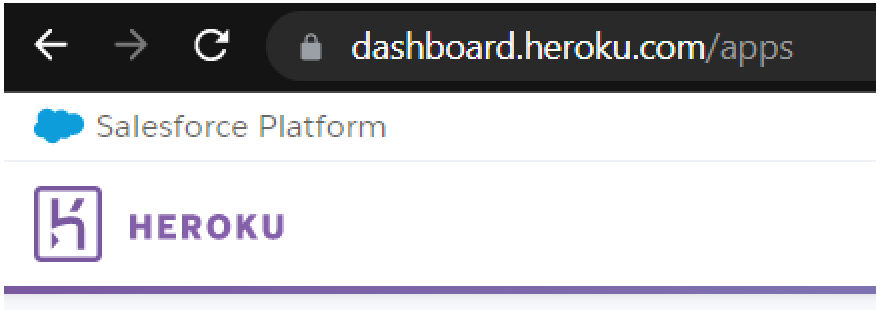
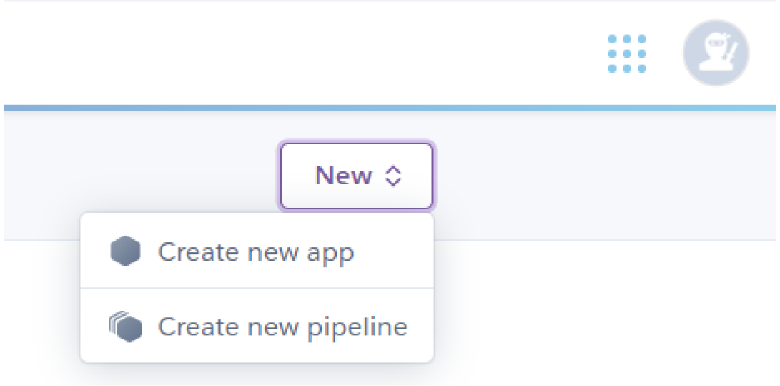
(GUI)

Deploying Models to Heroku For REST API Calls

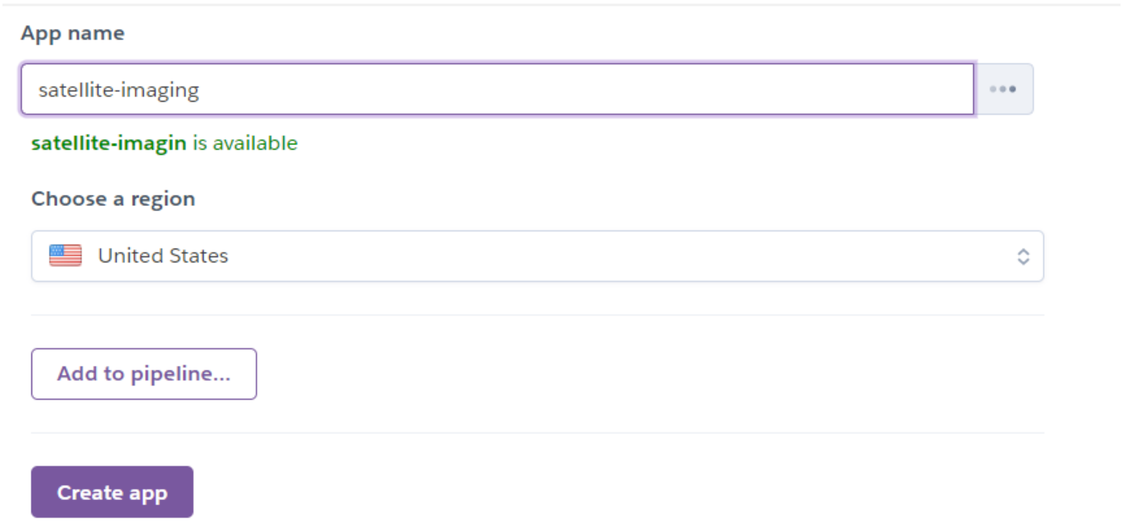
Step 1:- Create an Heroku account



Step 2:- Create an New Application



Step 3:- Create an New Application



App name

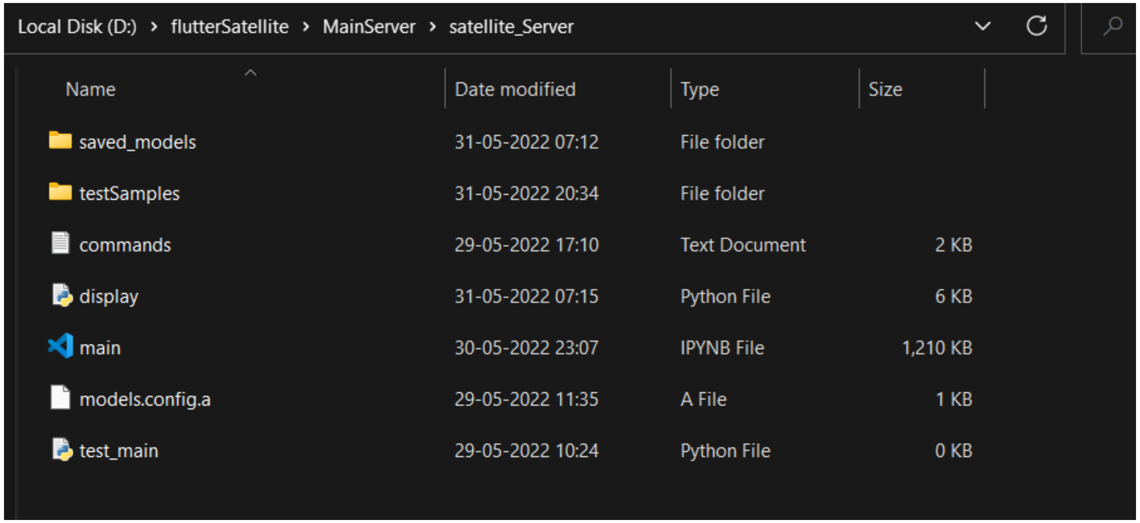
satellite-imaging

satellite-imagin is available

Choose a region

United States

Step 3:- Use the folder containing the following code (Currently Inside MainServer/Satellite)

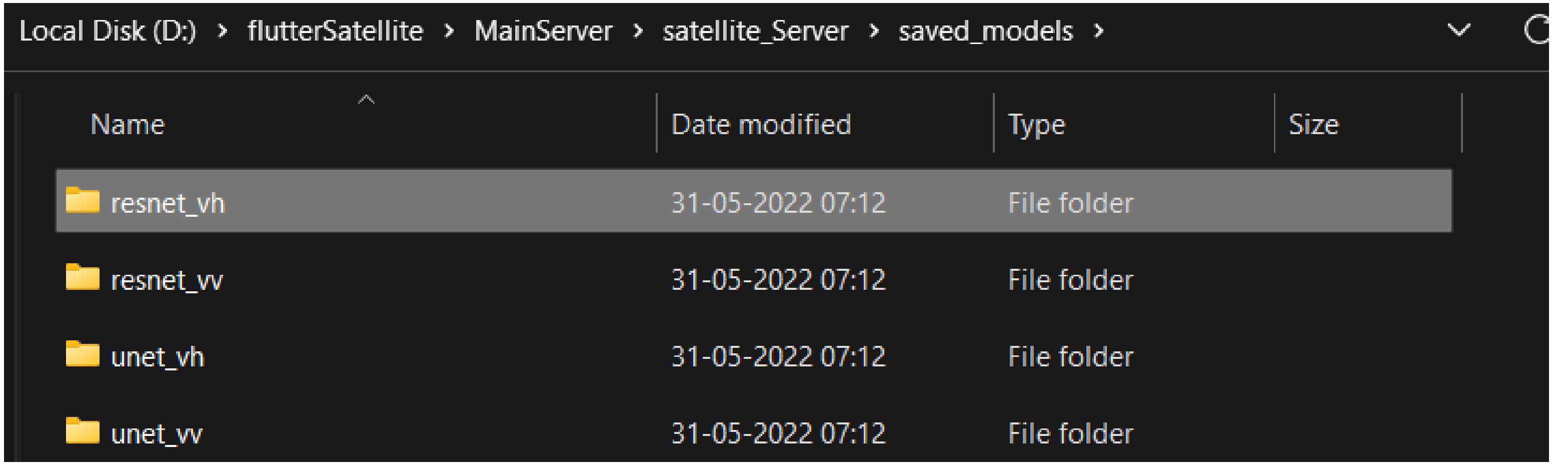


Local Disk (D:) › flutterSatellite › MainServer › satellite_Server

Name	Date modified	Type	Size
saved_models	31-05-2022 07:12	File folder	
testSamples	31-05-2022 20:34	File folder	
commands	29-05-2022 17:10	Text Document	2 KB
display	31-05-2022 07:15	Python File	6 KB
main	30-05-2022 23:07	IPYNB File	1,210 KB
models.config.a	29-05-2022 11:35	A File	1 KB
test_main	29-05-2022 10:24	Python File	0 KB

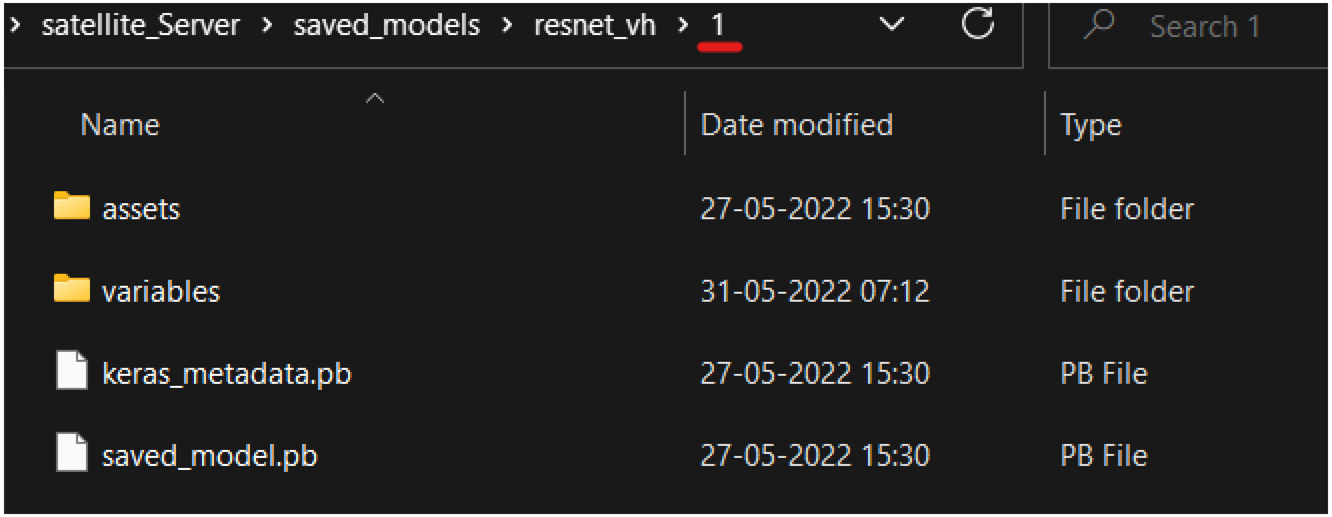
3.a:- Saving format for tensorflow models :- For Suitable Deployment Purpose

- Folder Name:- saved_models
- For Each model eg:- resnet_vh do model.save(“./resnet_vh/1/”)
- We save under another subfolder named “ 1 ” (Denotes the saved version of resnet_vh which is understood by creating subfolder 1 in resnet_vh folder to save that respective version1 weights)



Local Disk (D:) › flutterSatellite › MainServer › satellite_Server › saved_models ›

Name	Date modified	Type	Size
resnet_vh	31-05-2022 07:12	File folder	
resnet_vv	31-05-2022 07:12	File folder	
unet_vh	31-05-2022 07:12	File folder	
unet_vv	31-05-2022 07:12	File folder	

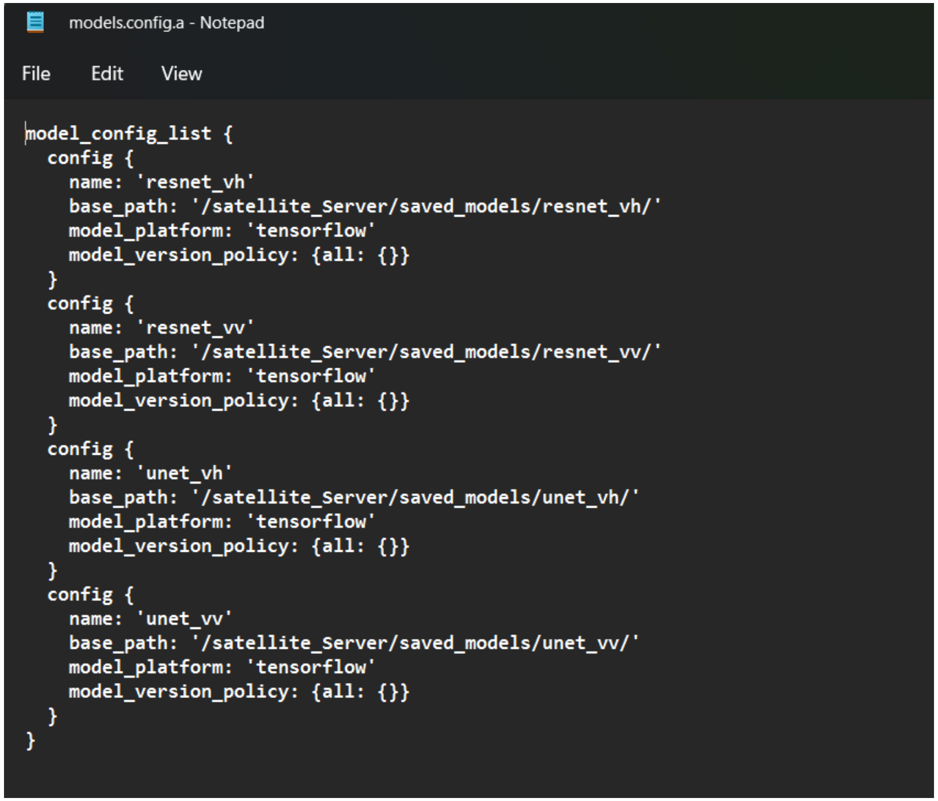


› satellite_Server › saved_models › resnet_vh › 1

Name	Date modified	Type
assets	27-05-2022 15:30	File folder
variables	31-05-2022 07:12	File folder
keras_metadata.pb	27-05-2022 15:30	PB File
saved_model.pb	27-05-2022 15:30	PB File

3.b:- Choose the necessary model to deploy :- models.config.a

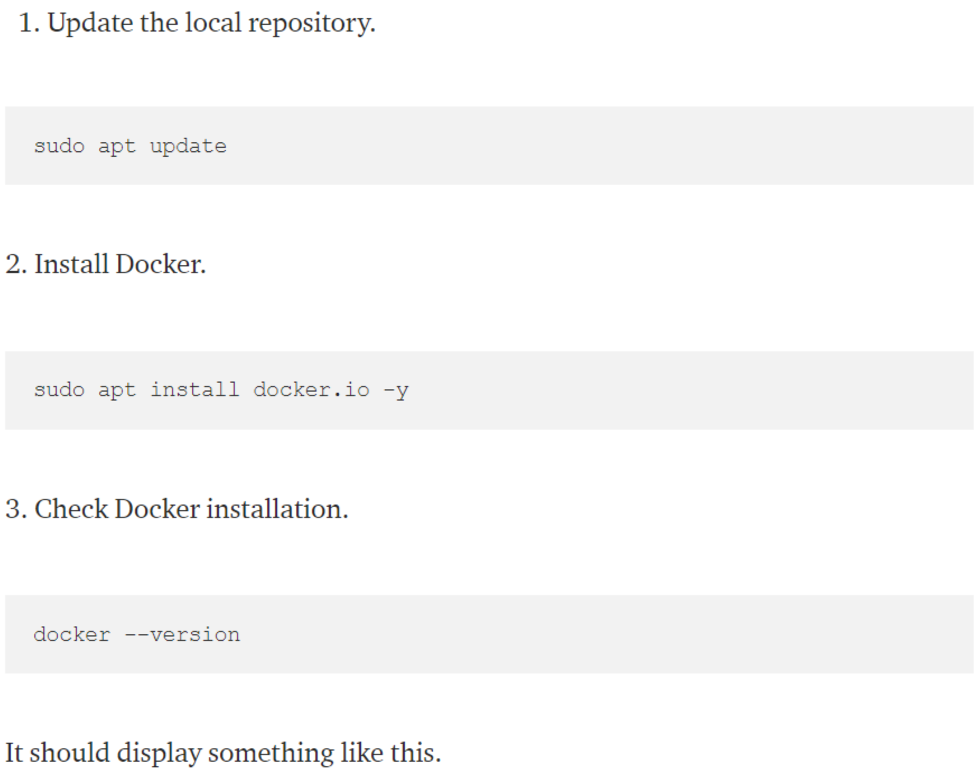
- Here each config is a model detail
- all: {} is used for denoting to choose all the versions present in that directory
- We currently created only subfolder named “1” under each module
- So tensorflow_serving serves 1 model



```
model_config_list {
  config {
    name: 'resnet_vh'
    base_path: '/satellite_Server/saved_models/resnet_vh/'
    model_platform: 'tensorflow'
    model_version_policy: {all: {}}
  }
  config {
    name: 'resnet_vv'
    base_path: '/satellite_Server/saved_models/resnet_vv/'
    model_platform: 'tensorflow'
    model_version_policy: {all: {}}
  }
  config {
    name: 'unet_vh'
    base_path: '/satellite_Server/saved_models/unet_vh/'
    model_platform: 'tensorflow'
    model_version_policy: {all: {}}
  }
  config {
    name: 'unet_vv'
    base_path: '/satellite_Server/saved_models/unet_vv/'
    model_platform: 'tensorflow'
    model_version_policy: {all: {}}
  }
}
```

4. INSTALL DOCKER

For More Details :-
https://medium.com/geekculture/run-docker-in-windows-10-11-wsl-without-docker-desktop-a2a7eb90556d



1. Update the local repository.

```
sudo apt update
```

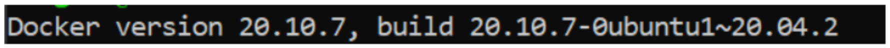
2. Install Docker.

```
sudo apt install docker.io -y
```

3. Check Docker installation.

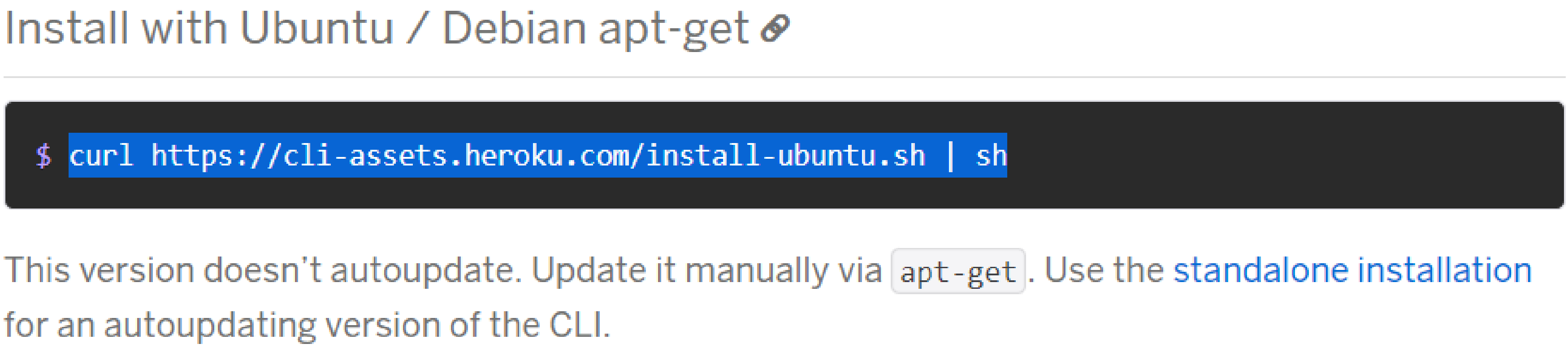
```
docker --version
```

It should display something like this.



```
Docker version 20.10.7, build 20.10.7-0ubuntu1~20.04.2
```

4.a INSTALL HEROKU



Install with Ubuntu / Debian apt-get

```
$ curl https://cli-assets.heroku.com/install-ubuntu.sh | sh
```

This version doesn't autoupdate. Update it manually via `apt-get`. Use the standalone installation for an autoupdating version of the CLI.

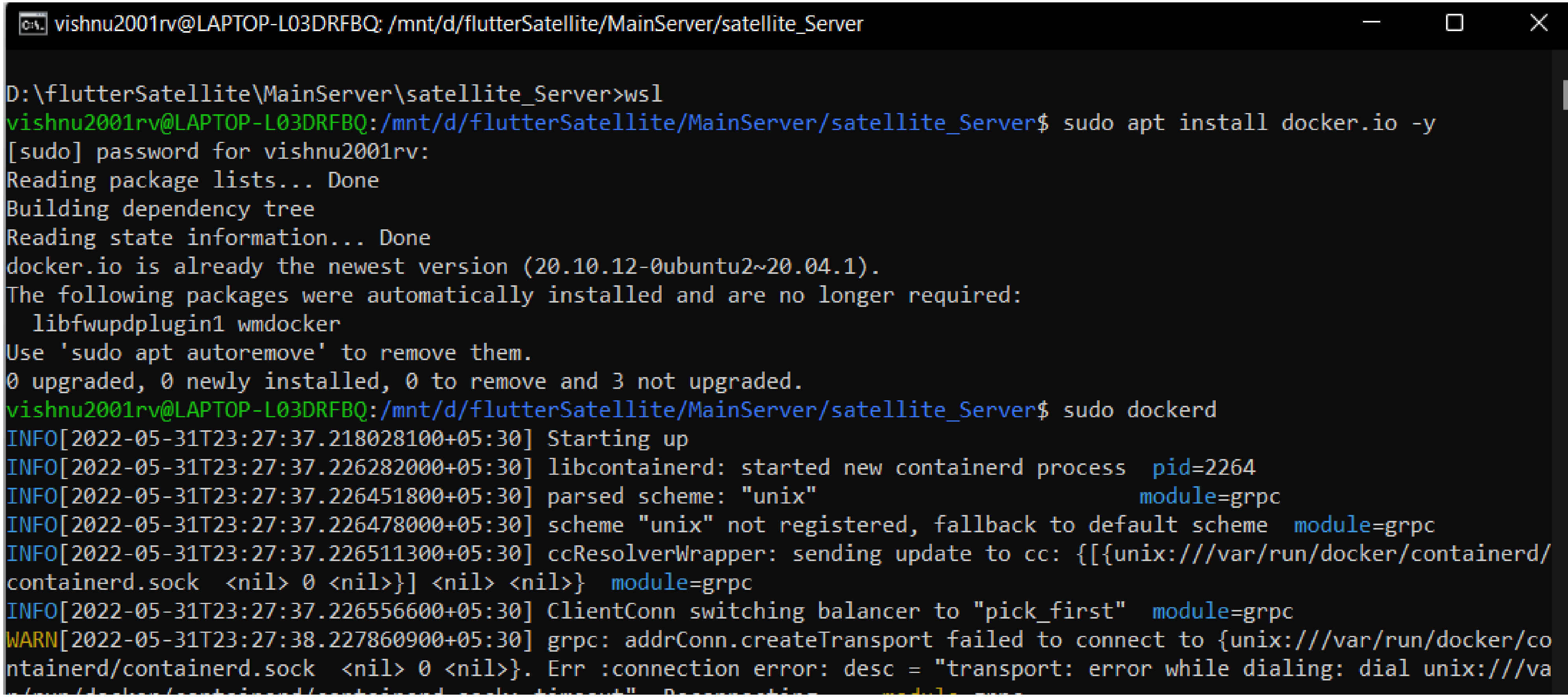
If Heroku is unrecogonized Do the following

alias heroku=/home/bin/heroku

Where /home/bin/heroku is where the curl installed the heroku
Later can usere heroku command directly eg heroku –version

5. Running the Local Server

- Start Docker(Server) open new terminal in the location



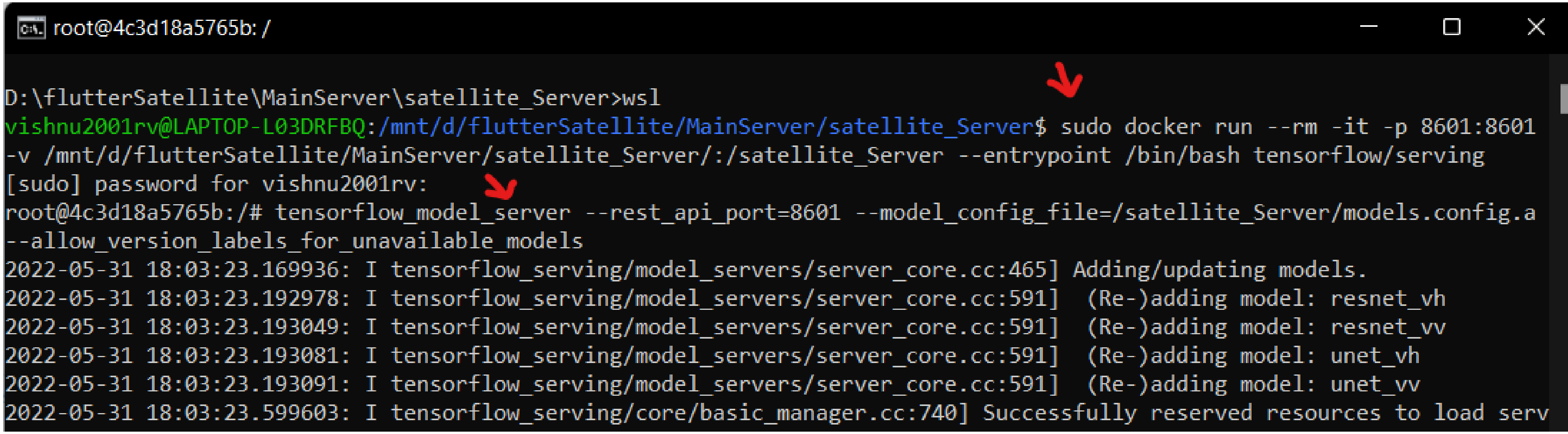
vishnu2001rv@LAPTOP-L03DRFBQ: /mnt/d/flutterSatellite/MainServer/satellite_Server

```
D:\flutterSatellite\MainServer\satellite_Server>wsl
vishnu2001rv@LAPTOP-L03DRFBQ:/mnt/d/flutterSatellite/MainServer/satellite_Server$ sudo apt install docker.io -y
[sudo] password for vishnu2001rv:
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (20.10.12-0ubuntu2~20.04.1).
The following packages were automatically installed and are no longer required:
  libfwupdplugin1 wmdocker
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
vishnu2001rv@LAPTOP-L03DRFBQ:/mnt/d/flutterSatellite/MainServer/satellite_Server$ sudo dockerd
INFO[2022-05-31T23:27:37.218028100+05:30] Starting up
INFO[2022-05-31T23:27:37.226282000+05:30] libcontainerd: started new containerd process  pid=2264
INFO[2022-05-31T23:27:37.226451800+05:30] parsed scheme: "unix"                         module=grpc
INFO[2022-05-31T23:27:37.226478000+05:30] scheme "unix" not registered, fallback to default scheme  module=grpc
INFO[2022-05-31T23:27:37.226511300+05:30] ccResolverWrapper: sending update to cc: {[{unix:///var/run/docker/containerd/
containerd.sock  <nil> 0 <nil>}] <nil> <nil>}  module=grpc
INFO[2022-05-31T23:27:37.226556600+05:30] ClientConn switching balancer to "pick_first"  module=grpc
WARN[2022-05-31T23:27:38.227860900+05:30] grpc: addrConn.createTransport failed to connect to {unix:///var/run/docker/co
ntainerd/containerd.sock  <nil> 0 <nil>}. Err :connection error: desc = "transport: error while dialing: dial unix:///va
r/run/docker/containerd/containerd.sock: timeout". Reconnecting...  module=grpc
```

6. Start the tensorflow serving

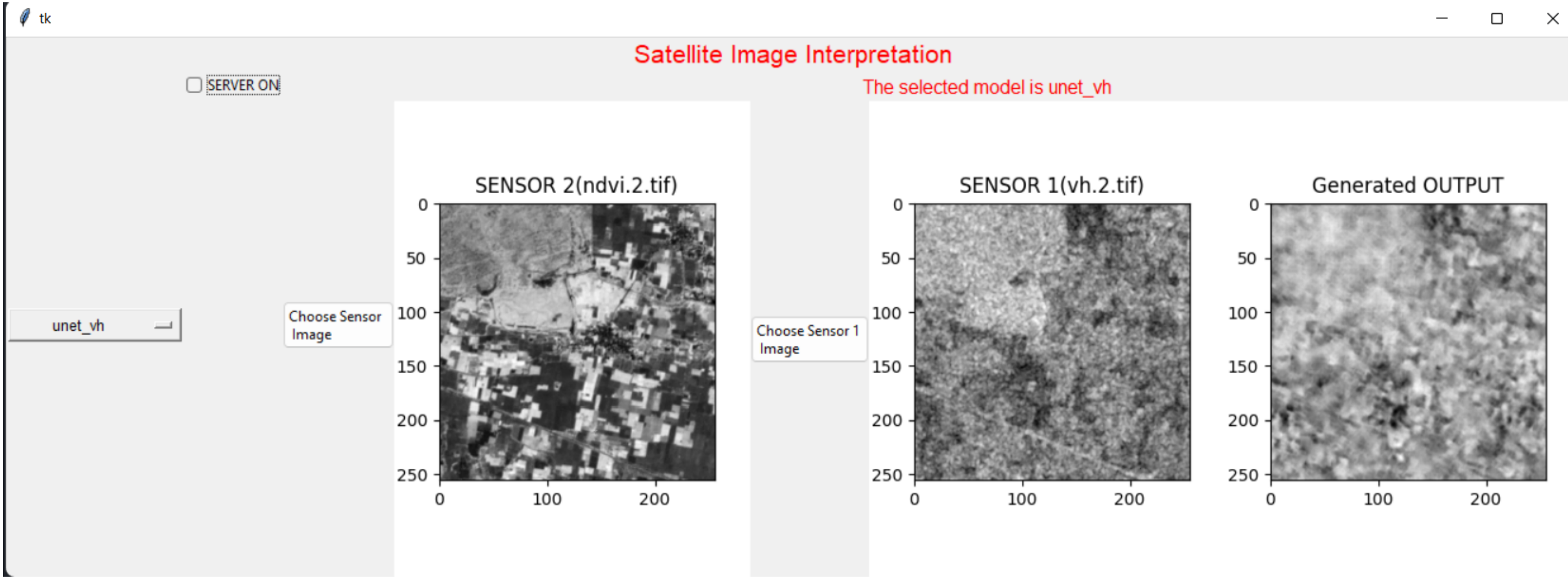
sudo docker run --rm -it -p 8601:8601 -v /mnt/d/flutterSatellite/MainServer/satellite_Server/:/satellite_Server --entrypoint /bin/bash tensorflow/serving

tensorflow_model_server --rest_api_port=8601 --model_config_file=/satellite_Server/models.config.a --allow_version_labels_for_unavailable_models



```
D:\flutterSatellite\MainServer\satellite_Server>wsl
vishnu2001rv@LAPTOP-L03DRFBQ:/mnt/d/flutterSatellite/MainServer/satellite_Server$ sudo docker run --rm -it -p 8601:8601
-v /mnt/d/flutterSatellite/MainServer/satellite_Server/:/satellite_Server --entrypoint /bin/bash tensorflow/serving
[sudo] password for vishnu2001rv:
root@4c3d18a5765b:/# tensorflow_model_server --rest_api_port=8601 --model_config_file=/satellite_Server/models.config.a
--allow_version_labels_for_unavailable_models
2022-05-31 18:03:23.169936: I tensorflow_serving/model_servers/server_core.cc:465] Adding/updating models.
2022-05-31 18:03:23.192978: I tensorflow_serving/model_servers/server_core.cc:591]  (Re-)adding model: resnet_vh
2022-05-31 18:03:23.193049: I tensorflow_serving/model_servers/server_core.cc:591]  (Re-)adding model: resnet_vv
2022-05-31 18:03:23.193081: I tensorflow_serving/model_servers/server_core.cc:591]  (Re-)adding model: unet_vh
2022-05-31 18:03:23.193091: I tensorflow_serving/model_servers/server_core.cc:591]  (Re-)adding model: unet_vv
2022-05-31 18:03:23.599603: I tensorflow_serving/core/basic_manager.cc:740] Successfully reserved resources to load serv
```

7. Check In GUI (PUT Server Off) then choose Sensor1 image and select model to see the output



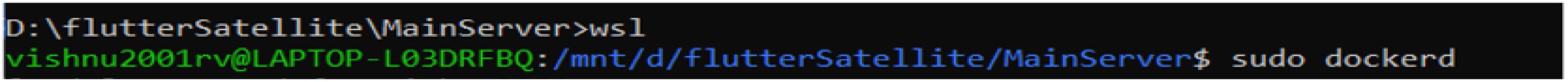
Satellite Image Interpretation

□ SERVER ON

The selected model is unet_vh

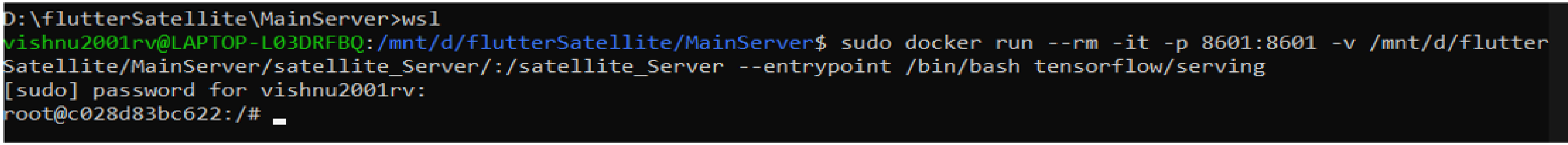
8. For Deploying to Server

- Go to the Previous Directory. (Main Server)
- Do the following Steps



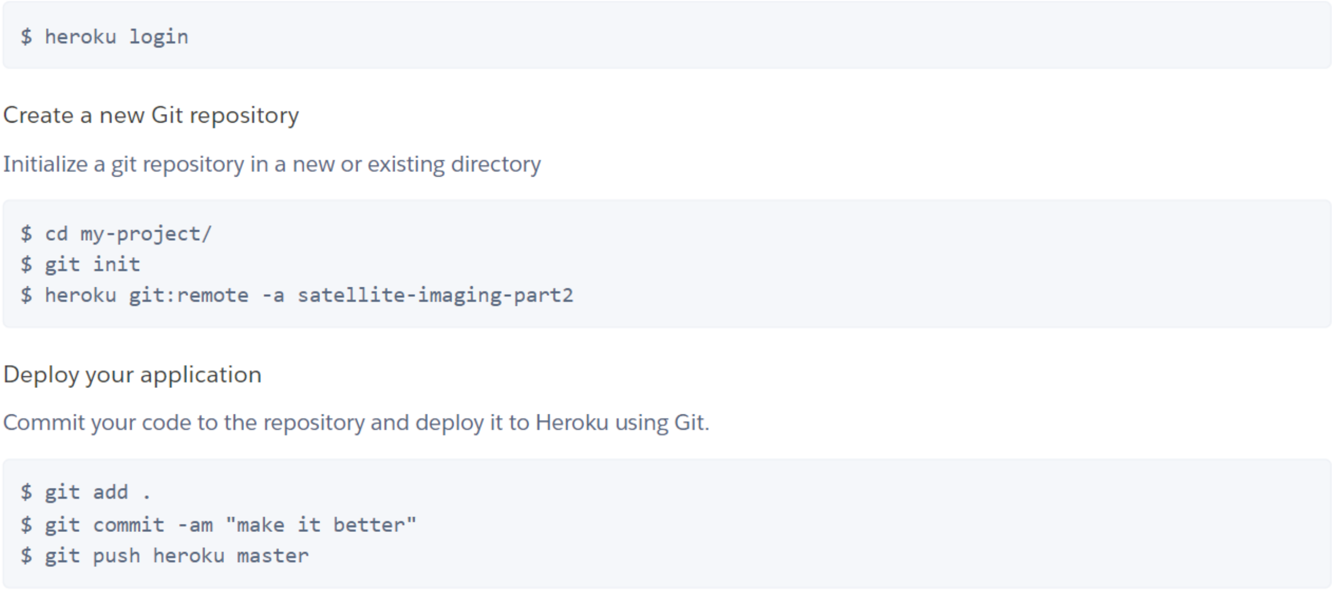
```
D:\flutterSatellite\MainServer>wsl
vishnu2001rv@LAPTOP-L03DRFBQ:/mnt/d/flutterSatellite/MainServer$ sudo dockerd
```

Open another terminal and type wsl



```
D:\flutterSatellite\MainServer>wsl
vishnu2001rv@LAPTOP-L03DRFBQ:/mnt/d/flutterSatellite/MainServer$ sudo docker run --rm -it -p 8601:8601 -v /mnt/d/flutter
Satellite/MainServer/satellite_Server/:/satellite_Server --entrypoint /bin/bash tensorflow/serving
[sudo] password for vishnu2001rv:
root@c028d83bc622:/# _
```

In MainServer folder open a terminal with wsl and do the following Steps to Initialize heroku



```
$ heroku login
```

Create a new Git repository

Initialize a git repository in a new or existing directory

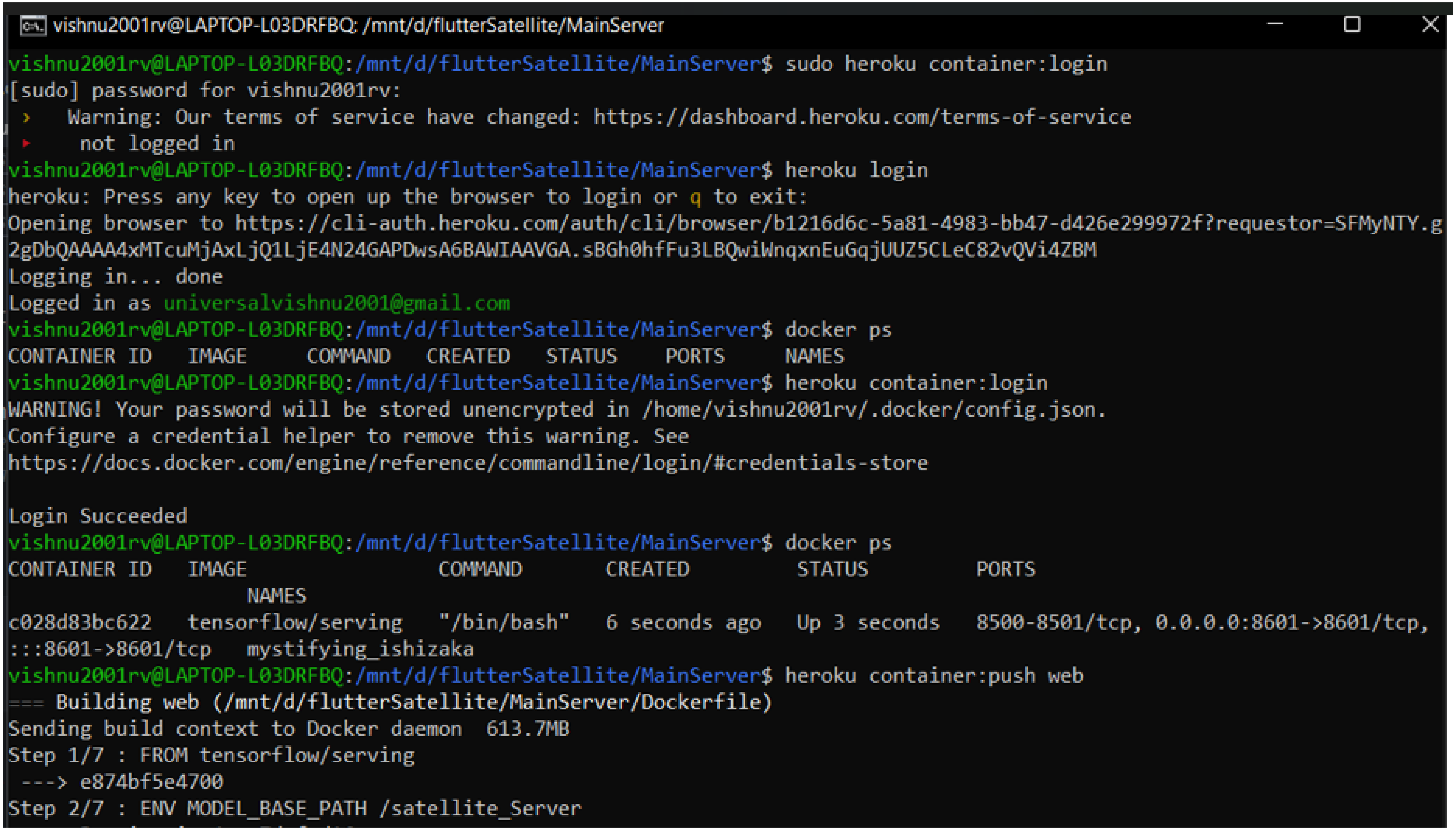
```
$ cd my-project/
$ git init
$ heroku git:remote -a satellite-imaging-part2
```

Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.

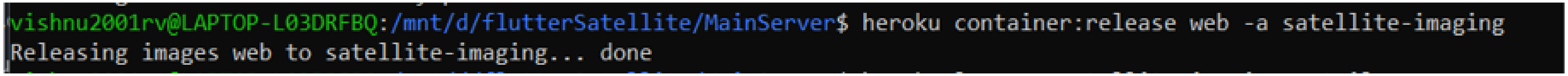
```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```

To Deploy the docker container to heroku Do the following.


vishnu2001rv@LAPTOP-L03DRFBQ: /mnt/d/flutterSatellite/MainServer

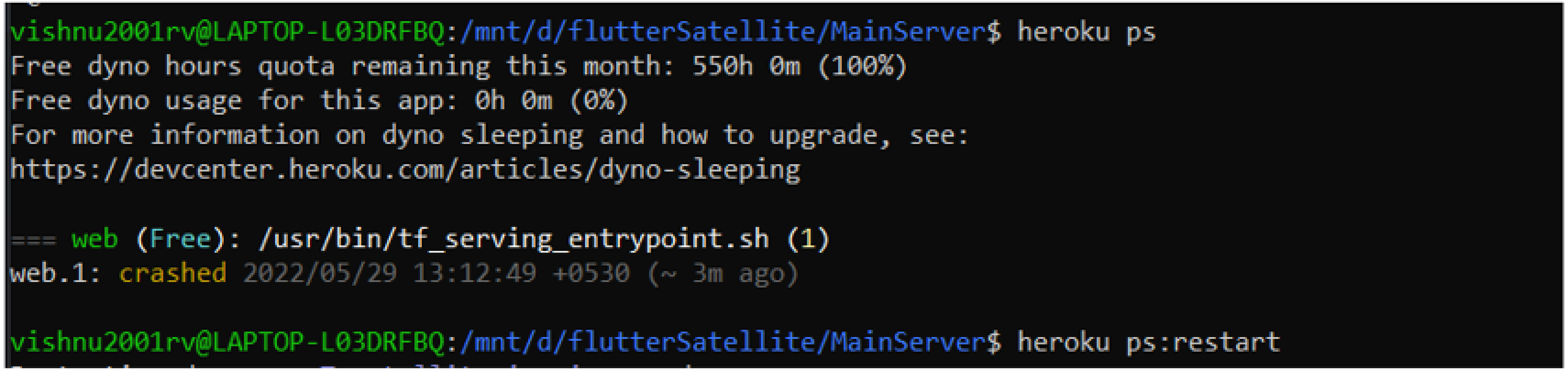
```
vishnu2001rv@LAPTOP-L03DRFBQ:/mnt/d/flutterSatellite/MainServer$ sudo heroku container:login
[sudo] password for vishnu2001rv:
 ›   Warning: Our terms of service have changed: https://dashboard.heroku.com/terms-of-service
 ▸    not logged in
vishnu2001rv@LAPTOP-L03DRFBQ:/mnt/d/flutterSatellite/MainServer$ heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/b1216d6c-5a81-4983-bb47-d426e299972f?requestor=SFMyNTY.g
2gDbQAAAA4xMTcuMjAxLjQ1LjE4N24GAPDwsA6BAWIAAVGA.sBGh0hfFu3LBQwiWnqxnEuGqjUUZ5CLeC82vQVi4ZBM
Logging in... done
Logged in as universalvishnu2001@gmail.com
vishnu2001rv@LAPTOP-L03DRFBQ:/mnt/d/flutterSatellite/MainServer$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
vishnu2001rv@LAPTOP-L03DRFBQ:/mnt/d/flutterSatellite/MainServer$ heroku container:login
WARNING! Your password will be stored unencrypted in /home/vishnu2001rv/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
vishnu2001rv@LAPTOP-L03DRFBQ:/mnt/d/flutterSatellite/MainServer$ docker ps
CONTAINER ID   IMAGE                COMMAND       CREATED         STATUS         PORTS
                 NAMES
c028d83bc622   tensorflow/serving   "/bin/bash"   6 seconds ago   Up 3 seconds   8500-8501/tcp, 0.0.0.0:8601->8601/tcp,
:::8601->8601/tcp   mystifying_ishizaka
vishnu2001rv@LAPTOP-L03DRFBQ:/mnt/d/flutterSatellite/MainServer$ heroku container:push web
=== Building web (/mnt/d/flutterSatellite/MainServer/Dockerfile)
Sending build context to Docker daemon  613.7MB
Step 1/7 : FROM tensorflow/serving
 ---> e874bf5e4700
Step 2/7 : ENV MODEL_BASE_PATH /satellite_Server
```

Release the container (Publishing the Tensorflow Server)

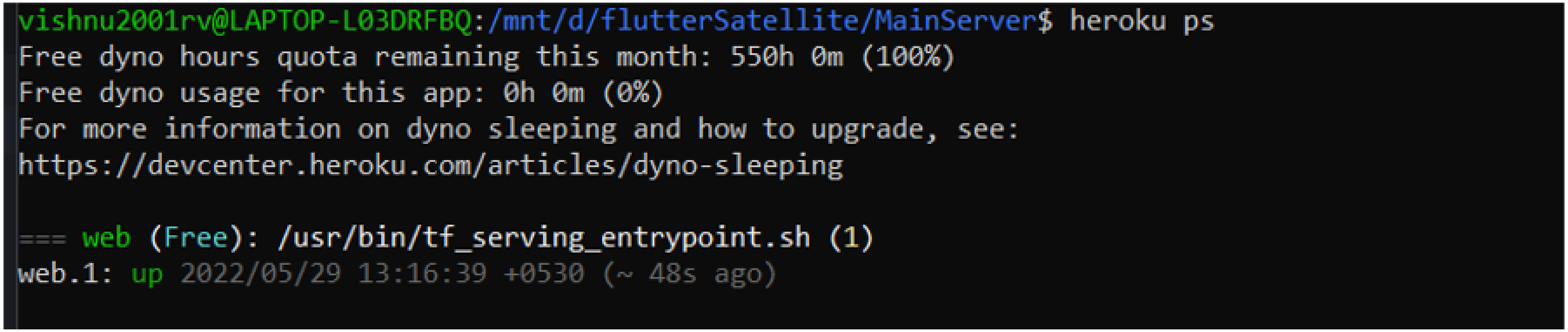


```
vishnu2001rv@LAPTOP-L03DRFBQ:/mnt/d/flutterSatellite/MainServer$ heroku container:release web -a satellite-imaging
Releasing images web to satellite-imaging... done
```

To Check the status of the Server:- **(Do ps:restart only if Server crashed)** **(Upgrading heroku can get more ram, thus preventing Server Crashes)**

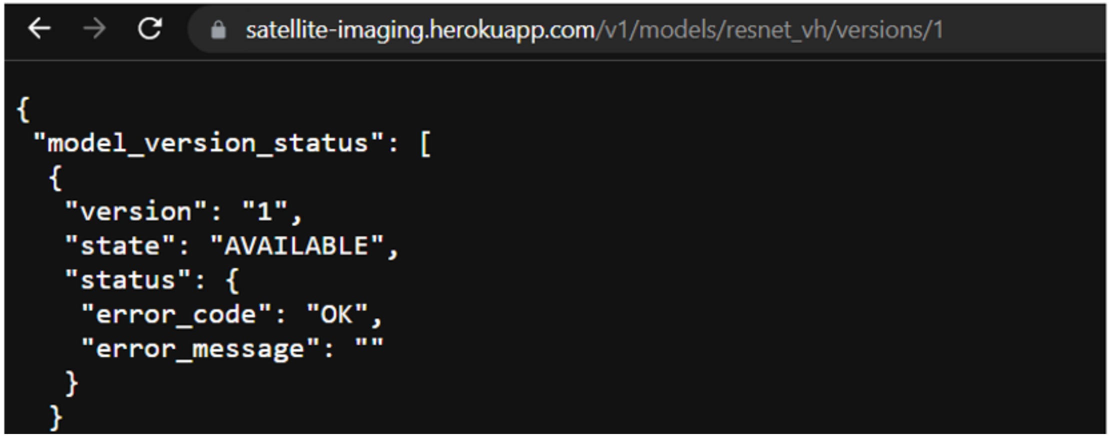


```
vishnu2001rv@LAPTOP-L03DRFBQ:/mnt/d/flutterSatellite/MainServer$ heroku ps
Free dyno hours quota remaining this month: 550h 0m (100%)
Free dyno usage for this app: 0h 0m (0%)
For more information on dyno sleeping and how to upgrade, see:
https://devcenter.heroku.com/articles/dyno-sleeping

=== web (Free): /usr/bin/tf_serving_entrypoint.sh (1)
web.1: crashed 2022/05/29 13:12:49 +0530 (~ 3m ago)

vishnu2001rv@LAPTOP-L03DRFBQ:/mnt/d/flutterSatellite/MainServer$ heroku ps:restart
```



```
vishnu2001rv@LAPTOP-L03DRFBQ:/mnt/d/flutterSatellite/MainServer$ heroku ps
Free dyno hours quota remaining this month: 550h 0m (100%)
Free dyno usage for this app: 0h 0m (0%)
For more information on dyno sleeping and how to upgrade, see:
https://devcenter.heroku.com/articles/dyno-sleeping

=== web (Free): /usr/bin/tf_serving_entrypoint.sh (1)
web.1: up 2022/05/29 13:16:39 +0530 (~ 48s ago)
```

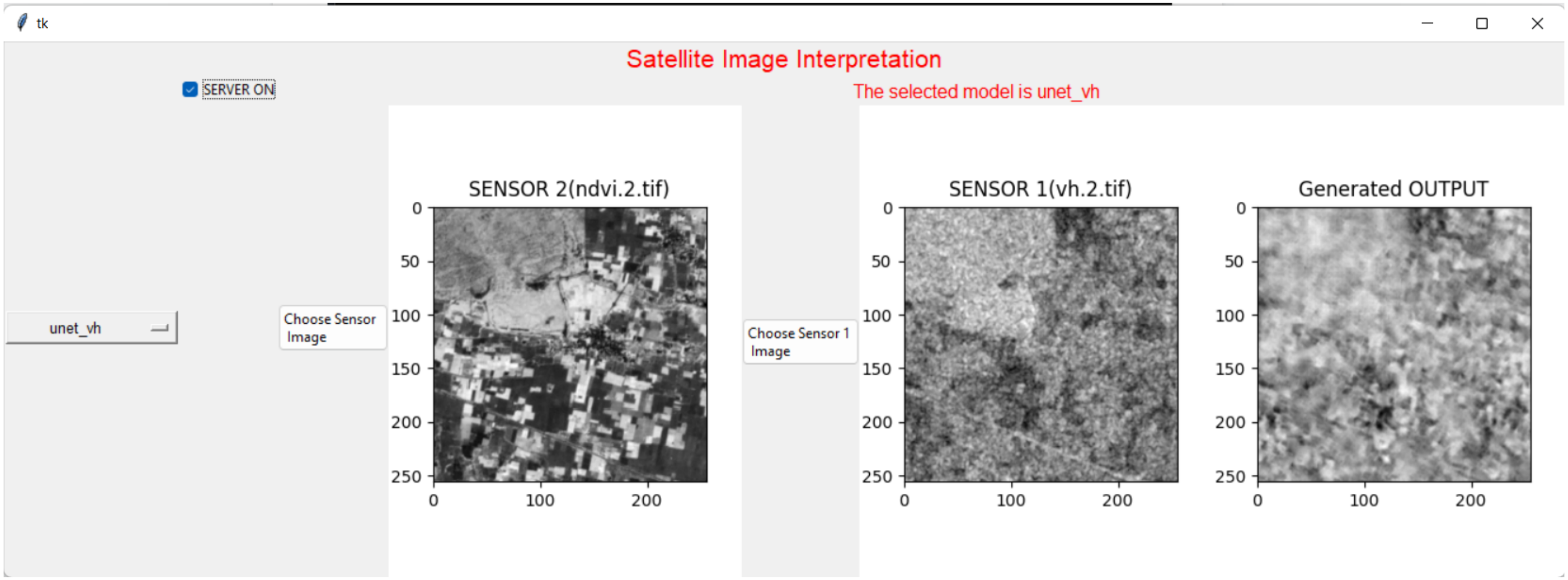
To check If the Server Runs Type this in URL



satellite-imaging.herokuapp.com/v1/models/resnet_vh/versions/1

```
{
 "model_version_status": [
  {
   "version": "1",
   "state": "AVAILABLE",
   "status": {
    "error_code": "OK",
    "error_message": ""
   }
  }
```

To Check Server with GUI Use:- Server ON option Note:- GUI gets Paused when LOADING predictions. Then It Resumes.



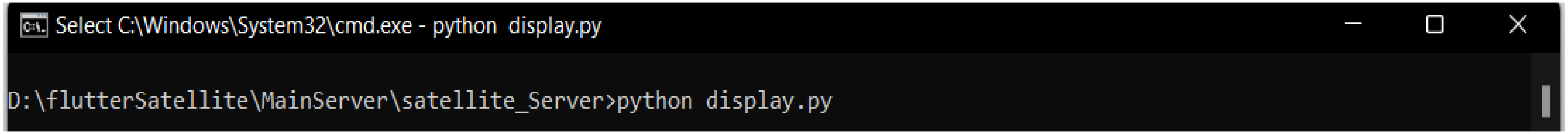
Satellite Image Interpretation

☑ SERVER ON

The selected model is unet_vh

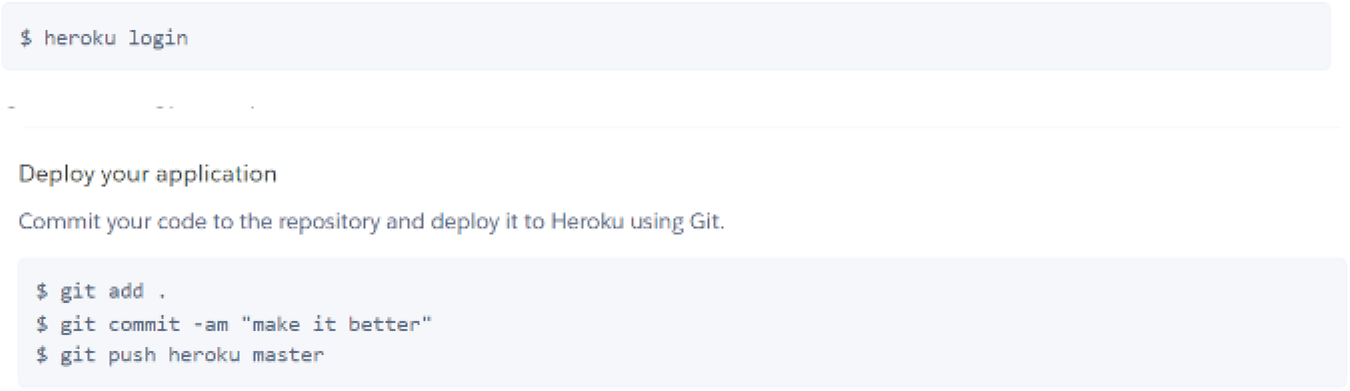
unet_vh

To Run the Portable GUI With Server/ Local Server Enabled.



```
D:\flutterSatellite\MainServer\satellite_Server>python display.py
```

Later on Updating the
Component eg saved_models directory do the following



```
$ heroku login
```

Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.

```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```

And Repeat the same pushing of container.