**Lab Exercise 21- Jobs in Kubernetes**

Step 1: Create a Kubernetes Job

Create a file named **simple-job.yaml** with the following content:

apiVersion: batch/v1

kind: Job

metadata:

name: simple-job

spec:

template:

spec:

containers:

- name: simple-job

image: busybox

command: ["sh", "-c", "echo Hello from pod $(hostname); sleep 5"]



restartPolicy: Never

**Apply the Job**

Use the kubectl apply command to create the job:

kubectl apply -f simple-job.yaml

**Check the Job Status**

Check the status of the job to see its progress:

kubectl get jobs

You should see output indicating the number of completions and the parallelism.

**Check the Pods Status**

List the pods created by the job:

kubectl get pods --selector=job-name=simple-job

You should see multiple pods running or completed.

**View Logs of Completed Pods**

To see the output of the job, you need to find the names of the completed pods and then view their logs:

POD\_NAMES=$(kubectl get pods --selector=job-name=simple-job --output=jsonpath='{.items[\*].metadata.name}')

for POD\_NAME in $POD\_NAMES

do

echo "Logs for $POD\_NAME:"

kubectl logs $POD\_NAME

done

The output should display Hello from pod <pod-name> for each pod.

**Step 2: Create a Kubernetes Job with Parallelism and Completions**

Define a Job Manifest with Parallelism and Completions

Create a file named **parallel-completion-job.yaml** with the following content:

apiVersion: batch/v1

kind: Job

metadata:

name: parallel-completion-job

spec:

completions: 5

parallelism: 2

template:

spec:

containers:

- name: parallel-completion-job

image: busybox

command: ["sh", "-c", "echo Hello from pod $(hostname); sleep 5"]

restartPolicy: Never

In this job:

* completions: 5 specifies that the job should run a total of 5 pods to completion.
* parallelism: 2 specifies that a maximum of 2 pods should run in parallel at any given time.

**Apply the Job**

Use the kubectl apply command to create the job:

kubectl apply -f parallel-completion-job.yaml

**Check the Job Status**

Check the status of the job to see its progress:

kubectl get jobs

You should see output indicating the number of completions and the parallelism.

**Check the Pods Status**

List the pods created by the job:

kubectl get pods --selector=job-name=parallel-completion-job

You should see multiple pods running or completed.

**View Logs of Completed Pods**

To see the output of the job, you need to find the names of the completed pods and then view their logs:

POD\_NAMES=$(kubectl get pods --selector=job-name=parallel-completion-job --output=jsonpath='{.items[\*].metadata.name}')

for POD\_NAME in $POD\_NAMES

do

echo "Logs for $POD\_NAME:"

kubectl logs $POD\_NAME

done

The output should display Hello from pod <pod-name> for each pod.

**Step 2: Experiment with Different Parallelism and Completions Settings**

Update the Job Manifest

Modify the **parallel-completion-job.yaml** file to change the parallelism and completions settings:

apiVersion: batch/v1

kind: Job

metadata:

name: parallel-completion-job

spec:

completions: 10

parallelism: 3

template:

spec:

containers:

- name: parallel-completion-job

image: busybox

command: ["sh", "-c", "echo Hello from updated pod $(hostname); sleep 5"]

restartPolicy: Never

In this job:

* completions: 10 specifies that the job should run a total of 10 pods to completion.
* parallelism: 3 specifies that a maximum of 3 pods should run in parallel at any given time.

**Apply the Updated Job**

Use the kubectl apply command to update the job:

kubectl apply -f parallel-completion-job.yaml

Check the Updated Job Status

Check the status of the updated job to see its progress:

kubectl get jobs

Check the Updated Pods Status

List the pods created by the updated job:

kubectl get pods --selector=job-name=parallel-completion-job

View Logs of Completed Pods

View the logs of the completed pods:

POD\_NAMES=$(kubectl get pods --selector=job-name=parallel-completion-job --output=jsonpath='{.items[\*].metadata.name}')

for POD\_NAME in $POD\_NAMES

do

echo "Logs for $POD\_NAME:"

kubectl logs $POD\_NAME

done

Step 3: Clean Up

After completing the exercise, clean up the resources created:

kubectl delete job simple-job

kubectl delete job parallel-completion-job