Varshni D S

05.02.2025

Minikube: Local Kubernetes Cluster Setup

Overview

This guide provides a step-by-step approach to setting up and using Minikube for running a Kubernetes cluster in a local development environment. It covers installation, initialization, application deployment, service exposure, and service access.

Setup Process

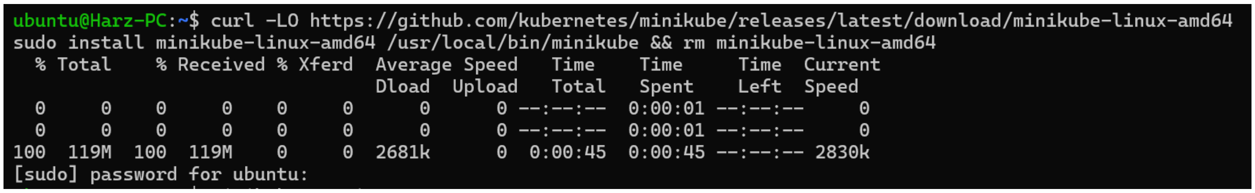
System Requirements

- Virtualization support enabled.
- Installed version of kubectl.
- A hypervisor such as Docker, VirtualBox, or Hyper-V.

Installing Minikube

To install Minikube, run:

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

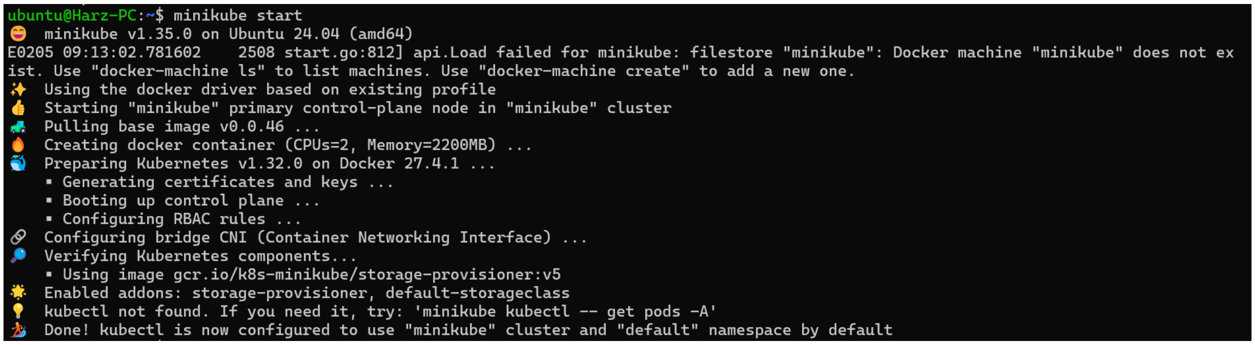


```
ubuntu@Harz-PC:~$ curl -LO https://github.com/kubernetes/minikube/releases/latest/download/minikube-linux-amd64
sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:--  0:00:01 --:--:--     0
  0     0    0     0    0     0      0      0 --:--:--  0:00:01 --:--:--     0
100  119M  100  119M    0     0  2681k      0  0:00:45  0:00:45 --:--:-- 2830k
[sudo] password for ubuntu:
```

Initializing Minikube

To start a Minikube cluster, execute:

minikube start



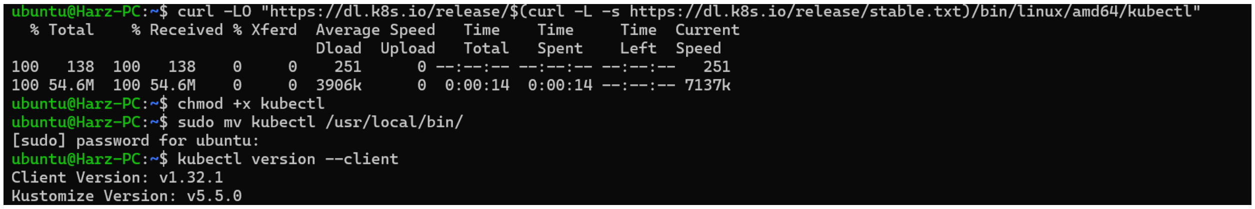
```
ubuntu@Harz-PC:~$ minikube start
😄  minikube v1.35.0 on Ubuntu 24.04 (amd64)
E0205 09:13:02.781602    2508 start.go:812] api.Load failed for minikube: filestore "minikube": Docker machine "minikube" does not ex
ist. Use "docker-machine ls" to list machines. Use "docker-machine create" to add a new one.
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.46 ...
🔥  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 initializes and runs a Minikube cluster with default settings.

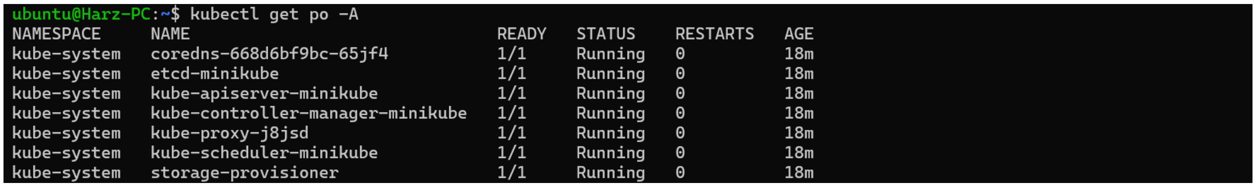
Installing kubectl

Ensure kubectl is installed using:

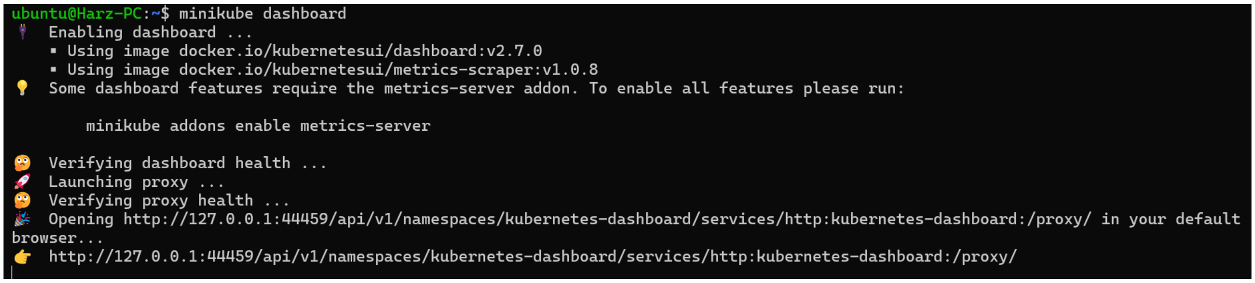
curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl && rm kubectl



```
ubuntu@Harz-PC:~$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0    251      0 --:--:-- --:--:-- --:--:--   251
100 54.6M  100 54.6M    0     0  3906k      0  0:00:14  0:00:14 --:--:-- 7137k
ubuntu@Harz-PC:~$ chmod +x kubectl
ubuntu@Harz-PC:~$ sudo mv kubectl /usr/local/bin/
[sudo] password for ubuntu:
ubuntu@Harz-PC:~$ kubectl version --client
Client Version: v1.32.1
Kustomize Version: v5.5.0
```



```
ubuntu@Harz-PC:~$ kubectl get po -A
NAMESPACE     NAME                               READY   STATUS    RESTARTS   AGE
kube-system   coredns-668d6bf9bc-65jf4           1/1     Running   0          18m
kube-system   etcd-minikube                      1/1     Running   0          18m
kube-system   kube-apiserver-minikube            1/1     Running   0          18m
kube-system   kube-controller-manager-minikube   1/1     Running   0          18m
kube-system   kube-proxy-j8jsd                   1/1     Running   0          18m
kube-system   kube-scheduler-minikube            1/1     Running   0          18m
kube-system   storage-provisioner                1/1     Running   0          18m
```



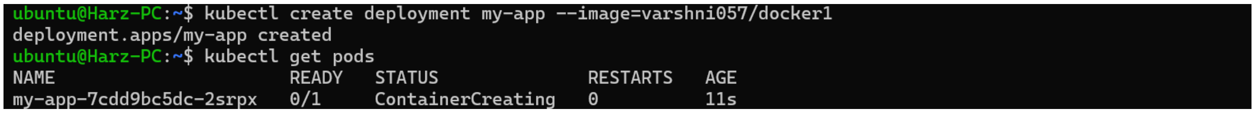
```
ubuntu@Harz-PC:~$ minikube dashboard
🔌  Enabling dashboard ...
    ▪ Using image docker.io/kubernetesui/dashboard:v2.7.0
    ▪ Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
💡  Some dashboard features require the metrics-server addon. To enable all features please run:

        minikube addons enable metrics-server

🤔  Verifying dashboard health ...
🚀  Launching proxy ...
🤔  Verifying proxy health ...
🎉  Opening http://127.0.0.1:44459/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default
browser...
👉  http://127.0.0.1:44459/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/
```

This fetches and installs the latest kubectl version.

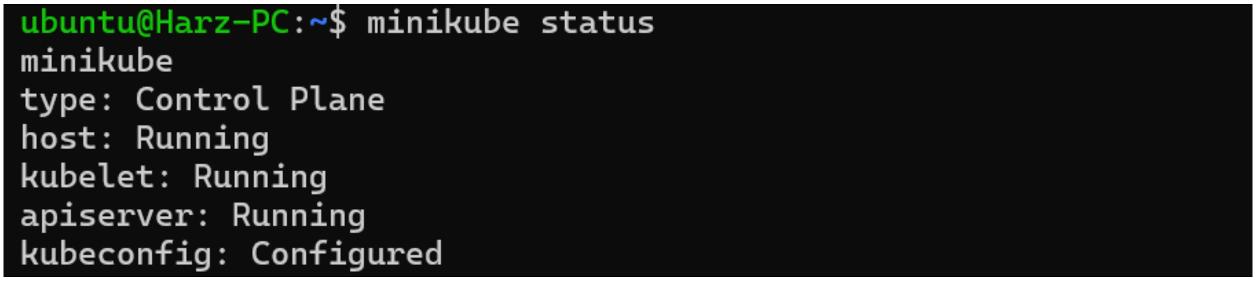
Deploying an Application

To create a deployment using the Nginx image, run:

kubectl create deployment my-deployment --image=nginx



```
ubuntu@Harz-PC:~$ kubectl create deployment my-app --image=varshni057/docker1
deployment.apps/my-app created
ubuntu@Harz-PC:~$ kubectl get pods
NAME                      READY   STATUS              RESTARTS   AGE
my-app-7cdd9bc5dc-2srpx   0/1     ContainerCreating   0          11s
```

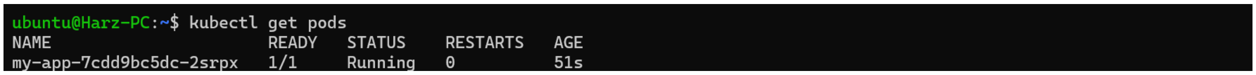


```
ubuntu@Harz-PC:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

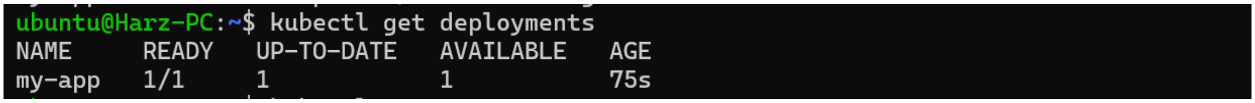
Viewing Deployments and Pods

To check active deployments and pods, use:

kubectl get deployments
kubectl get pods



```
ubuntu@Harz-PC:~$ kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
my-app-7cdd9bc5dc-2srpx   1/1     Running   0          51s
```

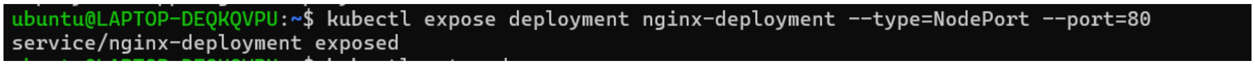


```
ubuntu@Harz-PC:~$ kubectl get deployments
NAME     READY   UP-TO-DATE   AVAILABLE   AGE
my-app   1/1     1            1           75s
```

Exposing the Deployment

To make the application accessible, expose it as a service:

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

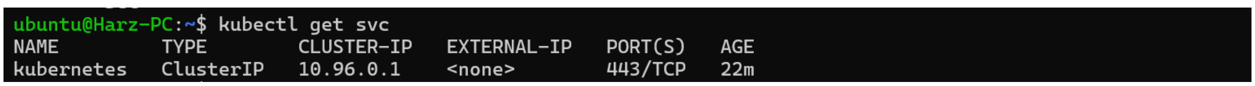


```
ubuntu@LAPTOP-DEQKQVPU:~$ kubectl expose deployment nginx-deployment --type=NodePort --port=80
service/nginx-deployment exposed
```

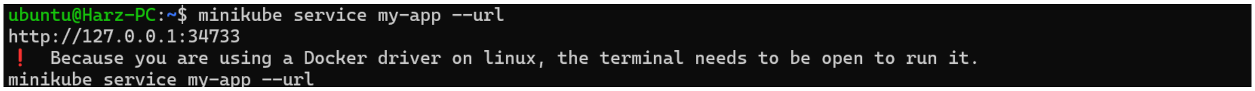
Retrieving the Service URL

To access the exposed service, execute:

minikube service my-deployment --url



```
ubuntu@Harz-PC:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   22m
```



```
ubuntu@Harz-PC:~$ minikube service my-app --url
http://127.0.0.1:34733
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
minikube service my-app --url
```

This returns the URL needed to interact with the running service.

Summary

Following these steps enables users to set up Minikube, deploy applications, and expose services efficiently. This guide ensures a smooth workflow for running a Kubernetes cluster in a local environment.