

Pre requisite: Install k8s cluster (minikube) + kubectl

Notes: minikube can be deployed as a VM, a container

Start it using minikube start --driver=docker OR minikube start --driver=virtualbox

This makes kubectl configured to use “minikube” cluster and “default” namespace my default

1- Create a pod with the name “imperative-nginx” and with the image nginx and latest tag. using Imperative command (not yaml).

```
amr@amrgomaa: ~  
amr@amrgomaa:~$ kubectl run imperative-nginx --image=nginx:latest  
pod/imperative-nginx created  
amr@amrgomaa:~$ kubectl get pods  
NAME             READY   STATUS    RESTARTS   AGE  
imperative-nginx 1/1     Running   0           6s  
amr@amrgomaa:~$
```

2- Create a pod with the name webserver and with the image “nginx123” Use a pod-definition YAML file.

```
apiVersion: v1  
kind: Pod  
metadata:  
  creationTimestamp: null  
  labels:  
    run: webserver  
  name: webserver  
spec:  
  containers:  
  - image: nginx123  
    name: webserver  
    resources: {}  
  dnsPolicy: ClusterFirst  
  restartPolicy: Always  
status: {}  
~  
~
```

```
amr@amrgomaa: ~  
amr@amrgomaa:~$ kubectl apply -f sprints.yml  
pod/webserver created  
amr@amrgomaa:~$
```

3- What is the nginx pod status?

```
amr@amrgomaa:~$ kubectl get pods
NAME                READY   STATUS              RESTARTS   AGE
imperative-nginx    1/1    Running             0          4m5s
webserver           0/1    ImagePullBackOff    0          50s
amr@amrgomaa:~$
```

Error in the image pulling

4- Change the nginx pod image to “nginx” check the status again
change the image in the yaml file

```
amr@amrgomaa:~$ kubectl apply -f sprints.yml
pod/webserver configured
amr@amrgomaa:~$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
imperative-nginx    1/1    Running   0          9m46s
webserver           1/1    Running   0          6m31s
amr@amrgomaa:~$
```

5- How many pods are running in the system? Type the command to show this

kubectl get pods

```
amr@amrgomaa:~$ kubectl apply -f sprints.yml
pod/webserver configured
amr@amrgomaa:~$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
imperative-nginx    1/1    Running   0          9m46s
webserver           1/1    Running   0          6m31s
amr@amrgomaa:~$
```

6- What does READY column in the output of get pods command indicate?

The number of container in the pod

7- Delete first pod named imperative-nginx you just created. Type the command to do this

```
amr@amrgomaa:~$ kubectl delete pod imperative-nginx
pod "imperative-nginx" deleted
amr@amrgomaa:~$
```

8- Which node is pod named webserver running on (list two commands to do this)

```
amr@amrgomaa:~$ kubectl get pods -owide
NAME          READY   STATUS    RESTARTS   AGE   IP            NODE       NOMINATED
NODE   READINESS GATES
webserver     1/1     Running   0           21m   172.17.0.3    minikube   <none>
```

```
amr@amrgomaa:~$ kubectl describe pod webserver | grep Node
Node:          minikube/192.168.59.101
Node-Selectors: <none>
amr@amrgomaa:~$
```

9- Get a shell to the running container i.e ssh into it (figure out the command)

```
amr@amrgomaa:~$ kubectl exec -it webserver -- //bin/bash
root@webserver:/#
```

10- Run cat /etc/os-release inside the container

```
root@webserver:/# cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 11 (bullseye)"
NAME="Debian GNU/Linux"
VERSION_ID="11"
VERSION="11 (bullseye)"
VERSION_CODENAME=bullseye
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
root@webserver:/#
```

11- Exit from the shell (/bin/bash) session

```
root@webserver:/# exit
exit
amr@amrgomaa:~$
```

12- Get logs of pod, what are logs and what they are used for?

The importance of logs is to keep track of what our pod/application is doing or to keep track of users, new requests, etc. And we need them for troubleshooting; whenever something goes wrong or our application crashes, we check the logs.

```
amr@amrgomaa:~$ kubectl logs webserver
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/01/17 13:07:05 [notice] 1#1: using the "epoll" event method
2023/01/17 13:07:05 [notice] 1#1: nginx/1.23.3
2023/01/17 13:07:05 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/01/17 13:07:05 [notice] 1#1: OS: Linux 5.10.57
2023/01/17 13:07:05 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/01/17 13:07:05 [notice] 1#1: start worker processes
2023/01/17 13:07:05 [notice] 1#1: start worker process 29
2023/01/17 13:07:05 [notice] 1#1: start worker process 30
```

13- How many ReplicaSets exist on the system?

```
amr@amrgomaa:~$ kubectl get rs
No resources found in default namespace.
```

14- create a ReplicaSet
withname= replica-set-1
image= busybox
replicas= 3

```
amr@amrgomaa: ~  
apiVersion: apps/v1  
kind: ReplicaSet  
metadata:  
  name: replica-set-1  
  labels:  
    app: busybox-app  
    tier: frontend  
spec:  
  replicas: 3  
  selector:  
    matchLabels:  
      tier: frontend  
  template:  
    metadata:  
      labels:  
        tier: frontend  
    spec:  
      containers:  
      - name: sprint-pod  
        image: busybox  
        tty: true  
~  
~  
~  
~  
~  
~  
"replica-1.yml" 22L, 347B 22,17 All  
amr@amrgomaa:~$ kubectl apply -f replica-1.yml  
replicaset.apps/replica-set-1 created
```

15- Scale the ReplicaSet replica-set-1 to 5 PODs.

```
amr@amrgomaa: ~  
amr@amrgomaa:~$ kubectl scale --replicas=5 -f replica-1.yml  
replicaset.apps/replica-set-1 scaled  
amr@amrgomaa:~$ kubectl get rs  
NAME                DESIRED    CURRENT    READY    AGE  
replica-set-1        5          5          5        2m50s  
amr@amrgomaa:~$
```

16- How many PODs are READY in the replica-set-1?

5 pods are ready

```
amr@amrgomaa:~$ kubectl get pods  
NAME                READY    STATUS    RESTARTS    AGE  
replica-set-1-hwsrr  1/1      Running   0            48s  
replica-set-1-sk6zg  1/1      Running   0            3m31s  
replica-set-1-t2b82  1/1      Running   0            3m31s  
replica-set-1-v2gcv  1/1      Running   0            48s  
replica-set-1-zpdlr  1/1      Running   0            3m31s  
webserver            1/1      Running   0            97m  
amr@amrgomaa:~$
```

17- Delete any one of the 5 PODs then check How many PODs exist now?

```
amr@amrgomaa:~$ kubectl delete pod replica-set-1-hwsrr  
pod "replica-set-1-hwsrr" deleted  
amr@amrgomaa:~$ kubectl get pods  
NAME                READY    STATUS    RESTARTS    AGE  
replica-set-1-sk6zg  1/1      Running   0            4m59s  
replica-set-1-t2b82  1/1      Running   0            4m59s  
replica-set-1-v2gcv  1/1      Running   0            2m16s  
replica-set-1-zpdlr  1/1      Running   0            4m59s  
replica-set-1-zrzrz  1/1      Running   0            44s  
webserver            1/1      Running   0            99m  
amr@amrgomaa:~$
```

Why are there still 5 PODs, even after you deleted one?

Because one of the replicaset features is to keep the number of running pods equals to the desired replicas in the yaml file so once the pod is deleted another one is created