



OCTOBER 17-18, 2019 BRATISLAVA, SLOVAKIA

Accenture conference on emerging
technologies and open source

OpenShift Pipelines with Tekton

Organizers

accenture

STU FIIT STU FEI



Partners



Red Hat

SIEMENS



mongoDB®

Media partner

robíme it

Supporters

Pivotal

ableneo
The innovation enablers

SEGES
space for solutions

WHITESTEIN
Technologies

tuli

OPENSLAVA'19



LOS TRES AMIGOS

ABOUT US

CONFIDENTIAL Designator



OPENSLAVA'19



Krain Arnold is a Senior Specialist Solution Architect at Red Hat with a focus on container management and automation.

T: @krain_arnold

E: krain@redhat.com

ABOUT US

CONFIDENTIAL Designator



OPENSLAVA'19



Natale Vinto is a Senior Specialist Solution Architect at Red Hat focusing on OpenShift and DevOps

T: @natalevinto

E: nvinto@redhat.com

ABOUT US

CONFIDENTIAL Designator



Rafael Cardona is a Senior Specialist Solution Architect at Red Hat focusing on Infrastructure and Automation

T: @rcardona_cloud
E: rcardona@redhat.com



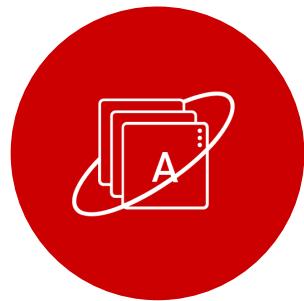
What are OpenShift Pipelines?



Cloud-Native CI/CD

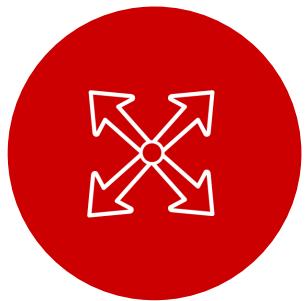


OPENSLSAVA'19



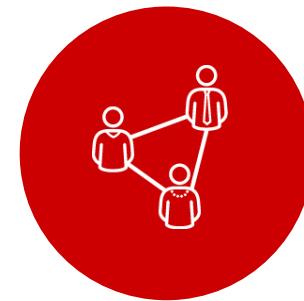
Containers

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

OpenShift Pipelines

a Cloud-Native CI/CD Experience on OpenShift



Standard Kubernetes-style pipelines

Declarative pipelines with standard Kubernetes custom resources (CRDs) based on Tekton*



Build images with Kubernetes tools

Use tools of your choice (source-to-image, buildah, kaniko, jib, etc) for building container images



Run pipelines in containers

Scale pipeline executions on-demand with containers on Kubernetes



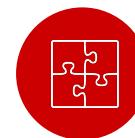
Deploy to multiple platforms

Deploy applications to multiple platforms like serverless, virtual machines and Kubernetes



Powerful command-line tool

Run and manage pipelines with an interactive command-line tool



Integrated CI/CD experience**

A CI/CD experience integrated with OpenShift Developer Console, Visual Studio Code Plugin, Eclipse Che Plugins and more



TEKTON



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



Governed by the Continuous Delivery Foundation
Contributions from Google, Red Hat, Cloudbees, IBM, Pivotal and many more

Pipelines Concepts



OPENSJAVA'19

Step

Run commands in a container with volumes, env vars, etc

Pipeline Resource

Inputs and outputs to tasks and pipelines (git, image, etc)

Task

A list of steps that run sequentially in the same pod

Task Run

An invocation of a task with inputs and outputs

Pipeline

A graph of tasks with inputs and outputs executed in a certain order

Pipeline Run

An invocation of a pipeline with inputs and outputs

Steps

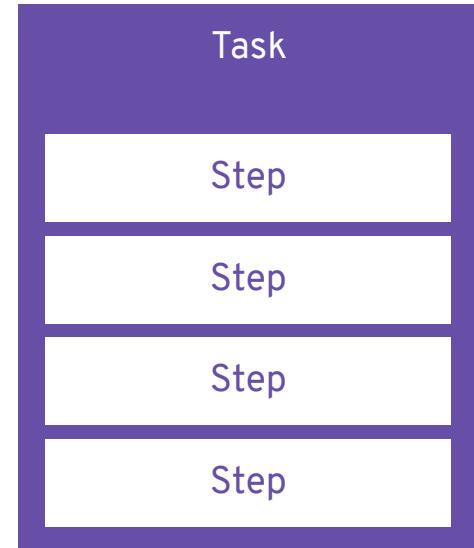


- Run commands in a container
- Kubernetes container spec
 - Env vars
 - Volumes
 - Config maps
 - Secrets
- Part of Task CRD

```
- name: build  
image: maven:3.6.0-jdk-8-slim  
command:  
- /usr/bin/mvn  
args:  
- package
```

Task

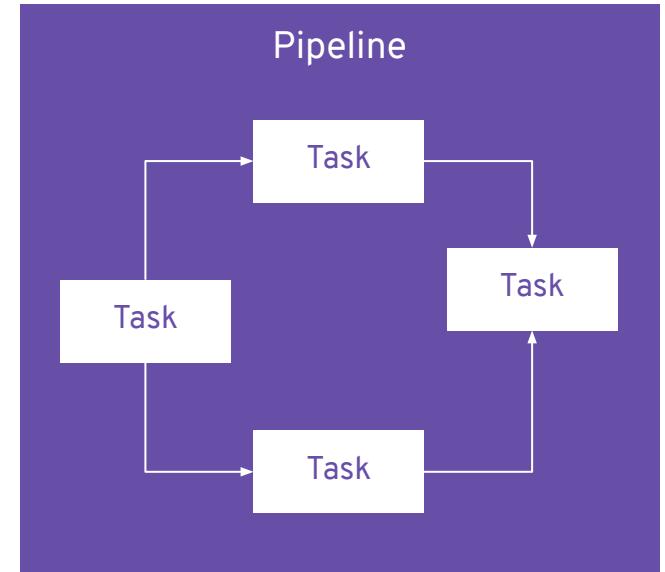
- Defines a unit of work to be executed
- A list of steps run sequentially
- Step containers run in the task pod
- Has inputs, outputs and parameters
- Can run independent of pipelines



Pipeline



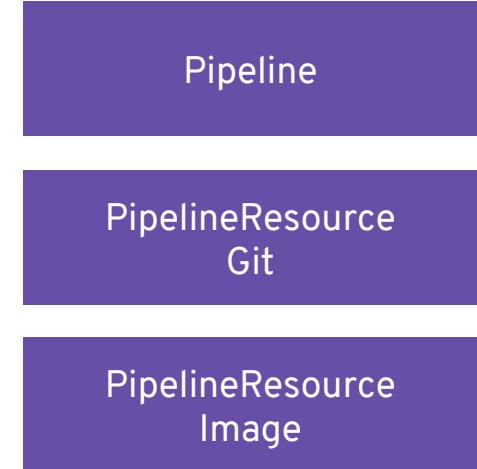
- Combine multiple tasks
- Expresses task order (graph)
- Has inputs and parameters
- Links task inputs and outputs
- Pipeline tasks run on different nodes



PipelineResource



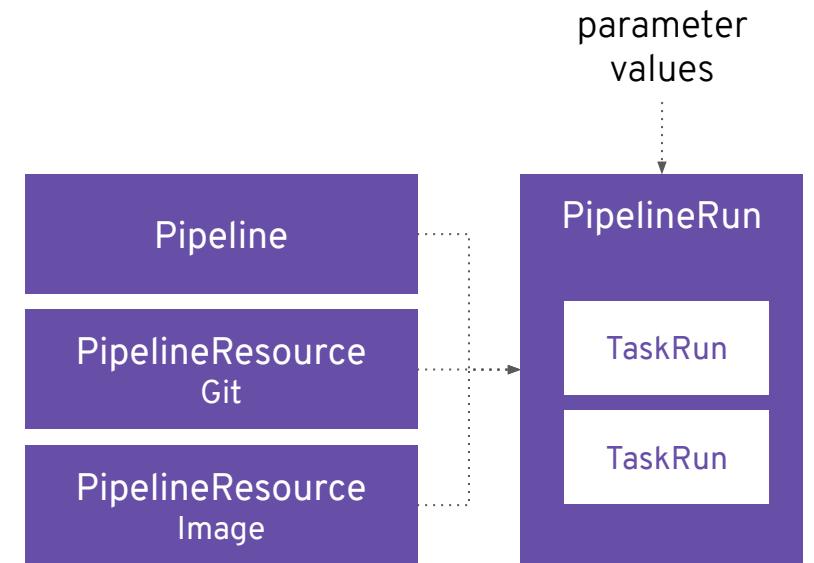
- Inputs and outputs of tasks and pipelines
 - git repository
 - image in a registry
 - ...and more
- Decoupled from pipeline definition
- Reusable across pipelines



TaskRun and PipelineRun



- Runtime CRDs
- Invocation of Task and Pipeline
- Reference tasks and pipelines
- Provide inputs, outputs and params

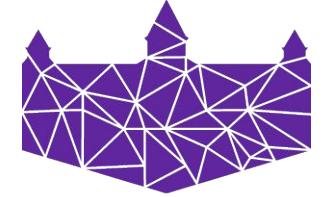


Task Catalog

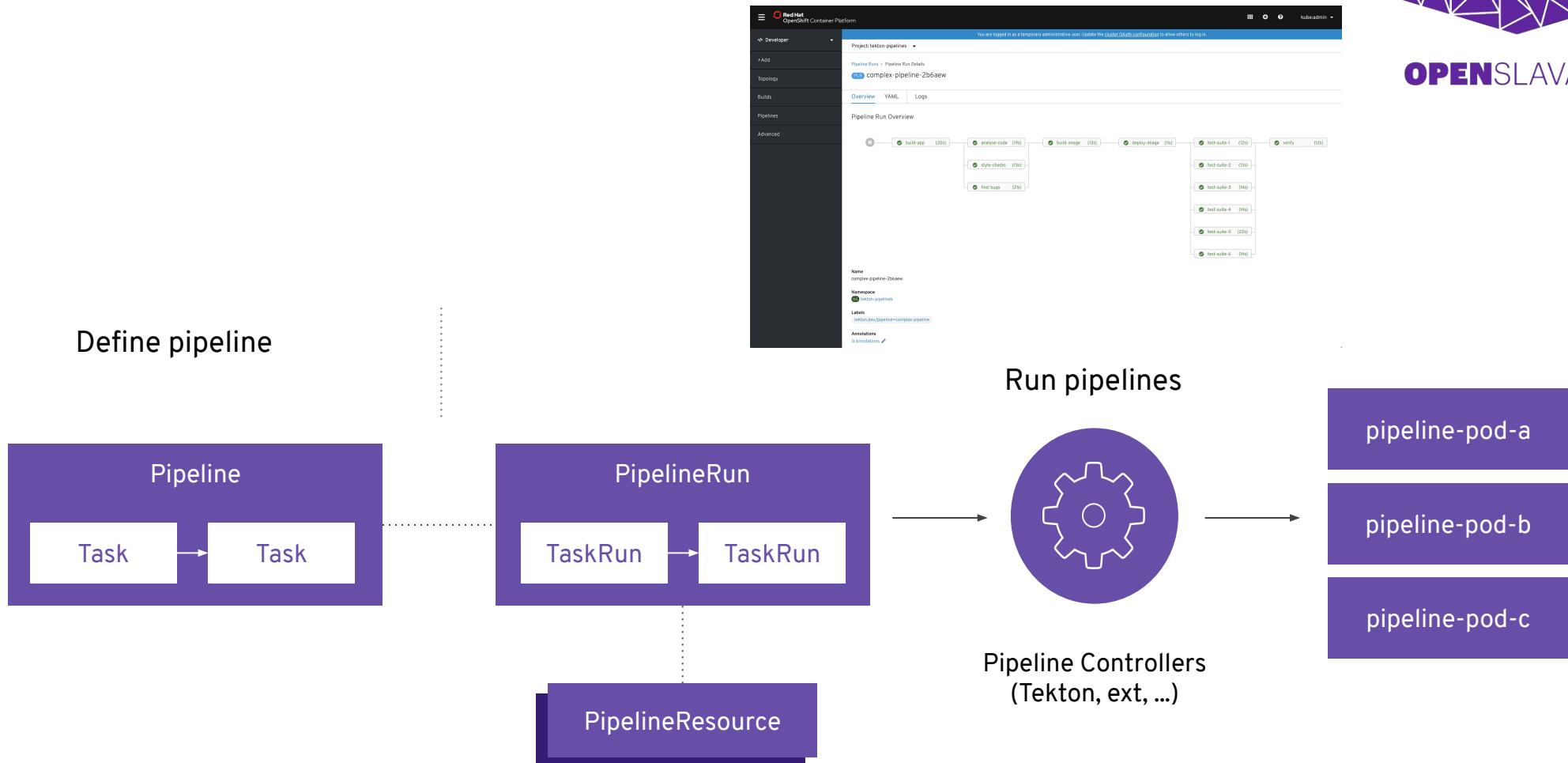


- Catalog of reusable Tasks
 - Image build: buildah, kaniko, jib, buildpacks, etc
 - Source-to-Image: Java, Python, Go, Ruby, etc
 - More to come soon
- Import and compose pipelines
- Available catalogs
 -  [openshift/pipelines-catalog](#)
 -  [tektoncd/catalog](#)

OpenShift Pipelines Architecture

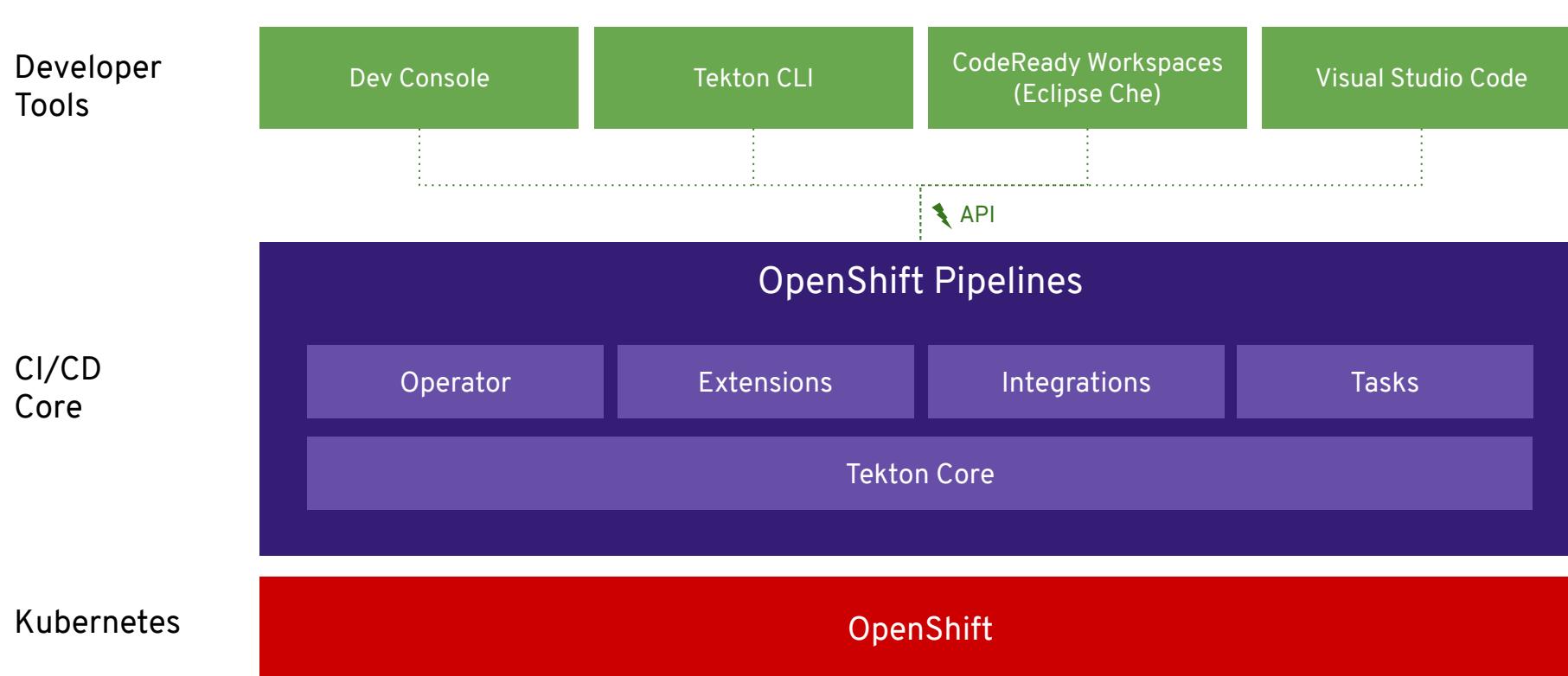


OPENSLAVA'19





OPENSLAVA'19

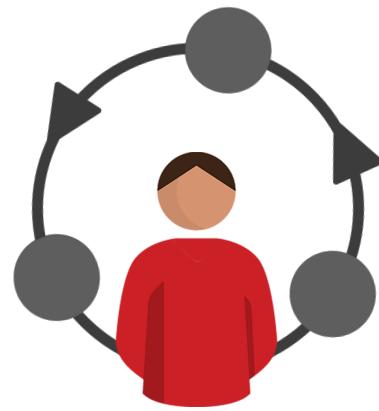


What are Kubernetes Operators?

Stateless is easy,
Stateful is hard

“An Operator is an application-specific controller that extends the Kubernetes API to create, configure, and manage instances of complex stateful applications on behalf of a Kubernetes user. It builds upon the basic Kubernetes resource and controller concepts but includes domain or application-specific knowledge to automate common tasks.”

Encoding and automating Ops knowledge



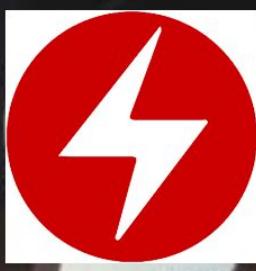
WITHOUT OPERATORS: REACTIVE

Continually checks for anomalies
Alert humans for response
Requires manual change to fix

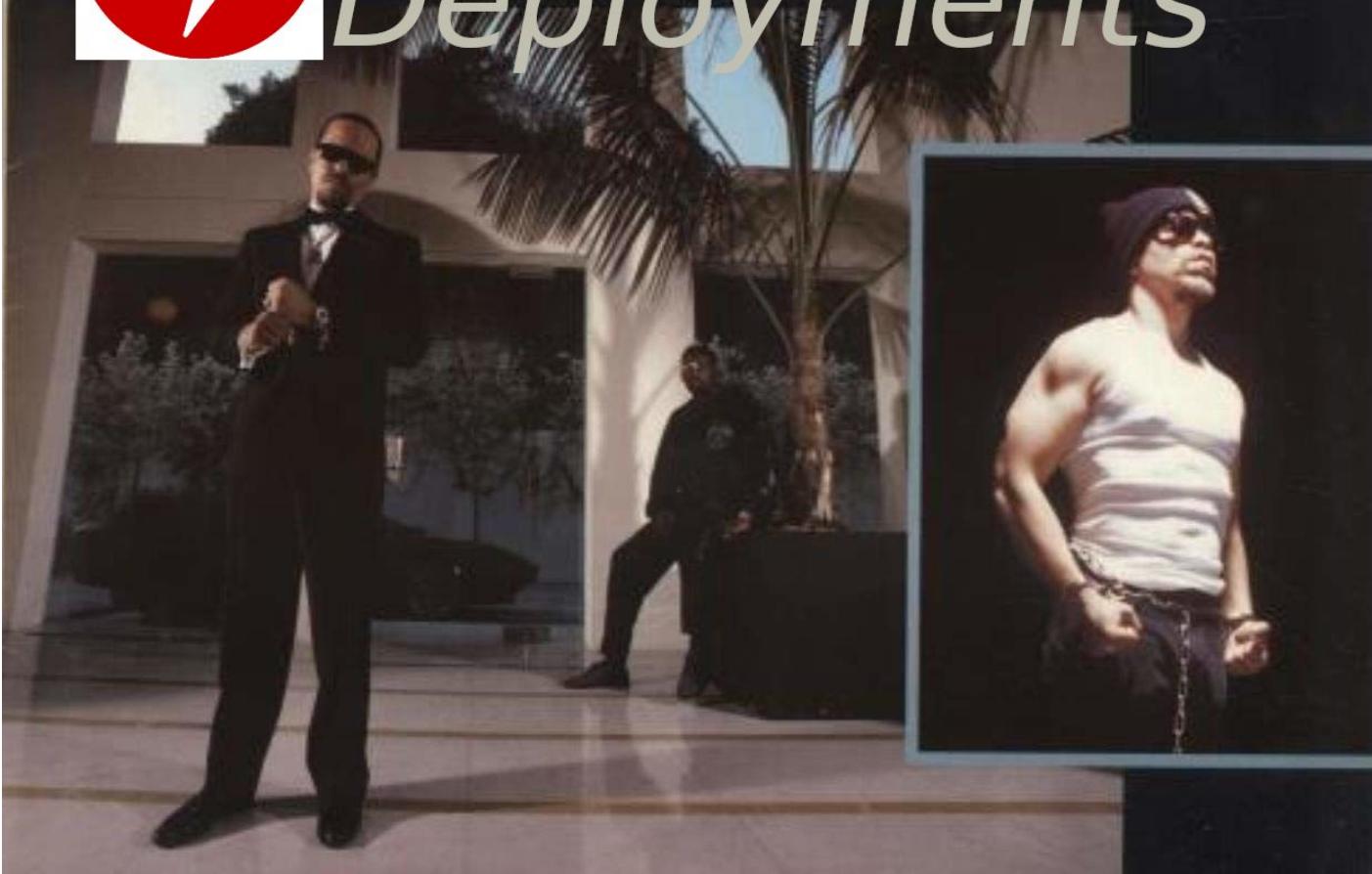


WITH OPERATORS: PROACTIVE

Continually adjusts to optimal state
Automatically acts in milliseconds



ReplicaSets & Deployments

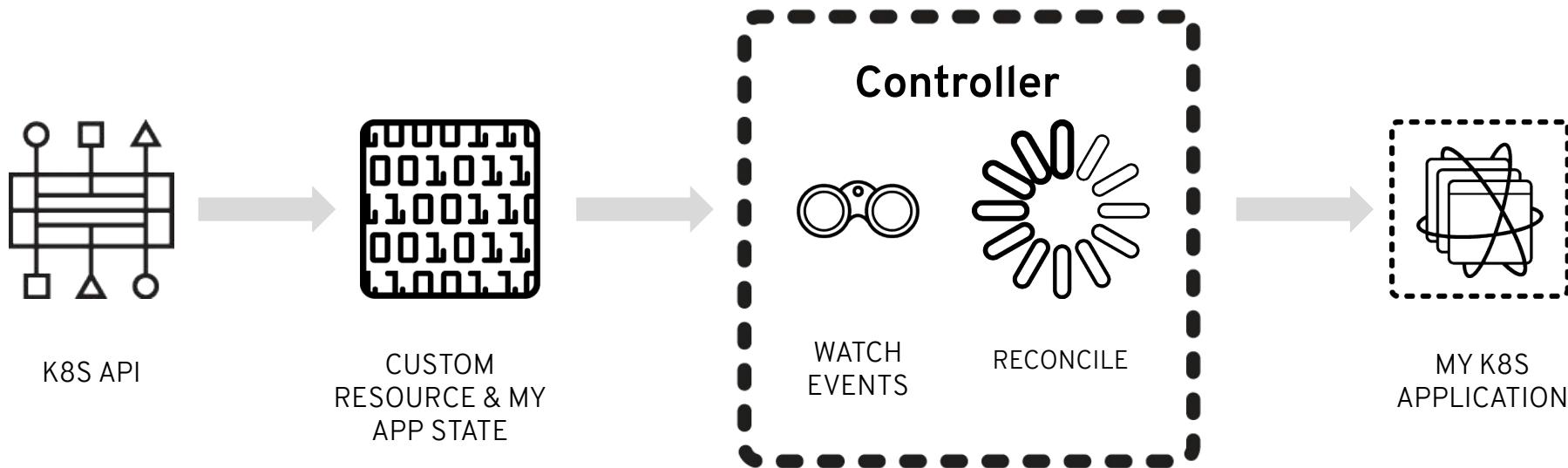


Original Gangster

OG



The Operator Pattern



Operator Framework

An open source toolkit to manage application instances
on Kubernetes in an automated, scalable way



OPERATOR
SDK

Build Operators without
specialized knowledge of
the Kubernetes API



OPERATOR
LIFECYCLE MANAGER

Install, update, and manage
Operators and their
dependencies



OPERATOR
METERING

Enable usage reporting
for Operators

github.com/operator-framework

Operator capability level



Basic Install

Automated application provisioning and configuration management

Seamless Upgrades

Patch and minor version upgrades supported

Phase III

Full Lifecycle

App lifecycle, storage lifecycle (backup, failure recovery)

Phase IV

Deep Insights

Metrics, alerts, log processing and workload analysis

Phase V

Auto Pilot

Horizontal/vertical scaling, auto config tuning, abnormal detection, scheduling tuning



"The Opera Bull"



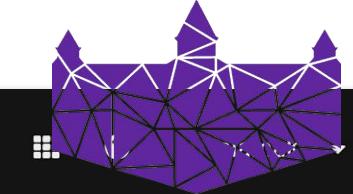
Install Operator from OperatorHub

The screenshot shows the Red Hat OpenShift Container Platform interface. The left sidebar has a dark theme with the Red Hat logo at the top. The 'Catalog' section is expanded, showing 'OperatorHub' is currently selected. The main content area is titled 'OperatorHub' and describes it as a place to discover operators from the Kubernetes community and Red Hat partners. It highlights that operators can be installed to provide optional add-ons and shared services. The 'Integration & Delivery' category is selected, showing 7 items:

- Automation Broker Operator** (Community) provided by Red Hat, Inc. Description: Automation Broker is an implementation of the Open Service Broker API managed by the Operator.
- Business Automation Operator** (Community) provided by Red Hat, Inc. Description: Business Automation Operator can deploy RHPAM/RHDM environments in the form of KieApp objects.
- OpenShift Pipelines Operator** (Community) provided by Red Hat, Inc. Description: OpenShift Pipelines is a cloud-native CI/CD solution for building, testing, and deploying applications.
- Federation** (Community) provided by Red Hat. Description: Federation allows multiple clusters to be federated together.
- Microcks Operator** (Community) provided by Microcks.io. Description: Microcks Operator provides a way to manage and expose APIs.
- Opsmx Spinnaker Operator** (Community) provided by Opsmx. Description: Opsmx Spinnaker Operator integrates Spinnaker with OpenShift.

Pipelines in web console **Developer** perspective

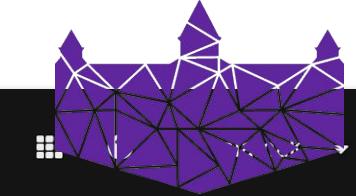
GENERAL DISTRIBUTION



The screenshot displays the Red Hat OpenShift Container Platform interface. The top navigation bar includes the Red Hat logo, 'OpenShift Container Platform', and a purple decorative graphic. On the left, a sidebar menu lists 'Developer' (selected), '+ Add', 'Topology', 'Builds', 'Pipelines' (selected), and 'Advanced'. The main content area shows a 'Project: Project01' dropdown and an 'OPEN SLAVA '19' badge. The 'Pipelines' section has a 'Tech Preview' button and a 'Filter Pipelines by name' input field. Below, a table lists five pipelines: pipeline01, pipeline02, pipeline03, pipeline04, and pipeline05, along with their last run details and status bars. A context menu is open for pipeline05, showing options like 'Restart last run', 'Start', and 'Delete Pipeline'. The bottom right corner features a '☰' icon.

Name	Last Pipeline Run	Last Run Status	Last Task Status	Last Run Started
pipeline01	PRun01A	Succeeded	Green	3 seconds ago
pipeline02	PRun02C	Succeeded	Green	2 minutes
pipeline03	PRun03B	Succeeded	Green	4 minutes
pipeline04	PRun04A	Succeeded	Green	6 minutes
pipeline05	PRun05D	Running	Green (partially blue)	8 minutes ago

Pipelines in web console **Developer** perspective



Red Hat
OpenShift Container Platform

Developer

+ Add

Topology

Builds

Pipelines

Advanced >

Project: Project01 ▾

Pipelines > Pipeline Run Details

PR pipelinerun01a Running

Overview YAML Logs Actions ▾

Tech Preview

Pipeline Run Overview

```
graph LR; A[code compile] --> B[compile & test]; B --> C[unit test]; C --> D[image build]; C -.- E[security check]
```

Name
pipelinerun01a

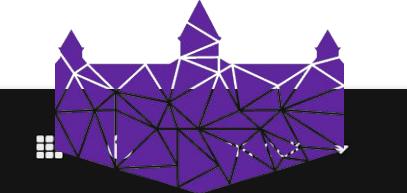
Namespace
NS project01

Labels
`app=dummy-mongo-pod-test` `bap.me/environment=dev`
`bap.me/track=experimental` `bap.me/tier=backend`

Annotations
0 Annotations >

Created At
Aug 9, 4:00 pm

Pipelines in web console **Developer** perspective



The screenshot shows the Red Hat OpenShift web console interface. The left sidebar has a dark theme with the following navigation items:

- Developer (selected)
- + Add
- Topology
- Builds
- Pipelines (selected)
- Advanced >

The main content area is titled "Project: Project01". Below it, the breadcrumb navigation shows "Pipelines > Pipeline Run Details". The pipeline run is identified as "PR pipelinerun01a" and is currently "Running". The "Logs" tab is selected, showing the log output for the "image build" step. The logs are displayed in a monospaced font. At the top right of the logs section, there are "Download" and "Expand" buttons. The overall interface is clean and modern, with a white background and blue highlights for selected items.

Project: Project01

Pipelines > Pipeline Run Details

PR pipelinerun01a (Running)

Actions

Overview YAML Logs

Download | Expand

code compile

compile & test

unit test

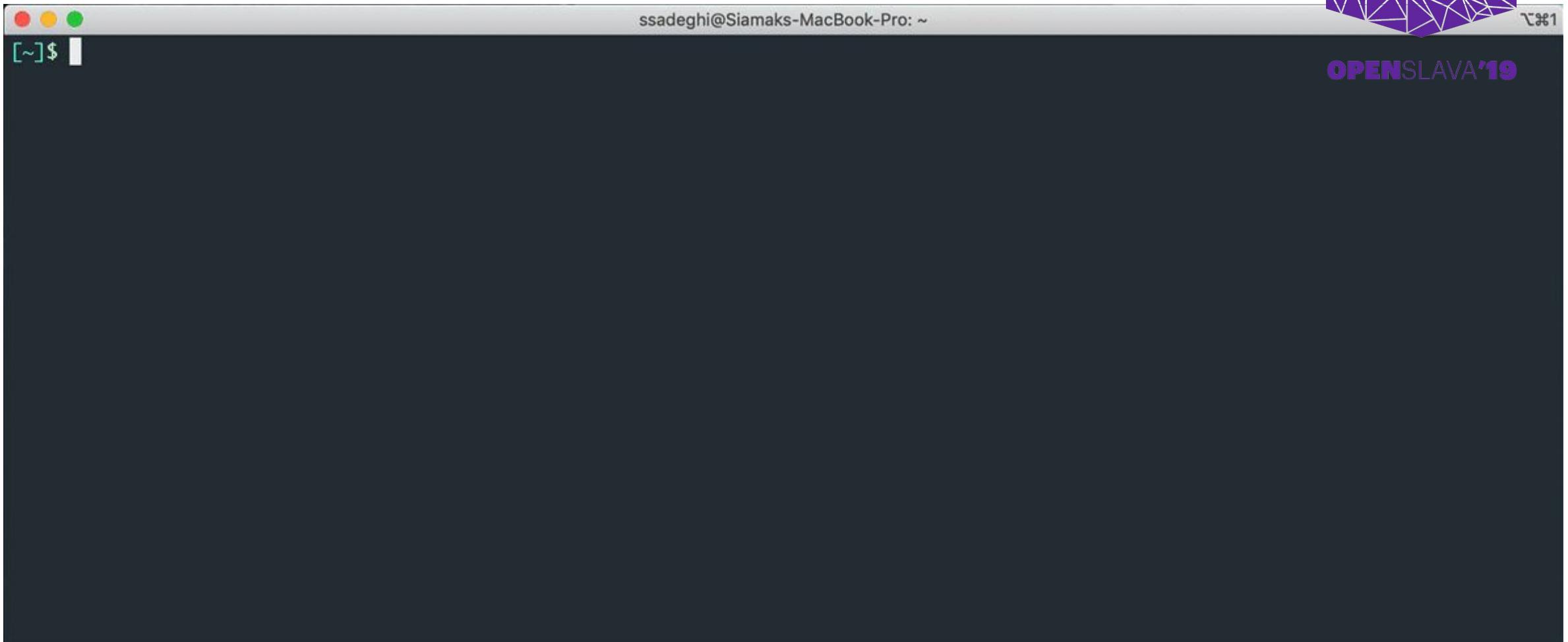
security check

image build

```
[Plack::Sandbox::_2fopt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsg: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_2epsg: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_2epsg: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_2epsg:54] core @2018-08-23

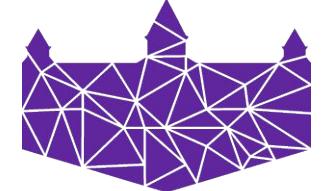
[Plack::Sandbox::_2fopt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsg:54] core @2018-08-23 18:28:53> looking for get /health in extlib/lib/perl5/Dancer2/Core/App.pm l. 36
[Plack::Sandbox::_2fopt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsg:54] core @2018-08-23 18:28:53> Entering hook core.error.init in (eval 306) l. 1
[Plack::Sandbox::_2fopt_2fapp_2droot_2fsrc_2fbin_2fapp_2epsg: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_2epsg:54] core @2018-08-23
```

Manage Pipelines with Tekton CLI

A screenshot of a macOS terminal window. The title bar shows the user's name and host: "ssadeghi@Siamaks-MacBook-Pro: ~". The window is dark-themed. In the top right corner, there is a watermark or logo for "OPENSLAVA'19" in purple text. The main area of the terminal is empty, showing only the command prompt "[~]\$".



GENERAL DISTRIBUTION



OPEN SLAVA '19

LAB TIME!

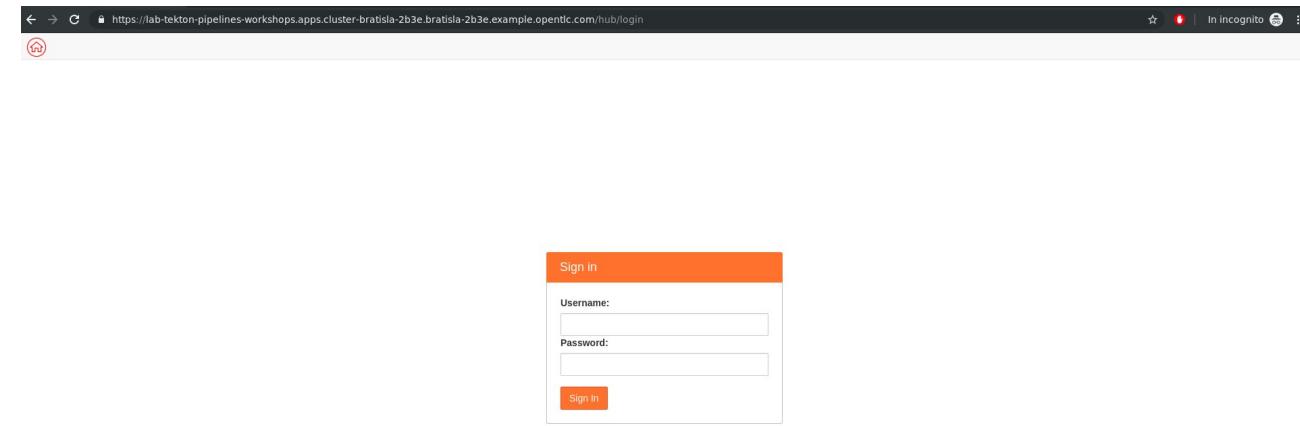
[http://bit.ly/tekton-lab-openslava](http://bit.ly/tekton-labOpenslava)

[http://bit.ly/tekton-lab-openslava-code](http://bit.ly/tekton-labOpenslava-code)

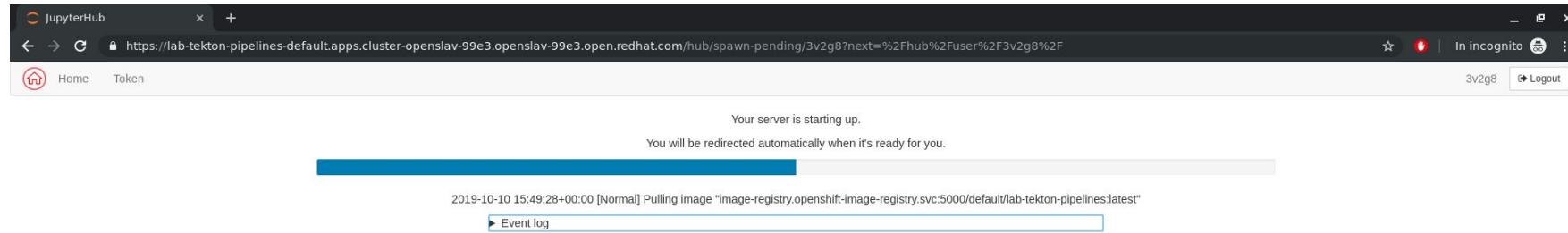


1. Go to the Lab URL

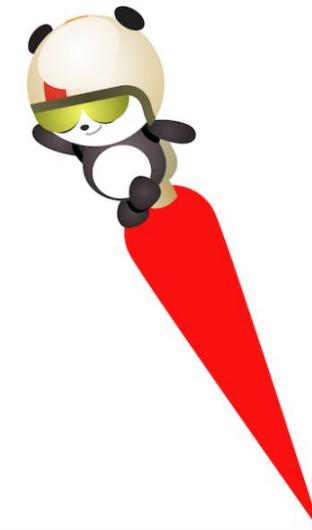
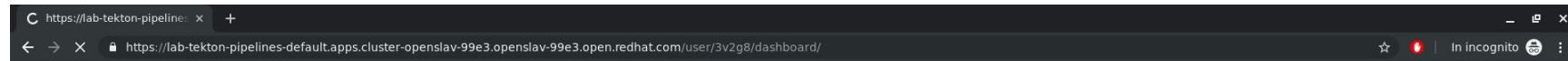
- Insert credentials:
 - User: your email address
 - Pass: **openshift**



2 - Launch the Server



3 - Lab is Launching



4 - Lab is Ready

The screenshot shows two side-by-side browser windows. The left window is a 'Workshop Introduction' page titled 'Workshop Introduction' with a sub-section 'Continuous Integration, Continuous Delivery (CI/CD)'. It contains a detailed explanation of CI/CD pipelines. The right window is the 'Red Hat OpenShift' web console. The top navigation bar shows 'Terminal' and 'Console'. The sidebar on the left has 'Administrator' selected, followed by 'Home', 'Projects', 'Search', 'Explore', 'Events', 'Operators', 'Workloads', 'Networking', 'Storage', 'Builds', and 'Administration'. The main content area is titled 'Projects' and lists a single project: 'lab-tekton-pipelines-default-3v2g8'. This project is shown in a table with columns: Name, Status, Requester, and Labels. The 'Status' column shows 'Active' with a green checkmark. The 'Labels' column shows several labels: 'a...=lab-tekton-pipelines...', 'class=session', 'spawner=learning-portal', and 'user=3v2g8'. A 'Filter by name...' input field is also present.

5 - Hands on!

- From Console, switch to **lab-tekton-pipelines-default-user<X>** project

nodejs-ex also needs a MongoDB database. You can deploy a container with MongoDB to your OpenShift project by running the following command:

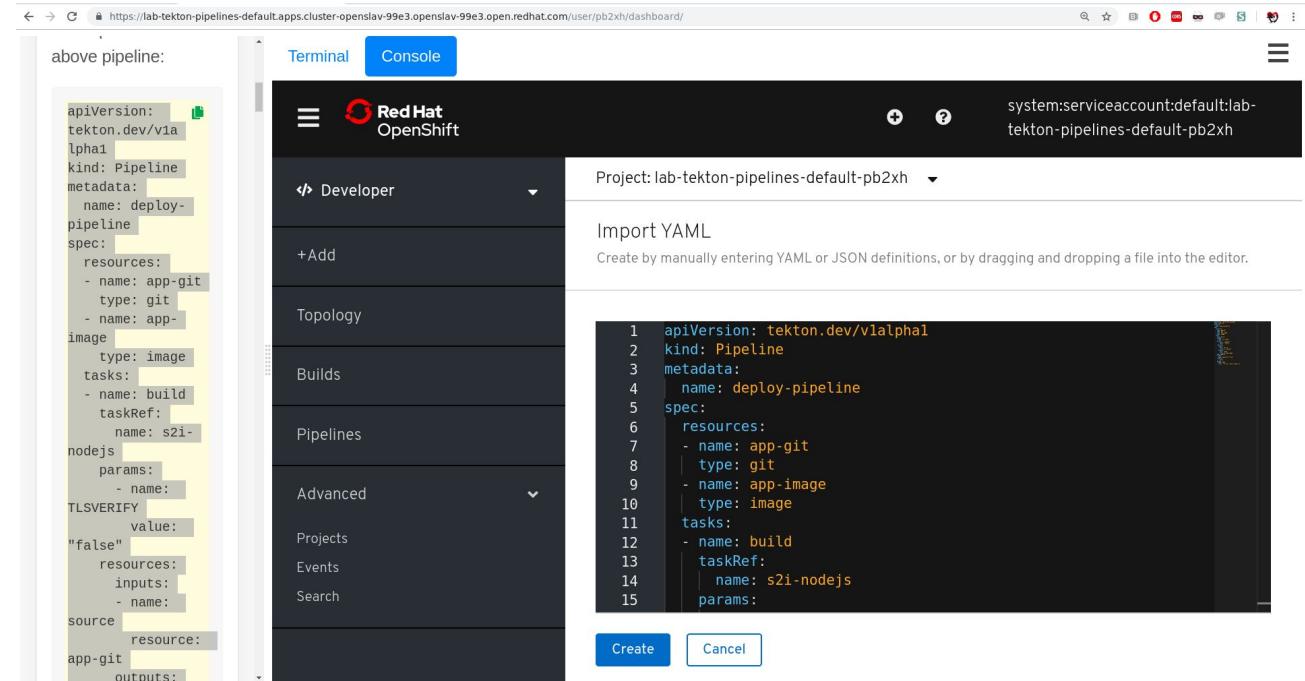
```
oc new-app centos/mongodb-36-centos7 -e MONGODB_USER=admin MONGODB_DATABASE=mo ngo_db MONGODB_PASSWORD=secret MONGODB_ADMIN_PASSWD=super-secret
```

You should see → Success in the output of the command, which verifies the successful deployment of the container image.

The command above

6 - Hands on!

- From Console, import Tekton Pipeline YAML you find in the guide and create your first cloud native pipeline!



The screenshot shows the Red Hat OpenShift web console interface. The title bar indicates the URL is <https://lab-tekton-pipelines-default.apps.cluster-openslav-99e3.openslav-99e3.open.redhat.com/user/pb2xh/dashboard/>. The top navigation bar includes icons for search, star, refresh, and other system functions. The main header says "Red Hat OpenShift" and "Project: lab-tekton-pipelines-default-pb2xh". The left sidebar has sections for Developer, Topology, Builds, Pipelines, Advanced, Projects, Events, and Search. The right panel is titled "Import YAML" with the sub-instruction "Create by manually entering YAML or JSON definitions, or by dragging and dropping a file into the editor." Below this is a code editor containing a Tekton Pipeline YAML definition:

```
1 apiVersion: tekton.dev/v1alpha1
2 kind: Pipeline
3 metadata:
4   name: deploy-pipeline
5 spec:
6   resources:
7     - name: app-git
8       type: git
9     - name: app-image
10    type: image
11   tasks:
12     - name: build
13       taskRef:
14         name: s2i-nodejs
15       params:
16         - name: TLSVERIFY
17           value:
18             "false"
19         resources:
20           inputs:
21             - name: source
22               resource:
23                 app-git
24               outputs:
```

At the bottom of the right panel are "Create" and "Cancel" buttons.

7 - Hands on!

- Follow Pipelines progress from Terminal logs and from Console UI

The screenshot shows the Red Hat OpenShift Pipeline Run Overview interface. On the left, there's a sidebar with options like Developer, +Add, Topology, Builds, Pipelines, Advanced, Projects, Events, and Search. The main area is titled "Pipeline Run Overview" and shows a pipeline named "lab-tekton-pipelines-default-pb2xh". The pipeline has three steps: "generate" (status: 28s), "build" (status: 14s), and "push" (status: 11s). Below the pipeline run, there are sections for Labels (tekton.dev/pipeline=deploy-pipeline), Annotations (0 Annotations), and Created At (2 minutes ago). A terminal window on the left displays the command used to start the pipeline: `tkn pipeline start deploy-pipeline \ -r app- git=nodejs-ex-git \ -r app- image=nodejs-ex-image \ -s pipeline`. A note above the terminal says: "be included in a pipeline run. You can see the git and image pipeline resources you created earlier. You will also notice the -s flag for specifying a service account. This is how you can add your pipeline service account to the pipeline run." Another note below the terminal says: "Run the command below to kick off the pipeline run:".

8 - Hands on!

- Follow Pipelines UI, click on Pipeline Runs and Running tab so see the progress

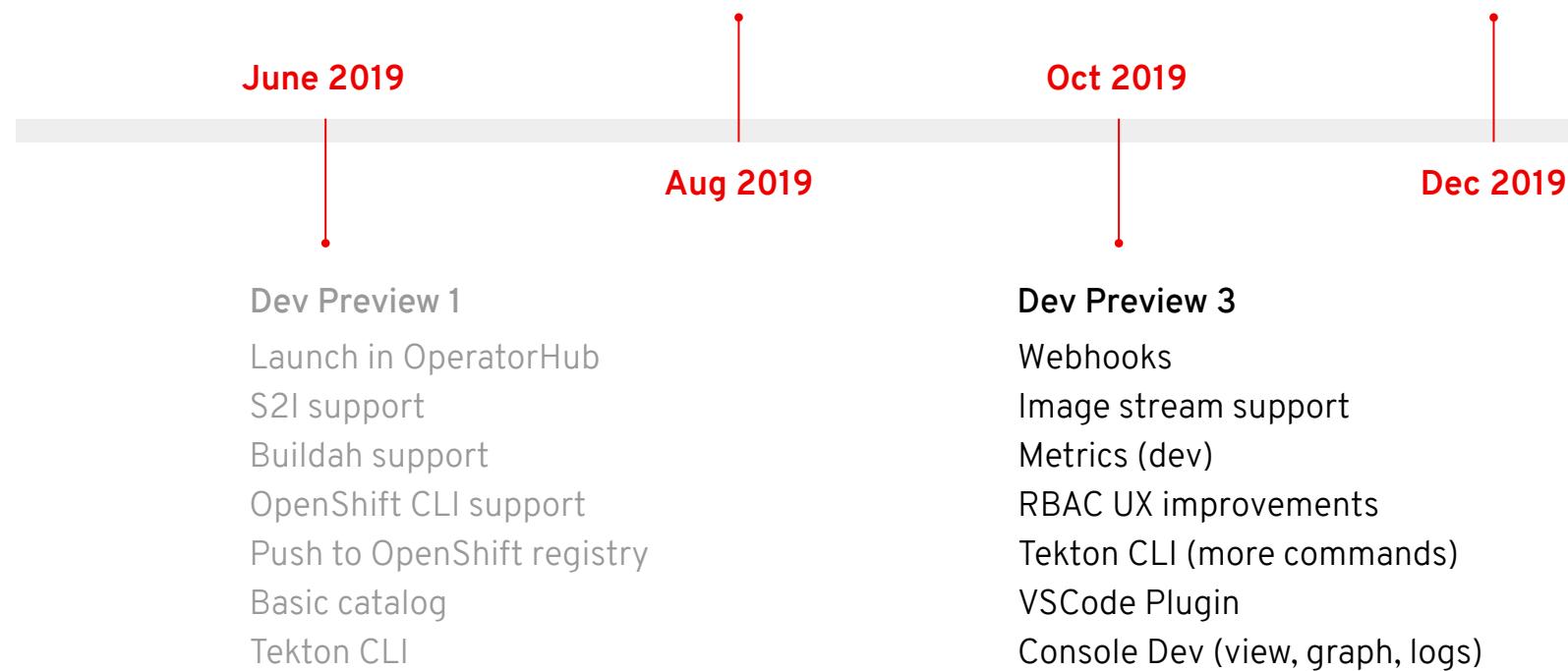
The screenshot shows the Red Hat OpenShift Pipelines UI. On the left, a sidebar menu includes Developer, Topology, Builds, Pipelines (which is selected), Advanced, Projects, Events, and Search. The main content area displays a pipeline named "deploy-pipeline". At the top, there are tabs for Terminal and Console. Below the tabs, there is a terminal window showing logs from a pipeline run. The logs include entries like "[build : generate]", "[build : build]", "[build : push]", "[deploy : oc]", and "[deploy : oc] deploymentconfig.apps.openshift.io/nodejs-ex rolled out". A message at the top of the logs says: "or use the pipeline run using the environment variable to specify the name of the pipeline run: tkn pr logs \$RUN -t". To the right of the logs, there is a section titled "Project: lab-tekton-pipelines-workshops-user1" with tabs for Overview, YAML, Pipeline Runs, Parameters, and Resources. A search bar below these tabs contains the placeholder "Filter by name...". Under the Pipeline Runs tab, there is a table with one row. The row shows a pipeline run named "TLP deploy-pipeline-run-mcwby" that started "a minute ago" and is currently "Running". The status column shows a blue bar indicating progress. Above the table, there are filter buttons for "Complete", "Failed", and "Running", with "Running" being highlighted and circled in red. The URL in the browser address bar is https://lab-tekton-pipelines-workshops.apps.cluster-bratislava-2b3e.bratislava-2b3e.example.opentlc.com/user/user1/dashboard/.

Lab Agenda

- Workshop Introduction
- Workshop Overview
- Tekton Concepts
- Install OpenShift Pipelines Operator
- Verify Workshop Environment
- Pipeline Service Account
- Deploy a Sample Application
- Create Tasks
- Task Resource Definitions
- Create a Pipeline
- Create Pipeline Resources
- Trigger a Pipeline
- Web Console Visualization
- Pipeline Run Breakdown
- Verify Deployment
- Workshop Summary

What's Next?

OpenShift Pipelines Roadmap



Resources

- <https://github.com/openshift/pipelines-tutorial>
- <https://github.com/tektoncd/catalog>
- <https://github.com/openshift-labs/lab-tekton-pipelines/>
- <https://learn.openshift.com/>
- <https://github.com/code-ready/crc>
 - \$ crc start



TEKTON





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://www.linkedin.com/company/red-hat)



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



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



twitter.com/RedHat