

Practical No.09

Aim: Study of GCP Console: Compute Engine

- Running a basic Apache web server under GCP
- Deploy basic Apps over the web server.
- Host sample static website

Theory:

- Google Cloud Platform (GCP) Compute Engine is a versatile and scalable Infrastructure as a Service (IaaS) solution that allows businesses and developers to run virtual machines in the cloud.
- With Compute Engine, users can create and manage virtual instances with a high degree of customization, choosing from a variety of machine types and configurations to match their specific needs.
- These virtual machines run on Google's global network infrastructure, providing excellent reliability and low-latency performance.
- One of the key benefits of GCP Compute Engine is its flexibility.
- Users can select from a wide range of virtual machine instance types, including general-purpose, memory-optimized, and CPU-optimized, to meet the demands of their workloads.
- These instances can be used for a variety of purposes, from hosting web applications and databases to running high-performance computing tasks.
- GCP Compute Engine also offers features such as auto-scaling, which automatically adjusts the number of instances based on demand, ensuring efficient resource utilization and cost savings.
- Security and management are top priorities with GCP Compute Engine.
- Google's infrastructure is designed with security in mind, and Compute Engine provides features like firewall rules, virtual private clouds, and encryption to safeguard data and resources.
- Additionally, it integrates seamlessly with other GCP services, making it easy to build and deploy applications with a variety of tools and services within the GCP ecosystem.
- Whether you are a small startup or a large enterprise, GCP Compute Engine provides the infrastructure needed to run your workloads efficiently and reliably in the cloud.

Running a basic Apache web server under GCP:

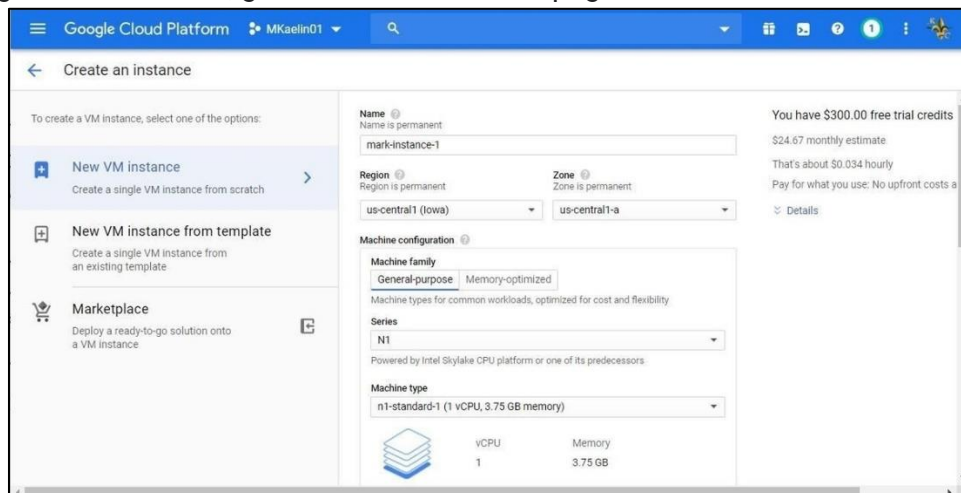
- Running a basic Apache web server on Google Cloud Platform (GCP) is a straightforward process. GCP offers a Compute Engine service that allows you to create and manage virtual machines in the cloud.
- To set up an Apache web server, you can start by creating a virtual machine instance with the desired operating system and configuration. Once your instance is up and running, you can easily install and configure the Apache web server software, enabling you to host websites, applications, or content on the internet.
- GCP's user-friendly interface and extensive documentation make it accessible for users of all skill levels to deploy and manage a basic Apache web server with ease.

A VM on Compute Engine can be controlled like any standard Linux server. Deploy an Apache web server to learn the basics of running a server on a VM instance.

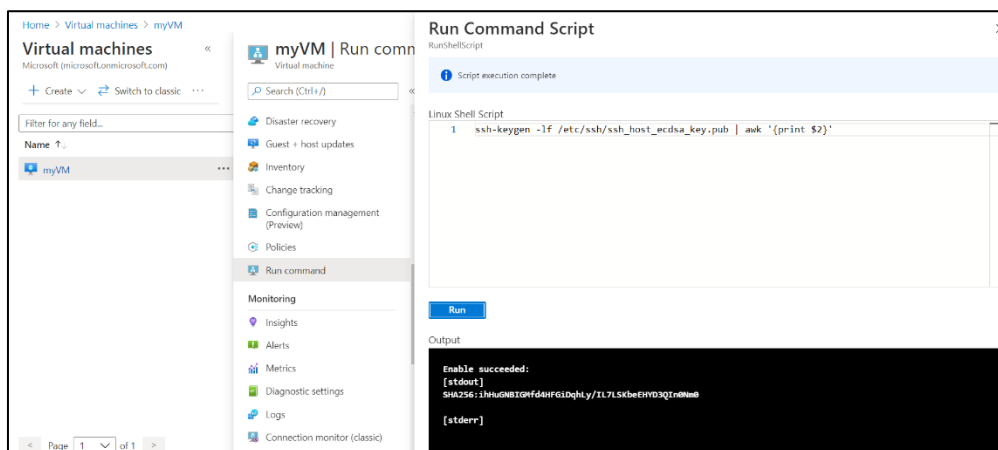
Steps for Running Apache Web Server:

Install Apache:

1. In the Google Cloud console, go to the **VM Instances** page.



2. To connect to the Linux VM you just created, click **SSH** in the row of the VM.



3. To update the available packages and install the apache2 package, use the system package manager for that operating system. If you followed the Quickstart, this creates an Ubuntu VM. To update an Ubuntu VM, run the following command:

sudo apt update && sudo apt -y install apache2

4. Verify that Apache is running:

sudo systemctl status apache2

5. Overwrite the Apache web server default web page:

echo '<!doctype html><html><body><h1>Hello World!</h1></body></html>' | sudo tee /var/www/html/index.html



Deploy basic Apps over the web server and hosting a website

- You can easily deploy basic web applications on Google Cloud Platform (GCP) by utilizing their web server infrastructure.
- GCP provides a straightforward and reliable environment for hosting and running web applications, allowing you to showcase your apps to a global audience over the internet.
- Whether it's a simple website, an online portfolio, or a small web-based tool, GCP's web server capabilities make it accessible and efficient to deploy your applications and share them with users worldwide.

Steps for Running Apache Web Server:

Django Application Deployment using GCP:

1. For deploying firstly download and install google cloud sdk on your machine . After successful installation you should see similar output by running “gcloud version” command.
2. For django we will require production server like *gunicorn so install gunicorn*.
3. Google App Engine uses app.yaml file for configuring app engine instance so lets create *app.yaml* in folder where *manage.py* file is located. Insert following code in app.yaml file.
Change ‘*gfg.wsgi:application*’ to *<YOUR_APP_NAME>.wsgi:application* .

```
runtime: python310
env: standard
entrypoint: gunicorn -b :8081 gfg.wsgi:application

handlers:
- url: /*
  script: auto

runtime_config:
  python_version: 3
```

4. create a *.gcloudignore* file which specifies which files to be not copied to instance . The file should look like below:

```
# .gcloudignore

.gcloudignore

# Ignore local .env file
.env

# If you would like to upload your .git directory, .gitignore file, or files
# from your .gitignore file, remove the corresponding line
# below:
.git
.gitignore

# Python pycache:
__pycache__/

# Ignore collected static and media files
mediafiles/
staticfiles/

# Ignore the local DB
db.sqlite3

# Ignored by the build system
/setup.cfg
venv/
```

5. Run below command where *app.yaml* file is located which will deploy our django application.

`gcloud app deploy`

6. Review details press Y to continue . After successful deployment you will see below output:

```
C:\Users\      \Desktop\New folder\gfg>gcloud app deploy
Services to deploy:

descriptor:          [C:\Users\Deepak\Desktop\New folder\gfg\app.yaml]
source:              [C:\Users\Deepak\Desktop\New folder\gfg]
target project:      [gfg-django]
target service:      [default]
target version:      [20230814t201214]
target url:          [https://gfg-django.ue.r.appspot.com]
target service account: [gfg-django@appspot.gserviceaccount.com]

Do you want to continue (Y/n)? y

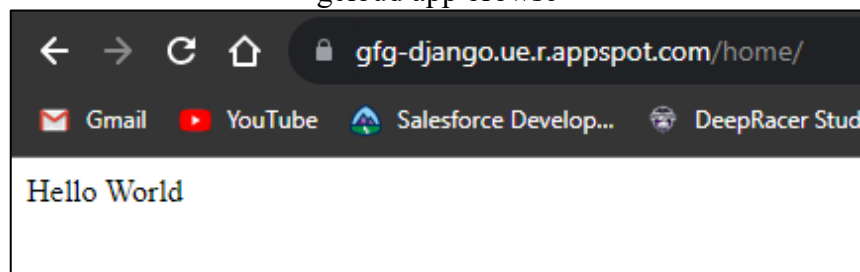
Beginning deployment of service [default]...
#=====#
#= Uploading 9 files to Google Cloud Storage      =#
#=====#
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://gfg-django.ue.r.appspot.com]

You can stream logs from the command line by running:
$ gcloud app logs tail -s default

To view your application in the web browser run:
$ gcloud app browse
```

7. Now to go to browser and go to service url of application OR run below command .

`gcloud app browse`



In above web application we've successfully deployed website with proper GCP console.

Conclusion: From this practical, we learnt about the deployment mechanism with GCP Console with implementation in apache server.