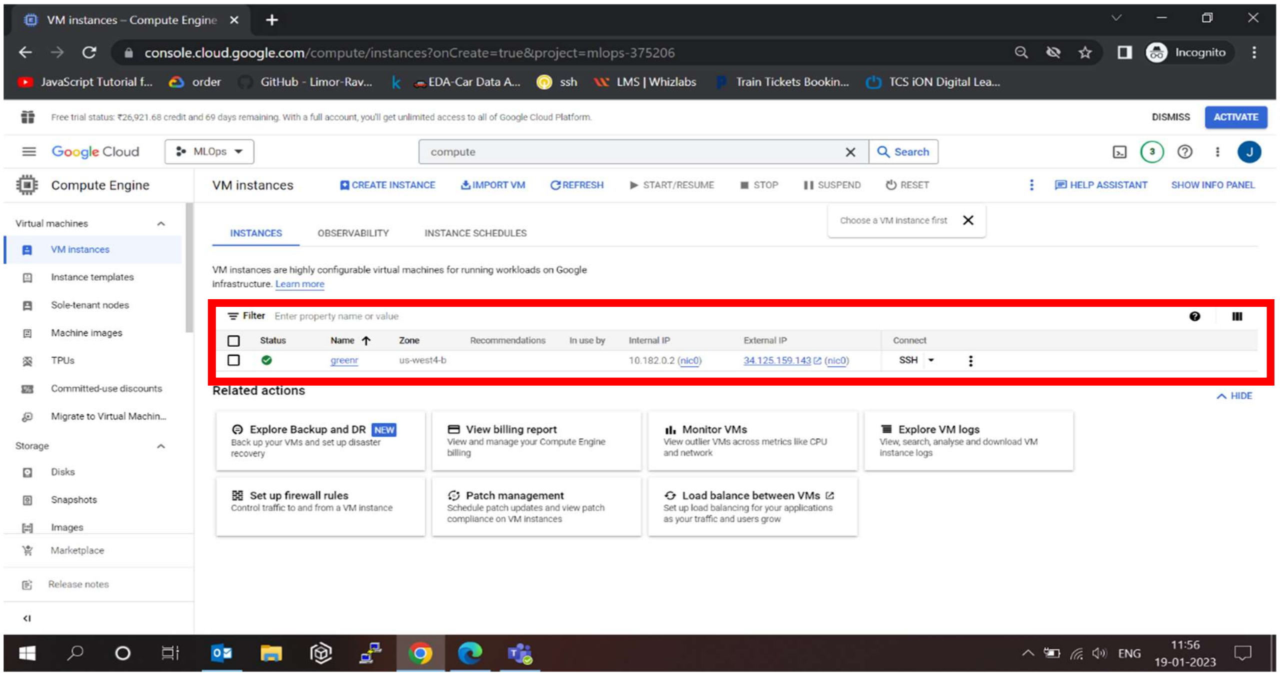
Topic: Deployment of a FLASK application in GCP by making use of DOCKER image

Task 1: Creation of Virtual Machine

- Sign in to the GCP account and search for Compute Engine services.
- Enable the Compute Engine API to work further.
- Click on Create Instances and create it.
- Configure the VM by selecting the required Series, Image, Access scope, and Firewall options.
- After a few minutes, the instance will be active.



Free trial status: ₹26,921.68 credit and 69 days remaining. With a full account, you'll get unlimited access to all of Google Cloud Platform.

VM instances are highly configurable virtual machines for running workloads on Google infrastructure. Learn more

□	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
□	✓	greenr	us-west4-b			10.182.0.2 (nic0)	34.125.159.143 (nic0)	SSH ▾

Explore Backup and DR NEW — Back up your VMs and set up disaster recovery

View billing report — View and manage your Compute Engine billing

Monitor VMs — View outlier VMs across metrics like CPU and network

Explore VM logs — View, search, analyse and download VM instance logs

Set up firewall rules — Control traffic to and from a VM instance

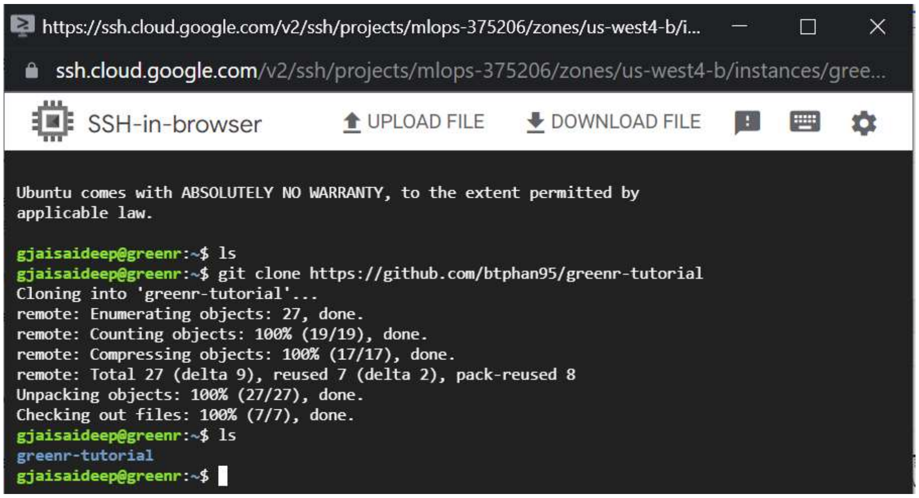
Patch management — Schedule patch updates and view patch compliance on VM instances

Load balance between VMs — Set up load balancing for your applications as your traffic and users grow

Task 2: Pulling the Trained Model from Github

- GitHub repository contains the pre-trained ML model.
- In order to make use of the contents in it Cloning of the repository inside the VM should take place.
- To achieve this, click the **SSH** option under the 'Connect' section of the created instance.
- Copy the repo directory link from GitHub.
- By making use of **git clone** command we can clone the whole repo inside required VM.

```
git clone https://github.com/btphan95/greenr-tutorial
```

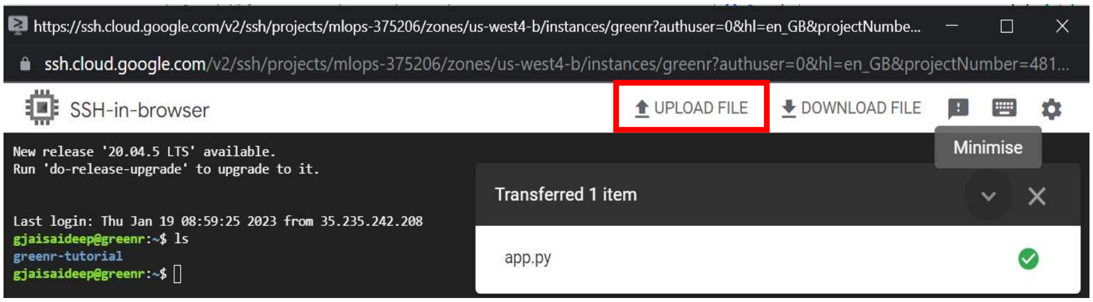

https://ssh.cloud.google.com/v2/ssh/projects/mlops-375206/zones/us-west4-b/i...
ssh.cloud.google.com/v2/ssh/projects/mlops-375206/zones/us-west4-b/instances/gree...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

gjaisaideep@greenr:~$ ls
gjaisaideep@greenr:~$ git clone https://github.com/btphan95/greenr-tutorial
Cloning into 'greenr-tutorial'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 27 (delta 9), reused 7 (delta 2), pack-reused 8
Unpacking objects: 100% (27/27), done.
Checking out files: 100% (7/7), done.
gjaisaideep@greenr:~$ ls
greenr-tutorial
gjaisaideep@greenr:~$ █
```

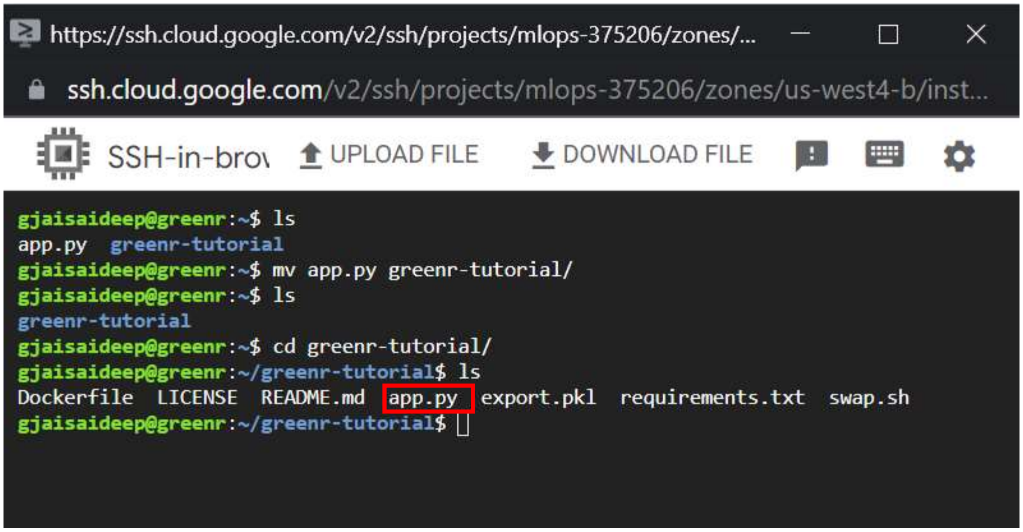
Task 3: Serving model on the web.

- Upload the **app.py** into the VM by making use of the upload button in the activated SSH window.
- Check whether the file got uploaded successfully by making use of **ls** command.
- Move the uploaded file from the existing directory to the required folder (greenr-tutorial) and confirm the same making use of **cd** and **ls** commands.


https://ssh.cloud.google.com/v2/ssh/projects/mlops-375206/zones/us-west4-b/instances/greenr?authuser=0&hl=en_GB&projectNumbe...
ssh.cloud.google.com/v2/ssh/projects/mlops-375206/zones/us-west4-b/instances/greenr?authuser=0&hl=en_GB&projectNumber=481...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
Minimise
Transferred 1 item
app.py

```
New release '20.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Thu Jan 19 08:59:25 2023 from 35.235.242.208
gjaisaideep@greenr:~$ ls
greenr-tutorial
gjaisaideep@greenr:~$ 
```

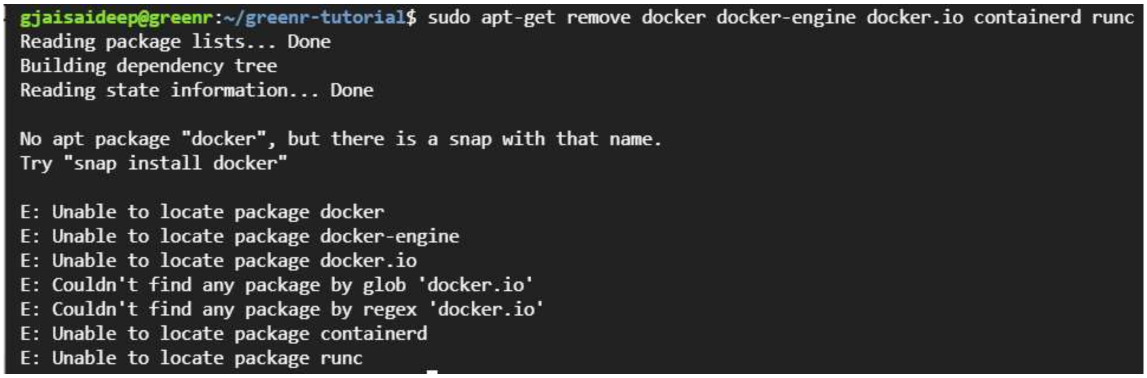

https://ssh.cloud.google.com/v2/ssh/projects/mlops-375206/zones/...
ssh.cloud.google.com/v2/ssh/projects/mlops-375206/zones/us-west4-b/inst...
SSH-in-brov
UPLOAD FILE
DOWNLOAD FILE

```
gjaisaideep@greenr:~$ ls
app.py  greenr-tutorial
gjaisaideep@greenr:~$ mv app.py greenr-tutorial/
gjaisaideep@greenr:~$ ls
greenr-tutorial
gjaisaideep@greenr:~$ cd greenr-tutorial/
gjaisaideep@greenr:~/greenr-tutorial$ ls
Dockerfile  LICENSE  README.md  app.py  export.pkl  requirements.txt  swap.sh
gjaisaideep@greenr:~/greenr-tutorial$ 
```

Task 4: Containerize App Using Docker

- Using Docker to build a container where the app will live. This lets the feasibility to run the app in its own environment, anywhere.
- Install Docker in our VM and uninstall the old versions in it by running the following command in SSH window.

```
sudo apt-get remove docker docker-engine docker.io containerd runc
```



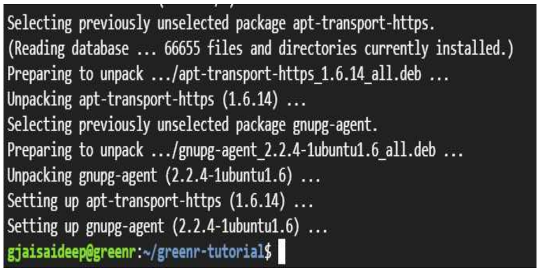
```
gjaisaideep@greenr:~/greenr-tutorial$ sudo apt-get remove docker docker-engine docker.io containerd runc
Reading package lists... Done
Building dependency tree
Reading state information... Done

No apt package "docker", but there is a snap with that name.
Try "snap install docker"

E: Unable to locate package docker
E: Unable to locate package docker-engine
E: Unable to locate package docker.io
E: Couldn't find any package by glob 'docker.io'
E: Couldn't find any package by regex 'docker.io'
E: Unable to locate package containerd
E: Unable to locate package runc
```

- By making use of below commands set up the repository.

```
sudo apt-get update
sudo apt-get install \
apt-transport-https \
ca-certificates \
curl \
gnupg-agent \
software-properties-common
```



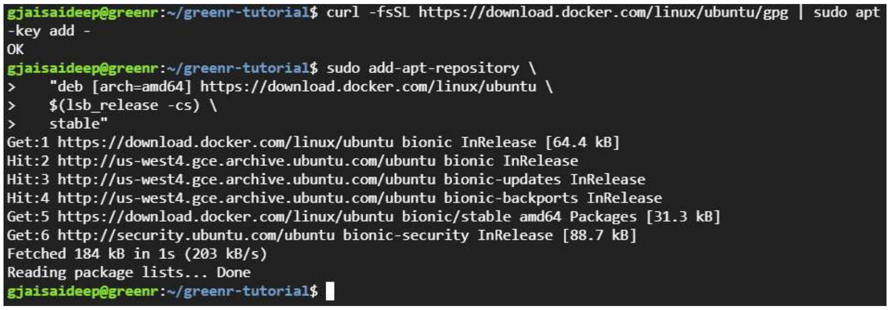
```
Selecting previously unselected package apt-transport-https.
(Reading database ... 66655 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_1.6.14_all.deb ...
Unpacking apt-transport-https (1.6.14) ...
Selecting previously unselected package gnupg-agent.
Preparing to unpack .../gnupg-agent_2.2.4-1ubuntu1.6_all.deb ...
Unpacking gnupg-agent (2.2.4-1ubuntu1.6) ...
Setting up apt-transport-https (1.6.14) ...
Setting up gnupg-agent (2.2.4-1ubuntu1.6) ...
gjaisaideep@greenr:~/greenr-tutorial$
```

- Add add Docker's official GPG key by making use of below command.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

- Add the stable repository by runnin gthe below command inside SSH terminal.

```
sudo add-apt-repository \
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
stable"
```

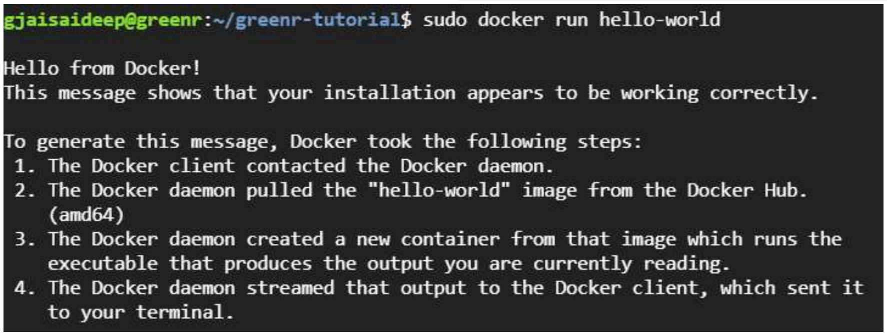


```
gjaisaideep@greenr:~/greenr-tutorial$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt
-key add -
OK
gjaisaideep@greenr:~/greenr-tutorial$ sudo add-apt-repository \
>    "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
>    $(lsb_release -cs) \
>    stable"
Get:1 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Hit:2 http://us-west4.gce.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://us-west4.gce.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://us-west4.gce.archive.ubuntu.com/ubuntu bionic-backports InRelease
Get:5 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [31.3 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Fetched 184 kB in 1s (203 kB/s)
Reading package lists... Done
gjaisaideep@greenr:~/greenr-tutorial$
```

- Update and Install Docker Engine.

```
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

- To verify the succsful installation of Docker run `sudo docker run hello-world`



```
gjaisaideep@greenr:~/greenr-tutorial$ sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.
```

- In the greenr directory, build the Docker container with the following command

```
sudo docker image build -t app:latest .
```

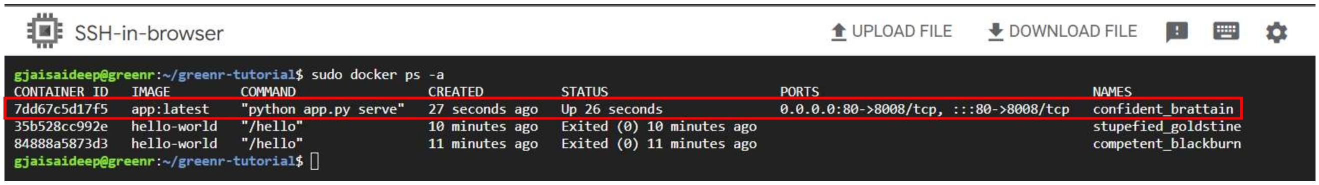
Task 5: Run Docker Container

- Run the Docker container.

```
sudo docker run -d -p 80:8008 app:latest
```

- List the details of containers by making use of below command.

```
sudo docker ps -a
```



```
gjaisaideep@greenr:~/greenr-tutorial$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS                      PORTS                                   NAMES
7dd67c5d17f5   app:latest     "python app.py serve"    27 seconds ago   Up 26 seconds               0.0.0.0:80->8008/tcp, :::80->8008/tcp   confident_brattain
35b528cc992e   hello-world    "/hello"                 10 minutes ago   Exited (0) 10 minutes ago                                           stupefied_goldstine
84888a5873d3   hello-world    "/hello"                 11 minutes ago   Exited (0) 11 minutes ago                                           competent_blackburn
gjaisaideep@greenr:~/greenr-tutorial$ 
```

Thanks and Regards,

Jai Sai Deep Gangavarapu

Software Engineer Trainee

Phone: +91 8106812015 **| Email:** jaisaideep17@gmail.com

Location: 84/1 & 86, Bascon Maeru, 3rd Floor, Kodambakkam High Road, Nungambakkam, Chennai-600034