DevOps Day 4 Task

Kubernetes,Namespace:

Kubernetes (K8s)

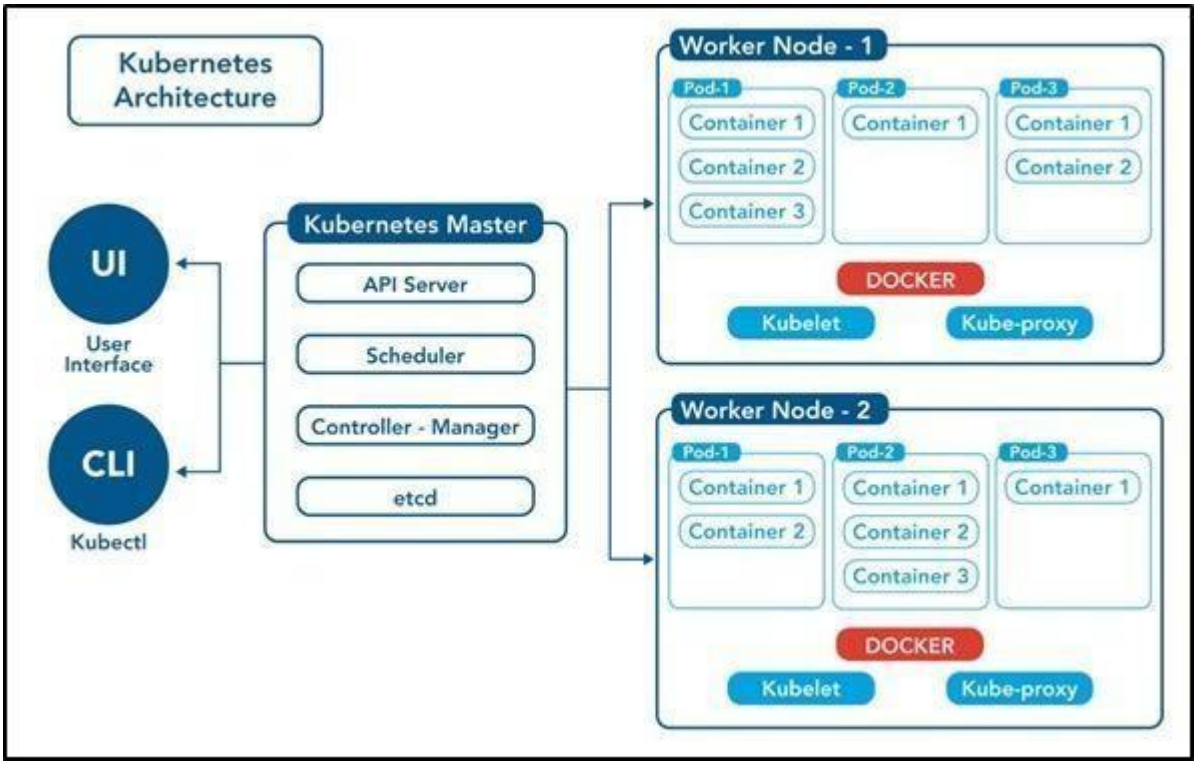
Kubernetes is an open source container orchestration engine for automating deployment, scaling, and management of containerized applications. The open source project is hosted by the Cloud Native Computing Foundation (CNCF).
It provides a scalable and resilient framework for automating the deployment, scaling, and management of applications across clusters of servers.

A SMALL HISTORY OF K8S:

- In the early 2000s, Google started developing a system called Borg to manage their internal containerized applications.
- Borg enabled Google to run applications at scale, providing features such as automatic scaling, service discovery, and fault tolerance.
- In 2014, Google open-sourced a version of Borg called Kubernetes.
- Kubernetes was donated to the Cloud Native Computing Foundation (CNCF), a neutral home for open-source cloud-native projects, in July 2015.

- Kubernetes 1.8 added significant enhancements for storage, security, and networking. Key features included the stable release of the stateful sets API, expanded support for volume plugins, and improvements in security policies.

- Check URL: https://kubernetes.io/releases/ for more release details.



Control Plane /Master Node

The control plane's components make global decisions about the cluster (for example, scheduling), as well as detecting and responding to cluster events (for example, starting up a new pod when a deployment's replicas field is unsatisfied).

Control plane components can be run on any machine in the cluster. Do not run user containers on this machine.

Node Components / Worker Nodes

Node components run on every node, maintaining running pods and providing the Kubernetes runtime environment.

1. Master Node: The master node is responsible for managing the cluster and coordinating the overall state of the system. It includes the following components:

a. API Server: The API server is the central control point for all interactions with the cluster. It exposes the Kubernetes API and handles requests from users and other components.

b. Scheduler: The scheduler is responsible for assigning workloads (pods) to individual worker nodes based on resource requirements, constraints, and other policies.

c. Controller Manager: The controller manager runs various controllers that monitor the cluster state and drive it towards the desired state. Examples include the replication controller, node controller, and service controller.

d. etcd: etcd is a distributed key-value store used by Kubernetes to store cluster state and configuration data.

1. Pod: The basic building block of Kubernetes. A pod represents a single instance of a running process within the cluster. It can encapsulate one or more containers that share the same network and storage resource

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>

kubectl exec -it my-pod

[4:34 PM, 3/20/2025] +91 90928 13114: Namespace (short name = ns):

namespace is a virtual cluster or logical partition within a cluster that provides a way to organize and isolate resources. It allows multiple teams or projects to share the same physical cluster while maintaining resource separation and access control.

[4:34 PM, 3/20/2025] +91 90928 13114: # 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

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: maven-web-app
        image: aswinprabusiva/webapp1:latest ports:
```

```
        - containerPort: 8000


---


apiVersion:
v1 kind:
Service
metadata:
  name: my-service
  labels:
    app: my-service
spec:
  type: NodePort
  ports:
    - port: 8000
      targetPort: 8080
      nodePort: 30007
```

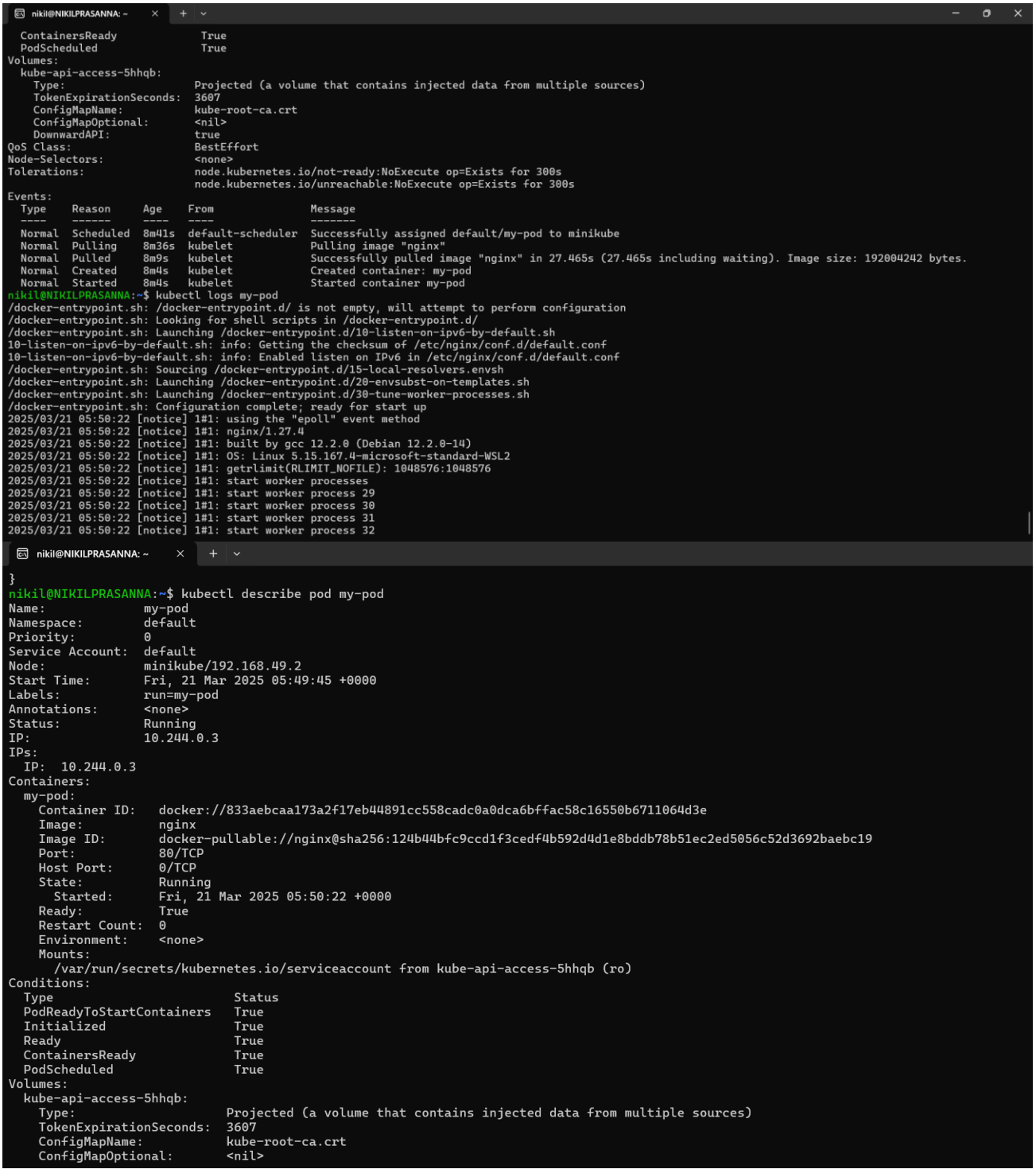
```
nikil@NIKILPRASANNA: ~
minikube   Ready    control-plane   2m38s   v1.32.0
nikil@NIKILPRASANNA:~$ kubectl get status
error: the server doesn't have a resource type "status"
nikil@NIKILPRASANNA:~$ kubectl status
error: unknown command "status" for "kubectl"
nikil@NIKILPRASANNA:~$ kubectl run my-pod --image=nginx --port=80
pod/my-pod created
nikil@NIKILPRASANNA:~$ kubectl get pods
NAME     READY   STATUS              RESTARTS   AGE
my-pod   0/1     ContainerCreating   0          13s
nikil@NIKILPRASANNA:~$ kubectl get replicaset
No resources found in default namespace.
nikil@NIKILPRASANNA:~$ kubectl apply -f replicaset.yaml
error: the path "replicaset.yaml" does not exist
nikil@NIKILPRASANNA:~$ kubectl get replicaset -A   # To check in all namespaces
NAMESPACE     NAME                 DESIRED   CURRENT   READY   AGE
kube-system   coredns-668d6bf9bc   1         1         1       30m
nikil@NIKILPRASANNA:~$ kubectl get replicaset -n <namespace-name>
bash: syntax error near unexpected token `newline'
nikil@NIKILPRASANNA:~$ kubectl get replicaset -n kube-system
NAME                 DESIRED   CURRENT   READY   AGE
coredns-668d6bf9bc   1         1         1       31m
nikil@NIKILPRASANNA:~$ kubectl get namespaces
NAME              STATUS   AGE
default           Active   32m
kube-node-lease   Active   32m
kube-public       Active   32m
kube-system       Active   32m
nikil@NIKILPRASANNA:~$ kubectl get replicaset -A
NAMESPACE     NAME                 DESIRED   CURRENT   READY   AGE
kube-system   coredns-668d6bf9bc   1         1         1       32m
nikil@NIKILPRASANNA:~$ kubectl get namespaces
NAME              STATUS   AGE
default           Active   32m
kube-node-lease   Active   32m
kube-public       Active   32m
kube-system       Active   32m
nikil@NIKILPRASANNA:~$ kubectl apply -f replicaset.yaml
error: the path "replicaset.yaml" does not exist
nikil@NIKILPRASANNA:~$ ls
nikil@NIKILPRASANNA:~$ cd /path/to/your/directory
```

```
nikil@NIKILPRASANNA: ~
kube-system       Active   32m
nikil@NIKILPRASANNA:~$ kubectl apply -f replicaset.yaml
error: the path "replicaset.yaml" does not exist
nikil@NIKILPRASANNA:~$ ls
nikil@NIKILPRASANNA:~$ cd /path/to/your/directory
bash: cd: /path/to/your/directory: No such file or directory
nikil@NIKILPRASANNA:~$ kubectl run my-pod --image=nginx --port=80
Error from server (AlreadyExists): pods "my-pod" already exists
nikil@NIKILPRASANNA:~$ kubectl get pods
NAME     READY   STATUS    RESTARTS   AGE
my-pod   1/1     Running   0          7m30s
nikil@NIKILPRASANNA:~$ kubectl get pods -A
NAMESPACE     NAME                                 READY   STATUS    RESTARTS      AGE
default       my-pod                               1/1     Running   0             7m38s
kube-system   coredns-668d6bf9bc-wxs8v             1/1     Running   0             34m
kube-system   etcd-minikube                        1/1     Running   0             34m
kube-system   kube-apiserver-minikube              1/1     Running   0             34m
kube-system   kube-controller-manager-minikube     1/1     Running   0             34m
kube-system   kube-proxy-lf4g5                     1/1     Running   0             34m
kube-system   kube-scheduler-minikube              1/1     Running   0             34m
kube-system   storage-provisioner                  1/1     Running   1 (34m ago)   34m
nikil@NIKILPRASANNA:~$ kubectl get pods -n kube-system
NAME                                 READY   STATUS    RESTARTS      AGE
coredns-668d6bf9bc-wxs8v             1/1     Running   0             34m
etcd-minikube                        1/1     Running   0             34m
kube-apiserver-minikube              1/1     Running   0             34m
kube-controller-manager-minikube     1/1     Running   0             34m
kube-proxy-lf4g5                     1/1     Running   0             34m
kube-scheduler-minikube              1/1     Running   0             34m
storage-provisioner                  1/1     Running   1 (34m ago)   34m
nikil@NIKILPRASANNA:~$ kubectl get pods my-pod
NAME     READY   STATUS    RESTARTS   AGE
my-pod   1/1     Running   0          8m4s
nikil@NIKILPRASANNA:~$ kubectl get pods my-pod -o yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: "2025-03-21T05:49:44Z"
  labels:
    run: my-pod
  name: my-pod
```

```
nikil@NIKILPRASANNA: ~
nikil@NIKILPRASANNA:~$ kubectl get pods my-pod
NAME     READY   STATUS    RESTARTS   AGE
my-pod   1/1     Running   0          8m4s
nikil@NIKILPRASANNA:~$ kubectl get pods my-pod -o yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: "2025-03-21T05:49:44Z"
  labels:
    run: my-pod
  name: my-pod
  namespace: default
  resourceVersion: "1685"
  uid: e97c60d8-f987-4288-97a4-d29e4e29ed63
spec:
  containers:
  - image: nginx
    imagePullPolicy: Always
    name: my-pod
    ports:
    - containerPort: 80
      protocol: TCP
    resources: {}
    terminationMessagePath: /dev/termination-log
    terminationMessagePolicy: File
    volumeMounts:
    - mountPath: /var/run/secrets/kubernetes.io/serviceaccount
      name: kube-api-access-5hhqb
      readOnly: true
  dnsPolicy: ClusterFirst
  enableServiceLinks: true
  nodeName: minikube
  preemptionPolicy: PreemptLowerPriority
  priority: 0
  restartPolicy: Always
  schedulerName: default-scheduler
  securityContext: {}
  serviceAccount: default
  serviceAccountName: default
  terminationGracePeriodSeconds: 30
  tolerations:
    name: my-pod
    ready: true
    restartCount: 0
    started: true
    state:
      running:
        startedAt: "2025-03-21T05:50:22Z"
    volumeMounts:
    - mountPath: /var/run/secrets/kubernetes.io/serviceaccount
      name: kube-api-access-5hhqb
      readOnly: true
      recursiveReadOnly: Disabled
  hostIP: 192.168.49.2
  hostIPs:
  - ip: 192.168.49.2
  phase: Running
  podIP: 10.244.0.3
  podIPs:
  - ip: 10.244.0.3
  qosClass: BestEffort
  startTime: "2025-03-21T05:49:45Z"
nikil@NIKILPRASANNA:~$ kubectl get pods my-pod -o json
{
    "apiVersion": "v1",
    "kind": "Pod",
    "metadata": {
        "creationTimestamp": "2025-03-21T05:49:44Z",
        "labels": {
            "run": "my-pod"
        },
        "name": "my-pod",
        "namespace": "default",
        "resourceVersion": "1685",
        "uid": "e97c60d8-f987-4288-97a4-d29e4e29ed63"
    },
    "spec": {
        "containers": [
            {
                "image": "nginx",
                "imagePullPolicy": "Always",
                "name": "my-pod",
```

```
nikil@NIKILPRASANNA: ~
  startTime: "2025-03-21T05:49:45Z"
nikil@NIKILPRASANNA:~$ kubectl get pods my-pod -o json
{
    "apiVersion": "v1",
    "kind": "Pod",
    "metadata": {
        "creationTimestamp": "2025-03-21T05:49:44Z",
        "labels": {
            "run": "my-pod"
        },
        "name": "my-pod",
        "namespace": "default",
        "resourceVersion": "1685",
        "uid": "e97c60d8-f987-4288-97a4-d29e4e29ed63"
    },
    "spec": {
        "containers": [
            {
                "image": "nginx",
                "imagePullPolicy": "Always",
                "name": "my-pod",
                "ports": [
                    {
                        "containerPort": 80,
                        "protocol": "TCP"
                    }
                ],
                "resources": {},
                "terminationMessagePath": "/dev/termination-log",
                "terminationMessagePolicy": "File",
                "volumeMounts": [
                    {
                        "mountPath": "/var/run/secrets/kubernetes.io/serviceaccount",
                        "name": "kube-api-access-5hhqb",
                        "readOnly": true
                    }
                ]
            }
        ],
        "dnsPolicy": "ClusterFirst",
        "enableServiceLinks": true,
nikil@NIKILPRASANNA: ~
            {
                "effect": "NoExecute",
                "key": "node.kubernetes.io/not-ready",
                "operator": "Exists",
                "tolerationSeconds": 300
            },
            {
                "effect": "NoExecute",
                "key": "node.kubernetes.io/unreachable",
                "operator": "Exists",
                "tolerationSeconds": 300
            }
        ],
        "volumes": [
            {
                "name": "kube-api-access-5hhqb",
                "projected": {
                    "defaultMode": 420,
                    "sources": [
                        {
                            "serviceAccountToken": {
                                "expirationSeconds": 3607,
                                "path": "token"
                            }
                        },
                        {
                            "configMap": {
                                "items": [
                                    {
                                        "key": "ca.crt",
                                        "path": "ca.crt"
                                    }
                                ],
                                "name": "kube-root-ca.crt"
                            }
                        },
                        {
                            "downwardAPI": {
                                "items": [
                                    {
                                        "fieldRef": {
```


nikil@NIKILPRASANNA: ~
},
"status": {
"conditions": [
{
"lastProbeTime": null,
"lastTransitionTime": "2025-03-21T05:50:22Z",
"status": "True",
"type": "PodReadyToStartContainers"
},
{
"lastProbeTime": null,
"lastTransitionTime": "2025-03-21T05:49:45Z",
"status": "True",
"type": "Initialized"
},
{
"lastProbeTime": null,
"lastTransitionTime": "2025-03-21T05:50:22Z",
"status": "True",
"type": "Ready"
},
{
"lastProbeTime": null,
"lastTransitionTime": "2025-03-21T05:50:22Z",
"status": "True",
"type": "ContainersReady"
},
{
"lastProbeTime": null,
"lastTransitionTime": "2025-03-21T05:49:45Z",
"status": "True",
"type": "PodScheduled"
}
],
"containerStatuses": [
{
"containerID": "docker://833aebcaa173a2f17eb44891cc558cadc0a0dca6bffac58c16550b6711064d3e",
"image": "nginx:latest",
"imageID": "docker-pullable://nginx@sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056c52d3692baebc19",
"lastState": {},
"name": "my-pod",
nikil@NIKILPRASANNA: ~
},
"volumeMounts": [
{
"mountPath": "/var/run/secrets/kubernetes.io/serviceaccount",
"name": "kube-api-access-5hhqb",
"readOnly": true,
"recursiveReadOnly": "Disabled"
}
]
}
],
"hostIP": "192.168.49.2",
"hostIPs": [
{
"ip": "192.168.49.2"
}
],
"phase": "Running",
"podIP": "10.244.0.3",
"podIPs": [
{
"ip": "10.244.0.3"
}
],
"qosClass": "BestEffort",
"startTime": "2025-03-21T05:49:45Z"
}
}
nikil@NIKILPRASANNA:~$ kubectl describe pod my-pod
Name: my-pod
Namespace: default
Priority: 0
Service Account: default
Node: minikube/192.168.49.2
Start Time: Fri, 21 Mar 2025 05:49:45 +0000
Labels: run=my-pod
Annotations: <none>
Status: Running
IP: 10.244.0.3
IPs:
IP: 10.244.0.3


nikil@NIKILPRASANNA: ~
ContainersReady True
PodScheduled True
Volumes:
kube-api-access-5hhqb:
Type: Projected (a volume that contains injected data from multiple sources)
TokenExpirationSeconds: 3607
ConfigMapName: kube-root-ca.crt
ConfigMapOptional: <nil>
DownwardAPI: true
QoS Class: BestEffort
Node-Selectors: <none>
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 8m41s default-scheduler Successfully assigned default/my-pod to minikube
Normal Pulling 8m36s kubelet Pulling image "nginx"
Normal Pulled 8m9s kubelet Successfully pulled image "nginx" in 27.465s (27.465s including waiting). Image size: 192004242 bytes.
Normal Created 8m4s kubelet Created container: my-pod
Normal Started 8m4s kubelet Started container my-pod
nikil@NIKILPRASANNA:~$ kubectl logs my-pod
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/03/21 05:50:22 [notice] 1#1: using the "epoll" event method
2025/03/21 05:50:22 [notice] 1#1: nginx/1.27.4
2025/03/21 05:50:22 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2025/03/21 05:50:22 [notice] 1#1: OS: Linux 5.15.167.4-microsoft-standard-WSL2
2025/03/21 05:50:22 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/03/21 05:50:22 [notice] 1#1: start worker processes
2025/03/21 05:50:22 [notice] 1#1: start worker process 29
2025/03/21 05:50:22 [notice] 1#1: start worker process 30
2025/03/21 05:50:22 [notice] 1#1: start worker process 31
2025/03/21 05:50:22 [notice] 1#1: start worker process 32
nikil@NIKILPRASANNA: ~
}
nikil@NIKILPRASANNA:~$ kubectl describe pod my-pod
Name: my-pod
Namespace: default
Priority: 0
Service Account: default
Node: minikube/192.168.49.2
Start Time: Fri, 21 Mar 2025 05:49:45 +0000
Labels: run=my-pod
Annotations: <none>
Status: Running
IP: 10.244.0.3
IPs:
IP: 10.244.0.3
Containers:
my-pod:
Container ID: docker://833aebcaa173a2f17eb44891cc558cadc0a0dca6bffac58c16550b6711064d3e
Image: nginx
Image ID: docker-pullable://nginx@sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056c52d3692baebc19
Port: 80/TCP
Host Port: 0/TCP
State: Running
Started: Fri, 21 Mar 2025 05:50:22 +0000
Ready: True
Restart Count: 0
Environment: <none>
Mounts:
/var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-5hhqb (ro)
Conditions:
Type Status
PodReadyToStartContainers True
Initialized True
Ready True
ContainersReady True
PodScheduled True
Volumes:
kube-api-access-5hhqb:
Type: Projected (a volume that contains injected data from multiple sources)
TokenExpirationSeconds: 3607
ConfigMapName: kube-root-ca.crt
ConfigMapOptional: <nil>