ML - Deployment on Heroku.

AppliedAICourse.com

APPLIED ROOTS

Pre-req:

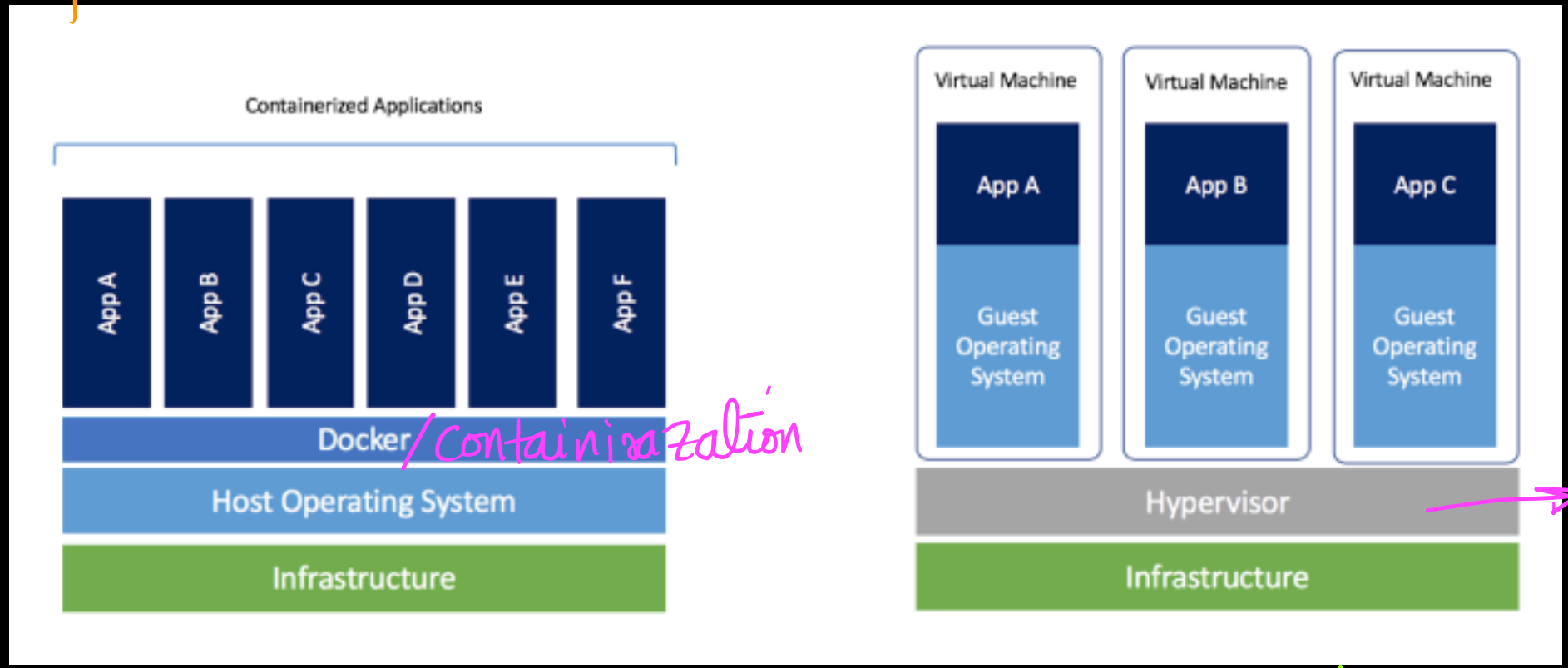
1. Flask APIs & Gunicorn
2. Streamlit webapps
3. Docker & Containerization
4. AWS EC2

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

APPLIED ROOTS

Recap: Virtualization vs Containirazation

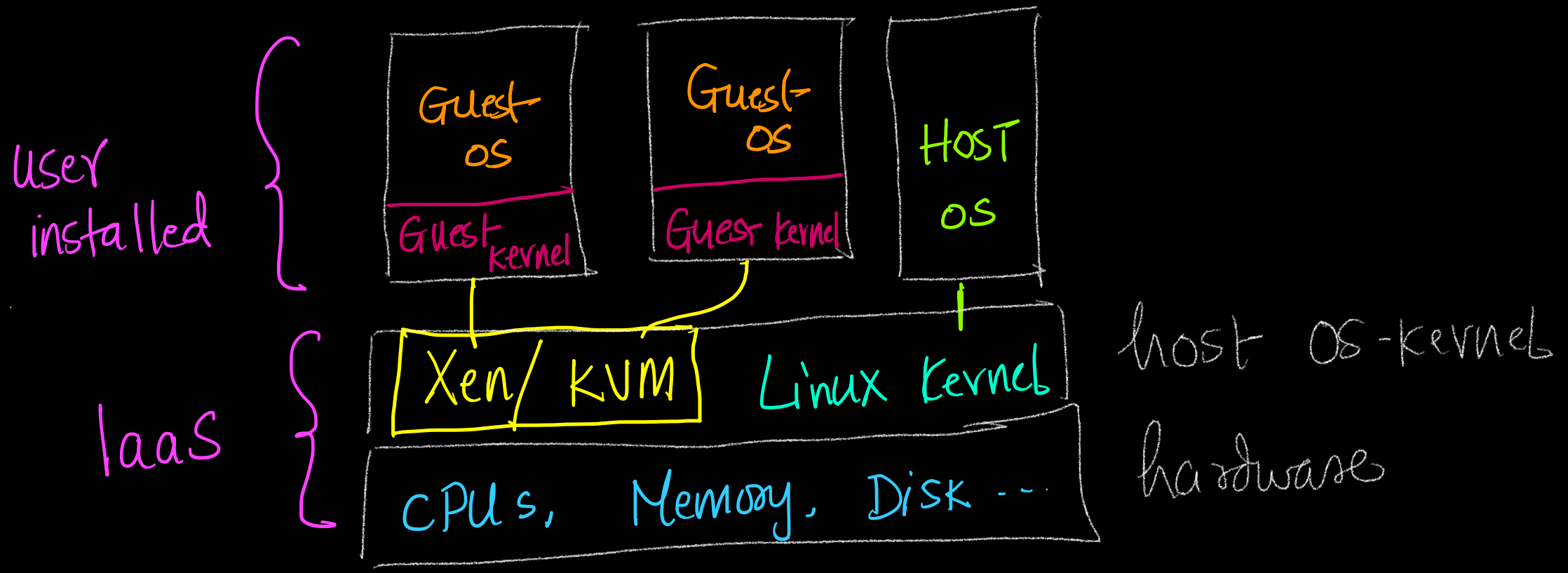


VM - monitor
e.g: Xen, KVM
VM-ware

Source: Docker.com

APPLIED ROOTS

AWS - EC2 [IaaS - model]



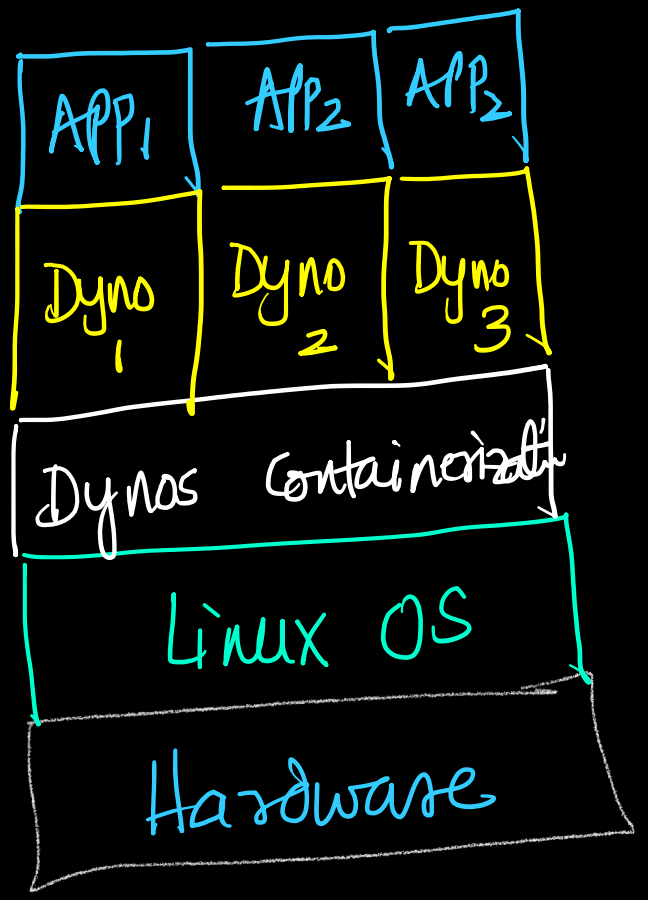
AWS - EC2 - instance

AMI → { OS, Database, ...; libraries, dependencies, ... }

- ssh into EC2 - instance
- run your code remotely

APPLIED ROOTS

Heroku [PaaS]



Heroku simplifies app-dev

→ requirements.txt

→ Procfile

→ Autodetect the app type

→ git push & start running

→ [550 + 450] hrs of free runtime per month

→ autosleeping

- Web dynos vs worker dynos
- Dyno types: https://devcenter.heroku.com/articles/dyno-types
- Slug: prepacked & compressed form of our app
- Databases → Postgres, Redis, ...
- Aws - Paas: Elastic Beanstalk

Python Virtual env

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]
```

APPLIED ROOTS

Flask - API + Gunicorn

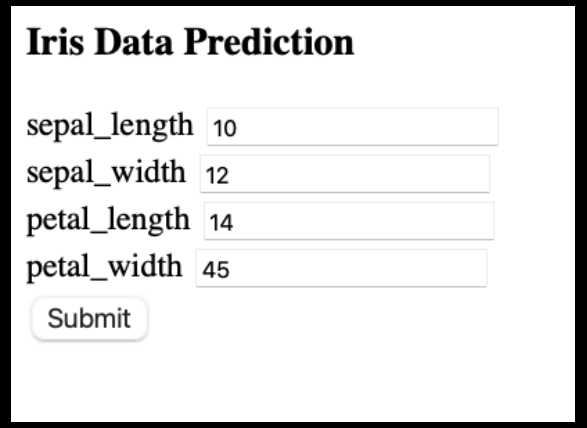
In the VirtualEnv, install packages that we want to train the model and run Flask-API locally

```
$ 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 ------>



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

APPLIED ROOTS

Required Python packages

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

→ from current virtualenv

Create Procfile

Procfile →

```
web: gunicorn app:app
```

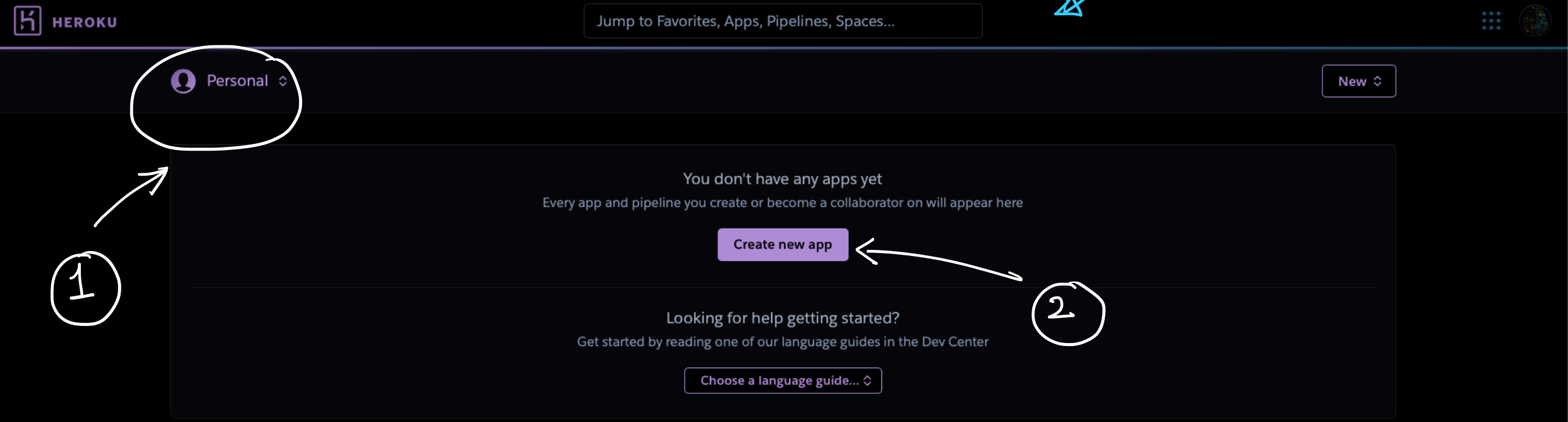
IRIS folder

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

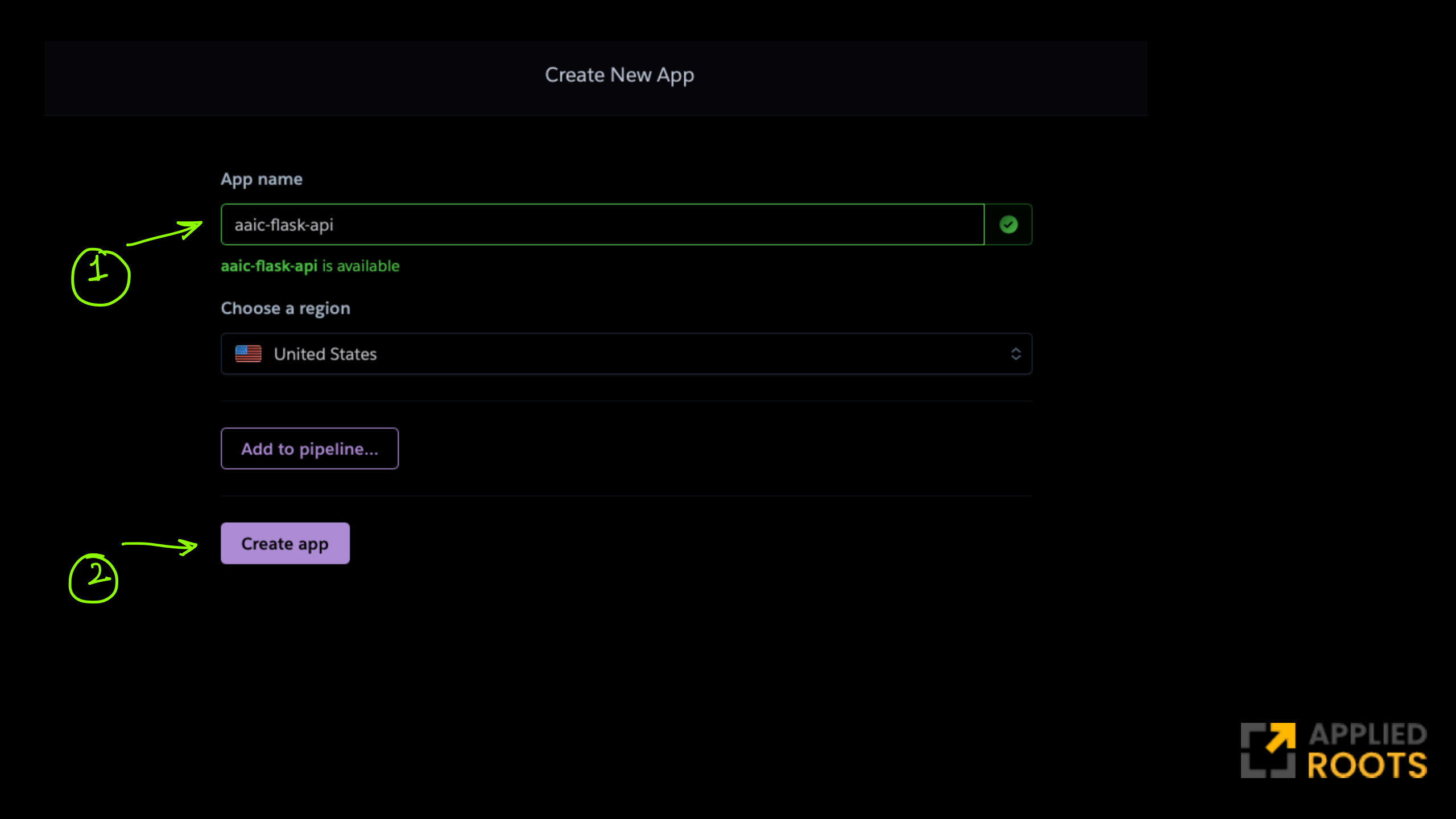
APPLIED ROOTS

Heroku.com : signup & login

Dashboard



APPLIED ROOTS


Create New App
App name
aaic-flask-api
aaic-flask-api is available
Choose a region
United States
Add to pipeline...
Create app
1
2

Personal > aaic-flask-api

Open app | More

Overview | Resources | Deploy | Metrics | Activity | Access | Settings

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

Heroku Git
Use Heroku CLI

GitHub
Connect to GitHub

Container Registry
Use Heroku CLI

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
```

APPLIED ROOTS

Heroku - CLI

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

- install Git
- install Heroku - CLI

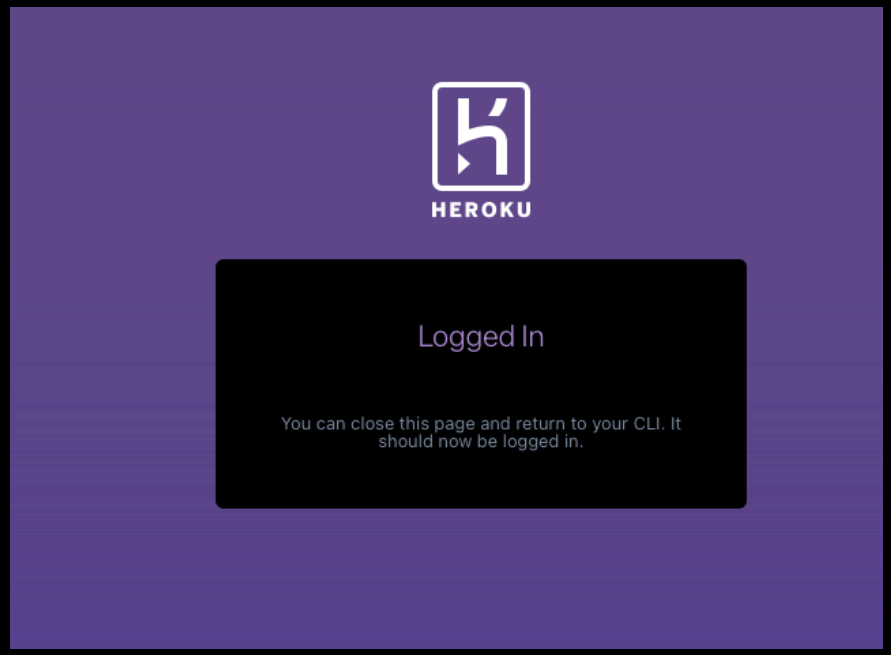
OS - dependent

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:
```

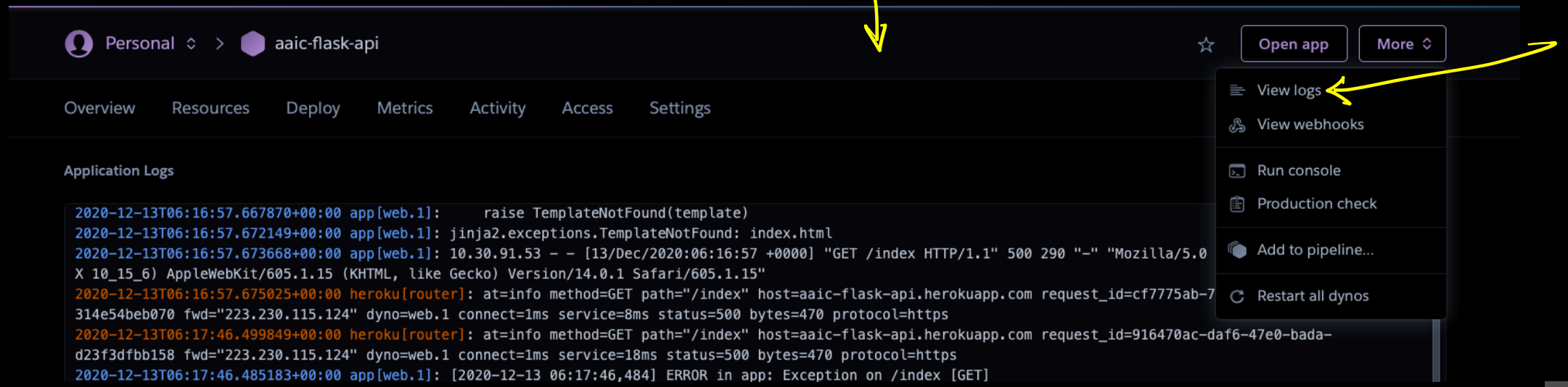


APPLIED ROOTS

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



APPLIED ROOTS

Heroku app folder-structure

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

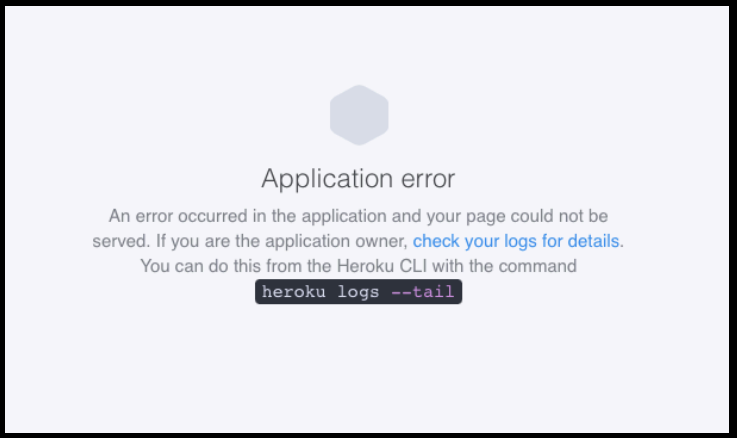
Heroku stop-app

```
$ 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



APPLIED ROOTS

Streamlit on Heroku:

Download code from https://drive.google.com/file/d/16RHjtlMlJKymma7tiu3jwqFtG7ngH0UI/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
```

APPLIED ROOTS

Q&A

APPLIED ROOTS