

EXPERIMENT NO:01

Title: Implementation and case study of Google App Engine

Solution:

Google App Engine (often referred to as GAE or simply App Engine) is a cloud computing platform as a service for developing and hosting web applications in Google- managed data centers. Applications are sandboxed and run across multiple servers. App Engine offers automatic scaling for web applications—as the number of requests increases for an application, App Engine automatically allocates more resources for the web application to handle the additional demand.

Google App Engine primarily supports Go, PHP, Java, Python, Node.js, .NET, and Ruby applications, although it can also support other languages via "custom runtimes". The service is free up to a certain level of consumed resources and only in standard environment but not in flexible environment.

Fees are charged for additional storage, bandwidth, or instance hours required by the application. It was first released as a preview version in April 2008 and came out of preview in September 2011.

Why Google App Engine is used?

Google allows you to add your web application code to the platform while managing the infrastructure for you. The engine ensures that your web apps are secure and running and saves them from malware and threats by enabling the firewall.

Feature of Google Cloud App Engine

1	Popular Language	Build your application in Node.js,Java,Ruby,C#,Python or PHP-or bring your own language runtime
2	Open and flexible	Custom runtimes allow you to bring any library and framework to App Engine by supplying a Docker container.
3	Fully Managed	A fully managed environment lets you focus on code while App Engine manages infrastructure concerns.
4	Powerful application diagnostics	Use Cloud Monitoring and Cloud Logging to monitor the health and performance of your app and Cloud Debugger and Error Reporting to diagnose and fix bugs quickly.

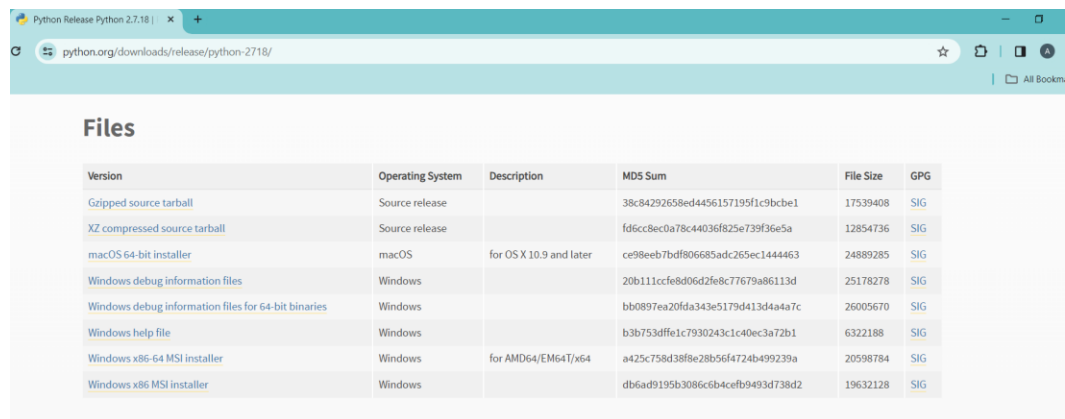
5	Application Versioning	Easily host different versions of your app, and easily create development, test, staging, and production environments.
6	Traffic Splitting	Route incoming requests to different app versions, A/B test, and do incremental feature rollouts.
7	Application Security	Help safeguard your application by defining access rules with App Engine firewall and leverage managed SSL/TLS certificates by default on your custom domain at no additional cost.
8	Services Ecosystem	Tap a growing ecosystem of Google Cloud services from your app including an excellent suite of cloud developer tools.

1. Installing Python and Google Cloud Sdk

i. Installing Python 2.7.18:

Go to link: <https://www.python.org/downloads/release/python-2718/> and install the appropriate version.

After installation, go to environment variables>user variables>path><Enter your python file path>, according to the folder location.

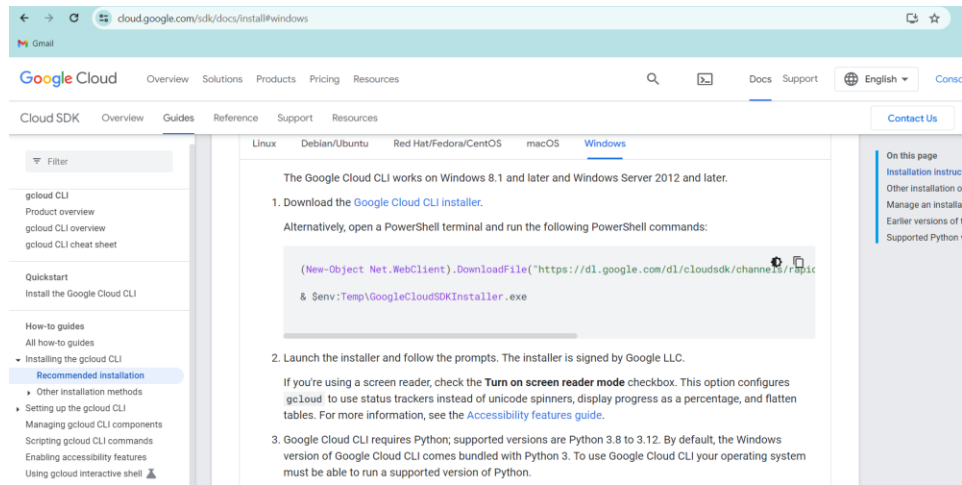


The screenshot shows a web browser window with the URL [python.org/downloads/release/python-2718/](https://www.python.org/downloads/release/python-2718/). The page title is "Python Release Python 2.7.18". Below the title, there is a section titled "Files" containing a table with the following data:

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		38c84292658ed4456157195f1c9bcbe1	17539408	SIG
XZ compressed source tarball	Source release		fd6cc8ec0a78c44036f825e739f36e5a	12854736	SIG
macOS 64-bit installer	macOS	for OS X 10.9 and later	ce98eeb7bdf806685adc265ec1444463	24889285	SIG
Windows debug information files	Windows		20b111ccfe8d06d2fe8c77679a86113d	25178278	SIG
Windows debug information files for 64-bit binaries	Windows		bb0897ea20fda343e5179d413d4a4a7c	26005670	SIG
Windows help file	Windows		b3b753dffe1c7930243c1c40ec3a72b1	6322188	SIG
Windows x86-64 MSI installer	Windows	for AMD64/EM64T/x64	a425c758d38f8e28b56f4724b499239a	20598784	SIG
Windows x86 MSI installer	Windows		db6ad9195b3086c6b4cefb9493d738d2	19632128	SIG

ii. Installing Google Cloud Sdk:

- Go to link: <https://cloud.google.com/sdk/docs/install> and install the Google Cloud CLI installer.



2. Writing the Program

Create a folder and create 2 files.

1. Main.py

Write the following program in main.py file. Make sure webapp2 is installed.

```
Cloud > main.py
1 import webapp2
2
3 class MainPage(webapp2.RequestHandler):
4     def get(self):
5         self.response.write("Hello World!")
6
7 app = webapp2.WSGIApplication([('/', MainPage)], debug=True)
8
```

2. App.yaml

Write the following instructions in the app.yaml file.

```
Cloud > app.yaml
1 runtime: python27
2 api_version: 1
3 threadsafe: true
4
5 handlers:
6 - url: /
7   script: main.app
```

3. Running the Program

- i. Check if the CLI recognizes the python version by entering: `python -V`
- ii. To run the program, open Google Cloud SDK Shell and type the following command:
`py google-cloud-sdk\bin\dev_appserver.py <Folder location where both programs are located>`

4. Output

This is the generated output window where link to the localhost is listed.

```
Allow dev_appserver to check for updates on startup? (Y/n): y
dev_appserver will check for updates on startup. To change this setting, edit /
home/mike/.appcfg_nag
INFO      2013-04-30 20:39:40,481 sdk_update_checker.py:244] Checking for updates
to the SDK.
INFO      2013-04-30 20:39:40,778 sdk_update_checker.py:272] The SDK is up to dat
e.
WARNING   2013-04-30 20:39:40,968 simple_search_stub.py:977] Could not read searc
h indexes from /tmp/appengine.greetings.mike/search_indexes
INFO      2013-04-30 20:39:40,969 api_server.py:152] Starting API server at: http
://localhost:45291
INFO      2013-04-30 20:39:40,986 dispatcher.py:150] Starting server "default" ru
nning at: http://localhost:8080
INFO      2013-04-30 20:39:40,988 admin_server.py:117] Starting admin server at:
http://localhost:8000
```

Go to browser and paste the localhost address to see the output.



