

Ragul R R

05.02.2025

Minikube Setup and Usage Documentation

Introduction

This documentation describes the process of setting up and using Minikube to run a Kubernetes cluster on a local development environment. It includes installing Minikube, starting the Minikube cluster, deploying an application, exposing the application via a service, and accessing the service.

Installation

Prerequisites

- A system with virtualization enabled.
- kubectl installed.
- A hypervisor such as Docker, VirtualBox, or Hyper-V.

Installing Minikube

curl -LO

<https://github.com/kubernetes/minikube/releases/latest/download/minikube-linux-amd64>

sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64

```
ubuntu@LAPTOP-DEQKQVPU:~$ curl -LO https://github.com/kubernetes/minikube/releases/latest/download/minikube-linux-amd64
sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
  0     0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
  0     0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
100 119M 100 119M    0     0 4949k    0  0:00:24  0:00:24 --:--:-- 8540k
```

Starting Minikube

minikube start

```

ubuntu@LAPTOP-DEQKQVPU:~$ minikube start
🐳 minikube v1.35.0 on Ubuntu 24.04 (amd64)
🔧 Automatically selected the docker driver. Other choices: none, ssh
🔥 Using Docker driver with root privileges
👉 Starting "minikube" primary control-plane node in "minikube" cluster
📡 Pulling base image v0.0.46 ...
📦 Downloading Kubernetes v1.32.0 preload ...
> preloaded-images-k8s-v18-v1...: 333.57 MiB / 333.57 MiB 100.00% 3.40 Mi
> gcr.io/k8s-minikube/kicbase...: 500.31 MiB / 500.31 MiB 100.00% 4.67 Mi
🔥 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
💡 kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
🎉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

```

This command initializes and starts a Minikube cluster using the default configurations.

Installing kubectl

```

curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl && rm kubectl

```

```

ubuntu@LAPTOP-DEQKQVPU:~$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 138    100 138    0     0  185      0  --:--:-- --:--:-- --:--:--  185
100 54.6M  100 54.6M  0     0 2027k    0  0:00:27  0:00:27 --:--:-- 6075k
ubuntu@LAPTOP-DEQKQVPU:~$ chmod +x kubectl
ubuntu@LAPTOP-DEQKQVPU:~$ sudo mv kubectl /usr/local/bin/
ubuntu@LAPTOP-DEQKQVPU:~$ kubectl version --client
Client Version: v1.32.1
Kustomize Version: v5.5.0

```

```

ubuntu@LAPTOP-DEQKQVPU:~$ kubectl get po -A
NAMESPACE   NAME                                     READY   STATUS    RESTARTS   AGE
kube-system  coredns-668d6bf9bc-wngp5              1/1     Running   0           11m
kube-system  etcd-minikube                         1/1     Running   0           11m
kube-system  kube-apiserver-minikube               1/1     Running   0           11m
kube-system  kube-controller-manager-minikube      1/1     Running   0           11m
kube-system  kube-proxy-dwl6b                     1/1     Running   0           11m
kube-system  kube-scheduler-minikube              1/1     Running   0           11m
kube-system  storage-provisioner                  1/1     Running   2 (9m11s ago)  11m
ubuntu@LAPTOP-DEQKQVPU:~$ minikube version
minikube version: v1.35.0
commit: dd5d320e41b5451cdf3c01891bc4e13d189586ed-dirty

```

```

ubuntu@LAPTOP-DEQKQVPU:~$ minikube dashboard
🔧 Enabling dashboard ...
  ▪ Using image docker.io/kubernetesui/dashboard:v2.7.0
  ▪ Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
💡 Some dashboard features require the metrics-server addon. To enable all features please run:

    minikube addons enable metrics-server

🐳 Verifying dashboard health ...
🔥 Launching proxy ...
📡 Verifying proxy health ...

```

This installs the latest version of kubectl.

Deploying an Application

kubectl create deployment my-deployment --image=nginx

```
ubuntu@LAPTOP-DEQKQVPU:~$ kubectl create deployment nginx-deployment --image=nginx
deployment.apps/nginx-deployment created
ubuntu@LAPTOP-DEQKQVPU:~$ kubectl expose deployment nginx-deployment --type=NodePort --port=80
service/nginx-deployment exposed
ubuntu@LAPTOP-DEQKQVPU:~$ kubectl get pods
NAME                                READY   STATUS             RESTARTS   AGE
nginx-deployment-6cfb98644c-756zd   0/1     ContainerCreating   0           22s

ubuntu@LAPTOP-DEQKQVPU:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

This creates a new deployment named my-deployment using the Nginx image.

Getting Deployments and Pods

kubectl get deployments

kubectl get pods

```
ubuntu@LAPTOP-DEQKQVPU:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6cfb98644c-756zd   1/1     Running   0           3m31s

ubuntu@LAPTOP-DEQKQVPU:~$ kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment    1/1     1             1           8m52s

ubuntu@LAPTOP-DEQKQVPU:~$ kubectl get svc
NAME                TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes          ClusterIP   10.96.0.1       <none>        443/TCP          26m
nginx-deployment    NodePort    10.99.103.111   <none>        80:30583/TCP     8m54s
```

These commands retrieve the list of deployments and pods in the cluster.

Exposing the Application

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

```
ubuntu@LAPTOP-DEQKQVPU:~$ kubectl expose deployment nginx-deployment --type=NodePort --port=80
service/nginx-deployment exposed
```

This exposes the my-deployment as a service accessible on port 80 using NodePort.

Accessing the Service

minikube service my-deployment --url

```
ubuntu@LAPTOP-DEQKQVPU:~$ minikube service nginx-deployment --url
http://127.0.0.1:37233
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```

This command retrieves the URL for accessing the service in the Minikube cluster.

Conclusion

This documentation provides a step-by-step guide to setting up Minikube, deploying an application, and exposing it as a service. By following these steps, users can effectively run a Kubernetes cluster on a local environment.