# Advanced Programming with Python

## Deploying Python apps

Pepe García jgarciah@faculty.ie.edu

# Deployment

When we talk about deployment, we mean running our application somewhere in the internet.

In this case we will run it on **Google App Engine**.

# Google App Engine

Google App Engine (**GAE**) is a PaaS Platform by Google Cloud. We can host applications written in a range of different programming languages, but we'll focus on Python ourselves. It's similar to other services such as **Heroku**, **Elastic Beanstalk**, **Engineyard**.


Figure 1: GAE logo

# gcloud

We will interact with Google Cloud using the **gcloud** sdk. It's a command line interface that provides access to all features in Google Cloud.

https://cloud.google.com/sdk/gcloud

# Practice

In this practice we will create a toy Flask application and deploy it to Google App Engine.

# Practice

## steps

- Create a project in `gcloud`
- Create a simple flask application
- Create needed files
- Deploying the app in Google App Engine

# Practice

## steps

1. **Create a project in `gcloud`**
2. Create a simple flask application
3. Create needed files
4. Deploying the app in Google App Engine

# Practice. Create project

We can have more than one project under the same Google account running on Google Cloud. We tell them appart using projects.

# Practice. Create project

```
$ gcloud projects create mcsbt-app-session-13-pepegar
```

# Practice. Create project

```
$ gcloud projects create mcsbt-app-session-13-pepegar
```

**Attention**

the id we give to the project must be unique

# Practice. Create project

```
$ gcloud projects list
PROJECT_ID                      NAME                    PROJ
mcsbt-app-session-13-pepegar    mcsbt-flask-example-1   4217
```

# Practice. Create project

We'll also configure the project as the current one:

```
$ gcloud config set project mcsbt-app-session-13-pepegar
```

# Practice

## Checkpoint

At this point we should have:

```
$ gcloud projects list
PROJECT_ID                          NAME                    PROJ
mcsbt-app-session-13-pepegar  mcsbt-flask-example-1  4217

$ gcloud config get-value project
mcsbt-flask-example-1
```

# Practice

## steps

Create a project in `gcloud`

**Create a simple flask application**

Create needed files

Deploying the app in Google App Engine

# Practice. Flask application

Now, let's create a **very simple** Flask app (`main.py`):

# Practice. Flask application

Now, let's create a **very simple** Flask app (`main.py`):

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def index():
    return "Hello, Appengine!"

# app.run()
```

# Practice. Flask application

Now, let's create a **very simple** Flask app (`main.py`):

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def index():
    return "Hello, Appengine!"

# app.run()
```

**Attention**

appengine will look for a variable named app and run it, we must not do it manually.

# Practice

## Checkpoint

Now there should be a **`main.py`** file in our folder, **without the `app.run()`**. Remember that **GAE** is gonna look for the variable called app for us, and run it. We won't need to run it manually.

```
$ ls
main.py
```

# Practice

## steps

Create a project in `gcloud`
Create a simple flask application
**Create needed files**
Deploying the app in Google App Engine

# Practice. Creating the needed files

There are two files we will need to create in our project in order to make it run on **GAE**:

# Practice. Creating the needed files

There are two files we will need to create in our project in order to make it run on **GAE**:

`requirements.txt`
`app.yaml`

# Practice. Creating the needed files

**`requirements.txt`** describes the dependencies of our application and their versions. In our case, it's only going to be Flask.

```
Flask==1.1.2
```

# Practice. Creating the needed files

`app.yaml` declares information needed for Google App Engine. In our case, we'll just specify the version of python we need.

```
runtime: python39
```

# Practice

## Checkpoint

```
$ ls
app.yaml                main.py                requirements.txt
```

# Practice

## steps

- Create a project in `gcloud`
- Create a simple flask application
- Create needed files
- **Deploying the app in Google App Engine**

# Practice. Deploying our app to GAE

```
$ gcloud app create

...

Please enter your numeric choice:  8

Creating App Engine application in project [pepegar-test-
Success! The app is now created. Please use `gcloud app d
```

# Practice. Deploying our app to GAE

```
$ gcloud app deploy

...

Do you want to continue (Y/n)?  Y

Beginning deployment of service [default]...

  Uploading 3 files to Google Cloud Storage

File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://pepegar-test-1.ew.

You can stream logs from the command line by running:
```

# Practice. Deploying our app to GAE

**Attention**

With a real GCP account, last command may fail because of Google Cloud Build, you'll need to activate it.