PRACTICAL NO : 02

Title : Assignment to install and configure Google App Engine

---

Practical Title: Installation and configure Google App Engine.

Objectives:

- To learn basic of Google App Engine.
- To install and configure Google App Engine.

Hardware Requirements:

- Pentium IV with latest configuration

Software Requirements:

- Ubuntu 20.04, Web application i.e. Google App Engine

Theory:

Introduction

Google App Engine is a web application hosting service. By "web application," we mean an application or service accessed over the Web, usually with a web browser: storefronts with shopping carts, social networking sites, multiplayer games, mobile applications, survey applications, project management, collaboration, publishing, and all the other things we're discovering are good uses for the Web. App Engine can serve traditional website content too, such as documents and images, but the environment is especially designed for real-time dynamic applications. Of course, a web browser is merely one kind of client: web application infrastructure is well suited to mobile applications, as well.

In particular, Google App Engine is designed to host applications with many simultaneous users. When an application can serve many simultaneous users without degrading performance, we say it scales. Applications written for App Engine scale automatically. Asmore people use the application, App Engine allocates more resources for the application and manages the use of those resources. The application itself does not need to know anything about the resources it is using.

The app engine is a Cloud-based platform, is quite comprehensive and combines infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS). The app engine supports the delivery, testing and development of software on demand in a Cloud computing environment that supports millions of users and is highly scalable.

The company extends its platform and infrastructure to the Cloud through its app engine. It presents the platform to those who want to develop SaaS solutions at competitive costs. Have you ever wondered as to who stands to benefit the most from the Google app engine? If you are a business SME or enterprise which owns any web-based application that needs to be scaled

without any compromise on the performance then Google App Engine is a good fit. Companies like Best Buy and Khan Academy have chosen Google App Engine for their apps.

Google App Engine:

It is a platform-as-a-service (PaaS) Cloud computing platform that is fully managed and uses inbuilt services to run your apps. You can start development almost instantly after downloading the software development kit (SDK). You can go on to the developer's guide right away when you click on the language you wish to develop your app in.

As soon as you have signed up for a Cloud account, you can build your app:

- With the template/HTML package in Go
- With Jinja2 and webapp2 in Python
- With Cloud SQL in PHP
- With Maven in Java

Generally Available Features

These are covered by the depreciation policy and the service-level agreement of the app engine. Any changes made to such a feature are backward-compatible and implementation of such a feature is usually stable. These include data storage, retrieval, and search; communications; process management; computation; app configuration and management.

- Data storage, retrieval, and search include features such as HRD migration tool, Google Cloud SQL, logs, datastore, dedicated Memcache, blobstore, Memcache and search. Cloud
- Communications include features such as XMPP. channel, URL fetch, mail, and Google Endpoints.
- Process management includes features like scheduled tasks and task queue
- Computation includes images.
- App management and configuration cover app identity, users, capabilities, traffic splitting,

modules, SSL for custom domains, modules, remote access, and multitenancy.

Advantages of Google App Engine:

Infrastructure for Security

Around the world, the Internet infrastructure that Google has is probably the most secure. There is rarely any type of unauthorized access till date as the application data and code are stored in highly secure servers. You can be sure that your app will be available to users worldwide at all times since Google has several hundred servers globally. Google's security and privacy policies are applicable to the apps developed using Google's infrastructure.

Scalability

For any app's success, this is among the deciding factors. Google creates its own apps using GFS, Big Table and other such technologies, which are available to you when you

utilize the Google app engine to create apps. You only have to write the code for the app and Google looks after the testing on account of the automatic scaling feature that the app engine has. Regardless of the amount of data or number of users that your app stores, the app engine can meet your needs by scaling up or down as required.

□ Performance and Reliability

Google is among the leaders worldwide among global brands. So, when you discuss performance and reliability you have to keep that in mind. In the past 15 years, the company has created new benchmarks based on its services' and products' performance. The app engine provides the same reliability and performance as any other Google product.

□ Cost Savings

You don't have to hire engineers to manage your servers or to do that yourself. You can invest the money saved into other parts of your business.

□ Platform Independence

You can move all your data to another environment without any difficulty as there is not many dependencies on the app engine platform.

STEPS TO INSTALL & CONFIGURE :

Before begin

1. Create a Google Cloud Platform project, if you don't have one already.
2. Make sure that Python 2.7 is installed on your

   system: python -V

   Note: As of Cloud SDK version 206.0.0, the gcloud CLI has experimental support for running using a Python 3.4+ interpreter (run gcloud topic startup for exclusions and more information on configuring your Python interpreter). All other Cloud SDK tools still requirea Python 2.7 interpreter.

3. **Download the archive file best suited to your operating system**. Most machines will run the 64-bit package. If you'd like to check, run uname -m to verify if you're running a 64-bit system.

Platform	Package	Size	SHA256 Checksum
Linux			
For 64-bit			
	google-cloud-sdk-229.0.0-linux-	25.6 MB	b1c87fc9451598a76cf66978dd8aa06482bfced639b56cf31559dc2c7f8b7b90
For 32-bit			
	google-cloud-sdk-229.0.0-linux-	25.2 MB	ee8c45f8018d0fee92b07c32cc6d8c891241da0b88bfe289d4e58e6746c3f668

(x86) x86.tar.gz

Alternatively, to download the Linux 64-bit archive file from your command-line, run:

curl -O https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloud-sdk- 229.0.0-linux-x86_64.tar.gz

32-bit archive file, run:

curl -O https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloud-sdk- 229.0.0-linux-x86.tar.gz

4. Extract the archive to any location on your file system; preferably, your Home folder. On Linux, you can extract the archive file by running this command:

   tar zxvf [ARCHIVE_FILE] google-cloud-sdk

5. If you're having trouble getting the gcloud command to work, ensure your $PATH is defined appropriately. Use the install script to add Cloud SDK tools to your path. You will also be able to opt-in to command-completion for your bash shell and usage statistics collection during the installation process. Run the script using this command:

   ./google-cloud-sdk/install.sh

Restart your terminal for the changes to take effect.

Alternatively, you can call Cloud SDK after extracting the downloaded archive by invoking its executables via the full path.

Initialize the SDK

Use the gcloud init command to perform several common SDK setup tasks. These include authorizing the SDK tools to access Google Cloud Platform using your user account credentials and setting up the default SDK configuration.

To initialize the SDK:

1. Run the following at a command

   prompt: gcloud init

   Note: To prevent the command from launching a web browser, use gcloud init --console- only instead. To authorize without a web browser and non-interactively, create a service account with the appropriate scopes using the Google Cloud Platform Console and use gcloud auth activate-service-account with the corresponding JSON key file.

2. Accept the option to log in using your Google user account:

   To continue, you must log in. Would you like to log in (Y/n)? Y

3. In your browser, log in to your Google user account when prompted and click **Allow** to grant permission to access Google Cloud Platform resources.
4. At the command prompt, select a Cloud Platform project from the list of those where you have **Owner**, **Editor** or **Viewer** permissions:

Pick cloud project to use:

[1] [my-project-1]

[2] [my-project-2]

...

Please enter your numeric choice:

If you only have one project, gcloud init selects it for you.

5. If you have the Google Compute Engine API enabled, gcloud init allows you to choose adefault Compute Engine zone:

Which compute zone would you like to use as project default?

[1] [asia-east1-a]

[2] [asia-east1-b]

...

[14] Do not use default zone

Please enter your numeric

choice:

gcloud init confirms that you have complete the setup steps successfully:gcloud has now been configured!

You can use [gcloud config] to change more gcloud settings.

Your active configuration is: [default]

Run core gcloud commands

Run these gcloud commands to view information about your SDK installation:

1. To list accounts whose credentials are stored on the local

system:gcloud auth list

gcloud displays a list of credentialed accounts:

Credentialed

Accounts ACTIVE

ACCO

UNT

* example-user-

1@gmail.com example-

user-2@gmail.com

2. To list the properties in your active SDK configuration:

gcloud config list

gcloud displays the list of properties:

[core]
account = example-user-
1@gmail.com
disable_usage_reporting = False
project = example-project

3. To view information about your Cloud SDK installation and the active SDK configuration: gcloud info

   gcloud displays a summary of information about your Cloud SDK installation. This includes information about your system, the installed SDK components, the active user account and current project, and the properties in the active SDK configuration.

4. To view information about gcloud commands and other topics from the command line: gcloud help

   For example, to view the help for gcloud compute instances create:

   gcloud help compute instances create

   gcloud displays a help topic that contains a description of the command, a list of commandflags and arguments, and examples of how to use it.

How to Run Program:

Now as we have finished installing app engine, now it's time to createand upload an app. In this case we will be taking example of a "HELLO WORLD" app in python.

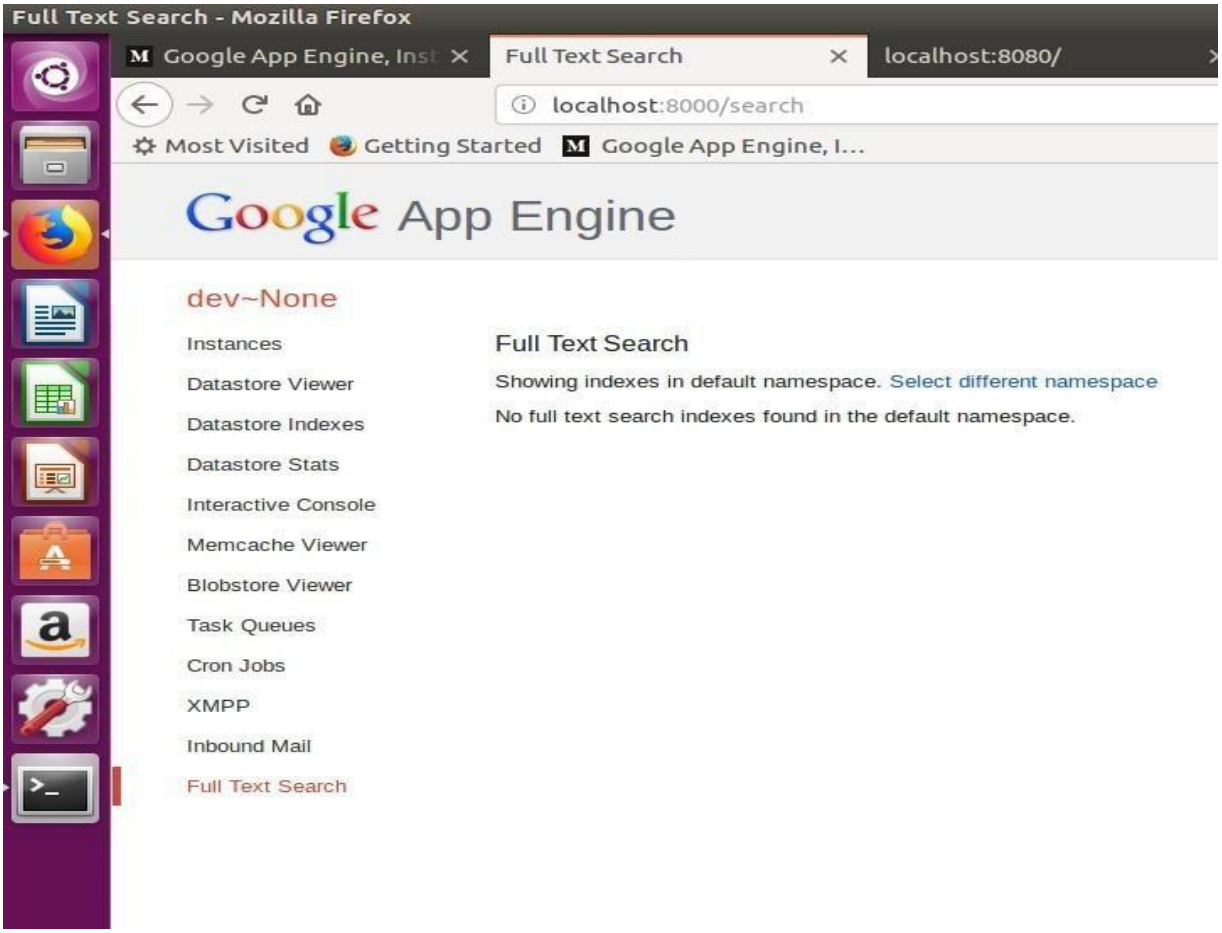
1. As we already have made sure that we have python installed in our system, It will be easier for usto clone existing code and deploy it rather than creating our own so we will use python docs- sample. Run the command "**git clone https://github.com/GoogleCloudPlatform/python-docs- samples**".

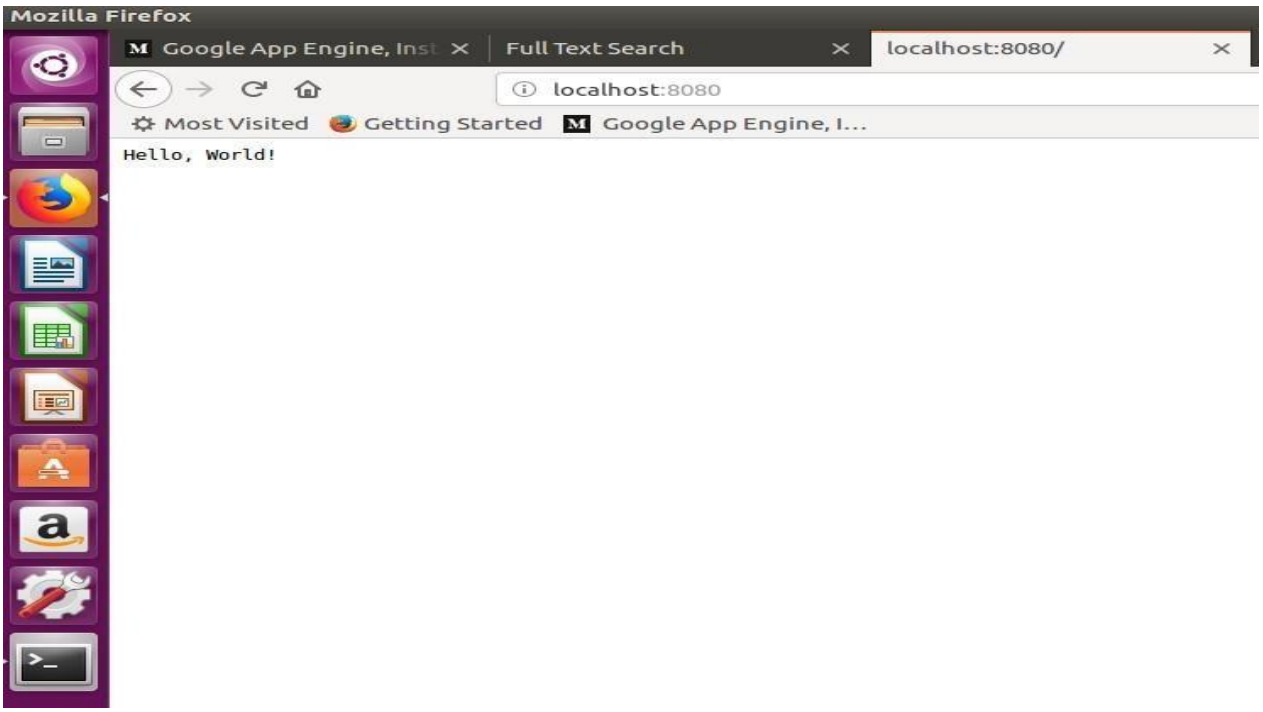
2. **cd python docs- samples/appengine/standard/hello_world**

3. **dev_appserver.py app.yaml**

It will run and give you the url of default and admin. If you go to the link of default you see the text hello world like this.

Output :


Full Text Search - Mozilla Firefox
Google App Engine, Inst
Full Text Search
localhost:8080/
localhost:8000/search
Most Visited
Getting Started
Google App Engine, I...
Google App Engine
dev~None
Instances
Datastore Viewer
Datastore Indexes
Datastore Stats
Interactive Console
Memcache Viewer
Blobstore Viewer
Task Queues
Cron Jobs
XMPP
Inbound Mail
Full Text Search
Full Text Search
Showing indexes in default namespace. Select different namespace
No full text search indexes found in the default namespace.


Mozilla Firefox
Google App Engine, Inst
Full Text Search
localhost:8080/
localhost:8080
Most Visited
Getting Started
Google App Engine, I...
Hello, World!

OR

Google App Engine, Installing it the right way in ubuntu.

1. Make sure you have python installed in your ubuntu system. run the command "*python -V*" and most probably you will get "Python 2.7.6" or above.
2. Crul `https://sdk.cloud.google.com` and use bash to run the commands by typing this command `curl https://sdk.cloud.google.com | bash`
3. Whenever you get to choose directories just hit enter, "YEAH IT WILL BE FINE".
4. Follow the instructions in the installation process.
5. Then run `gcloud init`
6. Follow the installation instructions as they are very straight forward.
7. Choose the account you want to use for google app engine.
8. Choose the project with numeric choice (don't use textual, you might make mistake). If you do not already have a google app engine project create a app engine project by following this link. https://console.cloud.google.com/start
9. Enable google api by pressing Y in the command line prompt.

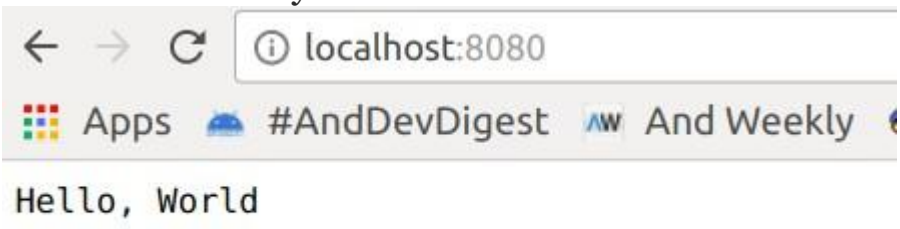
Now as we have finished installing appengine, now it's time to create and upload an app. In this case we will be taking example of a "HELLO WORLD" app in python.

1. As we already have made sure that we have python installed in our system, It will be easier for us to clone existing code and deploy it rather than creating our own so we will use python-docs-sample. Run the command "git

clone https://github.com/GoogleCloudPlatform/python-docs-samples".

2. cd to hello world sample by typing the command " cd python-docs-samples/appengine/standard/hello_world".

3. Then run the command "dev_appserver.py app.yml". It will run and give you the url of default and admin. If you go to the link of default you see the text hello world like this.



This is how you run the python app in your local server. But what we have to do is hosting the app in google app engine. To do so Now let's follow the following instructions.

1. Run the command *Ctrl + C* .

2. Being in the same working directory hello-world runt he command
gcloud app deploy

3. Select the project you want to deploy the app , press Y and enter to continue. after that you will get the console output "Deployed service[default] to [Your web url for appengine] "

4. If you copy and paste the url, you will see the hello world in the browser too.

Conclusion:

Thus, we have installed and Configured Google App Engine.