Red Hat OpenShift Pipelines 1.14

OpenShift Pipelines について

OpenShift Pipelines の概要

Last Updated: 2024-03-25

Red Hat OpenShift Pipelines 1.14 OpenShift Pipelines について

OpenShift Pipelines の概要

法律上の通知

概要

このドキュメントでは、OpenShift Pipelines 機能の概要を説明します。また、リリースノートおよびサポートを受ける方法の詳細も含まれています。

目次

第1章 RED HAT OPENSHIFT PIPELINES リリースノート

Red Hat OpenShift Pipelines は、以下を提供する Tekton プロジェクトをベースとするクラウドネイティブの CI/CD エクスペリエンスです。

- 標準の Kubernetes ネイティブパイプライン定義 (CRD)
- CI サーバー管理のオーバーヘッドのないサーバーレスのパイプライン。
- S2I、Buildah、JIB、Kaniko などの Kubernetes ツールを使用してイメージをビルドするための拡張性。
- Kubernetes ディストリビューションでの移植性。
- パイプラインと対話するための強力な CLI。
- OpenShift Container Platform Web コンソールの **Developer** パースペクティブと統合されたユーザーエクスペリエンス。

Red Hat OpenShift Pipelines の概要は、OpenShift Pipelines について を参照してください。

1.1. 互換性およびサポート表

現在、今回のリリースに含まれる機能には テクノロジープレビュー のものがあります。これらの実験的機能は、実稼働環境での使用を目的としていません。

以下の表では、機能は以下のステータスでマークされています。

TP	テクノロジープレビュー
GA	一般公開 (GA)

表1.1 互換性およびサポート表

Red Hat OpenShift Pipelines バージョン	コンポーネントのバージョン							OpenShift バージョン	サポートステータス
Operator	パイプライン	トリガー	CLI	チェーン	ハブ	コードとしてのパイプライン	結果		
1.14	0.56.x	0.26.x	0.35.x	0.20.x (GA)	1.16.x (TP)	0.24.x (GA)	0.9.x (TP)	4.12, 4.13, 4.14, 4.15	GA

Red Hat OpenShift Pipelines バージョン	コンポーネントのバージョン							OpenShift バージョン	サポートステータス
1.13	0.53.x	0.25.x	0.33.x	0.19.x (GA)	1.15.x (TP)	0.22.x (GA)	0.8.x (TP)	4.12, 4.13, 4.14, 4.15	GA
1.12	0.50.x	0.25.x	0.32.x	0.17.x (GA)	1.14.x (TP)	0.21.x (GA)	0.8.x (TP)	4.12、4.13、4.14	GA
1.11	0.47.x	0.24.x	0.31.x	0.16.x (GA)	1.13.x (TP)	0.19.x (GA)	0.6.x (TP)	4.12、4.13、4.14	GA
1.10	0.44.x	0.23.x	0.30.x	0.15.x (TP)	1.12.x (TP)	0.17.x (GA)	NA	4.10, 4.11, 4.12, 4.13	GA
1.9	0.41.x	0.22.x	0.28.x	0.13.x (TP)	1.11.x (TP)	0.15.x (GA)	NA	4.10, 4.11, 4.12, 4.13	GA
1.8	0.37.x	0.20.x	0.24.x	0.9.0 (TP)	1.8.x (TP)	0.10.x (TP)	NA	4.10、4.11、4.12	GA
1.7	0.33.x	0.19.x	0.23.x	0.8.0 (TP)	1.7.0 (TP)	0.5.x (TP)	NA	4.9、4.10、4.11	GA
1.6	0.28.x	0.16.x	0.21.x	NA	NA	NA	NA	4.9	GA
1.5	0.24.x	0.14.x (TP)	0.19.x	NA	NA	NA	NA	4.8	GA
1.4	0.22.x	0.12.x (TP)	0.17.x	NA	NA	NA	NA	4.7	GA

質問やフィードバックについては、製品チームに pipelines-interest@redhat.com 宛のメールを送信してください。

1.2. 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、Red Hat CTO である Chris Wright のメッセージ をご覧ください。

1.3. RED HAT OPENSHIFT PIPELINES 一般提供 1.14 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.14 が OpenShift Container Platform 4.12 以降のバージョンで利用できるようになりました。

1.3.1. 新機能

以下では、修正および安定性の面での改善点に加え、OpenShift Pipelines 1.14 の主な新機能について説明します。

1.3.1.1. Pipelines

- 今回の更新により、タスクまたはパイプラインのパラメーター、または前のタスクの結果を使用して、バインドのリソース名をワークスペースに指定できるようになりました (例: **name: $(params.name)-configmap)**。
- 今回の更新により、OpenShift Pipelines は、パイプライン内のビルドプロセスで Red Hat Enterprise Linux の既存のエンタイトルメントを使用できるようになります。組み込み **buildah** クラスタータスクでこれらのエンタイトルメントを使用できるようになりました。
- 今回の更新により、パイプライン実行またはタスク実行で **pipeline** サービスアカウントが使用される場合、パイプラインまたはタスクで CSI ボリュームタイプを使用できるようになります。
- 今回の更新により、**StepAction** カスタムリソース (CR) を使用して、再利用可能なスクリプトアクションを定義して任意数のタスクから呼び出すことができるようになりました。この機能を使用するには、**TektonConfig** CR の **Pipeline.options.configMaps.feature-flags.data.enable-step-actions** 仕様を **true** に設定する必要があります。
- 今回の更新により、オブジェクトパラメーターと配列結果がデフォルトで有効になりました。これらを使用するためにフラグを設定する必要はありません。
- この更新により、次の例に示すように、HTTP リゾルバーを使用して HTTP URL からパイプラインまたはタスクをフェッチできるようになりました。

  タスクの使用例

```
apiVersion: tekton.dev/v1
kind: TaskRun
metadata:
  name: remote-task-reference
spec:
  taskRef:
    resolver: http
    params:
```

```
    - name: url
      value: https://raw.githubusercontent.com/tektoncd-catalog/git-clone/main/task/git-
clone/git-clone.yaml
```

パイプラインの使用例

```
apiVersion: tekton.dev/v1beta1
kind: PipelineRun
metadata:
  name: http-demo
spec:
  pipelineRef:
    resolver: http
    params:
    - name: url
      value: https://raw.githubusercontent.com/tektoncd/catalog/main/pipeline/build-push-gke-
deploy/0.1/build-push-gke-deploy.yaml
```

- この更新により、次の例に示すように、enum 宣言を使用して、パイプラインまたはタスクのパラメーターに指定できる値を制限できるようになりました。この機能を使用するには、**TektonConfig** CR の **pipeline.options.configMaps.feature-flags.data.enable-param-enum** 仕様を **true** に設定する必要があります。

  使用例

```
apiVersion: tekton.dev/v1
kind: Pipeline
metadata:
  name: pipeline-param-enum
spec:
  params:
  - name: message
    enum: ["v1", "v2"]
    default: "v1"
# ...
```

- この更新により、認証されたソースコントロール管理 (SCM) API で Git リゾルバーを使用するときに、設定したデフォルトのトークン、SCM タイプ、およびサーバー URL をオーバーライドできるようになりました。以下の例を参照してください。

  使用例

```
apiVersion: tekton.dev/v1beta1
kind: TaskRun
metadata:
  name: git-api-demo-tr
spec:
  taskRef:
    resolver: git
    params:
    - name: org
      value: tektoncd
    - name: repo
      value: catalog
```

```
    - name: revision
      value: main
    - name: pathInRepo
      value: task/git-clone/0.6/git-clone.yaml
    # create the my-secret-token secret in the namespace where the
    # pipelinerun is created. The secret must  contain a GitHub personal access
    # token in the token key of the secret.
    - name: token
      value: my-secret-token
    - name: tokenKey
      value: token
    - name: scmType
      value: github
    - name: serverURL
      value: https://ghe.mycompany.com
```

- 今回の更新により、タスクの実行時に OpenShift Pipelines が作成する Pod 内のコンテナーと init-container のデフォルトのリソース要件を定義できるようになりました。これらの要件を設定するには、**TektonConfig** CR の **Pipeline.options.configMaps.config-defaults.default-container-resource-requirements** 仕様を使用します。すべてのコンテナーのデフォルト値を設定することも、特定のコンテナーのデフォルト値を名前または接頭辞 (**sidecar-*** など) で設定することもできます。

1.3.1.2. Operator

- この更新により、OpenShift Pipelines は Operator プロキシー Webhook の horizontal pod autoscaling をサポートします。Operator プロキシー Webhook を実行する Pod の CPU 使用率が 85% に達すると、オートスケーラーは Pod の別のレプリカを作成します。起動時に Operator プロキシー Webhook に複数のレプリカを使用する場合は、**TektonConfig** CR の **options.horizontalPodAutoscalers** 仕様でこの数を設定する必要があります。

- この更新により、OpenShift Pipelines のいくつかのコンポーネントの内部リーダーの選出が改善されました。Operator コントローラー、Operator Webhook、プロキシー Webhook、Pipelines as Code watcher、Pipelines as Code Webhook、および Tekton Chains コントローラーは、個別のリーダー選出 ConfigMap を使用するようになりました。リーダーの選出は、コンポーネントのどのレプリカがリクエストを処理するかに影響します。

- この更新前は、OpenShift Pipelines コントローラーのレプリカの数をスケールアップする場合、新しいレプリカの使用を有効にするために手動介入が必要でした。つまり、リーダー選挙でリースを削除する必要がありました。今回の更新により、OpenShift Pipelines コントローラーのレプリカの数をスケールアップすると、リーダーの選出に新しいレプリカが自動的に含まれるようになり、新しいレプリカで情報を処理できるようになります。

- この更新により、**TektonConfig** CR の **spec.pipeline** 仕様に次のフラグをオプションで設定できるようになりました。

  - **coschedule**
  - **enable-cel-in-whenexpression**
  - **enable-param-enum**
  - **enable-step-actions**
  - **enforce-nonfalsifiability**

- **keep-pod-on-cancel**
- **max-result-size**
- **metrics.count.enable-reason**
- **results-from**
- **set-security-context**
- **default-resolver-type**

1.3.1.3. トリガー

- 今回の更新により、Triggers インターセプターの CEL 式を指定するときに、**最初** と **最後** の関数を使用して JSON 配列内の値にアクセスできるようになりました。
- 今回の更新により、次の例のように、Triggers インターセプターの CEL 式を指定するときに、正規表現を利用して文字を指定した文字列に置き換えることが容易になる **translate** 関数を使用できるようになりました。

translate 関数の使用例

```
".translate("[^a-z0-9]+", "ABC")
```

入力文字列の例

```
This is $an Invalid5String
```

結果文字列の例

```
ABChisABCisABCanABCnvalid5ABCtring
```

1.3.1.4. Web コンソール

- 今回の更新により、OpenShift Pipelines の Web コンソールプラグインを有効にできるようになりました。プラグインを有効にすると、**Pipelines overview** ページとパイプラインのページでパイプラインおよびタスクの実行統計を表示できます。この情報を表示するには、Tekton Results をインストールする必要があります。

注記

OpenShift Pipelines の Web コンソールプラグインを使用するには、少なくとも以下の OpenShift Container Platform リリースを使用する必要があります。

- OpenShift Container Platform バージョン 4.12 の場合: 4.12.51
- OpenShift Container Platform バージョン 4.13 の場合: 4.13.34
- OpenShift Container Platform バージョン 4.14 の場合: 4.14.13
- OpenShift Container Platform バージョン 4.15 の場合: 任意のリリース

- この更新により、OpenShift Container Platform 4.15 を使用していてコンソールプラグインを有効にしている場合、過去のパイプライン実行とタスク実行に関するアーカイブ情報を表示できるようになりました。Tekton Results は、この情報を提供します。

- この更新により、Web コンソールの **Developer** または **Administrator** パースペクティブ両方からアクセスできる **PipelineRun** の詳細ページに、**Vulnerabilities** の行が導入されました。この新しい行では、特定された脆弱性が重大度 (重大、高、中、低) 別に分類されて視覚的に表示されます。この機能を有効にするには、タスクと関連パイプラインを指定された形式に更新します。さらに、有効にすると、パイプライン実行リストビューページの **Vulnerabilities** 列から、特定された脆弱性に関する情報にアクセスすることもできます。

- 今回の更新により、Web コンソールの **Developer** パースペクティブまたは **Administrator** パースペクティブの両方からアクセスできる **PipelineRun** の詳細ページで、ソフトウェア BOM (Software Bill of Materials) をダウンロードまたは表示するオプションが提供され、透過性および制御を強化されました。この機能を有効にするには、タスクと関連パイプラインを指定された形式に更新します。

1.3.1.5. CLI

- この更新により、Tekton Hub コンポーネントがインストールされている場合、**tkn version** コマンドでそのバージョンが表示されるようになりました。

- この更新により、**tkn customrun list** コマンドを使用してカスタム実行をリスト表示できるようになりました。

- 今回の更新により、**tkn task start** コマンドを使用するときに、**-i** または **--image** 引数で OCI イメージの URL を指定できるようになりました。このコマンドはイメージをプルし、このイメージから指定されたタスクを実行します。

- この更新により、**opc version** コマンドは、**opc** ユーティリティーの一部である Tekton Results CLI コンポーネントのバージョンを表示します。

1.3.1.6. Pipelines as Code

- 今回の更新により、Pipelines as Code を使用するときに、パイプラインの実行に **Pipelinesascode.tekton.dev/pipeline** アノテーションを指定して、Tekton Hub インスタンスからパイプラインをフェッチできるようになりました。このアノテーションの値は、Tekton Hub 上の単一のパイプラインを参照する必要があります。

- この更新により、異なる設定と異なるシークレットを使用して、追加の Pipelines as Code コントローラーをデプロイできるようになります。複数の Pipelines as Code コントローラーを使用して、複数の GitHub インスタンスと対話できます。

- この更新により、Pipelines as Code には GitLab プロバイダーと BitBucket プロバイダーのメトリクス公開が含まれます。メトリクスには、Pipelines as Code コントローラーの **/metrics** パスと、ウォッチャーサービス、ポート 9090 を使用してアクセスできます。

- この更新により、**Pipelinesascode.tekton.dev/on-cel-expression** で CEL 式を使用してパイプライン実行の実行条件を指定するときに、Git リポジトリー内のファイルの存在を確認できるようになりました。

  - 全ファイルの場合 **files.all.exists(x, x.matches('<path_or_regular_expression>'))**

  - このパイプラインの最後の実行以降に追加されたファイルの場合 **files.added.exists (x、x.matches ('<path_or_normal_expression>'))**

- このパイプラインの最後の実行以降に変更されたファイルの場合 **files.modified.exists (x, x.matches ('<path_or_regular_expression>'))**
- このパイプラインの最後の実行以降に削除されたファイルの場合 **files.deleted.exists (x, x.matches ('<path_or_regular_expression>'))**
- このパイプラインの最後の実行以降に名前が変更されたファイルの場合 **files.renamed.exists (x, x.matches ('<path_or_regular_expression>'))**

1.3.1.7. Tekton Chains

- この更新により、Tekton Chains は API バージョンの **v1** 値をサポートします。
- 今回の更新により、**TektonConfig** CR で **artifacts.pipelinerun.enable-deep-inspection** パラメーターを設定できるようになりました。このパラメーターが **true** の場合、Tekton Chains はパイプライン実行の子タスク実行の結果を記録します。このパラメーターが **false** の場合、Tekton Chains はパイプライン実行の結果を記録しますが、子タスクの実行は記録しません。
- 今回の更新により、**TektonConfig** CR で **builddefinition.buildtype** パラメーターを設定して、in-toto アテステーションのビルドタイプを設定できるようになりました。このパラメーターが **https://tekton.dev/chains/v2/slsa** の場合、Tekton Chains は SLSA v1.0 仕様に厳密に準拠して in-toto 証明書を記録します。このパラメーターが **https://tekton.dev/chains/v2/slsa-tekton** の場合、Tekton Chains は、各タスク実行およびパイプライン実行のラベルやアノテーションなどの追加情報を含む in-toto 証明を記録し、パイプライン実行の各タスクを **solvedDependency** の下に追加します。
- この更新前は、Tekton Chains が **gcs** ストレージを使用するように設定されていた場合、Tekton Chains はパイプライン実行情報を記録しませんでした。今回の更新により、Tekton Chains はこのストレージにパイプライン実行情報を記録します。
- この更新により、Tekton Chains のパフォーマンスメトリクスが利用できるようになりました。メトリックにアクセスするには、**tekton-chains-metrics** サービスを公開し、ポート 9090 のこのサービスの **/metrics** パスを使用します。これらのメトリクスは、OpenShift Container Platform Monitoring スタックでも利用できます。
- この更新により、Tekton Chains は、**v1** バージョン値を使用するパイプライン実行およびタスク実行を記録するときに、新しい **v2alpha3** レコード形式バージョンを使用します。
- この更新により、Tekton Chains は内部で **v1** バージョンのパイプライン実行およびタスク実行形式を使用するようになりました。

1.3.1.8. Tekton Results

- この更新により、Tekton Results がインストールされている場合、Tekton Results は Pipelines as Code を使用して開始されたパイプライン実行の概要と記録データを記録します。
- この更新により、Tekton Results はパイプラインまたはタスクについて最大 100 MB のログ情報を提供します。
- この更新により、認証されたユーザーはすべて、openshift-pipelines namespace の **tekton-results-api-service** ルートを表示し、REST API を使用して Tekton Results と対話できるようになります。
- 今回の更新により、Tekton Results API には、レコードのリストの概要と集計を取得するための新しいエンドポイントが含まれています。

- 今回の更新により、Tekton Results API の **GetLog** エンドポイントは、**text/plain** コンテンツタイプで raw バイトを返します。

- 今回の更新により、**TektonResult** CR の **options.configMaps.tekton-results-api-config.data.config.DB_SSLROOTCERT** 仕様でカスタム CA 証明書をオプションで指定できるようになりました。この場合、Tekton Results はデータベースサーバーへの SSL 接続を必要とし、接続にこの証明書を使用します。この設定を使用する場合は、Tekton Results を設定するときに、次の表に示すように、他のいくつかの設定パラメーターに対して代替仕様も使用する必要があります。通常のパラメーター仕様と代替パラメーターの仕様の両方が **TektonResult** CR に含まれています。

表1.2 Tekton Results の代替設定パラメーター

通常のパラメーター仕様	代替パラメーターの仕様
logs_api	**options.configMaps.tekton-results-api-config.data.config.LOGS_API**
log_level	**options.configMaps.tekton-results-api-config.data.config.LOG_LEVEL**
db_port	**options.configMaps.tekton-results-api-config.data.config.DB_PORT**
db_host	**options.configMaps.tekton-results-api-config.data.config.DB_HOST**
logs_path	**options.configMaps.tekton-results-api-config.data.config.LOGS_PATH**
logs_type	**options.configMaps.tekton-results-api-config.data.config.LOGS_TYPE**
logs_buffer_size	**options.configMaps.tekton-results-api-config.data.config.LOGS_BUFFER_SIZE**
auth_disable	**options.configMaps.tekton-results-api-config.data.config.AUTH_DISABLE**
db_enable_auto_migration	**options.configMaps.tekton-results-api-config.data.config.DB_ENABLE_AUTO_MIGRATION**
server_port	**options.configMaps.tekton-results-api-config.data.config.SERVER_PORT**
prometheus_port	**options.configMaps.tekton-results-api-config.data.config.PROMETHEUS_PORT**
gcs_bucket_name	**options.configMaps.tekton-results-api-config.data.config.GCS_BUCKET_NAME**

この表に記載されていない設定パラメーターについては、ドキュメントで説明されているように通常の仕様を使用してください。

重要

DB_SSLROOTCERT 設定を使用する必要がある場合にのみ、代替パラメーター仕様を使用してください。

1.3.2. 互換性を失わせる変更点

- この更新により、バンドルリゾルバーを使用する場合、**serviceAccount** パラメーターを指定できなくなりました。代わりに、**secret** パラメーターを指定して、レジストリーの認証情報を含むシークレットの名前を指定できます。また、**secret** パラメーターを使用するように、Bundles リゾルバーの **serviceAccount** パラメーターを使用するタスクまたはパイプラインを更新する必要があります。**TektonConfig** CR の **pipeline.bundles-resolver-config.default-service-account** 仕様はサポートされなくなりました。

1.3.3. 既知の問題

- **tkn Pipeline logs -f** コマンドは、パイプラインの進行中に **retries: X** パラメーターを使用してパイプラインに定義されたタスクのログを表示しません。

1.3.4. 修正された問題

- この更新の前は、GitHub Enterprise を使用する場合、受信 Webhook が機能しませんでした。この更新により、GitHub Enterprise で受信 Webhook を使用できるようになります。

- この更新より前は、タスク実行またはパイプライン実行でタイムアウトが無効になっている場合、OpenShift Pipelines はタスク実行またはパイプライン実行に対して一連の調整をすぐに実行し、コントローラーのパフォーマンスを低下させていました。今回の更新により、コントローラーは通常タイムアウトを無効にしてタスク実行とパイプライン実行を調整します。

- 今回の更新以前は、カスタムの namespace を使用して Tekton Hub をインストールした場合、そのインストールにより **openshift-pipelines** namespace が削除され、OpenShift Pipelines インストールが削除されました。今回の更新により、カスタムの namespace を使用して Tekton Hub をインストールできるようになり、OpenShift Pipelines のインストールは影響を受けません。

- この更新より前は、GitLab で Pipelines as Code を使用する場合、ユーザーが **/test** などのマージリクエスト内のコメントを使用してパイプラインの実行をトリガーした場合、Pipelines as Code はマージリクエストでのパイプライン実行のステータスを報告しませんでした。今回の更新により、Pipelines as Code はマージリクエスト上のパイプライン実行のステータスを正しく報告します。

- この更新より前は、次の例に示すように、Tekton Results でサブグループを含む CEL フィルターを使用すると、サブグループが正しく機能しませんでした。今回の更新により、サブグループが正しく機能するようになりました。

  サブグループを含む CEL フィルターの例

  ```
  "data_type==TASK_RUN &&
  (data.spec.pipelineSpec.tasks[0].name=='hello'||data.metadata.name=='hello')"
  ```

- この更新より前は、パイプライン実行がキャンセルされた場合、Tekton Results はこのパイプライン実行のログを記録しませんでした。この更新により、Tekton Results はキャンセルされたパイプライン実行のログを記録します。

1.3.5. Red Hat OpenShift Pipelines 一般提供 1.14.1 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.14.1 が OpenShift Container Platform 4.12 以降のバージョンで利用できるようになりました。

1.3.5.1. 修正された問題

- この更新より前は、異なる GitHub アプリで設定された複数の Pipelines as Code コントローラーを使用すると、Pipelines as Code ウォッチャーコンポーネントが **nilerror** メッセージでクラッシュしました。この更新により、Pipelines as Code は、異なる GitHub アプリで設定された複数のコントローラーで正常に機能します。

1.3.6. Red Hat OpenShift Pipelines 一般提供 1.14.2 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.14.2 が OpenShift Container Platform 4.12 以降のバージョンで利用できるようになりました。

1.3.6.1. 修正された問題

- この更新より前は、Pipelines as Code を使用してパイプライン実行を開始した場合、Tekton Results にはこのパイプライン実行に関する情報が保存されませんでした。この問題が原因で、Web コンソールプラグインでは、実行統計表示にパイプライン実行が含まれませんでした。今回の更新により、Tekton Results は Pipelines as Code のパイプライン実行に関する情報を保存し、これらのパイプライン実行が実行統計表示に含まれるようになりました。

- この更新より前は、Pipelines as Code を使用して多くのパイプライン実行を同時に開始し、これらのパイプライン実行に **max-keep-run** アノテーションが含まれていた場合、Pipelines as Code ウォッチャーコンポーネントは保留中のパイプライン実行の一部を処理できず、それらのパイプライン実行は、Pending の状態でそのまま残りました。今回の更新により、Pipelines as Code パイプライン実行が正しく処理されます。

1.4. RED HAT OPENSHIFT PIPELINES 一般提供 1.13 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.13 が OpenShift Container Platform 4.12 以降のバージョンで利用できるようになりました。

1.4.1. 新機能

以下では、修正および安定性の面での改善点に加え、OpenShift Pipelines 1.13 の主な新機能について説明します。

注記

Red Hat OpenShift Pipelines Operator 1.13 にアップグレードする前に、少なくとも OpenShift Container Platform 4.12.19 または 4.13.1 バージョンがクラスターにインストールされていることを確認してください。

1.4.1.1. Pipelines

- この更新前は、Source-to-Image (S2I) クラスタータスクでは、テクノロジープレビューにあっ

たベース S2I コンテナーイメージが使用されていました。この更新により、S2I クラスタータスクは、リリースされ完全にサポートされるベース S2I コンテナーイメージを使用するようになりました。

- この更新により、オプションで、タスク実行がキャンセルされたときに、OpenShift Pipelines がタスク実行の Pod を停止しますが、Pod は削除しないように設定を有効にすることができます。この設定を有効にするには、**TektonConfig** カスタムリソース (CR) で、**pipeline.options.configMaps.feature-flags.data.keep-pod-on-cancel** 仕様を **true** に設定し、**pipeline.enable-api-fields** 仕様を **alpha** に設定します。

- この更新前は、タスクレベルでコンピューティングリソース制限を設定するには、アルファ機能を有効にする必要がありました。この更新により、**TaskRun** CR の **computeResources** 仕様を使用してタスクのリソース制限を設定できるようになりました。

- この更新により、タスクを指定し、**displayName** パラメーターを使用するときに、表示名にパラメーター、結果、またはコンテキスト変数の値を含むパラメーターを使用できるようになりました (例: **$(params.application)**、**$(tasks.scan.results.report)**、**$(context.pipeline.name)**)。

- この更新により、ハブリゾルバーを使用してリモートパイプラインまたはタスクを指定するときに、**version** パラメーターで **>=0.2.0,< 1.0.0** などの不等式制約を使用できるようになりました。

- 今回の更新により、タスクを指定するときに **when** 式で Common Expression Language (CEL) 式を使用できるようになりました。この機能を使用するには、**TektonConfig** CR で **pipeline.options.configMaps.feature-flags.data.enable-cel-in-whenexpression** 仕様を **true** に設定する必要があります。

- この更新により、**PipelineRun** CR 仕様でパイプラインを指定するときに、インラインタスクによって生成された結果を後続のインラインタスクで参照できるようになりました。

  使用例

```
apiVersion: tekton.dev/v1
kind: Task
metadata:
  name: uid-task
spec:
  results:
    - name: uid
  steps:
    - name: uid
      image: alpine
      command: ["/bin/sh", "-c"]
      args:
        - echo "1001" | tee $(results.uid.path)
---
apiVersion: tekton.dev/v1
kind: PipelineRun
metadata:
  name: uid-pipeline-run
spec:
  pipelineSpec:
    tasks:
    - name: add-uid
      taskRef:
```

```
      name: uid-task
  - name: show-uid
    taskSpec:
      steps:
        - name: show-uid
          image: alpine
          command: ["/bin/sh", "-c"]
          args:
            - echo $(tasks.add-uid.results.uid)
```

- この更新により、クラスターリゾルバーを設定するときに、**blocked-namespaces** パラメーターの値を * に設定できるようになりました。この設定では、**allowed-namespaces** パラメーターにリストされている namespace のみが許可され、他の namespace はすべてブロックされます。

1.4.1.2. Operator

- この更新により、**disable-affinity-assistant** 機能フラグは非推奨になり、将来のリリースでは削除される可能性があります。代わりに、**TektonConfig** CR で、**pipeline.options.configMaps.feature-flags.data.coschedule** 仕様を次のいずれかの値に設定できます。
  - **workspaces**: ワークスペースが永続ボリューム要求を割り当てる場合、OpenShift Pipelines は、同じワークスペースを共有するすべてのタスクの実行を同じノードにスケジュールします。これはデフォルト設定です。
  - **pipelineruns**: OpenShift Pipelines は、同じノードへのパイプライン実行内のすべてのタスクの実行をスケジュールします。
  - **isolate-pipelinerun**: OpenShift Pipelines は、同じノードへのパイプライン実行内のすべてのタスク実行をスケジュールし、ノード上で同時に実行できるパイプライン実行は 1 つだけです。すべてのノードが他のパイプライン実行に使用されている場合は、この設定によりパイプラインの実行が遅れる可能性があります。
  - **disabled**: OpenShift Pipelines は、ノードへのタスク実行の割り当てに関する特定のポリシーを適用しません。

1.4.1.3. トリガー

- この更新が行われる前は、コアインターセプターは起動時に常に TLS シークレットを作成していました。この更新により、TLS シークレットがクラスター上に存在しない場合、または既存のシークレット内の証明書の有効期限が切れた場合、コアインターセプターは TLS シークレットを作成します。

1.4.1.4. CLI

- 今回の更新により、**tkn bundle push** コマンドを使用すると、作成時刻が **1970-01-01T00:00:00Z** (Unix エポック時間) に設定されたバンドルが作成されます。この変更により、同じソースから作成されたバンドルイメージが常に同一になることが保証されます。**--ctime** パラメーターを使用して、RFC3339 形式で作成時間を設定できます。**SOURCE_DATE_EPOCH** 環境変数を使用して作成時刻を設定することもできます。

1.4.1.5. Pipelines as Code

- この更新により、Pipelines as Code で高度なイベント一致 (pipelinesascode.tekton.dev/on-cel-

expression) に CEL 式を使用するときに、**header** フィールドと **body** フィールドを使用して、Git リポジトリープロバイダー本体が渡す完全ペイロードにアクセスできます。この機能を使用すると、Git リポジトリーが送信する情報によってイベントをフィルターできます。

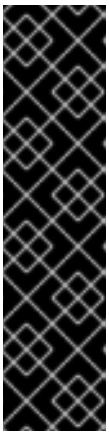
重要

イベント一致のための CEL 式でのペイロードのヘッダーと本文の使用は、テクノロジープレビューのみの機能です。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- この更新により、Pipelines as Code パイプラインの実行がプッシュイベントによってトリガーされたときに、対応するコミットコメントに対して **/test**、**/test branch:<branchname>**、**/retest**、**/retest branch:<branchname>**、**/cancel**、および **/cancel branch:<branchname>** コマンドを使用して、パイプラインの実行を再実行またはキャンセルします。

- この更新により、Pipelines as Code として使用するときに、リモートパイプラインでリモートタスクを使用できるようになります。したがって、複数のリポジトリー間で完全なリモートパイプラインを再利用できます。同じ名前のタスクを追加することで、リモートパイプラインからのタスクをオーバーライドできます。



重要

リモートパイプラインでのリモートタスクの使用とタスクのオーバーライドは、テクノロジープレビューのみの機能です。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- この更新により、Pipelines as Code として使用するときに、Git リポジトリープロバイダーでの CI パイプラインの実行に関する改善された情報を表示できるようになります。この情報には、namespace と関連するパイプラインの実行が含まれます。

1.4.2. 互換性を失わせる変更点

- この更新前は、Pipelines as Code でポリシーグループを使用する場合、ポリシーグループの一部ではなく、明示的に許可されていないが (組織の所有権などを通じて) CI の実行が許可されているユーザーが、プルリクエストのようなイベントを作成、または **ok_to_test** などのコマンドを実行して、パイプライン実行を行える場合がありました。この更新により、ポリシーグループが設定されている場合は、必要なポリシーグループに追加されたユーザーのみがパイプライン実行を行えるようになり、所有者組織の一部であるがポリシーグループに設定されていないユーザーはパイプライン実行を行えなくなります。

1.4.3. 既知の問題

- タスクの実行がキャンセルされたときに Pod を保持できるようにするには、**TektonConfig** CR で **pipeline.options.configMaps.feature-flags.data.keep-pod-on-cancel** 仕様を **true** に設定するとともに、**TektonConfig** CR の **pipeline.enable-api-fields** 仕様を **alpha** に設定する必要があります。

- タスク実行がキャンセルされたとき、デフォルトのタイムアウトのため、またはパイプライン仕様でタイムプットを設定したためにタスク実行がキャンセルされたときに Pod の保持を有効にすると、OpenShift Pipelines は Pod を削除します。

1.4.4. 修正された問題

- この更新前は、パイプラインの実行で Git 認証に使用されるシークレットが、クリーンアップ中にクラスターから削除される可能性がありました。この更新により、シークレットは、それを使用するすべてのパイプライン実行が削除された場合にのみ削除されます。

- この更新前は、複数のシークレットが同じ接頭辞を共有し、git インターフェイスを使用してログに記録された場合、隠蔽プロセスが短いシークレットから開始され、長いシークレットの一部がログに表示されることがありました。今回の更新により、ログ内のシークレットを隠蔽する場合は、最も長いシークレットからプロセスが開始されるようになり、ログにはシークレットの一部が表示されなくなります。

- この更新前は、パイプラインの **results** 仕様を指定した場合、パイプラインの実行がタイプの不一致エラーで誤って失敗する可能性がありました。この更新により、パイプラインの **results** 仕様を指定すると、パイプラインによって提供される結果が正しく処理されるようになりました。

- この更新より前は、Tekton Chains が KMS を使用して Hashicorp Vault として設定されていて、Vault への接続中に根本的なエラーが発生すると Pod がクラッシュし始めました。この問題は修正され、エラーが Tekton Chains コントローラーのログに記録されるようになりました。

- この更新前は、Tekton Chains を使用するときに **storage.oci.repository** パラメーターを設定すると、Tekton Chains コントローラーログにエラーが報告されていました。この更新により、**storage.oci.repository** パラメーターが正しく処理されるようになりました。

- この更新より前は、Tekton Chains が Hashicorp Vault KMS で設定されており、Vault への接続に問題があると、Tekton Chains コントローラー Pod がクラッシュする可能性がありました。この更新により、エラーが処理され、Tekton Chains コントローラーログに記録されます。

1.4.5. Red Hat OpenShift Pipelines 一般提供 1.13.1 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.13.1 が OpenShift Container Platform 4.12 以降のバージョンで利用できるようになりました。

1.4.5.1. 修正された問題

- この更新が行われる前は、タスクの実行が失敗し、**cannot stop the sidecar** というエラーメッセージが表示されることがありました。今回の更新により、この障害の原因となったコントローラー間の競合状態が修正されました。

- この更新より前は、タスクの実行がキャンセルされたときに Pod を保持できるようにするには、**TektonConfig** CR で **pipeline.options.configMaps.feature-flags.data.keep-pod-on-cancel** 仕様を **true** に設定するとともに、**TektonConfig** CR の **pipeline.enable-api-fields** 仕様を **alpha** に設定する必要がありました。今回の更新により、**TektonConfig** CR で

Pipeline.options.configMaps.feature-flags.data.keep-pod-on-cancel 仕様を **true** に設定すると、タスクの実行がキャンセルされたときに Pod を保持できるようになり、追加の設定は必要なくなりました。

- 今回の更新前は、タスクのサイドカーを定義した場合、OpenShift Pipelines では **Task** および **TaskRun** カスタムリソース (CR) の作成時に定義内のコンテナーイメージが検証されませんでした。ランタイム時に、サイドカーに無効なコンテナーイメージが含まれていたことが原因で、タスクの実行が失敗しました。今回の更新により、OpenShift Pipelines は **Task** および **TaskRun** CR の作成時にサイドカー定義のコンテナーイメージを検証するようになりました。

- 今回の更新の前は、タスクがパラメーターを評価しているときに OpenShift Pipelines コントローラーがクラッシュすることがありました。今回の更新により、コントローラーがクラッシュしなくなりました。

- 今回の更新より前は、パイプライン実行の最後のタスクが失敗するかスキップされた場合に、OpenShift Pipelines はパイプライン実行の検証エラーを報告することがありました。今回の更新により、OpenShift Pipelines はパイプライン実行のステータスを正しく報告します。

1.5. RED HAT OPENSHIFT PIPELINES 一般提供 1.12 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.12 が OpenShift Container Platform 4.12 以降のバージョンで利用できるようになりました。

1.5.1. 新機能

以下では、修正および安定性の面での改善点に加え、OpenShift Pipelines 1.12 の主な新機能について説明します。

注記

Red Hat OpenShift Pipelines Operator 1.12 にアップグレードする前に、少なくとも OpenShift Container Platform 4.12.19 または 4.13.1 バージョンがクラスターにインストールされていることを確認してください。

1.5.1.1. Pipelines

- 今回の更新により、Web コンソールに Pipelines 実行とタスク実行の新しいゲージメトリクスが含まれます。このメトリクスは、namespace で定義されたリソースクォータポリシー、または基礎となるノードのリソース制約が理由で、基礎となる Pod が OpenShift Container Platform によってスロットルされているかどうかを示します。

- 今回の更新により、新しい **set-security-context** 機能フラグがデフォルトで **true** に設定され、制限された Pod セキュリティーアドミッションポリシーが割り当てられた namespace でタスク実行とパイプライン実行を可能にします。

- 今回の更新により、**enable-api-fields** フラグはデフォルトで **beta** に設定されます。コード内で **beta** としてマークされているすべての機能は、追加で変更を加えることなく使用できます。

- 今回の更新により、**results.tekton.dev/*** and **chains.tekton.dev/*** の予約アノテーションは、パイプライン実行から作成されるタスク実行に渡されなくなります。

- 今回の更新の前は、CSI ボリュームと投影されたボリュームはデフォルトでは有効になっていませんでした。今回の更新により、設定フィールドを変更せずに、Pipelines で CSI ボリュームと投影ボリュームを使用できるようになります。

- 今回の更新により、分離ワークスペース機能がデフォルトで有効になります。この機能を使用すると、タスク実行全体でワークスペースを共有せずに、指定したステップおよびサイドカーとワークスペースを共有できます。

1.5.1.2. Operator

- 今回の更新により、OpenShift Pipelines がパイプライン実行およびタスク実行用に作成する Pod のデフォルトの security context constraint (SCC) を設定できるようになりました。異なる namespace に対して個別に SCC を設定したり、任意の namespace に対して設定できる最大 (最も制限の少ない) SCC を設定したりすることもできます。

- 今回の更新により、TektonConfig 仕様の各コンポーネントの下に新しい **options:** の見出しが追加されました。この見出しの下のパラメーターを使用して、さまざまなコンポーネントの設定を制御できます。特に、**platforms.openshift.pipelinesAsCode.options.configMaps.pac-config-logging.data** 仕様のパラメーターを使用して、Pipelines as Code のコンポーネントのロギングレベルを設定できます。

- 今回の更新により、新しい **spec.pipeline.performance.replicas** パラメーターを使用して、OpenShift Pipelines コントローラー Pod 用に作成されるレプリカの数を設定できるようになりました。これまでに、デプロイメント内のレプリカ数を手動で設定している場合は、この設定を使用してレプリカ数を制御する必要があります。

- 今回の更新により、Operator を使用すると、保存された API バージョンが、OpenShift Pipelines のデプロイメント全体で同じバージョンに確保できるようになります。OpenShift Pipelines 1.12 に保存されている API バージョンは **v1** です。

- 今回の更新により、シークレットを使用して S3 バケットストレージを設定し、Tekton Results ログ情報を保存できるようになりました。S3 バケットストレージを設定する場合、**TektonResult** カスタムリソース (CR) の新しい **Secret_name** 仕様を使用して、S3 ストレージ認証情報を含むシークレットを提供する必要があります。

1.5.1.3. Tekton Results

重要

Tekton Results はテクノロジープレビューのみの機能です。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- 今回の更新により、外部 PostgreSQL サーバーにデータを保存するように Tekton Results を設定できるようになります。

- 今回の更新により、Google Cloud Storage (GCS) を使用して Tekton Results ログ情報を保存できるようになります。GCS ストレージ認証情報を使用してシークレットを指定し、**TektonResult** 仕様のプロパティーでシークレット名、秘密鍵、バケット名を指定できます。認証に Workload Identity Federation を使用することもできます。

- 今回の更新により、OpenShift Pipelines で認証されたサービスアカウントはすべて **TektonResult** CR にアクセスできるようになります。

- 今回の更新により、Tekton Results には、admin、edit、および view のロールが割り当てられたサービスアカウントのクラスターロールの集約が含まれます。これらのサービスアカウントが Tekton Results API を使用して結果とレコードにアクセスするために、クラスターロールバインドは必要なくなりました。

- この更新により、**TektonConfig** CR の **prune-per-resource** ブール型フィールドを設定することで、各 **PipelineRun** または **TaskRun** リソースのプルーニングを設定できるようになりました。また、**operator.tekton.dev/prune.prune-per-resource=true** アノテーションをネームスペースに追加することで、ネームスペース内の各 **PipelineRun** または **TaskRun** リソースのプルーニングを設定することもできます。

- 今回の更新により、OpenShift Container Platform クラスター全体のプロキシーに変更があった場合、Operator Lifecycle Manager (OLM) は Red Hat OpenShift Pipelines Operator を再作成します。

- この更新により、**TektonConfig** CR で **config.pruner.disabled** フィールドの値を **true** に設定することで、プルーナー機能を無効にすることができます。

1.5.1.4. トリガー

- 今回の更新により、**Trigger** CR で readiness プローブと liveness プローブを設定できるようになりました。プローブの失敗しきい値を設定することもできます。デフォルト値は 3 です。

- 今回の更新により、OpenShift Pipelines トリガーで、Cloud Events リクエストに対するレスポンスを作成するときに **Type** と **Subject** の値が追加されます。

1.5.1.5. CLI

- 今回の更新により、**tkn Pipeline logs** コマンドは、リゾルバーを使用して参照されるパイプラインまたはタスクのログを表示します。

- 今回の更新により、**tkn Bundle Push** コマンドを入力するときに **--annotate** フラグを使用して追加のアノテーションを指定できるようになりました。

1.5.1.6. Pipelines as Code

- 今回の更新により、Pipelines as Code パイプラインの実行に、複数の Artifact Hub または Tekton Hub インスタンス、および同じハブインスタンス内の異なるカタログからフェッチされたリモートタスクを含めることができるようになりました。

- 今回の更新により、**TektonConfig** CR の **platforms.openshift.pipelinesAsCode.options.configMaps.pac-config-logging.data** 仕様のパラメーターを使用して、Pipelines as Code コンポーネントのロギングレベルを設定できるようになりました。

- 今回の更新により、特定のアクションをチームのメンバーにのみ許可し、他のユーザーがアクションを要求した場合はそのアクションを拒否するポリシーを設定できるようになります。現在、**pull_request** アクションと **ok_to_test** アクションは、そのようなポリシーの設定をサポートしています。

- 今回の更新により、受信 Webhook で任意のパラメーターを JSON ペイロードとして渡すことができるようになりました。OpenShift Pipelines は、これらのパラメーターをパイプライン実行に渡します。セキュリティー層を追加するには、**Repository** CR で許可されたパラメーターを明示的に定義する必要があります。

- 今回の更新により、Pipelines as Code 内の多数のリモートアノテーションと大規模なパイプライン実行のマッチングが最適化されます。Pipelines as Code は、一致したパイプライン実行のリモートタスクのみをフェッチします。

- 今回の更新により、パイプライン実行テンプレートで **source_url** 変数を使用して、プルリクエストやプッシュリクエストなどのイベントをトリガーするフォークしたリポジトリーに関する情報を取得します。

  使用例

```
apiVersion: tekton.dev/v1beta1
kind: PipelineRun
# ...
spec:
  params:
    - name: source_url
      value: "{{ source_url }}"
  pipelineSpec:
    params:
      - name: source_url
# ...
```

- 今回の更新により、承認されたユーザーが、承認されていないユーザーからのプルリクエストに対してパイプライン実行をトリガーするために **ok-to-test** コメントを提供し、作成者がブランチにさらに変更を加えた場合、Pipelines as Code によってパイプラインがトリガーされます。承認されたユーザーが新しい **ok-to-test** コメントを提供するまでパイプラインのトリガーを無効にするには、**TektonConfig** CR の **PipelinesAsCode.settings.remember-ok-to-test** 仕様を false に設定します。

- 今回の更新により、GitHub のステータス確認ページで、すべてのタスクのステータスを示すテーブルに各タスクの表示名が含まれるようになりました。

- 今回の更新により、GitLab で実行されるパイプラインで **tags push** イベントを設定できるようになりました。

- 今回の更新により、Pipelines の **target_url** と **source_url** フィールドを Code Common Expression Language (CEL) 式フィルタリングアノテーションとして使用して、特定のターゲットまたはソースのリクエストをフィルタリングできるようになりました。

- 今回の更新により、トークンを使用してリモート GitHub URL を取得するように設定する場合に、スラッシュを含むブランチ名を追加できるようになりました。この **https://github.com/organization/repository/blob/feature%2Fmainbranch/path/file** の URL 例のように、パイプラインをコードとして適切に解析するには、ブランチ名内のスラッシュを **%2F** としてエンコードする必要があります。この URL の例では、ブランチ名は **feature/mainbranch** で、フェッチするファイルの名前は **/path/file** です。

- 今回の更新により、**tkn pacsolve** コマンドで **--v1beta1** フラグを使用できるようになりました。パイプライン実行が **v1beta1** API バージョンスキーマで生成される場合は、このフラグを使用します。

1.5.2. 互換性を失わせる変更点

- 今回の更新により、**openshift-operators** namespace を OpenShift Pipelines をインストールするためのターゲット namespace として使用できなくなります。**openshift-operators** 名前空間をターゲット namespace として使用した場合は、Red Hat OpenShift Pipelines Operator バー

ジョン 1.12 にアップグレードする前にターゲット namespace を変更してください。そうしないと、アップグレード後に、**targetNamespace** 設定を除く **TektonConfig** CR の設定を変更できなくなります。

- 今回の更新により、新しい **spec.pipeline.performance.replicas** パラメーターは、パイプライン実行またはタスク実行の Pod ごとに作成されるレプリカの数を制御します。これまでにデプロイメントのレプリカ数を手動で設定した場合は、OpenShift Pipelines バージョン 1.12 にアップグレードした後に、このパラメーターを使用してレプリカ数を制御する必要があります。
- 今回の更新により、次のパラメーターは **TektonResult** CR でサポートされなくなりました。
  - **db_user**
  - **db_password**
  - **s3_bucket_name**
  - **s3_endpoint**
  - **s3_hostname_immutable**
  - **s3_region**
  - **s3_access_key_id**
  - **s3_secret_access_key**
  - **s3_multi_part_size**
    これらのパラメーターはシークレットを使用して指定する必要があります。OpenShift Pipelines バージョン 1.12 にアップグレードした後、**TektonResult** CR を削除して再作成し、これらのパラメーターを提供する必要があります。
- 今回の更新により、**tkn pac bootstrap** コマンドは **--github-hostname** フラグをサポートします。**--github-api-url** フラグは非推奨になりました。

1.5.3. 既知の問題

- namespace に制限範囲が設定されているが、Pod の一時ストレージが制限範囲に設定されていない場合、Pod はエラーの段階になり、**Pod ephemeral local storage usage exceeds the total limit of containers 0** というメッセージが表示されることがあります。
- **TektonResult** CR の設定を変更する場合は、既存の **TektonResult** CR を削除して、新しい CR を作成する必要があります。既存の **TektonResult** CR を変更した場合、その変更は Tekton Results の既存のデプロイメントには適用されません。たとえば、接続を内部データベースサーバーから外部データベースサーバーに変更した場合、またはその逆に変更した場合、API は古いデータベースに接続されたままになります。

1.5.4. 修正された問題

- 今回の更新が行われる前は、Pipelines as Code はブランチベース名のみに基づいてパイプライン実行を行い、同じベース名にもかかわらず、異なるブランチ名でパイプライン実行を誤ってトリガーする可能性がありました。今回の更新により、Pipelines as Code はパイプライン実行のベース名と、正確なブランチ名の両方をチェックします。

- 今回の更新以前は、受信 Webhook イベントによって、他のイベント用に設定された複数のパイプライン実行がトリガーされる可能性がありました。今回の更新により、受信 Webhook イベントは、Webhook イベント用に設定されたパイプライン実行のみをトリガーします。

- 今回の更新により、パイプライン実行の **ownerRef** が削除された場合に備えて、パイプライン実行をクリーンアップするときに **pac-gitauth** シークレットが明示的に削除されるようになりました。

- 今回の更新より前は、パイプライン実行内のタスクが理由を含むメッセージを表示して失敗すると、パイプライン実行全体が **PipelineValidationFailed** という理由で失敗していました。今回の更新により、パイプラインの実行が失敗し、失敗したタスクと同じ理由のメッセージが表示されます。

- 今回の更新前は、**disable-ha** フラグの値が Pipelines コントローラーに正しく渡されず、高可用性 (HA) 機能が有効になることはありませんでした。今回の更新により、**TektonConfig** CR の **disable-ha** フラグの値を **false** に設定することで HA 機能を有効化できます。

- 今回の更新が行われる前は、config map データに記載されているイメージをコピーしようとすると、**skopeo-copy** クラスタータスクに失敗していました。今回の更新により、**skopeo-copy** クラスタータスクが適切に完了します。

- 今回の更新により、**tkn pac generated – language=java** コマンドによって自動的に生成されたパイプライン実行に、正しいアノテーションとパラメーター名が付けられるようになりました。

- 今回の更新が行われる前は、管理権限を持つユーザーのみが **tkn pac create repository** コマンドを正常に実行できました。今回の更新により、許可されたユーザーはすべて **tkn pac create repository** コマンドを実行できるようになります。

- 今回の更新より前は、特定のパイプラインを指定する **/test <run-name>** および **/retest <run-name>** ユーザーコメントによって、期待どおりにパイプラインの実行がトリガーされませんでした。今回の更新では、これらのコメントによりパイプライン実行が正常にトリガーされます。

- 今回の更新より前は、**Name** フィールドではなく、**generateName** フィールドが含まれる **.tekton** フォルダー内で複数のパイプライン実行があった場合、パイプライン実行は失敗していました。今回の更新でこの問題が修正されています。

- 今回の更新以前は、GitLab を使用する場合の Pipelines as Code では、ラベルの追加やステータスの設定など、マージリクエスト内の任意のイベントによってパイプラインの実行がトリガーされていました。今回の更新により、パイプラインの実行は、オープン、再オープン、またはプッシュイベントがある場合にのみトリガーされます。チェックのステータスを含むコメントがマージリクエストに投稿されるようになりました。

- 今回の更新より前は、パイプライン実行が承認を待機している間、GitHub および Gitea プルリクエストのチェックセクションにチェックのステータスが **skipped** と表示されることがありました。今回の更新により、正しい **pending approval** のステータスが表示されるようになりました。

- 今回の更新以前は、バンドルリゾルバーはパイプラインを取得しようとするとタイプを **Task** に設定することがあり、取得エラーが発生することがありました。今回の更新により、リゾルバーは正しいタイプを使用してパイプラインを取得するようになりました。

- 今回の更新により、Tekton Results をクエリーする際の共通表現言語 (CEL) の NOT 演算子の処理で発生していたエラーが修正されました。

- 今回の更新により、レコードの **LIST** 操作が要求され、指定された結果が **-** であった場合に Tekton Results API で生成される **404** エラー応答が修正されました。

- 今回の更新前は、**EventListener** オブジェクトで、**status.address.url** フィールドは常にデフォルトのポートに設定されていました。今回の更新により、**status.address.url** フィールドは、**spec.resources.kubernetesresource.serviceport** パラメーターで指定されたポートと一致するように設定されます。

- 今回の更新前は、GitHub API がページネーションされたレスポンスを提供した場合に、Pipelines as Code はレスポンスの最初のページのみを使用していました。今回の更新により、ページネーションされたレスポンスがすべてが完全に処理されます。

- 今回の更新が行われる前は、KMS Hashicorp Vault のホストアドレスが正しく設定されていない場合、または Tekton Chains が KMS Hashicorp Vault に接続できない場合、Tekton Chains コントローラーがクラッシュしていました。今回の更新により、Tekton Chains は接続エラーをログに記録し、クラッシュしなくなります。

1.5.5. Red Hat OpenShift Pipelines 一般提供 1.12.1 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.12.1 が OpenShift Container Platform 4.12 以降のバージョンで利用できるようになりました。

1.5.5.1. 修正された問題

- この更新前は、カスタムコンソールに出力するようにカスタムコンソールドライバーを使用して Pipelines as Code を設定した場合、特定のケースで Pipelines as Code コントローラーがクラッシュしていました。変更をプルリクエストにプッシュした後、このプルリクエストの CI ステータスチェックが **waiting for status to be reported** になり、関連するパイプラインの実行が完了しない可能性があります。この更新により、Pipelines as Code コントローラーは正常に動作するようになりました。変更をプルリクエストにプッシュすると、関連するパイプラインの実行が正常に完了し、プルリクエストの CI ステータスチェックが更新されます。

- Pipelines as Code を使用する場合、特定のユーザーを含まないアクセスポリシーを **Repository** カスタムリソース (CR) に作成し、そのユーザーを Git リポジトリーの **OWNER** ファイルに追加すると、そのユーザーは Pipelines as Code CI プロセスに対する権利を持ちませんでした。たとえば、ユーザーが Git リポジトリーにプルリクエストを作成したとします。CI プロセスは、このプルリクエストに対して自動的には実行されません。この更新により、**Repository** CR のアクセスポリシーには含まれていないが、**OWNER** ファイルには含まれているユーザーが、リポジトリーの CI プロセスを実行できるようになりました。

- この更新により、HTTP/2.0 プロトコルは Webhook でサポートされなくなりました。Red Hat OpenShift Pipelines へのすべての Webhook 呼び出しでは、HTTP/1.1 プロトコルを使用する必要があります。

1.5.6. Red Hat OpenShift Pipelines 一般提供 1.12.2 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.12.2 が OpenShift Container Platform 4.12 以降のバージョンで利用できるようになりました。

1.5.6.1. 修正された問題

- この更新前は、**max-keep-runs** パラメーターを超えると、最新のパイプライン実行に対して生成された Git シークレットが削除されました。この更新により、Git シークレットは最新のパイプライン実行時に削除されなくなりました。

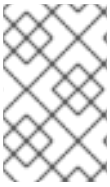
- この更新により、S2I クラスタータスクは一般提供のコンテナーイメージを使用します。

1.6. RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.11 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.11 が OpenShift Container Platform 4.12 以降のバージョンで利用できるようになりました。

1.6.1. 新機能

以下では、修正および安定性の面での改善点に加え、OpenShift Pipelines 1.11 の主な新機能について説明します。



注記

Red Hat OpenShift Pipelines Operator 1.11 にアップグレードする前に、少なくとも OpenShift Container Platform 4.12.19 または 4.13.1 バージョンがクラスターにインストールされていることを確認してください。

1.6.1.1. Pipelines

- この更新により、ARM ハードウェア上で実行される OpenShift Container Platform クラスター上で Red Hat OpenShift Pipelines を使用できるようになります。イメージが利用可能な **ClusterTask** リソースと、ARM ハードウェア上の Tekton CLI ツールがサポートされています。

- この更新により、**TektonConfig** CR で **enable-api-fields** 機能フラグを **beta** 値に設定した場合の、結果、オブジェクトパラメーター、配列結果、および配列へのインデックス作成のサポートが追加されました。

- 今回の更新により、伝播されたパラメーターが安定した機能の一部になりました。この機能により、埋め込み仕様のパラメーターを補間して、Tekton リソースの冗長性を軽減できます。

- この更新により、伝播されたワークスペースが安定した機能の一部になりました。伝達されたワークスペース機能を有効にするには、**enable-api-fields** 機能フラグを **alpha** 値または **beta** 値に設定します。

- 今回の更新により、**TaskRun** オブジェクトは、Pod の実行に失敗したときに初期コンテナーの失敗メッセージを取得してユーザーに表示します。

- この更新により、次のガイドラインに従ってマトリックスを設定しながら、パイプラインタスクのパラメーター、結果、およびコンテキストを置き換えることができます。

  - **matrix.params** 設定内の配列を **array** パラメーターに置き換えるか、文字列を **string**、**array**、または **object** パラメーターに置き換えます。

  - 文字列を **matrix.include** 設定の **string**、**array**、または **object** パラメーターに置き換えます。

  - パイプラインタスクのコンテキストを、**matrix.include** 設定内の別のコンテキストに置き換えます。

- この更新により、**TaskRun** リソース検証プロセスは、**matrix.include** パラメーターも検証します。検証では、すべてのパラメーターに値があり、指定された型と一致するかどうか、またオブジェクトパラメーターに必要なすべてのキーがあるかどうかがチェックされます。

- この更新により、**default-configs** config map に新しい **default-resolver-type** フィールドが追加されます。このフィールドの値を設定して、デフォルトのリゾルバーを設定できます。
- この更新により、**pipelineRun.workspaces.subPath** 設定で **PipelineRun** コンテキスト変数を定義して使用できるようになりました。
- この更新により、**ClusterResolver**、**BundleResolver**、**HubResolver**、および **GitResolver** の機能がデフォルトで利用できるようになりました。

1.6.1.2. トリガー

- この更新により、Tekton トリガーは **EventListener** 仕様の **Affinity** および **TopologySpreadConstraints** 値をサポートします。これらの値を使用して、**EventListener** オブジェクトの Kubernetes およびカスタムリソースを設定できます。
- この更新により、Slack でスラッシュコマンドを使用してフィールドを抽出できるようにする Slack インターセプターが追加されました。抽出されたフィールドは、HTTP リクエストのフォームデータセクションで送信されます。

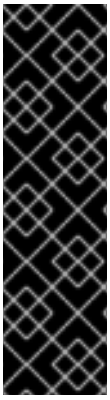
1.6.1.3. Operator

- この更新により、**TektonConfig** CR の **prune-per-resource** ブール型フィールドを設定することで、各 **PipelineRun** または **TaskRun** リソースのプルーニングを設定できるようになりました。また、**operator.tekton.dev/prune.prune-per-resource=true** アノテーションをネームスペースに追加することで、ネームスペース内の各 **PipelineRun** または **TaskRun** リソースのプルーニングを設定することもできます。
- 今回の更新により、OpenShift Container Platform クラスター全体のプロキシーに変更があった場合、Operator Lifecycle Manager (OLM) は Red Hat OpenShift Pipelines Operator を再作成します。
- この更新により、**TektonConfig** CR で **config.pruner.disabled** フィールドの値を **true** に設定することで、プルーナー機能を無効にすることができます。

1.6.1.4. Tekton Chains

- この更新により、Tekton Chains が一般的に使用できるようになりました。
- この更新により、Tekton チェーンで skopeo ツールを使用して、**cosign** 署名スキームで使用されるキーを生成できるようになりました。
- Red Hat OpenShift Pipelines Operator 1.11 にアップグレードすると、以前の Tekton Chains 設定が上書きされるため、**TektonConfig** CR で再度設定する必要があります。

1.6.1.5. Tekton Hub



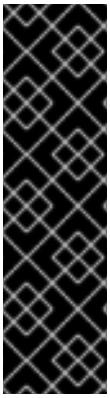
重要

Tekton Hub はテクノロジープレビュー機能としてのみ提供されます。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- この更新により、**resource** API レスポンスに新しい **resource/<catalog_name>/<kind>/<resource_name>/raw** エンドポイントと新しい **resourceURLPath** フィールドが追加されます。この更新は、リソースの最新の生の YAML ファイルを取得するのに役立ちます。

1.6.1.6. Tekton Results



重要

Tekton Results はテクノロジープレビューのみの機能です。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- この更新により、Tekton Results がオプションのコンポーネントとして Tekton Operator に追加されます。

1.6.1.7. Pipelines as Code

- 今回の更新により、Pipelines as Code では、**params** フィールドを使用して **PipelineRun** リソース内のカスタムパラメーターをデプロイメントできるようになりました。**Repository** CR のテンプレート内のカスタムパラメーターの値を指定できます。指定された値は、パイプライン実行のカスタムパラメーターを置き換えます。また、カスタムパラメーターを定義し、指定された条件が Common Expression Language (CEL) フィルターと互換性がある場合にのみその拡張を使用することもできます。

- この更新により、GitHub インターフェイスの **Checks** タブにある **Re-run all checks** ボタンをクリックすることで、特定のパイプラインまたはすべてのパイプラインを再実行できるようになりました。

- この更新により、新しい **tkn pac info** コマンドが Pipelines as Code CLI に追加されます。管理者は、**tkn pac info** コマンドを使用して、Pipelines as Code のインストールに関する次の詳細を取得できます。

  - Pipelines as Code がインストールされる場所。

  - Pipelines as Code のバージョン番号。

    - クラスター上に作成された **Repository** CR とリポジトリーに関連付けられた URL の概要。
    - インストールされている GitHub アプリケーションの詳細。
      このコマンドでは、**--github-api-url** 引数を使用してカスタム GitHub API URL を指定することもできます。

- この更新により、デフォルトですべての **PipelineRun** リソースのエラー検出が有効になります。Pipelines as Code は、**PipelineRun** リソースの実行が失敗したかどうかを検出し、エラーの最後の数行のスニペットを表示します。GitHub アプリケーションの場合、Pipelines as Code はコンテナーログ内のエラーメッセージを検出し、プルリクエストのアノテーションとして公開します。

- この更新により、プライベート Git リポジトリーに接続されたプライベート Tekton Hub インスタンスからタスクをフェッチできるようになります。この更新を有効にするために、Pipelines as Code は、GitHub の生 URL を使用する代わりに、プライベート Tekton Hub インスタンスの内部生 URL を使用します。

- この更新の前に、Pipelines as Code は namespace の詳細を含まないログを提供していました。この更新により、Pipelines as Code は namespace 情報をパイプラインログに追加するため、namespace に基づいてログをフィルタリングし、簡単にデバッグできるようになります。

- この更新により、**PipelineRun** リソース定義の取得元の出所ソースを定義できるようになりました。デフォルトでは、Pipelines as Code は、イベントがトリガーされたブランチから **PipelineRun** リソース定義をフェッチします。**Pipelinerun_provenance** 設定の値を **default_branch** に設定できるようになりました。これにより、**PipelineRun** リソース定義が GitHub で設定されたリポジトリーのデフォルトブランチからフェッチされるようになります。

- この更新により、GitHub トークンのスコープを次のレベルで拡張できます。
    - リポジトリーレベル: このレベルを使用して、元のリポジトリーが存在するのと同じ名前空間に存在するリポジトリーにスコープを拡張します。
    - グローバルレベル: このレベルを使用して、スコープを別の名前空間に存在するリポジトリーに拡張します。

- 今回の更新により、Pipelines as Code は、所有者、コラボレーター、またはパブリックメンバーではないユーザー、または **owner** ファイルにリストされていないがリポジトリーに変更をプッシュする権限を持つユーザーが作成したプルリクエストに対して、CI パイプラインをトリガーします。

- この更新により、カスタムコンソール設定により、**Repository** CR のカスタムパラメーターを使用できるようになります。

- この更新により、Pipelines as Code はすべての **PipelineRun** ラベルを **PipelineRun** アノテーションに変更します。**PipelineRun** ラベルを使用する代わりに、**PipelineRun** アノテーションを使用して Tekton リソースをマークできます。

- この更新により、**pac-config-logging** config map をウォッチャーおよび Webhook リソースに使用できるようになりますが、コードコントローラーとしてのパイプラインには使用できません。

1.6.2. 互換性を失わせる変更点

- この更新により、パイプライン仕様内の **resource-verification-mode** 機能フラグが新しい **Trusted-resources-verification-no-match-policy** フラグに置き換えられます。

- この更新により、Tekton Chains CR を編集できなくなります。代わりに、**TektonConfig** CR を編集して Tekton Chains を設定します。

1.6.3. 非推奨および削除された機能

- この更新により、Tekton CLI から **PipelineResource** コマンドと参照のサポートが削除されます。
  - クラスタータスクからのパイプラインリソースの削除
  - タスクからのパイプラインリソースの削除
  - パイプラインからのパイプラインリソースの削除
  - リソースコマンドの削除
  - **clustertask describe** コマンドからの入出力リソースの削除
- この更新により、Tekton CLI から **full** 埋め込みステータスのサポートが削除されます。
- **taskref.bundle** と **Pipelineref.bundle** バンドルは非推奨であり、将来のリリースでは削除される予定です。
- Red Hat OpenShift Pipelines 1.11 では、**PipelineResource** CR のサポートが削除されているため、代わりに **Task** CR を使用してください。
- Red Hat OpenShift Pipelines 1.11 では、**v1alpha1.Run** オブジェクトのサポートが削除されました。このリリースにアップグレードする前に、オブジェクトを **v1alpha1.Run** から **v1beta1.CustomRun** にアップグレードする必要があります。
- Red Hat OpenShift Pipelines 1.11 では、**custom-task-version** 機能フラグが削除されました。
- Red Hat OpenShift Pipelines 1.11 では、**pipelinerun.status.taskRuns** フィールドと **Pipelinerun.status.runs** フィールドが **embedded-status** 機能フラグとともに削除されました。代わりに、**pipelinerun.status.childReferences** フィールドを使用してください。

1.6.4. 既知の問題

- **prune-per-resource** のブール型フィールドを設定しても、**PipelineRun** リソースまたは **TaskRun** リソースがパイプラインまたはタスクの一部ではない場合、それらのリソースは削除されません。
- Tekton CLI では、リゾルバーを使用して作成された **PipelineRun** リソースのログは表示されません。
- **order_by=created_time+desc&page_size=1** クエリーに基づいてパイプライン結果をフィルターすると、出力に **nextPageToken** 値のないレコードがゼロになります。
- **loglevel.pipelinesascode** フィールドの値を **debug** に設定すると、Pipelines as Code コントローラー Pod にデバッグログは生成されません。回避策として、パイプラインをコードコントローラー Pod として再起動します。

1.6.5. 修正された問題

- この更新が行われる前は、Pipelines as Code は **PipelineRun** CR の **generateName** フィールドの検出中に **PipelineRun** リソースの作成に失敗していました。この更新により、Pipelines as Code は **PipelineRun** CR での **generateName** フィールドの提供をサポートします。

- この更新より前は、Web コンソールから **PipelineRun** リソースを作成すると、すべてのアノテーションがパイプラインからコピーされ、実行中のノードに問題が発生していました。この更新により問題が解決されました。

- この更新により、**keep** フラグの **tkn pr delete** コマンドが修正されました。ここで、**keep** フラグの値が関連するタスクの実行またはパイプラインの実行の数と等しい場合、コマンドはメッセージとともに終了コード **0** を返します。

- この更新前は、Tekton Operator はカスタマイズのためのパフォーマンス設定フィールドを公開していませんでした。この更新により、クラスター管理者は、ニーズに基づいて **TektonConfig** CR の次のパフォーマンス設定フィールドをカスタマイズできます。

  - **disable-ha**
  - **buckets**
  - **kube-api-qps**
  - **kube-api-burst**
  - **threads-per-controller**

- この更新により、バンドル内の **dev.tekton.image.kind** アノテーション値と **kind** フィールドの大文字と小文字を区別しない比較を実行するようにリモートバンドルリゾルバーが修正されました。

- この更新前は、大規模な Git リポジトリーのクローンを作成するときにメモリー不足が原因で、リモートリゾルバーの Pod が終了していました。この更新プログラムでは問題が修正され、リモートリゾルバーをデプロイメントするためのメモリー制限が増加します。

- この更新により、**v1** タイプのタスクおよびパイプラインリソースがリモート解決でサポートされます。

- この更新により、API からの埋め込み **TaskRun** ステータスの削除が元に戻ります。埋め込み **TaskRun** ステータスは、古いバージョンのクライアント/サーバーとの互換性をサポートするために、非推奨の機能として利用できるようになりました。

- この更新前は、実行に必要でない場合でも、すべてのアノテーションが **PipelineRun** と **TaskRun** リソースにマージされていました。この更新により、アノテーションを **PipelineRun** と **TaskRun** リソースにマージすると、**last-applied-configuration** のアノテーションがスキップされます。

- この更新により、回帰の問題が修正され、パイプライン結果でスキップされたタスクの結果が検証されなくなります。たとえば、パイプライン結果がスキップされた **PipelineTask** リソースを参照している場合、パイプライン結果は出力されず、結果の欠落によって **PipelineRun** の実行が失敗することはありません。

- この更新では、Pod のステータスメッセージを使用して、Pod 終了の原因を特定します。

- この更新前は、**finally** タスクの実行に対してデフォルトのリゾルバーが設定されていませんでした。この更新により、**finally** タスクのデフォルトのリゾルバーが設定されます。

- この更新により、Red Hat OpenShift Pipelines は、リモート解決を使用するときに時々発生する **TaskRun** または **PipelineRun** の実行の失敗を回避します。

- この更新が行われる前は、長時間のパイプラインの実行は、タイムアウト後でもクラスター上で実行状態のままになります。今回の更新でこの問題が修正されています。

- この更新により、**keep** フラグを正しく使用できるように **tkn pr delete** コマンドが修正されました。この更新で、**keep** フラグの値が関連するタスクの実行またはパイプラインの実行の数と等しい場合、**tkn pr delete** コマンドはメッセージとともに終了コード **0** を返します。

1.6.6. Red Hat OpenShift Pipelines General Availability 1.11.1 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.11.1 が OpenShift Container Platform 4.12 以降のバージョンで利用できるようになりました。

1.6.6.1. 修正された問題

- この更新より前は、実行中または保留中の Pod がプリエンプトされると、タスクの実行が失敗し、マウントパスのエラーメッセージが表示されることがありました。この更新により、クラスターによって Pod が削除され再作成された場合でも、タスクの実行は失敗しません。

- この更新前は、タスク内のシェルスクリプトを root として実行する必要がありました。この更新により、シェルスクリプトイメージに非 root ユーザー ID が設定されるようになり、**git-clone** タスクなどのシェルスクリプトを含むタスクを Pod 内で非 root ユーザーとして実行できるようになりました。

- この更新より前の Red Hat OpenShift Pipelines 1.11.0 では、Pipelines as Code を使用してパイプライン実行が定義されると、Git リポジトリー内の定義は **tekton.dev/v1beta1** API バージョンを参照し、**spec.pipelineRef.bundle** エントリー、バンドル参照の **kind** パラメーターが誤って **Task** に設定されました。この問題は、Red Hat OpenShift Pipelines の以前のバージョンには存在しませんでした。今回の更新により、**kind** パラメーターが正しく設定されるようになりました。

- この更新前は、**disable-ha** フラグが **tekton-pipelines** コントローラーに正しく渡されなかったため、Red Hat OpenShift Pipelines の高可用性 (HA) 機能を有効にすることができませんでした。この更新により、**disable-ha** フラグが正しく渡され、必要に応じて HA 機能を有効にできるようになりました。

- この更新前は、Tekton Hub と Artifact Hub の URL をハブリゾルバーに設定できませんでした。そのため、Tekton Hub と Artifact Hub のプリセットアドレスのみを使用できました。この更新により、インストールしたカスタム Tekton Hub インスタンスを使用するなど、ハブリゾルバーの Tekton Hub および Artifact Hub の URL を設定できるようになります。

- 今回の更新により、**git-init** イメージの SHA ダイジェストは、イメージの現在のリリースバージョンであるバージョン 1.10.5 に対応します。

- この更新前は、**tekton-pipelines-controller** コンポーネントは **config-leader-election** という名前の config map を使用していました。この名前は knative コントローラーのデフォルト値であるため、OpenShift Pipelines の設定プロセスが他のコントローラーに影響を与える可能性があり、またその逆も同様です。この更新により、コンポーネントは一意の設定名を使用するため、OpenShift Pipelines の設定プロセスは他のコントローラーに影響を与えず、他のコントローラーの影響も受けません。

- この更新より前は、GitHub リポジトリーへの書き込みアクセス権を持たないユーザーがプルリクエストを開いた場合、Pipelines as Code CI/CD アクションは GitHub で **skipped** ものとして表示されていました。この更新により、Pipelines as Code CI/CD アクションが GitHub で **Pending approval** として表示されます。

- この更新が行われる前は、Pipelines as Code は、設定されたブランチ名と一致するブランチへ

のすべてのプルリクエストに対して CI/CD アクションを実行していました。今回の更新により、Pipelines as Code は、プルリクエストのソースブランチが、設定されたブランチ名と正確に一致する場合にのみ CI/CD アクションを実行するようになります。

- この更新が行われる前は、Pipelines as Code コントローラーのメトリクスは OpenShift Container Platform 開発者コンソールに表示されませんでした。今回の更新により、Pipelines as Code コントローラーのメトリクスが開発者コンソールに表示されます。

- この更新より前の Red Hat OpenShift Pipelines 1.11.0 では、Operator は常に Tekton Chains をインストールしており、Tekton Chains コンポーネントのインストールを無効にすることはできませんでした。今回の更新により、**TektonConfig** CR で **disabled** パラメーターの値を **true** に設定して、okindf Tekton Chains のインストールを無効にすることができます。

- この更新の前に、**TektonChain** CR を使用して古いバージョンの OpenShift Pipelines で Tekton Chains を設定し、その後 OpenShift Pipelines バージョン 1.11.0 にアップグレードした場合、設定情報は上書きされました。この更新により、OpenShift Pipelines の古いバージョンからアップグレードし、**TektonConfig** がインストールされているのと同じ namespace (**openshift-pipelines**) で Tekton Chains が設定されていた場合、Tekton Chains の設定情報は保持されます。

1.6.7. Red Hat OpenShift Pipelines 一般提供 1.11.2 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.11.2 が OpenShift Container Platform 4.12 以降のバージョンで利用できるようになりました。

この更新には、**tkn** コマンドラインツールの更新バージョンが含まれています。このツールの更新バージョンは次の場所からダウンロードできます。

- Linux (x86_64, amd64)
- Linux on IBM zSystems および IBM® LinuxONE (s390x)
- Linux on IBM Power (ppc64le)
- ARM 上の Linux (aarch64、arm64)
- Windows
- MacOS
- ARM 上の macOS

Red Hat Enterprise Linux (RHEL) 上の RPM を使用して **tkn** コマンドラインツールをインストールした場合は、**yum update** コマンドを使用して更新されたバージョンをインストールします。

1.6.7.1. 修正された問題

- この更新前は、**tkn pac resolve -f** コマンドは、Git リポジトリーでの認証のための既存のシークレットを検出しませんでした。今回の更新により、このコマンドはシークレットを正常に検出できるようになりました。

- 今回の更新により、**tkn pacsolve** コマンドで **--v1beta1** フラグを使用できるようになりました。**v1beta1** API バージョンスキーマで実行するパイプラインを生成する場合は、このフラグを使用します。

- この更新前は、パイプライン実行がリゾルバーを参照している場合、**tkn pr logs** コマンドはパイプライン実行のログを表示できませんでした。今回の更新により、コマンドでログが表示されるようになりました。

- 今回の更新により、**git-init** イメージの SHA ダイジェストは、イメージの現在のリリースバージョンであるバージョン 1.12.1 に対応します。

- この更新により、HTTP/2.0 プロトコルは Webhook でサポートされなくなりました。Red Hat OpenShift Pipelines へのすべての Webhook 呼び出しでは、HTTP/1.1 プロトコルを使用する必要があります。

1.6.8. 既知の問題

- Bundles リゾルバーを使用してパイプライン実行を定義し、このパイプライン実行に対して **tkn pac resolve --v1beta1** コマンドを使用すると、コマンドによって不正な YAML 出力が生成されます。バンドルの **kind** パラメーターは、YAML 出力の **Task** に設定されます。回避策として、YAML データに正しい値を手動で設定できます。あるいは、**opc pac resolve --v1beta1** コマンドを使用するか、OpenShift Pipelines バージョン 1.12.0 以降に含まれるバージョンの **tkn** ツールを使用することもできます。

1.6.9. Red Hat OpenShift Pipelines 一般提供 1.11.3 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.11.3 が OpenShift Container Platform 4.11、4.12、4.13 に加え、4.10 でも利用できるようになりました。

1.6.9.1. 修正された問題

- 今回の更新より前は、パイプラインの最後のタスクが失敗するかスキップされた場合、OpenShift Pipelines は検証エラーを報告しました。今回の更新により、パイプラインは、最後のタスクが失敗するかスキップされた場合でも成功するようになりました。

1.7. RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.10 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.10 が OpenShift Container Platform 4.11、4.12、および 4.13 で利用できるようになりました。

1.7.1. 新機能

以下では、修正および安定性の面での改善点に加え、OpenShift Pipelines 1.10 の主な新機能について説明します。

1.7.1.1. Pipelines

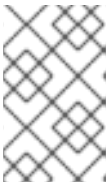
- 今回の更新により、**PipelineRun** または **TaskRun** Pod テンプレートで環境変数を指定して、タスクまたはステップで設定されている変数を上書きまたは追加できるようになりました。また、デフォルトの Pod テンプレートで環境変数を指定して、それらの変数をすべての **PipelineRuns** および **TaskRuns** に対してグローバルに使用することもできます。今回の更新では、Pod テンプレートからの伝播中に環境変数をフィルター処理する、**obhibited-envs** という名前の新しいデフォルト設定も追加されています。

- 今回の更新により、パイプラインのカスタムタスクがデフォルトで有効になります。

注記

この更新を無効にするには、**feature-flags** config カスタムリソースで **enable-custom-tasks** フラグを **false** に設定します。

- この更新プログラムは、カスタムタスクの **v1beta1.CustomRun** API バージョンをサポートします。
- 今回の更新により、カスタム実行を作成するための **PipelineRun** reconcilerのサポートが追加されました。たとえば、**custom-task-version** 機能フラグがデフォルト値の **v1alpha1** ではなく **v1beta1** に設定されている場合、**PipelineRuns** から作成されたカスタム **TaskRun** は**v1alpha1.Run** の代わりに **v1beta1.CustomRun** API バージョンを使用できるようになりました。



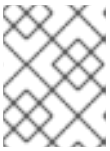
注記

v1beta1.CustomRun 要求に応答するには 、***v1alpha1.Run** ではなく ***v1beta1.CustomRun** API バージョンをリッスンするようにカスタムタスクコントローラーを更新する必要があります。

- この更新により、新しい **retries** フィールドが **v1beta1.TaskRun** および **v1.TaskRun** 仕様に追加されます。

1.7.1.2. トリガー

- 今回の更新により、トリガーは、**v1beta1** API バージョンの **CustomRun** オブジェクトと共に、**v1** API バージョンの **Pipelines**、**Tasks**、**PipelineRuns**、および **TaskRuns** オブジェクトの作成をサポートします。
- 今回の更新により、GitHub Interceptor は、所有者または所有者による設定可能なコメントで呼び出されない限り、プルリクエストトリガーの実行をブロックします。



注記

この更新を有効または無効にするには、GitHub Interceptor 設定ファイルで **githubOwners** パラメーターの値を **true** または **false** に設定します。

- 今回の更新により、GitHub Interceptor は、プッシュおよびプルリクエストイベント用に変更されたすべてのファイルのコンマ区切りのリストを追加できるようになりました。変更されたファイルのリストは、最上位の拡張フィールドのイベントペイロードの **changed_files** がプロパティーに追加されます。
- 今回の更新により、TLS の **MinVersion** が **tls.VersionTLS12** に変更され、Federal Information Processing Standards (FIPS) モードが有効になっている場合に OpenShift Container Platform でトリガーが実行されるようになります。

1.7.1.3. CLI

- 今回の更新で、**Task**、**ClusterTask** または **Pipeline** が開始時に Container Storage Interface (CSI) ファイルをワークスペースとして渡すためのサポートが追加されました。
- この更新により、タスク、パイプライン、パイプライン実行、およびタスク実行リソースに関連付けられたすべての CLI コマンドに **v1** API サポートが追加されます。Tekton CLI は、これらのリソースの **v1beta1** と **v1** API の両方で動作します。

- 今回の更新で、**start** コマンドと **describe** コマンドにオブジェクトタイプパラメーターのサポートが追加されました。

1.7.1.4. Operator

- 今回の更新により、オプションのパイプラインプロパティーに **default-forbidden-env** パラメーターが追加されました。パラメーターには、Pod テンプレートを介して提供された場合に伝播されるべきではない、禁止された環境変数が含まれています。
- この更新により、Tekton Hub UI でのカスタムロゴのサポートが追加されます。カスタムロゴを追加するには、**customLogo** パラメーターの値を、Tekton Hub CR の base64 でエンコードされたロゴの URI に設定します。
- この更新により、git-clone タスクのバージョン番号が 0.9 に増加します。

1.7.1.5. Tekton Chains

重要

Tekton Chains はテクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- 今回の更新により、**PipelineRun** および **TaskRun** 設定証明にアノテーションとラベルが追加されました。
- この更新により、**slsa/v1** という名前の新しい形式が追加されます。これは、**in-toto** 形式で要求したときに生成されるものと同じ来歴を生成します。
- 今回の更新により、Sigstore 機能が実験的機能から除外されました。
- 今回の更新により、**predicate.materials** 関数に、**TaskRun** オブジェクトのすべてのステップとサイドカーからのイメージ URI とダイジェスト情報が含まれるようになりました。

1.7.1.6. Tekton Hub

重要

Tekton Hub はテクノロジープレビュー機能としてのみ提供されます。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- この更新は、クラスターでの **v1** API バージョンの Tekton リソースのインストール、アップグレード、またはダウングレードをサポートします。
- この更新では、UI の Tekton Hub ロゴの代わりにカスタムロゴを追加できます。
- 今回の更新では、アーティファクトハブからリソースを取得してクラスターにインストールする **--type アーティファクト** フラグを追加することで、**tkn ハブインストール** コマンドの機能を拡張します。
- 今回の更新により、Artifact Hub からクラスターにインストールされるリソースにラベルとしてサポート層、カタログ、および組織情報が追加されます。

1.7.1.7. Pipelines as Code

- この更新により、着信 Webhook のサポートが強化されます。OpenShift Container Platform クラスターにインストールされた GitHub アプリケーションの場合、受信 Webhook に **git_provider** 仕様を提供する必要はありません。代わりに、Pipelines as Code がシークレットを検出し、それを着信 Webhook に使用します。
- 今回の更新により、同じトークンを使用して、GitHub 上の同じホストからデフォルト以外のブランチでリモートタスクを取得できるようになりました。
- 今回の更新により、Pipelines as Code は Tekton **v1** テンプレートをサポートします。**v1** および **v1beta1** テンプレートを使用できます。これは、Pipelines as Code が PR 生成のために読み取るものです。PR はクラスターで **v1** として作成されます。
- この更新の前は、OpenShift コンソール UI は、OpenShift namespace でランタイムテンプレートが見つからない場合、ハードコーディングされたパイプライン実行テンプレートをフォールバックテンプレートとして使用していました。**Pipelines-as-Code** config map のこの更新により、使用するコンソール用に、**pipelines-as-code-template-default** という名前の新しいデフォルトのパイプライン実行テンプレートが提供されます。
- 今回の更新により、Pipelines as Code は Tekton Pipelines 0.44.0 最小ステータスをサポートします。
- 今回の更新により、Pipelines as Code は Tekton **v1** API をサポートします。これは、Pipelines as Code が Tekton v0.44 以降と互換性を持つようになったことを意味します。
- 今回の更新により、OpenShift のコンソールと k8s の Tekton ダッシュボードの設定に加えて、カスタムコンソールダッシュボードを設定できるようになりました。
- 今回の更新により、Pipelines as Code は **tkn pac create repo** コマンドを使用して開始された GitHub アプリケーションのインストールを検出し、グローバルにインストールされている場合は GitHub Webhook を必要としません。
- この更新の前は、**PipelineRun** にアタッチされたタスクではなく **PipelineRun** の実行でエラーが発生した場合、Pipelines as Code は失敗を適切に報告しませんでした。今回の更新により、コードとしてのパイプラインは、**PipelineRun** を作成できなかった場合に GitHub チェックでエラーを適切に報告します。
- 今回の更新により、Pipelines as Code には、**PipelineRun** が実行される現在実行中の namespace にデプロイメントされる **target_namespace** 変数が含まれています。
- 今回の更新により、Pipelines as Code を使用すると、CLI ブートストラップ GitHub アプリケーションで GitHub エンタープライズの質問をバイパスできます。

- 今回の更新により、Pipelines as Code はリポジトリー CR が見つからない場合にエラーを報告しなくなりました。
- 今回の更新により、Pipelines as Code は、同じ名前の複数のパイプライン実行が見つかった場合にエラーを報告します。

1.7.2. 互換性を失わせる変更点

- 今回の更新により、以前のバージョンの **tkn** コマンドは Red Hat OpenShift Pipelines 1.10 と互換性がなくなりました。
- この更新により、Tekton CLI から **Cluster** および **CloudEvent** パイプラインリソースのサポートが削除されます。**tkn pipelineresource create** コマンドを使用してパイプラインリソースを作成することはできません。また、パイプラインリソースは、タスク、クラスタータスク、またはパイプラインの **start** コマンドでサポートされなくなりました。
- この更新により、Tekton Chains から来歴フォーマットとしての **tekton** が削除されます。

1.7.3. 非推奨および削除された機能

- Red Hat OpenShift Pipelines 1.10 では、**ClusterTask** コマンドが非推奨になり、将来のリリースで削除される予定です。**tkn task create** コマンドも、この更新で非推奨になりました。
- Red Hat OpenShift Pipelines 1.10 では、**v1** API がパイプラインリソースをサポートしていないため、**tkn task start** コマンドで使用されたフラグ **-i** および **-o** は非推奨になりました。
- Red Hat OpenShift Pipelines 1.10 では、**v1** API がパイプラインリソースをサポートしていないため、**tkn pipeline start** コマンドで使用されたフラグ **-r** は非推奨になりました。
- Red Hat OpenShift Pipelines 1.10 の更新では、**openshiftDefaultEmbeddedStatus** パラメーターが **full** 埋め込みステータスと **min** 埋め込みステータスの **both** に設定されます。デフォルトの埋め込みステータスを変更するフラグも非推奨であり、削除されます。さらに、パイプラインのデフォルトの埋め込みステータスは、将来のリリースで **minimal** に変更される予定です。

1.7.4. 既知の問題

- この更新には、以下の下位互換性のない変更が含まれています。
  - **PipelineResources** クラスターの削除
  - **PipelineResources** クラウドイベントの削除
- クラスターのアップグレード後にパイプラインメトリクス機能が動作しない場合は、回避策として次のコマンドを実行します。

```
$ oc get tektoninstallersets.operator.tekton.dev | awk '/pipeline-main-static/ {print $1}' | xargs oc delete tektoninstallersets
```

- 今回の更新により、Crunchy PostgreSQL などの外部データベースの使用は、IBM Power、IBM zSystems、および IBM® LinuxONE ではサポートされなくなりました。代わりに、デフォルトの Tekton Hub データベースを使用してください。

1.7.5. 修正された問題

- この更新の前は、**opc pac** コマンドはヘルプを表示する代わりにランタイムエラーを生成していました。今回の更新により、**opc pac** コマンドがヘルプメッセージを表示するように修正されました。

- この更新の前は、**tkn pac create repo** コマンドを実行するには、リポジトリーを作成するための webhook の詳細が必要でした。今回の更新により、GitHub アプリケーションがインストールされている場合、**tkn-pac create repo** コマンドは Webhook を設定しません。

- この更新の前は、Tekton Pipelines で **PipelineRun** リソースの作成に問題があった場合、Pipelines as Code はパイプライン実行の作成エラーを報告しませんでした。たとえば、パイプラインの実行に存在しないタスクは、ステータスを表示しません。今回の更新により、Pipelines as Code は、欠落しているタスクとともに Tekton Pipelines からの適切なエラーメッセージを表示します。

- この更新プログラムは、認証が成功した後の UI ページのリダイレクトを修正します。これで、Tekton Hub にログインしようとしたのと同じページにリダイレクトされます。

- 今回の更新では、クラスタータスク、個々のタスク、およびパイプラインに対して、これらのフラグ **--all-namespaces** および **--output=yaml** を使用した **list** コマンドが修正されました。

- 今回の更新により、**repo.spec.url** URL の末尾にあるスラッシュが削除され、GitHub からの URL と一致するようになりました。

- この更新の前は、**marshalJSON** 関数はオブジェクトのリストをマーシャリングしませんでした。今回の更新で、**marshalJSON** 関数はオブジェクトのリストをマーシャリングします。

- 今回の更新により、Pipelines as Code を使用すると、CLI ブートストラップ GitHub アプリケーションで GitHub エンタープライズの質問をバイパスできます。

- この更新により、リポジトリーに 100 人を超えるユーザーがいる場合の GitHub コラボレーターチェックが修正されます。

- 今回の更新により、タスクまたはパイプラインの **sign** および **verify** コマンドは、kubernetes 設定ファイルなしで機能するようになりました。

- 今回の更新により、namespace でプルーナーがスキップされた場合、Tekton Operator は残りのプルーナー cron ジョブをクリーンアップします。

- この更新の前に、API **ConfigMap** オブジェクトは、カタログ更新間隔のユーザー設定値で更新されませんでした。この更新により、Tekon Hub CR の **CATALOG_REFRESH_INTERVAL** API が修正されます。

- この更新プログラムは、**EmbeddedStatus** 関数フラグを変更するときの **PipelineRunStatus** の調整を修正します。この更新により、次のパラメーターがリセットされます。

  - **status.runs** および **status.taskruns** パラメーターを最小の **EmbeddedStatus** で **nil** に設定

  - **full EmbeddedStatus** で **status.childReferences** パラメーターを **nil** に

- 今回の更新で、**ResolutionRequest** CRD に変換設定が追加されました。この更新により、**v1alpha1.ResolutionRequest** リクエストから **v1beta1.ResolutionRequest** リクエストへの変換が適切に設定されます。

- この更新プログラムは、パイプラインタスクに関連付けられている重複したワークスペースをチェックします。

- この更新により、コードでリゾルバーを有効にするためのデフォルト値が修正されます。

- この更新プログラムは、リゾルバーを使用した **TaskRef** および **PipelineRef** 名の変換を修正します。

1.7.6. Red Hat OpenShift Pipelines General Availability 1.10.1 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.10.1 が OpenShift Container Platform 4.11、4.12、および 4.13 で利用できるようになりました。

1.7.6.1. Pipelines as Code の修正された問題

- この更新の前は、ペイロードからのソースブランチ情報に **refs/heads/** が含まれていたが、ユーザーが設定したターゲットブランチにブランチ名 **main** のみが CEL 式に含まれていた場合、プッシュリクエストは失敗していました。今回の更新により、ベースブランチまたはターゲットブランチのペイロードに **refs/heads/** がある場合、Pipelines as Code はプッシュリクエストを渡し、パイプラインをトリガーします。

- この更新の前は、**PipelineRun** オブジェクトを作成できなかった場合、Tekton コントローラーから受け取ったエラーがユーザーに報告されませんでした。今回の更新により、Pipelines as Code はエラーメッセージを GitHub インターフェイスに報告し、ユーザーがエラーをトラブルシューティングできるようにします。Pipelines as Code は、パイプラインの実行中に発生したエラーも報告します。

- 今回の更新により、Pipelines as Code は、インフラストラクチャーの問題により OpenShift Container Platform クラスターでシークレットを作成できなかった場合に、シークレットを GitHub のチェックインターフェイスにエコーしません。

- 今回の更新により、使用されなくなった非推奨の API が Red Hat OpenShift Pipelines から削除されます。

1.7.7. Red Hat OpenShift Pipelines General Availability 1.10.2 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.10.2 が OpenShift Container Platform 4.11、4.12、および 4.13 で利用できるようになりました。

1.7.7.1. 修正された問題

この更新前は、Tekton Operator の問題により、ユーザーは **enable-api-fields** フラグの値を **beta** に設定できませんでした。今回の更新でこの問題が修正されています。**TektonConfig** CR で、**enable-api-fields** フラグの値を **beta** に設定できるようになりました。

1.7.8. Red Hat OpenShift Pipelines General Availability 1.10.3 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.10.3 が OpenShift Container Platform 4.11、4.12、および 4.13 で利用できるようになりました。

1.7.8.1. 修正された問題

この更新前は、Tekton Operator はカスタマイズのためのパフォーマンス設定フィールドを公開していませんでした。この更新により、クラスター管理者は、ニーズに基づいて **TektonConfig** CR の次のパフォーマンス設定フィールドをカスタマイズできます。

- **disable-ha**

- **buckets**

- **kube-api-qps**
- **kube-api-burst**
- **threads-per-controller**

1.7.9. Red Hat OpenShift Pipelines General Availability 1.10.4 のリリースノート

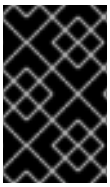
今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.10.4 が OpenShift Container Platform 4.11、4.12、および 4.13 で利用できるようになりました。

1.7.9.1. 修正された問題

- この更新により、パイプライン実行における **PipelineRef** フィールドのバンドルリゾルバー変換の問題が修正されます。現在、変換機能は、変換後に **kind** フィールドの値を **Pipeline** に設定します。
- この更新前は、**pipelinerun.timeouts** フィールドは **timeouts.pipeline** 値にリセットされ、**timeouts.tasks** 値と **timeouts.finally** 値は無視されました。この更新により問題が修正され、**PipelineRun** リソースの正しいデフォルトのタイムアウト値が設定されます。
- この更新前は、コントローラーのログに不要なデータが含まれていました。今回の更新でこの問題が修正されています。

1.7.10. Red Hat OpenShift Pipelines General Availability 1.10.5 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.10.5 が OpenShift Container Platform 4.11、4.12、4.13 に加え、4.10 でも利用できるようになりました。



重要

Red Hat OpenShift Pipelines 1.10.5 は、OpenShift Container Platform 4.10、4.11、4.12、および 4.13 の **pipelines-1.10** チャネルでのみ使用できます。OpenShift Container Platform バージョンの **latest** チャネルでは利用できません。

1.7.10.1. 修正された問題

- この更新が行われる前は、**oc** および **tkn** コマンドを使用しても、大規模なパイプライン実行がリストされたり、削除されませんでした。この更新では、この問題の原因となっていた巨大なアノテーションを圧縮することで、この問題を軽減します。圧縮後もパイプラインの実行が大きすぎる場合は、同じエラーが再発することに注意してください。
- この更新より前は、**pipelineRun.spec.taskRunSpecs.podTemplate** オブジェクトで指定された Pod テンプレートのみがパイプライン実行の対象となります。この更新により、**pipelineRun.spec.podTemplate** オブジェクトで指定された Pod テンプレートも考慮され、**pipelineRun.spec.taskRunSpecs.podTemplate** オブジェクトで指定されたテンプレートとマージされます。

1.7.11. Red Hat OpenShift Pipelines 一般提供 1.10.6 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.10.6 が OpenShift Container Platform 4.10、4.11、4.12、および 4.13 で利用できるようになりました。

この更新には、**tkn** コマンドラインツールの更新バージョンが含まれています。このツールの更新バージョンは次の場所からダウンロードできます。

- Linux (x86_64, amd64)
- Linux on IBM zSystems および IBM® LinuxONE (s390x)
- Linux on IBM Power (ppc64le)
- ARM 上の Linux (aarch64、arm64)
- Windows
- MacOS
- ARM 上の macOS

Red Hat Enterprise Linux (RHEL) 上の RPM を使用して **tkn** コマンドラインツールをインストールした場合は、**yum update** コマンドを使用して更新されたバージョンをインストールします。

1.7.11.1. 既知の問題

- **tkn task start** コマンドまたは **tkn clustertask start** コマンドを入力すると、**tkn** コマンドラインユーティリティーによってエラーメッセージが表示されます。回避策として、コマンドラインを使用してタスクまたはクラスタータスクを開始するには、OpenShift Pipelines 1.11 以降のバージョンに付属の **tkn** ユーティリティーのバージョンを使用します。

1.7.11.2. 修正された問題

- この更新により、S2I クラスタータスクは一般提供のコンテナーイメージを使用します。
- この更新により、HTTP/2.0 プロトコルは Webhook でサポートされなくなりました。Red Hat OpenShift Pipelines へのすべての Webhook 呼び出しでは、HTTP/1.1 プロトコルを使用する必要があります。

1.8. RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.9 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.9 が OpenShift Container Platform 4.11、4.12、および 4.13 で利用できるようになりました。

1.8.1. 新機能

以下では、修正および安定性の面での改善点に加え、OpenShift Pipelines 1.9 の主な新機能について説明します。

1.8.1.1. Pipelines

- 今回の更新により、パイプラインパラメーターと結果を配列とオブジェクトディクショナリー形式で指定できるようになりました。
- この更新により、Container Storage Interface (CSI) およびワークスペースの projected ボリュームがサポートされます。
- 今回の更新により、パイプラインステップを定義するときに **stdoutConfig** および **stderrConfig** パラメーターを指定できるようになりました。これらのパラメーターを定義すると、ステップに関連付けられた標準出力と標準エラーをローカルファイルにキャプチャーするのに役立ちます。

- 今回の更新により、**steps.onError** イベントハンドラーに **$(params.CONTINUE)** などの変数を追加できるようになりました。

- 今回の更新により、**PipelineResults** 定義で **finally** タスクからの出力を使用できるようになりました。たとえば **$(finally.<pipelinetask-name>.result.<result-name>)** では、**<pipelinetask-name>** はパイプラインタスク名を表し、**<result-name>** は結果名を表します。

- この更新では、タスク実行のタスクレベルのリソース要件をサポートがされます。

- 今回の更新により、名前に基づいて、パイプラインと定義されたタスクの間で共有されるパラメーターを再作成する必要がなくなりました。この更新は、開発者プレビュー機能の一部です。

- この更新により、組み込みの git、クラスター、バンドル、およびハブリゾルバーなどのリモート解決のサポートが追加されます。

1.8.1.2. トリガー

- 今回の更新では、**NamespacedInterceptor** を定義する **Interceptor** CRD が追加されました。**NamespacedInterceptor** は、トリガー内のインターセプター参照の **kind** セクションまたは **EventListener** 仕様で使用できます。

- この更新により **CloudEvents** が有効になります。

- 今回の更新により、トリガーを定義するときに Webhook ポート番号を設定できるようになりました。

- 今回の更新では、トリガー **eventID** を使用した **TriggerBinding** への入力がサポートされるようになりました。

- この更新では、**ClusterInterceptor** サーバーの証明書の検証とローテーションがサポートされています。

    - トリガーは、コアインターセプターの証明書を検証し、証明書の有効期限が切れると新しい証明書を **ClusterInterceptor** にローテーションします。

1.8.1.3. CLI

- 今回の更新では、**describe** コマンドでのアノテーションの表示がサポートされています。

- 今回の更新では、**pr describe** コマンドでのパイプライン、タスク、およびタイムアウトの表示がサポートされています。

- 今回の更新では、**pipeline start** コマンドでパイプライン、タスク、およびタイムアウトを提供するフラグが追加されました。

- 今回の更新では、タスクとパイプラインの **describe** コマンドで、オプションまたは必須のワークスペースの存在を表示できるようになりました。

- 今回の更新では、タイムスタンプ付きのログを表示するための **timestamps** フラグが追加されました。

- 今回の更新では、**PipelineRun** に関連付けられた **TaskRun** の削除を無視する新しいフラグ **--ignore-running-pipelinerun** が追加されました。

- 今回の更新では、実験的なコマンドのサポートが追加されました。今回の更新では、試験的なサブコマンドである **sign** と **verify** も **tkn** CLI ツールに追加されました。
- 今回の更新では、ファイルを生成せずに Z シェル (Zsh) 補完機能を使用できるようになりました。
- 今回の更新では、**opc** という新しい CLI ツールが導入されました。今後のリリースで、**tkn** CLI ツールが **opc** に置き換えられることが予想されます。

重要

- 新しい CLI ツール **opc** はテクノロジープレビュー機能です。
- **opc** は **tkn** の代替となり、Red Hat OpenShift Pipelines 固有の追加機能を備えていますが、それらは必ずしも **tkn** に適合するとは限りません。

1.8.1.4. Operator

- 今回の更新により、Pipelines as Code がデフォルトでインストールされます。**-p** フラグを使用して、Pipelines as Code を無効にすることができます。

```
$ oc patch tektonconfig config --type="merge" -p '{"spec": {"platforms": {"openshift":
{"pipelinesAsCode": {"enable": false}}}}}'
```

- 今回の更新により、**TektonConfig** CRD で Pipelines as Code 設定の変更も可能になりました。
- 今回の更新により、開発者パースペクティブを無効にした場合に Operator が開発者コンソール関連のカスタムリソースをインストールしなくなりました。
- 今回の更新には、Bitbucket Server および Bitbucket Cloud の **ClusterTriggerBinding** サポートが含まれており、クラスター全体で **TriggerBinding** を再利用するのに役立ちます。

1.8.1.5. リゾルバー

重要

リゾルバーはテクノロジープレビュー機能です。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- 今回の更新により、**TektonConfig** CRD でパイプラインリゾルバーを設定できるようになりました。パイプラインリゾルバー **enable-bundles-resolver**、**enable-cluster-resolver**、**enable-git-resolver**、**enable-hub-resolver** を、有効または無効にできます。

```
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 でリゾルバー固有の設定も指定できます。たとえば、次のフィールドを **mapstringstring** 形式で定義して、個々のリゾルバーを設定できます。

```
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
...
```

1.8.1.6. Tekton Chains

重要

Tekton Chains はテクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- この更新の前は、Open Container Initiative (OCI) イメージのみが in-toto 出所エージェントの **TaskRun** の出力としてサポートされていました。この更新では、**ARTIFACT_URI** および **ARTIFACT_DIGEST** の接尾辞を使用して、出所メタデータが出力として追加されます。
- この更新の前は、**TaskRun** 構成証明のみがサポートされていました。この更新では、**PipelineRun** 構成証明のサポートも追加されます。
- この更新では、Pod テンプレートから **imgPullSecret** パラメーターを取得するための Tekton Chains のサポートが追加されます。この更新により、サービスアカウントを変更せずに、各パイプライン実行またはタスク実行に基づいてリポジトリー認証を設定できます。

1.8.1.7. Tekton Hub

重要

Tekton Hub はテクノロジープレビュー機能としてのみ提供されます。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- この更新では、管理者は、デフォルトの Tekton Hub データベースを使用する代わりに、Crunchy PostgreSQL などの外部データベースを Tekton Hub で 使用できるようになりました。この更新は、次のアクションを実行するのに役立ちます。
  - Tekton Hub で使用する外部データベースの座標指定。
  - Operator によってデプロイされたデフォルトの Tekton Hub データベースの無効化。
- この更新では、外部 Git リポジトリーから **config.yaml** の依存関係が削除され、完全な設定データが API **ConfigMap** に移動されます。この更新は、管理者が次のアクションを実行するのに役立ちます。
  - Tekton Hub カスタムリソースへの、カテゴリー、カタログ、スコープ、defaultScopes などの設定データの追加。
  - クラスター上の Tekton Hub 設定データの変更。すべての変更は、Operator をアップグレードしても保持されます。
  - Tekton Hub のカタログリストの更新。
  - Tekton Hub のカテゴリーの変更。

注記

設定データを追加しない場合は、Tekton Hub 設定用の API **ConfigMap** のデフォルトデータを使用できます。

1.8.1.8. Pipelines as Code

- この更新では、**Repository** CRD で同時実行制限のサポートが追加され、一度にリポジトリーで実行される **PipelineRuns** の最大数が定義できます。プルリクエストまたはプッシュイベントからの **PipelineRun** は、アルファベット順にキューに入れられます。
- この更新では、リポジトリーの最新パイプライン実行のログを表示するための新しいコマンド **tkn pac logs** が追加されます。
- この更新では、GitHub および GitLab へのプッシュリクエストとプルリクエストのファイルパスにおける高度なイベントマッチングがサポートされています。たとえば、**docs** ディレクトリー内のマークダウンファイルのパスが変更された場合にのみ、Common Expression Language (CEL) を使用してパイプラインを実行できます。

```
...
annotations:
  pipelinesascode.tekton.dev/on-cel-expression: |
```

```
event == "pull_request" && "docs/*.md".pathChanged()
```

- 今回の更新により、アノテーションを使用して、**pipelineRef:** オブジェクトでリモートパイプラインを参照できるようになります。
- 今回の更新により、Pipelines as Code を使用して新しい GitHub リポジトリーを自動設定できるようになります。これにより、namespace が設定され、GitHub リポジトリーの **Repository** CRD が作成されます。
- 今回の更新により、Pipelines as Code は、プロバイダー情報を使用して **PipelineRuns** のメトリクスを生成します。
- この更新では、**tkn-pac** プラグインに次の機能拡張が提供されます。
  - 実行中のパイプラインを正しく検出します。
  - 障害完了時間がない場合に期間表示を修正します。
  - エラースニペットを表示し、**tkn-pac describe** コマンドのエラー正規表現パターンを強調表示します。
  - **use-real-time** スイッチを **tkn-pac ls** および **tkn-pac describe** コマンドに追加します。
  - **tkn-pac** ログのドキュメントをインポートします。
  - **tkn-pac ls** および **tkn-pac describe** コマンドで、**pipelineruntimeout** を失敗として表示します。
  - **--target-pipelinerun** オプションを使用して、特定のパイプライン実行の失敗を表示します。
- 今回の更新により、バージョン管理システム (VCS) コメントまたは GitHub チェックの小さなスニペットの形式で、パイプライン実行のエラーを表示できます。
- 今回の更新により、Pipelines as Code は、タスクが単純な形式である場合にオプションでタスク内のエラーを検出し、それらのタスクを GitHub のアノテーションとして追加できます。この更新は、開発者プレビュー機能の一部です。
- この更新では、次の新しいコマンドが追加されます。
  - **tkn-pac webhook add**: プロジェクトリポジトリー設定に Webhook を追加し、リポジトリーを更新せずに、既存の **k8s Secret** オブジェクトの **webhook.secret** キーを更新します。
  - **tkn-pac webhook update-token**: リポジトリーを更新せずに、既存の **k8s Secret** オブジェクトのプロバイダートークンを更新します。
- この更新により、**tkn-pac create repo** コマンドの機能が強化されます。このコマンドは、GitHub、GitLab、および BitbucketCloud の Webhook を作成および設定し、リポジトリーを作成します。
- この更新により、**tkn-pac describe** コマンドは 50 件の最新イベントが順に表示されます。
- この更新では、**tkn-pac logs** コマンドに **--last** オプションが追加されます。
- この更新により、**tkn-pac resolve** コマンドは、ファイルテンプレートで **git_auth_secret** を検出すると、トークンの入力を求めます。

- この更新により、Pipelines as Code はシークレットをログスニペットから非表示にして、GitHub インターフェイスでシークレットが公開されるのを回避します。

- この更新により、**git_auth_secret** に対して自動的に生成されるシークレットは、**PipelineRun** による所有者参照になります。シークレットは、パイプライン実行の実行後ではなく、**PipelineRun** で消去されます。

- この更新により、**/cancel** コメントを使用したパイプライン実行のキャンセルがサポートされます。

- この更新の前は、GitHub アプリのトークンスコープが定義されておらず、すべてのリポジトリーインストールでトークンが使用されていました。この更新により、次のパラメーターを使用して、GitHub アプリトークンの範囲をターゲットリポジトリーに設定できます。

  - **secret-github-app-token-scoped**: アプリのインストールがアクセスできるすべてのリポジトリーではなく、ターゲットリポジトリーにアプリトークンのスコープを設定します。

  - **secret-github-app-scope-extra-repos**: 追加の所有者またはリポジトリーを使用して、アプリトークンのスコープをカスタマイズします。

- この更新により、GitLab でホストされている独自の Git リポジトリーで Pipelines as Code を使用できるようになります。

- この更新により、namespace の kubernetes イベント形式でパイプライン実行の詳細にアクセスできるようになります。その詳細は、admin namespace へのアクセスを必要とせずにパイプラインエラーをトラブルシューティングするのに役立ちます。

- この更新により、Git プロバイダーを使用した Pipelines as Code での URL 認証がサポートされます。

- この更新により、**pipelines-as-code** config map の設定を使用して、ハブカタログの名前を設定できるようになります。

- この更新により、**max-keep-run** パラメーターの上限とデフォルトの制限を設定できるようになります。

- 今回の更新では、Pipelines as Code にカスタム Secure Sockets Layer (SSL) 証明書を挿入し、カスタム証明書を使用してプロバイダーインスタンスに接続する方法を説明したドキュメントが追加されます。

- この更新により、**PipelineRun** リソース定義にログ URL がアノテーションとして含まれるようになります。たとえば、**tkn-pac describe** コマンドは、**PipelineRun** を記述するときにログリンクを表示します。

- 今回の更新により、**tkn-pac** ログに **PipelineRun** 名ではなくリポジトリー名が表示されるようになります。

1.8.2. 互換性を失わせる変更点

- 今回の更新では、**Conditions** カスタムリソース定義 (CRD) タイプが削除されました。代わりに **WhenExpressions** を使用します。

- 今回の更新では、Pipeline、PipelineRun、Task、Clustertask、TaskRun などの **tekton.dev/v1alpha1** API パイプラインリソースのサポートが削除されました。

- 今回の更新では、**tkn-pac setup** コマンドが削除されました。代わりに、**tkn-pac webhook add** コマンドを使用して、Webhook を既存の Git リポジトリーに再度追加します。また、**tkn-**

pac webhook update-token コマンドを使用して、Git リポジトリー内の既存のシークレットオブジェクトの個人プロバイダーアクセストークンを更新します。

- 今回の更新により、デフォルト設定でパイプラインを実行する namespace は、**pod-security.kubernetes.io/enforce:privileged** ラベルをワークロードに適用しません。

1.8.3. 非推奨および削除された機能

- Red Hat OpenShift Pipelines 1.9.0 リリースでは、**ClusterTasks** が非推奨となり、今後のリリースで削除される予定です。代わりに、**Cluster Resolver** を使用できます。

- Red Hat OpenShift Pipelines 1.9.0 リリースでは、単一の **EventListener** 仕様で **triggers** と **namespaceSelector** フィールドを使用することは推奨されておらず、今後のリリースで削除される予定です。これらのフィールドは、異なる **EventListener** 仕様では正常に使用できます。

- Red Hat OpenShift Pipelines 1.9.0 リリースでは、**tkn pipelinerun describe** コマンドは **PipelineRun** リソースのタイムアウトを表示しません。

- Red Hat OpenShift Pipelines 1.9.0 リリースでは、PipelineResource カスタムリソース (CR) が非推奨になりました。**PipelineResource** CR はテクノロジープレビュー機能であり、**tekton.dev/v1alpha1** API の一部でした。

- Red Hat OpenShift Pipelines 1.9.0 リリースでは、クラスタータスクからのカスタムイメージパラメーターは非推奨になりました。代わりとして、クラスタータスクをコピーして、その中でカスタムイメージを使用できます。

1.8.4. 既知の問題

- Red Hat OpenShift Pipelines Operator をアンインストールすると、**chains-secret** および **chains-config** config map が削除されます。これらにはユーザーデータが含まれているため、削除せずに保持する必要があります。

- Windows でコマンドの **tkn pac** セットを実行すると、**Command finished with error: not supported by Windows.** のエラーメッセージが表示される場合があります。
  回避策: **NO_COLOR** 環境変数を **true** に設定します。

- **tkn pac resolve** コマンドがテンプレート化されたパラメーター値を使用して機能する場合、**tkn pac resolve -f <filename> | oc create -f** コマンドを実行しても、想定どおりの結果が得られない場合があります。
  回避策: この問題を軽減するには、**tkn pac resolve -f <filename> -o tempfile.yaml** コマンドを実行して **tkn pac resolve** の出力を一時ファイルに保存してから、**oc create -f tempfile.yaml** コマンドを実行します。例: **tkn pac resolve -f <filename> -o /tmp/pull-request-resolved.yaml && oc create -f /tmp/pull-request-resolved.yaml**。

1.8.5. 修正された問題

- この更新の前は、空の配列を置き換えた後、元の配列は中のパラメーターを無効にして空の文字列を返していました。今回の更新により、この問題は解決され、元の配列は空として返されます。

- この更新の前は、パイプライン実行のサービスアカウントに重複するシークレットが存在すると、タスク Pod の作成に失敗していました。今回の更新により、この問題が解決され、サービスアカウントに重複するシークレットが存在する場合でもタスク Pod は正常に作成されるようになりました。

- この更新の前は、TaskRun の **spec.StatusMessage** フィールドを見ても、**TaskRun** がユーザーによってキャンセルされたのか、その一部である **PipelineRun** によってキャンセルされたのかを区別できませんでした。今回の更新により、この問題は解決され、ユーザーは TaskRun の **spec.StatusMessage** フィールドを見て、**TaskRun** のステータスを区別できるようになりました。

- この更新の前は、無効なオブジェクトの古いバージョンを削除すると、webhook の検証が削除されていました。今回の更新で、この問題は解決されました。

- 今回の更新の前は、**timeouts.pipeline** パラメーターを **0** に設定すると、**timeouts.tasks** パラメーターまたは **timeouts.finally** パラメーターを設定できませんでした。今回の更新で問題が解決されました。これで、**timeouts.pipeline** パラメーター値を設定するときに、`timeouts.tasks` パラメーターまたは **timeouts.finally** パラメーターのいずれかの値を設定できます。以下に例を示します。

```
yaml
kind: PipelineRun
spec:
  timeouts:
    pipeline: "0"  # No timeout
    tasks: "0h3m0s"
```

- この更新の前は、別のツールが PipelineRun または TaskRun のラベルまたはアノテーションを更新すると、競合状態が発生する可能性がありました。今回の更新により、この問題は解決され、ラベルまたはアノテーションを結合できるようになりました。

- この更新の前は、ログキーにパイプラインコントローラーと同じキーはありませんでした。今回の更新により、この問題は解決され、パイプラインコントローラーのログストリームと一致するようにログキーが更新されました。ログのキーは、ts から timestamp、level から severity、message から msg に変更されました。

- この更新の前は、PipelineRun が不明ステータスで削除された場合、エラーメッセージは生成されませんでした。今回の更新により、この問題は解決され、エラーメッセージが生成されるようになります。

- この更新の前は、**list** や **push** などのバンドルコマンドにアクセスするには、**kubeconfig** ファイルを使用する必要がありました。今回の更新により、この問題は解決され、**kubeconfig** ファイルはバンドルコマンドにアクセスする必要がなくなりました。

- この更新の前は、TaskRun の削除中に親の PipelineRun が実行されていた場合、TaskRun が削除されていました。今回の更新により、この問題は解決され、親 PipelineRun が実行されていても TaskRuns は削除されなくなりました。

- この更新の前は、ユーザーがパイプラインコントローラーで許可されているよりも多くのオブジェクトを含むバンドルのビルドを試みた場合、Tekton CLI はエラーメッセージを表示しませんでした。今回の更新により、この問題は解決され、ユーザーがパイプラインコントローラーで許可されている制限を超える数のオブジェクトを含むバンドルを構築しようとすると、Tekton CLI にエラーメッセージが表示されるようになります。

- この更新の前は、クラスターから namespace が削除されても、operator は **ClusterInterceptor ClusterRoleBinding** サブジェクトから namespace を削除しませんでした。今回の更新により、この問題は解決され、operator は **ClusterInterceptor ClusterRoleBinding** サブジェクトから namespace を削除するようになります。

- この更新の前は、デフォルトの Red Hat OpenShift Pipelines Operator インストールで、**pipelines-scc-rolebinding security context constraint** (SCC) ロールバインディングリ

ソースがクラスターに残りました。今回の更新により、デフォルトの Red Hat OpenShift Pipelines Operator インストールで、**pipelines-scc-rolebinding security context constraint** (SCC) ロールバインディングリソースがクラスターから削除されるようになります。

- この更新の前は、Pipelines as Code は Pipelines as Code **ConfigMap** オブジェクトから更新された値を取得しませんでした。今回の更新により、この問題は修正され、Pipelines as Code **ConfigMap** オブジェクトが新しい変更を検索するようになります。

- この更新の前は、Pipelines as Code コントローラーは **tekton.dev/pipeline** ラベルが更新されるのを待たずに **checkrun id** ラベルを追加して、競合状態を引き起こしていました。今回の更新により、Pipelines as Code コントローラーは **tekton.dev/pipeline** ラベルが更新されるのを待ってから **checkrun id** ラベルを追加するようになりました。これは、競合状態の回避に役立ちます。

- この更新の前は、git リポジトリーに **PipelineRun** がすでに存在する場合、**tkn-pac create repo** コマンドはそれをオーバーライドしませんでした。今回の更新では **tkn-pac create** コマンドが修正され、git リポジトリーに **PipelineRun** が存在する場合はそれをオーバーライドするようになり、問題は解決されました。

- この更新の前は、**tkn pac describe** コマンドはすべてのメッセージの理由を表示しませんでした。今回の更新により、この問題は修正され、**tkn pac describe** コマンドはすべてのメッセージの理由を表示するようになります。

- この更新の前は、アノテーションのユーザーが **refs/head/rel-*** などの正規表現形式を使用して値を指定した場合、プルリクエストは失敗していました。ベースブランチに **refs/heads** がないため、プルリクエストは失敗していました。今回の更新では接頭辞が追加され、一致するかどうかもチェックされます。これで問題が解決し、プルリクエストが成功するようになります。

1.8.6. Red Hat OpenShift Pipelines General Availability 1.9.1 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.9.1 が OpenShift Container Platform 4.11、4.12、および 4.13 で利用できるようになりました。

1.8.7. 修正された問題

- この更新の前は、**tkn pac repo list** コマンドは Microsoft Windows で実行できませんでした。今回の更新で問題が修正され、Microsoft Windows で **tkn pac repo list** コマンドを実行できるようになりました。

- この更新の前は、Pipelines as Code ウォッチャーは設定変更イベントをすべて受信するわけではありませんでした。今回の更新により、Pipelines as Code ウォッチャーが更新され、Pipelines as Code ウォッチャーが設定変更イベントを見逃さなくなりました。

- この更新の前は、Pipelines as Code によって作成された **TaskRuns** や **PipelineRuns** などの Pod は、クラスター内のユーザーによって公開されたカスタム証明書にアクセスできませんでした。今回の更新で問題が修正され、クラスター内で **TaskRuns** または **PipelineRuns** Pod からカスタム証明書にアクセスできるようになりました。

- この更新の前は、FIPS が有効になっているクラスターで、**Trigger** リソースで使用される **tekton-triggers-core-interceptors** コアインターセプターは、Pipelines Operator がバージョン 1.9 にアップグレードされた後に機能しませんでした。今回の更新で問題が解決されました。現在、OpenShift はすべてのコンポーネントに MInTLS 1.2 を使用しています。その結果、**tekton-triggers-core-interceptors** コアインターセプターが TLS バージョン 1.2 に更新され、その機能は正確に実行されるようになりました。

- この更新の前は、内部 OpenShift イメージレジストリーでパイプライン実行を使用する場合、パイプライン実行定義でイメージへの URL をハードコーディングする必要がありました。以下に例を示します。

```
...
  - name: IMAGE_NAME
    value: 'image-registry.openshift-image-
registry.svc:5000/<test_namespace>/<test_pipelinerun>'
...
```

  Pipelines as Code のコンテキストでパイプライン実行を使用する場合、ハードコーディングされた値により、異なるクラスターおよび namespace でパイプライン実行定義をしようできませんでした。

  今回の更新により、namespace とパイプライン実行名の値をハードコーディングする代わりに動的テンプレート変数を使用して、パイプライン実行定義を一般化できます。以下に例を示します。

```
...
  - name: IMAGE_NAME
    value: 'image-registry.openshift-image-registry.svc:5000/{{ target_namespace
}}/$(context.pipelineRun.name)'
...
```

- この更新の前は、Pipelines as Code は同じ GitHub トークンを使用して、デフォルトの GitHub ブランチの同じホストでのみ使用可能なリモートタスクを取得していました。今回の更新で問題が解決されました。Pipelines as Code は同じ GitHub トークンを使用して、任意の GitHub ブランチからリモートタスクを取得するようになりました。

1.8.8. 既知の問題

- Tekton Hub CR で使用される Hub API **ConfigMap** オブジェクト内のフィールドである **CATALOG_REFRESH_INTERVAL** の値が、ユーザーが指定したカスタム値で更新されません。
  回避策: なし。問題 SRVKP-2854 を確認してください。

1.8.9. 互換性を失わせる変更点

- 今回の更新で、OpenShift Container Platform のアップグレードを妨げる OLM のご設定の問題が発生しました。この問題は今後のリリースで修正される予定です。

1.8.10. Red Hat OpenShift Pipelines General Availability 1.9.2 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.9.2 が OpenShift Container Platform 4.11、4.12、および 4.13 で利用できるようになりました。

1.8.11. 修正された問題

- この更新前は、リリースの以前のバージョンで OLM の誤設定の問題が発生しており、OpenShift Container Platform のアップグレードが妨げられていました。今回の更新により、この誤設定の問題が修正されました。

1.8.12. Red Hat OpenShift Pipelines General Availability 1.9.3 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.9.3 が OpenShift Container Platform 4.11、4.12、4.13 に加え、4.10 でも利用できるようになりました。

1.8.13. 修正された問題

- 今回の更新により、大規模パイプラインのパフォーマンスの問題が修正されました。これにより、CPU 使用率は 61%、メモリー使用率は 44% 削減されました。
- この更新前は、**when** 式が原因でタスクが実行されない場合、パイプラインの実行は失敗していました。今回の更新により、パイプライン結果でスキップされたタスクの結果が検証されないようにすることで問題を修正しました。現在は、パイプラインの結果は出力されず、結果の欠落を原因とするパイプライン実行の失敗は発生しません。
- 今回の更新により、**pipelineref.bundle** を **v1beta1** API のバンドルリゾルバーに変換する動作が修正されました。現在は変換機能により、変換後に **kind** フィールドの値が **Pipeline** に設定されます。
- この更新前は、OpenShift Pipelines Operator の問題により、ユーザーは **spec.pipeline.enable-api-fields** フィールドの値を **beta** に設定できませんでした。今回の更新でこの問題が修正されています。現在は、**TektonConfig** カスタムリソースで値を **alpha**、**stable**、**beta** に設定できます。
- この更新前は、Pipelines as Code はクラスターエラーが原因でシークレットを作成できなかった場合、GitHub チェック実行でパブリックな一時トークンが表示されていました。今回の更新でこの問題が修正されています。現在は、シークレットの作成に失敗しても、GitHub チェックインターフェイスにトークンは表示されません。

1.8.14. 既知の問題

- 現在、OpenShift Container Platform Web コンソールでのパイプライン実行の **stop** オプションに関する既知の問題があります。**Actions** ドロップダウンリストの **stop** オプションが期待どおりに機能せず、パイプラインの実行がキャンセルされません。
- 現在、カスタムリソース定義の変換の失敗が原因で発生する、OpenShift Pipelines バージョン 1.9.x へのアップグレードに関する既知の問題があります。
  回避策: OpenShift Pipelines バージョン 1.9.x にアップグレードする前に、Red Hat カスタマーポータルの solution に記載されている手順を実行してください。

1.9. RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.8 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.8 が OpenShift Container Platform 4.10、4.11、および 4.12 で利用できるようになりました。

1.9.1. 新機能

以下では、修正および安定性の面での改善点に加え、OpenShift Pipelines 1.8 の主な新機能について説明します。

1.9.1.1. Pipelines

- 今回の更新により、ARM ハードウェアで実行されている OpenShift Container Platform クラスターで Red Hat OpenShift Pipelines GA 1.8 以降を実行できるようになりました。これには、**ClusterTask** リソースと **tkn** CLI ツールのサポートが含まれます。

重要

ARM ハードウェアでの Red Hat OpenShift Pipelines の実行は、テクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- この更新では、**TaskRun** リソースの **Step** および **Sidecar** オーバーライドが実装されています。

- この更新により、**PipelineRun** ステータス内に最小限の **TaskRun** および **Run** ステータスが追加されます。
  この機能を有効にするには、**TektonConfig** カスタムリソース定義の **パイプライン** セクションで、**enable-api-fields** フィールドを **alpha** に設定する必要があります。

- 今回の更新により、パイプライン実行機能の正常な終了がアルファ機能から安定した機能に昇格されました。その結果、以前に廃止された **PipelineRunCancelled** ステータスは引き続き廃止され、将来のリリースで削除される予定です。
  この機能はデフォルトで使用できるため、**TektonConfig** カスタムリソース定義で **pipeline.enable-api-fields** フィールドを **alpha** に設定する必要がなくなりました。

- 今回の更新により、ワークスペースの名前を使用してパイプラインタスクのワークスペースを指定できるようになりました。この変更により、**Pipeline** および **PipelineTask** リソースのペアに共有ワークスペースを指定できるようになりました。ワークスペースを明示的にマップすることもできます。
  この機能を有効にするには、**TektonConfig** カスタムリソース定義の **パイプライン** セクションで、**enable-api-fields** フィールドを **alpha** に設定する必要があります。

- 今回の更新により、埋め込み仕様のパラメーターが変更なしに伝播されるようになりました。

- 今回の更新により、アノテーションとラベルを使用して、**PipelineRun** リソースによって参照される **Task** リソースの必要なメタデータを指定できるようになりました。これにより、実行コンテキストに依存する **Task** メタデータは、パイプライン実行時に利用できます。

- この更新により、**params** と **results** の値にオブジェクトまたはディクショナリータイプのサポートが追加されました。この変更は後方互換性に影響し、以前のクライアントを新しい Red Hat OpenShift Pipelines バージョンで使用するなど、前方互換性を損なう場合があります。この更新により、**ArrayOrStruct** 構造が変更されます。これは、Go 言語 API をライブラリーとして使用するプロジェクトに影響します。

- この更新により、**SkippingReason** 値が **PipelineRun** ステータスフィールドの **SkippedTasks** フィールドに追加され、特定の PipelineTask がスキップされた理由をユーザーが知ることができるようになりました。

- この更新プログラムは、**Task** オブジェクトから結果を発行するために **array** 型を使用できるアルファ機能をサポートします。結果の型は **string** から **ArrayOrString** に変更されています。たとえば、タスクはタイプを指定してアレイの結果を生成できます。

```
kind: Task
apiVersion: tekton.dev/v1beta1
metadata:
```

```
  name: write-array
  annotations:
    description: |
      A simple task that writes array
spec:
  results:
    - name: array-results
      type: array
      description: The array results
...
```

さらに、タスクスクリプトを実行して、結果をアレイで入力できます。

```
$ echo -n "[\"hello\",\"world\"]" | tee $(results.array-results.path)
```

この機能を有効にするには、**TektonConfig** カスタムリソース定義の **パイプライン** セクションで、**enable-api-fields** フィールドを **alpha** に設定する必要があります。

この機能は進行中であり、TEP-0076 の一部です。

1.9.1.2. トリガー

- この更新により、**EventListener** 仕様の **TriggerGroups** フィールドがアルファ機能から安定した機能に移行します。このフィールドを使用すると、トリガーのグループを選択および実行する前にインターセプターのセットを指定できます。
  この機能はデフォルトで使用できるため、**TektonConfig** カスタムリソース定義で **pipeline.enable-api-fields** フィールドを **alpha** に設定する必要がなくなりました。

- 今回の更新により、**Trigger** リソースは、HTTPS を使用して **ClusterInterceptor** サーバーを実行することにより、エンドツーエンドの安全な接続をサポートします。

1.9.1.3. CLI

- 今回の更新では、**tkn taskrun export** コマンドを使用して、ライブタスクの実行をクラスターから YAML ファイルにエクスポートできます。これを使用して、タスクの実行を別のクラスターにインポートできます。

- 今回の更新により、**tkn pipeline start** コマンドに **-o name** フラグを追加して、開始直後にパイプライン実行の名前を出力できるようになりました。

- 今回の更新により、利用可能なプラグインの一覧が **tkn --help** コマンドの出力に追加されました。

- 今回の更新により、パイプラインの実行またはタスクの実行を削除する際に、**--keep** フラグと **--keep-since** フラグの両方を一緒に使用できるようになりました。

- 今回の更新により、非推奨の **PipelineRunCancelled** 値ではなく、**spec.status** フィールドの値として **Canceled** を使用できるようになりました。

1.9.1.4. Operator

- 今回の更新により、管理者はローカルの Tekton Hub インスタンスを設定して、デフォルトデータベースではなくカスタムデータベースを使用できるようになりました。

- 今回の更新では、クラスター管理者としてローカルの Tekton Hub インスタンスを有効にすると、データベースが定期的に更新され、カタログの変更が Tekton Hub Web コンソールに表示されるようになります。更新の間隔は調整できます。
  以前は、カタログ内のタスクとパイプラインをデータベースに追加するために、そのタスクを手動で実行するか、cron ジョブをセットアップして実行していました。

- 今回の更新で、最小限の設定で Tekton Hub インスタンスをインストールし、実行できるようになりました。これにより、チームと連携して、必要な追加カスタマイズを決定できます。

- 今回の更新で、**GIT_SSL_CAINFO** が **git-clone** タスクに追加され、セキュアなリポジトリーをクローンできるようになりました。

1.9.1.5. Tekton Chains

重要

Tekton Chains はテクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- 今回の更新により、静的トークンではなく OIDC を使用して Vault にログインすることができます。この変更により、Spire は OIDC 認証情報を生成し、信頼されるワークロードのみが vault にログインできます。また、Vault アドレスを環境変数として挿入するのではなく、設定値として渡すこともできます。

- Red Hat OpenShift Pipelines Operator を使用してインストールした場合、設定マップの直接更新はサポートされないため、**openshift-pipelines** namespace の Tekton チェーンの **chain-config** 設定マップは、Red Hat OpenShift Pipelines Operator のアップグレード後に自動的にデフォルトにリセットされます。ただし、今回の更新により、**TektonChain** カスタムリソースを使用して Tekton Chains を設定できるようになりました。この機能により、アップグレード中に上書きされる **chain-config** 設定マップとは異なり、アップグレード後も設定を維持できます。

1.9.1.6. Tekton Hub

重要

Tekton Hub はテクノロジープレビュー機能としてのみ提供されます。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- 今回の更新により、Operator を使用して Tekton Hub の新しいインスタンスをインストールす

ると、Tekton Hub ログインがデフォルトで無効になります。ログインおよび評価機能を有効にするには、Tekton Hub のインストール時に Hub API シークレットを作成する必要があります。

注記

Red Hat OpenShift Pipelines 1.7 では Tekton Hub ログインがデフォルトで有効になっているため、Operator をアップグレードすると、Red Hat OpenShift Pipelines 1.8 ではログインがデフォルトで有効になります。このログインを無効にするには、OpenShift Pipelines 1.7.x -→ 1.8.x からアップグレードした後の Tekton Hub ログインの無効化 を参照してください。

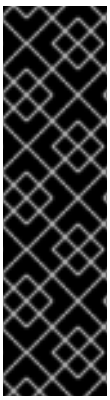
- 今回の更新により、管理者はローカルの Tekton Hub インスタンスを設定して、デフォルトデータベースではなくカスタム PostgreSQL 13 データベースを使用できるようになりました。これを行うには、**tekton-hub-db** という名前の **Secret** リソースを作成します。以下に例を示します。

```
apiVersion: v1
kind: Secret
metadata:
  name: tekton-hub-db
  labels:
    app: tekton-hub-db
type: Opaque
stringData:
  POSTGRES_HOST: <hostname>
  POSTGRES_DB: <database_name>
  POSTGRES_USER: <username>
  POSTGRES_PASSWORD: <password>
  POSTGRES_PORT: <listening_port_number>
```

- 今回の更新により、カタログからデータベースにリソースを追加するために Tekton Hub Web コンソールにログインする必要がなくなりました。現在、これらのリソースは、Tekton Hub API が初めて実行を開始したときに自動的に追加されます。

- この更新プログラムは、カタログ更新 API ジョブを呼び出すことにより、30 分ごとにカタログを自動的に更新します。この間隔は user-configurable です。

1.9.1.7. Pipelines as Code



重要

コードとしてのパイプライン (PAC) は、テクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- 今回の更新により、開発者は、重複したリポジトリーを Pipelines as Code run に追加しようとすると、**tkn-pac** CLI ツールから通知を受け取ります。**tkn pac create repository** を入力する

場合、各リポジトリーには一意の URL が必要です。この通知は、ハイジャックエクスプロイトの防止にも役立ちます。

- 今回の更新により、開発者は新しい **tkn-pac setup cli** コマンドを使用して、Webhook メカニズムを使用して Git リポジトリーを Pipelines as Code に追加できるようになりました。このように、GitHub アプリを使用できない場合でも、Pipelines as Code を使用できます。この機能には、GitHub、GitLab、BitBucket のリポジトリーのサポートが含まれます。

- 今回の更新により、Pipelines as Code は、次のような機能を備えた GitLab 統合をサポートします。

  - プロジェクトまたはグループの ACL (アクセス制御リスト)

  - 許可されたユーザーからの **/OK-to-test** サポート

  - **/retest** サポート。

- 今回の更新により、Common Expression Language (CEL) を使用して高度なパイプラインフィルタリングを実行できます。CEL では、**PipelineRun** リソースのアノテーションを使用して、パイプラインの実行をさまざまな Git プロバイダーイベントと一致させることができます。以下に例を示します。

```
...
annotations:
  pipelinesascode.tekton.dev/on-cel-expression: |
   event == "pull_request" && target_branch == "main" && source_branch == "wip"
```

- 以前は、開発者は、プルリクエストなどの Git イベントごとに **.tekton** ディレクトリーで 1 つのパイプラインしか実行できませんでした。今回の更新により、**.tekton** ディレクトリーに複数のパイプラインを実行できるようになりました。Web コンソールは、実行のステータスとレポートを表示します。パイプラインは並行して動作し、Git プロバイダーインターフェイスに報告します。

- 今回の更新により、プルリクエストで **/test** または **/retest** にコメントすることで、パイプラインの実行をテストまたは再テストできるようになりました。名前でパイプライン実行を指定することもできます。たとえば、**/test <pipelinerun_name>** または **/retest <pipelinerun-name>** を入力できます。

- 今回の更新により、新しい **tkn-pac delete repository** コマンドを使用して、リポジトリーカスタムリソースとそれに関連付けられたシークレットを削除できるようになりました。

1.9.2. 互換性を失わせる変更点

- この更新により、**TaskRun** および **PipelineRun** リソースのデフォルトのメトリクスレベルが次の値に変更されます。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: config-observability
  namespace: tekton-pipelines
  labels:
    app.kubernetes.io/instance: default
    app.kubernetes.io/part-of: tekton-pipelines
data:
  _example: |
```

```
...
  metrics.taskrun.level: "task"
  metrics.taskrun.duration-type: "histogram"
  metrics.pipelinerun.level: "pipeline"
  metrics.pipelinerun.duration-type: "histogram"
```

- 今回の更新により、アノテーションまたはラベルが **Pipeline** および **PipelineRun** リソースの両方にある場合、**Run** タイプの値が優先されます。アノテーションまたはラベルが **Task** および **TaskRun** リソースにある場合も同様です。

- Red Hat OpenShift Pipelines 1.8 では、以前に非推奨の **PipelineRun.Spec.ServiceAccountNames** フィールドが削除されました。代わりに **PipelineRun.Spec.TaskRunSpecs** フィールドを使用してください。

- Red Hat OpenShift Pipelines 1.8 では、以前に非推奨の **TaskRun.Status.ResourceResults.ResourceRef** フィールドが削除されました。代わりに **TaskRun.Status.ResourceResults.ResourceName** フィールドを使用してください。

- Red Hat OpenShift Pipelines 1.8 では、以前に非推奨となった **Conditions** リソースタイプが削除されました。**Conditions** リソースを、これが含まれる **Pipeline** リソース定義から削除します。代わりに **PipelineRun** 定義で **when** 式を使用してください。

- Tekton Chains では、**tekton-provenance** 形式は本リリースで削除されました。代わりに、**TektonChain** カスタムリソースで **"artifacts.taskrun.format": "in-toto"** を設定して、**in-toto** 形式を使用します。

- Pipeline が Code 0.5.x として同梱される Red Hat OpenShift Pipelines 1.7.x。現在の更新には、Pipeline が Code 0.10.x として同梱されます。この変更により、新規コントローラーの **openshift-pipelines** namespace に新規ルートが作成されます。このルートは、Pipeline を Code として使用する GitHub Apps または Webhook で更新する必要があります。ルートを取得するには、以下のコマンドを使用します。

```
$ oc get route -n openshift-pipelines pipelines-as-code-controller \
  --template='https://{{ .spec.host }}'
```

- 今回の更新で、コードとしてのパイプラインは、**Repository** カスタムリソース定義 (CRD) のデフォルトの秘密鍵の名前を変更します。CRD で、**token** を **provider.token** に置き換え、**secret** を **webhook.secret** に置き換えます。

- 今回の更新で、Pipelines as Code は、特別なテンプレート変数を、プライベートリポジトリーの複数のパイプライン実行をサポートするものに置き換えます。パイプラインの実行で、**secret: pac-git-basic-auth-{{repo_owner}}-{{repo_name}}** を **secret: {{ git_auth_secret }}** に置き換えます。

- 今回の更新により、コードとしての Pipeline が **tkn-pac** CLI ツールで以下のコマンドを更新するようになりました。

  - **tkn pac repository create** を **tkn pac create repository** に置き換えます。

  - **tkn pac repository delete** を **tkn pac delete repository** に置き換えます。

  - **tkn pac repository list** を **tkn pac list** に置き換えます。

1.9.3. 非推奨および削除された機能

- 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** チャネルは削除されています。Operator をインストールしてアップグレードするには、適切な **pipelines-<version>** チャネル、または最新の安定バージョンの **latest** チャネルを使用します。たとえば、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** チャネルを使用します。

注記

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。

- Red Hat OpenShift Pipelines GA 1.6 で廃止された **tekton.dev/v1alpha1** API バージョンのサポートは、今後の Red Hat OpenShift Pipelines GA 1.9 リリースで削除される予定です。この変更は、**TaskRun**、**PipelineRun**、**Task**、**Pipeline**、および同様の **tekton.dev/v1alpha1** リソースを含むパイプラインコンポーネントに影響します。別の方法として、 Migrating From Tekton v1alpha1 to Tekton v1beta1 で説明されているように、既存のリソースを **apiVersion: tekton.dev/v1beta1** を使用するように更新します。

  tekton.dev/v1alpha1 API バージョンのバグ修正とサポートは、現在の GA 1.8 ライフサイクルの終了までのみ提供されます。

重要

Tekton Operator の場合、**operator.tekton.dev/v1alpha1** API バージョンは **非**推奨ではありません。この値を変更する必要はありません。

- Red Hat OpenShift Pipelines 1.8 では、**PipelineResource** カスタムリソース (CR) が利用可能ですが、サポートされなくなりました。**PipelineResource** CR は Tech Preview 機能であり、**tekton.dev/v1alpha1** API の一部であり、廃止予定であり、今後の Red Hat OpenShift Pipelines GA 1.9 リリースで削除される予定でした。

- Red Hat OpenShift Pipelines 1.8 では、**Condition** カスタムリソース (CR) が削除されています。**Condition** CR は **tekton.dev/v1alpha1** API の一部でしたが、これは非推奨であり、今後の Red Hat OpenShift Pipelines GA 1.9 リリースで削除される予定です。

- Red Hat OpenShift Pipelines 1.8 では、**gsutil** の **gcr.io** イメージが削除されました。この削除により、このイメージに依存する **Pipeline** リソースを含むクラスターが壊れる可能性があります。バグ修正とサポートは、Red Hat OpenShift Pipelines 1.7 ライフサイクルが終了するまでのみ提供されます。

- Red Hat OpenShift Pipelines 1.8 では、**PipelineRun.Status.TaskRuns** および **PipelineRun.Status.Runs** フィールドは非推奨となり、将来のリリースで削除される予定です。TEP-0100: PipelineRuns に埋め込まれた TaskRuns と Runs Status を参照してください。

- Red Hat OpenShift Pipelines 1.8 では、**pipelineRunCancelled** 状態は非推奨となり、今後のリリースで削除される予定です。**PipelineRun** オブジェクトの正常な終了は、アルファ機能から安定した機能にプロモートされるようになりました。(TEP-0058: パイプライン実行の正常な終了 を参照してください。) 別の方法として、**Cancelled** 状態を使用できます。これは **pipelineRunCancelled** 状態を置き換えます。
  Pipeline および **Task** リソースを変更する必要はありません。パイプラインの実行をキャンセルするツールがある場合は、次のリリースでツールを更新する必要があります。この変更は、CLI、IDE 拡張機能などのツールにも影響を与え、新しい **PipelineRun** ステータスをサポートするようにします。

この機能はデフォルトで使用できるため、**TektonConfig** カスタムリソース定義で **pipeline.enable-api-fields** フィールドを **alpha** に設定する必要がなくなりました。

- Red Hat OpenShift Pipelines 1.8 では、**PipelineRun** の **timeout** フィールドが非推奨になりました。代わりに、**PipelineRun.Timeouts** フィールドを使用してください。これは現在、アルファ機能から安定した機能に昇格しています。
  この機能はデフォルトで使用できるため、**TektonConfig** カスタムリソース定義で **pipeline.enable-api-fields** フィールドを **alpha** に設定する必要がなくなりました。

- Red Hat OpenShift Pipelines 1.8 では、**init** コンテナーは **LimitRange** オブジェクトのデフォルトのリクエスト計算から省略されています。

1.9.4. 既知の問題

- **s2i-nodejs** パイプラインは、**nodejs:14-ubi8-minimal** イメージストリームを使用して、source-to-image (S2I) ビルドを実行できません。そのイメージストリームを使用すると **error building at STEP "RUN /usr/libexec/s2i/assemble": exit status 127** メッセージが生成されます。
  回避策: **nodejs:14-ubi8-minimal** イメージストリームではなく、**nodejs:14-ubi8** を使用します。

- Maven および Jib Maven クラスタータスクを実行する場合には、デフォルトのコンテナーイメージは Intel(x86) アーキテクチャーでのみサポートされます。したがって、タスクは ARM、IBM Power Systems (ppc64le)、IBM Z、および LinuxONE (s390x) クラスターで失敗します。
  回避策: **MAVEN_IMAGE** パラメーター値を **maven:3.6.3-adoptopenjdk-11** に設定して、カスタムイメージを指定します。

  ヒント

  tkn hub を使用して、ARM、IBM Power Systems (ppc64le)、IBM Z、および LinuxONE (s390x) に Tekton カタログに基づくタスクをインストールする前に、これらのプラットフォームでタスクを実行できるかどうかを確認してください。**ppc64le** および **s390x** がタスク情報の Platforms セクションに一覧表示されているかどうかを確認するには、**tkn hub info task <name>** コマンドを実行します。

- ARM、IBM Power Systems、IBM Z、および LinuxONE では、**s2i-dotnet** クラスタータスクはサポートされていません。

- 暗黙的なパラメーターマッピングは、最上位の **Pipeline** または **PipelineRun** 定義から **taskRef** タスクにパラメーターを誤って渡します。マッピングは、トップレベルのリソースからインライン **taskSpec** 仕様のタスクにのみ行う必要があります。この問題は、**TektonConfig** カスタムリソース定義の **pipeline** セクションで **enable-api-fields** フィールドを **alpha** に設定することにより、この機能が有効になっているクラスターにのみ影響します。

1.9.5. 修正された問題

- この更新の前は、Web コンソールの開発者ビューでのパイプライン実行のメトリクスは不完全で古くなっていました。今回の更新で問題が修正され、指標が正しくなりました。

- この更新の前は、パイプラインに失敗した 2 つの並列タスクがあり、そのうちの 1 つが **retries=2** であった場合、最後のタスクは実行されず、パイプラインはタイムアウトして実行に失敗しました。たとえば、**pipelines-operator-subscription** タスクが次のエラーメッセージで断続的に失敗しました。**Unable to connect to the server: EOF**。今回の更新で、最終タスクが常に実行されるように問題が修正されました。

- この更新の前は、タスクの実行が失敗したためにパイプラインの実行が停止した場合、他のタスクの実行が再試行を完了しない可能性がありました。その結果、**finally** タスクがスケジュールされず、パイプラインがハングしました。 今回の更新で問題が解決されました。**TaskRuns** および **Run** オブジェクトは、パイプラインの実行が停止したときに (正常な停止によっても) 再試行できるため、パイプラインの実行を完了できます。

- この更新により、**TaskRun** オブジェクトが存在する namespace に 1 つ以上の **LimitRange** オブジェクトが存在する場合のリソース要件の計算方法が変更されます。スケジューラーは、**step** コンテナーを考慮し、**LimitRange** オブジェクトからの要求を因数分解するときに、サイドカーコンテナーなどの他のすべてのアプリコンテナーを除外するようになりました。

- この更新の前は、特定の条件下で、フラグパッケージが二重ダッシュフラグターミネータ **--** の直後のサブコマンドを誤って解析する場合がありました。その場合、実際のコマンドではなく、エントリーポイントサブコマンドが実行されました。今回の更新では、このフラグ解析の問題が修正され、エントリーポイントが正しいコマンドを実行できるようになりました。

- この更新の前は、イメージのプルが失敗した場合、またはそのプルステータスが不完全であった場合、コントローラーが複数のパニックを生成する可能性がありました。この更新では、**status.TaskSpec** 値ではなく **step.ImageID** 値をチェックすることで問題が修正されています。

- この更新の前は、スケジュールされていないカスタムタスクを含むパイプラインの実行をキャンセルすると、**PipelineRunCouldntCancel** エラーが発生していました。今回の更新でこの問題が修正されています。エラーを生成することなく、スケジュールされていないカスタムタスクを含むパイプラインの実行をキャンセルできます。

- この更新の前は、**$params["<NAME>"]** または **$params['<NAME>']** の **<NAME>** にドット文字 (**.**) が含まれている場合、ドットの右側の名前のどの部分も含まれていませんでした。たとえば、**$params["org.ipsum.lorem"]** から、**org** のみが抽出されました。
  今回の更新で問題が修正され、**$params** が完全な値を取得するようになりました。たとえば、**$params["org.ipsum.lorem"]** と **$params['org.ipsum.lorem']** は有効で、**<NAME>** の値全体である **org.ipsum.lorem** が抽出されます。

  <NAME> が一重引用符または二重引用符で囲まれていない場合にも、エラーが出力されます。たとえば、**$params.org.ipsum.lorem** は有効ではなく、検証エラーが発生します。

- 今回の更新により、**Trigger** リソースはカスタムインターセプターをサポートし、カスタムインターセプターサービスのポートが **ClusterInterceptor** 定義ファイルのポートと同じになるようにします。

- この更新の前は、Tekton Chains および Operator コンポーネントの **tkn version** コマンドが正しく機能しませんでした。今回の更新で問題が修正され、コマンドが正しく機能し、それらのコンポーネントのバージョン情報が返されるようになりました。

- この更新の前に、**tkn pr delete --ignore-running** コマンドを実行し、パイプラインの実行に **status.condition** 値がない場合、**tkn** CLI ツールは null-pointer エラー (NPE) を生成しました。今回の更新で問題が修正され、CLI ツールがエラーを生成し、実行中のパイプライン実行を正しく無視するようになりました。

- この更新の前に、**tkn pr delete --keep <value>** または **tkn tr delete --keep <value>** コマンドを使用し、パイプラインの実行またはタスクの実行の数が値よりも少ない場合、コマンドは予想通りのエラー。今回の更新で問題が修正され、これらの条件下でコマンドが正しくエラーを返すようになりました。

- この更新の前に、**-p** または **-t** フラグと **--ignore-running** フラグを指定して **tkn pr delete** または **tkn tr delete** コマンドを使用した場合、コマンドは実行中または保留中のリソースを誤って削除しました。今回の更新で問題が修正され、これらのコマンドが実行中または保留中のリ

ソースを正しく無視するようになりました。

- 今回の更新により、**TektonChain** カスタムリソースを使用して Tekton Chains を設定できるようになりました。この機能により、アップグレード中に上書きされる **chain-config** 設定マップとは異なり、アップグレード後も設定を維持できます。

- 今回の更新により、**buildah** および **s2i** クラスタータスクを除き、**ClusterTask** リソースはデフォルトで root として実行されなくなりました。

- 今回の更新前は、最初の引数として **init** を使用し、その後に 2 つ以上の引数を使用すると、Red Hat OpenShift Pipelines 1.7.1 でのタスクが失敗していました。今回の更新により、フラグが正しく解析され、タスクが正常に実行されるようになりました。

- 今回の更新以前は、無効なロールバインディングにより、OpenShift Container Platform 4.9 および 4.10 への Red Hat OpenShift Pipelines Operator のインストールは、以下のエラーメッセージと共に失敗していました。

  ```
  error updating rolebinding openshift-operators-prometheus-k8s-read-binding:
  RoleBinding.rbac.authorization.k8s.io
  "openshift-operators-prometheus-k8s-read-binding" is invalid:
  roleRef: Invalid value: rbac.RoleRef{APIGroup:"rbac.authorization.k8s.io", Kind:"Role",
  Name:"openshift-operator-read"}: cannot change roleRef
  ```

  今回の更新で問題が修正され、障害が発生しなくなりました。

- 以前は、Red Hat OpenShift Pipelines Operator をアップグレードすると **pipeline** サービスアカウントが再作成され、サービスアカウントにリンクされたシークレットが失われていました。今回の更新でこの問題が修正されています。アップグレード中に、Operator は **pipeline** サービスアカウントを再作成しなくなりました。その結果、**pipeline** サービスアカウントにアタッチされたシークレットはアップグレード後も保持され、リソース (タスクとパイプライン) は引き続き正しく機能します。

- 今回の更新により、**TektonConfig** カスタムリソース (CR) でインフラストラクチャーノード設定が設定されている場合、Pipelines as Code Pod はインフラストラクチャーノードで実行されます。

- 以前は、リソースプルーナーを使用して、各 namespace Operator が個別のコンテナーで実行されるコマンドを作成していました。この設計は、namespaces の数が多いクラスターで大量のリソースを消費しました。たとえば、1 つのコマンドを実行するために、1000 個の namespace を持つクラスターは、Pod 内に 1000 個のコンテナーを生成しました。
  今回の更新でこの問題が修正されています。すべてのコマンドがループ内の 1 つのコンテナーで実行されるように、namespace ベースの設定をジョブに渡します。

- Tekton Chains では、**signing-secrets** と呼ばれるシークレットを定義して、タスクとイメージの署名に使用されるキーを保持する必要があります。ただし、この更新の前に、Red Hat OpenShift Pipelines Operator を更新すると、このシークレットがリセットまたは上書きされ、キーが失われました。今回の更新でこの問題が修正されています。これで、Operator を介して Tekton Chains をインストールした後にシークレットが設定された場合、シークレットは保持され、アップグレードによって上書きされなくなりました。

- 今回の更新以前は、すべての S2I ビルドタスクが以下の様なエラーメッセージと共に失敗していました。

  ```
  Error: error writing "0 0 4294967295\n" to /proc/22/uid_map: write /proc/22/uid_map:
  operation not permitted
  time="2022-03-04T09:47:57Z" level=error msg="error writing \"0 0 4294967295\\n\" to
  ```

```
/proc/22/uid_map: write /proc/22/uid_map: operation not permitted"
time="2022-03-04T09:47:57Z" level=error msg="(unable to determine exit status)"
```

今回の更新により、**pipelines-scc** セキュリティーコンテキスト制約 (SCC) は、**Buildah** および **S2I** クラスタータスクに必要な **SETFCAP** 機能と互換性が確保されています。その結果、**Buildah** および **S2I** ビルドタスクを正常に実行できます。

さまざまな言語やフレームワークで書かれたアプリケーションに対して **Buildah** クラスタータスクおよび **S2I** ビルドタスクを正常に実行するには、**build** や **push** などの適切な **steps** オブジェクトに以下のスニペットを追加します。

```
securityContext:
  capabilities:
    add: ["SETFCAP"]
```

- 今回の更新前は、Red Hat OpenShift Pipelines Operator のインストールに予想以上に時間がかかりました。この更新プログラムは、インストールプロセスを高速化するために一部の設定を最適化します。
- 今回の更新により、Buildah および S2I クラスタータスクの手順が以前のバージョンよりも少なくなりました。一部のステップは 1 つのステップに結合されているため、**ResourceQuota** および **LimitRange** オブジェクトでより適切に機能し、必要以上のリソースを必要としません。
- この更新により、クラスタータスクの Buildah、**tkn** CLI ツール、および **skopeo** CLI ツールのバージョンがアップグレードされます。
- 今回の更新前は、いずれかの namespace が **Terminating** 状態の場合、RBAC リソースの作成時に Operator が失敗していました。今回の更新により、Operator は **Terminating** 状態の namespace を無視し、RBAC リソースを作成します。
- この更新の前は、予想どおり、prune cronjobs の Pod がインフラストラクチャーノードでスケジュールされていませんでした。代わりに、それらはワーカーノードでスケジュールされているか、まったくスケジュールされていませんでした。今回の更新により、**TektonConfig** カスタムリソース (CR) で設定されている場合、これらのタイプの Pod をインフラストラクチャーノードでスケジュールできるようになりました。

1.9.6. Red Hat OpenShift Pipelines General Availability (GA) 1.8.1 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.8.1 が OpenShift Container Platform 4.10、4.11、および 4.12 で利用できるようになりました。

1.9.6.1. 既知の問題

- デフォルトでは、セキュリティーを強化するために、コンテナーのアクセス権が制限されています。制限付きのアクセス許可は、Red Hat OpenShift Pipelines Operator のすべてのコントローラー Pod と、一部のクラスタータスクに適用されます。アクセス権が制限されているため、特定の設定では **git-clone** クラスタータスクが失敗します。
  回避策: なし。問題 SRVKP-2634 を確認してください。
- インストーラーセットが失敗した状態の場合、**TektonConfig** カスタムリソースのステータスが **False** ではなく **True** として誤表示されます。

  例: 失敗したインストーラセット

```
$ oc get tektoninstallerset
```

```
NAME                                READY   REASON
addon-clustertasks-nx5xz            False   Error
addon-communityclustertasks-cfb2p   True
addon-consolecli-ftrb8              True
addon-openshift-67dj2               True
addon-pac-cf7pz                     True
addon-pipelines-fvllm               True
addon-triggers-b2wtt                True
addon-versioned-clustertasks-1-8-hqhnw   False   Error
pipeline-w75ww                      True
postpipeline-lrs22                  True
prepipeline-ldlhw                   True
rhosp-rbac-4dmgb                    True
trigger-hfg64                       True
validating-mutating-webhoook-28rf7  True
```

例: 正しくない TektonConfig ステータス

```
$ oc get tektonconfig config
NAME     VERSION   READY   REASON
config   1.8.1     True
```

1.9.6.2. 修正された問題

- この更新まで、プルーナーは実行中のパイプラインのタスク実行を削除し、警告 **some tasks were indicated completed without ancestors being done** を表示していました。今回の更新により、プルーナーは、実行中のパイプラインの一部であるタスク実行を保持します。

- この更新まで、**pipeline-1.8** が Red Hat OpenShift Pipelines Operator 1.8.x をインストールするためのデフォルトのチャネルでした。今回の更新により、**latest** がデフォルトのチャネルになりました。

- この更新まで、コードとしてのパイプラインのコントローラー Pod は、ユーザーによって公開された証明書にアクセスできませんでした。今回の更新により、コードとしてのパイプラインは、自己署名証明書またはカスタム証明書によって保護されたルートと Git リポジトリーにアクセスできるようになりました。

- この更新まで、Red Hat OpenShift Pipelines 1.7.2 から 1.8.0 にアップグレードすると、タスクが RBAC エラーで失敗していました。今回の更新により、タスクは RBAC エラーなしで正常に実行されます。

- この更新まで、**tkn** CLI ツールを使用して、**array** 型の **result** オブジェクトを含むタスク実行とパイプライン実行を削除できませんでした。今回の更新により、**tkn** CLI ツールを使用して、**array** 型の **result** オブジェクトを含むタスク実行とパイプライン実行を削除できます。

- この更新まで、パイプライン仕様に **array** 型の **ENV_VARS** パラメーターを持つタスクが含まれていた場合、パイプラインの実行は **invalid input params for task func-buildpacks: param types don't match the user-specified type: [ENV_VARS]** エラーで失敗していました。今回の更新により、そのようなパイプラインおよびタスク仕様でのパイプライン実行は失敗しなくなりました。

- この更新まで、クラスター管理者は、コンテナーレジストリーにアクセスするための **Buildah** クラスタータスクに **config.json** ファイルを提供できませんでした。今回の更新により、クラスター管理者は、**dockerconfig** ワークスペースを使用して、**Buildah** クラスタータスクに **config.json** ファイルを提供できるようになりました。

1.9.7. Red Hat OpenShift Pipelines General Availability (GA) 1.8.2 のリリースノート

今回の更新により、Red Hat OpenShift Pipelines General Availability (GA) 1.8.2 が OpenShift Container Platform 4.10、4.11、および 4.12 で利用できるようになりました。

1.9.7.1. 修正された問題

- この更新の前は、SSH キーを使用してリポジトリーのクローンを作成すると、**git-clone** タスクが失敗していました。今回の更新により、**git-init** タスクでの root 以外のユーザーのロールが削除され、SSH プログラムは **$HOME/.ssh/** ディレクトリーで正しいキーを検索します。

1.10. RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.7 のリリースノート

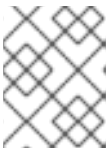
Red Hat OpenShift Pipelines General Availability (GA) 1.7 が OpenShift Container Platform 4.9、4.10、および 4.11 で利用可能になりました。

1.10.1. 新機能

以下では、修正および安定性の面での改善点に加え、OpenShift Pipelines 1.7 の主な新機能について説明します。

1.10.1.1. Pipelines

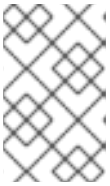
- 今回の更新では、**pipelines-<version>** が Red Hat OpenShift Pipelines Operator をインストールするためのデフォルトのチャネルです。たとえば、OpenShift Pipelines Operator バージョン **1.7** をインストールするデフォルトのチャネルは、**pipelines-1.7** です。クラスター管理者は、**latest** チャネルを使用して、Operator の最新の安定バージョンをインストールすることもできます。



注記

preview チャネルと **stable** チャネルは廃止され、将来のリリースで削除される予定です。

- ユーザー namespace でコマンドを実行すると、コンテナーは **root** (ユーザー ID **0**) として実行されますが、ホストに対するユーザー特権があります。この更新では、ユーザー namespace で pod を実行するには、CRI-O が期待するアノテーションを渡す必要があります。
  - すべてのユーザーにこれらのアノテーションを追加するには、**oc edit clustertask buildah** コマンドを実行し、**buildah** クラスタータスクを編集します。
  - 特定の namespace にアノテーションを追加するには、クラスタータスクをタスクとしてその namespace にエクスポートします。
- この更新の前は、特定の条件が満たされない場合、**when** 式は **Task** オブジェクトとその依存タスクをスキップしていました。今回の更新により、**when** 式のスコープを設定して、従属タスクではなく、**Task** オブジェクトのみを保護できるようになりました。この更新を有効にするには、**TektonConfig** CRD で **scope-when-expressions-to-task** フラグを **true** に設定します。



注記

scope-when-expressions-to-task フラグは非推奨であり、将来のリリースで削除される予定です。OpenShift Pipelines のベストプラクティスとして、保護された **Task** のみをスコープとする **when** 式を使用します。

- この更新では、タスク内のワークスペースの **subPath** フィールドで変数置換を使用できます。
- 今回の更新では、一重引用符または二重引用符を含む角かっこ表記を使用して、パラメーターと結果を参照できます。この更新以前は、ドット表記しか使用できませんでした。たとえば、次は同等になりました。
    - **$(param.myparam)**、**$(param['myparam'])**、および **$(param["myparam"])**。
      一重引用符または二重引用符を使用して、**"."** などの問題のある文字を含むパラメーター名を囲むことができます。たとえば、**$(param['my.param'])** と **$(param["my.param"])**。
- この更新により、**enable-api-fields** フラグを有効にせずに、タスク定義にステップの **onError** パラメーターを含めることができます。

1.10.1.2. トリガー

- この更新により、**feature-flag-triggers** 設定マップに新しいフィールド **labels-exclusion-pattern** が追加されました。このフィールドの値を正規表現 (regex) パターンに設定できます。コントローラーは、正規表現パターンに一致するラベルを、イベントリスナーからイベントリスナー用に作成されたリソースへの伝播から除外します。
- この更新により、**TriggerGroups** フィールドが **EventListener** 仕様に追加されました。このフィールドを使用すると、トリガーのグループを選択して実行する前に実行するインターセプターのセットを指定できます。この機能を有効にするには、**TektonConfig** カスタムリソース定義の **パイプライン** セクションで、**enable-api-fields** フィールドを **alpha** に設定する必要があります。
- この更新により、**Trigger** リソースは、**TriggerTemplate** テンプレートによって定義されたカスタム実行をサポートします。
- この更新により、トリガーは **EventListener** Pod からの Kubernetes イベントの生成をサポートします。
- この更新により、次のオブジェクトのカウントメトリクスが使用可能になります: **ClusterInteceptor**、**EventListener**、**TriggerTemplate**、**ClusterTriggerBinding**、および **TriggerBinding**。
- この更新により、**ServicePort** 仕様が Kubernetes リソースに追加されます。この仕様を使用して、イベントリスナーサービスを公開するポートを変更できます。デフォルトのポートは **8080** です。
- この更新では、**EventListener** 仕様の **targetURI** フィールドを使用して、トリガー処理中にクラウドイベントを送信できます。この機能を有効にするには、**TektonConfig** カスタムリソース定義の **パイプライン** セクションで、**enable-api-fields** フィールドを **alpha** に設定する必要があります。
- この更新により、**tekton-triggers-eventlistener-roles** オブジェクトには、既存の **create** 動詞に加えて、**patch** 動詞が含まれるようになりました。
- この更新により、**securityContext.runAsUser** パラメーターがイベントリスナーのデプロイメントから削除されます。

1.10.1.3. CLI

- この更新では、**tkn [pipeline | pipelinerun] export** コマンドは、パイプラインまたはパイプライン実行を YAML ファイルとしてエクスポートします。以下に例を示します。
  - **openshift-pipelines** namespace に **test_pipeline** という名前のパイプラインをエクスポートします。

```
$ tkn pipeline export test_pipeline -n openshift-pipelines
```

  - **openshift-pipelines** namespace に **test_pipeline_run** という名前のパイプランランをエクスポートします。

```
$ tkn pipelinerun export test_pipeline_run -n openshift-pipelines
```

- この更新により、**--grace** オプションが **tkn pipelinerun cancel** に追加されます。**--grace** オプションを使用して、パイプラインの実行を強制的に終了するのではなく、適切に終了します。この機能を有効にするには、**TektonConfig** カスタムリソース定義の **パイプライン** セクションで、**enable-api-fields** フィールドを **alpha** に設定する必要があります。
- この更新により、Operator バージョンと Chains バージョンが **tkn version** コマンドの出力に追加されます。

重要

Tekton Chains はテクノロジープレビュー機能です。

- この更新により、パイプラインの実行をキャンセルすると、**tkn pipelinerun describe** コマンドはキャンセルされたすべてのタスクの実行を表示します。この修正以前は、1 つのタスク実行のみが表示されていました。
- この更新により、**tkn [t | p | ct] start** コマンドのスキップを **--skip-optional-workspace** フラグで実行したときに、オプションのワークスペースの要求仕様を省略できるようになりました。インタラクティブモードで実行している場合はスキップすることもできます。
- この更新では、**tkn chains** コマンドを使用して Tekton Chains を管理できます。**--chains-namespace** オプションを使用し Tekton Chains をインストールする namespace を指定することもできます。

重要

Tekton Chains はテクノロジープレビュー機能です。

1.10.1.4. Operator

- この更新では、Red Hat OpenShift Pipelines Operator を使用して、Tekton Hub および Tekton Chains をインストールおよびデプロイできます。

重要

Tekton Chains とクラスターへの Tekton Hub のデプロイメントは、テクノロジープレビュー機能です。

- この更新により、アドオンオプションとして Pipelines as Code (PAC) を見つけて使用できるようになります。

重要

Pipelines as Code は、テクノロジープレビュー機能です。

- この更新により、**communityClusterTasks** パラメーターを **false** に設定することにより、コミュニティークラスタータスクのインストールを無効にできるようになりました。以下に例を示します。

```
...
spec:
  profile: all
  targetNamespace: openshift-pipelines
  addon:
    params:
    - name: clusterTasks
      value: "true"
    - name: pipelineTemplates
      value: "true"
    - name: communityClusterTasks
      value: "false"
...
```

- この更新では、**TektonConfig** カスタムリソースの **enable-devconsole-integration** フラグを **false** に設定することで、Tekton Hub と **Developer** パースペクティブの統合を無効にできます。以下に例を示します。

```
...
hub:
  params:
    - name: enable-devconsole-integration
      value: "true"
...
```

- 今回の更新により、**operator-config.yaml** 設定マップにより、**tkn version** コマンドの出力で Operator バージョンを表示できるようになります。

- この更新により、**argocd-task-sync-and-wait** タスクのバージョンが **v0.2** に変更されます。

- この **TektonConfig**CRD の更新により、**oc get tektonconfig** コマンドは OPerator のバージョンを表示します。

- この更新により、サービスモニターがトリガーメトリクスに追加されます。

1.10.1.5. ハブ

重要

Tekton Hub をクラスターにデプロイすることは、テクノロジープレビュー機能です。

Tekton Hub は、CI/CD ワークフローの再利用可能なタスクとパイプラインを検出、検索、および共有するのに役立ちます。Tekton Hub のパブリックインスタンスは、hub.tekton.dev で利用できます。

Red Hat OpenShift Pipelines 1.7 を確認しながら、クラスター管理者は Tekton Hub のカスタムインスタンスをエンタープライズクラスターにインストールしてデプロイすることもできます。組織に固有の再利用可能なタスクとパイプラインを使用してカタログをキュレートできます。

1.10.1.6. チェーン

重要

Tekton Chains はテクノロジープレビュー機能です。

Tekton Chains は、Kubernetes カスタムリソース定義 (CRD) コントローラーです。これを使用して、Red Hat OpenShift Pipelines を使用して作成されたタスクおよびパイプラインのサプライチェーンセキュリティーを管理できます。

デフォルトでは、Tekton Chains は OpenShift Container Platform クラスターで実行されるタスクをモニターします。Chains は、完了したタスク実行のスナップショットを取得し、それらを 1 つ以上の標準ペイロード形式に変換し、すべてのアーティファクトに署名して保存します。

Tekton Chains は、次の機能をサポートしています。

- 暗号化キータイプと **cosign** などのサービスを使用して、タスク実行、タスク実行結果、および OCI レジストリーイメージに署名できます。
- **in-toto** などの認証形式を使用できます。
- OCI リポジトリーをストレージバックエンドとして使用して、署名と署名されたアーティファクトを安全に保存できます。

1.10.1.7. Pipelines as Code (PAC)

重要

Pipelines as Code は、テクノロジープレビュー機能です。

Pipelines as Code を使用すると、クラスター管理者と必要な権限を持つユーザーは、パイプラインテンプレートをソースコード Git リポジトリーの一部として定義できます。設定された Git リポジトリーのソースコードプッシュまたはプルリクエストによってトリガーされると、この機能はパイプラインを実行し、ステータスを報告します。

Pipelines as Code は、次の機能をサポートしています。

- プルリクエストのステータス。プルリクエストを反復処理する場合、プルリクエストのステータスと制御は Git リポジトリーをホストしているプラットフォームで実行されます。
- GitHub は API をチェックして、再チェックを含むパイプライン実行のステータスを設定します。
- GitHub のプルリクエストとコミットイベント。
- **/retest** などのコメントでリクエストアクションをプルします。
- Git イベントのフィルタリング、およびイベントごとの個別のパイプライン。

- ローカルタスク、Tekton Hub、およびリモート URL の OpenShift Pipelines での自動タスク解決。
- 設定を取得するための GitHub blobs およびオブジェクト API の使用。
- GitHub 組織を介して、または Prow スタイルの **OWNER** ファイルを使用したアクセス制御リスト (ACL)。
- **tkn** CLI ツール用の **tkn pac** プラグイン。これを使用して Pipelines as Code リポジトリーとブートストラップを管理できます。
- GitHub アプリケーション、GitHub Webhook、Bitbucket Server、および Bitbucket Cloud のサポート。

1.10.2. 非推奨の機能

- 重大な変更: この更新により、**TektonConfig** カスタムリソース (CR) から **disable-working-directory-overwrite** および **disable-home-env-overwrite** フィールドが削除されます。その結果、**TektonConfig** CR は **$HOME** 環境変数と **workingDir** パラメーターを自動的に設定しなくなりました。**タスク** カスタムリソース定義 (CRD) の **env** および **workingDir** フィールドを使用して、引き続き **$HOME** 環境変数と **workingDir** パラメーターを設定できます。
- **Conditions** カスタムリソース定義 (CRD) タイプは非推奨であり、将来のリリースで削除される予定です。代わりに、推奨される **When** 式を使用してください。
- 重大な変更: **EventListener** と **TriggerBinding** の値を指定しない場合、**Triggers** リソースはテンプレートを検証し、エラーを生成します。

1.10.3. 既知の問題

- Maven および Jib Maven クラスタータスクを実行する場合には、デフォルトのコンテナーイメージは Intel(x86) アーキテクチャーでのみサポートされます。したがって、タスクは ARM、IBM Power Systems (ppc64le)、IBM Z、および LinuxONE (s390x) クラスターで失敗します。回避策として、**MAVEN_IMAGE** パラメーターの値を **maven:3.6.3-adoptopenjdk-11** に設定すると、カスタムイメージを指定できます。

  ヒント

  tkn hub を使用して、ARM、IBM Power Systems (ppc64le)、IBM Z、および LinuxONE (s390x) に Tekton カタログに基づくタスクをインストールする前に、これらのプラットフォームでタスクを実行できるかどうかを確認してください。**ppc64le** および **s390x** がタスク情報の Platforms セクションに一覧表示されているかどうかを確認するには、**tkn hub info task <name>** コマンドを実行します。

- IBM Power Systems、IBM Z、および LinuxONE では、**s2i-dotnet** クラスタータスクはサポートされません。
- **nodejs:14-ubi8-minimal** イメージストリームを使用すると、以下のエラーが生成されるため、使用できません。

```
STEP 7: RUN /usr/libexec/s2i/assemble
/bin/sh: /usr/libexec/s2i/assemble: No such file or directory
subprocess exited with status 127
```

```
subprocess exited with status 127
error building at STEP "RUN /usr/libexec/s2i/assemble": exit status 127
time="2021-11-04T13:05:26Z" level=error msg="exit status 127"
```

- 暗黙的なパラメーターマッピングは、最上位の **Pipeline** または **PipelineRun** 定義から **taskRef** タスクにパラメーターを誤って渡します。マッピングは、トップレベルのリソースからインライン **taskSpec** 仕様のタスクにのみ行う必要があります。この問題は、**TektonConfig** カスタムリソース定義の **pipeline** セクションで **enable-api-fields** フィールドを **alpha** に設定することにより、この機能が有効になっているクラスターにのみ影響します。

1.10.4. 修正された問題

- 今回の更新では、**labels** や **annotations** などのメタデータが **Pipeline** オブジェクト定義と **PipelineRun** オブジェクト定義の両方に存在する場合、**PipelineRun** タイプの値が優先されます。**Task** オブジェクトと **TaskRun** オブジェクトで同様の動作が見られます。

- この更新では、**timeouts.tasks** フィールドまたは **timeouts.finally** フィールドが **0** に設定されている場合、**timeouts.pipeline** も **0** に設定されます。

- この更新により、シバンを使用しないスクリプトから **-x** セットフラグが削除されました。この修正により、スクリプト実行による潜在的なデータ漏洩が減少します。

- この更新により、Git クレデンシャルのユーザー名に存在するバックスラッシュ文字は、**.gitconfig** ファイルの追加のバックスラッシュでエスケープされます。

- この更新により、**EventListener** オブジェクトの **finalizer** プロパティーは、ロギングおよび設定マップのクリーンアップに必要なくなりました。

- この更新により、イベントリスナーサーバーに関連付けられているデフォルトの HTTP クライアントが削除され、カスタム HTTP クライアントが追加されます。その結果、タイムアウトが改善されました。

- この更新により、トリガークラスターのロールが所有者の参照で機能するようになりました。

- この更新では、複数のインターセプターが拡張機能を返す場合、イベントリスナーの競合状態は発生しません。

- この更新により、**tkn pr delete** コマンドは、**ignore-running** フラグで実行されているパイプラインを削除しません。

- この更新では、アドオンパラメーターを変更しても、Operator Pod は再起動し続けません。

- この更新により、サブスクリプションおよび設定カスタムリソースで設定されていない場合、**tkn serve** CLI Pod はインフラノードでスケジュールされます。

- この更新では、指定されたバージョンのクラスタータスクはアップグレード中に削除されません。

1.10.5. Red Hat OpenShift Pipelines General Availability 1.7.1 のリリースノート

Red Hat OpenShift Pipelines General Availability (GA) 1.7.1 が OpenShift Container Platform 4.9、4.10、および 4.11 で利用可能になりました。

1.10.5.1. 修正された問題

- 今回の更新以前は、Red Hat OpenShift Pipelines Operator をアップグレードすると、Tekton

Hub に関連付けられたデータベースのデータが削除され、新規データベースがインストールされていました。今回の更新により、Operator のアップグレードでデータが保存されるようになりました。

- 今回の更新以前は、クラスター管理者のみが OpenShift Container Platform コンソールでパイプラインメトリクスにアクセスできていました。今回の更新により、他のクラスターロールを持つユーザーもパイプラインメトリクスにアクセスできるようになりました。

- 今回の更新以前は、大量の終了メッセージを生成するタスクが含まれるパイプラインの場合、パイプラインの実行に失敗しました。Pod 内のすべてのコンテナーの終了メッセージの合計サイズは 12 KB を超えることができないために、パイプライン実行が失敗しました。今回の更新により、同じイメージを使用する **place-tools** および **step-init** 初期化コンテナーがマージされ、各タスクの Pod で実行されているコンテナーの数が減りました。このソリューションにより、タスクの Pod で実行されているコンテナーの数を最小限にすることにより、パイプライン実行に失敗する可能性を減らすことができます。ただし、終了メッセージの最大許容サイズの制限は削除されません。

- 今回の更新以前は、Tekton Hub Web コンソールからリソースの URL に直接アクセスしようとすると、Nginx **404** エラーが発生しました。今回の更新で、Tekton Hub Web コンソールイメージは、Tekton Hub Web コンソールから直接リソースの URL にアクセスできるように修正されました。

- 今回の更新以前は、namespace ごとにリソースプルーナージョブがリソースのプルーニング用に別個のコンテナーを作成していました。今回の更新により、リソースプルーナージョブはすべての namespace のコマンドを 1 つのコンテナーのループとして実行するようになりました。

1.10.6. Red Hat OpenShift Pipelines General Availability 1.7.2 のリリースノート

Red Hat OpenShift Pipelines General Availability (GA) 1.7.2 が OpenShift Container Platform 4.9、4.10、およびそれ以降のバージョンで利用可能になりました。

1.10.6.1. 既知の問題

- **openshift-pipelines** namespace の Tekton Chains の **chains-config** 設定マップは、Red Hat OpenShift Pipelines Operator のアップグレード後に自動的にデフォルト値にリセットされます。現在、この問題に対する回避策はありません。

1.10.6.2. 修正された問題

- 今回の更新前は、OpenShift Pipelines 1.7.1 のタスクは、最初の引数として **init** を使用し、その後に 2 つ以上の引数を使用すると失敗しました。今回の更新により、フラグが正しく解析され、タスクが正常に実行されるようになりました。

- 今回の更新以前は、無効なロールバインディングにより、OpenShift Container Platform 4.9 および 4.10 への Red Hat OpenShift Pipelines Operator のインストールは、以下のエラーメッセージと共に失敗していました。

  ```
  error updating rolebinding openshift-operators-prometheus-k8s-read-binding:
  RoleBinding.rbac.authorization.k8s.io "openshift-operators-prometheus-k8s-read-binding" is
  invalid: roleRef: Invalid value: rbac.RoleRef{APIGroup:"rbac.authorization.k8s.io",
  Kind:"Role", Name:"openshift-operator-read"}: cannot change roleRef
  ```

  今回の更新により、Red Hat OpenShift Pipelines Operator は個別のロールバインディング namespace でインストールし、他の Operator のインストールとの競合を回避するようになりました。

- 今回の更新以前は、Operator をアップグレードすると、Tekton Chains の **signing-secrets** シークレットキーがデフォルト値にリセットされていました。今回の更新により、カスタムシークレットキーは Operator のアップグレード後も永続するようになりました。

注記

Red Hat OpenShift Pipelines 1.7.2 へのアップグレードにより、キーがリセットされます。ただし、それ以降のリリースにアップグレードすると、キーは永続化される予定です。

- 今回の更新以前は、すべての S2I ビルドタスクが以下の様なエラーメッセージと共に失敗していました。

```
Error: error writing "0 0 4294967295\n" to /proc/22/uid_map: write /proc/22/uid_map:
operation not permitted
time="2022-03-04T09:47:57Z" level=error msg="error writing \"0 0 4294967295\\n\" to
/proc/22/uid_map: write /proc/22/uid_map: operation not permitted"
time="2022-03-04T09:47:57Z" level=error msg="(unable to determine exit status)"
```

今回の更新により、**pipelines-scc** セキュリティーコンテキスト制約 (SCC) は、 **Buildah** および **S2I** クラスタータスクに必要な **SETFCAP** 機能と互換性が確保されています。その結果、**Buildah** および **S2I** ビルドタスクを正常に実行できます。

さまざまな言語やフレームワークで書かれたアプリケーションに対して **Buildah** クラスタータスクおよび **S2I** ビルドタスクを正常に実行するには、**build** や **push** などの適切な **steps** オブジェクトに以下のスニペットを追加します。

```
securityContext:
  capabilities:
    add: ["SETFCAP"]
```

1.10.7. Red Hat OpenShift Pipelines General Availability 1.7.3 のリリースノート

Red Hat OpenShift Pipelines General Availability (GA) 1.7.3 が OpenShift Container Platform 4.9、4.10、および 4.11 で利用可能になりました。

1.10.7.1. 修正された問題

- 今回の更新前は、いずれかの namespace が **Terminating** 状態の場合、RBAC リソースの作成時に Operator が失敗していました。今回の更新により、Operator は **Terminating** 状態の namespace を無視し、RBAC リソースを作成します。

- 以前は、Red Hat OpenShift Pipelines Operator をアップグレードすると **pipeline** サービスアカウントが再作成され、サービスアカウントにリンクされたシークレットが失われていました。今回の更新でこの問題が修正されています。アップグレード中に、Operator は **pipeline** サービスアカウントを再作成しなくなりました。その結果、**pipeline** サービスアカウントにアタッチされたシークレットはアップグレード後も保持され、リソース (タスクとパイプライン) は引き続き正しく機能します。

1.11. RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY (GA) 1.6 のリリースノート

Red Hat OpenShift Pipelines General Availability (GA) 1.6 が OpenShift Container Platform 4.9 で利用可能になりました。

1.11.1. 新機能

以下では、修正および安定性の面での改善点に加え、OpenShift Pipelines 1.6 の主な新機能について説明します。

- 今回の更新により **--output <string>** を使用して、YAML または JSON 形式の文字列を返すようにパイプラインまたはタスクの **start** コマンドを設定できるようになりました。ここでは、**<string>** は **yaml** または **json** に置き換えます。**--output** オプションを指定しないと、**start** コマンドは人間による解読はしやすくなりますが、他のプログラムによる解析が難しいメッセージを返します。継続的インテグレーション (CI) 環境では、YAML または JSON 形式の文字列を返す機能は便利です。たとえば、リソースの作成後に **yq** または **jq** を使用して、リソースに関する YAML または JSON 形式のメッセージを解析し、**showlog** オプションを使用せずにそのリソースが終了するまで待機します。

- 今回の更新により、Podman の **auth.json** 認証ファイルを使用してレジストリーに対して認証できるようになりました。たとえば、**tkn bundle push** を使用して、Docker CLI ではなく Podman を使用してリモートレジストリーにプッシュできます。

- 今回の更新により、**tkn [taskrun | pipelinerun] delete --all** コマンドを使用すると、新規の **--keep-since <minutes>** オプションを使用して、指定した期間よりも後の実行を保持できます。たとえば、5 分未満の実行を維持するには、**tkn [taskrun | pipelinerun] delete -all --keep-since 5** を入力します。

- 今回の更新により、タスク実行またはパイプライン実行を削除する際に、**--parent-resource** と **--keep-since** オプションを同時に使用できるようになりました。たとえば、**tkn pipelinerun delete --pipeline pipelinename --keep-since 5** コマンドは、親リソースの名前が **pipelinename** で、その経過時間が 5 分以下であるパイプラインの実行を保持します。**tkn tr delete -t <taskname> --keep-since 5** および **tkn tr delete --clustertask <taskname> --keep-since 5** コマンドはタスク実行と同様に機能します。

- 今回の更新により、**v1beta1** リソースと連携するトリガーリソースのサポートが追加されました。

- 今回の更新により、**ignore-running** オプションが **tkn pipelinerun delete** および **tkn taskrun delete** コマンドに追加されています。

- 今回の更新により、**create** サブコマンドが **tkn task** と **tkn clustertask** コマンドに追加されました。

- 今回の更新により、**tkn pipelinerun delete --all** コマンドを使用すると、新規の **--label <string>** オプションを使用して、ラベルでパイプライン実行をフィルターできるようになりました。オプションで、**--label** オプションに **=** と **==** を等価演算子として、または **!=** を不等価演算子として指定して使用できます。たとえば、**tkn pipelinerun delete --all --label asdf** および **tkn pipelinerun delete --all --label==asdf** コマンドはどちらも、**asdf** ラベルが割り当てられたすべてのパイプライン実行を削除します。

- 今回の更新では、設定マップからインストールされた Tekton コンポーネントのバージョンを取得するか、設定マップがない場合はデプロイメントコントローラーから取得できるようになりました。

- 今回の更新では、機能フラグを設定し、デフォルト値をそれぞれ設定するために **feature-flags** と **config-defaults** 設定マップをサポートするようになりました。

- 今回の更新では、新しいメトリクス **eventlistener_event_count** が追加され、**EventListener** リソースが受信するイベントをカウントできるようになりました。

- 今回の更新では、**v1beta1** Go API タイプが追加されました。今回の更新では、トリガーが **v1beta1** API バージョンをサポートするようになりました。
  現在のリリースでは、**v1alpha1** 機能が非推奨となり、今後のリリースで削除されます。代わりに **v1beta1** 機能の使用を開始します。

- 現在のリリースでは、リソースの自動実行がデフォルトで有効になっています。さらに、以下の新規アノテーションを使用して、namespace ごとにタスク実行およびパイプライン実行を自動実行するように設定できます。

  - **operator.tekton.dev/prune.schedule**: このアノテーションの値が **TektonConfig** カスタムリソース定義で指定された値と異なる場合には、その namespace に新規の cron ジョブが作成されます。

  - **operator.tekton.dev/prune.skip**: **true** に設定されている場合、設定先の namespace はプルーニングされません。

  - **operator.tekton.dev/prune.resources**: このアノテーションではリソースのコンマ区切りのリストを使用できます。パイプライン実行などの単一リソースをプルーニングするには、このアノテーションを **pipelinerun** に設定します。task run や pipeline run などの複数のリソース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。

  - **operator.tekton.dev/prune.keep**: このアノテーションを使用して、プルーニングなしでリソースを保持します。

  - **operator.tekton.dev/prune.keep-since**: このアノテーションを使用して、経過時間をもとにリソースを保持します。このアノテーションの値は、リソースの経過時間 (分単位) と等しくなければなりません。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 および **keep-since** アノテーションは同時に使用できません。リソースには、どちらか 1 つだけを使用する必要があります。

  - **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。

- 管理者はクラスター全体に対する **pipeline** サービスアカウントの作成を無効にし、紐付けされた SCC (**anyuid** と非常に似ている) の悪用による権限昇格を防ぎます。

- **TektonConfig** カスタムリソース (CR) および、**TektonPipeline** と **TektonTriggers** などの個々のコンポーネントの CR を使用して、機能フラグおよびコンポーネントを設定できるようになりました。この詳細レベルは、個々のコンポーネントの Tekton OCI バンドルなどのアルファ機能のカスタマイズおよびテストに役立ちます。

- **PipelineRun** リソースのオプションの **Timeouts** フィールドを設定できるようになりました。たとえば、パイプライン実行、各タスク実行、および **finally** タスクに個別にタイムアウトを設定できます。

- **TaskRun** リソースで生成される Pod を使用して、Pod の **activeDeadlineSeconds** フィールドが設定されるようになりました。これにより、OpenShift はこの値を終了として考慮でき、Pod に具体的にスコープを指定した **ResourceQuota** オブジェクトを使用できます。

- configmaps を使用して、タスク実行、パイプライン実行、タスク、およびパイプラインのメトリクスタグまたはラベルタイプを削除できます。さらに、ヒストグラム、ゲージ、最終値など、測定期間に、さまざまな種類のメトリクスを設定できます。

- Tekton は **Min**、**Max**、**Default** および **DefaultRequest** フィールドを考慮して **LimitRange** オブジェクトを完全にサポートするため、一貫性をもたせて Pod への要求および制限を定義できます。

- 以下のアルファ機能が導入されました。

  - パイプライン実行は、以前の動作のように、すべてのタスク実行を直接停止するのではなく、**finally** タスクの実行後に停止できるようになりました。今回の更新により、以下の **spec.status** 値が追加されました。

    - **StoppedRunFinal** は、完了後、現在実行中のタスクを停止し、**finally** タスクを実行します。

    - **CancelledRun** は、実行中のタスクをすぐにキャンセルしてから、**finally** タスクを実行します。

    - **Cancelled** は、**PipelineRunCancelled** ステータスで提供される以前の動作を保持します。

      注記

      非推奨となった **PipelineRunCancelled** ステータスは **v1** バージョンで削除され、**Cancelled** ステータスに置き換えられます。

  - **oc debug** コマンドを使用して、タスク実行をデバッグモードに配置できるようになりました。これにより、実行を一時停止し、Pod で特定の手順を検査できるようになりました。

  - ステップの **onError** フィールドを **continue** に設定すると、ステップの終了コードが記録され、後続のステップに渡されます。ただし、タスク実行は失敗しないので、タスクの残りのステップの実行は継続されます。既存の動作を維持するには、**onError** フィールドの値を **stopAndFail** に設定します。

  - タスクは、実際に使用されているよりも多くのパラメーターを受け入れるようになりました。アルファ機能フラグを有効にすると、パラメーターは暗黙的にインライン仕様に伝播できます。たとえば、インラインのタスクは、タスクの各パラメーターを明示的に定義せずに、親パイプライン実行のパラメーターにアクセスできます。

  - アルファ機能のフラグを有効にすると、**when** 式の条件が、直接関連付けられたタスクにのみ適用され、タスクに依存することはありません。**When** 式を関連タスクとその依存に適用するには、式を依存タスクごとに個別に関連付ける必要があります。今後、これが Tekton の新規 API バージョンの **When** 式のデフォルト動作になることに注意してください。今回の更新が優先され、既存のデフォルト動作は非推奨になりました。

- 現在のリリースでは、**nodeSelector** および **tolerations** の値を **TektonConfig カスタムリソース** (CR) に指定することで、ノードの選択を設定できます。Operator はこれらの値を、作成するすべてのデプロイメントに追加します。

  - Operator のコントローラーおよび Webhook デプロイメントのノード選択を設定するには、Operator のインストール後に **Subscription** CR の仕様で **config.nodeSelector** および **config.tolerations** フィールドを編集します。

- OpenShift Pipelines の残りのコントロールプレーン Pod をインフラストラクチャーノードにデプロイするには、**nodeSelector** および **tolerations** フィールドで **TektonConfig** CR を更新します。その後、変更は Operator で作成されるすべての Pod に適用されます。

1.11.2. 非推奨の機能

- CLI 0.21.0 では、**clustertask**、**task**、**taskrun**、**pipeline**、および **pipelinerun** コマンドに対するすべての **v1alpha1** リソースのサポートが非推奨になりました。クラスターローダーが非推奨になり、今後のリリースで削除されます。

- Tekton Triggers v0.16.0 では、重複する **status** ラベルは **EventListener** リソースのメトリクスから削除されます。

重要

重大な変更:**status** ラベルは **eventlistener_http_duration_seconds_*** メトリクスから削除されました。**status** ラベルに基づくクエリーを削除します。

- 現在のリリースでは、**v1alpha1** 機能が非推奨となり、今後のリリースで削除されます。代わりに、今回の更新では、**v1beta1** Go API タイプの使用を開始できるようになりました。トリガーが **v1beta1** API バージョンをサポートするようになりました。

- 現在のリリースでは、**EventListener** リソースはトリガーの終了処理前に応答を送信します。

重要

重大な変更: 今回の変更により、**EventListener** リソースがリソースの作成時に **201 Created** ステータスコードに応答しなくなります。代わりに **202 Accepted** 応答コードで応答します。

- 今回のリリースで、**podTemplate** フィールドが **EventListener** リソースから削除されます。

重要

重大な変更: #1100 の一部として非推奨となった **podTemplate** フィールドが削除されました。

- 今回のリリースで、非推奨の **replicas** フィールドが **EventListener** リソースの仕様から削除されます。

重要

重大な変更: 非推奨の **replicas** フィールドが削除されました。

- Red Hat OpenShift Pipelines 1.6 では、**HOME="/tekton/home"** および **workingDir="/workspace"** の値が **Step** オブジェクトの仕様から削除されます。
代わりに、Red Hat OpenShift Pipelines は、**Step** オブジェクトを実行するコンテナーで定義される値に **HOME** および **workingDir** を設定します。これらの値は、**Step** オブジェクトの仕様で上書きできます。

  以前の動作を使用するには、**TektonConfig** CR の **disable-working-directory-overwrite** フィールドおよび **disable-home-env-overwrite** フィールドを **false** に変更します。

```
apiVersion: operator.tekton.dev/v1alpha1
  kind: TektonConfig
  metadata:
    name: config
  spec:
    pipeline:
      disable-working-directory-overwrite: false
      disable-home-env-overwrite: false
  ...
```

重要

TektonConfig CR の **disable-working-directory-overwrite** と **disable-home-env-overwrite** フィールドは非推奨となり、今後のリリースで削除されます。

1.11.3. 既知の問題

- Maven および Jib Maven クラスタータスクを実行する場合には、デフォルトのコンテナーイメージは Intel(x86) アーキテクチャーでのみサポートされます。したがって、IBM Power Systems(ppc64le)、IBM Z、および LinuxONE(s390x) クラスターではタスクに失敗します。回避策として、**MAVEN_IMAGE** パラメーターの値を **maven:3.6.3-adoptopenjdk-11** に設定すると、カスタムイメージを指定できます。

- IBM Power Systems、IBM Z、および LinuxONE では、**s2i-dotnet** クラスタータスクはサポートされません。

- **tkn hub** を使用して IBM Power Systems(ppc64le)、IBM Z、および LinuxONE(s390x) の Tekton Catalog をもとにタスクをインストールする前に、タスクがこれらのプラットフォームで実行できるかどうかを確認します。**ppc64le** および **s390x** がタスク情報の Platforms セクションにリスト表示されているかどうかを確認するには、**tkn hub info task <name>** コマンドを実行します。

- **nodejs:14-ubi8-minimal** イメージストリームを使用すると、以下のエラーが生成されるため、使用できません。

```
STEP 7: RUN /usr/libexec/s2i/assemble
/bin/sh: /usr/libexec/s2i/assemble: No such file or directory
subprocess exited with status 127
subprocess exited with status 127
error building at STEP "RUN /usr/libexec/s2i/assemble": exit status 127
time="2021-11-04T13:05:26Z" level=error msg="exit status 127"
```

1.11.4. 修正された問題

- IBM Power Systems、IBM Z、および LinuxONE では、**tkn hub** コマンドはサポート対象外になりました。

- この更新以前は、ユーザーが **tkn** コマンドの実行後にターミナルを利用できず、**再試行** が指定された場合でもパイプライン実行が行われていました。タスク実行またはパイプライン実行のタイムアウトの指定には影響がありません。今回の更新で問題が修正され、コマンド実行後にターミナルが利用できるようになります。

- 今回の更新以前は、**tkn pipelinerun delete --all** を実行すると、すべてのリソースが削除されました。今回の更新で、実行中の状態のリソースが削除されなくなりました。

- 今回の更新以前は、**tkn version --component=<component>** コマンドを使用しても、コンポーネントのバージョンが返されませんでした。今回の更新でこの問題が修正され、このコマンドを使用すると、コンポーネントのバージョンを返すようになりました。

- 今回の更新以前は、**tkn pr logs** コマンドを使用すると、パイプラインの出力ログでタスクの順番が間違って表示されていました。今回の更新で問題は解決され、完了した **PipelineRun** のログで、**TaskRun** 実行順序を適切に表示するようになりました。

- 今回の更新以前は、実行中のパイプラインの仕様を編集すると、パイプライン実行が完了時に停止できなくなる可能性がありました。今回の更新では、定義を 1 度だけフェッチし、検証用にステータスに保存されている仕様を使用して問題を修正しています。今回の変更により、**PipelineRun** または **TaskRun** が実行中の **Pipeline** または **Task** を参照する場合に競合状態に陥る確率が削減されます。

- **when** 式値に、**[$(params.arrayParam[*])]** などの配列パラメーター参照を指定できるようになりました。

1.11.5. Red Hat OpenShift Pipelines General Availability 1.6.1 のリリースノート

1.11.5.1. 既知の問題

- 古いバージョンから Red Hat OpenShift Pipelines 1.6.1 にアップグレードした後、OpenShift Pipelines は一貫性のない状態になり、Tekton リソース (タスクおよびパイプライン) に対して操作 (作成/削除/適用) を実行できない場合があります。たとえば、リソースの削除中に、以下のエラーが発生する可能性があります。

> Error from server (InternalError): Internal error occurred: failed calling webhook "validation.webhook.pipeline.tekton.dev": Post "https://tekton-pipelines-webhook.openshift-pipelines.svc:443/resource-validation?timeout=10s": service "tekton-pipelines-webhook" not found.

1.11.5.2. 修正された問題

- Red Hat OpenShift Pipelines によって設定される **SSL_CERT_DIR** 環境変数 (**/tekton-custom-certs**) は、以下のデフォルトのシステムディレクトリーを証明書ファイルで上書きしません。
  - **/etc/pki/tls/certs**
  - **/etc/ssl/certs**
  - **/system/etc/security/cacerts**

- Horizontal Pod Autoscaler は、Red Hat OpenShift Pipelines Operator によって制御されるデプロイメントのレプリカ数を管理できます。このリリース以降、エンドユーザーまたはクラスター上のエージェントによってカウントが変更された場合、Red Hat OpenShift Pipelines Operator はそれによって管理されるデプロイメントのレプリカカウントをリセットしません。ただし、Red Hat OpenShift Pipelines Operator のアップグレード時にレプリカはリセットされます。

- **tkn** CLI を提供する Pod は、ノードセレクターおよび **TektonConfig** カスタムリソースで指定される容認制限に基づいて、ノードにスケジュールされるようになりました。

1.11.6. Red Hat OpenShift Pipelines General Availability 1.6.2 のリリースノート

1.11.6.1. 既知の問題

- 新規プロジェクトの作成時に、**pipeline** サービスアカウントの作成が遅延し、既存のクラスタータスクおよびパイプラインテンプレートの削除に 10 分以上かかります。

1.11.6.2. 修正された問題

- 今回の更新以前は、古いバージョンから Red Hat OpenShift Pipelines 1.6.1 にアップグレードした後に、Tekton インストーラーセットの複数のインスタンスがパイプライン用に作成されました。今回の更新では、Operator により、アップグレード後に **TektonInstallerSet** の各タイプのインスタンスが 1 つだけ存在するようになりました。

- 今回の更新以前は、Operator のすべてのリコンサイラーはコンポーネントバージョンを使用して、古いバージョンから Red Hat OpenShift Pipelines 1.6.1 へのアップグレード時にリソース再作成を決定していました。その結果、アップグレード時にコンポーネントのバージョンが変更されなかったリソースは再作成されませんでした。今回の更新により、Operator はコンポーネントのバージョンではなく Operator バージョンを使用して、アップグレード時にリソースの再作成を決定するようになりました。

- この更新の前は、アップグレード後にパイプライン Webhook サービスがクラスターにありませんでした。これは、設定マップのアップグレードのデッドロックが原因でした。今回の更新により、設定マップがクラスターにない場合に Webhook 検証を無効にするメカニズムが追加されました。その結果、パイプライン Webhook サービスはアップグレード後もクラスターで永続化します。

- 今回の更新以前は、namespace への設定変更後に自動プルーニングの cron ジョブは再作成されていました。今回の更新により、namespace に関連するアノテーションが変更された場合のみ、自動プルーニングの Cron ジョブは再作成されるようになりました。

- Tekton Pipelines のアップストリームバージョンは **v0.28.3** に改訂され、以下の修正が加えられました。

    - **PipelineRun** または **TaskRun** オブジェクトを修正し、ラベルまたはアノテーションの伝搬を許可します。

    - 暗黙的なパラメーターの場合:

        - **PipelineSpec** パラメーターを **TaskRefs** オブジェクトに適用しないでください。

        - **Pipeline** オブジェクトの暗黙的なパラメーター動作を無効にします。

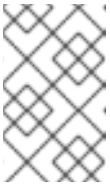
1.11.7. Red Hat OpenShift Pipelines General Availability 1.6.3 のリリースノート

1.11.7.1. 修正された問題

- 今回の更新以前は、Red Hat OpenShift Pipelines Operator は Pipeline および Trigger などのコンポーネントから Pod セキュリティーポリシーをインストールしていました。ただし、コンポーネントの一部として同梱される Pod セキュリティーポリシーは、以前のリリースで非推奨となりました。今回の更新により、Operator はコンポーネントから Pod セキュリティーポリシーをインストールするのを止めました。その結果、以下のアップグレードパスが影響を受けます。

    - OpenShift Pipelines 1.6.1 または 1.6.2 から OpenShift Pipelines 1.6.3 にアップグレードすると、Pipelines および Triggers コンポーネントからのものを含む Pod セキュリティーポリシーが削除されます。

- OpenShift Pipelines 1.5.x から 1.6.3 へのアップグレードでは、コンポーネントからインストールされた Pod セキュリティーポリシーが保持されます。クラスター管理者は、それらを手動で削除できます。



注記

今後のリリースにアップグレードすると、Red Hat OpenShift Pipelines Operator は古くなったすべての Pod セキュリティーポリシーを自動的に削除します。

- 今回の更新以前は、クラスター管理者のみが OpenShift Container Platform コンソールでパイプラインメトリクスにアクセスできていました。今回の更新により、他のクラスターロールを持つユーザーもパイプラインメトリクスにアクセスできるようになりました。

- 今回の更新前は、OpenShift Pipelines Operator のロールベースのアクセス制御 (RBAC) の問題により、コンポーネントのアップグレードまたはインストールで問題が発生していました。今回の更新により、各種の Red Hat OpenShift Pipelines コンポーネントをインストールする際の信頼性および一貫性が向上しました。

- 今回の更新以前は、**TektonConfig** CR で **clusterTasks** および **pipelineTemplates** フィールドを **false** に設定すると、クラスタータスクおよびパイプラインテンプレートの削除が遅くなりました。この更新により、クラスタータスクやパイプラインテンプレートなどの Tekton リソースのライフサイクル管理の速度が改善されました。

1.11.8. Red Hat OpenShift Pipelines General Availability (GA) 1.6.4 のリリースノート

1.11.8.1. 既知の問題

- Red Hat OpenShift Pipelines 1.5.2 から 1.6.4 にアップグレードした後、イベントリスナールートにアクセスすると **503** エラーが返されます。
  回避策: YAML ファイルで、イベントリスナーのルートのターゲットポートを変更します。

1. 関連する namespace のルート名を抽出します。

```
$ oc get route -n <namespace>
```

2. ルートを編集して、**targetPort** フィールドの値を変更します。

```
$ oc edit route -n <namespace> <el-route_name>
```

例: 既存のイベントリスナールート

```
...
spec:
  host: el-event-listener-q8c3w5-test-upgrade1.apps.ve49aws.aws.ospqa.com
  port:
    targetPort: 8000
  to:
    kind: Service
    name: el-event-listener-q8c3w5
    weight: 100
  wildcardPolicy: None
...
```

例: 変更されたイベントリスナールート

```
...
spec:
  host: el-event-listener-q8c3w5-test-upgrade1.apps.ve49aws.aws.ospqa.com
  port:
    targetPort: http-listener
  to:
    kind: Service
    name: el-event-listener-q8c3w5
    weight: 100
  wildcardPolicy: None
...
```

1.11.8.2. 修正された問題

- 今回の更新前は、いずれかの namespace が **Terminating** 状態の場合、RBAC リソースの作成時に Operator が失敗していました。今回の更新により、Operator は **Terminating** 状態の namespace を無視し、RBAC リソースを作成します。
- この更新の前は、関連する Tekton コントローラーのリリースバージョンを指定するアノテーションがないため、タスクの実行が失敗するか、再起動されました。今回の更新により、適切な注釈の組み込みが自動化され、タスクは失敗や再起動なしで実行されます。

1.12. RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY (GA) 1.5 のリリースノート

Red Hat OpenShift Pipelines General Availability (GA) 1.5 が OpenShift Container Platform 4.8 で利用可能になりました。

1.12.1. 互換性およびサポート表

現在、今回のリリースに含まれる機能には テクノロジープレビュー のものがあります。これらの実験的機能は、実稼働環境での使用を目的としていません。

以下の表では、機能は以下のステータスでマークされています。

TP	テクノロジープレビュー
GA	一般公開 (GA)

これらの機能に関しては、Red Hat カスタマーポータルの以下のサポート範囲を参照してください。

表1.3 互換性およびサポート表

機能	バージョン	サポートステータス
Pipelines	0.24	GA
CLI	0.19	GA

機能	バージョン	サポートステータス
カタログ	0.24	GA
トリガー	0.14	TP
パイプラインリソース	-	TP

質問やフィードバックについては、製品チームに pipelines-interest@redhat.com 宛のメールを送信してください。

1.12.2. 新機能

以下では、修正および安定性の面での改善点に加え、OpenShift Pipelines 1.5 の主な新機能について説明します。

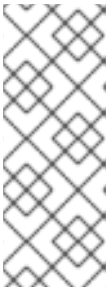
- パイプライン実行およびタスク実行は、ターゲット namespace の cron ジョブによって自動的にプルーニングされます。cron ジョブは **IMAGE_JOB_PRUNER_TKN** 環境変数の値を使用して **tkn image** の値を取得します。今回の機能拡張により、以下のフィールドが **TektonConfig** カスタムリソースに導入されるようになりました。

```
...
pruner:
  resources:
    - pipelinerun
    - taskrun
  schedule: "*/5 * * * *" # cron schedule
  keep: 2 # delete all keeping n
...
```

- OpenShift Container Platform で、Tekton Add-ons コンポーネントのインストールをカスタマイズするには、**TektonConig** カスタムリソースの新規パラメーター **clusterTasks** および **pipelinesTemplates** の値を変更します。

```
apiVersion: operator.tekton.dev/v1alpha1
kind: TektonConfig
metadata:
  name: config
spec:
  profile: all
  targetNamespace: openshift-pipelines
  addon:
    params:
    - name: clusterTasks
      value: "true"
    - name: pipelineTemplates
      value: "true"
...
```

  カスタマイズは、**TektonConfig** を使用してアドオンを作成するか、Tekton Add-ons を使用して直接アドオンを作成する場合に許可されます。ただし、パラメーターが渡されない場合、コントローラーはデフォルト値でパラメーターを追加します。



注記

- アドオンが **TektonConfig** カスタムリソースを使用して作成され、**Addon** カスタムリソースでパラメーター値を変更すると、**TektonConfig** カスタムリソースの値が変更を上書きします。
- **pipelinesTemplates** パラメーターの値は、**clusterTasks** パラメーターの値が **true** の場合のみ **true** に設定できます。

- **enableMetrics** パラメーターが **TektonConfig** カスタムリソースに追加されます。これを使用して、OpenShift Container Platform の Tekton Pipeline の一部であるサービスモニターを無効にすることができます。

```
apiVersion: operator.tekton.dev/v1alpha1
kind: TektonConfig
metadata:
  name: config
spec:
  profile: all
  targetNamespace: openshift-pipelines
  pipeline:
    params:
    - name: enableMetrics
      value: "true"
...
```

- プロセスレベルでメトリクスをキャプチャーする EventListener OpenCensus メトリクスが追加されました。
- トリガーにはラベルセレクターが追加され、ラベルを使用してイベントリスナーのトリガーを設定できるようになりました。
- インターセプターを登録する **ClusterInterceptor** カスタムリソース定義が追加され、プラグインできる新しい **Interceptor** タイプを登録できるようになりました。さらに、以下の関連する変更が行われます。
  - トリガー仕様では、**ref** フィールドが含まれる新しい API を使用してインターセプターを設定し、クラスターインターセプターを参照できます。さらに、**params** フィールドを使用して、処理用のインターセプターに渡すパラメーターを追加することができます。
  - バンドルされたインターセプター CEL、GitHub、GitLab、および BitBucket が移行されました。新しい **ClusterInterceptor** カスタムリソース定義を使用して実装されます。
  - コアインターセプターは新しい形式に移行され、古い構文を使用して作成された新しいトリガーは自動的に新しい **ref** または **params** ベースの構文に切り替わります。
- ログの表示中にタスクまたはステップの名前の接頭辞を無効にするには、**log** コマンドに **--prefix** オプションを使用します。
- 特定のコンポーネントのバージョンを表示するには、**tkn version** コマンドで新しい **--component** フラグを使用します。
- **tkn hub check-upgrade** コマンドが追加され、他のコマンドはパイプラインのバージョンに基づいて変更されます。さらに、カタログ名は **search** コマンドの出力に表示されます。
- 任意のワークスペースのサポートは **start** コマンドに追加されます。

- プラグインが **plugins** ディレクトリーに存在しない場合は、現在のパスで検索されます。
- **tkn start [task | clustertask | pipeline]** コマンドは、対話的に開始し、デフォルトパラメーターが指定されている場合でも **params** 値の入力を求めます。対話式プロンプトを停止するには、コマンドの呼び出し時に **--use-param-defaults** フラグを渡します。以下に例を示します。

```
$ tkn pipeline start build-and-deploy \
    -w name=shared-
workspace,volumeClaimTemplateFile=https://raw.githubusercontent.com/openshift/pipelines-
tutorial/pipelines-1.14/01_pipeline/03_persistent_volume_claim.yaml \
    -p deployment-name=pipelines-vote-api \
    -p git-url=https://github.com/openshift/pipelines-vote-api.git \
    -p IMAGE=image-registry.openshift-image-registry.svc:5000/pipelines-tutorial/pipelines-
vote-api \
    --use-param-defaults
```

- **version** フィールドは **tkn task describe** コマンドに追加されます。
- **TriggerTemplate**、**TriggerBinding**、**ClusterTriggerBinding**、**Eventlistener** などのリソースを自動的に選択するオプションのいずれか 1 つが存在する場合は、**describe** コマンドに追加されます。
- **tkn pr describe** コマンドでは、省略されたタスクのセクションが追加されます。
- **tkn clustertask logs** のサポートが追加されました。
- **config.yaml** からの YAML マージおよび変数は削除されます。さらに、**release.yaml** ファイルは、**kustomize** や **ytt** などのツールでより簡単に消費されるようになりました。
- ドット文字 (".") を含むリソース名のサポートが追加されました。
- **PodTemplate** 仕様の **hostAliases** 配列が、ホスト名解決の Pod レベルの上書きに追加されます。これには、**/etc/hosts** ファイルを変更します。
- タスクのアグリゲート実行ステータスにアクセスするために、変数 **$(tasks.status)** が導入されました。
- Windows のエントリーポイントバイナリービルドが追加されます。

1.12.3. 非推奨の機能

- **when** 式では、PascalCase で記述されたフィールドのサポートが削除されます。**when** 式は、小文字で記述されたフィールドのみをサポートします。

注記

Tekton Pipelines **v0.16** (Operator **v1.2.x**) の **when** 式のあるパイプラインを適用している場合は、これを再度適用する必要があります。

- Red Hat OpenShift Pipelines Operator を **v1.5** にアップグレードする場合、**openshift-client** および **openshift-client-v-1-5-0** クラスタータスクには **SCRIPT** パラメーターがあります。ただし、**ARGS** パラメーターおよび **git** リソースは **openshift-client** クラスタータスクの仕様から削除されます。これは重大な変更であり、**ClusterTask** リソースの **name** フィールドに特定のバージョンのないクラスタータスクがシームレスにアップグレードされます。

パイプラインの実行が中断しないようにするには、アップグレード後に **SCRIPT** パラメーターを使用します。これは、**ARGS** パラメーターで以前に指定された値がクラスタータスクの **SCRIPT** パラメーターに移動するためです。以下に例を示します。

```
...
- name: deploy
  params:
  - name: SCRIPT
    value: oc rollout status <deployment-name>
  runAfter:
    - build
  taskRef:
    kind: ClusterTask
    name: openshift-client
...
```

- Red Hat OpenShift Pipelines Operator **v1.4** から **v1.5** にアップグレードする場合は、**TektonConfig** カスタムリソースがインストールされるプロファイル名が変更になりました。

表1.4 TektonConfig カスタムリソースのプロファイル

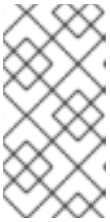
Pipelines 1.5 のプロファイル	Pipelines 1.4 の対応するプロファイル	インストールされた Tekton コンポーネント
すべて (**デフォルトプロファイル**)	すべて (**デフォルトプロファイル**)	Pipelines、Triggers、Add-ons
Basic	デフォルト	Pipeline、Triggers
Lite	Basic	Pipelines

注記

TektonConfig カスタムリソースの **config** インスタンスで **profile: all** を使用した場合は、リソース仕様を変更する必要はありません。

ただし、インストールされた Operator がアップグレード前に Default または Basic プロファイルのいずれかにある場合は、アップグレード後に **TektonConfig** カスタムリソースの **config** インスタンスを編集する必要があります。たとえば、アップグレードの前に設定が **profile: basic** の場合は、Pipeline 1.5 へのアップグレード後にこれが **profile: lite** であることを確認します。

- **disable-home-env-overwrite** フィールドおよび **disable-working-dir-overwrite** フィールドは非推奨となり、今後のリリースで削除されます。本リリースでは、後方互換性のために、これらのフラグのデフォルト値が **true** に設定されます。



注記

次回のリリース (Red Hat OpenShift Pipelines 1.6) では、**HOME** 環境変数は自動的に **/tekton/home** に設定されず、デフォルトの作業ディレクトリーはタスク実行の **/workspace** に設定されていません。これらのデフォルトは、この手順のイメージの Dockerfile で設定されているすべての値と競合します。

- **ServiceType** フィールドおよび **podTemplate** フィールドは **EventListener** 仕様から削除されます。
- コントローラーサービスアカウントは、namespace のリスト表示および監視に対してクラスター全体のパーミッションを要求しなくなりました。
- **EventListener** リソースのステータスには、**Ready** という新規条件があります。

注記

今後、**EventListener** リソースの他のステータス条件は非推奨となり、**Ready** ステータス条件が優先されます。

- **EventListener** 応答の **eventListener** フィールドおよび **namespace** フィールドは非推奨になりました。代わりに **eventListenerUID** フィールドを使用してください。
- **replicas** フィールドは **EventListener** 仕様から非推奨になります。その代わりに、**spec.replicas** フィールドは **KubernetesResource** 仕様の **spec.resources.kubernetesResource.replicas** に移動されます。

注記

replicas フィールドは今後のリリースで削除されます。

- コアインターセプターの設定における古い方法は非推奨になりました。ただし、今後のリリースで削除されるまでこれらの作業は継続されます。代わりに、**Trigger** リソースのインターセプターが新しい **ref** および **params** ベースの構文を使用して設定されるようになりました。作成されるデフォルトの Webhook は、新規トリガーの古い構文の使用を新規構文に自動的に切り替えます。
- **ClusterRoleBinding** リソースに非推奨の **rbac.authorization.k8s.io/v1beta1** ではなく **rbac.authorization.k8s.io/v1** を使用します。
- クラスターロールでは、**serviceaccounts**、**secrets**、**configmaps**、**limitranges** などのリソースへのクラスター全体の書き込みアクセスが削除されます。さらに、**deployments**、**statefulsets**、**deployment/finalizers** などのリソースにクラスター全体のアクセスが削除されます。
- **caching.internal.knative.dev** グループの **image** カスタムリソース定義は Tekton により使用されず、本リリースで除外されます。

1.12.4. 既知の問題

- git-cli クラスタータスクは、alpine/git ベースイメージから構築されます。これは、**/root** がユーザーのホームディレクトリーであると想定します。ただし、これは **git-cli** クラスタータスクに明示的に設定されません。

Tekton では、特に指定がない場合は、デフォルトのホームディレクトリーはタスクのすべての手順で **/tekton/home** で上書きされます。ベースイメージの **$HOME** 環境変数を上書きすると、**git-cli** クラスタータスクが失敗します。

この問題は、今後のリリースで修正される予定です。Red Hat OpenShift Pipelines 1.5 以前のバージョンでは、**以下の回避策のいずれかを使用** して、**git-cli** クラスタータスクの失敗を防ぐことができます。

- この手順で **$HOME** 環境変数を設定します。これにより、上書きされないようにします。

  1. [オプション] Operator を使用して Red Hat OpenShift Pipeline をインストールしている場合は、**git-cli** クラスタータスクを別のタスクにクローンします。このアプローチにより、Operator はクラスタータスクに加えられた変更を上書きしないようにします。

  2. **oc edit clustertasks git-cli** コマンドを実行します。

  3. 予想される **HOME** 環境変数をステップの YAML に追加します。

```
...
steps:
  - name: git
    env:
    - name: HOME
      value: /root
    image: $(params.BASE_IMAGE)
    workingDir: $(workspaces.source.path)
...
```

警告

オペレーターがインストールした Red Hat OpenShift Pipelines の場合、**HOME** 環境変数を変更する前に **git-cli** クラスタータスクを別のタスクに複製しないと、Operator の調整中に変更が上書きされます。

- **feature-flags** 設定マップで **HOME** 環境変数の上書きを無効にします。

  1. **oc edit -n openshift-pipelines configmap feature-flags** コマンドを実行します。

  2. **disable-home-env-overwrite** フラグの値を **true** に設定します。

警告

- Operator を使用して Red Hat OpenShift Pipelines をインストールしている場合、変更は Operator の調整時に上書きされます。
- **disable-home-env-overwrite** フラグのデフォルト値を変更すると、すべてのタスクのデフォルトの動作を変更するため、他のタスクやクラスタータスクが破損する可能性があります。

  - パイプラインのデフォルトサービスアカウントが使用される場合に **HOME** 環境変数の上書きを行うため、**git-cli** クラスタータスクに別のサービスアカウントを使用します。

    1. 新規のサービスアカウントを作成します。
    2. 作成したサービスアカウントに Git シークレットをリンクします。
    3. タスクまたはパイプラインの実行中にサービスアカウントを使用します。

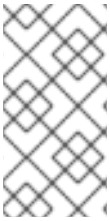
- IBM Power Systems、IBM Z、および LinuxONE では、**s2i-dotnet** クラスタータスクと **tkn hub** コマンドはサポートされません。
- Maven および Jib Maven クラスタータスクを実行する場合には、デフォルトのコンテナーイメージは Intel(x86) アーキテクチャーでのみサポートされます。したがって、IBM Power Systems(ppc64le)、IBM Z、および LinuxONE(s390x) クラスターではタスクに失敗します。回避策として、**MAVEN_IMAGE** パラメーターの値を **maven:3.6.3-adoptopenjdk-11** に設定すると、カスタムイメージを指定できます。

1.12.5. 修正された問題

- **dag** タスクの **when** 式は、他のタスクの実行ステータス (**$(tasks.<pipelineTask>.status)**) にアクセスするコンテキスト変数を指定できません。
- **PipelineRun** リソースがすぐに削除されてから再作成される状況で、**volumeClaimTemplate** PVC を削除することにより作成される競合状態を回避するのに役立つため、所有者名の代わりに所有者 UID を使用します。
- root 以外のユーザーによってトリガーされる **build-base** イメージの **pullrequest-init** に新しい Dockerfile が追加されます。
- パイプラインまたはタスクが **-f** オプションで実行され、その定義の **param** に **type** が定義されていない場合は、パイプラインまたはタスク実行が失敗する代わりに検証エラーが生成されます。
- **tkn start [task | pipeline | clustertask]** コマンドの場合は、**--workspace** フラグの説明に一貫性が保たれました。
- パラメーターを解析する際に、空の配列が発生すると、対応する対話的なヘルプが空の文字列として表示されるようになりました。

1.13. RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY (GA) 1.4 のリリースノート

Red Hat OpenShift Pipelines General Availability (GA) 1.4 が OpenShift Container Platform 4.7 で利用可能になりました。



注記

stable および preview Operator チャネルのほかに、Red Hat OpenShift Pipelines Operator 1.4.0 には ocp-4.6、ocp-4.5、および ocp-4.4 の非推奨チャネルが同梱されます。これらの非推奨チャネルおよびそれらのサポートは、Red Hat OpenShift Pipelines の以下のリリースで削除されます。

1.13.1. 互換性およびサポート表

現在、今回のリリースに含まれる機能には テクノロジープレビュー のものがあります。これらの実験的機能は、実稼働環境での使用を目的としていません。

以下の表では、機能は以下のステータスでマークされています。

TP	テクノロジープレビュー
GA	一般公開 (GA)

これらの機能に関しては、Red Hat カスタマーポータルの以下のサポート範囲を参照してください。

表1.5 互換性およびサポート表

機能	バージョン	サポートステータス
Pipelines	0.22	GA
CLI	0.17	GA
カタログ	0.22	GA
トリガー	0.12	TP
パイプラインリソース	-	TP

質問やフィードバックについては、製品チームに pipelines-interest@redhat.com 宛のメールを送信してください。

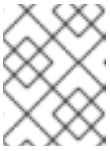
1.13.2. 新機能

以下のセクションでは、修正および安定性の面での改善点に加え、OpenShift Pipelines 1.4 の主な新機能について説明します。

- カスタムタスクには、以下の機能強化が含まれます。

- パイプラインの結果として、カスタムタスクで生成される結果を参照できるようになりました。
- カスタムタスクはワークスペース、サービスアカウント、および Pod テンプレートを使用して、より複雑なカスタムタスクをビルドできるようになりました。

- **finally** タスクには、以下の機能強化が含まれます。
  - **when** 式は **最後** のタスクでサポートされます。これにより、効率的に保護された実行が可能になり、タスクの再利用性が向上します。
  - **finally** タスクは、同じパイプライン内のタスクの結果を使用するように設定できます。



注記

when 式および **finally** タスクのサポートは OpenShift Container Platform 4.7 Web コンソールでは利用できません。

- **dockercfg** または **dockerconfigjson** タイプの複数のシークレットのサポートがランタイム時に認証用に追加されました。
- **git-clone** タスクでスパースチェックをサポートする機能が追加されました。これにより、ローカルコピーとしてリポジトリーのサブセットのみをクローンすることができ、これはクローン作成したリポジトリーのサイズを制限するのに便利です。
- 実際に起動せずに、パイプライン実行を保留中の状態で作成できます。負荷が大きいクラスターでは、これにより、Operator はパイプライン実行の開始時間を制御することができます。
- コントローラー用に **SYSTEM_NAMESPACE** 環境変数を手動で設定していることを確認します。これは以前はデフォルトで設定されていました。
- root 以外のユーザーがパイプラインのビルドベースイメージに追加され、**git-init** がリポジトリーのクローンを root 以外のユーザーとして作成できるようになりました。
- パイプライン実行の開始前に解決されたリソース間で依存関係を検証するサポートが追加されています。パイプラインのすべての結果変数は有効でなければならず、パイプラインからのオプションのワークスペースは、パイプライン実行の開始に使用することが予想されているタスクにのみ渡すことができます。
- コントローラーおよび Webhook は root 以外のグループとして実行され、それらの必要以上の機能は削除され、よりセキュアになりました。
- **tkn pr logs** コマンドを使用して、再試行されたタスク実行のログストリームを表示できます。
- **tkn tr delete** コマンドで **--clustertask** オプションを使用して、特定のクラスタータスクに関連付けられたすべてのタスク実行を削除できます。
- **EventListener** リソースでの Knative サービスのサポートは、新規の **customResource** フィールドを導入して追加されます。
- イベントペイロードが JSON 形式を使用しない場合にエラーメッセージが表示されます。
- GitLab、BitBucket、GitHub などのソース制御インターセプターは、新規の **InterceptorRequest** または **InterceptorResponse** を使用できるようになりました。
- 新しい CEL 関数の **marshalJSON** が実装され、JSON オブジェクトまたは配列を文字列にエンコードできます。

- CEL およびソース制御コアインターセプターを提供する HTTP ハンドラーが追加されました。これは、**tekton-pipelines** namespace にデプロイされる単一の HTTP サーバーに 4 つのコアインターセプターをパッケージ化します。**EventListener** オブジェクトは、HTTP サーバー経由でイベントをインターセプターに転送します。それぞれのインターセプターは異なるパスで利用できます。たとえば、CEL インターセプターは **/cel** パスで利用できます。

- **pipelines-scc** SCC (Security Context Constraint) は、パイプラインのデフォルト **pipeline** サービスアカウントで使用されます。この新規サービスアカウントは **anyuid** と似ていますが、OpenShift Container Platform 4.7 の SCC について YAML に定義されるように若干の違いがあります。

```
fsGroup:
  type: MustRunAs
```

1.13.3. 非推奨の機能

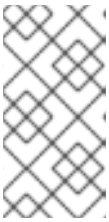
- パイプラインリソースストレージの **build-gcs** サブタイプ、および **gcs-fetcher** イメージは、サポートされていません。

- クラスタータスクの **taskRun** フィールドで、**tekton.dev/task** ラベルが削除されます。

- Webhook の場合、フィールド **admissionReviewVersions** に対応する値 **v1beta1** は削除されます。

- ビルドおよびデプロイ用の **creds-init** ヘルパーイメージが削除されます。

- トリガー仕様およびバインディングでは、**template.ref** が優先されるため、非推奨フィールドの **template.name** が削除されます。**ref** フィールドを使用するには、**eventListener** のすべての定義を更新する必要があります。



注記

OpenShift Pipelines 1.3.x 以前のバージョンから OpenShift Pipelines 1.4.0 にアップグレードすると、**template.name** フィールドが使用できないため、イベントリスナーが壊れます。このような場合、OpenShift Pipelines 1.4.1 を使用して、復元された **template.name** フィールドを利用します。

- **EventListener** カスタムリソース/オブジェクトの場合、**Resource** が優先されるために、**PodTemplate** および **ServiceType** フィールドは非推奨になりました。

- 非推奨の仕様スタイルの埋め込みバインディングは削除されています。

- **spec** フィールドは **triggerSpecBinding** から削除されています。

- イベント ID 表現は、5 文字のランダムな文字列から UUID に変更されています。

1.13.4. 既知の問題

- **Developer** パースペクティブでは、Pipeline メトリクスおよびトリガー機能は OpenShift Container Platform 4.7.6 以降のバージョンでのみ利用できます。

- IBM Power Systems、IBM Z、および LinuxONE では、**tkn hub** コマンドはサポートされません。

- IBM Power Systems (ppc64le)、IBM Z、および LinuxONE (s390x) クラスターで Maven および Jib Maven クラスタータスクを実行する場合、**MAVEN_IMAGE** パラメーターの値を **maven:3.6.3-adoptopenjdk-11** に設定します。

- トリガーは、トリガーバインディングに以下の設定がある場合は、JSON 形式の正しくない処理によって生じるエラーを出力します。

```
params:
  - name: github_json
    value: $(body)
```

この問題を解決するには、以下を実行します。

  - トリガー v0.11.0 以降を使用している場合、**marshalJSON** 関数を使用して JSON オブジェクトまたは配列を取得し、そのオブジェクトまたは配列の JSON エンコーディングを文字列として返します。

  - 古いバージョンのトリガーを使用している場合は、以下のアノテーションをトリガーテンプレートに追加します。

```
annotations:
  triggers.tekton.dev/old-escape-quotes: "true"
```

- OpenShift Pipelines 1.3.x から 1.4.x にアップグレードする場合、ルートを再作成する必要があります。

1.13.5. 修正された問題

- 以前のバージョンでは、**tekton.dev/task** ラベルがクラスタータスクのタスク実行から削除され、**tekton.dev/clusterTask ラベル** が導入されました。この変更により生じる問題は 、**clustertask describe** および **delete** コマンドを修正して解決されています。さらに、タスクの **lastrun** 機能は変更され、古いバージョンのパイプラインでタスクとクラスタータスクの両方のタスク実行に適用される **tekton.dev/task** ラベルの問題を修正できるになりました。

- 対話的な **tkn pipeline start pipelinename** を実行する場合、**PipelineResource** が対話的に作成されます。**tkn p start** コマンドは、リソースのステータスが **nil** ではない場合にリソースのステータスを出力します。

- 以前のバージョンでは、**tekton.dev/task=name** ラベルは、クラスタータスクから作成されるタスク実行から削除されました。今回の修正により、**--last** フラグの指定される **tkn clustertask start** コマンドが変更され、作成されたタスク実行で **tekton.dev/task=name** ラベルの有無がチェックされるようになりました。

- タスクがインラインのタスク仕様を使用する場合、対応するタスク実行は **tkn pipeline describe** コマンドの実行時にパイプラインに組み込まれ、タスク名は埋め込まれた状態で返されます。

- **tkn version** コマンドは、設定された **kubeConfiguration namespace** やクラスターへのアクセスなしに、インストールされた Tekton CLI ツールのバージョンを表示するように修正されています。

- 引数が予期せずに使用されるか、複数の引数が使用される場合、**tkn completion** コマンドでエラーが発生します。

- 以前のバージョンでは、パイプライン仕様でネスト化された **finally** タスクのあるパイプライン実行は、**v1alpha1** バージョンに変換され、**v1beta1** バージョンに戻されると、それらの

finally タスクを失うことがあります。変換中に発生するこのエラーは修正され、潜在的データ損失を防ぐことができます。**finally** タスクがパイプライン仕様でネスト化されたパイプライン実行はシリアライズされ、アルファバージョンに保存されてデシリアライズは後に実行されるようになりました。

- 以前のバージョンでは、サービスアカウントで **secrets** フィールドに **{}** があると、Pod の生成でエラーが発生しました。空のシークレット名を持つ GET 要求がエラーがリソース名が空ではないことを示すエラーを返すため、タスク実行は **CouldntGetTask** で失敗しました。この問題は、**kubeclient** GET 要求で空のシークレット名を使用しないことで解決されています。

- **v1beta1** API バージョンのあるパイプラインは、**finally** タスクを失うことなく、**v1alpha1** バージョンと共に要求できるようになりました。返される **v1alpha1** バージョンを適用すると、リソースが **v1beta1** として保存され、**finally** セクションがその元の状態に戻ります。

- 以前のバージョンでは、コントローラーの **selfLink** フィールドが設定されていないと、Kubernetes v1.20 クラスターでエラーが発生しました。一時的な修正として、**CloudEvent** ソースフィールドは、自動設定される **selfLink** フィールドの値なしに現在のソース URI に一致する値に設定されます。

- 以前のバージョンでは、**gcr.io** などのドットの付いたシークレット名により、タスク実行の作成が失敗しました。これは、シークレット名がボリュームマウント名の一部として内部で使用されるために生じました。ボリュームマウント名は RFC1123 DNS ラベルに準拠し、名前の一部として使用されるドットを許可しません。この問題は、ドットをダッシュに置き換えることで解決し、これにより名前の読み取りが可能になりました。

- コンテキスト変数は、**finally** タスクで検証されるようになりました。

- 以前のバージョンでは、タスク実行リコンサイラーが渡され、作成した Pod の名前を含む直前のステータス更新を持たないタスク実行があると、タスク実行リコンサイラーはタスク実行に関連付けられた Pod をリスト表示しました。タスク実行リコンサイラーは、Pod を検索するために、Pod に伝播されるタスク実行のラベルを使用しました。タスク実行の実行中にこれらのラベルを変更すると、コードが既存の Pod を見つけることができませんでした。その結果、重複した Pod が作成されました。この問題は、Pod の検索時に **tekton.dev/taskRun** の Tekton で制御されるラベルのみを使用するようにタスク実行リコンサイラーを変更することで修正されています。

- 以前のバージョンでは、パイプラインがオプションのワークスペースを受け入れ、これをパイプラインタスクに渡すと、パイプライン実行リコンサイラーは、ワークスペースが提供されておらず、欠落しているワークスペースのバインディングがオプションのワークスペースについて有効な場合でも、エラーを出して停止しました。この問題は、オプションのワークスペースが指定されていない場合でも、パイプライン実行リコンサイラーがタスク実行の作成に失敗しないようにすることで修正されています。

- ステップのステータスの並び順は、ステップコンテナーの順序と一致します。

- 以前のバージョンでは、Pod で **CreateContainerConfigError** の理由が出されると、タスク実行のステータスは **unknown** に設定されました。これは、タスクおよびパイプラインが Pod がタイムアウトするまで実行されることを意味しました。この問題は、Pod で **CreateContainerConfigError** の理由が出される際にタスクを失敗 (failed) として設定できるようにタスク実行ステータスを **false** に設定することで解決されています。

- 以前のバージョンでは、パイプライン実行の完了後に、パイプラインの結果は最初の調整で解決されました。これにより解決が失敗し、パイプライン実行の **Succeeded** 状態が上書きされる可能性がありました。その結果、最終のステータス情報が失われ、パイプライン実行の状態を監視するすべてのサービスに混乱を生じさせる可能性がありました。この問題は、パイプライン実行が **Succeeded** または **True** 状態になる際に、パイプラインの結果の解決を調整の最後に移行することにで解決されました。

- 実行ステータス変数が検証されるようになりました。これにより、実行ステータスにアクセスするためのコンテキスト変数の検証中に、タスク結果が検証されることを防ぐことができます。

- 以前のバージョンでは、無効な変数を含むパイプラインの結果は、変数のリテラル式はそのままの状態でパイプライン実行に追加されます。そのため、結果が正しく設定されているかどうかを評価することは容易ではありませんでした。この問題は、失敗したタスク実行を参照するパイプライン実行結果でフィルタリングすることで解決されています。無効な変数を含むパイプラインの結果は、パイプライン実行によって出されなくなりました。

- **tkn eventlistener describe** コマンドは、テンプレートなしでクラッシュを回避できるように修正されています。また、トリガーの参照に関する情報も表示します。

- OpenShift Pipelines 1.3.x 以前のバージョンから OpenShift Pipelines 1.4.0 にアップグレードすると、**template.name** が利用できないため、イベントリスナーが壊れます。OpenShift Pipelines 1.4.1 では、**template.name が** 復元され、トリガーのイベントリスナーが壊れないようになりました。

- OpenShift Pipelines 1.4.1 では、**ConsoleQuickStart** カスタムリソースが更新され、OpenShift Container Platform 4.7 の機能および動作と一致するようになりました。

1.14. RED HAT OPENSHIFT PIPELINES テクノロジープレビュー 1.3 のリリースノート

1.14.1. 新機能

Red Hat OpenShift Pipelines テクノロジープレビュー (TP) 1.3 が OpenShift Container Platform 4.7 で利用可能になりました。Red Hat OpenShift Pipelines TP 1.3 が以下をサポートするように更新されています。

- Tekton Pipelines 0.19.0

- Tekton **tkn** CLI 0.15.0

- Tekton Triggers 0.10.2

- Tekton Catalog 0.19.0 をベースとするクラスタータスク

- OpenShift Container Platform 4.7 での IBM Power Systems

- OpenShift Container Platform 4.7 での IBM Z および LinuxONE

以下のセクションでは、修正および安定性の面での改善点に加え、OpenShift Pipelines 1.3 の主な新機能について説明します。

1.14.1.1. Pipelines

- S2I や Buildah タスクなどのイメージをビルドするタスクが、イメージの SHA を含むビルドされたイメージの URL を生成するようになりました。

- **Condition** カスタムリソース定義 (CRD) が非推奨となっているため、カスタムタスクを参照するパイプラインタスクの条件は許可されません。

- **spec.steps[].imagePullPolicy** および **spec.sidecar[].imagePullPolicy** フィールドの **Task** CRD に変数の拡張が追加されました。

- **disable-creds-init** feature-flag を **true** に設定すると、Tekton のビルトイン認証情報メカニズムを無効にすることができます。
- 解決済みの When 式は、**PipelineRun** 設定の **Status** フィールドの **Skipped Tasks** および **Task Runs** セクションにリスト表示されるようになりました。
- **git init** コマンドが、再帰的なサブモジュールのクローンを作成できるようになりました。
- **Task** CR の作成者は、**Task** 仕様のステップのタイムアウトを指定できるようになりました。
- エントリーポイントイメージを **distroless/static:nonroot** イメージにベースとして作成し、ベースイメージに存在する **cp** コマンドを使用せずに、これを宛先にコピーするモードを許可できるようになりました。
- Git SSH シークレットの既知のホストの省略を許可しないように、設定フラグ **require-git-ssh-secret-known-hosts** を使用できるようになりました。フラグ値が **true** に設定されている場合には、Git SSH シークレットに **known_host** フィールドを含める必要があります。フラグのデフォルト値は **false** です。
- オプションのワークスペースの概念が導入されました。タスクまたはパイプラインはワークスペースオプションを宣言し、その存在に基づいて動作を条件的に変更する可能性があります。タスク実行またはパイプライン実行により、そのワークスペースが省略され、タスクまたはパイプラインの動作が変更される可能性があります。デフォルトのタスク実行ワークスペースは、省略されたオプションのワークスペースの代わりに追加されることはありません。
- Tekton の認証情報の初期化により、SSH 以外の URL で使用する SSH 認証情報が検出されるほか、Git パイプラインリソースでは SSH URL で使用する http 認証情報が検出され、Step コンテナーで警告がログに記録されるようになりました。
- タスク実行コントローラーは、Pod テンプレートで指定されたアフィニティーがアフィニティーアシスタントによって上書きされる場合に警告イベントを生成します。
- タスク実行リコンサイラーは、タスク実行が完了すると生成されるクラウドイベントのメトリクスを記録するようになりました。これには再試行が含まれます。

1.14.1.2. Pipelines CLI

- **--no-headers flag** のサポートが、次のコマンドに追加されました: **tkn condition list**、**tkn triggerbinding list**、**tkn eventlistener list**、**tkn clustertask list**、**tkn clustertriggerbinding list**
- 併用した場合、**--last** または **--use** オプションは、**--prefix-name** および **--timeout** オプションを上書きします。
- **tkn eventlistener logs** コマンドが追加され、**EventListener** ログが表示されるようになりました。
- **tekton hub** コマンドは **tkn** CLI に統合されるようになりました。
- **--nocolour** オプションは **--no-color** に変更されました。
- **--all-namespaces** フラグは、次のコマンドに追加されました: **tkn triggertemplate list**、**tkn condition list**、**tkn triggerbinding list**、**tkn eventlistener list**

1.14.1.3. トリガー

- **EventListener** テンプレートでリソース情報を指定できるようになりました。

- すべてのトリガーリソースの **get** 動詞に加えて、**EventListener** サービスアカウントに **list** および **watch** 動詞が設定されることが必須になりました。これにより、**Listers** を使用して **EventListener**、 **Trigger**、**TriggerBinding**、**TriggerTemplate**、および **ClusterTriggerBinding** リソースからデータを取得することができます。この機能を使用して、複数のインフォーマーを指定するのではなく **Sink** オブジェクトを作成し、API サーバーを直接呼び出すことができます。

- イミュータブルな入力イベント本体をサポートする新たな **Interceptor** インターフェイスが追加されました。インターセプターはデータまたはフィールドを新しい **extensions** フィールドに追加できるようになり、入力本体を変更できなくなったことでイミュータブルとなりました。CEL インターセプターはこの新たな **Interceptor** インターフェイスを使用します。

- **namespaceSelector** フィールドは **EventListener** リソースに追加されます。これを使用して、**EventListener** リソースがイベント処理用に **Trigger** オブジェクトを取得できる namespace を指定します。**namespaceSelector** フィールドを使用するには、**EventListener** のサービスアカウントにクラスターロールが必要です。

- トリガー **EventListener** リソースは、**eventlistener** Pod へのエンドツーエンドのセキュアな接続をサポートするようになりました。

- **"** を **\"** に置き換えることで、**TriggerTemplates** リソースのエスケープパラメーター動作が削除されました。

- Kubernetes リソースをサポートする新規 **resources** フィールドは、**EventListener** 仕様の一部として導入されます。

- ASCII 文字列の大文字と小文字へのサポートが含まれる CEL インターセプターの新機能が追加されました。

- **TriggerBinding** リソースは、トリガーの **name** および **value** フィールドを使用するか、イベントリスナーを使用して埋め込むことができます。

- **PodSecurityPolicy** 設定は、制限された環境で実行されるように更新されます。これにより、コンテナーは root 以外のユーザーとして実行する必要があります。さらに、Pod セキュリティーポリシーを使用するためのロールベースのアクセス制御は、クラスタースコープから namespace スコープに移行されます。これにより、トリガーは namespace に関連しない他の Pod セキュリティーポリシーを使用することができません。

- 埋め込みトリガーテンプレートのサポートが追加されました。**name** フィールドを使用して埋め込みテンプレートを参照するか、**spec** フィールド内にテンプレートを埋め込むことができます。

1.14.2. 非推奨の機能

- **PipelineResources** CRD を使用する Pipeline テンプレートは非推奨となり、今後のリリースで削除されます。

- **template.ref** フィールドが優先されるため、**template.name** フィールドは非推奨となり、今後のリリースで削除されます。

- **--check** コマンドの短縮形である **-c** が削除されました。さらに、グローバル **tkn** フラグが **version** コマンドに追加されます。

1.14.3. 既知の問題

- CEL オーバーレイは、受信イベント本体を変更する代わりに、フィールドを新しい最上位の

extensions 関数に追加します。**TriggerBinding** リソースは、**$(extensions.<key>)** 構文を使用して、この新しい **extensions** 関数内の値にアクセスできます。**$(body.<overlay-key>)** の代わりに **$(extensions.<key>)** 構文を使用するようにバインディングを更新します。

- **"** を **\"** に置き換えることで、エスケープパラメーター動作が削除されました。古いエスケープパラメーターの動作を保持する必要がある場合は、**tekton.dev/old-escape-quotes: true"** アノテーションを **TriggerTemplate** 仕様に追加します。

- **TriggerBinding** リソースは、トリガーまたはイベントリスナー内の **name** および **value** フィールドを使用して組み込みことができます。ただし、単一のバインディングに **name** および **ref** フィールドの両方を指定することはできません。**ref** フィールドを使用して **TriggerBinding** リソースおよび埋め込みバインディングの **name** フィールドを参照します。

- インターセプターは、**EventListener** リソースの namespace 外で **secret** の参照を試行することはできません。シークレットを `EventListener` の namespace に含める必要があります。

- Trigger 0.9.0 以降では、本体またはヘッダーベースの **TriggerBinding** パラメーターが見つからないか、イベントペイロードで形式が正しくない場合に、エラーを表示する代わりにデフォルト値が使用されます。

- JSON アノテーションを修正するには、Tekton および Pipelines 0.16.x を使用して **WhenExpression** オブジェクトで作成されたタスクおよびパイプラインを再適用する必要があります。

- パイプラインがオプションのワークスペースを受け入れ、これをタスクに付与すると、ワークスペースが指定されていない場合はパイプライン実行が停止します。

- 非接続環境で Buildah クラスタータスクを使用するには、Dockerfile が内部イメージストリームをベースイメージとして使用していることを確認してから、これを S2I クラスタータスクと同じ方法で使用します。

1.14.4. 修正された問題

- CEL インターセプターによって追加された拡張機能は、イベント本体内に **Extensions** フィールドを追加して Webhook インターセプターに渡されます。

- ログリーダーのアクティビティータイムアウトは、**LogOptions** フィールドを使用して設定できるようになりました。ただし、10 秒のタイムアウトのデフォルト動作は保持されます。

- **log** コマンドは、タスク実行またはパイプライン実行が完了したときに **--follow** フラグを無視し、ライブログではなく利用可能なログを読み取ります。

- 以下の Tekton リソースへの参照:
EventListener、**TriggerBinding**、**ClusterTriggerBinding**、**Condition**、および **TriggerTemplate** は、**tkn** コマンドのすべてのユーザーに表示されるメッセージで標準化され、一貫性を保つようになりました。

- 以前は、**--use-taskrun <canceled-task-run-name>**、**--use-pipelinerun <canceled-pipeline-run-name>** または **--last** フラグを使用してキャンセルされたタスク実行またはパイプライン実行を開始した場合、新規の実行はキャンセルされました。このバグは修正されています。

- **tkn pr desc** コマンドが強化され、パイプラインが各種の状態で実行された場合に失敗しなくなりました。

- **--task** オプションで **tkn tr delete** コマンドを使用してタスク実行を削除し、クラスタータスクが同じ名前で存在する場合、クラスタータスクのタスク実行も削除されます。回避策として、**TaskRefKind** フィールドを使用して、タスク実行をフィルタリングします。

- **tkn triggertemplate describe** コマンドは、出力内の **apiVersion** 値の一部のみを表示します。たとえば、**triggers.tekton.dev/v1alpha1** ではなく、**triggers.tekton.dev** のみが表示されました。このバグは修正されています。

- 特定の条件下で Webhook はリースの取得に失敗し、正常に機能しません。このバグは修正されています。

- v0.16.3 で作成した When 式を持つパイプラインは、v0.17.1 以降で実行できるようになりました。アップグレード後に、アノテーションの最初の大文字と小文字の両方がサポートされるようになったため、以前のバージョンで作成されたパイプライン定義を再適用する必要はありません。

- デフォルトでは、**leader-election-ha** フィールドが高可用性に対して有効にされるようになりました。コントローラーフラグ **disable-ha** を **true** に設定すると、高可用性サポートが無効になります。

- 重複したクラウドイベントに関する問題が修正されています。クラウドイベントは、条件が状態、理由、またはメッセージを変更する場合にのみ送信されるようになりました。

- サービスアカウント名が **PipelineRun** または **TaskRun** 仕様にない場合、コントローラーは **config-defaults** 設定マップからサービスアカウント名を使用します。サービスアカウント名が **config-defaults** 設定マップにもない場合、コントローラーはこれを仕様で **default** に設定するようになりました。

- アフィニティーアシスタントとの互換性の検証は、同じ永続ボリューム要求 (PVC) が複数のワークスペースに使用される場合にサポートされるようになりましたが、サブパスは異なります。

1.15. RED HAT OPENSHIFT PIPELINES テクノロジープレビュー 1.2 のリリースノート

1.15.1. 新機能

Red Hat OpenShift Pipelines テクノロジープレビュー (TP) 1.2 が OpenShift Container Platform 4.6 で利用可能になりました。Red Hat OpenShift Pipelines TP 1.2 が以下をサポートするように更新されています。

- Tekton Pipelines 0.16.3

- Tekton **tkn** CLI 0.13.1

- Tekton Triggers 0.8.1

- Tekton Catalog 0.16 をベースとするクラスタータスク

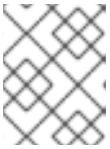
- OpenShift Container Platform 4.6 での IBM Power Systems

- OpenShift Container Platform 4.6 での IBM Z および LinuxONE

以下では、修正および安定性の面での改善点に加え、OpenShift Pipelines 1.2 の主な新機能について説明します。

1.15.1.1. Pipelines

- Red Hat OpenShift Pipelines のリリースでは、非接続インストールのサポートが追加されました。



注記

制限された環境でのインストールは現時点で、IBM Power Systems、IBM Z、および LinuxONE ではサポートされていません。

- **conditions** リソースの代わりに **when** フィールドを使用して、特定の条件が満たされる場合にのみタスクを実行できるようになりました。**WhenExpression** の主なコンポーネントは **Input**、 **Operator**、および **Values** です。すべての When 式が **True** に評価されると、タスクが実行されます。When 式のいずれかが **False** に評価されると、タスクはスキップされます。

- ステップのステータスは、タスクの実行がキャンセルまたはタイムアウトすると更新されるようになりました。

- **git-init** が使用するベースイメージをビルドするために、Git Large File Storage (LFS) のサポートが利用できるようになりました。

- **taskSpec** フィールドを使用して、タスクがパイプラインに組み込まれる際に、ラベルやアノテーションなどのメタデータを指定できるようになりました。

- クラウドイベントがパイプラインの実行でサポートされるようになりました。**backoff** を使用した再試行が、クラウドイベントパイプラインリソースによって送信されるクラウドイベントに対して有効になりました。

- **Task** リソースが宣言するものの、**TaskRun** リソースが明示的に指定しないワークスペースのデフォルトの **Workspace** 設定を設定できるようになりました。

- サポートは、**PipelineRun** namespace および **TaskRun** namespace の namespace 変数の補間に利用できます。

- **TaskRun** オブジェクトの検証が追加され、**TaskRun** リソースが Affinity Assistant に関連付けられる際に複数の永続ボリューム要求 (PVC) ワークスペースが使用されていないことを確認するようになりました。複数の永続ボリューム要求 (PVC) ワークスペースが使用されていると、タスクの実行は **TaskRunValidationFailed** の状態で失敗します。デフォルトで、Affinity Assistant は Red Hat OpenShift Pipelines で無効にされているため、これを使用できるように有効にする必要があります。

1.15.1.2. Pipelines CLI

- **tkn task describe**、**tkn taskrun describe**、**tkn clustertask describe**、**tkn pipeline describe**、および **tkn pipelinerun describe** コマンドが以下を実行するようになりました。
  - **Task**、**TaskRun**、**ClusterTask**、**Pipeline** および **PipelineRun** リソースのいずれかが 1 つしかない場合、それぞれを自動的に選択します。
  - 出力に **Task**、**TaskRun**、**ClusterTask**、**Pipeline** および **PipelineRun** リソースの結果をそれぞれ表示します。
  - 出力に **Task**、**TaskRun**、**ClusterTask**、**Pipeline** および **PipelineRun** リソースで宣言されたワークスペースをそれぞれ表示します。

- **tkn clustertask start** コマンドに **--prefix-name** オプションを指定して、タスク実行の名前に接頭辞を指定できるようになりました。

- インタラクティブモードのサポートが **tkn clustertask start** コマンドに提供されるようになりました。

- **TaskRun** および **PipelineRun** オブジェクトのローカルまたはリモートファイル定義を使用して、パイプラインでサポートされる **PodTemplate** プロパティーを指定できるようになりました。

- **--use-params-defaults** オプションを **tkn clustertask start** コマンドに指定して、**ClusterTask** 設定に設定したデフォルト値を使用して、タスク実行を作成できるようになりました。

- **tkn pipeline start** コマンドの **--use-param-defaults** フラグで、デフォルトの値が一部のパラメーターに指定されていない場合に対話モードをプロンプトで表示するようになりました。

1.15.1.3. トリガー

- YAML 文字列を文字列のマップに解析するために、**parseYAML** という名前の Common Expression Language (CEL) 関数が追加されました。

- 式を評価する際や、評価環境を作成するためにフック本体を解析する際に、CEL 式の解析を行うエラーメッセージの詳細度が上がりました。

- ブール値とマップが CEL オーバーレイメカニズムで式の値として使用されている場合に、それらをマーシャリングするためのサポートが利用できるようになりました。

- 以下のフィールドが **EventListener** オブジェクトに追加されました。

  - **replicas** フィールドは、YAML ファイルのレプリカ数を指定して、イベントリスナーが複数の Pod を実行できるようにします。

  - **NodeSelector** フィールドでは、**EventListener** オブジェクトがイベントリスナー Pod を特定のノードにスケジュールできるようにします。

- Webhook インターセプターは **EventListener-Request-URL** ヘッダーを解析し、イベントリスナーによって処理される元のリクエスト URL からパラメーターを抽出できるようになりました。

- イベントリスナーからのアノテーションがデプロイメント、サービス、およびその他の Pod に伝播できるようになりました。サービスまたはデプロイメントのカスタムアノテーションは上書きされるため、イベントリスナーアノテーションに追加して伝播できるようにする必要があります。

- **EventListener** 仕様のレプリカの適切な検証が、ユーザーが **spec.replicas** 値を **negative** または **zero** として指定する場合に利用できるようになりました。

- **TriggerCRD** オブジェクトを、**TriggerRef** フィールドを使用して参照として **EventListener** 仕様内に指定し、**TriggerCRD** オブジェクトを別個に作成してから、これを **EventListener** 仕様内でバインドできるようになりました。

- **TriggerCRD** オブジェクトの検証およびデフォルト値が利用可能になりした。

1.15.2. 非推奨の機能

- **$(params)** パラメーターは **triggertemplate** リソースから削除され、**$(tt.params)** に置き換えられ、これにより **resourcetemplate** と **triggertemplate** パラメーター間の混乱が生じなくなります。

- オプションの **EventListenerTrigger** ベースの認証レベルの **ServiceAccount** 参照が

ServiceAccountName 文字列へのオブジェクト参照から変更されました。これにより、**ServiceAccount** 参照が **EventListenerTrigger** オブジェクトと同じ namespace に置かれるようになりました。

- **Conditions** カスタムリソース定義 (CRD) は非推奨となり、代わりに **WhenExpressions** CRD が使用されます。
- **PipelineRun.Spec.ServiceAccountNames** オブジェクトは非推奨となり、**PipelineRun.Spec.TaskRunSpec[].ServiceAccountName** オブジェクトによって置き換えられます。

1.15.3. 既知の問題

- Red Hat OpenShift Pipelines のリリースでは、非接続インストールのサポートが追加されました。ただし、クラスタータスクで使用される一部のイメージは、非接続クラスターで動作するようにミラーリングする必要があります。
- **openshift** namespace のパイプラインは、Red Hat OpenShift Pipelines Operator のアンインストール後に削除されません。**oc delete pipelines -n openshift --all** コマンドを使用してパイプラインを削除します。
- Red Hat OpenShift Pipelines Operator をアンインストールしても、イベントリスナーは削除されません。
  回避策として、**EventListener** および **Pod** CRD を削除するには、以下を実行します。

  1. **EventListener** オブジェクトを **foregroundDeletion** ファイナライザーで編集します。

     ```
     $ oc patch el/<eventlistener_name> -p '{"metadata":{"finalizers":["foregroundDeletion"]}}'
     --type=merge
     ```

     以下に例を示します。

     ```
     $ oc patch el/github-listener-interceptor -p '{"metadata":{"finalizers":
     ["foregroundDeletion"]}}' --type=merge
     ```

  2. **EventListener** CRD を削除します。

     ```
     $ oc patch crd/eventlisteners.triggers.tekton.dev -p '{"metadata":{"finalizers":[]}}' --
     type=merge
     ```

- IBM Power Systems (ppc64le) または IBM Z (s390x) クラスターでコマンド仕様なしにマルチアーキテクチャーコンテナーイメージタスクを実行すると、**TaskRun** リソースは以下のエラーを出して失敗します。

  ```
  Error executing command: fork/exec /bin/bash: exec format error
  ```

  回避策として、アーキテクチャー固有のコンテナーイメージを使用するか、正しいアーキテクチャーを参照する sha256 ダイジェストを指定します。sha256 ダイジェストを取得するには、以下を実行します。

  ```
  $ skopeo inspect --raw <image_name>| jq '.manifests[] | select(.platform.architecture == "
  <architecture>") | .digest'
  ```

1.15.4. 修正された問題

- CEL フィルター、Webhook バリデーターのオーバーレイ、およびインターセプターの式を確認するための簡単な構文検証が追加されました。
- Trigger は、基礎となるデプロイメントおよびサービスオブジェクトに設定されたアノテーションを上書きしなくなりました。
- 以前のバージョンでは、イベントリスナーはイベントの受け入れを停止しました。今回の修正により、この問題を解決するために **EventListener** シンクの 120 秒のアイドルタイムアウトが追加されました。
- 以前のバージョンでは、**Failed(Canceled)** 状態でパイプラインの実行を取り消すと、成功のメッセージが表示されました。これは、代わりにエラーが表示されるように修正されました。
- **tkn eventlistener list** コマンドがリスト表示されたイベントリスナーのステータスを提供するようになり、利用可能なイベントリスナーを簡単に特定できるようになりました。
- トリガーがインストールされていない場合や、リソースが見つからない場合に、**triggers list** および **triggers describe** コマンドについて一貫性のあるエラーメッセージが表示されるようになりました。
- 以前のバージョンでは、多くのアイドル接続がクラウドイベントの配信時に増大しました。この問題を修正するために、**DisableKeepAlives: true** パラメーターが **cloudeventclient** 設定に追加されました。新規の接続がすべてのクラウドイベントに設定されます。
- 以前のバージョンでは、特定のタイプの認証情報が指定されていない場合であっても、**creds-init** コードが空のファイルをディスクに書き込みました。今回の修正により、**creds-init** コードが変更され、正しくアノテーションが付けられたシークレットから実際にマウントされた認証情報のみのファイルを書き込むようになりました。

1.16. RED HAT OPENSHIFT PIPELINES テクノロジープレビュー 1.1 のリリースノート

1.16.1. 新機能

Red Hat OpenShift Pipelines テクノロジープレビュー (TP) 1.1 が OpenShift Container Platform 4.5 で利用可能になりました。Red Hat OpenShift Pipelines TP 1.1 が以下をサポートするように更新されています。

- Tekton Pipelines 0.14.3
- Tekton **tkn** CLI 0.11.0
- Tekton Triggers 0.6.1
- Tekton Catalog 0.14 をベースとするクラスタータスク

以下では、修正および安定性の面での改善点に加え、OpenShift Pipelines 1.1 の主な新機能について説明します。

1.16.1.1. Pipelines

- ワークスペースをパイプラインリソースの代わりに使用できるようになりました。パイプラインリソースはデバッグが容易ではなく、スコープの制限があり、タスクの再利用を可能にしな

いため、OpenShift Pipelines ではワークスペースを使用することが推奨されます。ワークスペースの詳細は、OpenShift Pipelines のセクションを参照してください。

- ボリューム要求テンプレートのワークスペースのサポートが追加されました。
  - パイプライン実行およびタスク実行のボリューム要求テンプレートがワークスペースのボリュームソースとして追加できるようになりました。次に、tekton-controller はパイプラインのすべてのタスク実行の PVC として表示されるテンプレートを使用して永続ボリューム要求 (PVC) を作成します。したがって、複数のタスクにまたがるワークスペースをバインドするたびに PVC 設定を定義する必要はありません。
  - ボリューム要求テンプレートがボリュームソースとして使用される場合の PVC の名前検索のサポートが、変数の置換を使用して利用できるようになりました。
- 監査を強化するサポート:
  - **PipelineRun.Status** フィールドには、パイプラインのすべてのタスク実行のステータスと、パイプライン実行の進捗をモニターするためにパイプライン実行をインスタンス化する際に使用するパイプライン仕様が含まれるようになりました。
  - Pipeline の結果が Pipeline 仕様および **PipelineRun** ステータスに追加されました。
  - **TaskRun.Status** フィールドには、**TaskRun** リソースのインスタンス化に使用される実際のタスク仕様が含まれるようになりました。
- デフォルトパラメーターを各種の状態に適用するサポート。
- クラスタータスクを参照して作成されるタスク実行は、**tekton.dev/task** ラベルではなく **tekton.dev/clusterTask** ラベルを追加するようになりました。
- kube config writer は、kubeconfig-creator タスクでパイプラインリソースタイプクラスターの置き換えを有効にするために **ClientKeyData** および **ClientCertificateData** 設定をリソース構造に追加できるようになりました。
- **feature-flags** および **config-defaults** 設定マップの名前はカスタマイズ可能になりました。
- タスク実行で使用される Pod テンプレートのホストネットワークのサポートが追加されました。
- Affinity Assistant が、ワークスペースボリュームを共有するタスク実行のノードのアフィニティーをサポートするようになりました。デフォルトで、これは OpenShift Pipelines で無効にされます。
- Pod テンプレートは、Pod の起動時にコンテナーイメージのプルを許可するためにコンテナーランタイムが使用するシークレットを特定するために **imagePullSecrets** を指定するように更新されました。
- コントローラーがタスク実行の更新に失敗した場合にタスク実行コントローラーから警告イベントを出すためのサポート。
- アプリケーションまたはコンポーネントに属するリソースを特定するために、すべてのリソースに標準または推奨される k8s ラベルが追加されました。
- **Entrypoint** プロセスがシグナルについて通知されるようになり、これらのシグナルは **Entrypoint** プロセスの専用の PID グループを使用して伝播されるようになりました。
- Pod テンプレートはタスク実行仕様を使用してランタイム時にタスクレベルで設定できるようになりました。

- Kubernetes イベントを生成するサポート。
  - コントローラーは、追加のタスク実行ライフサイクルイベント (**taskrun started** および **taskrun running**) のイベントを生成するようになりました。
  - パイプライン実行コントローラーは、パイプラインの起動時に毎回イベントを生成するようになりました。
- デフォルトの Kubernetes イベントのほかに、タスク実行のクラウドイベントのサポートが利用可能になりました。コントローラーは、クラウドイベントとして create、started、および failed などのタスク実行イベントを送信するように設定できます。
- パイプライン実行およびタスク実行の場合に適切な名前を参照するための **$context.<taskRun|pipeline|pipelineRun>.name** 変数を使用するサポート。
- パイプライン実行パラメーターの検証が、パイプラインで必要なすべてのパラメーターがパイプライン実行によって提供できるようにするために利用可能になりました。これにより、パイプライン実行は必要なパラメーターに加えて追加のパラメーターを指定することもできます。
- パイプライン YAML ファイルの **finally** フィールドを使用して、すべてのタスクが正常に終了するか、パイプラインのタスクの失敗後、パイプラインが終了する前に常に実行されるパイプライン内でタスクを指定できるようになりました。
- **git-clone** クラスタータスクが利用できるようになりました。

1.16.1.2. Pipelines CLI

- 組み込まれた Trigger バインディングのサポートが、**tkn evenlistener describe** コマンドで利用できるようになりました。
- 正しくないサブコマンドが使用される場合にサブコマンドを推奨し、提案するためのサポート。
- **tkn task describe** コマンドは、1つのタスクのみがパイプラインに存在する場合にタスクを自動的に選択できるようになりました。
- **--use-param-defaults** フラグを **tkn task start** コマンドに指定することにより、デフォルトのパラメーター値を使用してタスクを起動できるようになりました。
- **--workspace** オプションを **tkn pipeline start** または **tkn task start** コマンドで使用して、パイプライン実行またはタスク実行のボリューム要求テンプレートを指定できるようになりました。
- **tkn pipelinerun logs** コマンドに、**finally** セクションにリスト表示される最終タスクのログが表示されるようになりました。
- インタラクティブモードのサポートが、以下の **tkn** リソース向けに **tkn task start** コマンドおよび **describe** サブコマンドに追加されました: **pipeline**、**pipelinerun**、**task**、**taskrun**、**clustertask** および **pipelineresource**。
- **tkn version** コマンドで、クラスターにインストールされているトリガー のバージョンが表示されるようになりました。
- **tkn pipeline describe** コマンドで、パイプラインで使用されるタスクに指定されたパラメーター値およびタイムアウトが表示されるようになりました。

- 最近のパイプライン実行またはタスク実行をそれぞれ記述できるように、**tkn pipelinerun describe** および **tkn taskrun describe** コマンドの **--last** オプションのサポートが追加されました。
- **tkn pipeline describe** コマンドに、パイプラインのタスクに適用される各種の状態が表示されるようになりました。
- **--no-headers** および **--all-namespaces** フラグを **tkn resource list** コマンドで使用できるようになりました。

1.16.1.3. トリガー

- 以下の Common Expression Language (CEL) 機能が利用できるようになりました。
  - **parseURL**: URL の一部を解析し、抽出します。
  - **parseJSON**: **deployment** webhook の **payload** フィールドの文字列に埋め込まれた JSON 値タイプを解析します。
- Bitbucket からの Webhook の新規インターセプターが追加されました。
- イベントリスナーは、**kubectl get** コマンドでリスト表示される際の追加フィールドとして **Address URL** および **Available status** を表示します。
- トリガーテンプレートパラメーターは、**$(params.<paramName>)** ではなく **$(tt.params.<paramName>)** 構文を使用するようになり、トリガーテンプレートとリソーステンプレートパラメーター間で生じる混乱が軽減されました。
- **EventListener** CRD に **tolerations** を追加し、セキュリティーや管理上の問題によりすべてのノードにテイントのマークが付けられる場合でもイベントリスナーが同じ設定でデプロイされるようにできるようになりました。
- イベントリスナー Deployment の Readiness Probe を **URL/live** に追加できるようになりました。
- イベントリスナートリガーでの **TriggerBinding** 仕様の埋め込みのサポート。
- Trigger リソースに推奨される **app.kubernetes.io** ラベルでアノテーションが付けられるようになりました。

1.16.2. 非推奨の機能

本リリースでは、以下の項目が非推奨になりました。

- **clustertask** コマンドおよび **clustertriggerbinding** コマンドを含む、クラスター全体のすべてのコマンドの **--namespace** または **-n** フラグが非推奨になりました。これは今後のリリースで削除されます。
- **ref** フィールドが優先されるため、イベントリスナー内の **triggers.bindings** の **name** フィールドは非推奨となり、今後のリリースで削除されます。
- **$(tt.params)** が優先されるため、**$(params)** を使用したトリガーテンプレートの変数の補間が非推奨となり、これにより、パイプライン変数の補間構文に関連した混乱が軽減されました。**$(params.<paramName>)** 構文は今後のリリースで削除されます。
- **tekton.dev/task** ラベルはクラスタータスクで非推奨になりました。

- **TaskRun.Status.ResourceResults.ResourceRef** フィールドは非推奨となり、今後削除されます。
- **tkn pipeline create**、 **tkn task create**、および **tkn resource create -f** サブコマンドが削除されました。
- namespace の検証が **tkn** コマンドから削除されました。
- **tkn ct start** コマンドのデフォルトタイムアウトの **1h** および **-t** フラグが削除されました。
- **s2i** クラスタータスクが非推奨になりました。

1.16.3. 既知の問題

- 各種の状態はワークスペースには対応しません。
- **--workspace** オプションとおよびインタラクティブモードは **tkn clustertask start** コマンドではサポートされていません。
- **$(params.<paramName>)** 構文の後方互換性のサポートにより、トリガーテンプレートがパイプライン固有のパラメーターで強制的に使用されます。 トリガー webhook がトリガーパラメーターとパイプラインパラメーターを区別できないためです。
- Pipeline メトリクスは、**tekton_taskrun_count** および **tekton_taskrun_duration_seconds_count** の promQL を実行する際に正しくない値を報告します。
- パイプライン実行およびタスク実行は、存在しない PVC 名がワークスペースに指定されている場合でも、それぞれ **Running** および **Running(Pending)** の状態のままになります。

1.16.4. 修正された問題

- 以前のバージョンでは、タスクおよびクラスタータスクの名前が同じ場合、**tkn task delete <name> --trs** コマンドは、タスクとクラスタータスクの両方を削除しました。今回の修正により、コマンドはタスク **<name>** で作成されるタスク実行のみを削除するようになりました。
- 以前のバージョンでは、**tkn pr delete -p <name> --keep 2** コマンドは、**--keep** フラグと共に使用する場合に **-p** フラグを無視し、最新の 2 つのパイプライン実行を除きすべてのパイプライン実行を削除しました。今回の修正により、コマンドは最新の 2 つのパイプライン実行を除き、パイプライン **<name>** で作成されるパイプライン実行のみを削除するようになりました。
- **tkn triggertemplate describe** 出力には、YAML 形式ではなくテーブル形式でリソーステンプレートが表示されるようになりました。
- 以前のバージョンでは、**buildah** クラスタータスクは、新規ユーザーがコンテナーに追加されると失敗していました。今回の修正により、この問題は解決されています。

1.17. RED HAT OPENSHIFT PIPELINES テクノロジープレビュー 1.0 のリリースノート

1.17.1. 新機能

Red Hat OpenShift Pipelines テクノロジープレビュー (TP) 1.0 が OpenShift Container Platform 4.4 で利用可能になりました。Red Hat OpenShift Pipelines TP 1.0 が以下をサポートするように更新されています。

- Tekton Pipelines 0.11.3
- Tekton **tkn** CLI 0.9.0
- Tekton Triggers 0.4.0
- Tekton Catalog 0.11 をベースとするクラスタータスク

以下では、修正および安定性の面での改善点に加え、OpenShift Pipelines 1.0 の主な新機能について説明します。

1.17.1.1. Pipelines

- v1beta1 API バージョンのサポート。
- 改善された制限範囲のサポート。以前のバージョンでは、制限範囲はタスク実行およびパイプライン実行に対してのみ指定されていました。制限範囲を明示的に指定する必要がなくなりました。namespace 間で最小の制限範囲が使用されます。
- タスク結果およびタスクパラメーターを使用してタスク間でデータを共有するためのサポート。
- パイプラインは、**HOME** 環境変数および各ステップの作業ディレクトリーを上書きしないように設定できるようになりました。
- タスクステップと同様に、**sidecars** がスクリプトモードをサポートするようになりました。
- タスク実行 **podTemplate** リソースに別のスケジューラーの名前を指定できるようになりました。
- Star Array Notation を使用した変数置換のサポート。
- Tekton コントローラーは、個別の namespace を監視するように設定できるようになりました。
- パイプライン、タスク、クラスタータスク、リソース、および状態 (condition) の仕様に新規の説明フィールドが追加されました。
- Git パイプラインリソースへのプロキシーパラメーターの追加。

1.17.1.2. Pipelines CLI

- **describe** サブコマンドが以下の **tkn** リソースについて追加されました。**EventListener**、**Condition**、**TriggerTemplate**、**ClusterTask**、および **TriggerSBinding**。
- **v1beta1** についてのサポートが、**v1alpha1** の後方互換性と共に以下のコマンドに追加されました。**ClusterTask**、**Task**、 **Pipeline**、**PipelineRun**、および **TaskRun**。
- 以下のコマンドは、**--all-namespaces** フラグオプションを使用してすべての namespace からの出力をリスト表示できるようになりました。これらは、**tkn task list**、**tkn pipeline list**、**tkn taskrun list**、**tkn pipelinerun list** です。
  これらのコマンドの出力は、**--no-headers** フラグオプションを使用してヘッダーなしで情報を表示するように強化されています。
- **--use-param-defaults** フラグを **tkn pipelines start** コマンドに指定することにより、デフォルトのパラメーター値を使用してパイプラインを起動できるようになりました。

- ワークスペースのサポートが **tkn pipeline start** および **tkn task start** コマンドに追加されるようになりました。
- 新規の **clustertriggerbinding** コマンドが以下のサブコマンドと共に追加されました。**describe**、**delete**、および **list**。
- ローカルまたはリモートの **yaml** ファイルを使用してパイプラインの実行を直接開始できるようになりました。
- **describe** サブコマンドには、強化され、詳細化した出力が表示されるようになりました。**description**、**timeout**、 **param description**、および **sidecar status** などの新規フィールドの追加により、コマンドの出力に特定の **tkn** リソースについてのより詳細な情報が提供されるようになりました。
- **tkn task log** コマンドには、1 つのタスクが namespace に存在する場合にログが直接表示されるようになりました。

1.17.1.3. トリガー

- Trigger は **v1alpha1** および **v1beta1** の両方のパイプラインリソースを作成できるようになりました。
- 新規 Common Expression Language (CEL) インターセプター機能 **compareSecret** のサポート。この機能は、文字列と CEL 式のシークレットを安全な方法で比較します。
- イベントリスナーのトリガーレベルでの認証および認可のサポート。

1.17.2. 非推奨の機能

本リリースでは、以下の項目が非推奨になりました。

- 環境変数 **$HOME**、および **Steps** 仕様の変数 **workingDir** が非推奨となり、今後のリリースで変更される可能性があります。現時点で **Step** コンテナーでは、**HOME** および **workingDir** 変数が **/tekton/home** および **/workspace** 変数にそれぞれ上書きされます。
  今後のリリースでは、これらの 2 つのフィールドは変更されず、コンテナーイメージおよび **Task** YAML で定義される値に設定されます。本リリースでは、**disable-home-env-overwrite** および **disable-working-directory-overwrite** フラグを使用して、**HOME** および **workingDir** 変数の上書きを無効にします。
- 以下のコマンドは非推奨となり、今後のリリースで削除される可能性があります。**tkn pipeline create**、 **tkn task create**。
- **tkn resource create** コマンドの **-f** フラグは非推奨になりました。これは今後のリリースで削除される可能性があります。
- **tkn clustertask create** コマンドの **-t** フラグおよび **--timeout** フラグ (秒単位の形式) は非推奨になりました。期間タイムアウトの形式のみがサポートされるようになりました (例: **1h30s**)。これらの非推奨のフラグは今後のリリースで削除される可能性があります。

1.17.3. 既知の問題

- 以前のバージョンの Red Hat OpenShift Pipelines からアップグレードする場合は、既存のデプロイメントを削除してから Red Hat OpenShift Pipelines バージョン 1.0 にアップグレードする必要があります。既存のデプロイメントを削除するには、まずカスタムリソースを削除してから Red Hat OpenShift Pipelines Operator をアンインストールする必要があります。詳細は、Red Hat OpenShift Pipelines のアンインストールについてのセクションを参照してください。

- 同じ **v1alpha1** タスクを複数回送信すると、エラーが発生します。**v1alpha1** タスクの再送信時に、**oc apply** ではなく **oc replace** コマンドを使用します。

- **buildah** クラスタータスクは、新規ユーザーがコンテナーに追加されると機能しません。Operator がインストールされると、**buildah** クラスタータスクの **--storage-driver** フラグが指定されていないため、フラグはデフォルト値に設定されます。これにより、ストレージドライバーが正しく設定されなくなることがあります。新規ユーザーが追加されると、storage-driver が間違っている場合に、**buildah** クラスタータスクが以下のエラーを出して失敗します。

```
useradd: /etc/passwd.8: lock file already used
useradd: cannot lock /etc/passwd; try again later.
```

回避策として、**buildah-task.yaml** ファイルで **--storage-driver** フラグの値を **overlay** に手動で設定します。

1. **cluster-admin** としてクラスターにログインします。

```
$ oc login -u <login> -p <password> https://openshift.example.com:6443
```

2. **oc edit** コマンドを使用して **buildah** クラスタータスクを編集します。

```
$ oc edit clustertask buildah
```

buildah clustertask YAML ファイルの現行バージョンが **EDITOR** 環境変数で設定されたエディターで開かれます。

3. **Steps** フィールドで、以下の **command** フィールドを見つけます。

```
 command: ['buildah', 'bud', '--format=$(params.FORMAT)', '--tls-
verify=$(params.TLSVERIFY)', '--layers', '-f', '$(params.DOCKERFILE)', '-t',
'$(resources.outputs.image.url)', '$(params.CONTEXT)']
```

4. **command** フィールドを以下に置き換えます。

```
 command: ['buildah', '--storage-driver=overlay', 'bud', '--format=$(params.FORMAT)', '--
tls-verify=$(params.TLSVERIFY)', '--no-cache', '-f', '$(params.DOCKERFILE)', '-t',
'$(params.IMAGE)', '$(params.CONTEXT)']
```

5. ファイルを保存して終了します。

または、**Pipelines → Cluster Tasks → buildah** に移動して、**buildah** クラスタータスク YAML ファイルを Web コンソール上で直接変更することもできます。**Actions** メニューから **Edit Cluster Task** を選択し、直前の手順のように **command** フィールドを置き換えます。

1.17.4. 修正された問題

- 以前のリリースでは、**DeploymentConfig** タスクは、イメージのビルドがすでに進行中であっても新規デプロイメントビルドをトリガーしていました。これにより、パイプラインのデプロイメントが失敗していました。今回の修正により、**deploy task** コマンドが **oc rollout status** コマンドに置き換えられ、進行中のデプロイメントが終了するまで待機するようになりました。

- **APP_NAME** パラメーターのサポートがパイプラインテンプレートに追加されました。

- 以前のバージョンでは、Java S2I のパイプラインテンプレートはレジストリーでイメージを検索できませんでした。今回の修正により、イメージはユーザーによって提供される **IMAGE_NAME** パラメーターの代わりに既存イメージのパイプラインリソースを使用して検索されるようになりました。

- OpenShift Pipelines イメージはすべて、Red Hat Universal Base Images (UBI) をベースにしています。

- 以前のバージョンでは、パイプラインが **tekton-pipelines** 以外の namespace にインストールされている場合、**tkn version** コマンドはパイプラインのバージョンを **unknown** と表示していました。今回の修正により、**tkn version** コマンドにより、正しいパイプラインのバージョンがすべての namespace で表示されるようになりました。

- **-c** フラグは **tkn version** コマンドでサポートされなくなりました。

- 管理者以外のユーザーがクラスタートリガーバインディングをリスト表示できるようになりました。

- イベントリスナーの **CompareSecret** 機能が、CEL インターセプターについて修正されました。

- タスクおよびクラスタータスクの **list**、**describe**、および **start** サブコマンドは、タスクおよびクラスタータスクが同じ名前を持つ場合に出力に正常に表示されるようになりました。

- 以前のバージョンでは、OpenShift Pipelines Operator は特権付き SCC (Security Context Constraints) を変更していました。これにより、クラスターのアップグレード時にエラーが発生しました。このエラーは修正されています。

- **tekton-pipelines** namespace では、設定マップを使用して、すべてのタスク実行およびパイプライン実行のタイムアウトが **default-timeout-minutes** フィールドの値に設定されるようになりました。

- 以前のバージョンでは、Web コンソールのパイプラインセクションは管理者以外のユーザーには表示されませんでした。この問題は解決されています。

第2章 RED HAT OPENSHIFT PIPELINE について

Red Hat OpenShift Pipelines は、Kubernetes リソースをベースとしたクラウドネイティブの継続的インテグレーションおよび継続的デリバリー (CI/CD) ソリューションです。これは Tekton ビルディングブロックを使用し、基礎となる実装の詳細を抽象化することで、複数のプラットフォームでのデプロイメントを自動化します。Tekton では、Kubernetes ディストリビューション間で移植可能な CI/CD パイプラインを定義するための標準のカスタムリソース定義 (CRD) が多数導入されています。

注記

Red Hat OpenShift Pipelines は OpenShift Container Platform とは異なる頻度でリリースされるため、Red Hat OpenShift Pipelines ドキュメントは製品のマイナーバージョンごとに個別のドキュメントセットとして利用できるようになりました。

Red Hat OpenShift Pipelines ドキュメントは https://docs.openshift.com/pipelines/ で利用できます。

特定のバージョンのドキュメントは、バージョンセレクターのドロップダウンリストを使用するか、URL にバージョンを直接追加 (https://docs.openshift.com/pipelines/1.14 など) することによって利用できます。

さらに、Red Hat OpenShift Pipelines のドキュメントは、Red Hatカスタマーポータルの https://access.redhat.com/documentation/ja-jp/red_hat_openshift_pipelines/ でも入手できます。

Red Hat OpenShift Pipelines のライフサイクルとサポートされるプラットフォームに関する追加情報は、プラットフォームのライフサイクルポリシー を参照してください。

第3章 OPENSHIFT PIPELINES について

Red Hat OpenShift Pipelines は、Kubernetes リソースをベースとしたクラウドネイティブの継続的インテグレーションおよび継続的デリバリー (CI/CD) ソリューションです。これは Tekton ビルディングブロックを使用し、基礎となる実装の詳細を抽象化することで、複数のプラットフォームでのデプロイメントを自動化します。Tekton では、Kubernetes ディストリビューション間で移植可能な CI/CD パイプラインを定義するための標準のカスタムリソース定義 (CRD) が多数導入されています。

3.1. 主な特長

- Red Hat OpenShift Pipelines は、分離されたコンテナーで必要なすべての依存関係と共にパイプラインを実行するサーバーレスの CI/CD システムです。
- Red Hat OpenShift Pipelines は、マイクロサービスベースのアーキテクチャーで機能する分散型チーム向けに設計されています。
- Red Hat OpenShift Pipelines は、拡張および既存の Kubernetes ツールとの統合を容易にする標準の CI/CD パイプライン定義を使用し、オンデマンドのスケーリングを可能にします。
- Red Hat OpenShift Pipelines を使用して、Kubernetes プラットフォーム全体で移植可能な S2I (Source-to-Image)、Buildah、Buildpacks、および Kaniko などの Kubernetes ツールを使用してイメージをビルドできます。
- OpenShift Container Platform Developer Console を使用して、Tekton リソースの作成、パイプライン実行のログの表示、OpenShift Container Platform namespace でのパイプラインの管理を実行できます。

3.2. OPENSHIFT PIPELINE の概念

本書では、パイプラインの各種概念を詳述します。

3.2.1. タスク

Task リソースは、パイプラインのビルディングブロックであり、順次実行されるステップで設定されます。これは基本的に入出力の機能です。Task は個別に実行することも、パイプラインの一部として実行することもできます。タスクは再利用可能で、複数のパイプラインで使用できます。

Step は、イメージのビルドなど、Task によって順次実行され、特定の目的を達成するための一連のコマンドです。各タスクは Pod として実行され、各ステップは同じ Pod 内のコンテナーとして実行されます。Step は同じ Pod 内で実行されるため、ファイル、設定マップ、およびシークレットをキャッシュするために同じボリュームにアクセスできます。

以下の例は、**apply-manifests** Task を示しています。

```
apiVersion: tekton.dev/v1 1
kind: Task 2
metadata:
  name: apply-manifests 3
spec: 4
  workspaces:
  - name: source
  params:
    - name: manifest_dir
      description: The directory in source that contains yaml manifests
```

```
    type: string
    default: "k8s"
steps:
  - name: apply
    image: image-registry.openshift-image-registry.svc:5000/openshift/cli:latest
    workingDir: /workspace/source
    command: ["/bin/bash", "-c"]
    args:
      - |-
        echo Applying manifests in $(params.manifest_dir) directory
        oc apply -f $(params.manifest_dir)
        echo ----------------------------------
```

1. タスク API バージョン **v1**。
2. Kubernetes オブジェクトのタイプ **Task**。
3. この Task の一意の名前。
4. Task のパラメーターおよび Step と、Task によって使用される Workspace のリスト。

この Task は Pod を起動し、指定されたコマンドを実行するために指定されたイメージを使用して Pod 内のコンテナーを実行されます。

注記

OpenShift Pipelines 1.6 以降、ステップ YAML ファイルから以下のデフォルトが削除されます。

- **HOME** 環境変数が **/tekton/home** ディレクトリーにデフォルト設定されない
- **workingDir** フィールドがデフォルトで **/workspace** ディレクトリーにない

代わりに、この手順のコンテナーは **HOME** 環境変数と **workingDir** フィールドを定義します。ただし、この手順の YAML ファイルにカスタム値を指定すると、デフォルト値を上書きできます。

一時的な手段として、以前の OpenShift Pipelines バージョンとの下位互換性を維持するために、**TektonConfig** カスタムリソース定義の以下のフィールドを **false** に設定できます。

```
spec:
  pipeline:
    disable-working-directory-overwrite: false
    disable-home-env-overwrite: false
```

3.2.2. when 式

when 式で、パイプライン内のタスクの実行の条件を設定して、タスク実行を保護します。これには、特定の条件が満たされる場合にのみタスクを実行できるようにします。when 式は、パイプライン YAML ファイルの **finally** フィールドを使用して指定される最終タスクセットでもサポートされます。

when 式の主要なコンポーネントは、以下のとおりです。

- **input**: パラメーター、タスクの結果、実行ステータスなどの静的入力または変数を指定します。有効な入力を入力する必要があります。有効な入力を入力しない場合は、デフォルトで空の文字列に設定されます。
- **operator**: **values** セットへの入力の関係を指定します。operator の値として **in** または **notin** を入力します。
- **values**: 文字列値の配列を指定します。ワークスペースに、パラメーター、結果、バインドされたステータスなどの静的値や変数の空でない配列を入力します。

宣言された when 式が、タスクの実行前に評価されます。when 式の値が **True** の場合は、タスクが実行します。when 式の値が **False** の場合、タスクはスキップします。

さまざまなユースケースで when 式を使用できます。たとえば、次のいずれかです。

- 以前のタスクの結果は期待どおりに実行される。
- Git リポジトリーのファイルが以前のコミットで変更になる。
- イメージがレジストリーに存在する。
- 任意のワークスペースが利用可能である。

以下の例は、パイプライン実行の when 式を示しています。パイプライン実行は、次の基準が満たされた場合にのみ **create-file** タスクを実行します。**path** パラメーターが **README.md** です。また、**check-file** タスクから生じる **exists** が **yes** の場合に限り、**echo-file-exists** タスクが実行します。

```
apiVersion: tekton.dev/v1
kind: PipelineRun (1)
metadata:
  generateName: guarded-pr-
spec:
  taskRunTemplate:
    serviceAccountName: pipeline
  pipelineSpec:
    params:
      - name: path
        type: string
        description: The path of the file to be created
    workspaces:
      - name: source
        description: |
          This workspace is shared among all the pipeline tasks to read/write common resources
    tasks:
      - name: create-file (2)
        when:
          - input: "$(params.path)"
            operator: in
            values: ["README.md"]
        workspaces:
          - name: source
            workspace: source
        taskSpec:
          workspaces:
            - name: source
              description: The workspace to create the readme file in
```

```
        steps:
          - name: write-new-stuff
            image: ubuntu
            script: 'touch $(workspaces.source.path)/README.md'
      - name: check-file
        params:
          - name: path
            value: "$(params.path)"
        workspaces:
          - name: source
            workspace: source
        runAfter:
          - create-file
        taskSpec:
          params:
            - name: path
          workspaces:
            - name: source
              description: The workspace to check for the file
          results:
            - name: exists
              description: indicates whether the file exists or is missing
          steps:
            - name: check-file
              image: alpine
              script: |
                if test -f $(workspaces.source.path)/$(params.path); then
                  printf yes | tee /tekton/results/exists
                else
                  printf no | tee /tekton/results/exists
                fi
      - name: echo-file-exists
        when: 3
          - input: "$(tasks.check-file.results.exists)"
            operator: in
            values: ["yes"]
        taskSpec:
          steps:
            - name: echo
              image: ubuntu
              script: 'echo file exists'
...
      - name: task-should-be-skipped-1
        when: 4
          - input: "$(params.path)"
            operator: notin
            values: ["README.md"]
        taskSpec:
          steps:
            - name: echo
              image: ubuntu
              script: exit 1
...
    finally:
      - name: finally-task-should-be-executed
        when: 5
```

```
        - input: "$(tasks.echo-file-exists.status)"
          operator: in
          values: ["Succeeded"]
        - input: "$(tasks.status)"
          operator: in
          values: ["Succeeded"]
        - input: "$(tasks.check-file.results.exists)"
          operator: in
          values: ["yes"]
        - input: "$(params.path)"
          operator: in
          values: ["README.md"]
      taskSpec:
        steps:
          - name: echo
            image: ubuntu
            script: 'echo finally done'
params:
  - name: path
    value: README.md
workspaces:
  - name: source
    volumeClaimTemplate:
      spec:
        accessModes:
          - ReadWriteOnce
        resources:
          requests:
            storage: 16Mi
```

1. Kubernetes オブジェクトのタイプを指定します。この例では、**PipelineRun** です。
2. パイプラインで使用されるタスク **create-file**。
3. **check-file** タスクから生じた **exists** が **yes** になった場合に限り、**echo-file-exists** タスクを実行するのに指定する **when** 式。
4. **path** パラメーターが **README.md** の場合に限り、**task-should-be-skipped-1** タスクをスキップすることを指定する **when** 式。
5. **echo-file-exists** タスクの実行ステータス、およびタスクステータスが **Succeeded** で、**check-file** タスクから生じる **exists** が **yes** になり、**path** パラメーターが **README.md** となる場合に限り、**finally-task-should-be-executed** タスクを実行するのに指定する **when** 式。

OpenShift Container Platform Web コンソールの **Pipeline Run details**ページには、以下のようにタスクと When 式が表示されます。

- すべての基準が満たされています。タスクと、ひし形で表される when 式の記号は緑色です。
- いずれかの基準が満たされていません。タスクはスキップされます。スキップされたタスクと when 式記号は灰色になります。
- 満たされていない基準はありません。タスクはスキップされます。スキップされたタスクと when 式記号は灰色になります。
- タスクの実行が失敗する: 失敗したタスクと when 式の記号が赤で表示されます。

3.2.3. 最後のタスク

finally のタスクは、パイプライン YAML ファイルの **finally** フィールドを使用して指定される最終タスクのセットです。**finally** タスクは、パイプライン実行が正常に実行されるかどうかに関係なく、パイプライン内でタスクを常に実行します。**finally** のタスクは、対応するパイプラインが終了する前に、すべてのパイプラインの実行後に並行して実行されます。

同じパイプライン内のタスクの結果を使用するように、**finally** タスクを設定できます。このアプローチでは、この最終タスクが実行される順序は変更されません。これは、最終以外のタスクすべての実行後に他の最終タスクと並行して実行されます。

以下の例は、**clone-cleanup-workspace** パイプラインのコードスニペットを示しています。このコードは、リポジトリーを共有ワークスペースにクローンし、ワークスペースをクリーンアップします。パイプラインタスクの実行後に、パイプライン YAML ファイルの **finally** セクションで指定される **cleanup** タスクがワークスペースをクリーンアップします。

```
apiVersion: tekton.dev/v1
kind: Pipeline
metadata:
  name: clone-cleanup-workspace (1)
spec:
  workspaces:
    - name: git-source (2)
  tasks:
    - name: clone-app-repo (3)
      taskRef:
        name: git-clone-from-catalog
      params:
        - name: url
          value: https://github.com/tektoncd/community.git
        - name: subdirectory
          value: application
      workspaces:
        - name: output
          workspace: git-source
  finally:
    - name: cleanup (4)
      taskRef: (5)
        name: cleanup-workspace
      workspaces: (6)
        - name: source
          workspace: git-source
    - name: check-git-commit
      params: (7)
        - name: commit
          value: $(tasks.clone-app-repo.results.commit)
      taskSpec: (8)
        params:
          - name: commit
        steps:
          - name: check-commit-initialized
            image: alpine
            script: |
```

```
        if [[ ! $(params.commit) ]]; then
          exit 1
        fi
```

1. Pipeline の一意の名前。
2. git リポジトリーのクローンが作成される共有ワークスペース。
3. アプリケーションリポジトリーを共有ワークスペースにクローンするタスク。
4. 共有ワークスペースをクリーンアップするタスク。
5. タスク実行で実行されるタスクへの参照。
6. 入力を受信し、出力を提供するために Pipeline のタスクがランタイム時に必要とする共有ストレージボリュームを宣言します。
7. タスクに必要なパラメーターのリスト。パラメーターに暗黙的なデフォルト値がない場合は、その値を明示的に設定する必要があります。
8. 埋め込まれたタスク定義。

3.2.4. TaskRun

TaskRun は、クラスター上の特定の入出力、および実行パラメーターで実行するためにタスクをインスタンス化します。これは独自に起動することも、パイプラインの各タスクの PipelineRun の一部として起動することもできます。

タスクはコンテナーイメージを実行する 1 つ以上のステップで設定され、各コンテナーイメージは特定のビルド作業を実行します。タスク実行では、すべてのステップが正常に実行されるか、失敗が発生するまで、指定された順序でタスクのステップが実行されます。**TaskRun** は、パイプライン内の各タスクに対して **PipelineRun** によって自動的に作成されます。

次の例は、関連する入力パラメーターを使用して **apply-manifests** タスクを実行するタスク実行を示しています。

```
apiVersion: tekton.dev/v1 1
kind: TaskRun 2
metadata:
  name: apply-manifests-taskrun 3
spec: 4
  taskRunTemplate:
    serviceAccountName: pipeline
  taskRef: 5
    kind: Task
    name: apply-manifests
  workspaces: 6
  - name: source
    persistentVolumeClaim:
      claimName: source-pvc
```

1. タスク実行 API バージョン **v1**。
2. Kubernetes オブジェクトのタイプを指定します。この例では、**TaskRun** です。

3 このタスク実行を識別する一意の名前。

4 タスク実行の定義。このタスクの実行では、タスクと必要なワークスペースが指定されています。

5 このタスクの実行に使用されるタスク参照の名前。このタスク実行は、**apply-manifests** タスクを実行します。

6 タスクの実行で使用されるワークスペース。

3.2.5. Pipelines

Pipeline は、特定の実行順序で配置された **Task** リソースのコレクションです。これらは、アプリケーションのビルド、デプロイメント、およびデリバリーを自動化する複雑なワークフローを構築するために実行されます。1 つ以上のタスクを含むパイプラインを使用して、アプリケーションの CI/CD ワークフローを定義できます。

Pipeline 定義は、多くのフィールドまたは属性で設定され、Pipeline が特定の目的を達成することを可能にします。各 **Pipeline** リソース定義には、特定の入力を取り込み、特定の出力を生成する **Task** が少なくとも 1 つ含まれる必要があります。パイプライン定義には、アプリケーション要件に応じて **Conditions**、**Workspaces**、 **Parameters**、または **Resources** をオプションで含めることもできます。

以下の例は、**buildah ClusterTask** を使用して Git リポジトリーからアプリケーションイメージをビルドする **build-and-deploy** パイプラインを示しています。

```
apiVersion: tekton.dev/v1 1
kind: Pipeline 2
metadata:
  name: build-and-deploy 3
spec: 4
  workspaces: 5
  - name: shared-workspace
  params: 6
  - name: deployment-name
    type: string
    description: name of the deployment to be patched
  - name: git-url
    type: string
    description: url of the git repo for the code of deployment
  - name: git-revision
    type: string
    description: revision to be used from repo of the code for deployment
    default: "pipelines-1.14"
  - name: IMAGE
    type: string
    description: image to be built from the code
  tasks: 7
  - name: fetch-repository
    taskRef:
      name: git-clone
      kind: ClusterTask
    workspaces:
    - name: output
      workspace: shared-workspace
    params:
```

```
    - name: url
      value: $(params.git-url)
    - name: subdirectory
      value: ""
    - name: deleteExisting
      value: "true"
    - name: revision
      value: $(params.git-revision)
  - name: build-image 8
    taskRef:
      name: buildah
      kind: ClusterTask
    params:
    - name: TLSVERIFY
      value: "false"
    - name: IMAGE
      value: $(params.IMAGE)
    workspaces:
    - name: source
      workspace: shared-workspace
    runAfter:
    - fetch-repository
  - name: apply-manifests 9
    taskRef:
      name: apply-manifests
    workspaces:
    - name: source
      workspace: shared-workspace
    runAfter: 10
    - build-image
  - name: update-deployment
    taskRef:
      name: update-deployment
    workspaces:
    - name: source
      workspace: shared-workspace
    params:
    - name: deployment
      value: $(params.deployment-name)
    - name: IMAGE
      value: $(params.IMAGE)
    runAfter:
    - apply-manifests
```

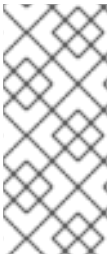
1. Pipeline API バージョン **v1**
2. Kubernetes オブジェクトのタイプを指定します。この例では、**Pipeline** です。
3. この Pipeline の一意の名前。
4. Pipeline の定義と構造を指定します。
5. Pipeline のすべてのタスクで使用される Workspace。
6. Pipeline のすべてのタスクで使用されるパラメーター。

7 Pipeline で使用されたタスクの一覧を指定します。

8 タスク **build-image**: **buildah ClusterTask** を使用して、所定の Git リポジトリーからアプリケーションイメージをビルドします。

9 タスク **apply-manifests**: 同じ名前のユーザー定義のタスクを使用します。

10 タスクが Pipeline で実行されるシーケンスを指定します。この例では、**apply-manifests** タスクは **build-image** タスクが完了した後にのみ実行されます。



注記

Red Hat OpenShift Pipelines Operator は Buildah クラスタータスクをインストールし、イメージのビルドおよびプッシュを実行するのに十分なパーミッションを割り当てて、**パイプライン** サービスアカウントを作成します。Buildah クラスタータスクは、パーミッションが不十分な別のサービスアカウントに関連付けられていると失敗する可能性があります。

3.2.6. PipelineRun

PipelineRun は、パイプライン、ワークスペース、認証情報、および CI/CD ワークフローを実行するシナリオ固有のパラメーター値のセットをバインドするリソースタイプです。

パイプライン実行は、パイプラインの実行中のインスタンスです。これは、クラスター上の特定の入力、出力、および実行パラメーターで実行されるパイプラインをインスタンス化します。また、パイプライン実行に、タスクごとのタスク実行も作成します。

パイプラインは、完了するか、タスクが失敗するまでタスクを順次実行します。**status** フィールドは、監視および監査のために、Task run ごとの進捗を追跡し、保存します。

以下の例は、関連するリソースおよびパラメーターで **build-and-deploy** Pipeline を実行しています。

```
apiVersion: tekton.dev/v1 1
kind: PipelineRun 2
metadata:
  name: build-deploy-api-pipelinerun 3
spec:
  pipelineRef:
    name: build-and-deploy 4
  params: 5
  - name: deployment-name
    value: vote-api
  - name: git-url
    value: https://github.com/openshift-pipelines/vote-api.git
  - name: IMAGE
    value: image-registry.openshift-image-registry.svc:5000/pipelines-tutorial/vote-api
  workspaces: 6
  - name: shared-workspace
    volumeClaimTemplate:
      spec:
        accessModes:
          - ReadWriteOnce
```

```
    resources:
      requests:
        storage: 500Mi
```

1. パイプライン実行 API バージョン **v1**。
2. Kubernetes オブジェクトのタイプ。この例では、**PipelineRun** です。
3. この Pipeline Run を識別する一意の名前。
4. 実行する Pipeline の名前。この例では、**build-and-deploy** です。
5. Pipeline の実行に必要なパラメーターのリスト。
6. Pipeline Run で使用する Workspace。

関連情報

- git シークレットを使用したパイプラインの認証

3.2.7. Workspaces

注記

Red Hat OpenShift Pipelines の **PipelineResource** CR の代わりにワークスペースを使用することを推奨します。これは、**PipelineResource** CR はデバッグが難しく、範囲が限定されており、タスクの再利用性が低下するためです。

Workspace は、入力を受信し、出力を提供するために Pipeline の Task がランタイム時に必要とする共有ストレージボリュームを宣言します。Workspace では、ボリュームの実際の場所を指定する代わりに、ランタイム時に必要となるファイルシステムまたはファイルシステムの一部を宣言できます。Task または Pipeline は Workspace を宣言し、ボリュームの特定の場所の詳細を指定する必要があります。その後、これは TaskRun または PipelineRun の Workspace にマウントされます。ランタイムストレージボリュームからボリューム宣言を分離することで、Task を再利用可能かつ柔軟にし、ユーザー環境から切り離すことができます。

Workspace を使用すると、以下が可能になります。

- Task の入力および出力の保存
- Task 間でのデータの共有
- Secret に保持される認証情報のマウントポイントとして使用
- config maps に保持される設定のマウントポイントとして使用
- 組織が共有する共通ツールのマウントポイントとして使用
- ジョブを高速化するビルドアーティファクトのキャッシュの作成

以下を使用して、**TaskRun** または **PipelineRun** で Workspace を指定できます。

- 読み取り専用の config map またはシークレット
- 他のタスクと共有されている既存の永続ボリューム要求

- 提供されたボリュームクレームテンプレートからの永続的なボリュームクレーム
- タスクの実行が完了すると破棄される **emptyDir**

以下の例は、Pipeline で定義される、**build-image** および **apply-manifests** Task の **shared-workspace** Workspace を宣言する **build-and-deploy** Pipeline のコードスニペットを示しています。

```
apiVersion: tekton.dev/v1
kind: Pipeline
metadata:
  name: build-and-deploy
spec:
  workspaces: 1
  - name: shared-workspace
  params:
...
  tasks: 2
  - name: build-image
    taskRef:
      name: buildah
      kind: ClusterTask
    params:
    - name: TLSVERIFY
      value: "false"
    - name: IMAGE
      value: $(params.IMAGE)
    workspaces: 3
    - name: source 4
      workspace: shared-workspace 5
    runAfter:
    - fetch-repository
  - name: apply-manifests
    taskRef:
      name: apply-manifests
    workspaces: 6
    - name: source
      workspace: shared-workspace
    runAfter:
      - build-image
...
```

1. Pipeline で定義される Task 間で共有される Workspace の一覧。Pipeline は、必要な数の Workspace を定義できます。この例では、**shared-workspace** という名前の 1 つの Workspace のみが宣言されます。
2. Pipeline で使用される Task の定義。このスニペットは、共通の Workspace を共有する **build-image** および **apply-manifests** の 2 つの Task を定義します。
3. **build-image** Task で使用される Workspace の一覧。Task 定義には、必要な数の Workspace を含めることができます。ただし、Task が最大 1 つの書き込み可能な Workspace を使用することが推奨されます。
4. Task で使用される Workspace を一意に識別する名前。この Task は、**source** という名前の 1 つの Workspace を使用します。

5 Task によって使用される Pipeline Workspace の名前。Workspace **source** は Pipeline Workspace の **shared-workspace** を使用することに注意してください。

6 **apply-manifests** Task で使用される Workspace の一覧。この Task は、 **build-image** Task と **source** Workspace を共有することに注意してください。

Workspace はタスクがデータを共有する際に使用でき、これにより、パイプラインの各タスクが実行時に必要となる 1 つまたは複数のボリュームを指定することができます。永続ボリューム要求 (PVC) を作成するか、永続ボリューム要求 (PVC) を作成するボリューム要求テンプレートを指定できます。

以下の **build-deploy-api-pipelinerun** PipelineRun のコードスニペットは、 **build-and-deploy** Pipeline で使用される **shared-workspace** Workspace のストレージボリュームを定義するための永続ボリューム要求 (PVC) を作成するために永続ボリュームテンプレートを使用します。

```
apiVersion: tekton.dev/v1
kind: PipelineRun
metadata:
  name: build-deploy-api-pipelinerun
spec:
  pipelineRef:
    name: build-and-deploy
  params:
...

  workspaces: 1
  - name: shared-workspace 2
    volumeClaimTemplate: 3
      spec:
        accessModes:
          - ReadWriteOnce
        resources:
          requests:
            storage: 500Mi
```

1 Pipeline にボリュームバインディングを提供する Pipeline Workspace の一覧を指定します。

2 ボリュームが提供されている Pipeline の Workspace の名前。

3 ワークスペースのストレージボリュームを定義するために永続ボリューム要求 (PVC) を作成するボリューム要求テンプレートを指定します。

3.2.8. トリガー

Trigger をパイプラインと併用して、Kubernetes リソースで CI/CD 実行全体を定義する本格的な CI/CD システムを作成します。Trigger は、Git プルリクエストなどの外部イベントをキャプチャーし、それらのイベントを処理して情報の主要な部分を抽出します。このイベントデータを事前に定義されたパラメーターのセットにマップすると、Kubernetes リソースを作成およびデプロイし、パイプラインをインスタンス化できる一連のタスクがトリガーされます。

たとえば、アプリケーションの Red Hat OpenShift Pipelines を使用して CI/CD ワークフローを定義します。アプリケーションリポジトリーで新たな変更を有効にするには、パイプラインを開始する必要があります。トリガーは変更イベントをキャプチャーし、処理することにより、また新規イメージを最新の変更でデプロイするパイプライン実行をトリガーして、このプロセスを自動化します。

Trigger は、再利用可能で分離した自律型 CI/CD システムを設定するように連携する以下の主なリソースで設定されています。

- **TriggerBinding** リソースは、イベントペイロードからフィールドを抽出し、それらをパラメーターとして保存します。
  以下の例は、**TriggerBinding** リソースのコードスニペットを示しています。これは、受信イベントペイロードから Git リポジトリー情報を抽出します。

```
apiVersion: triggers.tekton.dev/v1beta1 1
kind: TriggerBinding 2
metadata:
  name: vote-app 3
spec:
  params: 4
  - name: git-repo-url
    value: $(body.repository.url)
  - name: git-repo-name
    value: $(body.repository.name)
  - name: git-revision
    value: $(body.head_commit.id)
```

1. **TriggerBinding** リソースの API バージョン。この例では、**v1beta1** です。
2. Kubernetes オブジェクトのタイプを指定します。この例では、**TriggerBinding** です。
3. この **TriggerBinding** を識別する一意の名前。
4. 受信イベントペイロードから抽出され、 **TriggerTemplate** に渡されるパラメーターのリスト。この例では、Git リポジトリー URL、名前、およびリビジョンはイベントペイロードの本体から抽出されます。

- **TriggerTemplate** リソースは、リソースの作成方法の標準として機能します。これは、**TriggerBinding** リソースからのパラメーター化されたデータが使用される方法を指定します。トリガーテンプレートは、トリガーバインディングから入力を受信し、新規パイプラインリソースの作成および新規パイプライン実行の開始につながる一連のアクションを実行します。
  以下の例は、**TriggerTemplate** リソースのコードスニペットを示しています。これは、作成した **TriggerBinding** リソースから受信される Git リポジトリー情報を使用してパイプライン実行を作成します。

```
apiVersion: triggers.tekton.dev/v1beta1 1
kind: TriggerTemplate 2
metadata:
  name: vote-app 3
spec:
  params: 4
  - name: git-repo-url
    description: The git repository url
  - name: git-revision
    description: The git revision
    default: pipelines-1.14
  - name: git-repo-name
    description: The name of the deployment to be created / patched
```

```
  resourcetemplates: 5
  - apiVersion: tekton.dev/v1
    kind: PipelineRun
    metadata:
      name: build-deploy-$(tt.params.git-repo-name)-$(uid)
    spec:
      taskRunTemplate:
        serviceAccountName: pipeline
      pipelineRef:
        name: build-and-deploy
      params:
      - name: deployment-name
        value: $(tt.params.git-repo-name)
      - name: git-url
        value: $(tt.params.git-repo-url)
      - name: git-revision
        value: $(tt.params.git-revision)
      - name: IMAGE
        value: image-registry.openshift-image-registry.svc:5000/pipelines-
tutorial/$(tt.params.git-repo-name)
      workspaces:
      - name: shared-workspace
        volumeClaimTemplate:
          spec:
            accessModes:
             - ReadWriteOnce
            resources:
              requests:
                storage: 500Mi
```

1. **TriggerTemplate** リソースの API バージョン。この例では、**v1beta1** です。
2. Kubernetes オブジェクトのタイプを指定します。この例では、**TriggerTemplate** です。
3. **TriggerTemplate** リソースを識別するための一意の名前。
4. **TriggerBinding** リソースによって提供されるパラメーター。
5. **TriggerBinding** または **EventListener** リソースを使用して受信されるパラメーターを使用してリソースを作成する必要がある方法を指定するテンプレートの一覧。

- **Trigger** リソースは、**TriggerBinding** リソースおよび **TriggerTemplate** リソースと、オプションで **interceptors** イベントプロセッサーを組み合わせます。
  インターセプターは、**TriggerBinding** リソースの前に実行される特定プラットフォームのすべてのイベントを処理します。インターセプターを使用して、ペイロードのフィルタリング、イベントの検証、トリガー条件の定義およびテスト、および他の有用な処理を実装できます。インターセプターは、イベント検証にシークレットを使用します。イベントデータがインターセプターを通過したら、ペイロードデータをトリガーバインディングに渡す前にトリガーに移動します。インターセプターを使用して、**EventListener** 仕様で参照される関連付けられたトリガーの動作を変更することもできます。

  以下の例は、**TriggerBinding** および **TriggerTemplate** リソースを接続する **vote-trigger** という名前の **Trigger** リソースのコードスニペットと、**interceptors** イベントプロセッサーを示しています。

```
apiVersion: triggers.tekton.dev/v1beta1 (1)
kind: Trigger (2)
metadata:
  name: vote-trigger (3)
spec:
  taskRunTemplate:
    serviceAccountName: pipeline (4)
  interceptors:
    - ref:
        name: "github" (5)
      params: (6)
        - name: "secretRef"
          value:
            secretName: github-secret
            secretKey: secretToken
        - name: "eventTypes"
          value: ["push"]
  bindings:
    - ref: vote-app (7)
  template: (8)
     ref: vote-app
---
apiVersion: v1
kind: Secret (9)
metadata:
  name: github-secret
type: Opaque
stringData:
  secretToken: "1234567"
```

(1) **Trigger** リソースの API バージョン。この例では、**v1beta1** です。

(2) Kubernetes オブジェクトのタイプを指定します。この例では、**Trigger** です。

(3) この **Trigger** リソースを識別するための一意の名前。

(4) 使用されるサービスアカウント名。

(5) 参照されるインターセプター名。この例では、**github** です。

(6) 指定する必要のあるパラメーター。

(7) **TriggerTemplate** リソースに接続する **TriggerBinding** リソースの名前。

(8) **TriggerBinding** リソースに接続するための **TriggerTemplate** リソースの名前。

(9) イベントの検証に使用されるシークレット。

- **EventListener** は、JSON ペイロードを含む受信 HTTP ベースイベントをリッスンするエンドポイントまたはイベントシンクを提供します。これは各 **TriggerBinding** リソースからイベントパラメーターを抽出し、次にこのデータを処理し、対応する **TriggerTemplate** リソースによって指定される Kubernetes リソースを作成します。**EventListener** リソースは、イベントの **interceptors** を使用してペイロードで軽量イベント処理または基本的なフィルターを実行します。これはペイロードのタイプを特定し、オプションでこれを変更します。現時点で、パイプ

ライントリガーは **Webhook インターセプター**、**GitHub インターセプター**、**GitLab インターセプター**、**Bitbucket インターセプター**、および **Common Expression Language (CEL) インターセプター** の 4 種類のインターセプターをサポートします。
以下の例は、**vote-trigger** という名前の **Trigger** リソースを参照する **EventListener** リソースを示しています。

```
apiVersion: triggers.tekton.dev/v1beta1 1
kind: EventListener 2
metadata:
  name: vote-app 3
spec:
  taskRunTemplate:
    serviceAccountName: pipeline 4
  triggers:
    - triggerRef: vote-trigger 5
```

1. **EventListener** リソースの API バージョン。この例では、**v1beta1** です。
2. Kubernetes オブジェクトのタイプを指定します。この例では、**EventListener** です。
3. **EventListener** リソースを識別するための一意の名前。
4. 使用されるサービスアカウント名。
5. **EventListener** リソースによって参照される **Trigger** リソースの名前。

3.3. 関連情報

- OpenShift Pipelines のインストールの詳細は、OpenShift Pipelines のインストール を参照してください。
- カスタム CI/CD ソリューションの作成の詳細は、OpenShift Pipelines を使用したアプリケーション用の CI/CD ソリューションの作成 を参照してください。
- re-encrypt TLS 終端の詳細は、再暗号化終端 を参照してください。
- セキュリティー保護されたルートの詳細は、セキュリティー保護されたルート セクションを参照してください。