DEPLOY A MULTI-TIER WEB APPLICATION ON KUBERNETES

Step 1: Start Minikube

```
master@master-vm:~$ minikube start
😄 minikube v1.35.0 on Ubuntu 20.04
▪ MINIKUBE_ACTIVE_DOCKERD=minikube
✨ Using the docker driver based on existing profile
👍 Starting "minikube" primary control-plane node in "minikube" cluster
🚜 Pulling base image v0.0.46 ...
📌 Noticed you have an activated docker-env on docker driver in this terminal:
❗ Please re-eval your docker-env, To ensure your environment variables have updated ports:

    'minikube -p minikube docker-env'


🏃 Updating the running docker "minikube" container ...
🐳 Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
🔎 Verifying Kubernetes components...
▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: storage-provisioner, default-storageclass

❗ /usr/bin/kubectl is version 1.29.15, which may have incompatibilities with Kubernetes 1.32.0.
▪ Want kubectl v1.32.0? Try 'minikube kubectl -- get pods -A'
🏄 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Step 2: Run cluster info

```
master@master-vm:~$ kubectl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

Step 3: Get Nodes

```
master@master-vm:~$ kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   18m   v1.32.0
```

Step 4: Create and run mysql-pv.yaml file

```
master@master-vm:~$ nano mysql-pv.yaml
master@master-vm:~$ kubectl apply -f mysql-pv.yaml
persistentvolume/mysql-pv created
```

Step 5: Create and run mysql-secret.yaml file

```
master@master-vm:~$ nano mysql-secret.yaml
master@master-vm:~$ kubectl apply -f mysql-secret.yaml
```

Step 6: Get Pods

```
master@master-vm:~$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
mysql-0   1/1     Running   0          4m59s
master@master-vm:~$ nano app.py
master@master-vm:~$ nano Dockerfile
```

Step 7: Create a app.py file

```
master@master-vm:~$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
mysql-0   1/1     Running   0          4m59s
master@master-vm:~$ nano app.py
master@master-vm:~$ nano Dockerfile
```

Step 8: Create a Dockerfile

```
master@master-vm:~$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
mysql-0   1/1     Running   0          4m59s
master@master-vm:~$ nano app.py
master@master-vm:~$ nano Dockerfile
```

Step 9: Build and Push the flask-app using docker credentials

```
master@master-vm:~$ docker build -t ankitamohanty1509/flask-app .
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  1.164GB
Step 1/6 : FROM python:3.11
3.11: Pulling from library/python
155ad54a8b28: Pull complete
8031108f3cda: Pull complete
1d281e50d3e4: Pull complete
447713e77b4f: Pull complete
441749a24fb5: Pull complete
ae604eab20d6: Pull complete
672d84e58157: Pull complete
Digest: sha256:68a8863d0625f42d47e0684f33ca02f19d6094ef859a8af237aaf645195ed477
Status: Downloaded newer image for python:3.11
 ---> 78553a4d82cb
Step 2/6 : WORKDIR /app
 ---> Running in c4d70a7b5f67
 ---> Removed intermediate container c4d70a7b5f67
 ---> 9ccc114e61ec
Step 3/6 : COPY . /app
 ---> 1b49eb393907
Step 4/6 : RUN pip install flask
 ---> Running in 02b40cdd0a0b
Collecting flask
  Downloading flask-3.1.0-py3-none-any.whl.metadata (2.7 kB)
Collecting Werkzeug>=3.1 (from flask)
  Downloading werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Collecting Jinja2>=3.1.2 (from flask)
  Downloading jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
Collecting itsdangerous>=2.2 (from flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting click>=8.1.3 (from flask)
  Downloading click-8.1.8-py3-none-any.whl.metadata (2.3 kB)
```

```
Downloading MarkupSafe-3.0.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (23 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, blinker, Werkzeug, Jinja2, flask
Successfully installed Jinja2-3.1.6 MarkupSafe-3.0.2 Werkzeug-3.1.3 blinker-1.9.0 click-8.1.8 flask-3.1.0 itsdangerous-2.2.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 24.0 -> 25.0.1
[notice] To update, run: pip install --upgrade pip
 ---> Removed intermediate container 02b40cdd0a0b
 ---> 23acb1489981
Step 5/6 : EXPOSE 5000
 ---> Running in 1ead304bd0cd
 ---> Removed intermediate container 1ead304bd0cd
 ---> fdf4c8c6b1c0
Step 6/6 : CMD ["python", "app.py"]
 ---> Running in 715f15349ae7
 ---> Removed intermediate container 715f15349ae7
 ---> 2a49cee978a7
Successfully built 2a49cee978a7
Successfully tagged ankitamohanty1509/flask-app:latest
master@master-vm:~$ docker push ankitamohanty1509/flask-app
Using default tag: latest
The push refers to repository [docker.io/ankitamohanty1509/flask-app]
5233cae342a0: Pushed
d7cde034ff1c: Pushed
9bcfcb81bcaa: Pushed
b723da6e1cf4: Mounted from library/python
7af6b2a8a1a8: Mounted from library/python
71030c5d3283: Mounted from library/python
4b017a36fd9c: Mounted from library/python
20a9b386e10e: Mounted from library/python
f8217d7865d2: Mounted from library/python
01c9a2a5f237: Mounted from library/python
latest: digest: sha256:2b94a9b1c0175b57c34945dd6b4e8e200b81cd14ce92d9af86560c3e6ee8f57c size: 2426
master@master-vm:~$
```

Step 10: Create and run flask-deployment.yaml file

```
master@master-vm:~$ nano flask-deployment.yaml
master@master-vm:~$ kubectl apply -f flask-deployment.yaml
deployment.apps/flask-app configured
```

Step 11: Get pods

```
master@master-vm:~$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
flask-app-7c7b6d9476-2gj5c    1/1     Running   0          6m15s
flask-app-7c7b6d9476-x8kht    1/1     Running   0          6m46s
mysql-0                       1/1     Running   0          40m
```

Step 12: Create and run nginx-deployment.yaml file

```
master@master-vm:~$ nano nginx-deployment.yaml
master@master-vm:~$ kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx created
```

Step 13: Create and run nginx-service.yaml

```
master@master-vm:~$ nano nginx-service.yaml
master@master-vm:~$ kubectl apply -f nginx-service.yaml
service/nginx-service created
```

Step 14: Get pods

```
master@master-vm:~$ kubectl get pods
NAME                        READY   STATUS    RESTARTS        AGE
flask-app-7c7b6d9476-2gj5c  1/1     Running   1 (7m8s ago)    15h
flask-app-7c7b6d9476-x8kht  1/1     Running   1               15h
mysql-0                     1/1     Running   1 (7m14s ago)   16h
nginx-86c57bc6b8-7ffgt      1/1     Running   0               111s
```

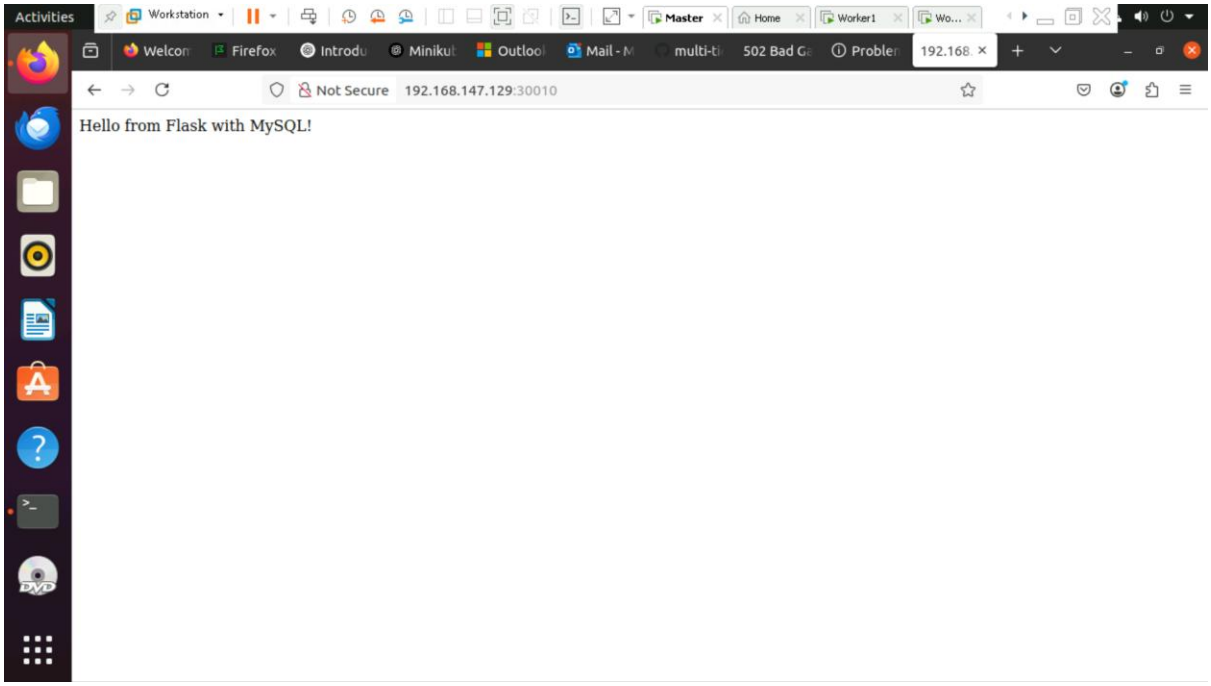
Step 15: Get svc

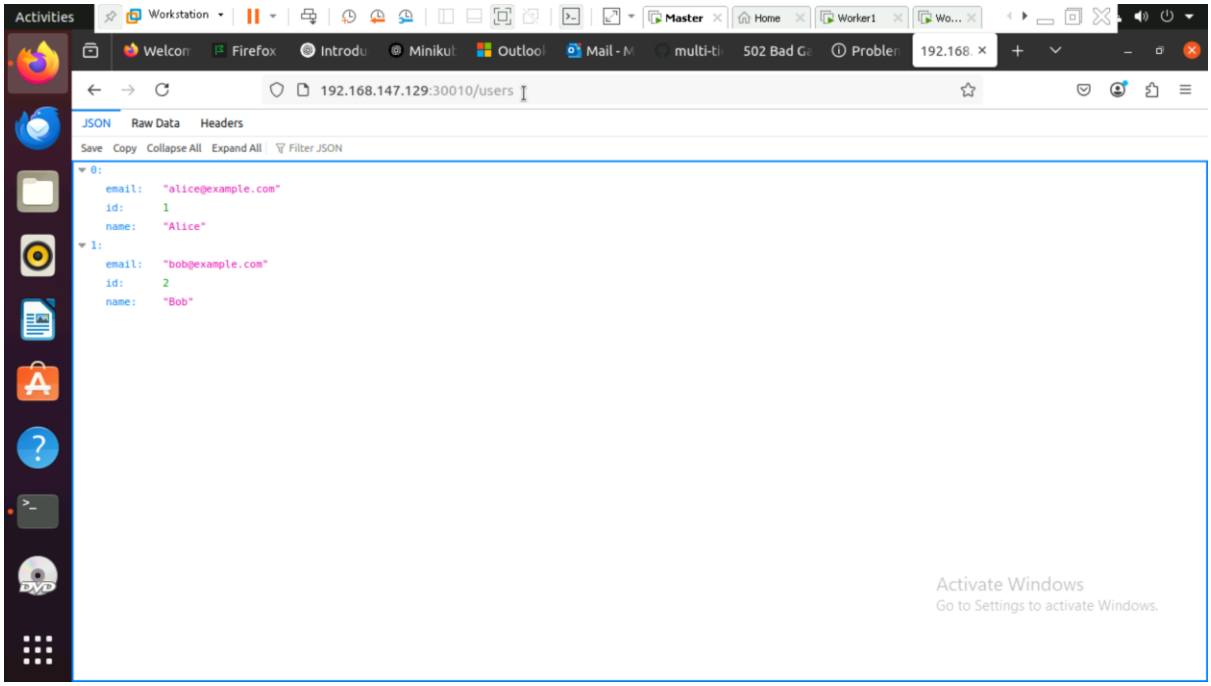
```
master@master-vm:~$ kubectl get svc
NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes      ClusterIP   10.96.0.1      <none>        443/TCP        16h
nginx-service   NodePort    10.99.215.77   <none>        80:30007/TCP   32s
```

Step 16: Run minikube --url

```
master@master-vm:~$ minikube service nginx-service --url
http://192.168.49.2:30007
master@master-vm:~$
```

Step 17: Run the command in the browser




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"
Activate Windows
Go to Settings to activate Windows.