



Red Hat OpenShift Pipelines 1.14

インストールおよび設定

OpenShift パイプラインのインストールと設定

Red Hat OpenShift Pipelines 1.14 インストールおよび設定

OpenShift パイプラインのインストールと設定

法律上の通知

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.

概要

このドキュメントでは、OpenShift Pipelines のインストールと設定に関する情報を提供します。

目次

第1章 OPENSIFT PIPELINES のインストール	3
前提条件	3
1.1. WEB コンソールでの RED HAT OPENSIFT PIPELINES OPERATOR のインストール	3
1.2. CLI を使用した OPENSIFT PIPELINES OPERATOR のインストール	6
1.3. 制限された環境での RED HAT OPENSIFT PIPELINES OPERATOR	6
1.4. 関連情報	7
第2章 OPENSIFT PIPELINES のアンインストール	8
2.1. RED HAT OPENSIFT PIPELINES コンポーネントおよびカスタムリソースの削除	8
2.2. RED HAT OPENSIFT PIPELINES OPERATOR のアンインストール	9
第3章 TEKTONCONFIG カスタムリソース設定のカスタマイズ	10
3.1. 前提条件	10
3.2. TEKTONCONFIG CR を使用したパフォーマンスチューニング	10
3.3. RED HAT OPENSIFT PIPELINES コントロールプレーンの設定	12
3.4. OPENSIFT PIPELINES のデフォルトサービスアカウントの変更	15
3.5. サービス 모니터の無効化	15
3.6. パイプラインリゾルバーの設定	15
3.7. クラスタタスクとパイプラインテンプレートの無効化	16
3.8. TEKTON HUB 統合の無効化	17
3.9. RBAC リソースの自動作成の無効化	17
3.10. タスク実行とパイプライン実行の自動プルーニング	18
3.11. 関連情報	20

第1章 OPENSIFT PIPELINES のインストール

以下では、クラスター管理者を対象に、Red Hat OpenShift Pipelines Operator の OpenShift Container Platform クラスターへのインストールプロセスについて説明します。

前提条件

- **cluster-admin** パーミッションを持つアカウントを使用して OpenShift Container Platform クラスターにアクセスできる。
- **oc** CLI がインストールされている。
- **OpenShift Pipelines (tkn) CLI** がローカルシステムにインストールされている。
- クラスターで **Marketplace 機能** が有効になっているか、Red Hat Operator カタログソースが手動で設定されている。



注記

Windows ノードと Linux ノードの両方が含まれるクラスターでは、Red Hat OpenShift Pipelines は Linux ノードでのみ実行できます。

1.1. WEB コンソールでの RED HAT OPENSIFT PIPELINES OPERATOR のインストール

OpenShift Container Platform OperatorHub にリスト表示されている Operator を使用して Red Hat OpenShift Pipelines をインストールできます。Red Hat OpenShift Pipelines Operator をインストールする際に、パイプラインの設定に必要なカスタムリソース (CR) が Operator と共に自動的にインストールされます。

デフォルトの Operator カスタムリソース定義 (CRD) の **config.operator.tekton.dev** が **tektonconfigs.operator.tekton.dev** に置き換えられました。さらに Operator は、個別に管理される OpenShift Pipelines コンポーネントに追加の CRD (**tektonpipelines.operator.tekton.dev**、**tektontriggers.operator.tekton.dev** および **tektonaddons.operator.tekton.dev**) を提供します。

OpenShift Pipelines がクラスターにすでにインストールされている場合、既存のインストールはシームレスにアップグレードされます。Operator は必要に応じて、クラスターの **config.operator.tekton.dev** のインスタンスを **tektonconfigs.operator.tekton.dev** のインスタンスと、その他の CRD の追加オブジェクトに置き換えます。



警告

既存のインストールを手動で変更した場合 (**resource name - cluster** フィールドに変更を加えて **config.operator.tekton.dev** CRD インスタンスのターゲット namespace を変更する場合など)、アップグレードパスはスムーズではありません。このような場合は、インストールをアンインストールして Red Hat OpenShift Pipelines Operator を再インストールするワークフローが推奨されます。

Red Hat OpenShift Pipelines Operator では、**TektonConfig** カスタムリソース (CR) の一部としてプロ

ファイルを指定することで、インストールするコンポーネントを選択するオプションが提供されるようになりました。**TektonConfig** CR は Operator のインストール時に自動的にインストールされます。サポートされるプロファイルは以下のとおりです。

- Lite: Tekton パイプラインのみをインストールします。
- 基本: Tekton Pipelines、Tekton Triggers、および Tekton Chains をインストールします。
- All: **TektonConfig** CR のインストール時に使用されるデフォルトプロファイルです。このプロファイルは、Tekton Pipelines、Tekton Triggers、Tekton Chains、Pipelines as Code、Tekton Addons を含むすべての Tekton コンポーネントをインストールします。Tekton Addons には、**ClusterTasks**、**ClusterTriggerBindings**、**ConsoleCLIDownload**、**ConsoleQuickStart**、および **ConsoleYAMLSample** リソースが含まれています。

手順

1. Web コンソールの **Administrator** パースペクティブで、**Operators** → **OperatorHub** に移動します。
2. **Filter by keyword** ボックスを使用して、カタログで **Red Hat OpenShift Pipelines Operator** を検索します。**Red Hat OpenShift Pipelines Operator** タイルをクリックします。
3. **Red Hat OpenShift Pipelines Operator** ページで Operator についての簡単な説明を参照してください。**Install** をクリックします。
4. **Install Operator** ページで以下を行います。
 - a. **Installation Mode** で **All namespaces on the cluster (default)** を選択します。このモードは、デフォルトの **openshift-operators** namespace に Operator をインストールします。これにより、Operator はクラスター内のすべての namespace を監視し、これらの namespace に対して利用可能になります。
 - b. **Approval Strategy** で **Automatic** を選択します。これにより、Operator への今後のアップグレードは Operator Lifecycle Manager (OLM) によって自動的に処理されます。**Manual** 承認ストラテジーを選択すると、OLM は更新要求を作成します。クラスター管理者は、Operator を新規バージョンに更新できるように OLM 更新要求を手動で承認する必要があります。
 - c. **Update Channel** を選択します。
 - **latest** チャンネルにより、Red Hat OpenShift Pipelines Operator の最新の stable バージョンがインストール可能になります。これは、Red Hat OpenShift Pipelines Operator のインストールに使用する現在のデフォルトチャンネルです。
 - Red Hat OpenShift Pipelines Operator の特定のバージョンをインストールする場合、クラスター管理者は対応する **pipelines-<version>** チャンネルを使用できます。たとえば、Red Hat OpenShift Pipelines Operator バージョン **1.8.x** をインストールするには、**pipelines-1.8** チャンネルを使用できます。



注記

OpenShift Container Platform 4.11 以降、Red Hat OpenShift Pipelines Operator をインストールおよびアップグレードするための **preview** および **stable** チャンネルは利用できません。ただし、OpenShift Container Platform 4.10 以前のバージョンでは、**preview** および **stable** チャンネルを使用して Operator をインストールおよびアップグレードできます。

5. **Install** をクリックします。Operator が **Installed Operators** ページにリスト表示されます。



注記

Operator は **openshift-operators** namespace に自動的にインストールされます。

6. **Status** が **Succeeded Up to date** に設定され、Red Hat OpenShift Pipelines Operator のインストールが正常に行われたことを確認します。



警告

他のコンポーネントのインストールが進行中の場合でも、成功ステータスが **Succeeded Up to date** として表示される場合があります。したがって、ターミナルで手動でインストールを確認することが重要です。

7. Red Hat OpenShift Pipelines Operator のすべてのコンポーネントが正常にインストールされたことを確認します。ターミナルでクラスターにログインし、次のコマンドを実行します。

```
$ oc get tektonconfig config
```

出力例

```
NAME      VERSION  READY  REASON
config    1.14.0   True
```

READY 条件が **True** の場合、Operator とそのコンポーネントは正常にインストールされています。

さらに、次のコマンドを実行して、コンポーネントのバージョンを確認します。

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

出力例

```
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. CLI を使用した OPENSIFT PIPELINES OPERATOR のインストール

CLI を使用して OperatorHub から Red Hat OpenShift Pipelines Operator をインストールできます。

手順

- Subscription オブジェクトの YAML ファイルを作成し、namespace を Red Hat OpenShift Pipelines Operator にサブスクライブします (例: **sub.yaml**)。

Subscription の例

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: openshift-pipelines-operator
  namespace: openshift-operators
spec:
  channel: <channel name> ❶
  name: openshift-pipelines-operator-rh ❷
  source: redhat-operators ❸
  sourceNamespace: openshift-marketplace ❹
```

- ❶ Operator のチャンネル名。デフォルトチャンネルは **pipelines-<version>** です。たとえば、Red Hat OpenShift Pipelines Operator バージョン **1.7** のデフォルトチャンネルは **pipelines-1.7** です。 **latest** チャンネルにより、Red Hat OpenShift Pipelines Operator の最新の stable バージョンがインストール可能になります。
- ❷ サブスクライブする Operator の名前。
- ❸ Operator を提供する CatalogSource の名前。
- ❹ CatalogSource の namespace。デフォルトの OperatorHub CatalogSource には **openshift-marketplace** を使用します。

- Subscription オブジェクトを作成します。

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

サブスクリプションにより、Red Hat OpenShift Pipelines Operator が **openshift-operators** namespace にインストールされます。Operator は、OpenShift Pipelines をデフォルトの **openshift-pipelines** ターゲット namespace に自動的にインストールします。

1.3. 制限された環境での RED HAT OPENSIFT PIPELINES OPERATOR

Red Hat OpenShift Pipelines Operator は、ネットワークが制限された環境でのパイプラインのインストールに対するサポートを有効にします。

Operator は、**cluster** プロキシオブジェクトに基づいて tekton-controllers によって作成される Pod

のコンテナにプロキシ環境変数を設定するプロキシ Webhook をインストールします。また、プロキシ環境変数を **TektonPipelines**、**TektonTriggers**、**Controllers**、**Webhooks**、および **Operator Proxy Webhook** リソースに設定します。

デフォルトで、プロキシ Webhook は **openshift-pipelines** namespace に対して無効にされます。他の namespace に対してこれを無効にするには、**operator.tekton.dev/disable-proxy: true** ラベルを **namespace** オブジェクトに追加します。

1.4. 関連情報

- OpenShift Container Platform への Operator のインストールの詳細は、[クラスターへの Operator の追加](#) セクションを参照してください。
- Red Hat OpenShift Pipelines Operator を使用して Tekton Chains をインストールするには、[Using Tekton Chains for Red Hat OpenShift Pipelines supply chain security](#) を参照してください。
- クラスター内の Tekton Hub をインストールしてデプロイするには、[Red Hat OpenShift Pipeline での Tekton Hub の使用](#) を参照してください。
- 制限された環境でパイプラインを使用する方法についての詳細は、以下を参照してください。
 - [制限された環境でパイプラインを実行するためのイメージのミラーリング](#)
 - [制限されたクラスターの Samples Operator の設定](#)
 - [ミラーリングされたレジストリーでのクラスターの作成](#)

第2章 OPENSIFT PIPELINES のアンインストール

クラスター管理者は、以下のステップを実行することにより、Red Hat OpenShift Pipelines Operator をアンインストールできます。

1. Red Hat OpenShift Pipelines Operator のインストール時にデフォルトで追加されたカスタムリソース (CR) を削除します。
2. Operator に依存する Tekton Hub などのオプションコンポーネントの CR を削除します。

注意

オプションコンポーネントの CR を削除せずに Operator をアンインストールした場合、後で削除できません。

3. Red Hat OpenShift Pipelines Operator をアンインストールします。

Operator のみをアンインストールしても、Operator のインストール時にデフォルトで作成される Red Hat OpenShift Pipelines コンポーネントは削除されません。

2.1. RED HAT OPENSIFT PIPELINES コンポーネントおよびカスタムリソースの削除

Red Hat OpenShift Pipelines Operator のインストール時にデフォルトで作成されるカスタムリソース (CR) を削除します。

手順

1. Web コンソールの **Administrator** パースペクティブで、**Administration** → **Custom Resource Definition** に移動します。
2. **Filter by name** ボックスに **config.operator.tekton.dev** を入力し、Red Hat OpenShift Pipelines Operator CR を検索します。
3. **CRD Config** をクリックし、**Custom Resource Definition Details** ページを表示します。
4. **Actions** ドロップダウンメニューをクリックし、**Delete Custom Resource Definition** を選択します。



注記

CR を削除すると、Red Hat OpenShift Pipelines コンポーネントが削除され、クラスター上のすべてのタスクとパイプラインが失われます。

5. **Delete** をクリックし、CR の削除を確認します。



重要

Operator をアンインストールする前に、この手順を繰り返して Tekton Hub などのオプションコンポーネントの CR を見つけ、削除します。オプションコンポーネントの CR を削除せずに Operator をアンインストールした場合、後で削除できません。

2.2. RED HAT OPENSIFT PIPELINES OPERATOR のアンインストール

Web コンソールの **Administrator** パースペクティブを使用して、Red Hat OpenShift Pipelines Operator をアンインストールできます。

手順

1. **Operators** → **OperatorHub** ページから、**Filter by keyword** ボックスを使用して **Red Hat OpenShift Pipelines Operator** を検索します。
2. **Red Hat OpenShift Pipelines Operator** タイルをクリックします。Operator タイルは、Operator がインストールされていることを示します。
3. **Red Hat OpenShift Pipelines Operator** の説明ページで、**Uninstall** をクリックします。



警告

{pipeline-shortname} Operator をアンインストールすると、Operator がインストールされているターゲット namespace (デフォルトでは **openshift-pipelines** namespace) 内のすべてのリソースが、それに設定したシークレットを含めて失われます。

関連情報

- OpenShift Container Platform での Operator のアンインストールの詳細は [クラスターからの Operator の削除](#) セクションを参照してください。

第3章 TEKTONCONFIG カスタムリソース設定のカスタマイズ

Red Hat OpenShift Pipelines では、**TektonConfig** カスタムリソース (CR) を使用して以下の設定をカスタマイズできます。

- OpenShift Pipelines コントローラーの高可用性モードを含む、OpenShift Pipelines パフォーマンスの最適化
- Red Hat OpenShift Pipelines コントロールプレーンの設定
- デフォルトサービスアカウントの変更
- サービスモニターの無効化
- パイプラインリゾルバーの設定
- クラスタタスクとパイプラインテンプレートの無効化
- Tekton Hub 統合の無効化
- RBAC リソースの自動作成の無効化
- タスク実行とパイプライン実行のプルーニング

3.1. 前提条件

- Red Hat OpenShift Pipelines Operator がインストールされている。

3.2. TEKTONCONFIG CR を使用したパフォーマンスチューニング

TektonConfig カスタムリソース (CR) の **.spec.pipeline.performance** パラメーターの下のフィールドを変更して、OpenShift Pipelines コントローラーの高可用性 (HA) サポートとパフォーマンス設定を変更できます。

TektonConfig パフォーマンスフィールドの例

```
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
```

すべてのフィールドは任意です。これらを設定すると、Red Hat OpenShift Pipelines Operator には、**openshift-pipelines-controller** コンテナの下にある **openshift-pipelines-controller** デプロイメントにほとんどのフィールドが引数として含まれます。OpenShift Pipelines Operator は **openshift-pipelines** namespace の下の **config-leader-election** 設定マップの **buckets** フィールドも更新します。

値を指定しない場合、OpenShift Pipelines Operator はそれらのフィールドを更新せず、OpenShift Pipelines コントローラーのデフォルト値を適用します。



注記

いずれかのパフォーマンスフィールドを変更または削除すると、OpenShift Pipelines Operator は **openshift-pipelines-controller** デプロイメントおよび **config-leader-election** 設定マップ (**buckets** フィールドが変更された場合) を更新し、**openshift-pipelines-controller** Pod を再作成します。

高可用性 (HA) モードは、パイプライン実行およびタスク実行定義に基づいて Pod を作成および開始する OpenShift Pipelines コントローラーに適用されます。HA モードを使用しない場合、単一の Pod がこれらの操作を実行するため、高負荷時には大幅な遅延が発生する可能性があります。

HA モードでは、OpenShift Pipelines は複数の Pod (レプリカ) を使用してこれらの操作を実行します。最初に、OpenShift Pipelines はすべてのコントローラー操作をバケットに割り当てます。各レプリカは1つ以上のバケットからオペレーションを選択します。2つのレプリカが同時に同じ操作を選択できる場合、コントローラーはこの操作を実行する **リーダー** を内部で決定します。

HA モードは、Pod の作成後にタスク実行には影響しません。

表3.1 OpenShift Pipelines のパフォーマンスを調整するための変更可能なフィールド

Name	Description	OpenShift Pipelines コントローラーのデフォルト値
disable-ha	高可用性 (HA) モードを有効または無効にします。デフォルトでは、HA モードが有効になっています。	false
buckets	HA モードの場合、コントローラー操作の処理に使用されるバケットの数。最大値は 10 です。	1
replicas	HA モードの場合、コントローラー操作を処理するために作成された Pod の数。この値は buckets 値と同じまたはそれよりも小さい番号に設定します。	1
threads-per-controller	OpenShift Pipelines コントローラーのワークキューが処理されるときに使用するスレッド (ワーカー) の数。	2
kube-api-qps	REST クライアントからクラスターマスターへの1秒あたりの最大クエリー数 (QPS)。	5.0
kube-api-burst	スロットルの最大バースト。	10



注記

OpenShift Pipelines Operator は、OpenShift Pipelines コントローラーのレプリカの数
を制御しません。デプロイメントの **replica** 設定によって、レプリカの数が決ま
ります。たとえば、レプリカ数を 3 に変更するには、次のコマンドを入力し
ます。

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



重要

kube-api-qps と **kube-api-burst** フィールドは、OpenShift Pipelines コントローラーで
2 倍にされます。たとえば、**kube-api-qps** および **kube-api-burst** の値が **10** の場合、実
際の QPS およびバースト値は **20** になります。

3.3. RED HAT OPENSIFT PIPELINES コントロールプレーンの設定

TektonConfig カスタムリソース (CR) の設定フィールドを編集して、OpenShift Pipelines コントロー
ルプレーンをカスタマイズできます。Red Hat OpenShift Pipelines Operator は設定フィールドにデ
フォルト値を自動的に追加し、OpenShift Pipelines コントロールプレーンを使用可能な状態にします。

手順

1. Web コンソールの **Administrator** パースペクティブで、**Administration** → **CustomResourceDefinitions** に移動します。
2. **Search by name** ボックスを使用して、**tektonconfigs.operator.tekton.dev** カスタムリソース
定義 (CRD) を検索します。**TektonConfig** をクリックし、CRD の詳細ページを表示します。
3. **Instances** タブをクリックします。
4. **config** インスタンスをクリックして、**TektonConfig** CR の詳細を表示します。
5. **YAML** タブをクリックします。
6. 要件に応じて **TektonConfig** YAML ファイルを編集します。

デフォルト値が適用された TektonConfig CR の例

```
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
```



```

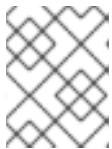
enable-tekton-oci-bundles: false
metrics.taskrun.level: task
metrics.pipelinerun.level: pipeline
enable-api-fields: stable
enable-provenance-in-status: false
enable-custom-tasks: true
disable-creds-init: false
disable-affinity-assistant: true

```

3.3.1. デフォルト値が適用された変更可能フィールド

次のリストには、デフォルト値が適用された **TektonConfig** CR の変更可能フィールドがすべて含まれています。

- **running-in-environment-with-injected-sidecars** (デフォルト: **true**): Istio などの注入済みサイドカーを使用しないクラスターでパイプラインを実行する場合は、このフィールドを **false** に設定します。**false** に設定すると、パイプラインがタスク実行を開始するまでにかかる時間が短縮されます。



注記

注入されたサイドカーを使用するクラスターの場合、このフィールドを **false** に設定すると、予期しない動作が発生する可能性があります。

- **await-sidecar-readiness** (デフォルト: **true**): **TaskRun** サイドカーコンテナの実行を待たずに OpenShift が動作を開始するには、このフィールドを **false** に設定します。これにより、**downwardAPI** ボリュームタイプをサポートしない環境でのタスク実行が可能になります。
- **default-service-account** (デフォルト: **Pipeline**): 特に指定されていない場合、このフィールドには **TaskRun** および **PipelineRun** リソースに使用するデフォルトのサービスアカウント名が設定されます。
- **require-git-ssh-secret-known-hosts** (デフォルト: **false**): このフィールドを **true** に設定するには、Git SSH シークレットに **known_hosts** フィールドが含まれている必要があります。
 - Git SSH シークレットの設定について、詳しくは [関連情報](#) セクションの [Git の SSH 認証の設定](#) を参照してください。
- **Enable-tekton-oci-bundles** (デフォルト: **false**): このフィールドを **true** に設定すると、Tekton OCI バンドルという名前の実験的アルファ機能の使用が可能になります。
- **Enable-api-fields** (デフォルト: **stable**): このフィールドを設定すると、どの機能が有効になるかが決まります。使用できる値は **stable**、**beta**、または **alpha** です。



注記

Red Hat OpenShift Pipelines で **alpha** 値はサポートされていません。

- **Enable-provenance-in-status** (デフォルト: **false**): このフィールドを **true** に設定すると、**TaskRun** ステータスおよび **PipelineRun** ステータスの **provenance** フィールドへの入力が可能になります。**provenance** フィールドには、リモートタスクまたはパイプライン定義の取得元などの、タスク実行およびパイプライン実行で使用されるリソースのメタデータが含まれます。

- **Enable-custom-tasks** (デフォルト: **true**): このフィールドを **false** に設定すると、パイプラインでのカスタムタスクの使用が無効になります。
- **disable-creds-init** (デフォルト: **false**): OpenShift Pipelines が接続されたサービスアカウントをスキャンしてステップに認証情報を挿入しないようにするには、このフィールドを **true** に設定します。
- **disable-affinity-assistant** (デフォルト: **true**): 永続ボリューム要求ワークスペースを共有する各 **TaskRun** リソースに対してアフィニティアシスタントを有効にするには、このフィールドを **false** に設定します。

メトリクスオプション

TektonConfig CR の次のメトリクスフィールドのデフォルト値を変更できます。

- **metrics.taskrun.duration-type** と **metrics.pipelinerun.duration-type** (デフォルト: **histogram**): これらのフィールドを設定すると、タスクまたはパイプライン実行の期間のタイプが決まります。使用できる値は、**gauge** または **histogram** です。
- **metrics.taskrun.level** (デフォルト: **task**): このフィールドにより、タスク実行メトリクスのレベルが決まります。使用できる値は、**taskrun**、**task**、または **namespace** です。
- **metrics.pipelinerun.level** (デフォルト: **Pipeline**): このフィールドにより、パイプライン実行メトリクスのレベルが決まります。使用できる値は、**pipelinerun**、**pipeline**、または **namespace** です。

3.3.2. 任意の設定フィールド

次のフィールドにはデフォルト値がなく、設定した場合にのみ考慮されます。デフォルトでは、Operator はこれらのフィールドを **TektonConfig** カスタムリソース (CR) に追加も設定もしません。

- **default-timeout-minutes**: **TaskRun** および **PipelineRun** リソースの作成時に指定していない場合、このフィールドがデフォルトのタイムアウトを設定します。タスク実行またはパイプライン実行にかかる時間が設定された分数より長いと、タスク実行またはパイプライン実行はタイムアウトになり、キャンセルされます。たとえば、**default-timeout-minutes: 60** はデフォルトを 60 分に設定します。
- **default-managed-by-label-value**: このフィールドには、**app.kubernetes.io/managed-by** ラベルに指定されたデフォルト値が含まれます。このデフォルト値は、何も指定されていない場合にすべての **TaskRun** Pod に適用されます。たとえば、**default-managed-by-label-value: tekton-pipelines** です。
- **default-pod-template**: このフィールドは、指定されていない場合にデフォルトの **TaskRun** および **PipelineRun** Pod テンプレートを設定します。
- **default-cloud-events-sink**: このフィールドは、何も指定されていない場合に、**TaskRun** および **PipelineRun** リソースに使用されるデフォルトの **CloudEvents** シンクを設定します。
- **default-task-run-workspace-binding**: このフィールドには、**Task** リソースが宣言するワークスペースのデフォルトワークスペース設定が含まれますが、**TaskRun** リソースは明示的に宣言されません。
- **default-affinity-assistant-pod-template**: このフィールドは、何も指定されていない場合にアフィニティアシスタント Pod が使用するデフォルトの **PipelineRun** Pod テンプレートを設定します。

- **default-max-matrix-combinations-count**: このフィールドには、何も指定されていない場合の、マトリクスから生成される組み合わせの最大数のデフォルト値が含まれます。

3.4. OPENSIFT PIPELINES のデフォルトサービスアカウントの変更

OpenShift Pipeline のデフォルトサービスアカウントは、**.spec.pipeline** および **.spec.trigger** 仕様の **default-service-account** フィールドを編集して変更できます。デフォルトのサービスアカウントの名前は **pipeline** です。

例

```
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. サービスモニターの無効化

OpenShift Pipeline の一部であるサービスモニターを無効にして、Telemetry データを公開できます。サービスモニターを無効にするには、**TektonConfig** カスタムリソース (CR) の **.spec.pipeline** 仕様で **enableMetrics** パラメーターを **false** に設定します。

例

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

3.6. パイプラインリゾルバーの設定

パイプラインリゾルバーは、**TektonConfig** カスタムリソース (CR) で設定できます。パイプラインリゾルバーは、有効または無効にできます。

- **enable-bundles-resolver**
- **enable-cluster-resolver**
- **enable-git-resolver**
- **enable-hub-resolver**

例

```

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 CR でリゾルバー固有の設定も指定できます。たとえば、**map[string]string** 形式で以下のフィールドを定義し、各パイプラインリゾルバーを設定できます。

例

```

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

```

3.7. クラスタータスクとパイプラインテンプレートの無効化

デフォルトでは、**TektonAddon** カスタムリソース (CR) は、クラスター上の OpenShift Pipeline と併せて **clusterTasks** および **pipelineTemplates** リソースをインストールします。

clusterTasks および **pipelineTemplates** リソースのインストールを無効にするには、**.spec.addon** 仕様でパラメーターの値を **false** に設定します。さらに、**communityClusterTasks** パラメーターも無効にできます。

例

```

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'

```

3.8. TEKTON HUB 統合の無効化

Web コンソールの **Developer** パースペクティブで Tekton Hub の統合を無効にするには、**TektonConfig** カスタムリソース (CR) の **enable-devconsole-integration** パラメーターを **false** に設定します。

Tekton Hub 無効化の例

```

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

```

3.9. RBAC リソースの自動作成の無効化

Red Hat OpenShift Pipelines Operator のデフォルトインストールは、**^(openshift|kube)-*** 正規表現パターンに一致する namespace を除き、クラスター内のすべての namespace について複数のロールベースアクセス制御 (RBAC) リソースを作成します。これらの RBAC リソースの中で、**pipelines-scc-rolebinding** SCC (security context constraint) のロールバインディングリソースは、関連する **pipelines-scc** SCC に **RunAsAny** 権限があるため、セキュリティ上の問題となる可能性があります。

Red Hat OpenShift Pipelines Operator のインストール後にクラスター全体の RBAC リソースの自動作成を無効にするには、クラスター管理者は、クラスターレベルの **TektonConfig** カスタムリソース (CR) で **createRbacResource** パラメーターを **false** に設定します。

TektonConfig CR の例

```

apiVersion: operator.tekton.dev/v1alpha1
kind: TektonConfig
metadata:
  name: config
spec:
  params:
    - name: createRbacResource
      value: "false"
  ...

```

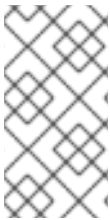


警告

クラスター管理者または適切な権限を持つユーザーとして、すべての namespace の RBAC リソースの自動作成を無効にすると、デフォルトの **ClusterTask** リソースは機能しません。**ClusterTask** リソースを機能させるには、それぞれの意図された namespace について RBAC リソースを手動で作成する必要があります。

3.10. タスク実行とパイプライン実行の自動プルーニング

古い **TaskRun** オブジェクトと **PipelineRun** オブジェクト、およびそれらの実行されたインスタンスは、アクティブな実行に使用できる物理リソースを占有します。これらのリソースを最適に利用するために、Red Hat OpenShift Pipelines は、さまざまな namespace 内の未使用のオブジェクトとそのインスタンスを自動的に削除するプルーナーコンポーネントを提供します。



注記

TektonConfig カスタムリソースを使用してインストール全体のプルーナーを設定し、namespace のアノテーションを使用して namespace の設定を変更できます。ただし、namespace で個々のタスク実行とパイプライン実行を選択的に自動プルーニングすることはできません。

3.10.1. プルーナーの設定

TektonConfig カスタムリソースを使用して、パイプラインの実行とタスクの実行に関連付けられたリソースの定期的なプルーニングを設定できます。

次の例は、デフォルト設定に対応します。

プルーナー設定の例

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

表3.2 プルーナー設定でサポートされているパラメーター

パラメーター	Description
schedule	プルーナープロセスを実行するための Cron スケジュール。デフォルトのスケジュールでは、プロセスは毎日 08:00 に実行されます。Cron スケジュール構文の詳細は、Kubernetes ドキュメントの Cron schedule syntax を参照してください。
resources	プルーナーが適用されるリソースのタイプ。使用可能なリソースの種類は、 taskrun と Pipelinerun です。
keep	保持するすべてのタイプの最新のリソースの数。
prune-per-resource	<p>false に設定した場合、keep パラメーターの値はタスク実行またはパイプライン実行の合計数を示します。たとえば、keep が 100 に設定されている場合、プルーナーは最新のタスク実行 100 件と最新のパイプライン実行 100 件を保持し、他のすべてのリソースを削除します。</p> <p>true に設定すると、keep パラメーターの値は、各パイプラインを参照するパイプライン実行と、各タスクを参照するタスク実行に対して個別に計算されます。たとえば、keep が 100 に設定されている場合、プルーナーは Pipeline1 の最新のパイプライン実行 100 件、Pipeline2 の最新のパイプライン実行 100 件、Task1 の最新のタスク実行 100 件などを保持し、他のすべてのリソースを削除します。</p>
keep-since	リソースを保持する最大時間 (分単位)。たとえば、6 日以上前に作成されたリソースを保持するには、 keep-since を 7200 に設定します。



注記

keep パラメーターと **keep-since** パラメーターは相互に排他的です。設定ではそのうちの 1 つだけを使用してください。

3.10.2. タスク実行とパイプライン実行を自動的にプルーニングするためのアノテーション

ネームスペース内のタスク実行とパイプライン実行の自動プルーニングの設定を変更するには、namespace にアノテーションを設定します。

次の namespace のアノテーションは、**TektonConfig** カスタムリソースの対応するキーと同じ意味を持ちます。

- **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`



注記

`Operator.tekton.dev/prune.resources` アノテーションは、コンマ区切りのリストを受け入れます。タスク実行とパイプライン実行の両方をプルーニングするには、このアノテーションを "`taskrun, pipelinerun`" に設定します。

次の追加の namespace アノテーションを使用できます。

- `Operator.tekton.dev/prune.skip: true` に設定すると、アノテーションが設定されている namespace はプルーニングされません。
- `operator.tekton.dev/prune.strategy`: このアノテーションの値を `keep` または `keep-since` のいずれかに設定します。

たとえば、次のアノテーションは、過去 5 日間に作成されたすべてのタスク実行とパイプライン実行を保持し、古いリソースを削除します。

自動プルーニングアノテーションの例

```
kind: Namespace
apiVersion: v1
# ...
spec:
  annotations:
    operator.tekton.dev/prune.resources: "taskrun, pipelinerun"
    operator.tekton.dev/prune.keep-since: 7200
# ...
```

3.11. 関連情報

- [Git の SSH 認証の設定](#)
- [バージョン管理されていないクラスタタスクおよびバージョン管理されたクラスタタスクの管理](#)
- [Administrator パースペクティブでのパイプラインテンプレートの作成](#)
- [リソースを回収するためのオブジェクトのプルーニング](#)