

2- Create a pod with the name redis and with the image redis.

```
! redis-pod.yml > {} spec > [ ] containers > {} 0 > [ ] ports > {} 0
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: redis
5  spec:
6    containers:
7    - name: redis
8      image: redis:5.0.4
9      ports:
10     - containerPort: 6379
11       protocol: TCP
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS D:\K8S> kubectl create -f redis-pod.yml
pod/redis created
PS D:\K8S> kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
redis   1/1     Running   0           7s
PS D:\K8S>
```

3- Create a pod with the name nginx and with the image “nginx123”

4- What is the nginx pod status?

```
! nginx-pod.yml > {} spec > [ ] containers > {} 0 > [ ] image
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: nginx
5  spec:
6    containers:
7    - name: nginx
8      image: nginx123
9      ports:
10     - containerPort: 80
11
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS D:\K8S> kubectl create -f nginx-pod.yml
pod/nginx created
PS D:\K8S> kubectl get pods
NAME    READY   STATUS             RESTARTS   AGE
nginx   0/1     ContainerCreating   0           3s
redis   1/1     Running             0           3m27s
PS D:\K8S> kubectl get pods
NAME    READY   STATUS             RESTARTS   AGE
nginx   0/1     ErrImagePull        0           12s
redis   1/1     Running             0           3m36s
PS D:\K8S>
```

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

```
! nginx-pod.yml > {} spec > [ ] containers > {} 0 > image
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: nginx
5  spec:
6    containers:
7      - name: nginx
8        image: nginx
9        ports:
10       - containerPort: 80
11
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   COMMENTS

```
PS D:\K8S> kubectl apply -f nginx-pod.yml
pod/nginx configured
PS D:\K8S> kubectl get pods
NAME      READY   STATUS              RESTARTS   AGE
nginx     0/1     ImagePullBackOff    0           4m57s
redis     1/1     Running             0           8m21s
PS D:\K8S> kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx     1/1     Running   1 (37s ago) 5m21s
redis     1/1     Running   0           8m45s
PS D:\K8S>
```

6- How many ReplicaSets exist on the system?

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   COMMENTS

```
PS D:\K8S> kubectl get replicaset
No resources found in default namespace.
PS D:\K8S>
```

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

! redis-pod.yml ! nginx-pod.yml ! replica-set-1.yml X ! deployment.yml ! nginx-deployment.yml

```
! replica-set-1.yml > {} spec > # replicas
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: replica-set-1
5    labels:
6      app: replica-set-1
7      tier: replica-set-1
8  spec:
9    replicas: 3
10   selector:
11     matchLabels:
12       tier: replica-set-1
13   template:
14     metadata:
15       labels:
16         tier: replica-set-1
17     spec:
18       containers:
19         - name: busybox
20           image: busybox
21           command: [ "sleep", "3600" ]
22
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS D:\K8S> kubectl create -f replica-set-1.yml
replicaset.apps/replica-set-1 created
PS D:\K8S> kubectl get replicaset
NAME           DESIRED  CURRENT  READY  AGE
replica-set-1  3        3        1      3s
PS D:\K8S> kubectl get replicaset
NAME           DESIRED  CURRENT  READY  AGE
replica-set-1  3        3        3      45s
PS D:\K8S>
```

## 8- Scale the ReplicaSet replica-set-1 to 5 PODs.

! redis-pod.yml

! nginx-pod.yml

! replica-set-1.yml X

! deployment.yml

! nginx-deployment.yml

```
! replica-set-1.yml > {} spec > # replicas
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: replica-set-1
5    labels:
6      app: replica-set-1
7      tier: replica-set-1
8  spec:
9    replicas: 5
10   selector:
11     matchLabels:
12       tier: replica-set-1
13   template:
14     metadata:
15       labels:
16         tier: replica-set-1
17     spec:
18       containers:
19         - name: busybox
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS D:\K8S> kubectl replace -f replica-set-1.yml
replicaset.apps/replica-set-1 replaced
PS D:\K8S> kubectl get pods -l tier=replica-set-1
NAME                                READY   STATUS    RESTARTS   AGE
replica-set-1-d8cdk                 1/1     Running   0           16m
replica-set-1-n57t9                 1/1     Running   0           19s
replica-set-1-npxn9                 1/1     Running   0           16m
replica-set-1-p59z9                 1/1     Running   0           19s
replica-set-1-pqkch                 1/1     Running   0           15m
PS D:\K8S> 
```

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

```
! redis-pod.yml x ! nginx-pod.yml ! replica-set-1.yml x ! deployment.yml ! nginx-deployment.yml
! replica-set-1.yml > {} spec > {} template > {} metadata > {} labels > tier
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: replica-set-1
5    labels:
6      app: replica-set-1
7      tier: replica-set-1
8  spec:
9    replicas: 5
10   selector:
11     matchLabels:
12       tier: replica-set-1
13   template:
14     metadata:
15       labels:
16         tier: replica-set-1
17     spec:
18       containers:
19         - name: busybox
20           image: busybox
21           command: [ "sleep", "3600" ]
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS D:\K8S> kubectl get pods -l tier=replica-set-1
NAME                                READY   STATUS    RESTARTS   AGE
replica-set-1-d8cdk                 1/1     Running   0           12m
replica-set-1-hbcqb                 1/1     Running   0           11m
replica-set-1-npxn9                 1/1     Running   0           12m
replica-set-1-pqkch                 1/1     Running   0           11m
replica-set-1-rsmfv                 1/1     Running   0           12m
PS D:\K8S>
```

## 10- 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?

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS D:\K8S> kubectl get pods -l tier=replica-set-1
NAME                                READY   STATUS    RESTARTS   AGE
replica-set-1-d8cdk                 1/1     Running   0           16m
replica-set-1-n57t9                 1/1     Running   0           50s
replica-set-1-npxn9                 1/1     Running   0           16m
replica-set-1-p59z9                 1/1     Running   0           50s
replica-set-1-pqkch                 1/1     Running   0           15m
PS D:\K8S> kubectl delete pod replica-set-1-pqkch
pod "replica-set-1-pqkch" deleted
PS D:\K8S> kubectl get pods -l tier=replica-set-1
NAME                                READY   STATUS    RESTARTS   AGE
replica-set-1-6qdgm                 1/1     Running   0           45s
replica-set-1-d8cdk                 1/1     Running   0           18m
replica-set-1-n57t9                 1/1     Running   0           2m33s
replica-set-1-npxn9                 1/1     Running   0           18m
replica-set-1-p59z9                 1/1     Running   0           2m33s
PS D:\K8S>
```

11- How many Deployments and ReplicaSets exist on the system?

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS D:\K8S> kubectl get replicaset
NAME          DESIRED  CURRENT  READY  AGE
replica-set-1  5        5        5      19m
PS D:\K8S> kubectl get deployments
No resources found in default namespace.
PS D:\K8S> 
```

12- create a Deployment with name= deployment-1 image= busybox replicas= 3

```
! redis-pod.yml ! nginx-pod.yml ! replica-set-1.yml ! deplyment-1.yml X
! deplyment-1.yml > {} spec > {} template > {} spec > [ ] containers > {} 0 > [ ] command
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: deployment-1
5    labels:
6      app: deployment-1
7  spec:
8    replicas: 3
9    selector:
10     matchLabels:
11       app: deployment-1
12   template:
13     metadata:
14       labels:
15         app: deployment-1
16     spec:
17       containers:
18       - name: busybox
19         image: busybox
20         command: [ "sleep", "3600" ]
21

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS D:\K8S> kubectl create -f deplyment-1.yml
deployment.apps/deployment-1 created
PS D:\K8S> kubectl get deployments
NAME          READY  UP-TO-DATE  AVAILABLE  AGE
deployment-1  3/3    3           3          14s
PS D:\K8S> 
```

13- How many Deployments and ReplicaSets exist on the system now?

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS D:\K8S> kubectl get deployments
NAME          READY    UP-TO-DATE    AVAILABLE    AGE
deployment-1   3/3      3             3            41s
PS D:\K8S> kubectl get replicaset
NAME          DESIRED    CURRENT    READY    AGE
deployment-1-5454847857  3          3          3        43s
replica-set-1  5          5          5        24m
PS D:\K8S>
```

14- How many pods are ready with the deployment-1?

```
PS D:\K8S> kubectl get deployment deployment-1
NAME          READY    UP-TO-DATE    AVAILABLE    AGE
deployment-1   3/3      3             3            112s
PS D:\K8S>
```

15- Update deployment-1 image to nginx then check the ready pods again

! redis-pod.yml ! nginx-pod.yml ! replica-set-1.yml ! deplyment-1.yml X

```
! deplyment-1.yml > {} spec > {} template > {} spec > [ ] containers > {} 0 > image
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: deployment-1
5    labels:
6      app: deployment-1
7  spec:
8    replicas: 3
9    selector:
10     matchLabels:
11       app: deployment-1
12   template:
13     metadata:
14       labels:
15         app: deployment-1
16     spec:
17       containers:
18         - name: busybox
19           image: nginx
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS D:\K8S> kubectl apply -f deplyment-1.yml
deployment.apps/deployment-1 configured
PS D:\K8S> kubectl get pods -l app=deployment-1
NAME                                READY   STATUS    RESTARTS   AGE
deployment-1-5b4d9b947-4wvck        1/1     Running   0           3m1s
deployment-1-5b4d9b947-62t5q        1/1     Running   0           3m9s
deployment-1-5b4d9b947-wkpm4        1/1     Running   0           3m5s
PS D:\K8S>
```



16- Run `kubectl describe deployment deployment-1` and check events What is the deployment strategy used to upgrade the deployment-1?

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS D:\K8S> kubectl describe deployment deployment-1
Name: deployment-1
Namespace: default
CreationTimestamp: Sat, 01 Mar 2025 23:47:09 +0200
Labels: app=deployment-1
Annotations: deployment.kubernetes.io/revision: 2
Selector: app=deployment-1
Replicas: 3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels: app=deployment-1
  Containers:
    busybox:
      Image: nginx
      Port: <none>
      Host Port: <none>
      Command:
        sleep
        3600
      Environment: <none>
      Mounts: <none>
```

## 17- Rollback the deployment-1 What is the used image with the deployment-1?

```
PS D:\K8S> kubectl rollout undo deployment deployment-1
deployment.apps/deployment-1 rolled back
PS D:\K8S> kubectl get deployment deployment-1
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1  3/3     3            3           12m
PS D:\K8S> kubectl describe deployment deployment-1
Name:          deployment-1
Namespace:     default
CreationTimestamp: Sat, 01 Mar 2025 23:47:09 +0200
Labels:        app=deployment-1
Annotations:   deployment.kubernetes.io/revision: 3
Selector:      app=deployment-1
Replicas:      3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=deployment-1
  Containers:
    busybox:
      Image:      busybox
      Port:       <none>
      Host Port:  <none>
      Command:
        sleep
        3600
      Environment:  <none>
      Mounts:       <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
```

18- Create a deployment using nginx image with latest tag only and remember to mention tag i.e nginx:latest and name it as nginx-deployment. App labels should be app: nginx-app and type: front-end. The container should be named as nginx-container; also make sure replica counts are 3.

! redis-pod.yml ! nginx-pod.yml ! replica-set-1.yml ! deployment-1.yml ! nginx-deployment.yml X

```
! nginx-deployment.yml > {} spec > {} template > {} spec > [ ] containers > {} 0 > [ ] ports > {} 0
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: nginx-deployment
5    labels:
6      app: nginx-app
7      type: front-end
8  spec:
9    replicas: 3
10   selector:
11     matchLabels:
12       app: nginx-app
13       type: front-end
14   template:
15     metadata:
16       labels:
17         app: nginx-app
18         type: front-end
19     spec:
20       containers:
21       - name: nginx-container
22         image: nginx:latest
23         ports:
24         - containerPort: 80
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS D:\K8S> kubectl create -f nginx-deployment.yml
deployment.apps/nginx-deployment created
PS D:\K8S> kubectl get deployment nginx-deployment
NAME          READY  UP-TO-DATE  AVAILABLE  AGE
nginx-deployment  3/3    3           3          12s
PS D:\K8S>
```