

DEPLOY A MULTI-TIER WEB APPLICATION ON KUBERNETES

The File Structure

```
master@master-vm: ~/Desktop/k8s-project/nginx
master@master-vm:~$ cd Desktop/
master@master-vm:~/Desktop$ mkdir k8s-project
master@master-vm:~/Desktop$ cd k8s-project
master@master-vm:~/Desktop/k8s-project$ mkdir mysql
master@master-vm:~/Desktop/k8s-project$ cd mysql
master@master-vm:~/Desktop/k8s-project/mysql$ nano mysql-deployment.yaml
master@master-vm:~/Desktop/k8s-project/mysql$ nano mysql-secret.yaml
master@master-vm:~/Desktop/k8s-project/mysql$ nano mysql-pv.yaml
master@master-vm:~/Desktop/k8s-project/mysql$ cd ..
master@master-vm:~/Desktop/k8s-project$ mkdir flask
master@master-vm:~/Desktop/k8s-project$ cd flask
master@master-vm:~/Desktop/k8s-project/flask$ nano Dockerfile
master@master-vm:~/Desktop/k8s-project/flask$ nano app.py
master@master-vm:~/Desktop/k8s-project/flask$ nano requirements.txt
master@master-vm:~/Desktop/k8s-project/flask$ nano flask-deployment.yaml
master@master-vm:~/Desktop/k8s-project/flask$ nano flask-service.yaml
master@master-vm:~/Desktop/k8s-project/flask$ cd ..
master@master-vm:~/Desktop/k8s-project$ mkdir nginx
master@master-vm:~/Desktop/k8s-project$ cd nginx
master@master-vm:~/Desktop/k8s-project/nginx$ nano nginx-configmap.yaml
master@master-vm:~/Desktop/k8s-project/nginx$ nano nginx-deployment.yaml
master@master-vm:~/Desktop/k8s-project/nginx$ nano nginx-service.yaml
master@master-vm:~/Desktop/k8s-project/nginx$
```

Persistent Volume (mysql-pv.yaml) file

```
GNU nano 4.8 mysql-pv.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mysql-pv
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/mnt/data"
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

StatefulSet (mysql-deployment.yaml) file

```
GNU nano 4.8 mysql-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:5.7
          env:
            - name: MYSQL_ROOT_PASSWORD
              value: "rootpassword"
            - name: MYSQL_DATABASE
              value: "mydb"
            - name: MYSQL_USER
              value: "user"
            - name: MYSQL_PASSWORD
              value: "password"
          ports:
            - containerPort: 3306
          volumeMounts:
            - name: mysql-storage
              mountPath: /var/lib/mysql
```

Read 45 lines

Activate Windows
Go to Settings to activate Windows.

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo Mark Text
Exit Read File Replace Paste Text To Spell Go To Line Redo M-A M-B M-C M-D

Secret (mysql-secret.yaml) file

```
master@master-vm: ~/Desktop/k8s-project/mysql
GNU nano 4.8 mysql-secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: mysql-secret
type: Opaque
data:
  mysql-root-password: cGFzc3dvcmQ= # Base64 encoded "password"
```

Flask app.py file

```
master@master-vm: ~/Desktop/k8s-project/flask
GNU nano 4.8 app.py
from flask import Flask, jsonify
import mysql.connector
import os
import time

app = Flask(__name__)

# Load MySQL credentials from environment variables
MYSQL_HOST = os.getenv("MYSQL_HOST", "mysql.default.svc.cluster.local")
MYSQL_USER = os.getenv("MYSQL_USER", "user")
MYSQL_PASSWORD = os.getenv("MYSQL_PASSWORD", "password")
MYSQL_DATABASE = os.getenv("MYSQL_DATABASE", "mydb")

def get_db_connection():
    """Establish a connection to MySQL with retry mechanism"""
    for _ in range(5): # Retry up to 5 times
        try:
            conn = mysql.connector.connect(
                host=MYSQL_HOST,
                user=MYSQL_USER,
                password=MYSQL_PASSWORD,
                database=MYSQL_DATABASE
            )
            print("✅ Connected to MySQL successfully!")
            return conn
        except mysql.connector.Error as err:
            print(f"❌ MySQL Connection Error: {err}")
            time.sleep(2)
    return None

@app.route("/")
```

Flask- Dockerfile

```
master@master-vm: ~/Desktop/k8s-project/flask
GNU nano 4.8 Dockerfile
FROM python:3.8
WORKDIR /app
COPY app.py .
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
RUN apt update && apt install -y mysql-client
CMD ["python", "app.py"]
```

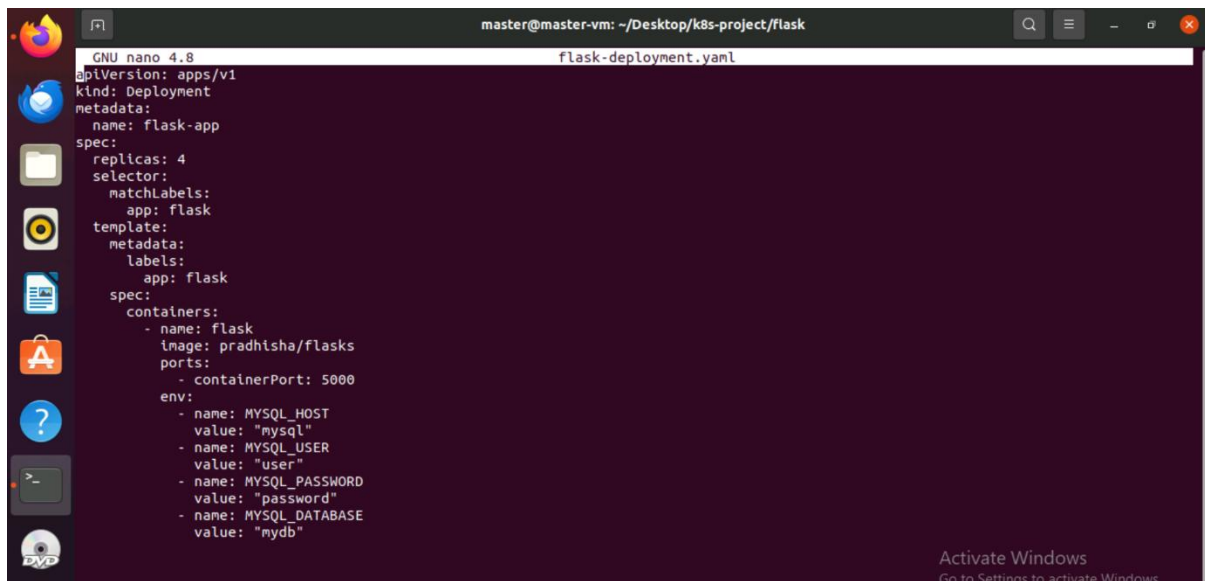
Flask- Requirements.txt file

```
master@master-vm: ~/Desktop/k8s-project/flask
GNU nano 4.8 requirements.txt
flask
mysql-connector-python
```

Service (flask-service.yaml)

```
master@master-vm: ~/Desktop/k8s-project/flask
GNU nano 4.8 flask-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: flask-service
spec:
  type: NodePort
  selector:
    app: flask
  ports:
    - protocol: TCP
      port: 5000
      targetPort: 5000
      nodePort: 30010
```

Deployment (flask-deployment.yaml)

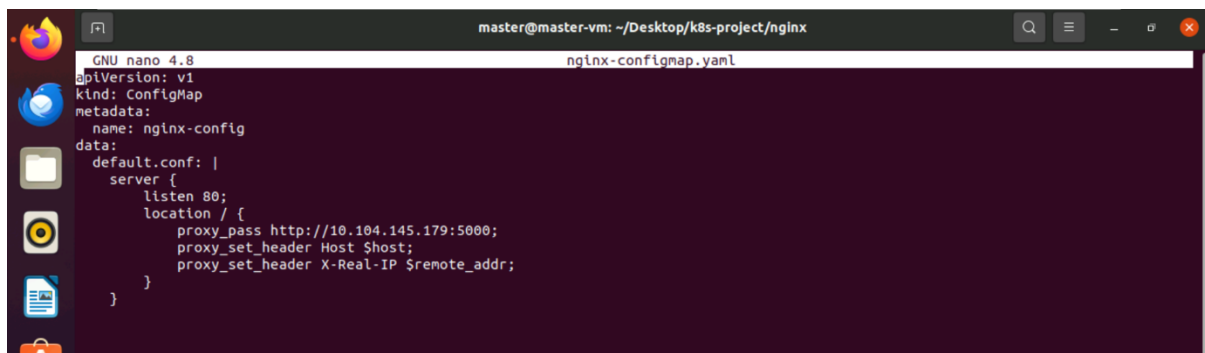


The screenshot shows a terminal window titled 'master@master-vm: ~/Desktop/k8s-project/flask' with the nano 4.8 editor open to 'flask-deployment.yaml'. The file content is as follows:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-app
spec:
  replicas: 4
  selector:
    matchLabels:
      app: flask
  template:
    metadata:
      labels:
        app: flask
    spec:
      containers:
      - name: flask
        image: pradhisha/flasks
        ports:
        - containerPort: 5000
        env:
        - name: MYSQL_HOST
          value: "mysql"
        - name: MYSQL_USER
          value: "user"
        - name: MYSQL_PASSWORD
          value: "password"
        - name: MYSQL_DATABASE
          value: "mydb"
```

An 'Activate Windows' watermark is visible in the bottom right corner.

ConfigMap (nginx-configmap.yaml) file

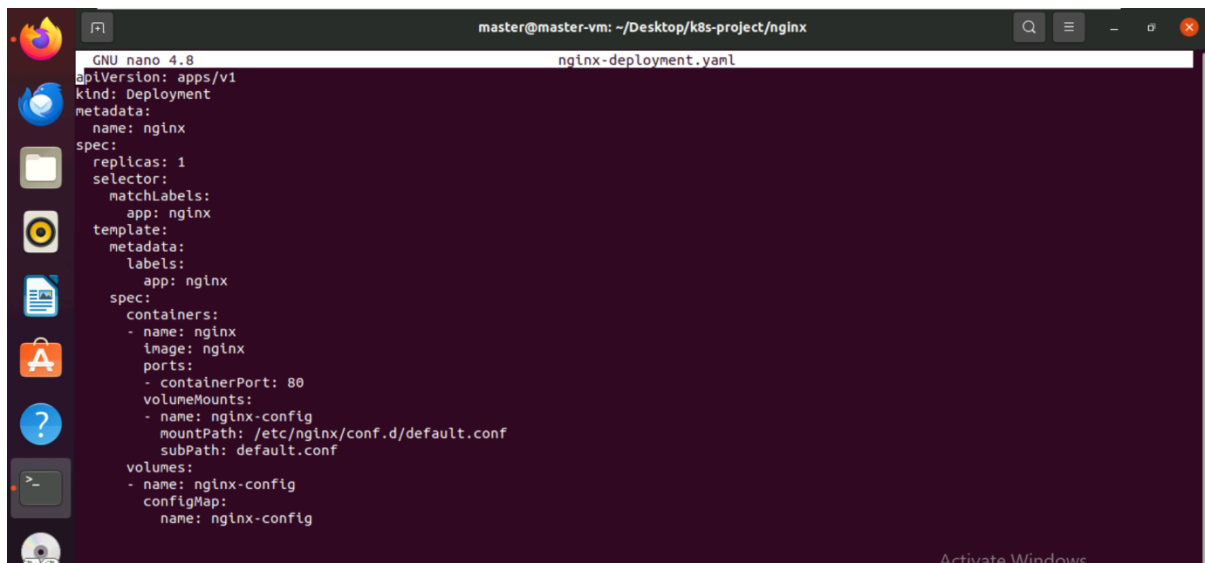


The screenshot shows a terminal window titled 'master@master-vm: ~/Desktop/k8s-project/nginx' with the nano 4.8 editor open to 'nginx-configmap.yaml'. The file content is as follows:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: nginx-config
data:
  default.conf: |
    server {
      listen 80;
      location / {
        proxy_pass http://10.104.145.179:5000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
      }
    }
```

An 'Activate Windows' watermark is visible in the bottom right corner.

Deployment (nginx-deployment.yaml) file

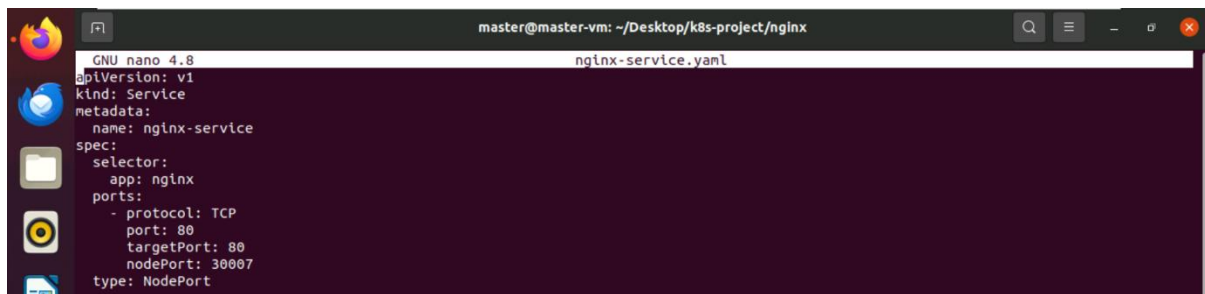


The screenshot shows a terminal window titled 'master@master-vm: ~/Desktop/k8s-project/nginx' with the nano 4.8 editor open to 'nginx-deployment.yaml'. The file content is as follows:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
        volumeMounts:
        - name: nginx-config
          mountPath: /etc/nginx/conf.d/default.conf
          subPath: default.conf
      volumes:
      - name: nginx-config
        configMap:
          name: nginx-config
```

An 'Activate Windows' watermark is visible in the bottom right corner.

Service (nginx-service.yaml) file



```
master@master-vm: ~/Desktop/k8s-project/nginx
GNU nano 4.8 nginx-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30007
  type: NodePort
```

1. Build and Push Docker Image

1. Navigate to the Flask application directory:

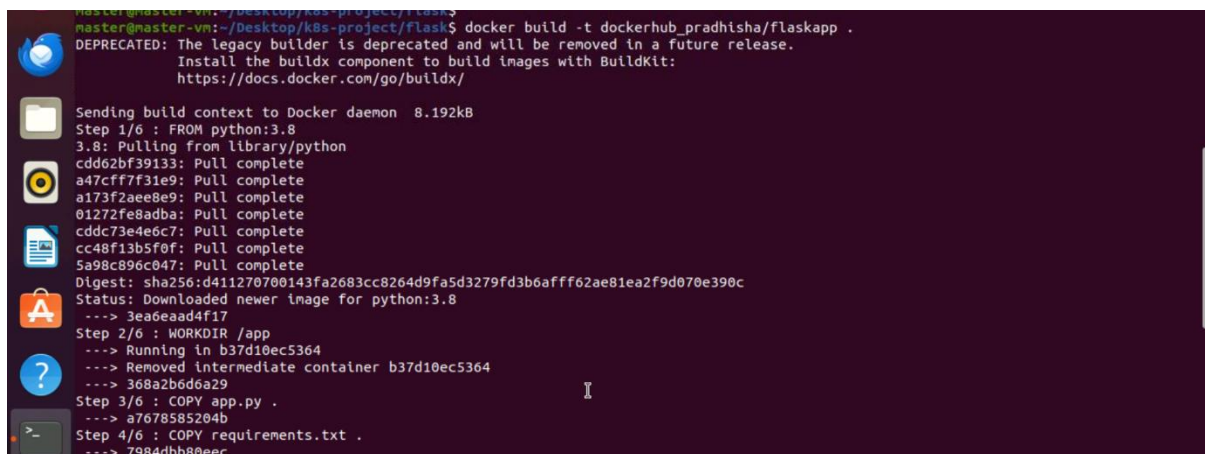
- `cd flask`



```
master@master-vm: ~/Desktop/k8s-project$ cd flask
```

2. Build the Docker image:

- `docker build -t dockerhub_username/flaskapp .`




```
master@master-vm: ~/Desktop/k8s-project/flask$ docker build -t dockerhub_pradhisha/flaskapp .
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  8.192kB
Step 1/6 : FROM python:3.8
3.8: Pulling from library/python
cdd62bf39133: Pull complete
a47c7f7f31e9: Pull complete
a173f2aee8e9: Pull complete
01272fe8adba: Pull complete
cddc73e4e6c7: Pull complete
cc48f13b5f0f: Pull complete
5a98c896c047: Pull complete
Digest: sha256:d411270700143fa2683cc8264d9fa5d3279fd3b6afff62ae81ea2f9d070e390c
Status: Downloaded newer image for python:3.8
--> 3ea6eaad4f17
Step 2/6 : WORKDIR /app
--> Running in b37d10ec5364
--> Removed intermediate container b37d10ec5364
--> 368a2b6d6a29
Step 3/6 : COPY app.py .
--> a7678585204b
Step 4/6 : COPY requirements.txt .
--> 7984dbb80eec
```

1. Push the image to Docker Hub:

- `docker push dockerhub_username/flaskapp`

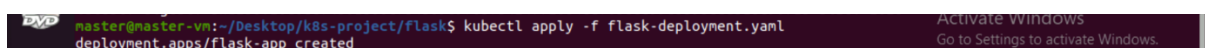


```
master@master-vm: ~/Desktop/k8s-project/flask$ docker push pradhisha/flaskapp
Using default tag: latest
The push refers to repository [docker.io/pradhisha/flaskapp]
8767dfc0d204: Pushed
b6ea8b3da943: Pushed
09addba15cab: Pushed
322fd7415106: Pushed
32ee710ca3c7: Pushed
1767e4d52b5a: Pushed
45b98afd69b3: Pushed
2bce433c3a29: Pushed
f91dc7a486d9: Pushed
3e14a6961052: Pushed
d50132f2fe78: Pushed
latest: digest: sha256:20c6ae74501806c754509e9f63fb4247f2084ea763dee99d3b93aaab18c90715 size: 2628
```

2. Deploy the Flask Application

4. Apply the Flask Deployment manifest:

- `kubectl apply -f flask-deployment.yaml`



```
master@master-vm: ~/Desktop/k8s-project/flask$ kubectl apply -f flask-deployment.yaml
deployment.apps/flask-app created
```

5. Apply the Flask Service manifest:

- `kubectl apply -f flask-service.yaml`

```
master@master-vn:~/Desktop/k8s-project/flask$ kubectl apply -f flask-service.yaml
service/flask-service created
master@master-vn:~/Desktop/k8s-project/flask$
```

3. Deploy the MySQL Database

6. Navigate to the MySQL directory:

- `cd .. && cd mysql`

7. Apply the MySQL Deployment manifest:

- `kubectl apply -f mysql-deployment.yaml`

```
master@master-vn:~/Desktop/k8s-project/mysql$ kubectl apply -f mysql-deployment.yaml
deployment.apps/mysql unchanged
service/mysql unchanged
```

8. Apply the MySQL Persistent Volume manifest:

- `kubectl apply -f mysql-pv.yaml`

```
master@master-vn:~/Desktop/k8s-project/mysql$ kubectl apply -f mysql-pv.yaml
persistentvolume/mysql-pv unchanged
persistentvolumeclaim/mysql-pvc unchanged
```

9. Apply the MySQL Secret manifest:

- `kubectl apply -f mysql-secret.yaml`

```
master@master-vn:~/Desktop/k8s-project/mysql$ kubectl apply -f mysql-secret.yaml
secret/mysql-secret unchanged
master@master-vn:~/Desktop/k8s-project/mysql$ cd ..
```

4. Deploy the Nginx Service

10. Navigate to the Nginx directory:

- `cd .. && cd nginx`

11. Apply the Nginx ConfigMap manifest:

- `kubectl apply -f nginx-configmap.yaml`

```
master@master-vn:~/Desktop/k8s-project/nginx$ kubectl apply -f nginx-configmap.yaml
configmap/nginx-config unchanged
```

12. Apply the Nginx Deployment manifest:

- `kubectl apply -f nginx-deployment.yaml`

```
master@master-vn:~/Desktop/k8s-project/nginx$ kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx unchanged
```

13. Apply the Nginx Service manifest:

- `kubectl apply -f nginx-service.yaml`

```
master@master-vn:~/Desktop/k8s-project/nginx$ kubectl apply -f nginx-service.yaml
service/nginx-service unchanged
```


5. Verify the Deployment

14. Check the status of all Kubernetes resources:

- `kubectl get all -o wide`

```
naster@naster-vm: ~/Desktop/k8s-project/nginx$ kubectl get all -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
pod/flask-app-5bdc6ddf8d-44bln	1/1	Running	0	9m57s	192.168.94.218	worker1-vm	<none>	<none>
pod/flask-app-5bdc6ddf8d-gfplj	1/1	Running	0	10m	192.168.94.217	worker1-vm	<none>	<none>
pod/flask-app-5bdc6ddf8d-r2bwt	1/1	Running	0	11m	192.168.94.215	worker1-vm	<none>	<none>
pod/flask-app-5bdc6ddf8d-xvvz5	1/1	Running	0	11m	192.168.94.216	worker1-vm	<none>	<none>
pod/mysql-66d468f74c-hks24	1/1	Running	0	22m	192.168.94.209	worker1-vm	<none>	<none>
pod/nginx-78b5b78b99-89vp8	1/1	Running	0	22m	192.168.94.208	worker1-vm	<none>	<none>

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	SELECTOR
service/flask-service	ClusterIP	10.96.105.212	<none>	5000/TCP	16h	app=flask
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	20d	<none>
service/mysql	ClusterIP	10.108.12.84	<none>	3306/TCP	15h	app=mysql
service/nginx-service	NodePort	10.96.88.25	<none>	80:30007/TCP	15h	app=nginx

NAME	READY	UP-TO-DATE	AVAILABLE	AGE	CONTAINERS	IMAGES	SELECTOR
deployment.apps/flask-app	4/4	4	4	16h	flask	pradhisha/flaskapp	app=flask
deployment.apps/mysql	1/1	1	1	15h	mysql	mysql:5.7	app=mysql
deployment.apps/nginx	1/1	1	1	15h	nginx	nginx	app=nginx

NAME	DESIRED	CURRENT	READY	AGE	CONTAINERS	IMAGES	SELECTOR
replicaset.apps/flask-app-55c765985d	0	0	0	16h	flask	pradhisha/flaskapp	app=flask,pod-template-ha
sh=55c765985d							
replicaset.apps/flask-app-5bdc6ddf8d	4	4	4	11m	flask	pradhisha/flaskapp	app=flask,pod-template-ha
sh=5bdc6ddf8d							
replicaset.apps/mysql-66d468f74c	1	1	1	15h	mysql	mysql:5.7	app=mysql,pod-template-ha
sh=66d468f74c							
replicaset.apps/nginx-78b5b78b99	1	1	1	15h	nginx	nginx	app=nginx,pod-template-ha
sh=78b5b78b99							

6. Access the Application

15. Retrieve the worker node IP:

- `kubectl get nodes -o wide`

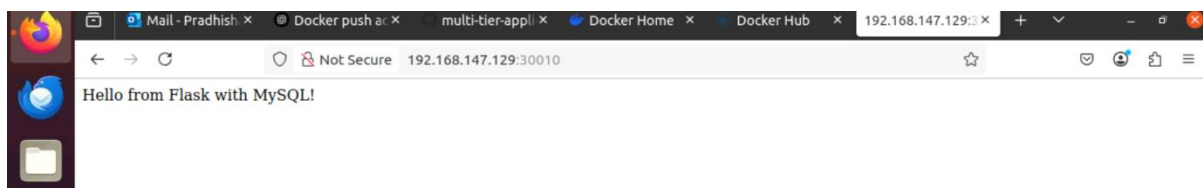
```
naster@naster-vm: ~/Desktop/k8s-project/nginx$ kubectl get nodes -o wide
```

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNEL-VERSION	CON
TAI-NER-RUNTIME									
master-vm	Ready	control-plane	20d	v1.28.15	192.168.147.128	<none>	Ubuntu 20.04.6 LTS	5.15.0-134-generic	con
tai-nerd://1.7.24									
worker1-vm	Ready	<none>	20d	v1.28.15	192.168.147.129	<none>	Ubuntu 20.04.6 LTS	5.15.0-134-generic	con
tai-nerd://1.7.24									
worker2-vm	Ready	<none>	20d	v1.28.15	192.168.147.130	<none>	Ubuntu 20.04.6 LTS	5.15.0-131-generic	con
tai-nerd://1.7.24									

```
naster@naster-vm: ~/Desktop/k8s-project/nginx$
```

16. Access the application in the browser:

<http://192.168.147.129:30010/>



7. Configure MySQL Database

17. Access the MySQL pod:

`kubectl exec -it mysql-0 -- mysql -u root -p`

18. Create and configure the database:

```
master@master-vn:~/Desktop/k8s-project/mysql$ kubectl exec -it mysql-66d468f74c-p4pqg -- mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.7.44 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE mydb;
ERROR 1007 (HY000): Can't create database 'mydb'; database exists
mysql>
mysql> USE mydb;
Database changed
mysql>
mysql> SHOW TABLES;
Empty set (0.00 sec)

mysql>
mysql> CREATE TABLE users (
->   -> id INT AUTO_INCREMENT PRIMARY KEY,
->   -> name VARCHAR(100),
->   -> email VARCHAR(100)
->   -> );
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql> INSERT INTO users (name, email) VALUES ('Alice', 'alice@example.com');
Query OK, 1 row affected (0.02 sec)

mysql>
mysql> INSERT INTO users (name, email) VALUES ('Bob', 'bob@example.com');
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> SELECT * FROM users;
+-----+-----+-----+
| id | name | email |
+-----+-----+-----+
| 1 | Alice | alice@example.com |
| 2 | Bob | bob@example.com |
+-----+-----+-----+
2 rows in set (0.00 sec)

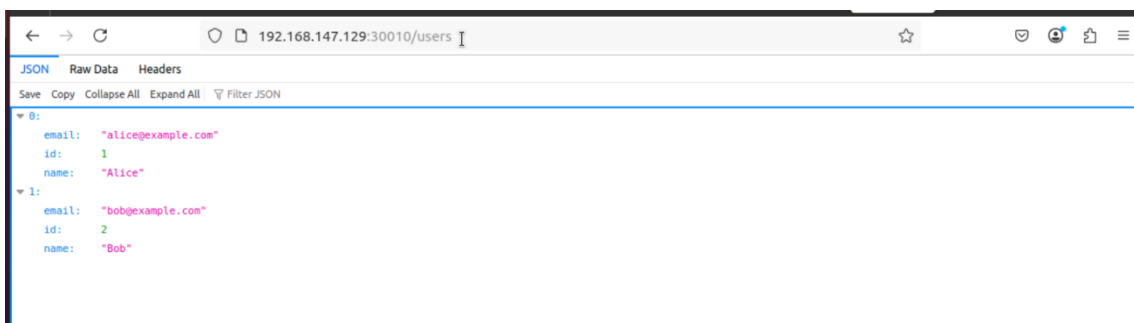
mysql>
mysql> GRANT ALL PRIVILEGES ON mydb.* TO 'user'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> FLUSH PRIVILEGES; exit
Query OK, 0 rows affected (0.00 sec)

-> ^C
mysql> exit
Bye
```

19. Access the users API endpoint:

<http://192.168.147.129:30010/users>



```
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON

0:
  email: "alice@example.com"
  id: 1
  name: "Alice"
1:
  email: "bob@example.com"
  id: 2
  name: "Bob"
```