DevOps-Day 04:Task_Completed

Command:

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


kind: ReplicaSet
metadata:
  name: my-rs
  labels:
    name: my-rs
spec:
  replicas: 4
  selector:
    matchLabels:
      apptype: web-backend
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
      - name: my-app
        image:
        ports:
          - containerPort: 8080
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 4
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
```

```
containers:
- name: my-app
  image:
  ports:
        - containerPort: 7070
```

Services (short name = svc):
Service is an abstraction that defines a logical set of pods and a policy to access them. Services enable network connectivity and load balancing to the pods that are part of the service, allowing other components within or outside the cluster to interact with the application.
Service Types: Kubernetes supports different types of services:
1. NodePort: Exposes the service on a static port on each selected node's IP. This type makes the service accessible from outside the cluster by the <NodeIP>:<NodePort> combination.

2. ClusterIP: Exposes the service on a cluster-internal IP. This type makes the service only reachable within the cluster.

3. LoadBalancer: Creates an external load balancer in cloud environments, which routes traffic to the service.
2. 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

3. 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
4. View Deployment Description
$ kubectl describe deploy <deployment-name>
5. 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 usi…
[2:33 PM, 3/20/2025] +91 90928 13114: 2. Create ReplicaSet by executing above YAML file
$ kubectl create -f rs-test.yml
# Do necessary modifications if exist, else create new
$ kubectl apply -f rs-test.yml
# Completely Modify Pod Template
$ kubectl replace –f rs-test.yml

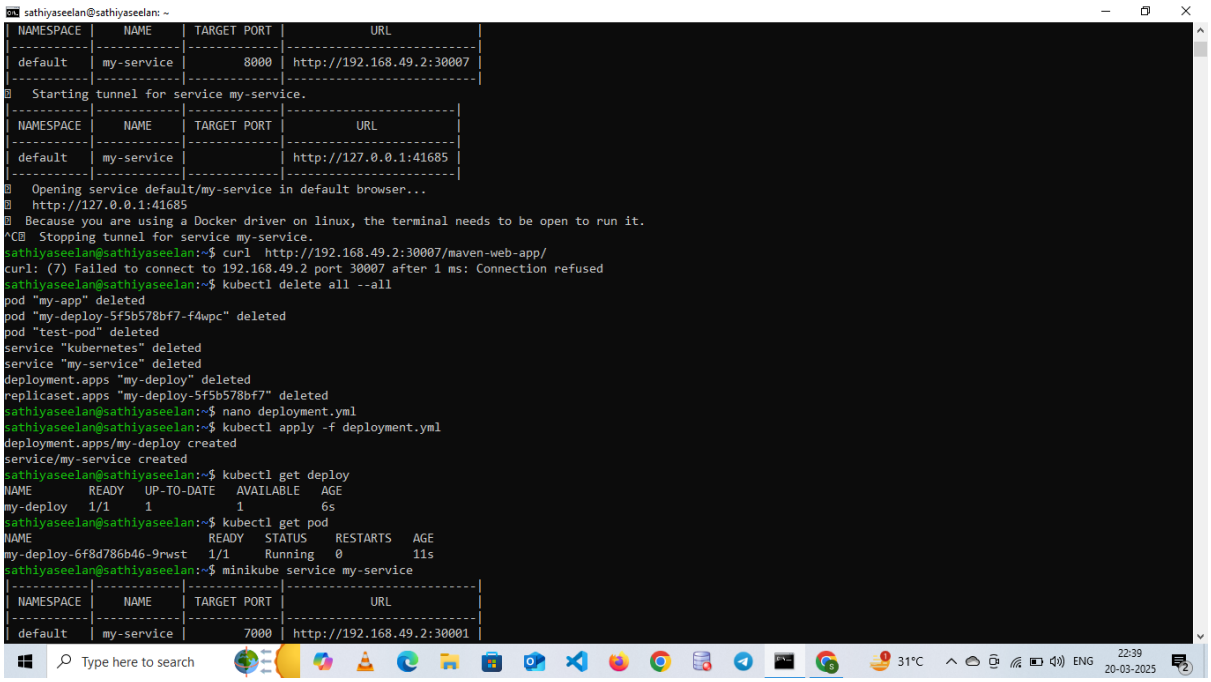
3. View ReplicaSets
$ kubectl get replicasets
$ kubectl get rs
$ kubectl get rs –o wide
$ kubectl get rs <replica-set-name> –o json
$ kubectl get rs <replica-set-name> –o yaml

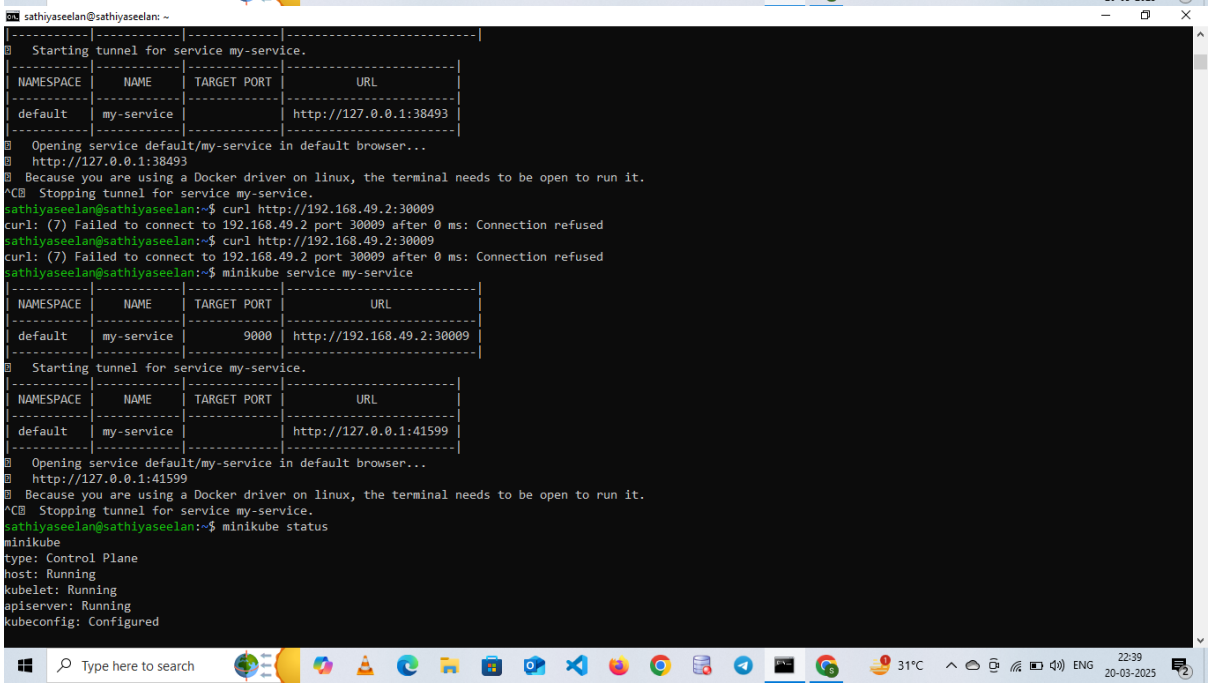
4. View ReplicaSet Description
$ kubectl describe rs <replica-set-name>
5. We can modify generated/updated YAML file
$ kubectl edit rs <replica-set-name>
change replicas: count to any other value then (ESC):wq

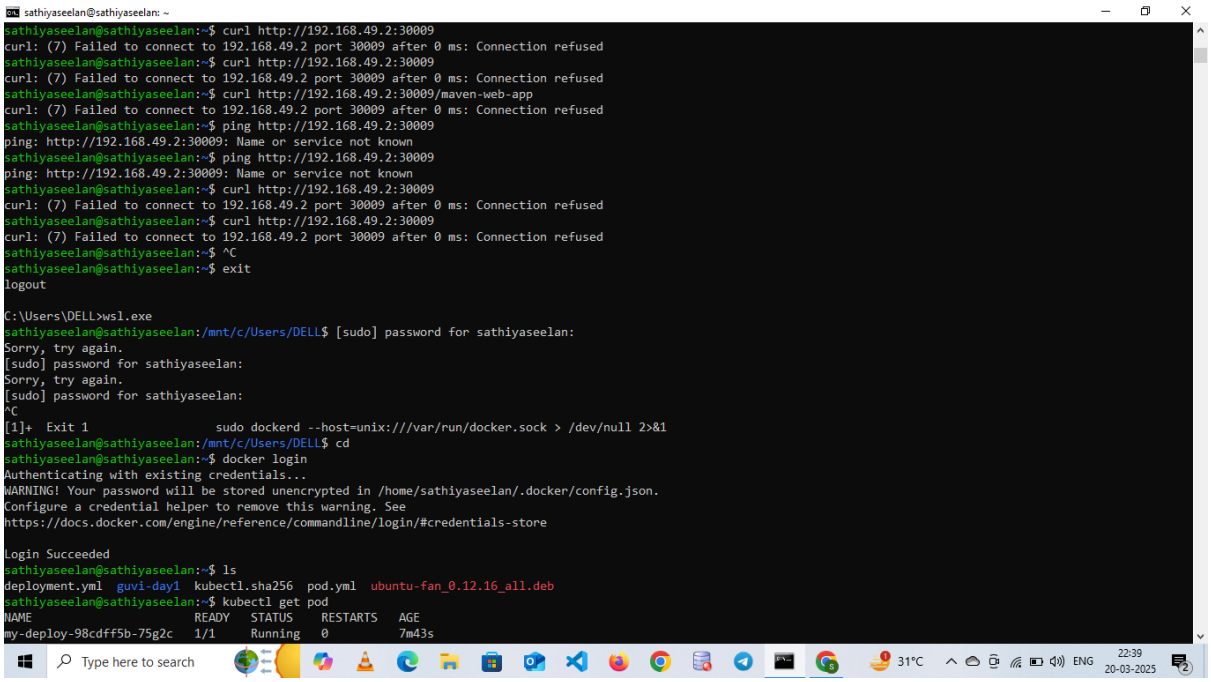
# We can modify our YAML file and then execute apply command
$ kubectl apply -f rs-test.yml

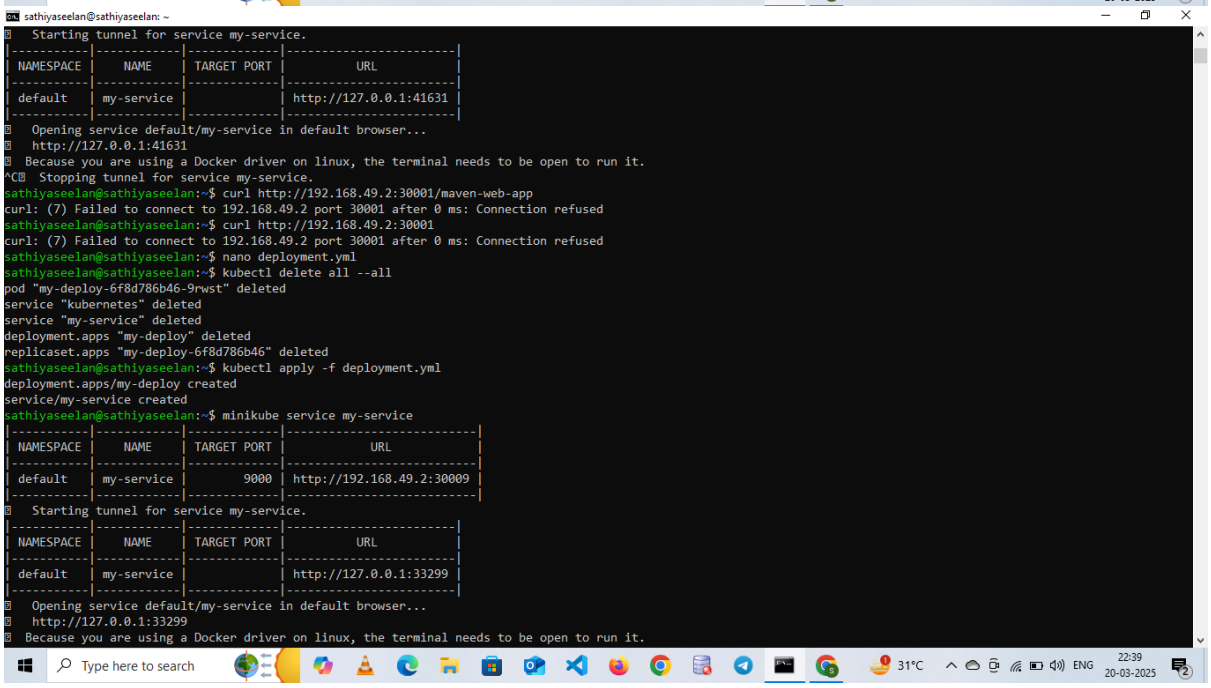
We can Even scale using command also
$ kubectl scale replicaset <replicaset-name> --replicas=<desired-replica-count>

6. Delete ReplicaSet
$ kubectl delete rs <replica-set-name>
$ kubectl delete -f rs-test.yml


sathiyaseelan@sathiyaseelan: ~
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | 8000 | http://192.168.49.2:30007 |
Starting tunnel for service my-service.
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | | http://127.0.0.1:41685 |
Opening service default/my-service in default browser...
http://127.0.0.1:41685
Because you are using a Docker driver on linux, the terminal needs to be open to run it.
^C Stopping tunnel for service my-service.
sathiyaseelan@sathiyaseelan:~$ curl http://192.168.49.2:30007/maven-web-app/
curl: (7) Failed to connect to 192.168.49.2 port 30007 after 1 ms: Connection refused
sathiyaseelan@sathiyaseelan:~$ kubectl delete all --all
pod "my-app" deleted
pod "my-deploy-5f5b578bf7-f4wpc" deleted
pod "test-pod" deleted
service "kubernetes" deleted
service "my-service" deleted
deployment.apps "my-deploy" deleted
replicaset.apps "my-deploy-5f5b578bf7" deleted
sathiyaseelan@sathiyaseelan:~$ nano deployment.yml
sathiyaseelan@sathiyaseelan:~$ kubectl apply -f deployment.yml
deployment.apps/my-deploy created
service/my-service created
sathiyaseelan@sathiyaseelan:~$ kubectl get deploy
NAME READY UP-TO-DATE AVAILABLE AGE
my-deploy 1/1 1 1 6s
sathiyaseelan@sathiyaseelan:~$ kubectl get pod
NAME READY STATUS RESTARTS AGE
my-deploy-6f8d786b46-9rwst 1/1 Running 0 11s
sathiyaseelan@sathiyaseelan:~$ minikube service my-service
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | 7000 | http://192.168.49.2:30001 |
Type here to search
31°C
ENG
22:39
20-03-2025


sathiyaseelan@sathiyaseelan: ~
Starting tunnel for service my-service.
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | | http://127.0.0.1:38493 |
Opening service default/my-service in default browser...
http://127.0.0.1:38493
Because you are using a Docker driver on linux, the terminal needs to be open to run it.
^C Stopping tunnel for service my-service.
sathiyaseelan@sathiyaseelan:~$ curl http://192.168.49.2:30009
curl: (7) Failed to connect to 192.168.49.2 port 30009 after 0 ms: Connection refused
sathiyaseelan@sathiyaseelan:~$ curl http://192.168.49.2:30009
curl: (7) Failed to connect to 192.168.49.2 port 30009 after 0 ms: Connection refused
sathiyaseelan@sathiyaseelan:~$ minikube service my-service
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | 9000 | http://192.168.49.2:30009 |
Starting tunnel for service my-service.
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | | http://127.0.0.1:41599 |
Opening service default/my-service in default browser...
http://127.0.0.1:41599
Because you are using a Docker driver on linux, the terminal needs to be open to run it.
^C Stopping tunnel for service my-service.
sathiyaseelan@sathiyaseelan:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
Type here to search
31°C
ENG
22:39
20-03-2025


sathiyaseelan@sathiyaseelan: ~
sathiyaseelan@sathiyaseelan:~$ curl http://192.168.49.2:30009
curl: (7) Failed to connect to 192.168.49.2 port 30009 after 0 ms: Connection refused
sathiyaseelan@sathiyaseelan:~$ curl http://192.168.49.2:30009
curl: (7) Failed to connect to 192.168.49.2 port 30009 after 0 ms: Connection refused
sathiyaseelan@sathiyaseelan:~$ curl http://192.168.49.2:30009/maven-web-app
curl: (7) Failed to connect to 192.168.49.2 port 30009 after 0 ms: Connection refused
sathiyaseelan@sathiyaseelan:~$ ping http://192.168.49.2:30009
ping: http://192.168.49.2:30009: Name or service not known
sathiyaseelan@sathiyaseelan:~$ ping http://192.168.49.2:30009
ping: http://192.168.49.2:30009: Name or service not known
sathiyaseelan@sathiyaseelan:~$ curl http://192.168.49.2:30009
curl: (7) Failed to connect to 192.168.49.2 port 30009 after 0 ms: Connection refused
sathiyaseelan@sathiyaseelan:~$ curl http://192.168.49.2:30009
curl: (7) Failed to connect to 192.168.49.2 port 30009 after 0 ms: Connection refused
sathiyaseelan@sathiyaseelan:~$ ^C
sathiyaseelan@sathiyaseelan:~$ exit
logout
C:\Users\DELL>wsl.exe
sathiyaseelan@sathiyaseelan:/mnt/c/Users/DELL$ [sudo] password for sathiyaseelan:
Sorry, try again.
[sudo] password for sathiyaseelan:
Sorry, try again.
[sudo] password for sathiyaseelan:
^C
[1]+ Exit 1 sudo dockerd --host=unix:///var/run/docker.sock > /dev/null 2>&1
sathiyaseelan@sathiyaseelan:/mnt/c/Users/DELL$ cd
sathiyaseelan@sathiyaseelan:~$ docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/sathiyaseelan/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
sathiyaseelan@sathiyaseelan:~$ ls
deployment.yml guvi-day1 kubectl.sha256 pod.yml ubuntu-fan_0.12.16_all.deb
sathiyaseelan@sathiyaseelan:~$ kubectl get pod
NAME READY STATUS RESTARTS AGE
my-deploy-98cdff5b-75g2c 1/1 Running 0 7m43s
Type here to search
31°C
ENG
22:39
20-03-2025


sathiyaseelan@sathiyaseelan: ~
Starting tunnel for service my-service.
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | | http://127.0.0.1:41631 |
Opening service default/my-service in default browser...
http://127.0.0.1:41631
Because you are using a Docker driver on linux, the terminal needs to be open to run it.
^C Stopping tunnel for service my-service.
sathiyaseelan@sathiyaseelan:~$ curl http://192.168.49.2:30001/maven-web-app
curl: (7) Failed to connect to 192.168.49.2 port 30001 after 0 ms: Connection refused
sathiyaseelan@sathiyaseelan:~$ curl http://192.168.49.2:30001
curl: (7) Failed to connect to 192.168.49.2 port 30001 after 0 ms: Connection refused
sathiyaseelan@sathiyaseelan:~$ nano deployment.yml
sathiyaseelan@sathiyaseelan:~$ kubectl delete all --all
pod "my-deploy-6f8d786b46-9rwst" deleted
service "kubernetes" deleted
service "my-service" deleted
deployment.apps "my-deploy" deleted
replicaset.apps "my-deploy-6f8d786b46" deleted
sathiyaseelan@sathiyaseelan:~$ kubectl apply -f deployment.yml
deployment.apps/my-deploy created
service/my-service created
sathiyaseelan@sathiyaseelan:~$ minikube service my-service
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | 9000 | http://192.168.49.2:30009 |
Starting tunnel for service my-service.
| NAMESPACE | NAME | TARGET PORT | URL |
| default | my-service | | http://127.0.0.1:33299 |
Opening service default/my-service in default browser...
http://127.0.0.1:33299
Because you are using a Docker driver on linux, the terminal needs to be open to run it.
Type here to search
31°C
ENG
22:39
20-03-2025