




[2] Pod

Pod

- : k8s
- Create a new namespace and create a pod in the namespace.
 - namespace name: ecommerce
 - Pod Name: eshop-main
 - image: nginx:1.17
 - env: DB=mysql



```
#    
[user@console ~]$ kubectl config use-context k8s  
  
# namespace   
[user@console ~]$ kubectl create namespace ecommerce  
  
#  namespace   
[user@console ~]$ kubectl get namespaces  
  
# pod   
#    --dry-run=client          
[user@console ~]$ kubectl run eshop-main --image=nginx:1.17 --env=DB=mysql --namespace=ecommerce --  
dry-run=client  
  
#   --dry-client       
[user@console ~]$ kubectl run eshop-main --image=nginx:1.17 --env=DB=mysql --namespace=ecommerce  
  
# Pod    
[user@console ~]$ kubectl get pods --namespace ecommerce
```

□□□□ (2□)

Create 2 nginx image pods in which one of them is labelled with env=prod and another one labelled with env=dev and verify the same

- □□

```
kubectl run nginx-prod --image=nginx --labels=env=prod
```

```
kubectl run nginx-dev --image=nginx --labels=env=dev
```

```
# □□  
kubectl get pods  
  
kubectl get pod nginx-prod -o yaml  
--> labels□□ □□  
  
kubectl get pod nginx-dev -o yaml  
--> labels□□ □□
```

□□□□ (3□)

Create an nginx pod and list the pod with different levels of verbosity

- □□

```
kubectl run nginx --image=nginx
```

```
kubectl get pod nginx --v=7
```

```
kubectl get pod nginx --v=8
```

```
kubectl get pod nginx --v=9
```

- □□ (verbosity)

<https://kubernetes.io/ko/docs/reference/kubectl/cheatsheet/>

□□□□ (4□)

Create a pod with environment variables as var1=value1. Check the environment variable in pod

- □□

```
kubectl run test-pod --image=nginx --env=var1=value1
```

```
# □□
```

```
kubectl get pods
```

```
kubectl describe pod test-pod
```

```
--> env □□ □□
```

□□□□ (6□)

List "nginx-dev" and "nginx-prod" pod and delete those pods

- □□

```
# □□ □□□□□□ □□ pod □□
```

```
kubectl get pods -A
```

```
nginx-dev, nginx-prod pod □□ □□
```

```
kubectl delete pod nginx-dev
```

```
kubectl delete pod nginx-prod
```

□□□□ (8□)

Create a pod as follows:

- Name: non-persistent-redis
- container Image: redis
- Volume with name: cache-control
- Mount path: /data/redis

The pod should launch in the staging namespace and the volume must not be persistent.

- `❏`

```
kubectl create namespace staging
```

```
# YAML ❏ ❏  
vi volume.yaml  
  
apiVersion: v1  
kind: Pod  
metadata:  
  creationTimestamp: null  
  labels:  
    run: non-persistent-redis  
  name: non-persistent-redis  
  namespace: staging  
spec:  
  containers:  
  - image: redis  
    name: redis  
    resources: {}  
    volumeMounts:  
    - mountPath: /data/redis  
      name: cache-control  
  volumes:  
  - name: cache-control  
    emptyDir: {}  
  dnsPolicy: ClusterFirst  
  restartPolicy: Always  
status: {}  
  
:wq
```

```
kubectl apply -f volume.yaml
```

❏❏❏ (9❏)

Create a pod with i traffic on port 80

-

```
kubectl run nginx --image=nginx --port=80
```

(11)

Create a file: /opt/KUCC00302/kucc00302.txt that lists all pods that implement service baz in namespace development. The format of the file should be one pod name per line.

-

```
# baz  service   
kubectl get svc baz -n development
```

```
kubectl describe svc baz -n development  
-->  name=foo  .
```

```
kubectl get pods -l name=foo -n development
```

```
kubectl get pods -l name=foo -n development -o NAME > /opt/KUCC00302/kucc00302.txt
```

(12)

Create a pod named kucc8 with a single app container for each of the following images running inside (there may be between 1 and 4 images specified): nginx + redis + memcached.

-

```
kubectl run kucc8 --image=nginx --dry-run=client -o yaml  
--> yaml 
```

```
vi kucc.yaml  
  
apiVersion: v1  
kind: Pod  
metadata:
```

```
creationTimestamp: null
labels:
  run: kucc8
name: kucc8
spec:
  containers:
  - image: nginx
    name: nginx
  - image: redis
    name: redis
  - image: memcached
    name: memcached
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}

:wq
```

```
kubectl apply -f kucc.yaml
```

```
# []
kubectl get pods

kubectl describe pod kucc8
--> [] [] 3 [] [] [] []
```

0000 (180)

Check the image version in pod name nginx without the describe command

- []

```
kubectl get pod nginx -o jsonpath='{.spec.container[].image}'
```

0000 (190)

Print pod name and start time to "/opt/pod-status" file

-

```
# json    start time      
kubectl get pods -o json
```

```
kubectl get pods -o
jsonpath='{range.items[*]}{.metadata.name}{"\t"}{.status.startTime}{"\n"}{end}' > /opt/pod-
status
```

(23)

Get IP address of the pod "nginx-dev"

-

```
kubectl get pod nginx-dev -o jsonpath='{.status.podIP}{"\n"}'
```

(24)

List all the pods sorted by name

-

```
kubectl get pods --sort-by=.metadata.name
```

(26)

List the nginx pod with custom columns POD_NAME and POD_STATUS

-

```
kubectl get pod -o=custom-columns="POD_NAME:.metadata.name,
POD_STATUS:.status.containerStatuses[.state]"
```

-

□□□□ (27□)

List all the pods sorted by created timestamp

- □□

```
kubectl get pods --sort-by=.metadata.creationTimestamp
```

□□□□ (29□)

Get list of all pods in all namespaces and write it to file "/opt/pods-list.yaml"

- □□

```
kubectl get pods -A > /opt/pods-list.yaml
```

□□□□ (34□)

Get list of all the pods showing name and namespace with a jsonpath expression.

- □□

```
kubectl get pods -o jsonpath='{range .items[*]} {.metadata.name}{"\t"} {.metadata.namespace}{"\n"}{end}'
```

□□□□ (38□)

Create a busybox pod and add "sleep 3600" command

- □□

```
kubectl run busybox --image=busybox -- /bin/sh -c "sleep 3600"
```

□□□□ (42□)

Create a nginx pod with label env=test in engineering namespace

- □□

```
kubectl create namespace engineering
```

```
kubectl run nginx --image=nginx --labels=env=test -n engineering
```

□□□□ (44□)

Check the Image version of nginx-dev pod using jsonpath

- □□

```
kubectl get pod nginx-dev -o json
```

```
kubectl get pod nginx-dev -o jsonpath='{.spec.containers[].image}'{"\n"}
```

□□□□ (57□)

List all the pods showing name and namespace with a json path expression

- □□

```
kubectl get pods -A -o=jsonpath='{range .items[*]}{.metadata.name}{"\t"}{.metadata.namespace}{"\n"}{end}'
```

□□□□ (59□)

Create a pod as follows: Name: mongo Using Image: mongo In a new Kubernetes namespace named: my-website

