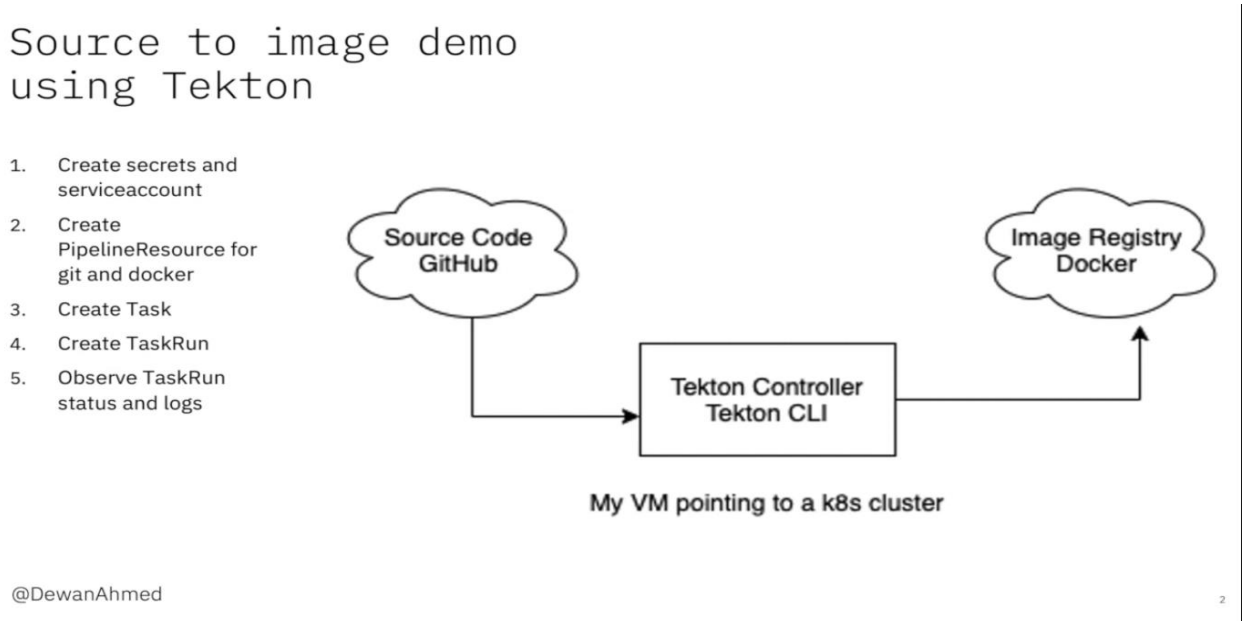
TekTon

TekTon is a powerful yet **flexible Kubernetes- native open source framework for creating Continuous Integration and Continuous Delivery (CI/CD)** system.

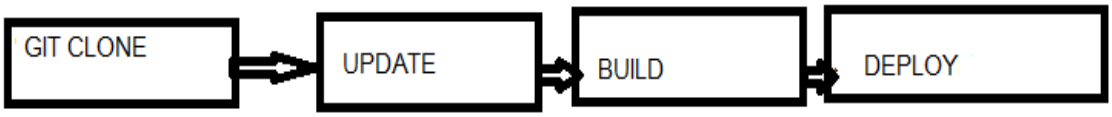
It lets you to **build, test and deploy across multiple cloud providers** (or) on-promises system by abstracting away the underlying implementation details.

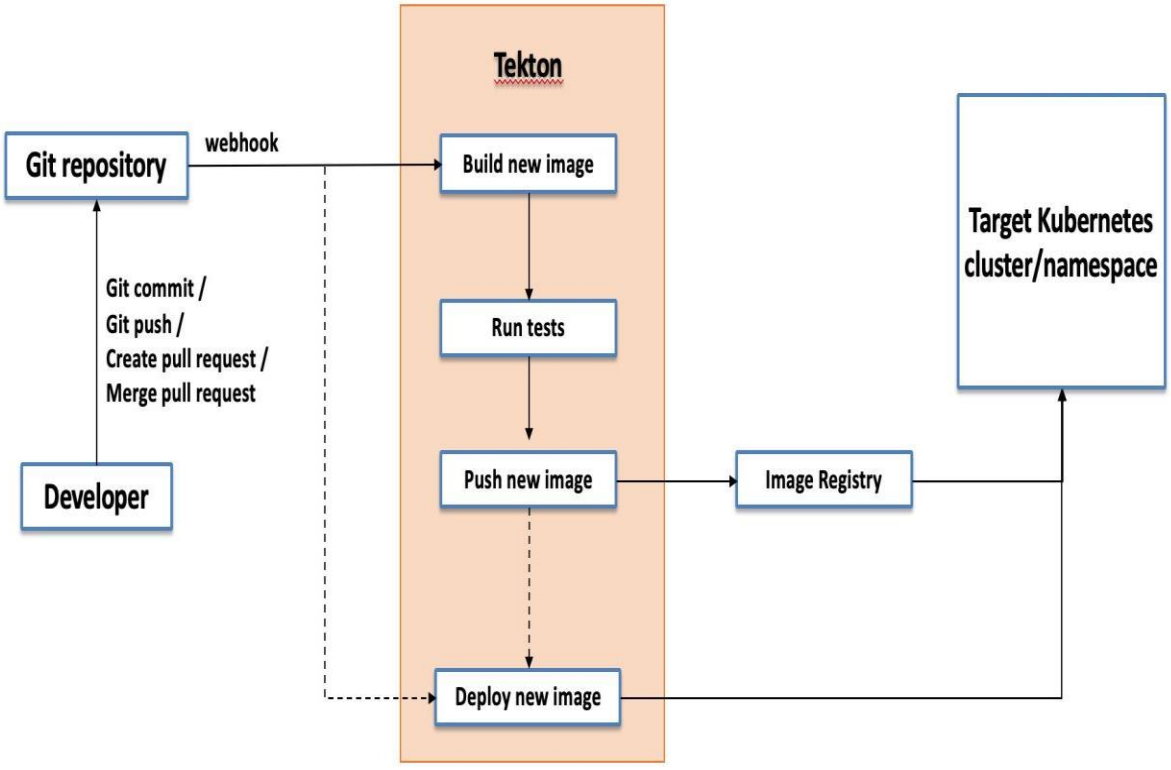


What is TekTon?

- Formerly a part of knative and was called Knative Build pipelines.
- Builds by Google but contributed by cloud Bees, Red Hat, IBM, etc.
- Kubernetes – native.
- Flexible and supports many advanced CI/CD patterns including rolling, blue/green and canary Deployments.
- Written in GO Long.
- CLI and a dashboard.

Tekton Vision





Key components:

- **Pipelines:** Basic building blocks (tasks and pipelines) of a CI/CD workflow.
- **Trigger:** Event trigger for a CI/CD workflow , like a GitHub trigger(I want to trigger a pipeline based on the git commit , Want to do a git push it basically triggers a pipeline.
- **CLI:** Command- line-interface for CI/CD workflow management, view logs, etc.
- **Dashboard:** General- purpose of this web based UI for pipeline.

Step

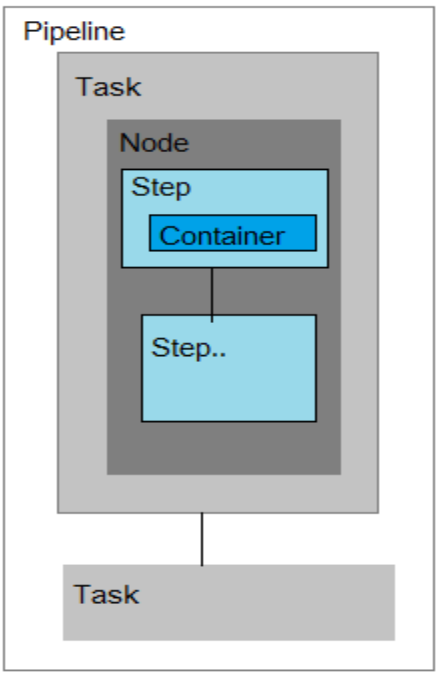
The **most basic of Tekton components are the steps**, essentially a kubernetes container spec which is an existing resource type **lets you define an image and the information you need to run it.**

Task

A **task is composed of one or more steps** (you can have a granular or fine tasks as you wish) and is a **unit of work in a pipeline that achieves a specific goal** (built jar archive, Docker image, test run etc..)

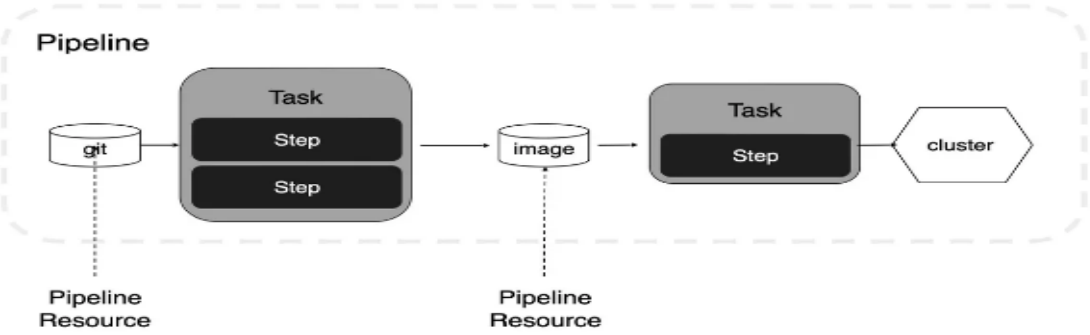
Pipeline

Pipeline is a collection of Tasks that you define and arrange in a specific order of execution as part of your continuous integration flow. Each Task in a pipeline execution as a Pod o your Kubernetes cluster. You can configure various execution conditions to fit your business needs. Pipeline can be both the workflow of part of workflow as you desire. Here's a diagrammatic representation of what a pipeline would achieve in Tekton.



Defining a Pipeline

Let's see how to define a **Continuous Delivery Pipeline** in Tekton. This **pipeline is composed of two tasks**. The first **task clones the project from GitHub**, **builds** a Java project using Maven (it could be any other build tool or even a different language), **creates a container image**, and **pushes it to a container registry**. And a second task **deploys** the services to a Kubernetes cluster.

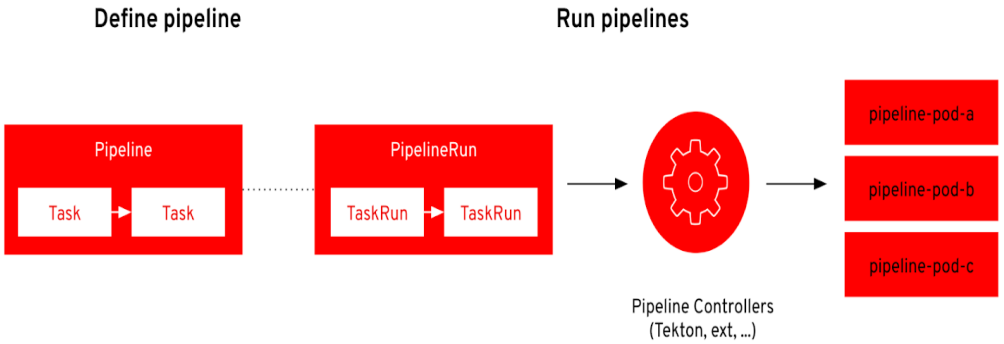


Kubernetes:

Kubernetes is a open source orchestration tool developed by Google for managing micro services or containerized application across a distributed cluster of nodes.

Kubernetes-native CI/CD concepts

By extending Kubernetes/OpenShift with Custom Resource Definitions (CRDs), OpenShift Pipelines makes CI/CD concepts such as a "**pipeline**", a "**task**", a "**step**" natively instantiable so it can use the **scalability**, **security**, **ease of deployment capabilities** of Kubernetes.

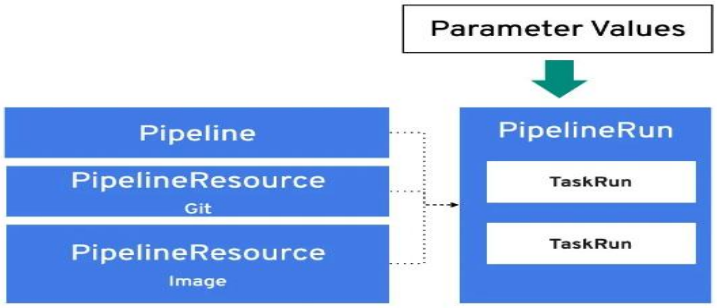


Concepts that define the pipeline

- **Pipeline**: the definition of the pipeline and the **Tasks** that it should perform
- **Task**: a reusable, loosely coupled number of steps that perform a specific task (e.g. building a container image).

Concepts that run the pipeline

- **PipelineRun:** The execution and result of running an instance of a pipeline, which includes a number of **TaskRuns**
- **TaskRun:** the execution and result of running an instance of a **Task**.



Tekton defines the following basic Kubernetes Custom Resource Definitions (CRDs) to build a pipeline:

A `PipelineResource` defines referable resources such as source code repositories or container images.
A `Task` defines a list of steps executed in sequential order. A step executes commands within a container. A task is a Kubernetes Pod containing as many containers as steps.
A `TaskRun` instantiates a `Task` for execution with concrete inputs, outputs, and parameters.
A `Pipeline` defines a list of tasks to execute in a particular order.
A `PipelineRun` instantiates a `Pipeline` for execution with concrete inputs, outputs, and parameters. It automatically creates `TaskRun` instances for each `Task`.
A `Task` may be run individually by creating a `TaskRun` object or as a part of a `Pipeline`.

OpenShift Pipelines concepts

Openshift pipeline provide a set of standard Custom Resource Definitions(CRDs) that **act as the building blocks from which you can assemble a CI/CD pipeline for your application**.

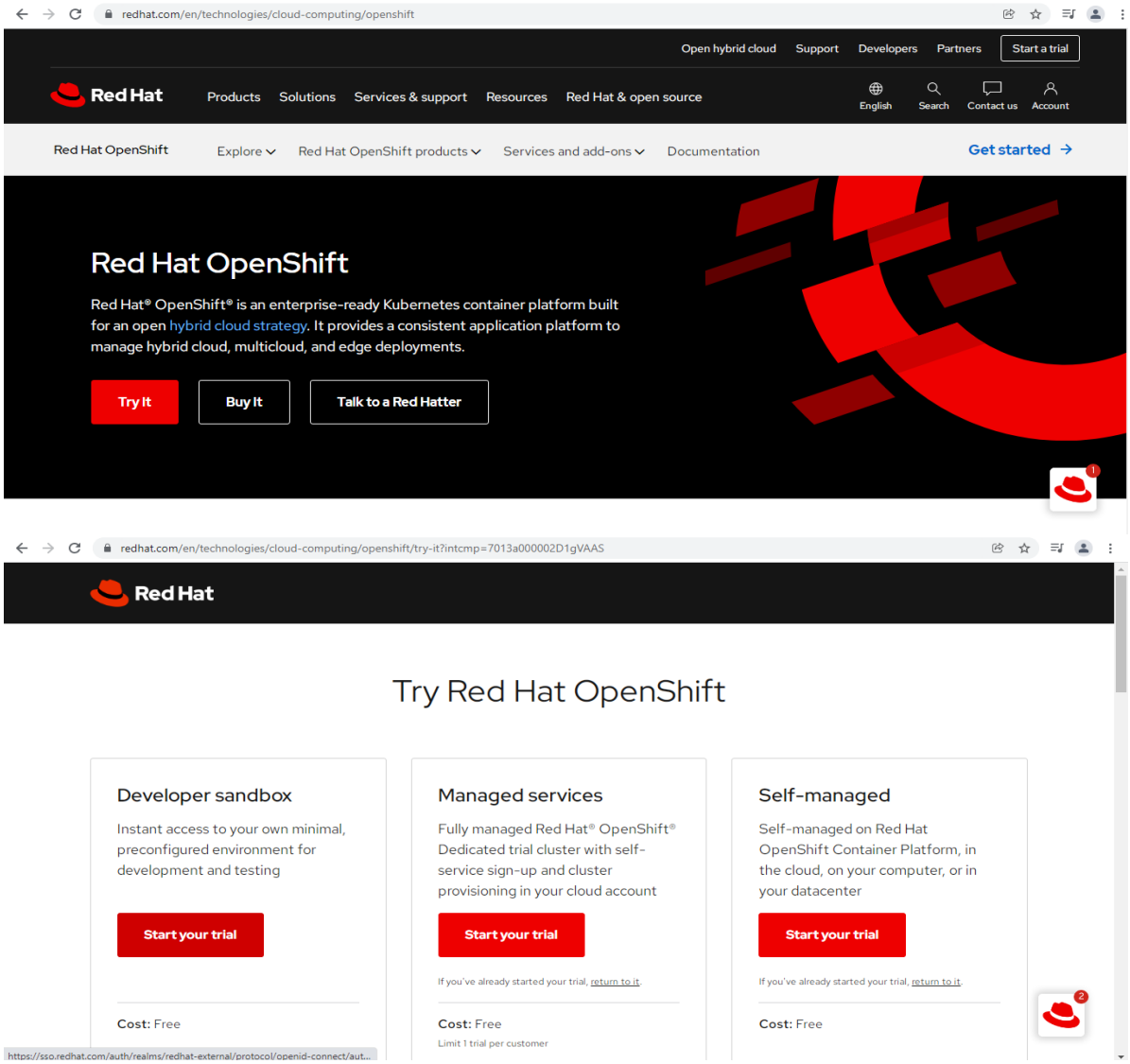
OpenShift Pipelines: Cloud-Native Continuous Integration

Tekton is the core of OpenShiftPipelines and provides a Kubernetes-native framework for creating pipelines that automate the delivery of applications and run native as pods on the cluster. Tekton is built on top of Kubernetes concepts, an operational model that significantly reduces the operational overhead of continuous integration infrastructure for organizations when combined with the serverless execution model.

OpenShift Pipelines features:

- Standard CI/CD pipeline definition based on Tekton.
- **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 **decentralized** teams.
- Integrated with the OpenShift Developer Console.

Screenshot:


redhat.com/en/technologies/cloud-computing/openshift
Open hybrid cloud
Support
Developers
Partners
Start a trial
Red Hat
Products
Solutions
Services & support
Resources
Red Hat & open source
English
Search
Contact us
Account
Red Hat OpenShift
Explore
Red Hat OpenShift products
Services and add-ons
Documentation
Get started
Red Hat OpenShift
Red Hat® OpenShift® is an enterprise-ready Kubernetes container platform built for an open hybrid cloud strategy. It provides a consistent application platform to manage hybrid cloud, multicloud, and edge deployments.
Try It
Buy It
Talk to a Red Hatter
redhat.com/en/technologies/cloud-computing/openshift/try-it?intcmp=7013a000002D1gVAAS
Red Hat
Try Red Hat OpenShift
Developer sandbox
Instant access to your own minimal, preconfigured environment for development and testing
Start your trial
Cost: Free
Managed services
Fully managed Red Hat® OpenShift® Dedicated trial cluster with self-service sign-up and cluster provisioning in your cloud account
Start your trial
If you've already started your trial, return to it.
Cost: Free
Limit 1 trial per customer
Self-managed
Self-managed on Red Hat OpenShift Container Platform, in the cloud, on your computer, or in your datacenter
Start your trial
If you've already started your trial, return to it.
Cost: Free
https://sso.redhat.com/auth/realms/redhat-external/protocol/openid-connect/aut...

developers.redhat.com/developer-sandbox

Welcome to the Developer Sandbox for Red Hat OpenShift

Access Red Hat's products and technologies without setup or configuration, and start developing quicker than ever before with our new sandbox environments for Red Hat OpenShift and CodeReady Workspaces. Try your hand at the technologies with our library of activities as well.

Overview | Get started in the Sandbox | Sandbox IDE | Sandbox activities

What is the Sandbox?

The sandbox provides you with a private OpenShift environment in a shared, multi-tenant OpenShift cluster that is pre-configured with a set of developer tools. You can easily create containers from your source code or Dockerfile, build new applications using the samples and stacks provided, add services such as databases from our templates catalog, deploy Helm charts, and much more. Discover the rich capabilities of the full developer experience on OpenShift with the sandbox.

Get started in the Sandbox | Learn about our IDE

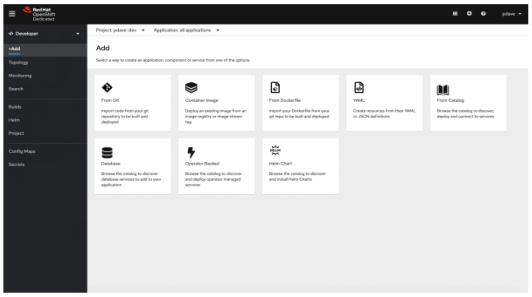
https://developers.redhat.com/developer-sandbox/get-started

developers.redhat.com/developer-sandbox/get-started

Overview | Get started in the Sandbox | Sandbox IDE | Sandbox activities

Start your OpenShift experience in four simple steps

If you're exploring how to run your code in containers, our Developer Sandbox makes it simple. Not only can you easily deploy your application from a Git repo, you can also set up a cloud IDE for your entire team. Follow these four steps to quickly get started:

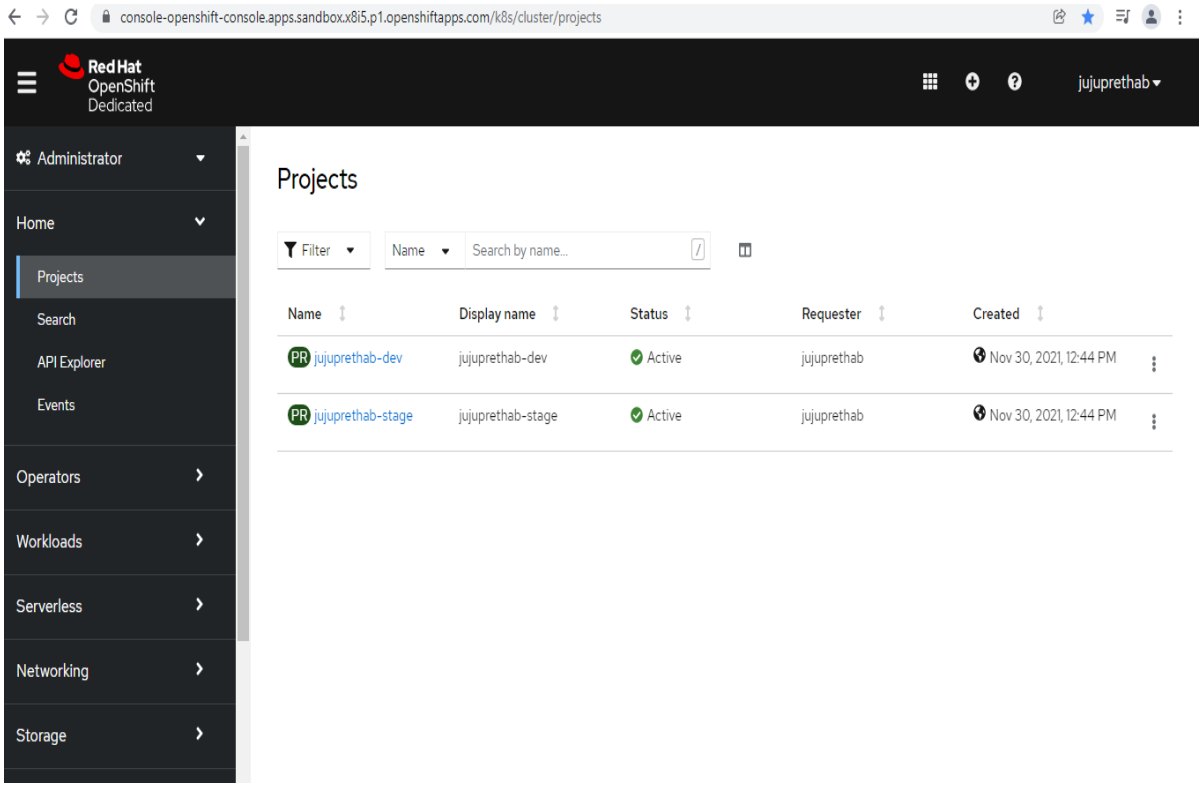


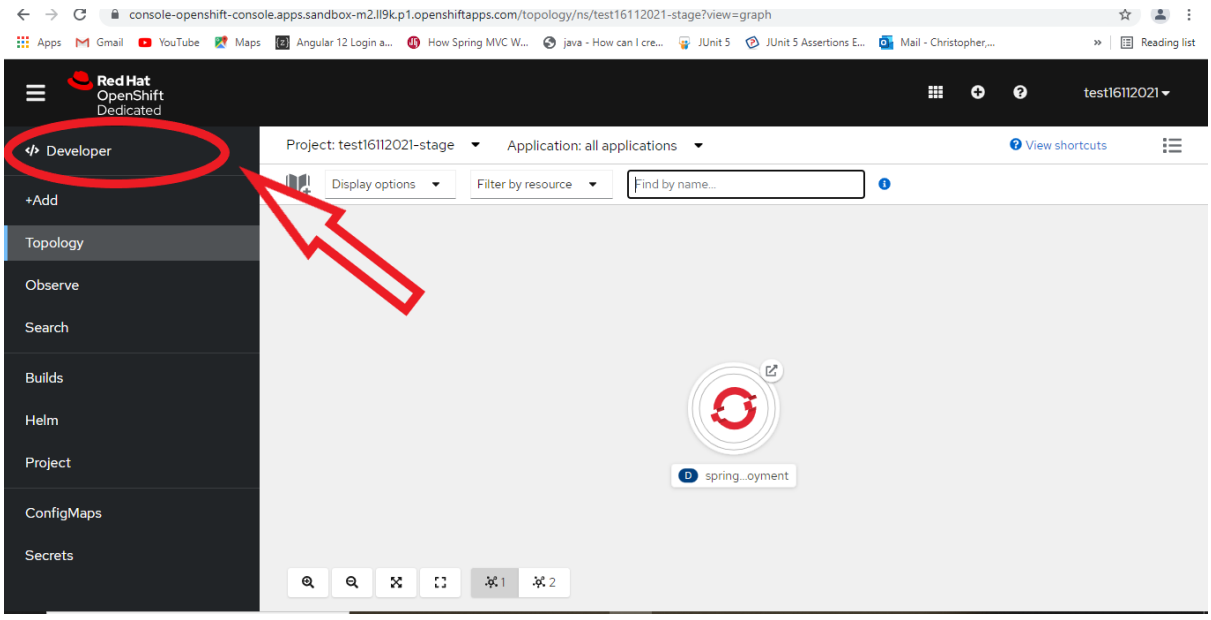
You're ready to get started!

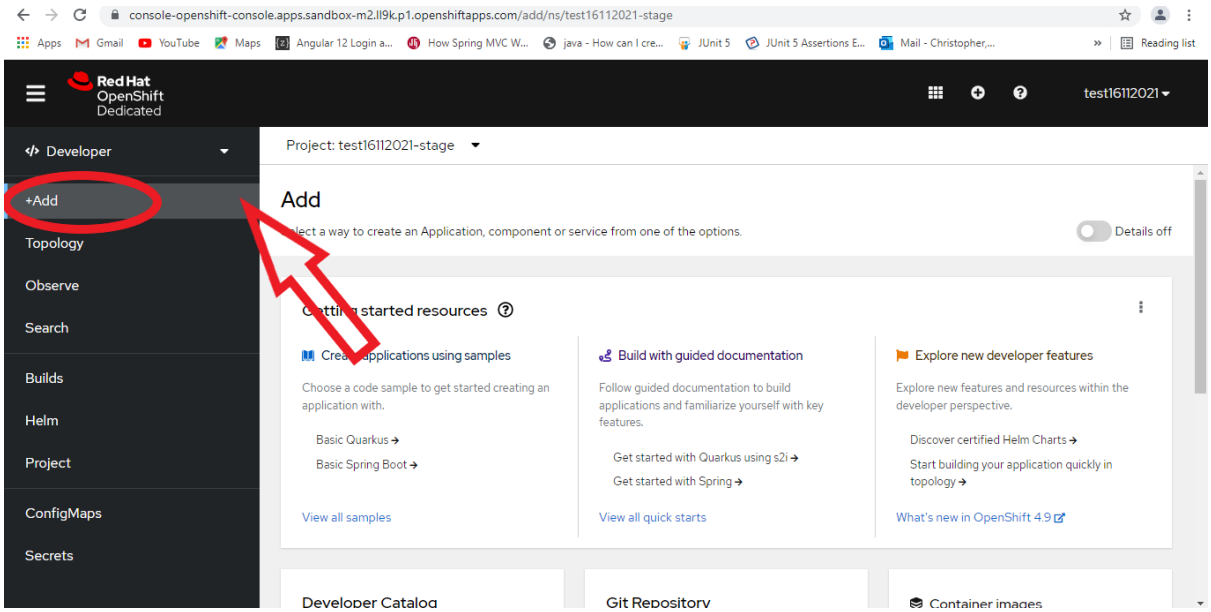
To launch your sandbox, click the button below and select DevSandbox when prompted.

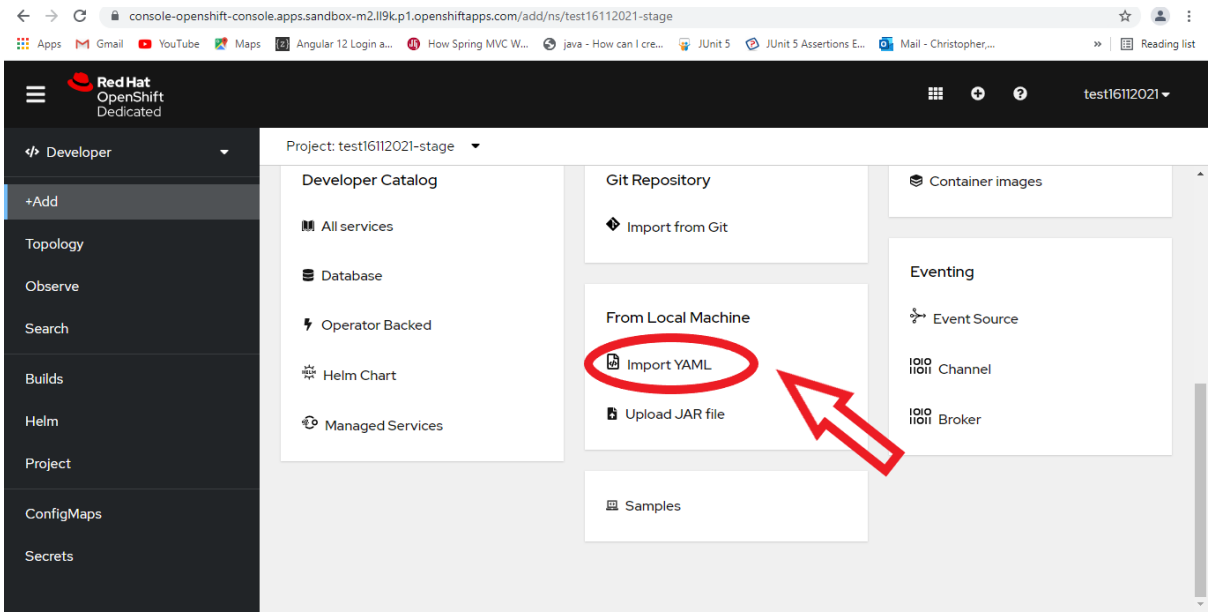
Start using your sandbox

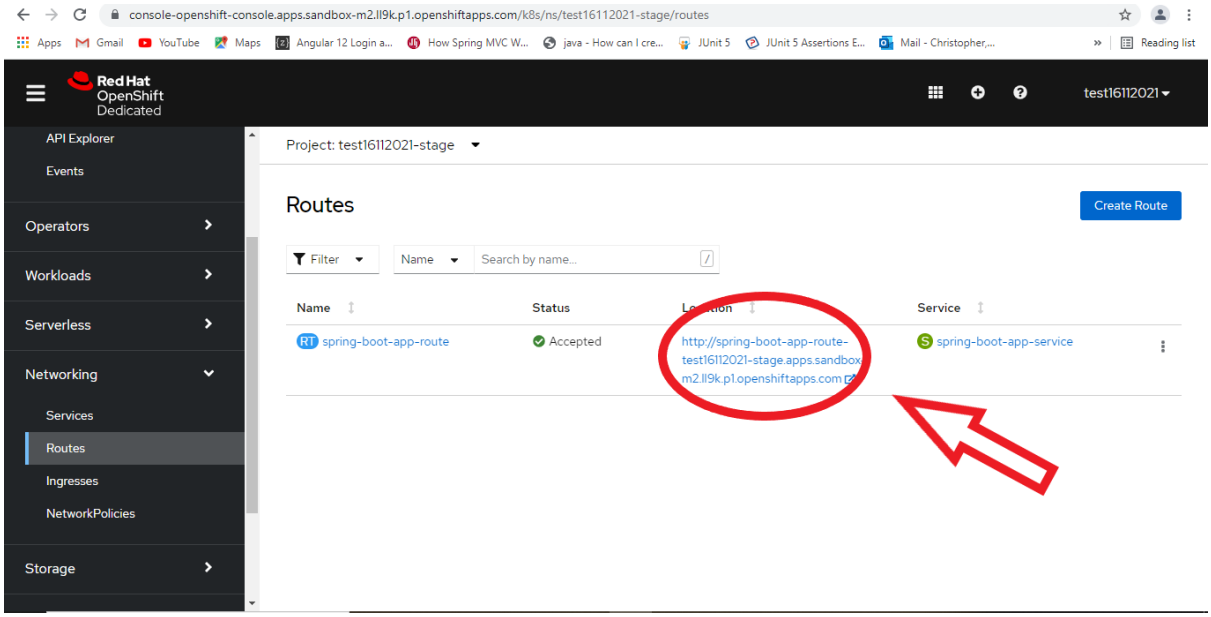
1. Provision your free Red Hat OpenShift development cluster
2. Deploy a sample application or bring your own repo
3. Edit code from the integrated Eclipse Che–based cloud IDE


console-openshift-console.apps.sandbox.x8i5.p1.openshiftapps.com/k8s/cluster/projects
Red Hat OpenShift Dedicated
jujuprethab
Administrator
Home
Projects
Search
API Explorer
Events
Operators
Workloads
Serverless
Networking
Storage
Projects
Filter
Name
Search by name...
Name
Display name
Status
Requester
Created
jujuprethab-dev
jujuprethab-dev
Active
jujuprethab
Nov 30, 2021, 12:44 PM
jujuprethab-stage
jujuprethab-stage
Active
jujuprethab
Nov 30, 2021, 12:44 PM


console-openshift-console.apps.sandbox-m2.ll9k.p1.openshiftapps.com/topology/ns/test16112021-stage?view=graph
Red Hat OpenShift Dedicated
test16112021
Developer
+Add
Topology
Observe
Search
Builds
Helm
Project
ConfigMaps
Secrets
Project: test16112021-stage
Application: all applications
View shortcuts
Display options
Filter by resource
Find by name...
spring...oyment


console-openshift-console.apps.sandbox-m2.ll9k.p1.openshiftapps.com/add/ns/test16112021-stage
Red Hat OpenShift Dedicated
test16112021
Developer
Project: test16112021-stage
+Add
Topology
Observe
Search
Builds
Helm
Project
ConfigMaps
Secrets
Add
Details off
started resources
Create applications using samples
Choose a code sample to get started creating an application with.
Basic Quarkus
Basic Spring Boot
View all samples
Build with guided documentation
Follow guided documentation to build applications and familiarize yourself with key features.
Get started with Quarkus using s2i
Get started with Spring
View all quick starts
Explore new developer features
Explore new features and resources within the developer perspective.
Discover certified Helm Charts
Start building your application quickly in topology
What's new in OpenShift 4.9
Developer Catalog
Git Repository
Container images


console-openshift-console.apps.sandbox-m2.ll9k.p1.openshiftapps.com/add/ns/test16112021-stage
Red Hat OpenShift Dedicated
test16112021
Developer
Project: test16112021-stage
+Add
Topology
Observe
Search
Builds
Helm
Project
ConfigMaps
Secrets
Developer Catalog
All services
Database
Operator Backed
Helm Chart
Managed Services
Git Repository
Import from Git
From Local Machine
Import YAML
Upload JAR file
Samples
Container images
Eventing
Event Source
Channel
Broker


console-openshift-console.apps.sandbox-m2.ll9k.p1.openshiftapps.com/k8s/ns/test16112021-stage/routes
Red Hat OpenShift Dedicated
test16112021
API Explorer
Events
Operators
Workloads
Serverless
Networking
Services
Routes
Ingresses
NetworkPolicies
Storage
Project: test16112021-stage
Routes
Create Route
Filter
Name
Search by name...
Name
Status
Service
spring-boot-app-route
Accepted
http://spring-boot-app-route-test16112021-stage.apps.sandbox-m2.ll9k.p1.openshiftapps.com
spring-boot-app-service