

# Kubernetes Lab Exercise

Mohit Namdev

## Exercise 1: Setting Up Your Kubernetes Cluster

Objective: Set up a local Kubernetes environment using Minikube and kubectl.

minikube start

```
PS M:\Mohit\Kubernetes Assignment> minikube start
minikube v1.36.0 on Microsoft Windows 11 Home Single Language 10.0.22631.5771 Build 22631.5771
🔧 Automatically selected the docker driver
🔧 Using Docker Desktop driver with root privileges
🔥 Starting "minikube" primary control-plane node in "minikube" cluster
📦 Pulling base image v0.0.47 ...
📥 Downloading Kubernetes v1.33.1 preload ...
> gcr.io/k8s-minikube/kicbase...: 502.26 MiB / 502.26 MiB 100.00% 3.26 Mi
> preloaded-images-k8s-v18-v1...: 347.04 MiB / 347.04 MiB 100.00% 1.35 Mi
🔥 Creating docker container (CPUs=2, Memory=2200MB) ...
❗ Failing to connect to https://registry.k8s.io/ from inside the minikube container
💡 To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
🔧 Preparing Kubernetes v1.33.1 on Docker 28.1.1 ...
  ▪ 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
```

kubectl cluster-info

```
PS M:\Mohit\Kubernetes Assignment> kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:55830
CoreDNS is running at https://127.0.0.1:55830/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

kubectl get nodes

```
PS M:\Mohit\Kubernetes Assignment> kubectl get nodes
NAME          STATUS    ROLES          AGE   VERSION
minikube      Ready     control-plane  38m   v1.33.1
```

## Exercise 2: Creating and Managing Pods

Objective: Learn how to create and manage Pods.

kubectl run nginx-pod --image=nginx --restart=Never

```
PS M:\Mohit\Kubernetes Assignment> kubectl run nginx-pod --image=nginx --restart=Never
pod/nginx-pod created
PS M:\Mohit\Kubernetes Assignment>
```

kubectl get pods

```
PS M:\Mohit\Kubernetes Assignment> kubectl get pods
NAME          READY   STATUS              RESTARTS   AGE
nginx-pod     0/1     ContainerCreating   0           40s
```

kubectl logs nginx-pod

```
PS M:\Mohit\Kubernetes Assignment> kubectl logs nginx-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
2025/08/25 17:14:41 [notice] 1#1: using the "epoll" event method
2025/08/25 17:14:41 [notice] 1#1: nginx/1.29.1
2025/08/25 17:14:41 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14+deb12u1)
2025/08/25 17:14:41 [notice] 1#1: OS: Linux 6.6.87.2-microsoft-standard-WSL2
2025/08/25 17:14:41 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/08/25 17:14:41 [notice] 1#1: start worker processes
2025/08/25 17:14:41 [notice] 1#1: start worker process 30
2025/08/25 17:14:41 [notice] 1#1: start worker process 31
2025/08/25 17:14:41 [notice] 1#1: start worker process 32
2025/08/25 17:14:41 [notice] 1#1: start worker process 33
2025/08/25 17:14:41 [notice] 1#1: start worker process 34
2025/08/25 17:14:41 [notice] 1#1: start worker process 35
2025/08/25 17:14:41 [notice] 1#1: start worker process 36
2025/08/25 17:14:41 [notice] 1#1: start worker process 37
2025/08/25 17:14:41 [notice] 1#1: start worker process 38
2025/08/25 17:14:41 [notice] 1#1: start worker process 39
2025/08/25 17:14:41 [notice] 1#1: start worker process 40
2025/08/25 17:14:41 [notice] 1#1: start worker process 41
```

kubectl expose pod nginx-pod --type=NodePort --port=80

```
PS M:\Mohit\Kubernetes Assignment> kubectl expose pod nginx-pod --type=NodePort --port=80
service/nginx-pod exposed
```

kubectl delete pod nginx-pod

```
PS M:\Mohit\Kubernetes Assignment> kubectl delete pod nginx-pod
pod "nginx-pod" deleted
```

Checkpoint: What happens when you delete a pod?

After a pod is deleted, Kubernetes terminates it, and then it can't be accessed. Since the pod was created directly, it is not automatically recreated. The cluster continues to function, and other resources may remain unless explicitly deleted.

### Exercise 3: Working with Deployments

Objective: Use Deployments for managing replicated Pods.

```
PS M:\Mohit\Kubernetes Assignment> kubectl create deployment nginx-deployment --image=nginx deployment.apps/nginx-deployment created
```

kubectl scale deployment nginx-deployment --replicas=3

```
PS M:\Mohit\Kubernetes Assignment> kubectl scale deployment nginx-deployment --replicas=3 deployment.apps/nginx-deployment scaled
```

kubectl get deployments

```
PS M:\Mohit\Kubernetes Assignment> kubectl get deployments
NAME                READY    UP-TO-DATE    AVAILABLE    AGE
nginx-deployment    3/3      3             3            2m30s
```

kubectl set image deployment/nginx-deployment nginx=nginx:1.25

```
PS M:\Mohit\Kubernetes Assignment> kubectl set image deployment/nginx-deployment nginx=nginx:1.25 deployment.apps/nginx-deployment image updated
```

kubectl rollout history deployment/nginx-deployment

```
PS M:\Mohit\Kubernetes Assignment> kubectl rollout history deployment/nginx-deployment deployment.apps/nginx-deployment
REVISION  CHANGE-CAUSE
1          <none>
2          <none>
```

kubectl rollout undo deployment/nginx-deployment

```
PS M:\Mohit\Kubernetes Assignment> kubectl rollout undo deployment/nginx-deployment deployment.apps/nginx-deployment rolled back
```

Checkpoint: What does deployment rollout history show? How would you roll back a deployment?

It shows the record of revisions for a deployment, changes like image updates. To roll back a deployment, we can use `kubectl rollout undo deployment/` to revert to the previous revision, or specify a revision with `-to-revision=`

## Exercise 4: Services and Networking

Objective: Expose your app using Kubernetes services

kubectl expose deployment nginx-deployment --type=NodePort --port=80

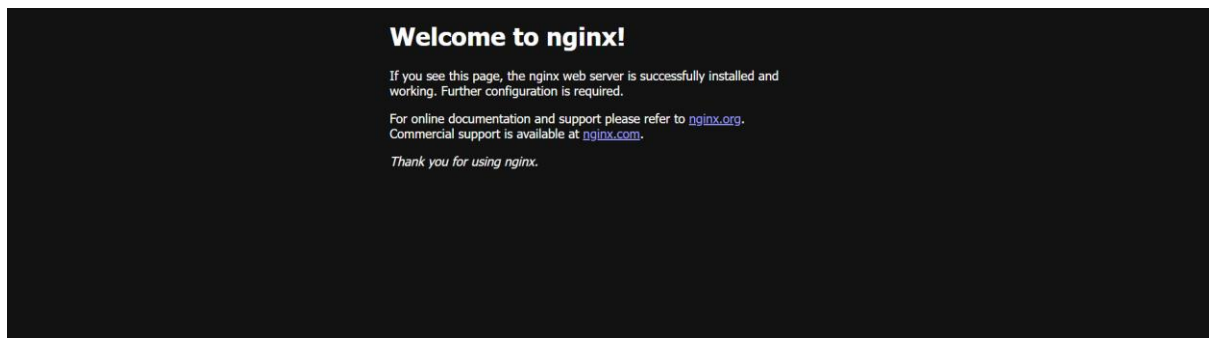
```
PS M:\Mohit\Kubernetes Assignment> kubectl expose deployment nginx-deployment --type=NodePort --port=80
service/nginx-deployment exposed
```

kubectl get svc

```
PS M:\Mohit\Kubernetes Assignment> kubectl get svc
NAME                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes           ClusterIP     10.96.0.1     <none>         443/TCP          10h
nginx-deployment     NodePort      10.97.217.29  <none>         80:31922/TCP     84s
nginx-pod            NodePort      10.102.96.2   <none>         80:32507/TCP     9h
```

minikube service nginx-deployment --url

```
PS M:\Mohit\Kubernetes Assignment> minikube service nginx-deployment --url
http://127.0.0.1:64980
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```



Checkpoint: Difference between ClusterIP, NodePort, and LoadBalancer services?

ClusterIP exposes a service internally within the cluster, it is ideal for internal communication.

NodePort exposes the service on a specific port of each node, which makes it suitable for external access in development.

LoadBalancer assigns an external IP via a cloud provider, it is best for production-grade external access.

## Exercise 5: ConfigMaps and Secrets

Objective: Manage configurations using ConfigMaps and Secrets.

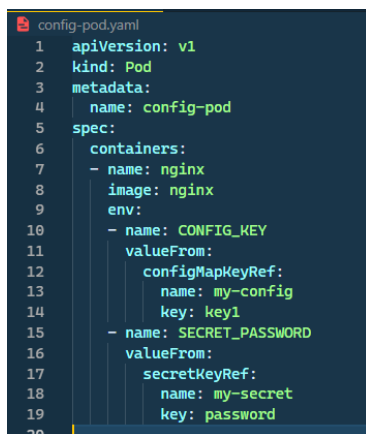
kubectl create configmap my-config --from-literal=key1=value1

```
PS M:\Mohit\Kubernetes Assignment> kubectl create configmap my-config --from-literal=key1=value1
configmap/my-config created
```

kubectl create secret generic my-secret --from-literal=password=mohit123

```
PS M:\Mohit\Kubernetes Assignment> kubectl create secret generic my-secret --from-literal=password=mohit123
secret/my-secret created
```

kubectl apply -f config-pod.yaml



```
config-pod.yaml
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: config-pod
5  spec:
6    containers:
7    - name: nginx
8      image: nginx
9      env:
10     - name: CONFIG_KEY
11       valueFrom:
12         configMapKeyRef:
13           name: my-config
14           key: key1
15     - name: SECRET_PASSWORD
16       valueFrom:
17         secretKeyRef:
18           name: my-secret
19           key: password
20
```

```
PS M:\Mohit\Kubernetes Assignment> kubectl apply -f config-pod.yaml
pod/config-pod created
```

Checkpoint: Accessing ConfigMap or Secret in an application

ConfigMaps and Secrets can be accessed as environment variables (shown in above yaml file) or mounted as volumes in a pod. Applications retrieve these values using standard environment variable access or by reading files in the mounted volume path.

## Exercise 6: Persistent Volumes (PVs) and Persistent Volume Claims (PVCs)

Objective: Use PVs and PVCs for persistent data storage.

```
pv-pvc.yaml
1  apiVersion: v1
2  kind: PersistentVolume
3  metadata:
4    name: my-pv
5  spec:
6    capacity:
7      storage: 1Gi
8    accessModes:
9      - ReadWriteOnce
10   hostPath:
11     path: "/mnt/data"
12   ---
13   apiVersion: v1
14   kind: PersistentVolumeClaim
15   metadata:
16     name: my-pvc
17   spec:
18     accessModes:
19       - ReadWriteOnce
20     resources:
21       requests:
22         storage: 1Gi
23   ---
24   apiVersion: v1
25   kind: Pod
26   metadata:
27     name: pv-pod
28   spec:
29     containers:
30       - name: nginx
31         image: nginx
32         volumeMounts:
33           - mountPath: "/data"
34             name: storage
35     volumes:
36       - name: storage
37         persistentVolumeClaim:
38           claimName: my-pvc
39
```

kubectl apply -f pv-pvc.yaml

```
PS M:\Mohit\Kubernetes Assignment> kubectl apply -f pv-pvc.yaml
persistentvolume/my-pv created
persistentvolumeclaim/my-pvc created
pod/pv-pod created
```

kubectl exec pv-pod -- sh -c "echo 'test data' > /data/test.txt"

```
PS M:\Mohit\Kubernetes Assignment> kubectl exec pv-pod -- sh -c "echo 'test data' > /data/test.txt"
PS M:\Mohit\Kubernetes Assignment>
```

kubectl delete pod pv-pod

```
PS M:\Mohit\Kubernetes Assignment> kubectl delete pod pv-pod
pod "pv-pod" deleted
```

kubectl apply -f pv-pvc.yaml

```
PS M:\Mohit\Kubernetes Assignment> kubectl apply -f pv-pvc.yaml
persistentvolume/my-pv unchanged
persistentvolumeclaim/my-pvc unchanged
pod/pv-pod created
```

kubectl exec pv-pod -- cat /data/test.txt

```
PS M:\Mohit\Kubernetes Assignment> kubectl exec pv-pod -- cat /data/test.txt
'test data'
```

Checkpoint: What happens if the PVC is deleted?

When a PVC is deleted, the associated pod loses its binding to the Persistent Volume, and any new pod cannot access the volume, we need to create a new PVC. The underlying Persistent Volume is not deleted unless its reclaim policy is set to "Delete" from the "Released" state.

## Exercise 7: StatefulSets

Objective: Use StatefulSets for managing stateful applications.

```
mysql-statefulset.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: mysql-service
5    labels:
6      app: mysql
7  spec:
8    ports:
9      - port: 3306
10      name: mysql
11      clusterIP: None
12    selector:
13      app: mysql
14  ---
15  apiVersion: apps/v1
16  kind: StatefulSet
17  metadata:
18    name: mysql
19  spec:
20    serviceName: mysql-service
21    replicas: 2
22    selector:
23      matchLabels:
24        app: mysql
25    template:
26      metadata:
27        labels:
28          app: mysql
29      spec:
30        containers:
31          - name: mysql
32            image: mysql:8
33            env:
34              - name: MYSQL_ROOT_PASSWORD
35                value: "rootpassword"
36            ports:
37              - containerPort: 3306
38
```

kubectl apply -f mysql-statefulset.yaml

```
PS M:\Mohit\Kubernetes Assignment> kubectl apply -f mysql-statefulset.yaml
service/mysql-service created
statefulset.apps/mysql created
```

kubectl get statefulset

```
PS M:\Mohit\Kubernetes Assignment> kubectl get statefulset
NAME      READY   AGE
mysql     2/2     4m28s
```

kubectl get svc mysql-service

```
PS M:\Mohit\Kubernetes Assignment> kubectl get svc mysql-service
NAME            TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
mysql-service   ClusterIP   None         <none>        3306/TCP   5m7s
```

Checkpoint: Differences between StatefulSets and Deployments

StatefulSets provide stable network identities and persistent storage for pods, which helps in ordered deployment and scaling, it is ideal for stateful applications.

Deployments manage stateless applications with identical pods, suitable for web servers. We can use StatefulSets when pod identity and data persistence are critical.



## Exercise 8: Horizontal Pod Autoscaling (HPA)

Objective: Scale your application automatically based on metrics.

```
hpa.yaml
1  apiVersion: autoscaling/v2
2  kind: HorizontalPodAutoscaler
3  metadata:
4    name: nginx-hpa
5  spec:
6    scaleTargetRef:
7      apiVersion: apps/v1
8      kind: Deployment
9      name: nginx-deployment
10   minReplicas: 1
11   maxReplicas: 5
12   metrics:
13     - type: Resource
14       resource:
15         name: cpu
16         target:
17           type: Utilization
18           averageUtilization: 70
19
```

minikube addons enable metrics-server

```
PS M:\Mohit\Kubernetes Assignment> minikube addons enable metrics-server
⚠ metrics-server is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
  ▪ Using image registry.k8s.io/metrics-server/metrics-server:v0.7.2
🌟 The 'metrics-server' addon is enabled
PS M:\Mohit\Kubernetes Assignment>
```

kubectl apply -f hpa.yaml

```
PS M:\Mohit\Kubernetes Assignment> kubectl apply -f hpa.yaml
horizontalpodautoscaler.autoscaling/nginx-hpa created
```

kubectl get hpa

```
PS M:\Mohit\Kubernetes Assignment> kubectl get hpa
NAME          REFERENCE                TARGETS          MINPODS  MAXPODS  REPLICAS  AGE
nginx-hpa     Deployment/nginx-deployment  cpu: <unknown>/70%    1         5         3         4m54s
```

Checkpoint: How does the HPA decide when to scale?

HPA decides to scales on the basis of metrics like CPU or memory utilization, defined in the HPA configuration. It monitors metrics via the Metrics Server and adjusts the number of replicas between minReplicas and maxReplicas to meet the requirements.

## Exercise 9: Helm Basics

Objective: Use Helm to manage Kubernetes applications.

winget install Helm.Helm

```
PS M:\Mohit\Kubernetes Assignment> winget install Helm.Helm
The 'msstore' source requires that you view the following agreements before using.
Terms of Transaction: https://aka.ms/microsoft-store-terms-of-transaction
The source requires the current machine's 2-letter geographic region to be sent to the backend service to function properly (ex. "US").

Do you agree to all the source agreements terms?
[Y] Yes [N] No: y
Found Helm [Helm.Helm] Version 3.18.6
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://get.helm.sh/helm-v3.18.6-windows-amd64.zip
17.5 MB / 17.5 MB
Successfully verified installer hash
Extracting archive...
Successfully extracted archive
Starting package install...
Path environment variable modified; restart your shell to use the new value.
Command line alias added: "helm"
Successfully installed
PS M:\Mohit\Kubernetes Assignment> |
```

helm repo add bitnami <https://charts.bitnami.com/bitnami>

```
PS M:\Mohit\Kubernetes Assignment> helm repo add bitnami https://charts.bitnami.com/bitnami
"bitnami" has been added to your repositories
```

helm install my-nginx bitnami/nginx

```
PS M:\Mohit\Kubernetes Assignment> helm install my-nginx bitnami/nginx
NAME: my-nginx
LAST DEPLOYED: Tue Aug 26 12:44:51 2025
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: nginx
CHART VERSION: 21.1.23
APP VERSION: 1.29.1

⚠WARNING: Since August 28th, 2025, only a limited subset of images/charts are available for free.
  Subscribe to Bitnami Secure Images to receive continued support and security updates.
  More info at https://bitnami.com and https://github.com/bitnami/containers/issues/83267

** Please be patient while the chart is being deployed **
NGINX can be accessed through the following DNS name from within your cluster:

    my-nginx.default.svc.cluster.local (port 80)

To access NGINX from outside the cluster, follow the steps below:

1. Get the NGINX URL by running these commands:

    NOTE: It may take a few minutes for the LoadBalancer IP to be available.
    Watch the status with: 'kubectl get svc --namespace default -w my-nginx'

    export SERVICE_PORT=$(kubectl get --namespace default -o jsonpath="{.spec.ports[0].port}" services my-nginx)
    export SERVICE_IP=$(kubectl get svc --namespace default my-nginx -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
    echo "http://${SERVICE_IP}:${SERVICE_PORT}"

WARNING: There are "resources" sections in the chart not set. Using "resourcesPreset" is not recommended for production. For production installations, please set the following values according to your workload needs:
- cloneStaticSiteFromGit.gitSync.resources
- resources
+info https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
```

helm list

NAME	NAMESPACE	REVISION	UPDATED	STATUS	CHART	APP VERSION
my-nginx	default	1	2025-08-26 12:44:51.6227197 +0530 IST	deployed	nginx-21.1.23	1.29.1

Checkpoint: Advantages of using Helm

Helm simplifies Kubernetes resource management by packaging applications into charts, and enabling easy installation, upgrades, and rollbacks. It reduces manual YAML configuration, and it ensures consistency, and supports reusable templates for complex deployments.

## Exercise 10: Debugging and Troubleshooting

Objective: Learn how to troubleshoot issues in Kubernetes

kubectl describe pod nginx-deployment-6cfb98644c-dw7r7

```
PS M:\Mohit\Kubernetes Assignment> kubectl describe pod nginx-deployment-6cfb98644c-dw7r7
Name:          nginx-deployment-6cfb98644c-dw7r7
Namespace:     default
Priority:       0
Service Account: default
Node:          minikube/192.168.49.2
Start Time:    Mon, 25 Aug 2025 23:34:18 +0530
Labels:        app=nginx-deployment
               pod-template-hash=6cfb98644c
Annotations:   <none>
Status:        Running
IP:            10.244.0.15
IPs:           IP: 10.244.0.15
Controlled By: ReplicaSet/nginx-deployment-6cfb98644c
Containers:
  nginx:
    Container ID:  docker://23d8dc31567895c1122e6d95f626f966208ae2f903e2252d17542c7e34ffbed5
    Image:         nginx
    Image ID:      docker-pullable://nginx@sha256:33e0bbc7ca9ecf108140af6288c7c9d1ecc77548cbfd3952fd8466a75edefe57
    Port:          <none>
    Host Port:     <none>
    State:         Running
      Started:     Tue, 26 Aug 2025 08:47:25 +0530
    Last State:    Terminated
      Reason:      Error
      Exit Code:   255
      Started:     Mon, 25 Aug 2025 23:34:22 +0530
      Finished:    Tue, 26 Aug 2025 08:46:08 +0530
    Ready:         True
    Restart Count: 1
    Environment:   <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-7nrdk (ro)
Conditions:
  Type                 Status
  PodReadyToStartContainers  True
  Initialized            True
  Ready                  True
  ContainersReady        True
  PodScheduled           True
Volumes:
  kube-api-access-7nrdk:
    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:                <none>
```

kubectl get nodes

```
PS M:\Mohit\Kubernetes Assignment> kubectl get nodes
NAME          STATUS    ROLES          AGE   VERSION
minikube      Ready     control-plane  14h   v1.33.1
```

kubectl get events

```

PS N:\Mohit\kubernetes Assignment> kubectl get events
LAST SEEN   TYPE      REASON   OBJECT                MESSAGE
17m         Normal    Scheduled pod/load-generator      Successfully assigned default/load-generator to minikube
16m         Normal    Pulling   pod/load-generator    Pulling image "busybox"
17m         Normal    Pulled    pod/load-generator     Successfully pulled image "busybox" in 6.809s (6.01s including waiting). Image size: 4429366 bytes.
16m         Normal    Created   pod/load-generator     Created container: load-generator
16m         Normal    Started   pod/load-generator     Started container load-generator
16m         Normal    Pulled    pod/load-generator     Successfully pulled image "busybox" in 2.31s (2.31s including waiting). Image size: 4429366 bytes.
5m51s       Normal    Scheduled pod/my-nginx-594d78ffc7-qnrw2  Successfully assigned default/my-nginx-594d78ffc7-qnrw2 to minikube
5m42s       Normal    Pulling   pod/my-nginx-594d78ffc7-qnrw2  Pulling image "docker.io/bitnami/nginx:1.29.1-debian-12-r8"
4m36s       Normal    Pulled    pod/my-nginx-594d78ffc7-qnrw2  Successfully pulled image "docker.io/bitnami/nginx:1.29.1-debian-12-r8" in 1m5.745s (1m5.745s including waiting). Image size: 185996216 bytes.
4m36s       Normal    Created   pod/my-nginx-594d78ffc7-qnrw2  Created container: preserve-logs-symlinks
4m35s       Normal    Started   pod/my-nginx-594d78ffc7-qnrw2  Started container preserve-logs-symlinks
4m34s       Normal    Pulled    pod/my-nginx-594d78ffc7-qnrw2  Container image "docker.io/bitnami/nginx:1.29.1-debian-12-r8" already present on machine
4m33s       Normal    Created   pod/my-nginx-594d78ffc7-qnrw2  Created container: nginx
4m32s       Normal    Started   pod/my-nginx-594d78ffc7-qnrw2  Started container nginx
5m55s       Normal    SuccessfulCreate replicaset/my-nginx-594d78ffc7-qnrw2  Created pod: my-nginx-594d78ffc7-qnrw2
5m57s       Normal    NoPods    poddisruptionbudget/my-nginx  No matching pods found
5m53s       Normal    ScalingReplicaSet deployment/my-nginx  Scaled up replica set my-nginx-594d78ffc7 from 0 to 1
5m          Normal    ExternalProvisioning persistentvolumeclaim/my-pvc  Waiting for a volume to be created either by the external provisioner 'k8s.io/minikube-hostpath' or manually by the sys
ten administrator. If volume creation is delayed, please verify that the provisioner is running and correctly registered.
5m          Normal    Provisioning persistentvolumeclaim/my-pvc  External provisioner is provisioning volume for claim "default/my-pvc"
5m          Normal    ProvisioningSucceeded persistentvolumeclaim/my-pvc  Successfully provisioned volume pvc-30e35e97-fa74-48bf-bd3e-cbd5599e8ud
3m          Normal    Scheduled pod/mysql-0              Successfully assigned default/mysql-0 to minikube
3m          Normal    Pulled    pod/mysql-0            Pulling image "mysql:8"
3m          Normal    Created   pod/mysql-0            Successfully pulled image "mysql:8" in 2m33.333s (2m33.333s including waiting). Image size: 785726559 bytes.
3m          Normal    Started   pod/mysql-0            Created container: mysql
3m          Normal    Scheduled pod/mysql-1              Successfully assigned default/mysql-1 to minikube
3m          Normal    Pulled    pod/mysql-1            Container image "mysql:8" already present on machine
3m          Normal    Started   pod/mysql-1            Started container mysql
3m          Normal    Started   pod/mysql-1            create Pod mysql-0 in StatefulSet mysql successful
3m          Normal    SuccessfulCreate statefulset/mysql      create Pod mysql-1 in StatefulSet mysql successful
3m          Normal    FailedGetResourceMetric horizontalpodautoscaler/nginx-hpa  failed to get cpu utilization: unable to get metrics for resource cpu: unable to fetch metrics from resource metrics AP
I: the server is currently unable to handle the request (get pods.metrics.k8s.io)
17m         Warning   FailedComputeMetricsReplicas horizontalpodautoscaler/nginx-hpa  invalid metrics (1 invalid out of 1), first error is: failed to get cpu resource metric value: failed to get cpu utiliz
ation: unable to get metrics for resource cpu: unable to fetch metrics from resource metrics API: the server is currently unable to handle the request (get pods.metrics.k8s.io)
3m9s        Warning   FailedGetResourceMetric horizontalpodautoscaler/nginx-hpa  failed to get cpu utilization: missing request for cpu in container nginx of Pod nginx-deployment-6cfb98644c-dw7r7
1m          Warning   FailedComputeMetricsReplicas horizontalpodautoscaler/nginx-hpa  invalid metrics (1 invalid out of 1), first error is: failed to get cpu resource metric value: failed to get cpu utiliz
ation: missing request for cpu in container nginx of Pod nginx-deployment-6cfb98644c-dw7r7
13m         Warning   FailedGetResourceMetric horizontalpodautoscaler/nginx-hpa  failed to get cpu utilization: missing request for cpu in container nginx of Pod nginx-deployment-6cfb98644c-g95c7
5m          Warning   FailedScheduling       pod/pv-pod             0/1 nodes are available: pod has unbound immediate PersistentVolumeClaims. preemption: 0/1 nodes are available: 1 Preem
ption is not helpful for scheduling.
55m         Normal    Scheduled pod/pv-pod              Successfully assigned default/pv-pod to minikube
55m         Normal    Pulling   pod/pv-pod             Pulling image "nginx"
55m         Normal    Pulled    pod/pv-pod             Successfully pulled image "nginx" in 2.861s (2.861s including waiting). Image size: 192385880 bytes.
55m         Normal    Created   pod/pv-pod             Created container: nginx
55m         Normal    Started   pod/pv-pod             Started container nginx
5m          Normal    Killing   pod/pv-pod             Stopping container nginx
53m         Normal    Scheduled pod/pv-pod              Successfully assigned default/pv-pod to minikube
53m         Normal    Pulling   pod/pv-pod             Pulling image "nginx"
53m         Normal    Pulled    pod/pv-pod             Successfully pulled image "nginx" in 2.299s (2.299s including waiting). Image size: 192385880 bytes.

```

## kubectl logs nginx-deployment-6cfb98644c-dw7r7

```

10.244.0.24 - - [26/Aug/2025:07:14:10 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.244.0.24 - - [26/Aug/2025:07:14:10 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.244.0.24 - - [26/Aug/2025:07:14:10 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.244.0.24 - - [26/Aug/2025:07:14:10 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.244.0.24 - - [26/Aug/2025:07:14:10 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.244.0.24 - - [26/Aug/2025:07:14:10 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.244.0.24 - - [26/Aug/2025:07:14:10 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.244.0.24 - - [26/Aug/2025:07:14:10 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.244.0.24 - - [26/Aug/2025:07:14:10 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.244.0.24 - - [26/Aug/2025:07:14:10 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.244.0.24 - - [26/Aug/2025:07:14:10 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"
10.244.0.24 - - [26/Aug/2025:07:14:10 +0000] "GET / HTTP/1.1" 200 615 "-" "Wget" "-"

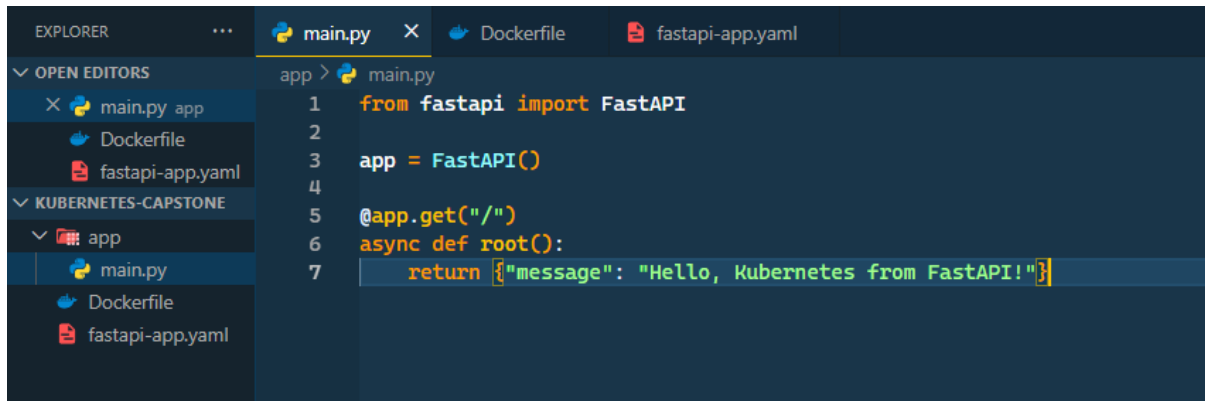
```

## Checkpoint: Common reasons for CrashLoopBackOff

CrashLoopBackOff occurs when a pod repeatedly fails to start, it often occurs due to misconfigured environment variables, missing dependencies, incorrect image versions, or application errors. We can diagnose these issues by checking kubectl logs and events (kubectl describe pod).

## Capstone Project: Kubernetes Application Deployment

Objective: Deploy a production-grade multi-container application using Kubernetes.

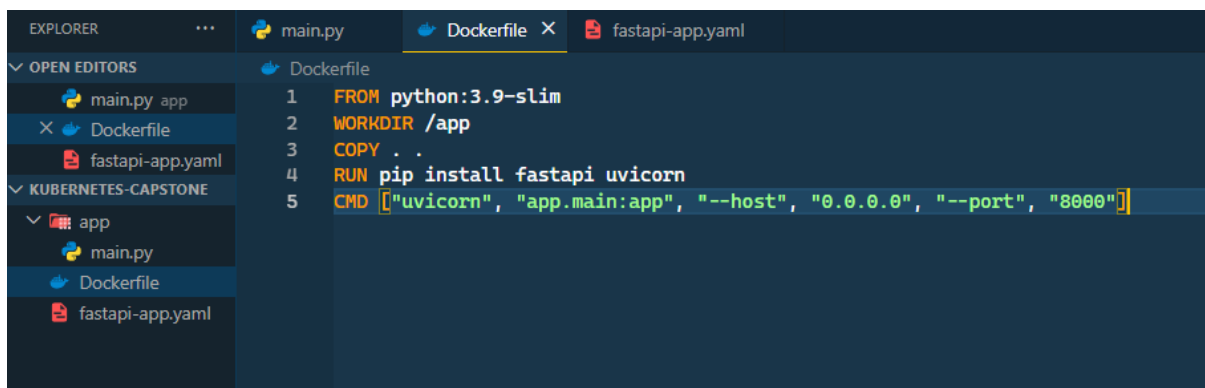


The screenshot shows the VS Code interface with the Explorer panel on the left and the Open Editors panel on the right. The Explorer panel shows the project structure with the following files and folders:

- main.py
- Dockerfile
- fastapi-app.yaml
- KUBERNETES-CAPSTONE
  - app
    - main.py
    - Dockerfile
    - fastapi-app.yaml

The Open Editors panel shows the main.py file with the following code:

```
1 from fastapi import FastAPI
2
3 app = FastAPI()
4
5 @app.get("/")
6 async def root():
7     return {"message": "Hello, Kubernetes from FastAPI!"}
```

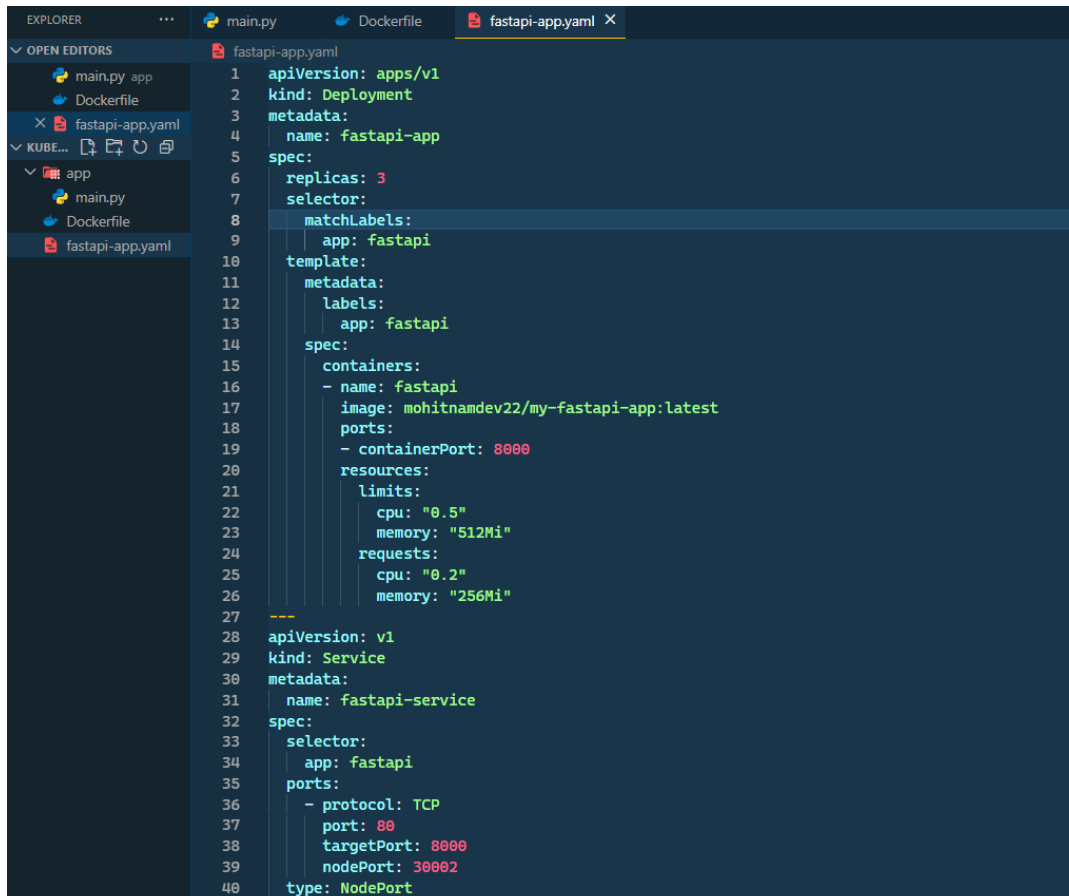


The screenshot shows the VS Code interface with the Explorer panel on the left and the Open Editors panel on the right. The Explorer panel shows the project structure with the following files and folders:

- main.py
- Dockerfile
- fastapi-app.yaml
- KUBERNETES-CAPSTONE
  - app
    - main.py
    - Dockerfile
    - fastapi-app.yaml

The Open Editors panel shows the Dockerfile file with the following code:

```
1 FROM python:3.9-slim
2 WORKDIR /app
3 COPY . .
4 RUN pip install fastapi uvicorn
5 CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]
```



The screenshot shows the VS Code interface with the Explorer panel on the left and the Open Editors panel on the right. The Explorer panel shows the project structure with the following files and folders:

- main.py
- Dockerfile
- fastapi-app.yaml
- KUBERNETES-CAPSTONE
  - app
    - main.py
    - Dockerfile
    - fastapi-app.yaml

The Open Editors panel shows the fastapi-app.yaml file with the following code:

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: fastapi-app
5 spec:
6   replicas: 3
7   selector:
8     matchLabels:
9       app: fastapi
10  template:
11    metadata:
12      labels:
13        app: fastapi
14    spec:
15      containers:
16        - name: fastapi
17          image: mohitnamdev22/my-fastapi-app:latest
18          ports:
19            - containerPort: 8000
20          resources:
21            limits:
22              cpu: "0.5"
23              memory: "512Mi"
24            requests:
25              cpu: "0.2"
26              memory: "256Mi"
27 ---
28 apiVersion: v1
29 kind: Service
30 metadata:
31   name: fastapi-service
32 spec:
33   selector:
34     app: fastapi
35   ports:
36     - protocol: TCP
37       port: 80
38       targetPort: 8000
39       nodePort: 30002
40   type: NodePort
```

docker build -t my-fastapi-app:latest .

```
PS M:\Mohit\Kubernetes Assignment\capstone\kubernetes-capstone> docker build -t my-fastapi-app:latest .
[+] Building 41.4s (10/10) FINISHED
=> [internal] load build definition from Dockerfile                                docker:desktop-linux 0.7s
=> => transferring dockerfile: 188B                                              0.1s
=> [internal] load metadata for docker.io/library/python:3.9-slim                4.1s
=> [auth] library/python:pull token for registry-1.docker.io                    0.0s
=> [internal] load .dockerignore                                                 0.1s
=> => transferring context: 2B                                                  0.0s
=> [1/4] FROM docker.io/library/python:3.9-slim@sha256:914169c7c8398b1b90c0b0ff921c8027445e39d7c25dc440337e56ce0 0.3s
=> => resolve docker.io/library/python:3.9-slim@sha256:914169c7c8398b1b90c0b0ff921c8027445e39d7c25dc440337e56ce0 0.2s
=> [internal] load build context                                                0.2s
=> => transferring context: 1.17kB                                              0.1s
=> CACHED [2/4] WORKDIR /app                                                    0.0s
=> [3/4] COPY . .                                                                0.1s
=> [4/4] RUN pip install fastapi uvicorn                                       29.1s
=> exporting to image                                                            5.8s
=> => exporting layers                                                            3.3s
=> => exporting manifest sha256:ca1f07725267d6951e7bfb275b81b1a28e09fcd50e2055dada54ec6d6a0ae904      0.0s
=> => exporting config sha256:efa953d0c2eda9f97e37c6ac4c97d619a5839b90bd88e2dd29ab457eeb456105        0.0s
=> => exporting attestation manifest sha256:1daa9572a8391d483d2f4640629af31aa22ded960031119fdb5d4c5e55b9b61d 0.1s
=> => exporting manifest list sha256:5991fb17740fc365b02b311e41b3505afe0e96b2bc7331d3e484aee76a6879b3    0.0s
=> => naming to docker.io/library/my-fastapi-app:latest                        0.0s
=> => unpacking to docker.io/library/my-fastapi-app:latest                      2.1s
```

docker push mohitnamdev22/my-fastapi-app:latest

```
PS M:\Mohit\Kubernetes Assignment\capstone\kubernetes-capstone> docker push mohitnamdev22/my-fastapi-app:latest
The push refers to repository [docker.io/mohitnamdev22/my-fastapi-app]
5ec99fe17015: Pushed
396b1da7636e: Mounted from library/postgres
0219e1e5e6ef: Pushed
770480a1ec4b: Pushed
7493a7aecd6e: Pushed
ea3499df304f: Pushed
23c6b5cea222: Pushed
5a15fe23c360: Pushed
latest: digest: sha256:5991fb17740fc365b02b311e41b3505afe0e96b2bc7331d3e484aee76a6879b3 size: 856
```

kubectl apply -f fastapi-app.yaml

```
PS M:\Mohit\Kubernetes Assignment\capstone\kubernetes-capstone> kubectl apply -f fastapi-app.yaml
deployment.apps/fastapi-app created
service/fastapi-service created
```

minikube service fastapi-service --url

```
PS M:\Mohit\Kubernetes Assignment\capstone\kubernetes-capstone> minikube service fastapi-service --url
http://127.0.0.1:62544
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```

kubectl get pods

```
PS M:\Mohit\Kubernetes Assignment\capstone\kubernetes-capstone> kubectl get pods
NAME                                READY    STATUS    RESTARTS    AGE
config-pod                         1/1      Running   0            123m
fastapi-app-65fc889b9c-gzm2c       1/1      Running   0            2m58s
fastapi-app-65fc889b9c-rc8j8       1/1      Running   0            2m58s
fastapi-app-65fc889b9c-v9nsd       1/1      Running   0            2m57s
load-generator                     1/1      Running   1 (46m ago)  47m
my-nginx-594d78ffc7-qnrw2          1/1      Running   0            36m
mysql-0                             1/1      Running   0            64m
mysql-1                             1/1      Running   0            62m
nginx-deployment-6cfb98644c-dw7r7  1/1      Running   1 (4h35m ago) 13h
nginx-deployment-6cfb98644c-g95c7  1/1      Running   1 (4h35m ago) 13h
nginx-deployment-6cfb98644c-x65pz  1/1      Running   1 (4h35m ago) 13h
pv-pod                             1/1      Running   0            83m
PS M:\Mohit\Kubernetes Assignment\capstone\kubernetes-capstone> |
```

kubectl get svc

```
PS M:\Mohit\Kubernetes Assignment\capstone\kubernetes-capstone> kubectl get svc
NAME                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
fastapi-service     NodePort      10.99.154.32   <none>         80:30002/TCP     11m
kubernetes           ClusterIP     10.96.0.1      <none>         443/TCP          15h
my-nginx            LoadBalancer 10.101.92.247  <pending>      80:32028/TCP,443:31127/TCP 38m
mysql-service       ClusterIP     None           <none>         3306/TCP         66m
nginx-deployment    NodePort      10.97.217.29   <none>         80:31922/TCP     4h31m
nginx-pod           NodePort      10.102.96.2    <none>         80:32507/TCP     14h
```