DR. D. Y. PATIL INSTITUTE OF ENGINEERING, MANAGEMENT AND RESEARCH,

AKURDI, PUNE-44

Electronics & Telecommunication Department

LAB MANUAL

Subject – Cloud Computing

Subject code: 404186

Class – BE

Institute Vision:

To strive for excellence by providing quality technical education and facilitate research for

the welfare of society

Institute Mission:

1. To educate students with strong fundamentals by providing conducive environment.
2. To strengthen leadership, team-work, professional & communication skills and ethical standards
3. To promote industry-institute collaboration & prepare students for lifelong learning in context

   of technological change.
4. To inculcate research through creativity & innovation.

Department Vision:

To impart quality education to produce competent E&TC Engineers

Department Mission:

1. To equip students with strong basics through excellent blend of theory and practical knowledge
2. To develop students with communication skills and ethical standards to meet the professional needs
3. To give the knowledge about all possible areas of E&TC by interacting with professional world
4. To inculcate creativity and innovation through curricular and co-curricular activities

Program Educational Objectives (PEOs):

1. The graduate shall utilize the basic knowledge to address the Engineering problems
2. The graduate shall attain the qualities of professional leadership with ethical and moral standards
3. The graduate shall develop their capabilities for lifelong learning throughout their professional
   career and higher education
4. The graduate shall explore engineering capabilities through creativity and innovation

Program Specific Outcomes (PSOs):

1. Apply principles of Electronics and communication, digital systems, signal processing,
   software programming in the field of Embedded, Telecommunication & Software services for
   real world applications
2. Comprehend the technological advancements, demonstrate the proficiency in the usage of
   engineering tools to analyze and design systems for variety of applications.
3. Demonstrate professional ethics, apply communication skills for successful career and higher studies.

Program Outcomes:

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant

to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, andneed for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

University course Code	SAR course code	Course Outcomes	BT Level

404186	C416.1	Use GAE launcher to launch the web applications and simulate a cloud scenario using CloudSim	BT 3
	C416.2	Design and deploy a PaaS environment and find procedure to transfer the files from one virtual machine to another virtual machine	BT 6

INDEX

Experiment No. 1

Title: Install Google App Engine. Create hello world app and other simple web applications using python / java.

Software Required: Google App Engine Installer, Python 2.5.4

Procedure:

Pre--Requisites: Python 2.5.4

If you don't already have Python 2.5.4 installed in your computer, download and Install Python 2.5.4 from:

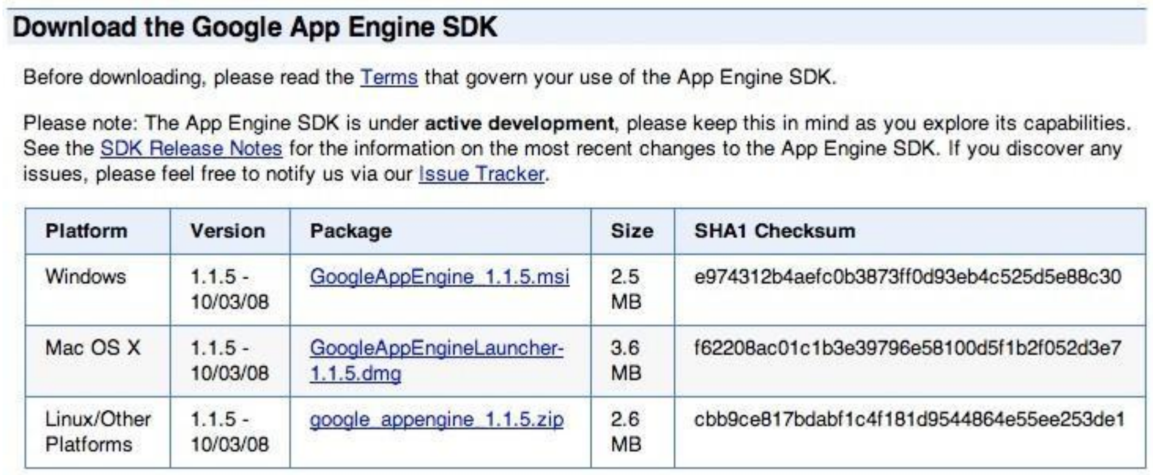
```
http://www.python.org/download/releases/2.5.4/
```

Download and Install

You can download the Google App Engine SDK by going to:

```
http://code.google.com/appengine/downloads.html
```

and download the appropriate install package.



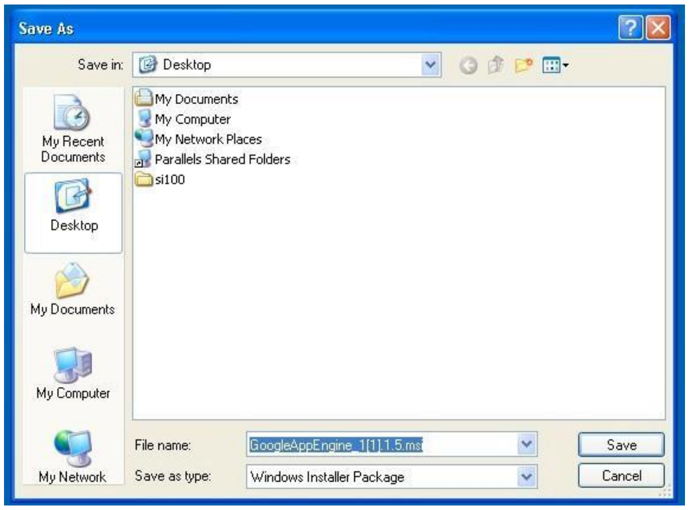
Download the Google App Engine SDK

Before downloading, please read the Terms that govern your use of the App Engine SDK.

Please note: The App Engine SDK is under **active development**, please keep this in mind as you explore its capabilities. See the SDK Release Notes for the information on the most recent changes to the App Engine SDK. If you discover any issues, please feel free to notify us via our Issue Tracker.

Platform	Version	Package	Size	SHA1 Checksum
Windows	1.1.5 - 10/03/08	GoogleAppEngine_1.1.5.msi	2.5 MB	e974312b4aefc0b3873ff0d93eb4c525d5e88c30
Mac OS X	1.1.5 - 10/03/08	GoogleAppEngineLauncher-1.1.5.dmg	3.6 MB	f62208ac01c1b3e39796e58100d5f1b2f052d3e7
Linux/Other Platforms	1.1.5 - 10/03/08	google_appengine_1.1.5.zip	2.6 MB	cbb9ce817bdabf1c4f181d9544864e55ee253de1

Download the Windows installer – the simplest thing is to download it to your Desktop or another folder that you remember.

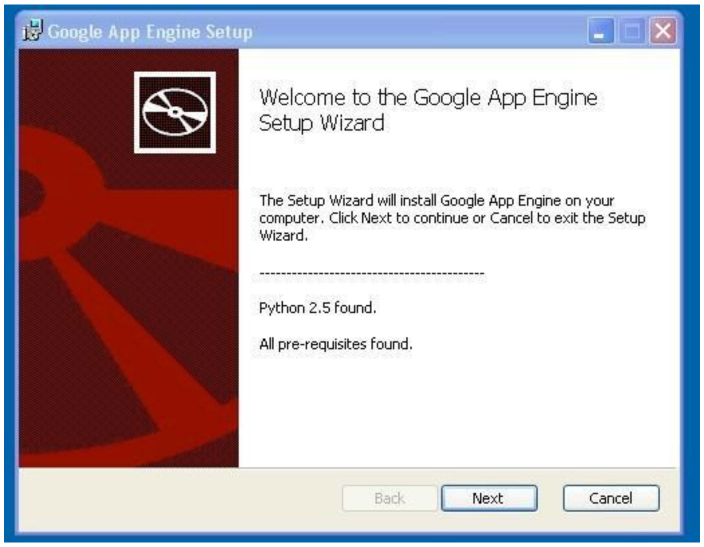


Save in: Desktop

My Documents
My Computer
My Network Places
Parallels Shared Folders
si100

File name: GoogleAppEngine_1[1].1.5.msi
Save as type: Windows Installer Package

Double Click on the **Google Application Engine** installer.



Welcome to the Google App Engine Setup Wizard

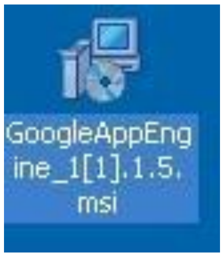
The Setup Wizard will install Google App Engine on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Python 2.5 found.

All pre-requisites found.

Click through the installation wizard, and it should install the App Engine. If you do not have Python 2.5, it will install Python 2.5 as well.

Once the install is complete you can discard the downloaded installer



Making your First Application

Now you need to create a simple application. We could use the "+" option to have the launcher make us an application – but instead we will do it by hand to get a better sense of what is going on.

Make a folder for your Google App Engine applications. I am going to make the Folder on my Desktop called "**apps**" – the path to this folder is:

C:\Documents and Settings\csev\Desktop\apps

And then make a sub---folder in within **apps** called "**ae--01--trivial**" – the path to this folder would be:

C:\ Documents and Settings \csev\Desktop\apps\ae--01--trivial

Using a text editor such as JEdit (www.jedit.org), create a file called **app.yaml** in the

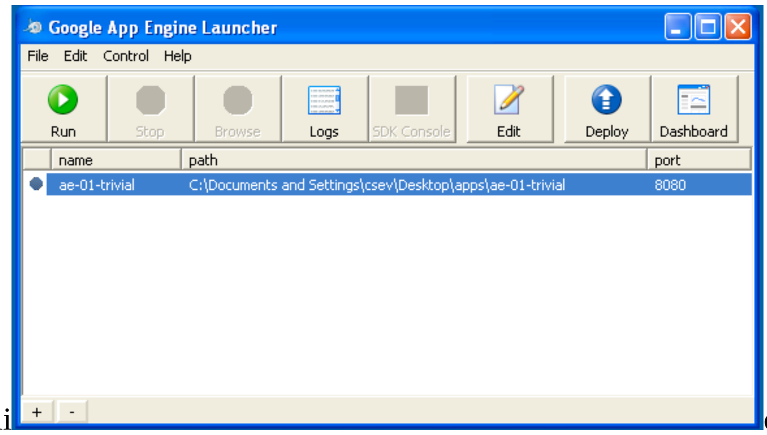
ae--01--trivial folder with the following contents:

```
application: ae-01-trivial version: 1

runtime: python api_version: 1


handlers:

- url: /.*

    script: index.py
```

Note: Please do not copy and paste these lines into your text editor – you might end up with strange characters – simply type them into your editor.

Then create a file in the **ae--01--trivial** folder called **index.py** with three lines in it:

```
print 'Content-Type: text/plain' print ' '

print 'Hello there Chuck'
```

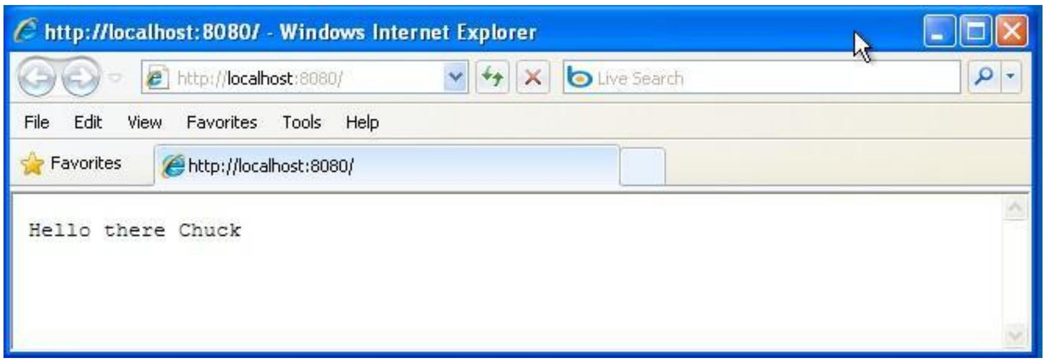
Then start the **Google App Engine Launcher** program that can be found under **Applications**. Use the **File --> Add Existing Application** command and navigate into the **apps** directory and select the **ae--01--trivial** folder. Once you have added the application, select it so that you can control the appli

name	path	port
ae-01-trivial	C:\Documents and Settings\csev\Desktop\apps\ae-01-trivial	8080

cation using the launcher.

Once you have selected your application and press **Run**. After a few moments your application will start and the launcher will show a little green icon next to your application. Then press **Browse** to open a browser pointing at your application which is running at **http://localhost:8080/**

Paste **http://localhost:8080** into your browser and you should see your application as follows:



http://localhost:8080/ - Windows Internet Explorer
http://localhost:8080/
Live Search
File Edit View Favorites Tools Help
Favorites
http://localhost:8080/

```
Hello there Chuck
```

Screen Shot of Output Page:

Conclusion:

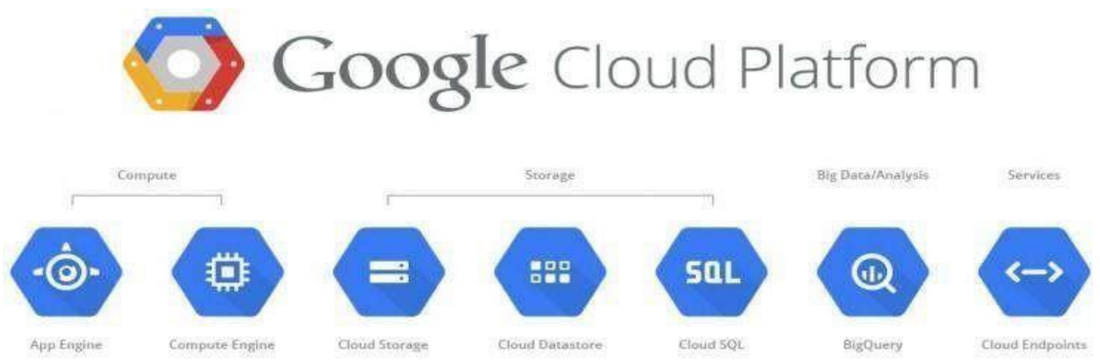
Experiment No. 2

Title: Use GAE launcher to launch the web applications

Software Required: Google App Engine Installer, Google cloud SDK installer, Python 2.5.4

Procedure:

- *Google Cloud Platform (GCP)*
  - **Google Cloud Platform** (**GCP**), offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube.
  - Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning.
  - Google Cloud Platform provides infrastructure as a service, platform as a service, and serverless computing environments.



➢ *Platform as a Service (PaaS)*
  - Cloud computing service which provides a computing platform and a solution stack as a service.
  - Consumer creates the software using tools and/or libraries from the provider.
  - Provider provides the networks, servers, storage, etc.

➢

Google App Engine:

- Google App Engine was first released as a beta version in April 2008.
- It is a is a Platform as a Service (PaaS) cloud computing platform for developing and hosting web applications in Google-managed data centers.
- Google's App Engine opens Google's production to any person in the world at no charge.
- Google App Engine is software that facilitates the user to run his web applications on Google infrastructure.
- It is more reliable because failure of any server will not affect either the performance of the end user or the service of the Google.
- It virtualizes applications across multiple servers and data centers.
    - Other cloud-based platforms include offerings such as Amazon Web Services and Microsoft's Azure Services Platform.

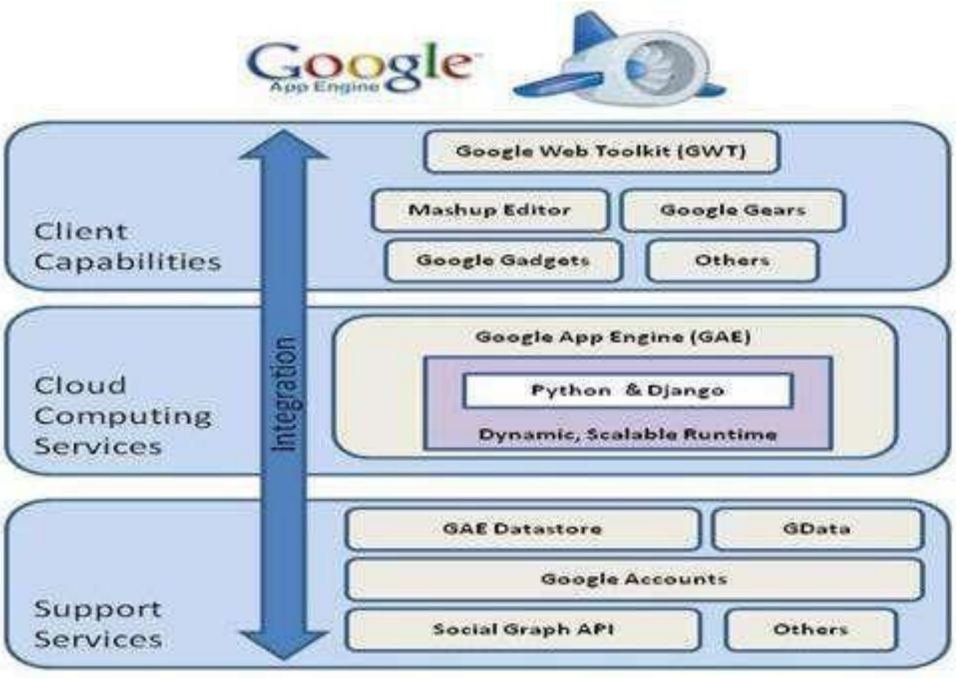
➢ *Introduction of Google App Engine*

- Google App Engine lets you run your web applications on Google's infrastructure. App Engine applications are easy to build, easy to maintain,and easy to scale as your traffic and data storage needs grow. With App Engine, there are no servers to maintain: You just upload your application, and it's ready to serve your users.
- You can serve your app from your own domain name (such as https://www.example.com/) using Google Apps. Or, you can serve your app using a free name on the appspot.com domain. You can share your application with the world, or limit access to members of your organization.
- Google App Engine supports apps written in several programming languages. With App Engine's Java runtime environment, you can build your app using standard Java technologies, including the JVM, Java servlets, and the Java programming language—or any other language using a JVM-based interpreter or compiler, such as JavaScript or Ruby. App Engine also features a dedicated Python runtime environment, which includes a fast Python interpreter and the

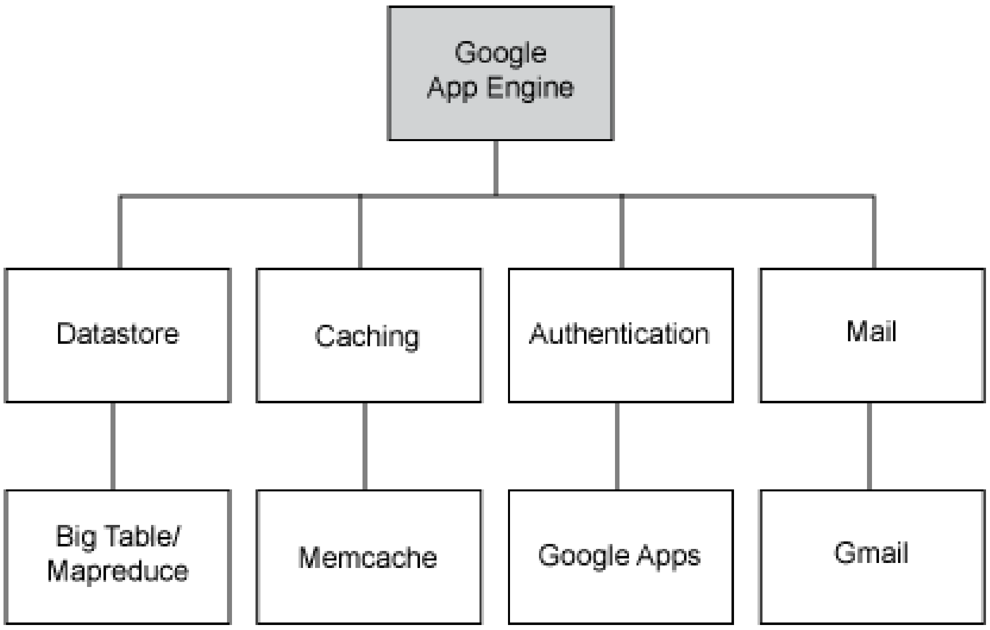
Python standard library. The Java and Python runtime environments are built to ensure that your application runs quickly, securely, and without interference from other apps on the system.

With App Engine, you only pay for what you use. There are no set-up costs and no recurring fees. The resources your application uses, such as storage and bandwidth, are measured by the gigabyte, and billed at competitive rates. You control the maximum amounts of resources your app can consume, so it always stays within your budget. App Engine costs nothing to get started. All applications can use up to 500 MB of storage and enough CPU and bandwidth to support an efficient app serving around 5 million page views a month, absolutely free. When you enable billing for your application, your free limitsare raised, and you only pay for resources you use above the free levels.

➢ *Architecture of Google App Engine*



- **Features of Google App Engine**



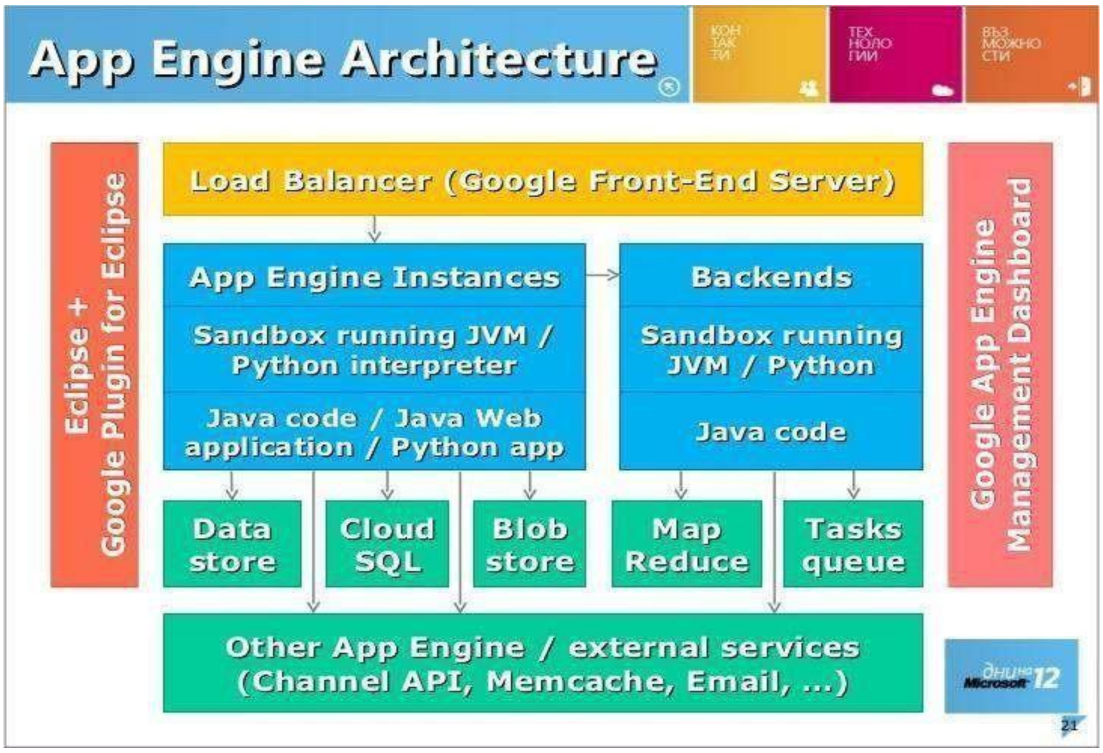
- *GAE Application Environment:*

  - Google App Engine makes it easy to build an application that runs reliably, even under heavy load and with large amounts of data. App Engine includes the following features:
  - Persistent storage with queries, sorting and transactions
  - Automatic scaling and load balancing
  - APIs for authenticating users and sending email using Google Accounts
  - Task queues for performing work outside of the scope of a web request
  - Scheduled tasks for triggering events at specified times and regular intervals
  - Dynamic web serving, with full support for common web technologies

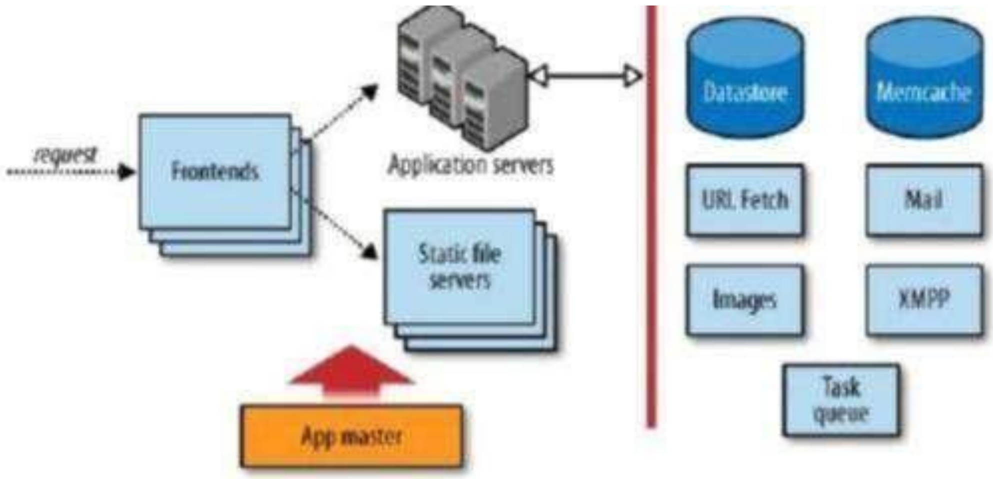
- *Java Runtime Environment*

  - You can develop your application for the Java runtime environment using common Java web development tools and API standards. Your app interacts with the environment using the Java Servlets standard, and can use common web application technologies such as Java Server Pages

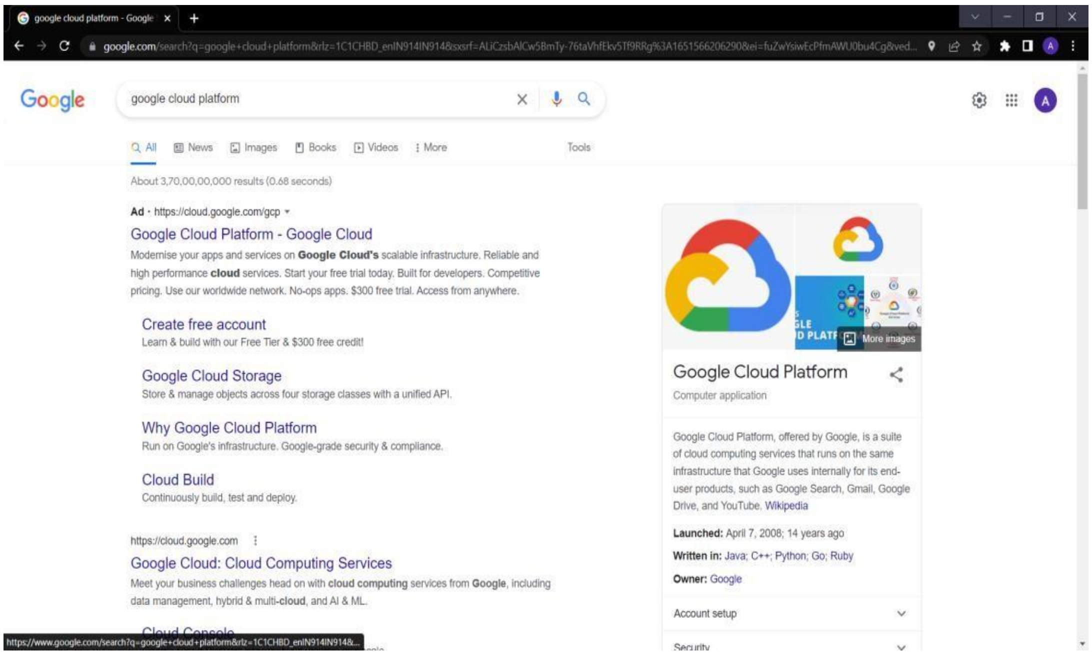
- The Java runtime environment uses Java 6. The App Engine Java SDK supports developing apps using either Java 5 or 6. The environment includes the Java SE Runtime Environment (JRE) 6 platform and libraries. The restrictions of the sandbox environment are implemented in the JVM. An app can use any JVM byte code or library feature, as long as it does not exceed the sandbox restrictions. For instance, byte code that attempts to open a socket or write to a file will throw a runtime exception.
- Your app accesses most App Engine services using Java standard APIs. For the App Engine data store, the Java SDK includes implementations of the Java Data Objects (JDO) and Java Persistence API (JPA) interfaces. Your app can use the JavaMail API to send email messages with the App Engine Mail service. The java.net HTTP APIs accesses the App Engine URL fetch service.
- App Engine also includes low-level APIs for its services to implement additional adapters, or to use directly from the application. See the documentation for the data store, memcache, URL fetch, mail, images and Google Accounts APIs. Typically, Java developers use the Java programming language and APIs to implement web applications for the JVM. With the use of JVM-compatible compilers or interpreters, you can also use other languages to develop web applications, such as JavaScript, Ruby


App Engine Architecture
КОНТАКТИ
ТЕХНОЛОГИИ
ВЪЗМОЖНОСТИ
Eclipse + Google Plugin for Eclipse
Load Balancer (Google Front-End Server)
App Engine Instances
Sandbox running JVM / Python interpreter
Java code / Java Web application / Python app
Backends
Sandbox running JVM / Python
Java code
Data store
Cloud SQL
Blob store
Map Reduce
Tasks queue
Other App Engine / external services (Channel API, Memcache, Email, ...)
Google App Engine Management Dashboard
дни на Microsoft 12
21

➢ *Workflow of Google App Engine*



.Step1 : Login to www.cloud.google.com



google cloud platform

All News Images Books Videos More Tools

About 3,70,00,00,000 results (0.68 seconds)

Ad · https://cloud.google.com/gcp

Google Cloud Platform - Google Cloud

Modernise your apps and services on **Google Cloud's** scalable infrastructure. Reliable and high performance **cloud** services. Start your free trial today. Built for developers. Competitive pricing. Use our worldwide network. No-ops apps. $300 free trial. Access from anywhere.

Create free account
Learn & build with our Free Tier & $300 free credit!

Google Cloud Storage
Store & manage objects across four storage classes with a unified API.

Why Google Cloud Platform
Run on Google's infrastructure. Google-grade security & compliance.

Cloud Build
Continuously build, test and deploy.

https://cloud.google.com

Google Cloud: Cloud Computing Services

Meet your business challenges head on with **cloud computing** services from **Google**, including data management, hybrid & multi-**cloud**, and AI & ML.

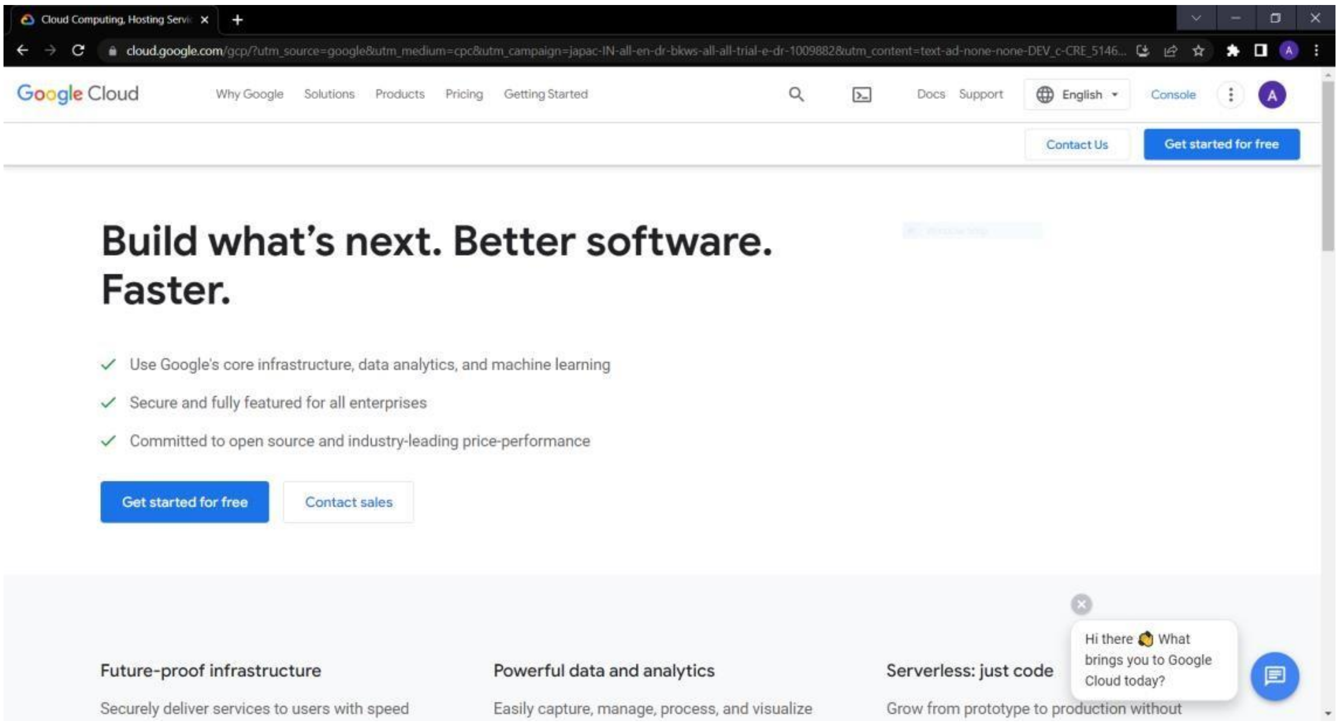
Google Cloud Platform
Computer application

Google Cloud Platform, offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, Google Drive, and YouTube. Wikipedia

Launched: April 7, 2008; 14 years ago
Written in: Java; C++; Python; Go; Ruby
Owner: Google

Account setup

Step2 : Goto Console



Google Cloud   Why Google   Solutions   Products   Pricing   Getting Started   Docs   Support   English   Console

Contact Us   Get started for free

Build what's next. Better software. Faster.

✓ Use Google's core infrastructure, data analytics, and machine learning
✓ Secure and fully featured for all enterprises
✓ Committed to open source and industry-leading price-performance

Get started for free   Contact sales

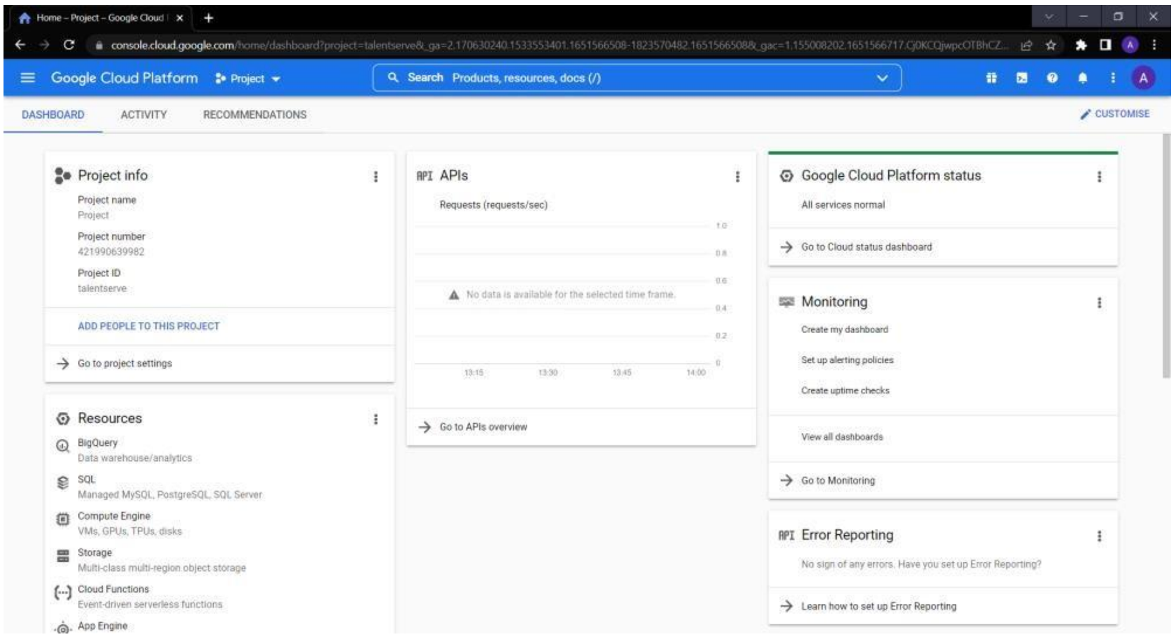
Future-proof infrastructure
Securely deliver services to users with speed

Powerful data and analytics
Easily capture, manage, process, and visualize

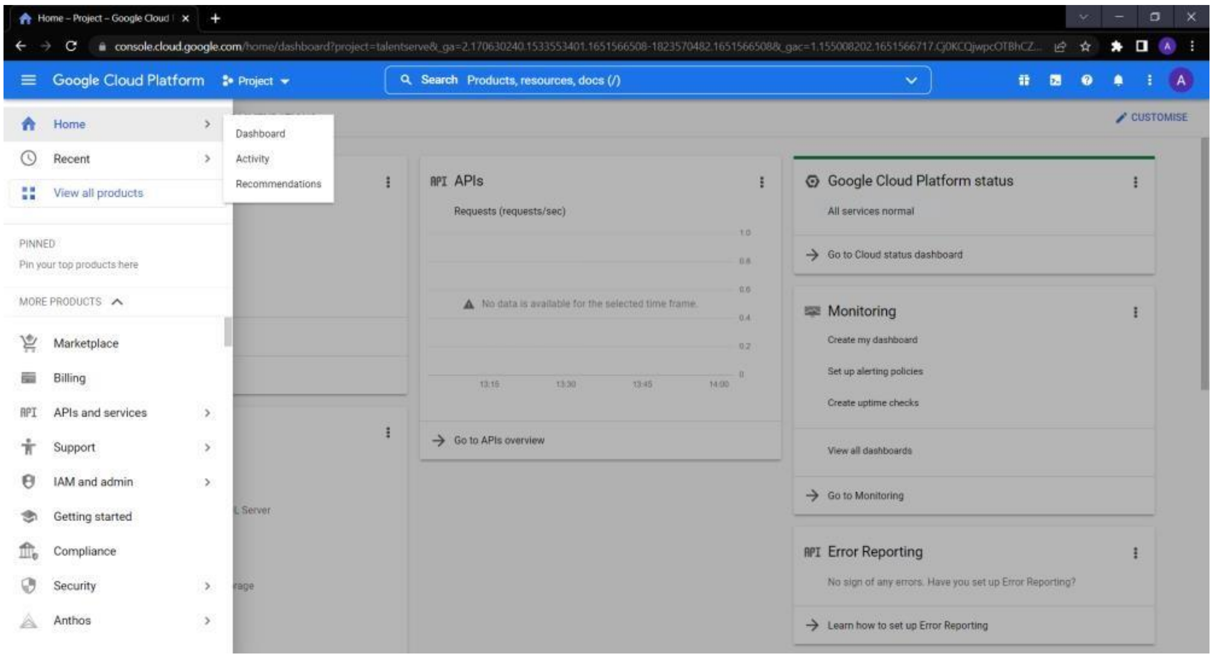
Serverless: just code
Grow from prototype to production without

Hi there What brings you to Google Cloud today?

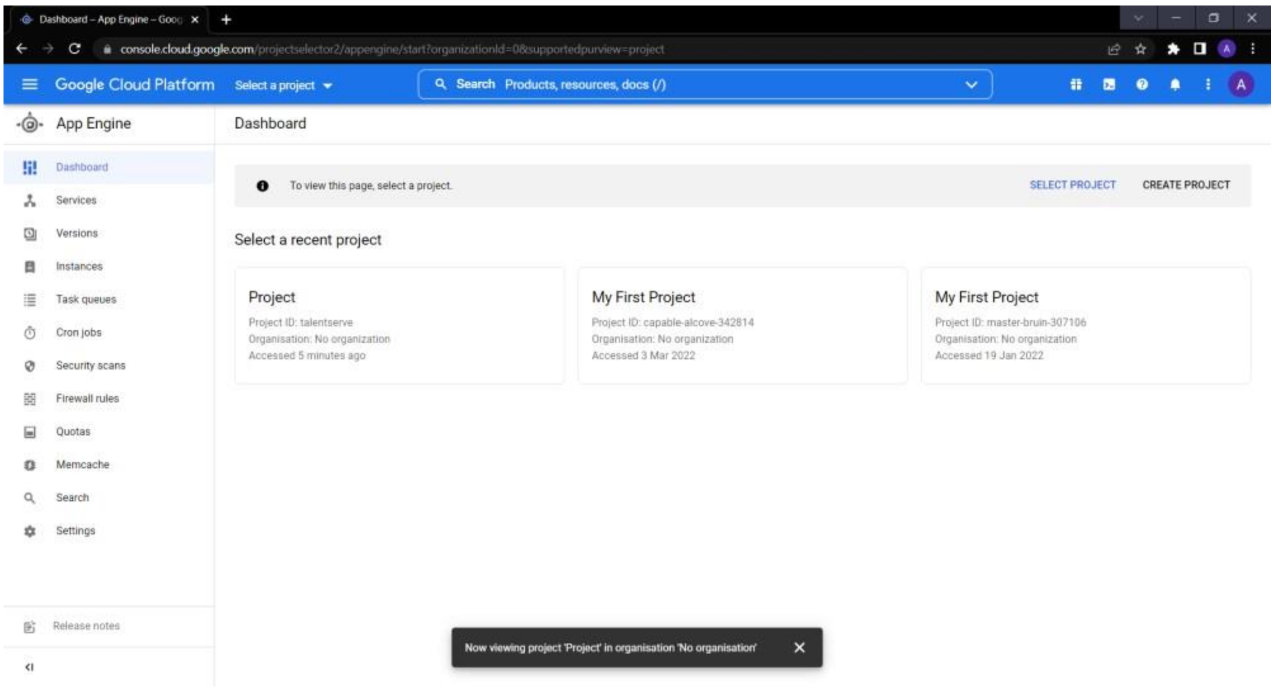
Step 3 : Google Cloud Platform is shown


Home – Project – Google Cloud
console.cloud.google.com/home/dashboard?project=talentserve&_ga=2.170630240.1533553401.1651566508-1823570482.1651566508&_gac=1.155008202.1651566717.Cj0KCQjwpcOTBhCZ...
Google Cloud Platform
Project
Search Products, resources, docs (/)
DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMISE
Project info
Project name
Project
Project number
421990639982
Project ID
talentserve
ADD PEOPLE TO THIS PROJECT
Go to project settings
Resources
BigQuery
Data warehouse/analytics
SQL
Managed MySQL, PostgreSQL, SQL Server
Compute Engine
VMs, GPUs, TPUs, disks
Storage
Multi-class multi-region object storage
Cloud Functions
Event-driven serverless functions
App Engine
APIs
Requests (requests/sec)
No data is available for the selected time frame.
Go to APIs overview
Google Cloud Platform status
All services normal
Go to Cloud status dashboard
Monitoring
Create my dashboard
Set up alerting policies
Create uptime checks
View all dashboards
Go to Monitoring
Error Reporting
No sign of any errors. Have you set up Error Reporting?
Learn how to set up Error Reporting

Step 4 : Click Dashboard in the Google Cloud Plaform


Home – Project – Google Cloud
console.cloud.google.com/home/dashboard?project=talentserve&_ga=2.170630240.1533553401.1651566508-1823570482.1651566508&_gac=1.155008202.1651566717.Cj0KCQjwpcOTBhCZ...
Google Cloud Platform
Project
Search Products, resources, docs (/)
Home
Recent
View all products
Dashboard
Activity
Recommendations
PINNED
Pin your top products here
MORE PRODUCTS
Marketplace
Billing
APIs and services
Support
IAM and admin
Getting started
Compliance
Security
Anthos
CUSTOMISE
APIs
Requests (requests/sec)
No data is available for the selected time frame.
Go to APIs overview
Google Cloud Platform status
All services normal
Go to Cloud status dashboard
Monitoring
Create my dashboard
Set up alerting policies
Create uptime checks
View all dashboards
Go to Monitoring
Error Reporting
No sign of any errors. Have you set up Error Reporting?
Learn how to set up Error Reporting

Step 5 : Dashboard in the Google Cloud Plaform



Dashboard – App Engine – Goog

console.cloud.google.com/projectselector2/appengine/start?organizationId=0&supportedpurview=project

Google Cloud Platform   Select a project   Search Products, resources, docs (/)

App Engine   Dashboard

Dashboard
Services
Versions
Instances
Task queues
Cron jobs
Security scans
Firewall rules
Quotas
Memcache
Search
Settings
Release notes

To view this page, select a project.   SELECT PROJECT   CREATE PROJECT

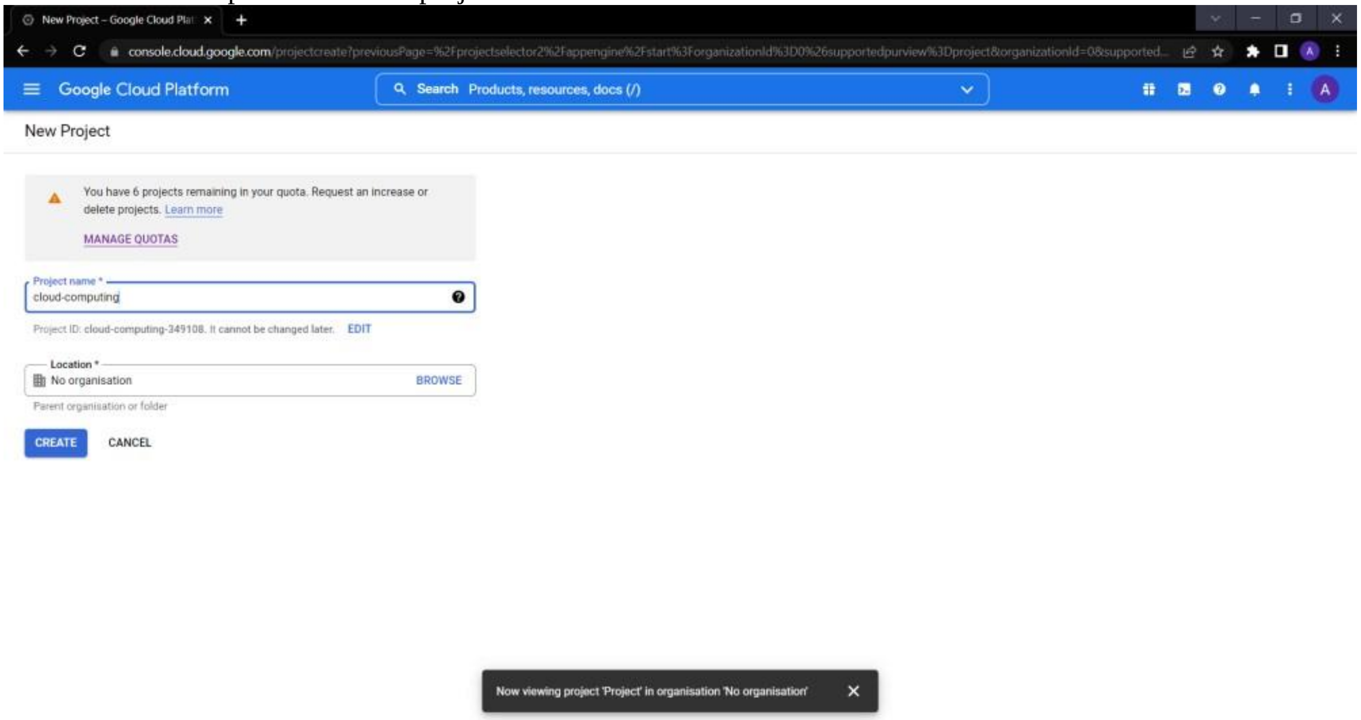
Select a recent project

Project	My First Project	My First Project
Project ID: talentserve	Project ID: capable-alcove-342814	Project ID: master-bruin-307106
Organisation: No organization	Organisation: No organization	Organisation: No organization
Accessed 5 minutes ago	Accessed 3 Mar 2022	Accessed 19 Jan 2022

Now viewing project 'Project' in organisation 'No organisation'

Step 6 : Click New Project and give unique Project Name.

Example : kcet-cloud-project



New Project – Google Cloud Plat

console.cloud.google.com/projectcreate?previousPage=%2Fprojectselector2%2Fappengine%2Fstart%3ForganizationId%3D0%26supportedpurview%3Dproject&organizationId=0&supported...

Google Cloud Platform   Search Products, resources, docs (/)

New Project

You have 6 projects remaining in your quota. Request an increase or delete projects. Learn more

MANAGE QUOTAS

Project name *
cloud-computing

Project ID: cloud-computing-349108. It cannot be changed later. EDIT

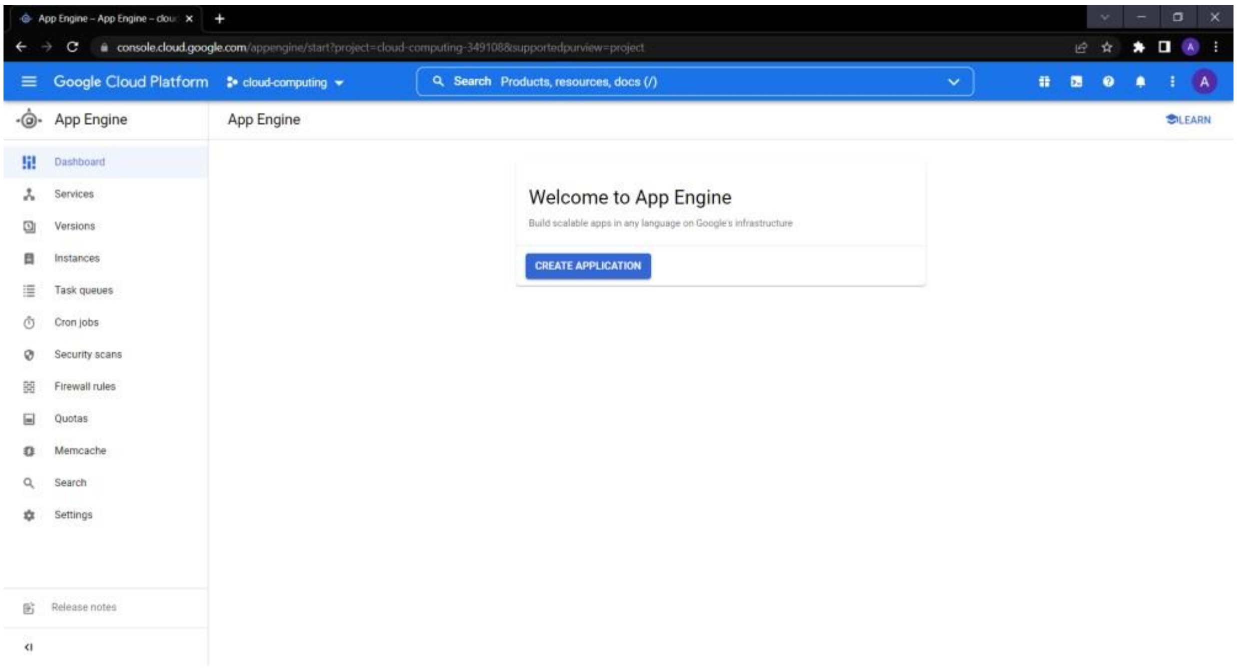
Location *
No organisation   BROWSE

Parent organisation or folder

CREATE   CANCEL

Now viewing project 'Project' in organisation 'No organisation'

Step 7 : Google App Engine is initated

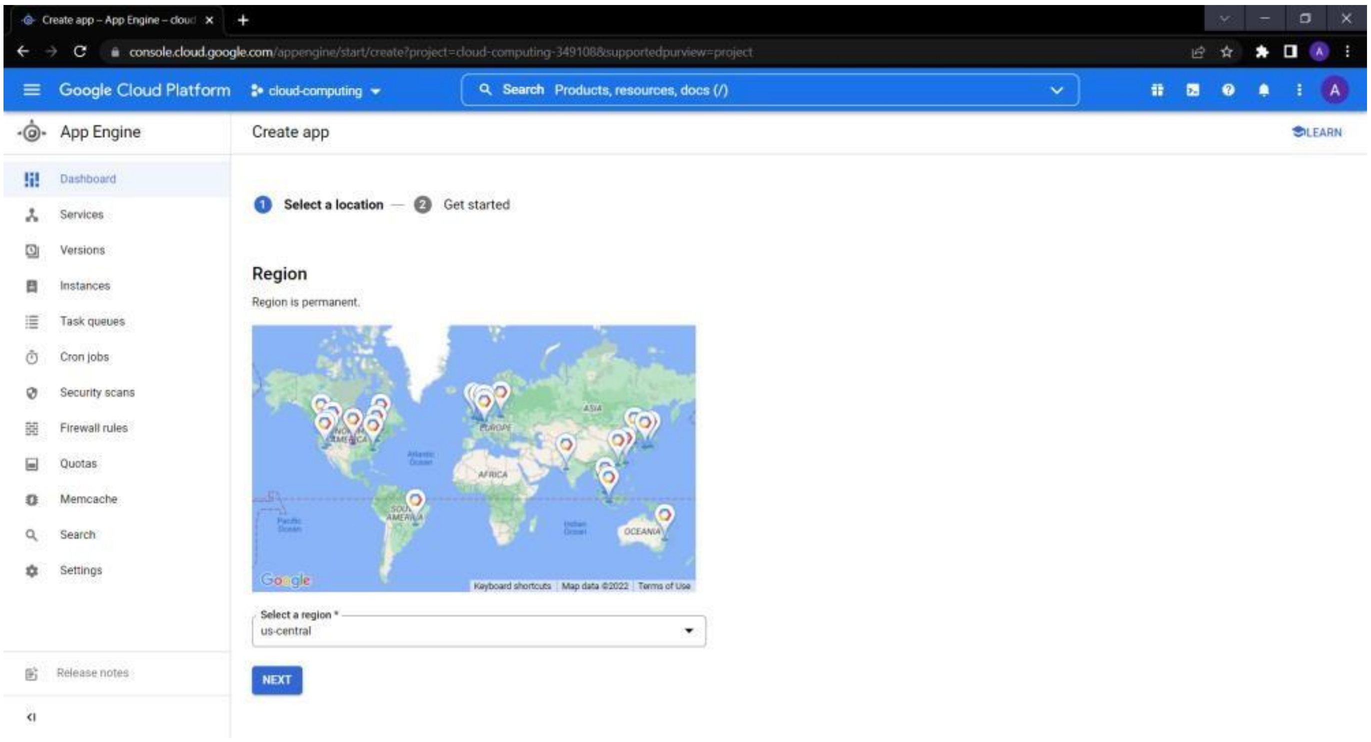


Welcome to App Engine

Build scalable apps in any language on Google's infrastructure

CREATE APPLICATION

Step 8 : Click create Application

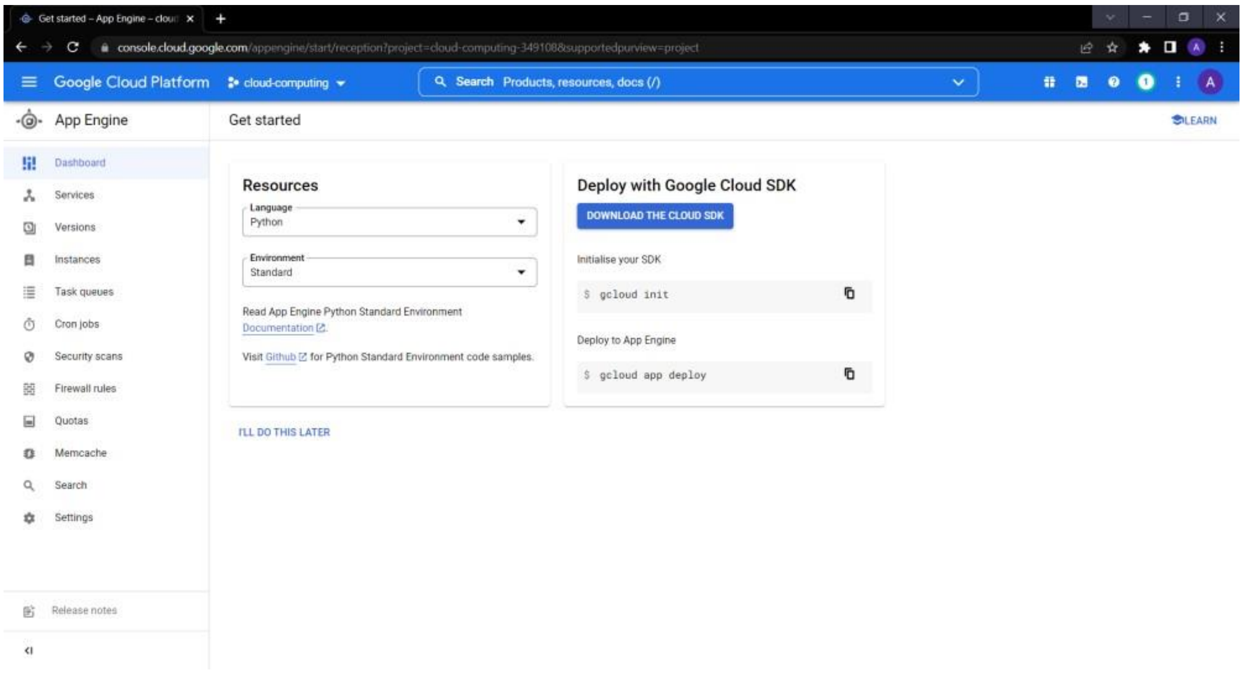


Region

Region is permanent.

Select a region * us-central

Step 9 : Create app and Select Language Python



Resources

Language: Python

Environment: Standard

Read App Engine Python Standard Environment Documentation.

Visit Github for Python Standard Environment code samples.

Deploy with Google Cloud SDK

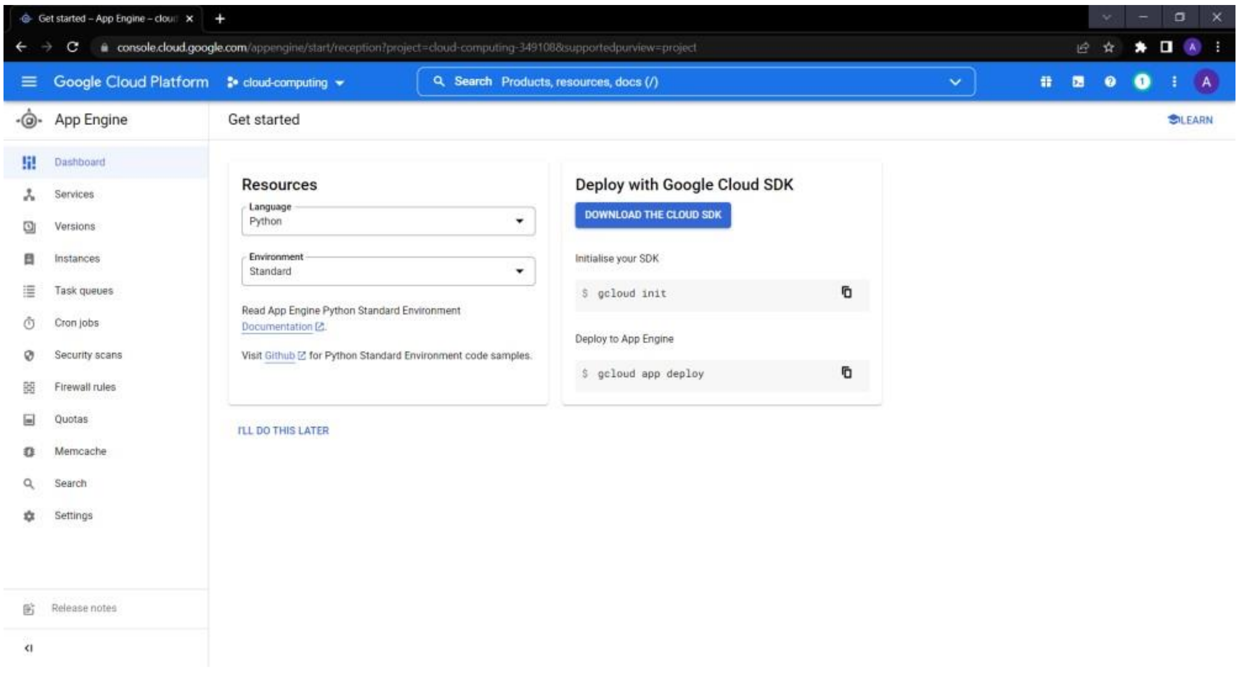
Initialise your SDK

```
$ gcloud init
```

Deploy to App Engine

```
$ gcloud app deploy
```

Step 10 : Python app is created in Google App Engine



Resources

Language: Python

Environment: Standard

Read App Engine Python Standard Environment Documentation.

Visit Github for Python Standard Environment code samples.

Deploy with Google Cloud SDK

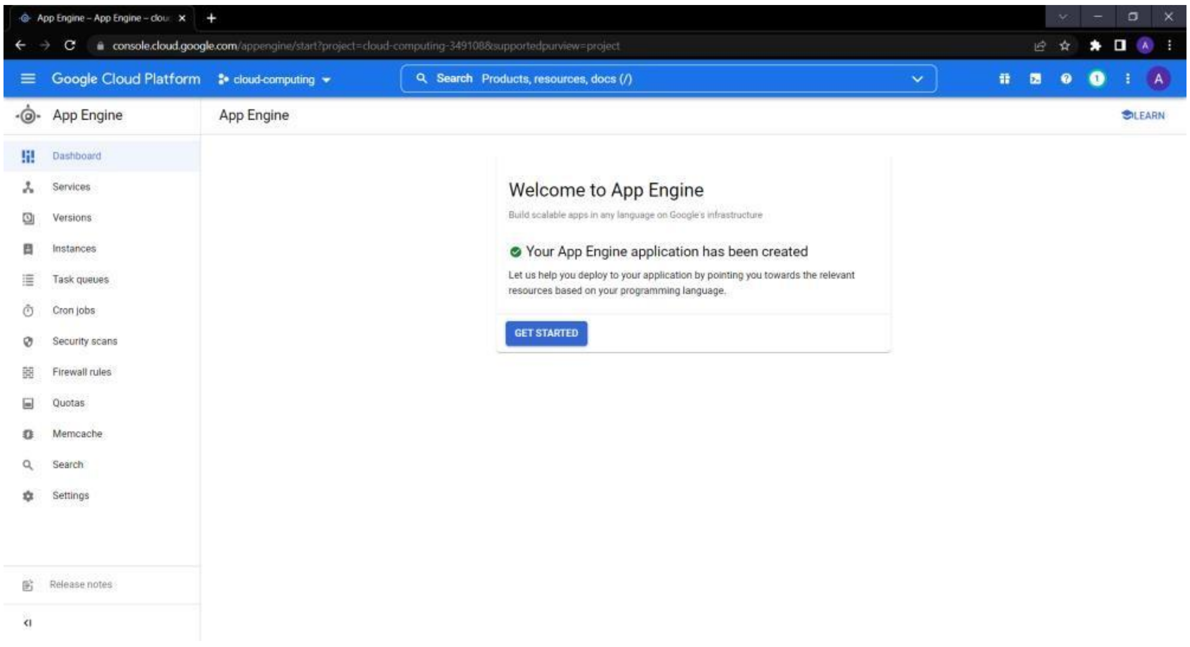
Initialise your SDK

```
$ gcloud init
```

Deploy to App Engine

```
$ gcloud app deploy
```

Step 11 : Python app Engine application is created



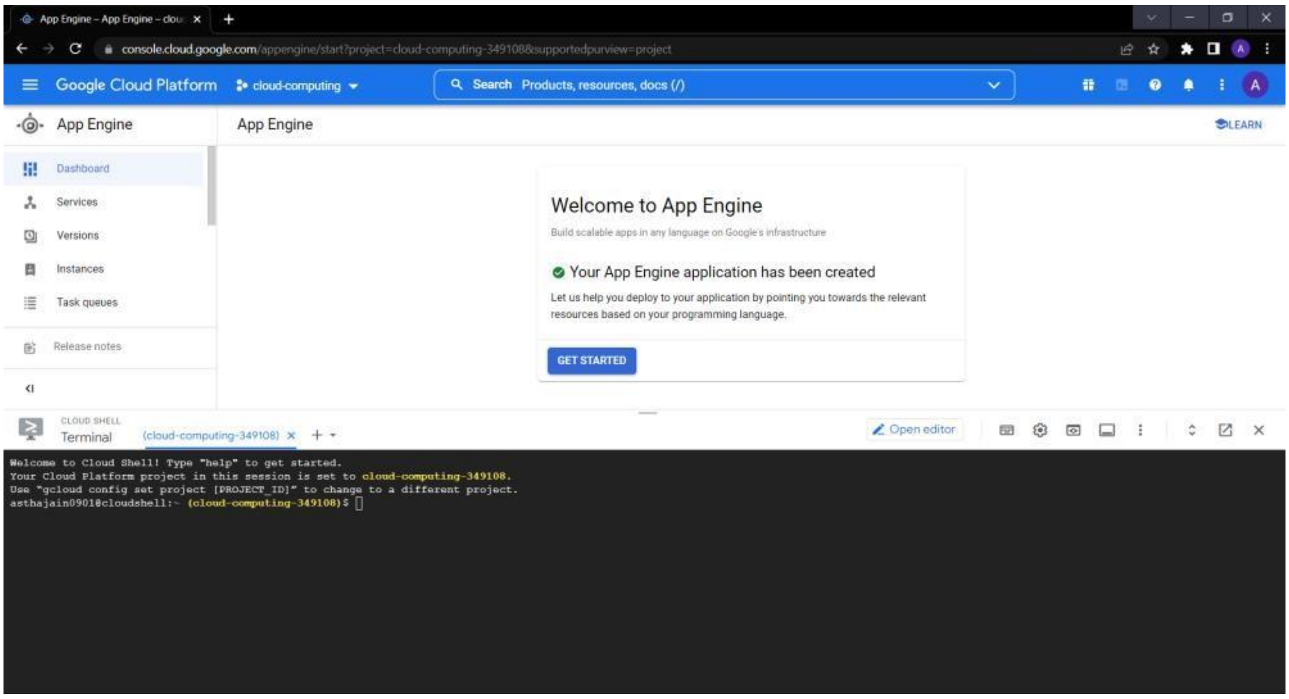
Welcome to App Engine

Build scalable apps in any language on Google's infrastructure

Your App Engine application has been created

Let us help you deploy to your application by pointing you towards the relevant resources based on your programming language.

Step 12 : Click Cloud Shell in the Kathir-Cloud-Project



Welcome to App Engine

Build scalable apps in any language on Google's infrastructure

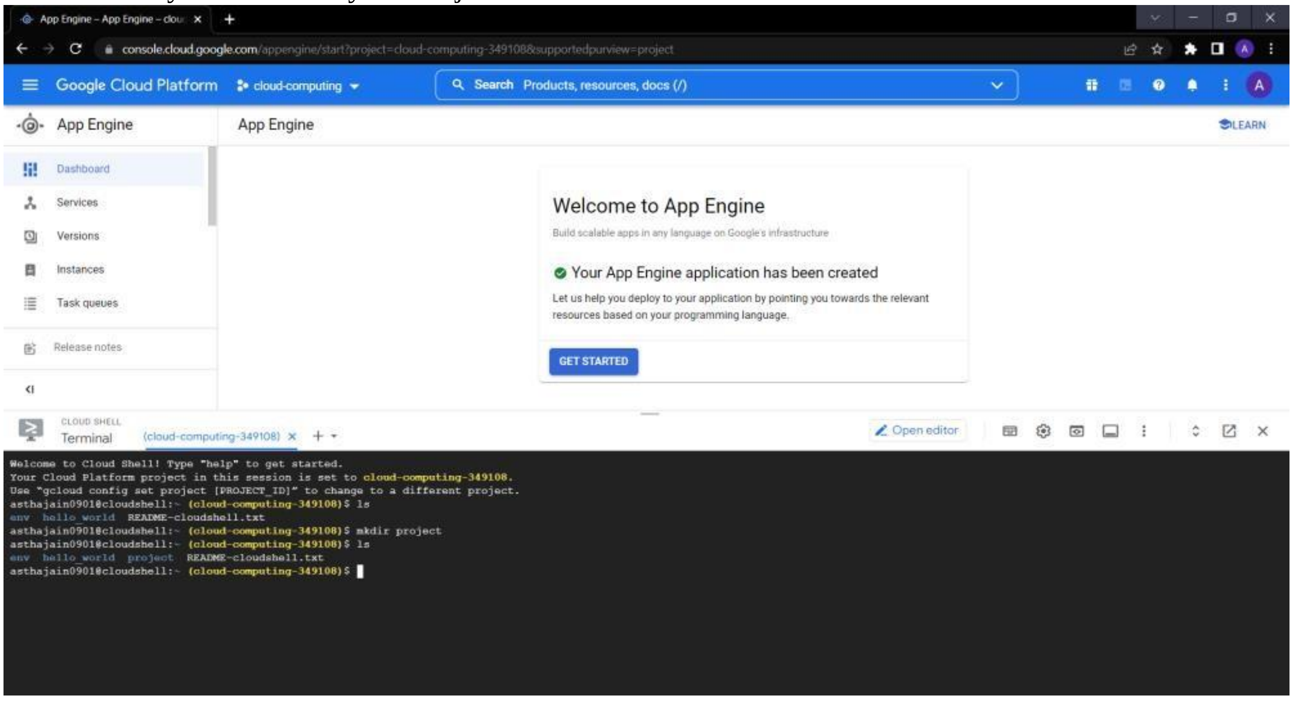
Your App Engine application has been created

Let us help you deploy to your application by pointing you towards the relevant resources based on your programming language.

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to cloud-computing-349108.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
asthajain0901@cloudshell:~ (cloud-computing-349108)$ 
```

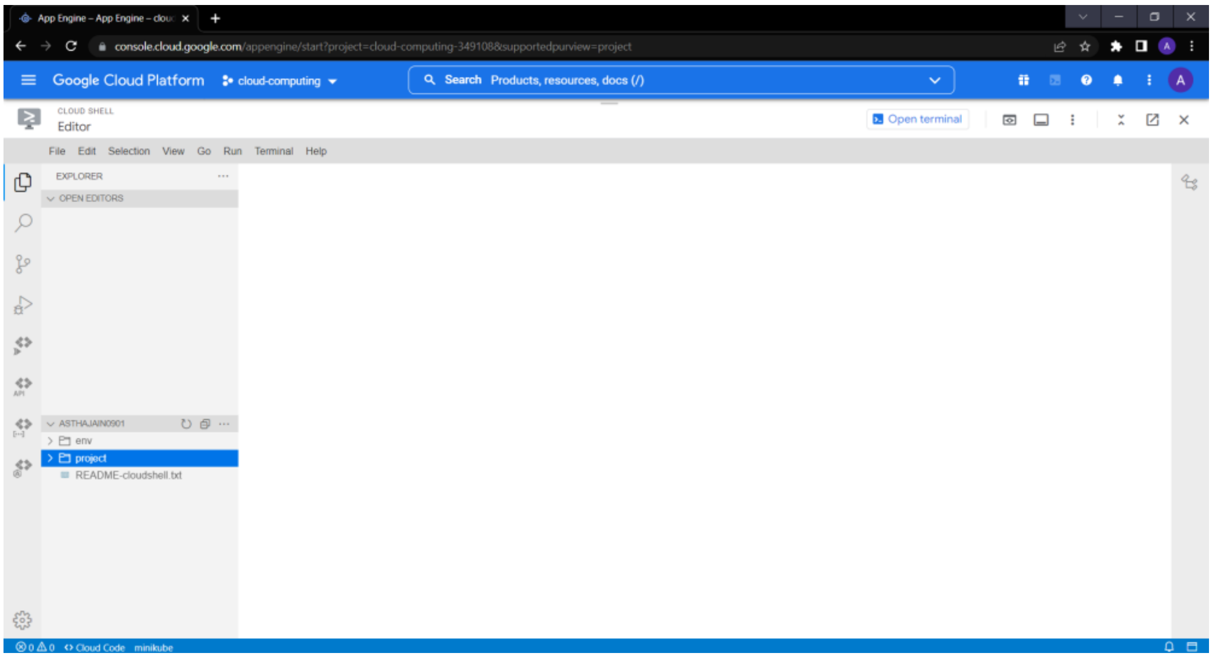
Step 13 : Create a Directory PythonProject using mkdir command

Syntax : mkdir PythonProject

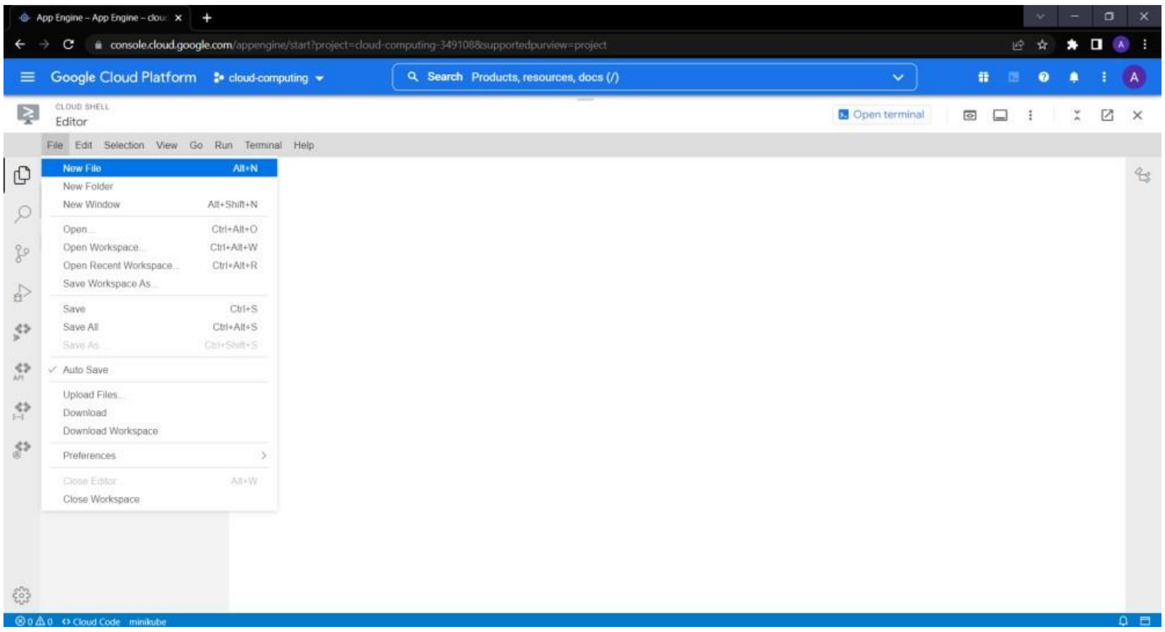

App Engine – App Engine – clou
console.cloud.google.com/appengine/start?project=cloud-computing-349108&supportedpurview=project
Google Cloud Platform
cloud-computing
Search Products, resources, docs (/)
App Engine
Dashboard
Services
Versions
Instances
Task queues
Release notes
LEARN
Welcome to App Engine
Build scalable apps in any language on Google's infrastructure
Your App Engine application has been created
Let us help you deploy to your application by pointing you towards the relevant resources based on your programming language.
GET STARTED
CLOUD SHELL
Terminal
(cloud-computing-349108)
Open editor

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to cloud-computing-349108.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
asthajain0901@cloudshell:~ (cloud-computing-349108)$ ls
env  hello_world  README-cloudshell.txt
asthajain0901@cloudshell:~ (cloud-computing-349108)$ mkdir project
asthajain0901@cloudshell:~ (cloud-computing-349108)$ ls
env  hello_world  project  README-cloudshell.txt
asthajain0901@cloudshell:~ (cloud-computing-349108)$
```

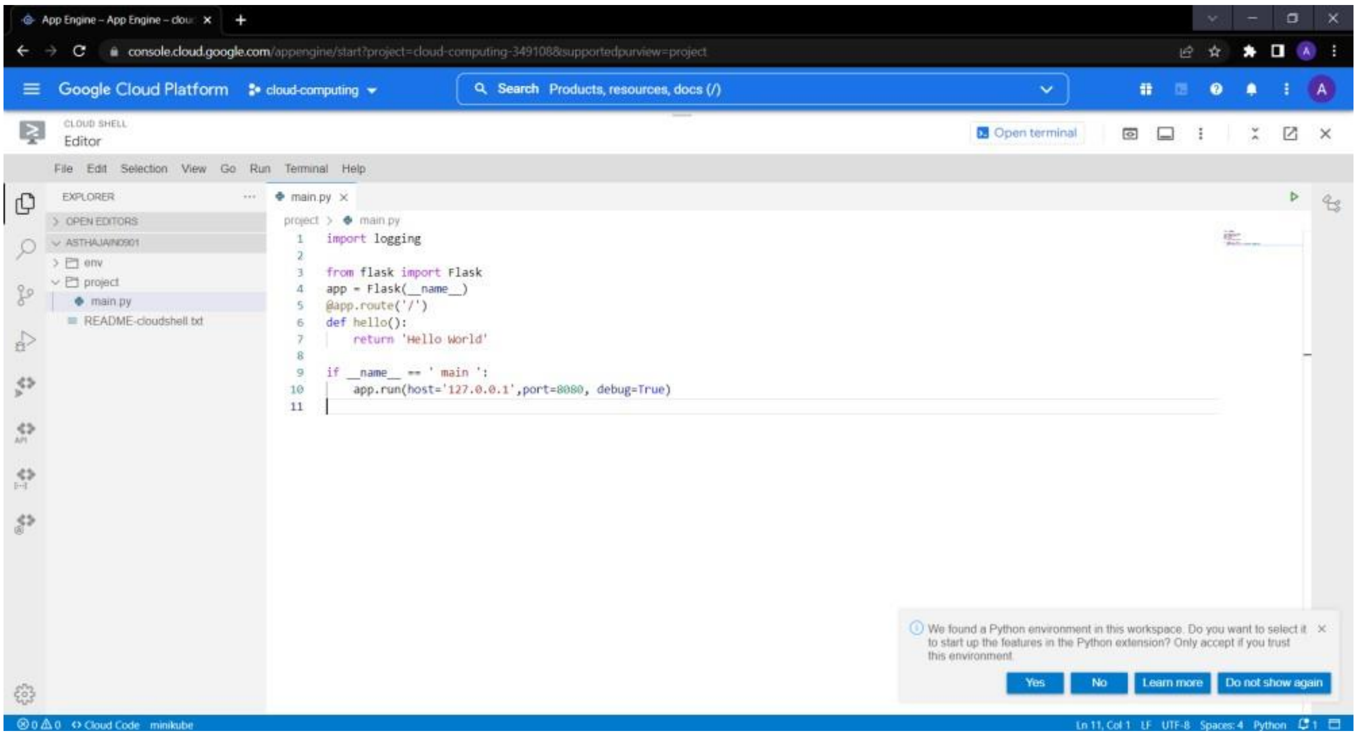
Step 14 : Click Editor to create Python application


App Engine – App Engine – clou
console.cloud.google.com/appengine/start?project=cloud-computing-349108&supportedpurview=project
Google Cloud Platform
cloud-computing
Search Products, resources, docs (/)
CLOUD SHELL
Editor
Open terminal
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
ASTHAJAIN0901
env
project
README-cloudshell.txt
Cloud Code minikube

Step 15 : Click New File in the PythonProject Folder (Python file)



Step 16 : Create main.py file


App Engine – App Engine – clou
console.cloud.google.com/appengine/start?project=cloud-computing-349108&supportedpurview=project
Google Cloud Platform
cloud-computing
Search Products, resources, docs (/)
CLOUD SHELL
Editor
Open terminal
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
ASTHAJAIN0901
env
project
main.py
README-cloudshell.txt
main.py
project > main.py
Yes
No
Learn more
Do not show again
Cloud Code minikube
Ln 11, Col 1 LF UTF-8 Spaces: 4 Python

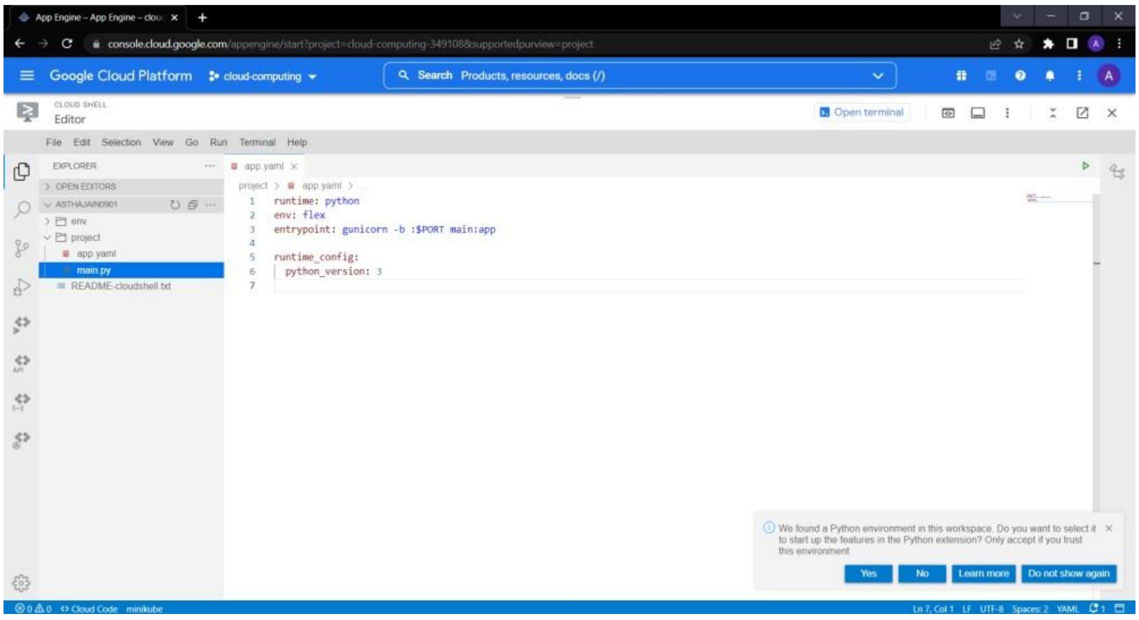
```
import logging

from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello():
    return 'Hello World'

if __name__ == ' main ':
    app.run(host='127.0.0.1',port=8080, debug=True)

```

We found a Python environment in this workspace. Do you want to select it to start up the features in the Python extension? Only accept if you trust this environment.

main.py file

```
import &nbsplogging

from &nbspflask import &nbspFlask

app =&nbspFlask( name ) _

@app.route('/')
def hello():
    return 'Hello&nbspWorld'

if___name___== '_main_':
   &nbspapp.run(host='127.0.0.1',port=8080, debug=True)
```

Step 17 : Create app.yaml file



App Engine – App Engine – clou
console.cloud.google.com/appengine/start?project=cloud-computing-349108&supportedpurview=project
Google Cloud Platform
cloud-computing
Search Products, resources, docs (/)
CLOUD SHELL
Editor
Open terminal
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
ASTHAJAIN0901
env
project
app.yaml
main.py
README-cloudshell.txt
app.yaml
project > app.yaml >
Yes
No
Learn more
Do not show again
Cloud Code minikube
Ln 7, Col 1 LF UTF-8 Spaces: 2 YAML

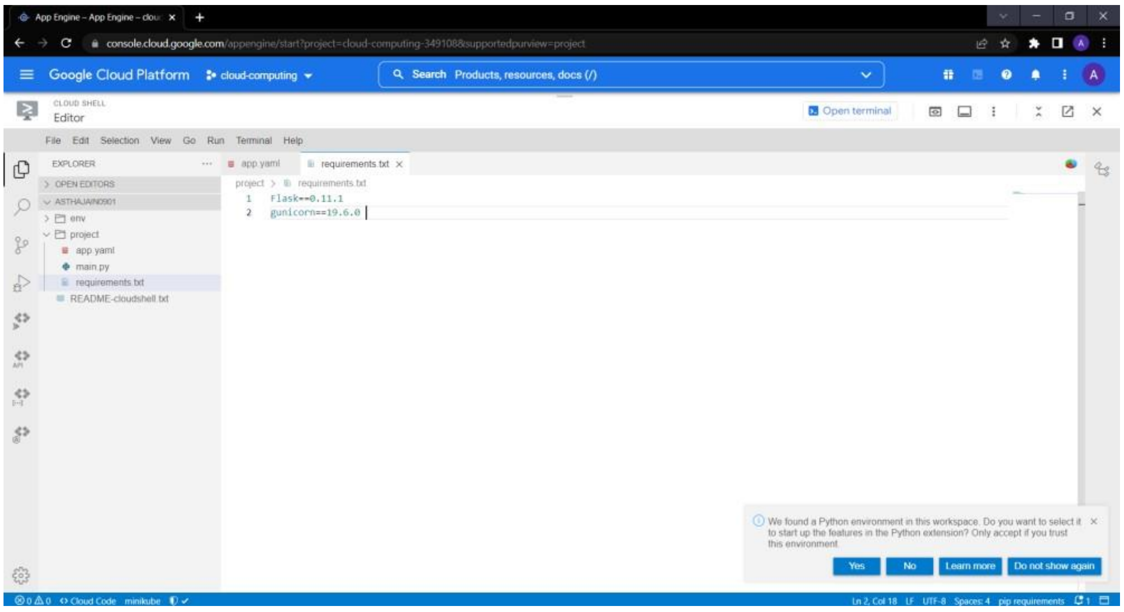
```
runtime: python
env: flex
entrypoint: gunicorn -b :$PORT main:app

runtime_config:
  python_version: 3
```

We found a Python environment in this workspace. Do you want to select it to start up the features in the Python extension? Only accept if you trust this environment.

app.yaml

```
runtime: python
env: flex
entrypoint: gunicorn -b :$PORT&nbspmain:app

runtime_config:
    python_version: 3
```

Step 18 : Create requirements.txt file



```
Flask==0.11.1
gunicorn==19.6.0
```

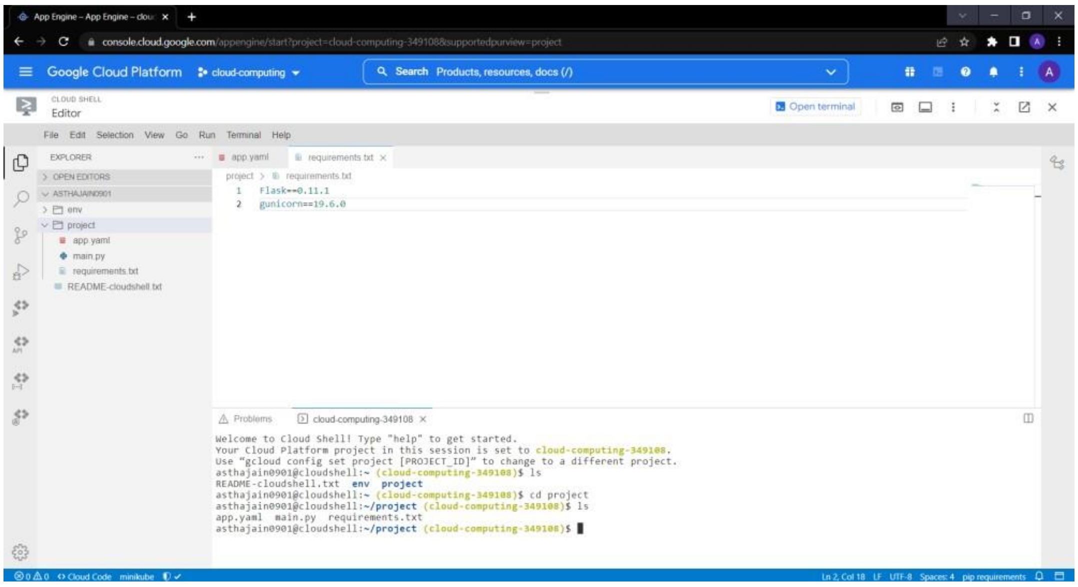
We found a Python environment in this workspace. Do you want to select it to start up the features in the Python extension? Only accept if you trust this environment.

requirements.txt

Flask==0.11.1

gunicorn==19.6.0

Step 19 : Move to Cloud Shell Environment to run the application

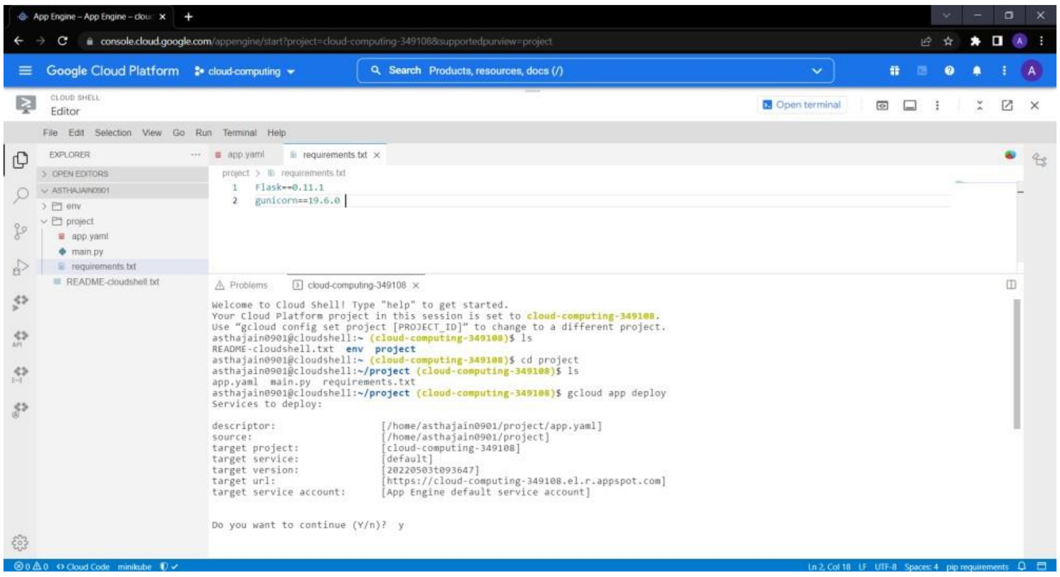


```
Flask==0.11.1
gunicorn==19.6.0
```

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to cloud-computing-349108.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
asthajain0901@cloudshell:~ (cloud-computing-349108)$ ls
README-cloudshell.txt  env  project
asthajain0901@cloudshell:~ (cloud-computing-349108)$ cd project
asthajain0901@cloudshell:~/project (cloud-computing-349108)$ ls
app.yaml  main.py  requirements.txt
asthajain0901@cloudshell:~/project (cloud-computing-349108)$ 
```

Step 20 : Move to Cloud Shell Environment to run the application

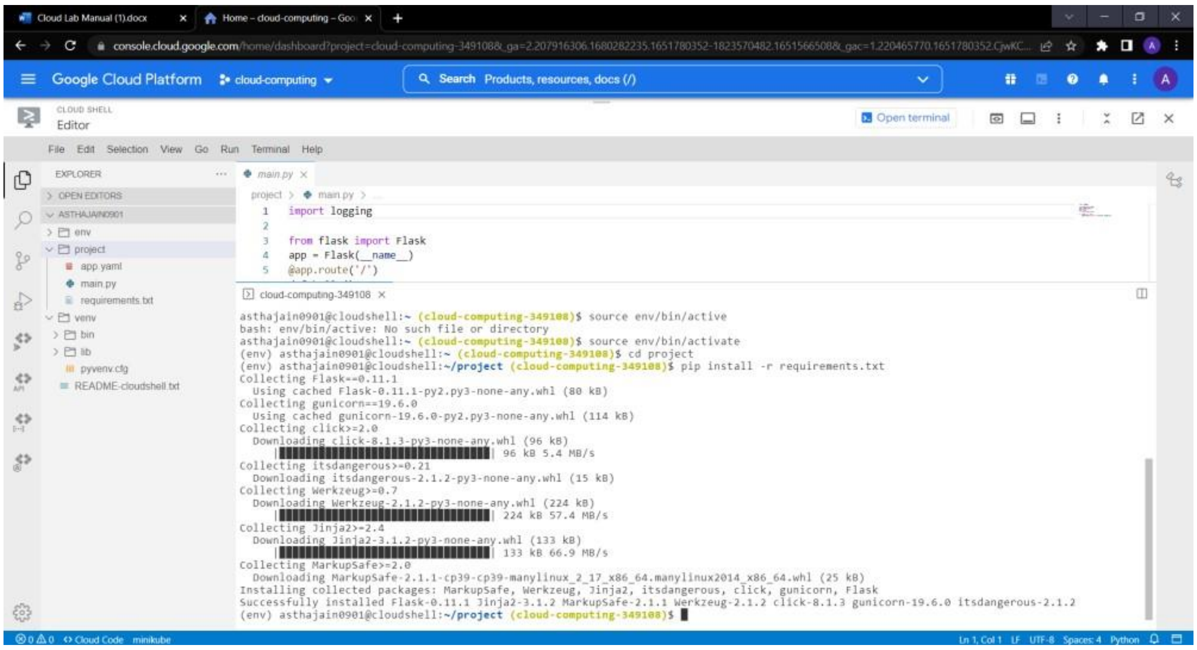
Syntax : gcloud app deploy



```
Flask==0.11.1
gunicorn==19.6.0
```

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to cloud-computing-349108.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
asthajain0901@cloudshell:~ (cloud-computing-349108)$ ls
README-cloudshell.txt  env  project
asthajain0901@cloudshell:~ (cloud-computing-349108)$ cd project
asthajain0901@cloudshell:~/project (cloud-computing-349108)$ ls
app.yaml  main.py  requirements.txt
asthajain0901@cloudshell:~/project (cloud-computing-349108)$ gcloud app deploy
Services to deploy:

descriptor:                  [/home/asthajain0901/project/app.yaml]
source:                      [/home/asthajain0901/project]
target project:              [cloud-computing-349108]
target service:              [default]
target version:              [20220503t093647]
target url:                  [https://cloud-computing-349108.el.r.appspot.com]
target service account:      [App Engine default service account]


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

1Continue the application. It enables service on the given project

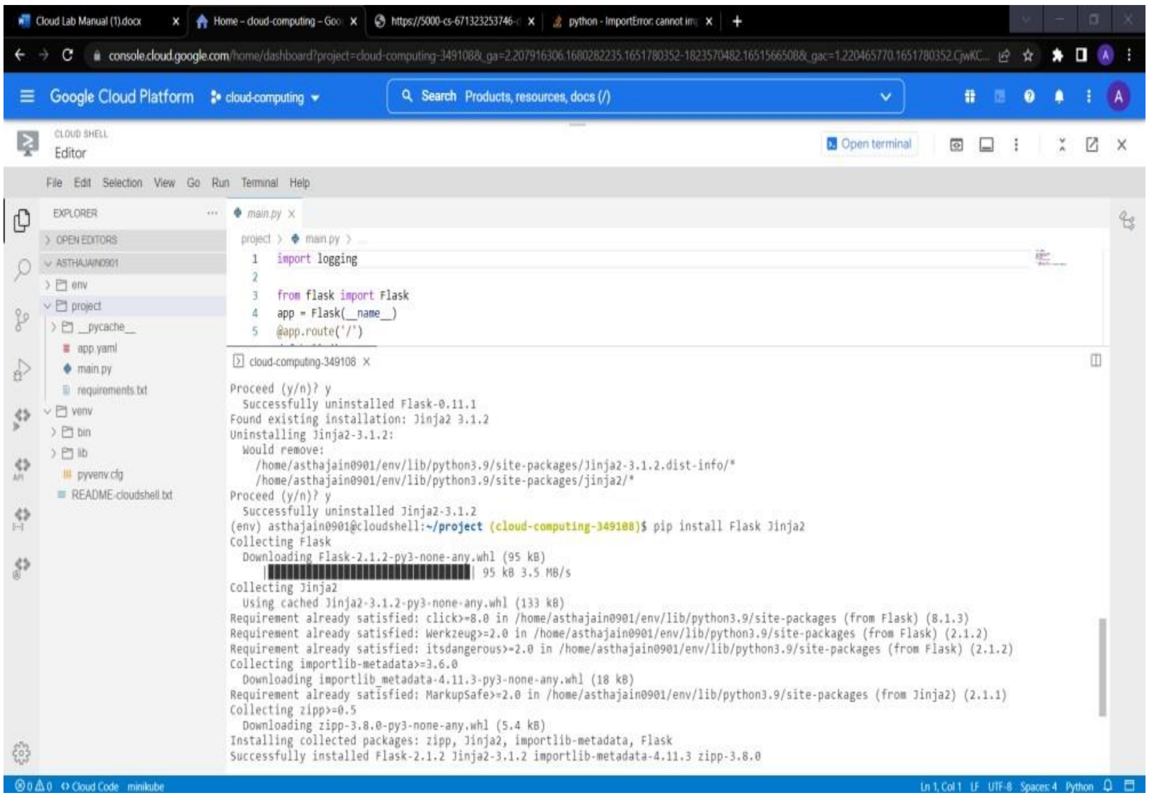


```
import logging

from flask import Flask
app = Flask(__name__)
@app.route('/')
```

```
asthajain0901@cloudshell:~ (cloud-computing-349108)$ source env/bin/active
bash: env/bin/active: No such file or directory
asthajain0901@cloudshell:~ (cloud-computing-349108)$ source env/bin/activate
(env) asthajain0901@cloudshell:~ (cloud-computing-349108)$ cd project
(env) asthajain0901@cloudshell:~/project (cloud-computing-349108)$ pip install -r requirements.txt
Collecting Flask==0.11.1
  Using cached Flask-0.11.1-py2.py3-none-any.whl (80 kB)
Collecting gunicorn==19.6.0
  Using cached gunicorn-19.6.0-py2.py3-none-any.whl (114 kB)
Collecting click>=2.0
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
     |████████████████████████████████| 96 kB 5.4 MB/s
Collecting itsdangerous>=0.21
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting Werkzeug>=0.7
  Downloading Werkzeug-2.1.2-py3-none-any.whl (224 kB)
     |████████████████████████████████| 224 kB 57.4 MB/s
Collecting Jinja2>=2.4
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
     |████████████████████████████████| 133 kB 66.9 MB/s
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Installing collected packages: MarkupSafe, Werkzeug, Jinja2, itsdangerous, click, gunicorn, Flask
Successfully installed Flask-0.11.1 Jinja2-3.1.2 MarkupSafe-2.1.1 Werkzeug-2.1.2 click-8.1.3 gunicorn-19.6.0 itsdangerous-2.1.2
(env) asthajain0901@cloudshell:~/project (cloud-computing-349108)$
```

It started building the object and fetching the storage object for the created application

It is updating the service



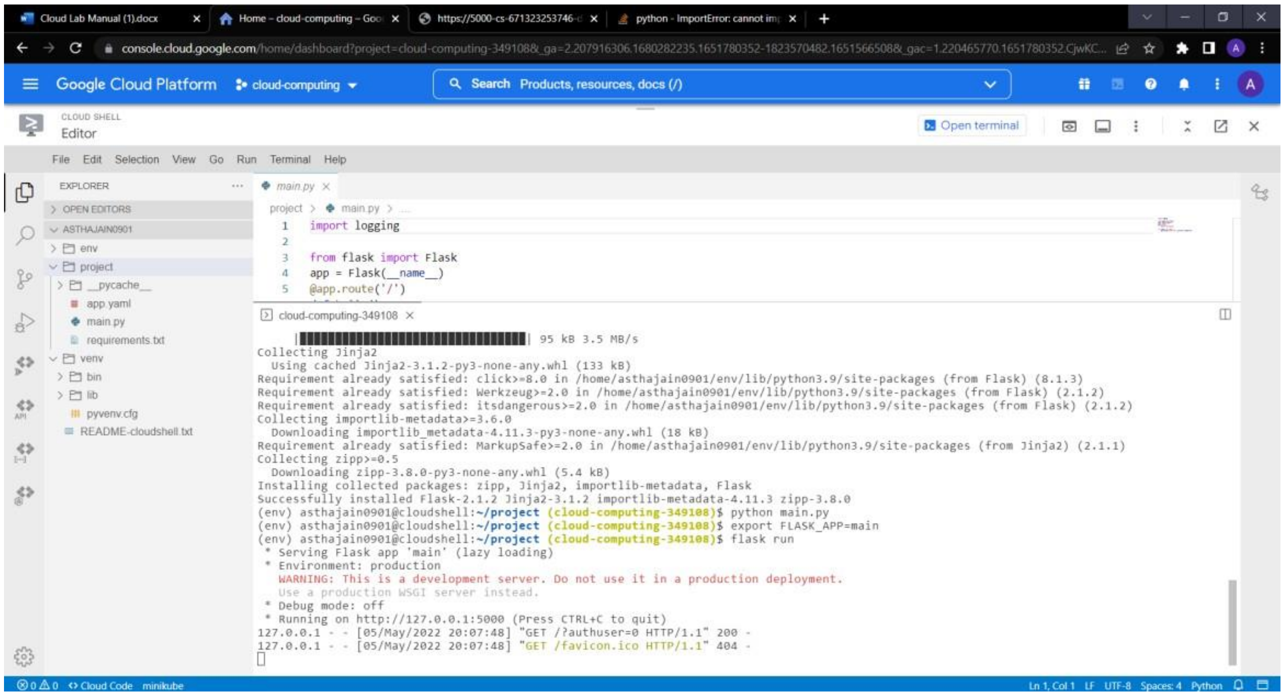
```python
import logging

from flask import Flask
app = Flask(__name__)
@app.route('/')
```

```
Proceed (y/n)? y
  Successfully uninstalled Flask-0.11.1
Found existing installation: Jinja2 3.1.2
Uninstalling Jinja2-3.1.2:
  Would remove:
    /home/asthajain0901/env/lib/python3.9/site-packages/Jinja2-3.1.2.dist-info/*
    /home/asthajain0901/env/lib/python3.9/site-packages/jinja2/*
Proceed (y/n)? y
  Successfully uninstalled Jinja2-3.1.2
(env) asthajain0901@cloudshell:~/project (cloud-computing-349108)$ pip install Flask Jinja2
Collecting Flask
  Downloading Flask-2.1.2-py3-none-any.whl (95 kB)
     |████████████████████████████████| 95 kB 3.5 MB/s
Collecting Jinja2
  Using cached Jinja2-3.1.2-py3-none-any.whl (133 kB)
Requirement already satisfied: click>=8.0 in /home/asthajain0901/env/lib/python3.9/site-packages (from Flask) (8.1.3)
Requirement already satisfied: Werkzeug>=2.0 in /home/asthajain0901/env/lib/python3.9/site-packages (from Flask) (2.1.2)
Requirement already satisfied: itsdangerous>=2.0 in /home/asthajain0901/env/lib/python3.9/site-packages (from Flask) (2.1.2)
Collecting importlib-metadata>=3.6.0
  Downloading importlib_metadata-4.11.3-py3-none-any.whl (18 kB)
Requirement already satisfied: MarkupSafe>=2.0 in /home/asthajain0901/env/lib/python3.9/site-packages (from Jinja2) (2.1.1)
Collecting zipp>=0.5
  Downloading zipp-3.8.0-py3-none-any.whl (5.4 kB)
Installing collected packages: zipp, Jinja2, importlib-metadata, Flask
Successfully installed Flask-2.1.2 Jinja2-3.1.2 importlib-metadata-4.11.3 zipp-3.8.0
```

The application is successfully deployed and URL is

https://expanded-curve-289413.uc.r.appspot.com

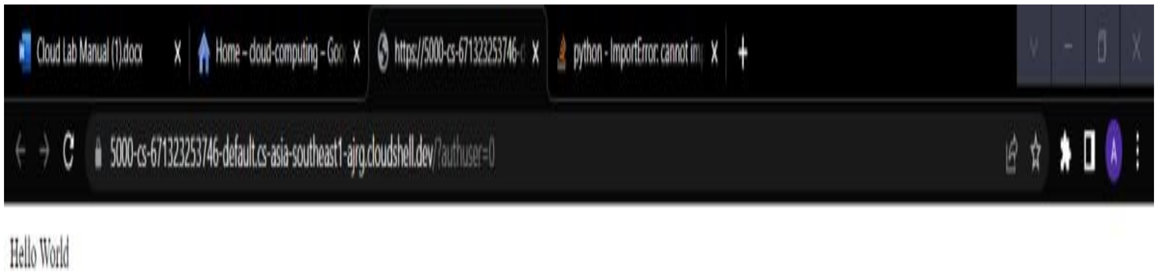
Step 21 : Run your program in the browser



```
import logging

from flask import Flask
app = Flask(__name__)
@app.route('/')
```

```
    |████████████████████████████████| 95 kB 3.5 MB/s
Collecting Jinja2
  Using cached Jinja2-3.1.2-py3-none-any.whl (133 kB)
Requirement already satisfied: click>=8.0 in /home/asthajain0901/env/lib/python3.9/site-packages (from Flask) (8.1.3)
Requirement already satisfied: Werkzeug>=2.0 in /home/asthajain0901/env/lib/python3.9/site-packages (from Flask) (2.1.2)
Requirement already satisfied: itsdangerous>=2.0 in /home/asthajain0901/env/lib/python3.9/site-packages (from Flask) (2.1.2)
Collecting importlib-metadata>=3.6.0
  Downloading importlib_metadata-4.11.3-py3-none-any.whl (18 kB)
Requirement already satisfied: MarkupSafe>=2.0 in /home/asthajain0901/env/lib/python3.9/site-packages (from Jinja2) (2.1.1)
Collecting zipp>=0.5
  Downloading zipp-3.8.0-py3-none-any.whl (5.4 kB)
Installing collected packages: zipp, Jinja2, importlib-metadata, Flask
Successfully installed Flask-2.1.2 Jinja2-3.1.2 importlib-metadata-4.11.3 zipp-3.8.0
(env) asthajain0901@cloudshell:~/project (cloud-computing-349108)$ python main.py
(env) asthajain0901@cloudshell:~/project (cloud-computing-349108)$ export FLASK_APP=main
(env) asthajain0901@cloudshell:~/project (cloud-computing-349108)$ flask run
 * Serving Flask app 'main' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
127.0.0.1 - - [05/May/2022 20:07:48] "GET /?authuser=0 HTTP/1.1" 200 -
127.0.0.1 - - [05/May/2022 20:07:48] "GET /favicon.ico HTTP/1.1" 404 -
```

Step 22 : Hello World Program is sucessfully run in the browser



```
5000-cs-671323253746-default.cs-asia-southeast1-ajrg.cloudshell.dev/?authuser=0
```

Hello World

Screen Shot of Output Page:

Conclusion:

Experiment No. 3

Title: Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim

Software Required: JDK, JRE, Cloudsim 4.0, Eclipse

Theory :

CloudSim comprises huge volume of libraries for simulating the cloud based computing systems. It offers necessary classes for creating users, applications, virtual machines, data centers, computational resources and policies for controlling the different parts of the system as provisioning and scheduling.

The challenging issues in cloud computing is the frequent performance assessment of resources, workloads and other provisioning polices in regardless of unpredictable user needs, system set-ups and network configurations.

Cloudsim plays a key player role to tackle this issue by modeling and simulating cloud environs through unique generic application provisioning approach. At present, it supports both inter-networked (federation) and single cloud(s) to analyze the behavior of deployed cloud components. In overall, Cloudsim is cloud simulation toolkit to satisfy the cloud user requirements. Here, we have included the benefits of using Cloudsim for testing the system performance.

Advantages of Cloudsim

- **Flexibility and Applicability** – In different cloud environments, it is simple to design and verify the overall applications / services performance.
  - For instance: MS Azure, Amazon EC2 and many more
- **Time Effectiveness** – Designing the cloud application with test platform will consume minimum time and effort

Let's talk over more detailed information of ***Cloudsim in Cloud Computing***. Basically, Cloudsim has specific functionalities to perform the better cloud simulation. Just to let you know our developer have listed some fundamental operations of cloudsim. **This toolkit allows to:**

- Do frequent test on developed system / application in the monitoring environment
- Deploy the application in real cloud after identifying the performance bottlenecks
- Design the application distribution approaches by conducting more number of test on various resources and workload setups

- Model and Simulate the following things,
    - Federated Cloud
    - Custom Based Virtualized Server Hosts
    - Broad-Size Cloud Data Centers And Network Structure
    - Message-Passing Paradigms
    - Application Containers
    - Energy-Aware Computational Resources
- Dynamically insert entities and resume / stop simulation
- Allocate both host and host resources to VM based on user-defined policies

Steps :

1.JDK

2.JRE

3.Cloudsim 4.0

4.Eclipse

Download all the above-mentioned software.

Save all five configuration files into one folder mentioned in description box.

Extract the cloud sim and save two files in any folder we need to copy it later.

Flow to follow

- 1. Create java project
- 2. Create package
- 3. Add source file inside the package
- 4. Add 2 jar files
- 5. Run the program and observe the output

Procedure:

- Install any of the IDE for running JAVA applications (eclipse recommended)
- Install JDK and JRE for the same
- Add the jdk\bin path to the environment variables Open environment variables window, add the following to the path variable
- Do include your bin path wherever you have installed JDK
- Mine is as following :
- > C:\Program Files\Java\jdk-14.0.1\bin

- Open eclipse in your confined workspace
- Click on new and open a new JAVA Project, give it a name
- Create a package inside the src folder.
- Dump in all the files here inside the package.
- Now right click on the project and choose configure build path.
- Click on the libraries section and add external jars
- Extract the Cloudsim.tar file you downloaded
- Now import the jar files in that Cloudsim.tar into the external jars.
- Do remember to change the name of the package in all the source files.
- Now right click on the project and run the project as JAVA Application.
- Select the SJF_Scheduler.java file present in the list.

Conclusion:

Experiment No. 4

Title: Find a procedure to transfer the files from one virtual machine to another virtual machine.

Software Required: Oracle's Virtual Box

Theory:

Virtualization:

Virtualization:

Virtualization is a process that allows for more efficient utilization of physical computer hardware and is the foundation of cloud computing. Virtualization uses software to create an abstraction layer over computer hardware that allows the hardware elements of a single computer—processors, memory, storage and more—to be divided into multiple virtual computers, commonly called virtual machines (VMs). Each VM runs its own operating system (OS) and behaves like an independent computer, even though it is running on just a portion of the actual underlying computer hardware. t follows that virtualization enables more efficient utilization of physical computer hardware and allows a greater return on an organization's hardware investment.

Today, virtualization is a standard practice in enterprise IT architecture. It is also the technology that drives cloud computing economics. Virtualization enables cloud providers to serve users with their existing physical computer hardware; it enables cloud users to purchase only the computing resources they need when they need it, and to scale those resources cost-effectively as their workloads grow.

Virtual Machine

A virtual computer system is known as a "virtual machine" (VM): a tightly isolated software container with an operating system and application inside. Each self-contained VM is completely independent. Putting multiple VMs on a single computer enables several operating systems and applications to run on just one physical server, or "host."

A thin layer of software called a "hypervisor" decouples the virtual machines from the host and dynamically allocates computing resources to each virtual machine as needed.

Oracle VM Virtual Box

Oracle VM VirtualBox is cross-platform virtualization software. It allows users to extend their existing computer to run multiple operating systems including Microsoft Windows, Mac OS X, Linux, and Oracle Solaris, at the same time. Designed for IT professionals and developers, Oracle VM VirtualBox is ideal for testing, developing, demonstrating, and deploying solutions across multiple platforms from one machine. Software developers rely on Oracle VM VirtualBox Enterprise for the development and debugging of their applications in multiple operating systems and environments on one device. Developers can clone an environment on their personal desktop/laptop without impact to production services.

Private Cloud
Public
Hybrid Cloud
Download and Test
Dev and Deploy

Steps

1. Download and install Oracle's Virtual Box. (Reboot needed after installation)

   Install VirtualBox 5.0 or higher

2. Download Ubuntu VMDK Image

   https://releases.ubuntu.com/18.04/

3. Launch Virtualbox and create a new VM.

4. Click on new and mention the Name and the machine folder along with the Type and Version of the Machine to be created.

5. Assign memory size for our VM (1024 MB sufficient for now).

6. Select the option Use an existing virtual hard disk file and locate the donwloaded VMDK image below and create VM.

7. Now we have to create a NAT Network so go to File -> Preferences -> Network -> Add a New NAT Network (Click on +)

8. Right click and edit the Network name and CIDR if needed. Example :

Name - My VMbox Network
CIDR - 172.168.2.0/24 and save the changes.

9. Repeat the process of launching the VM for 2 instances.

10. Now go to the setting, go to the network setting and change the adapter to NAT Network and and select the NAT Network you made (in our case : My VMbox Network) and click ok.

11. Launch the VM now.

12. Install the net-tools to know the IP's of the instance.

```
$ sudo apt install net-tools
$ sudo apt update
```

13. To know the IP address

```
$ ifconfig
```

Now the IP will be in the range of 172.168.2.*

* - any number in the range of 1 to 254 (total 256 addresses)

14. Now create a file and write something into it.

```
$ touch tranfer.txt
$ nano transfer.txt
-> hey, How are you?
ctrl + X and save
```

Some Commands for Linux Based Distros:

ls - list all the files and directories
cat - show the content inside a file
scp - it will help us to copy files from one vm to other
cd - change directory
mkdir - make a new directory
touch - it makes a new file
nano - nano is a editor inside linux os

15. If your file is on the VM with IP 172.168.2.4 and the second VM's IP is 172.168.2.5.

16. Tranfer the file using SCP

```
$ scp tranfer.txt vagrant@172.168.2.5:/home/vagrant
```

Put in the password of the 2nd VM and done.

17. Check for the file in the Second VM under the /home/vagrant directory.

18. Done..!!!!

Screen Shot of Output Page:

Conclusion:

Experiment No. 5

Title: Find a procedure to launch virtual machine using try stack (Online Openstack Demo Version)

Software Required: Oracle's Virtual Box

Theory:

Virtualization:

Virtualization is a process that allows for more efficient utilization of physical computer hardware and is the foundation of cloud computing. Virtualization uses software to create an abstraction layer over computer hardware that allows the hardware elements of a single computer—processors, memory, storage and more—to be divided into multiple virtual computers, commonly called virtual machines (VMs). Each VM runs its own operating system (OS) and behaves like an independent computer, even though it is running on just a portion of the actual underlying computer hardware. t follows that virtualization enables more efficient utilization of physical computer hardware and allows a greater return on an organization's hardware investment.

Today, virtualization is a standard practice in enterprise IT architecture. It is also the technology that drives cloud computing economics. Virtualization enables cloud providers to serve users with their existing physical computer hardware; it enables cloud users to purchase only the computing resources they need when they need it, and to scale those resources cost-effectively as their workloads grow.

Virtual Machine

A virtual computer system is known as a "virtual machine" (VM): a tightly isolated software container with an operating system and application inside. Each self-contained VM is completely independent. Putting multiple VMs on a single computer enables several operating systems and applications to run on just one physical server, or "host."

A thin layer of software called a "hypervisor" decouples the virtual machines from the host and dynamically allocates computing resources to each virtual machine as needed.

Oracle VM Virtual Box

Oracle VM VirtualBox is cross-platform virtualization software. It allows users to extend their existing computer to run multiple operating systems including Microsoft Windows, Mac OS X, Linux, and Oracle Solaris, at the same time. Designed for IT professionals and developers, Oracle VM VirtualBox is ideal for testing, developing, demonstrating, and deploying solutions across multiple platforms from one machine. Software developers rely on Oracle VM VirtualBox Enterprise for the development and debugging of their applications in multiple operating systems and environments on one device. Developers can clone an environment on their personal desktop/laptop without impact to production services.

Private Cloud
Public
Hybrid Cloud
Download and Test
Dev and Deploy

Steps

9. Download and install Oracle's Virtual Box. (Reboot needed after installation)

   Install VirtualBox 5.0 or higher

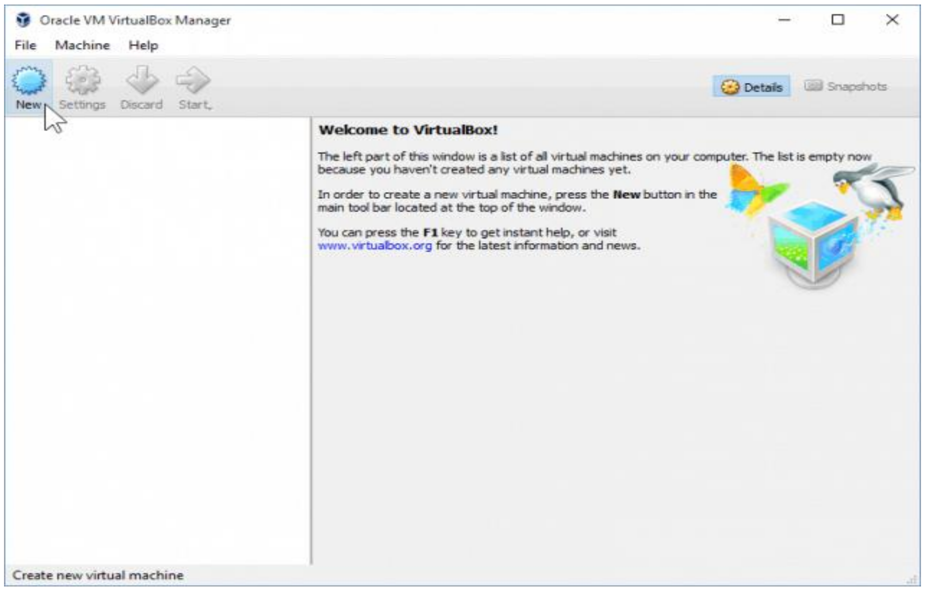
10. Download Ubuntu VMDK Image or Windows –xp- iso image

    https://releases.ubuntu.com/18.04/

    Installation media for Windows 10 (ISO or DVD)

    https://www.partitionwizard.com/partitionmagic/windows-xp-iso-download.html

11. Launch Virtualbox and create a new VM.


Oracle VM VirtualBox Manager

Welcome to VirtualBox!

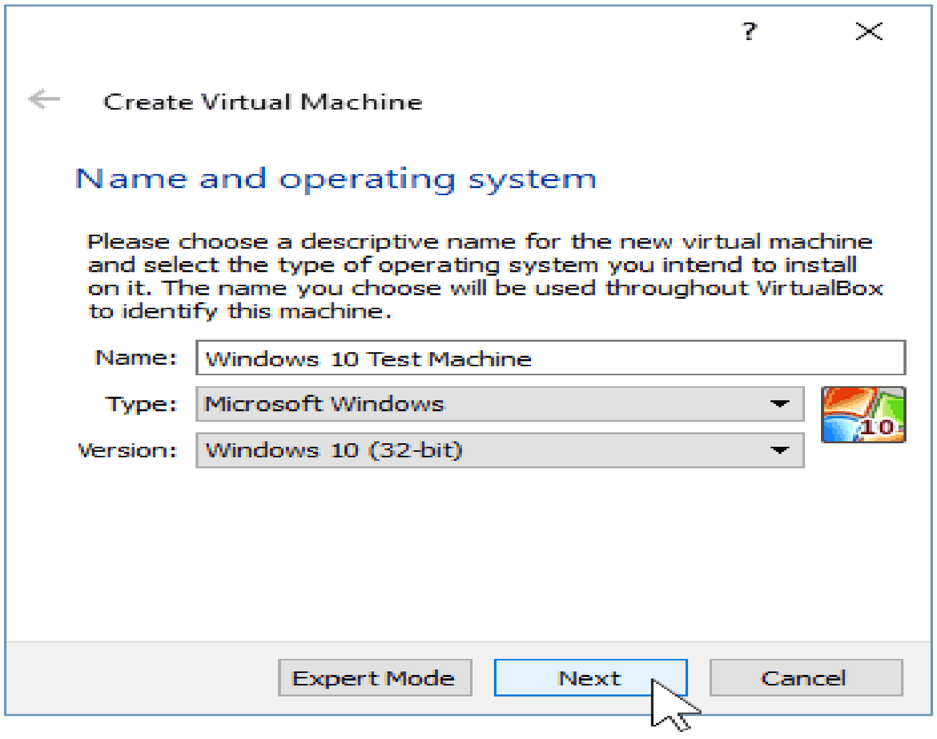
The left part of this window is a list of all virtual machines on your computer. The list is empty now because you haven't created any virtual machines yet.

In order to create a new virtual machine, press the **New** button in the main tool bar located at the top of the window.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for the latest information and news.

Create new virtual machine

12. Click on new and mention the Name and the machine folder along with the Type and Version of the Machine to be created.


Create Virtual Machine

Name and operating system

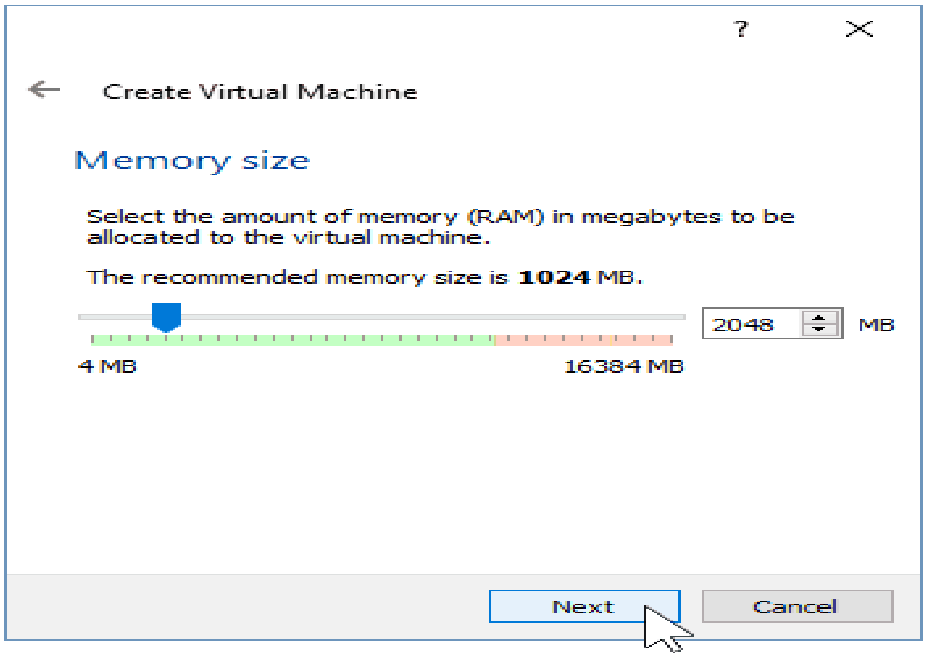
Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: Windows 10 Test Machine

Type: Microsoft Windows

Version: Windows 10 (32-bit)

13. Assign memory size for our VM (1024 MB sufficient for now).



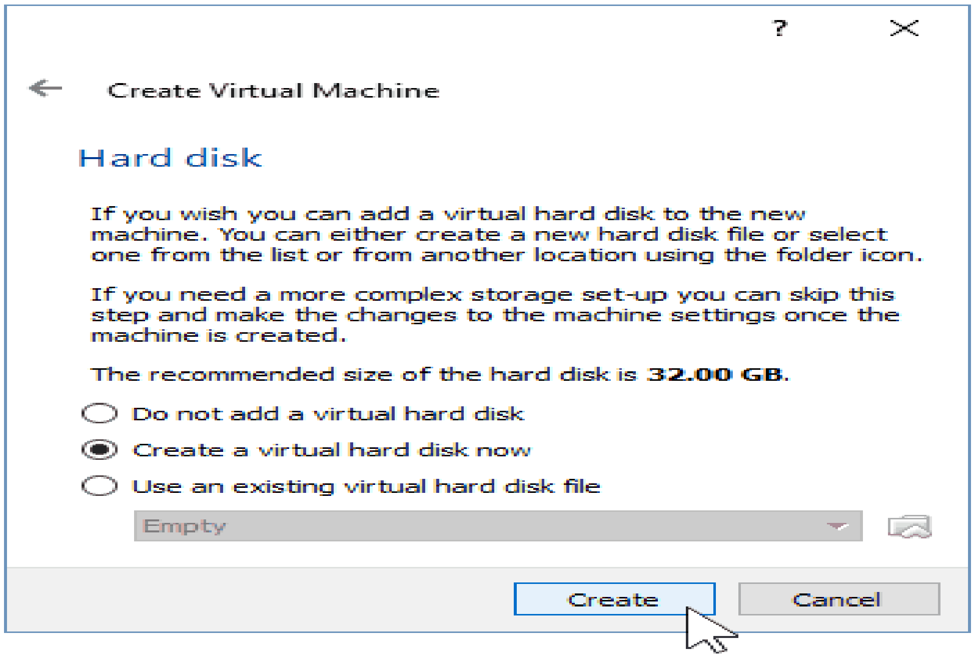
Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024** MB.

2048 MB

14. Configuring the HDD, you have to pick "Create a virtual hard disk now," assuming you don't have one created previously.



Hard disk

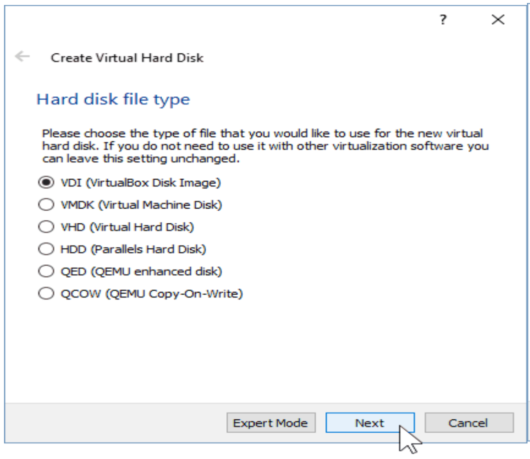
If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **32.00 GB**.

- Do not add a virtual hard disk
- [✓ Create a virtual hard disk now]
- Use an existing virtual hard disk file

15. You can leave the next two screens at their default settings since they work just fine with the new OS.

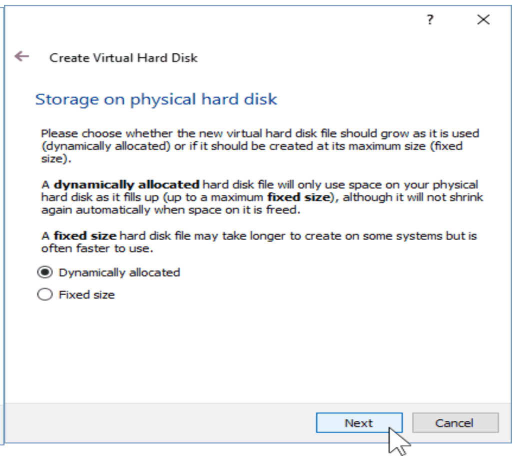


Create Virtual Hard Disk

Hard disk file type

Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

- [✓ VDI (VirtualBox Disk Image)]
- VMDK (Virtual Machine Disk)
- VHD (Virtual Hard Disk)
- HDD (Parallels Hard Disk)
- QED (QEMU enhanced disk)
- QCOW (QEMU Copy-On-Write)



Create Virtual Hard Disk

Storage on physical hard disk

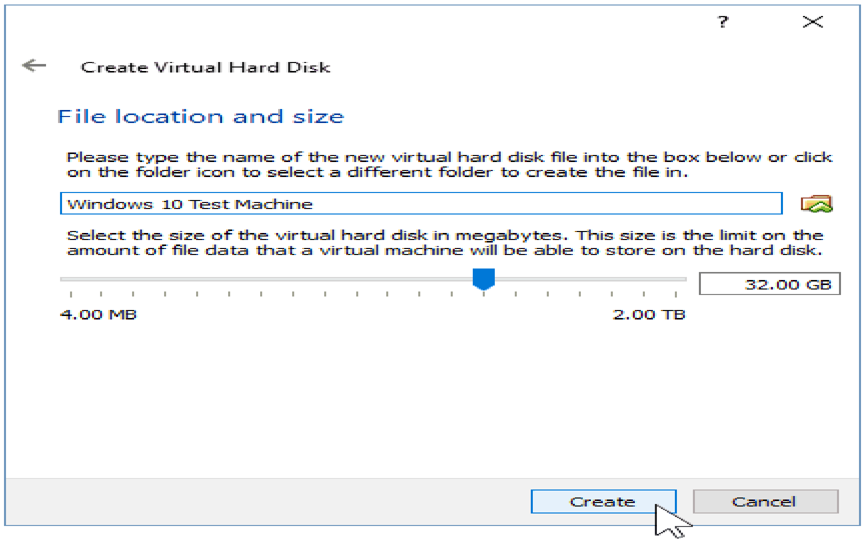
Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

- [✓ Dynamically allocated]
- Fixed size

16. At the final virtual hard disk screen, you’ll be able to change the location of the drive as well as its size.



Create Virtual Hard Disk

File location and size

Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

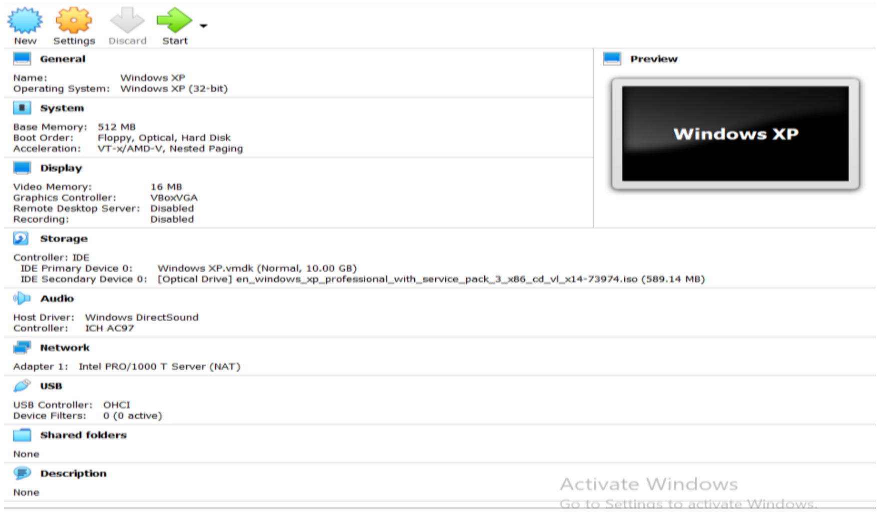
Windows 10 Test Machine

Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

32.00 GB

4.00 MB 2.00 TB

17. After clicking the “Create” button, you should be able to see your created VM



General

```
Name:              Windows XP
Operating System:  Windows XP (32-bit)
```

System

```
Base Memory:   512 MB
Boot Order:    Floppy, Optical, Hard Disk
Acceleration:  VT-x/AMD-V, Nested Paging
```

Display

```
Video Memory:            16 MB
Graphics Controller:     VBoxVGA
Remote Desktop Server:   Disabled
Recording:               Disabled
```

Storage

```
Controller: IDE
  IDE Primary Device 0:     Windows XP.vmdk (Normal, 10.00 GB)
  IDE Secondary Device 0:   [Optical Drive] en_windows_xp_professional_with_service_pack_3_x86_cd_vl_x14-73974.iso (589.14 MB)
```

Audio

```
Host Driver:   Windows DirectSound
Controller:    ICH AC97
```

Network

```
Adapter 1:  Intel PRO/1000 T Server (NAT)
```

USB

```
USB Controller:  OHCI
Device Filters:  0 (0 active)
```

Shared folders

None

Description

None

Screen Shot of Output Page:

Conclusion:

Experiment No. 6

Title: Design and deploy a PaaS environment.

Theory:

What is PaaS?

- Platform as a service (PaaS) is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app.

Common PaaS scenarios

- Organisations typically use PaaS for these scenarios:
- Development framework. PaaS provides a framework that developers can build upon to develop or customise cloud-based applications. Similar to the way you create an Excel macro, PaaS lets developers create applications using built-in software components. Cloud features such as scalability, high-availability and multi-tenant capability are included, reducing the amount of coding that developers must do.
- Analytics or business intelligence. Tools provided as a service with PaaS allow organisations to analyse and mine their data, finding insights and patterns and predicting outcomes to improve forecasting, product design decisions, investment returns and other business decisions.
- Additional services. PaaS providers may offer other services that enhance applications, such as workflow, directory, security and scheduling.

Advantages of PaaS

By delivering infrastructure as a service, PaaS offers the same advantages as IaaS. But its additional features—middleware, development tools and other business tools—give you more advantages:

Cut coding time. PaaS development tools can cut the time it takes to code new apps with pre-coded application components built into the platform, such as workflow, directory services, security features, search and so on.

Add development capabilities without adding staff. Platform as a Service component can give your development team new capabilities without your needing to add staff having the required skills.

Develop for multiple platforms—including mobile—more easily. Some service providers give you development options for multiple platforms, such as computers, mobile devices and browsers making cross-platform apps quicker and easier to develop.

Use sophisticated tools affordably. A pay-as-you-go model makes it possible for individuals or organizations to use sophisticated development software and business intelligence and analytics tools that they could not afford to purchase outright.

Support geographically distributed development teams. Because the development environment is accessed over the Internet, development teams can work together on projects even when team members are in remote locations.

Efficiently manage the application lifecycle. PaaS provides all of the capabilities that you need to support the complete web application lifecycle: building, testing, deploying, managing and updating within the same integrated environment.

AWS Amplify

AWS Amplify is a set of purpose-built tools and features that lets frontend web and mobile developers quickly and easily build full-stack applications on AWS, with the flexibility to leverage the breadth of AWS services as your use cases evolve. With Amplify, you can configure a web or mobile app backend, connect your app in minutes, visually build a web frontend UI, and easily manage app content outside the AWS console. Ship faster and scale effortlessly—with no cloud expertise needed.

Procedure:

1. Login to the AWS console
2. Find for AWS Amplify in the services.
3. Get Started with Amplify service.
4. Click on Host a Web App.
5. Then choose to launch it with Github and authenticate your GitHub account for the same..
6. After that choose the Repository containing your source code (subfolder if needed)
7. Then Launch the application with the default configurations provided by AWS Amplify
8. Configurations may be different on type of framework / technology you are launching your application (REACT is my case).

Conclusion:

Case Study 1

Case Study 2