**Install minikube using brew in macos**

brew install minikube

**Start minikube**

minimube start

**Get available nodes in Kubernetes**

Kubectl get nodes

NAME STATUS ROLES AGE VERSION

minikube Ready control-plane 3m53s v1.26.3

**Check minikube installed version with latest version**

minikube update-check

**Stop minikube cluster running**

minikube stop

**Delete minikube cluster**

minikube delete

**Get namespaces in kubernetes**

kubectl get namespaces

NAME STATUS AGE

default Active 4d15h

kube-node-lease Active 4d15h

kube-public Active 4d15h

kube-system Active 4d15h

**Create namespace.yaml file**

kubectl apply -f namespace.yaml

namespace/development created

**Check again namespaces in kubernetes**

kubectl get namespaces

NAME STATUS AGE

default Active 4d15h

development Active 45s

kube-node-lease Active 4d15h

kube-public Active 4d15h

kube-system Active 4d15h

**Create new namespace production in same file and run yaml file:**

kubectl apply -f namespace.yaml

namespace/development unchanged

namespace/production created

**Check again namespaces in Kubernetes**

kubectl get namespaces

NAME STATUS AGE

default Active 4d15h

development Active 4m27s

kube-node-lease Active 4d15h

kube-public Active 4d15h

kube-system Active 4d15h

production Active 76s

**To delete namesapces from Kubernetes run:**

kubectl delete -f namespace.yaml

namespace "development" deleted

namespace "production" deleted

**Deploy an application**

**Get namespaces from Kubernetes**

kubectl get ns

NAME STATUS AGE

default Active 4d15h

kube-node-lease Active 4d15h

kube-public Active 4d15h

kube-system Active 4d15h

**Create development namespace:**

kubectl apply -f namespace.yaml

namespace/development created

namespace/production created

kubectl apply -f deployment.yaml

deployment.apps/pod-info-deployment created

**Confirmed by checking development namespace**

kubectl get deployments -n development

NAME READY UP-TO-DATE AVAILABLE AGE

pod-info-deployment 3/3 3 3 4m34s

**Get pods created by development**

kubectl get pods -n development

NAME READY STATUS RESTARTS AGE

pod-info-deployment-68c8476764-lcjmk 1/1 Running 0 5m48s

pod-info-deployment-68c8476764-xzrkv 1/1 Running 0 5m48s

pod-info-deployment-68c8476764-zbm6w 1/1 Running 0 5m49s

**Check always three pods is running (We are going to delete one pod and check again)**

kubectl delete pod pod-info-deployment-68c8476764-lcjmk -n development

pod "pod-info-deployment-68c8476764-lcjmk" deleted

**Now check again running pods count**

kubectl get pods -n development

NAME READY STATUS RESTARTS AGE

pod-info-deployment-68c8476764-xzrkv 1/1 Running 0 9m35s

pod-info-deployment-68c8476764-zbm6w 1/1 Running 0 9m36s

pod-info-deployment-68c8476764-zj24m 1/1 Running 0 92s

**(Deleted pod "pod-info-deployment-68c8476764-lcjmk" get removed and new pod created as per replica count requirement)**

**Check pod log and describe about pod**

kubectl describe pod pod-info-deployment-68c8476764-xzrkv -n development

Name: pod-info-deployment-68c8476764-xzrkv

Namespace: development

Priority: 0

Service Account: default

Node: minikube/192.168.49.2

Start Time: Mon, 14 Jul 2025 13:01:59 +0530

Labels: app=pod-info

pod-template-hash=68c8476764

Annotations: <none>

Status: Running

IP: 10.244.0.12

IPs:

IP: 10.244.0.12

Controlled By: ReplicaSet/pod-info-deployment-68c8476764

Containers:

pod-info-container:

Container ID: docker://95b9c1e137fe8e5bcb1fa773446c57ee17d12f1c43676a1868193dc7ddce21ad

Image: kimschles/pod-info-app:latest

Image ID: docker-pullable://kimschles/pod-info-app@sha256:fa4f33bc2301bb242bdd078ac206d0e379dfed2e225d46a6952ff444ae6f4a7a

Port: 3000/TCP

Host Port: 0/TCP

State: Running

Started: Mon, 14 Jul 2025 13:02:18 +0530

Ready: True

Restart Count: 0

Environment:

POD\_NAME: pod-info-deployment-68c8476764-xzrkv (v1:metadata.name)

POD\_NAMESPACE: development (v1:metadata.namespace)

POD\_IP: (v1:status.podIP)

Mounts:

/var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-bxgp6 (ro)

Conditions:

Type Status

PodReadyToStartContainers True

Initialized True

Ready True

ContainersReady True

PodScheduled True

Volumes:

kube-api-access-bxgp6:

Type: Projected (a volume that contains injected data from multiple sources)

TokenExpirationSeconds: 3607

ConfigMapName: kube-root-ca.crt

Optional: false

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 27m default-scheduler Successfully assigned development/pod-info-deployment-68c8476764-xzrkv to minikube

Normal Pulling 27m kubelet Pulling image "kimschles/pod-info-app:latest"

Normal Pulled 27m kubelet Successfully pulled image "kimschles/pod-info-app:latest" in 3.326s (18.823s including waiting). Image size: 206915626 bytes.

Normal Created 27m kubelet Created container: pod-info-container

Normal Started 27m kubelet Started container pod-info-container

**Delete all pods created using deployment.yaml for development namespace**

kubectl delete -f deployment.yaml

deployment.apps "pod-info-deployment" deleted

kubectl get pods -n development

No resources found in development namespace.

**Check that our application is working with BusyBox**

**Create Busybox pod using busybox.yaml**

kubectl apply -f busybox.yaml

deployment.apps/busybox created

**Get running pods**

kubectl get pods

NAME READY STATUS RESTARTS AGE

busybox-6c747767dd-9rtzp 1/1 Running 0 81s

**Run exce command:**

**exce : execute**

**-it : interactive terminal**

kubectl exec -it busybox-6c747767dd-9rtzp -- /bin/sh

/ # wget

BusyBox v1.37.0 (2024-09-26 21:31:42 UTC) multi-call binary.

Usage: wget [-cqS] [--spider] [-O FILE] [-o LOGFILE] [--header STR]

[--post-data STR | --post-file FILE] [-Y on/off]

[--no-check-certificate] [-P DIR] [-U AGENT] [-T SEC] URL...

Retrieve files via HTTP or FTP

--spider Only check URL existence: $? is 0 if exists

--header STR Add STR (of form 'header: value') to headers

--post-data STR Send STR using POST method

--post-file FILE Send FILE using POST method

--no-check-certificate Don't validate the server's certificate

-c Continue retrieval of aborted transfer

-q Quiet

-P DIR Save to DIR (default .)

-S Show server response

-T SEC Network read timeout is SEC seconds

-O FILE Save to FILE ('-' for stdout)

-o LOGFILE Log messages to FILE

-U STR Use STR for User-Agent header

-Y on/off Use proxy

**Know the ip address of Kubernetes clusters from development namespace**

kubectl get pods -n development -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

pod-info-deployment-68c8476764-l5vx9 1/1 Running 0 31s 10.244.0.15 minikube <none> <none>

pod-info-deployment-68c8476764-n6kb6 1/1 Running 0 31s 10.244.0.16 minikube <none> <none>

pod-info-deployment-68c8476764-tb8qd 1/1 Running 0 31s 10.244.0.17 minikube <none> <none>

**Now copy IP Address of one of the pod and use with wget command to connect:**

/ # wget 10.244.0.15

Connecting to 10.244.0.15 (10.244.0.15:80)

wget: can't connect to remote host (10.244.0.15): Connection refused

**Ip Address is unable to connect because it is using port 80 but in our deployment.yaml file we have assigned container port 3000**

/ # wget 10.244.0.15:3000

Connecting to 10.244.0.15:3000 (10.244.0.15:3000)

saving to 'index.html'

index.html 100% |\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*| 104 0:00:00 ETA

'index.html' saved

**Now see index.html file using cat command**

/ # cat index.html

{"pod\_name":"pod-info-deployment-68c8476764-l5vx9","pod\_namespace":"development","pod\_ip":"10.244.0.15"}/ #

**Now exit busybox using exit command**

/ # exit

**View Application logs from pod’s container**

kubectl logs pod-info-deployment-68c8476764-l5vx9 -n development

undefined

Example app listening on port 3000

**Expose application to the internet with a LoadBalancer**

**Create service.yaml file for LoadBalancer**

1. **Start minikube tunnel**

minikube tunnel

✅ Tunnel successfully started

📌 NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessible ...

1. **Create Service:**

kubectl apply -f service.yaml

service/demo-service created

1. **Get IP Address of service:**

kubectl get services -n development

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

demo-service LoadBalancer 10.102.92.203 127.0.0.1 80:30692/TCP 101s

1. **Copy External-IP and paste into browser, the tunnel will ask for password of system to run on browser:**

The service/ingress demo-service requires privileged ports to be exposed: [80]

🔑 sudo permission will be asked for it.

🏃 Starting tunnel for service demo-service.

Password:

1. **Browser will now show the details as below:**

{

"pod\_name": "pod-info-deployment-68c8476764-tb8qd",

"pod\_namespace": "development",

"pod\_ip": "10.244.0.17"

}

**Add resource requests and limits to the pod**

Change in deployment.yaml file and create new yaml file with name deployment\_resource.yaml

resources:

requests:

cpu: 250m

memory: 64Mi

limits:

cpu: 500m

memory: 128Mi

**Create pods using deployment\_resource.yaml:**

kubectl apply -f deployment\_resource.yaml

deployment.apps/pod-info-deployment configured

**Check running pods:**

kubectl get pods -n development

NAME READY STATUS RESTARTS AGE

pod-info-deployment-79b94dfd9c-96wmp 1/1 Running 0 109s

pod-info-deployment-79b94dfd9c-gm2b2 1/1 Running 0 100s

pod-info-deployment-79b94dfd9c-pmb65 1/1 Running 0 104s

**Use describe command to get details of one of the pods from namespace development:**

kubectl describe pod pod-info-deployment-79b94dfd9c-96wmp -n development

Name: pod-info-deployment-79b94dfd9c-96wmp

Namespace: development

Priority: 0

Service Account: default

Node: minikube/192.168.49.2

Start Time: Mon, 14 Jul 2025 18:10:09 +0530

Labels: app=pod-info

pod-template-hash=79b94dfd9c

Annotations: <none>

Status: Running

IP: 10.244.0.18

IPs:

IP: 10.244.0.18

Controlled By: ReplicaSet/pod-info-deployment-79b94dfd9c

Containers:

pod-info-container:

Container ID: docker://c71d82de01a9059e1157e1338f575822c6c5a4964384e6651a473accaa16db0e

Image: kimschles/pod-info-app:latest

Image ID: docker-pullable://kimschles/pod-info-app@sha256:fa4f33bc2301bb242bdd078ac206d0e379dfed2e225d46a6952ff444ae6f4a7a

Port: 3000/TCP

Host Port: 0/TCP

State: Running

Started: Mon, 14 Jul 2025 18:10:14 +0530

Ready: True

Restart Count: 0

Limits:

cpu: 500m

memory: 128Mi

Requests:

cpu: 250m

memory: 64Mi

Environment:

POD\_NAME: pod-info-deployment-79b94dfd9c-96wmp (v1:metadata.name)

POD\_NAMESPACE: development (v1:metadata.namespace)

POD\_IP: (v1:status.podIP)

Mounts:

/var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-zrbxk (ro)

Conditions:

Type Status

PodReadyToStartContainers True

Initialized True

Ready True

ContainersReady True

PodScheduled True

Volumes:

kube-api-access-zrbxk:

Type: Projected (a volume that contains injected data from multiple sources)

TokenExpirationSeconds: 3607

ConfigMapName: kube-root-ca.crt

Optional: false

DownwardAPI: true

QoS Class: Burstable

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 5m4s default-scheduler Successfully assigned development/pod-info-deployment-79b94dfd9c-96wmp to minikube

Normal Pulling 5m4s kubelet Pulling image "kimschles/pod-info-app:latest"

Normal Pulled 5m kubelet Successfully pulled image "kimschles/pod-info-app:latest" in 3.406s (3.407s including waiting). Image size: 206915626 bytes.

Normal Created 5m kubelet Created container: pod-info-container

Normal Started 5m kubelet Started container pod-info-container

**Delete our Kubernetes objects and tear down our cluster**

**Delete file using:**

**Delete busybox.yaml**

kubectl delete -f busybox.yaml

deployment.apps "busybox" deleted

**Delete deployment.yaml**

kubectl delete -f deployment.yaml

deployment.apps "pod-info-deployment" deleted

**Delete quote.yaml**

kubectl delete -f quote.yaml

deployment.apps "quote -service" deleted

**Delete service.yaml**

kubectl delete -f service.yaml

service "demo-service" deleted

**Delete namespace.yaml**

kubectl delete -f namespace.yaml

namespace "development" deleted

namespace "production" deleted

**Finally delete minikube cluster**

minikube delete

🔥 Deleting "minikube" in docker ...

🔥 Deleting container "minikube" ...

🔥 Removing /Users/ravishankarkushwaha/.minikube/machines/minikube ...

💀 Removed all traces of the "minikube" cluster.

**Kubernetes Architecture**

**To know all the api resources run below command:**

kubectl api-resources

NAME SHORTNAMES APIVERSION NAMESPACED KIND

bindings v1 true Binding

componentstatuses cs v1 false ComponentStatus

configmaps cm v1 true ConfigMap

endpoints ep v1 true Endpoints

events ev v1 true Event

limitranges limits v1 true LimitRange

namespaces ns v1 false Namespace

nodes no v1 false Node

persistentvolumeclaims pvc v1 true PersistentVolumeClaim

persistentvolumes pv v1 false PersistentVolume

pods po v1 true Pod

podtemplates v1 true PodTemplate

replicationcontrollers rc v1 true ReplicationController

resourcequotas quota v1 true ResourceQuota

secrets v1 true Secret

serviceaccounts sa v1 true ServiceAccount

services svc v1 true Service

mutatingwebhookconfigurations admissionregistration.k8s.io/v1 false MutatingWebhookConfiguration

validatingadmissionpolicies admissionregistration.k8s.io/v1 false ValidatingAdmissionPolicy

validatingadmissionpolicybindings admissionregistration.k8s.io/v1 false ValidatingAdmissionPolicyBinding

validatingwebhookconfigurations admissionregistration.k8s.io/v1 false ValidatingWebhookConfiguration

customresourcedefinitions crd,crds apiextensions.k8s.io/v1 false CustomResourceDefinition

apiservices apiregistration.k8s.io/v1 false APIService

controllerrevisions apps/v1 true ControllerRevision

daemonsets ds apps/v1 true DaemonSet

deployments deploy apps/v1 true Deployment

replicasets rs apps/v1 true ReplicaSet

statefulsets sts apps/v1 true StatefulSet

selfsubjectreviews authentication.k8s.io/v1 false SelfSubjectReview

tokenreviews authentication.k8s.io/v1 false TokenReview

localsubjectaccessreviews authorization.k8s.io/v1 true LocalSubjectAccessReview

selfsubjectaccessreviews authorization.k8s.io/v1 false SelfSubjectAccessReview

selfsubjectrulesreviews authorization.k8s.io/v1 false SelfSubjectRulesReview

subjectaccessreviews authorization.k8s.io/v1 false SubjectAccessReview

horizontalpodautoscalers hpa autoscaling/v2 true HorizontalPodAutoscaler

cronjobs cj batch/v1 true CronJob

jobs batch/v1 true Job

certificatesigningrequests csr certificates.k8s.io/v1 false CertificateSigningRequest

leases coordination.k8s.io/v1 true Lease

endpointslices discovery.k8s.io/v1 true EndpointSlice

events ev events.k8s.io/v1 true Event

flowschemas flowcontrol.apiserver.k8s.io/v1 false FlowSchema

prioritylevelconfigurations flowcontrol.apiserver.k8s.io/v1 false PriorityLevelConfiguration

ingressclasses networking.k8s.io/v1 false IngressClass

ingresses ing networking.k8s.io/v1 true Ingress

ipaddresses ip networking.k8s.io/v1 false IPAddress

networkpolicies netpol networking.k8s.io/v1 true NetworkPolicy

servicecidrs networking.k8s.io/v1 false ServiceCIDR

runtimeclasses node.k8s.io/v1 false RuntimeClass

poddisruptionbudgets pdb policy/v1 true PodDisruptionBudget

clusterrolebindings rbac.authorization.k8s.io/v1 false ClusterRoleBinding

clusterroles rbac.authorization.k8s.io/v1 false ClusterRole

rolebindings rbac.authorization.k8s.io/v1 true RoleBinding

roles rbac.authorization.k8s.io/v1 true Role

priorityclasses pc scheduling.k8s.io/v1 false PriorityClass

csidrivers storage.k8s.io/v1 false CSIDriver

csinodes storage.k8s.io/v1 false CSINode

csistoragecapacities storage.k8s.io/v1 true CSIStorageCapacity

storageclasses sc storage.k8s.io/v1 false StorageClass

volumeattachments storage.k8s.io/v1 false VolumeAttachment

**Kube API server is containerized application run as pod, to see all pods run kube-system namespace with the kubectl get pods command:**

kubectl get pods -n kube-system

NAME READY STATUS RESTARTS AGE

coredns-674b8bbfcf-npv2x 1/1 Running 1 (4m7s ago) 4m20s

etcd-minikube 1/1 Running 0 4m26s

kube-apiserver-minikube 1/1 Running 0 4m26s

kube-controller-manager-minikube 1/1 Running 0 4m26s

kube-proxy-2j6pj 1/1 Running 0 4m21s

kube-scheduler-minikube 1/1 Running 0 4m26s

storage-provisioner 1/1 Running 1 (3m51s ago) 4m24s

**Find the etcd pod in the kube system namespace:**

kubectl logs etcd-minikube -n kube-system | jq .

**{**

**"level":** "warn"**,**

**"ts":** "2025-07-14T13:01:46.980490Z"**,**

**"caller":** "embed/config.go:689"**,**

**"msg":** "Running http and grpc server on single port. This is not recommended for production."

**}**

**{**

**"level":** "warn"**,**

**"ts":** "2025-07-14T13:01:46.980841Z"**,**

**"caller":** "etcdmain/config.go:389"**,**

**"msg":** "--proxy-refresh-interval is deprecated in 3.5 and will be decommissioned in 3.6."

**}**

**{**

**"level":** "info"**,**

**"ts":** "2025-07-14T13:01:46.980886Z"**,**

**"caller":** "etcdmain/etcd.go:73"**,**

**"msg":** "Running: "**,**

**"args": [**

"etcd"**,**

"--advertise-client-urls=https://192.168.49.2:2379"**,**

"--cert-file=/var/lib/minikube/certs/etcd/server.crt"**,**

"--client-cert-auth=true"**,**

"--data-dir=/var/lib/minikube/etcd"**,**

"--experimental-initial-corrupt-check=true"**,**

"--experimental-watch-progress-notify-interval=5s"**,**

"--initial-advertise-peer-urls=https://192.168.49.2:2380"**,**

"--initial-cluster=minikube=https://192.168.49.2:2380"**,**

"--key-file=/var/lib/minikube/certs/etcd/server.key"**,**

"--listen-client-urls=https://127.0.0.1:2379,https://192.168.49.2:2379"**,**

"--listen-metrics-urls=http://127.0.0.1:2381"**,**

"--listen-peer-urls=https://192.168.49.2:2380"**,**

"--name=minikube"**,**

"--peer-cert-file=/var/lib/minikube/certs/etcd/peer.crt"**,**

"--peer-client-cert-auth=true"**,**

"--peer-key-file=/var/lib/minikube/certs/etcd/peer.key"**,**

"--peer-trusted-ca-file=/var/lib/minikube/certs/etcd/ca.crt"**,**

"--proxy-refresh-interval=70000"**,**

"--snapshot-count=10000"**,**

"--trusted-ca-file=/var/lib/minikube/certs/etcd/ca.crt"

**]**

**}**