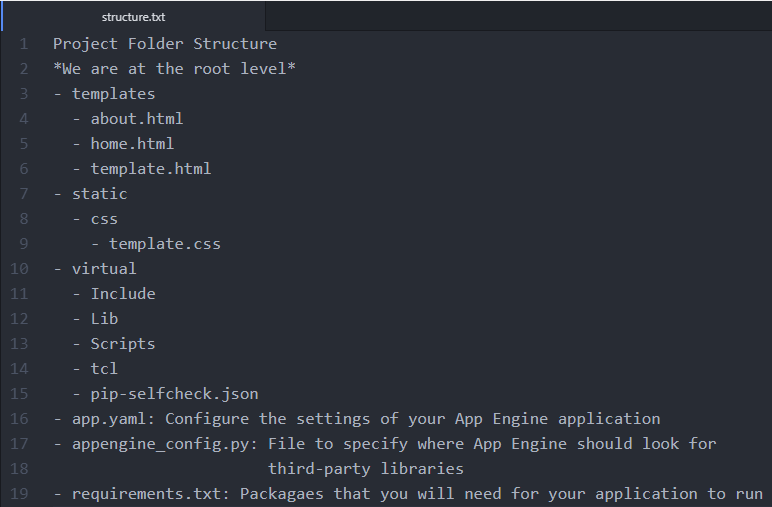
**An Introduction to Web Applications and Deploying Them to the Cloud  
Part 3: Let’s Send it to the Cloud**

By Salvador Villalon

**Deploy Your Web Application to the Cloud**

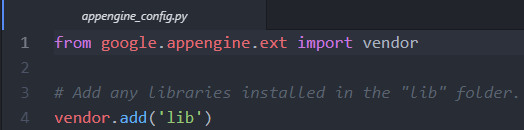
* To deploy our Web Application to the Cloud we will use [Google App Engine](https://cloud.google.com/appengine/) (Standard Environment)
* What do we need to deploy our web application?
  + We will use **Google App Engine** which is an example of a **Platform as a Service (PaaS)**
  + **PaaS** refers to the **delivery of operating systems and associated services over the internet without downloads or installation**. The approach lets customers create and deploy applications without having to invest in the underlying infrastructure. (TechTarget)
    - **More info:** [**https://searchitchannel.techtarget.com/definition/cloud-services**](https://searchitchannel.techtarget.com/definition/cloud-services)
  + **Google App Engine** is a platform as a service offering that allows developers and businesses to build and run applications using Google's advanced infrastructure (TechOpedia)
* **Before you Start:**
  + You will need a **Google Account**. You can create a Google Account Here: <https://accounts.google.com/signup/v2/webcreateaccount?hl=en&flowName=GlifWebSignIn&flowEntry=SignUp>
  + Once you create an account, go to the Google Cloud Platform Console (<https://console.cloud.google.com>) and create a new project
  + Install the Google Cloud SDK: <https://cloud.google.com/sdk/>
* At the end of this tutorial your project structure will look like this:



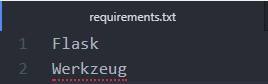
* We will need to create three new files:
  + **app.yaml**
  + **appengine\_config.py**
  + **requirements.txt**
* Content of app.yaml



* + <https://cloud.google.com/appengine/docs/standard/python/getting-started/python-standard-env>
  + If you were to check the part where they **talk about content of the app.yaml**, it does not include what I typed in lines 11 – 13
  + When I first attempted to deploy my simple web app, my deployment never worked. By adding the SSL Library, it allows us to [create secure connections between client and server](https://wiki.python.org/moin/SSL). Every time the user goes to our website they will need to connect to a server that is run by Google App Engine. We need to create secure connection for this
* Content of appengine\_config.py



* Content of requirements.txt



* Now inside our virtual environment **(make sure your virtualenv is activated)** we are going to install the new dependencies we have in requirements.txt
  + Do this command: **pip install -t lib -r requirements.txt**
    - **- t lib :** This flag copies the libraries into a lib folder, which is uploaded to App Engine during deployment
    - **-r requirements.txt :** Tells pip to install everything from requirements.txt
* **Deploying the Application**
  + To deploy the application to Google App Engine, use this command.
  + **gcloud app deploy**
  + I usually include **-- project [ID of Project]**
  + This specifies what project you are deploying. The command will look like

**gcloud app deploy -- project [ID of Project]**

* **The Application**
  + Now check the URL of your application
  + The application will be store in the following way:
    - **“your project id”.appspot.com**
  + My application is here: <http://sal-flask-tutorial.appspot.com>

**Conclusion**

* **From this tutorial, you all learned how to**
  + Use the framework called Flask to use Python as a Server Side Language
  + Learned how to use HTML, CSS, and Flask to make a website
  + Learned how to create Virtual Environments using virtualenv
  + Use Google App Engine Standard Environment to deploy an application to the cloud
* **What I learned?**
  + I learned three important things from this small project.
    - First, I learned about the difference between a static website and a web application
      * Static Websites:
        + Means that the server is just serving HTML, CSS, and JavaScript files to the client. The content of the site does not change when the user interacts with it.
      * Web Applications:
        + A Web Application or Dynamic websites generate content based on retrieved data (most of the time is a database) that changes based on a user's interaction with the site. In a Web Application the server is responsible for querying, retrieving, and updating data. This causes web applications to be slower and more difficult to deploy than static websites for simple applications
      * Server Side and Client Side:
        + I learned that a web application has two sides. The client side and the server side. The client side is what the user interacts with and the server side is where the all the information that the user inputted is processed
    - Second, I learned about Cloud Services
      * Most of my previous projects where static websites and to deploy them I used this service called [GitHub Pages](https://pages.github.com/). GitHub Pages is a free static site hosting service design to host any kind of project from a GitHub Repository.
      * When working with web applications, I could not use GitHub Pages to host them. GitHub Pages is only meant for static websites not for something dynamic like a web application. I had to use Cloud Services such as [Amazon Web Services](https://aws.amazon.com) or [Heroku](https://www.heroku.com/)
    - Third, I learned how to use Python as a Server Side Language
      * In order to create the server side of the web application we had to use a server side language. I learned that I could use the framework called Flask to use Python as the Server Side Language
* **Contact Info:**
  + If you need to contact me:
    - LinkedIn: <https://www.linkedin.com/in/salvadorvillalon/>
    - School Email: [salvav1@uci.edu](mailto:salvav1@uci.edu)
    - Work Email: [Salvador.villalon@spglobal.com](mailto:Salvador.villalon@spglobal.com)
* **Next Steps:** 
  + You can build all sorts of things with Flask. I realized that Flask helps make the code behind the website easier to read. I have made the following applications during this Summer 2018 and I hope to make more
    - Personal Projects
      * A Twilio SMS App: <http://twilio-pokedex.appspot.com/>
      * My Personal Website: <http://salvador-villalon.appspot.com/>
    - During my internship
      * Part of a project where I learned about Docker and Containers <http://spgi2018-container-project.appspot.com/>
  + Here is the list of resources that helped me create this tutorial
* <https://pythonhow.com/building-a-website-with-python-flask/>
* [https://cloud.google.com/appengine/docs/standard/python/getting-started/python-standard-env](https://cloud.google.com/appengine/docs/standard/python/getting-started/python-standard-env#test_the_application)
* <https://youtu.be/j5wysXqaIV8>
* <https://realpython.com/python-virtual-environments-a-primer/>
* <http://flask.pocoo.org/docs/0.12/installation/#installation>
* <https://www.techopedia.com/definition/31267/google-app-engine-gae>
* <https://cloud.google.com/appengine/>
* <https://searchitchannel.techtarget.com/definition/cloud-services>
* <https://www.techopedia.com/definition/31267/google-app-engine-gae>
* <https://www.reddit.com/r/Python/comments/1iewqt/deploying_static_flask_sites_for_free_on_github/>