

Lab 1 – Kubernetes

Toka Abdelhakim

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

Command: `Kubectl run redis --image redis`

Output:

redis	0/1	ContainerCreating	0	5s
-------	-----	-------------------	---	----

3- Create a pod with the name `nginx` and with the image `nginx123`.
Use a pod-definition YAML file. And yes, the image name is wrong!

Yaml file:

```
pod.yaml x
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: nginx
5  spec:
6    containers:
7      - name: nginx
8        image: nginx123
9
```

4- What is the `nginx` pod status?

Command: `Kubectl create -f pod.yaml`

Output:

nginx	0/1	ErrImagePull	0	40s
-------	-----	--------------	---	-----

5- Change the `nginx` pod image to `nginx` check the status again

Command: `Kubectl apply -f pod.yaml`

Output:

NAME	READY	STATUS	RESTARTS	AGE
nginx	1/1	Running	0	96s

6- How many ReplicaSets exist on the system?

Two replicaset

Command: `kubectl get replicaset -n kube-system`

Output:

NAME	DESIRED	CURRENT	READY	AGE
coredns-fb8b8dccc	2	2	2	36m
katacoda-cloud-provider-7c4ff69fcb	1	1	1	36m

controlplane \$

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

Yaml file:

```
replicaset.yaml x replicaset-defintion.ya... x
1  apiVersion: apps/v1
2  kind: ReplicaSet
3  metadata:
4    name: replica-set-1
5    labels:
6      name: myapp
7      type: front-end
8  spec:
9    template:
10   metadata:
11     name: myapp-pod
12     labels:
13       name: myapp
14       type: front-end
15   spec:
16     containers:
17     - name: replica-set-pod
18       image: busybox
19   replicas: 3
20   selector:
21     matchLabels:
22       type: front-end
```

Command: `Kubectl create -f replicaset.yaml`

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

Change the number of replicase in file

Command: `Kubectl apply -f replicaset.yaml`

```
controlplane $ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
replica-set-1-f6zw6                 1/1     Running   0           2m50s
replica-set-1-fbzw4                 1/1     Running   0           2m50s
replica-set-1-rcpbs                  1/1     Running   0           94s
replica-set-1-tchtg                 1/1     Running   0           2m50s
replica-set-1-v7476                 1/1     Running   0           94s
controlplane $
```

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

No-one

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?

Because the replicaset creates the 5th pod to keep number of replicas.

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

Kube-system: Two of Deployments and ReplicaSets

Output:

```
controlplane $ kubectl get deployment -n kube-system
NAME                                READY  UP-TO-DATE  AVAILABLE  AGE
coredns                            2/2    2           2          74m
kacoda-cloud-provider              0/1    1           0          74m
controlplane $
```

My create: No- deployment One Replicaset

Output:

```
controlplane $ kubectl get deployment
No resources found.
controlplane $ kubectl get replicaset
NAME            DESIRED  CURRENT  READY  AGE
replica-set-1   5        5        5      15m
controlplane $
```

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

yaml file:

```
deploy.yaml x
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: deployment-1
5    labels:
6      name: myapp
7      type: front-end
8  spec:
9    template:
10     metadata:
11       name: myapp-pod
12     labels:
13       name: myapp
14       type: front-end
15     spec:
16       containers:
17       - name: replica-set-pod
18         image: busybox
19     replicas: 3
20     selector:
21       matchLabels:
22         type: front-end
23
```

command:

```
controlplane $ kubectl create -f deploy.yaml
deployment.apps/deployment-1 created
controlplane $
```

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

One Deployment – Two ReplicaSets

```
controlplane $ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment-1  3/3     1            3           2m18s
controlplane $ kubectl get replicaset
NAME          DESIRED   CURRENT   READY   AGE
deployment-1-97f7484d  1         1         0       2m49s
replica-set-1  3         3         3       25m
controlplane $
```

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

No-one

```
controlplane $ kubectl get pods
NAME                               READY   STATUS             RESTARTS   AGE
deployment-1-6db9c4bbc8-nbbj2     0/1     ContainerCreating   0          3s
deployment-1-6db9c4bbc8-qrbm9     0/1     ContainerCreating   0          3s
deployment-1-6db9c4bbc8-vvqgh     0/1     ContainerCreating   0          3s
```

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

Three pods are ready

```
NAME                                READY   STATUS    RESTARTS   AGE
deployment-1-79648bff8d-2rsvp      1/1     Running   0           24s
deployment-1-79648bff8d-t2smq      1/1     Running   0           24s
deployment-1-79648bff8d-z42t4      1/1     Running   0           24s
controlplane $
```

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

RollingUpdate strategy

```
Conditions:
  Type           Status  Reason
  ----           -
  Available      False   MinimumReplicasUnavailable
  Progressing    True    ReplicaSetUpdated
OldReplicaSets: <none>
NewReplicaSet:  deployment-1-6db9c4bbc8 (3/3 replicas created)
Events:
  Type       Reason              Age   From          Message
  ----       -
  Normal     ScalingReplicaSet   2m26s deployment-controller Scaled up replica set deployment-1-6db9c4bbc8 to 3
controlplane $
```

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

Command: `controlplane $ kubectl rollout undo deployment/deployment-1`
`deployment.extensions/deployment-1 rolled back`
`controlplane $`

Image is busybox

```
Labels: name=myapp
        type=front-end
Containers:
  deployment-pod:
    Image:          busybox
    Port:           <none>
    Host Port:      <none>
```

18- How many Namespaces exist on the system?

Command: `kubectl get namespace`

Output:

```
kube-system
controlplane $ kubectl get namespace
NAME                STATUS    AGE
default             Active    40m
kube-node-lease     Active    41m
kube-public         Active    41m
kube-system         Active    41m
controlplane $
```

19- How many pods exist in the kube-system namespace?

Command: `kubectl get pods --namespace=kube-system`

Output:

```
controlplane $ kubectl get pods --namespace=kube-system
NAME                                READY   STATUS    RESTARTS   AGE
coredns-fb8b8dcdf-86dh7            1/1     Running   1           47m
coredns-fb8b8dcdf-dnmpm            1/1     Running   1           47m
etcd-controlplane                  1/1     Running   0           47m
katakoda-cloud-provider-9fb8587c-m4d5z 0/1     CrashLoopBackOff 15          47m
kube-apiserver-controlplane         1/1     Running   0           47m
kube-controller-manager-controlplane 1/1     Running   2           47m
kube-keepalived-vip-9xts8           1/1     Running   0           47m
kube-proxy-8df4p                   1/1     Running   0           47m
kube-proxy-qxwkh                    1/1     Running   0           47m
kube-scheduler-controlplane         1/1     Running   2           47m
weave-net-567sr                     2/2     Running   1           47m
weave-net-t9zxz                     2/2     Running   1           47m
controlplane $
```

20- Create a deployment with Name: beta Image: redis Replicas: 2 Namespace: finance Resources Requests: CPU: .5 vcpu Mem: 1G Resources Limits: CPU: 1 vcpu Mem: 2G

Files:

```
deployment.yaml× namespace.yaml×
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: beta
5    namespace: finance
6  spec:
7    replicas: 2
8    selector:
9      matchLabels:
10       app: test-namespace
11    # use replica set definition
12  template:
13    metadata:
14      labels:
15       app: test-namespace
16    spec:
17      containers:
18      - image: redis
19        name: redis-deploy
20        resources:
21          requests:
22            memory: "1G"
23            cpu: "1"
24          limits:
25            memory: "2G"
26            cpu: "5"
27
```

```
deployment.yaml× namespace.yaml×
1  apiVersion: v1
2  kind: Namespace
3  metadata:
4    name: finance
5
```

Confirm Creation:

```
controlplane $ kubectl get pods --namespace=finance
NAME                                READY   STATUS    RESTARTS   AGE
beta-6cf66db7c4-9z5z8              0/1     Pending   0           2s
beta-6cf66db7c4-ns4xr              0/1     Pending   0           2s
```

```
controlplane $ kubectl describe -f deploy.yaml
Name:                                beta
Namespace:                           finance
CreationTimestamp:                    Thu, 16 Sep 2021 14:32:32 +0000
```