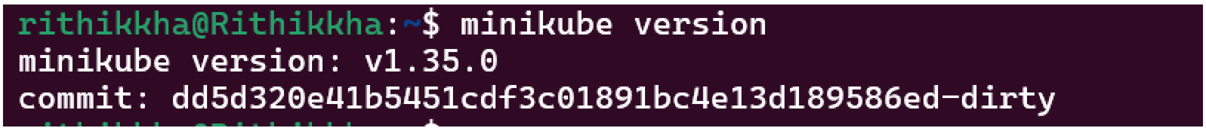
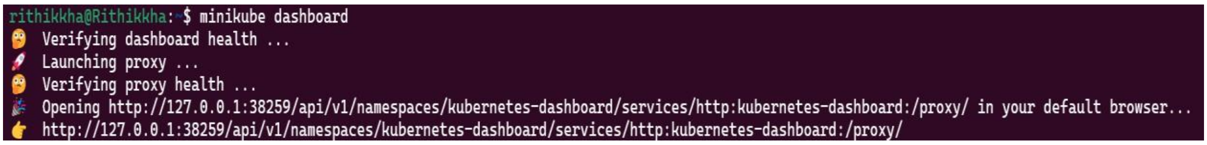
TASK 3 MINIKUBE

MINIKUBE VERSION

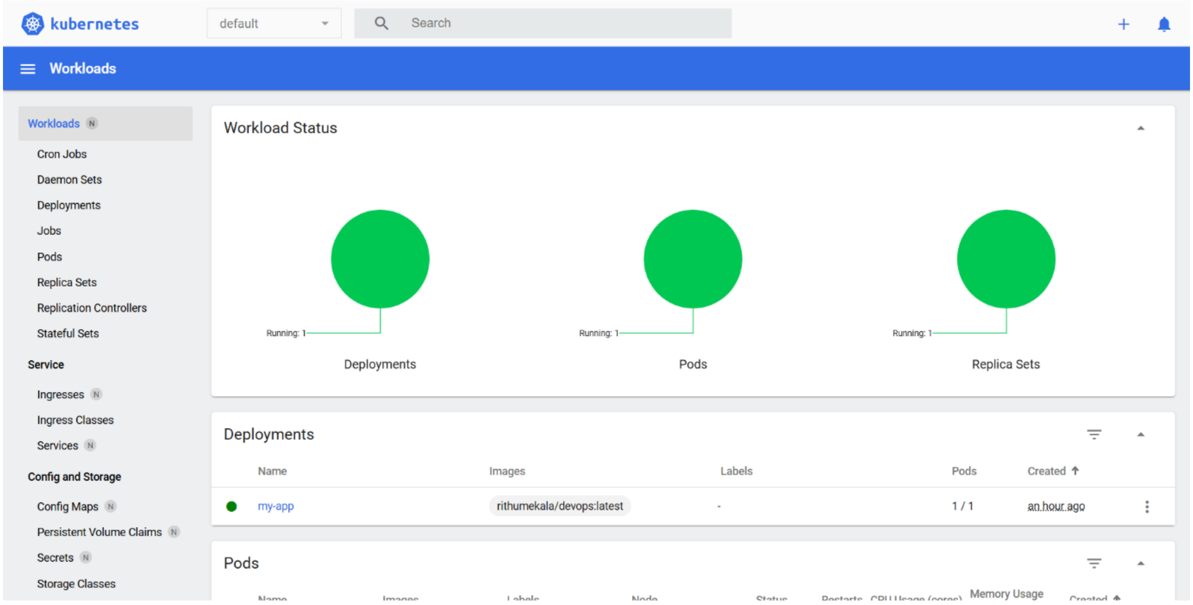


```
rithikkha@Rithikkha:~$ minikube version
minikube version: v1.35.0
commit: dd5d320e41b5451cdf3c01891bc4e13d189586ed-dirty
```

MINIKUBE DASHBOARD



```
rithikkha@Rithikkha:~$ minikube dashboard
🤔  Verifying dashboard health ...
🚀  Launching proxy ...
🤔  Verifying proxy health ...
🎉  Opening http://127.0.0.1:38259/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
👉  http://127.0.0.1:38259/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/
```



Deployments

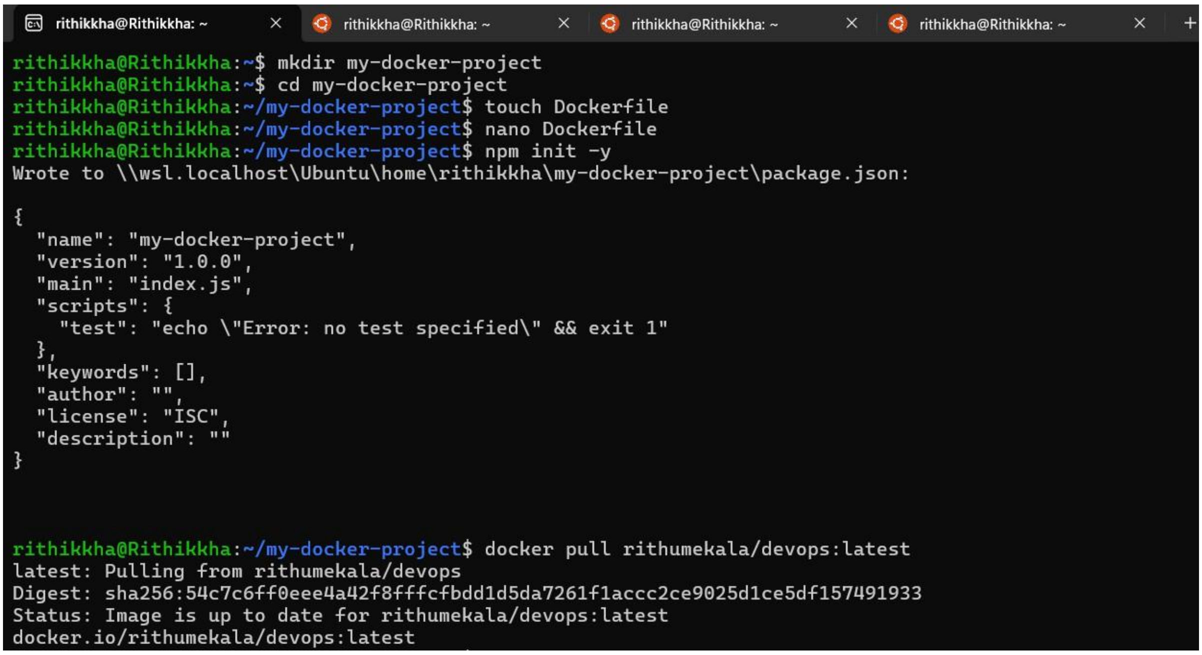
	Name	Images	Labels	Pods	Created ↑
●	my-app	rithumekala/devops:latest	-	1 / 1	an hour ago

Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage	Created ↑

TASK 3 MINIKUBE

CREATED A PROJECT



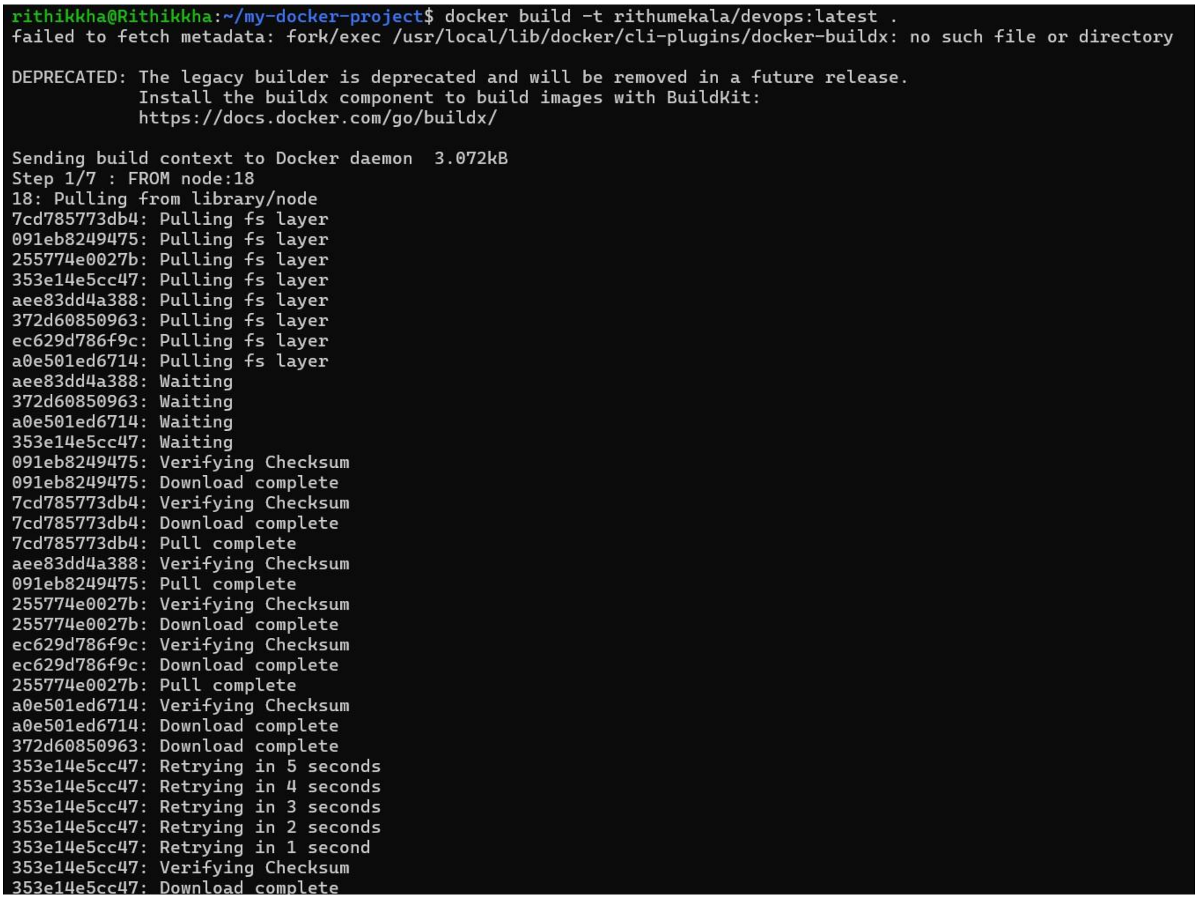
```
rithikkha@Rithikkha:~$ mkdir my-docker-project
rithikkha@Rithikkha:~$ cd my-docker-project
rithikkha@Rithikkha:~/my-docker-project$ touch Dockerfile
rithikkha@Rithikkha:~/my-docker-project$ nano Dockerfile
rithikkha@Rithikkha:~/my-docker-project$ npm init -y
Wrote to \\wsl.localhost\Ubuntu\home\rithikkha\my-docker-project\package.json:

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



rithikkha@Rithikkha:~/my-docker-project$ docker pull rithumekala/devops:latest
latest: Pulling from rithumekala/devops
Digest: sha256:54c7c6ff0eee4a42f8fffcfbdd1d5da7261f1accc2ce9025d1ce5df157491933
Status: Image is up to date for rithumekala/devops:latest
docker.io/rithumekala/devops:latest
```

SENDING BUILD CONTEXT TO DOCKER

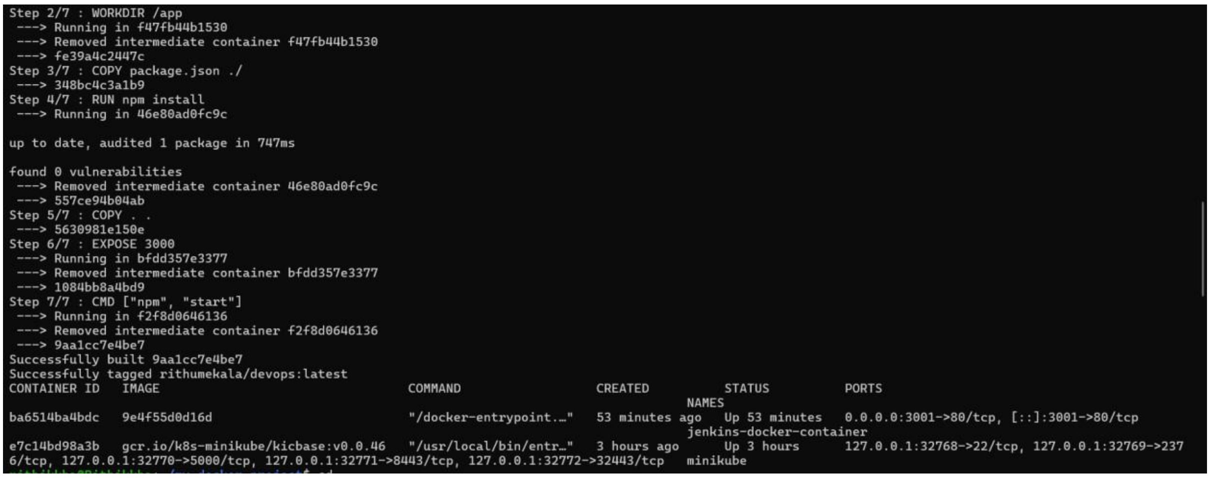


```
rithikkha@Rithikkha:~/my-docker-project$ docker build -t rithumekala/devops:latest .
failed to fetch metadata: fork/exec /usr/local/lib/docker/cli-plugins/docker-buildx: no such file or directory

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: Pulling fs layer
091eb8249475: Pulling fs layer
255774e0027b: Pulling fs layer
353e14e5cc47: Pulling fs layer
aee83dd4a388: Pulling fs layer
372d60850963: Pulling fs layer
ec629d786f9c: Pulling fs layer
a0e501ed6714: Pulling fs layer
aee83dd4a388: Waiting
372d60850963: Waiting
a0e501ed6714: Waiting
353e14e5cc47: Waiting
091eb8249475: Verifying Checksum
091eb8249475: Download complete
7cd785773db4: Verifying Checksum
7cd785773db4: Download complete
7cd785773db4: Pull complete
aee83dd4a388: Verifying Checksum
091eb8249475: Pull complete
255774e0027b: Verifying Checksum
255774e0027b: Download complete
ec629d786f9c: Verifying Checksum
ec629d786f9c: Download complete
255774e0027b: Pull complete
a0e501ed6714: Verifying Checksum
a0e501ed6714: Download complete
372d60850963: Download complete
353e14e5cc47: Retrying in 5 seconds
353e14e5cc47: Retrying in 4 seconds
353e14e5cc47: Retrying in 3 seconds
353e14e5cc47: Retrying in 2 seconds
353e14e5cc47: Retrying in 1 second
353e14e5cc47: Verifying Checksum
353e14e5cc47: Download complete
```

TASK 3 MINIKUBE

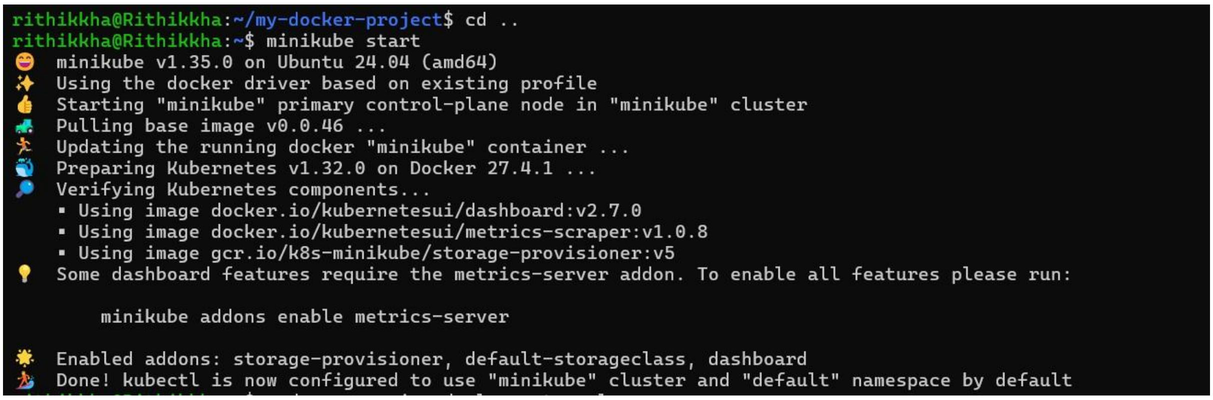


```
Step 2/7 : WORKDIR /app
 ---> Running in f47fb44b1530
 ---> Removed intermediate container f47fb44b1530
 ---> fe39a4c2447c
Step 3/7 : COPY package.json ./
 ---> 348bc4c3a1b9
Step 4/7 : RUN npm install
 ---> Running in 46e80ad0fc9c

up to date, audited 1 package in 747ms

found 0 vulnerabilities
 ---> Removed intermediate container 46e80ad0fc9c
 ---> 557ce94b04ab
Step 5/7 : COPY . .
 ---> 5630981e150e
Step 6/7 : EXPOSE 3000
 ---> Running in bfdd357e3377
 ---> Removed intermediate container bfdd357e3377
 ---> 1084bb8a4bd9
Step 7/7 : CMD ["npm", "start"]
 ---> Running in f2f8d0646136
 ---> Removed intermediate container f2f8d0646136
 ---> 9aa1cc7e4be7
Successfully built 9aa1cc7e4be7
Successfully tagged rithumekala/devops:latest
CONTAINER ID   IMAGE                               COMMAND                  CREATED          STATUS          PORTS
                                                                                   NAMES
ba6514ba4bdc   9e4f55d0d16d                        "/docker-entrypoint.…"   53 minutes ago   Up 53 minutes   0.0.0.0:3001->80/tcp, [::]:3001->80/tcp
                                                                                   jenkins-docker-container
e7c14bd98a3b   gcr.io/k8s-minikube/kicbase:v0.0.46   "/usr/local/bin/entr…"   3 hours ago      Up 3 hours      127.0.0.1:32768->22/tcp, 127.0.0.1:32769->237
6/tcp, 127.0.0.1:32770->5000/tcp, 127.0.0.1:32771->8443/tcp, 127.0.0.1:32772->32443/tcp   minikube
```

MINIKUBE START

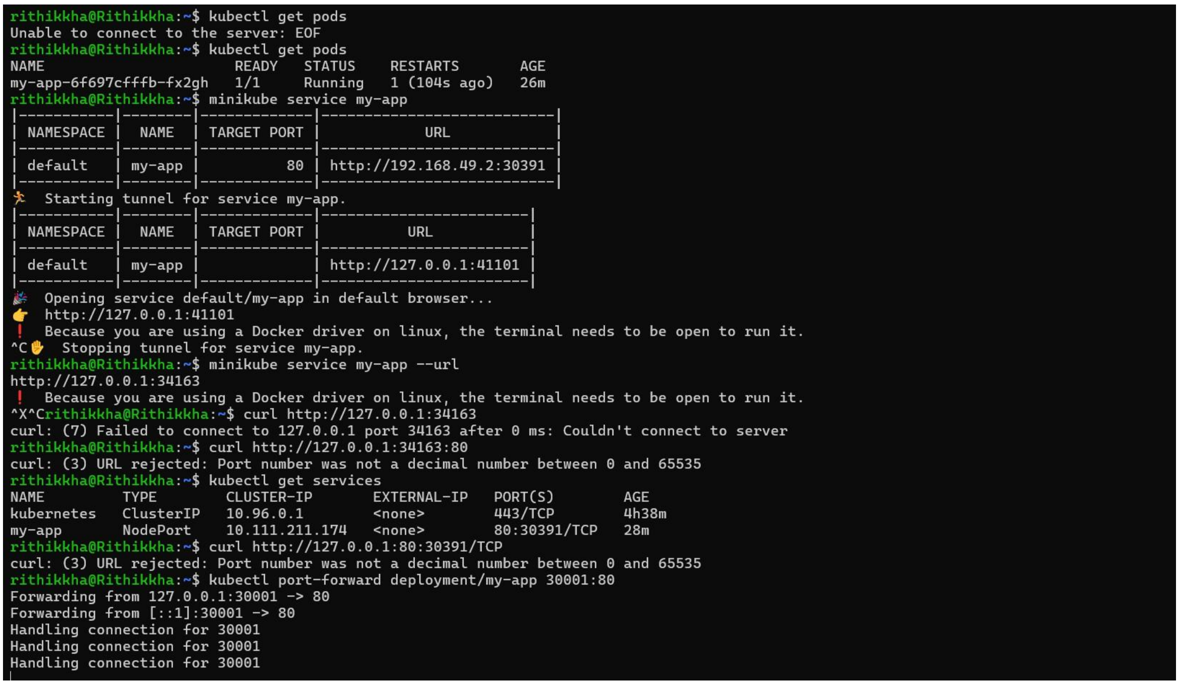


```
rithikkha@Rithikkha:~/my-docker-project$ cd ..
rithikkha@Rithikkha:~$ minikube start
😄  minikube v1.35.0 on Ubuntu 24.04 (amd64)
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.46 ...
🏃  Updating the running docker "minikube" container ...
🐳  Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
🔎  Verifying Kubernetes components...
    ▪ Using image docker.io/kubernetesui/dashboard:v2.7.0
    ▪ Using image docker.io/kubernetesui/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, default-storageclass, dashboard
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

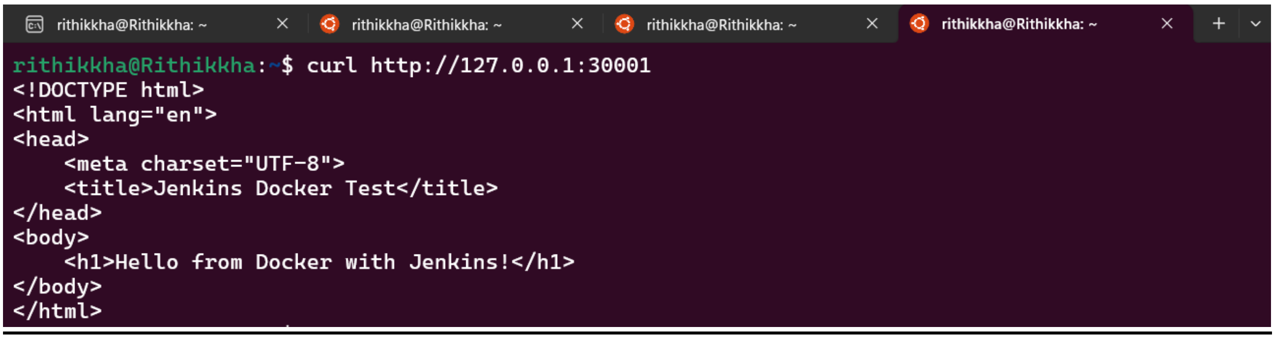
MINIKUBE SERVICE MY-APP



```
rithikkha@Rithikkha:~$ kubectl get pods
Unable to connect to the server: EOF
rithikkha@Rithikkha:~$ kubectl get pods
NAME                      READY   STATUS    RESTARTS       AGE
my-app-6f697cfffb-fx2gh   1/1     Running   1 (104s ago)   26m
rithikkha@Rithikkha:~$ minikube service my-app
|-----------|--------|-------------|---------------------------|
| NAMESPACE |  NAME  | TARGET PORT |            URL            |
|-----------|--------|-------------|---------------------------|
| default   | my-app |          80 | http://192.168.49.2:30391 |
|-----------|--------|-------------|---------------------------|
🏃  Starting tunnel for service my-app.
|-----------|--------|-------------|------------------------|
| NAMESPACE |  NAME  | TARGET PORT |          URL           |
|-----------|--------|-------------|------------------------|
| default   | my-app |             | http://127.0.0.1:41101 |
|-----------|--------|-------------|------------------------|
🎉  Opening service default/my-app in default browser...
👉  http://127.0.0.1:41101
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
^C✋  Stopping tunnel for service my-app.
rithikkha@Rithikkha:~$ minikube service my-app --url
http://127.0.0.1:34163
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
^X^Crithikkha@Rithikkha:~$ curl http://127.0.0.1:34163
curl: (7) Failed to connect to 127.0.0.1 port 34163 after 0 ms: Couldn't connect to server
rithikkha@Rithikkha:~$ curl http://127.0.0.1:34163:80
curl: (3) URL rejected: Port number was not a decimal number between 0 and 65535
rithikkha@Rithikkha:~$ kubectl get services
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP        4h38m
my-app       NodePort    10.111.211.174   <none>        80:30391/TCP   28m
rithikkha@Rithikkha:~$ curl http://127.0.0.1:80:30391/TCP
curl: (3) URL rejected: Port number was not a decimal number between 0 and 65535
rithikkha@Rithikkha:~$ kubectl port-forward deployment/my-app 30001:80
Forwarding from 127.0.0.1:30001 -> 80
Forwarding from [::1]:30001 -> 80
Handling connection for 30001
Handling connection for 30001
Handling connection for 30001
```

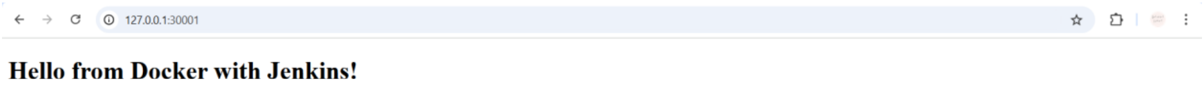
TASK 3 MINIKUBE

CURL LOCAL HOST



```
rithikkha@Rithikkha:~$ curl http://127.0.0.1:30001
<!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>
```

OUTPUT



Hello from Docker with Jenkins!