Minikube

Installation

Step 1 : Install required packages:

1. Update the package information on the system by entering the following command:

sudo apt update

2. Install curl and apt-transport-https:

sudo apt install curl apt-transport-https

```
poojashree@poojashree-virtual-machine:~$ sudo apt install curl apt-transport-https
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (7.81.0-1ubuntu1.20).
apt-transport-https is already the newest version (2.4.13).
0 upgraded, 0 newly installed, 0 to remove and 213 not upgraded.
```

Step 2 : Download Minikube Binary

1. Use **curl** to download the latest Minikube binary:

curl -O https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

```
poojashree@poojashree-virtual-machine:~$ curl -O https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  119M  100  119M    0     0  3641k      0  0:00:33  0:00:33 --:--:-- 4985k
```

2. Copy the downloaded file and store it in the */usr/local/bin/* directory:

sudo cp minikube-linux-amd64 /usr/local/bin/minikube

The command prints no output.

Step 3 : Enable Minikube Binary Execution

sudo chmod 755 /usr/local/bin/minikube

Verify the installation by checking the Minikube version:

minikube version

```
poojashree@poojashree-virtual-machine:~$ minikube version
minikube version: v1.35.0
commit: dd5d320e41b5451cdf3c01891bc4e13d189586ed-dirty
```

Step 4 : Install kubectl

sudo snap install kubectl –classic

Step 5 : Start Minikube

minikube start

```
poojashree@poojashree-virtual-machine:~$ minikube start
😄  minikube v1.35.0 on Ubuntu 22.04
✨  Automatically selected the docker driver. Other choices: ssh, none

❗  The requested memory allocation of 1920MiB does not leave room for system overhead (total system memory: 1920MiB). You may face stability issues.
💡  Suggestion: Start minikube with less memory allocated: 'minikube start --memory=1920mb'

📌  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% 854.17
    > gcr.io/k8s-minikube/kicbase...:  500.31 MiB / 500.31 MiB  100.00% 595.31
🔥  Creating docker container (CPUs=2, Memory=1920MB) ...
🐳  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
```

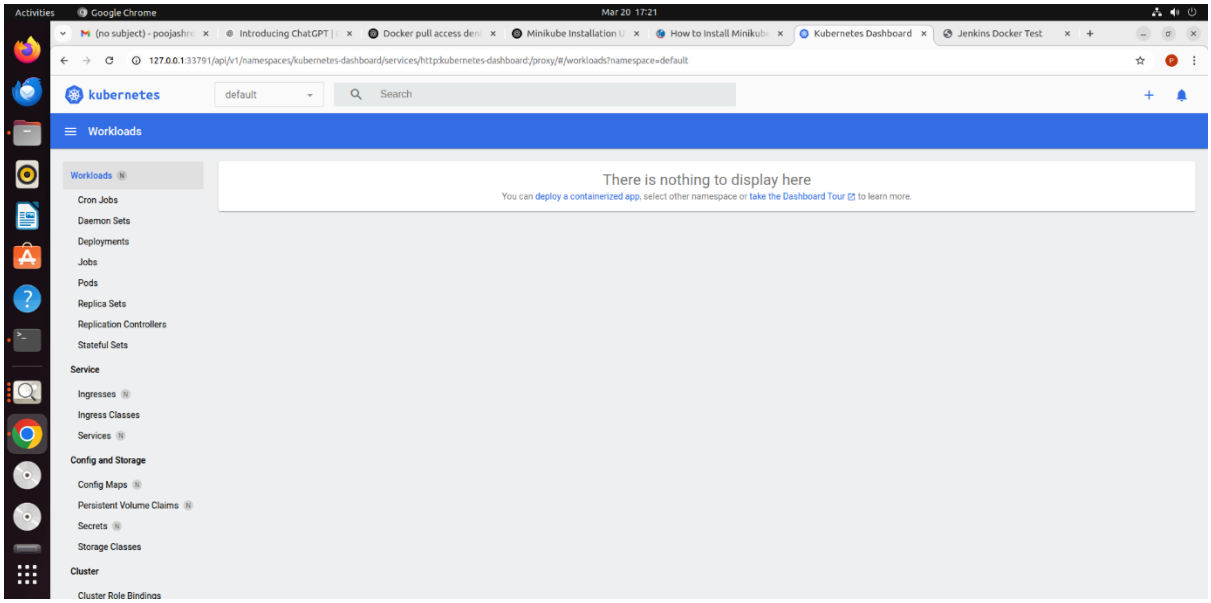
minikube status

```
poojashree@poojashree-virtual-machine:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

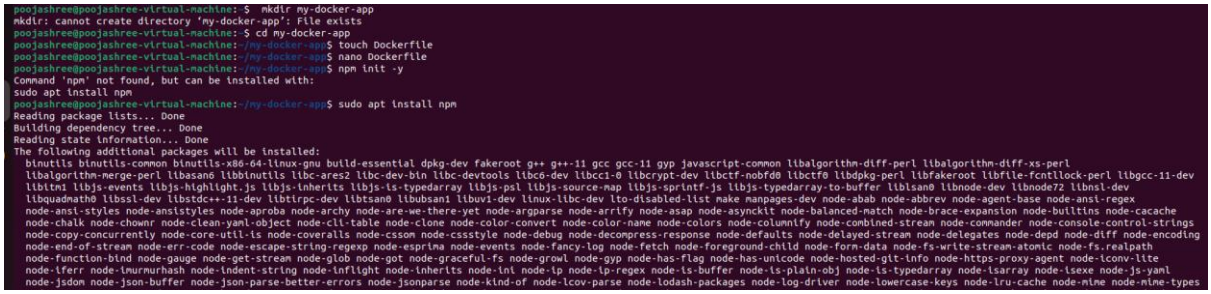
Access minikube dashboard

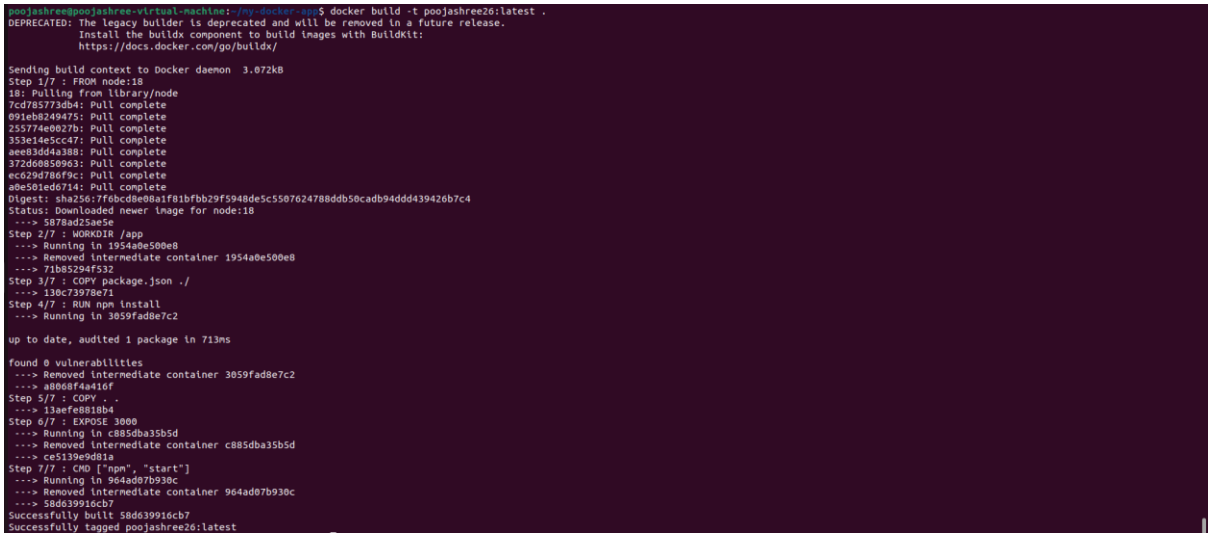
minikube dashboard

```
poojashree@poojashree-virtual-machine:~$ 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://127.0.0.1:33791/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
Opening in existing browser session.
```



Task 3





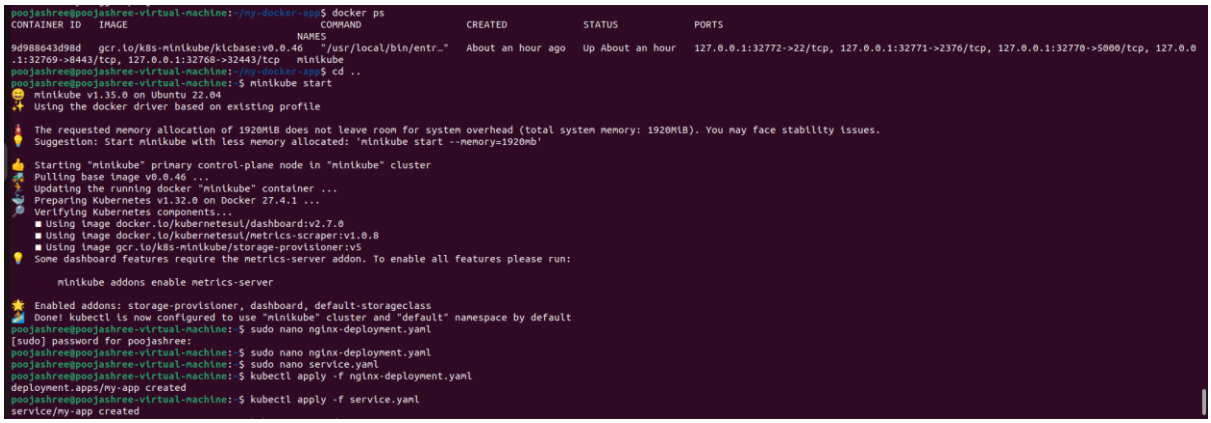
sudo nano ngnix-deployment.yaml

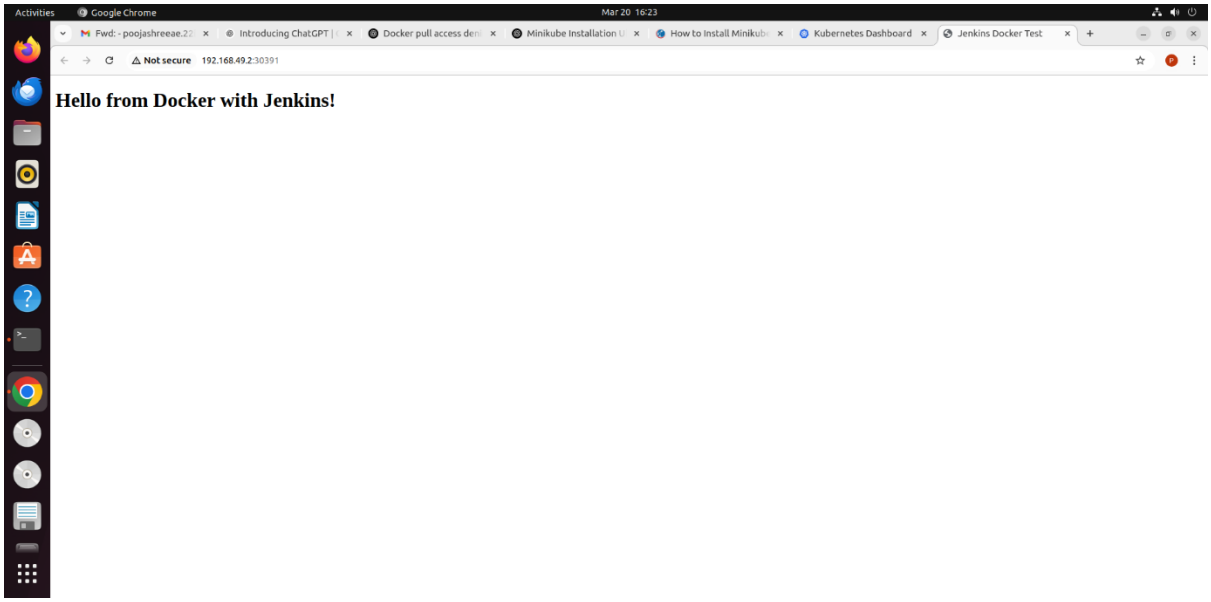
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
      - name: my-app
```

```
      image: nadinc/docker_pipe:latest
      imagePullPolicy: IfNotPresent
      ports:
      - containerPort: 80
```

sudo nano service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: my-app
  namespace: default
spec:
  type: NodePort # Ensures external access via a specific port
  selector:
    app: my-app
  ports:
    - protocol: TCP
      port: 80 # Service port inside the cluster
      targetPort: 80 # The container's port
      nodePort: 30391 # Externally accessible port
```




Activities
Google Chrome
Mar 20 16:23
Jenkins Docker Test
Not secure
192.168.49.2:30391
Hello from Docker with Jenkins!