CMPUT 404 Lab 6

9-10th February 2016

OVERVIEW

Learn how to host a Django application on a PaaS (Openshift, Heroku). Learn about the other cloud application PaaS services available (AWS, Windows Azure, Google App Engine)

OPENSHIFT STEPS

1. Steps are at the openshift-django repository: https://github.com/awwong1/openshift-django
2. Sign up for a free account at https://openshift.com
3. Click create your first application, select Python 2.7 (or Python 3)
4. Set your namespace and application name, click “Create Application” (This may take 5-10 minutes)
5. Install the openshift cli tools

```
gem install rhc
# if on lab machine
# gem install --user-install rhc
# Add the following to your path
# /cshome/<csid>/.gem/ruby/1.9.1/bin
```

6. Log into your openshift account from terminal

```
rhc setup
```

7. Clone your application locally to your workspace

```
rhc apps
rhc git-clone <app-name>
```

8. Add the database cartridge to your application

```
rhc add-cartridge postgresql-9.2 --app <app-name>
OR
rhc add-cartridge mysql-5.5 --app <app-name>
```

9. Add the django seed repository as the upstream repo

```
cd django
git remote add upstream -m master
https://github.com/awwong1/openshift-django.git
git pull -s recursive -X theirs upstream master
```

10. Set the WSGI application to be Django's built in WSGI application

```
rhc env set OPENSHIFT_PYTHON_WSGI_APPLICATION=wsgi/wsgi.py
--app <app-name>
```

11. Push the repo upstream

```
git push
```

12. SSH into the application to create a django superuser.

```
python app-root/repo/manage.py createsuperuser
```

13. Now use your browser to connect to the Admin site.
14. Add the repository from Abram's fork for CMPUT 404 to your github remote

```
git remote add github
https://github.com/<github-username>/CMPUT404-project-socialdis
tribution
```

15. Pull the remote contents, handle the merge conflicts.

HEROKU STEPS

1. Official steps are at the devcenter for Heroku: https://devcenter.heroku.com/articles/getting-started-with-python#introduction
2. Sign up for a free account at https://heroku.com
3. Download the heroku dev tools (ensuring Ruby is installed)

```
#Not supported on lab machines, as they require su
wget -O- https://toolbelt.heroku.com/install-ubuntu.sh | sh
wget -O- https://toolbelt.heroku.com/install.sh | sh

# Lab machine supported, but not 'official'
gem install --user-install heroku
```

```
# Add the following to your path
# /cshome/<csid>/.gem/ruby/1.9.1/bin
```

4. Log in using the heroku cli

   ```
   heroku login
   ```

5. Clone the heroku starter application

   ```
   git clone https://github.com/heroku/python-getting-started.git
   ```

6. Create an app on heroku after navigating within the newly created project

   ```
   cd python-getting-started
   heroku create
   ```

7. Push the code to the newly created application

   ```
   git push heroku master
   ```

8. Ensure one instance of your application is running

   ```
   heroku ps:scale web=1
   ```

9. Open your application

   ```
   heroku open
   ```

QUESTIONS

Question 1:

What does WSGI stand for? What does it do?

Question 2:

What does PaaS stand for?

Question 3:

What are some of the benefits to using a PaaS to host your applications? What are some of the drawbacks?

Question 4:

List three different PaaS vendors. Also, specify which vendor are you likely going to use for your CMPUT project. (self hosted, Heroku, OpenShift, etc.)