Jobs

* Vi job.yml

( eg:- apiVersion: batch/v1

kind: Job

metadata:

name: testjob

spec:

template:

metadata:

name: testjob

spec:

containers:

- name: counter

image: centos:7

command: ["bin/bash", "-c", "echo Technical-Guftgu; sleep 5"]

restartPolicy: Never )

* Kubectl apply -f job.yml
* Kubectl get pods
* Watch !! ( pod will delete after 5 seconds )

Parallel jobs

* Vi job2.yml

(eg:- apiVersion: batch/v1

kind: Job

metadata:

name: testjob

spec:

parallelism: 5 # Runs for pods in parallel

activeDeadlineSeconds: 10 # Timesout after 30 sec

template:

metadata:

name: testjob

spec:

containers:

- name: counter

image: centos:7

command: ["bin/bash", "-c", "echo Technical-Guftgu; sleep 20"]

restartPolicy: Never )

* Kubectl apply -f job2.yml
* Kubectl get pods
* Watch !! ( 5 pod will delete after 5 seconds )

Cronjob

* Vi cronjob.yml

(eg:- apiVersion: batch/v1beta1

kind: CronJob

metadata:

name: bhupi

spec:

schedule: "\* \* \* \* \*"

jobTemplate:

spec:

template:

spec:

containers:

- image: ubuntu

name: bhupi

command: ["/bin/bash", "-c", "echo Technical-Guftgu; sleep 5"]

restartPolicy: Never )

* Kubectl get pods ( after every one minute one pod will create with container )

init container

* Vi init.yml

( eg:- apiVersion: v1

kind: pod

mertadata:

name: initcontainer

spec:

initcontainers:

- name: c1

image: centos

command: ["/bin/sh", "-c", "echo HEY ITS HARISH > /tmp/xchange/testfile; sleep 30"]

volumeMounts:

- name: xchange

mountPath: "/tmp/xchange"

containers:

- name: c2

image: centos

command: ["/bin/bash", "-c", "while true; do echo 'cat /tmp/data/testfile'; sleep 5; done"]

volumeMounts:

- name: xchange

mountPath: "/tmp/data"

volumes:

- name: xchange

emptyDir: () )

* Kubectl apply -f init.yml
* Watch kubectl get pods ( you will see init:0/1 )
* Watch kuberctl get pods ( now your pod is running )
* Kubectl logs -f pod/initcontainer ( to see the output of container )