Name:- Goregaonkar Purva Pradip.
Roll No: 33313
Batch: K11
Subject: cloud computing lab.

Assignment 1.

Title:- Install Google App Engine. Create hello world ap
and other simple web application using python/Java

Requirements:-
1. Google App Engine
2. Python Interpreter (Python 2.7x)
3. Text Editor
4. Browser.

Theory:-
A> Google App Engine:-
1. Google App Engine (GAE) is a platform-as-a servi
product that provides web app developers and
enterprises with access to Google's scalable
hosting and tier 1 internet service.
2. GAE requires that applications be written in Jav
or python, store data in Google Bigtable and use th
Google query language.
3. Noncompliant applications requires modificatio

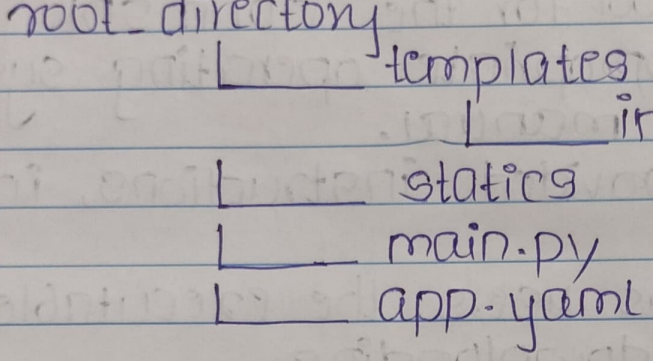
to use GAE.
4. GAE provides more infrastructure than other scalable hosting services, such as Amazon Elastic Compute cloud (EC2).
5. GAE also eliminates some system administration and development tasks to make writing scalable application easier.
6 Google provides GAE free up to a certain amount of use for resources like CPU, storages, API calls and concurrent requests.

B> Google cloud SDK.
1. Google cloud SDK (Software Development Kit), in simple terms, is a set of tools that are used to manage applications and resources that are hosted on the Google cloud Platform.
2. It is composed of the gsutil, gcloud, and bqcommond line tools.
3. The gcloudtool is automatically downloaded with the cloud SDK.
4. Google cloud SDK run on specific platforms. Windows, Linux and macOS, and requires python 2.7x.
5. SDK might have further necessities like Java tools used for the development of Google App Engine needs Java 1.7 or the later one.
6. It can be used to locally deploy and test

web applications.

c. Directory structure for creating hello world applications:-

1. The web applications to be developed can be organized in the following directory structure.



2. The templates directory can be used to store the web templates of the web application (HTML files).

3. The statics directory can be used to store the web static ~~fields~~ ~~both~~ files which contain the stylish and the business logic data for the web application (CSS and JS files).

4. The main.py is used to defines the routes, rendering logic, data acquisition logic.

5. It provides the WSGI abstraction to the application.

6. The app.yaml file provides the runtime environment, URLs for routes and launch configuration

of the application in the form of key value pairs.

Steps:-

A> Install Google cloud SDK on windows or Linux machines.

1. Visit the https://cloud.google.com/sdk/docs/install link to download the CLI (command line interface) tool for the cloud SDK.
2. Select the appropriate operating system from the installation manual.
3. Follow the provided instructions, in the displayed section.

a> for windows users, the executable downloader is provided for downloading.

b> for ubuntu and fedora users, terminal commands for installation are provided using apt and dnf repositories respectively.

B> Creating the application:-

1> The application must be initialized using the above-mentioned directory structure.

2> It is recommended format for organization and redability of code.

3> The app.yaml file should be contain the following content.

Contents of app.yaml:

```
runtime : python27
api_version : 1
threadsafe : true

handlers
-url: /
 script : main.app
```

4) The logic of the application, i.e, the web server interaction code of the application must be placed in the main.py file

5) A simple code displaying the hello world on a web page is as follows

Contents of main.py for the Hello World application

```
import webapp2
class MainPage (webapp2.RequestHandler):

 def get (self):
     self.response.write ("Hello World")

app = webapp2.WSGIApplication([("/", Main
page)], debug = True
```

6. Finally, after saving the above code, the application can be run on the localhost server using the following command. (The command must be run on the Google cloud shell or the terminal in case of the Ubuntu).

Command:-

python <path_to_sdk>/bin/devappserver.py <path_to_application_directory>