Webex Teams Hackathon 2018

Lab4 – Using Heroku Platform to Deploy Python REST API Back End

Objectives

In this lab, you will complete the following objectives:

- create GitHub repo for your project
- register Heroku account
- create an app in Heroku, connect to the GitHub repo
- start deployment manually
- test your running back end on Heroku
- enable automatic deploys
- commit and push the project changes to test autodeploy

Background / Scenario

For developing application which use Webex Teams bot capabilities should be accessed directly on the Internet.

Heroku (https://heroku.com) is a cloud-based development platform as a service (PaaS) provider. The Heroku platform supports development in Ruby on Rails, Java, Node.js, Python, Scala and Clojure. Heroku allow developers with Heroku accounts to create and deploy Web apps quickly. The application is sent to Heroku using either of the following: Git, GitHub, Dropbox, or via an API.

When completed, the **lab4-deploy-heroku.py** program will be deployed in Heroku and can be accessed from anywhere from the Internet.

Required Resources

- Postman application
- Python 3 with IDLE
- Python code files

Step 1: Create Project for Deployment

In this step, you will create your next project based on the Lab3 project.

a. Create your project directory called `lab4-heroku`. Copy the final version of your `lab3-serve-static.py` file from `lab3-server-static` directory to `lab4-heroku` directory. Rename the new file to `lab4-heroku.py`.

b. Install gunicorn webserver which will serve your application in Heroku environment instead of the development server we used in our development environment.

Note: We will not use gunicorn in our development environment, but installing gunicorn is a necessary step.

c. Change your main part of your project to:

```
InitDatabase()
pushDataToDatabase("Charles Webex", 15)
if __name__ == "__main__":
      app.run()
```

d. **Note:** This modification disallows of executing `app.run()` command in the Heroku environment. In development environment `app.run()` launches development web server. In Heroku environment we use gunicorn as production web server.

e. Create `requirements.txt` file to make your app dependencies available for deployment. `requirements.txt` will be used by Heroku environment to download and install packages which are necessary for running your app. In command prompt at root of your project directory run the following command:

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

`requirements.txt` file will list all packages and their version number that have been installed in your environment.

Note: Previously installed packages which are not used in your actual project also appear in `requirements.txt`. If you would like to avoid it, you should use virtual environment for your project development. Working on a python project in an isolated python environment is recommended so that python modules and packages don't meddle with that of other projects or even that of the operating system. `Virtualenv` is a great tool for creating isolated python environments. See the details on the following page: http://timmyreilly.azurewebsites.net/python-pip-virtualenv-installation-on-windows/. When you create a virtual environment for your project you will have a separate and clean Python installation which not contain any additional package.

f. In order to successfully deploy an application to Heroku, we have to add a `Procfile` to the application. Create a new file with `Procfile` as the name and do not add any extension. Add this line below

```
web: gunicorn lab4-heroku:app
```

`web` is used by Heroku to start a web server for the application. The `app:app` specifies the module and application name. In our application we have the `lab4-heroku` module and our flask application is called `app`.

Step 2: Create a repo for your project on GitHub

a. Install Git on your computer if you did not do it before. https://git-scm.com/download/win

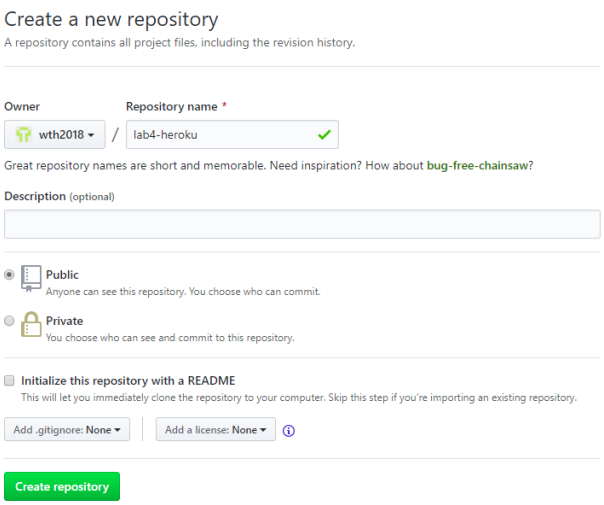
b. Open a command prompt at your project directory. Run the following commands for creating a local git repo and make your first commit:

```
git init
git add .
git commit –m "init"
```

c. Create a GitHub (https://github.com/) account if you do not have yet.

d. Login in with your credential on GitHub.

e. Create a new repo called `lab4-heroku`. Leave everything else on their default.



f. According to the hint on the next page, type the following command on the command prompt at your project directory.

```
git remote add origin https://github.com/<your-github-user-name>/lab4-heroku.git
git push -u origin master
```

Step 3: Create Heroku account

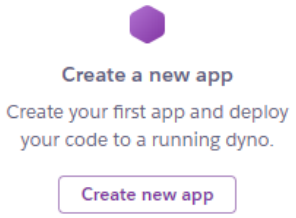
a. Navigate to the https://www.heroku.com/ page and click on the “Sign up” button.

b. Add your personal details and set “Primary development language” field to “Python” than click to “Create Free Account” button.

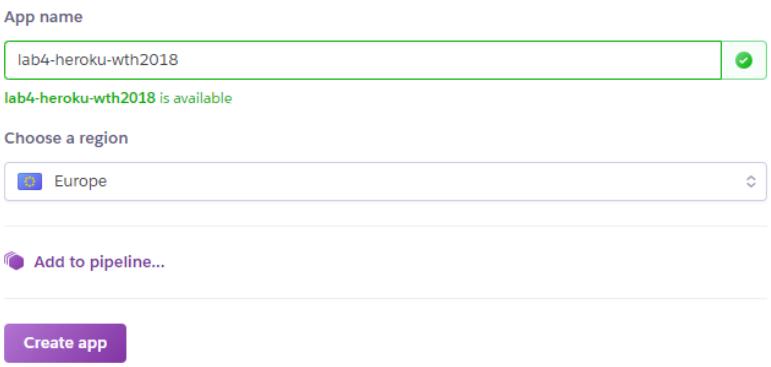
c. Check your email to confirm your account. Open the link in your verification email then give your password as the last step in your registration process.

Step 4: Create your first Heroku app

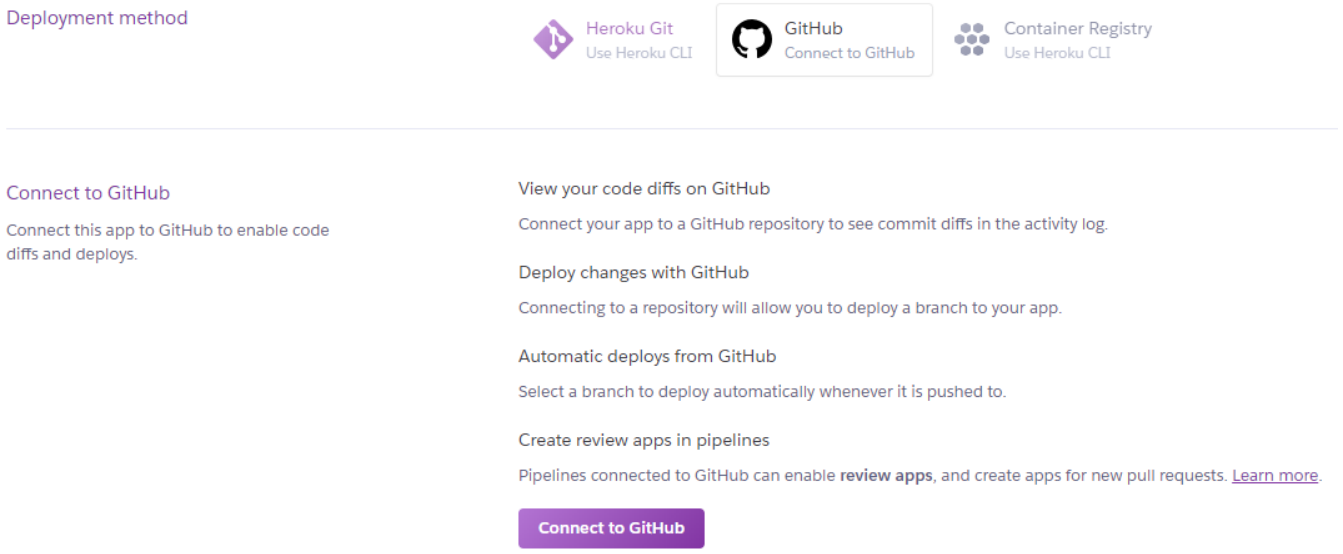
a. After you successfully logged in to Heroku click to Create new app button.



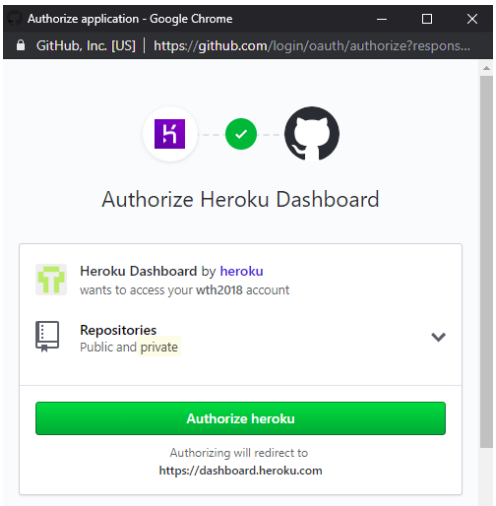
b. Type `lab4-heroku-<your-github-user-name>` as app name, select “Europe” under “Choose region” then click to “Create App” button.



c. On the next page select “GitHub” under the “Deployment method” section then click to “Connect to GitHub” button.



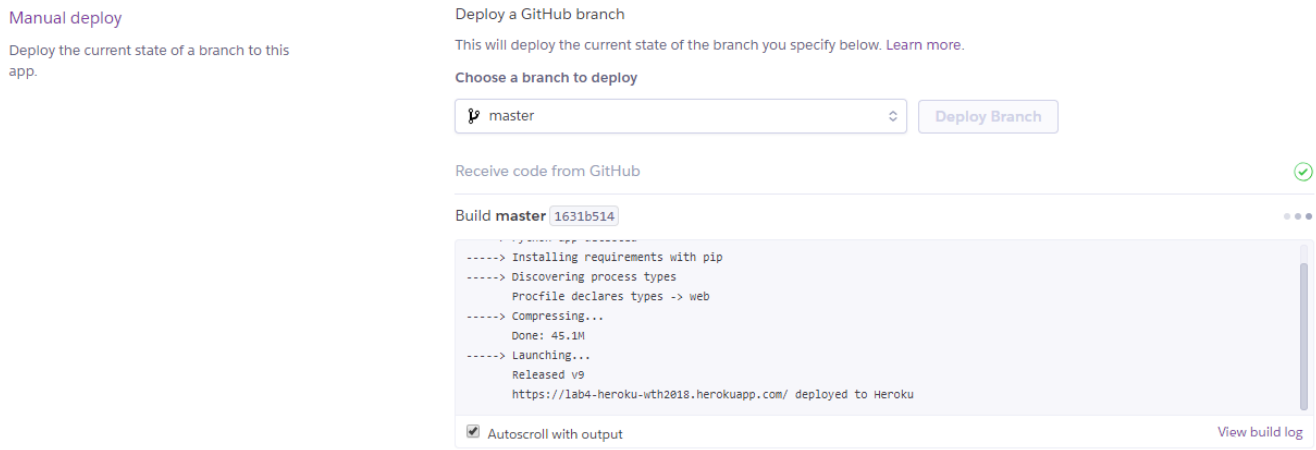
d. Grant access to Heroku to your GitHub account.



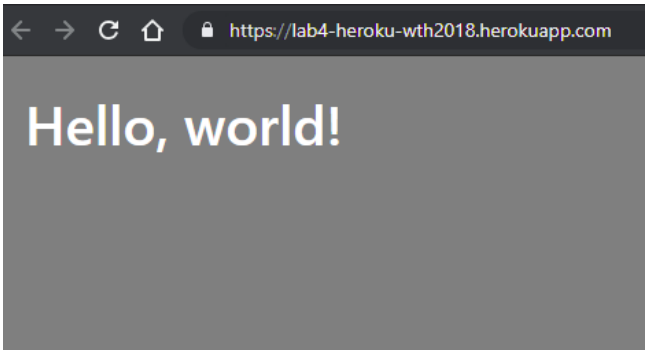
e. Search for your `lab4-heroku` repo and when it is listed click to “Connect” button.

f. Click to “Enable Automatic Deploys” button.

g. Click to “Deploy Branch” at the “Manual deploy” section. It will start a new deployment process.

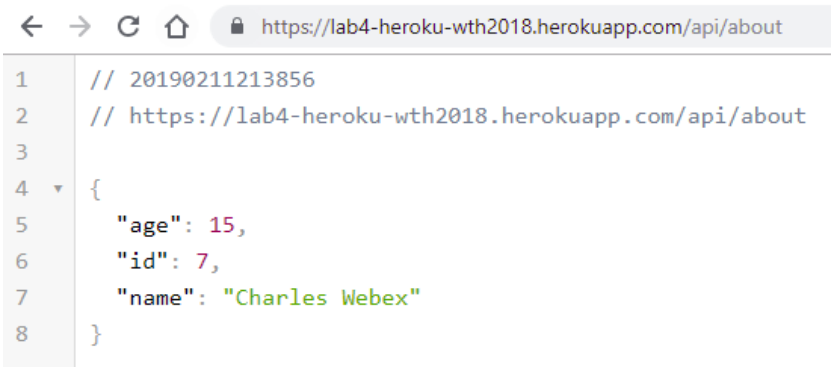


h. Click “Open app” button on the top right side of the browser window:



i. Check if `/api/about` REST API endpoint works. In your browser open the following link:

https://lab4-heroku-<your-github-user-name>.herokuapp.com/api/about



Step 4: Modify your project to test automatic deploys function

a. Change in your `/static/index,html` file “Hello, World” text to “Hello, Webex Teams”.

b. Also change the `pushDatabase` command in the main part of your python project file to `pushDataToDatabase("Victoria Teams", 15).`

c. Delete `about.db` file.

d. Commit and push your changes. In the command line at your project directory root type the following command:

```
git add .
git commit –m "test autodeploy function "
git push
```

e. Go back to your Heroku app. Click More button om the top right side of the browser window and select View logs

```
2019-02-11T20:51:31.000000+00:00 app[api]: Build started by user wth@http-foundation.hu
2019-02-11T20:51:47.186975+00:00 app[api]: Release v10 created by user wth@http-foundation.hu
2019-02-11T20:51:47.186975+00:00 app[api]: Deploy 4385741c by user wth@http-foundation.hu
2019-02-11T20:51:48.421649+00:00 heroku[web.1]: Restarting
2019-02-11T20:51:48.505567+00:00 heroku[web.1]: State changed from up to starting
2019-02-11T20:51:49.411232+00:00 heroku[web.1]: Stopping all processes with SIGTERM
2019-02-11T20:51:49.421071+00:00 app[web.1]: [2019-02-11 20:51:49 +0000] [10] [INFO] Worker exiting (pid: 10)
2019-02-11T20:51:49.421090+00:00 app[web.1]: [2019-02-11 20:51:49 +0000] [11] [INFO] Worker exiting (pid: 11)
2019-02-11T20:51:49.421981+00:00 app[web.1]: [2019-02-11 20:51:49 +0000] [4] [INFO] Handling signal: term
2019-02-11T20:51:49.522482+00:00 app[web.1]: [2019-02-11 20:51:49 +0000] [4] [INFO] Shutting down: Master
2019-02-11T20:51:49.607955+00:00 heroku[web.1]: Process exited with status 0
2019-02-11T20:51:52.687799+00:00 heroku[web.1]: Starting process with command `gunicorn lab4-heroku:app`
2019-02-11T20:51:54.795347+00:00 app[web.1]: [2019-02-11 20:51:54 +0000] [4] [INFO] Starting gunicorn 19.9.0
2019-02-11T20:51:54.796167+00:00 app[web.1]: [2019-02-11 20:51:54 +0000] [4] [INFO] Listening at: http://0.0.0.0:22882 (4)
2019-02-11T20:51:54.796326+00:00 app[web.1]: [2019-02-11 20:51:54 +0000] [4] [INFO] Using worker: sync
2019-02-11T20:51:54.801321+00:00 app[web.1]: [2019-02-11 20:51:54 +0000] [10] [INFO] Booting worker with pid: 10
2019-02-11T20:51:54.819605+00:00 app[web.1]: [2019-02-11 20:51:54 +0000] [11] [INFO] Booting worker with pid: 11
2019-02-11T20:51:55.000000+00:00 app[api]: Build succeeded
2019-02-11T20:51:56.324500+00:00 heroku[web.1]: State changed from starting to up
```

You can follow the deployment process and see the result of it on the log window.

Note: Every time when you push a new commit to your GitHub repo, a new deployment process starts automatically.

f. Open your app and refresh your browser window (Ctrl-F5) if it is necessary.

g. Check if /api/about give Victoria Teams' details.

https://lab4-heroku-<your-github-user-name>.herokuapp.com/api/about