

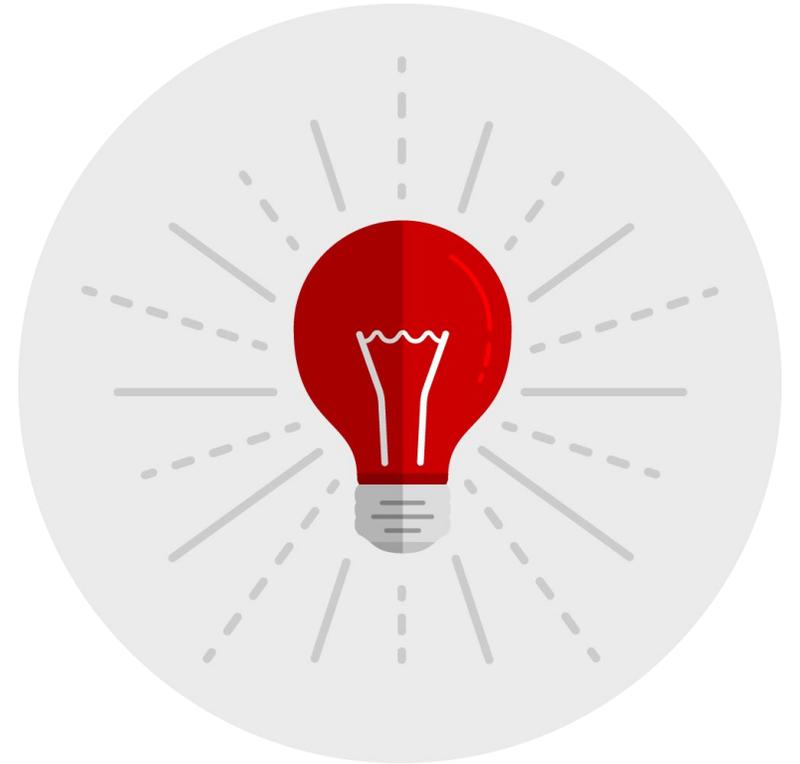
# Tekton in Action with Red Hat OpenShift Pipelines

Natale Vinto  
Developer Advocate  
[@natalevinto](#)



# AGENDA

- OpenShift
- What is CI/CD?
- Cloud Native CI/CD
- OpenShift Pipelines
- Tekton components
- Tekton in action



DevOps is the key to meet the insatiable  
demand for delivering quality applications  
rapidly

# OpenShift

## A Comprehensive DevOps Platform for Hybrid Cloud

Build container images from source code using Kubernetes tools



Traditional and Kubernetes-native CI/CD



Declarative GitOps for multi-cluster continuous delivery

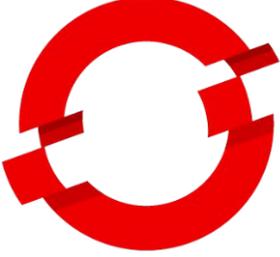


OpenShift  
Builds

OpenShift  
Pipelines

OpenShift  
GitOps

OpenShift

The logo consists of a red circular icon with a white center and a red ring, and the text "Red Hat OpenShift" in a bold, black, sans-serif font.

# Red Hat OpenShift

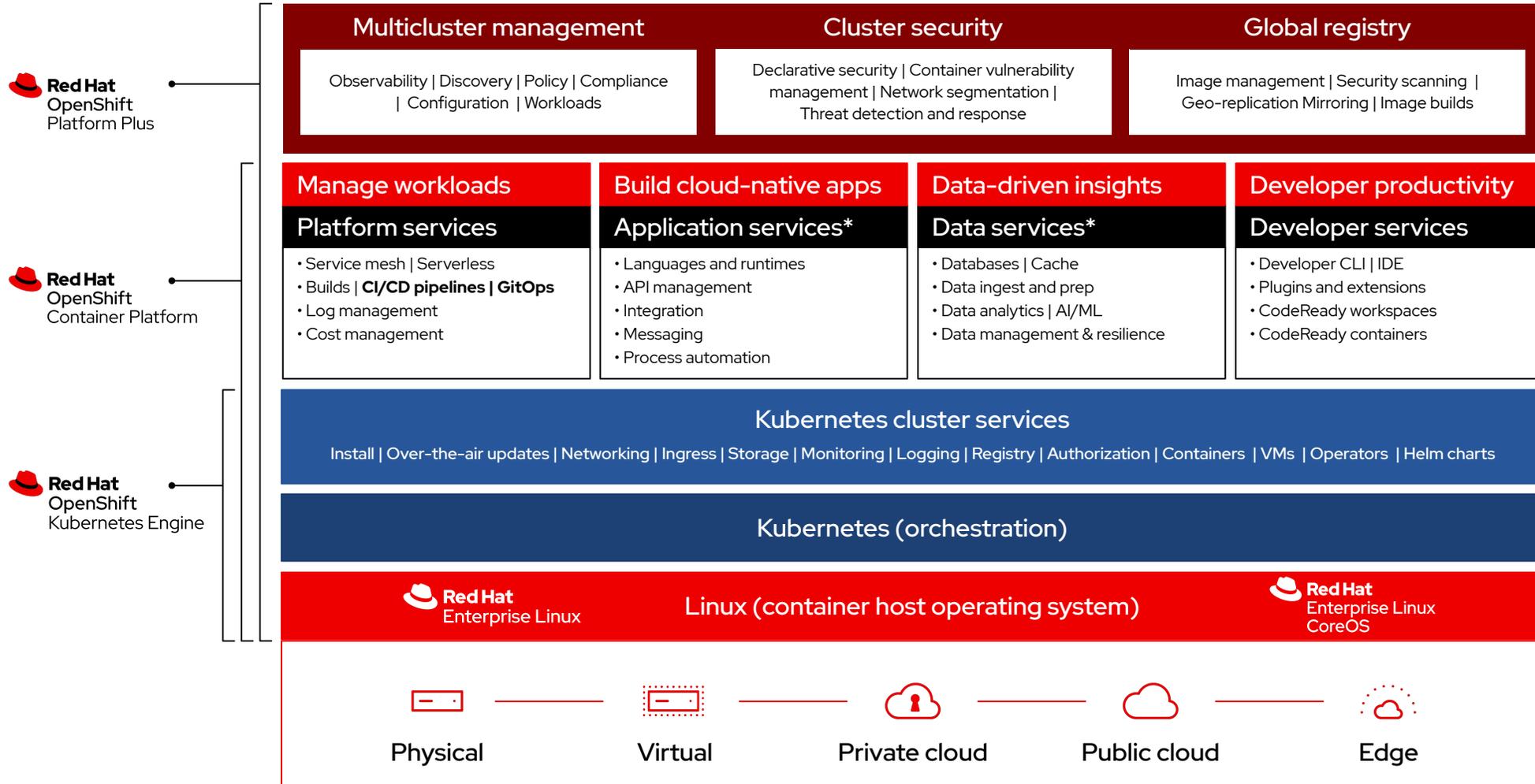
A secure and enterprise-grade container application platform based on **Kubernetes** for traditional and cloud-native applications

# OpenShift Platform Plus

 **Red Hat**  
Advanced Cluster Management  
for Kubernetes

 **Red Hat**  
Advanced Cluster Security  
for Kubernetes

 **Red Hat**  
Quay

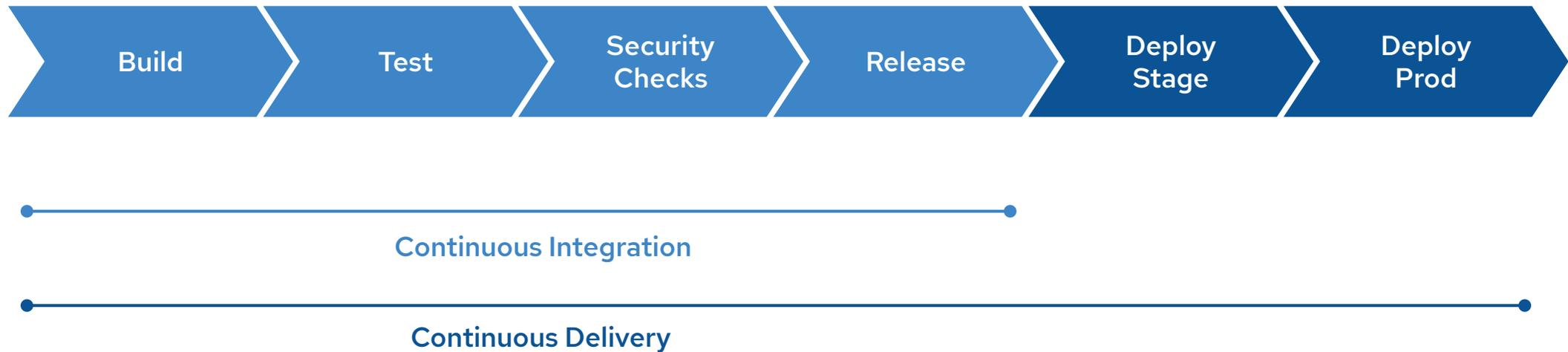


\* Red Hat OpenShift® includes supported runtimes for popular languages/frameworks/databases. Additional capabilities listed are from the Red Hat Application and Data Services portfolio.

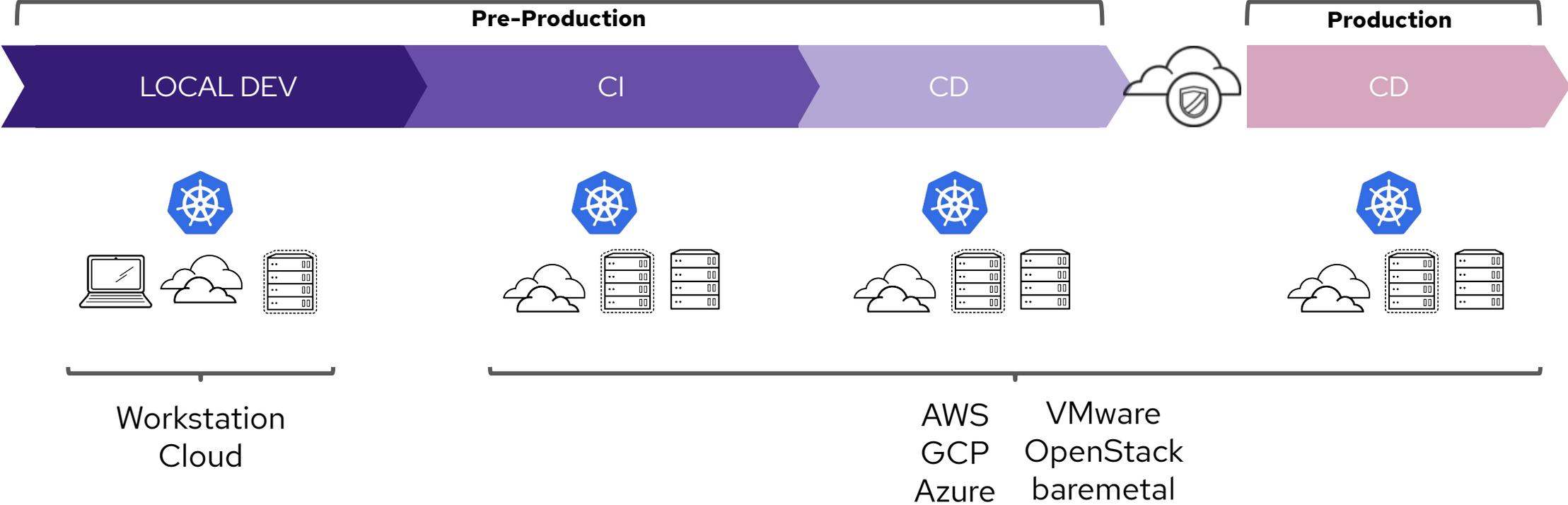
# What is CI/CD?



# Continuous Integration & Continuous Delivery



# Fact: Kubernetes is the target platform



# One Continuous Delivery

Multiple Clouds

Multiple Platforms

DEVELOPMENT

CONTINUOUS INTEGRATION

CONTINUOUS DELIVERY



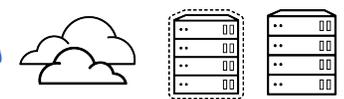
Workstation



Kubernetes



Kubernetes



Kubernetes

Azure

AWS

GCP

VMware

OpenStack

baremetal

# Cloud Native CI/CD

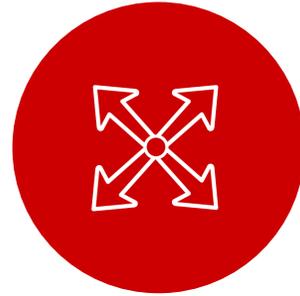


# What is Cloud Native CI/CD?



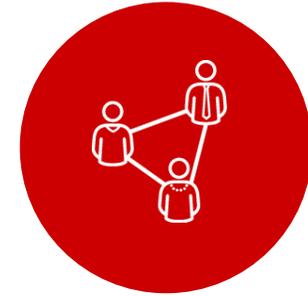
## Containers

Built for container apps and runs on Kubernetes



## Serverless

Runs serverless with no CI/CD engine to manage and maintain



## DevOps

Designed with microservices and distributed teams in mind

# Why Cloud-Native CI/CD?

Traditional CI/CD	Cloud-Native CI/CD
Designed for Virtual Machines	Designed for Containers and Kubernetes
Requires IT Ops for CI engine maintenance	Pipeline as a service with no Ops overhead
Plugins shared across CI engine	Pipelines fully isolated from each other
Plugin dependencies with undefined update cycles	Everything lifecycle'd as container images
No interoperability with Kubernetes resources	Native Kubernetes resources
Admin manages persistence	Platform manages persistence
Config baked into CI engine container	Configured via Kubernetes ConfigMaps

# Why Cloud-Native CI/CD?

Traditional CI/CD	Cloud-Native CI/CD
Designed for Virtual Machines	Designed for Containers and Kubernetes
Require IT Ops for CI engine maintenance	Pipeline as a service with no Ops overhead
 <b>Jenkins</b> Plugins shared across CI engine Plugin dependencies with undefined update cycles	 <b>TEKTON</b> Pipelines fully isolated from each other Evaluating lines of code in containers
No interoperability with Kubernetes resources	Native Kubernetes resources
Admin manages persistence	Platform manages persistence
Config baked into CI engine container	Configured via Kubernetes ConfigMaps



An open-source project for providing a set of shared and standard components for building Kubernetes-style CI/CD systems



CD.FOUNDATION

Governed by the Continuous Delivery Foundation

Contributions from Google, Red Hat, Cloudbees, IBM, Pivotal and many more



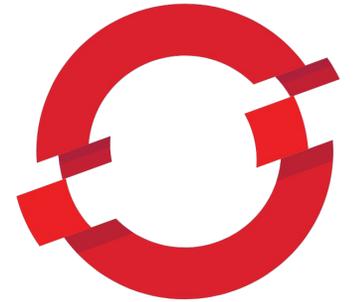
**Composable**

**Declarative**



**Reproducible**

**Cloud Native**



# OpenShift Pipelines



# OpenShift Pipelines



## Built for Kubernetes

Cloud-native pipelines taking advantage of Kubernetes execution and, operational model and concepts



## Scale on-demand

Pipelines run and scale on-demand in isolated containers, with repeatable and predictable outcomes



## Secure pipeline execution

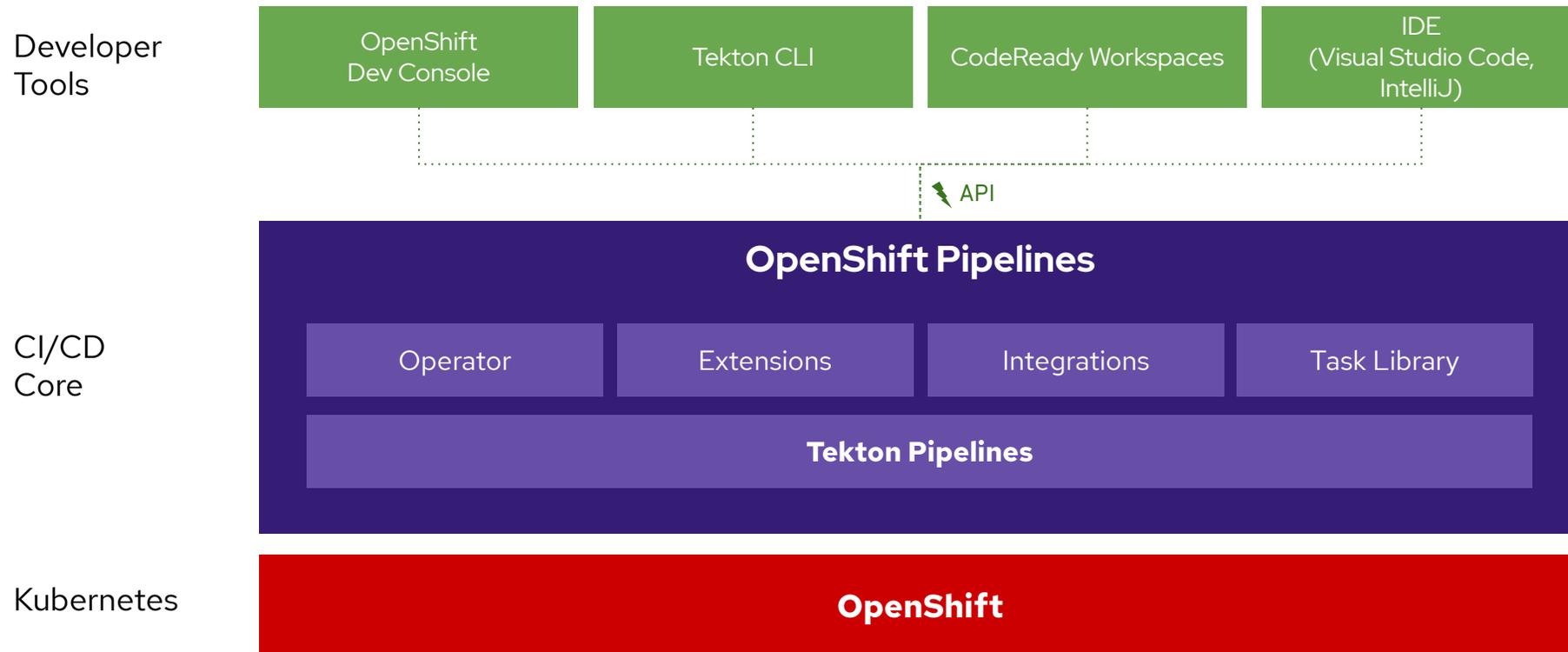
Kubernetes RBAC and security model ensures security consistently across pipelines and workloads



## Flexible and powerful

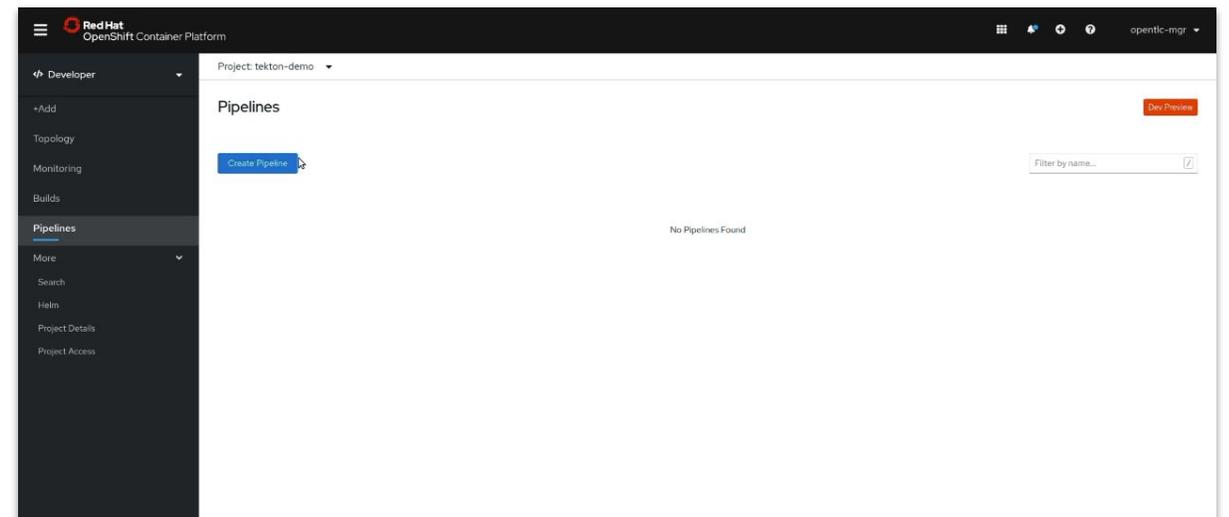
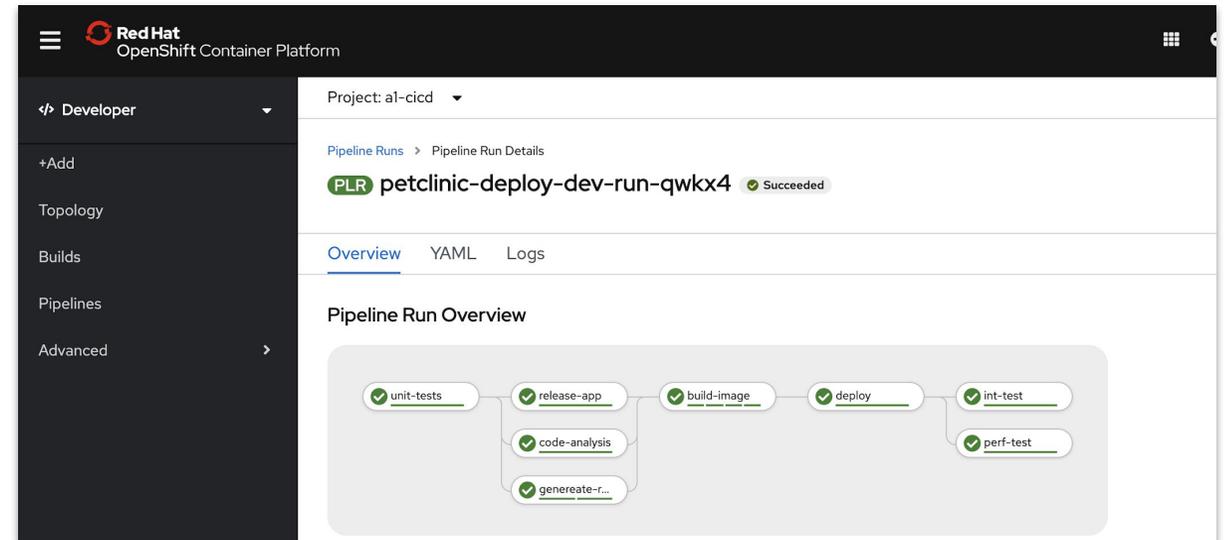
Granular control over pipeline execution details on Kubernetes, to support your exact requirements

# OpenShift Pipelines

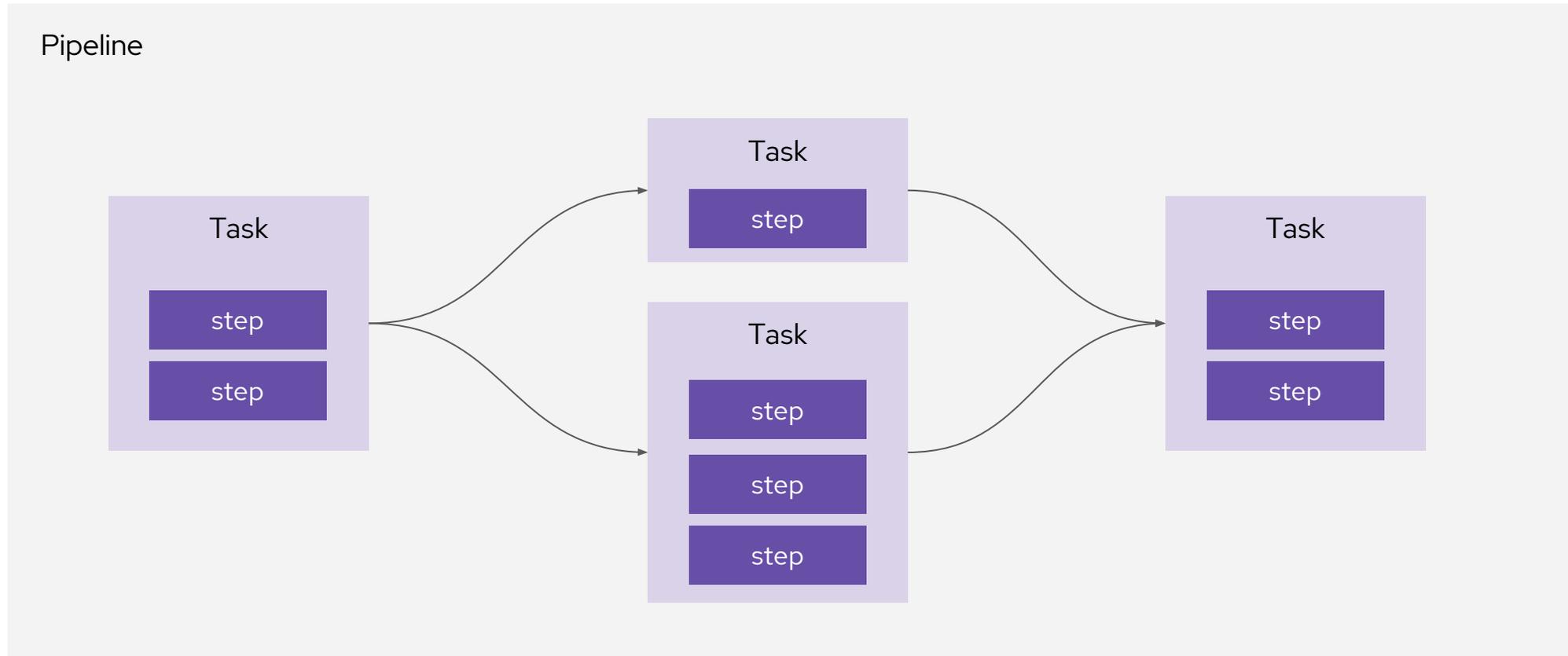


# OpenShift Pipelines

- Based on Tekton Pipelines
- Kubernetes-native declarative CI/CD
- Pipelines run on-demand in isolated containers
- No central server to maintain! No plugin conflicts!
- Task library and integration with Tekton Hub
- Secure pipelines aligned with Kubernetes RBAC
- Visual and IDE-based pipeline authoring
- Pipeline templates when importing apps
- Automated install and upgrades via OperatorHub
- CLI, Web, VS Code and IntelliJ plugins



# Tekton Concepts



## Tekton Concepts: step

- Run command or script in a container
- Kubernetes container spec
  - Env vars
  - Volumes
  - Config maps
  - Secrets

```
- name: build
  image: maven:3.6.0-jdk-8-slim
  command: ["mvn"]
  args: ["install"]
```

```
- name: parse-yaml
  image: python3
  script: |-
    #!/usr/bin/env python3
    ...
```

## Tekton Concepts: Task

- Performs a specific task
- List of steps
- Steps run sequentially
- Reusable

```
kind: Task
metadata:
  name: buildah
spec:
  params:
    - name: IMAGE
  steps:
    - name: build
      image: quay.io/buildah/stable:latest
      command: ["buildah"]
      args: ["bud", ".", "-t", "$(params.IMAGE)"]
    - name: push
      image: quay.io/buildah/stable:latest
      script: |
        buildah push $(params.IMAGE) docker://$(params.IMAGE)
```

# Tekton Hub

Search, discover and install Tekton Tasks

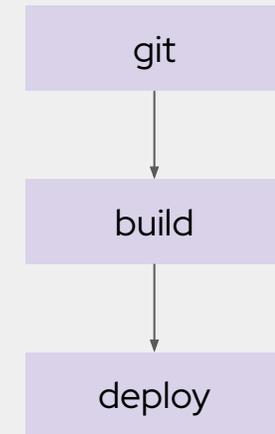
The screenshot shows the Tekton Hub (BETA) interface. At the top, there is a navigation bar with the Tekton Hub logo and a 'Login' link. Below the navigation bar, a large banner reads 'Welcome to Tekton Hub' and 'Discover, search and share reusable Tasks and Pipelines'. The main content area features a search bar and a 'Sort' dropdown menu set to 'Name'. On the left side, there is a 'Refine By' sidebar with filters for 'Kind' (Task, Pipeline), 'Support Tier' (Official, Verified, Community), and 'Categories' (Build Tools, CLI, Cloud, Deploy, Image Build, Notification, Others, Test Framework). The main area displays a grid of task cards. Each card includes a task icon, a star rating, the task name and version, a brief description, the update date, and tags for the task's kind.

Task Name	Version	Rating	Update Date	Tags
Ansible Runner	v0.1	4.5	Updated 3 weeks ago	cli
ansible tower cli	v0.1	2.0	Updated 3 weeks ago	ansible, cli
argocd	v0.1	3.0	Updated 3 weeks ago	deploy
aws cli	v0.1	5.0	Updated 3 weeks ago	cli
Amazon ECR Login	v0.1	4.0	Updated 3 weeks ago	aws, ecr
azure cli	v0.1	1.0	Updated 4 months ago	cli
bentoml	v0.1	0.0	Updated 3 weeks ago	cli
Python Black	v0.1	0.0	Updated 3 weeks ago	formatter, python

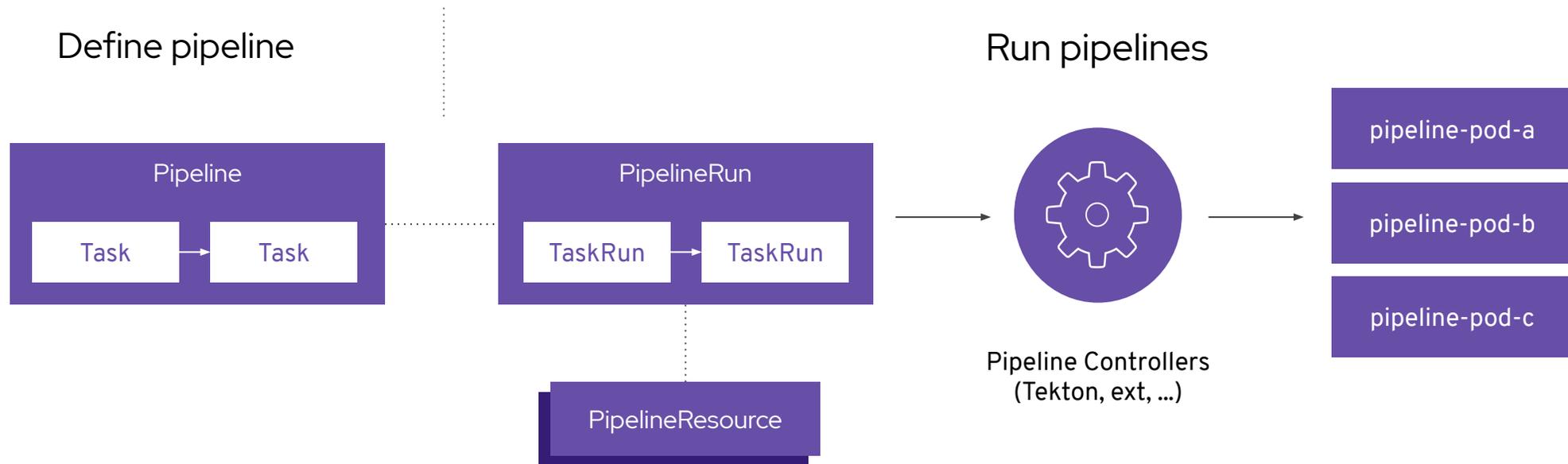
# Tekton Concepts: Pipeline

- A graph of Tasks: concurrent & sequential
- Tasks run on different nodes
- Task execution logic
  - Conditional
  - Retries
- Share data between tasks

```
kind: Pipeline
metadata:
  name: deploy-dev
spec:
  params:
    - name: IMAGE_TAG
  tasks:
    - name: git
      taskRef:
        name: git-clone
        params: [...]
    - name: build
      taskRef:
        name: maven
        params: [...]
      runAfter: ["git"]
    - name: deploy
      taskRef:
        name: knative-deploy
        params: [...]
      runAfter: ["build"]
```



# OpenShift Pipelines Architecture



# Migrate from Jenkins to Tekton



## Jenkins Pipeline

```
pipeline {
  stages {
    stage('Git Clone') {
      steps { ... }
    }
    stage('Build App') {
      steps { ... }
    }
    stage('Test') {
      steps { ... }
    }
    stage('Code Analysis') {
      steps { ... }
    }
  }
}
```

## Tekton Pipeline

```
kind: Pipeline
spec:
  tasks:
  - name: git-clone
  - name: build-app
  - name: test
  - name: code-analysis
```

## Jenkins Pipeline

```
pipeline {  
  
  agent {  
    label 'maven'  
  }  
  
  stages {  
    stage ('Clone') {  
      git url: 'https://github.com/...'  
    }  
    stage ('Build App') {  
      withMaven(maven: 'maven-3') {  
        sh "mvn clean verify"  
      }  
    }  
    ...  
  }  
}
```

## Tekton Pipeline

```
kind: Pipeline  
spec:  
  tasks:  
  
  - name: git-clone  
    taskRef:  
      name: git-clone  
    params:  
    - name: url  
      value: https://github.com/...  
    workspaces:  
    - name: app-workspace  
      workspace: app-source  
  
  - name: build-app  
    taskRef:  
      name: maven  
    params:  
    - name: GOALS  
      value: ["clean", "verify"]  
    runAfter:  
    - git-clone  
    workspaces:  
    - name: app-workspace  
      workspace: app-source
```

# Tekton in action



# Install Pipeline via OperatorHub marketplace

The screenshot shows the OpenShift OperatorHub marketplace interface. On the left, there is a navigation sidebar with categories like Administrator, Home, Projects, Search, Explore, Events, Operators, Workloads, Networking, Storage, Builds, Monitoring, Compute, User Management, and Administration. The main content area displays the 'OperatorHub' page with a search bar and a list of operators. The 'OpenShift Pipelines Operator' is highlighted, showing its details in a modal window.

**OperatorHub**  
Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. You can purchase services to your developers. After installation, the Operator capabilities will appear in the Developer Catalog page.

**OpenShift Pipelines Operator**  
1.0.1 provided by Red Hat

**Operator Version**  
1.0.1

**Capability Level**

- Basic Install
- Seamless Upgrades
- Full Lifecycle
- Deep Insights
- Auto Pilot

**Provider Type**  
Red Hat

**Provider**  
Red Hat

**Repository**  
<https://github.com/openshift/tektoncd-pipeline-operator>

**Container Image**  
registry.redhat.io/openshift-pipelines-tech-preview/pipelines-rhel8-operator@sha256:78260d7b70e43ec4782176fe892fae2998e5885943f67391

**Features**

- Standard CI/CD pipelines definition
- Build images with Kubernetes tools such as S2I, Buildah, Buildpacks, Kaniko, etc
- Deploy applications to multiple platforms such as Kubernetes, serverless and VMs
- Easy to extend and integrate with existing tools
- Scale pipelines on-demand
- Portable across any Kubernetes platform
- Designed for microservices and decentralised team
- Integrated with OpenShift Developer Console

**Installation**  
*OpenShift Pipelines Operator* gets installed into a single namespace (openshift-operators) which would then install *OpenShift Pipelines* into the openshift-pipelines namespace. *OpenShift Pipelines* is however cluster-wide and can run pipelines created in any namespace.

**Components**

- OpenShift-Pipelines: v0.11.3
- OpenShift-Pipelines-Triggers: v0.4.0
- OpenShift-Pipelines-ClusterTasks: v0.11

**Note:** If you are already subscribed to the community version of OpenShift-Pipelines-Operator, then please uninstall the community version of the operator before subscribing to this operator.

<https://console-openshift-console.natalie-test-4-5-6-f5541308b177087861a229b886140c95-0000.us-east.containers.appdomain.cloud/operatorhub/subscribe?pkq=openshift-pipelines-operator-rh&catalog=redhat-operators&catalogNamespace=openshift-marketplace&targetNamespace=...>

# Run Pipelines

Red Hat OpenShift Container Platform

Project: a3-cicd

Pipeline Runs > Pipeline Run Details

**PLR** petclinic-dev-eb123ee Succeeded Tech preview Actions

[Details](#) [YAML](#) [Task Runs](#) [Logs](#) [Events](#)

### Pipeline Run Details

```

graph LR
    source-clone[✓ source-clone] --> code-analysis[✓ code-analysis]
    source-clone --> dependency[✓ dependency...]
    source-clone --> unit-tests[✓ unit-tests]
    code-analysis --> release-app[✓ release-app]
    dependency --> release-app
    unit-tests --> release-app
    release-app --> build-image[✓ build-image]
    build-image --> config-clone[✓ config-clone]
    build-image --> tests-clone[✓ tests-clone]
    config-clone --> deploy-dev[✓ deploy-dev]
    tests-clone --> deploy-dev
    deploy-dev --> int-test[✓ int-test]
    deploy-dev --> perf-test[✓ perf-test]
  
```

**Name**  
petclinic-dev-eb123ee

**Status**  
✓ Succeeded

**Namespace**  
NS a3-cicd

**Pipeline**  
PL petclinic-deploy-dev

**Labels** Edit

tekton.dev/pipeline=petclinic-deploy-dev  
trigger.tekton.dev/webhook

**Triggered by:**  
EL webhook

# Check logs of running pipelines

The screenshot displays the Red Hat OpenShift Container Platform interface. The top navigation bar includes the Red Hat logo, the text 'OpenShift Container Platform', and a user profile 'siamak'. The left sidebar shows navigation options: Developer, + Add, Topology, Builds, Pipelines (selected), and Advanced. The main content area shows 'Project: Project01' and 'Pipeline Run Details' for 'pipelinerun01a' in a 'Running' state. A 'Tech Preview' badge and an 'Actions' dropdown are visible. The 'Logs' tab is active, showing a list of pipeline steps on the left: code compile, compile & test, unit test, security check, and image build (selected). The 'image build' step logs show the following output:

```

image build

[Plack::Sandbox::_2fopt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsqi:54] core @2018-08-23 18:28:53> looking
for get /health in extlib/lib/perl5/Dancer2/Core/App.pm l. 36
[Plack::Sandbox::_2fopt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsqi:54] core @2018-08-23 18:28:53> Entering
hook core.error.init in (eval 306) l. 1
[Plack::Sandbox::_2fopt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsqi:54] core @2018-08-23 18:28:53> Entering
hook core.error.before in (eval 306) l. 1
[Plack::Sandbox::_2fopt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsqi:54] core @2018-08-23

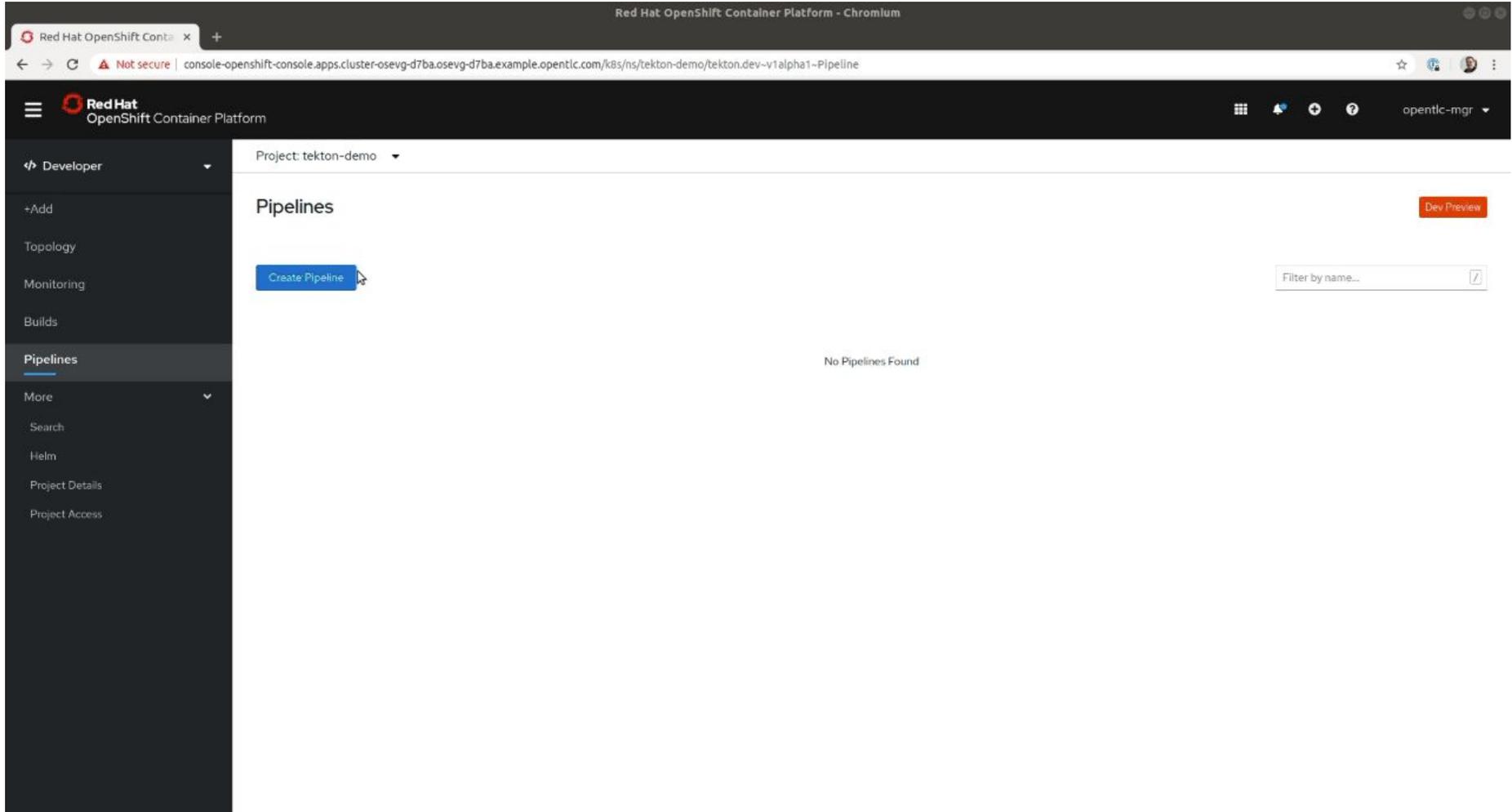
[Plack::Sandbox::_2fopt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsqi:54] core @2018-08-23 18:28:53> looking
for get /health in extlib/lib/perl5/Dancer2/Core/App.pm l. 36
[Plack::Sandbox::_2fopt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsqi:54] core @2018-08-23 18:28:53> Entering
hook core.error.init in (eval 306) l. 1
[Plack::Sandbox::_2fopt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsqi:54] core @2018-08-23 18:28:53> Entering
hook core.error.before in (eval 306) l. 1
[Plack::Sandbox::_2fopt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsqi:54] core @2018-08-23
  
```

# Create apps with Pipelines

The screenshot displays the Red Hat OpenShift Container Platform Developer console. The top navigation bar includes the Red Hat logo, the text 'Red Hat OpenShift Container Platform', and user information 'siamak'. The left sidebar shows the 'Developer' view with a '+Add' button highlighted. The main content area is titled 'Add' and shows 'Project: pipelines-demo' and 'Application: all applications'. A light blue message box states 'No workloads found' and provides instructions: 'To add content to your project, create an application, component or service using one of these options.' Below this, six options are presented in a grid:

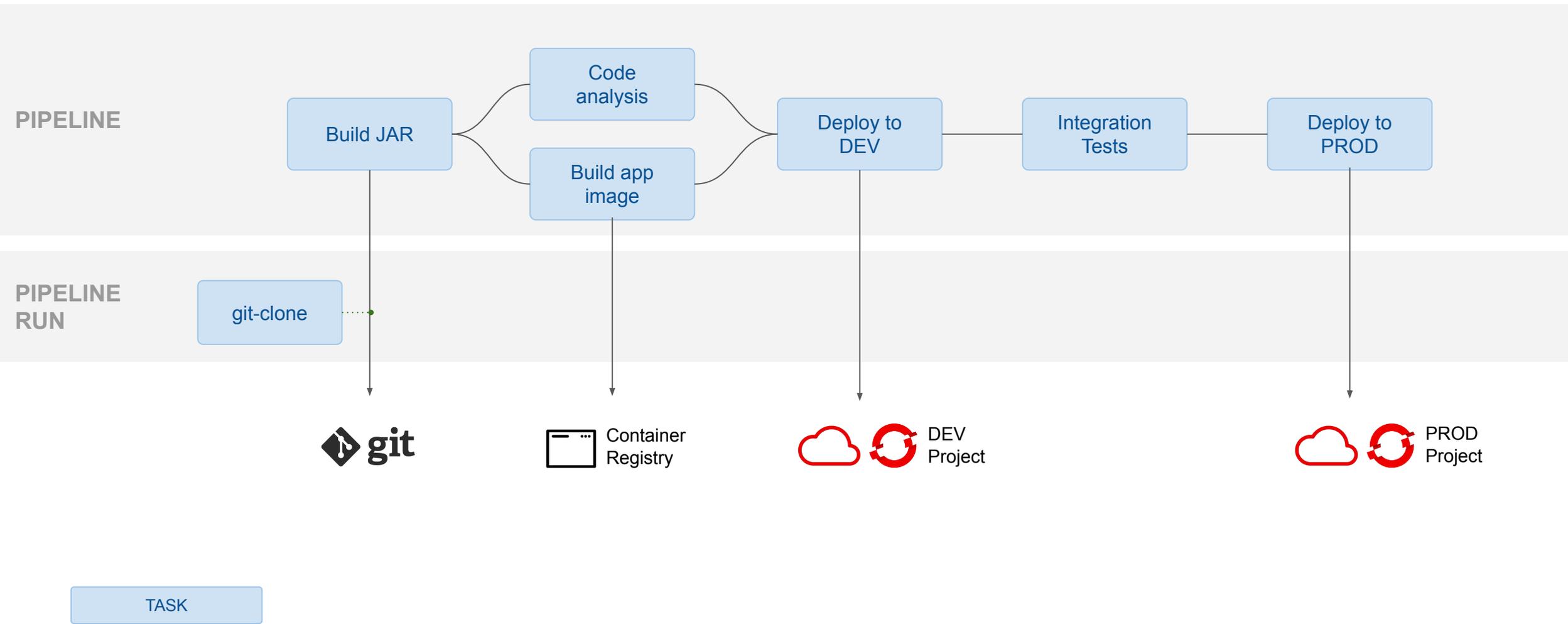
- From Git**: Import code from your git repository to be built and deployed.
- Container Image**: Deploy an existing image from an image registry or image stream tag.
- From Catalog**: Browse the catalog to discover, deploy and connect to services.
- From Dockerfile**: Import your Dockerfile from your git repo to be built & deployed.
- YAML**: Create resources from their YAML or JSON definitions.
- Database**: Browse the catalog to discover database services to add to your application.

# Create Pipelines with Pipeline UI



The screenshot displays the Red Hat OpenShift Container Platform Pipeline UI. The browser address bar shows the URL: `console-openshift-console.apps.cluster-osevg-d7ba.osevg-d7ba.example.opentlc.com/k8s/ns/tekton-demo/tekton.dev-v1alpha1-Pipeline`. The page title is "Pipelines" and it includes a "Dev Preview" badge. A "Create Pipeline" button is visible, along with a search filter input labeled "Filter by name...". The main content area displays "No Pipelines Found". The left sidebar contains navigation options: Developer, +Add, Topology, Monitoring, Builds, Pipelines (selected), More, Search, Helm, Project Details, and Project Access. The top navigation bar shows the Red Hat logo, the text "Red Hat OpenShift Container Platform", and the user "opentlc-mgr".

# OpenShift Pipeline Example



## Interactive Learning Portal

Our Interactive Learning Scenarios provide you with a pre-configured OpenShift® instance, accessible from your browser without any downloads or configuration. Use it to experiment, learn OpenShift and see how we can help solve real-world problems.

[learn.openshift.com](https://learn.openshift.com)

Foundations of  
OpenShift

START COURSE

Building Applications On  
OpenShift

START COURSE

Subsystems,  
Components, and  
Internals

START COURSE

OpenShift Playgrounds

START COURSE

Service Mesh Workshop  
with Istio

START COURSE

Building Operators on  
OpenShift

START COURSE

AI and Machine Learning  
on OpenShift

- ▶ Setup
- ▶ Pipeline Resources
- ▶ Tasks
- ▼ Pipelines
  - Add Tasks from Catalog
  - Create Pipeline**
  - Deploy Pipeline
  - Run Pipeline
  - Test Pipeline
  - Clean
- ▶ Workspaces
- ▶ Private Registries and Repositories
- ▶ Triggers
- ▶ OpenShift Pipelines

## Deploy Pipeline

The Kubernetes service deployment Pipeline could be created using the command

```
kubectl apply -n tektontutorial -f svc-deploy.yaml
```

We will use the Tekton cli to inspect the created resources

```
tkn pipeline ls
```

The above command should list one Pipeline as shown below:

NAME	AGE	LAST RUN	STARTED	DURATION
svc-deploy	4 seconds ago	---	---	---

### TIP

Use the command **help** via `tkn pipeline --help` to see more options

## Run Pipeline

[dn.dev/tekton-tutorial](https://dn.dev/tekton-tutorial)

# Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

 [linkedin.com/company/red-hat](https://linkedin.com/company/red-hat)

 [youtube.com/user/RedHatVideos](https://youtube.com/user/RedHatVideos)

 [facebook.com/redhatinc](https://facebook.com/redhatinc)

 [twitter.com/RedHat](https://twitter.com/RedHat)