

ML-Deployment on Heroku.

Applied AI Course.com

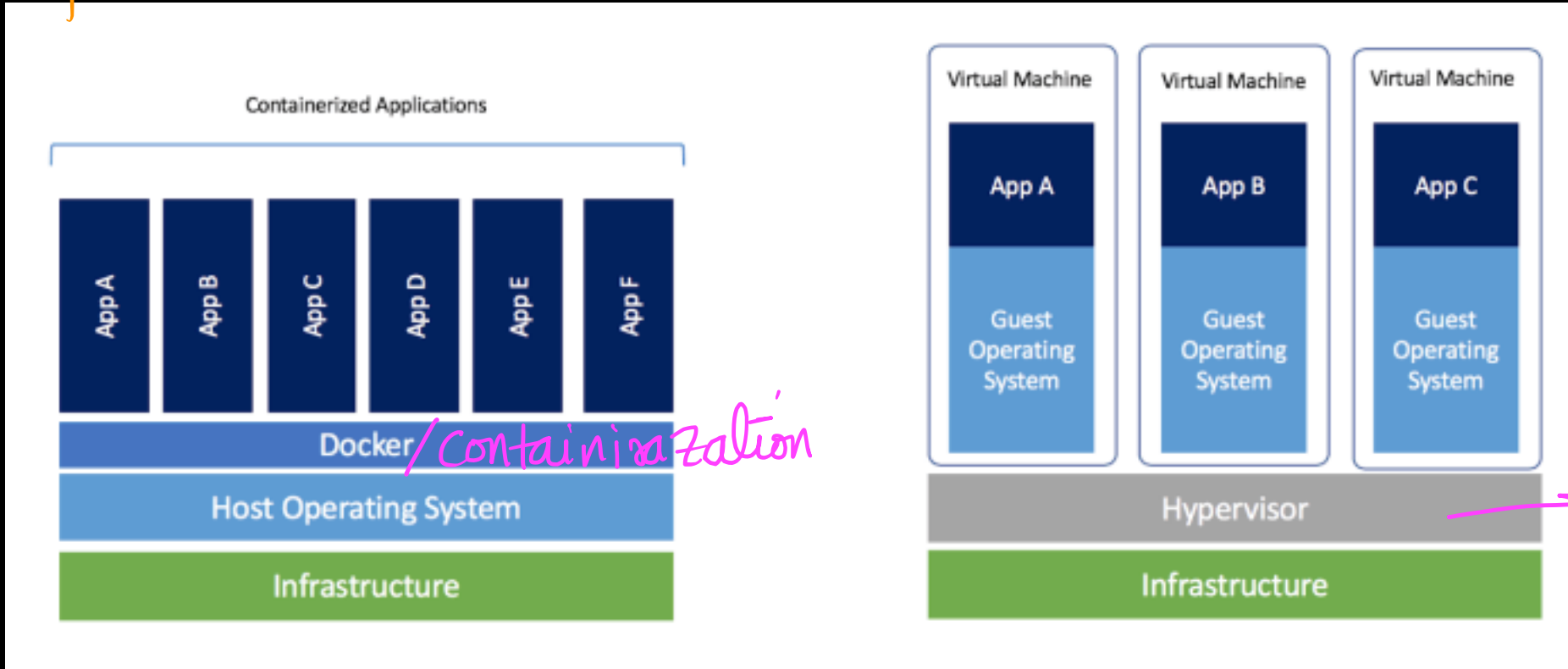
Pre-req:

1. Flask APIs & Gunicorn
2. Streamlit Webapps
3. Docker & Containerization
4. AWS EC2

Agenda:

1. EC2 vs Heroku architectures
2. Python Virtual Env
3. Flask API + Gunicorn
4. Deploying Flask APIs on Heroku
5. Deploying streamlit on Heroku

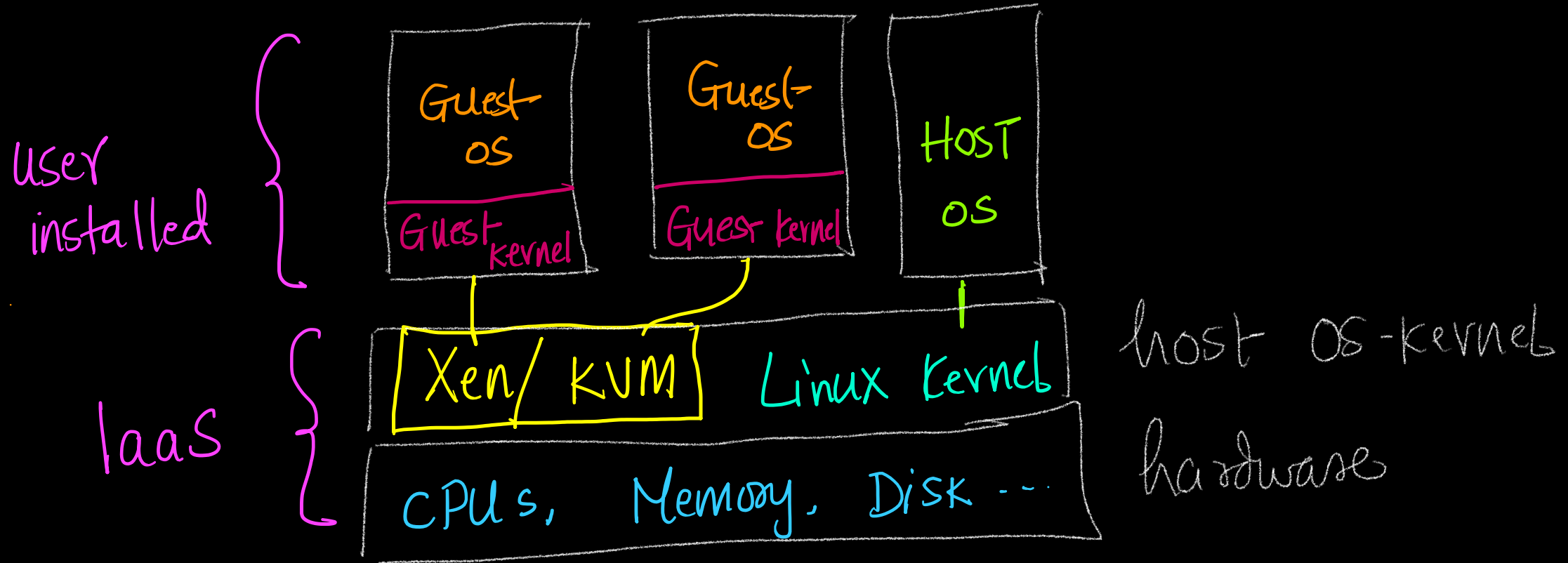
Recap: Virtualization vs Containerization



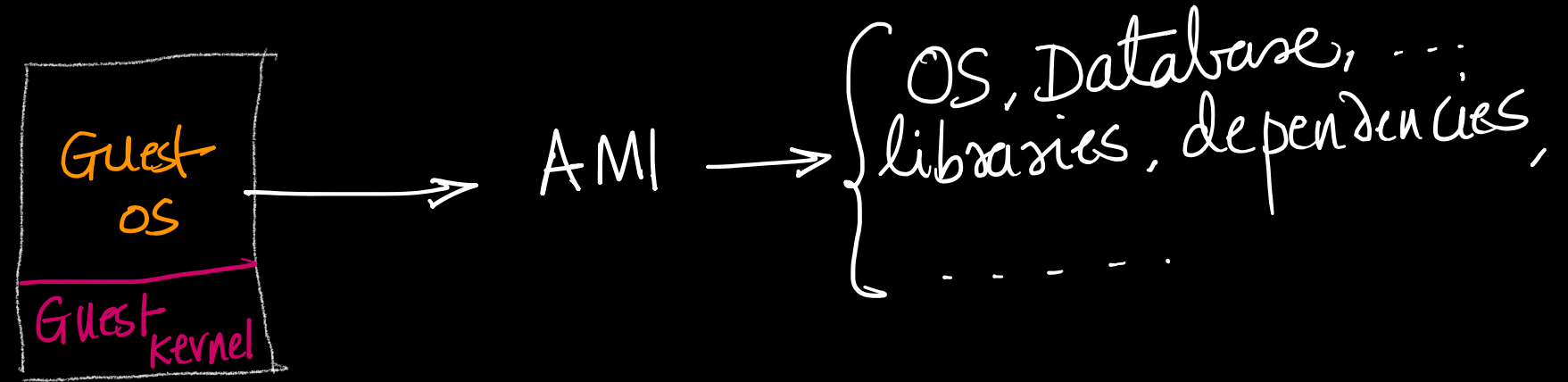
Source: Docker.com

VM-monitor
e.g.: Xen, KVM
VM-ware

AWS - EC2 [IaaS - model]



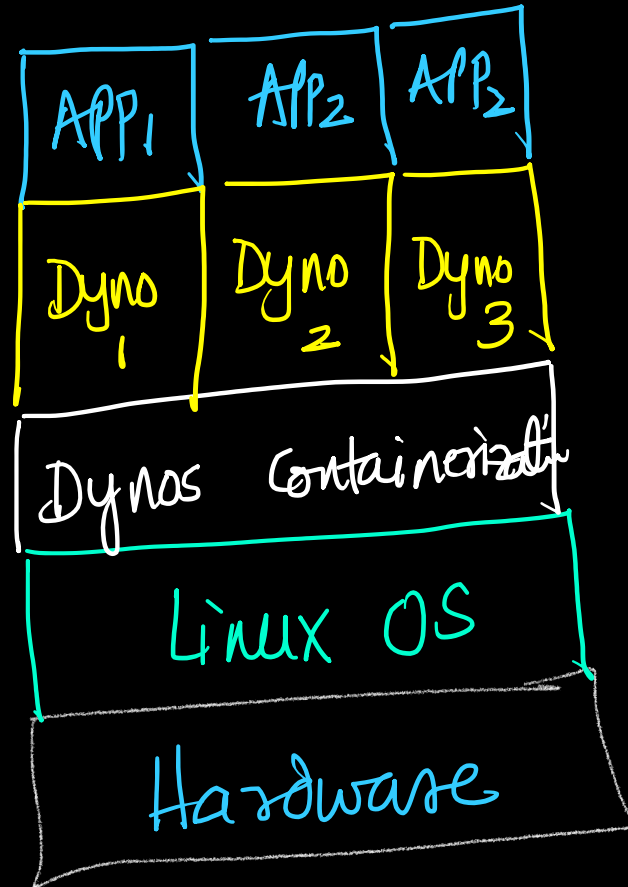
AWS - EC2 - instance



- ssh into EC2-instance

- run your code remotely

Heroku [PaaS]



Heroku simplifies app-dev

→ requirements.txt

→ Procfile

→ Autodetect the app type

→ git push & start running

→ [550+] hrs of free
runtime per
month

→ autosleeping

— Web dynos vs worker dynos

— Dyno types: <https://devcenter.heroku.com/articles/dyno-types>

— Slog: prepacked & compressed form of our app

— Databases → Postgres, Redis,

— AWS-PaaS: Elastic Beanstalk

Python Virtualenv

Refer: <https://packaging.python.org/guides/installing-using-pip-and-virtual-environments/>

Local Box:

```
$ pip install virtualenv
```

```
$ mkdir HerokuDeployment
```

```
$ cd HerokuDeployment
```

```
$ python -m venv venv/
```

```
$ source venv/bin/activate -----> .\venv\Scripts\activate [on windows]
```

```
$ which python
```

```
$ deactivate ----->[Don't do it now]
```

Flask - API + Gunicorn

In the VirtualEnv, install packages that we want to train the model and run Flask-API locally

```
$ pip install flask  
$ pip install gunicorn  
$ pip install numpy  
$ pip install sklearn  
$ pip install pandas
```

Download IRIS (from Flask-API and AWS session)----->

https://drive.google.com/file/d/15E1oFiBMO_UTGZuqydF0Gu9yMC0UC9Lq/view

```
$ cd IRIS  
$ python model.py  
$ python app.py
```

Browser -----> <http://localhost:8080/index> ----->

Iris Data Prediction

sepal_length	<input type="text" value="10"/>
sepal_width	<input type="text" value="12"/>
petal_length	<input type="text" value="14"/>
petal_width	<input type="text" value="45"/>

-----> {"prediction":["virginica"]}

Required Python packages

```
$ pip freeze > requirements.txt
```

→ from current virtualenv

Create Procfile

Procfile → web: gunicorn app:app

IRIS folder

```
iris.csv  
templates  
model.pkl  
model.py  
app.py  
requirements.txt  
Procfile
```

Heroku.com: signup & login

Dashboard

The screenshot shows the Heroku dashboard interface. At the top left is the Heroku logo. In the center of the top bar is a search bar with the text "Jump to Favorites, Apps, Pipelines, Spaces...". On the right side of the top bar is a "New" button with a dropdown arrow. Below the top bar, on the left, is a user profile section with a person icon and the text "Personal" followed by a dropdown arrow. This section is circled in white, and a white arrow points from a circled "1" to it. The main content area has a dark background and contains the following text: "You don't have any apps yet", "Every app and pipeline you create or become a collaborator on will appear here", a blue "Create new app" button, "Looking for help getting started?", "Get started by reading one of our language guides in the Dev Center", and a "Choose a language guide..." button with a dropdown arrow. A white arrow points from a circled "2" to the "Create new app" button.

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Personal ▾

New ▾

You don't have any apps yet

Every app and pipeline you create or become a collaborator on will appear here

Create new app

Looking for help getting started?

Get started by reading one of our language guides in the Dev Center

Choose a language guide... ▾

Create New App

App name

aaic-flask-api



aaic-flask-api is available

Choose a region



United States



Add to pipeline...

Create app

Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

①



Add this app to a stage in a pipeline to enable additional features



Pipelines let you connect multiple apps together and **promote code** between them. [Learn more.](#)

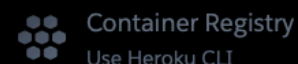
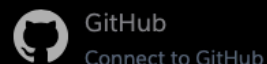


Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. [Learn more.](#)

Choose a pipeline

Deployment method

②



Deploy using Heroku Git

Use git in the command line or a GUI tool to deploy this app.

③



Install the Heroku CLI

Download and install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

Heroku - CLI

Refer: <https://devcenter.heroku.com/articles/heroku-cli>

- install Git

- install Heroku - CLI

↓
OS - dependent

Create a Git-repo

```
$ cd IRIS
```

```
$ git init .
```

```
$ git add *
```

```
$ git commit -m "commit_1"
```

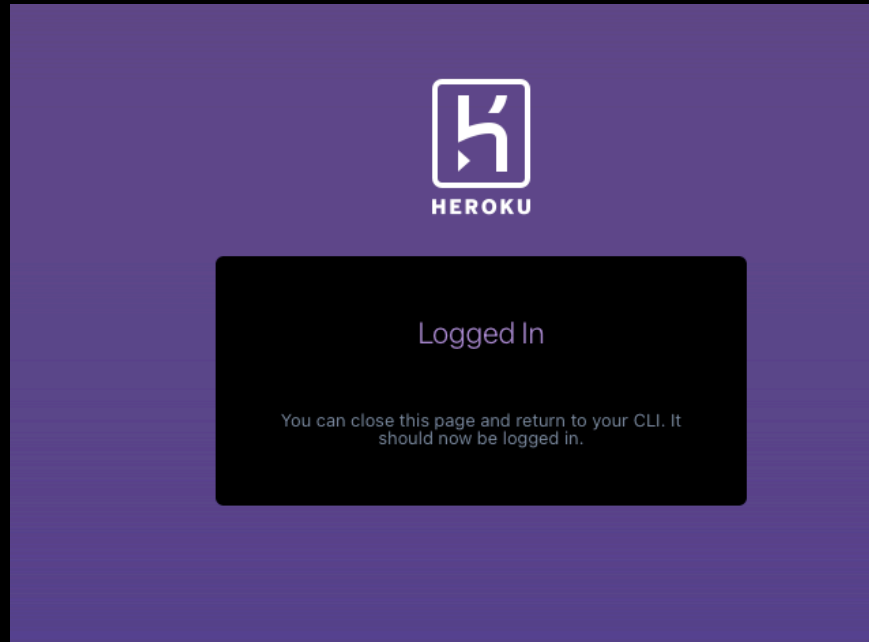
Heroku-CLI login:

```
$ heroku login
```

```
> Warning: Our terms of service have changed:
```

```
https://dashboard.heroku.com/terms-of-service
```

```
heroku: Press any key to open up the browser to login or q to exit:
```



Git - push to Heroku - app:

\$ heroku git:remote -a aaic-flask-api -----> add your local repo to a remote one


\$ git push heroku master -----> push to heroku

<https://dashboard.heroku.com/apps/aaic-flask-api/logs>

The screenshot shows the Heroku dashboard for the application 'aaic-flask-api'. The 'More' dropdown menu is open, and the 'View logs' option is highlighted with a yellow arrow. The application logs are displayed below the menu, showing a 500 error.

```
2020-12-13T06:16:57.667870+00:00 app[web.1]: raise TemplateNotFound(template)
2020-12-13T06:16:57.672149+00:00 app[web.1]: jinja2.exceptions.TemplateNotFound: index.html
2020-12-13T06:16:57.673668+00:00 app[web.1]: 10.30.91.53 - - [13/Dec/2020:06:16:57 +0000] "GET /index HTTP/1.1" 500 290 "-" "Mozilla/5.0
X 10_15_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.0.1 Safari/605.1.15"
2020-12-13T06:16:57.675025+00:00 heroku[router]: at=info method=GET path="/index" host=aaic-flask-api.herokuapp.com request_id=cf7775ab-7
314e54beb070 fwd="223.230.115.124" dyno=web.1 connect=1ms service=8ms status=500 bytes=470 protocol=https
2020-12-13T06:17:46.499849+00:00 heroku[router]: at=info method=GET path="/index" host=aaic-flask-api.herokuapp.com request_id=916470ac-daf6-47e0-bada-
d23f3dfbb158 fwd="223.230.115.124" dyno=web.1 connect=1ms service=18ms status=500 bytes=470 protocol=https
2020-12-13T06:17:46.485183+00:00 app[web.1]: [2020-12-13 06:17:46,484] ERROR in app: Exception on /index [GET]
```

Heroku app folder-structure

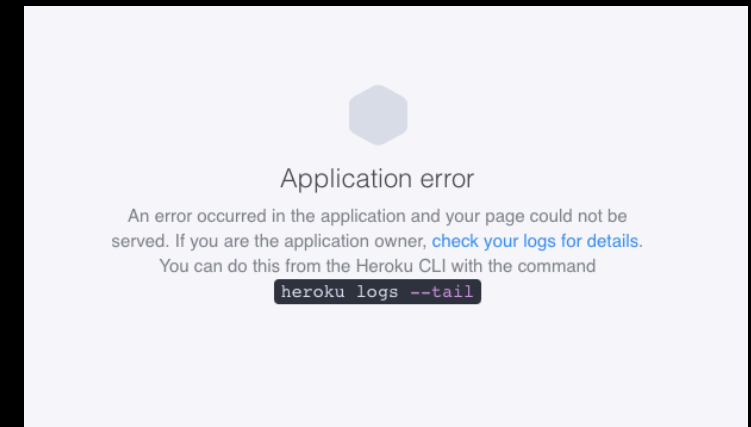
```
$ heroku run bash -a aaic-flask-api  
Running bash on  aaic-flask-api... up, run.3578 (Free)  
~$
```

Heroku stop-app

```
$ heroku ps:scale web=0 -a aaic-flask-api
```

```
Scaling dynos... done, now running web at 0:Free  
Refer: https://devcenter.heroku.com/articles/scaling
```

<https://aaic-flask-api.herokuapp.com>



Streamlit on Heroku:

Download code from <https://drive.google.com/file/d/16RHjtIMIJKymma7tiu3jwqFtG7ngH0UI/view?usp=sharing>

```
$ python -m venv venvStreamlit/
```

```
$ source venvStreamlit/bin/activate -----> .\ venvStreamlit\Scripts\activate [on windows]
```

```
$ which python
```

```
$ pip install streamlit
```

```
$ cd streamlitPlot
```

```
$ streamlit run ./plotting.py -----> Run on local box to check if its working
```

```
$ pip freeze > requirements.txt
```

Refer: <https://discuss.streamlit.io/t/issue-with-serving-a-streamlit-heroku-app-in-a-browser/2543/6>

```
- setup.sh
```

```
- Procfile
```

```
$ git init .
```

```
$ git add *
```

```
$ git commit -m "commit_i"
```

```
$ heroku git:remote -a aaic-streamlit-app
```

```
$ git push heroku master -----> Version number changed in requirements.txt to avoid errors
```

Q & A