

LAB CYCLE 1

Date:

EXPERIMENT 1 : Install Virtual Machines using VirtualBox

Theory :

VirtualBox is an open-source virtualization software that allows users to run multiple operating systems on a single physical machine simultaneously. It works by creating virtual machines that simulate the hardware environment required by different Operating Systems. Learning VirtualBox is essential because virtualization is widely used in IT for testing, development, training, and server consolidation. By understanding how to set up and manage virtual machines, we gain practical skills in system configuration, OS installation, and resource allocation. This also helps in experimenting with different OS flavors without affecting the host machine's stability or data.

Question :

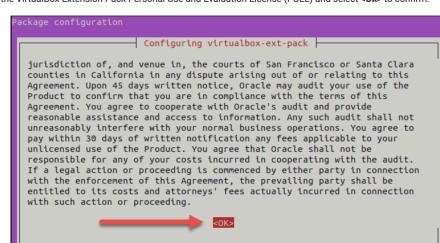
Install Virtualbox/VMware Workstation with different flavors of linux or windows OS on top of windows7 or 8.

Steps :

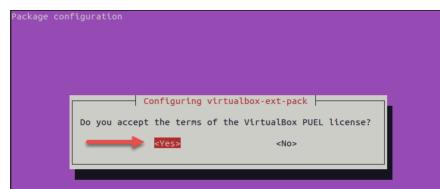
I. Open terminal:

```
sudo apt update  
sudo apt install virtualbox  
sudo apt install virtualbox-ext-pack
```

Read the VirtualBox Extension Pack Personal Use and Evaluation License (PUEL) and select <Ok> to confirm.



Accept the VirtualBox PUEL license terms by selecting <Yes> and hitting Enter.

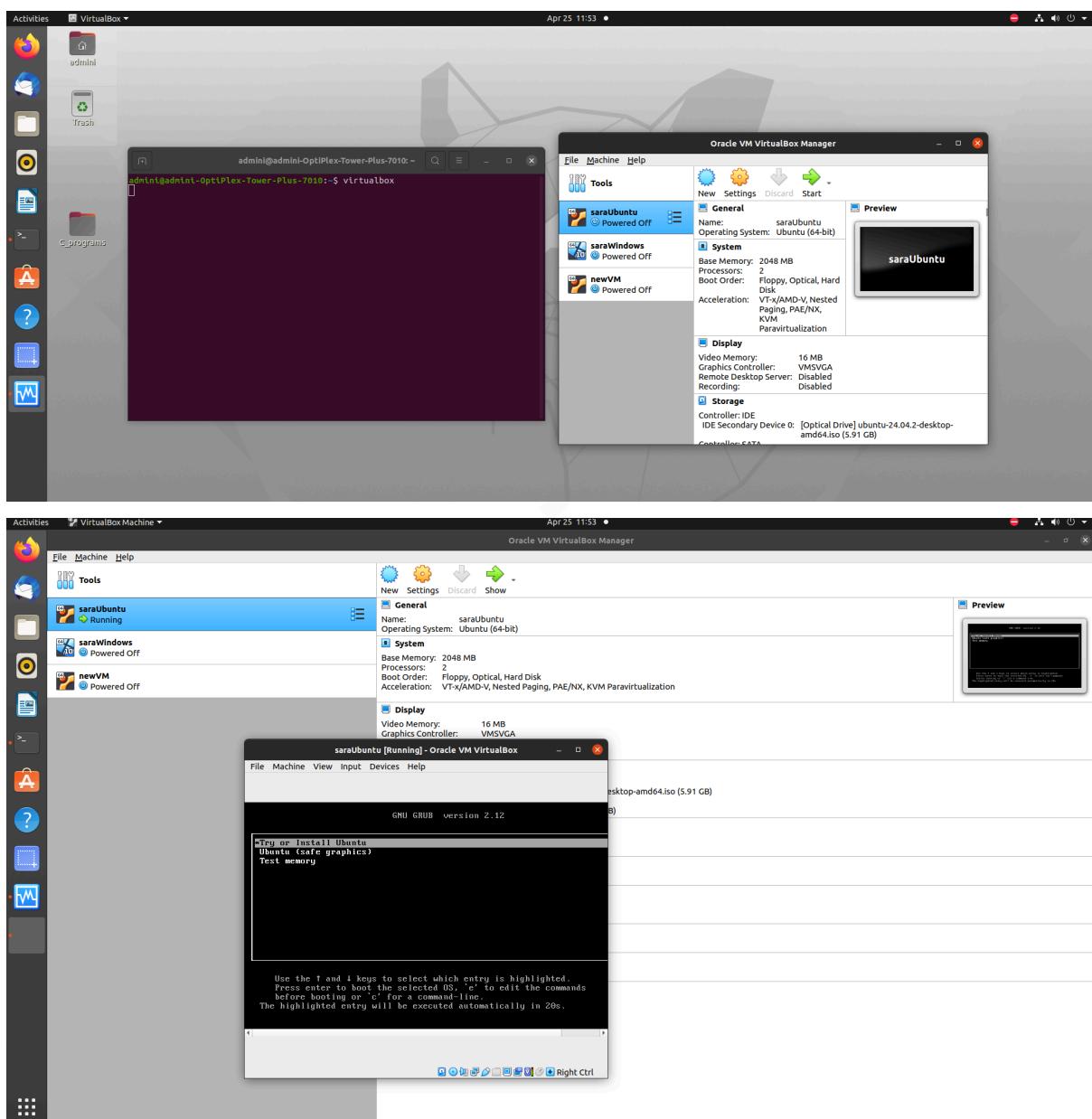


virtualbox

- II. Once installed, VirtualBox allows users to create and manage virtual machines. The software lets users run multiple guest operating systems concurrently on the host machine.
 - III. Create Your First Virtual Machine: VirtualBox's primary function is to install and run an operating system within another operating system by creating virtual machines.
 - a. Launch the VirtualBox interface: Open terminal: **virtualbox**
 - b. In Oracle VM VirtualBox Manager Interface, Click the New button to create a new virtual machine.
 - c. Type the OS Name. The Type and Version are filled out automatically based on the name. However, change them manually if needed.
 - d. In the same window, the system prompts you to select an ISO file to boot from. This ISO file can be obtained from the official website of the operating system you wish to install (e.g., Ubuntu, Windows, etc.). Choose the file and click Next. (This screen may or may not be visible)
 - e. Choose the username, password. Once completed, click Next. (Not required for certain setup)
 - f. Allocate the base memory and the virtual CPU count. Make sure the sliders stay on the green side to ensure the allocated resources are within the recommended range. This step ensures optimal performance of both the host and guest operating systems.
 - g. Mark the Create a Virtual Hard Disk Now checkbox, adjust the Disk Size slider, and click Next.
 - h. Check all the settings in the Summary section and click Finish. Once finished, the machine starts installing and configuring.
 - i. Once installation completes, a new virtual machine appears in the left column of VirtualBox Dashboard. Select it and click the green arrow Start button.
 - j. A new window opens and boots up the virtual machine.
 - k. Another way of adding ISO file:
 - i. Open Settings of the Virtual OS and go to Storage.
 - ii. Under Controller:IDE, Add Optical Drive and Upload ISO file from your host system (Downloads folder)
- IV. Virtualbox vs VMware: Comparison -
<https://phoenixnap.com/kb/virtualbox-vs-vmware>

Conclusion :

Through this experiment, I learned how to install and configure VirtualBox to run multiple virtual machines on a single host system. I gained hands-on experience in creating virtual environments, allocating resources, and installing different operating systems without altering the host OS. This exercise improved my understanding of virtualization concepts and prepared me to use such tools for testing, learning, and deployment in real-world scenarios.



Date:

EXPERIMENT 2 : Execute C Programs in the Virtual Machines

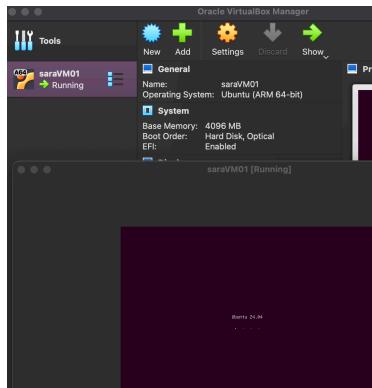
Theory :

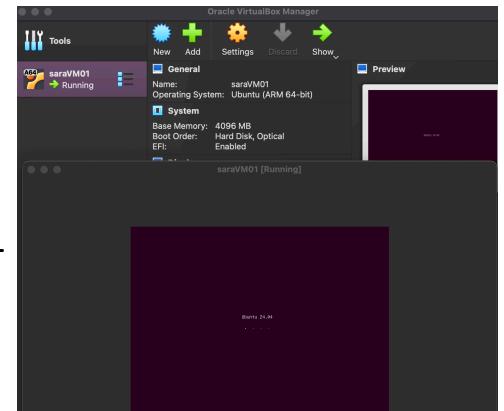
A C compiler is a program that translates C source code into executable machine code. Popular examples include GCC (GNU Compiler Collection) and MinGW. Installing a C compiler inside a virtual machine allows us to develop and run programs in an isolated environment, ensuring that the host system remains unaffected. This approach is useful for learning programming, testing software, and working with different OS setups without compatibility issues. By combining virtualization with programming, we can simulate real-world development environments and gain flexibility in testing across multiple platforms.

Question :

Install a C compiler in the virtual machine created using virtual box and execute simple programs

Steps :

- I. Open Ubuntu VM created on VirtualBox. Choose the option “Try/Install Ubuntu”.
 - II. Open terminal:
`sudo apt update`
`sudo apt install build-essential`
`gcc --version`
Install Text editor in VM
`sudo apt install gedit`
 - III. Execute 3 C programs inside the VM:
 - a. Use the touch command in the terminal to create a C program file.
`touch program1.c`
`touch program2.c`
`touch program3.c`



- b. Use the gedit command in the terminal to edit your C program file or use Text editor.

gedit program1.c

- c. Execute the C Programs.

gcc program1.c -o output1

- d. Run the executable OUTPUT file in the terminal using:

./output1

- IV. To make sure you can Drag and Drop from HOST OS to GUEST OS and vice versa. Please enable “Bidirectional” Option in General -> Advanced tab of VM Settings.

Conclusion :

From this experiment, I learned how to set up a programming environment inside a virtual machine by installing a C compiler. I successfully wrote, compiled, and executed simple C programs, which helped me understand the workflow from source code to execution. This task reinforced my understanding of compiler usage and demonstrated how virtualization can provide a safe, controlled environment for software development and testing.

```
sara@eccmtech:~$ gcc --version
gcc (Ubuntu 13.3.0-6ubuntu2-24.04) 13.3.0
Copyright (C) 2023 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

sara@eccmtech:~$ sudo apt update
[sudo] password for sara:
Get:1 https://packages.microsoft.com/repos/code stable InRelease [3,590 B]
Get:2 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Get:3 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:4 https://packages.microsoft.com/repos/code/stable/main armhf Packages [20.1 kB]
Get:5 https://dl.google.com/linux/chrome/deb/stable/main amd64 Packages [1,215 B]
Hit:6 http://in.archive.ubuntu.com/ubuntu noble InRelease
Get:7 https://packages.microsoft.com/repos/code/stable/main amd64 Packages [19.9 kB]
Get:8 https://packages.microsoft.com/repos/code/stable/main arm64 Packages [20.0 kB]
Hit:9 https://repo.waydro.id noble InRelease
Get:10 https://brave-browser-apt-release.s3.brave.com stable InRelease [7,546 B]
Get:11 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:12 https://brave-browser-apt-release.s3.brave.com stable/main amd64 Packages [23.2 kB]
Get:13 https://brave-browser-apt-release.s3.brave.com stable/main arm64 Packages [23.2 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1,054 kB]
Get:15 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Hit:16 https://repo.radeon.com/amdgpu/6.4.1/ubuntu noble InRelease
Get:17 http://in.archive.ubuntu.com/ubuntu noble-updates/main i386 Packages [506 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.5 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.3 kB]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Hit:22 https://repo.radeon.com/rocm/apt/6.4.1 noble InRelease
Ign:23 https://packages.openvpn.net/openvpn3/debian DISTRIBUTION InRelease
Get:24 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,313 kB]
```

```
sara@eccmtech:~$ sudo apt install gcc
[sudo] password for sara:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gcc is already the newest version (4:13.2.0-7ubuntu1).
gcc set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 178 not upgraded.
```

```
sara@eccmtech: ~/Videos/C_Programming
sara@eccmtech:~/Videos/C_Programming$ cat program1.c
#include <stdio.h>
int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);

    if(num % 2 == 0)
        printf("%d is even.", num);
    else
        printf("%d is odd.", num);

    return 0;
}sara@eccmtech:~/Videos/C_Programming$ gcc program1.c -o ./output1.c
sara@eccmtech:~/Videos/C_Programming$ ./output1.c
Enter a number: 2
2 is even.sara@eccmtech:~/Videos/C_Programming$
```

```
sara@eccmtech: ~/Videos/C_Programming
sara@eccmtech:~/Videos/C_Programming$ cat program2.c
#include <stdio.h>
int main() {
    int t1 = 0, t2 = 1, nextTerm = 0, n;
    printf("Enter a positive number upto which Fibonacci sequence to be displayed: ");
    scanf("%d", &n);

    printf("Fibonacci Series: %d, %d, ", t1, t2);
    nextTerm = t1 + t2;

    while (nextTerm <= n) {
        printf("%d, ", nextTerm);
        t1 = t2;
        t2 = nextTerm;
        nextTerm = t1 + t2;
    }

    return 0;
}sara@eccmtech:~/Videos/C_Programming$ gcc program2.c -o output2.c
sara@eccmtech:~/Videos/C_Programming$ ./output2.c
Enter a positive number upto which Fibonacci sequence to be displayed: 14
Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, sara@eccmtech:~/Videos/C_Programming$
```

```
sara@eccmtech: ~/Videos/C_Programming
sara@eccmtech:~/Videos/C_Programming$ cat program3.c
#include <stdio.h>

int main(void)
{
    for (unsigned int i = 1; i <= 100; i++) {
        if (i % 15 == 0) {
            printf("FizzBuzz\n");
        } else if (i % 3 == 0) {
            printf("Fizz\n");
        } else if (i % 5 == 0){
            printf("Buzz\n");
        } else {
            printf("%u\n", i);
        }
    }
    return 0;
}sara@eccmtech:~/Videos/C_Programming$ gcc program3.c -o output3.c
sara@eccmtech:~/Videos/C_Programming$ ./output3.c
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz
Fizz
22
```

Date:

EXPERIMENT 3 : Create Applications using Google App Engine

Theory :

Google App Engine (GAE) is a cloud-based platform-as-a-service (PaaS) provided by Google for building and hosting web applications. It supports multiple programming languages such as Python, Java, PHP, and Go. GAE automatically manages infrastructure, scaling, and load balancing, so developers can focus on coding rather than server maintenance. Learning GAE is important because it provides real-world exposure to cloud deployment, serverless architecture, and scalable application development. By building and deploying simple web applications on GAE, we gain practical experience with cloud services, application configuration, and version control for hosted projects.

Question :

Install Google App Engine. Create a hello world app and other simple web applications using Python/Java.

Steps :

- I. Install Google Cloud SDK on your system from the official Google Cloud website. Get the Google Cloud CLI at
<https://cloud.google.com/sdk/docs/install>

 Try Google Cloud for free

Welcome to the Google Cloud CLI!

To help improve the quality of this product, we collect anonymized usage data and anonymized stacktraces when crashes are encountered; additional information is available in our [Privacy Policy](#) and [Google Cloud Terms of Use](#).

This data is handled in accordance with our [privacy policy](#).

[View the latest privacy notice](#). You may choose to opt in this collection now or at any time via the prompt, or at any time in the future by running the following command:

```
gcloud config set disable_usage_reporting false
```

Do you want to help improve the Google Cloud CLI (y/N)? N

Your current Google Cloud CLI version is: 533.0.0
The latest available version is: 533.0.0

Components				
Status	Name	ID	Size	
Not Installed	App Engine Go Extensions	app-engine-go	4.5 MiB	
Not Installed	Applets	applets	1.8 MiB	
Not Installed	Artifact Registry Go Module Package Helper	artifact-go-module	1.3 MiB	
Not Installed	Bigtable Command Line Tool	cht	19.3 MiB	
Not Installed	Cloud Build	cloudbuild	5.1 MiB	
Not Installed	Cloud Datastore Emulator	cloud-datastore-emulator	63.7 MiB	
Not Installed	Cloud Firestore Emulator	cloud-firestore-emulator	42.9 MiB	
Not Installed	Cloud Functions Emulator	cloud-functions-emulator	11.3 MiB	
Not Installed	Cloud Run Proxy	cloud-run-proxy	14.7 MiB	
Not Installed	Cloud SQL Proxy v2	cloud-sql-proxy	1.4 MiB	
Not Installed	Docker Credential Registry's Docker credential helper	docker-credential-dkr	7.4 MiB	
Not Installed	Kustomize	kustomize	1.4 MiB	
Not Installed	Local Testing	lbt	14.5 MiB	
Not Installed	Managed File Client	managed-filclient	383.4 MiB	
Not Installed	Minikube	minikube	45.9 MiB	
Not Installed	Minikube API	minikube-api	13.5 MiB	
Not Installed	On-Demand Scanning API extraction helper	local-extract	23.4 MiB	
Not Installed	Spanner CAPI	spanner-capi	2.4 MiB	
Not Installed	Spanner CLI	spanner-cli	12.2 MiB	
Not Installed	Terraform Tools	terraform-tools	64.8 MiB	
Not Installed	TensorFlow	tensorflow	34.9 MiB	
Not Installed	config-connector	config-connector	138.7 MiB	
Not Installed	enterprise-certIFICATE-proxy	enterprise-certIFICATE-proxy	7.6 MiB	
Not Installed	enterprise-COMMANDS	enterprise-COMMANDS	1.9 MiB	
Not Installed	gcloud Beta Commands	beta	5 MiB	
Not Installed	gcloud Compute Extensions	compute	137.4 MiB	
Not Installed	gcloud app Python Extensions	app-engine-python	3.8 MiB	
Not Installed	gcloud app Python Extensions (Extra Libraries)	app-engine-python-extras	1.3 MiB	
Not Installed	gcloud auth-outside-plugin	auth-outside	1.3 MiB	
Not Installed	linter	linter	20.4 MiB	
Not Installed	Minikube	minikube	1.4 MiB	
Not Installed	kubectl	kubectl	29.9 MiB	
Not Installed	kubectl-oidc	kubectl-oidc	2.1 MiB	
Installed	BigQuery Command Line Tool	bq	1.8 MiB	
Installed	Cloud Storage Command Line Tool	gsutil	12.4 MiB	
Installed	Cloud Spanner C API Libraries	gcsfuse	22.1 MiB	
Installed	Google Cloud CRC32C Hash Tool	gcloud-crc32c	1.4 MiB	

To install or remove components at any time, run `gcloud components`.

II. Open a terminal and initialize the SDK using :

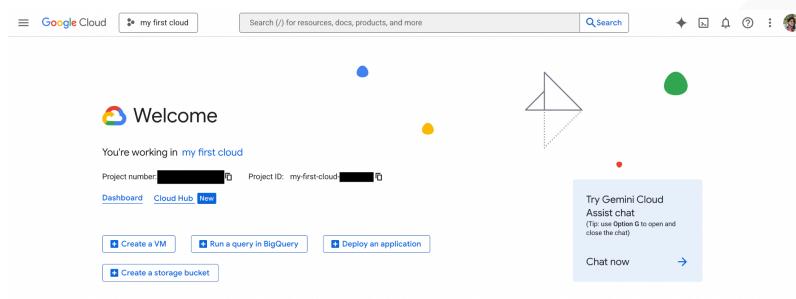
```
gcloud init
```

```
Last login: [REDACTED] on ttys000
[aneettasara@[REDACTED] ~] google-cloud-sdk % gcloud --version
Google Cloud SDK 533.0.0
bq 2.1.22
core 2025.08.01
gcloud-crc32c 1.0.0
gsutil 5.35
[aneettasara@[REDACTED] ~] google-cloud-sdk % gcloud init
Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [default] are:
core:
  account: [REDACTED]
  disable_usage_reporting: 'True'

Pick configuration to use:
[1] Re-initialize this configuration [default] with new settings
[2] Create a new configuration
Please enter your numeric choice: [REDACTED]
```

III. Create a new project in Google Cloud Console and note the Project ID.



IV. Install the App Engine component using :
While using Python -

```
gcloud components install app-engine-python
```

```
aneettasara@[REDACTED] ~ google-cloud-sdk % gcloud components install app-engine-python
Your current Google Cloud CLI version is: 533.0.0
Installing components from version: 533.0.0

These components will be installed.
+---+-----+-----+
| Name | Version | Size |
+---+-----+-----+
| Cloud Datastore Emulator | 2.3.1 | 36.2 MiB |
| gRPC Python library (Platform Specific) | 1.78.0 | < 1 MiB |
| gcloud app Python Extensions | 1.9.117 | 3.8 MiB |
| gcloud cli dependencies (Platform Specific) | 2021.04.16 | < 1 MiB |

For the latest full release notes, please visit:
https://cloud.google.com/sdk/release_notes

Once started, canceling this operation may leave your SDK installation in an
inconsistent state.

Do you want to continue (Y/n)? y
Performing in place update...

+ Downloading: Cloud Datastore Emulator
+ Downloading: gRPC Python library
+ Downloading: gRPC Python library (Platform Specific)
+ Downloading: gcloud app Python Extensions
+ Downloading: gcloud cli dependencies (Platform Specific)
+ Installing: Cloud Datastore Emulator
+ Installing: gRPC Python library
+ Installing: gRPC Python library (Platform Specific)
+ Installing: gcloud app Python Extensions
+ Installing: gcloud cli dependencies (Platform Specific)

Performing post processing steps...done.
Google Cloud CLI works best with Python 3.12 and certain modules.
Python 3.12 installation detected in
/Library/Frameworks/Python.framework/Versions/3.12/bin/python3, install
recommended modules? (Y/n)? [REDACTED]
```

While using Java -

```
gcloud components install app-engine-java
```

V. Create a working directory for your app.

VI. Write a simple “Hello World” application :

While using Python - Create main.py with basic Flask code.

The screenshot shows a terminal window with the following content:

```
main.py
1 from flask import Flask
2
3 app = Flask(__name__)
4
5 @app.route('/')
6 def hello():
7     return "Hello, BVRITH! This is my first application on Google App Engine."
8
9 if __name__ == '__main__':
10    app.run(host='0.0.0.0', port=8080, debug=True)

PROBLEMS OUTPUT TERMINAL PORTS TRUFFLE DEBUG CONSOLE
aneettasara@... my_first_gcodes % python3 -m venv my-env1
aneettasara@... my_first_gcodes % source my-env1/bin/activate
(my-env1) aneettasara@... my_first_gcodes % pip install flask
Collecting flask
  Using cached flask-3.1.1-py3-none-any.whl.metadata (3.0 kB)
Collecting blinker>=1.9.0 (from flask)
  Using cached blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting click>=8.1.3 (from flask)
  Using cached click-8.2.1-py3-none-any.whl.metadata (2.5 kB)
Collecting itsdangerous>=2.2.0 (from flask)
  Using cached itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting jinja2>=3.1.2 (from flask)
  Using cached jinja2-3.1.2-py3-none-any.whl.metadata (2.9 kB)
Collecting markupsafe<2.1.1 (from flask)
  Downloading MarkupSafe-3.0.2-cp313-cp313-macosx_11_0_arm64.whl.metadata (4.0 kB)
Collecting werkzeug>=3.1.0 (from flask)
  Using cached werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Using cached flask-3.1.1-py3-none-any.whl (103 kB)
Using cached blinker-1.9.0-py3-none-any.whl (8.5 kB)
Using cached click-8.2.1-py3-none-any.whl (102 kB)
Using cached itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Using cached jinja2-3.1.6-py3-none-any.whl (134 kB)
Downloaded MarkupSafe-3.0.2-cp313-cp313-macosx_11_0_arm64.whl (12 kB)
Using cached werkzeug-3.1.3-py3-none-any.whl (224 kB)
Successfully installed packages markupsafe itsdangerous click blinker werkzeug jinja2 flask
[notice] A new release of pip is available: 25.1.1 -> 25.2
[notice] To update, run: pip install --upgrade pip
```

While using Java - Create HelloServlet.java with a simple servlet.

VII. Add the `app.yaml` configuration file with runtime and environment settings.

The screenshot shows a terminal window with the following content:

```
! app.yaml
1 runtime: python313
2
3 handlers:
4   - url: /.*
5     script: auto

my_first_gcodes
! app.yaml
main.py
requirements.txt
```

The terminal shows the output of running the application:

```
127.0.0.1:8080
Hello, BVRITH! This is my first application on Google App Engine.
```

VIII. Open Terminal and Deploy the app to Google App Engine using :

`gcloud config set account ACCOUNT`

`gcloud config set project VALUE`

`gcloud app deploy ./my_first_gcodes/app.yaml`

```

gcloud app deploy --project my-first-google-app.yaml
You are creating an app for project [my-first-cloud-44641].
WARNING: Creating an App Engine application for a project is irreversible and the region
cannot be changed. Learn more at
https://cloud.google.com/appengine/docs/location.

WARNING: Starting from March, 2023, App Engine sets the automatic scaling maximum instances
to 20 by default. If you have existing apps, you can change this behavior by providing a value for automatic_scaling.max_instances. Starting from March, 2023, App Engine sets the automatic
scaling maximum instances default for standard environment deployments to 20. This change doesn't impact existing apps. To override the default, specify the new max_instances value in your app.yaml file.
For more details on max_instances, see
https://cloud.google.com/appengine/docs/standard/python/config/appref#max_instances.

NOTE: Cloud Run will reflect any modern fully managed application hosting experience
with lower minimum billable times and support for GPU on demand for your AI/ML workloads.
Cloud Run is currently in preview. Learn more about Cloud Run at
https://cloud.google.com/run/docs/concepts/build-and-deploy-a-web-service

Please choose the region where you want your App Engine application located:
[1] asia-east1 (supports standard and flexible)
[2] asia-north1 (supports standard and flexible and search_api)
[3] asia-northeast1 (supports standard and flexible and search_api)
[4] asia-south1 (supports standard and flexible and search_api)
[5] asia-northwest1 (supports standard and flexible and search_api)
[6] asia-southeast1 (supports standard and flexible and search_api)
[7] asia-southwest1 (supports standard and flexible and search_api)
[8] asia-southeast2 (supports standard and flexible and search_api)
[9] australia-southeast1 (supports standard and flexible and search_api)
[10] europe-central2 (supports standard and flexible and search_api)
[11] europe-west1 (supports standard and flexible and search_api)
[12] europe-west2 (supports standard and flexible and search_api)
[13] europe-west3 (supports standard and flexible and search_api)
[14] europe-west4 (supports standard and flexible and search_api)
[15] northamerica-northeast1 (supports standard and flexible and search_api)
[16] us-central1 (supports standard and flexible and search_api)
[17] us-central2 (supports standard and flexible and search_api)
[18] us-east1 (supports standard and flexible and search_api)
[19] us-east4 (supports standard and flexible and search_api)
[20] us-west1 (supports standard and flexible and search_api)
[21] us-west2 (supports standard and flexible and search_api)
[22] us-west3 (supports standard and flexible and search_api)
[23] us-west4 (supports standard and flexible and search_api)

Please enter your numeric choice: 7B

```

IX. Open the deployed app in a browser using :

gcloud app browse

Google App will be enabled at the Google Cloud Project ID.

Conclusion :

This experiment helped me to set up Google App Engine and deploy simple web applications using Python or Java. I gained hands-on experience in creating project configurations, writing minimal application code, and hosting it in a cloud environment. This helped me understand the fundamentals of cloud-based application deployment, scalability, and how serverless platforms simplify infrastructure management.