

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

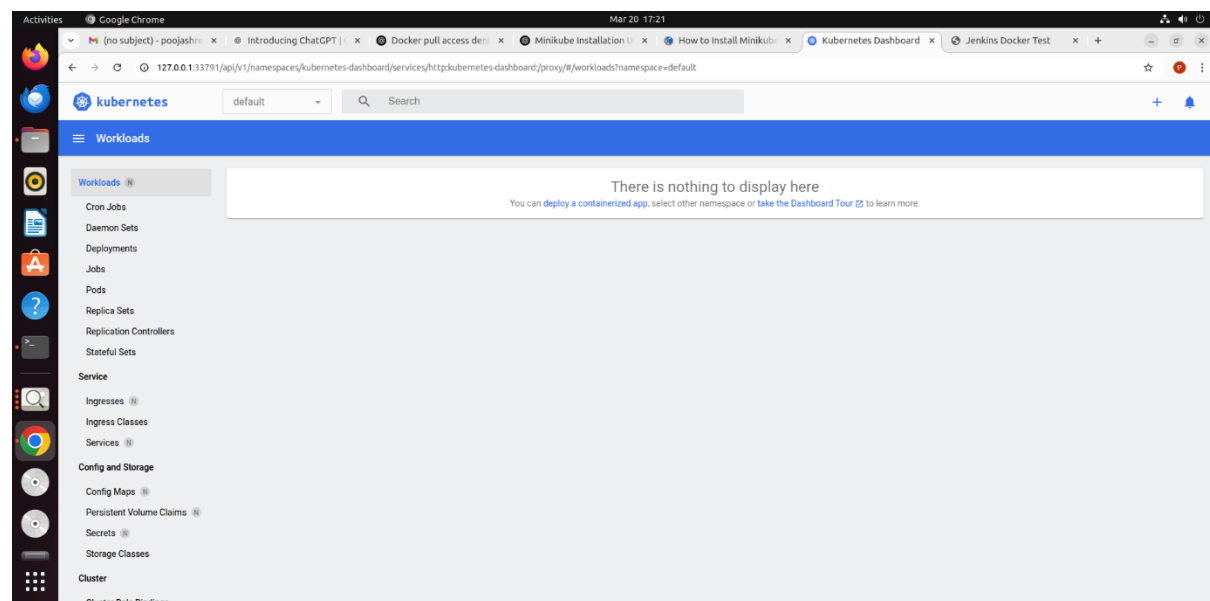
```
poojashree@poojashree-virtual-machine:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

Access minikube dashboard

minikube dashboard

```
poojashree@poojashree-virtual-machine:~$ minikube dashboard
Enabling dashboard ...
  Using image docker.io/kubernetes/dashboard:v2.7.0
  Using image docker.io/kubernetes/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

```
poojashree@poojashree-virtual-machine:~$ mkdir my-docker-app
Mkdir: cannot create directory 'my-docker-app': File exists
poojashree@poojashree-virtual-machine:~$ cd my-docker-app
poojashree@poojashree-virtual-machine:~/my-docker-app$ touch Dockerfile
poojashree@poojashree-virtual-machine:~/my-docker-app$ nano Dockerfile
poojashree@poojashree-virtual-machine:~/my-docker-app$ npm init -y
Command 'npm' not found, but can be installed with:
sudo apt install npm
poojashree@poojashree-virtual-machine:~/my-docker-app$ sudo apt install npm
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev fakeroot g++ g++-11 gcc gcc-11 gyp javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan6 libbinutils libc-ares2 libc-dev-bin libc-devtools libcbo-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libfakeroot libfile-fcntllock-perl libgcc-11-dev
  libitm1 libjs-events libjs-highlight.js libjs-inherits libjs-is-typedarray libjs-psl libjs-source-map libjs-sprintf.js libjs-typedarray-to-buffer liblsan0 libnode-dev libnode72 libnsl-dev
  libquadmath0 libssl-dev libstdc++-11-dev libtirpc-dev libtsan0 libubsan1 libuv1-dev linux-libc-dev lto-disabled-list make manpages-dev node-abab node-abbrev node-agent-base node-ansi-regex
  node-ansi-styles node-anisstyles node-aproba node-archy node-are-we-there-yet node-argparse node-arrify node-asap node-async-l1 node-balanced-match node-brace-expansion node-bulldoze node-cacache
  node-chalk node-chownr node-clean-yaml-object node-cli-table node-clone node-color-convert node-color-name node-colors node-columnify node-combined-stream node-commander node-console-control-strings
  node-copy-concurrently node-core-util-is node-coveralls node-cssson node-cssstyle node-debug node-decompress-response node-defaults node-delayed-stream node-delegates node-depd node-diff node-encoding
  node-end-of-stream node-err-code node-escape-string-regexp node-esprima node-events node-fancy-log node-fetch node-foreground-child node-form-data node-fs-write-stream-atomic node-fs.realpath
  node-function-bind node-gauge node-get-stream node-glob node-got node-graceful-fs node-growl node-gyp node-has-flag node-has-unicode node-hosted-git-info node-https-proxy-agent node-tconv-lite
  node-i18n node-lunurhash node-indent-string node-inflight node-inherits node-ini node-ip node-ip-regex node-is-buffer node-is-plain-obj node-is-typedarray node-isarray node-ls-exe node-js-yaml
  node-json node-json-buffer node-json-parse-better-errors node-jsonparse node-kind-of node-lcov-parse node-lodash-packages node-log-driver node-lowercase-keys node-lru-cache node-mime node-mime-types
```

```
poojashree@poojashree-virtual-machine:~/my-docker-app$ npm init -y
wrote to /home/poojashree/my-docker-app/package.json:

{
  "name": "my-docker-app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

```
poojashree@poojashree-virtual-machine:~/my-docker-app$ docker build -t poojashree26:latest .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 3.072kB
Step 1/7 : FROM node:18
18: Pulling from library/node
7cd785773db4: Pull complete
691eb0249475: Pull complete
255774e0027b: Pull complete
353e14e5cc47: Pull complete
aee83dd4a388: Pull complete
372d00859631: Pull complete
ec629d786f9c: Pull complete
a8e501ed0714: Pull complete
Digest: sha256:7f6bdc0e08a181bb29f594de5c507624788db58cadb94dd439426b7c4
Status: Downloaded newer image for node:18
--> 5878ad25ae5e
Step 2/7 : WORKDIR /app
--> Running in 1954a0e500e8
--> Removed intermediate container 1954a0e500e8
--> 71b85294f532
Step 3/7 : COPY package.json ./
--> 13c73297a271
Step 4/7 : RUN npm install
--> Running in 3059fad8e7c2

up to date, audited 1 package in 713ms

found 0 vulnerabilities
--> Removed intermediate container 3059fad8e7c2
--> a8008f4a416f
Step 5/7 : COPY . .
--> 13afe8818b4
Step 6/7 : EXPOSE 3000
--> Running in c885dba35b5d
--> Removed intermediate container c885dba35b5d
--> ce5139e9d81a
Step 7/7 : CMD ["npm", "start"]
--> Running in 964ad07b930c
--> Removed intermediate container 964ad07b930c
--> 58d639916cb7
Successfully built 58d639916cb7
Successfully tagged poojashree26:latest
```

sudo nano nginx-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

```
poojashree@poojashree-virtual-machine:~$ docker ps
CONTAINER ID   IMAGE                                NAMES      COMMAND                  CREATED        STATUS        PORTS
9d988643d9ad   gcr.io/k8s-minikube/kicbase:v0.0.46 /usr/local/bin/entr... About an hour ago Up About an hour 127.0.0.1:32769->8443/tcp, 127.0.0.1:32768->32443/tcp minikube
poojashree@poojashree-virtual-machine:~$ minikube start
minikube v1.35.0 on Ubuntu 22.04
Using the docker driver based on existing profile

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'

Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.46 ...
Updating the running docker "minikube" container ...
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
Verifying Kubernetes components...
  Using image docker.io/kubernetes/dashboard:v2.7.0
  Using image docker.io/kubernetes/metrics-scraper:v1.0.8
  Using image gcr.io/k8s-minikube/storage-provisioner:v5
Some dashboard features require the metrics-server addon. To enable all features please run:

minikube addons enable metrics-server

Enabled addons: storage-provisioner, dashboard, default-storageclass
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
poojashree@poojashree-virtual-machine:~$ sudo nano nginx-deployment.yaml
[sudo] password for poojashree:
poojashree@poojashree-virtual-machine:~$ sudo nano nginx-deployment.yaml
poojashree@poojashree-virtual-machine:~$ sudo nano service.yaml
deployment.apps/my-app created
poojashree@poojashree-virtual-machine:~$ kubectl apply -f service.yaml
service/my-app created
```

```
poojashree@poojashree-virtual-machine:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
my-app-7848f8498-cngrw   0/1     ContainerCreating   0          48s
poojashree@poojashree-virtual-machine:~$ kubectl get svc my-app
NAME    TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
my-app  NodePort    10.97.232.84   <none>        80:30391/TCP   63s
poojashree@poojashree-virtual-machine:~$ minikube service my-app --url
http://192.168.49.2:30391
```

```
poojashree@poojashree-virtual-machine:~$ curl http://192.168.49.2:30391
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Jenkins Docker Test</title>
</head>
<body>
  <h1>Hello from Docker with Jenkins!</h1>
</body>
</html>
```

