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

- Kubectl run imperative-nginx –image=nginx
- Kubectl get pod

```
controlplane $ kubectl run imperative-nginx --image=nginx
pod/imperative-nginx created
controlplane $ kubectl get pod
NAME              READY   STATUS    RESTARTS   AGE
imperative-nginx   1/1     Running   0          24s
```

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

a. Yaml File

```
Editor   Tab 1   +
apiVersion: v1
kind: Pod
metadata:
    name: webserver
spec:
    containers:
        - name: nginx
          image: nginx123
~
~
```

```
controlplane $ kubectl apply -f pod-definition.yaml
pod/webserver created
```

```
controlplane $ vim pod-definition.yaml
controlplane $ kubectl apply -f pod-definition.yaml
pod/webserver created
controlplane $ kubectl get pod
NAME              READY   STATUS              RESTARTS   AGE
imperative-nginx   1/1     Running             0          7m3s
webserver         0/1     ContainerCreating   0          3s
```

3- What is the nginx pod status?

a. *State:ImagePullBackOff*

4- Change the nginx pod image to “nginx” check the status again

```
controlplane $ vim pod-definition.yaml
controlplane $ kubectl apply -f pod-definition.yaml
pod/webserver configured
controlplane $ kubectl get pod
NAME              READY   STATUS    RESTARTS   AGE
imperative-nginx  1/1     Running   0          12m
webserver         1/1     Running   0          5m50s
```

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

```
controlplane $ kubectl get pod
NAME               READY   STATUS    RESTARTS   AGE
imperative-nginx   1/1     Running   0          18m
webserver          1/1     Running   0          2m25s
```

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

it shows how many containers in a pod are considered ready

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

```
controlplane $ kubectl delete pod/imperative-nginx
pod "imperative-nginx" deleted
controlplane $
```

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

```
controlplane $ kubectl get pod -o wide
NAME        READY   STATUS    RESTARTS   AGE   IP            NODE     NOMINATED NODE   READINESS GATES
webserver   1/1     Running   0          11m   192.168.1.4   node01   <none>           <none>
controlplane $
```

```
controlplane $ kubectl describe pod webserver
Name:             webserver
Namespace:        default
Priority:         0
Service Account:  default
Node:             node01/172.30.2.2
Start Time:       Sun, 22 Jan 2023 21:28:54 +0000
Labels:           <none>
Annotations:      cni.projectcalico.org/containerID: 82a4f01318f11a97c4f7e87d56ea3718167500814129cca3ffc213b977fdafc0
                  cni.projectcalico.org/podIP: 192.168.1.4/32
                  cni.projectcalico.org/podIPs: 192.168.1.4/32
Status:           Running
IP:               192.168.1.4
IPs:
  IP:  192.168.1.4
Containers:
  nginx:
    Container ID:   containerd://25ac813abd9f42aace974ef7576c5293ab4600dcbca71a84090284dc51ef6d5e
    Image:          nginx
    Image ID:       docker.io/library/nginx@sha256:b8f2383a95879e1ae064940d9a200f67a6c79e710ed82ac42263397367e7cc4e
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Sun, 22 Jan 2023 21:31:15 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-mcfht (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
```

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

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

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

```
controlplane $ kubectl exec -it webserver -- /bin/bash
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:/# exit
exit
controlplane $
```

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

```
controlplane $ 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
2023/01/22 21:31:15 [notice] 1#1: using the "epoll" event method
2023/01/22 21:31:15 [notice] 1#1: nginx/1.23.3
2023/01/22 21:31:15 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/01/22 21:31:15 [notice] 1#1: OS: Linux 5.4.0-131-generic
2023/01/22 21:31:15 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/01/22 21:31:15 [notice] 1#1: start worker processes
2023/01/22 21:31:15 [notice] 1#1: start worker process 29
controlplane $
```

13- How many ReplicaSets exist on the system?

```
controlplane $ kubectl get rs
No resources found in default namespace.
controlplane $
```

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

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: frontend
  labels:
    app: guestbook
    tier: frontend
spec:
  replicas: 3
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
      - name: busybox-1
        image: busybox
        tty: true
```

```
controlplane $ kubectl apply -f my-rs
replicaset.apps/frontend created
controlplane $
```

```
controlplane $ kubectl get pod
NAME              READY   STATUS    RESTARTS   AGE
frontend-4sdt9    1/1     Running   0          67s
frontend-r9khc    1/1     Running   0          67s
frontend-xj5fp    1/1     Running   0          67s
webserver         1/1     Running   0          27m
controlplane $
```

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

```
controlplane $ kubectl get pod
NAME             READY   STATUS    RESTARTS   AGE
frontend-4sdt9   1/1     Running   0          67s
frontend-r9khc   1/1     Running   0          67s
frontend-xj5fp   1/1     Running   0          67s
webserver        1/1     Running   0          27m
controlplane $ kubectl scale --replicas=5 -f my-rs
replicaset.apps/frontend scaled
controlplane $
```

```
replicaset.apps/frontend scaled
controlplane $ kubectl get pod
NAME             READY   STATUS    RESTARTS   AGE
frontend-4sdt9   1/1     Running   0          2m18s
frontend-5czgg   1/1     Running   0          16s
frontend-bpzgb   1/1     Running   0          16s
frontend-r9khc   1/1     Running   0          2m18s
frontend-xj5fp   1/1     Running   0          2m18s
webserver        1/1     Running   0          29m
controlplane $
```

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

5

```
replicaset.apps/frontend scaled
controlplane $ kubectl get pod
NAME             READY   STATUS    RESTARTS   AGE
frontend-4sdt9   1/1     Running   0          2m18s
frontend-5czgg   1/1     Running   0          16s
frontend-bpzgb   1/1     Running   0          16s
frontend-r9khc   1/1     Running   0          2m18s
frontend-xj5fp   1/1     Running   0          2m18s
webserver        1/1     Running   0          29m
controlplane $
```

17- Delete any one of the 5 PODs then check How many PODs exist now?
Why are there still 5 PODs, even after you deleted one?

```
controlplane $ vim my-rs
controlplane $ kubectl apply -f my-rs
replicaset.apps/frontend created
controlplane $ kubectl scale --replicas=5 -f my-rs
replicaset.apps/frontend scaled
controlplane $ kubectl get pod
NAME             READY   STATUS    RESTARTS   AGE
frontend-6mgpw   1/1     Running   0          3m
frontend-jjwqg   1/1     Running   0          2m19s
frontend-pmjt5   1/1     Running   0          3m
frontend-t4qmx   1/1     Running   0          3m
frontend-tphtq   1/1     Running   0          2m19s
controlplane $ kubectl get pod
NAME             READY   STATUS    RESTARTS   AGE
frontend-6mgpw   1/1     Running   0          3m32s
frontend-jjwqg   1/1     Running   0          2m51s
frontend-pmjt5   1/1     Running   0          3m32s
frontend-t4qmx   1/1     Running   0          3m32s
frontend-tphtq   1/1     Running   0          2m51s
controlplane $ kubectl delete pod/^C
controlplane $ kubectl delete pod/frontend-tphtq
pod "frontend-tphtq" deleted
controlplane $ kubectl get pod
NAME             READY   STATUS    RESTARTS   AGE
frontend-66jxn   1/1     Running   0          38s
frontend-6mgpw   1/1     Running   0          5m17s
frontend-jjwqg   1/1     Running   0          4m36s
frontend-pmjt5   1/1     Running   0          5m17s
frontend-t4qmx   1/1     Running   0          5m17s
controlplane $
```

---

apiVersion: apps/v1

kind: ReplicaSet

metadata:

  name: frontend

```
  labels:
    app: guestbook
    tier:  frontend
spec:
  replicas: 3
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
      - name: busybox-1
        image: busybox
        tty: true
```