Red Hat OpenShift Pipelines 1.15

About OpenShift Pipelines

Introduction to OpenShift Pipelines

Last Updated: 2024-06-26

Red Hat OpenShift Pipelines 1.15 About OpenShift Pipelines

Introduction to OpenShift Pipelines

Legal Notice

Abstract

This document provides an overview of OpenShift Pipelines features. It also includes release notes and details on how to get support.

Table of Contents

CHAPTER 1. RED HAT OPENSHIFT PIPELINES RELEASE NOTES 7
1.1. COMPATIBILITY AND SUPPORT MATRIX 7
1.2. MAKING OPEN SOURCE MORE INCLUSIVE 8
1.3. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.15 8
1.3.1. New features 8
1.3.1.1. Pipelines 9
1.3.1.2. Operator 11
1.3.1.3. Triggers 12
1.3.1.4. Manual approval 13
1.3.1.5. CLI 14
1.3.1.6. Pipelines as Code 14
1.3.1.7. Tekton Results 15
1.3.2. Breaking changes 17
1.3.3. Known issues 18
1.3.4. Fixed issues 18
1.4. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.14 21
1.4.1. New features 21
1.4.1.1. Pipelines 22
1.4.1.2. Operator 24
1.4.1.3. Triggers 24
1.4.1.4. Web console 25
1.4.1.5. CLI 25
1.4.1.6. Pipelines as Code 26
1.4.1.7. Tekton Chains 26
1.4.1.8. Tekton Results 27
1.4.2. Breaking changes 28
1.4.3. Known issues 28
1.4.4. Fixed issues 29
1.4.5. Release notes for Red Hat OpenShift Pipelines General Availability 1.14.1 29
1.4.5.1. Fixed issues 29
1.4.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.14.2 29
1.4.6.1. Fixed issues 29
1.4.7. Release notes for Red Hat OpenShift Pipelines General Availability 1.14.3 30
1.4.7.1. Fixed issues 30
1.4.8. Release notes for Red Hat OpenShift Pipelines General Availability 1.14.4 30
1.4.8.1. Fixed issues 30
1.5. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.13 31
1.5.1. New features 31
1.5.1.1. Pipelines 31
1.5.1.2. Operator 32
1.5.1.3. Triggers 33
1.5.1.4. CLI 33
1.5.1.5. Pipelines as Code 33
1.5.2. Breaking changes 34
1.5.3. Known issues 34
1.5.4. Fixed issues 34
1.5.5. Release notes for Red Hat OpenShift Pipelines General Availability 1.13.1 35
1.5.5.1. Fixed issues 35
1.6. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.12 35
1.6.1. New features 35
1.6.1.1. Pipelines 36

1.6.1.2. Operator 36
1.6.1.3. Tekton Results 37
1.6.1.4. Triggers 37
1.6.1.5. CLI 37
1.6.1.6. Pipelines as Code 38
1.6.2. Breaking changes 39
1.6.3. Known issues 39
1.6.4. Fixed issues 40
1.6.5. Release notes for Red Hat OpenShift Pipelines General Availability 1.12.1 41
1.6.5.1. Fixed issues 41
1.6.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.12.2 41
1.6.6.1. Fixed issues 42
1.7. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.11 42
1.7.1. New features 42
1.7.1.1. Pipelines 42
1.7.1.2. Triggers 43
1.7.1.3. Operator 43
1.7.1.4. Tekton Chains 43
1.7.1.5. Tekton Hub 43
1.7.1.6. Tekton Results 44
1.7.1.7. Pipelines as Code 44
1.7.2. Breaking changes 45
1.7.3. Deprecated and removed features 45
1.7.4. Known issues 46
1.7.5. Fixed issues 46
1.7.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.11.1 47
1.7.6.1. Fixed issues 48
1.7.7. Release notes for Red Hat OpenShift Pipelines General Availability 1.11.2 49
1.7.7.1. Fixed issues 49
1.7.8. Known issues 49
1.7.9. Release notes for Red Hat OpenShift Pipelines General Availability 1.11.3 50
1.7.9.1. Fixed issues 50
1.8. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.10 50
1.8.1. New features 50
1.8.1.1. Pipelines 50
1.8.1.2. Triggers 51
1.8.1.3. CLI 51
1.8.1.4. Operator 51
1.8.1.5. Tekton Chains 51
1.8.1.6. Tekton Hub 52
1.8.1.7. Pipelines as Code 52
1.8.2. Breaking changes 53
1.8.3. Deprecated and removed features 53
1.8.4. Known issues 54
1.8.5. Fixed issues 54
1.8.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.10.1 55
1.8.6.1. Fixed issues for Pipelines as Code 55
1.8.7. Release notes for Red Hat OpenShift Pipelines General Availability 1.10.2 55
1.8.7.1. Fixed issues 55
1.8.8. Release notes for Red Hat OpenShift Pipelines General Availability 1.10.3 56
1.8.8.1. Fixed issues 56
1.8.9. Release notes for Red Hat OpenShift Pipelines General Availability 1.10.4 56
1.8.9.1. Fixed issues 56

1.8.10. Release notes for Red Hat OpenShift Pipelines General Availability 1.10.5 56
1.8.10.1. Fixed issues 56
1.8.11. Release notes for Red Hat OpenShift Pipelines General Availability 1.10.6 57
1.8.11.1. Known issues 57
1.8.11.2. Fixed issues 57
1.9. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.9 57
1.9.1. New features 57
1.9.1.1. Pipelines 58
1.9.1.2. Triggers 58
1.9.1.3. CLI 58
1.9.1.4. Operator 59
1.9.1.5. Resolvers 59
1.9.1.6. Tekton Chains 60
1.9.1.7. Tekton Hub 61
1.9.1.8. Pipelines as Code 61
1.9.2. Breaking changes 63
1.9.3. Deprecated and removed features 63
1.9.4. Known issues 64
1.9.5. Fixed issues 64
1.9.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.9.1 66
1.9.7. Fixed issues 66
1.9.8. Known issues 67
1.9.9. Breaking changes 67
1.9.10. Release notes for Red Hat OpenShift Pipelines General Availability 1.9.2 67
1.9.11. Fixed issues 67
1.9.12. Release notes for Red Hat OpenShift Pipelines General Availability 1.9.3 67
1.9.13. Fixed issues 67
1.9.14. Known issues 68
1.10. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.8 68
1.10.1. New features 68
1.10.1.1. Pipelines 68
1.10.1.2. Triggers 69
1.10.1.3. CLI 70
1.10.1.4. Operator 70
1.10.1.5. Tekton Chains 70
1.10.1.6. Tekton Hub 71
1.10.1.7. Pipelines as Code 72
1.10.2. Breaking changes 73
1.10.3. Deprecated and removed features 74
1.10.4. Known issues 75
1.10.5. Fixed issues 76
1.10.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.8.1 78
1.10.6.1. Known issues 78
1.10.6.2. Fixed issues 79
1.10.7. Release notes for Red Hat OpenShift Pipelines General Availability 1.8.2 80
1.10.7.1. Fixed issues 80
1.11. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.7 80
1.11.1. New features 80
1.11.1.1. Pipelines 80
1.11.1.2. Triggers 81
1.11.1.3. CLI 81
1.11.1.4. Operator 82
1.11.1.5. Hub 83

1.11.1.6. Chains 83
1.11.1.7. Pipelines as Code (PAC) 84
1.11.2. Deprecated features 85
1.11.3. Known issues 85
1.11.4. Fixed issues 85
1.11.5. Release notes for Red Hat OpenShift Pipelines General Availability 1.7.1 86
1.11.5.1. Fixed issues 86
1.11.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.7.2 87
1.11.6.1. Known issues 87
1.11.6.2. Fixed issues 87
1.11.7. Release notes for Red Hat OpenShift Pipelines General Availability 1.7.3 88
1.11.7.1. Fixed issues 88
1.12. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.6 88
1.12.1. New features 88
1.12.2. Deprecated features 91
1.12.3. Known issues 92
1.12.4. Fixed issues 93
1.12.5. Release notes for Red Hat OpenShift Pipelines General Availability 1.6.1 93
1.12.5.1. Known issues 93
1.12.5.2. Fixed issues 93
1.12.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.6.2 94
1.12.6.1. Known issues 94
1.12.6.2. Fixed issues 94
1.12.7. Release notes for Red Hat OpenShift Pipelines General Availability 1.6.3 94
1.12.7.1. Fixed issues 95
1.12.8. Release notes for Red Hat OpenShift Pipelines General Availability 1.6.4 95
1.12.8.1. Known issues 95
1.12.8.2. Fixed issues 96
1.13. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.5 96
1.13.1. Compatibility and support matrix 96
1.13.2. New features 97
1.13.3. Deprecated features 99
1.13.4. Known issues 101
1.13.5. Fixed issues 103
1.14. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.4 103
1.14.1. Compatibility and support matrix 103
1.14.2. New features 104
1.14.3. Deprecated features 105
1.14.4. Known issues 106
1.14.5. Fixed issues 106
1.15. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES TECHNOLOGY PREVIEW 1.3 108
1.15.1. New features 108
1.15.1.1. Pipelines 109
1.15.1.2. Pipelines CLI 109
1.15.1.3. Triggers 110
1.15.2. Deprecated features 110
1.15.3. Known issues 111
1.15.4. Fixed issues 111
1.16. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES TECHNOLOGY PREVIEW 1.2 112
1.16.1. New features 112
1.16.1.1. Pipelines 112
1.16.1.2. Pipelines CLI 113
1.16.1.3. Triggers 114

1.16.2. Deprecated features 114
1.16.3. Known issues 114
1.16.4. Fixed issues 115
1.17. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES TECHNOLOGY PREVIEW 1.1 116
1.17.1. New features 116
1.17.1.1. Pipelines 116
1.17.1.2. Pipelines CLI 118
1.17.1.3. Triggers 118
1.17.2. Deprecated features 119
1.17.3. Known issues 119
1.17.4. Fixed issues 119
1.18. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES TECHNOLOGY PREVIEW 1.0 120
1.18.1. New features 120
1.18.1.1. Pipelines 120
1.18.1.2. Pipelines CLI 121
1.18.1.3. Triggers 121
1.18.2. Deprecated features 121
1.18.3. Known issues 122
1.18.4. Fixed issues 123

CHAPTER 2. ABOUT RED HAT OPENSHIFT PIPELINES 124

CHAPTER 3. UNDERSTANDING OPENSHIFT PIPELINES 125
3.1. KEY FEATURES 125
3.2. OPENSHIFT PIPELINES CONCEPTS 125
3.2.1. Tasks 125
3.2.2. When expression 126
3.2.3. Finally tasks 130
3.2.4. TaskRun 131
3.2.5. Pipelines 132
3.2.6. PipelineRun 134
3.2.7. Workspaces 135
3.2.8. Triggers 138
3.3. ADDITIONAL RESOURCES 141

CHAPTER 1. RED HAT OPENSHIFT PIPELINES RELEASE NOTES

Red Hat OpenShift Pipelines is a cloud-native CI/CD experience based on the Tekton project which provides:

- Standard Kubernetes-native pipeline definitions (CRDs).
- Serverless pipelines with no CI server management overhead.
- Extensibility to build images using any Kubernetes tool, such as S2I, Buildah, JIB, and Kaniko.
- Portability across any Kubernetes distribution.
- Powerful CLI for interacting with pipelines.
- Integrated user experience with the **Developer** perspective of the OpenShift Container Platform web console.

For an overview of Red Hat OpenShift Pipelines, see Understanding OpenShift Pipelines.

1.1. COMPATIBILITY AND SUPPORT MATRIX

Some features in this release are currently in Technology Preview. These experimental features are not intended for production use.

In the table, features are marked with the following statuses:

TP	Technology Preview
GA	General Availability

Table 1.1. Compatibility and support matrix

Red Hat OpenShift Pipelines Version	Component Version								OpenShift Version	Support Status
Operator	Pipelines	Triggers	CLI	Chains	Hub	Pipelines as Code	Results	Manual Approval Gate		
1.15	0.59.x	0.27.x	0.37.x	0.20.x (GA)	1.17.x (TP)	0.27.x (GA)	0.10.x (TP)	0.2.x (TP)	4.14, 4.15, 4.16	GA

Red Hat OpenShift Pipelines Version	Component Version								OpenShift Version	Support Status
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)	NA	4.12, 4.13, 4.14, 4.15, 4.16	GA
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)	NA	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)	NA	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)	NA	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	NA	4.10, 4.11, 4.12, 4.13	GA

For questions and feedback, you can send an email to the product team at pipelines-interest@redhat.com.

1.2. MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message .

1.3. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.15

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.15 is available on OpenShift Container Platform 4.14 and later versions.

1.3.1. New features

In addition to fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.15:

1.3.1.1. Pipelines

- With this update, when incorporating a step from another custom resource (CR) using a **stepRef:** section, you can use parameters in the same way that you use parameters in **taskRef:** and **pipelineRef:** sections.

  Example usage

  ```
  apiVersion: tekton.dev/v1
  kind: Task
  metadata:
    name: test-task
  spec:
    steps:
    - name: fetch-repository
      stepRef:
        resolver: git
        params:
        - name: url
          value: https://github.com/tektoncd/catalog.git
        - name: revision
          value: main
        - name: pathInRepo
          value: stepaction/git-clone/0.1/git-clone
      params:
      - name: url
        value: $(params.repo-url)
      - name: revision
        value: $(params.tag-name)
      - name: output-path
        value: $(workspaces.output.path)
  ```

- Before this update, when using a resolver to incorporate a task or pipeline from a remote source, if one of the parameters expected an array you had to specify the type of the parameter explicitly. With this update, when using a resolver to incorporate a task or pipeline from a remote source, you do not have to set the type of any parameters.

- With this update, when specifying the use of a workspace in a pipeline run or task run, you can use parameters and other variables in the specification in the **secret**, **configMap**, and **projected.sources** sections.

  Example usage

  ```
  apiVersion: tekton.dev/v1
  kind: Task
  metadata:
    generateName: something-
  spec:
    params:
    - name: myWorkspaceSecret
    steps:
    - image: registry.redhat.io/ubi/ubi8-minimal:latest
  ```

```
    script: |
      echo “Hello World”
  workspaces:
  - name: myworkspace
    secret:
      secretName: $(params.myWorkspaceSecret)
```

- By default, when OpenShift Pipelines fails to pull the container image that is required for the execution of a task, the task fails. With this release, you can configure an image pull backoff timeout. If you configure this timeout, when OpenShift Pipelines fails to pull the container image that is required for the execution of a task, it continues to attempt to pull the image for the specified time period. The task fails if OpenShift Pipelines is unable to pull the image within the specified period.

  Example specification

  ```
  apiVersion: operator.tekton.dev/v1alpha1
  kind: TektonConfig
  metadata:
    name: config
  spec:
    pipeline:
      options:
        configMaps:
            config-defaults:
                data:
                     default-imagepullbackoff-timeout: "5m"
  ```

- With this release, the YAML manifest of a completed pipeline run or task run includes a **displayName** field in the **childReferences** section. This field contains the display name of the pipeline run or task run, which can differ from the full name of the pipeline run or task run.

- With this update, the YAML manifest of every step in a completed **TaskRun** CR includes a new **terminationReason** field. This field contains the reason the step execution ended. OpenShift Pipelines uses the following values for the **terminationReason** field:

  - **Completed**: The step completed successfully and any commands invoked in the step ended with exit code 0.

  - **Continued**: There was an error during execution of the step, for example a command returned a non-zero exit code, but the step execution continued because the **onError** value was set to **continue**. See the log output for the details of the error.

  - **Error**: There was an error during execution of the step, for example a command returned a non-zero exit code, and this error caused the step to fail. See the log output for the details of the error.

  - **TimeoutExceeded**: The execution of the step timed out. See the log output for the details of the timeout.

  - **Skipped**: The step was skipped because a previous step failed.

  - **TaskRunCancelled**: The task run was cancelled.

- With this update, you can use the **pipeline.disable-inline-spec** spec in the **TektonConfig** CR to disable specifying pipelines and tasks inside **PipelineRun** CRs, specifying tasks inside **Pipeline**

CRs, or specifying tasks inside **TaskRun** CRs. If you use this option, you must refer to pipelines by using the **pipelineRef:** specification and refer to tasks by using the **taskRef:** specification.

- With this update, some metrics for Prometheus monitoring of OpenShift Pipelines were renamed to ensure compliance with the Prometheus naming convention. Gauge and Counter metric names no longer end with **count**.

1.3.1.2. Operator

- With this update, several tasks are added to the **openshift-pipelines** namespace in the **resolverTasks** add-on. You can incorporate these tasks in your pipelines using the cluster resolver. Most of these tasks were previously available as cluster tasks (**ClusterTask** resources). You can access the following tasks by using the cluster resolver:
  - **buildah**
  - **git-cli**
  - **git-clone**
  - **kn**
  - **kn-apply**
  - **maven**
  - **openshift-client**
  - **s2i-dotnet**
  - **s2i-go**
  - **s2i-java**
  - **s2i-nodejs**
  - **s2i-perl**
  - **s2i-php**
  - **s2i-python**
  - **s2i-ruby**
  - **skopeo-copy**
  - **tkn**
- With this update, you can set the **pruner.startingDeadlineSeconds** spec in the **TektonConfig** CR. If the pruner job that removes old resources associated with pipeline runs and task runs is not started at the scheduled time for any reason, this setting configures the maximum time, in seconds, in which the job can still be started. If the job is not started within the specified time, OpenShift Pipelines considers this job failed and starts the pruner at the next scheduled time.
- With this update, you can use the **targetNamespaceMetadata** spec in the **TektonConfig** CR to set labels and annotations for the **openshift-pipelines** namespace in which the Operator installs OpenShift Pipelines.

- With this update, error messages for the OpenShift Pipelines Operator include additional context information such as namespace.

1.3.1.3. Triggers

- With this update, you can use the **TriggerTemplate** CR to specify templates for any types of resources. When the trigger is invoked, OpenShift Pipelines creates the resources that you define in the **TriggerTemplate** CR for the trigger. In the following example, a **ConfigMap** resource is created when the trigger is invoked:

  Example TriggerTemplate CR

```
apiVersion: triggers.tekton.dev/v1beta1
kind: TriggerTemplate
metadata:
  name: create-configmap-template
spec:
  params:
    - name: action
  resourcetemplates:
    - apiVersion: v1
      kind: ConfigMap
      metadata:
        generateName: sample-
      data:
        field: "Action is : $(tt.params.action)"
```

- With this update, you can define the **ServiceType** in an **EventListener** CR as **NodePort** and define the port number for the event listener, as shown in the following example:

  Example EventListener CR defining a port number

```
apiVersion: triggers.tekton.dev/v1beta1
kind: EventListener
metadata:
  name: simple-eventlistener
spec:
  serviceAccountName:  simple-tekton-robot
  triggers:
    - name: simple-trigger
      bindings:
      - ref: simple-binding
      template:
        ref: simple-template
  resources:
    kubernetesResource:
      serviceType: NodePort
      servicePort: 38080
```

- With this update, if you use a **serviceType** value of **LoadBalancer** in an **EventListener** CR, you can optionally specify a load balancer class in the **serviceLoadBalancerClass** field. If your cluster provides multiple load balancer controllers, you can use the load balancer class to select one of these controllers. For more information about setting a load balancer class, see the Kubernetes documentation.

Example specifying a LoadBalancerClass setting

```
apiVersion: triggers.tekton.dev/v1beta1
kind: EventListener
metadata:
  name: listener-loadbalancerclass
spec:
  serviceAccountName: tekton-triggers-example-sa
  triggers:
    - name: example-trig
      bindings:
        - ref: pipeline-binding
        - ref: message-binding
      template:
        ref: pipeline-template
  resources:
    kubernetesResource:
      serviceType: LoadBalancer
      serviceLoadBalancerClass: private
```

1.3.1.4. Manual approval

With this update, OpenShift Pipelines includes the new Manual Approval Gate functionality.

Manual Approval Gate is a custom resource definition (CRD) controller. You can use this controller to add manual approval points in the pipeline so that the pipeline stops at that point and waits for a manual approval before continuing execution. To use this feature, specify an **ApprovalTask** in the pipeline, in a similar way to specifying a **Task**. Users can provide the approval by using the web console or by using the **opc** command line utility.

The Manual Approval Gate controller includes the following features:

- You must set the following parameters in the **ApprovalTask** specification:
  - **approvers**: The users who can approve or reject the **approvalTask** to unblock the pipeline
  - **numberOfApprovalsRequired**: The number of approvals required to unblock the pipeline
  - **description**: (Optional) Description of the **approvalTask** that OpenShift Pipelines displays to the users
- The manual approval gate supports approval from multiple users:
  - The approval requires the configured minimum number of approvals from the configured users. Until this number is reached, the approval task does not finish and its **approvalState** value remains **pending**.
  - If any one approver rejects the approval, the **ApprovalTask** controller changes the **approvalState** of the task to **rejected** and the pipeline run fails.
  - If one user approves the task but the configured number of approvals is still not reached, the same user can change to rejecting the task and the pipeline run fails.
- Users can provide approval using the **opc approvaltask** CLI and the OpenShift web console. Approval in the OpenShift web console requires installation of the OpenShift Pipelines web console plugin. This plugin requires OpenShift Container Platform version 4.15 or later.

- Users can add messages while approving or rejecting the **approvalTask**.
- You can add a timeout setting to the **approvalTask** specification. If the required number of approvals is not provided during this time period, the pipeline run fails.

IMPORTANT

The manual approval gate is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

1.3.1.5. CLI

- With this update, the **tkn** command line utility supports the **-E** or **--exit-with-pipelinerun-error** option for the **pipeline showlog** command. With this option, the command line utility returns an error code of **0** if the pipeline run completed successfully, **1** if the pipeline run ended with an error, and **2** if the status of the pipeline run is unknown.
- With this update, the **tkn** command line utility supports the **--label** option for the **bundle push** command. With this option, you can provide the value of a label in the **<label-name>=<value>** format; the utility adds the label to the OCI image that it creates. You can use this option several times to provide several labels for the same image.

1.3.1.6. Pipelines as Code

- With this update, when using Pipelines as Code, you can set a **pipelinesascode.tekton.dev/on-comment** annotation on a pipeline run to start the pipeline run when a developer adds a matching comment to a pull request. This setting is supported only for pull requests and only for GitHub and GitLab repository providers.

IMPORTANT

Matching a comment event to a pipeline run is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- With this update, when using Pipelines as Code, you can enter the **/test <pipeline_run_name>** comment on a pull request to start any Pipelines as Code pipeline run on the repository, whether or not it was triggered by an event for this pipeline run. This feature is Technological preview only.
- With this update, when providing a **/test** or **/retest** command for Pipelines as Code in a Git request comment, you can now set any standard or custom parameters for the pipeline run.

Example commands in a Git request comment

```
/test pipelinerun1 revision=main param1="value1" param2="value \"value2\" with quotes"
```

This command runs the **pipelinerun1** pipeline run on the **main** branch instead of the pull request branch.

```
/test checker target_branch=backport-branch
```

This command runs the **checker** pipeline run on a backport (cherry-pick) of the pull request to the **backport-branch** branch.

- With this update, when using Pipelines as Code, you can create a global **Repository** CR with the name **pipelines-as-code** in the namespace in which OpenShift Pipelines is installed, normally **openshift-pipelines**. In this CR, you can set the configuration options that apply to all **Repository** CRs. You can override any of these default options by setting different values in the **Repository** CR for a particular repository.

> IMPORTANT
>
> The global **Repository** CR is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.
>
> For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- With this update, Pipelines as Code processes both the **OWNERS** file and the **OWNERS_ALIASES** file when determining which users can trigger pipeline runs. However, if the **OWNERS** file includes a **filters** section, Pipelines as Code matches approvers and reviewers only against the **.*** filter.

- With this update, when Pipelines as Code generates a random secret name for storing the GitHub temporary token, it uses two additional random characters. This change decreases the probability of a collision in secret names.

- With this update, when a pipeline run defined by using Pipelines as Code causes a YAML validation error, OpenShift Pipelines reports the error and the pipeline run name in the event log of the user namespace where the pipeline run executes, as well as in the OpenShift Pipelines controller log. The error report is also displayed in the Git repository provider, for example, in the GitHub CheckRun user interface. With this change, a user who does not have access to the controller namespace can access the error messages.

1.3.1.7. Tekton Results

- Tekton Results uses an **UpdateLog** operation to store logging information in the database. With this update, you can use the TektonResult CR to specify a timeout value for this operation. If the operation does not complete within the specified time period, Tekton Results ends the operation.

Example specification

```
apiVersion: operator.tekton.dev/v1
kind: TektonResult
metadata:
  name: result
spec:
     options:
        deployments:
           tekton-results-watcher:
              spec:
                  template:
                     spec:
                        containers:
                        - name: watcher
                          args:
                          - "--updateLogTimeout=60s"
```

- With this update, when configuring Tekton Results, you can optionally specify the following database configuration settings in the **options.configMaps.tekton-results-api-config.data.config** section of the **TektonResult** CR:
  - **DB_MAX_IDLE_CONNECTIONS**: The maximum number of idle connections to the database server that can remain open
  - **DB_MAX_OPEN_CONNECTIONS**: The maximum total number of connections to the database server that can remain open
  - **GRPC_WORKER_POOL**: The size of the GRPC worker pool
  - **K8S_QPS**: The Kubernetes client QPS setting
  - **K8S_BURST**: The Kubernetes client burst QPS setting
    If you want to use this setting, when configuring Tekton Results you must also use alternate specs for several other configuration parameters, as listed in the following table. Both the regular and the alternate parameter specs are in the **TektonResult** CR.

Table 1.2. Alternate configuration parameters for Tekton Results

Regular parameter spec	Alternate parameter spec
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**

Regular parameter spec	Alternate parameter spec
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**

For the configuration parameters not listed in this table, use the regular specs as described in the documentation.

> IMPORTANT
>
> Use the alternate parameter specs only if you need to use the additional settings in the **options.configMaps.tekton-results-api-config.data.config** section of the **TektonResult** CR.

- With this update, you can use the Tekton Results API to retrieve the Go profiling data for Tekton Results.

- Before this update, Tekton Results checked the user authentication when displaying every fragment of log data. With this update, Tekton Results checks the user authentication only once per log data request. This change improves the response time for the Tekton Results log API, which is used for displaying logs using the command line utility.

1.3.2. Breaking changes

- With this update, the OpenShift Pipelines console plugin, which is required for viewing pipeline and task execution statistics in the web console and for using the manual approval gate, requires OpenShift Container Platform version 4.15 or a later version.

- Before this update, Pipelines as Code set the **git-provider**, **sender**, and **branch** labels in a pipeline run. With this update, Pipelines as Code no longer sets these labels, Instead, it sets the **pipelinesascode.tekton.dev/git-provider**, **pipelinesascode.tekton.dev/sender**, and **pipelinesascode.tekton.dev/branch** annotations.

- With this update, you can no longer use the **jaeger** exporter for OpenTelemetry tracing. You can use the **oltptraceexporter** for tracing.

1.3.3. Known issues

- The new **skopeo-copy** task, which is available from the **openshift-pipelines** namespace by using the cluster resolver, does not work when the **VERBOSE** parameter is set to **false**, which is the default setting. As a workaround, when you use this task, set the **VERBOSE** parameter to **true**. The issue does not apply to the **skopeo-copy ClusterTask**.

- The new **skopeo-copy** task, which is available from the **openshift-pipelines** namespace by using the cluster resolver, fails when you use it to push or pull an image to or from an OpenShift Container Platform internal image repository, such as **image-registry.openshift-image-registry.svc:5000**. As a workaround, set the **DEST_TLS_VERIFY** or **SRC_TLS_VERIFY** parameter to **false**. Alternatively, use an external image repository that has a valid SSL certificate. The issue does not apply to the **skopeo-copy ClusterTask**.

- The new **s2i-*** tasks, which are available from the **openshift-pipelines** namespace by using the cluster resolver, fail if you clone a Git tepository to a subdirectory of the **source** workspace and then set the **CONTEXT** parameter of the task. As a workaround, when you use these tasks, do not set the **CONTEXT** parameter. The issue does not apply to the **s2i-* ClusterTasks**.

- The new **git-clone** task, which is available from the **openshift-pipelines** namespace by using the cluster resolver, does not set the **COMMIT** result value. The issue does not apply to the **git-clone ClusterTask**.

- The **jib-maven ClusterTask** does not work if you are using OpenShift Container Platform version 4.16.

- When using Pipelines as Code, if you set the **concurrency_limit** spec in the global **Repository** CR named **pipelines-as-code** in the **openshift-pipelines** namespace, which provides default settings for all **Repository** CRs, the Pipelines as Code watcher crashes. As a workaround, do not set this spec in this CR. Instead, set the **concurrency_limit** spec in the other **Repository** CRs that you create.

- When using Pipelines as Code, if you set the **settings.pipelinerun_provenance** spec in the global **Repository** CR named **pipelines-as-code** in the **openshift-pipelines** namespace, which provides default settings for all **Repository** CRs, the Pipelines as Code controller crashes. As a workaround, do not set this spec in this CR. Instead, set the **settings.pipelinerun_provenance** spec in the other **Repository** CRs that you create.

1.3.4. Fixed issues

- Before this update, many **info** messages about **ClusterTask** resources being repeatedly reconciled were present in the OpenShift Pipelines Operator log. With this update, the excessive reconciliation no longer happens and the excessive messages do not appear.

  If the reconciliation messages still appear, remove the earlier version of the **ClusterTask installerset** resource. However, if you remove the **installerset** resource, you cannot reference **ClusterTasks** with this specified version in your pipelines.

  Enter the following command to list **installerset** resources:

```
$ oc get tektoninstallersets
```

The names for versioned **ClusterTask installerset** resources are **addon-versioned-clustertasks-<version>-<unique_id>**, for example, **addon-versioned-clustertasks-1.12-fblb8**.

Enter the following command to remove an **installerset** resource:

```
$ oc delete tektoninstallerset <installerset_name>
```

- Before this update, if a task run or pipeline run referenced a service account and this service account referenced a secret that did not exist, the task run or pipeline run failed. With this update, the task run or pipeline run logs a warning and continues.

- Before this update, when you referenced a **StepAction** CR inside a step of a task, OpenShift Pipelines passed all parameters of the step to the **StepAction** CR. With this update, OpenShift Pipelines passes only the parameters defined in the **StepAction** CR to the step action.

- Before this update, if you defined a parameter of a task within a pipeline twice, OpenShift Pipelines logged the wrong path to the definition in the error message. With this update, the error message contains the correct path.

- Before this update, if you specified a task under the **finally:** clause of a pipeline, used an expression in the **when:** clause of this task, and referenced the status of another task in this expression (for example, **'$(tasks.a-task.status)' == 'Succeeded'**), this expression was not evaluated correctly. With this update, the expression is evaluated correctly.

- Before this update, if you specified a negative number of retries when specifying a task run, OpenShift Pipelines did not detect the error. With this update, OpenShift Pipelines detects and reports this error.

- Before this update, when you use a **pipelineRef:** section inside a task of a pipeline to reference another pipeline or when you use a **pipelineSpec:** section inside a task of a pipeline to specify another pipeline, the OpenShift Pipelines controller could crash. With this update, the crash does not happen and the correct error message is logged. Use of **pipelineRef:** and **pipelineSpec:** sections inside a pipeline is not supported.

- Before this update, when you configured a task to use a workspace using the **workspace.<workspace_name>.volume** keyword and then the task failed and was retried, creation of the pod for the task failed. With this update, the pod is created successfully.

- Before this update, OpenShift Pipelines sometimes modified recorded annotations on a completed pipeline run or task run after its completion. For example, the **pipeline.tekton.dev/release** annotation records the version information of the pipeline, and if the pipeline version was updated after the execution of the pipeline run, this annotation could be changed to reflect the new version instead of the version that was run. With this update, the annotations reflect the status of the pipeline run when it was completed and OpenShift Pipelines does not modify the annotations later.

- Before this update, if a YAML manifest that a pipeline run uses (for example, the manifest of a task or pipeline) had syntax errors, the logged error message was unspecific or no error message was logged. With this update, the logged error message includes the syntax errors.

- Before this update, when you used the **buildah** cluster task with a secret with the **.dockerconfigjson** file provided using a workspace, the task failed during the **cp** command because the /**root/.docker** directory did not exist. With this update, the task completes successfully.

- Before this update, if a pipeline run timed out and a **TaskRun** or **CustomRun** resource that this pipeline run included was deleted, the pipeline run execution was blocked and never completed. With this update, the execution correctly ends, logging a canceled state.

- Before this update, when using a resolver to incorporate a task from a remote source, the resolver automatically added the **kind** value of **Task** to the resulting specification. With this update, the resolver does not add a **kind** value to the specification.

- Before this update, when you set configuration options using an **options:** section in the **TektonConfig** CR, these options were sometimes not applied correctly. With this update, the options are applied correctly.

- Before this update, if you set the **enable-api-fields** field and certain other fields in the **TektonConfig** CR, the settings were lost after any update of OpenShift Pipelines. With this update, the settings are preserved during updates.

- Before this update, if you configured the horizontal pod autoscaler (HPA) using the options section in the **TektonConfig** CR, any existing HPA was updated correctly but a new HPA was not created when required. With this update, HPA configuration using the options section works correctly.

- Before this update, you could erroneously change the **targetNamespace** field in the **TektonConfig** CR, creating an unsupported configuration. With this update, you can no longer change this field. Changing the target namespace name from **openshift-pipelines** is not supported.

- Before this update, if the **pipelines-scc-rolebinding** rolebinding was missing or deleted in any namespace, the OpenShift Pipelines Operator controller failed to create default resources in new namespaces correctly. With this update, the controller functions correctly.

- Before this update, when you specified a **namespaceSelector** value when defining a **triggerGroup** in an **EventListener** CR, the event listener was unable to access triggers in the specified namespace if it was not the same as the namespace of the event listener. With this update, the event listener can access triggers in the specified namespace.

- Before this update, when a request was sent to an **EventListener** route URL with a **Content-Type** header, this header was not passed to the interceptor. With this update, the header is passed to the interceptor.

- With this update, several potential causes for Tekton Results becoming unresponsive, crashing, or consuming a large amount of memory were removed.

- Before this update, in the **Pipeline details** page of the web console, if a **when** expression using CEL was configured for a task, information was not displayed correctly. With this update, the information is displayed correctly.

- Before this update, in the **Pipeline details** page of the web console, the menu was not visible when you enabled dark mode in the web console. With this update, the menu is visible.

- Before this update, in the **Pipelines** page of the web console, information about running statistics of pipelines did not include the information saved in Tekton Results. With this update, the page includes all available running statistics information for every pipeline.

- Before this update, when you viewed a list of tasks for a namespace in the web console, a task from another namespace was sometimes displayed in the list. With this update, the web console correctly lists tasks for each namespace.

- Before this update, when you viewed the list of task runs in the web console, the status for each task run was not displayed. With this update, the list of task runs in the web console includes the status for each task run.

- Before this update, if you disabled cluster tasks in your OpenShift Pipelines deployment, the Pipeline Builder in the web console did not work. With this update, if you disable cluster tasks, the Pipeline Builder in the web console works correctly.

- Before this update, the OpenShift Pipelines console plugin pod did not move to the node specified using the **nodeSelector**, **tolerations**, and **priorityClassName** settings. With this update, the OpenShift Pipelines plugin pod moves to the correct node.

- Before this update, the Pipelines as Code watcher sometimes crashed when processing a pipeline run for which a concurrency limit was not set. With this update, these pipeline runs are processed correctly.

- Before this update, in Pipelines as Code, a concurrency limit setting of **0** was not interpreted as disabling the concurrency limit. With this update, a concurrency limit setting of **0** disables the concurrency limit.

- Before this update, when you defined annotations and labels for a task in Pipelines as Code, the annotations and labels were not set on the pod that is running the task. With this update, Pipelines as Code correctly sets the configured annotations and labels on the pod that is running the task.

- Before this update, Pipelines as Code sometimes caused a load on the Kubernetes service by re-reading an internal configuration **ConfigMap** resource frequently. With this update, Pipelines as Code no longer causes this load, because it reloads the **ConfigMap** resource only after the **ConfigMap** resource is modified.

- Before this update, when using Pipelines as Code, when you deleted a comment on a pull request such as **/test** or **/retest**, Pipelines as Code executed the command in the comment again. With this update, Pipelines as Code executes a command only when you add the comment.

- Before this update, when using Pipelines as Code, if some pipeline runs for a pull request failed and then re-ran successfully after a **/test** or **/retest** command without pushing a new commit, the user interface of the Git provider, such as GitHub, displayed the previous failure result along with the new result. With this update, the up-to-date status is displayed.

- Before this update, when you used the **tkn pr logs -f** command to view the logs for a running pipeline, the command line utility stopped responding, even if the pipeline run completed successfully. With this update, the **tkn pr logs -f** command properly displays the log information and exits.

1.4. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.14

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.14 is available on OpenShift Container Platform 4.12 and later versions.

1.4.1. New features

In addition to fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.14:

1.4.1.1. Pipelines

- With this update, you can use a parameter of a task or pipeline or a result of a previous task to specify the name of a resource to bind to a workspace, for example, **name: $(params.name)-configmap**.

- With this update, OpenShift Pipelines supports using your existing entitlements for Red Hat Enterprise Linux in build processes within your pipelines. The built-in **buildah** cluster task can now use these entitlements.

- With this update, if a pipeline run or task run uses the **pipeline** service account, you can use CSI volume types in the pipeline or task.

- With this update, you can use a **StepAction** custom resource (CR) to define a reusable scripted action that you can invoke from any number of tasks. To use this feature, you must set the **pipeline.options.configMaps.feature-flags.data.enable-step-actions** spec in the **TektonConfig** CR to **true**.

- With this update, object parameters and array results are enabled by default. You do not need to set any flags to use them.

- With this update, you can use the HTTP resolver to fetch a pipeline or task from an HTTP URL, as shown in the following examples:

  Example usage for a task

  ```
  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
  ```

  Example usage for a pipeline

  ```
  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
  ```

- With this update, you can use an enum declaration to limit the values that you can supply for a parameter of a pipeline or task, as shown in the following example. To use this feature, you must set the **pipeline.options.configMaps.feature-flags.data.enable-param-enum** spec in the **TektonConfig** CR to **true**.

  ### Example usage

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

- With this update, when using the Git resolver with the authenticated source control management (SCM) API, you can override the default token, SCM type, and server URL that you configured. See the following example:

  ### Example usage

  ```
  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
  ```

- With this update, you can define default resource requirements for containers and init-containers in pods that OpenShift Pipelines creates when executing task runs. Use the **pipeline.options.configMaps.config-defaults.default-container-resource-requirements** spec in the **TektonConfig** CR to set these requirements. You can set the default values for all containers and also for particular containers by name or by prefix, such as **sidecar-***.

1.4.1.2. Operator

- With this update, OpenShift Pipelines supports horizontal pod autoscaling for the Operator proxy webhook. If the pod that runs the Operator proxy webhook reaches 85% CPU utilization, the autoscaler creates another replica of the pod. If you want to use more than one replica for the Operator proxy webhook at startup, you must configure this number in the **options.horizontalPodAutoscalers** spec of the **TektonConfig** CR.

- With this update, the internal leader election for several components of OpenShift Pipelines was improved. The Operator controller, Operator webhook, proxy webhook, Pipelines as Code watcher, Pipelines as Code webhook, and the Tekton Chains controller now use separate leader election ConfigMaps. The leader election affects which replica of a component processes a request.

- Before this update, when you scaled up the number of replicas of the OpenShift Pipelines controller, manual intervention was required to enable the use of the new replicas; namely, you needed to delete leases in the leader election. With this update, when you scale up the number of replicas of the OpenShift Pipelines controller, the leader election includes the new replicas automatically, so the new replicas can process information.

- With this update, you can optionally set the following flags in the **spec.pipeline** spec of the **TektonConfig** CR:

    - **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**

1.4.1.3. Triggers

- With this update, when specifying CEL expressions for the Triggers interceptor, you can use the **first** and **last** functions to access values in a JSON array.

- With this update, when specifying CEL expressions for the Triggers interceptor, you can use the **translate** function that facilitates the utilization of regular expressions to replace characters with specified strings, as in the following example:

Sample use of the translate function

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

Sample input string

```
This is $an Invalid5String
```

Sample result string

```
ABChisABCisABCanABCnvalid5ABCtring
```

1.4.1.4. Web console

- With this update, you can enable the web console plugin for OpenShift Pipelines. If you enable the plugin, you can view pipeline and task execution statistics in the **Pipelines overview** page and in the page of a pipeline. You must install Tekton Results to view this information.

> NOTE
>
> To use the web console plugin for OpenShift Pipelines, you must use, at a minimum, the following OpenShift Container Platform releases:
>
> - For OpenShift Container Platform version 4.12: 4.12.51
> - For OpenShift Container Platform version 4.13: 4.13.34
> - For OpenShift Container Platform version 4.14: 4.14.13
> - For OpenShift Container Platform version 4.15: any release

- With this update, if you are using OpenShift Container Platform 4.15 and you enabled the console plugin, you can view archive information about past pipeline runs and task runs. Tekton Results provides this information.

- With this update, the **PipelineRun** details page, accessible from both the **Developer** or **Administrator** perspective of the web console, introduces a **Vulnerabilities** row. This new row offers a visual representation of identified vulnerabilities, categorized by severity (critical, high, medium, and low). To enable this feature, update your tasks and associated pipelines to the specified format. Additionally, once enabled, you can also access the information about identified vulnerabilities through the **Vulnerabilities** column in the pipeline run list view page.

- With this update, the **PipelineRun** details page, accessible from both the **Developer** or **Administrator** perspective of the web console, provides an option to download or view Software Bill of Materials (SBOMs) for enhanced transparency and control. To enable this feature, update your tasks and associated pipelines to the specified format.

1.4.1.5. CLI

- With this update, the **tkn version** command displays the version of the Tekton Hub component if this component is installed.

- With this update, you can use the **tkn customrun list** command to list custom runs.

- With this update, when using the **tkn task start** command, you can specify a URL for an OCI image in the **-i** or **--image** argument. The command pulls the image and runs the specified task from this image.

- With this update, the **opc version** command displays the version of the Tekton Results CLI component, which is a part of the **opc** utility.

1.4.1.6. Pipelines as Code

- With this update, when using Pipelines as Code, you can specify the **pipelinesascode.tekton.dev/pipeline** annotation on a pipeline run to fetch the pipeline from a Tekton Hub instance. The value of this annotation must refer to a single pipeline on Tekton Hub.

- With this update, you can deploy an additional Pipelines as Code controller with different configuration settings and different secrets. You can use multiple Pipelines as Code controllers to interact with multiple GitHub instances.

- With this update, Pipelines as Code includes metrics publication for the GitLab and BitBucket providers. You can access the metrics using the **/metrics** path on the Pipelines as Code controller and watcher service, port 9090.

- With this update, when specifying the conditions for executing a pipeline run using a CEL expression with **pipelinesascode.tekton.dev/on-cel-expression** , you can check for existence of files in the Git repository:

  - **files.all.exists(x, x.matches('<path_or_regular_expression>'))** for all files

  - **files.added.exists(x, x.matches('<path_or_regular_expression>'))** for files that were added since the last run of this pipeline

  - **files.modified.exists(x, x.matches('<path_or_regular_expression>'))** for files that were modified since the last run of this pipeline

  - **files.deleted.exists(x, x.matches('<path_or_regular_expression>'))** for files that were deleted since the last run of this pipeline

  - **files.renamed.exists(x, x.matches('<path_or_regular_expression>'))** for files that were renamed since the last run of this pipeline; this expression checks the new names of the renamed files

1.4.1.7. Tekton Chains

- With this update, Tekton Chains supports the **v1** value of the API version.

- With this update, you can set the **artifacts.pipelinerun.enable-deep-inspection** parameter in the **TektonConfig** CR. When this parameter is **true**, Tekton Chains records the results of the child task runs of a pipeline run. When this parameter is **false**, Tekton Chains records the results of the pipeline run but not of its child task runs.

- With this update, you can set the **builddefinition.buildtype** parameter in the **TektonConfig** CR to set the build type for in-toto attestation. When this parameter is **https://tekton.dev/chains/v2/slsa**, Tekton Chains records in-toto attestations in strict conformance with the SLSA v1.0 specification. When this parameter is **https://tekton.dev/chains/v2/slsa-tekton**, Tekton Chains records in-toto attestations with additional information such as the labels and annotations in each task run and pipeline run, and also adds each task in a pipeline run under **resolvedDependencies**.

- Before this update, when Tekton Chains was configured to use **gcs** storage, Tekton Chains did not record pipeline run information. With this update, Tekton Chains records pipeline run information with this storage.

- With this update, performance metrics are available for Tekton Chains. To access the metrics, expose the **tekton-chains-metrics** service and then use the **/metrics** path on this service, port 9090. These metrics are also available in the OpenShift Container Platform Monitoring stack.

- With this update, Tekton Chains uses the new **v2alpha3** record format version when recording pipeline runs and task runs that use the **v1** version value.

- With this update, Tekton Chains uses the **v1** version of pipeline run and task run formats internally.

1.4.1.8. Tekton Results

- With this update, if Tekton Results is installed, Tekton Results records the summary and record data for pipeline runs started using Pipelines as Code.

- With this update, Tekton Results provides up to 100 megabytes of logging information for a pipeline or task.

- With this update, any authenticated user can view the **tekton-results-api-service** route in the openshift-pipelines namespace to interact with Tekton Results using a REST API.

- With this update, the Tekton Results API includes a new endpoint for fetching summary and aggregation for a list of records.

- With this update, the **GetLog** endpoint of the Tekton Results API returns raw bytes with the **text/plain** content type.

- With this update, you can optionally specify a custom CA certificate in the **options.configMaps.tekton-results-api-config.data.config.DB_SSLROOTCERT** spec in the **TektonResult** CR. In this case, Tekton Results requires an SSL connection to the database server and uses this certificate for the connection. If you want to use this setting, when configuring Tekton Results you must also use alternate specs for several other configuration parameters, as listed in the following table. Both the regular and the alternate parameter specs are in the **TektonResult** CR.

  Table 1.3. Alternate configuration parameters for Tekton Results

Regular parameter spec	Alternate parameter spec
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**

Regular parameter spec	Alternate parameter spec
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**

For the configuration parameters not listed in this table, use the regular specs as described in the documentation.

IMPORTANT

Use the alternate parameter specs only if you need to use the **DB_SSLROOTCERT** setting.

1.4.2. Breaking changes

- With this update, when using the Bundles resolver, you can no longer specify the **serviceAccount** parameter. Instead, you can specify the **secret** parameter to provide the name of a secret containing authentication information for the registry. You must update any tasks or pipelines that use the **serviceAccount** parameter of the Bundles resolver to use the **secret** parameter instead. The **pipeline.bundles-resolver-config.default-service-account** spec in the **TektonConfig** CR is no longer supported.

1.4.3. Known issues

- The **tkn pipeline logs -f** command does not display the logs of tasks that were defined in a pipeline with the **retries: X** parameter while this pipeline is in progress.

1.4.4. Fixed issues

- Before this update, when using GitHub Enterprise, an incoming webhook did not work. With this update, you can use incoming webhooks with GitHub Enterprise.

- Before this update, if a task run or pipeline run disabled timeouts, OpenShift Pipelines would run a series of rapid reconciliations on the task run or pipeline run, degrading the performance of the controller. With this update, the controller reconciles task runs and pipeline runs with disabled timeouts normally.

- Before this update, if you used a custom namespace to install Tekton Hub, the installation deleted the **openshift-pipelines** namespace, removing the OpenShift Pipelines installation. With this update, you can use a custom namespace to install Tekton Hub and your OpenShift Pipelines installation is unaffected

- Before this update, when using Pipelines as Code with GitLab, if the user triggered a pipeline run by using a comment in a merge request such as /**test**, Pipelines as Code did not report the status of the pipeline run on the merge request. With this update, Pipelines as Code correctly reports the status of the pipeline run on the merge request.

- Before this update, when using CEL filtering in Tekton Results with subgroups, as shown in the following example, the subgroups did not work correctly. With this update, subgroups work correctly.

  Example CEL filter with a subgroup

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

- Before this update, when a pipeline run was cancelled, Tekton Results did not record the logs of this pipeline run. With this update, Tekton Results records the logs of a cancelled pipeline run.

1.4.5. Release notes for Red Hat OpenShift Pipelines General Availability 1.14.1

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.14.1 is available on OpenShift Container Platform 4.12 and later versions.

1.4.5.1. Fixed issues

- Before this update, when using multiple Pipelines as Code controllers configured with different GitHub apps, the Pipelines as Code watcher component crashed with a **nilerror** message. With this update, Pipelines as Code functions normally with multiple controllers configured with different GitHub apps.

1.4.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.14.2

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.14.2 is available on OpenShift Container Platform 4.12 and later versions.

1.4.6.1. Fixed issues

- Before this update, when you started a pipeline run using Pipelines as Code, Tekton Results did not store information about this pipeline run. Because of this issue, the web console plugin did not include the pipeline run in the execution statistics display. With this update, Tekton Results stores information about Pipelines as Code pipeline runs and these pipeline runs are included in the execution statistics display.

- Before this update, when you started many pipeline runs using Pipelines as Code at the same time and these pipelines runs included a **max-keep-run** annotation, the Pipelines as Code watcher component failed to process some of the pending pipeline runs and they remained in a pending state. With this update, Pipelines as Code pipeline runs are processed correctly.

1.4.7. Release notes for Red Hat OpenShift Pipelines General Availability 1.14.3

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.14.3 is available on OpenShift Container Platform 4.12 and later versions.

1.4.7.1. Fixed issues

- Before this update, when you started many pipeline runs using Pipelines as Code at the same time and these pipelines runs included a **max-keep-run** annotation, the Pipelines as Code watcher was unable to reconcile the pipeline runs because of a race condition between deletion of existing pipeline runs and processing new pipeline runs. Because of this issue, some pipeline runs could not be processed. With this update, the Pipelines as Code watcher processes pipeline runs.

- Before this update, when you used the **tkn pr logs -f** command to view the logs for a running pipeline, the command line utility stopped responding, even if the pipeline run completed successfully. With this update, the **tkn pr logs -f** command properly displays the log information and exits.

1.4.8. Release notes for Red Hat OpenShift Pipelines General Availability 1.14.4

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.14.4 is available on OpenShift Container Platform 4.12 and later versions.

1.4.8.1. Fixed issues

- Before this update, a large number of error messages referencing **tekton-pipelines-webhook.ConversionWebhook** could be logged. With this update, unneeded conversion webhook configuration for the **ClusterTask** and **StepAction** Custom Resource Definitions (CRDs) was removed, and such error messages are no longer logged.

- Before this update, some configuration settings using the **options** sections in the **TektonConfig** custom resource (CR) did not work, because a race condition would occur if the same setting was configured in both the **options** section and another field in the **TektonConfig** CR. With this update, the settings work.

- Before this update, the OpenShift Pipelines console plugin pod did not move to the node specified using the **nodeSelector**, **tolerations**, and **priorityClassName** settings. With this update, the OpenShift Pipelines plugin pod moves to the correct node.

- Before this update, some error messages were logged in the operator controller logs without the proper context information. With this update, error messages contain the required information.

- Before this update, if the **pipelines-scc-rolebinding** rolebinding was missing or deleted in any namespace, the OpenShift Pipelines operator controller would fail to create default resources in new namespaces correctly. With this update, the controller functions correctly.

- Before this update, if you configured the Horizontal Pod Autoscaler (HPA) using the **options** section in the **TektonConfig** CR, any existing HPA was updated correctly but a new HPA was not created when required. With this update, Horizontal Pod Autoscaler configuration using the **options** section works correctly.

- Before this update, if a user or an OpenShift Pipelines controller used the OpenShift Pipelines API to modify a pipeline run that was in the process of being started by Pipelines as Code, Pipelines as Code could stop and the log contained "panic" messages. With this update, the pipeline being started by Pipelines as Code can be modified concurrently.

- Before this update, in Pipelines as Code, a concurrency limit setting of **0** was not interpreted as disabling the concurrency limit. With this update, a concurrency limit setting of **0** disables the concurrency limit.

1.5. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.13

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.13 is available on OpenShift Container Platform 4.12 and later versions.

1.5.1. New features

In addition to fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.13:

> NOTE
>
> Before upgrading to the Red Hat OpenShift Pipelines Operator 1.13, ensure that you installed at least the OpenShift Container Platform 4.12.19 or 4.13.1 version on your cluster.

1.5.1.1. Pipelines

- Before this update, the Source-to-Image (S2I) cluster tasks used a base S2I container image that was in Technology Preview. With this update, the S2I cluster tasks used a base S2I container image that is released and fully supported.

- With this update, optionally, you can enable a setting so that, when a task run is cancelled, OpenShift Pipelines stops the pod for the task run but does not delete the pod. To enable this setting, in the **TektonConfig** custom resource (CR) set the **pipeline.options.configMaps.feature-flags.data.keep-pod-on-cancel** spec to **true** and the **pipeline.enable-api-fields** spec to **alpha**.

- Before this update, you had to enable alpha features in order to set compute resource limits on the task level. With this update, you can use the **computeResources** spec for a **TaskRun** CR to set the resource limits for a task.

- With this update, when specifying a task and using the **displayName** parameter, you can use parameters that include the values of parameters, results, or context variables in the display name, for example, **$(params.application)**, **$(tasks.scan.results.report)**, **$(context.pipeline.name)**.

- With this update, when specifying a remote pipeline or task using the hub resolver, in the **version** parameter you can use inequation constraints such as **>=0.2.0,< 1.0.0**.

- With this update, when specifying a task you can use a Common Expression Language (CEL) expression in the **when** expression. To use this feature, you must set the **pipeline.options.configMaps.feature-flags.data.enable-cel-in-whenexpression** spec to **true** in the **TektonConfig** CR.

- With this update, when specifying a pipeline in a **PipelineRun** CR spec, you can reference the results produced by an inline task in a subsequent inline task.

  Example usage

```
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)
```

- With this update, when configuring the cluster resolver, you can set the value of the **blocked-namespaces** parameter to *. With this setting, only the namespaces listed in the **allowed-namespaces** parameter are allowed and all other namespaces are blocked.

1.5.1.2. Operator

- With this update, the **disable-affinity-assistant** feature flag has been deprecated and might be removed in a future release. Instead, in the **TektonConfig** CR, you can set the **pipeline.options.configMaps.feature-flags.data.coschedule** spec to one of the following values:
  - **workspaces**: OpenShift Pipelines schedules all task runs that share the same workspace to the same node if the workspace allocates a persistent volume claim. This is the default

setting.

  - **pipelineruns**: OpenShift Pipelines schedules all task runs in a pipeline run to the same node.

  - **isolate-pipelinerun**: OpenShift Pipelines schedules all task runs in a pipeline run to the same node and allows only one pipeline run to run on a node at the same time. This setting might delay pipeline runs if all nodes are used for other pipeline runs.

  - **disabled**: OpenShift Pipelines does not apply any specific policy about alocating task runs to nodes.

1.5.1.3. Triggers

- Before this update, the core interceptor always created TLS secrets when starting. With this update, the core interceptor creates TLS secrets if a TLS secret is not present on the cluster or when a certificate in the existing secret has expired.

1.5.1.4. CLI

- With this update, when you use the **tkn bundle push** command, the bundle is created with the creation time set to **1970-01-01T00:00:00Z** (Unix epoch time). This change ensures that bundle images created from the same source are always identical. You can use the **--ctime** parameter to set the creation time in the RFC3339 format. You can also use the **SOURCE_DATE_EPOCH** environment variable to set the creation time.

1.5.1.5. Pipelines as Code

- With this update, in Pipelines as Code, when using a CEL expression for advanced event matching (pipelinesascode.tekton.dev/on-cel-expression), you can use the **header** and **body** fields to access the full payload that is passed by the Git repository provider body. You can use this feature to filter events by any information that the Git repository sends.

  IMPORTANT

  Using the header and body of the payload in CEL expressions for event matching is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

  For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- With this update, when a Pipelines as Code pipeline run is triggered by a push event, you can use **/test**, **/test branch:<branchname>**, **/retest**, **/retest branch:<branchname>**, **/cancel**, and **/cancel branch:<branchname>** commands on the corresponding commit comment to re-run or cancel the pipeline run.

- With this update, when using Pipelines as Code, you can use remote tasks on remote pipelines. Therefore, you can reuse a complete remote pipeline across multiple repositories. You can override tasks from the remote pipeline by adding a task with the same name.

IMPORTANT

Using remote tasks on remote pipelines and overriding tasks is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- With this update, when using Pipelines as Code, you can view improved information about CI pipeline runs in the Git repository provider. This information includes the namespace and the associated pipeline run.

1.5.2. Breaking changes

- Before this update, in Pipelines as Code, when using a policy group, users who were not a part of policy groups and not allowed explicitly but allowed to run the CI (via org ownership or otherwise) could sometimes execute pipeline runs by creating events such as pull requests or by entering commands such as **ok_to_test**. With this update, if policy groups are configured, only users that are added to the required policy groups can execute pipeline runs, and users that are a part of the owner organization but not configured in policy groups cannot execute pipeline runs.

1.5.3. Known issues

- To enable keeping pods when a task run is cancelled, along with setting the **pipeline.options.configMaps.feature-flags.data.keep-pod-on-cancel** spec to **true** in the **TektonConfig** CR, you also need to set the **pipeline.enable-api-fields** spec to **alpha** in the **TektonConfig** CR.

- If you enable keeping pods when a task run is cancelled, when a task run is cancelled because of a default timeout or because you set a timeput in the pipeline specification, OpenShift Pipelines deletes the pod.

1.5.4. Fixed issues

- Before this update, a secret that a pipeline run uses for Git authentication could be deleted from the cluster during a cleanup. With this update, a secret is deleted only when all pipeline runs that use it are deleted.

- Before this update, in cases where multiple secrets shared the same prefix and were logged using the git interface, sometimes the concealing process started with a shorter secret, and a part of a longer secret could be displayed in the logs. With this update, when concealing secrets in logs, the process now starts from the longest secret, ensuring that no part of any secret is displayed in the logs.

- Before this update, if you specified a **results** spec for a pipeline, the pipeline run could wrongly fail with a mismatched types error. With this update, if you specify a **results** spec for a pipeline, the results provided by the pipeline are correctly processed.

- Before this update, when Tekton Chains was configured with KMS as Hashicorp Vault, the pod started crashing if there was an underlying error while connecting to Vault. This has now been fixed and the error is now recorded in the Tekton Chains controller log.
- Before this update, when using Tekton Chains, if you configured the **storage.oci.repository** parameter, errors were reported in the Tekton Chains controller log. With this update, the **storage.oci.repository** parameter is processed correctly.
- Before this update, when Tekton Chains was configured with the Hashicorp Vault KMS and there was an issue with the connection to Vault, the Tekton Chains controller pod could crash. With this update, the errors are processed and logged on the Tekton Chains controller log.

1.5.5. Release notes for Red Hat OpenShift Pipelines General Availability 1.13.1

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.13.1 is available on OpenShift Container Platform 4.12 and later versions.

1.5.5.1. Fixed issues

- Before this update, a task run sometimes failed with a **cannot stop the sidecar** error message. With this update, the race condition between controllers that caused this failure is fixed.
- Before this update, to enable keeping pods when a task run is cancelled, along with setting the **pipeline.options.configMaps.feature-flags.data.keep-pod-on-cancel** spec to **true** in the **TektonConfig** CR, you also need to set the **pipeline.enable-api-fields** spec to **alpha** in the **TektonConfig** CR. With this update, setting the **pipeline.options.configMaps.feature-flags.data.keep-pod-on-cancel** spec to **true** in the **TektonConfig** CR enables keeping pods when a task run is cancelled, and no additional setting is necessary.
- Before this update, if you defined a sidecar for a task, OpenShift Pipelines did not validate the container image in the definition when creating the **Task** and **TaskRun** custom resources (CRs). At run time, a sidecar with an invalid container image caused the task run to fail. With this update, OpenShift Pipelines validates the container image in the sidecar definition when creating the **Task** and **TaskRun** CRs.
- Before this update, the OpenShift Pipelines controller sometimes crashed when the task was evaluating parameters. With this update, the controller no longer crashes.
- Before this update, if the final task in a pipeline run failed or was skipped, OpenShift Pipelines sometimes reported a validation error for the pipeline run. With this update, OpenShift Pipelines reports the status of the pipeline run correctly.

1.6. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.12

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.12 is available on OpenShift Container Platform 4.12 and later versions.

1.6.1. New features

In addition to fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.12:

NOTE

Before upgrading to the Red Hat OpenShift Pipelines Operator 1.12, ensure that you have at least installed the OpenShift Container Platform 4.12.19 or 4.13.1 version on your cluster.

1.6.1.1. Pipelines

- With this update, the web console includes a new gauge metric for pipeline runs and task runs. This metric indicates whether the underlying pods are being throttled by OpenShift Container Platform either because of resource quota policies defined in the namespace or because of resource constraints on the underlying node.

- With this update, the new **set-security-context** feature flag is set to **true** by default, in order to enable task runs and pipeline runs to run in namespaces with restricted pod security admission policies.

- With this update, the **enable-api-fields** flag is set to **beta** by default. You can use all features that are marked as **beta** in the code without further changes.

- With this update, the **results.tekton.dev/*** and **chains.tekton.dev/*** reserved annotations are not passed from the pipeline run to the task runs that it creates.

- Before this update, CSI volumes and projected volumes were not enabled by default. With this update, you can use CSI volumes and projected volumes in your pipelines without changing any configuration fields.

- With this update, the isolated workspaces feature is enabled by default. You can use this feature to share a workspace with specified steps and sidecars without sharing it with the entire task run.

1.6.1.2. Operator

- With this update, you can configure the default security context constraint (SCC) for the pods that OpenShift Pipelines creates for pipeline runs and task runs. You can set the SCC separately for different namespaces and also configure the maximum (least restrictive) SCC that can be set for any namespace.

- With this update, a new **options:** heading is available under each component in the TektonConfig spec. You can use parameters under this headings to control settings for different components. In particular, you can use parameters under the **platforms.openshift.pipelinesAsCode.options.configMaps.pac-config-logging.data** spec to set logging levels for components of Pipelines as Code.

- With this update, you can use the new **spec.pipeline.performance.replicas** parameter to set the number of replicas that are created for the OpenShift Pipelines controller pod. If you previously set the replica counts in your deployment manually, you must now use this setting to control the replica counts.

- With this update, the Operator ensures that the stored API version remains the same throughout your deployment of OpenShift Pipelines. The stored API version in OpenShift Pipelines 1.12 is **v1**.

- With this update, you can use a secret to configure S3 bucket storage to store Tekton Results logging information. When configuring S3 bucket storage, you must provide the secret with the S3 storage credentials by using the new **secret_name** spec in the **TektonResult** custom resource (CR).

1.6.1.3. Tekton Results

IMPORTANT

Tekton Results is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- With this update, you can configure Tekton Results to store data in an external PostgreSQL server.

- With this update, you can use Google Cloud Storage (GCS) to store Tekton Results logging information. You can provide the secret with the GCS storage credentials and then provide the secret name, secret key, and bucket name in properties under the **TektonResult** spec. You can also use Workload Identity Federation for authentication.

- With this update, any service account authenticated with OpenShift Pipelines can access the **TektonResult** CR.

- With this update, Tekton Results includes cluster role aggregation for service accounts with admin, edit, and view roles. Cluster role binding is no longer required for these service accounts to access results and records using the Tekton Results API.

- With this update, you can configure pruning for each **PipelineRun** or **TaskRun** resource by setting a **prune-per-resource** boolean field in the **TektonConfig** CR. You can also configure pruning for each **PipelineRun** or **TaskRun** resource in a namespace by adding the **operator.tekton.dev/prune.prune-per-resource=true** annotation to that namespace.

- With this update, if there are any changes in the OpenShift Container Platform cluster-wide proxy, Operator Lifecycle Manager (OLM) recreates the Red Hat OpenShift Pipelines Operator.

- With this update, you can disable the pruner feature by setting the value of the **config.pruner.disabled** field to **true** in the **TektonConfig** CR.

1.6.1.4. Triggers

- With this update, you can configure readiness and liveness probes on **Trigger** CRs. You can also set the value of the failure threshold for the probes; the default value is 3.

- With this update, OpenShift Pipelines triggers add **Type** and **Subject** values when creating a response to a Cloud Events request.

1.6.1.5. CLI

- With this update, the **tkn pipeline logs** command displays the logs of a pipeline or task that is referenced using a resolver.

- [illegible]

- With this update, when entering the **tkn bundle push** command, you can use the **--annotate** flag to provide additional annotations.

1.6.1.6. Pipelines as Code

- With this update, a Pipelines as Code pipeline run can include remote tasks fetched from multiple Artifact Hub or Tekton Hub instances and from different catalogs in the same hub instance.

- With this update, you can use parameters under the **platforms.openshift.pipelinesAsCode.options.configMaps.pac-config-logging.data** spec in the **TektonConfig** CR to set logging levels for Pipelines as Code components.

- With this update, you can set policies that allow certain actions only to members of a team and reject the actions when other users request them. Currently, the **pull_request** and **ok_to_test** actions support setting such policies.

- With this update, you can pass arbitrary parameters in the incoming webhook as a JSON payload. OpenShift Pipelines passes these parameters to the pipeline run. To provide an additional security layer, you must explicitly define the permitted parameters in the **Repository** CR.

- With this update, matching a large set of pipeline runs with a large number of remote annotations in Pipelines as Code is optimized. Pipelines as Code fetches the remote tasks only for the pipeline run that has matched.

- With this update, you can use the **source_url** variable in a pipeline run template to retrieve information about the forked repository from where the event, such as a pull or push request, is triggered.

  #### Example usage

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

- With this update, if an authorized user provides an **ok-to-test** comment to trigger a pipeline run on the pull request from an unauthorized user and then the author makes further changes to the branch, Pipelines as Code triggers the pipelines. To disable triggering the pipeline until an authorized user provides a new **ok-to-test** comment, set the **pipelinesAsCode.settings.remember-ok-to-test** spec in the **TektonConfig** CR to false.

- With this update, on the GitHub status check page, the table that shows the status of all tasks includes the display name of every task.

- With this update, you can configure the **tags push** event in a pipeline run on GitLab.

- With this update, you can use the [illegible] fields in Pipelines as Code

- With this update, you can use the **target_url** and **source_url** fields in Pipelines as Code Common Expression Language (CEL) expression filtering annotations to filter the request for a specific target or source.

- With this update, when you configure fetching a remote GitHub URL using a token, you can include a branch name that contains a slash. You must encode the slash within the branch name as **%2F** to ensure proper parsing by Pipelines as Code, as in the following example URL: **https://github.com/organization/repository/blob/feature%2Fmainbranch/path/file**. In this example URL, the branch name is **feature/mainbranch** and the name of the file to fetch is **/path/file**.

- With this update, you can use **--v1beta1** flag in the **tkn pac resolve** command. Use this flag if the pipeline run is generated with the **v1beta1** API version schema.

1.6.2. Breaking changes

- With this update, you cannot use the **openshift-operators** namespace as the target namespace for installing OpenShift Pipelines. If you used the **openshift-operators** namespace as the target namespace, change the target namespace before upgrading to Red Hat OpenShift Pipelines Operator version 1.12. Otherwise, after the upgrade, you will not be able to change any configuration settings in the **TektonConfig** CR except the **targetNamespace** setting.

- With this update, the new **spec.pipeline.performance.replicas** parameter controls the number of replicas that is created for every pod for a pipeline run or task run. If you previously set the replica counts in your deployment manually, after upgrading to OpenShift Pipelines version 1.12 you must use this parameter to control the replica counts.

- With this update, the following parameters are no longer supported in the **TektonResult** CR:

  - **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**
    You must provide these parameters using secrets. After upgrading to OpenShift Pipelines version 1.12, you must delete and re-create the **TektonResult** CR to provide these parameters.

- With this update, the **tkn pac bootstrap** command supports the **--github-hostname** flag. The **--github-api-url** flag is deprecated.

1.6.3. Known issues

- If limit ranges are configured for a namespace, but pod ephemeral storage is not configured in the limit ranges, pods can go into an error stage with the message **Pod ephemeral local storage usage exceeds the total limit of containers 0**.

- If you want to make changes to the configuration in the **TektonResult** CR, you must delete the existing **TektonResult** CR and then create a new one. If you change an existing **TektonResult** CR, the changes are not applied to the existing deployment of Tekton Results. For example, if you change the connection from an internal database server to an external one or vice versa, the API remains connected to the old database.

1.6.4. Fixed issues

- Before this update, Pipelines as Code ran pipeline runs based only on branch base names, and could incorrectly trigger pipeline runs with the same base name but different branch name. With this update, Pipelines as Code checks both the base name and the exact branch name of a pipeline run.

- Before this update, an incoming webhook event could trigger multiple pipeline runs configured for other events. With this update, an incoming webhook event triggers only a pipeline run configured for the webhook event.

- With this update, the **pac-gitauth** secrets are now explicitly deleted when cleaning up a pipeline run, in case the **ownerRef** on the pipeline run gets removed.

- Before this update, when a task in a pipeline run failed with a reason message, the entire pipeline run failed with a **PipelineValidationFailed** reason. With this update, the pipeline run fails with the same reason message as the task that failed.

- Before this update, the **disable-ha** flag value was not correctly passed to the Pipelines controller, and the high availability (HA) functionality was never enabled. With this update, you can enable the HA functionality by setting the value of the **disable-ha** flag in the **TektonConfig** CR to **false**.

- Before this update, the **skopeo-copy** cluster task would fail when attempting to copy images mentioned in config map data. With this update, the **skopeo-copy** cluster task completes properly.

- With this update, a pipeline run automatically generated by the **tkn pac generate –language=java** command has correct annotations and parameter names.

- Before this update, only a user with the administrative permissions could successfully run the **tkn pac create repository** command. With this update, any authorized user can run the **tkn pac create repository** command.

- Before this update, the /**test <run-name>** and /**retest <run-name>** user comments, which specified a particular pipeline, did not trigger pipeline runs as expected. With this update, these comments trigger pipeline runs successfully.

- Before this update, if there were multiple pipeline runs in the **.tekton** folder with the **generateName** field and not the **Name** field, the pipeline runs failed. This update fixes the issue.

- Before this update, in Pipelines as Code when using GitLab, a pipeline run was triggered by any event in a merge request, including adding labels and setting status. With this update, the pipeline run is triggered only when there is an open, reopen, or push event. A comment containing the status of the checks is now posted on the merge request.

- Before this update, while a pipeline run was waiting for approval, the status of the check could be displayed as **skipped** in the checks section of GitHub and Gitea pull requests. With this update, the correct **pending approval** status is displayed.

- Before this update, the bundles resolver sometimes set the type to **Task** when attempting to retrieve a pipeline, leading to errors in retrieval. With this update, the resolver uses the correct type to retrieve a pipeline.

- This update fixes an error in processing the Common Expression Language (CEL) NOT operator when querying Tekton Results.

- This update fixed a **404** error response that was produced in the Tekton Results API when a **LIST** operation for records was requested and the specified result was **-**.

- Before this update, in an **EventListener** object, the **status.address.url** field was always set to the default port. With this update, the **status.address.url** field is set to match the port specified in the **spec.resources.kubernetesresource.serviceport** parameter.

- Before this update, if the GitHub API provided a paginated response, Pipelines as Code used only the first page of the response. With this update, all paginated responses are processed fully.

- Before this update, the Tekton Chains controller crashed when the host address of KMS Hashicorp Vault was configured incorrectly or when Tekton Chains was unable to connect to the KMS Hashicorp Vault. With this update, Tekton Chains logs the connection error and does not crash.

1.6.5. Release notes for Red Hat OpenShift Pipelines General Availability 1.12.1

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.12.1 is available on OpenShift Container Platform 4.12 and later versions.

1.6.5.1. Fixed issues

- Before this update, if you configured Pipelines as Code with the custom console driver to output to a custom console, the Pipelines as Code controller crashed in certain cases. After you pushed changes to a pull request, the CI status check for this pull request could remain as **waiting for status to be reported** and the associated pipeline run did not complete. With this update, the Pipelines as Code controller operates normally. After you push changes to a pull request, the associated pipeline run completes normally and the CI status check for the pull request is updated.

- Before this update, when using Pipelines as Code, if you created an access policy on the **Repository** custom resource (CR) that did not include a particular user and then added the user to the **OWNER** file in the Git repository, the user would have no rights for the Pipelines as Code CI process. For example, if the user created a pull request into the Git repository. the CI process would not run on this pull request automatically. With this update, a user who is not included in the access policy on the **Repository** CR but is included in the **OWNER** file is allowed to run the CI process for the repository.

- With this update, the HTTP/2.0 protocol is not supported for webhooks. All webhook calls to Red Hat OpenShift Pipelines must use the HTTP/1.1 protocol.

1.6.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.12.2

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.12.2 is available on OpenShift Container Platform 4.12 and later versions.

1.6.6.1. Fixed issues

- Before this update, the generated Git secret for the latest pipeline run was deleted when the **max-keep-runs** parameter was exceeded. With this update, the Git secret is no longer deleted on the latest pipeline run.
- With this update, the S2I cluster task uses a General Availability container image.

1.7. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.11

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.11 is available on OpenShift Container Platform 4.12 and later versions.

1.7.1. New features

In addition to fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.11:

> NOTE
>
> Before upgrading to the Red Hat OpenShift Pipelines Operator 1.11, ensure that you have at least installed the OpenShift Container Platform 4.12.19 or 4.13.1 version on your cluster.

1.7.1.1. Pipelines

- With this update, you can use Red Hat OpenShift Pipelines on the OpenShift Container Platform cluster that runs on ARM hardware. You have support for the **ClusterTask** resources where images are available and the Tekton CLI tool on ARM hardware.
- This update adds support for results, object parameters, array results, and indexing into an array when you set the **enable-api-fields** feature flag to **beta** value in the **TektonConfig** CR.
- With this update, propagated parameters are now part of a stable feature. This feature enables interpolating parameters in embedded specifications to reduce verbosity in Tekton resources.
- With this update, propagated workspaces are now part of a stable feature. You can enable the propagated workspaces feature by setting the **enable-api-fields** feature flag to **alpha** or **beta** value.
- With this update, the **TaskRun** object fetches and displays the init container failure message to users when a pod fails to run.
- With this update, you can replace parameters, results, and the context of a pipeline task while configuring a matrix as per the following guidelines:
  - Replace an array with an **array** parameter or a string with a **string**, **array**, or **object** parameter in the **matrix.params** configuration.
  - Replace a string with a **string**, **array**, or **object** parameter in the **matrix.include** configuration.
  - Replace the context of a pipeline task with another context in the **matrix.include** configuration.

- With this update, the **TaskRun** resource validation process also validates the **matrix.include** parameters. The validation checks whether all parameters have values and match the specified type, and object parameters have all the keys required.

- This update adds a new **default-resolver-type** field in the **default-configs** config map. You can set the value of this field to configure a default resolver.

- With this update, you can define and use a **PipelineRun** context variable in the **pipelineRun.workspaces.subPath** configuration.

- With this update, the **ClusterResolver**, **BundleResolver**, **HubResolver**, and **GitResolver** features are now available by default.

1.7.1.2. Triggers

- With this update, Tekton Triggers support the **Affinity** and **TopologySpreadConstraints** values in the **EventListener** specification. You can use these values to configure Kubernetes and custom resources for an **EventListener** object.

- This update adds a Slack interceptor that allows you to extract fields by using a slash command in Slack. The extracted fields are sent in the form data section of an HTTP request.

1.7.1.3. Operator

- With this update, you can configure pruning for each **PipelineRun** or **TaskRun** resource by setting a **prune-per-resource** boolean field in the **TektonConfig** CR. You can also configure pruning for each **PipelineRun** or **TaskRun** resource in a namespace by adding the **operator.tekton.dev/prune.prune-per-resource=true** annotation to that namespace.

- With this update, if there are any changes in the OpenShift Container Platform cluster-wide proxy, Operator Lifecycle Manager (OLM) recreates the Red Hat OpenShift Pipelines Operator.

- With this update, you can disable the pruner feature by setting the value of the **config.pruner.disabled** field to **true** in the **TektonConfig** CR.

1.7.1.4. Tekton Chains

- With this update, Tekton Chains is now generally available for use.

- With this update, you can use the skopeo tool with Tekton Chains to generate keys, which are used in the **cosign** signing scheme.

- When you upgrade to the Red Hat OpenShift Pipelines Operator 1.11, the previous Tekton Chains configuration will be overwritten and you must set it again in the **TektonConfig** CR.

1.7.1.5. Tekton Hub

IMPORTANT

Tekton Hub is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- This update adds a new **resource/<catalog_name>/<kind>/<resource_name>/raw** endpoint and a new **resourceURLPath** field in the **resource** API response. This update helps you to obtain the latest raw YAML file of the resource.

1.7.1.6. Tekton Results

IMPORTANT

Tekton Results is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- This update adds Tekton Results to the Tekton Operator as an optional component.

1.7.1.7. Pipelines as Code

- With this update, Pipelines as Code allows you to expand a custom parameter within your **PipelineRun** resource by using the **params** field. You can specify a value for the custom parameter inside the template of the **Repository** CR. The specified value replaces the custom parameter in your pipeline run. Also, you can define a custom parameter and use its expansion only when specified conditions are compatible with a Common Expression Language (CEL) filter.

- With this update, you can either rerun a specific pipeline or all pipelines by clicking the **Re-run all checks** button in the **Checks** tab of the GitHub interface.

- This update adds a new **tkn pac info** command to the Pipelines as Code CLI. As an administrator, you can use the **tkn pac info** command to obtain the following details about the Pipelines as Code installation:

    - The location where Pipelines as Code is installed.

    - The version number of Pipelines as Code.

    - An overview of the **Repository** CR created on the cluster and the URL associated with the repository.

    - Details of any installed GitHub applications.

    With this update, you can also specify a custom GitHub API URL by using the

With this command, you can also specify a custom GitHub API URL by using the **--github-api-url** argument.

- This update enables error detection for all **PipelineRun** resources by default. Pipelines as Code detects if a **PipelineRun** resource execution has failed and shows a snippet of the last few lines of the error. For a GitHub application, Pipelines as Code detects error messages in the container logs and exposes them as annotations on a pull request.

- With this update, you can fetch tasks from a private Tekton Hub instance attached to a private Git repository. To enable this update, Pipelines as Code uses the internal raw URL of the private Tekton Hub instance instead of using the GitHub raw URL.

- Before this update, Pipelines as Code provided logs that would not include the namespace detail. With this update, Pipelines as Code adds the namespace information to the pipeline logs so that you can filter them based on a namespace and debug easily.

- With this update, you can define the provenance source from where the **PipelineRun** resource definition is to be fetched. By default, Pipelines as Code fetches the **PipelineRun** resource definition from the branch where the event has been triggered. Now, you can configure the value of the **pipelinerun_provenance** setting to **default_branch** so that the **PipelineRun** resource definition is fetched from the default branch of the repository as configured on GitHub.

- With this update, you can extend the scope of the GitHub token at the following levels:

    - Repository-level: Use this level to extend the scope to the repositories that exist in the same namespace in which the original repository exists.

    - Global-level: Use this level to extend the scope to the repositories that exist in a different namespace.

- With this update, Pipelines as Code triggers a CI pipeline for a pull request created by a user who is not an owner, collaborator, or public member or is not listed in the **owner** file but has permission to push changes to the repository.

- With this update, the custom console setting allows you to use custom parameters from a **Repository** CR.

- With this update, Pipelines as Code changes all **PipelineRun** labels to **PipelineRun** annotations. You can use a **PipelineRun** annotation to mark a Tekton resource, instead of using a **PipelineRun** label.

- With this update, you can use the **pac-config-logging** config map for watcher and webhook resources, but not for the Pipelines as Code controller.

1.7.2. Breaking changes

- This update replaces the **resource-verification-mode** feature flag with a new **trusted-resources-verification-no-match-policy** flag in the pipeline specification.

- With this update, you cannot edit the Tekton Chains CR. Instead, edit the **TektonConfig** CR to configure Tekton Chains.

1.7.3. Deprecated and removed features

- This update removes support for the **PipelineResource** commands and references from Tekton CLI:

  - Removal of pipeline resources from cluster tasks
  - Removal of pipeline resources from tasks
  - Removal of pipeline resources from pipelines
  - Removal of resource commands
  - Removal of input and output resources from the **clustertask describe** command

- This update removes support for the **full** embedded status from Tekton CLI.
- The **taskref.bundle** and **pipelineref.bundle** bundles are deprecated and will be removed in a future release.
- In Red Hat OpenShift Pipelines 1.11, support for the **PipelineResource** CR has been removed, use the **Task** CR instead.
- In Red Hat OpenShift Pipelines 1.11, support for the **v1alpha1.Run** objects has been removed. You must upgrade the objects from **v1alpha1.Run** to **v1beta1.CustomRun** before upgrading to this release.
- In Red Hat OpenShift Pipelines 1.11, the **custom-task-version** feature flag has been removed.
- In Red Hat OpenShift Pipelines 1.11, the **pipelinerun.status.taskRuns** and **pipelinerun.status.runs** fields have been removed along with the **embedded-status** feature flag. Use the **pipelinerun.status.childReferences** field instead.

1.7.4. Known issues

- Setting the **prune-per-resource** boolean field does not delete **PipelineRun** or **TaskRun** resources if they were not part of any pipeline or task.
- Tekton CLI does not show logs of the **PipelineRun** resources that are created by using resolvers.
- When you filter your pipeline results based on the **order_by=created_time+desc&page_size=1** query, you get zero records without any **nextPageToken** value in the output.
- When you set the value of the **loglevel.pipelinesascode** field to **debug**, no debugging logs are generated in the Pipelines as Code controller pod. As a workaround, restart the Pipelines as Code controller pod.

1.7.5. Fixed issues

- Before this update, Pipelines as Code failed to create a **PipelineRun** resource while detecting the **generateName** field in the **PipelineRun** CR. With this update, Pipelines as Code supports providing the **generateName** field in the **PipelineRun** CR.
- Before this update, when you created a **PipelineRun** resource from the web console, all annotations would be copied from the pipeline, causing issues for the running nodes. This update now resolves the issue.
- This update fixes the **tkn pr delete** command for the **keep** flag. Now, if the value of the **keep** flag is equal to the number of the associated task runs or pipeline runs, then the command returns the exit code **0** along with a message.

- Before this update, the Tekton Operator did not expose the performance configuration fields for any customizations. With this update, as a cluster administrator, you can customize the following performance configuration fields in the **TektonConfig** CR based on your needs:
    - **disable-ha**
    - **buckets**
    - **kube-api-qps**
    - **kube-api-burst**
    - **threads-per-controller**
- This update fixes the remote bundle resolver to perform a case-insensitive comparison of the **kind** field with the **dev.tekton.image.kind** annotation value in the bundle.
- Before this update, pods for remote resolvers were terminated because of insufficient memory when you would clone a large Git repository. This update fixes the issue and increases the memory limit for deploying remote resolvers.
- With this update, task and pipeline resources of **v1** type are supported in remote resolution.
- This update reverts the removal of embedded **TaskRun** status from the API. The embedded **TaskRun** status is now available as a deprecated feature to support compatibility with older versions of the client-server.
- Before this update, all annotations were merged into **PipelineRun** and **TaskRun** resources even if they were not required for the execution. With this update, when you merge annotations into **PipelineRun** and **TaskRun** resources, the **last-applied-configuration** annotation is skipped.
- This update fixes a regression issue and prevents the validation of a skipped task result in pipeline results. For example, if the pipeline result references a skipped **PipelineTask** resource, then the pipeline result is not emitted and the **PipelineRun** execution does not fail due to a missing result.
- This update uses the pod status message to determine the cause of a pod termination.
- Before this update, the default resolver was not set for the execution of the **finally** tasks. This update sets the default resolver for the **finally** tasks.
- With this update, Red Hat OpenShift Pipelines avoids occasional failures of the **TaskRun** or **PipelineRun** execution when you use remote resolution.
- Before this update, a long pipeline run would be stuck in the running state on the cluster, even after the timeout. This update fixes the issue.
- This update fixes the **tkn pr delete** command for correctly using the **keep** flag. With this update, if the value of the **keep** flag equals the number of associated task runs or pipeline runs, the **tkn pr delete** command returns exit code **0** along with a message.

1.7.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.11.1

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.11.1 is available on OpenShift Container Platform 4.12 and later versions.

1.7.6.1. Fixed issues

- Before this update, a task run could fail with a mount path error message, when a running or pending pod was preempted. With this update, a task run does not fail when the cluster causes a pod to be deleted and re-created.

- Before this update, a shell script in a task had to be run as root. With this update, the shell script image has the non-root user ID set so that you can run a task that includes a shell script, such as the **git-clone** task, as a non-root user within the pod.

- Before this update, in Red Hat OpenShift Pipelines 1.11.0, when a pipeline run is defined using Pipelines as Code, the definition in the Git repository references the **tekton.dev/v1beta1** API version and includes a **spec.pipelineRef.bundle** entry, the **kind** parameter for the bundle reference was wrongly set to **Task**. The issue did not exist in earlier versions of Red Hat OpenShift Pipelines. With this update, the **kind** parameter is set correctly.

- Before this update, the **disable-ha** flag was not correctly passed to the **tekton-pipelines** controller, so the High Availability (HA) feature of Red Hat OpenShift Pipelines could not be enabled. With this update, the **disable-ha** flag is correctly passed and you can enable the HA feature as required.

- Before this update, you could not set the URL for Tekton Hub and Artifact Hub for the hub resolver, so you could use only the preset addresses of Tekton Hub and Artifact Hub. With this update, you can configure the URL for Tekton Hub and Artifact Hub for the hub resolver, for example, to use a custom Tekton Hub instance that you installed.

- With this update, the SHA digest of the **git-init** image corresponds to version 1.10.5, which is the current released version of the image.

- Before this update, the **tekton-pipelines-controller** component used a config map named **config-leader-election**. This name is the default value for knative controllers, so the configuration process for OpenShift Pipelines could affect other controllers and vice versa. With this update, the component uses a unique config name, so the configuration process for OpenShift Pipelines does not affect other controllers and is not affected by other controllers.

- Before this update, when a user without write access to a GitHub repository opened a pull request, Pipelines as Code CI/CD actions would show as **skipped** in GitHub. With this update, Pipelines as Code CI/CD actions are shown as **Pending approval** in GitHub.

- Before this update, Pipelines as Code ran CI/CD actions for every pull request into a branch that matched a configured branch name. With this update, Pipelines as Code runs CI/CD actions only when the source branch of the pull request matches the exact configured branch name.

- Before this update, metrics for the Pipelines as Code controller were not visible in the OpenShift Container Platform developer console. With this update, metrics for the Pipelines as Code controller are displayed in the developer console.

- Before this update, in Red Hat OpenShift Pipelines 1.11.0, the Operator always installed Tekton Chains and you could not disable installation of the Tekton Chains component. With this update, you can set the value of the **disabled** parameter to **true** in the **TektonConfig** CR to disable installation okindf Tekton Chains.

- Before this update, if you configured Tekton Chains on an older version of OpenShift Pipelines using the **TektonChain** CR and then upgraded to OpenShift Pipelines version 1.11.0, the configuration information was overwritten. With this update, if you upgrade from an older version of OpenShift Pipelines and Tekton Chains was configured in the same namespace where the **TektonConfig** is installed (**openshift-pipelines**), Tekton Chains configuration information is preserved.

1.7.7. Release notes for Red Hat OpenShift Pipelines General Availability 1.11.2

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.11.2 is available on OpenShift Container Platform 4.12 and later versions.

This update includes an updated version of the **tkn** command line tool. You can download the updated version of this tool at the following locations:

- Linux (x86_64, amd64)
- Linux on IBM zSystems and IBM® LinuxONE (s390x)
- Linux on IBM Power (ppc64le)
- Linux on ARM (aarch64, arm64)
- Windows
- macOS
- macOS on ARM

If you installed the **tkn** command line tool using RPM on Red Hat Enterprise Linux (RHEL), use the **yum update** command to install the updated version.

1.7.7.1. Fixed issues

- Before this update, the **tkn pac resolve -f** command did not detect the existing secret for authentication with the Git repository. With this update, this command successfully detects the secret.
- With this update, you can use **--v1beta1** flag in the **tkn pac resolve** command. Use this flag if you want to generate the pipeline run with the **v1beta1** API version schema.
- Before this update, the **tkn pr logs** command failed to display the logs for a pipeline run if this pipeline run referenced a resolver. With this update, the command displays the logs.
- With this update, the SHA digest of the **git-init** image corresponds to version 1.12.1, which is the current released version of the image
- With this update, the HTTP/2.0 protocol is not supported for webhooks. All webhook calls to Red Hat OpenShift Pipelines must use the HTTP/1.1 protocol.

1.7.8. Known issues

- If you use the Bundles resolver to define a pipeline run and then use the **tkn pac resolve --v1beta1** command for this pipeline run, the command generates incorrect YAML output. The **kind** parameter for the bundle is set to **Task** in the YAML output. As a workaround, you can set the correct value in the YAML data manually. Alternatively, you can use the **opc pac resolve --v1beta1** command or use the version of the **tkn** tool included with OpenShift Pipelines version 1.12.0 or later.

1.7.9. Release notes for Red Hat OpenShift Pipelines General Availability 1.11.3

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.11.3 is available on OpenShift Container Platform 4.11 in addition to 4.12 and later versions.

1.7.9.1. Fixed issues

- Before this update, if the final task of a pipeline has failed or was skipped, OpenShift Pipelines reported validation errors. With this update, a pipeline can succeed even if its final task fails or is skipped.

1.8. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.10

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.10 is available on OpenShift Container Platform 4.11, 4.12, and 4.13.

1.8.1. New features

In addition to fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.10.

1.8.1.1. Pipelines

- With this update, you can specify environment variables in a **PipelineRun** or **TaskRun** pod template to override or append the variables that are configured in a task or step. Also, you can specify environment variables in a default pod template to use those variables globally for all **PipelineRuns** and **TaskRuns**. This update also adds a new default configuration named **forbidden-envs** to filter environment variables while propagating from pod templates.

- With this update, custom tasks in pipelines are enabled by default.

  NOTE

  To disable this update, set the **enable-custom-tasks** flag to **false** in the **feature-flags** config custom resource.

- This update supports the **v1beta1.CustomRun** API version for custom tasks.

- This update adds support for the **PipelineRun** reconciler to create a custom run. For example, custom **TaskRuns** created from **PipelineRuns** can now use the **v1beta1.CustomRun** API version instead of **v1alpha1.Run**, if the **custom-task-version** feature flag is set to **v1beta1**, instead of the default value **v1alpha1**.

  NOTE

  You need to update the custom task controller to listen for the ***v1beta1.CustomRun** API version instead of ***v1alpha1.Run** in order to respond to **v1beta1.CustomRun** requests.

- This update adds a new **retries** field to the **v1beta1.TaskRun** and **v1.TaskRun** specifications.

1.8.1.2. Triggers

- With this update, triggers support the creation of **Pipelines**, **Tasks**, **PipelineRuns**, and **TaskRuns** objects of the **v1** API version along with **CustomRun** objects of the **v1beta1** API version.
- With this update, GitHub Interceptor blocks a pull request trigger from being executed unless invoked by an owner or with a configurable comment by an owner.

  NOTE

  To enable or disable this update, set the value of the **githubOwners** parameter to **true** or **false** in the GitHub Interceptor configuration file.
- With this update, GitHub Interceptor has the ability to add a comma delimited list of all files that have changed for the push and pull request events. The list of changed files is added to the **changed_files** property of the event payload in the top-level extensions field.
- This update changes the **MinVersion** of TLS to **tls.VersionTLS12** so that triggers run on OpenShift Container Platform when the Federal Information Processing Standards (FIPS) mode is enabled.

1.8.1.3. CLI

- This update adds support to pass a Container Storage Interface (CSI) file as a workspace at the time of starting a **Task**, **ClusterTask** or **Pipeline**.
- This update adds **v1** API support to all CLI commands associated with task, pipeline, pipeline run, and task run resources. Tekton CLI works with both **v1beta1** and **v1** APIs for these resources.
- This update adds support for an object type parameter in the **start** and **describe** commands.

1.8.1.4. Operator

- This update adds a **default-forbidden-env** parameter in optional pipeline properties. The parameter includes forbidden environment variables that should not be propagated if provided through pod templates.
- This update adds support for custom logos in Tekton Hub UI. To add a custom logo, set the value of the **customLogo** parameter to base64 encoded URI of logo in the Tekton Hub CR.
- This update increments the version number of the git-clone task to 0.9.

1.8.1.5. Tekton Chains

IMPORTANT

Tekton Chains is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- This update adds annotations and labels to the **PipelineRun** and **TaskRun** attestations.
- This update adds a new format named **slsa/v1**, which generates the same provenance as the one generated when requesting in the **in-toto** format.
- With this update, Sigstore features are moved out from the experimental features.
- With this update, the **predicate.materials** function includes image URI and digest information from all steps and sidecars for a **TaskRun** object.

1.8.1.6. Tekton Hub

IMPORTANT

Tekton Hub is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- This update supports installing, upgrading, or downgrading Tekton resources of the **v1** API version on the cluster.
- This update supports adding a custom logo in place of the Tekton Hub logo in UI.
- This update extends the **tkn hub install** command functionality by adding a **--type artifact** flag, which fetches resources from the Artifact Hub and installs them on your cluster.
- This update adds support tier, catalog, and org information as labels to the resources being installed from Artifact Hub to your cluster.

1.8.1.7. Pipelines as Code

- This update enhances incoming webhook support. For a GitHub application installed on the OpenShift Container Platform cluster, you do not need to provide the **git_provider** specification for an incoming webhook. Instead, Pipelines as Code detects the secret and use it for the incoming webhook.
- With this update, you can use the same token to fetch remote tasks from the same host on GitHub with a non-default branch.
- With this update, Pipelines as Code supports Tekton **v1** templates. You can have **v1** and **v1beta1** templates, which Pipelines as Code reads for PR generation. The PR is created as **v1** on cluster.
- Before this update, OpenShift console UI would use a hardcoded pipeline run template as a fallback template when a runtime template was not found in the OpenShift namespace. This update in the **pipelines-as-code** config map provides a new default pipeline run template named, **pipelines-as-code-template-default** for the console to use.
- With this update, Pipelines as Code supports Tekton Pipelines 0.44.0 minimal status.

- With this update, Pipelines as Code supports Tekton **v1** API, which means Pipelines as Code is now compatible with Tekton v0.44 and later.
- With this update, you can configure custom console dashboards in addition to configuring a console for OpenShift and Tekton dashboards for k8s.
- With this update, Pipelines as Code detects the installation of a GitHub application initiated using the **tkn pac create repo** command and does not require a GitHub webhook if it was installed globally.
- Before this update, if there was an error on a **PipelineRun** execution and not on the tasks attached to **PipelineRun**, Pipelines as Code would not report the failure properly. With this update, Pipelines as Code reports the error properly on the GitHub checks when a **PipelineRun** could not be created.
- With this update, Pipelines as Code includes a **target_namespace** variable, which expands to the currently running namespace where the **PipelineRun** is executed.
- With this update, Pipelines as Code lets you bypass GitHub enterprise questions in the CLI bootstrap GitHub application.
- With this update, Pipelines as Code does not report errors when the repository CR was not found.
- With this update, Pipelines as Code reports an error if multiple pipeline runs with the same name were found.

1.8.2. Breaking changes

- With this update, the prior version of the **tkn** command is not compatible with Red Hat OpenShift Pipelines 1.10.
- This update removes support for **Cluster** and **CloudEvent** pipeline resources from Tekton CLI. You cannot create pipeline resources by using the **tkn pipelineresource create** command. Also, pipeline resources are no longer supported in the **start** command of a task, cluster task, or pipeline.
- This update removes **tekton** as a provenance format from Tekton Chains.

1.8.3. Deprecated and removed features

- In Red Hat OpenShift Pipelines 1.10, the **ClusterTask** commands are now deprecated and are planned to be removed in a future release. The **tkn task create** command is also deprecated with this update.
- In Red Hat OpenShift Pipelines 1.10, the flags **-i** and **-o** that were used with the **tkn task start** command are now deprecated because the **v1** API does not support pipeline resources.
- In Red Hat OpenShift Pipelines 1.10, the flag **-r** that was used with the **tkn pipeline start** command is deprecated because the **v1** API does not support pipeline resources.
- The Red Hat OpenShift Pipelines 1.10 update sets the **openshiftDefaultEmbeddedStatus** parameter to **both** with **full** and **minimal** embedded status. The flag to change the default embedded status is also deprecated and will be removed. In addition, the pipeline default embedded status will be changed to **minimal** in a future release.

1.8.4. Known issues

- This update includes the following backward incompatible changes:
  - Removal of the **PipelineResources** cluster
  - Removal of the **PipelineResources** cloud event
- If the pipelines metrics feature does not work after a cluster upgrade, run the following command as a workaround:

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

- With this update, usage of external databases, such as the Crunchy PostgreSQL is not supported on IBM Power, IBM zSystems, and IBM® LinuxONE. Instead, use the default Tekton Hub database.

1.8.5. Fixed issues

- Before this update, the **opc pac** command generated a runtime error instead of showing any help. This update fixes the **opc pac** command to show the help message.
- Before this update, running the **tkn pac create repo** command needed the webhook details for creating a repository. With this update, the **tkn-pac create repo** command does not configure a webhook when your GitHub application is installed.
- Before this update, Pipelines as Code would not report a pipeline run creation error when Tekton Pipelines had issues creating the **PipelineRun** resource. For example, a non-existing task in a pipeline run would show no status. With this update, Pipelines as Code shows the proper error message coming from Tekton Pipelines along with the task that is missing.
- This update fixes UI page redirection after a successful authentication. Now, you are redirected to the same page where you had attempted to log in to Tekton Hub.
- This update fixes the **list** command with these flags, **--all-namespaces** and **--output=yaml**, for a cluster task, an individual task, and a pipeline.
- This update removes the forward slash in the end of the **repo.spec.url** URL so that it matches the URL coming from GitHub.
- Before this update, the **marshalJSON** function would not marshal a list of objects. With this update, the **marshalJSON** function marshals the list of objects.
- With this update, Pipelines as Code lets you bypass GitHub enterprise questions in the CLI bootstrap GitHub application.
- This update fixes the GitHub collaborator check when your repository has more than 100 users.
- With this update, the **sign** and **verify** commands for a task or pipeline now work without a kubernetes configuration file.
- With this update, Tekton Operator cleans leftover pruner cron jobs if pruner has been skipped on a namespace.

- Before this update, the API **ConfigMap** object would not be updated with a user configured value for a catalog refresh interval. This update fixes the **CATALOG_REFRESH_INTERVAL** API in the Tekon Hub CR.
- This update fixes reconciling of **PipelineRunStatus** when changing the **EmbeddedStatus** feature flag. This update resets the following parameters:
  - The **status.runs** and **status.taskruns** parameters to **nil** with **minimal EmbeddedStatus**
  - The **status.childReferences** parameter to **nil** with **full EmbeddedStatus**
- This update adds a conversion configuration to the **ResolutionRequest** CRD. This update properly configures conversion from the **v1alpha1.ResolutionRequest** request to the **v1beta1.ResolutionRequest** request.
- This update checks for duplicate workspaces associated with a pipeline task.
- This update fixes the default value for enabling resolvers in the code.
- This update fixes **TaskRef** and **PipelineRef** names conversion by using a resolver.

1.8.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.10.1

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.10.1 is available on OpenShift Container Platform 4.11, 4.12, and 4.13.

1.8.6.1. Fixed issues for Pipelines as Code

- Before this update, if the source branch information coming from payload included **refs/heads/** but the user-configured target branch only included the branch name, **main**, in a CEL expression, the push request would fail. With this update, Pipelines as Code passes the push request and triggers a pipeline if either the base branch or target branch has **refs/heads/** in the payload.
- Before this update, when a **PipelineRun** object could not be created, the error received from the Tekton controller was not reported to the user. With this update, Pipelines as Code reports the error messages to the GitHub interface so that users can troubleshoot the errors. Pipelines as Code also reports the errors that occurred during pipeline execution.
- With this update, Pipelines as Code does not echo a secret to the GitHub checks interface when it failed to create the secret on the OpenShift Container Platform cluster because of an infrastructure issue.
- This update removes the deprecated APIs that are no longer in use from Red Hat OpenShift Pipelines.

1.8.7. Release notes for Red Hat OpenShift Pipelines General Availability 1.10.2

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.10.2 is available on OpenShift Container Platform 4.11, 4.12, and 4.13.

1.8.7.1. Fixed issues

Before this update, an issue in the Tekton Operator prevented the user from setting the value of the **enable-api-fields** flag to **beta**. This update fixes the issue. Now, you can set the value of the **enable-api-fields** flag to **beta** in the **TektonConfig** CR.

1.8.8. Release notes for Red Hat OpenShift Pipelines General Availability 1.10.3

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.10.3 is available on OpenShift Container Platform 4.11, 4.12, and 4.13.

1.8.8.1. Fixed issues

Before this update, the Tekton Operator did not expose the performance configuration fields for any customizations. With this update, as a cluster administrator, you can customize the following performance configuration fields in the **TektonConfig** CR based on your needs:

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

1.8.9. Release notes for Red Hat OpenShift Pipelines General Availability 1.10.4

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.10.4 is available on OpenShift Container Platform 4.11, 4.12, and 4.13.

1.8.9.1. Fixed issues

- This update fixes the bundle resolver conversion issue for the **PipelineRef** field in a pipeline run. Now, the conversion feature sets the value of the **kind** field to **Pipeline** after conversion.
- Before this update, the **pipelinerun.timeouts** field was reset to the **timeouts.pipeline** value, ignoring the **timeouts.tasks** and **timeouts.finally** values. This update fixes the issue and sets the correct default timeout value for a **PipelineRun** resource.
- Before this update, the controller logs contained unnecessary data. This update fixes the issue.

1.8.10. Release notes for Red Hat OpenShift Pipelines General Availability 1.10.5

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.10.5 is available on OpenShift Container Platform 4.10 in addition to 4.11, 4.12, and 4.13.

IMPORTANT

Red Hat OpenShift Pipelines 1.10.5 is only available in the **pipelines-1.10** channel on OpenShift Container Platform 4.10, 4.11, 4.12, and 4.13. It is not available in the **latest** channel for any OpenShift Container Platform version.

1.8.10.1. Fixed issues

- Before this update, huge pipeline runs were not getting listed or deleted using the **oc** and **tkn** commands. This update mitigates this issue by compressing the huge annotations that were causing this problem. Remember that if the pipeline runs are still too huge after compression, then the same error still recurs.

- Before this update, only the pod template specified in the **pipelineRun.spec.taskRunSpecs[].podTemplate** object would be considered for a pipeline run. With this update, the pod template specified in the **pipelineRun.spec.podTemplate** object is also considered and merged with the template specified in the **pipelineRun.spec.taskRunSpecs[].podTemplate** object.

1.8.11. Release notes for Red Hat OpenShift Pipelines General Availability 1.10.6

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.10.6 is available on OpenShift Container Platform 4.10, 4.11, 4.12, and 4.13.

This update includes an updated version of the **tkn** command line tool. You can download the updated version of this tool at the following locations:

- Linux (x86_64, amd64)
- Linux on IBM zSystems and IBM® LinuxONE (s390x)
- Linux on IBM Power (ppc64le)
- Linux on ARM (aarch64, arm64)
- Windows
- macOS
- macOS on ARM

If you installed the **tkn** command line tool using RPM on Red Hat Enterprise Linux (RHEL), use the **yum update** command to install the updated version.

1.8.11.1. Known issues

- If you enter the **tkn task start** or **tkn clustertask start** command, the **tkn** command line utility displays an error message. As a workaround, to start tasks or cluster tasks using the command line, use the version of the **tkn** utility shipped with OpenShift Pipelines 1.11 or a later version.

1.8.11.2. Fixed issues

- With this update, the S2I cluster task uses a General Availability container image.
- With this update, the HTTP/2.0 protocol is not supported for webhooks. All webhook calls to Red Hat OpenShift Pipelines must use the HTTP/1.1 protocol.

1.9. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.9

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.9 is available on OpenShift Container Platform 4.11, 4.12, and 4.13.

1.9.1. New features

In addition to the fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.9.

1.9.1.1. Pipelines

- With this update, you can specify pipeline parameters and results in arrays and object dictionary forms.
- This update provides support for Container Storage Interface (CSI) and projected volumes for your workspace.
- With this update, you can specify the **stdoutConfig** and **stderrConfig** parameters when defining pipeline steps. Defining these parameters helps to capture standard output and standard error, associated with steps, to local files.
- With this update, you can add variables in the **steps[].onError** event handler, for example, **$(params.CONTINUE)**.
- With this update, you can use the output from the **finally** task in the **PipelineResults** definition. For example, **$(finally.<pipelinetask-name>.result.<result-name>)**, where **<pipelinetask-name>** denotes the pipeline task name and **<result-name>** denotes the result name.
- This update supports task-level resource requirements for a task run.
- With this update, you do not need to recreate parameters that are shared, based on their names, between a pipeline and the defined tasks. This update is part of a developer preview feature.
- This update adds support for remote resolution, such as built-in git, cluster, bundle, and hub resolvers.

1.9.1.2. Triggers

- This update adds the **Interceptor** CRD to define **NamespacedInterceptor**. You can use **NamespacedInterceptor** in the **kind** section of interceptors reference in triggers or in the **EventListener** specification.
- This update enables **CloudEvents**.
- With this update, you can configure the webhook port number when defining a trigger.
- This update supports using trigger **eventID** as input to **TriggerBinding**.
- This update supports validation and rotation of certificates for the **ClusterInterceptor** server.
    - Triggers perform certificate validation for core interceptors and rotate a new certificate to **ClusterInterceptor** when its certificate expires.

1.9.1.3. CLI

- This update supports showing annotations in the **describe** command.
- This update supports showing pipeline, tasks, and timeout in the **pr describe** command.
- This update adds flags to provide pipeline, tasks, and timeout in the **pipeline start** command.
- This update supports showing the presence of workspace, optional or mandatory, in the **describe** command of a task and pipeline.
- This update adds the **timestamps** flag to show logs with a timestamp.

- This update adds a new flag **--ignore-running-pipelinerun**, which ignores the deletion of **TaskRun** associated with **PipelineRun**.
- This update adds support for experimental commands. This update also adds experimental subcommands, **sign** and **verify** to the **tkn** CLI tool.
- This update makes the Z shell (Zsh) completion feature usable without generating any files.
- This update introduces a new CLI tool called **opc**. It is anticipated that an upcoming release will replace the **tkn** CLI tool with **opc**.

> IMPORTANT
>
> - The new CLI tool **opc** is a Technology Preview feature.
> - **opc** will be a replacement for **tkn** with additional Red Hat OpenShift Pipelines specific features, which do not necessarily fit in **tkn**.

1.9.1.4. Operator

- With this update, Pipelines as Code is installed by default. You can disable Pipelines as Code by using the **-p** flag:

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

- With this update, you can also modify Pipelines as Code configurations in the **TektonConfig** CRD.
- With this update, if you disable the developer perspective, the Operator does not install developer console related custom resources.
- This update includes **ClusterTriggerBinding** support for Bitbucket Server and Bitbucket Cloud and helps you to reuse a **TriggerBinding** across your entire cluster.

1.9.1.5. Resolvers

> IMPORTANT
>
> Resolvers is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.
>
> For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- With this update, you can configure pipeline resolvers in the **TektonConfig** CRD. You can enable or disable these pipeline resolvers: **enable-bundles-resolver**, **enable-cluster-resolver**, **enable-git-resolver**, and **enable-hub-resolver**.

```
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
...
```

You can also provide resolver specific configurations in **TektonConfig**. For example, you can define the following fields in the **map[string]string** format to set configurations for individual resolvers:

```
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
...
```

1.9.1.6. Tekton Chains

IMPORTANT

Tekton Chains is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- Before this update, only Open Container Initiative (OCI) images were supported as outputs of **TaskRun** in the in-toto provenance agent. This update adds in-toto provenance metadata as outputs with these suffixes, **ARTIFACT_URI** and **ARTIFACT_DIGEST**.

- Before this update, only **TaskRun** attestations were supported. This update adds support for **PipelineRun** attestations as well.

- This update adds support for Tekton Chains to get the **imgPullSecret** parameter from the pod template. This update helps you to configure repository authentication based on each pipeline run or task run without modifying the service account.

1.9.1.7. Tekton Hub

IMPORTANT

Tekton Hub is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- With this update, as an administrator, you can use an external database, such as Crunchy PostgreSQL with Tekton Hub, instead of using the default Tekton Hub database. This update helps you to perform the following actions:
  - Specify the coordinates of an external database to be used with Tekton Hub
  - Disable the default Tekton Hub database deployed by the Operator
- This update removes the dependency of **config.yaml** from external Git repositories and moves the complete configuration data into the API **ConfigMap**. This update helps an administrator to perform the following actions:
  - Add the configuration data, such as categories, catalogs, scopes, and defaultScopes in the Tekton Hub custom resource.
  - Modify Tekton Hub configuration data on the cluster. All modifications are preserved upon Operator upgrades.
  - Update the list of catalogs for Tekton Hub
  - Change the categories for Tekton Hub

  NOTE

  If you do not add any configuration data, you can use the default data in the API **ConfigMap** for Tekton Hub configurations.

1.9.1.8. Pipelines as Code

- This update adds support for concurrency limit in the **Repository** CRD to define the maximum number of **PipelineRuns** running for a repository at a time. The **PipelineRuns** from a pull request or a push event are queued in alphabetical order.
- This update adds a new command **tkn pac logs** for showing the logs of the latest pipeline run for a repository.
- This update supports advanced event matching on file path for push and pull requests to GitHub and GitLab. For example, you can use the Common Expression Language (CEL) to run a pipeline only if a path has changed for any markdown file in the **docs** directory.

```
...
annotations:
```

```
pipelinesascode.tekton.dev/on-cel-expression: |
 event == "pull_request" && "docs/*.md".pathChanged()
```

- With this update, you can reference a remote pipeline in the **pipelineRef:** object using annotations.

- With this update, you can auto-configure new GitHub repositories with Pipelines as Code, which sets up a namespace and creates a **Repository** CRD for your GitHub repository.

- With this update, Pipelines as Code generates metrics for **PipelineRuns** with provider information.

- This update provides the following enhancements for the **tkn-pac** plugin:

  - Detects running pipelines correctly
  - Fixes showing duration when there is no failure completion time
  - Shows an error snippet and highlights the error regular expression pattern in the **tkn-pac describe** command
  - Adds the **use-real-time** switch to the **tkn-pac ls** and **tkn-pac describe** commands
  - Imports the **tkn-pac** logs documentation
  - Shows **pipelineruntimeout** as a failure in the **tkn-pac ls** and **tkn-pac describe** commands.
  - Show a specific pipeline run failure with the **--target-pipelinerun** option.

- With this update, you can view the errors for your pipeline run in the form of a version control system (VCS) comment or a small snippet in the GitHub checks.

- With this update, Pipelines as Code optionally can detect errors inside the tasks if they are of a simple format and add those tasks as annotations in GitHub. This update is part of a developer preview feature.

- This update adds the following new commands:

  - **tkn-pac webhook add**: Adds a webhook to project repository settings and updates the **webhook.secret** key in the existing **k8s Secret** object without updating the repository.
  - **tkn-pac webhook update-token**: Updates provider token for an existing **k8s Secret** object without updating the repository.

- This update enhances functionality of the **tkn-pac create repo** command, which creates and configures webhooks for GitHub, GitLab, and BitbucketCloud along with creating repositories.

- With this update, the **tkn-pac describe** command shows the latest fifty events in a sorted order.

- This update adds the **--last** option to the **tkn-pac logs** command.

- With this update, the **tkn-pac resolve** command prompts for a token on detecting a **git_auth_secret** in the file template.

- With this update, Pipelines as Code hides secrets from log snippets to avoid exposing secrets in the GitHub interface.

- With this update, the secrets automatically generated for **git_auth_secret** are an owner reference with **PipelineRun**. The secrets get cleaned with the **PipelineRun**, not after the pipeline run execution.
- This update adds support to cancel a pipeline run with the /**cancel** comment.
- Before this update, the GitHub apps token scoping was not defined and tokens would be used on every repository installation. With this update, you can scope the GitHub apps token to the target repository using the following parameters:
  - **secret-github-app-token-scoped**: Scopes the app token to the target repository, not to every repository the app installation has access to.
  - **secret-github-app-scope-extra-repos**: Customizes the scoping of the app token with an additional owner or repository.
- With this update, you can use Pipelines as Code with your own Git repositories that are hosted on GitLab.
- With this update, you can access pipeline execution details in the form of kubernetes events in your namespace. These details help you to troubleshoot pipeline errors without needing access to admin namespaces.
- This update supports authentication of URLs in the Pipelines as Code resolver with the Git provider.
- With this update, you can set the name of the hub catalog by using a setting in the **pipelines-as-code** config map.
- With this update, you can set the maximum and default limits for the **max-keep-run** parameter.
- This update adds documents on how to inject custom Secure Sockets Layer (SSL) certificates in Pipelines as Code to let you connect to provider instance with custom certificates.
- With this update, the **PipelineRun** resource definition has the log URL included as an annotation. For example, the **tkn-pac describe** command shows the log link when describing a **PipelineRun**.
- With this update, **tkn-pac** logs show repository name, instead of **PipelineRun** name.

1.9.2. Breaking changes

- With this update, the **Conditions** custom resource definition (CRD) type has been removed. As an alternative, use the **WhenExpressions** instead.
- With this update, support for **tekton.dev/v1alpha1** API pipeline resources, such as Pipeline, PipelineRun, Task, Clustertask, and TaskRun has been removed.
- With this update, the **tkn-pac setup** command has been removed. Instead, use the **tkn-pac webhook add** command to re-add a webhook to an existing Git repository. And use the **tkn-pac webhook update-token** command to update the personal provider access token for an existing Secret object in the Git repository.
- With this update, a namespace that runs a pipeline with default settings does not apply the **pod-security.kubernetes.io/enforce:privileged** label to a workload.

1.9.3. Deprecated and removed features

- In the Red Hat OpenShift Pipelines 1.9.0 release, **ClusterTasks** are deprecated and planned to be removed in a future release. As an alternative, you can use **Cluster Resolver**.

- In the Red Hat OpenShift Pipelines 1.9.0 release, the use of the **triggers** and the **namespaceSelector** fields in a single **EventListener** specification is deprecated and planned to be removed in a future release. You can use these fields in different **EventListener** specifications successfully.

- In the Red Hat OpenShift Pipelines 1.9.0 release, the **tkn pipelinerun describe** command does not display timeouts for the **PipelineRun** resource.

- In the Red Hat OpenShift Pipelines 1.9.0 release, the PipelineResource` custom resource (CR) is deprecated. The **PipelineResource** CR was a Tech Preview feature and part of the **tekton.dev/v1alpha1** API.

- In the Red Hat OpenShift Pipelines 1.9.0 release, custom image parameters from cluster tasks are deprecated. As an alternative, you can copy a cluster task and use your custom image in it.

1.9.4. Known issues

- The **chains-secret** and **chains-config** config maps are removed after you uninstall the Red Hat OpenShift Pipelines Operator. As they contain user data, they should be preserved and not deleted.

- When running the **tkn pac** set of commands on Windows, you may receive the following error message: **Command finished with error: not supported by Windows.**
  Workaround: Set the **NO_COLOR** environment variable to **true**.

- Running the **tkn pac resolve -f <filename> | oc create -f** command may not provide expected results, if the **tkn pac resolve** command uses a templated parameter value to function.
  Workaround: To mitigate this issue, save the output of **tkn pac resolve** in a temporary file by running the **tkn pac resolve -f <filename> -o tempfile.yaml** command and then run the **oc create -f tempfile.yaml** command. For example, **tkn pac resolve -f <filename> -o /tmp/pull-request-resolved.yaml && oc create -f /tmp/pull-request-resolved.yaml**.

1.9.5. Fixed issues

- Before this update, after replacing an empty array, the original array returned an empty string rendering the paramaters inside it invalid. With this update, this issue is resolved and the original array returns as empty.

- Before this update, if duplicate secrets were present in a service account for a pipelines run, it resulted in failure in task pod creation. With this update, this issue is resolved and the task pod is created successfully even if duplicate secrets are present in a service account.

- Before this update, by looking at the TaskRun's **spec.StatusMessage** field, users could not distinguish whether the **TaskRun** had been cancelled by the user or by a **PipelineRun** that was part of it. With this update, this issue is resolved and users can distinguish the status of the **TaskRun** by looking at the TaskRun's **spec.StatusMessage** field.

- Before this update, webhook validation was removed on deletion of old versions of invalid objects. With this update, this issue is resolved.

- Before this update, if you set the **timeouts.pipeline** parameter to **0**, you could not set the **timeouts.tasks** parameter or the **timeouts.finally** parameters. This update resolves the issue. Now, when you set the **timeouts.pipeline** parameter value, you can set the value of either the`timeouts.tasks` parameter or the **timeouts.finally** parameter. For example:

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

- Before this update, a race condition could occur if another tool updated labels or annotations on a PipelineRun or TaskRun. With this update, this issue is resolved and you can merge labels or annotations.

- Before this update, log keys did not have the same keys as in pipelines controllers. With this update, this issue has been resolved and the log keys have been updated to match the log stream of pipeline controllers. The keys in logs have been changed from "ts" to "timestamp", from "level" to "severity", and from "message" to "msg".

- Before this update, if a PipelineRun was deleted with an unknown status, an error message was not generated. With this update, this issue is resolved and an error message is generated.

- Before this update, to access bundle commands like **list** and **push**, it was required to use the **kubeconfig** file . With this update, this issue has been resolved and the **kubeconfig** file is not required to access bundle commands.

- Before this update, if the parent PipelineRun was running while deleting TaskRuns, then TaskRuns would be deleted. With this update, this issue is resolved and TaskRuns are not getting deleted if the parent PipelineRun is running.

- Before this update, if the user attempted to build a bundle with more objects than the pipeline controller permitted, the Tekton CLI did not display an error message. With this update, this issue is resolved and the Tekton CLI displays an error message if the user attempts to build a bundle with more objects than the limit permitted in the pipeline controller.

- Before this update, if namespaces were removed from the cluster, then the operator did not remove namespaces from the **ClusterInterceptor ClusterRoleBinding** subjects. With this update, this issue has been resolved, and the operator removes the namespaces from the **ClusterInterceptor ClusterRoleBinding** subjects.

- Before this update, the default installation of the Red Hat OpenShift Pipelines Operator resulted in the **pipelines-scc-rolebinding security context constraint** (SCC) role binding resource remaining in the cluster. With this update, the default installation of the Red Hat OpenShift Pipelines Operator results in the **pipelines-scc-rolebinding security context constraint** (SCC) role binding resource resource being removed from the cluster.

- Before this update, Pipelines as Code did not get updated values from the Pipelines as Code **ConfigMap** object. With this update, this issue is fixed and the Pipelines as Code **ConfigMap** object looks for any new changes.

- Before this update, Pipelines as Code controller did not wait for the **tekton.dev/pipeline** label to be updated and added the **checkrun id** label, which would cause race conditions. With this update, the Pipelines as Code controller waits for the **tekton.dev/pipeline** label to be updated and then adds the **checkrun id** label, which helps to avoid race conditions.

- Before this update, the **tkn-pac create repo** command did not override a **PipelineRun** if it already existed in the git repository. With this update, **tkn-pac create** command is fixed to override a **PipelineRun** if it exists in the git repository and this resolves the issue successfully.

- Before this update, the **tkn pac describe** command did not display reasons for every message. With this update, this issue is fixed and the **tkn pac describe** command displays reasons for every message.

- Before this update, a pull request failed if the user in the annotation provided values by using a regex form, for example, **refs/head/rel-***. The pull request failed because it was missing **refs/heads** in its base branch. With this update, the prefix is added and checked that it matches. This resolves the issue and the pull request is successful.

1.9.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.9.1

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.9.1 is available on OpenShift Container Platform 4.11, 4.12, and 4.13.

1.9.7. Fixed issues

- Before this update, the **tkn pac repo list** command did not run on Microsoft Windows. This update fixes the issue, and now you can run the **tkn pac repo list** command on Microsoft Windows.

- Before this update, Pipelines as Code watcher did not receive all the configuration change events. With this update, the Pipelines as Code watcher is updated, and now the Pipelines as Code watcher does not miss the configuration change events.

- Before this update, the pods created by Pipelines as Code, such as **TaskRuns** or **PipelineRuns** could not access custom certificates exposed by the user in the cluster. This update fixes the issue, and you can now access custom certificates from the **TaskRuns** or **PipelineRuns** pods in the cluster.

- Before this update, on a cluster enabled with FIPS, the **tekton-triggers-core-interceptors** core interceptor used in the **Trigger** resource did not function after the Pipelines Operator was upgraded to version 1.9. This update resolves the issue. Now, OpenShift uses MInTLS 1.2 for all its components. As a result, the **tekton-triggers-core-interceptors** core interceptor updates to TLS version 1.2and its functionality runs accurately.

- Before this update, when using a pipeline run with an internal OpenShift image registry, the URL to the image had to be hardcoded in the pipeline run definition. For example:

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

  When using a pipeline run in the context of Pipelines as Code, such hardcoded values prevented the pipeline run definitions from being used in different clusters and namespaces.

  With this update, you can use the dynamic template variables instead of hardcoding the values for namespaces and pipeline run names to generalize pipeline run definitions. For example:

  ```
  ...
  ```

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

- Before this update, Pipelines as Code used the same GitHub token to fetch a remote task available in the same host only on the default GitHub branch. This update resolves the issue. Now Pipelines as Code uses the same GitHub token to fetch a remote task from any GitHub branch.

1.9.8. Known issues

- The value for **CATALOG_REFRESH_INTERVAL**, a field in the Hub API **ConfigMap** object used in the Tekton Hub CR, is not getting updated with a custom value provided by the user.
Workaround: None. You can track the issue SRVKP-2854.

1.9.9. Breaking changes

- With this update, an OLM misconfiguration issue has been introduced, which prevents the upgrade of the OpenShift Container Platform. This issue will be fixed in a future release.

1.9.10. Release notes for Red Hat OpenShift Pipelines General Availability 1.9.2

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.9.2 is available on OpenShift Container Platform 4.11, 4.12, and 4.13.

1.9.11. Fixed issues

- Before this update, an OLM misconfiguration issue had been introduced in the previous version of the release, which prevented the upgrade of OpenShift Container Platform. With this update, this misconfiguration issue has been fixed.

1.9.12. Release notes for Red Hat OpenShift Pipelines General Availability 1.9.3

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.9.3 is available on OpenShift Container Platform 4.10 in addition to 4.11, 4.12, and 4.13.

1.9.13. Fixed issues

- This update fixes the performance issues for huge pipelines. Now, the CPU usage is reduced by 61% and the memory usage is reduced by 44%.

- Before this update, a pipeline run would fail if a task did not run because of its **when** expression. This update fixes the issue by preventing the validation of a skipped task result in pipeline results. Now, the pipeline result is not emitted and the pipeline run does not fail because of a missing result.

- This update fixes the **pipelineref.bundle** conversion to the bundle resolver for the **v1beta1** API. Now, the conversion feature sets the value of the **kind** field to **Pipeline** after conversion.

- Before this update, an issue in the OpenShift Pipelines Operator prevented the user from setting the value of the **spec.pipeline.enable-api-fields** field to **beta**. This update fixes the issue. Now, you can set the value to **beta** along with **alpha** and **stable** in the **TektonConfig** custom resource.

- Before this update, when Pipelines as Code could not create a secret due to a cluster error, it would show the temporary token on the GitHub check run, which is public. This update fixes the issue. Now, the token is no longer displayed on the GitHub checks interface when the creation of the secret fails.

1.9.14. Known issues

- There is currently a known issue with the **stop** option for pipeline runs in the OpenShift Container Platform web console. The **stop** option in the **Actions** drop-down list is not working as expected and does not cancel the pipeline run.

- There is currently a known issue with upgrading to OpenShift Pipelines version 1.9.x due to a failing custom resource definition conversion.
  Workaround: Before upgrading to OpenShift Pipelines version 1.9.x, perform the step mentioned in the solution on the Red Hat Customer Portal.

1.10. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.8

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.8 is available on OpenShift Container Platform 4.10, 4.11, and 4.12.

1.10.1. New features

In addition to the fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.8.

1.10.1.1. Pipelines

- With this update, you can run Red Hat OpenShift Pipelines GA 1.8 and later on an OpenShift Container Platform cluster that is running on ARM hardware. This includes support for **ClusterTask** resources and the **tkn** CLI tool.

> IMPORTANT
>
> Running Red Hat OpenShift Pipelines on ARM hardware is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.
>
> For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- This update implements **Step** and **Sidecar** overrides for **TaskRun** resources.

- This update adds minimal **TaskRun** and **Run** statuses within **PipelineRun** statuses.
  To enable this feature, in the **TektonConfig** custom resource definition, in the **pipeline** section, you must set the **enable-api-fields** field to **alpha**.

- With this update, the graceful termination of pipeline runs feature is promoted from an alpha feature to a stable feature. As a result, the previously deprecated **PipelineRunCancelled** status remains deprecated and is planned to be removed in a future release.

Because this feature is available by default, you no longer need to set the **pipeline.enable-api-fields** field to **alpha** in the **TektonConfig** custom resource definition.

- With this update, you can specify the workspace for a pipeline task by using the name of the workspace. This change makes it easier to specify a shared workspace for a pair of **Pipeline** and **PipelineTask** resources. You can also continue to map workspaces explicitly.
  To enable this feature, in the **TektonConfig** custom resource definition, in the **pipeline** section, you must set the **enable-api-fields** field to **alpha**.

- With this update, parameters in embedded specifications are propagated without mutations.

- With this update, you can specify the required metadata of a **Task** resource referenced by a **PipelineRun** resource by using annotations and labels. This way, **Task** metadata that depends on the execution context is available during the pipeline run.

- This update adds support for object or dictionary types in **params** and **results** values. This change affects backward compatibility and sometimes breaks forward compatibility, such as using an earlier client with a later Red Hat OpenShift Pipelines version. This update changes the **ArrayOrStruct** structure, which affects projects that use the Go language API as a library.

- This update adds a **SkippingReason** value to the **SkippedTasks** field of the **PipelineRun** status fields so that users know why a given PipelineTask was skipped.

- This update supports an alpha feature in which you can use an **array** type for emitting results from a **Task** object. The result type is changed from **string** to **ArrayOrString**. For example, a task can specify a type to produce an array result:

  ```
  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
  ...
  ```

  Additionally, you can run a task script to populate the results with an array:

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

  To enable this feature, in the **TektonConfig** custom resource definition, in the **pipeline** section, you must set the **enable-api-fields** field to **alpha**.

  This feature is in progress and is part of TEP-0076.

1.10.1.2. Triggers

- This update transitions the **TriggerGroups** field in the **EventListener** specification from an alpha feature to a stable feature. Using this field, you can specify a set of interceptors before selecting and running a group of triggers.

Because this feature is available by default, you no longer need to set the **pipeline.enable-api-fields** field to **alpha** in the **TektonConfig** custom resource definition.

- With this update, the **Trigger** resource supports end-to-end secure connections by running the **ClusterInterceptor** server using HTTPS.

1.10.1.3. CLI

- With this update, you can use the **tkn taskrun export** command to export a live task run from a cluster to a YAML file, which you can use to import the task run to another cluster.

- With this update, you can add the **-o name** flag to the **tkn pipeline start** command to print the name of the pipeline run right after it starts.

- This update adds a list of available plugins to the output of the **tkn --help** command.

- With this update, while deleting a pipeline run or task run, you can use both the **--keep** and **--keep-since** flags together.

- With this update, you can use **Cancelled** as the value of the **spec.status** field rather than the deprecated **PipelineRunCancelled** value.

1.10.1.4. Operator

- With this update, as an administrator, you can configure your local Tekton Hub instance to use a custom database rather than the default database.

- With this update, as a cluster administrator, if you enable your local Tekton Hub instance, it periodically refreshes the database so that changes in the catalog appear in the Tekton Hub web console. You can adjust the period between refreshes.
  Previously, to add the tasks and pipelines in the catalog to the database, you performed that task manually or set up a cron job to do it for you.

- With this update, you can install and run a Tekton Hub instance with minimal configuration. This way, you can start working with your teams to decide which additional customizations they might want.

- This update adds **GIT_SSL_CAINFO** to the **git-clone** task so you can clone secured repositories.

1.10.1.5. Tekton Chains

> **IMPORTANT**
>
> Tekton Chains is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.
>
> For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- With this update, you can log in to a vault by using OIDC rather than a static token. This change means that Spire can generate the OIDC credential so that only trusted workloads are allowed to log in to the vault. Additionally, you can pass the vault address as a configuration value rather than inject it as an environment variable.

- The **chains-config** config map for Tekton Chains in the **openshift-pipelines** namespace is automatically reset to default after upgrading the Red Hat OpenShift Pipelines Operator because directly updating the config map is not supported when installed by using the Red Hat OpenShift Pipelines Operator. However, with this update, you can configure Tekton Chains by using the **TektonChain** custom resource. This feature enables your configuration to persist after upgrading, unlike the **chains-config** config map, which gets overwritten during upgrades.

1.10.1.6. Tekton Hub

IMPORTANT

Tekton Hub is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- With this update, if you install a fresh instance of Tekton Hub by using the Operator, the Tekton Hub login is disabled by default. To enable the login and rating features, you must create the Hub API secret while installing Tekton Hub.

  NOTE

  Because Tekton Hub login was enabled by default in Red Hat OpenShift Pipelines 1.7, if you upgrade the Operator, the login is enabled by default in Red Hat OpenShift Pipelines 1.8. To disable this login, see Disabling Tekton Hub login after upgrading from OpenShift Pipelines 1.7.x -→ 1.8.x

- With this update, as an administrator, you can configure your local Tekton Hub instance to use a custom PostgreSQL 13 database rather than the default database. To do so, create a **Secret** resource named **tekton-hub-db**. For example:

```
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>
```

- With this update, you no longer need to log in to the Tekton Hub web console to add resources from the catalog to the database. Now, these resources are automatically added when the Tekton Hub API starts running for the first time.

- This update automatically refreshes the catalog every 30 minutes by calling the catalog refresh API job. This interval is user-configurable.

1.10.1.7. Pipelines as Code

IMPORTANT

Pipelines as Code (PAC) is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

- With this update, as a developer, you get a notification from the **tkn-pac** CLI tool if you try to add a duplicate repository to a Pipelines as Code run. When you enter **tkn pac create repository**, each repository must have a unique URL. This notification also helps prevent hijacking exploits.

- With this update, as a developer, you can use the new **tkn-pac setup cli** command to add a Git repository to Pipelines as Code by using the webhook mechanism. This way, you can use Pipelines as Code even when using GitHub Apps is not feasible. This capability includes support for repositories on GitHub, GitLab, and BitBucket.

- With this update, Pipelines as Code supports GitLab integration with features such as the following:
  - ACL (Access Control List) on project or group
  - /**ok-to-test** support from allowed users
  - /**retest** support.

- With this update, you can perform advanced pipeline filtering with Common Expression Language (CEL). With CEL, you can match pipeline runs with different Git provider events by using annotations in the **PipelineRun** resource. For example:

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

- Previously, as a developer, you could have only one pipeline run in your **.tekton** directory for each Git event, such as a pull request. With this update, you can have multiple pipeline runs in your **.tekton** directory. The web console displays the status and reports of the runs. The pipeline runs operate in parallel and report back to the Git provider interface.

- With this update, you can test or retest a pipeline run by commenting **/test** or **/retest** on a pull request. You can also specify the pipeline run by name. For example, you can enter **/test <pipelinerun_name>** or **/retest <pipelinerun-name>**.

- With this update, you can delete a repository custom resource and its associated secrets by using the new **tkn-pac delete repository** command.

1.10.2. Breaking changes

- This update changes the default metrics level of **TaskRun** and **PipelineRun** resources to the following values:

```
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"
```

- With this update, if an annotation or label is present in both **Pipeline** and **PipelineRun** resources, the value in the **Run** type takes precedence. The same is true if an annotation or label is present in **Task** and **TaskRun** resources.

- In Red Hat OpenShift Pipelines 1.8, the previously deprecated **PipelineRun.Spec.ServiceAccountNames** field has been removed. Use the **PipelineRun.Spec.TaskRunSpecs** field instead.

- In Red Hat OpenShift Pipelines 1.8, the previously deprecated **TaskRun.Status.ResourceResults.ResourceRef** field has been removed. Use the **TaskRun.Status.ResourceResults.ResourceName** field instead.

- In Red Hat OpenShift Pipelines 1.8, the previously deprecated **Conditions** resource type has been removed. Remove the **Conditions** resource from **Pipeline** resource definitions that include it. Use **when** expressions in **PipelineRun** definitions instead.

- For Tekton Chains, the **tekton-provenance** format has been removed in this release. Use the **in-toto** format by setting **"artifacts.taskrun.format": "in-toto"** in the **TektonChain** custom resource instead.

- Red Hat OpenShift Pipelines 1.7.x shipped with Pipelines as Code 0.5.x. The current update ships with Pipelines as Code 0.10.x. This change creates a new route in the **openshift-pipelines** namespace for the new controller. You must update this route in GitHub Apps or webhooks that use Pipelines as Code. To fetch the route, use the following command:

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

- With this update, Pipelines as Code renames the default secret keys for the **Repository** custom resource definition (CRD). In your CRD, replace **token** with **provider.token**, and replace **secret** with **webhook.secret**.

- With this update, Pipelines as Code replaces a special template variable with one that supports multiple pipeline runs for private repositories. In your pipeline runs, replace **secret: pac-git-basic-auth-{{repo_owner}}-{{repo_name}}** with **secret: {{ git_auth_secret }}**.

- With this update, Pipelines as Code updates the following commands in the **tkn-pac** CLI tool:

  - Replace **tkn pac repository create** with **tkn pac create repository**.

  - Replace **tkn pac repository delete** with **tkn pac delete repository**.

  - Replace **tkn pac repository list** with **tkn pac list**.

1.10.3. Deprecated and removed features

- Starting with OpenShift Container Platform 4.11, the **preview** and **stable** channels for installing and upgrading the Red Hat OpenShift Pipelines Operator are removed. To install and upgrade the Operator, use the appropriate **pipelines-<version>** channel, or the **latest** channel for the most recent stable version. For example, to install the OpenShift Pipelines Operator version **1.8.x**, use the **pipelines-1.8** channel.

  NOTE

  In OpenShift Container Platform 4.10 and earlier versions, you can use the **preview** and **stable** channels for installing and upgrading the Operator.

- Support for the **tekton.dev/v1alpha1** API version, which was deprecated in Red Hat OpenShift Pipelines GA 1.6, is planned to be removed in the upcoming Red Hat OpenShift Pipelines GA 1.9 release.
  This change affects the pipeline component, which includes the **TaskRun**, **PipelineRun**, **Task**, **Pipeline**, and similar **tekton.dev/v1alpha1** resources. As an alternative, update existing resources to use **apiVersion: tekton.dev/v1beta1** as described in Migrating From Tekton v1alpha1 to Tekton v1beta1.

  Bug fixes and support for the **tekton.dev/v1alpha1** API version are provided only through the end of the current GA 1.8 lifecycle.

  IMPORTANT

  For the **Tekton Operator**, the **operator.tekton.dev/v1alpha1** API version is **not** deprecated. You do not need to make changes to this value.

- In Red Hat OpenShift Pipelines 1.8, the **PipelineResource** custom resource (CR) is available but no longer supported. The **PipelineResource** CR was a Tech Preview feature and part of the **tekton.dev/v1alpha1** API, which had been deprecated and planned to be removed in the upcoming Red Hat OpenShift Pipelines GA 1.9 release.

- In Red Hat OpenShift Pipelines 1.8, the **Condition** custom resource (CR) is removed. The **Condition** CR was part of the **tekton.dev/v1alpha1** API, which has been deprecated and is planned to be removed in the upcoming Red Hat OpenShift Pipelines GA 1.9 release.

- In Red Hat OpenShift Pipelines 1.8, the **gcr.io** image for **gsutil** has been removed. This removal might break clusters with **Pipeline** resources that depend on this image. Bug fixes and support are provided only through the end of the Red Hat OpenShift Pipelines 1.7 lifecycle.

- In Red Hat OpenShift Pipelines 1.8, the **PipelineRun.Status.TaskRuns** and **PipelineRun.Status.Runs** fields are deprecated and are planned to be removed in a future release. See TEP-0100: Embedded TaskRuns and Runs Status in PipelineRuns .

- In Red Hat OpenShift Pipelines 1.8, the **pipelineRunCancelled** state is deprecated and planned to be removed in a future release. Graceful termination of **PipelineRun** objects is now promoted from an alpha feature to a stable feature. (See TEP-0058: Graceful Pipeline Run Termination.) As an alternative, you can use the **Cancelled** state, which replaces the **pipelineRunCancelled** state.
  You do not need to make changes to your **Pipeline** and **Task** resources. If you have tools that cancel pipeline runs, you must update tools in the next release. This change also affects tools such as the CLI, IDE extensions, and so on, so that they support the new **PipelineRun** statuses.

  Because this feature is available by default, you no longer need to set the **pipeline.enable-api-fields** field to **alpha** in the **TektonConfig** custom resource definition.

- In Red Hat OpenShift Pipelines 1.8, the **timeout** field in **PipelineRun** has been deprecated. Instead, use the **PipelineRun.Timeouts** field, which is now promoted from an alpha feature to a stable feature.
  Because this feature is available by default, you no longer need to set the **pipeline.enable-api-fields** field to **alpha** in the **TektonConfig** custom resource definition.

- In Red Hat OpenShift Pipelines 1.8, **init** containers are omitted from the **LimitRange** object’s default request calculations.

1.10.4. Known issues

- The **s2i-nodejs** pipeline cannot use the **nodejs:14-ubi8-minimal** image stream to perform source-to-image (S2I) builds. Using that image stream produces an **error building at STEP "RUN /usr/libexec/s2i/assemble": exit status 127** message.
  Workaround: Use **nodejs:14-ubi8** rather than the **nodejs:14-ubi8-minimal** image stream.

- When you run Maven and Jib-Maven cluster tasks, the default container image is supported only on Intel (x86) architecture. Therefore, tasks will fail on ARM, IBM Power Systems (ppc64le), IBM Z, and LinuxONE (s390x) clusters.
  Workaround: Specify a custom image by setting the **MAVEN_IMAGE** parameter value to **maven:3.6.3-adoptopenjdk-11**.

  #### TIP

  Before you install tasks that are based on the Tekton Catalog on ARM, IBM Power Systems (ppc64le), IBM Z, and LinuxONE (s390x) using **tkn hub**, verify if the task can be executed on these platforms. To check if **ppc64le** and **s390x** are listed in the "Platforms" section of the task information, you can run the following command: **tkn hub info task <name>**

- On ARM, IBM Power Systems, IBM Z, and LinuxONE, the **s2i-dotnet** cluster task is unsupported.

- Implicit parameter mapping incorrectly passes parameters from the top-level **Pipeline** or **PipelineRun** definitions to the **taskRef** tasks. Mapping should only occur from a top-level resource to tasks with in-line **taskSpec** specifications. This issue only affects clusters where this feature was enabled by setting the **enable-api-fields** field to **alpha** in the **pipeline** section of the **TektonConfig** custom resource definition.

1.10.5. Fixed issues

- Before this update, the metrics for pipeline runs in the Developer view of the web console were incomplete and outdated. With this update, the issue has been fixed so that the metrics are correct.

- Before this update, if a pipeline had two parallel tasks that failed and one of them had **retries=2**, the final tasks never ran, and the pipeline timed out and failed to run. For example, the **pipelines-operator-subscription** task failed intermittently with the following error message: **Unable to connect to the server: EOF**. With this update, the issue has been fixed so that the final tasks always run.

- Before this update, if a pipeline run stopped because a task run failed, other task runs might not complete their retries. As a result, no **finally** tasks were scheduled, which caused the pipeline to hang. This update resolves the issue. **TaskRuns** and **Run** objects can retry when a pipeline run has stopped, even by graceful stopping, so that pipeline runs can complete.

- This update changes how resource requirements are calculated when one or more **LimitRange** objects are present in the namespace where a **TaskRun** object exists. The scheduler now considers **step** containers and excludes all other app containers, such as sidecar containers, when factoring requests from **LimitRange** objects.

- Before this update, under specific conditions, the flag package might incorrectly parse a subcommand immediately following a double dash flag terminator, **--**. In that case, it ran the entrypoint subcommand rather than the actual command. This update fixes this flag-parsing issue so that the entrypoint runs the correct command.

- Before this update, the controller might generate multiple panics if pulling an image failed, or its pull status was incomplete. This update fixes the issue by checking the **step.ImageID** value rather than the **status.TaskSpec** value.

- Before this update, canceling a pipeline run that contained an unscheduled custom task produced a **PipelineRunCouldntCancel** error. This update fixes the issue. You can cancel a pipeline run that contains an unscheduled custom task without producing that error.

- Before this update, if the **<NAME>** in **$params["<NAME>"]** or **$params['<NAME>']** contained a dot character (**.**), any part of the name to the right of the dot was not extracted. For example, from **$params["org.ipsum.lorem"]**, only **org** was extracted.
  This update fixes the issue so that **$params** fetches the complete value. For example, **$params["org.ipsum.lorem"]** and **$params['org.ipsum.lorem']** are valid and the entire value of **<NAME>**, **org.ipsum.lorem**, is extracted.

  It also throws an error if **<NAME>** is not enclosed in single or double quotes. For example, **$params.org.ipsum.lorem** is not valid and generates a validation error.

- With this update, **Trigger** resources support custom interceptors and ensure that the port of the custom interceptor service is the same as the port in the **ClusterInterceptor** definition file.

- Before this update, the **tkn version** command for Tekton Chains and Operator components did not work correctly. This update fixes the issue so that the command works correctly and returns version information for those components.

- Before this update, if you ran a **tkn pr delete --ignore-running** command and a pipeline run did not have a **status.condition** value, the **tkn** CLI tool produced a null-pointer error (NPE). This update fixes the issue so that the CLI tool now generates an error and correctly ignores pipeline runs that are still running.

- Before this update, if you used the **tkn pr delete --keep <value>** or **tkn tr delete --keep <value>** commands, and the number of pipeline runs or task runs was less than the value, the command did not return an error as expected. This update fixes the issue so that the command correctly returns an error under those conditions.

- Before this update, if you used the **tkn pr delete** or **tkn tr delete** commands with the **-p** or **-t** flags together with the **--ignore-running** flag, the commands incorrectly deleted running or pending resources. This update fixes the issue so that these commands correctly ignore running or pending resources.

- With this update, you can configure Tekton Chains by using the **TektonChain** custom resource. This feature enables your configuration to persist after upgrading, unlike the **chains-config** config map, which gets overwritten during upgrades.

- With this update, **ClusterTask** resources no longer run as root by default, except for the **buildah** and **s2i** cluster tasks.

- Before this update, tasks on Red Hat OpenShift Pipelines 1.7.1 failed when using **init** as a first argument followed by two or more arguments. With this update, the flags are parsed correctly, and the task runs are successful.

- Before this update, installation of the Red Hat OpenShift Pipelines Operator on OpenShift Container Platform 4.9 and 4.10 failed due to an invalid role binding, with the following error message:

  ```
  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
  ```

  This update fixes the issue so that the failure no longer occurs.

- Previously, upgrading the Red Hat OpenShift Pipelines Operator caused the **pipeline** service account to be recreated, which meant that the secrets linked to the service account were lost. This update fixes the issue. During upgrades, the Operator no longer recreates the **pipeline** service account. As a result, secrets attached to the **pipeline** service account persist after upgrades, and the resources (tasks and pipelines) continue to work correctly.

- With this update, Pipelines as Code pods run on infrastructure nodes if infrastructure node settings are configured in the **TektonConfig** custom resource (CR).

- Previously, with the resource pruner, each namespace Operator created a command that ran in a separate container. This design consumed too many resources in clusters with a high number of namespaces. For example, to run a single command, a cluster with 1000 namespaces produced 1000 containers in a pod.
  This update fixes the issue. It passes the namespace-based configuration to the job so that all the commands run in one container in a loop.

- In Tekton Chains, you must define a secret called **signing-secrets** to hold the key used for signing tasks and images. However, before this update, updating the Red Hat OpenShift Pipelines Operator reset or overwrote this secret, and the key was lost. This update fixes the issue. Now, if the secret is configured after installing Tekton Chains through the Operator, the secret persists, and it is not overwritten by upgrades.

- Before this update, all S2I build tasks failed with an error similar to the following message:

  ```
  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)"
  ```

  With this update, the **pipelines-scc** security context constraint (SCC) is compatible with the **SETFCAP** capability necessary for **Buildah** and **S2I** cluster tasks. As a result, the **Buildah** and **S2I** build tasks can run successfully.

  To successfully run the **Buildah** cluster task and **S2I** build tasks for applications written in various languages and frameworks, add the following snippet for appropriate **steps** objects such as **build** and **push**:

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

- Before this update, installing the Red Hat OpenShift Pipelines Operator took longer than expected. This update optimizes some settings to speed up the installation process.

- With this update, Buildah and S2I cluster tasks have fewer steps than in previous versions. Some steps have been combined into a single step so that they work better with **ResourceQuota** and **LimitRange** objects and do not require more resources than necessary.

- This update upgrades the Buildah, **tkn** CLI tool, and **skopeo** CLI tool versions in cluster tasks.

- Before this update, the Operator failed when creating RBAC resources if any namespace was in a **Terminating** state. With this update, the Operator ignores namespaces in a **Terminating** state and creates the RBAC resources.

- Before this update, pods for the prune cronjobs were not scheduled on infrastructure nodes, as expected. Instead, they were scheduled on worker nodes or not scheduled at all. With this update, these types of pods can now be scheduled on infrastructure nodes if configured in the **TektonConfig** custom resource (CR).

1.10.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.8.1

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.8.1 is available on OpenShift Container Platform 4.10, 4.11, and 4.12.

1.10.6.1. Known issues

- By default, the containers have restricted permissions for enhanced security. The restricted permissions apply to all controller pods in the Red Hat OpenShift Pipelines Operator, and to some cluster tasks. Due to restricted permissions, the **git-clone** cluster task fails under certain configurations.
  Workaround: None. You can track the issue SRVKP-2634.

- When installer sets are in a failed state, the status of the **TektonConfig** custom resource is incorrectly displayed as **True** instead of **False**.

  Example: Failed installer sets

  ```
  $ 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
  ```

  Example: Incorrect TektonConfig status

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

1.10.6.2. Fixed issues

- Before this update, the pruner deleted task runs of running pipelines and displayed the following warning: **some tasks were indicated completed without ancestors being done**. With this update, the pruner retains the task runs that are part of running pipelines.

- Before this update, **pipeline-1.8** was the default channel for installing the Red Hat OpenShift Pipelines Operator 1.8.x. With this update, **latest** is the default channel.

- Before this update, the Pipelines as Code controller pods did not have access to certificates exposed by the user. With this update, Pipelines as Code can now access routes and Git repositories guarded by a self-signed or a custom certificate.

- Before this update, the task failed with RBAC errors after upgrading from Red Hat OpenShift Pipelines 1.7.2 to 1.8.0. With this update, the tasks run successfully without any RBAC errors.

- Before this update, using the **tkn** CLI tool, you could not remove task runs and pipeline runs that contained a **result** object whose type was **array**. With this update, you can use the **tkn** CLI tool to remove task runs and pipeline runs that contain a **result** object whose type is **array**.

- Before this update, if a pipeline specification contained a task with an **ENV_VARS** parameter of **array** type, the pipeline run failed with the following error: **invalid input params for task func-buildpacks: param types don’t match the user-specified type: [ENV_VARS]**. With this update, pipeline runs with such pipeline and task specifications do not fail.

- Before this update, cluster administrators could not provide a **config.json** file to the **Buildah** cluster task for accessing a container registry. With this update, cluster administrators can provide the **Buildah** cluster task with a **config.json** file by using the **dockerconfig** workspace.

1.10.7. Release notes for Red Hat OpenShift Pipelines General Availability 1.8.2

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.8.2 is available on OpenShift Container Platform 4.10, 4.11, and 4.12.

1.10.7.1. Fixed issues

- Before this update, the **git-clone** task failed when cloning a repository using SSH keys. With this update, the role of the non-root user in the **git-init** task is removed, and the SSH program looks in the **$HOME/.ssh/** directory for the correct keys.

1.11. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.7

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.7 is available on OpenShift Container Platform 4.9, 4.10, and 4.11.

1.11.1. New features

In addition to the fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.7.

1.11.1.1. Pipelines

- With this update, **pipelines-<version>** is the default channel to install the Red Hat OpenShift Pipelines Operator. For example, the default channel to install the OpenShift Pipelines Operator version **1.7** is **pipelines-1.7**. Cluster administrators can also use the **latest** channel to install the most recent stable version of the Operator.

  NOTE

  The **preview** and **stable** channels will be deprecated and removed in a future release.

- When you run a command in a user namespace, your container runs as **root** (user id **0**) but has user privileges on the host. With this update, to run pods in the user namespace, you must pass the annotations that CRI-O expects.
  - To add these annotations for all users, run the **oc edit clustertask buildah** command and edit the **buildah** cluster task.
  - To add the annotations to a specific namespace, export the cluster task as a task to that namespace.
- Before this update, if certain conditions were not met, the **when** expression skipped a **Task** object and its dependent tasks. With this update, you can scope the **when** expression to guard the **Task** object only, not its dependent tasks. To enable this update, set the **scope-when-expressions-to-task** flag to **true** in the **TektonConfig** CRD.

NOTE

The **scope-when-expressions-to-task** flag is deprecated and will be removed in a future release. As a best practice for OpenShift Pipelines, use **when** expressions scoped to the guarded **Task** only.

- With this update, you can use variable substitution in the **subPath** field of a workspace within a task.
- With this update, you can reference parameters and results by using a bracket notation with single or double quotes. Prior to this update, you could only use the dot notation. For example, the following are now equivalent:
    - **$(param.myparam)**, **$(param['myparam'])**, and **$(param["myparam"])**.
      You can use single or double quotes to enclose parameter names that contain problematic characters, such as **"."**. For example, **$(param['my.param'])** and **$(param["my.param"])**.
- With this update, you can include the **onError** parameter of a step in the task definition without enabling the **enable-api-fields** flag.

1.11.1.2. Triggers

- With this update, the **feature-flag-triggers** config map has a new field **labels-exclusion-pattern**. You can set the value of this field to a regular expression (regex) pattern. The controller filters out labels that match the regex pattern from propagating from the event listener to the resources created for the event listener.
- With this update, the **TriggerGroups** field is added to the **EventListener** specification. Using this field, you can specify a set of interceptors to run before selecting and running a group of triggers. To enable this feature, in the **TektonConfig** custom resource definition, in the **pipeline** section, you must set the **enable-api-fields** field to **alpha**.
- With this update, **Trigger** resources support custom runs defined by a **TriggerTemplate** template.
- With this update, Triggers support emitting Kubernetes events from an **EventListener** pod.
- With this update, count metrics are available for the following objects: **ClusterInteceptor**, **EventListener**, **TriggerTemplate**, **ClusterTriggerBinding**, and **TriggerBinding**.
- This update adds the **ServicePort** specification to Kubernetes resource. You can use this specification to modify which port exposes the event listener service. The default port is **8080**.
- With this update, you can use the **targetURI** field in the **EventListener** specification to send cloud events during trigger processing. To enable this feature, in the **TektonConfig** custom resource definition, in the **pipeline** section, you must set the **enable-api-fields** field to **alpha**.
- With this update, the **tekton-triggers-eventlistener-roles** object now has a **patch** verb, in addition to the **create** verb that already exists.
- With this update, the **securityContext.runAsUser** parameter is removed from event listener deployment.

1.11.1.3. CLI

- With this update, the **tkn [pipeline | pipelinerun] export** command exports a pipeline or pipeline run as a YAML file. For example:
  - Export a pipeline named **test_pipeline** in the **openshift-pipelines** namespace:

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

  - Export a pipeline run named **test_pipeline_run** in the **openshift-pipelines** namespace:

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

- With this update, the **--grace** option is added to the **tkn pipelinerun cancel**. Use the **--grace** option to terminate a pipeline run gracefully instead of forcing the termination. To enable this feature, in the **TektonConfig** custom resource definition, in the **pipeline** section, you must set the **enable-api-fields** field to **alpha**.

- This update adds the Operator and Chains versions to the output of the **tkn version** command.

  IMPORTANT

  Tekton Chains is a Technology Preview feature.

- With this update, the **tkn pipelinerun describe** command displays all canceled task runs, when you cancel a pipeline run. Before this fix, only one task run was displayed.

- With this update, you can skip supplying the asking specifications for optional workspace when you run the **tkn [t | p | ct] start** command skips with the **--skip-optional-workspace** flag. You can also skip it when running in interactive mode.

- With this update, you can use the **tkn chains** command to manage Tekton Chains. You can also use the **--chains-namespace** option to specify the namespace where you want to install Tekton Chains.

  IMPORTANT

  Tekton Chains is a Technology Preview feature.

1.11.1.4. Operator

- With this update, you can use the Red Hat OpenShift Pipelines Operator to install and deploy Tekton Hub and Tekton Chains.

  IMPORTANT

  Tekton Chains and deployment of Tekton Hub on a cluster are Technology Preview features.

- With this update, you can find and use Pipelines as Code (PAC) as an add-on option.

  IMPORTANT

  Pipelines as Code is a Technology Preview feature.

- With this update, you can now disable the installation of community cluster tasks by setting the **communityClusterTasks** parameter to **false**. For example:

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

- With this update, you can disable the integration of Tekton Hub with the **Developer** perspective by setting the **enable-devconsole-integration** flag in the **TektonConfig** custom resource to **false**. For example:

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

- With this update, the **operator-config.yaml** config map enables the output of the **tkn version** command to display of the Operator version.

- With this update, the version of the **argocd-task-sync-and-wait** tasks is modified to **v0.2**.

- With this update to the **TektonConfig** CRD, the **oc get tektonconfig** command displays the OPerator version.

- With this update, service monitor is added to the Triggers metrics.

1.11.1.5. Hub

IMPORTANT

Deploying Tekton Hub on a cluster is a Technology Preview feature.

Tekton Hub helps you discover, search, and share reusable tasks and pipelines for your CI/CD workflows. A public instance of Tekton Hub is available at hub.tekton.dev.

Staring with Red Hat OpenShift Pipelines 1.7, cluster administrators can also install and deploy a custom instance of Tekton Hub on enterprise clusters. You can curate a catalog with reusable tasks and pipelines specific to your organization.

1.11.1.6. Chains

IMPORTANT

Tekton Chains is a Technology Preview feature.

Tekton Chains is a Kubernetes Custom Resource Definition (CRD) controller. You can use it to manage the supply chain security of the tasks and pipelines created using Red Hat OpenShift Pipelines.

By default, Tekton Chains monitors the task runs in your OpenShift Container Platform cluster. Chains takes snapshots of completed task runs, converts them to one or more standard payload formats, and signs and stores all artifacts.

Tekton Chains supports the following features:

- You can sign task runs, task run results, and OCI registry images with cryptographic key types and services such as **cosign**.
- You can use attestation formats such as **in-toto**.
- You can securely store signatures and signed artifacts using OCI repository as a storage backend.

1.11.1.7. Pipelines as Code (PAC)

IMPORTANT

Pipelines as Code is a Technology Preview feature.

With Pipelines as Code, cluster administrators and users with the required privileges can define pipeline templates as part of source code Git repositories. When triggered by a source code push or a pull request for the configured Git repository, the feature runs the pipeline and reports status.

Pipelines as Code supports the following features:

- Pull request status. When iterating over a pull request, the status and control of the pull request is exercised on the platform hosting the Git repository.
- GitHub checks the API to set the status of a pipeline run, including rechecks.
- GitHub pull request and commit events.
- Pull request actions in comments, such as **/retest**.
- Git events filtering, and a separate pipeline for each event.
- Automatic task resolution in OpenShift Pipelines for local tasks, Tekton Hub, and remote URLs.
- Use of GitHub blobs and objects API for retrieving configurations.
- Access Control List (ACL) over a GitHub organization, or using a Prow-style **OWNER** file.
- The **tkn pac** plugin for the **tkn** CLI tool, which you can use to manage Pipelines as Code repositories and bootstrapping.
- Support for GitHub Application, GitHub Webhook, Bitbucket Server, and Bitbucket Cloud.

1.11.2. Deprecated features

- Breaking change: This update removes the **disable-working-directory-overwrite** and **disable-home-env-overwrite** fields from the **TektonConfig** custom resource (CR). As a result, the **TektonConfig** CR no longer automatically sets the **$HOME** environment variable and **workingDir** parameter. You can still set the **$HOME** environment variable and **workingDir** parameter by using the **env** and **workingDir** fields in the **Task** custom resource definition (CRD).

- The **Conditions** custom resource definition (CRD) type is deprecated and planned to be removed in a future release. Instead, use the recommended **When** expression.

- Breaking change: The **Triggers** resource validates the templates and generates an error if you do not specify the **EventListener** and **TriggerBinding** values.

1.11.3. Known issues

- When you run Maven and Jib-Maven cluster tasks, the default container image is supported only on Intel (x86) architecture. Therefore, tasks will fail on ARM, IBM Power Systems (ppc64le), IBM Z, and LinuxONE (s390x) clusters. As a workaround, you can specify a custom image by setting the **MAVEN_IMAGE** parameter value to **maven:3.6.3-adoptopenjdk-11**.

  TIP

  Before you install tasks that are based on the Tekton Catalog on ARM, IBM Power Systems (ppc64le), IBM Z, and LinuxONE (s390x) using **tkn hub**, verify if the task can be executed on these platforms. To check if **ppc64le** and **s390x** are listed in the "Platforms" section of the task information, you can run the following command: **tkn hub info task <name>**

- On IBM Power Systems, IBM Z, and LinuxONE, the **s2i-dotnet** cluster task is unsupported.

- You cannot use the **nodejs:14-ubi8-minimal** image stream because doing so generates the following errors:

```
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"
```

- Implicit parameter mapping incorrectly passes parameters from the top-level **Pipeline** or **PipelineRun** definitions to the **taskRef** tasks. Mapping should only occur from a top-level resource to tasks with in-line **taskSpec** specifications. This issue only affects clusters where this feature was enabled by setting the **enable-api-fields** field to **alpha** in the **pipeline** section of the **TektonConfig** custom resource definition.

1.11.4. Fixed issues

- With this update, if metadata such as **labels** and **annotations** are present in both **Pipeline** and **PipelineRun** object definitions, the values in the **PipelineRun** type takes precedence. You can observe similar behavior for **Task** and **TaskRun** objects.

- With this update, if the **timeouts.tasks** field or the **timeouts.finally** field is set to **0**, then the **timeouts.pipeline** is also set to **0**.

- With this update, the **-x** set flag is removed from scripts that do not use a shebang. The fix reduces potential data leak from script execution.

- With this update, any backslash character present in the usernames in Git credentials is escaped with an additional backslash in the **.gitconfig** file.

- With this update, the **finalizer** property of the **EventListener** object is not necessary for cleaning up logging and config maps.

- With this update, the default HTTP client associated with the event listener server is removed, and a custom HTTP client added. As a result, the timeouts have improved.

- With this update, the Triggers cluster role now works with owner references.

- With this update, the race condition in the event listener does not happen when multiple interceptors return extensions.

- With this update, the **tkn pr delete** command does not delete the pipeline runs with the **ignore-running** flag.

- With this update, the Operator pods do not continue restarting when you modify any add-on parameters.

- With this update, the **tkn serve** CLI pod is scheduled on infra nodes, if not configured in the subscription and config custom resources.

- With this update, cluster tasks with specified versions are not deleted during upgrade.

1.11.5. Release notes for Red Hat OpenShift Pipelines General Availability 1.7.1

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.7.1 is available on OpenShift Container Platform 4.9, 4.10, and 4.11.

1.11.5.1. Fixed issues

- Before this update, upgrading the Red Hat OpenShift Pipelines Operator deleted the data in the database associated with Tekton Hub and installed a new database. With this update, an Operator upgrade preserves the data.

- Before this update, only cluster administrators could access pipeline metrics in the OpenShift Container Platform console. With this update, users with other cluster roles also can access the pipeline metrics.

- Before this update, pipeline runs failed for pipelines containing tasks that emit large termination messages. The pipeline runs failed because the total size of termination messages of all containers in a pod cannot exceed 12 KB. With this update, the **place-tools** and **step-init** initialization containers that uses the same image are merged to reduce the number of containers running in each tasks's pod. The solution reduces the chance of failed pipeline runs by minimizing the number of containers running in a task's pod. However, it does not remove the limitation of the maximum allowed size of a termination message.

- Before this update, attempts to access resource URLs directly from the Tekton Hub web console resulted in an Nginx **404** error. With this update, the Tekton Hub web console image is fixed to allow accessing resource URLs directly from the Tekton Hub web console.

- Before this update, for each namespace the resource pruner job created a separate container to prune resources. With this update, the resource pruner job runs commands for all namespaces as a loop in one container.

1.11.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.7.2

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.7.2 is available on OpenShift Container Platform 4.9, 4.10, and the upcoming version.

1.11.6.1. Known issues

- The **chains-config** config map for Tekton Chains in the **openshift-pipelines** namespace is automatically reset to default after upgrading the Red Hat OpenShift Pipelines Operator. Currently, there is no workaround for this issue.

1.11.6.2. Fixed issues

- Before this update, tasks on OpenShift Pipelines 1.7.1 failed on using **init** as the first argument, followed by two or more arguments. With this update, the flags are parsed correctly and the task runs are successful.

- Before this update, installation of the Red Hat OpenShift Pipelines Operator on OpenShift Container Platform 4.9 and 4.10 failed due to invalid role binding, with the following error message:

  > 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

  With this update, the Red Hat OpenShift Pipelines Operator installs with distinct role binding namespaces to avoid conflict with installation of other Operators.

- Before this update, upgrading the Operator triggered a reset of the **signing-secrets** secret key for Tekton Chains to its default value. With this update, the custom secret key persists after you upgrade the Operator.

  NOTE

  Upgrading to Red Hat OpenShift Pipelines 1.7.2 resets the key. However, when you upgrade to future releases, the key is expected to persist.

- Before this update, all S2I build tasks failed with an error similar to the following message:

  > 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)"

  With this update, the **pipelines-scc** security context constraint (SCC) is compatible with the **SETFCAP** capability necessary for **Buildah** and **S2I** cluster tasks. As a result, the **Buildah** and **S2I** build tasks can run successfully.

To successfully run the **Buildah** cluster task and **S2I** build tasks for applications written in various languages and frameworks, add the following snippet for appropriate **steps** objects such as **build** and **push**:

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

1.11.7. Release notes for Red Hat OpenShift Pipelines General Availability 1.7.3

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.7.3 is available on OpenShift Container Platform 4.9, 4.10, and 4.11.

1.11.7.1. Fixed issues

- Before this update, the Operator failed when creating RBAC resources if any namespace was in a **Terminating** state. With this update, the Operator ignores namespaces in a **Terminating** state and creates the RBAC resources.

- Previously, upgrading the Red Hat OpenShift Pipelines Operator caused the **pipeline** service account to be recreated, which meant that the secrets linked to the service account were lost. This update fixes the issue. During upgrades, the Operator no longer recreates the **pipeline** service account. As a result, secrets attached to the **pipeline** service account persist after upgrades, and the resources (tasks and pipelines) continue to work correctly.

1.12. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.6

With this update, Red Hat OpenShift Pipelines General Availability (GA) 1.6 is available on OpenShift Container Platform 4.9.

1.12.1. New features

In addition to the fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.6.

- With this update, you can configure a pipeline or task **start** command to return a YAML or JSON-formatted string by using the **--output <string>**, where **<string>** is **yaml** or **json**. Otherwise, without the **--output** option, the **start** command returns a human-friendly message that is hard for other programs to parse. Returning a YAML or JSON-formatted string is useful for continuous integration (CI) environments. For example, after a resource is created, you can use **yq** or **jq** to parse the YAML or JSON-formatted message about the resource and wait until that resource is terminated without using the **showlog** option.

- With this update, you can authenticate to a registry using the **auth.json** authentication file of Podman. For example, you can use **tkn bundle push** to push to a remote registry using Podman instead of Docker CLI.

- With this update, if you use the **tkn [taskrun | pipelinerun] delete --all** command, you can preserve runs that are younger than a specified number of minutes by using the new **--keep-since <minutes>** option. For example, to keep runs that are less than five minutes old, you enter **tkn [taskrun | pipelinerun] delete -all --keep-since 5**.

- With this update, when you delete task runs or pipeline runs, you can use the **--parent-resource** and **--keep-since** options together. For example, the **tkn pipelinerun delete --pipeline pipelinename --keep-since 5** command preserves pipeline runs whose parent resource is named **pipelinename** and whose age is five minutes or less. The **tkn tr delete -t <taskname> --keep-since 5** and **tkn tr delete --clustertask <taskname> --keep-since 5** commands work similarly for task runs.

- This update adds support for the triggers resources to work with **v1beta1** resources.
- This update adds an **ignore-running** option to the **tkn pipelinerun delete** and **tkn taskrun delete** commands.
- This update adds a **create** subcommand to the **tkn task** and **tkn clustertask** commands.
- With this update, when you use the **tkn pipelinerun delete --all** command, you can use the new **--label <string>** option to filter the pipeline runs by label. Optionally, you can use the **--label** option with **=** and **==** as equality operators, or **!=** as an inequality operator. For example, the **tkn pipelinerun delete --all --label asdf** and **tkn pipelinerun delete --all --label==asdf** commands both delete all the pipeline runs that have the **asdf** label.
- With this update, you can fetch the version of installed Tekton components from the config map or, if the config map is not present, from the deployment controller.
- With this update, triggers support the **feature-flags** and **config-defaults** config map to configure feature flags and to set default values respectively.
- This update adds a new metric, **eventlistener_event_count**, that you can use to count events received by the **EventListener** resource.
- This update adds **v1beta1** Go API types. With this update, triggers now support the **v1beta1** API version.
  With the current release, the **v1alpha1** features are now deprecated and will be removed in a future release. Begin using the **v1beta1** features instead.
- In the current release, auto-prunning of resources is enabled by default. In addition, you can configure auto-prunning of task run and pipeline run for each namespace separately, by using the following new annotations:
  - **operator.tekton.dev/prune.schedule**: If the value of this annotation is different from the value specified at the **TektonConfig** custom resource definition, a new cron job in that namespace is created.
  - **operator.tekton.dev/prune.skip**: When set to **true**, the namespace for which it is configured will not be prunned.
  - **operator.tekton.dev/prune.resources**: This annotation accepts a comma-separated list of resources. To prune a single resource such as a pipeline run, set this annotation to **"pipelinerun"**. To prune multiple resources, such as task run and pipeline run, set this annotation to **"taskrun, pipelinerun"**.
  - **operator.tekton.dev/prune.keep**: Use this annotation to retain a resource without prunning.
  - **operator.tekton.dev/prune.keep-since**: Use this annotation to retain resources based on their age. The value for this annotation must be equal to the age of the resource in minutes. For example, to retain resources which were created not more than five days ago, set **keep-since** to **7200**.

NOTE

The **keep** and **keep-since** annotations are mutually exclusive. For any resource, you must configure only one of them.

  - **operator.tekton.dev/prune.strategy**: Set the value of this annotation to either **keep** or **keep-since**.

- Administrators can disable the creation of the **pipeline** service account for the entire cluster, and prevent privilege escalation by misusing the associated SCC, which is very similar to **anyuid**.

- You can now configure feature flags and components by using the **TektonConfig** custom resource (CR) and the CRs for individual components, such as **TektonPipeline** and **TektonTriggers**. This level of granularity helps customize and test alpha features such as the Tekton OCI bundle for individual components.

- You can now configure optional **Timeouts** field for the **PipelineRun** resource. For example, you can configure timeouts separately for a pipeline run, each task run, and the **finally** tasks.

- The pods generated by the **TaskRun** resource now sets the **activeDeadlineSeconds** field of the pods. This enables OpenShift to consider them as terminating, and allows you to use specifically scoped **ResourceQuota** object for the pods.

- You can use configmaps to eliminate metrics tags or labels type on a task run, pipeline run, task, and pipeline. In addition, you can configure different types of metrics for measuring duration, such as a histogram, gauge, or last value.

- You can define requests and limits on a pod coherently, as Tekton now fully supports the **LimitRange** object by considering the **Min**, **Max**, **Default**, and **DefaultRequest** fields.

- The following alpha features are introduced:

  - A pipeline run can now stop after running the **finally** tasks, rather than the previous behavior of stopping the execution of all task run directly. This update adds the following **spec.status** values:

    - **StoppedRunFinally** will stop the currently running tasks after they are completed, and then run the **finally** tasks.

    - **CancelledRunFinally** will immediately cancel the running tasks, and then run the **finally** tasks.

    - **Cancelled** will retain the previous behavior provided by the **PipelineRunCancelled** status.

      NOTE

      The **Cancelled** status replaces the deprecated **PipelineRunCancelled** status, which will be removed in the **v1** version.

  - You can now use the **oc debug** command to put a task run into debug mode, which pauses the execution and allows you to inspect specific steps in a pod.

  - When you set the **onError** field of a step to **continue**, the exit code for the step is recorded and passed on to subsequent steps. However, the task run does not fail and the execution of the rest of the steps in the task continues. To retain the existing behavior, you can set the value of the **onError** field to **stopAndFail**.

- Tasks can now accept more parameters than are actually used. When the alpha feature flag is enabled, the parameters can implicitly propagate to inlined specs. For example, an inlined task can access parameters of its parent pipeline run, without explicitly defining each parameter for the task.

- If you enable the flag for the alpha features, the conditions under **When** expressions will only apply to the task with which it is directly associated, and not the dependents of the task. To apply the **When** expressions to the associated task and its dependents, you must associate the expression with each dependent task separately. Note that, going forward, this will be the default behavior of the **When** expressions in any new API versions of Tekton. The existing default behavior will be deprecated in favor of this update.

- The current release enables you to configure node selection by specifying the **nodeSelector** and **tolerations** values in the **TektonConfig** custom resource (CR). The Operator adds these values to all the deployments that it creates.
    - To configure node selection for the Operator’s controller and webhook deployment, you edit the **config.nodeSelector** and **config.tolerations** fields in the specification for the **Subscription** CR, after installing the Operator.
    - To deploy the rest of the control plane pods of OpenShift Pipelines on an infrastructure node, update the **TektonConfig** CR with the **nodeSelector** and **tolerations** fields. The modifications are then applied to all the pods created by Operator.

1.12.2. Deprecated features

- In CLI 0.21.0, support for all **v1alpha1** resources for **clustertask**, **task**, **taskrun**, **pipeline**, and **pipelinerun** commands are deprecated. These resources are now deprecated and will be removed in a future release.

- In Tekton Triggers v0.16.0, the redundant **status** label is removed from the metrics for the **EventListener** resource.

    IMPORTANT

    Breaking change: The **status** label has been removed from the **eventlistener_http_duration_seconds_*** metric. Remove queries that are based on the **status** label.

- With the current release, the **v1alpha1** features are now deprecated and will be removed in a future release. With this update, you can begin using the **v1beta1** Go API types instead. Triggers now supports the **v1beta1** API version.

- With the current release, the **EventListener** resource sends a response before the triggers finish processing.

    IMPORTANT

    Breaking change: With this change, the **EventListener** resource stops responding with a **201 Created** status code when it creates resources. Instead, it responds with a **202 Accepted** response code.

- The current release removes the **podTemplate** field from the **EventListener** resource.

IMPORTANT

Breaking change: The **podTemplate** field, which was deprecated as part of #1100, has been removed.

- The current release removes the deprecated **replicas** field from the specification for the **EventListener** resource.

IMPORTANT

Breaking change: The deprecated **replicas** field has been removed.

- In Red Hat OpenShift Pipelines 1.6, the values of **HOME="/tekton/home"** and **workingDir="/workspace"** are removed from the specification of the **Step** objects. Instead, Red Hat OpenShift Pipelines sets **HOME** and **workingDir** to the values defined by the containers running the **Step** objects. You can override these values in the specification of your **Step** objects.

  To use the older behavior, you can change the **disable-working-directory-overwrite** and **disable-home-env-overwrite** fields in the **TektonConfig** CR to **false**:

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

IMPORTANT

The **disable-working-directory-overwrite** and **disable-home-env-overwrite** fields in the **TektonConfig** CR are now deprecated and will be removed in a future release.

1.12.3. Known issues

- When you run Maven and Jib-Maven cluster tasks, the default container image is supported only on Intel (x86) architecture. Therefore, tasks will fail on IBM Power Systems (ppc64le), IBM Z, and LinuxONE (s390x) clusters. As a workaround, you can specify a custom image by setting the **MAVEN_IMAGE** parameter value to **maven:3.6.3-adoptopenjdk-11**.

- On IBM Power Systems, IBM Z, and LinuxONE, the **s2i-dotnet** cluster task is unsupported.

- Before you install tasks based on the Tekton Catalog on IBM Power Systems (ppc64le), IBM Z, and LinuxONE (s390x) using **tkn hub**, verify if the task can be executed on these platforms. To check if **ppc64le** and **s390x** are listed in the "Platforms" section of the task information, you can run the following command: **tkn hub info task <name>**

- You cannot use the **nodejs:14-ubi8-minimal** image stream because doing so generates the following errors:

```
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"
```

1.12.4. Fixed issues

- The **tkn hub** command is now supported on IBM Power Systems, IBM Z, and LinuxONE.

- Before this update, the terminal was not available after the user ran a **tkn** command, and the pipeline run was done, even if **retries** were specified. Specifying a timeout in the task run or pipeline run had no effect. This update fixes the issue so that the terminal is available after running the command.

- Before this update, running **tkn pipelinerun delete --all** would delete all resources. This update prevents the resources in the running state from getting deleted.

- Before this update, using the **tkn version --component=<component>** command did not return the component version. This update fixes the issue so that this command returns the component version.

- Before this update, when you used the **tkn pr logs** command, it displayed the pipelines output logs in the wrong task order. This update resolves the issue so that logs of completed **PipelineRuns** are listed in the appropriate **TaskRun** execution order.

- Before this update, editing the specification of a running pipeline might prevent the pipeline run from stopping when it was complete. This update fixes the issue by fetching the definition only once and then using the specification stored in the status for verification. This change reduces the probability of a race condition when a **PipelineRun** or a **TaskRun** refers to a **Pipeline** or **Task** that changes while it is running.

- **When** expression values can now have array parameter references, such as: **values: [$(params.arrayParam[*])]**.

1.12.5. Release notes for Red Hat OpenShift Pipelines General Availability 1.6.1

1.12.5.1. Known issues

- After upgrading to Red Hat OpenShift Pipelines 1.6.1 from an older version, OpenShift Pipelines might enter an inconsistent state where you are unable to perform any operations (create/delete/apply) on Tekton resources (tasks and pipelines). For example, while deleting a resource, you might encounter the following error:

  ```
  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.
  ```

1.12.5.2. Fixed issues

- The **SSL_CERT_DIR** environment variable (**/tekton-custom-certs**) set by Red Hat OpenShift Pipelines will not override the following default system directories with certificate files:

  - **/etc/pki/tls/certs**
  - **/etc/ssl/certs**
  - **/system/etc/security/cacerts**

- The Horizontal Pod Autoscaler can manage the replica count of deployments controlled by the Red Hat OpenShift Pipelines Operator. From this release onward, if the count is changed by an end user or an on-cluster agent, the Red Hat OpenShift Pipelines Operator will not reset the replica count of deployments managed by it. However, the replicas will be reset when you upgrade the Red Hat OpenShift Pipelines Operator.
- The pod serving the **tkn** CLI will now be scheduled on nodes, based on the node selector and toleration limits specified in the **TektonConfig** custom resource.

1.12.6. Release notes for Red Hat OpenShift Pipelines General Availability 1.6.2

1.12.6.1. Known issues

- When you create a new project, the creation of the **pipeline** service account is delayed, and removal of existing cluster tasks and pipeline templates takes more than 10 minutes.

1.12.6.2. Fixed issues

- Before this update, multiple instances of Tekton installer sets were created for a pipeline after upgrading to Red Hat OpenShift Pipelines 1.6.1 from an older version. With this update, the Operator ensures that only one instance of each type of **TektonInstallerSet** exists after an upgrade.
- Before this update, all the reconcilers in the Operator used the component version to decide resource recreation during an upgrade to Red Hat OpenShift Pipelines 1.6.1 from an older version. As a result, those resources were not recreated whose component versions did not change in the upgrade. With this update, the Operator uses the Operator version instead of the component version to decide resource recreation during an upgrade.
- Before this update, the pipelines webhook service was missing in the cluster after an upgrade. This was due to an upgrade deadlock on the config maps. With this update, a mechanism is added to disable webhook validation if the config maps are absent in the cluster. As a result, the pipelines webhook service persists in the cluster after an upgrade.
- Before this update, cron jobs for auto-pruning got recreated after any configuration change to the namespace. With this update, cron jobs for auto-pruning get recreated only if there is a relevant annotation change in the namespace.
- The upstream version of Tekton Pipelines is revised to **v0.28.3**, which has the following fixes:
  - Fix **PipelineRun** or **TaskRun** objects to allow label or annotation propagation.
  - For implicit params:
    - Do not apply the **PipelineSpec** parameters to the **TaskRefs** object.
    - Disable implicit param behavior for the **Pipeline** objects.

1.12.7. Release notes for Red Hat OpenShift Pipelines General Availability 1.6.3

1.12.7.1. Fixed issues

- Before this update, the Red Hat OpenShift Pipelines Operator installed pod security policies from components such as Pipelines and Triggers. However, the pod security policies shipped as part of the components were deprecated in an earlier release. With this update, the Operator stops installing pod security policies from components. As a result, the following upgrade paths are affected:
  - Upgrading from OpenShift Pipelines 1.6.1 or 1.6.2 to OpenShift Pipelines 1.6.3 deletes the pod security policies, including those from the Pipelines and Triggers components.
  - Upgrading from OpenShift Pipelines 1.5.x to 1.6.3 retains the pod security policies installed from components. As a cluster administrator, you can delete them manually.

  NOTE

  When you upgrade to future releases, the Red Hat OpenShift Pipelines Operator will automatically delete all obsolete pod security policies.

- Before this update, only cluster administrators could access pipeline metrics in the OpenShift Container Platform console. With this update, users with other cluster roles also can access the pipeline metrics.
- Before this update, role-based access control (RBAC) issues with the OpenShift Pipelines Operator caused problems upgrading or installing components. This update improves the reliability and consistency of installing various Red Hat OpenShift Pipelines components.
- Before this update, setting the **clusterTasks** and **pipelineTemplates** fields to **false** in the **TektonConfig** CR slowed the removal of cluster tasks and pipeline templates. This update improves the speed of lifecycle management of Tekton resources such as cluster tasks and pipeline templates.

1.12.8. Release notes for Red Hat OpenShift Pipelines General Availability 1.6.4

1.12.8.1. Known issues

- After upgrading from Red Hat OpenShift Pipelines 1.5.2 to 1.6.4, accessing the event listener routes returns a **503** error.
  Workaround: Modify the target port in the YAML file for the event listener's route.
  1. Extract the route name for the relevant namespace.

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

  2. Edit the route to modify the value of the **targetPort** field.

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

     Example: Existing event listener route

     ```
     ...
     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
...
```

Example: Modified event listener route

```
...
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
...
```

1.12.8.2. Fixed issues

- Before this update, the Operator failed when creating RBAC resources if any namespace was in a **Terminating** state. With this update, the Operator ignores namespaces in a **Terminating** state and creates the RBAC resources.

- Before this update, the task runs failed or restarted due to absence of annotation specifying the release version of the associated Tekton controller. With this update, the inclusion of the appropriate annotations are automated, and the tasks run without failure or restarts.

1.13. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.5

Red Hat OpenShift Pipelines General Availability (GA) 1.5 is now available on OpenShift Container Platform 4.8.

1.13.1. Compatibility and support matrix

Some features in this release are currently in Technology Preview. These experimental features are not intended for production use.

In the table, features are marked with the following statuses:

TP	Technology Preview
GA	General Availability

Note the following scope of support on the Red Hat Customer Portal for these features:

Table 1.4. Compatibility and support matrix

Feature	Version	Support Status
Pipelines	0.24	GA
CLI	0.19	GA
Catalog	0.24	GA
Triggers	0.14	TP
Pipeline resources	-	TP

For questions and feedback, you can send an email to the product team at pipelines-interest@redhat.com.

1.13.2. New features

In addition to the fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.5.

- Pipeline run and task runs will be automatically pruned by a cron job in the target namespace. The cron job uses the **IMAGE_JOB_PRUNER_TKN** environment variable to get the value of **tkn image**. With this enhancement, the following fields are introduced to the **TektonConfig** custom resource:

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

- In OpenShift Container Platform, you can customize the installation of the Tekton Add-ons component by modifying the values of the new parameters **clusterTasks** and **pipelinesTemplates** in the **TektonConfig** custom resource:

  ```
  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"
  ...
  ```

The customization is allowed if you create the add-on using **TektonConfig**, or directly by using Tekton Add-ons. However, if the parameters are not passed, the controller adds parameters with default values.

NOTE

- If add-on is created using the **TektonConfig** custom resource, and you change the parameter values later in the **Addon** custom resource, then the values in the **TektonConfig** custom resource overwrites the changes.

- You can set the value of the **pipelinesTemplates** parameter to **true** only when the value of the **clusterTasks** parameter is **true**.

- The **enableMetrics** parameter is added to the **TektonConfig** custom resource. You can use it to disable the service monitor, which is part of Tekton Pipelines for OpenShift Container Platform.

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

- Eventlistener OpenCensus metrics, which captures metrics at process level, is added.

- Triggers now has label selector; you can configure triggers for an event listener using labels.

- The **ClusterInterceptor** custom resource definition for registering interceptors is added, which allows you to register new **Interceptor** types that you can plug in. In addition, the following relevant changes are made:

  - In the trigger specifications, you can configure interceptors using a new API that includes a **ref** field to refer to a cluster interceptor. In addition, you can use the **params** field to add parameters that pass on to the interceptors for processing.

  - The bundled interceptors CEL, GitHub, GitLab, and BitBucket, have been migrated. They are implemented using the new **ClusterInterceptor** custom resource definition.

  - Core interceptors are migrated to the new format, and any new triggers created using the old syntax automatically switch to the new **ref** or **params** based syntax.

- To disable prefixing the name of the task or step while displaying logs, use the **--prefix** option for **log** commands.

- To display the version of a specific component, use the new **--component** flag in the **tkn version** command.

- The **tkn hub check-upgrade** command is added, and other commands are revised to be based on the pipeline version. In addition, catalog names are displayed in the **search** command output.

- Support for optional workspaces are added to the **start** command.

- If the plugins are not present in the **plugins** directory, they are searched in the current path.

- The **tkn start [task | clustertask | pipeline]** command starts interactively and ask for the **params** value, even when you specify the default parameters are specified. To stop the interactive prompts, pass the **--use-param-defaults** flag at the time of invoking the command. For example:

```
$ tkn pipeline start build-and-deploy \
    -w name=shared-
workspace,volumeClaimTemplateFile=https://raw.githubusercontent.com/openshift/pipelines-
tutorial/pipelines-1.15/01_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
```

- The **version** field is added in the **tkn task describe** command.

- The option to automatically select resources such as **TriggerTemplate**, or **TriggerBinding**, or **ClusterTriggerBinding**, or **Eventlistener**, is added in the **describe** command, if only one is present.

- In the **tkn pr describe** command, a section for skipped tasks is added.

- Support for the **tkn clustertask logs** is added.

- The YAML merge and variable from **config.yaml** is removed. In addition, the **release.yaml** file can now be more easily consumed by tools such as **kustomize** and **ytt**.

- The support for resource names to contain the dot character (".") is added.

- The **hostAliases** array in the **PodTemplate** specification is added to the pod-level override of hostname resolution. It is achieved by modifying the **/etc/hosts** file.

- A variable **$(tasks.status)** is introduced to access the aggregate execution status of tasks.

- An entry-point binary build for Windows is added.

1.13.3. Deprecated features

- In the **when** expressions, support for fields written is PascalCase is removed. The **when** expressions only support fields written in lowercase.

  NOTE

  If you had applied a pipeline with **when** expressions in Tekton Pipelines **v0.16** (Operator **v1.2.x**), you have to reapply it.

- When you upgrade the Red Hat OpenShift Pipelines Operator to **v1.5**, the **openshift-client** and the **openshift-client-v-1-5-0** cluster tasks have the **SCRIPT** parameter. However, the **ARGS** parameter and the **git** resource are removed from the specification of the **openshift-client**

cluster task. This is a breaking change, and only those cluster tasks that do not have a specific version in the **name** field of the **ClusterTask** resource upgrade seamlessly.
To prevent the pipeline runs from breaking, use the **SCRIPT** parameter after the upgrade because it moves the values previously specified in the **ARGS** parameter into the **SCRIPT** parameter of the cluster task. For example:

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

- When you upgrade from Red Hat OpenShift Pipelines Operator **v1.4** to **v1.5**, the profile names in which the **TektonConfig** custom resource is installed now change.

Table 1.5. Profiles forTektonConfig custom resource

Profiles in Pipelines 1.5	Corresponding profile in Pipelines 1.4	Installed Tekton components
All (*default profile*)	All (*default profile*)	Pipelines, Triggers, Add-ons
Basic	Default	Pipelines, Triggers
Lite	Basic	Pipelines

NOTE

If you used **profile: all** in the **config** instance of the **TektonConfig** custom resource, no change is necessary in the resource specification.

However, if the installed Operator is either in the Default or the Basic profile before the upgrade, you must edit the **config** instance of the **TektonConfig** custom resource after the upgrade. For example, if the configuration was **profile: basic** before the upgrade, ensure that it is **profile: lite** after upgrading to Pipelines 1.5.

- The **disable-home-env-overwrite** and **disable-working-dir-overwrite** fields are now deprecated and will be removed in a future release. For this release, the default value of these flags is set to **true** for backward compatibility.

NOTE

In the next release (Red Hat OpenShift Pipelines 1.6), the **HOME** environment variable will not be automatically set to **/tekton/home**, and the default working directory will not be set to **/workspace** for task runs. These defaults collide with any value set by image Dockerfile of the step.

- The **ServiceType** and **podTemplate** fields are removed from the **EventListener** spec.
- The controller service account no longer requests cluster-wide permission to list and watch namespaces.
- The status of the **EventListener** resource has a new condition called **Ready**.

  NOTE

  In the future, the other status conditions for the **EventListener** resource will be deprecated in favor of the **Ready** status condition.

- The **eventListener** and **namespace** fields in the **EventListener** response are deprecated. Use the **eventListenerUID** field instead.
- The **replicas** field is deprecated from the **EventListener** spec. Instead, the **spec.replicas** field is moved to **spec.resources.kubernetesResource.replicas** in the **KubernetesResource** spec.

  NOTE

  The **replicas** field will be removed in a future release.

- The old method of configuring the core interceptors is deprecated. However, it continues to work until it is removed in a future release. Instead, interceptors in a **Trigger** resource are now configured using a new **ref** and **params** based syntax. The resulting default webhook automatically switch the usages of the old syntax to the new syntax for new triggers.
- Use **rbac.authorization.k8s.io/v1** instead of the deprecated **rbac.authorization.k8s.io/v1beta1** for the **ClusterRoleBinding** resource.
- In cluster roles, the cluster-wide write access to resources such as **serviceaccounts**, **secrets**, **configmaps**, and **limitranges** are removed. In addition, cluster-wide access to resources such as **deployments**, **statefulsets**, and **deployment/finalizers** are removed.
- The **image** custom resource definition in the **caching.internal.knative.dev** group is not used by Tekton anymore, and is excluded in this release.

1.13.4. Known issues

- The git-cli cluster task is built off the alpine/git base image, which expects **/root** as the user's home directory. However, this is not explicitly set in the **git-cli** cluster task.
  In Tekton, the default home directory is overwritten with **/tekton/home** for every step of a task, unless otherwise specified. This overwriting of the **$HOME** environment variable of the base image causes the **git-cli** cluster task to fail.

  This issue is expected to be fixed in the upcoming releases. For Red Hat OpenShift Pipelines 1.5 and earlier versions, you can *use any one of the following workarounds* to avoid the failure of the **git-cli** cluster task:

  - Set the **$HOME** environment variable in the steps, so that it is not overwritten.

    1. [OPTIONAL] If you installed Red Hat OpenShift Pipelines using the Operator, then clone the **git-cli** cluster task into a separate task. This approach ensures that the Operator does not overwrite the changes made to the cluster task.

2. Execute the **oc edit clustertasks git-cli** command.

3. Add the expected **HOME** environment variable to the YAML of the step:

```
...
steps:
  - name: git
    env:
    - name: HOME
      value: /root
    image: $(params.BASE_IMAGE)
    workingDir: $(workspaces.source.path)
...
```

> **WARNING**
>
> For Red Hat OpenShift Pipelines installed by the Operator, if you do not clone the **git-cli** cluster task into a separate task before changing the **HOME** environment variable, then the changes are overwritten during Operator reconciliation.

- Disable overwriting the **HOME** environment variable in the **feature-flags** config map.

1. Execute the **oc edit -n openshift-pipelines configmap feature-flags** command.

2. Set the value of the **disable-home-env-overwrite** flag to **true**.

> **WARNING**
>
> - If you installed Red Hat OpenShift Pipelines using the Operator, then the changes are overwritten during Operator reconciliation.
> - Modifying the default value of the **disable-home-env-overwrite** flag can break other tasks and cluster tasks, as it changes the default behavior for all tasks.

- Use a different service account for the **git-cli** cluster task, as the overwriting of the **HOME** environment variable happens when the default service account for pipelines is used.

1. Create a new service account.

2. Link your Git secret to the service account you just created.

3. Use the service account while executing a task or a pipeline.

- On IBM Power Systems, IBM Z, and LinuxONE, the **s2i-dotnet** cluster task and the **tkn hub** command are unsupported.
- When you run Maven and Jib-Maven cluster tasks, the default container image is supported only on Intel (x86) architecture. Therefore, tasks will fail on IBM Power Systems (ppc64le), IBM Z, and LinuxONE (s390x) clusters. As a workaround, you can specify a custom image by setting the **MAVEN_IMAGE** parameter value to **maven:3.6.3-adoptopenjdk-11**.

1.13.5. Fixed issues

- The **when** expressions in **dag** tasks are not allowed to specify the context variable accessing the execution status (**$(tasks.<pipelineTask>.status)**) of any other task.
- Use Owner UIDs instead of Owner names, as it helps avoid race conditions created by deleting a **volumeClaimTemplate** PVC, in situations where a **PipelineRun** resource is quickly deleted and then recreated.
- A new Dockerfile is added for **pullrequest-init** for **build-base** image triggered by non-root users.
- When a pipeline or task is executed with the **-f** option and the **param** in its definition does not have a **type** defined, a validation error is generated instead of the pipeline or task run failing silently.
- For the **tkn start [task | pipeline | clustertask]** commands, the description of the **--workspace** flag is now consistent.
- While parsing the parameters, if an empty array is encountered, the corresponding interactive help is displayed as an empty string now.

1.14. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES GENERAL AVAILABILITY 1.4

Red Hat OpenShift Pipelines General Availability (GA) 1.4 is now available on OpenShift Container Platform 4.7.

NOTE

In addition to the stable and preview Operator channels, the Red Hat OpenShift Pipelines Operator 1.4.0 comes with the ocp-4.6, ocp-4.5, and ocp-4.4 deprecated channels. These deprecated channels and support for them will be removed in the following release of Red Hat OpenShift Pipelines.

1.14.1. Compatibility and support matrix

Some features in this release are currently in Technology Preview. These experimental features are not intended for production use.

In the table, features are marked with the following statuses:

TP	Technology Preview
GA	General Availability

Note the following scope of support on the Red Hat Customer Portal for these features:

Table 1.6. Compatibility and support matrix

Feature	Version	Support Status
Pipelines	0.22	GA
CLI	0.17	GA
Catalog	0.22	GA
Triggers	0.12	TP
Pipeline resources	-	TP

For questions and feedback, you can send an email to the product team at pipelines-interest@redhat.com.

1.14.2. New features

In addition to the fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.4.

- The custom tasks have the following enhancements:
  - Pipeline results can now refer to results produced by custom tasks.
  - Custom tasks can now use workspaces, service accounts, and pod templates to build more complex custom tasks.
- The **finally** task has the following enhancements:
  - The **when** expressions are supported in **finally** tasks, which provides efficient guarded execution and improved reusability of tasks.
  - A **finally** task can be configured to consume the results of any task within the same pipeline.

  NOTE

  Support for **when** expressions and **finally** tasks are unavailable in the OpenShift Container Platform 4.7 web console.

- Support for multiple secrets of the type **dockercfg** or **dockerconfigjson** is added for authentication at runtime.
- Functionality to support sparse-checkout with the **git-clone** task is added. This enables you to clone only a subset of the repository as your local copy, and helps you to restrict the size of the cloned repositories.
- You can create pipeline runs in a pending state without actually starting them. In clusters that are under heavy load, this allows Operators to have control over the start time of the pipeline runs.

- Ensure that you set the **SYSTEM_NAMESPACE** environment variable manually for the controller; this was previously set by default.

- A non-root user is now added to the build-base image of pipelines so that **git-init** can clone repositories as a non-root user.

- Support to validate dependencies between resolved resources before a pipeline run starts is added. All result variables in the pipeline must be valid, and optional workspaces from a pipeline can only be passed to tasks expecting it for the pipeline to start running.

- The controller and webhook runs as a non-root group, and their superfluous capabilities have been removed to make them more secure.

- You can use the **tkn pr logs** command to see the log streams for retried task runs.

- You can use the **--clustertask** option in the **tkn tr delete** command to delete all the task runs associated with a particular cluster task.

- Support for using Knative service with the **EventListener** resource is added by introducing a new **customResource** field.

- An error message is displayed when an event payload does not use the JSON format.

- The source control interceptors such as GitLab, BitBucket, and GitHub, now use the new **InterceptorRequest** or **InterceptorResponse** type interface.

- A new CEL function **marshalJSON** is implemented so that you can encode a JSON object or an array to a string.

- An HTTP handler for serving the CEL and the source control core interceptors is added. It packages four core interceptors into a single HTTP server that is deployed in the **tekton-pipelines** namespace. The **EventListener** object forwards events over the HTTP server to the interceptor. Each interceptor is available at a different path. For example, the CEL interceptor is available on the **/cel** path.

- The **pipelines-scc** Security Context Constraint (SCC) is used with the default **pipeline** service account for pipelines. This new service account is similar to **anyuid**, but with a minor difference as defined in the YAML for SCC of OpenShift Container Platform 4.7:

  ```
  fsGroup:
    type: MustRunAs
  ```

1.14.3. Deprecated features

- The **build-gcs** sub-type in the pipeline resource storage, and the **gcs-fetcher** image, are not supported.

- In the **taskRun** field of cluster tasks, the label **tekton.dev/task** is removed.

- For webhooks, the value **v1beta1** corresponding to the field **admissionReviewVersions** is removed.

- The **creds-init** helper image for building and deploying is removed.

- In the triggers spec and binding, the deprecated field **template.name** is removed in favor of **template.ref**. You should update all **eventListener** definitions to use the **ref** field.

> NOTE
>
> Upgrade from OpenShift Pipelines 1.3.x and earlier versions to OpenShift Pipelines 1.4.0 breaks event listeners because of the unavailability of the **template.name** field. For such cases, use OpenShift Pipelines 1.4.1 to avail the restored **template.name** field.

- For **EventListener** custom resources/objects, the fields **PodTemplate** and **ServiceType** are deprecated in favor of **Resource**.

- The deprecated spec style embedded bindings is removed.

- The **spec** field is removed from the **triggerSpecBinding**.

- The event ID representation is changed from a five-character random string to a UUID.

1.14.4. Known issues

- In the **Developer** perspective, the pipeline metrics and triggers features are available only on OpenShift Container Platform 4.7.6 or later versions.

- On IBM Power Systems, IBM Z, and LinuxONE, the **tkn hub** command is not supported.

- When you run Maven and Jib Maven cluster tasks on an IBM Power Systems (ppc64le), IBM Z, and LinuxONE (s390x) clusters, set the **MAVEN_IMAGE** parameter value to **maven:3.6.3-adoptopenjdk-11**.

- Triggers throw error resulting from bad handling of the JSON format, if you have the following configuration in the trigger binding:

  ```
  params:
    - name: github_json
      value: $(body)
  ```

  To resolve the issue:

  - If you are using triggers v0.11.0 and above, use the **marshalJSON** CEL function, which takes a JSON object or array and returns the JSON encoding of that object or array as a string.

  - If you are using older triggers version, add the following annotation in the trigger template:

    ```
    annotations:
      triggers.tekton.dev/old-escape-quotes: "true"
    ```

- When upgrading from OpenShift Pipelines 1.3.x to 1.4.x, you must recreate the routes.

1.14.5. Fixed issues

- Previously, the **tekton.dev/task** label was removed from the task runs of cluster tasks, and the **tekton.dev/clusterTask** label was introduced. The problems resulting from that change is resolved by fixing the **clustertask describe** and **delete** commands. In addition, the **lastrun** function for tasks is modified, to fix the issue of the **tekton.dev/task** label being applied to the task runs of both tasks and cluster tasks in older versions of pipelines.

- When doing an interactive **tkn pipeline start pipelinename**, a **PipelineResource** is created interactively. The **tkn p start** command prints the resource status if the resource status is not **nil**.

- Previously, the **tekton.dev/task=name** label was removed from the task runs created from cluster tasks. This fix modifies the **tkn clustertask start** command with the **--last** flag to check for the **tekton.dev/task=name** label in the created task runs.

- When a task uses an inline task specification, the corresponding task run now gets embedded in the pipeline when you run the **tkn pipeline describe** command, and the task name is returned as embedded.

- The **tkn version** command is fixed to display the version of the installed Tekton CLI tool, without a configured **kubeConfiguration namespace** or access to a cluster.

- If an argument is unexpected or more than one arguments are used, the **tkn completion** command gives an error.

- Previously, pipeline runs with the **finally** tasks nested in a pipeline specification would lose those **finally** tasks, when converted to the **v1alpha1** version and restored back to the **v1beta1** version. This error occurring during conversion is fixed to avoid potential data loss. Pipeline runs with the **finally** tasks nested in a pipeline specification is now serialized and stored on the alpha version, only to be deserialized later.

- Previously, there was an error in the pod generation when a service account had the **secrets** field as **{}**. The task runs failed with **CouldntGetTask** because the GET request with an empty secret name returned an error, indicating that the resource name may not be empty. This issue is fixed by avoiding an empty secret name in the **kubeclient** GET request.

- Pipelines with the **v1beta1** API versions can now be requested along with the **v1alpha1** version, without losing the **finally** tasks. Applying the returned **v1alpha1** version will store the resource as **v1beta1**, with the **finally** section restored to its original state.

- Previously, an unset **selfLink** field in the controller caused an error in the Kubernetes v1.20 clusters. As a temporary fix, the **CloudEvent** source field is set to a value that matches the current source URI, without the value of the auto-populated **selfLink** field.

- Previously, a secret name with dots such as **gcr.io** led to a task run creation failure. This happened because of the secret name being used internally as part of a volume mount name. The volume mount name conforms to the RFC1123 DNS label and disallows dots as part of the name. This issue is fixed by replacing the dot with a dash that results in a readable name.

- Context variables are now validated in the **finally** tasks.

- Previously, when the task run reconciler was passed a task run that did not have a previous status update containing the name of the pod it created, the task run reconciler listed the pods associated with the task run. The task run reconciler used the labels of the task run, which were propagated to the pod, to find the pod. Changing these labels while the task run was running, caused the code to not find the existing pod. As a result, duplicate pods were created. This issue is fixed by changing the task run reconciler to only use the **tekton.dev/taskRun** Tekton-controlled label when finding the pod.

- Previously, when a pipeline accepted an optional workspace and passed it to a pipeline task, the pipeline run reconciler stopped with an error if the workspace was not provided, even if a missing workspace binding is a valid state for an optional workspace. This issue is fixed by ensuring that the pipeline run reconciler does not fail to create a task run, even if an optional workspace is not provided.

- The sorted order of step statuses matches the order of step containers.
- Previously, the task run status was set to **unknown** when a pod encountered the **CreateContainerConfigError** reason, which meant that the task and the pipeline ran until the pod timed out. This issue is fixed by setting the task run status to **false**, so that the task is set as failed when the pod encounters the **CreateContainerConfigError** reason.
- Previously, pipeline results were resolved on the first reconciliation, after a pipeline run was completed. This could fail the resolution resulting in the **Succeeded** condition of the pipeline run being overwritten. As a result, the final status information was lost, potentially confusing any services watching the pipeline run conditions. This issue is fixed by moving the resolution of pipeline results to the end of a reconciliation, when the pipeline run is put into a **Succeeded** or **True** condition.
- Execution status variable is now validated. This avoids validating task results while validating context variables to access execution status.
- Previously, a pipeline result that contained an invalid variable would be added to the pipeline run with the literal expression of the variable intact. Therefore, it was difficult to assess whether the results were populated correctly. This issue is fixed by filtering out the pipeline run results that reference failed task runs. Now, a pipeline result that contains an invalid variable will not be emitted by the pipeline run at all.
- The **tkn eventlistener describe** command is fixed to avoid crashing without a template. It also displays the details about trigger references.
- Upgrades from OpenShift Pipelines 1.3.x and earlier versions to OpenShift Pipelines 1.4.0 breaks event listeners because of the unavailability of **template.name**. In OpenShift Pipelines 1.4.1, the **template.name** has been restored to avoid breaking event listeners in triggers.
- In OpenShift Pipelines 1.4.1, the **ConsoleQuickStart** custom resource has been updated to align with OpenShift Container Platform 4.7 capabilities and behavior.

1.15. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES TECHNOLOGY PREVIEW 1.3

1.15.1. New features

Red Hat OpenShift Pipelines Technology Preview (TP) 1.3 is now available on OpenShift Container Platform 4.7. Red Hat OpenShift Pipelines TP 1.3 is updated to support:

- Tekton Pipelines 0.19.0
- Tekton **tkn** CLI 0.15.0
- Tekton Triggers 0.10.2
- cluster tasks based on Tekton Catalog 0.19.0
- IBM Power Systems on OpenShift Container Platform 4.7
- IBM Z and LinuxONE on OpenShift Container Platform 4.7

In addition to the fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.3.

1.15.1.1. Pipelines

- Tasks that build images, such as S2I and Buildah tasks, now emit a URL of the image built that includes the image SHA.
- Conditions in pipeline tasks that reference custom tasks are disallowed because the **Condition** custom resource definition (CRD) has been deprecated.
- Variable expansion is now added in the **Task** CRD for the following fields: **spec.steps[].imagePullPolicy** and **spec.sidecar[].imagePullPolicy**.
- You can disable the built-in credential mechanism in Tekton by setting the **disable-creds-init** feature-flag to **true**.
- Resolved when expressions are now listed in the **Skipped Tasks** and the **Task Runs** sections in the **Status** field of the **PipelineRun** configuration.
- The **git init** command can now clone recursive submodules.
- A **Task** CR author can now specify a timeout for a step in the **Task** spec.
- You can now base the entry point image on the **distroless/static:nonroot** image and give it a mode to copy itself to the destination, without relying on the **cp** command being present in the base image.
- You can now use the configuration flag **require-git-ssh-secret-known-hosts** to disallow omitting known hosts in the Git SSH secret. When the flag value is set to **true**, you must include the **known_host** field in the Git SSH secret. The default value for the flag is **false**.
- The concept of optional workspaces is now introduced. A task or pipeline might declare a workspace optional and conditionally change their behavior based on its presence. A task run or pipeline run might also omit that workspace, thereby modifying the task or pipeline behavior. The default task run workspaces are not added in place of an omitted optional workspace.
- Credentials initialization in Tekton now detects an SSH credential that is used with a non-SSH URL, and vice versa in Git pipeline resources, and logs a warning in the step containers.
- The task run controller emits a warning event if the affinity specified by the pod template is overwritten by the affinity assistant.
- The task run reconciler now records metrics for cloud events that are emitted once a task run is completed. This includes retries.

1.15.1.2. Pipelines CLI

- Support for **--no-headers flag** is now added to the following commands: **tkn condition list**,**tkn triggerbinding list**,**tkn eventlistener list**,**tkn clustertask list**, **tkn clustertriggerbinding list**.
- When used together, the **--last** or **--use** options override the **--prefix-name** and **--timeout** options.
- The **tkn eventlistener logs** command is now added to view the **EventListener** logs.
- The **tekton hub** commands are now integrated into the **tkn** CLI.
- The **--nocolour** option is now changed to **--no-color**.

- The **--all-namespaces** flag is added to the following commands: **tkn triggertemplate list**, **tkn condition list**, **tkn triggerbinding list**, **tkn eventlistener list**.

1.15.1.3. Triggers

- You can now specify your resource information in the **EventListener** template.
- It is now mandatory for **EventListener** service accounts to have the **list** and **watch** verbs, in addition to the **get** verb for all the triggers resources. This enables you to use **Listers** to fetch data from **EventListener**, **Trigger**, **TriggerBinding**, **TriggerTemplate**, and **ClusterTriggerBinding** resources. You can use this feature to create a **Sink** object rather than specifying multiple informers, and directly make calls to the API server.
- A new **Interceptor** interface is added to support immutable input event bodies. Interceptors can now add data or fields to a new **extensions** field, and cannot modify the input bodies making them immutable. The CEL interceptor uses this new **Interceptor** interface.
- A **namespaceSelector** field is added to the **EventListener** resource. Use it to specify the namespaces from where the **EventListener** resource can fetch the **Trigger** object for processing events. To use the **namespaceSelector** field, the service account for the **EventListener** resource must have a cluster role.
- The triggers **EventListener** resource now supports end-to-end secure connection to the **eventlistener** pod.
- The escaping parameters behavior in the **TriggerTemplates** resource by replacing **"** with **\"** is now removed.
- A new **resources** field, supporting Kubernetes resources, is introduced as part of the **EventListener** spec.
- A new functionality for the CEL interceptor, with support for upper and lower-casing of ASCII strings, is added.
- You can embed **TriggerBinding** resources by using the **name** and **value** fields in a trigger, or an event listener.
- The **PodSecurityPolicy** configuration is updated to run in restricted environments. It ensures that containers must run as non-root. In addition, the role-based access control for using the pod security policy is moved from cluster-scoped to namespace-scoped. This ensures that the triggers cannot use other pod security policies that are unrelated to a namespace.
- Support for embedded trigger templates is now added. You can either use the **name** field to refer to an embedded template or embed the template inside the **spec** field.

1.15.2. Deprecated features

- Pipeline templates that use **PipelineResources** CRDs are now deprecated and will be removed in a future release.
- The **template.name** field is deprecated in favor of the **template.ref** field and will be removed in a future release.
- The **-c** shorthand for the **--check** command has been removed. In addition, global **tkn** flags are added to the **version** command.

1.15.3. Known issues

- CEL overlays add fields to a new top-level **extensions** function, instead of modifying the incoming event body. **TriggerBinding** resources can access values within this new **extensions** function using the **$(extensions.<key>)** syntax. Update your binding to use the **$(extensions.<key>)** syntax instead of the **$(body.<overlay-key>)** syntax.

- The escaping parameters behavior by replacing **"** with **\"** is now removed. If you need to retain the old escaping parameters behavior add the **tekton.dev/old-escape-quotes: true"** annotation to your **TriggerTemplate** specification.

- You can embed **TriggerBinding** resources by using the **name** and **value** fields inside a trigger or an event listener. However, you cannot specify both **name** and **ref** fields for a single binding. Use the **ref** field to refer to a **TriggerBinding** resource and the **name** field for embedded bindings.

- An interceptor cannot attempt to reference a **secret** outside the namespace of an **EventListener** resource. You must include secrets in the namespace of the `EventListener`resource.

- In Triggers 0.9.0 and later, if a body or header based **TriggerBinding** parameter is missing or malformed in an event payload, the default values are used instead of displaying an error.

- Tasks and pipelines created with **WhenExpression** objects using Tekton Pipelines 0.16.x must be reapplied to fix their JSON annotations.

- When a pipeline accepts an optional workspace and gives it to a task, the pipeline run stalls if the workspace is not provided.

- To use the Buildah cluster task in a disconnected environment, ensure that the Dockerfile uses an internal image stream as the base image, and then use it in the same manner as any S2I cluster task.

1.15.4. Fixed issues

- Extensions added by a CEL Interceptor are passed on to webhook interceptors by adding the **Extensions** field within the event body.

- The activity timeout for log readers is now configurable using the **LogOptions** field. However, the default behavior of timeout in 10 seconds is retained.

- The **log** command ignores the **--follow** flag when a task run or pipeline run is complete, and reads available logs instead of live logs.

- References to the following Tekton resources: **EventListener**, **TriggerBinding**, **ClusterTriggerBinding**, **Condition**, and **TriggerTemplate** are now standardized and made consistent across all user-facing messages in **tkn** commands.

- Previously, if you started a canceled task run or pipeline run with the **--use-taskrun <canceled-task-run-name>**, **--use-pipelinerun <canceled-pipeline-run-name>** or **--last** flags, the new run would be canceled. This bug is now fixed.

- The **tkn pr desc** command is now enhanced to ensure that it does not fail in case of pipeline runs with conditions.

- When you delete a task run using the **tkn tr delete** command with the **--task** option, and a cluster task exists with the same name, the task runs for the cluster task also get deleted. As a workaround, filter the task runs by using the **TaskRefKind** field.

- The **tkn triggertemplate describe** command would display only part of the **apiVersion** value in the output. For example, only **triggers.tekton.dev** was displayed instead of **triggers.tekton.dev/v1alpha1**. This bug is now fixed.

- The webhook, under certain conditions, would fail to acquire a lease and not function correctly. This bug is now fixed.

- Pipelines with when expressions created in v0.16.3 can now be run in v0.17.1 and later. After an upgrade, you do not need to reapply pipeline definitions created in previous versions because both the uppercase and lowercase first letters for the annotations are now supported.

- By default, the **leader-election-ha** field is now enabled for high availability. When the **disable-ha** controller flag is set to **true**, it disables high availability support.

- Issues with duplicate cloud events are now fixed. Cloud events are now sent only when a condition changes the state, reason, or message.

- When a service account name is missing from a **PipelineRun** or **TaskRun** spec, the controller uses the service account name from the **config-defaults** config map. If the service account name is also missing in the **config-defaults** config map, the controller now sets it to **default** in the spec.

- Validation for compatibility with the affinity assistant is now supported when the same persistent volume claim is used for multiple workspaces, but with different subpaths.

1.16. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES TECHNOLOGY PREVIEW 1.2

1.16.1. New features

Red Hat OpenShift Pipelines Technology Preview (TP) 1.2 is now available on OpenShift Container Platform 4.6. Red Hat OpenShift Pipelines TP 1.2 is updated to support:

- Tekton Pipelines 0.16.3

- Tekton **tkn** CLI 0.13.1

- Tekton Triggers 0.8.1

- cluster tasks based on Tekton Catalog 0.16

- IBM Power Systems on OpenShift Container Platform 4.6

- IBM Z and LinuxONE on OpenShift Container Platform 4.6

In addition to the fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.2.

1.16.1.1. Pipelines

- This release of Red Hat OpenShift Pipelines adds support for a disconnected installation.

NOTE

Installations in restricted environments are currently not supported on IBM Power Systems, IBM Z, and LinuxONE.

- You can now use the **when** field, instead of **conditions** resource, to run a task only when certain criteria are met. The key components of **WhenExpression** resources are **Input**, **Operator**, and **Values**. If all the when expressions evaluate to **True**, then the task is run. If any of the when expressions evaluate to **False**, the task is skipped.
- Step statuses are now updated if a task run is canceled or times out.
- Support for Git Large File Storage (LFS) is now available to build the base image used by **git-init**.
- You can now use the **taskSpec** field to specify metadata, such as labels and annotations, when a task is embedded in a pipeline.
- Cloud events are now supported by pipeline runs. Retries with **backoff** are now enabled for cloud events sent by the cloud event pipeline resource.
- You can now set a default **Workspace** configuration for any workspace that a **Task** resource declares, but that a **TaskRun** resource does not explicitly provide.
- Support is available for namespace variable interpolation for the **PipelineRun** namespace and **TaskRun** namespace.
- Validation for **TaskRun** objects is now added to check that not more than one persistent volume claim workspace is used when a **TaskRun** resource is associated with an Affinity Assistant. If more than one persistent volume claim workspace is used, the task run fails with a **TaskRunValidationFailed** condition. Note that by default, the Affinity Assistant is disabled in Red Hat OpenShift Pipelines, so you will need to enable the assistant to use it.

1.16.1.2. Pipelines CLI

- The **tkn task describe**, **tkn taskrun describe**, **tkn clustertask describe**, **tkn pipeline describe**, and **tkn pipelinerun describe** commands now:
  - Automatically select the **Task**, **TaskRun**, **ClusterTask**, **Pipeline** and **PipelineRun** resource, respectively, if only one of them is present.
  - Display the results of the **Task**, **TaskRun**, **ClusterTask**, **Pipeline** and **PipelineRun** resource in their outputs, respectively.
  - Display workspaces declared in the **Task**, **TaskRun**, **ClusterTask**, **Pipeline** and **PipelineRun** resource in their outputs, respectively.
- You can now use the **--prefix-name** option with the **tkn clustertask start** command to specify a prefix for the name of a task run.
- Interactive mode support has now been provided to the **tkn clustertask start** command.
- You can now specify **PodTemplate** properties supported by pipelines using local or remote file definitions for **TaskRun** and **PipelineRun** objects.
- You can now use the **--use-params-defaults** option with the **tkn clustertask start** command to use the default values set in the **ClusterTask** configuration and create the task run.

- The **--use-param-defaults** flag for the **tkn pipeline start** command now prompts the interactive mode if the default values have not been specified for some of the parameters.

1.16.1.3. Triggers

- The Common Expression Language (CEL) function named **parseYAML** has been added to parse a YAML string into a map of strings.
- Error messages for parsing CEL expressions have been improved to make them more granular while evaluating expressions and when parsing the hook body for creating the evaluation environment.
- Support is now available for marshaling boolean values and maps if they are used as the values of expressions in a CEL overlay mechanism.
- The following fields have been added to the **EventListener** object:
  - The **replicas** field enables the event listener to run more than one pod by specifying the number of replicas in the YAML file.
  - The **NodeSelector** field enables the **EventListener** object to schedule the event listener pod to a specific node.
- Webhook interceptors can now parse the **EventListener-Request-URL** header to extract parameters from the original request URL being handled by the event listener.
- Annotations from the event listener can now be propagated to the deployment, services, and other pods. Note that custom annotations on services or deployment are overwritten, and hence, must be added to the event listener annotations so that they are propagated.
- Proper validation for replicas in the **EventListener** specification is now available for cases when a user specifies the **spec.replicas** values as **negative** or **zero**.
- You can now specify the **TriggerCRD** object inside the **EventListener** spec as a reference using the **TriggerRef** field to create the **TriggerCRD** object separately and then bind it inside the **EventListener** spec.
- Validation and defaults for the **TriggerCRD** object are now available.

1.16.2. Deprecated features

- **$(params)** parameters are now removed from the **triggertemplate** resource and replaced by **$(tt.params)** to avoid confusion between the **resourcetemplate** and **triggertemplate** resource parameters.
- The **ServiceAccount** reference of the optional **EventListenerTrigger**-based authentication level has changed from an object reference to a **ServiceAccountName** string. This ensures that the **ServiceAccount** reference is in the same namespace as the **EventListenerTrigger** object.
- The **Conditions** custom resource definition (CRD) is now deprecated; use the **WhenExpressions** CRD instead.
- The **PipelineRun.Spec.ServiceAccountNames** object is being deprecated and replaced by the **PipelineRun.Spec.TaskRunSpec[].ServiceAccountName** object.

1.16.3. Known issues

- This release of Red Hat OpenShift Pipelines adds support for a disconnected installation. However, some images used by the cluster tasks must be mirrored for them to work in disconnected clusters.

- Pipelines in the **openshift** namespace are not deleted after you uninstall the Red Hat OpenShift Pipelines Operator. Use the **oc delete pipelines -n openshift --all** command to delete the pipelines.

- Uninstalling the Red Hat OpenShift Pipelines Operator does not remove the event listeners. As a workaround, to remove the **EventListener** and **Pod** CRDs:

  1. Edit the **EventListener** object with the **foregroundDeletion** finalizers:

     ```
     $ oc patch el/<eventlistener_name> -p '{"metadata":{"finalizers":["foregroundDeletion"]}}'
     --type=merge
     ```

     For example:

     ```
     $ oc patch el/github-listener-interceptor -p '{"metadata":{"finalizers":
     ["foregroundDeletion"]}}' --type=merge
     ```

  2. Delete the **EventListener** CRD:

     ```
     $ oc patch crd/eventlisteners.triggers.tekton.dev -p '{"metadata":{"finalizers":[]}}' --
     type=merge
     ```

- When you run a multi-arch container image task without command specification on an IBM Power Systems (ppc64le) or IBM Z (s390x) cluster, the **TaskRun** resource fails with the following error:

  ```
  Error executing command: fork/exec /bin/bash: exec format error
  ```

  As a workaround, use an architecture specific container image or specify the sha256 digest to point to the correct architecture. To get the sha256 digest enter:

  ```
  $ skopeo inspect --raw <image_name>| jq '.manifests[] | select(.platform.architecture == "
  <architecture>") | .digest'
  ```

1.16.4. Fixed issues

- A simple syntax validation to check the CEL filter, overlays in the Webhook validator, and the expressions in the interceptor has now been added.

- Triggers no longer overwrite annotations set on the underlying deployment and service objects.

- Previously, an event listener would stop accepting events. This fix adds an idle timeout of 120 seconds for the **EventListener** sink to resolve this issue.

- Previously, canceling a pipeline run with a **Failed(Canceled)** state gave a success message. This has been fixed to display an error instead.

- The **tkn eventlistener list** command now provides the status of the listed event listeners, thus enabling you to easily identify the available ones.

- Consistent error messages are now displayed for the **triggers list** and **triggers describe** commands when triggers are not installed or when a resource cannot be found.
- Previously, a large number of idle connections would build up during cloud event delivery. The **DisableKeepAlives: true** parameter was added to the **cloudeventclient** config to fix this issue. Thus, a new connection is set up for every cloud event.
- Previously, the **creds-init** code would write empty files to the disk even if credentials of a given type were not provided. This fix modifies the **creds-init** code to write files for only those credentials that have actually been mounted from correctly annotated secrets.

1.17. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES TECHNOLOGY PREVIEW 1.1

1.17.1. New features

Red Hat OpenShift Pipelines Technology Preview (TP) 1.1 is now available on OpenShift Container Platform 4.5. Red Hat OpenShift Pipelines TP 1.1 is updated to support:

- Tekton Pipelines 0.14.3
- Tekton **tkn** CLI 0.11.0
- Tekton Triggers 0.6.1
- cluster tasks based on Tekton Catalog 0.14

In addition to the fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.1.

1.17.1.1. Pipelines

- Workspaces can now be used instead of pipeline resources. It is recommended that you use workspaces in OpenShift Pipelines, as pipeline resources are difficult to debug, limited in scope, and make tasks less reusable. For more details on workspaces, see the Understanding OpenShift Pipelines section.
- Workspace support for volume claim templates has been added:
  - The volume claim template for a pipeline run and task run can now be added as a volume source for workspaces. The tekton-controller then creates a persistent volume claim (PVC) using the template that is seen as a PVC for all task runs in the pipeline. Thus you do not need to define the PVC configuration every time it binds a workspace that spans multiple tasks.
  - Support to find the name of the PVC when a volume claim template is used as a volume source is now available using variable substitution.
- Support for improving audits:
  - The **PipelineRun.Status** field now contains the status of every task run in the pipeline and the pipeline specification used to instantiate a pipeline run to monitor the progress of the pipeline run.
  - Pipeline results have been added to the pipeline specification and **PipelineRun** status.

  - The **TaskRun.Status** field now contains the exact task specification used to instantiate the **TaskRun** resource.

- Support to apply the default parameter to conditions.
- A task run created by referencing a cluster task now adds the **tekton.dev/clusterTask** label instead of the **tekton.dev/task** label.
- The kube config writer now adds the **ClientKeyData** and the **ClientCertificateData** configurations in the resource structure to enable replacement of the pipeline resource type cluster with the kubeconfig-creator task.
- The names of the **feature-flags** and the **config-defaults** config maps are now customizable.
- Support for the host network in the pod template used by the task run is now available.
- An Affinity Assistant is now available to support node affinity in task runs that share workspace volume. By default, this is disabled on OpenShift Pipelines.
- The pod template has been updated to specify **imagePullSecrets** to identify secrets that the container runtime should use to authorize container image pulls when starting a pod.
- Support for emitting warning events from the task run controller if the controller fails to update the task run.
- Standard or recommended k8s labels have been added to all resources to identify resources belonging to an application or component.
- The **Entrypoint** process is now notified for signals and these signals are then propagated using a dedicated PID Group of the **Entrypoint** process.
- The pod template can now be set on a task level at runtime using task run specs.
- Support for emitting Kubernetes events:
  - The controller now emits events for additional task run lifecycle events - **taskrun started** and **taskrun running**.
  - The pipeline run controller now emits an event every time a pipeline starts.
- In addition to the default Kubernetes events, support for cloud events for task runs is now available. The controller can be configured to send any task run events, such as create, started, and failed, as cloud events.
- Support for using the **$context.<task|taskRun|pipeline|pipelineRun>.name** variable to reference the appropriate name when in pipeline runs and task runs.
- Validation for pipeline run parameters is now available to ensure that all the parameters required by the pipeline are provided by the pipeline run. This also allows pipeline runs to provide extra parameters in addition to the required parameters.
- You can now specify tasks within a pipeline that will always execute before the pipeline exits, either after finishing all tasks successfully or after a task in the pipeline failed, using the **finally** field in the pipeline YAML file.
- The **git-clone** cluster task is now available.

1.17.1.2. Pipelines CLI

- Support for embedded trigger binding is now available to the **tkn evenlistener describe** command.
- Support to recommend subcommands and make suggestions if an incorrect subcommand is used.
- The **tkn task describe** command now auto selects the task if only one task is present in the pipeline.
- You can now start a task using default parameter values by specifying the **--use-param-defaults** flag in the **tkn task start** command.
- You can now specify a volume claim template for pipeline runs or task runs using the **--workspace** option with the **tkn pipeline start** or **tkn task start** commands.
- The **tkn pipelinerun logs** command now displays logs for the final tasks listed in the **finally** section.
- Interactive mode support has now been provided to the **tkn task start** command and the **describe** subcommand for the following **tkn** resources: **pipeline**, **pipelinerun**, **task**, **taskrun**, **clustertask**, and **pipelineresource**.
- The **tkn version** command now displays the version of the triggers installed in the cluster.
- The **tkn pipeline describe** command now displays parameter values and timeouts specified for tasks used in the pipeline.
- Support added for the **--last** option for the **tkn pipelinerun describe** and the **tkn taskrun describe** commands to describe the most recent pipeline run or task run, respectively.
- The **tkn pipeline describe** command now displays the conditions applicable to the tasks in the pipeline.
- You can now use the **--no-headers** and **--all-namespaces** flags with the **tkn resource list** command.

1.17.1.3. Triggers

- The following Common Expression Language (CEL) functions are now available:
  - **parseURL** to parse and extract portions of a URL
  - **parseJSON** to parse JSON value types embedded in a string in the **payload** field of the **deployment** webhook
- A new interceptor for webhooks from Bitbucket has been added.
- Event listeners now display the **Address URL** and the **Available status** as additional fields when listed with the **kubectl get** command.
- trigger template params now use the **$(tt.params.<paramName>)** syntax instead of **$(params.<paramName>)** to reduce the confusion between trigger template and resource templates params.

- You can now add **tolerations** in the **EventListener** CRD to ensure that event listeners are deployed with the same configuration even if all nodes are tainted due to security or management issues.

- You can now add a Readiness Probe for event listener Deployment at **URL/live**.

- Support for embedding **TriggerBinding** specifications in event listener triggers is now added.

- Trigger resources are now annotated with the recommended **app.kubernetes.io** labels.

1.17.2. Deprecated features

The following items are deprecated in this release:

- The **--namespace** or **-n** flags for all cluster-wide commands, including the **clustertask** and **clustertriggerbinding** commands, are deprecated. It will be removed in a future release.

- The **name** field in **triggers.bindings** within an event listener has been deprecated in favor of the **ref** field and will be removed in a future release.

- Variable interpolation in trigger templates using **$(params)** has been deprecated in favor of using **$(tt.params)** to reduce confusion with the pipeline variable interpolation syntax. The **$(params.<paramName>)** syntax will be removed in a future release.

- The **tekton.dev/task** label is deprecated on cluster tasks.

- The **TaskRun.Status.ResourceResults.ResourceRef** field is deprecated and will be removed.

- The **tkn pipeline create**, **tkn task create**, and **tkn resource create -f** subcommands have been removed.

- Namespace validation has been removed from **tkn** commands.

- The default timeout of **1h** and the **-t** flag for the **tkn ct start** command have been removed.

- The **s2i** cluster task has been deprecated.

1.17.3. Known issues

- Conditions do not support workspaces.

- The **--workspace** option and the interactive mode is not supported for the **tkn clustertask start** command.

- Support of backward compatibility for **$(params.<paramName>)** syntax forces you to use trigger templates with pipeline specific params as the trigger s webhook is unable to differentiate trigger params from pipelines params.

- Pipeline metrics report incorrect values when you run a promQL query for **tekton_taskrun_count** and **tekton_taskrun_duration_seconds_count**.

- pipeline runs and task runs continue to be in the **Running** and **Running(Pending)** states respectively even when a non existing PVC name is given to a workspace.

1.17.4. Fixed issues

- Previously, the **tkn task delete <name> --trs** command would delete both the task and cluster task if the name of the task and cluster task were the same. With this fix, the command deletes only the task runs that are created by the task **<name>**.
- Previously the **tkn pr delete -p <name> --keep 2** command would disregard the **-p** flag when used with the **--keep** flag and would delete all the pipeline runs except the latest two. With this fix, the command deletes only the pipeline runs that are created by the pipeline **<name>**, except for the latest two.
- The **tkn triggertemplate describe** output now displays resource templates in a table format instead of YAML format.
- Previously the **buildah** cluster task failed when a new user was added to a container. With this fix, the issue has been resolved.

1.18. RELEASE NOTES FOR RED HAT OPENSHIFT PIPELINES TECHNOLOGY PREVIEW 1.0

1.18.1. New features

Red Hat OpenShift Pipelines Technology Preview (TP) 1.0 is now available on OpenShift Container Platform 4.4. Red Hat OpenShift Pipelines TP 1.0 is updated to support:

- Tekton Pipelines 0.11.3
- Tekton **tkn** CLI 0.9.0
- Tekton Triggers 0.4.0
- cluster tasks based on Tekton Catalog 0.11

In addition to the fixes and stability improvements, the following sections highlight what is new in Red Hat OpenShift Pipelines 1.0.

1.18.1.1. Pipelines

- Support for v1beta1 API Version.
- Support for an improved limit range. Previously, limit range was specified exclusively for the task run and the pipeline run. Now there is no need to explicitly specify the limit range. The minimum limit range across the namespace is used.
- Support for sharing data between tasks using task results and task params.
- Pipelines can now be configured to not overwrite the **HOME** environment variable and the working directory of steps.
- Similar to task steps, **sidecars** now support script mode.
- You can now specify a different scheduler name in task run **podTemplate** resource.
- Support for variable substitution using Star Array Notation.
- Tekton controller can now be configured to monitor an individual namespace.

- A new description field is now added to the specification of pipelines, tasks, cluster tasks, resources, and conditions.

- Addition of proxy parameters to Git pipeline resources.

1.18.1.2. Pipelines CLI

- The **describe** subcommand is now added for the following **tkn** resources: **EventListener**, **Condition**, **TriggerTemplate**, **ClusterTask**, and **TriggerSBinding**.

- Support added for **v1beta1** to the following resources along with backward compatibility for **v1alpha1**: **ClusterTask**, **Task**, **Pipeline**, **PipelineRun**, and **TaskRun**.

- The following commands can now list output from all namespaces using the **--all-namespaces** flag option: **tkn task list**, **tkn pipeline list**, **tkn taskrun list**, **tkn pipelinerun list**
  The output of these commands is also enhanced to display information without headers using the **--no-headers** flag option.

- You can now start a pipeline using default parameter values by specifying **--use-param-defaults** flag in the **tkn pipelines start** command.

- Support for workspace is now added to **tkn pipeline start** and **tkn task start** commands.

- A new **clustertriggerbinding** command is now added with the following subcommands: **describe**, **delete**, and **list**.

- You can now directly start a pipeline run using a local or remote **yaml** file.

- The **describe** subcommand now displays an enhanced and detailed output. With the addition of new fields, such as **description**, **timeout**, **param description**, and **sidecar status**, the command output now provides more detailed information about a specific **tkn** resource.

- The **tkn task log** command now displays logs directly if only one task is present in the namespace.

1.18.1.3. Triggers

- Triggers can now create both **v1alpha1** and **v1beta1** pipeline resources.

- Support for new Common Expression Language (CEL) interceptor function - **compareSecret**. This function securely compares strings to secrets in CEL expressions.

- Support for authentication and authorization at the event listener trigger level.

1.18.2. Deprecated features

The following items are deprecated in this release:

- The environment variable **$HOME**, and variable **workingDir** in the **Steps** specification are deprecated and might be changed in a future release. Currently in a **Step** container, the **HOME** and **workingDir** variables are overwritten to **/tekton/home** and **/workspace** variables, respectively.
  In a later release, these two fields will not be modified, and will be set to values defined in the container image and the **Task** YAML. For this release, use the **disable-home-env-overwrite** and **disable-working-directory-overwrite** flags to disable overwriting of the **HOME** and **workingDir** variables.

- The following commands are deprecated and might be removed in the future release: **tkn pipeline create**, **tkn task create**.

- The **-f** flag with the **tkn resource create** command is now deprecated. It might be removed in the future release.

- The **-t** flag and the **--timeout** flag (with seconds format) for the **tkn clustertask create** command are now deprecated. Only duration timeout format is now supported, for example **1h30s**. These deprecated flags might be removed in the future release.

1.18.3. Known issues

- If you are upgrading from an older version of Red Hat OpenShift Pipelines, you must delete your existing deployments before upgrading to Red Hat OpenShift Pipelines version 1.0. To delete an existing deployment, you must first delete Custom Resources and then uninstall the Red Hat OpenShift Pipelines Operator. For more details, see the uninstalling Red Hat OpenShift Pipelines section.

- Submitting the same **v1alpha1** tasks more than once results in an error. Use the **oc replace** command instead of **oc apply** when re-submitting a **v1alpha1** task.

- The **buildah** cluster task does not work when a new user is added to a container.
  When the Operator is installed, the **--storage-driver** flag for the **buildah** cluster task is not specified, therefore the flag is set to its default value. In some cases, this causes the storage driver to be set incorrectly. When a new user is added, the incorrect storage-driver results in the failure of the **buildah** cluster task with the following error:

  ```
  useradd: /etc/passwd.8: lock file already used
  useradd: cannot lock /etc/passwd; try again later.
  ```

  As a workaround, manually set the **--storage-driver** flag value to **overlay** in the **buildah-task.yaml** file:

  1. Login to your cluster as a **cluster-admin**:

     ```
     $ oc login -u <login> -p <password> https://openshift.example.com:6443
     ```

  2. Use the **oc edit** command to edit **buildah** cluster task:

     ```
     $ oc edit clustertask buildah
     ```

     The current version of the **buildah** clustertask YAML file opens in the editor set by your **EDITOR** environment variable.

  3. Under the **Steps** field, locate the following **command** field:

     ```
      command: ['buildah', 'bud', '--format=$(params.FORMAT)', '--tls-verify=$(params.TLSVERIFY)', '--layers', '-f', '$(params.DOCKERFILE)', '-t', '$(resources.outputs.image.url)', '$(params.CONTEXT)']
     ```

  4. Replace the **command** field with the following:

```
 command: ['buildah', '--storage-driver=overlay', 'bud', '--format=$(params.FORMAT)', '--
tls-verify=$(params.TLSVERIFY)', '--no-cache', '-f', '$(params.DOCKERFILE)', '-t',
'$(params.IMAGE)', '$(params.CONTEXT)']
```

5. Save the file and exit.

Alternatively, you can also modify the **buildah** cluster task YAML file directly on the web console by navigating to **Pipelines** → **Cluster Tasks** → **buildah**. Select **Edit Cluster Task** from the **Actions** menu and replace the **command** field as shown in the previous procedure.

1.18.4. Fixed issues

- Previously, the **DeploymentConfig** task triggered a new deployment build even when an image build was already in progress. This caused the deployment of the pipeline to fail. With this fix, the **deploy task** command is now replaced with the **oc rollout status** command which waits for the in-progress deployment to finish.

- Support for **APP_NAME** parameter is now added in pipeline templates.

- Previously, the pipeline template for Java S2I failed to look up the image in the registry. With this fix, the image is looked up using the existing image pipeline resources instead of the user provided **IMAGE_NAME** parameter.

- All the OpenShift Pipelines images are now based on the Red Hat Universal Base Images (UBI).

- Previously, when the pipeline was installed in a namespace other than **tekton-pipelines**, the **tkn version** command displayed the pipeline version as **unknown**. With this fix, the **tkn version** command now displays the correct pipeline version in any namespace.

- The **-c** flag is no longer supported for the **tkn version** command.

- Non-admin users can now list the cluster trigger bindings.

- The event listener **CompareSecret** function is now fixed for the CEL Interceptor.

- The **list**, **describe**, and **start** subcommands for tasks and cluster tasks now correctly display the output in case a task and cluster task have the same name.

- Previously, the OpenShift Pipelines Operator modified the privileged security context constraints (SCCs), which caused an error during cluster upgrade. This error is now fixed.

- In the **tekton-pipelines** namespace, the timeouts of all task runs and pipeline runs are now set to the value of **default-timeout-minutes** field using the config map.

- Previously, the pipelines section in the web console was not displayed for non-admin users. This issue is now resolved.

CHAPTER 2. ABOUT RED HAT OPENSHIFT PIPELINES

Red Hat OpenShift Pipelines is a cloud-native, continuous integration and continuous delivery (CI/CD) solution based on Kubernetes resources. It uses Tekton building blocks to automate deployments across multiple platforms by abstracting away the underlying implementation details. Tekton introduces a number of standard custom resource definitions (CRDs) for defining CI/CD pipelines that are portable across Kubernetes distributions.

NOTE

Because Red Hat OpenShift Pipelines releases on a different cadence from OpenShift Container Platform, the Red Hat OpenShift Pipelines documentation is now available as separate documentation sets for each minor version of the product.

The Red Hat OpenShift Pipelines documentation is available at https://docs.openshift.com/pipelines/.

Documentation for specific versions is available using the version selector drop-down list, or directly by adding the version to the URL, for example, https://docs.openshift.com/pipelines/1.15.

In addition, the Red Hat OpenShift Pipelines documentation is also available on the Red Hat Customer Portal at https://access.redhat.com/documentation/en-us/red_hat_openshift_pipelines/.

For additional information about the Red Hat OpenShift Pipelines life cycle and supported platforms, refer to the Platform Life Cycle Policy .

CHAPTER 3. UNDERSTANDING OPENSHIFT PIPELINES

Red Hat OpenShift Pipelines is a cloud-native, continuous integration and continuous delivery (CI/CD) solution based on Kubernetes resources. It uses Tekton building blocks to automate deployments across multiple platforms by abstracting away the underlying implementation details. Tekton introduces a number of standard custom resource definitions (CRDs) for defining CI/CD pipelines that are portable across Kubernetes distributions.

3.1. KEY FEATURES

- Red Hat OpenShift Pipelines is a serverless CI/CD system that runs pipelines with all the required dependencies in isolated containers.
- Red Hat OpenShift Pipelines are designed for decentralized teams that work on microservice-based architecture.
- Red Hat OpenShift Pipelines use standard CI/CD pipeline definitions that are easy to extend and integrate with the existing Kubernetes tools, enabling you to scale on-demand.
- You can use Red Hat OpenShift Pipelines to build images with Kubernetes tools such as Source-to-Image (S2I), Buildah, Buildpacks, and Kaniko that are portable across any Kubernetes platform.
- You can use the OpenShift Container Platform Developer console to create Tekton resources, view logs of pipeline runs, and manage pipelines in your OpenShift Container Platform namespaces.

3.2. OPENSHIFT PIPELINES CONCEPTS

This guide provides a detailed view of the various pipeline concepts.

3.2.1. Tasks

Task resources are the building blocks of a pipeline and consist of sequentially executed steps. It is essentially a function of inputs and outputs. A task can run individually or as a part of the pipeline. Tasks are reusable and can be used in multiple pipelines.

Steps are a series of commands that are sequentially executed by the task and achieve a specific goal, such as building an image. Every task runs as a pod, and each step runs as a container within that pod. Because steps run within the same pod, they can access the same volumes for caching files, config maps, and secrets.

The following example shows the **apply-manifests** task.

```
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. The task API version, **v1**.
2. The type of Kubernetes object, **Task**.
3. The unique name of this task.
4. The list of parameters and steps in the task and the workspace used by the task.

This task starts the pod and runs a container inside that pod using the specified image to run the specified commands.

NOTE

Starting with OpenShift Pipelines 1.6, the following defaults from the step YAML file are removed:

- The **HOME** environment variable does not default to the **/tekton/home** directory
- The **workingDir** field does not default to the **/workspace** directory

Instead, the container for the step defines the **HOME** environment variable and the **workingDir** field. However, you can override the default values by specifying the custom values in the YAML file for the step.

As a temporary measure, to maintain backward compatibility with the older OpenShift Pipelines versions, you can set the following fields in the **TektonConfig** custom resource definition to **false**:

```
spec:
  pipeline:
    disable-working-directory-overwrite: false
    disable-home-env-overwrite: false
```

3.2.2. When expression

When expressions guard task execution by setting criteria for the execution of tasks within a pipeline. They contain a list of components that allows a task to run only when certain criteria are met. When expressions are also supported in the final set of tasks that are specified using the **finally** field in the pipeline YAML file.

The key components of a when expression are as follows:

- **input**: Specifies static inputs or variables such as a parameter, task result, and execution status. You must enter a valid input. If you do not enter a valid input, its value defaults to an empty string.

- **operator**: Specifies the relationship of an input to a set of **values**. Enter **in** or **notin** as your operator values.

- **values**: Specifies an array of string values. Enter a non-empty array of static values or variables such as parameters, results, and a bound state of a workspace.

The declared when expressions are evaluated before the task is run. If the value of a when expression is **True**, the task is run. If the value of a when expression is **False**, the task is skipped.

You can use the when expressions in various use cases. For example, whether:

- The result of a previous task is as expected.

- A file in a Git repository has changed in the previous commits.

- An image exists in the registry.

- An optional workspace is available.

The following example shows the when expressions for a pipeline run. The pipeline run will execute the **create-file** task only if the following criteria are met: the **path** parameter is **README.md**, and the **echo-file-exists** task executed only if the **exists** result from the **check-file** task is **yes**.

```
apiVersion: tekton.dev/v1
kind: PipelineRun ❶
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 ❷
        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: ❸
          - input: "$(tasks.check-file.results.exists)"
            operator: in
            values: ["yes"]
        taskSpec:
          steps:
            - name: echo
              image: ubuntu
              script: 'echo file exists'
...
      - name: task-should-be-skipped-1
        when: ❹
          - input: "$(params.path)"
            operator: notin
            values: ["README.md"]
        taskSpec:
          steps:
            - name: echo
              image: ubuntu
              script: exit 1
...
    finally:
      - name: finally-task-should-be-executed
        when: ❺
```

```
          - 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. Specifies the type of Kubernetes object. In this example, **PipelineRun**.

2. Task **create-file** used in the pipeline.

3. **when** expression that specifies to execute the **echo-file-exists** task only if the **exists** result from the **check-file** task is **yes**.

4. **when** expression that specifies to skip the **task-should-be-skipped-1** task only if the **path** parameter is **README.md**.

5. **when** expression that specifies to execute the **finally-task-should-be-executed** task only if the execution status of the **echo-file-exists** task and the task status is **Succeeded**, the **exists** result from the **check-file** task is **yes**, and the **path** parameter is **README.md**.

The **Pipeline Run details**page of the OpenShift Container Platform web console shows the status of the tasks and when expressions as follows:

- All the criteria are met: Tasks and the when expression symbol, which is represented by a diamond shape are green.

- Any one of the criteria are not met: Task is skipped. Skipped tasks and the when expression symbol are grey.

- None of the criteria are met: Task is skipped. Skipped tasks and the when expression symbol are grey.

- Task run fails: Failed tasks and the when expression symbol are red.

3.2.3. Finally tasks

The **finally** tasks are the final set of tasks specified using the **finally** field in the pipeline YAML file. A **finally** task always executes the tasks within the pipeline, irrespective of whether the pipeline runs are executed successfully. The **finally** tasks are executed in parallel after all the pipeline tasks are run, before the corresponding pipeline exits.

You can configure a **finally** task to consume the results of any task within the same pipeline. This approach does not change the order in which this final task is run. It is executed in parallel with other final tasks after all the non-final tasks are executed.

The following example shows a code snippet of the **clone-cleanup-workspace** pipeline. This code clones the repository into a shared workspace and cleans up the workspace. After executing the pipeline tasks, the **cleanup** task specified in the **finally** section of the pipeline YAML file cleans up the workspace.

```
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. Unique name of the pipeline.
2. The shared workspace where the git repository is cloned.
3. The task to clone the application repository to the shared workspace.
4. The task to clean-up the shared workspace.
5. A reference to the task that is to be executed in the task run.
6. A shared storage volume that a task in a pipeline needs at runtime to receive input or provide output.
7. A list of parameters required for a task. If a parameter does not have an implicit default value, you must explicitly set its value.
8. Embedded task definition.

3.2.4. TaskRun

A **TaskRun** instantiates a task for execution with specific inputs, outputs, and execution parameters on a cluster. It can be invoked on its own or as part of a pipeline run for each task in a pipeline.

A task consists of one or more steps that execute container images, and each container image performs a specific piece of build work. A task run executes the steps in a task in the specified order, until all steps execute successfully or a failure occurs. A **TaskRun** is automatically created by a **PipelineRun** for each task in a pipeline.

The following example shows a task run that runs the **apply-manifests** task with the relevant input parameters:

```
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. The task run API version **v1**.
2. Specifies the type of Kubernetes object. In this example, **TaskRun**.

3. Unique name to identify this task run.
4. Definition of the task run. For this task run, the task and the required workspace are specified.
5. Name of the task reference used for this task run. This task run executes the **apply-manifests** task.
6. Workspace used by the task run.

3.2.5. Pipelines

A **Pipeline** is a collection of **Task** resources arranged in a specific order of execution. They are executed to construct complex workflows that automate the build, deployment and delivery of applications. You can define a CI/CD workflow for your application using pipelines containing one or more tasks.

A **Pipeline** resource definition consists of a number of fields or attributes, which together enable the pipeline to accomplish a specific goal. Each **Pipeline** resource definition must contain at least one **Task** resource, which ingests specific inputs and produces specific outputs. The pipeline definition can also optionally include *Conditions*, *Workspaces*, *Parameters*, or *Resources* depending on the application requirements.

The following example shows the **build-and-deploy** pipeline, which builds an application image from a Git repository using the **buildah** task provided in the **openshift-pipelines** namespace:

```
apiVersion: tekton.dev/v1 ❶
kind: Pipeline ❷
metadata:
  name: build-and-deploy ❸
spec: ❹
  workspaces: ❺
  - name: shared-workspace
  params: ❻
  - 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-1.15"
  - name: IMAGE
    type: string
    description: image to be built from the code
  tasks: ❼
  - name: fetch-repository
    taskRef:
      resolver: cluster
      params:
      - name: kind
        value: task
      - name: name
        value: git-clone
      - name: namespace
      value: openshift-pipelines
  workspaces:
  - name: output
    workspace: shared-workspace
  params:
  - name: URL
    value: $(params.git-url)
  - name: SUBDIRECTORY
    value: ""
  - name: DELETE_EXISTING
    value: "true"
  - name: REVISION
    value: $(params.git-revision)
- name: build-image 8
  taskRef:
    resolver: cluster
    params:
    - name: kind
      value: task
    - name: name
      value: buildah
    - name: namespace
      value: openshift-pipelines
  workspaces:
  - name: source
    workspace: shared-workspace
  params:
  - name: TLSVERIFY
    value: "false"
  - name: IMAGE
    value: $(params.IMAGE)
  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 version **v1**.
2. Specifies the type of Kubernetes object. In this example, **Pipeline**.
3. Unique name of this pipeline.
4. Specifies the definition and structure of the pipeline.
5. Workspaces used across all the tasks in the pipeline.
6. Parameters used across all the tasks in the pipeline.
7. Specifies the list of tasks used in the pipeline.
8. Task **build-image**, which uses the **buildah** task provided in the **openshift-pipelines** namespace to build application images from a given Git repository.
9. Task **apply-manifests**, which uses a user-defined task with the same name.
10. Specifies the sequence in which tasks are run in a pipeline. In this example, the **apply-manifests** task is run only after the **build-image** task is completed.

NOTE

The Red Hat OpenShift Pipelines Operator installs the Buildah task in the **openshift-pipelines** namespace and creates the **pipeline** service account with sufficient permission to build and push an image. The Buildah task can fail when associated with a different service account with insufficient permissions.

3.2.6. PipelineRun

A **PipelineRun** is a type of resource that binds a pipeline, workspaces, credentials, and a set of parameter values specific to a scenario to run the CI/CD workflow.

A pipeline run is the running instance of a pipeline. It instantiates a pipeline for execution with specific inputs, outputs, and execution parameters on a cluster. It also creates a task run for each task in the pipeline run.

The pipeline runs the tasks sequentially until they are complete or a task fails. The **status** field tracks and the progress of each task run and stores it for monitoring and auditing purposes.

The following example runs the **build-and-deploy** pipeline with relevant resources and parameters:

```
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 Pipeline run API version **v1**.

2 The type of Kubernetes object. In this example, **PipelineRun**.

3 Unique name to identify this pipeline run.

4 Name of the pipeline to be run. In this example, **build-and-deploy**.

5 The list of parameters required to run the pipeline.

6 Workspace used by the pipeline run.

Additional resources

- Authenticating pipelines with repositories using secrets

3.2.7. Workspaces

NOTE

It is recommended that you use workspaces instead of the **PipelineResource** CRs in Red Hat OpenShift Pipelines, as **PipelineResource** CRs are difficult to debug, limited in scope, and make tasks less reusable.

Workspaces declare shared storage volumes that a task in a pipeline needs at runtime to receive input or provide output. Instead of specifying the actual location of the volumes, workspaces enable you to declare the filesystem or parts of the filesystem that would be required at runtime. A task or pipeline declares the workspace and you must provide the specific location details of the volume. It is then mounted into that workspace in a task run or a pipeline run. This separation of volume declaration from runtime storage volumes makes the tasks reusable, flexible, and independent of the user environment.

With workspaces, you can:

- Store task inputs and outputs

- Share data among tasks

- Use it as a mount point for credentials held in secrets

- Use it as a mount point for configurations held in config maps

- Use it as a mount point for common tools shared by an organization
- Create a cache of build artifacts that speed up jobs

You can specify workspaces in the **TaskRun** or **PipelineRun** using:

- A read-only config map or secret
- An existing persistent volume claim shared with other tasks
- A persistent volume claim from a provided volume claim template
- An **emptyDir** that is discarded when the task run completes

The following example shows a code snippet of the **build-and-deploy** pipeline, which declares a **shared-workspace** workspace for the **build-image** and **apply-manifests** tasks as defined in the pipeline.

```
apiVersion: tekton.dev/v1
kind: Pipeline
metadata:
  name: build-and-deploy
spec:
  workspaces: 1
  - name: shared-workspace
  params:
...
  tasks: 2
  - name: build-image
    taskRef:
      resolver: cluster
      params:
      - name: kind
        value: task
      - name: name
        value: buildah
      - name: namespace
        value: openshift-pipelines
    workspaces: 3
    - name: source 4
      workspace: shared-workspace 5
    params:
    - name: TLSVERIFY
      value: "false"
    - name: IMAGE
      value: $(params.IMAGE)
    runAfter:
    - fetch-repository
  - name: apply-manifests
    taskRef:
      name: apply-manifests
    workspaces: 6
    - name: source
      workspace: shared-workspace
    runAfter:
      - build-image
...
```

① List of workspaces shared between the tasks defined in the pipeline. A pipeline can define as many workspaces as required. In this example, only one workspace named **shared-workspace** is declared.

② Definition of tasks used in the pipeline. This snippet defines two tasks, **build-image** and **apply-manifests**, which share a common workspace.

③ List of workspaces used in the **build-image** task. A task definition can include as many workspaces as it requires. However, it is recommended that a task uses at most one writable workspace.

④ Name that uniquely identifies the workspace used in the task. This task uses one workspace named **source**.

⑤ Name of the pipeline workspace used by the task. Note that the workspace **source** in turn uses the pipeline workspace named **shared-workspace**.

⑥ List of workspaces used in the **apply-manifests** task. Note that this task shares the **source** workspace with the **build-image** task.

Workspaces help tasks share data, and allow you to specify one or more volumes that each task in the pipeline requires during execution. You can create a persistent volume claim or provide a volume claim template that creates a persistent volume claim for you.

The following code snippet of the **build-deploy-api-pipelinerun** pipeline run uses a volume claim template to create a persistent volume claim for defining the storage volume for the **shared-workspace** workspace used in the **build-and-deploy** pipeline.

```
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 Specifies the list of pipeline workspaces for which volume binding will be provided in the pipeline run.

② The name of the workspace in the pipeline for which the volume is being provided.

3 Specifies a volume claim template that creates a persistent volume claim to define the storage volume for the workspace.

3.2.8. Triggers

Use *Triggers* in conjunction with pipelines to create a full-fledged CI/CD system where Kubernetes resources define the entire CI/CD execution. Triggers capture the external events, such as a Git pull request, and process them to extract key pieces of information. Mapping this event data to a set of predefined parameters triggers a series of tasks that can then create and deploy Kubernetes resources and instantiate the pipeline.

For example, you define a CI/CD workflow using Red Hat OpenShift Pipelines for your application. The pipeline must start for any new changes to take effect in the application repository. Triggers automate this process by capturing and processing any change event and by triggering a pipeline run that deploys the new image with the latest changes.

Triggers consist of the following main resources that work together to form a reusable, decoupled, and self-sustaining CI/CD system:

- The **TriggerBinding** resource extracts the fields from an event payload and stores them as parameters.
  The following example shows a code snippet of the **TriggerBinding** resource, which extracts the Git repository information from the received event payload:

  ```
  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 The API version of the **TriggerBinding** resource. In this example, **v1beta1**.

  2 Specifies the type of Kubernetes object. In this example, **TriggerBinding**.

  3 Unique name to identify the **TriggerBinding** resource.

  4 List of parameters which will be extracted from the received event payload and passed to the **TriggerTemplate** resource. In this example, the Git repository URL, name, and revision are extracted from the body of the event payload.

- The **TriggerTemplate** resource acts as a standard for the way resources must be created. It specifies the way parameterized data from the **TriggerBinding** resource should be used. A trigger template receives input from the trigger binding, and then performs a series of actions that results in creation of new pipeline resources, and initiation of a new pipeline run.

The following example shows a code snippet of a **TriggerTemplate** resource, which creates a pipeline run using the Git repository information received from the **TriggerBinding** resource you just created:

```
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-1.15
  - 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. The API version of the **TriggerTemplate** resource. In this example, **v1beta1**.
2. Specifies the type of Kubernetes object. In this example, **TriggerTemplate**.
3. Unique name to identify the **TriggerTemplate** resource.
4. Parameters supplied by the **TriggerBinding** resource.

5. List of templates that specify the way resources must be created using the parameters received through the **TriggerBinding** or **EventListener** resources.

- The **Trigger** resource combines the **TriggerBinding** and **TriggerTemplate** resources, and optionally, the **interceptors** event processor.
  Interceptors process all the events for a specific platform that runs before the **TriggerBinding** resource. You can use interceptors to filter the payload, verify events, define and test trigger conditions, and implement other useful processing. Interceptors use secret for event verification. After the event data passes through an interceptor, it then goes to the trigger before you pass the payload data to the trigger binding. You can also use an interceptor to modify the behavior of the associated trigger referenced in the **EventListener** specification.

  The following example shows a code snippet of a **Trigger** resource, named **vote-trigger** that connects the **TriggerBinding** and **TriggerTemplate** resources, and the **interceptors** event processor.

```
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. The API version of the **Trigger** resource. In this example, **v1beta1**.
2. Specifies the type of Kubernetes object. In this example, **Trigger**.
3. Unique name to identify the **Trigger** resource.
4. Service account name to be used.

5. Interceptor name to be referenced. In this example, **github**.
6. Desired parameters to be specified.
7. Name of the **TriggerBinding** resource to be connected to the **TriggerTemplate** resource.
8. Name of the **TriggerTemplate** resource to be connected to the **TriggerBinding** resource.
9. Secret to be used to verify events.

- The **EventListener** resource provides an endpoint, or an event sink, that listens for incoming HTTP-based events with a JSON payload. It extracts event parameters from each **TriggerBinding** resource, and then processes this data to create Kubernetes resources as specified by the corresponding **TriggerTemplate** resource. The **EventListener** resource also performs lightweight event processing or basic filtering on the payload using event **interceptors**, which identify the type of payload and optionally modify it. Currently, pipeline triggers support five types of interceptors: *Webhook Interceptors*, *GitHub Interceptors*, *GitLab Interceptors*, *Bitbucket Interceptors*, and *Common Expression Language (CEL) Interceptors*.
  The following example shows an **EventListener** resource, which references the **Trigger** resource named **vote-trigger**.

```
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. The API version of the **EventListener** resource. In this example, **v1beta1**.
2. Specifies the type of Kubernetes object. In this example, **EventListener**.
3. Unique name to identify the **EventListener** resource.
4. Service account name to be used.
5. Name of the **Trigger** resource referenced by the **EventListener** resource.

3.3. ADDITIONAL RESOURCES

- For information on installing OpenShift Pipelines, see Installing OpenShift Pipelines.
- For more details on creating custom CI/CD solutions, see Creating CI/CD solutions for applications using OpenShift Pipelines.
- For more details on re-encrypt TLS termination, see Re-encryption Termination.
- For more details on secured routes, see the Secured routes section.