Simple Job to run echo “Hello World”

Create Simple Job from following configuration file

vim simplejob.yaml

```
apiVersion: batch/v1
kind: Job
metadata:
  name: nginx
spec:
  template:
    metadata:
      name: nginx
      labels:
        app: job
    spec:
      containers:
      - name: nginx
        image: nginx:1.9.1
        command: ["/bin/sh"]
        args: ["-c", "echo Hello World"]
      restartPolicy: Never
```

This configuration file create the job which require 1 completeion. It checks whether the pod successfully executed command. Once pod complete its execution successfully job get completed.

Deploy a job from above Yaml file.

$ kubectl apply -f simplejob.yaml

Get list of Jobs

$ kubectl get job

```
NAME     DESIRED   SUCCESSFUL   AGE
nginx    1         1            10s
```

Get the list of pod which are part of above Job.

```
$ kubectl get po --show-all
NAME        READY    STATUS     RESTARTS   AGE
nginx-wzb68   0/1      Completed   0         56s
```

Check the logs of above pod you will see output of that job.

```
$ kubectl logs -l app=job
Hello World
```

Working with Cron Jobs.

Create a cron job configuration from following file.

vim cronjob.yaml

```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: cron-job-demo
spec:
  schedule: "0 22 * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
          - name: demo
            image: nginx:1.9.1
            command: ["/bin/sh"]
            args: ["-c", "echo Time is 10PM"]
          restartPolicy: OnFailure
```

In above configuration file we have specified that schedule: "0 22 * * *" which mean our job will be scheduled at 0 minute and 22 hours. It follows 24 hour clock so the job gets scheduled at 10 PM everyday.

Deploy the cron job

$ kubectl apply -f cronjob.yaml

Get list of Cron job

$ kubectl get cronjob

NAME SCHEDULE SUSPEND ACTIVE LAST SCHEDULE AGE

cron-job-demo 0 22 * * * False 0 <none> 1s

Parallel Jobs and Job Completion

Create a job which specifies the number of replicas those should run in parallel to complete the job. Here we have specified 2 replicas and 10 completions that mean job will complete the 10 executions.

vim paralleljob.yaml

```
apiVersion: batch/v1
kind: Job
metadata:
  name: nginx-parallel
spec:
  completions: 10
  parallelism: 2
  template:
    metadata:
      name: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.9.1
        command: ["/bin/sh"]
        args: ["-c", "echo Hello World"]
```

restartPolicy: OnFailure

Deploy this job.

$ kubectl apply -f paralleljob.yaml

Check the status of the jobs.

$ kubectl get jobs

NAME DESIRED SUCCESSFUL AGE

nginx-parallel 10 0 8s

Check the status of Pods.

$ kubectl get po

NAME READY STATUS RESTARTS AGE

nginx-parallel-f955r 0/1 ContainerCreating 0 1s

nginx-parallel-pbt9s 0/1 ContainerCreating 0 1s

When the job completly get executed check the status of the jobs.

$ kubectl get jobs

NAME DESIRED SUCCESSFUL AGE

nginx 1 1 8m

nginx-parallel 10 10 2m

Delete Jobs and Cron Job.

$ kubectl delete jobs nginx-parallel

kubectl delete cronjob cron-job-demo