API 102 (Python3)

DAT-A SMASH! #02:
"API Machine Learning Deployment with Python and Cloud Platform"

SIVAPHONG NIYOMPHANICH

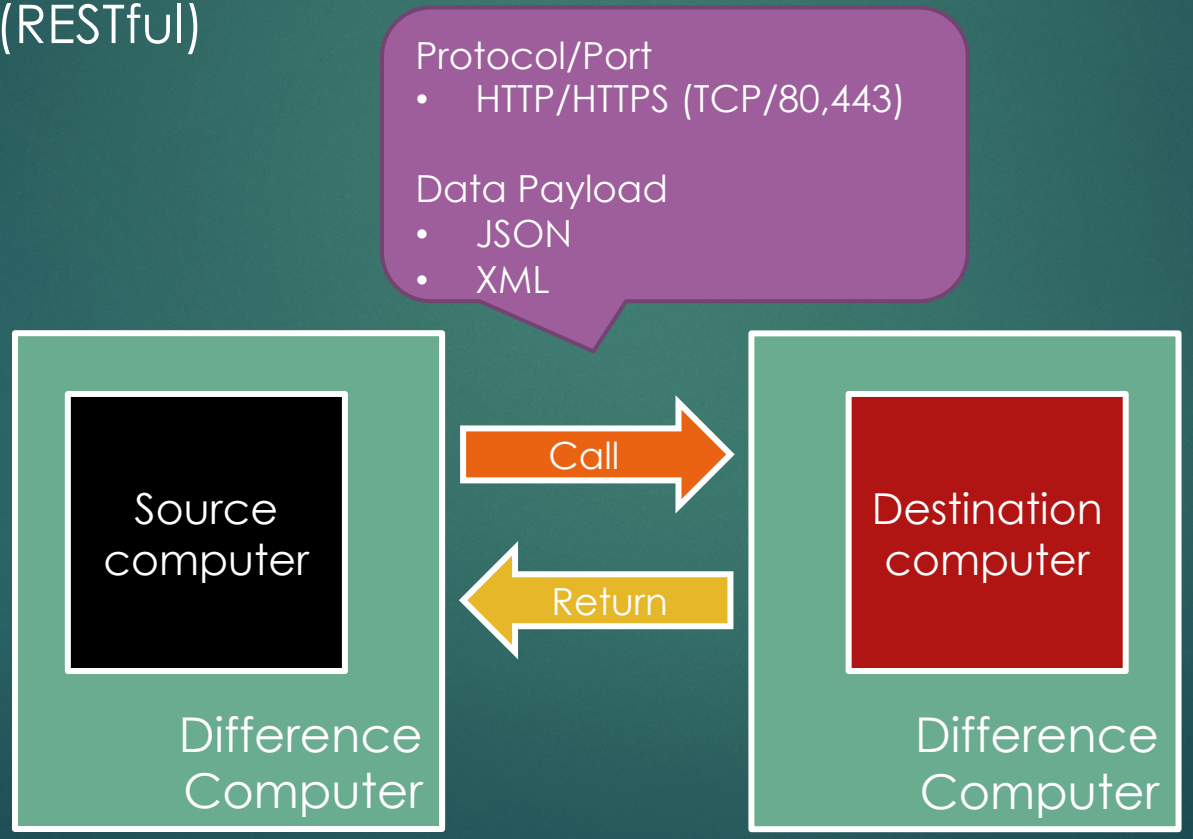
Agenda

- API Overview
- Python with Machine Learning (Train model and save model)
- API with Python (Using saved model and conduct API)
- Flask Framework
- Heroku and Cloud Platform

- Image Parameter API
- Trip & Trick for API (Speed and Problem)

Web API

- Exchange data between two or more objects by using HTTP protocol (RESTful)



Understanding HTTP Protocol

- HTTP = Hyper Text Transfer Protocol
- HTTPS = Secure using SSL/TLS

- HTTP Request Method
  - GET
  - POST
  - PUT
  - DELETE
  - PATCH

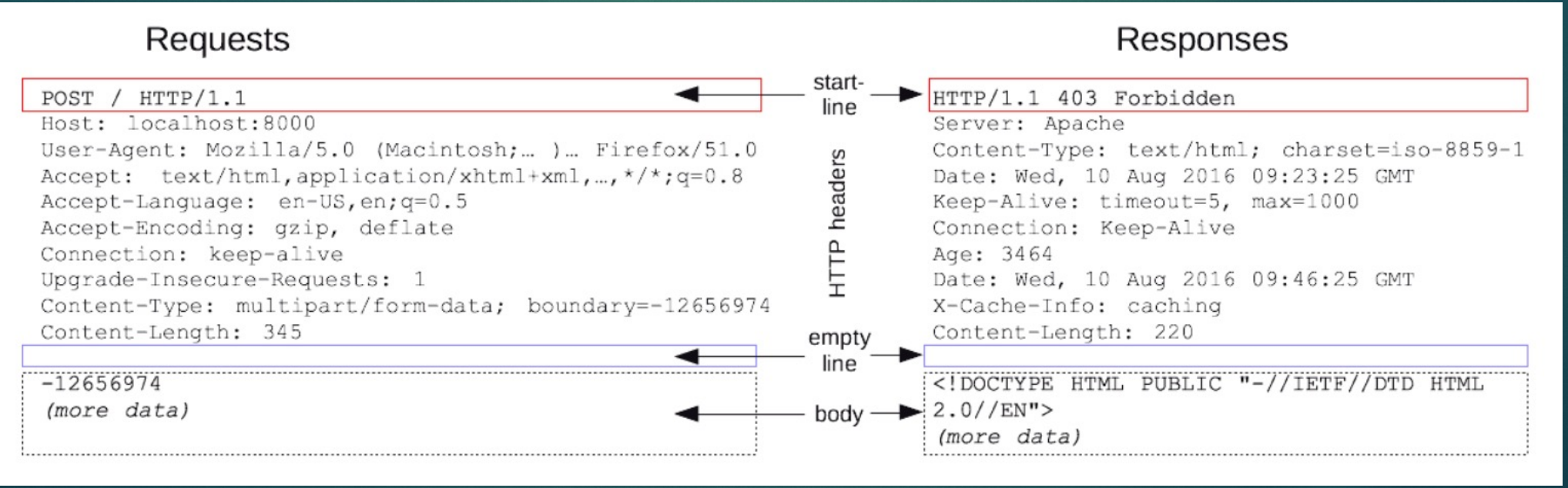
- HTTP = Hyper Text Transfer Protocol
- HTTPS = Secure using SSL/TLS

- HTTP Response Code
  - 1XX
  - 2XX
  - 3XX
  - 4XX
  - 5XX

https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

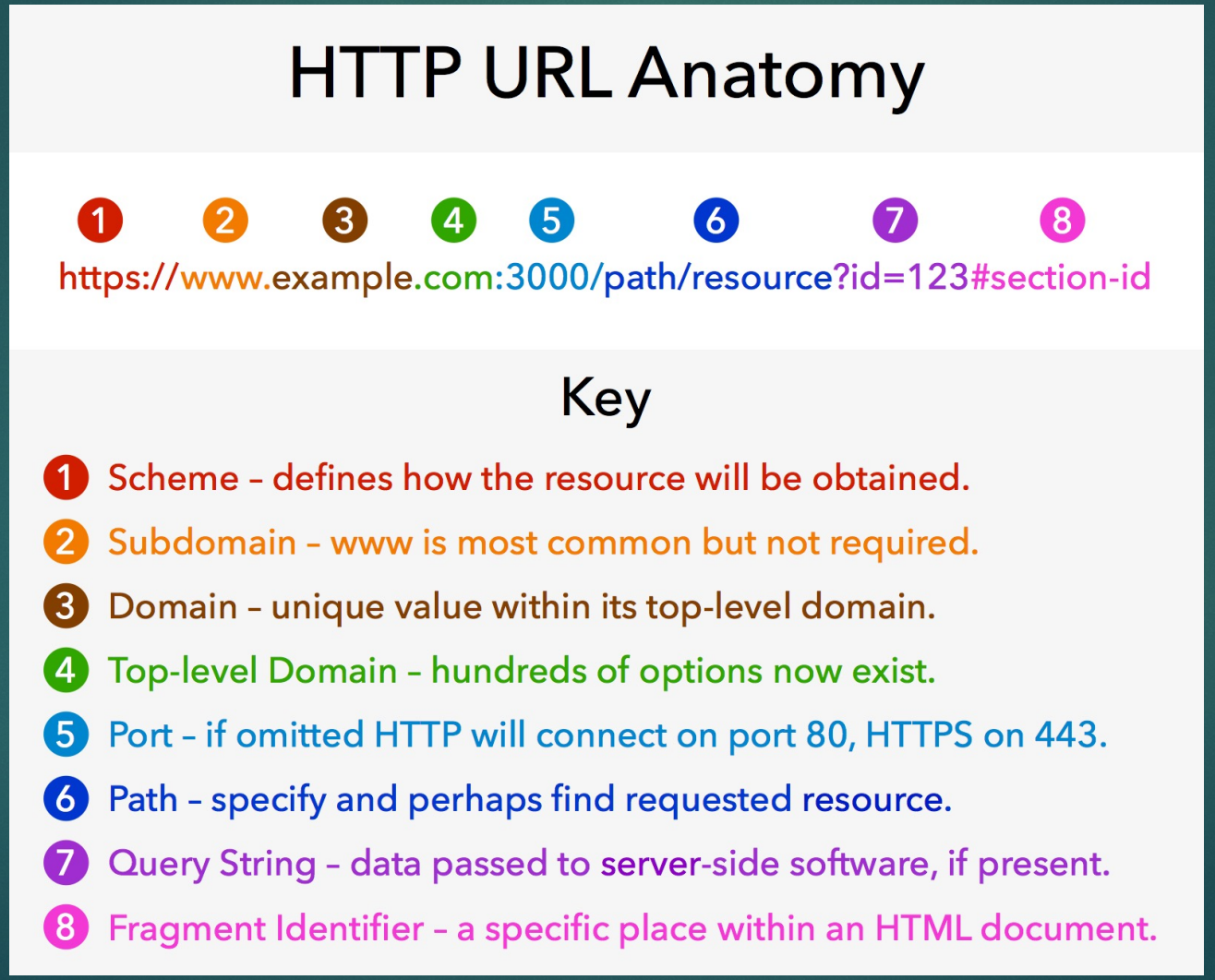
Understanding HTTP Protocol

- Payload (HTTP Message)
  - HTTP Header
  - HTTP Body



https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages

Understanding HTTP Protocol

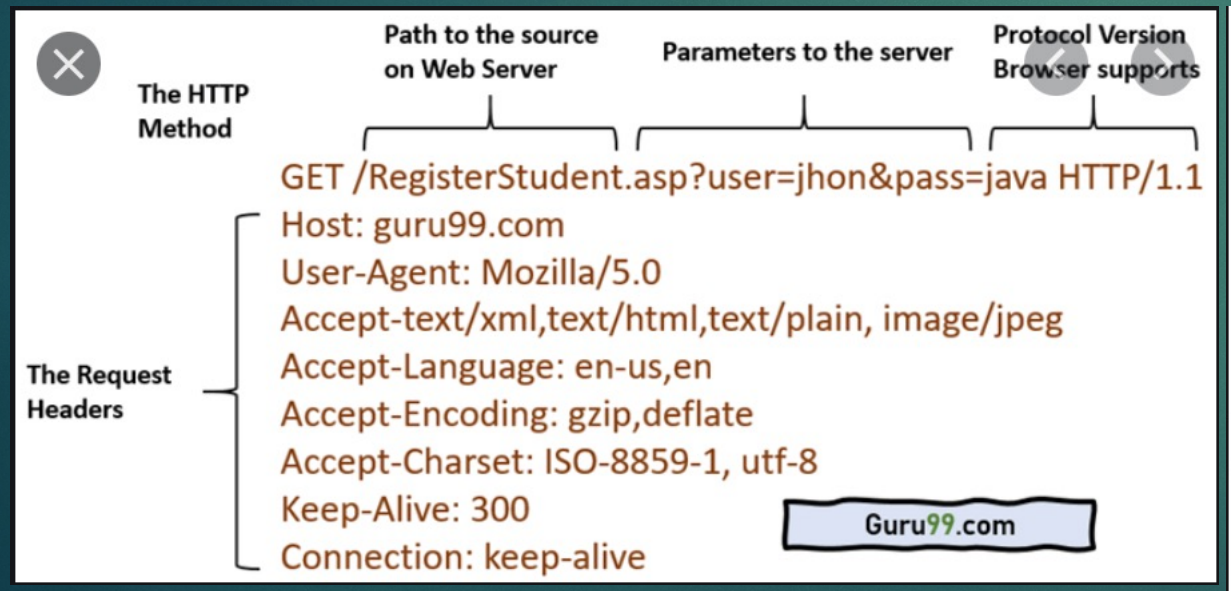


https://cascadingmedia.com/insites/2015/02/url-anatomy-seo-tips.html

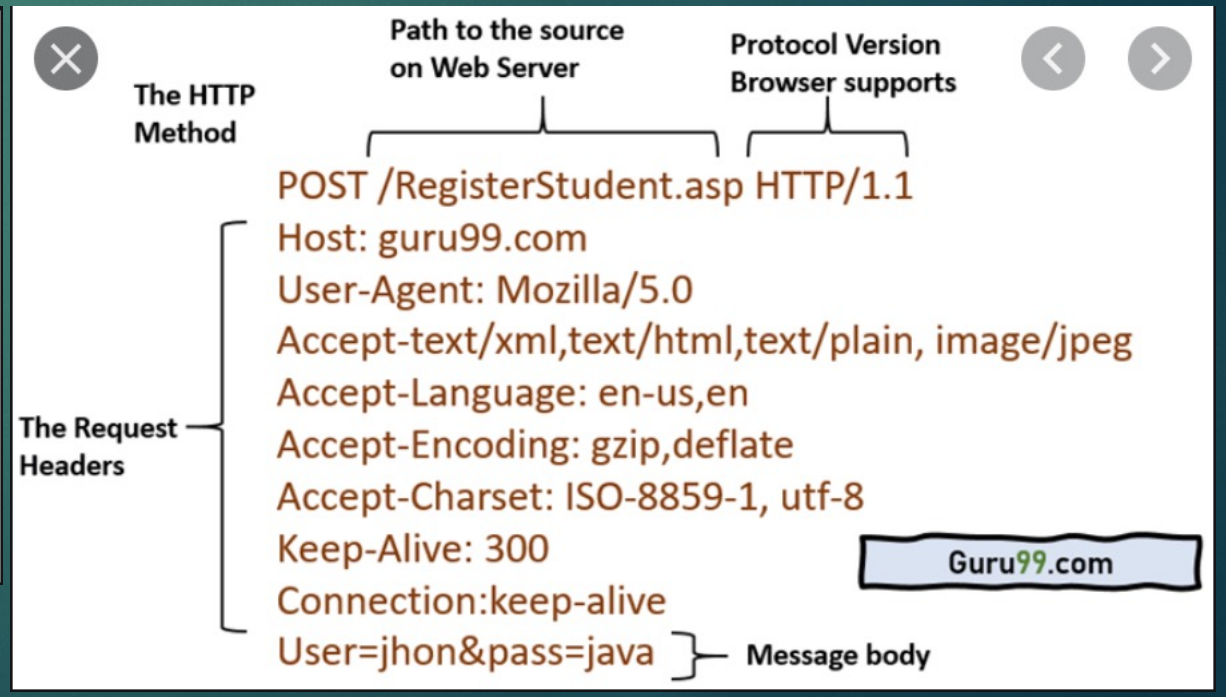
Understanding HTTP Protocol

- GET vs POST method

http://guru99.com/RegisterStudent.asp?user=jhon&pass=java



http://guru99.com/RegisterStudent.asp



First calling Web API

- Most helpful tool
- Postman
  - https://www.postman.com

- API
  - https://api.agify.io/?name=bella
- Keyword
  - URL
  - HTTP Method
  - Parameter/Payload (Both Request and Response)
  - Is need subscriber (e.g. token, key)

Python Machine Learning

- Regression
- Nueral Network

Preparation

- Create environment
  - Environment for ML development (e.g. MLDev)
  - Environment for API deployment (e.g. APIDep)

Preparation Development

- Install required library on ML development environment
  - pip install scikit-learn
  - pip install numpy
  - pip install matplotlib
  - pip install pickle
  - *pip install keras*

Let developing

```
# Linear Regression
```

```
#Step 1: Import packages and classes
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import pickle
```

```
#Step 2: Provide data
x = np.array([5, 15, 25, 35, 45, 55]).reshape((-1, 1))
y = np.array([5, 20, 14, 32, 22, 38])
```

```
print(x)
print(y)
```

```
plt.plot(x, y)
plt.show()
```

```
#Step 3: Create a model and fit it
model = LinearRegression()
model.fit(x, y)
```

```
#Step 4: Get results
r_sq = model.score(x, y)

print('coefficient of determination:', r_sq)
print('intercept:', model.intercept_)
print('slope:', model.coef_)
```

```
model.predict([[9]])
```

```
#Step 5: Predict response
y_pred = model.predict(x)
```

```
plt.plot(x, y, x, y_pred)
plt.show()
```

```
#Step 6: Save model to file
savefilename = 'saved_model.sav'
pickle.dump(model, open(savefilename, 'wb'))
```

```
#Step 7: Using model on other system
import pickle
loadfilename = 'saved_model.sav'
loaded_model = pickle.load(open(loadfilename, 'rb'))
```

```
loaded_model.predict([[9]])
```

Save your Machine Learning model

- Many ways to save and load ML model
  - pickle
  - joblib
  - Native
    - E.g. Keras

Preparation Deployment

- Install required library on API deployment environment
  - $ pip install scikit-learn
  - $ pip install pickle
  - $ pip install flask
  - $ pip install gunicorn
  - $ pip install markupsafe
- Prepared Cloud platform or something related
  - Heroku
- Model file
  - saved_model.sav

app.py

- Create file name app.py

app.py > ...

```python
from flask import Flask, jsonify, request
from markupsafe import escape
import pickle

loadfilename = 'saved_model.sav'
loaded_model = pickle.load(open(loadfilename, 'rb'))

```

```python
@app.route('/predict/<myparam>')
def pred_regression(myparam):
    pred = loaded_model.predict([[myparam]])
    return str(pred)

```

app.py – Full of code


```
from flask import Flask, jsonify, request
from markupsafe import escape
import pickle

loadfilename = 'saved_model.sav'
loaded_model = pickle.load(open(loadfilename, 'rb'))

app = Flask(__name__)

@app.route('/')
def hello():
    return "Hello Flask-Heroku"

@app.route('/predict/<myparam>')
def pred_regression(myparam):
    pred = loaded_model.predict([[myparam]])
    return str(pred)

if __name__ == "__main__":
    app.run(debug=False)
```

Procfile

- Create file named Procfile
  - Tell the (cloud) how to run this API

```
Procfile
1    web: gunicorn app:app
```

requirements.txt

- Generate file named requirements.txt
  - Tell the (cloud) which packages/library that required to running this API

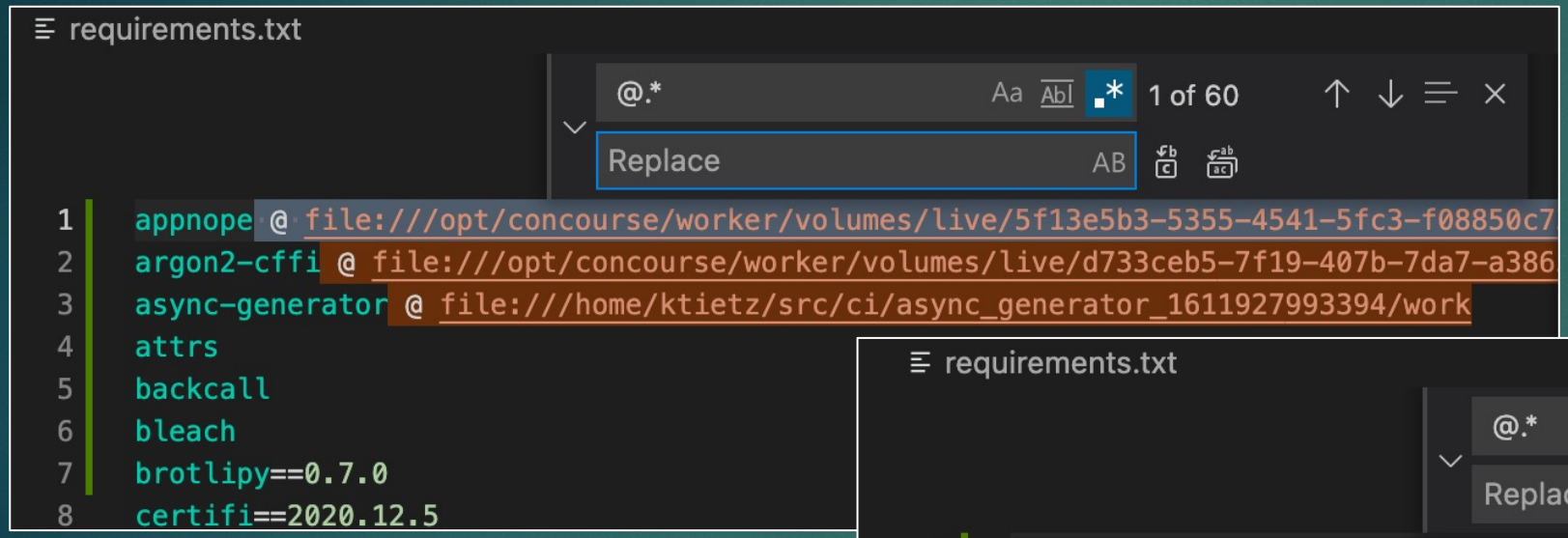
- $ pip freeze > PATH-TO-APP.PY-DIR/requirements.txt

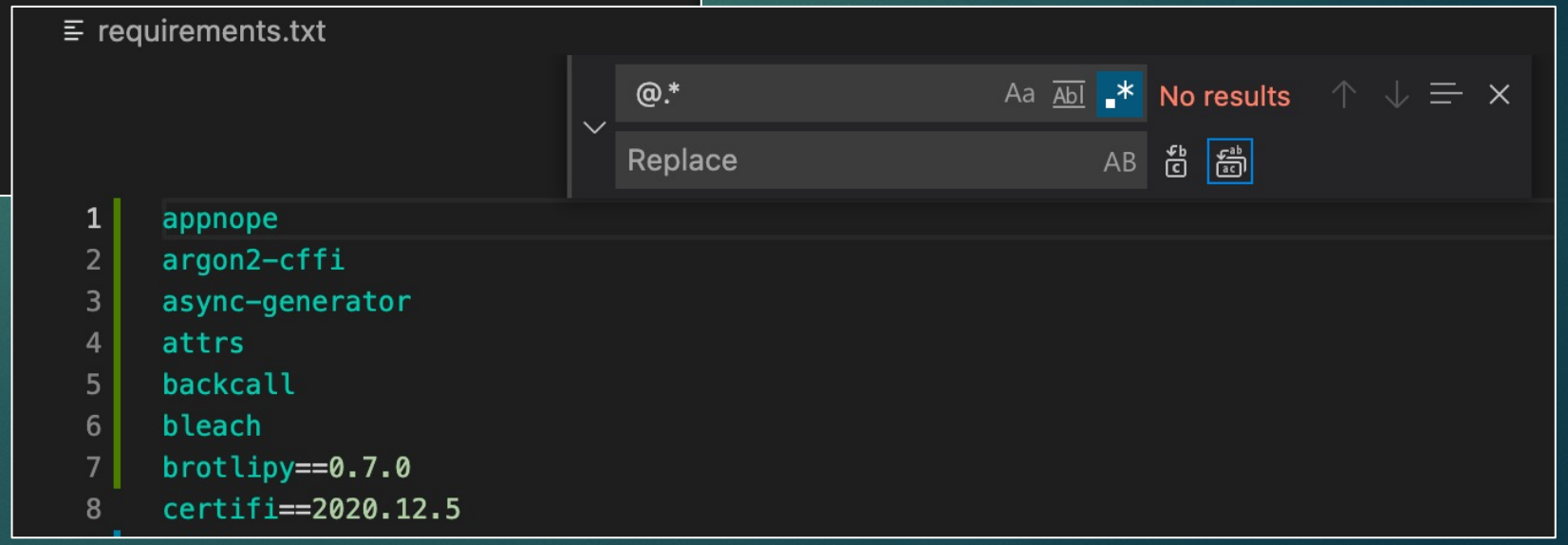
requirements.txt

```
appnope @ file:///opt/concourse/worker/volumes/live/5f13e5b3-5355-4541-5fc3-f08850c7
argon2-cffi @ file:///opt/concourse/worker/volumes/live/d733ceb5-7f19-407b-7da7-a386
async-generator @ file:///home/ktietz/src/ci/async_generator_1611927993394/work
attrs @ file:///tmp/build/80754af9/attrs_1604765588209/work
backcall @ file:///home/ktietz/src/ci/backcall_1611930011877/work
bleach @ file:///tmp/build/80754af9/bleach_1612211392645/work
brotlipy==0.7.0
certifi==2020.12.5
cffi @ file:///opt/concourse/worker/volumes/live/0ef369cc-6ba0-47e7-75da-208c6400381
chardet @ file:///opt/concourse/worker/volumes/live/c798b2ee-88b1-4341-6830-161a92c2
click @ file:///home/linux1/recipes/ci/click_1610990599742/work
cryptography @ file:///opt/concourse/worker/volumes/live/c515855a-effc-46df-74dc-542
decorator @ file:///tmp/build/80754af9/decorator_1617916966915/work
defusedxml @ file:///tmp/build/80754af9/defusedxml_1615228127516/work
entrypoints==0.3
Flask @ file:///home/ktietz/src/ci/flask_1611932660458/work
gunicorn @ file:///opt/concourse/worker/volumes/live/f958ac2d-63e5-4751-6df7-f751b63
```

requirements.txt

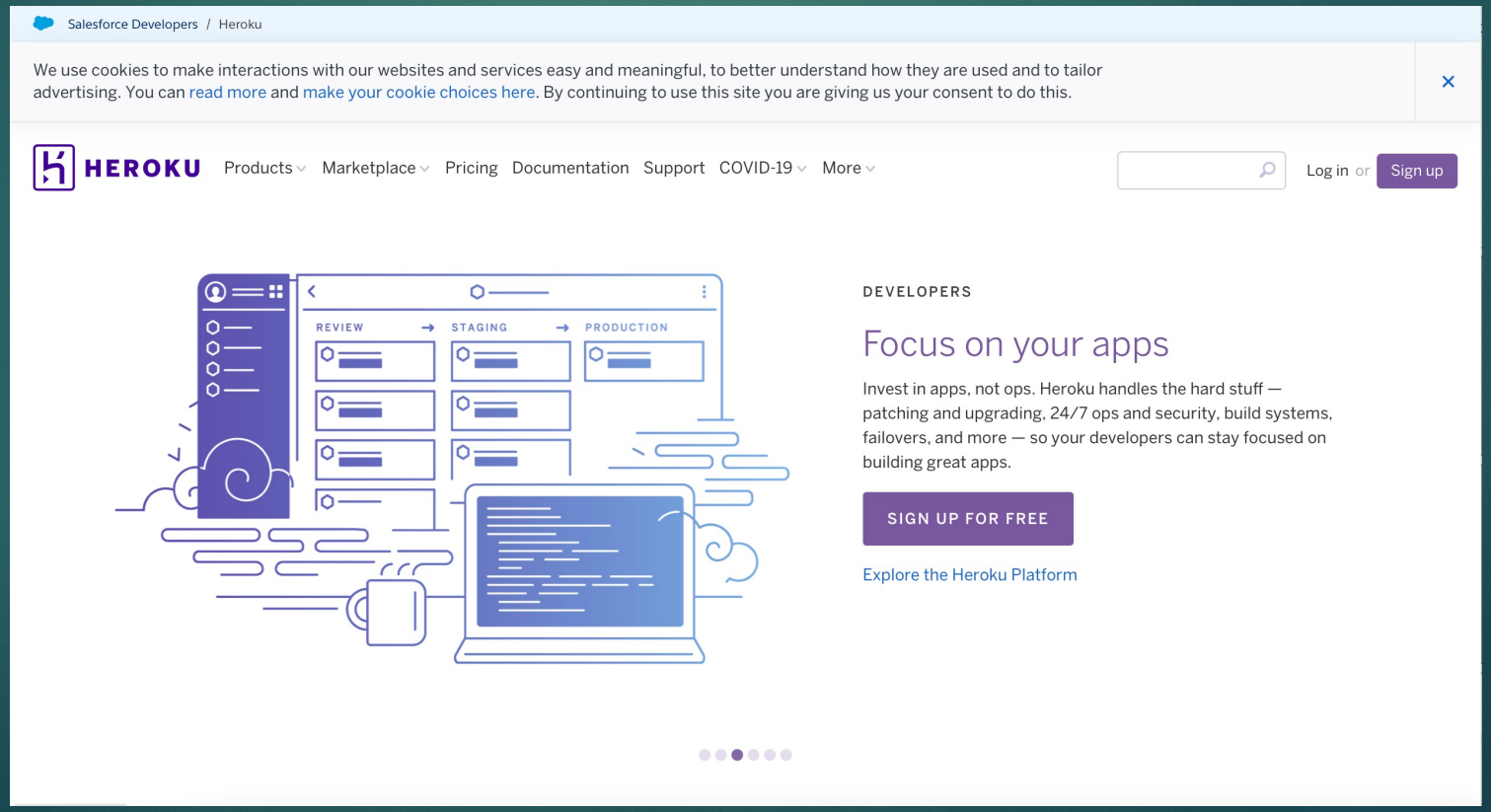
- Edit content in file requirements.txt (@.*)

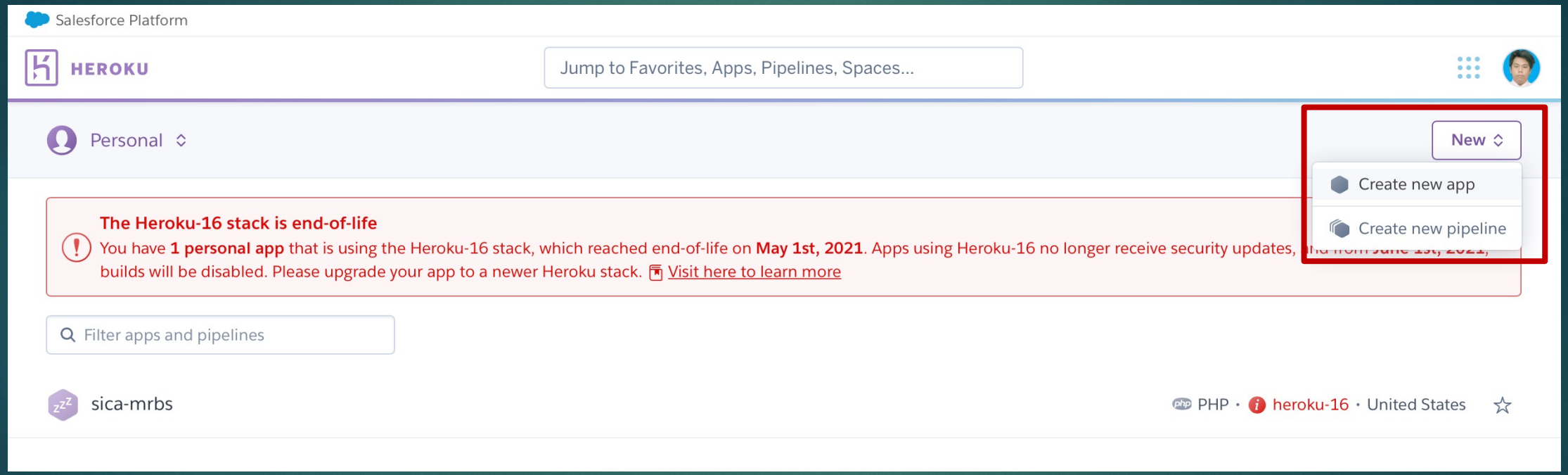


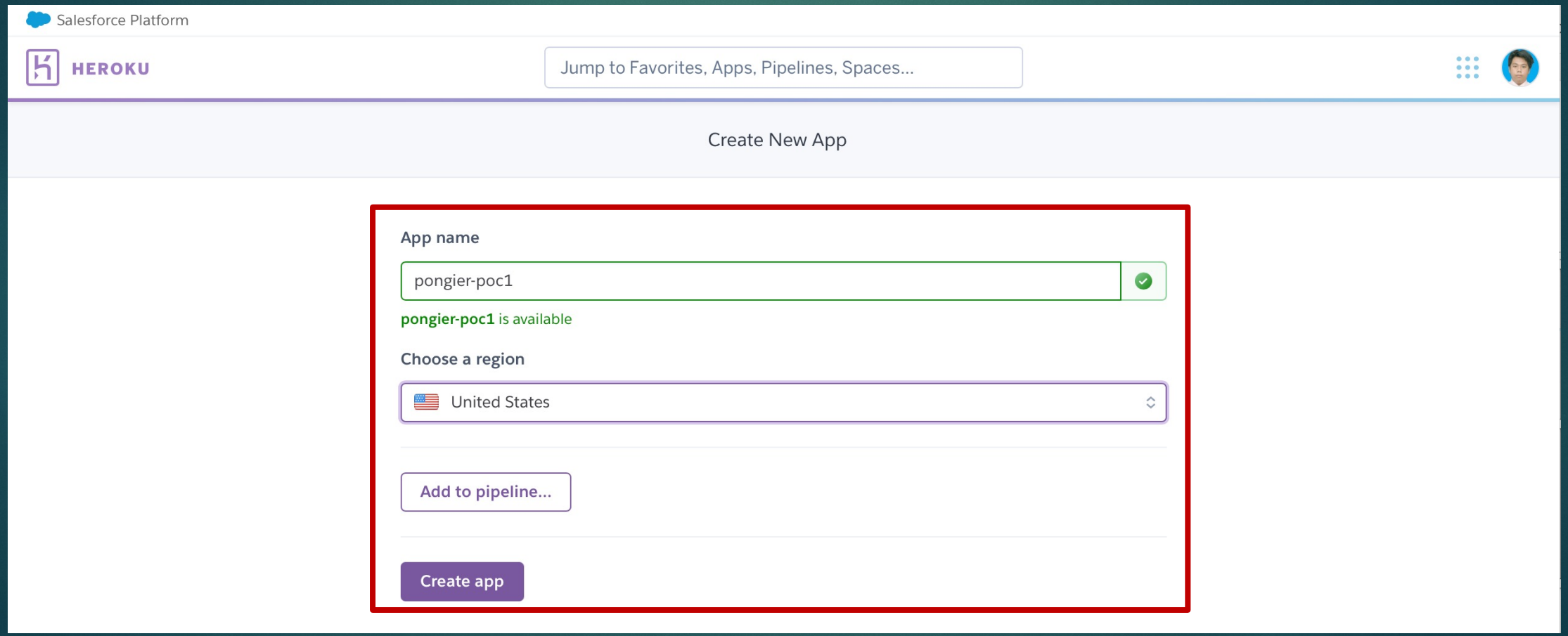


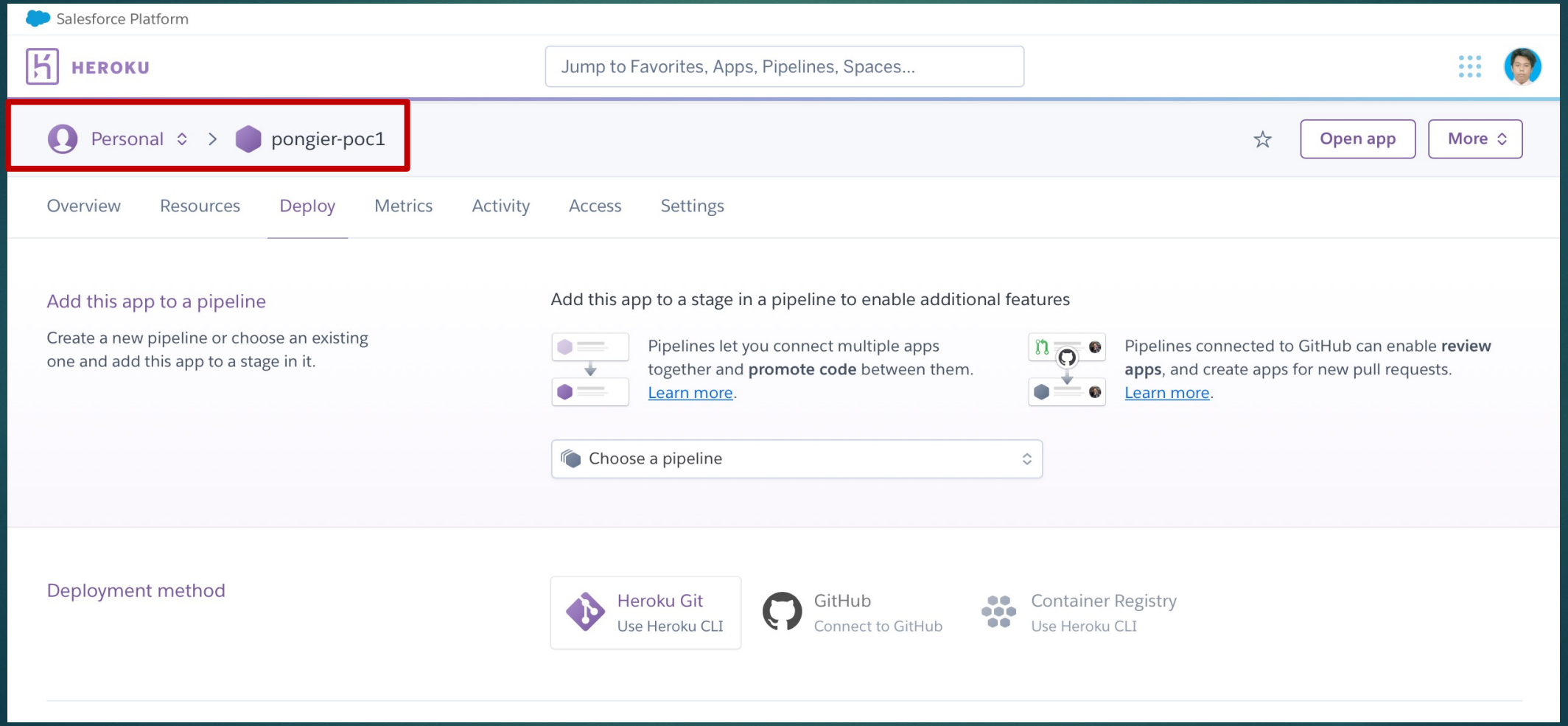
Preparing Heroku

- Register to Heroku
  - https://www.heroku.com




Salesforce Platform
HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...
Personal
New
Create new app
Create new pipeline
The Heroku-16 stack is end-of-life
You have 1 personal app that is using the Heroku-16 stack, which reached end-of-life on May 1st, 2021. Apps using Heroku-16 no longer receive security updates,
builds will be disabled. Please upgrade your app to a newer Heroku stack. Visit here to learn more
Filter apps and pipelines
sica-mrbs
PHP · heroku-16 · United States


Salesforce Platform
HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...
Create New App
App name
pongier-poc1
pongier-poc1 is available
Choose a region
United States
Add to pipeline...
Create app


Salesforce Platform
HEROKU
Jump to Favorites, Apps, Pipelines, Spaces...
Personal
pongier-poc1
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.
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
Heroku Git
Use Heroku CLI
GitHub
Connect to GitHub
Container Registry
Use Heroku CLI

Some Related Tools

- https://devcenter.heroku.com/articles/heroku-command-line
  - Heroku Client Tool
- https://git-scm.com
  - Git Tool

Git command

- Goto PATH-TO-APP.PY-DIR

- $ cd PATH-TO-APP.PY-DIR

- $ git init .
- $ git add app.py Procfile requirements.txt saved_model.sav
- $ git commit -a -m "first commit"

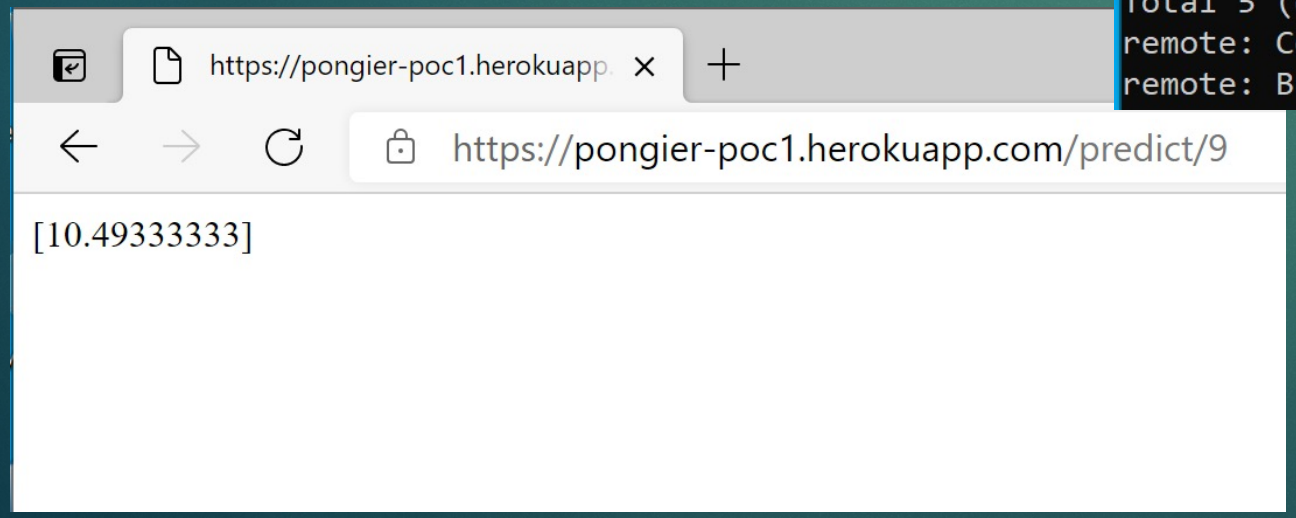
Heroku command

- $ heroku login -i or heroku login
- $ heroku git:remote -a {your-project-name}
- $ git push heroku master

```
Z:\API02>git push heroku master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.38 KiB | 176.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
```

```
remote: -----> Discovering process types
remote:        Procfile declares types -> web
remote:
remote: -----> Compressing...
remote:        Done: 149M
remote: -----> Launching...
remote:        Released v4
remote:        https://pongier-poc1.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/pongier-poc1.git
   5963570..8dde213  master -> master
```

- Testing…



Next Time

- $ git add FILE_IF_WANT
- $ git commit -a -m "YOUR COMMENT"
- $ git push heroku master

```
Z:\API02>git add saved_model.sav

Z:\API02>git commit -a -m "Predict API R1"
[master 3205d70] Predict API R1
 3 files changed, 73 insertions(+)
 create mode 100644 saved_model.sav

Z:\API02>git push heroku master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.38 KiB | 176.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
```

Trip & Trick

- Popular Problem on deployment
  - Library conflict
- CPU/RAM/GPU capability
  - CPU overload
  - RAM full
  - GPU required
- Speed improvement on Code Thinking
  - Lode model on starting

Some popular error – Library conflict


```
remote:          Downloading MarkupSafe-2.0.1-cp39-cp39-manylinux2010_x86_64.whl (30 kB)
remote:        Collecting mistune
remote:          Downloading mistune-0.8.4-py2.py3-none-any.whl (16 kB)
remote:        ERROR: Could not find a version that satisfies the requirement mkl-fft==1.3.0 (from -r /tmp/build_5d20fc3
7/requirements.txt (line 33)) (from versions: none)
remote:        ERROR: No matching distribution found for mkl-fft==1.3.0 (from -r /tmp/build_5d20fc37/requirements.txt (l
ine 33))
remote:  !     Push rejected, failed to compile Python app.
remote:
remote:  !     Push failed
remote: Verifying deploy...
remote:
remote: !       Push rejected to pongier-poc1.
remote:
To https://git.heroku.com/pongier-poc1.git
 ! [remote rejected] master -> master (pre-receive hook declined)
error: failed to push some refs to 'https://git.heroku.com/pongier-poc1.git'
```

- In this case
- Remove all **mkl** library

requirements.txt

```
28  jupyter-client
29  jupyter-core
30  jupyterlab-pygments
31  MarkupSafe
32  mistune
33  nbclient
34  nbconvert
```

requirements.txt

```
31  MarkupSafe
32  mistune
33  mkl-fft==1.3.0
34  mkl-random
35  mkl-service==2.3.0
36  nbclient
37  nbconvert
```

CPU/RAM/GPU capability

- API crash on th fly
- Some free tire cloud dose not meet ML capability requirement
  - E.g. Image classification model
  - E.g. Some algorithm required GPU for working
- CPU overload
- RAM full
- GPU required

Speed improvement on Code Thinking

Pongier
21 May
2021

- Which the better code

```python
from flask import Flask, jsonify, request
from markupsafe import escape
import pickle

loadfilename = 'saved_model.sav'
loaded_model = pickle.load(open(loadfilename, 'rb'))
```

```python
@app.route('/predict/<myparam>')
def pred_regression(myparam):
    pred = loaded_model.predict([[myparam]])
    return str(pred)
```

```python
@app.route('/predict/<myparam>')
def pred_regression(myparam):
    loadfilename = 'saved_model.sav'
    loaded_model = pickle.load(open(loadfilename, 'rb'))
    pred = loaded_model.predict([[myparam]])
    return str(pred)
```

Image Parameter API


```python
import imghdr
import os
from flask import Flask, render_template, request, redirect, url_for, abort, \
    send_from_directory
from werkzeug.utils import secure_filename

app = Flask(__name__)
app.config['MAX_CONTENT_LENGTH'] = 2 * 1024 * 1024
app.config['UPLOAD_EXTENSIONS'] = ['.jpg', '.png', '.gif']
app.config['UPLOAD_PATH'] = 'uploads'

def validate_image(stream):
    header = stream.read(512)
    stream.seek(0)
    format = imghdr.what(None, header)
    if not format:
        return None
    return '.' + (format if format != 'jpeg' else 'jpg')

@app.errorhandler(413)
def too_large(e):
    return "File is too large", 413
```

```python
@app.route('/')
def index():
    files = os.listdir(app.config['UPLOAD_PATH'])
    return render_template('index.html', files=files)

@app.route('/', methods=['POST'])
def upload_files():
    uploaded_file = request.files['file']
    filename = secure_filename(uploaded_file.filename)
    if filename != '':
        file_ext = os.path.splitext(filename)[1]
        if file_ext not in app.config['UPLOAD_EXTENSIONS'] or \
                file_ext != validate_image(uploaded_file.stream):
            return "Invalid image", 400
        uploaded_file.save(os.path.join(app.config['UPLOAD_PATH'], filename))
    return '', 204

@app.route('/uploads/<filename>')
def upload(filename):
    return send_from_directory(app.config['UPLOAD_PATH'], filename)
```

Q&A

- A&Q

Example deploy to Heroku

- https://stackpython.co/tutorial/flask-heroku-deploy-python
- https://www.digitalocean.com
- https://azure.microsoft.com/en-us/