Quick guide

RH OpenShift on IBM Cloud

And

Tekton Pipelines

Running up a minimal Openshift 4.3.5 beta on IBM Cloud and installing and testing Tekton.

Contents

[IBM Cloud – provision a Red Hat OpenShift on IBM Cloud service 3](#_Toc38299757)

[Accessing the Red Hat OpenShift service 6](#_Toc38299758)

[Installing Tekton 7](#_Toc38299759)

[C:\openshift>oc new-project tekton-pipelines 7](#_Toc38299760)

[C:\openshift>oc adm policy add-scc-to-user anyuid -z tekton-pipelines-controller 7](#_Toc38299761)

[C:\openshift>oc apply --filename https://storage.googleapis.com/tekton-releases/latest/release.yaml 7](#_Toc38299762)

[C:\openshift>oc get pods --namespace tekton-pipelines 8](#_Toc38299763)

[Correcting the paths to the image locations 9](#_Toc38299764)

[C:\openshift>oc get pods --namespace tekton-pipelines 11](#_Toc38299765)

[Test Tekton with an Echo example 11](#_Toc38299766)

[C:\openshift>oc apply -f Tekton-ivt\echo-hello-world-task.yaml 12](#_Toc38299767)

[C:\openshift>oc apply -f Tekton-ivt\echo-hello-world-task-run.yaml 12](#_Toc38299768)

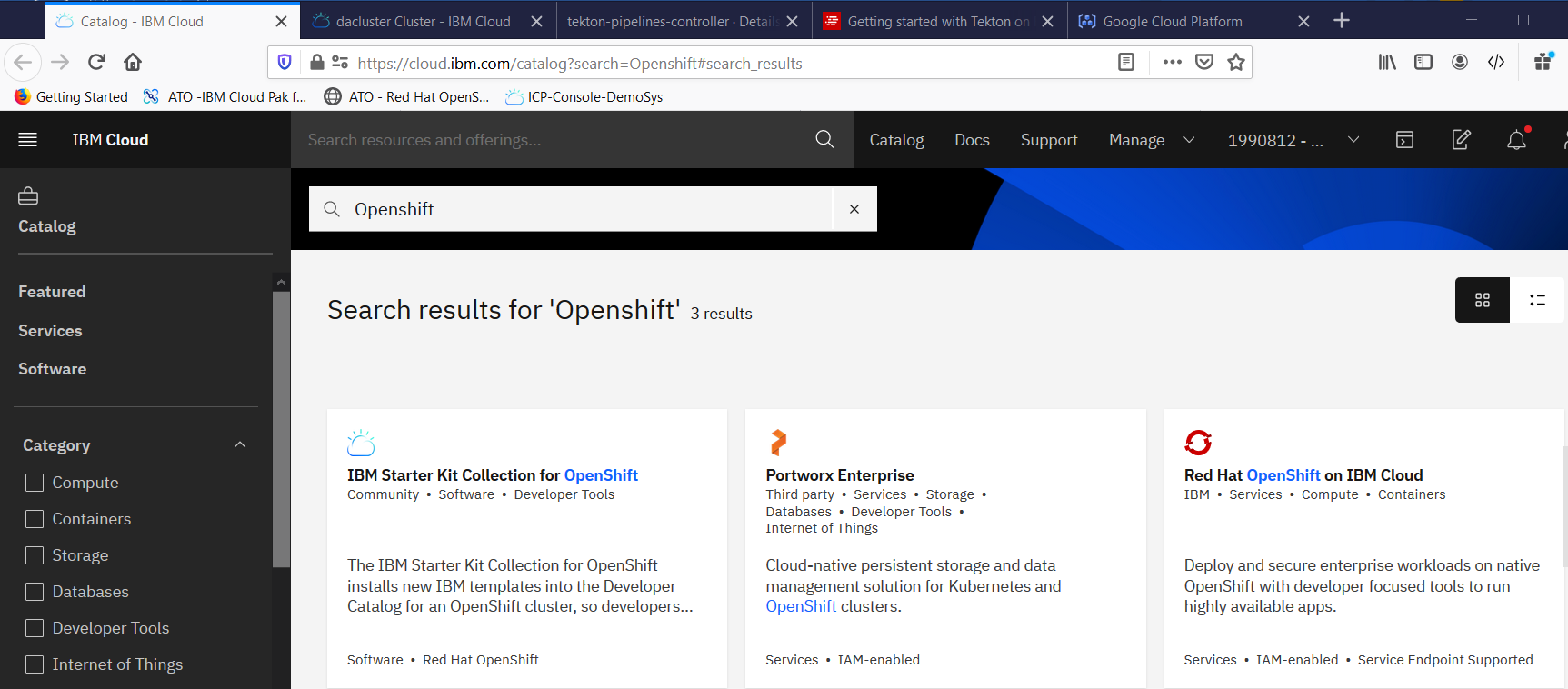
[C:\openshift>oc get taskruns/echo-hello-world-task-run -o yaml 12](#_Toc38299769)

## IBM Cloud – provision a Red Hat OpenShift on IBM Cloud service

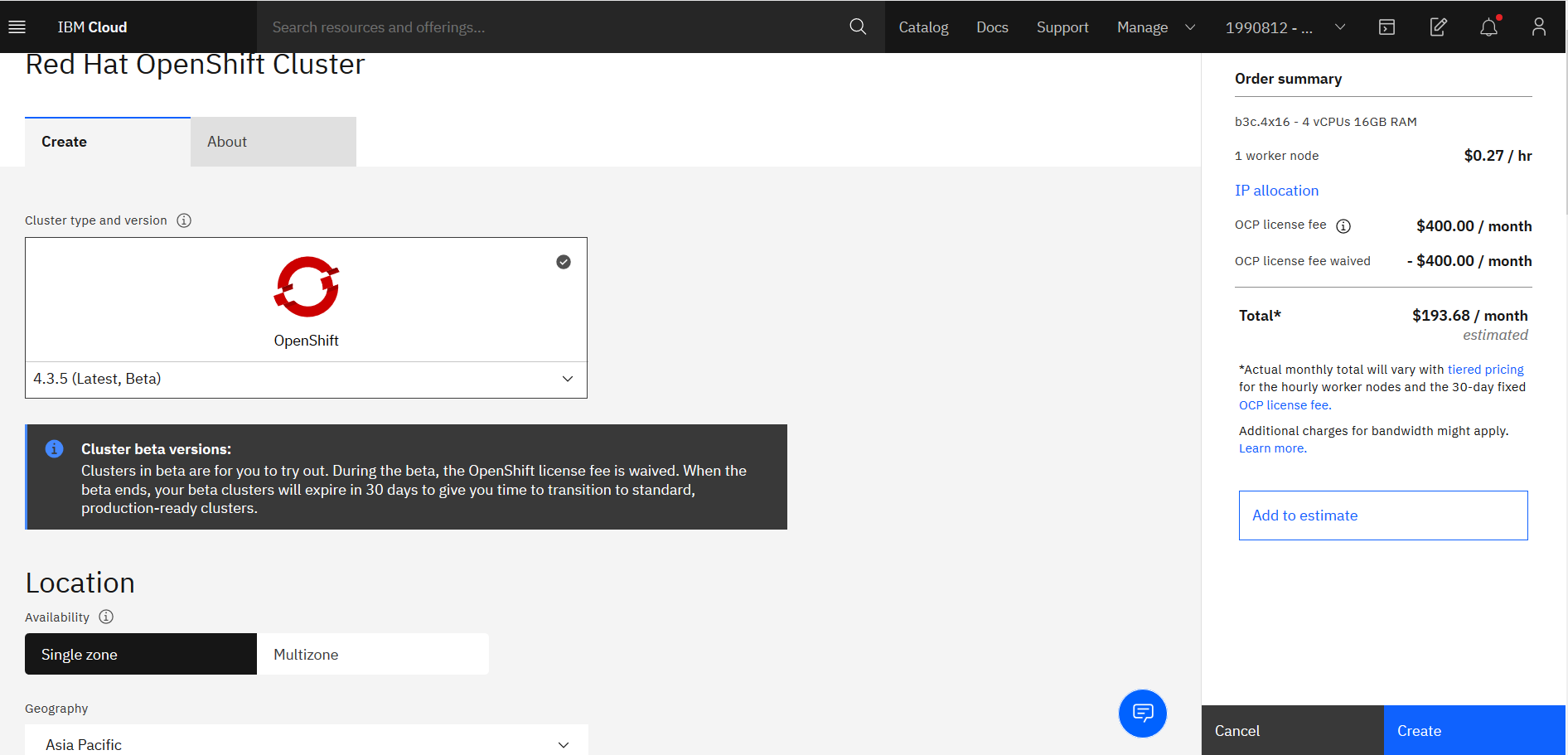
Log into IBM cloud

Goto the catalog and type openshift in the search

Select RedHat Openshift on IBM cloud



Select 4.3.5 Beta as there is no RHOCP license

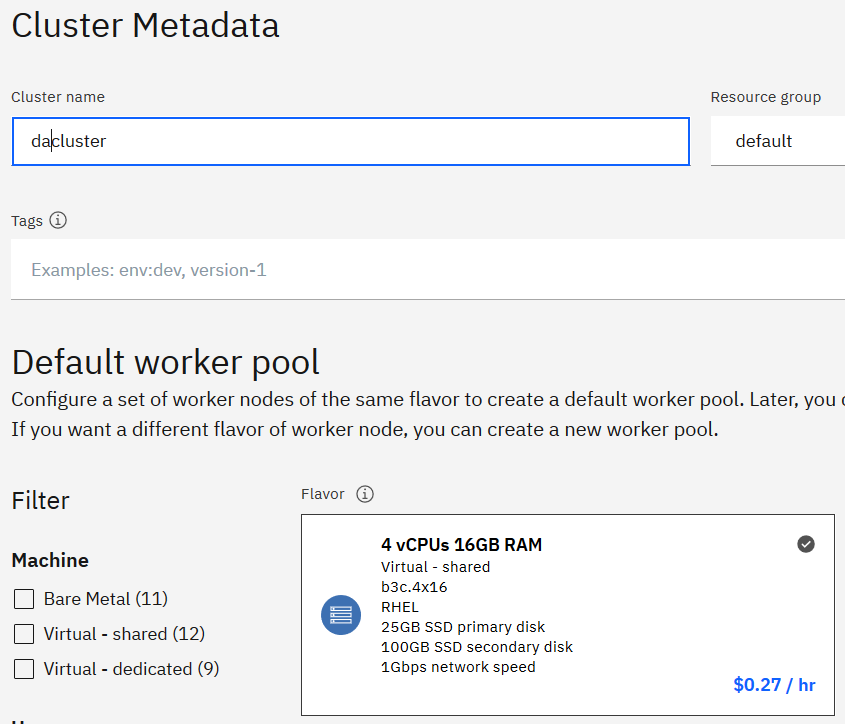


Switch to Single Zone

Asia Pacific

Sydney 01

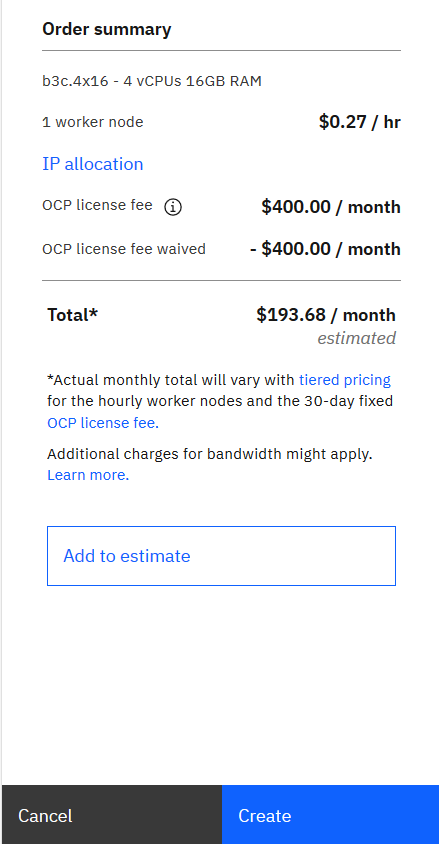
Name your cluster and select the small worker node size



Drop the worker node count from 3 to 1

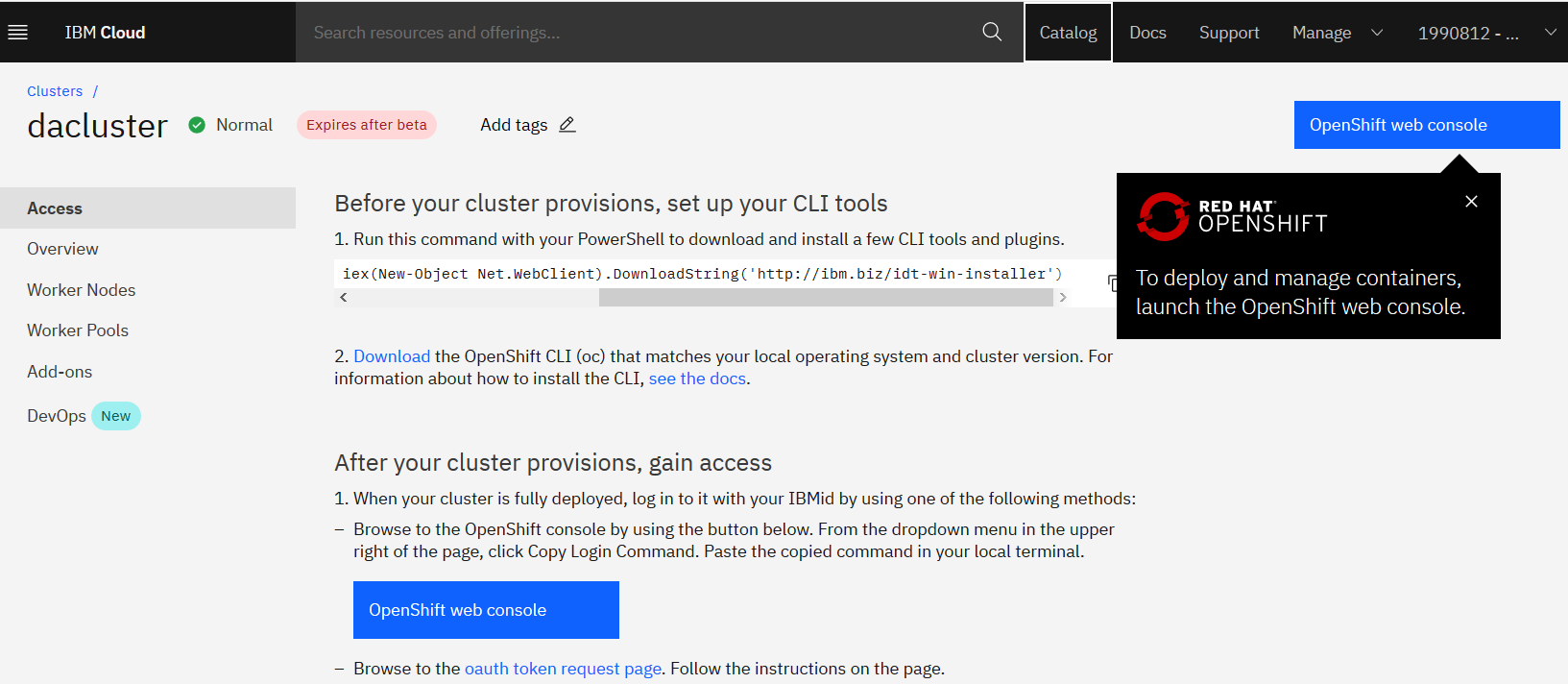


Price for 30 days is low as there is no Openshift license for the Beta version service. **This is point in time.**

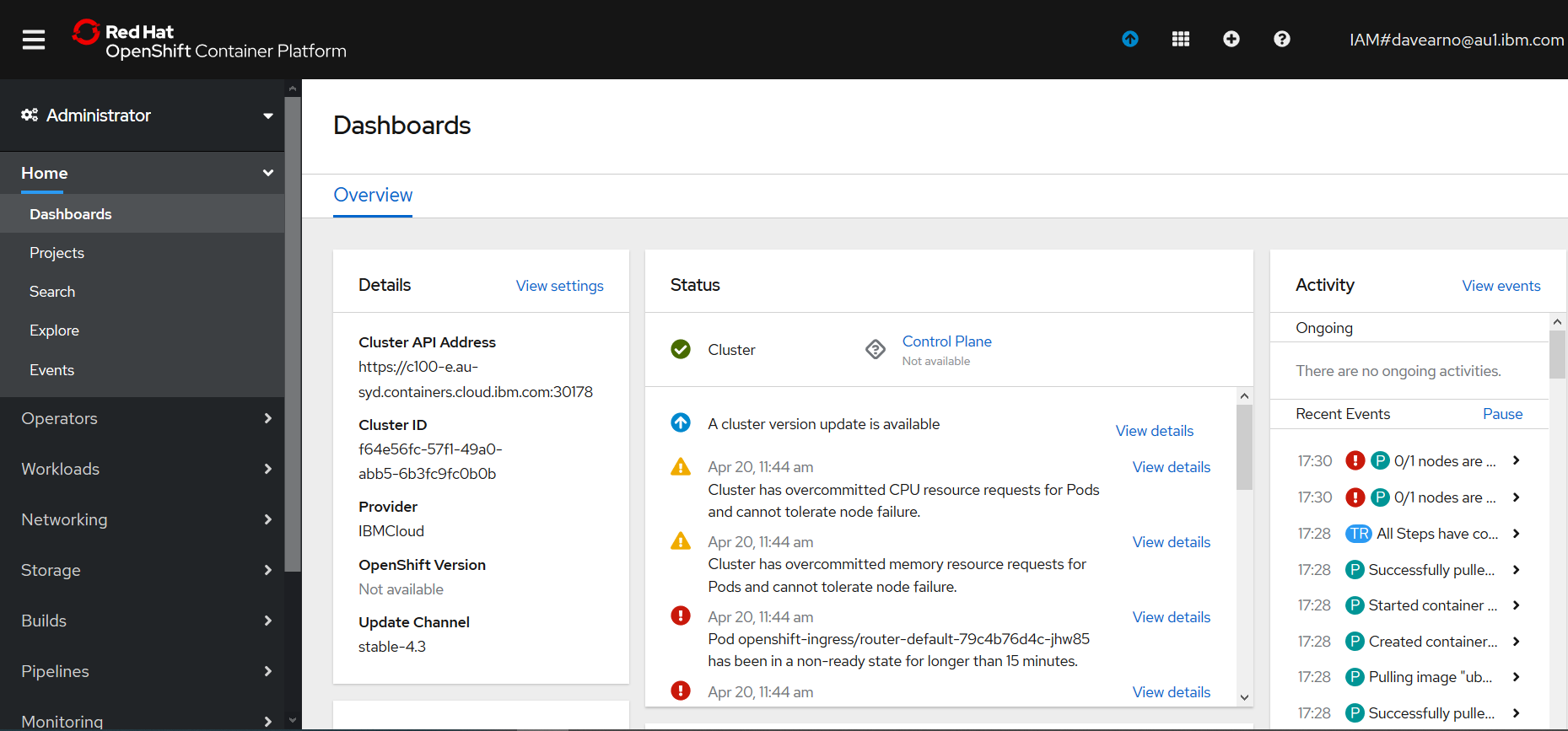


Click create and wait for the service to be provisioned

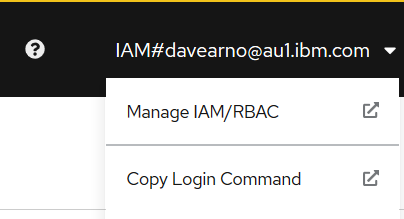
## Accessing the Red Hat OpenShift service

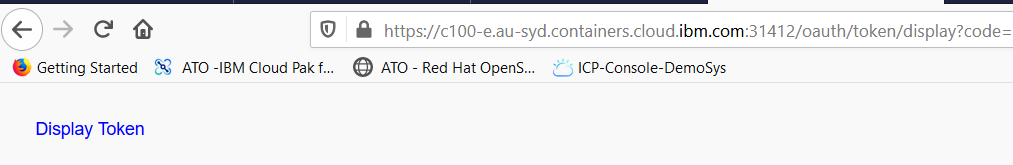


Open the web console



Log in via the command line



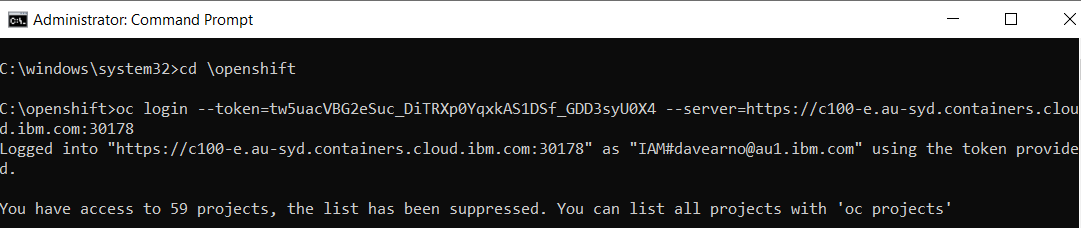


Click on display token



Copy the oc login command

(you need the RH Openshift client installed (the oc.exe))



## Installing Tekton

Follow the instructions in this article

https://developers.redhat.com/blog/2019/07/19/getting-started-with-tekton-on-red-hat-openshift/

### C:\openshift>oc new-project tekton-pipelines

Now using project "tekton-pipelines" on server "https://c100-e.au-syd.containers.cloud.ibm.com:30178".

You can add applications to this project with the 'new-app' command. For example, try:

oc new-app centos/ruby-25-centos7~https://github.com/sclorg/ruby-ex.git

to build a new example application in Ruby.

### C:\openshift>oc adm policy add-scc-to-user anyuid -z tekton-pipelines-controller

scc "anyuid" added to: ["system:serviceaccount:tekton-pipelines:tekton-pipelines-controller"]

### C:\openshift>oc apply --filename https://storage.googleapis.com/tekton-releases/latest/release.yaml

Warning: oc apply should be used on resource created by either oc create --save-config or oc apply

namespace/tekton-pipelines configured

podsecuritypolicy.policy/tekton-pipelines created

clusterrole.rbac.authorization.k8s.io/tekton-pipelines-admin created

serviceaccount/tekton-pipelines-controller created

clusterrolebinding.rbac.authorization.k8s.io/tekton-pipelines-controller-admin created

customresourcedefinition.apiextensions.k8s.io/clustertasks.tekton.dev created

customresourcedefinition.apiextensions.k8s.io/conditions.tekton.dev created

customresourcedefinition.apiextensions.k8s.io/images.caching.internal.knative.dev created

customresourcedefinition.apiextensions.k8s.io/pipelines.tekton.dev created

customresourcedefinition.apiextensions.k8s.io/pipelineruns.tekton.dev created

customresourcedefinition.apiextensions.k8s.io/pipelineresources.tekton.dev created

customresourcedefinition.apiextensions.k8s.io/tasks.tekton.dev created

customresourcedefinition.apiextensions.k8s.io/taskruns.tekton.dev created

service/tekton-pipelines-controller created

service/tekton-pipelines-webhook created

clusterrole.rbac.authorization.k8s.io/tekton-aggregate-edit created

clusterrole.rbac.authorization.k8s.io/tekton-aggregate-view created

configmap/config-artifact-bucket created

configmap/config-artifact-pvc created

configmap/config-defaults created

configmap/config-logging created

configmap/config-observability created

deployment.apps/tekton-pipelines-controller created

deployment.apps/tekton-pipelines-webhook created

### C:\openshift>oc get pods --namespace tekton-pipelines

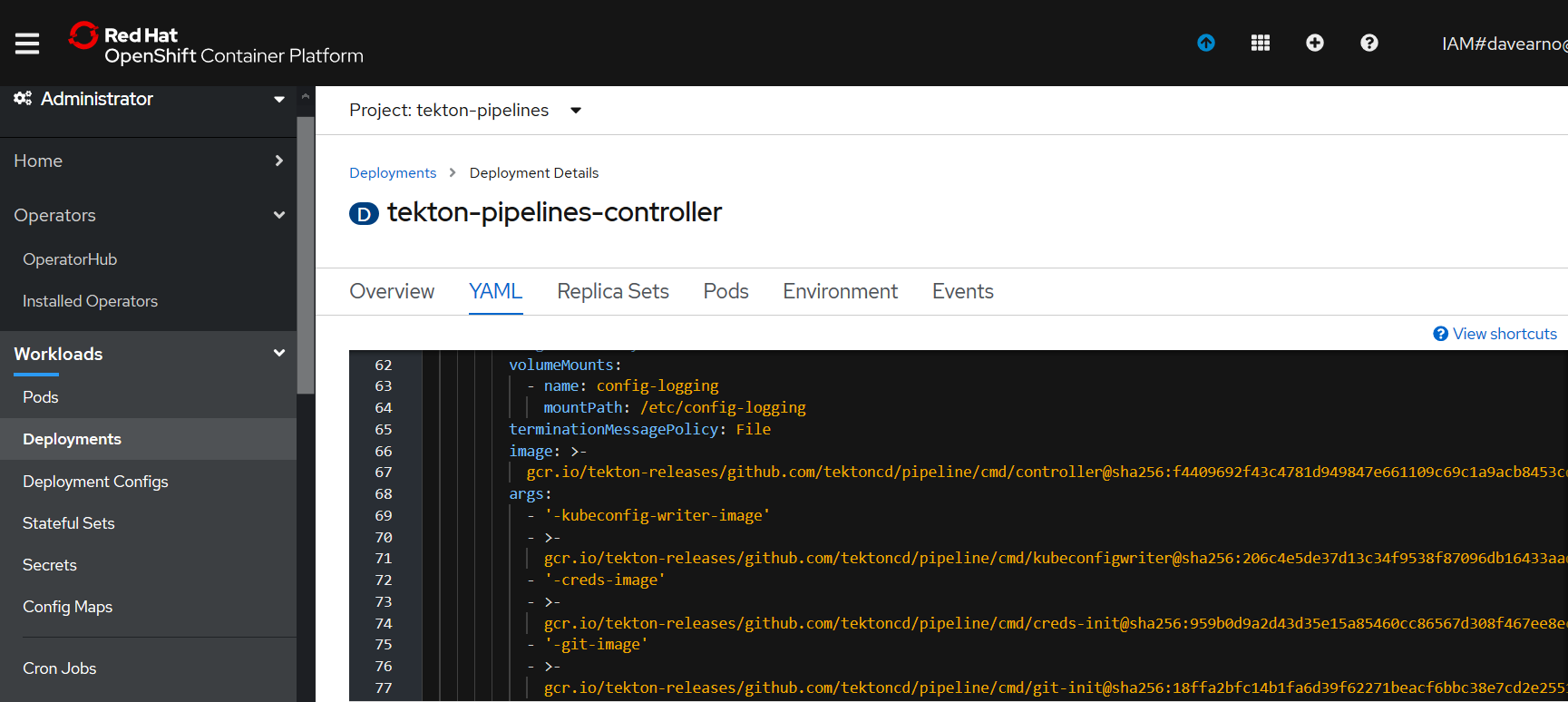
NAME READY STATUS RESTARTS AGE

tekton-pipelines-controller-5b75cdfb95-25gfg 0/1 ImagePullBackOff 0 11m

tekton-pipelines-webhook-b848dcd97-bdhnz 0/1 ImagePullBackOff 0 11m

### Correcting the paths to the image locations

Navigate to the RH Openshift Console->Workloads->Deploymenets->tekton-pipelines-controller



Edit the YAML

The deployment YAML seems to have a mismatch for the actual location of the images on google cloud. The YAML includes **:v0.10.1** this need to be removed as per the example below.

<https://console.cloud.google.com/gcr/images/tekton-releases/GLOBAL/github.com/tektoncd/pipeline/cmd>

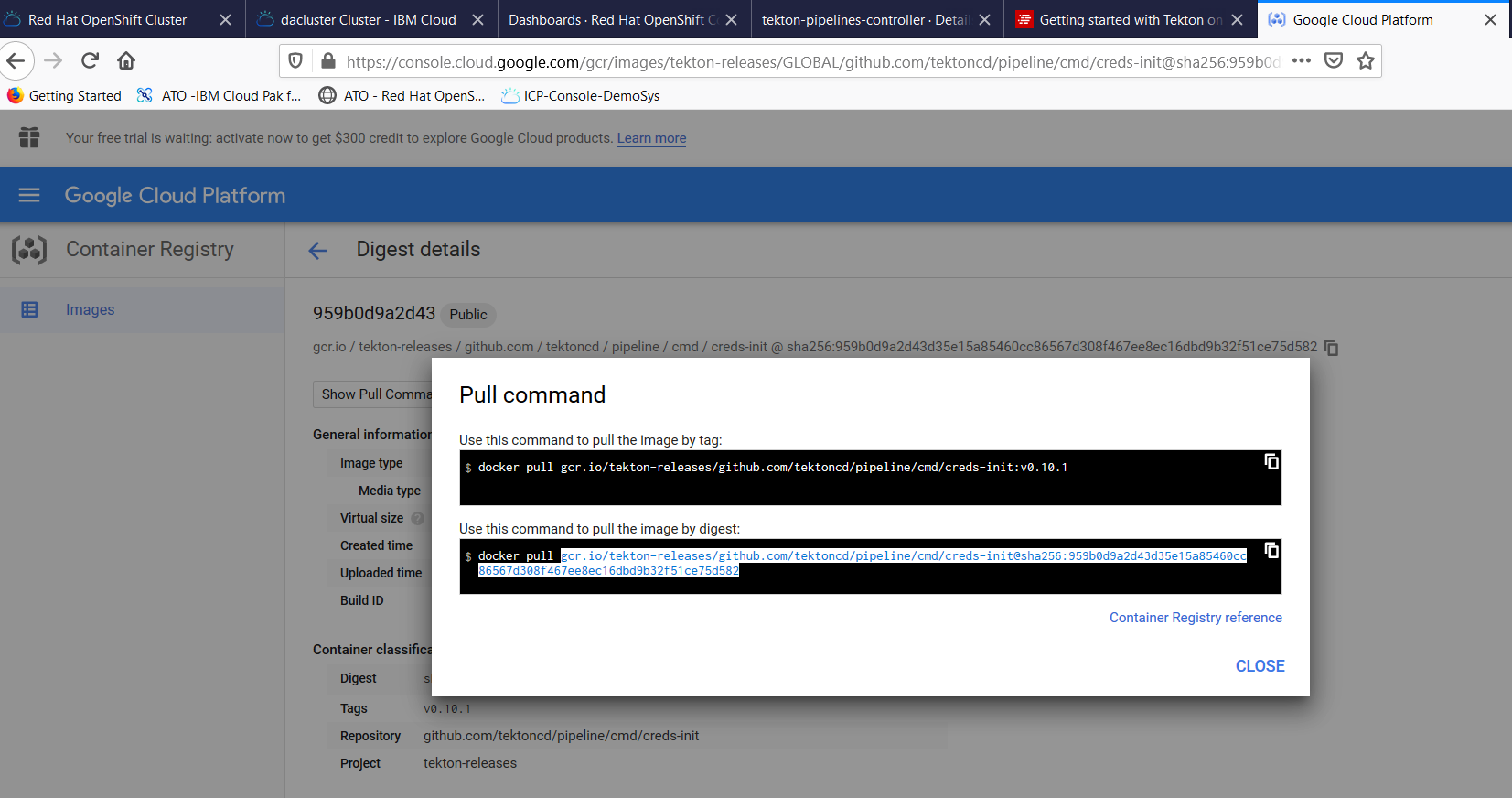
The YAML:

gcr.io/tekton-releases/github.com/tektoncd/pipeline/cmd/creds-init**:v0.10.1**@sha256:959b0d9a2d43d35e15a85460cc86567d308f467ee8ec16dbd9b32f51ce75d582

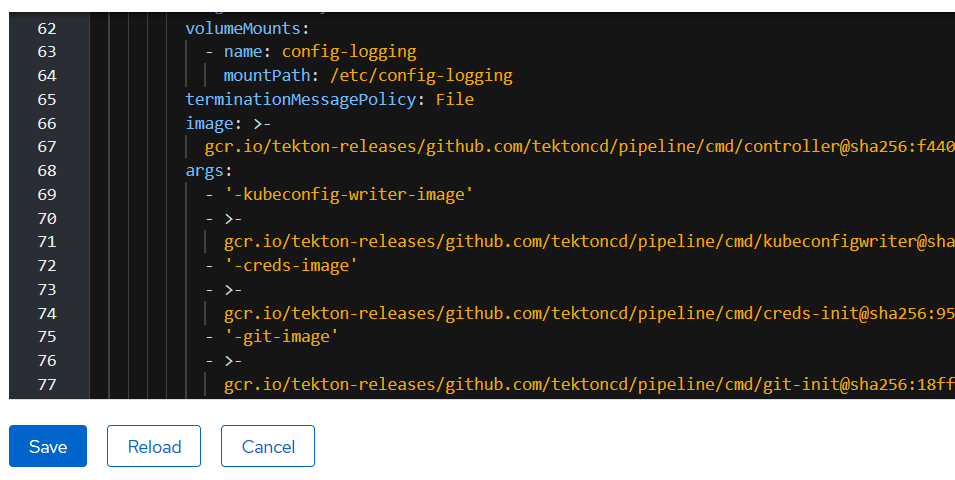
The location of the images on google:

[gcr.io/tekton-releases/github.com/tektoncd/pipeline/cmd/creds-init@sha256:959b0d9a2d43d35e15a85460cc86567d308f467ee8ec16dbd9b32f51ce75d582](mailto:gcr.io/tekton-releases/github.com/tektoncd/pipeline/cmd/creds-init@sha256:959b0d9a2d43d35e15a85460cc86567d308f467ee8ec16dbd9b32f51ce75d582)

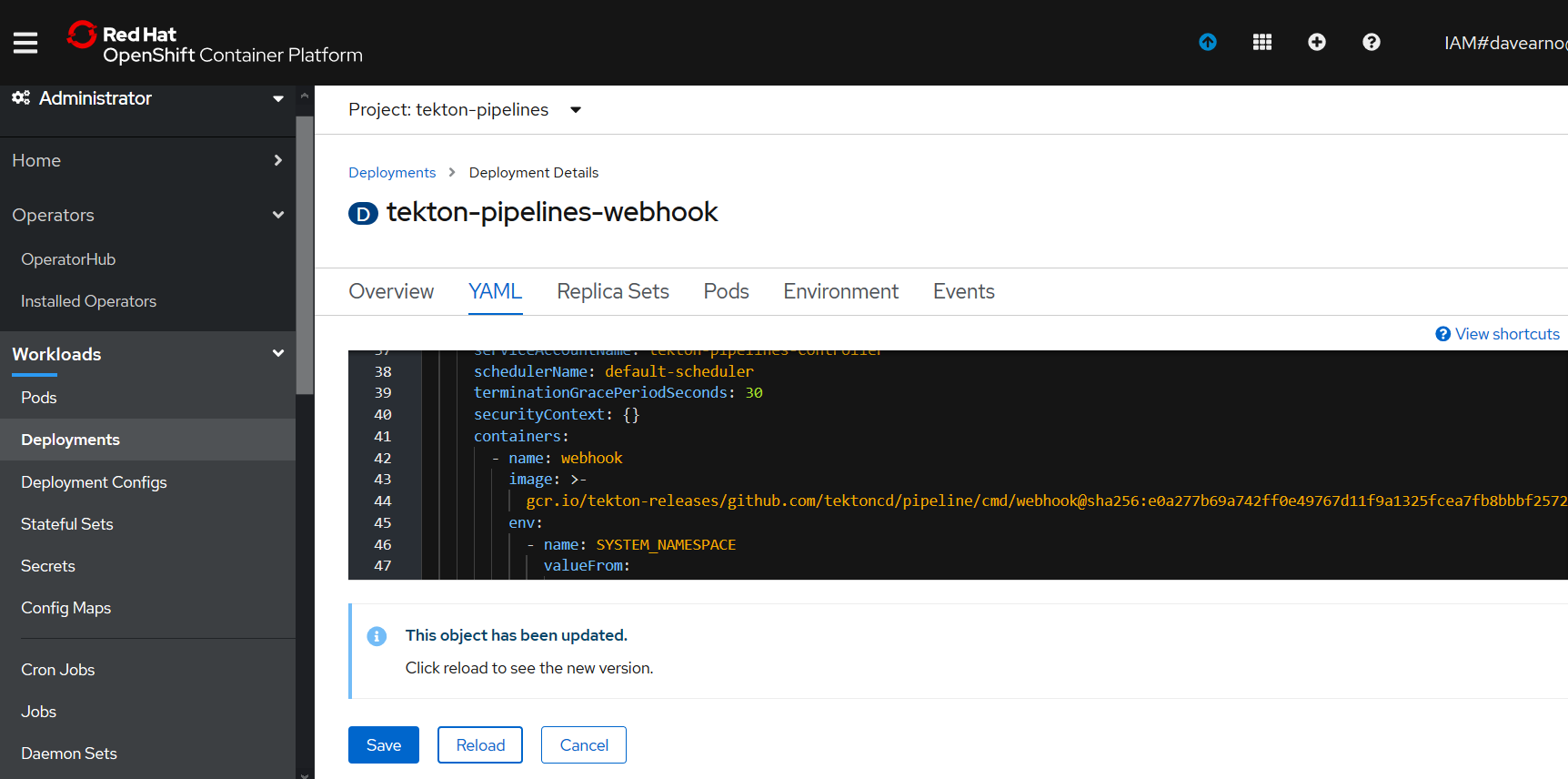
you can navigate to the image and copy and paste from the pull command.



Correct the YAML and hit Save and Reload.

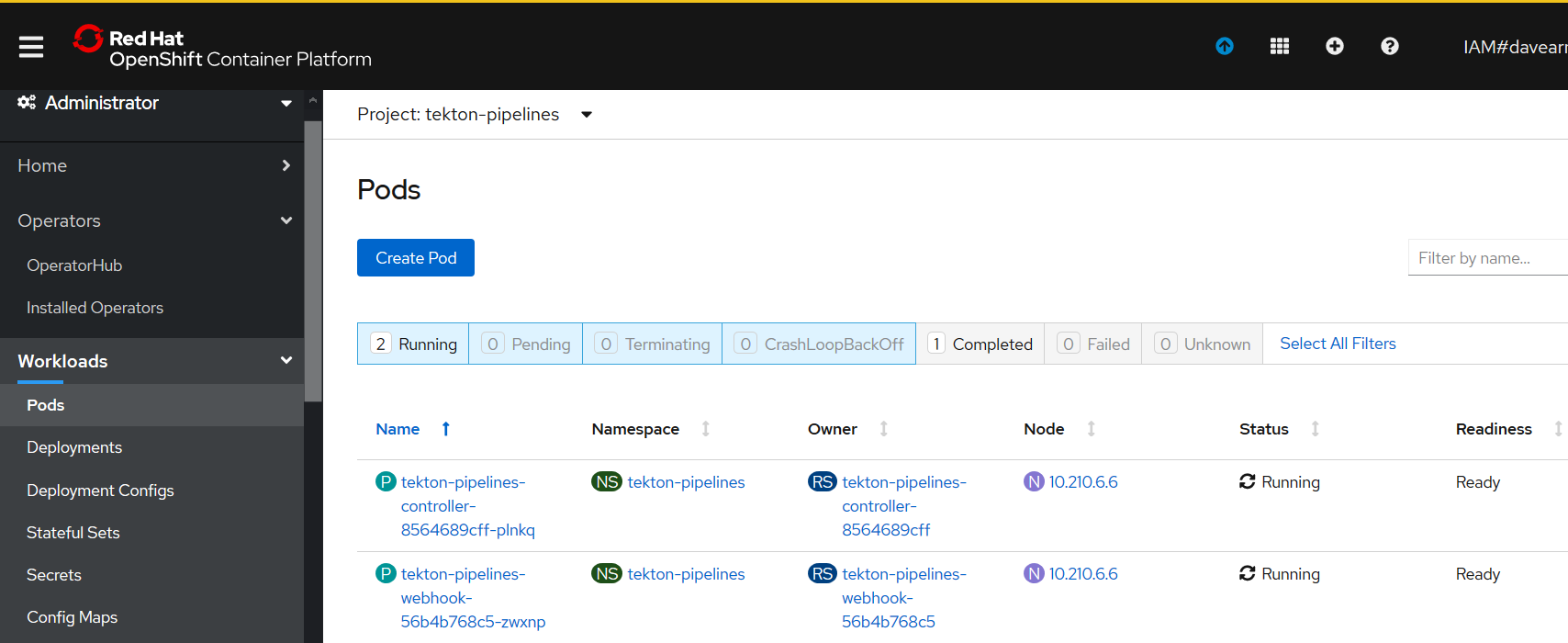


Repeat for the WebHook deployment YAML



[gcr.io/tekton-releases/github.com/tektoncd/pipeline/cmd/webhook@sha256:e0a277b69a742ff0e49767d11f9a1325fcea7fb8bbbf2572af9d49116cbb2385](mailto:gcr.io/tekton-releases/github.com/tektoncd/pipeline/cmd/webhook@sha256:e0a277b69a742ff0e49767d11f9a1325fcea7fb8bbbf2572af9d49116cbb2385)

The Pods should now start up correctly



### C:\openshift>oc get pods --namespace tekton-pipelines

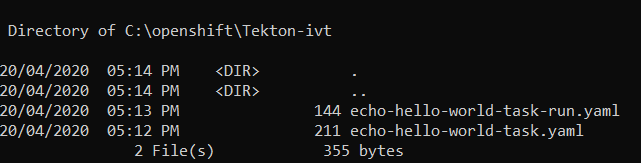
NAME READY STATUS RESTARTS AGE

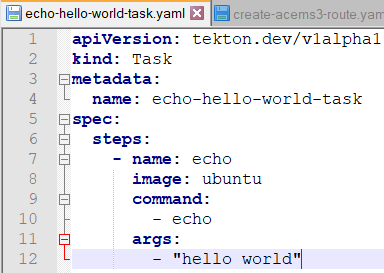
tekton-pipelines-controller-8564689cff-plnkq 1/1 Running 0 50m

tekton-pipelines-webhook-56b4b768c5-zwxnp 1/1 Running 0 3m12s

## Test Tekton with an Echo example

Create two YAML files





apiVersion: tekton.dev/v1alpha1

kind: Task

metadata:

name: echo-hello-world-task

spec:

steps:

- name: echo

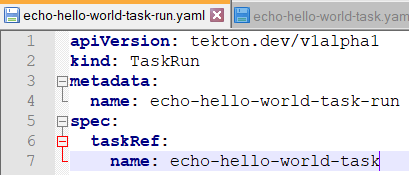
image: ubuntu

command:

- echo

args:

- "hello world"



apiVersion: tekton.dev/v1alpha1

kind: TaskRun

metadata:

name: echo-hello-world-task-run

spec:

taskRef:

name: echo-hello-world-task

### C:\openshift>oc apply -f Tekton-ivt\echo-hello-world-task.yaml

task.tekton.dev/echo-hello-world-task created

### C:\openshift>oc apply -f Tekton-ivt\echo-hello-world-task-run.yaml

taskrun.tekton.dev/echo-hello-world-task-run created

### C:\openshift>oc get taskruns/echo-hello-world-task-run -o yaml

Check the results.

apiVersion: tekton.dev/v1alpha1

kind: TaskRun

metadata:

annotations:

kubectl.kubernetes.io/last-applied-configuration: |

{"apiVersion":"tekton.dev/v1alpha1","kind":"Task","metadata":{"annotations":{},"name":"echo-hello-world-task","namespace":"tekton-pipelines"},"spec":{"steps":[{"args":["hello world"],"command":["echo"],"image":"ubuntu","name":"echo"}]}}

tekton.dev/release: devel

creationTimestamp: 2020-04-20T07:14:58Z

generation: 1

labels:

app.kubernetes.io/managed-by: tekton-pipelines

tekton.dev/task: echo-hello-world-task

name: echo-hello-world-task-run

namespace: tekton-pipelines

resourceVersion: "108232"

selfLink: /apis/tekton.dev/v1alpha1/namespaces/tekton-pipelines/taskruns/echo-hello-world-task-run

uid: 54bf7e0d-b1de-4989-914e-06444e067242

spec:

inputs: {}

outputs: {}

serviceAccountName: ""

taskRef:

kind: Task

name: echo-hello-world-task

timeout: 1h0m0s

status:

completionTime: 2020-04-20T07:28:25Z

conditions:

- lastTransitionTime: 2020-04-20T07:28:25Z

message: All Steps have completed executing

reason: Succeeded

status: "True"

type: Succeeded

podName: echo-hello-world-task-run-pod-pz6vx

startTime: 2020-04-20T07:14:58Z

steps:

- container: step-echo

imageID: docker.io/library/ubuntu@sha256:bec5a2727be7fff3d308193cfde3491f8fba1a2ba392b7546b43a051853a341d

name: echo

terminated:

containerID: cri-o://f04dd508f82092cb3978f14cfc0bedc64aa110082fc557790790577cf265d025

exitCode: 0

finishedAt: 2020-04-20T07:28:24Z

reason: Completed

startedAt: 2020-04-20T07:28:24Z