

# DEVEPOS

## DAY 4 Task

### Pod

1. Create a pod using run command

```
$ kubectl run <pod-name> --image=<image-name> --port=<container-port>
```

```
$ kubectl run my-pod --image=nginx --port=80
```

2. View all the pods

(In default namespace)

```
$ kubectl get pods
```

(In All namespace)

```
$ kubectl get pods -A
```

# For a specific namespace

```
$ kubectl get pods -n kube-system
```

# For a specific type

```
$ kubectl get pods <pod-name>
```

```
$ kubectl get pods <pod-name> -o wide
```

```
$ kubectl get pods <pod-name> -o yaml
```

```
$ kubectl get pods <pod-name> -o json
```

3. Describe a pod (View Pod details)

```
$ kubectl describe pod <pod-name>
```

```
$ kubectl describe pod my-pod
```

4. View Logs of a pod

```
$ kubectl logs <pod-name>
```

```
$ kubectl logs my-pod
```

5. Execute any command inside Pod (Inside Pod OS)

```
$ kubectl exec <pod-name> -- <command>
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
  name: my-pod
```

```
  labels:
```

```
    app: my-web-app
```

```
    type: backend
```

```
spec:
```

```
  containers:
```

```
    - name: nginx-container
```

```
      image: nginx
```

```
      ports:
```

```
        - containerPort: 80
```

```

$ kubectl run secrets/kubernetes.io/serviceaccount from kube-api-access-tnbvj (ro)
Conditions:
  Type             Status
PodReadyToStartContainers    True
  Initialized       True
  Ready             True
  ContainersReady    True
  PodScheduled      True
Volumes:
  kube-api-access-tnbvj:
    Type: Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName: kube-root-ca.crt
    ConfigMapOptional: nil
    DownwardAPI: true
    BearerToken: none
QoS Class:
Node-Selectors:
Tolerations:
node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:
  Type    Reason      Age    From          Message
  ----    -
Normal   Scheduled   3ms    default-scheduler   Successfully assigned default/my-app to minikube
Normal   Pulling     3m1s   kubelet         Pulling image "nginx"
Normal   Completed   2m56s  kubelet         Successfully pulled image "nginx" in 4.224s (4.224s including waiting). Image size: 19
2909252 bytes
Normal   Created     2m56s  kubelet         Created container my-app-container
Normal   Started     2m56s  kubelet         Started container my-app-container
$ kubectl get pod
NAME                READY   STATUS    RESTARTS   AGE
my-app              1/1    Running   0           267s
nginx-586947778c-cp9hr
$ kubectl exec $(kubectl get pod -o jsonpath='{.items[0].metadata.name}') --
$

```

```

type: RollingUpdate
template:
  metadata:
    labels:
      apptype: web-backend
  spec:
    containers:
      - name: my-app
        image: nginx
        ports:
          - containerPort: 7070

```

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
  labels:
    app: my-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
        - name: my-app
          image: nginx:latest
          ports:
            - containerPort: 7070

```

## Deploy

1. Create Deployment by executing above YAML file

```
$ kubectl create -f web-deploy.yml
```

# Do necessary modifications if exist, else create new

```
$ kubectl create -f web-deploy.yml
```

# Completely Modify Pod Template

```
$ kubectl replace -f web-deploy.yml
```

#Create deploy

```
kubectl create deployment webnginx2 --image=nginx:latest --replicas=1
```

2. View Deployments

```
$ kubectl get deployments
```

```
$ kubectl get deploy
```

```
$ kubectl get deploy -o wide
```

```
$ kubectl get deploy <deployment-name> -o json
```

```
$ kubectl get deploy <deployment-name> -o yaml
```

3. View Deployment Description

```
$ kubectl describe deploy <deployment-name>
```

4. We can modify generated/updated YAML file

```
$ kubectl edit deploy <deployment-name>
```

```
## change replicas: count to any other value then (ESC):wq
```

# We can modify our YAML file and then execute apply command

```
$ kubectl apply -f web-deploy.yml
```

## We can Even scale using command also

```
$ kubectl scale deploy <deployment-name> --replicas=<desired-replica-count>
```

## 5. Delete Deployment

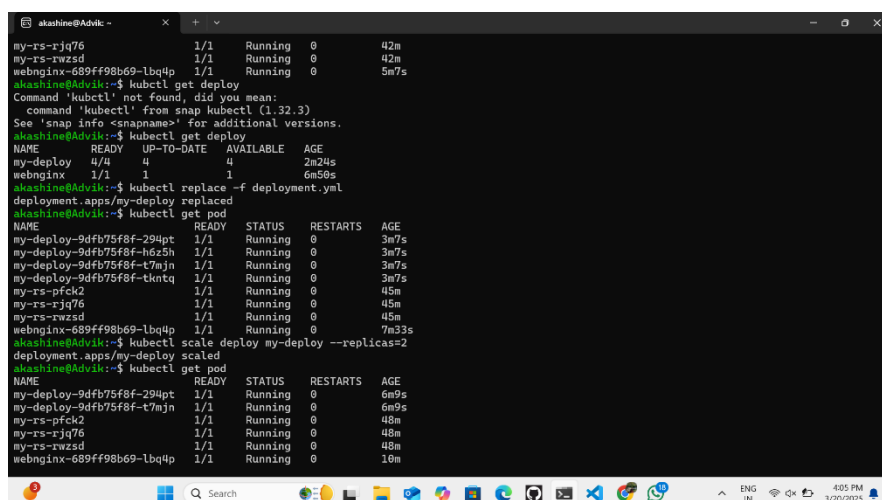
```
$ kubectl delete deploy <deployment-name>
```

```
$ kubectl delete -f web-deploy.yml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
        - name: my-app
          image:
          ports:
            - containerPort: 7070
```

---

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-service
    type: backend-app
spec:
  type: NodePort
  ports:
    - targetPort: 7070
      port: 7070
      nodePort: 30002
  selector:
    apptype: web-backend
```



A terminal window showing a series of Kubernetes commands and their outputs. The user is in a shell on a machine named 'akashine@Advik'. The commands and outputs are as follows:

```
akashine@Advik:~$ kubectl get deploy
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
my-deploy     4/4     4             4           2m24s
webnginx      1/1     1             1           6m50s
akashine@Advik:~$ kubectl get deploy
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
my-deploy     4/4     4             4           2m24s
webnginx      1/1     1             1           6m50s
akashine@Advik:~$ kubectl replace -f deployment.yml
deployment.apps/my-deploy replaced
akashine@Advik:~$ kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
my-deploy-9dfb75f8f-294pt  1/1     Running   0           3m7s
my-deploy-9dfb75f8f-h6z5h  1/1     Running   0           3m7s
my-deploy-9dfb75f8f-t7mjn  1/1     Running   0           3m7s
my-deploy-9dfb75f8f-tkntq  1/1     Running   0           3m7s
my-ts-pfck2      1/1     Running   0           45m
my-ts-rjq76      1/1     Running   0           45m
my-ts-rwzsd      1/1     Running   0           45m
webnginx-689ff98b69-lbq4p  1/1     Running   0           7m33s
akashine@Advik:~$ kubectl scale deploy my-deploy --replicas=2
deployment.apps/my-deploy scaled
akashine@Advik:~$ kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
my-deploy-9dfb75f8f-294pt  1/1     Running   0           6m9s
my-deploy-9dfb75f8f-t7mjn  1/1     Running   0           6m9s
my-ts-pfck2      1/1     Running   0           48m
my-ts-rjq76      1/1     Running   0           48m
my-ts-rwzsd      1/1     Running   0           48m
webnginx-689ff98b69-lbq4p  1/1     Running   0           10m
```

## Minikube Service

Minikube service

#need to create a yml file  
sudo nano deployment.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
        - name: my-app
          image:
            ports:
              - containerPort: 9000
---
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-service
spec:
  type: NodePort
  ports:
    - port: 9000
      targetPort: 8080
      nodePort: 30002
  selector:
    apptype: web-backend
```

#Apply the deployment  
kubectl apply -f deployment.yml

#replace the deployment  
kubectl replace -f deployment.yml

#Run the service  
minikube service my-service

#curl the url  
curl <url>/<file\_name>/

```
akashine@Advik: ~$ kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
my-deploy-6f6bc8b66d-hsrhb         1/1     Running   0           2m57s
my-rs-pfck2                         1/1     Running   0           175m
my-rs-rjg76                        1/1     Running   0           175m
my-rs-rwzsd                        1/1     Running   0           175m
webnginx-689ff98b69-lbq4p          1/1     Running   0           137m
akashine@Advik:~$ kubectl exec -it my-deploy-6f6bc8b66d-hsrhb -- /bin/bash/
OCI runtime exec failed: exec failed: unable to start container process: exec: "/bin/bash/": stat /bin/bash/: not a directory: unknown
n: Are you trying to mount a directory onto a file (or vice-versa)? Check if the specified host path exists and is the expected type
command terminated with exit code 126
akashine@Advik:~$ kubectl exec -it my-deploy-6f6bc8b66d-hsrhb -- /bin/bash
root@my-deploy-6f6bc8b66d-hsrhb:/usr/local/tomcat# ls
bin          conf          filtered-KEYS  LICENSE      native-jni-lib  README.md     RUNNING.txt   upstream-KEYS  webapps.dist
BUILDING.txt  CONTRIBUTING.md  lib          NOTICE      RELEASE-NOTES  src          webapps      work
exit
akashine@Advik:~$ curl http://192.168.49.2:30002/webapps/
<doctype html><html lang="en"><head><title>HTTP Status 404 - Not Found</title><style type="text/css">body {font-family:Tahoma,Arial,sans-serif;} h1, h2, h3, h4 {color:white;background-color:#525276;} h1 {font-size:22px;} h2 {font-size:16px;} h3 {font-size:14px;} p {font-size:12px;} a {color:black;} .line {height:1px;background-color:#525276;border:none;}</style></head><body><h1>HTTP Status 404 - Not Found</h1><hr class="line" /><p><b>Type</b></p><p><b>Status Report</b><p><b><b>Description</b></b> The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.</p><hr class="line" /><h3>Apache Tomcat/9.0.102</h3></body></html></C
akashine@Advik:~$ curl http://192.168.49.2:30002/maven-web-app
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
akashine@Advik:~$
```

## Namespace

```
# To create a namespace:
$ kubectl create namespace <namespace-name>
$ kubectl create ns my-bank
# To switch to a specific namespace: (make this as default type)
$ kubectl config set-context --current --namespace=<namespace-name>
# To list all namespaces:
$ kubectl get namespaces
# To get resources within a specific namespace:
$ kubectl get <resource-type> -n <namespace-name>
$ kubectl get deploy -n my-bank
$ kubectl get deploy --namespace my-bank
$ kubectl get all --namespace my-bank
# To delete a namespace and all associated resources:
$ kubectl delete namespace <namespace-name>
$ kubectl delete ns my-bank
```

```
kubectl create ns my-deploy
kubectl apply -f deploy.yml -n mydeploy
```

```
apiVersion: v1
kind: Namespace
metadata:
  name: my-demo-ns
```

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
  namespace: my-demo-ns
spec:
  containers:
  - name: my-container
    image: nginx:latest
```

## Namespace

```
service/my-service created
akashine@Advik:~$ kubectl get pod -n my-deploy
NAME          READY   STATUS    RESTARTS   AGE
my-deploy-6f6bc8b66d-m5mzq  1/1     Running   0           4m37s
akashine@Advik:~$ kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
my-deploy-6f6bc8b66d-hsrhb  1/1     Running   0           29m
my-rs-pfck2  1/1     Running   0           3h21m
my-rs-rjg76  1/1     Running   0           3h21m
my-rs-twzsd  1/1     Running   0           3h21m
webnginx-689ff98b69-lbq4p  1/1     Running   0           164m
akashine@Advik:~$ kubectl get pod -n my-deploy
NAME          READY   STATUS    RESTARTS   AGE
my-deploy-6f6bc8b66d-m5mzq  1/1     Running   0           5m16s
akashine@Advik:~$ kubectl get deploy -n my-deploy
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
my-deploy     1/1     1             1           5m38s
akashine@Advik:~$ kubectl get all --namespaces
error: unknown flag: --namespaces
See 'kubectl get --help' for usage.
akashine@Advik:~$ kubectl get all --namespace my-deploy
NAME          READY   STATUS    RESTARTS   AGE
pod/my-deploy-6f6bc8b66d-m5mzq  1/1     Running   0           6m27s
NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
service/my-service  NodePort      10.99.26.253  <none>         7070:30001/TCP   6m27s
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/my-deploy  1/1     1             1           6m27s
NAME          DESIRED   CURRENT   READY   AGE
replicaset.apps/my-deploy-6f6bc8b66d  1         1         1       6m27s
akashine@Advik:~$
```

## Namespace yml

```
webnginx-689ff98b69-lbq4p  1/1     Running   0           164m
akashine@Advik:~$ kubectl get pod -n my-deploy
NAME          READY   STATUS    RESTARTS   AGE
my-deploy-6f6bc8b66d-m5mzq  1/1     Running   0           5m16s
akashine@Advik:~$ kubectl get deploy -n my-deploy
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
my-deploy     1/1     1             1           5m38s
akashine@Advik:~$ kubectl get all --namespaces
error: unknown flag: --namespaces
See 'kubectl get --help' for usage.
akashine@Advik:~$ kubectl get all --namespace my-deploy
NAME          READY   STATUS    RESTARTS   AGE
pod/my-deploy-6f6bc8b66d-m5mzq  1/1     Running   0           6m27s
NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
service/my-service  NodePort      10.99.26.253  <none>         7070:30001/TCP   6m27s
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/my-deploy  1/1     1             1           6m27s
NAME          DESIRED   CURRENT   READY   AGE
replicaset.apps/my-deploy-6f6bc8b66d  1         1         1       6m27s
akashine@Advik:~$ sudo nano mydeploy.yml
[sudo] password for akashine:
akashine@Advik:~$ kubectl apply -f mydeploy.yml
namespace/my-demo-ns created
akashine@Advik:~$ sudo nano nspod.yml
akashine@Advik:~$ kubectl apply -f nspod.yml
pod/my-pod created
akashine@Advik:~$ kubectl get pod -n my-demo-ns
NAME          READY   STATUS    RESTARTS   AGE
my-pod        1/1     Running   0           36s
akashine@Advik:~$
```