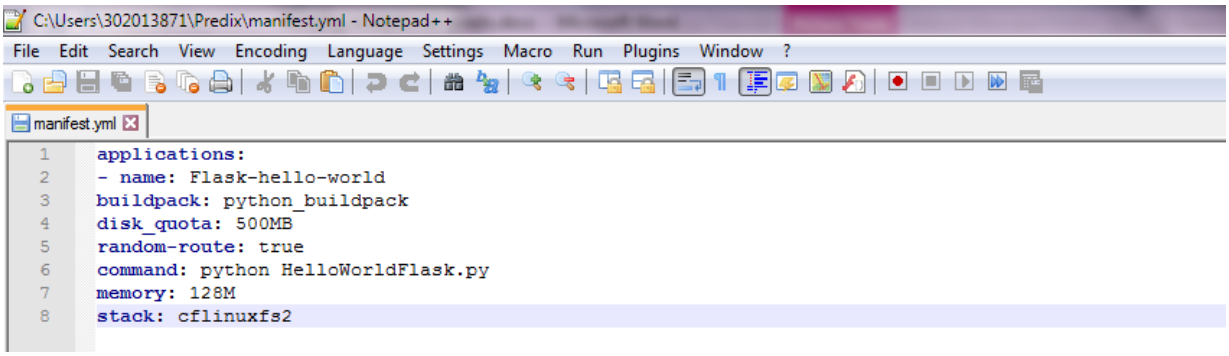
Developing and deploying "Hello World" step by step using Python on Predix:

1. Create 3 Files: requirements.txt, mainpythonfile.py, manifest.yml

2. Edit the manifest.yml as follows (Highlighted lines are mandatory):

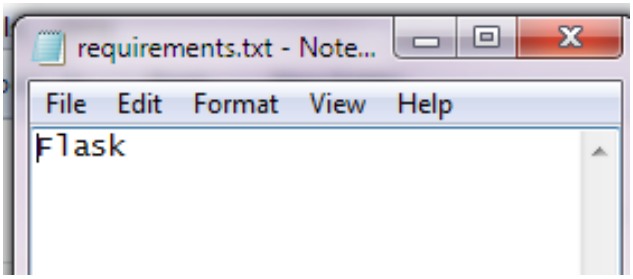
applications:	
- name: Flask-hello-world	(This the name of your app)
buildpack: python_buildpack	(This is the python buildpack)
disk_quota: 500MB	
random-route: true	(While this is not mandatory, its good to use it as of now. This provides the route to creating an URL. It can be random our enter your own URL)
command: python HelloWorldFlask.py	(This is the name of your python file)
memory: 128M	
stack: cflinuxfs2	



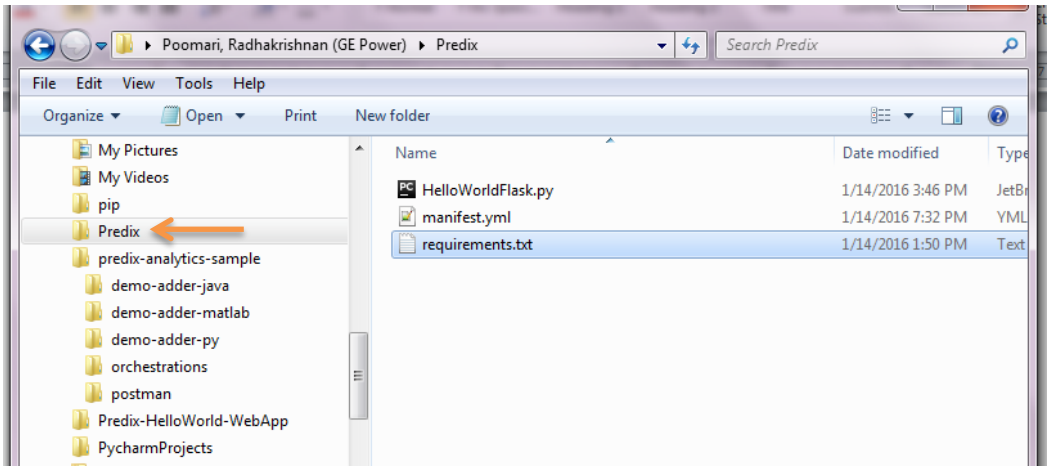
3. Include the following highlighted code segments in the python code: To know what is Flask checkout the link https://en.wikipedia.org/wiki/Flask_(web_framework)

```
import os
from flask import Flask

app = Flask(__name__)
port = int(os.getenv("VCAP_APP_PORT"))
@app.route('/')

### Your Python code goes starts here:
def hello_world():
   return 'Hello World! I am running on port' + str(port)

### Your Python code ends here:

if __name__ == '__main__':
   app.run(host='0.0.0.0',port=port)
```

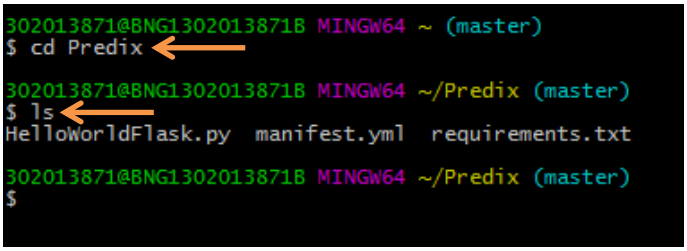
4. Edit the requirements.txt file as follows. Include all the import library names within this file.
    a. In the above example flask is the library imported. Include it in the txt file. Libraries such as os, math and other standard python libraries need not be mentioned in the requirements file.



5. Place all the three files – requirements.txt, manifest.yml and HelloWorldFlask.py in a folder:



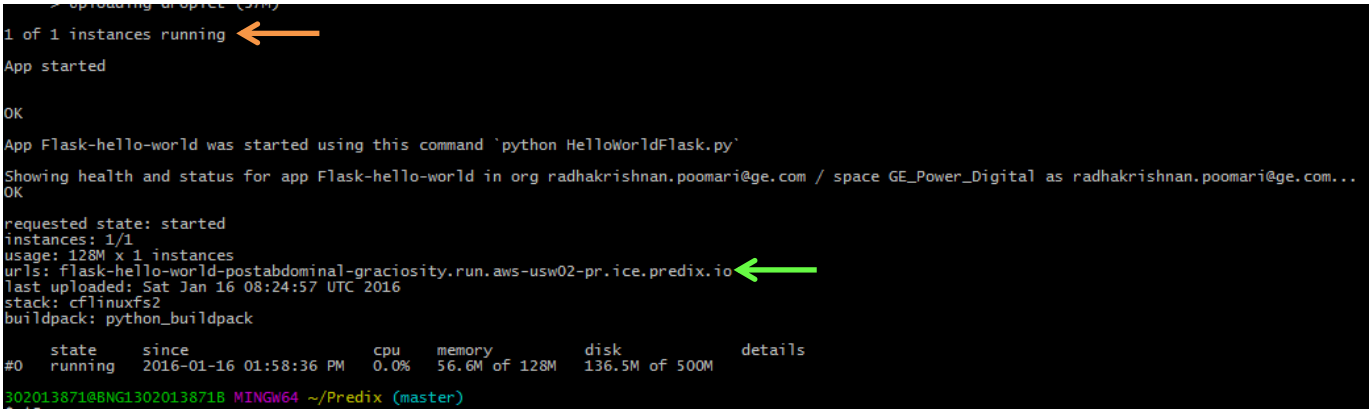
6. From the master directory, navigate to the folder where the three files are situated as follows:
    a. To get to the master directory:
    $ cd
    b. From master move to Predix folder
    $ cd Predix (Here Predix is the folder that as the three files we created)
    c. Make sure all three files are inside this folder using the command:
    $ ls



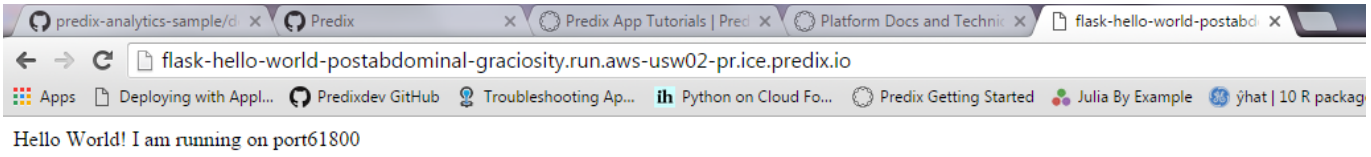
7. Push the app (we named it Flask-hello-world in our .yml file) as follows:

   $ cf push Flask-hello-world

   A successful start of the app will have the following message displayed.

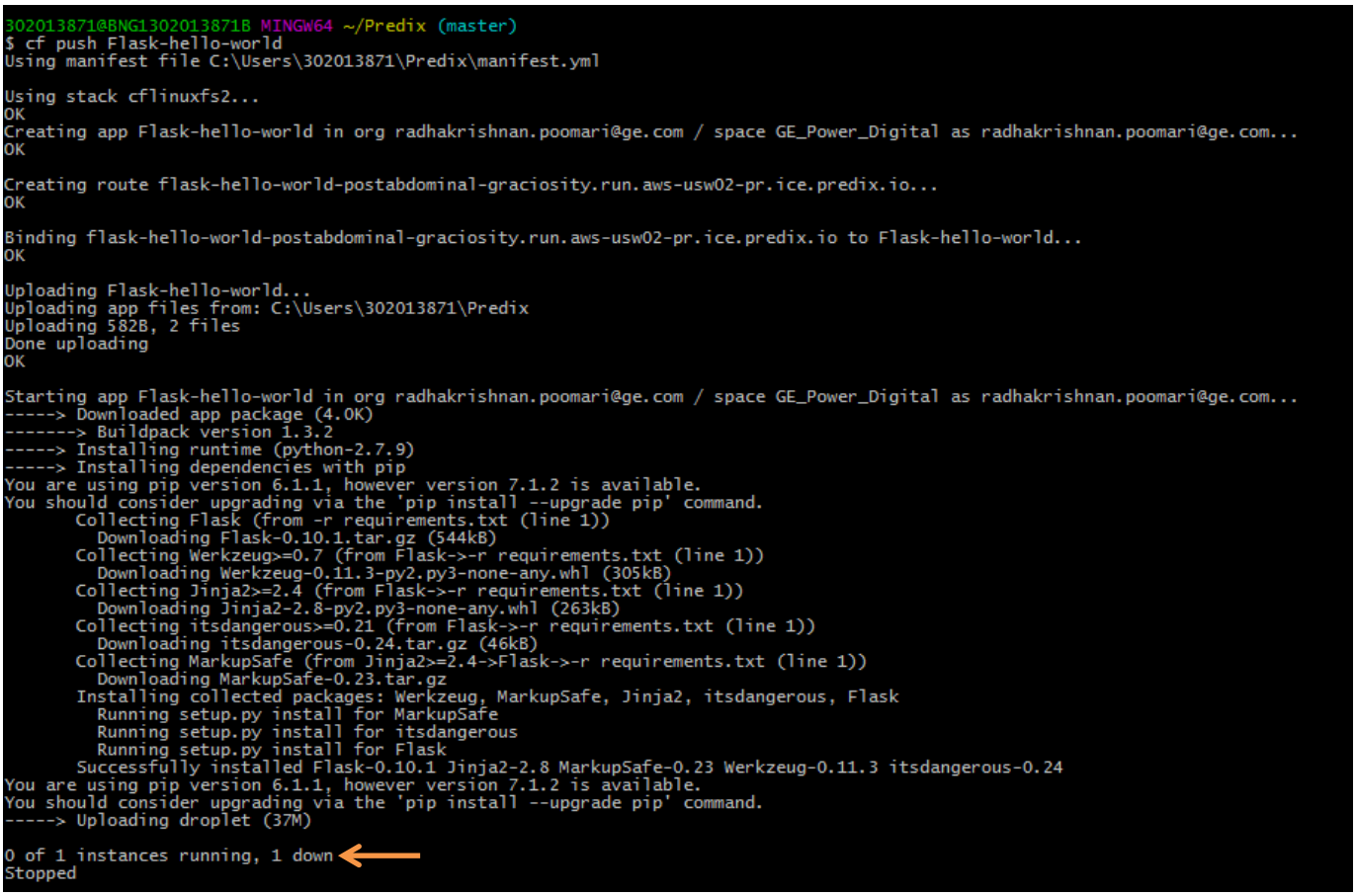


8. The URL randomly generated for our app is highlighted by the green arrow. Open the URL in a web browser. Congratulations! on running your first Python-based app on Predix.



Troubleshooting:

1. Getting the following highlighted message means, the app (python file) has an error. Make sure the template code is correct and the python file is free of errors. If you continue to get the message "0 of 1 instances running, 1 down", stop the process by pressing Ctrl z.



2. For any other errors check if the following are set right:
    a. All libraries in python file are imported (not the import os, import math etc) in the requirements.txt file
    b. Make sure the name of the requirements.txt and manifest.yml are spelled correctly
    c. Check if all the 3 files are placed in the same directory
    d. Check if your proxies are set correct. Refer to the previous documentation on how to set proxies.
    e. Make sure the main python file is error free. Execute it and check if the error code is 0.
    f. Make sure the build pack in manifest.yml for python is correctly specified as python_buildpack
    g. If you're using a VPN connection from home make sure working fine.

Key Cloudfoundry Commands

1. $cf m – List all services available in cloudfoundry
2. $cf marketplace – same as cf m
3. $cf apps – List all the apps that are created along with their status and URL
4. $cf buildpacks – lists all available buildpacks
5. $cf push *mapp* – pushes the app specified in the manifest file to the cloud
6. $cf scale *myapp* – creates multiple instances of myapp. Good for testing scalability of the applications. Eg. cf scale myapp -i 5 -m 1G
7. $cf bind-service *myapp* redis – binds a service to myapp. Service can be anything available in the marketplace.
8. $cf delete *myapp* – deletes *myapp*
9. $cf stop *myapp* – Stops a running app

Resources

1. http://www.ianhuston.net/2014/11/python-on-cloud-foundry/
2. https://docs.cloudfoundry.org/devguide/deploy-apps/cf-scale.html
3. https://docs.cloudfoundry.org/devguide/deploy-apps/manifest.html#minimal-manifest
4. https://github.com/PredixDev/predix-analytics-sample/tree/master/demo-adder-py
5. https://github.com/PredixDev