Thomas Noone – CISC 5550 – Homework 4 Report

1) **Local Development and Deploying the API**

Firstly, I made all the necessary local changes to split up the flask application into two:

todolist.py contains the web application

todolist_api.py contains the API service (and must sit next to the .db)

I ran .sh commands similar to the Homework 3 script. I created a new project and VM to run the API Service:

```
thoma@LAPTOP-NPVECTV0 MINGW64 ~/OneDrive/Documents/CISC 5500/homework-four
$ gcloud services enable compute.googleapis.com
Operation "operations/acf.p2-1033111204186-79e2f45e-7594-475e-ae00-abf844f616bf" finished successfully.

thoma@LAPTOP-NPVECTV0 MINGW64 ~/OneDrive/Documents/CISC 5500/homework-four
$ INSTANCE_NAME=todo-list-hw4-vm

thoma@LAPTOP-NPVECTV0 MINGW64 ~/OneDrive/Documents/CISC 5500/homework-four
$ ZONE=us-central1-a

thoma@LAPTOP-NPVECTV0 MINGW64 ~/OneDrive/Documents/CISC 5500/homework-four
$ IMAGE_FAMILY=ubuntu-2004-lts

thoma@LAPTOP-NPVECTV0 MINGW64 ~/OneDrive/Documents/CISC 5500/homework-four
$ IMAGE_PROJECT=ubuntu-os-cloud

thoma@LAPTOP-NPVECTV0 MINGW64 ~/OneDrive/Documents/CISC 5500/homework-four
$ MACHINE_TYPE=e2-medium

thoma@LAPTOP-NPVECTV0 MINGW64 ~/OneDrive/Documents/CISC 5500/homework-four
$ USER=$(whoami)
```

```
thoma@LAPTOP-NPVECTV0 MINGW64 ~/OneDrive/Documents/CISC 5500/homework-four
$ gcloud compute instances create $INSTANCE_NAME --zone=$ZONE --image-family=$IMAGE_FAMILY --image-project=$IMAGE_PROJECT --m
achine-type=$MACHINE_TYPE
Created [https://www.googleapis.com/compute/v1/projects/cisc5550-homework4/zones/us-central1-a/instances/todo-list-hw4-vm].
NAME             ZONE           MACHINE_TYPE  PREEMPTIBLE  INTERNAL_IP  EXTERNAL_IP    STATUS
todo-list-hw4-vm  us-central1-a  e2-medium                  10.128.0.2   34.27.234.150  RUNNING

thoma@LAPTOP-NPVECTV0 MINGW64 ~/OneDrive/Documents/CISC 5500/homework-four
$ EXTERNAL_IP=$(gcloud compute instances describe $INSTANCE_NAME --zone=$ZONE --format='get(networkInterfaces[0].accessConfig
s[0].natIP)')

thoma@LAPTOP-NPVECTV0 MINGW64 ~/OneDrive/Documents/CISC 5500/homework-four
$ echo $EXTERNAL_IP
34.27.234.150
```

```
thoma@LAPTOP-NPVECTV0 MINGW64 ~/OneDrive/Documents/CISC 5500/homework-four
$ gcloud compute ssh $INSTANCE_NAME --zone=$ZONE --command "
sudo mkdir -p /home/flask-app &&
sudo mkdir -p /home/flask-app/api &&
sudo chown $USER:$USER /home/flask-app &&
sudo chown $USER:$USER /home/flask-app/api &&
sudo chmod 755 /home/flask-app &&
sudo chmod 755 /home/flask-app/api
"

thoma@LAPTOP-NPVECTV0 MINGW64 ~/OneDrive/Documents/CISC 5500/homework-four
$ gcloud compute scp api/todolist_api.py api/todolist.db $INSTANCE_NAME:/home/flask-app/api/ --zone=$ZONE
todolist_api.py           | 1 kB |   1.9 kB/s | ETA: 00:00:00 | 100%
todolist.db               | 12 kB |  12.0 kB/s | ETA: 00:00:00 | 100%

thoma@LAPTOP-NPVECTV0 MINGW64 ~/OneDrive/Documents/CISC 5500/homework-four
$ gcloud compute firewall-rules create allow-port-5000-http --direction=INGRESS --priority=1000 --network=default --action=AL
LOW --rules=tcp:5001 --source-ranges=0.0.0.0/0 --target-tags=http-server
Creating firewall...
..Created [https://www.googleapis.com/compute/v1/projects/cisc5550-homework4/global/firewalls/allow-port-5000-http].
done.
NAME                 NETWORK  DIRECTION  PRIORITY  ALLOW     DENY  DISABLED
allow-port-5000-http  default  INGRESS    1000      tcp:5001        False

thoma@LAPTOP-NPVECTV0 MINGW64 ~/OneDrive/Documents/CISC 5500/homework-four
$ gcloud compute instances add-tags $INSTANCE_NAME --zone=$ZONE --tags=http-server
Updated [https://www.googleapis.com/compute/v1/projects/cisc5550-homework4/zones/us-central1-a/instances/todo-l
ist-hw4-vm].

thoma@LAPTOP-NPVECTV0 MINGW64 ~/OneDrive/Documents/CISC 5500/homework-four
$ gcloud compute ssh $INSTANCE_NAME --zone=$ZONE --command "
> sudo apt update &&
> sudo apt install -y python3-pip &&
> pip3 install flask &&
> cd /home/flask-app/api &&
> nohup python3 todolist_api.py &
> "
```

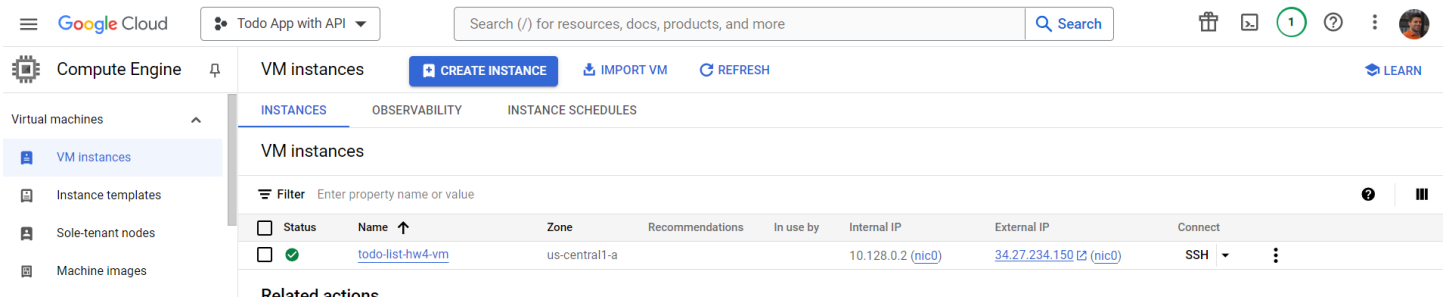
I adjusted the web application code locally to hit the VM's external IP:

```python
@app.route("/")
def show_list():
    resp = requests.get("http://34.27.234.150:5001/api/items")
    resp = resp.json()
    return render_template('index.html', todolist=resp)
```

Then when running the web application locally, it still works and I interact with the external API.

```
Press CTRL+C to quit
68.193.29.247 - - [28/Jul/2024 16:04:57] "GET /api/items HTTP/1.1" 200 -
68.193.29.247 - - [28/Jul/2024 16:05:24] "GET /api/items HTTP/1.1" 200 -
68.193.29.247 - - [28/Jul/2024 16:07:21] "PUT /api/mark HTTP/1.1" 200 -
68.193.29.247 - - [28/Jul/2024 16:07:21] "GET /api/items HTTP/1.1" 200 -
```

We can also see the VM running on Google Cloud Console:



2) Dockerizing the Web Application locally

I created a Dockerfile which included pip installing flask and requests, and then running `python todolist.py` as the primary `CMD`. See GitHub for the actual Dockerfile.

3) After creating a Docker account, I tagged and pushed the image:

```
Username: tnoone125
Password:

Login Succeeded
PS C:\Users\thoma\OneDrive\Documents\CISC 5500\homework-four\webapp> docker tag todolist_webapp:latest tnoone125/todolist_webapp:latest
PS C:\Users\thoma\OneDrive\Documents\CISC 5500\homework-four\webapp> docker push tnoone125/todolist_webapp:latest
The push refers to repository [docker.io/tnoone125/todolist_webapp]
c9f434ce8159: Pushed
```

4) Kubernetes

I installed kubectl and then created a deployment.yaml and service.yaml file (see the GitHub). Using gcloud and kubectl commands, I could create the cluster and apply the deployment.

`gcloud services enable container.googleapis.com`

was necessary first to enable Kubernetes for my project.

Here are the sequence of gcloud commands:

```
gcloud container clusters create todolist-cluster --num-nodes=1 --zone=us-central1-a
gcloud components install gke-gcloud-auth-plugin
gcloud container clusters get-credentials todolist-cluster --region=us-central1-a
kubectl get namespaces
kubectl get nodes
NAME                                                  STATUS   ROLES    AGE   VERSION
gke-todolist-cluster-default-pool-5efe6c92-9jbg   Ready    <none>   17m   v1.29.6-gke.1038001
kubectl apply -f .\deployment.yaml
kubectl apply -f .\service.yaml
kubectl get services
NAME              TYPE           CLUSTER-IP       EXTERNAL-IP    PORT(S)        AGE
kubernetes        ClusterIP      34.118.224.1     <none>         443/TCP        23m
todolist-webapp   LoadBalancer   34.118.235.111   35.224.13.20   80:30281/TCP   114s

gcloud compute firewall-rules create allow-http-5000 --allow tcp:5000
gcloud compute firewall-rules create allow-http-80 --allow tcp:80
gcloud compute firewall-rules create allow-https --allow tcp:443
```

Running Cluster:

```
kubeconfig entry generated for todolist-cluster.
NAME             LOCATION       MASTER_VERSION      MASTER_IP     MACHINE_TYPE  NODE_VERSION        NUM_NODES  STATUS
todolist-cluster  us-central1-a  1.29.6-gke.1038001  34.46.30.99  e2-medium     1.29.6-gke.1038001  1          RUNNING
```

And now I can access the web application via the external IP, port 80!