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

```
controlplane $ kubectl run imperative-nginx --image=nginx:latest
pod/imperative-nginx created
controlplane $ kubectl get pods
NAME              READY   STATUS    RESTARTS   AGE
imperative-nginx   1/1     Running   0          12s
controlplane $
```

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

```
apiVersion: v1

kind: Pod

metadata:

  name: webserver

spec:

  containers:

  - name: nginx

    image: nginx123
```

```
controlplane $ vim new-pod
controlplane $ kubectl create -f new-pod
pod/webserver created
controlplane $ kubectl get pods
NAME              READY   STATUS         RESTARTS   AGE
imperative-nginx   1/1     Running        0          6m43s
webserver          0/1     ErrImagePull   0          9s
controlplane $
```

3- What is the nginx pod status?

```
webserver          0/1     ImagePullBackOff   0          99s
```

image pullbackoff

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

```
apiVersion: v1

kinc: Poc

metacata:

  name: webserver

spec:

  containers:

  - name: nginx

    image: nginx
```

```
controlplane $ vim new-poc
controlplane $ kubectl create -f new-poc
Error from server (AlreacyExists): error when creating "new-poc": pocs "webserver" alreacy exists
```

Nothing changes we need to delete or apply the old pod with the same name first

```
controlplane $ kubectl get pocs
NAME               READY   STATUS             RESTARTS   AGE
imperative-nginx   1/1     Running            0          ⸤0m
webserver          0/1     ImagePullBack ff   0          13m
controlplane $ kubectl celete poc webserver
poc "webserver" celetec
controlplane $ kubectl create -f new-poc
poc/webserver createc
controlplane $ kubectl get pocs
NAME               READY   STATUS    RESTARTS   AGE
imperative-nginx   1/1     Running   0          ⸤1m
webserver          1/1     Running   0          4! s
controlplane $
```

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

```
controlplane $ kubectl get pocs
NAME               READY   STATUS    RESTARTS   AGE
imperative-nginx   1/1     Running   0          ⸤1m
webserver          1/1     Running   0          4! s
```

6- What does READY column in the output of get pods command indicate?
The number of containers ready in the pod

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

```
controlplane $ kubectl celete poc imperative-nginx
poc "imperative-nginx" celetec
controlplane $ kubectl get pocs
NAME        READY   STATUS    RESTARTS   AGE
webserver   1/1     Running   0          7m1⸤s
controlplane $
```

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

```
controlplane $ kubectl get pocs -o wice
NAME        READY   STATUS    RESTARTS   AGE   IP           N DE     N MINATED N DE   READINESS GATES
webserver   1/1     Running   0          10m   19⸤.168.1.!   noce01   <none>           <none>
controlplane $
```

```
controlplane $ kubectl describe poc webserver
Name:             webserver
Namespace:        default
Priority:         0
Service Account:  default
Node:             node01/172.30.2.2
```

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 get pods
NAME        READY   STATUS    RESTARTS   AGE
webserver   1/1     Running   0          27m
controlplane $ kubectl exec --stdin --tty 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
2023/01/19 17:33:03 [notice] 1#1: using the "epoll" event method
2023/01/19 17:33:03 [notice] 1#1: nginx/1.23.3
2023/01/19 17:33:03 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/01/19 17:33:03 [notice] 1#1: OS: Linux 5.4.0-131-generic
2023/01/19 17:33:03 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/01/19 17:33:03 [notice] 1#1: start worker processes
2023/01/19 17:33:03 [notice] 1#1: start worker process 28
```

13- How many ReplicaSets exist on the system?

zero

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

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

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: replica-set-1
  labels:
    app: example
    tier: frontend
spec:
  # modify replicas according to your case
  replicas: 3
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
      - name: example-replica
        image: busybox
```

```
controlplane $ vim replica
controlplane $ kubectl create -f replica
replicaset.apps/replica-set-1 created
controlplane $ kubectl get rs
NAME            DESIRED   CURRENT   READY   AGE
replica-set-1   3         3         0       12s
```

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

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: replica-set-1
  labels:
    app: example
    tier: frontend
spec:
  # modify replicas according to your case
  replicas: 5
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
      - name: example-replica
        image: busybox
```

```
controlplane $ kubectl apply -f replica
Warning: resource replicasets/replica-set-1 is missing the kubectl.kubernetes.io/last-applied-configuration annotat
ion which is required by kubectl apply. kubectl apply should only be used on resources created declaratively by eit
her kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.
replicaset.apps/replica-set-1 configured
controlplane $ kubectl get rs
NAME           DESIRED   CURRENT   READY   AGE
replica-set-1   5         5         0       18m
controlplane $
```

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

Zero

```
controlplane $ kubectl apply -f replica
Warning: resource replicasets/replica-set-1 is missing the kubectl.kubernetes.io/last-applied-configuration annotat
ion which is required by kubectl apply. kubectl apply should only be used on resources created declaratively by eit
her kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.
replicaset.apps/replica-set-1 configured
controlplane $ kubectl get rs
NAME           DESIRED   CURRENT   READY   AGE
replica-set-1   5         5         0       18m
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 $ kubectl get rs
NAME           DESIRED   CURRENT   READY   AGE
replica-set-1   5         5         0       24m
controlplane $ kubectl get pods
NAME                   READY   STATUS             RESTARTS        AGE
replica-set-1-gfvls    0/1     CrashLoopBackOff   9 (2m37s ago,   24m
replica-set-1-jcvhb    0/1     CrashLoopBackOff   5 (2m26s ago,   5m25s
replica-set-1-npkng    0/1     CrashLoopBackOff   9 (3m9s ago,    24m
replica-set-1-szw9t    0/1     CrashLoopBackOff   9 (2m31s ago,   24m
replica-set-1-v5fjj    0/1     CrashLoopBackOff   5 (2m27s ago,   5m25s
controlplane $ kubectl delete pod replica-set-1-gfvls
pod "replica-set-1-gfvls" deleted
controlplane $ kubectl get pods
NAME                   READY   STATUS             RESTARTS        AGE
replica-set-1-jcvhb    0/1     CrashLoopBackOff   6 (23s ago,     6m14s
replica-set-1-npkng    0/1     CrashLoopBackOff   9 (3m58s ago,   25m
replica-set-1-szw9t    0/1     CrashLoopBackOff   9 (3m20s ago,   25m
replica-set-1-v5fjj    0/1     CrashLoopBackOff   6 (34s ago,     6m14s
replica-set-1-ws2vs    0/1     CrashLoopBackOff   1 (3s ago,      6s
controlplane $ kubectl get rs
NAME           DESIRED   CURRENT   READY   AGE
replica-set-1   5         5         0       25m
controlplane $
```

Because the replicaset is obligated to create 5 pods and to replace any one of them in case anyone has crashed for any reason

P.s. the d characters and some othe number aren't clear this a probem either from killercuda or my browser alse note that we change the state of the replicaset pods by applying tty:true or give it a sleep command the pods aren't running because they have nothing to do so the just exit

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: replica-set-1
  labels:
    app: example
    tier: frontend
spec:
  # modify replicas according to your
  replicas: 5
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
      - name: example-replica
        image: busybox
        tty: true
```