

## TASK 3-MiniKube Deployment Task

### Step 1: Start the minikube server

Starting the minikube using “minikube start” command

```
sharan@sharan:~$ sudo usermod -aG docker $USER && newgrp docker
sharan@sharan:~$ 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...: 3.08 MiB / 333.57 MiB 0.92% 20.25 KiB p
> gcr.io/k8s-minikube/kicbase...: 500.31 MiB / 500.31 MiB 100.00% 495.52
🔥 Creating docker container (CPUs=2, Memory=2200MB) ...
> kubectll.sha256: 64 B / 64 B [-----] 100.00% ? p/s 0s
> kubelet.sha256: 64 B / 64 B [-----] 100.00% ? p/s 0s
> kubeadm.sha256: 64 B / 64 B [-----] 100.00% ? p/s 0s
> kubelet: 73.81 MiB / 73.81 MiB [-----] 100.00% 2.28 MiB p/s 33s
> kubectll: 54.67 MiB / 54.67 MiB [-----] 100.00% 1.23 MiB p/s 45s
> kubeadm: 67.66 MiB / 67.66 MiB [-----] 100.00% 1.35 MiB p/s 50s

▪ Generating certificates and keys ...
▪ Booting up control plane ...
▪ Configuring RBAC rules ...
```

### Step 2: Install kubectl

```
sudo apt install -y kubectl
Hit:1 https://download.docker.com/linux/ubuntu noble InRelease
Ign:2 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:3 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:6 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:7 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:8 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

### Step 3: Create deployment

Now create a deployment named r2 using the image ‘vishal15276/test1’

```
gcr.io/k8s-minikube/kicbase v0.0.46 e72c4cbe9b29 2 months ago 1.31GB
sharan@sharan:~$ kubectl create deployment y --image=sharanprasath/test2 --port=80
deployment.apps/y created
sharan@sharan:~$ kubectl get pods
```

#### Step 4: Verify the pods

Now give `kubectl get pods` to check if the container is running and wait until it starts running

NAME	READY	STATUS	RESTARTS	AGE
p-7db547f989-7srbn	0/1	ImagePullBackOff	0	2m32s
sp-bbf5f7896-tvxn7	0/1	ContainerCreating	0	9s

```
sharan@Sharan:~$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
p-7db547f989-7srbn	1/1	Running	0	7m4s
sp-bbf5f7896-tvxn7	1/1	Running	0	4m41s

#### Step 5: Expose the deployment

Now expose the deployment using the `expose` command

```
sharan@Sharan:~$ kubectl expose deployment sp --port=80 --type=NodePort
service/sp exposed
```

#### Step 6: Accessing the website

Now give `service` command to check the ip address of the deployed image

```
sharan@Sharan:~$ minikube service sp
```

NAMESPACE	NAME	TARGET PORT	URL
default	sp	80	http://192.168.49.2:31250

🚀 Starting tunnel for service sp.

NAMESPACE	NAME	TARGET PORT	URL
default	sp		http://127.0.0.1:44365

🌐 Opening service default/sp in default browser...  
👉 http://127.0.0.1:44365  
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.

## Step 7: Output page

The output will be displayed as follows

