

Devops class guvi (DAY-4)

20 March 2025 (DAY-4)

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
```

```

Terminal Shell Edit View Window Help
raswanthsabarish -- bash -- 179x49
> gcr.io/k8s-minikube/kicbase...: 398.08 MiB / 452.84 MiB 87.91% 1.20 MiB
> index.docker.io/kicbase/sta...: 0 B [====] 7% ? p/s ?
! minikube was unable to download gcr.io/k8s-minikube/kicbase:v0.0.46, but successfully downloaded docker.io/kicbase/stable:v0.0.46@sha256:fd2d445ddcc33ebc5c6b68a17e6219ea287ce6
3c905095ea1525296da2d1a279 as a fallback image
Creating docker container (CPUs=2, Memory=2200MB) ...
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
■ Generating certificates and keys ...
■ Booting up control plane ...
■ Configuring RBAC rules ...
Configuring bridge CNI (Container Networking Interface) ...
Verifying Kubernetes components...
■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: storage-provisioner, default-storageclass
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
Raswanths-MacBook-Air:~ raswanthsabarish$ 
Raswanths-MacBook-Air:~ raswanthsabarish$ 
Raswanths-MacBook-Air:~ raswanthsabarish$ 
Raswanths-MacBook-Air:~ raswanthsabarish$ kubectl delete all --all
service "kubernetes" deleted
Raswanths-MacBook-Air:~ raswanthsabarish$ kubectl run ipodname --image=nginx --port=8085
pod/ipodname created
Raswanths-MacBook-Air:~ raswanthsabarish$ kubectl get pods
NAME READY STATUS RESTARTS AGE
ipodname 1/1 Running 0 33s
Raswanths-MacBook-Air:~ raswanthsabarish$ kubectl get pods -A
NAMESPACE NAME READY STATUS RESTARTS AGE
default ipodname 1/1 Running 0 41s
kube-system coredns-648d6bf9bc-fx9rc 1/1 Running 0 2m10s
kube-system etcd-minikube 1/1 Running 0 2m15s
kube-system kube-apiserver-minikube 1/1 Running 0 2m15s
kube-system kube-controller-manager-minikube 1/1 Running 0 2m15s
kube-system kube-proxy-qpxmym 1/1 Running 0 2m16s
kube-system kube-scheduler-minikube 1/1 Running 0 2m16s
kube-system storage-provisioner 1/1 Running 1 (100s ago) 2m14s
Raswanths-MacBook-Air:~ raswanthsabarish$ kubectl get pods -n kube-system
NAME READY STATUS RESTARTS AGE
coredns-648d6bf9bc-fx9rc 1/1 Running 0 2m24s
etcd-minikube 1/1 Running 0 2m29s
kube-apiserver-minikube 1/1 Running 0 2m29s
kube-controller-manager-minikube 1/1 Running 0 2m29s
kube-proxy-qpxmym 1/1 Running 0 2m24s
kube-scheduler-minikube 1/1 Running 0 2m30s
storage-provisioner 1/1 Running 1 (114s ago) 2m28s
Raswanths-MacBook-Air:~ raswanthsabarish$ kubectl get pods ipodname
NAME READY STATUS RESTARTS AGE
ipodname 1/1 Running 0 70s
Raswanths-MacBook-Air:~ raswanthsabarish$ 

```

```

Terminal Shell Edit View Window Help
raswanthsabarish -- bash -- 179x49
2025/03/20 05:35:34 [notice] 1#1: start worker process 35
2025/03/20 05:35:34 [notice] 1#1: start worker process 36
Raswanths-MacBook-Air:~ raswanthsabarish$ kubectl exec ipodname --ls
error: unknown flag: --ls
See 'kubectl exec --help' for usage.
Raswanths-MacBook-Air:~ raswanthsabarish$ kubectl exec ipodname -- ls
bin
boot
dev
docker-entrypoint.d
docker-entrypoint.sh
etc
home
lib
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
Raswanths-MacBook-Air:~ raswanthsabarish$ kubectl exec ipodname -- pwd
/
Raswanths-MacBook-Air:~ raswanthsabarish$ kubectl get pods
NAME READY STATUS RESTARTS AGE
ipodname 1/1 Running 0 10m
Raswanths-MacBook-Air:~ raswanthsabarish$ sudo nano pod.yaml
Password:
Raswanths-MacBook-Air:~ raswanthsabarish$ sudo nano pod.yaml
Password:
Raswanths-MacBook-Air:~ raswanthsabarish$ kubectl apply -f pod.yaml
Error from server (BadRequest): error when creating "pod.yaml": Pod in version "v1" cannot be handled as a Pod: strict decoding error: unknown field "spec.ports"
Raswanths-MacBook-Air:~ raswanthsabarish$ sudo nano pod.yaml
Raswanths-MacBook-Air:~ raswanthsabarish$ kubectl apply -f pod.yaml
error: error parsing pod.yaml: error converting YAML to JSON: yaml: line 10: found character that cannot start any token
Raswanths-MacBook-Air:~ raswanthsabarish$ sudo nano pod.yaml
Raswanths-MacBook-Air:~ raswanthsabarish$ sudo nano pod.yaml
Raswanths-MacBook-Air:~ raswanthsabarish$ kubectl apply -f pod.yaml
pod/my-app created
Raswanths-MacBook-Air:~ raswanthsabarish$ kubectl get pods
NAME READY STATUS RESTARTS AGE
ipodname 1/1 Running 0 26m
my-app 0/1 ContainerCreating 0 13s
Raswanths-MacBook-Air:~ raswanthsabarish$ 

```

Kubernetes (K8s) Notes

Kubernetes is an open-source container orchestration platform for automating deployment, scaling, and management of containerized applications.

Originally developed by **Google** as **Borg** in the early 2000s and open-sourced in 2014 under CNCF.

Architecture: Kubernetes follows a **master-worker** model.

Control Plane (Master Node)

API Server – Entry point for all Kubernetes interactions.

• **Scheduler** – Assigns workloads to nodes based on resource availability.

Controller Manager – Ensures the cluster's desired state (e.g., scaling, replication).

etcd – Stores all cluster data persistently.

Worker Nodes (Slave Nodes)

Kubelet – Manages pods and communicates with the master node.

Kube Proxy – Handles networking and load balancing.

Container Runtime – Runs containers (e.g., Docker, containerd).

Key Kubernetes Concepts

Pods – Smallest deployable unit in Kubernetes, containing one or more containers.

• **Deployments** – Manage and scale pod replicas.

Services – Expose applications within and outside the cluster.

• **Ingress** – Manages external access to services using HTTP/HTTPS.

A screenshot of a Mac OS X desktop environment. At the top, a dock bar contains various application icons. Below it, a window titled "raswanthsbarish -- bash -- 179x49" displays the YAML configuration for a Kubernetes pod named "ipodname". The pod is running on node "minikube" and has a single container using the "nginx" image. It is mounted with a secret from the "kubernetes.io/serviceaccount" type.

```
ipodname 1/1 Running 0 118s 10.244.0.3 minikube <none> <none>
Raswanths-MacBook-Air:~ raswanthsbarish$ kubectl get pods ipodname -o yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: "2025-03-20T05:35:19Z"
  labels:
    run: ipodname
  name: ipodname
  namespace: default
  resourceVersion: "494"
  uid: 19adf8fc-aae6-4394-bb8c-1b25d5e3d2ea
spec:
  containers:
    - image: nginx
      imagePullPolicy: Always
      name: ipodname
      ports:
        - containerPort: 8085
          protocol: TCP
      resources: {}
      terminationMessagePath: /dev/termination-log
      terminationMessagePolicy: File
    volumes:
      - mountPath: /var/run/secrets/kubernetes.io/serviceaccount
        name: kube-api-access-sb4j8
        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:
    - 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-sb4j8
```

A screenshot of a Mac OS X desktop environment. At the top, a dock bar contains various application icons. Below it, a window titled "raswanthsbarish -- bash -- 179x49" displays the JSON representation of the same Kubernetes pod configuration. The JSON structure mirrors the YAML output, detailing the pod's metadata, spec, and status.

```
startTime: "2025-03-20T05:35:19Z"
Raswanths-MacBook-Air:~ raswanthsbarish$ kubectl get pods ipodname -o json
{
  "apiVersion": "v1",
  "kind": "Pod",
  "metadata": {
    "creationTimestamp": "2025-03-20T05:35:19Z",
    "labels": {
      "run": "ipodname"
    },
    "name": "ipodname",
    "namespace": "default",
    "resourceVersion": "494",
    "uid": "19adf8fc-aae6-4394-bb8c-1b25d5e3d2ea"
  },
  "spec": {
    "containers": [
      {
        "image": "nginx",
        "imagePullPolicy": "Always",
        "name": "ipodname",
        "ports": [
          {
            "containerPort": 8085,
            "protocol": "TCP"
          }
        ],
        "resources": {},
        "terminationMessagePath": "/dev/termination-log",
        "terminationMessagePolicy": "File",
        "volumeMounts": [
          {
            "mountPath": "/var/run/secrets/kubernetes.io/serviceaccount",
            "name": "kube-api-access-sb4j8",
            "readOnly": true
          }
        ]
      }
    ],
    "dnsPolicy": "ClusterFirst",
    "enableServiceLinks": true,
    "nodeName": "minikube",
    "preemptionPolicy": "PreemptLowerPriority",
    "priority": 0,
    "restartPolicy": "Always",
    "schedulerName": "default-scheduler",
    "securityContext": {},
    "serviceAccount": "default",
    "serviceAccountName": "default",
    "tolerations": []
  }
}
```

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
```