineRun** 상태의 검증 필드를 채울 수 있도록 이 필드를 **true** 로 설정합니다. **provenance** 필드에는 **원격 작업** 또는 **파이프라인** 정의가 가져온 소스의 소스와 같이 작업 실행 및 파이프라인 실행에 사용되는 리소스에 대한 메타데이터가 포함되어 있습니다.
- **enable-custom-tasks** (기본값: **true**): 파이프라인에서 사용자 지정 작업 사용을 비활성화하려면 이 필드를 **false** 로 설정합니다.

- **disable-creds-init** (기본값: **false**): OpenShift Pipelines에서 연결된 서비스 계정을 스캔하고 단계에 인증 정보를 삽입하지 못하도록 이 필드를 **true** 로 설정합니다.
- **disable-affinity-assistant** (기본값: **true**): 영구 볼륨 클레임 작업 공간을 공유하는 각 TaskRun 리소스의 선호도 도우미를 활성화하려면 이 필드를 **false** 로 설정합니다.

메트릭 옵션

TektonConfig CR에서 다음 메트릭 필드의 기본값을 수정할 수 있습니다.

- **metrics.taskrun.duration-type** 및 **metrics.pipelinerun.duration-type** (기본값: 히스토그램): 이러한 필드를 설정하면 작업 또는 파이프라인 실행의 기간 유형이 결정됩니다. 허용 가능한 값은 게이지 또는 히스토그램입니다.
- **metrics.taskrun.level** (기본값: 작업): 이 필드는 작업 실행 메트릭의 수준을 결정합니다. 허용 가능한 값은 **taskrun**, **task** 또는 **namespace** 입니다.
- **metrics.pipelinerun.level** (기본값: pipeline): 이 필드는 파이프라인 실행 메트릭의 수준을 결정합니다. 허용되는 값은 **pipelinerun**, **pipeline** 또는 **namespace** 입니다.

3.3.2. 선택적 구성 필드

다음 필드에는 기본값이 없으며 구성하는 경우에만 간주됩니다. 기본적으로 Operator는 TektonConfig CR(사용자 정의 리소스)에서 이러한 필드를 추가하고 구성하지 않습니다.

- **default-timeout-minutes**: 이 필드는 생성할 때 지정되지 않은 경우 TaskRun 및 PipelineRun 리소스에 대한 기본 시간 초과를 설정합니다. 작업 실행 또는 파이프라인 실행이 설정된 시간(분)보다 더 많은 시간이 걸리면 작업 실행 또는 파이프라인 실행이 시간 초과되고 취소됩니다. 예를 들어 **default-timeout-minutes: 60** 은 60분을 기본값으로 설정합니다.
- **default-managed-by-label-value**: 이 필드에는 모든 TaskRun Pod에 적용되는 **app.kubernetes.io/managed-by** 라벨에 지정된 기본값이 포함되어 있습니다. 예: **default-managed-by-label-value: tekton-pipelines**.
- **default-pod-template**: 이 필드는 지정되지 않은 경우 기본 TaskRun 및 PipelineRun Pod 템플릿을 설정합니다.

- **default-cloud-events-sink:** 이 필드는 지정되지 않은 경우 **TaskRun** 및 **PipelineRun** 리소스에 사용되는 기본 **CloudEvents** 싱크를 설정합니다.
- **default-task-run-workspace-binding:** 이 필드에는 **Task** 리소스에서 선언하는 작업 공간에 대한 기본 작업 공간 구성이 포함되어 있지만 **TaskRun** 리소스는 명시적으로 선언하지 않습니다.
- **default-affinity-assistant-pod-template:** 이 필드는 유사성 도우미 **Pod**에 사용되는 기본 **PipelineRun Pod** 템플릿을 설정합니다.
- **default-max-matrix-combinations-count:** 이 필드에는 매트릭스에서 생성된 기본 최대 조합 수가 포함됩니다.

3.4. OPENSIFT PIPELINES의 기본 서비스 계정 변경

`.spec.pipeline` 및 `.spec.trigger` 사양에서 **default-service-account** 필드를 편집하여 **OpenShift Pipelines**의 기본 서비스 계정을 변경할 수 있습니다. 기본 서비스 계정 이름은 **pipeline** 입니다.

예

```

apiVersion: operator.tekton.dev/v1alpha1
kind: TektonConfig
metadata:
  name: config
spec:
  pipeline:
    default-service-account: pipeline
  trigger:
    default-service-account: pipeline
  enable-api-fields: stable

```

3.5. OPENSIFT PIPELINES 설치 네임스페이스에 대한 라벨 및 주석 설정

Operator가 **OpenShift Pipelines**를 설치하는 **openshift-pipelines** 네임스페이스에 대한 레이블 및 주석을 설정할 수 있습니다.



참고

openshift-pipelines 네임스페이스의 이름 변경은 지원되지 않습니다.

TektonConfig CR(사용자 정의 리소스)의 **spec.targetNamespaceMetadata** 사양에 추가하여 레이블 및 주석을 지정합니다.

openshift-pipelines 네임스페이스에 대한 레이블 및 주석 설정 예

```
apiVersion: operator.tekton.dev/v1
kind: TektonConfig
metadata:
  name: config
spec:
  targetNamespaceMetadata:
    labels: {"example-label":"example-value"}
    annotations: {"example-annotation":"example-value"}
```

3.6. 서비스 모니터 비활성화

OpenShift Pipelines의 일부인 서비스 모니터를 비활성화하여 **Telemetry** 데이터를 노출할 수 있습니다. 서비스 모니터를 비활성화하려면 **TektonConfig CR**(사용자 정의 리소스)의 **.spec.pipeline** 사양에서 **enableMetrics** 매개변수를 **false** 로 설정합니다.

예

```
apiVersion: operator.tekton.dev/v1alpha1
kind: TektonConfig
metadata:
  name: config
spec:
  pipeline:
    params:
      - name: enableMetrics
        value: 'false'
```

3.7. 파이프라인 확인기 구성

TektonConfig CR(사용자 정의 리소스)에서 파이프라인 해석기를 구성할 수 있습니다. 이러한 파이프라인 확인기를 활성화하거나 비활성화할 수 있습니다.

- `enable-bundles-resolver`
- `enable-cluster-resolver`
- `enable-git-resolver`
- `enable-hub-resolver`

예

```
apiVersion: operator.tekton.dev/v1alpha1
kind: TektonConfig
metadata:
  name: config
spec:
  pipeline:
    enable-bundles-resolver: true
    enable-cluster-resolver: true
    enable-git-resolver: true
    enable-hub-resolver: true
```

TektonConfig CR에 해결자 특정 구성을 제공할 수도 있습니다. 예를 들어 `map[string]string` 형식으로 다음 필드를 정의하여 각 파이프라인 확인자에 대한 구성을 설정합니다.

예

```
apiVersion: operator.tekton.dev/v1alpha1
kind: TektonConfig
metadata:
```

```

name: config
spec:
  pipeline:
    bundles-resolver-config:
      default-service-account: pipelines
    cluster-resolver-config:
      default-namespace: test
    git-resolver-config:
      server-url: localhost.com
    hub-resolver-config:
      default-tekton-hub-catalog: tekton

```

3.8. 클러스터 작업 및 파이프라인 템플릿 비활성화

기본적으로 **TektonAddon CR**(사용자 정의 리소스)은 클러스터의 **OpenShift Pipelines**와 함께 **clusterTasks** 및 **pipelineTemplates** 리소스를 설치합니다.

.spec.addon 사양에서 매개변수 값을 **false** 로 설정하여 **clusterTasks** 및 **pipelineTemplates** 리소스 설치를 비활성화할 수 있습니다. 또한 **communityClusterTasks** 매개변수를 비활성화할 수 있습니다.

예

```

apiVersion: operator.tekton.dev/v1alpha1
kind: TektonConfig
metadata:
  name: config
spec:
  addon:
    params:
      - name: clusterTasks
        value: 'false'
      - name: pipelineTemplates
        value: 'false'
      - name: communityClusterTasks
        value: 'true'

```



중요

Red Hat OpenShift Pipelines 1.10에서 ClusterTask 기능은 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다.

3.9. TEKTON HUB 통합 비활성화

TektonConfig CR(사용자 정의 리소스)에서 `enable-devconsole-integration` 매개변수를 `false` 로 설정하여 웹 콘솔 개발자 화면에서 Tekton Hub의 통합을 비활성화할 수 있습니다.

Tekton Hub 비활성화 예

```
apiVersion: operator.tekton.dev/v1alpha1
kind: TektonConfig
metadata:
  name: config
spec:
  hub:
    params:
      - name: enable-devconsole-integration
        value: false
```

3.10. RBAC 리소스의 자동 생성 비활성화

Red Hat OpenShift Pipelines Operator의 기본 설치는 `^(openshift|kube)-*` 정규식 패턴과 일치하는 네임스페이스를 제외하고 클러스터의 모든 네임스페이스에 대해 여러 개의 RBAC(역할 기반 액세스 제어) 리소스를 생성합니다. 이러한 RBAC 리소스 중에서 `pipelines-scc-rolebinding` SCC(보안 컨텍스트 제약 조건) 역할 바인딩 리소스는 연결된 `pipelines-scc` SCC에 `RunAsAny` 권한이 있으므로 잠재적인 보안 문제입니다.

Red Hat OpenShift Pipelines Operator가 설치된 후 클러스터 전체 RBAC 리소스의 자동 생성을 비활성화하려면 클러스터 관리자가 클러스터 수준 TektonConfig CR(사용자 정의 리소스)에서 `createRbacResource` 매개변수를 `false` 로 설정할 수 있습니다.

TektonConfig CR의 예

```
apiVersion: operator.tekton.dev/v1alpha1
```



```

kind: TektonConfig
metadata:
  name: config
spec:
  params:
    - name: createRbacResource
      value: "false"
  ...

```



주의

클러스터 관리자 또는 적절한 권한이 있는 사용자는 모든 네임스페이스에 대한 **RBAC** 리소스 자동 생성을 비활성화하면 기본 **ClusterTask** 리소스가 작동하지 않습니다. **ClusterTask** 리소스가 작동하려면 의도한 각 네임스페이스에 대해 **RBAC** 리소스를 수동으로 생성해야 합니다.

3.11. 파이프라인 및 작업의 인라인 사양 비활성화

기본적으로 **OpenShift Pipelines**는 다음과 같은 경우 파이프라인 및 작업의 인라인 사양을 지원합니다.

- 다음 예와 같이 하나 이상의 작업 사양을 포함하는 **Pipeline CR**을 생성할 수 있습니다.

Pipeline CR의 인라인 사양의 예

```

apiVersion: operator.tekton.dev/v1
kind: Pipeline
metadata:
  name: pipelineInline
spec:
  tasks:
    taskSpec:
# ...

```

- 다음 예와 같이 파이프라인 사양을 포함하는 **PipelineRun CR**(사용자 정의 리소스)을 생성할 수 있습니다.

PipelineRun CR의 인라인 사양의 예

```
apiVersion: operator.tekton.dev/v1
kind: PipelineRun
metadata:
  name: pipelineRunInline
spec:
  pipelineSpec:
    tasks:
# ...
```

- 다음 예와 같이 작업 사양을 포함하는 **TaskRun CR**(사용자 정의 리소스)을 생성할 수 있습니다.

TaskRun CR의 인라인 사양의 예

```
apiVersion: operator.tekton.dev/v1
kind: TaskRun
metadata:
  name: taskRunInline
spec:
  taskSpec:
    steps:
# ...
```

이러한 경우 일부 또는 전부에서 인라인 사양을 비활성화할 수 있습니다. 인라인 사양을 비활성화하려면 다음 예와 같이 **TektonConfig CR**의 **.spec.pipeline** 사양의 **disable-inline-spec** 필드를 설정합니다.

인라인 사양을 비활성화하는 구성 예

```

apiVersion: operator.tekton.dev/v1alpha1
kind: TektonConfig
metadata:
  name: config
spec:
  pipeline:
    disable-inline-spec: "pipeline,pipelinerun,taskrun"
# ...

```

disable-inline-spec 매개변수를 단일 값 또는 여러 값의 쉼표로 구분된 목록으로 설정할 수 있습니다. 매개변수의 다음 값은 유효합니다.

표 3.2. **disable-inline-spec** 매개변수에 지원되는 값

현재의	설명
pipeline	taskSpec: spec을 사용하여 Pipeline CR 내에 작업을 정의할 수 없습니다. 대신 taskRef: spec을 사용하여 Task CR의 작업을 통합하거나 확인자를 사용하여 작업을 지정해야 합니다.
pipelinerun	pipelineSpec: spec을 사용하여 PipelineRun CR 내에 파이프라인을 정의할 수 없습니다. 대신 pipelineRef: spec을 사용하여 Pipeline CR의 파이프라인을 통합하거나 확인자를 사용하여 파이프라인을 지정해야 합니다.
taskrun	taskSpec: spec을 사용하여 TaskRun CR 내에 작업을 정의할 수 없습니다. 대신 taskRef: spec을 사용하여 Task CR의 작업을 통합하거나 확인자를 사용하여 작업을 지정해야 합니다.

3.12. 작업 실행 및 파이프라인 실행 자동 정리

오래된 **TaskRun** 및 **PipelineRun** 오브젝트와 실행된 인스턴스는 활성 실행에 사용할 수 있는 물리적 리소스를 차지합니다. 이러한 리소스를 최적으로 사용하기 위해 **Red Hat OpenShift Pipelines**는 다양한 네임스페이스에서 사용하지 않는 오브젝트와 해당 인스턴스를 자동으로 제거하는 정리기 구성 요소를 제공합니다.



참고

TektonConfig 사용자 정의 리소스를 사용하여 전체 설치에 대한 정리를 구성하고 네임스페이스 주석을 사용하여 네임스페이스 구성을 수정할 수 있습니다. 그러나 네임스페이스에서 개별 작업 실행 또는 파이프라인 실행을 선택적으로 자동 실행할 수 없습니다.

3.12.1. pruner 구성

TektonConfig 사용자 지정 리소스를 사용하여 파이프라인 실행 및 작업 실행과 관련된 리소스의 주기적 정리를 구성할 수 있습니다.

다음 예제는 기본 구성에 해당합니다.

정리기 구성의 예

```

apiVersion: operator.tekton.dev/v1alpha1
kind: TektonConfig
metadata:
  name: config
  # ...
spec:
  pruner:
    resources:
      - taskrun
      - pipelinerun
    keep: 100
    prune-per-resource: false
    schedule: "* 8 * * *"
    startingDeadlineSeconds: 60
  # ...
    
```

표 3.3. 정리기 구성에 지원되는 매개변수

매개변수	설명
스케줄	정리 프로세스 실행을 위한 Cron 스케줄입니다. 기본 일정은 매일 08:00에 프로세스를 실행합니다. Cron 일정 구문에 대한 자세한 내용은 Kubernetes 문서의 Cron schedule 구문 을 참조하십시오.

매개변수	설명
resources	pruner가 적용되는 리소스 유형입니다. 사용 가능한 리소스 유형은 taskrun 및 pipelinerun 입니다.
keep	유지할 모든 유형의 최근 리소스 수입니다.
prune-per-resource	false 로 설정하면 keep 매개변수의 값이 총 작업 실행 또는 파이프라인 실행 수를 나타냅니다. 예를 들어 keep 가 100 으로 설정된 경우 pruner는 최신 작업 실행 및 100개의 최신 파이프라인 실행과 기타 모든 리소스를 제거합니다. true 로 설정하면 각 파이프라인을 참조하는 파이프라인 실행과 각 작업을 참조하는 작업 실행에 대해 keep 매개변수의 값이 별도로 계산됩니다. 예를 들어 keep 가 100 으로 설정된 경우 pruner는 Pipeline1 에 대해 최신 파이프라인 실행 100개, Pipeline2 용 최신 파이프라인 실행, Task1 에 대한 최신 작업 실행 100개를 유지하고 다른 모든 리소스를 제거합니다.
keep-since	리소스를 유지할 최대 시간(분)입니다. 예를 들어 5일 전에 생성된 리소스를 유지하려면 keep-since 를 7200 으로 설정합니다.
startingDeadlineSeconds	이 매개변수는 선택 사항입니다. 예약된 시간에 어떤 이유로든 정리기 작업이 시작되지 않는 경우 이 설정은 작업을 계속 시작할 수 있는 최대 시간(초)을 구성합니다. 지정된 시간 내에 작업이 시작되지 않으면 OpenShift Pipelines에서 이 작업이 실패했다고 간주하고 다음 예약된 시간에 정리기를 시작합니다. 이 매개변수를 지정하지 않고 예약된 시간에 pruner 작업이 시작되지 않으면 OpenShift Pipelines에서 작업을 나중에 시작할 수 있습니다.



참고

keep 및 **keep-since** 매개변수는 함께 사용할 수 없습니다. 구성에 해당 중 하나만 사용합니다.

3.12.2. 작업 실행 및 파이프라인 실행을 자동으로 정리하기 위한 주석

네임스페이스에서 작업 실행 및 파이프라인 실행 자동 정리 구성을 수정하려면 네임스페이스에 주석을 설정할 수 있습니다.

다음 namespace 주석은 TektonConfig 사용자 정의 리소스의 해당 키와 동일한 의미를 갖습니다.

- `operator.tekton.dev/prune.schedule`
- `operator.tekton.dev/prune.resources`
- `operator.tekton.dev/prune.keep`
- `operator.tekton.dev/prune.prune-per-resource`
- `operator.tekton.dev/prune.keep-since`



참고

`operator.tekton.dev/prune.resources` 주석은 쉼표로 구분된 목록을 허용합니다. 작업 실행 및 파이프라인 실행을 모두 정리하려면 이 주석을 "`taskrun, pipelinerun`" 으로 설정합니다.

다음과 같은 추가 네임스페이스 주석을 사용할 수 있습니다.

- `operator.tekton.dev/prune.skip: true` 로 설정하면 주석이 구성된 네임스페이스가 정리되지 않습니다.
- `operator.tekton.dev/prune.strategy`: 이 주석의 값을 `keep` 또는 `keep-since` 로 설정합니다.

예를 들어 다음 주석은 지난 5일 동안 생성된 모든 작업 실행 및 파이프라인 실행을 유지하고 이전 리소스를 삭제합니다.

자동 실행 주석의 예

```
kind: Namespace
apiVersion: v1
# ...
spec:
```

```
annotations:  
  operator.tekton.dev/prune.resources: "taskrun, pipelinerun"  
  operator.tekton.dev/prune.keep-since: 7200  
# ...
```

3.13. 추가 리소스

- [보안을 사용하여 리포지터리로 파이프라인 인증](#)
- [버전이 지정되지 않은 클러스터 작업 관리](#)
- [관리자 화면에서 파이프라인 템플릿 생성](#)
- [리소스 회수를 위한 오브젝트 정리](#)