.EX.No:4 Installation of a GoogleApp Engine

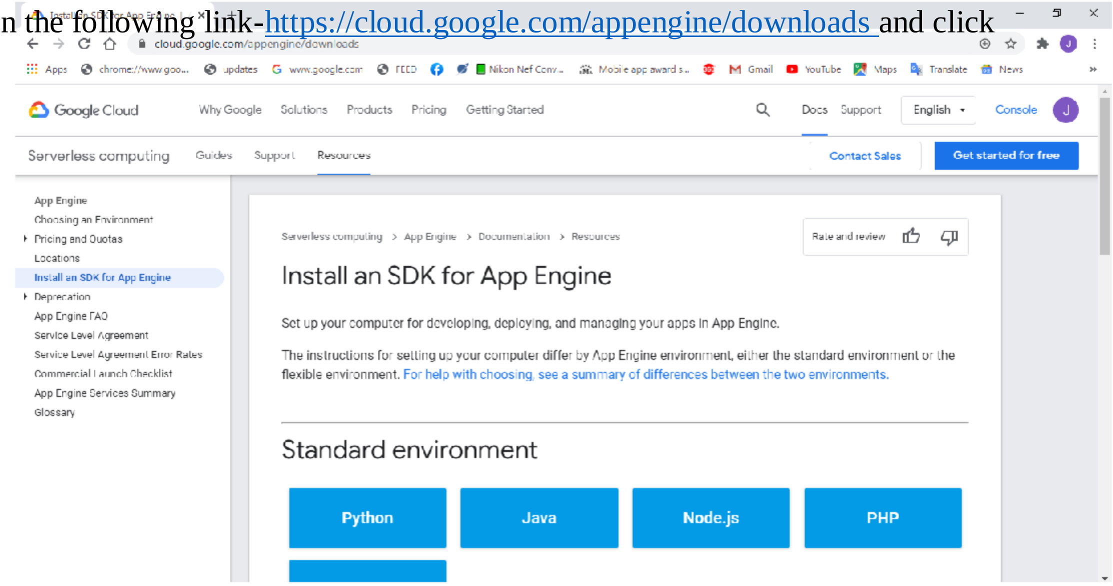
Date:

Aim:

To install the Google app engine in the system.

Procedure:

Step1: Open the following link-https://cloud.google.com/appengine/downloads and click python.



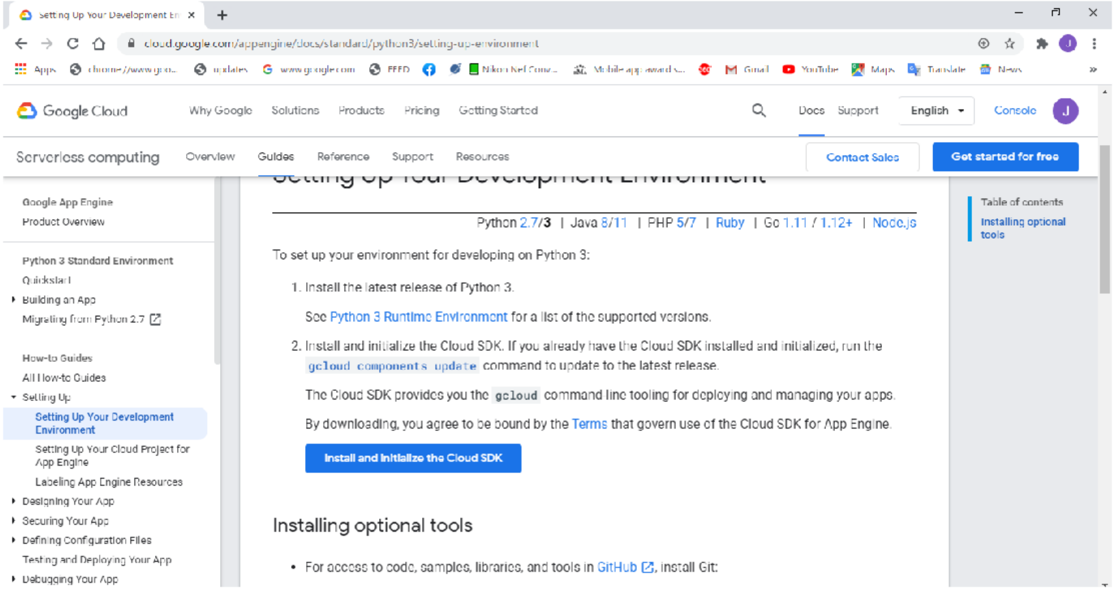
Install an SDK for App Engine

Set up your computer for developing, deploying, and managing your apps in App Engine.

The instructions for setting up your computer differ by App Engine environment, either the standard environment or the flexible environment. For help with choosing, see a summary of differences between the two environments.

Standard environment

Step2: Select setting up your environment development and click on install and initialize the cloud sdk.


Setting Up Your Development En
cloud.google.com/appengine/docs/standard/python3/setting-up-environment
Google Cloud
Why Google
Solutions
Products
Pricing
Getting Started
Docs
Support
English
Console
Serverless computing
Overview
Guides
Reference
Support
Resources
Contact Sales
Get started for free
Google App Engine
Product Overview
Python 3 Standard Environment
Quickstart
Building an App
Migrating from Python 2.7
How-to Guides
All How-to Guides
Setting Up
Setting Up Your Development Environment
Setting Up Your Cloud Project for App Engine
Labeling App Engine Resources
Designing Your App
Securing Your App
Defining Configuration Files
Testing and Deploying Your App
Debugging Your App
Table of contents
Installing optional tools
Python 2.7/3 | Java 8/11 | PHP 5/7 | Ruby | Go 1.11 / 1.12+ | Node.js
Install and Initialize the Cloud SDK

Setting Up Your Development Environment

To set up your environment for developing on Python 3:

1. Install the latest release of Python 3.

   See Python 3 Runtime Environment for a list of the supported versions.

2. Install and initialize the Cloud SDK. If you already have the Cloud SDK installed and initialized, run the `gcloud components update` command to update to the latest release.

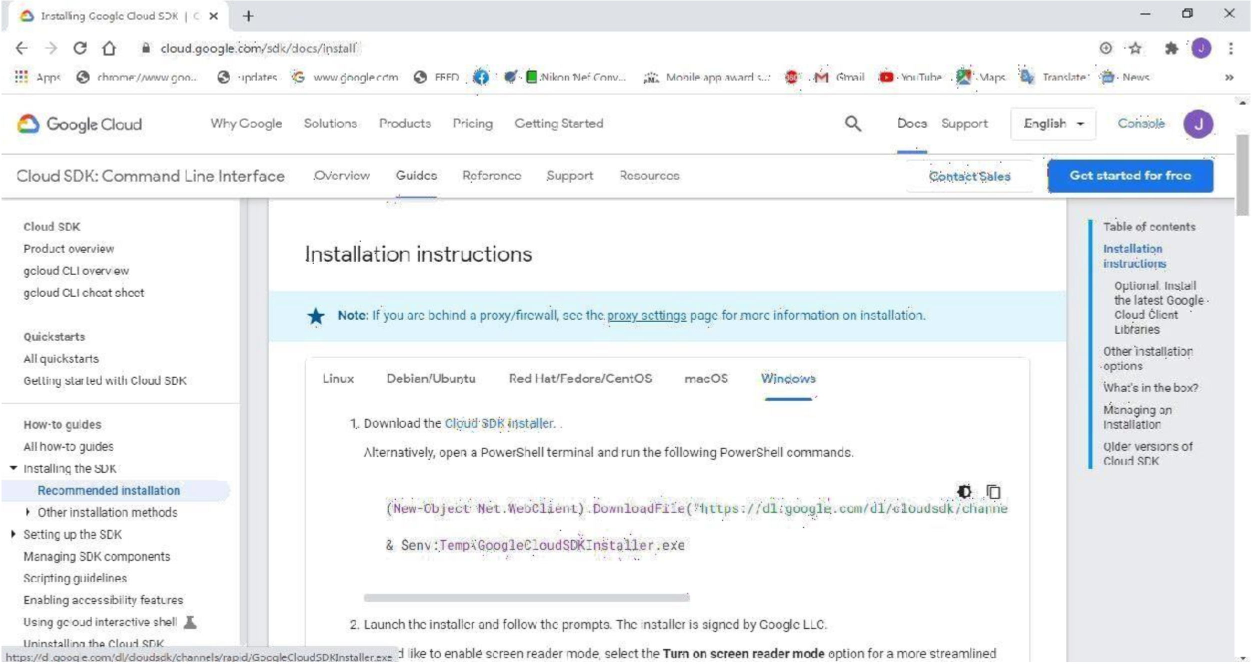
   The Cloud SDK provides you the `gcloud` command line tooling for deploying and managing your apps.

   By downloading, you agree to be bound by the Terms that govern use of the Cloud SDK for App Engine.

Installing optional tools

- For access to code, samples, libraries, and tools in GitHub, install Git:

Step3:Download the sdk installer and install it.



Installation instructions

Note: If you are behind a proxy/firewall, see the proxy settings page for more information on installation.

1. Download the Cloud SDK installer.

   Alternatively, open a PowerShell terminal and run the following PowerShell commands.

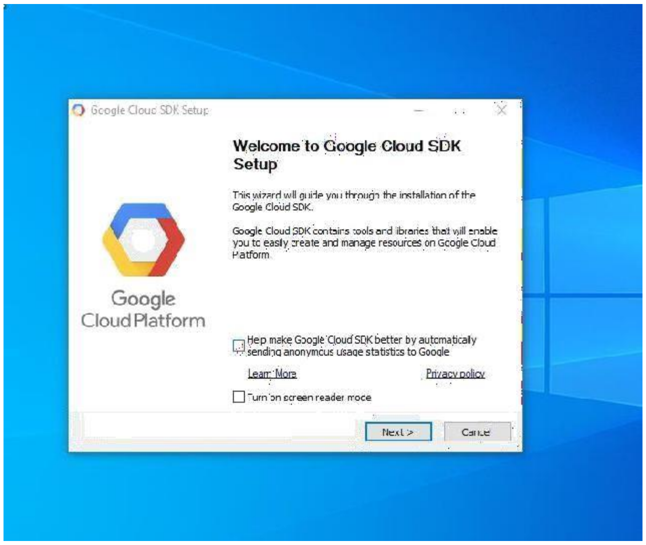
```
(New-Object Net.WebClient).DownloadFile("https://dl.google.com/dl/cloudsdk/channe

& $env:Temp\GoogleCloudSDKInstaller.exe
```

2. Launch the installer and follow the prompts. The installer is signed by Google LLC.

   d like to enable screen reader mode, select the **Turn on screen reader mode** option for a more streamlined

Step4: Click Next.



Welcome to Google Cloud SDK Setup

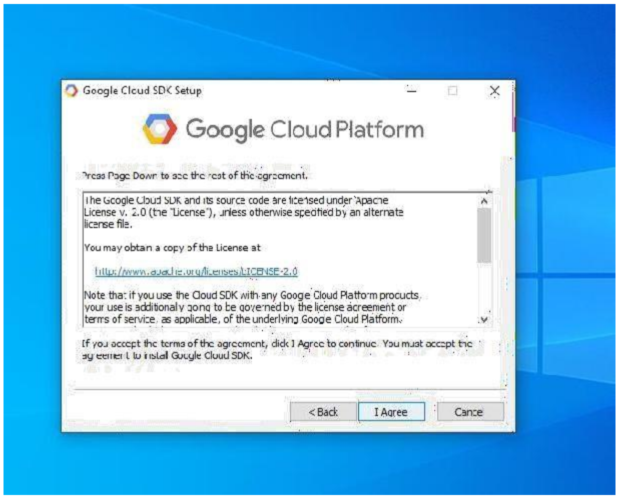
This wizard will guide you through the installation of the Google Cloud SDK.

Google Cloud SDK contains tools and libraries that will enable you to easily create and manage resources on Google Cloud Platform.

☑ Help make Google Cloud SDK better by automatically sending anonymous usage statistics to Google

□ Turn on screen reader mode

Step5:Click I Agree.



Google Cloud SDK Setup

Press Page Down to see the rest of the agreement.

The Google Cloud SDK and its source code are licensed under Apache License v. 2.0 (the "License"), unless otherwise specified by an alternate license file.

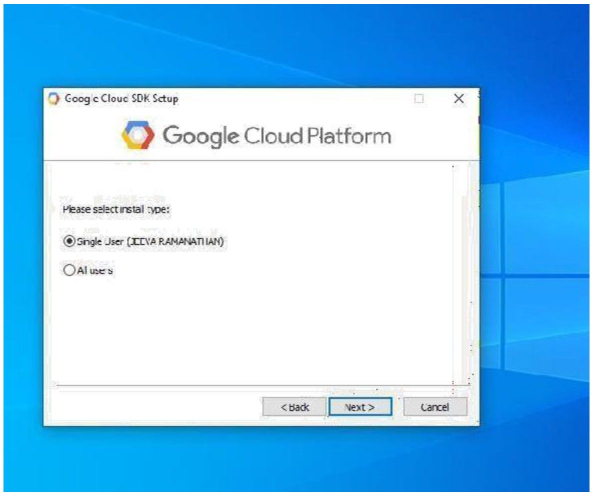
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Note that if you use the Cloud SDK with any Google Cloud Platform products, your use is additionally going to be governed by the license agreement or terms of service, as applicable, of the underlying Google Cloud Platform.

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Google Cloud SDK.

Step6: Select single user and click Next.



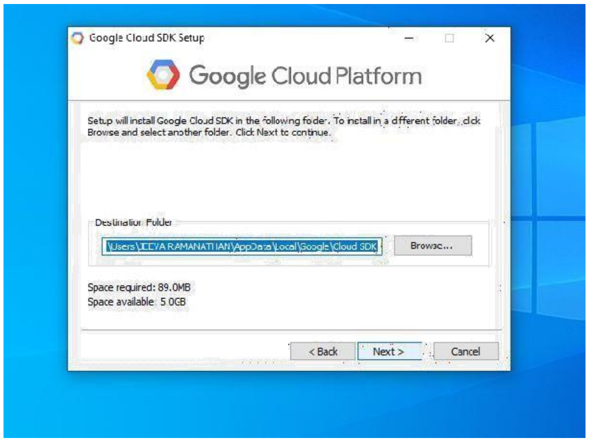
Google Cloud SDK Setup

Please select install type:

● Single User (JEEVA RAMANATHAN)

○ All users

Step7: Select the destination location and click Next.



Google Cloud SDK Setup

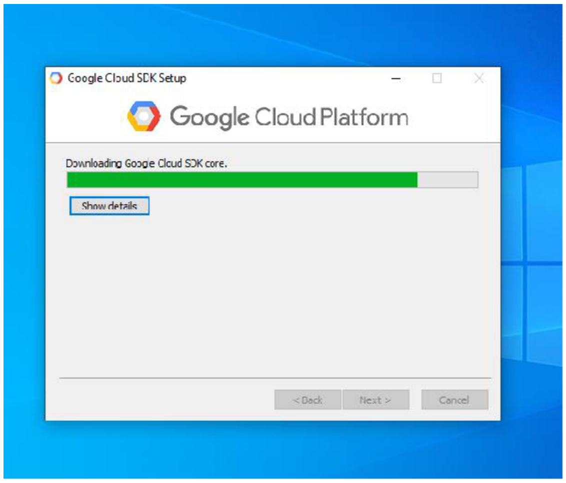
Setup will install Google Cloud SDK in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

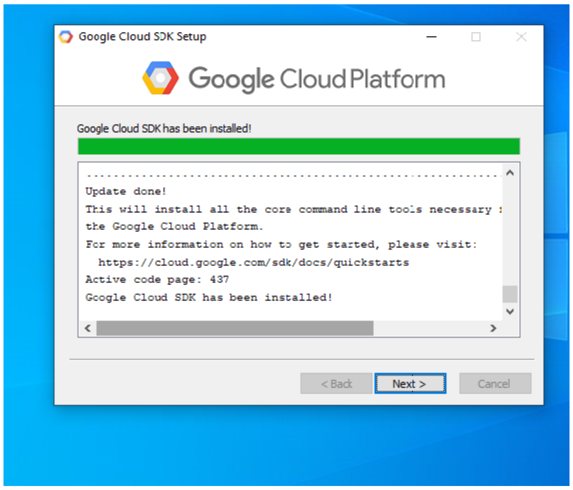
\Users\JEEVA RAMANATHAN\AppData\Local\Google\Cloud SDK

Space required: 89.0MB
Space available: 5.0GB

Step8: Downloading all the requirements and installing



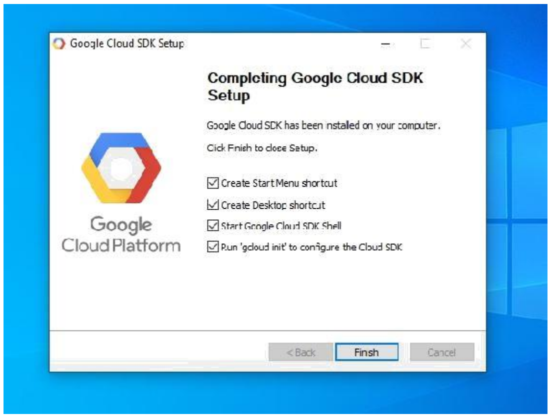
Downloading Googe Cloud SDK core.



Google Cloud SDK has been installed!

```
.............................................................
Update done!
This will install all the core command line tools necessary f
the Google Cloud Platform.
For more information on how to get started, please visit:
  https://cloud.google.com/sdk/docs/quickstarts
Active code page: 437
Google Cloud SDK has been installed!
```

Step9:Click Finish.



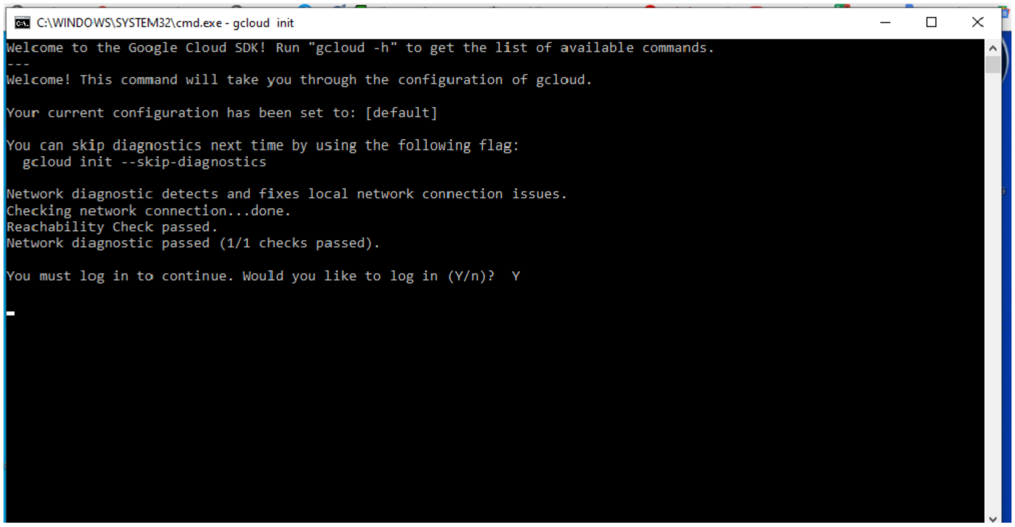
Completing Google Cloud SDK Setup

Google Cloud SDK has been nstaled on your computer.

Cick Finish to close Setup.

☑ Create Start Menu shortcut
☑ Create Desktop shortcut
☑ Start Google Cloud SDK Shell
☑ Run 'gcloud init' to configure the Cloud SDK

Step10:Once successfully installed cmd line in login with your Google account.



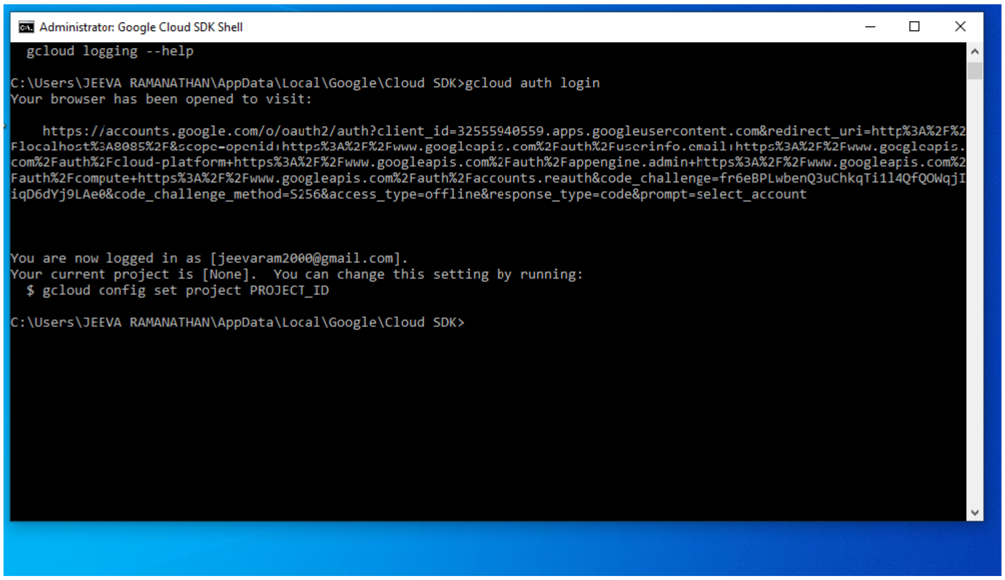
```
Welcome to the Google Cloud SDK! Run "gcloud -h" to get the list of available commands.
---
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

You must log in to continue. Would you like to log in (Y/n)?  Y
```



```
  gcloud logging --help

C:\Users\JEEVA RAMANATHAN\AppData\Local\Google\Cloud SDK>gcloud auth login
Your browser has been opened to visit:

    https://accounts.google.com/o/oauth2/auth?client_id=32555940559.apps.googleusercontent.com&redirect_uri=http%3A%2F%2
Flocalhost%3A8085%2F&scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.
com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2
Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&code_challenge=fr6eBPLwbenQ3uChkqTi1l4QfQOWqjI
iqD6dYj9LAe0&code_challenge_method=S256&access_type=offline&response_type=code&prompt=select_account


You are now logged in as [jeevaram2000@gmail.com].
Your current project is [None].  You can change this setting by running:
  $ gcloud config set project PROJECT_ID

C:\Users\JEEVA RAMANATHAN\AppData\Local\Google\Cloud SDK>
```

Result:

Thus googleapp engine is installed successfully in the system.

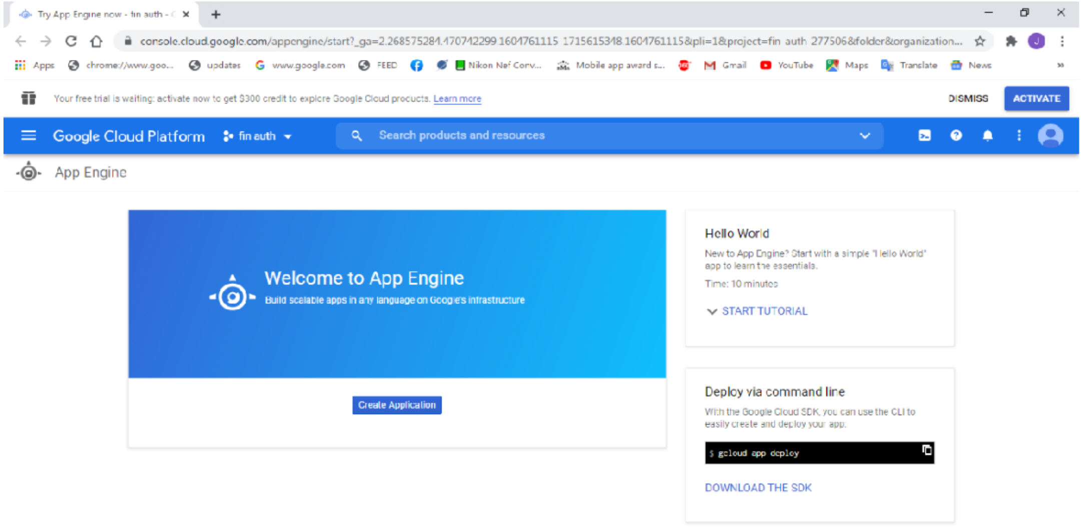
EX.No:4 Launch a web application using GAE launcher

Aim:

To launch the web applications by using the GAE launcher.

Procedure:

Step 1:Go to the following website https://console.cloud.google.com/start/appengine?_ga=2.268575284.470742299.1604761115-1715615348.1604761115&pli=1 and create a new project.



console.cloud.google.com/appengine/start?_ga=2.268575284.470742299.1604761115-1715615348.1604761115&pli=1&project=fin-auth-277506&folder&organization...

Your free trial is waiting: activate now to get $300 credit to explore Google Cloud products. Learn more

Welcome to App Engine

Build scalable apps in any language on Google's infrastructure

Hello World

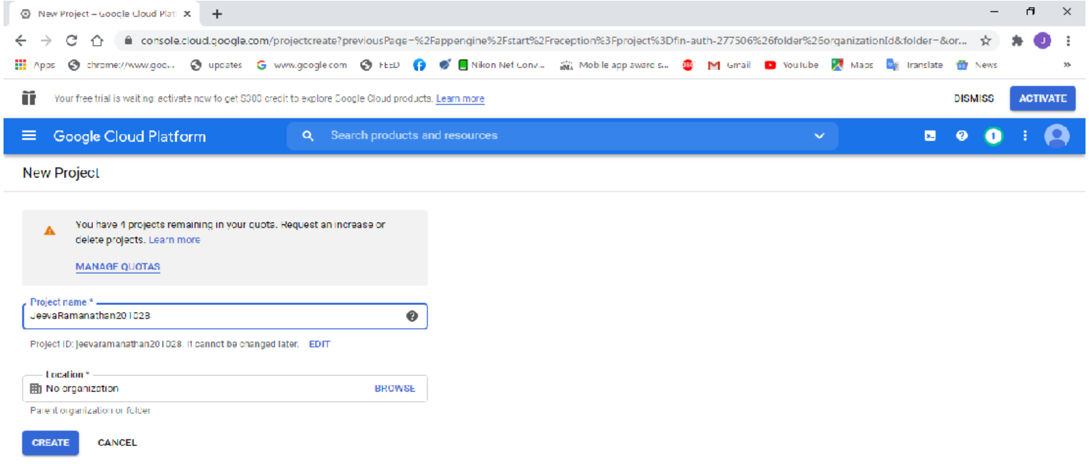
New to App Engine? Start with a simple "Hello World" app to learn the essentials.

Time: 10 minutes

Deploy via command line

With the Google Cloud SDK, you can use the CLI to easily create and deploy your app.

```
$ gcloud app deploy
```



console.cloud.google.com/projectcreate?previousPage=%2Fappengine%2Fstart%2Freception%3Fproject%3Dfin-auth-277506%26folder%26organizationId&folder=&or...

Your free trial is waiting: activate now to get $300 credit to explore Google Cloud products. Learn more

You have 4 projects remaining in your quota. Request an increase or delete projects. Learn more

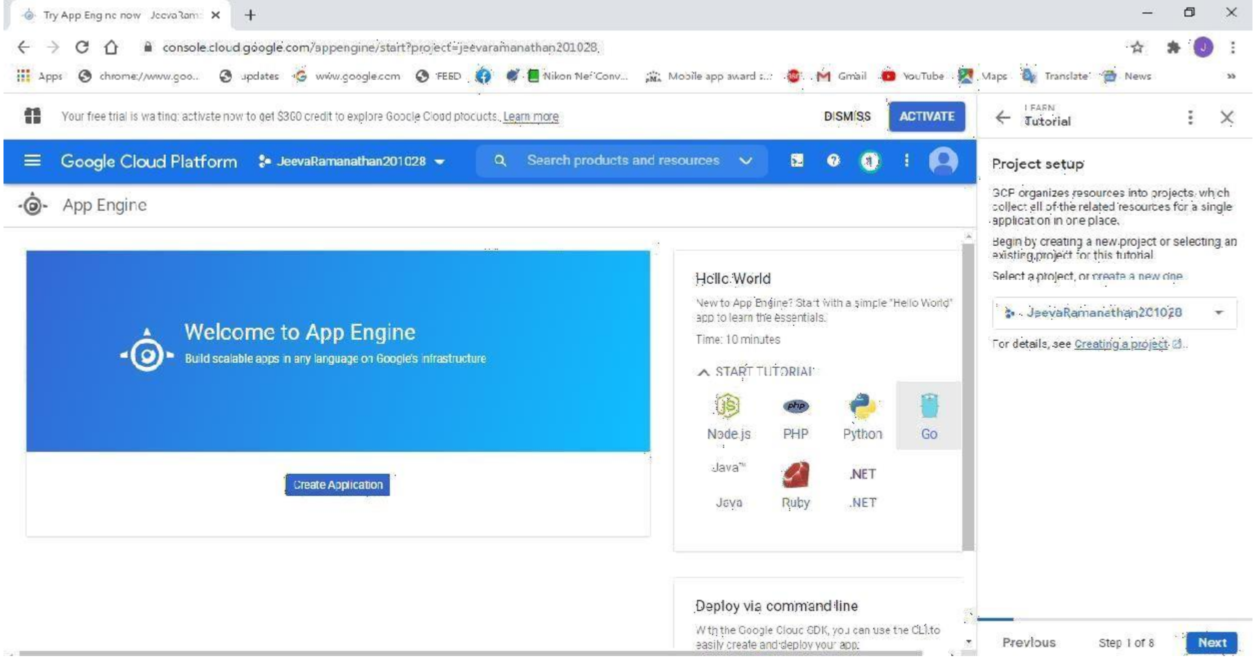
Project name *: JeevaRamanathan201028

Project ID: jeevaramanathan201028. It cannot be changed later. EDIT

Location *: No organization

Parent organization or folder

Step2:Select python and click next.



Your free trial is waiting: activate now to get $300 credit to explore Google Cloud products. Learn more

Hello World

New to App Engine? Start with a simple "Hello World" app to learn the essentials.

Time: 10 minutes

Deploy via command line

With the Google Cloud SDK, you can use the CLI to easily create and deploy your app.

Project setup

GCP organizes resources into projects, which collect all of the related resources for a single application in one place.

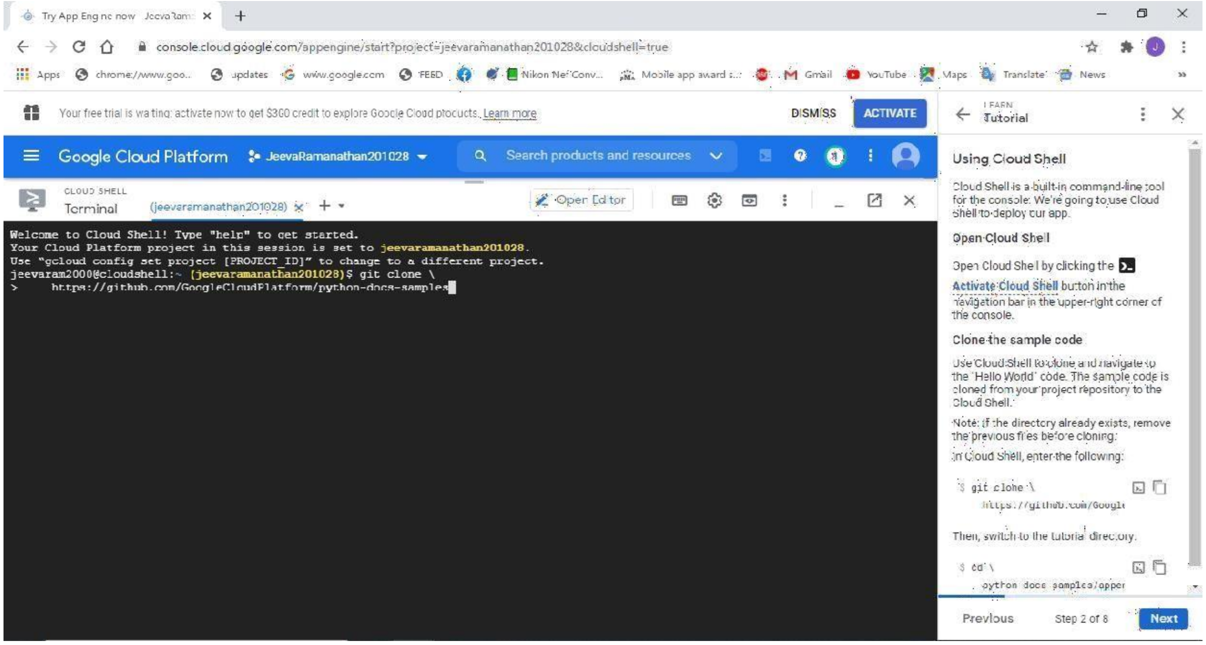
Begin by creating a new project or selecting an existing project for this tutorial.

Select a project, or create a new one.

For details, see Creating a project.

Step3: Open the cloud shell and follow the steps in the tutorial.

Clone the repository by using the given command



Your free trial is waiting: activate now to get $300 credit to explore Google Cloud products. Learn more

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to jeevaramanathan201028.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
jeevaram2000@cloudshell:~ (jeevaramanathan201028)$ git clone \
>     https://github.com/GoogleCloudPlatform/python-docs-samples
```

Using Cloud Shell

Cloud Shell is a built-in command-line tool for the console. We're going to use Cloud Shell to deploy our app.

Open Cloud Shell

Open Cloud Shell by clicking the Activate Cloud Shell button in the navigation bar in the upper-right corner of the console.

Clone the sample code

Use Cloud Shell to clone and navigate to the "Hello World" code. The sample code is cloned from your project repository to the Cloud Shell.

Note: If the directory already exists, remove the previous files before cloning:

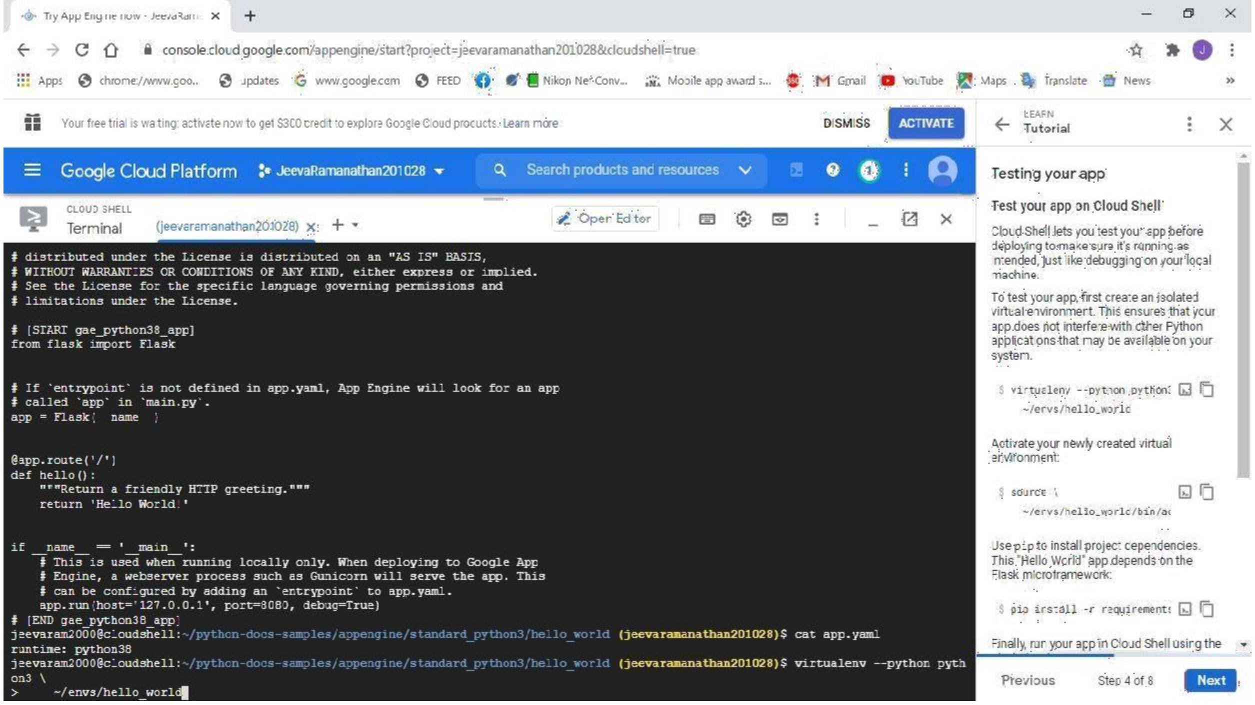
In Cloud Shell, enter the following:

```
$ git clone \
    https://github.com/Google
```

Then, switch to the tutorial directory.

```
$ cd \
    python-docs-samples/apper
```

Step4: Create the virtual environment


Try App Engine now - JeevaRam
console.cloud.google.com/appengine/start?project=jeevaramanathan201028&cloudshell=true
Apps
chrome://www.goo...
updates
www.google.com
FEED
Nikon Nef Conv...
Mobile app award s...
Gmail
YouTube
Maps
Translate
News
Your free trial is waiting: activate now to get $300 credit to explore Google Cloud products. Learn more
DISMISS
ACTIVATE
Google Cloud Platform
JeevaRamanathan201028
Search products and resources
CLOUD SHELL
Terminal
(jeevaramanathan201028)
Open Editor
LEARN
Tutorial

```
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# [START gae_python38_app]
from flask import Flask


# If `entrypoint` is not defined in app.yaml, App Engine will look for an app
# called `app` in `main.py`.
app = Flask(__name__)


@app.route('/')
def hello():
    """Return a friendly HTTP greeting."""
    return 'Hello World!'


if __name__ == '__main__':
    # This is used when running locally only. When deploying to Google App
    # Engine, a webserver process such as Gunicorn will serve the app. This
    # can be configured by adding an `entrypoint` to app.yaml.
    app.run(host='127.0.0.1', port=8080, debug=True)
# [END gae_python38_app]
jeevaram2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ cat app.yaml
runtime: python38
jeevaram2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ virtualenv --python pyth
on3 \
>     ~/envs/hello_world
```

Testing your app

Test your app on Cloud Shell

Cloud Shell lets you test your app before deploying to make sure it's running as intended, just like debugging on your local machine.

To test your app, first create an isolated virtual environment. This ensures that your app does not interfere with other Python applications that may be available on your system.

```
$ virtualenv --python python3
    ~/envs/hello_world
```

Activate your newly created virtual environment:

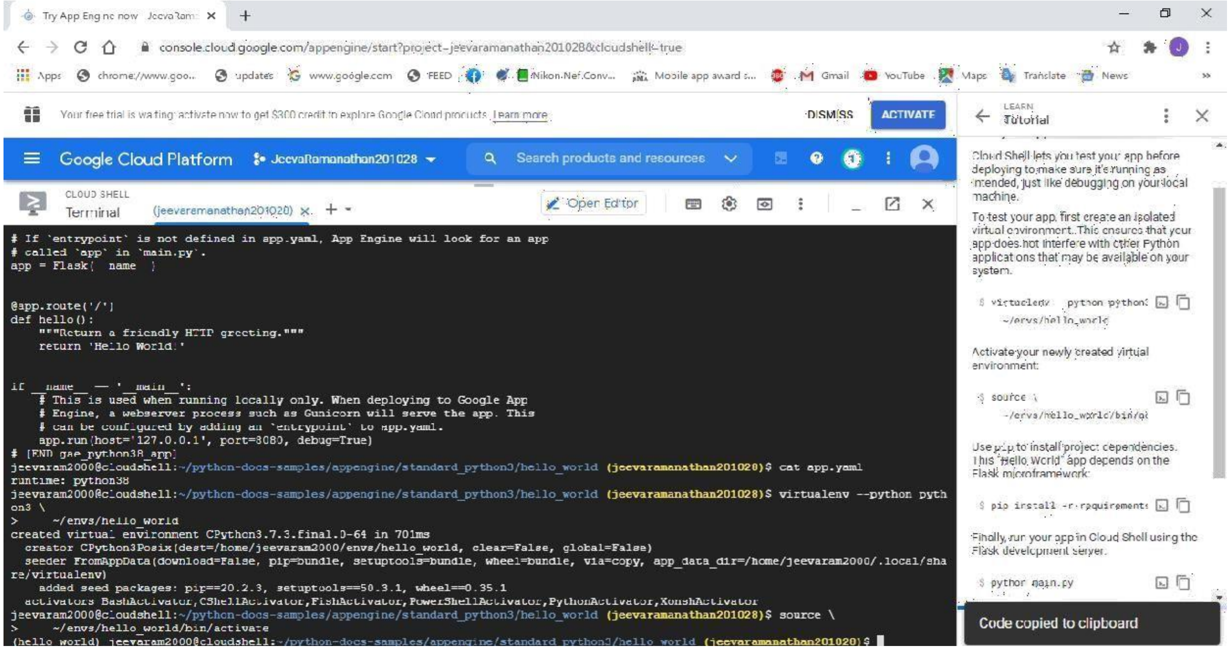
```
$ source \
    ~/envs/hello_world/bin/ac
```

Use pip to install project dependencies. This "Hello World" app depends on the Flask microframework:

```
$ pip install -r requirements
```

Finally, run your app in Cloud Shell using the

Step5: Activate your virtual environment.


Try App Engine now - JeevaRam
console.cloud.google.com/appengine/start?project=jeevaramanathan201028&cloudshell=true
Apps
chrome://www.goo...
updates
www.google.com
FEED
Nikon Nef Conv...
Mobile app award s...
Gmail
YouTube
Maps
Translate
News
Your free trial is waiting: activate now to get $300 credit to explore Google Cloud products. Learn more
DISMISS
ACTIVATE
Google Cloud Platform
JeevaRamanathan201028
Search products and resources
CLOUD SHELL
Terminal
(jeevaramanathan201028)
Open Editor
LEARN
Tutorial

```
# If `entrypoint` is not defined in app.yaml, App Engine will look for an app
# called `app` in `main.py`.
app = Flask(__name__)


@app.route('/')
def hello():
    """Return a friendly HTTP greeting."""
    return 'Hello World!'


if __name__ == '__main__':
    # This is used when running locally only. When deploying to Google App
    # Engine, a webserver process such as Gunicorn will serve the app. This
    # can be configured by adding an `entrypoint` to app.yaml.
    app.run(host='127.0.0.1', port=8080, debug=True)
# [END gae_python38_app]
jeevaram2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ cat app.yaml
runtime: python38
jeevaram2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ virtualenv --python pyth
on3 \
>     ~/envs/hello_world
created virtual environment CPython3.7.3.final.0-64 in 701ms
  creator CPython3Posix(dest=/home/jeevaram2000/envs/hello_world, clear=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/jeevaram2000/.local/sha
re/virtualenv)
    added seed packages: pip==20.2.3, setuptools==50.3.1, wheel==0.35.1
  activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
jeevaram2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ source \
>     ~/envs/hello_world/bin/activate
(hello_world) jeevaram2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$
```

Cloud Shell lets you test your app before deploying to make sure it's running as intended, just like debugging on your local machine.

To test your app, first create an isolated virtual environment. This ensures that your app does not interfere with other Python applications that may be available on your system.

```
$ virtualenv  python python3
    ~/envs/hello_world
```

Activate your newly created virtual environment:

```
$ source \
    ~/envs/hello_world/bin/ac
```

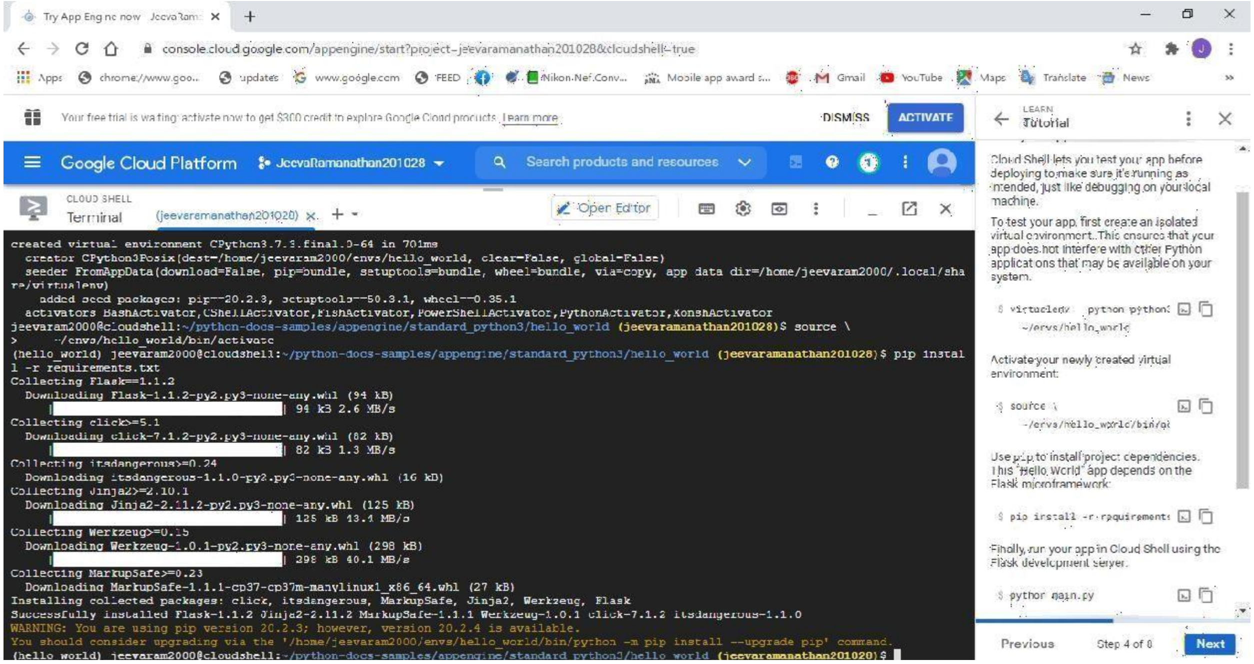
Use pip to install project dependencies. This "Hello World" app depends on the Flask microframework:

```
$ pip install -r requirements
```

Finally, run your app in Cloud Shell using the Flask development server.

```
$ python main.py
```

Step6: Installing requirements and run the app.



```
created virtual environment CPython3.7.3.final.0-64 in 701ms
  creator CPython3Posix(dest=/home/jeevaram2000/envs/hello_world, clear=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app data dir=/home/jeevaram2000/.local/sha
re/virtualenv)
    added seed packages: pip==20.2.3, setuptools==50.3.1, wheel==0.35.1
  activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
jeevaram2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ source \
>     ~/envs/hello_world/bin/activate
(hello_world) jeevaram2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ pip instal
l -r requirements.txt
Collecting Flask==1.1.2
  Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
     |████████████████████████████████| 94 kB 2.6 MB/s
Collecting click>=5.1
  Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
     |████████████████████████████████| 82 kB 1.3 MB/s
Collecting itsdangerous>=0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2>=2.10.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
     |████████████████████████████████| 125 kB 13.4 MB/s
Collecting Werkzeug>=0.15
  Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
     |████████████████████████████████| 298 kB 40.1 MB/s
Collecting MarkupSafe>=0.23
  Downloading MarkupSafe-1.1.1-cp37-cp37m-manylinux1_x86_64.whl (27 kB)
Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, Flask
Successfully installed Flask-1.1.2 Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 click-7.1.2 itsdangerous-1.1.0
WARNING: You are using pip version 20.2.3; however, version 20.2.4 is available.
You should consider upgrading via the '/home/jeevaram2000/envs/hello_world/bin/python -m pip install --upgrade pip' command.
(hello_world) jeevaram2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$
```

Cloud Shell lets you test your app before deploying to make sure it's running as intended, just like debugging on your local machine.

To test your app, first create an isolated virtual environment. This ensures that your app does not interfere with other Python applications that may be available on your system.

```
$ virtualenv --python python3 \
    ~/envs/hello_world
```

Activate your newly created virtual environment:

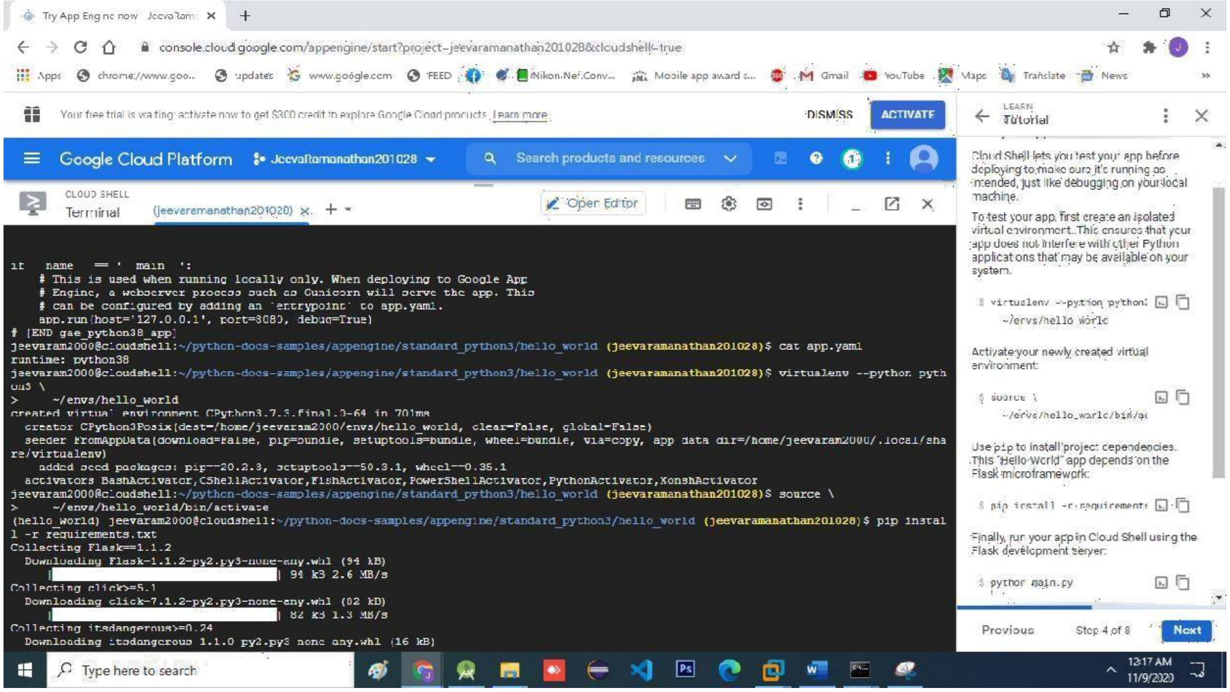
```
$ source \
    ~/envs/hello_world/bin/ac
```

Use pip to install project dependencies. This "Hello World" app depends on the Flask microframework:

```
$ pip install -r requirements
```

Finally, run your app in Cloud Shell using the Flask development server.

```
$ python main.py
```



```
if __name__ == '__main__':
    # This is used when running locally only. When deploying to Google App
    # Engine, a webserver process such as Gunicorn will serve the app. This
    # can be configured by adding an `entrypoint` to app.yaml.
    app.run(host='127.0.0.1', port=8080, debug=True)
# [END gae_python38_app]
jeevaram2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ cat app.yaml
runtime: python38
jeevaram2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ virtualenv --python pyth
on3 \
>     ~/envs/hello_world
created virtual environment CPython3.7.3.final.0-64 in 701ms
  creator CPython3Posix(dest=/home/jeevaram2000/envs/hello_world, clear=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app data dir=/home/jeevaram2000/.local/sha
re/virtualenv)
    added seed packages: pip==20.2.3, setuptools==50.3.1, wheel==0.35.1
  activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
jeevaram2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ source \
>     ~/envs/hello_world/bin/activate
(hello_world) jeevaram2000@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ pip instal
l -r requirements.txt
Collecting Flask==1.1.2
  Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
     |████████████████████████████████| 94 kB 2.6 MB/s
Collecting click>=5.1
  Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
     |████████████████████████████████| 82 kB 1.3 MB/s
Collecting itsdangerous>=0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
```

Cloud Shell lets you test your app before deploying to make sure it's running as intended, just like debugging on your local machine.

To test your app, first create an isolated virtual environment. This ensures that your app does not interfere with other Python applications that may be available on your system.

```
$ virtualenv --python python3 \
    ~/envs/hello_world
```

Activate your newly created virtual environment:

```
$ source \
    ~/envs/hello_world/bin/ac
```

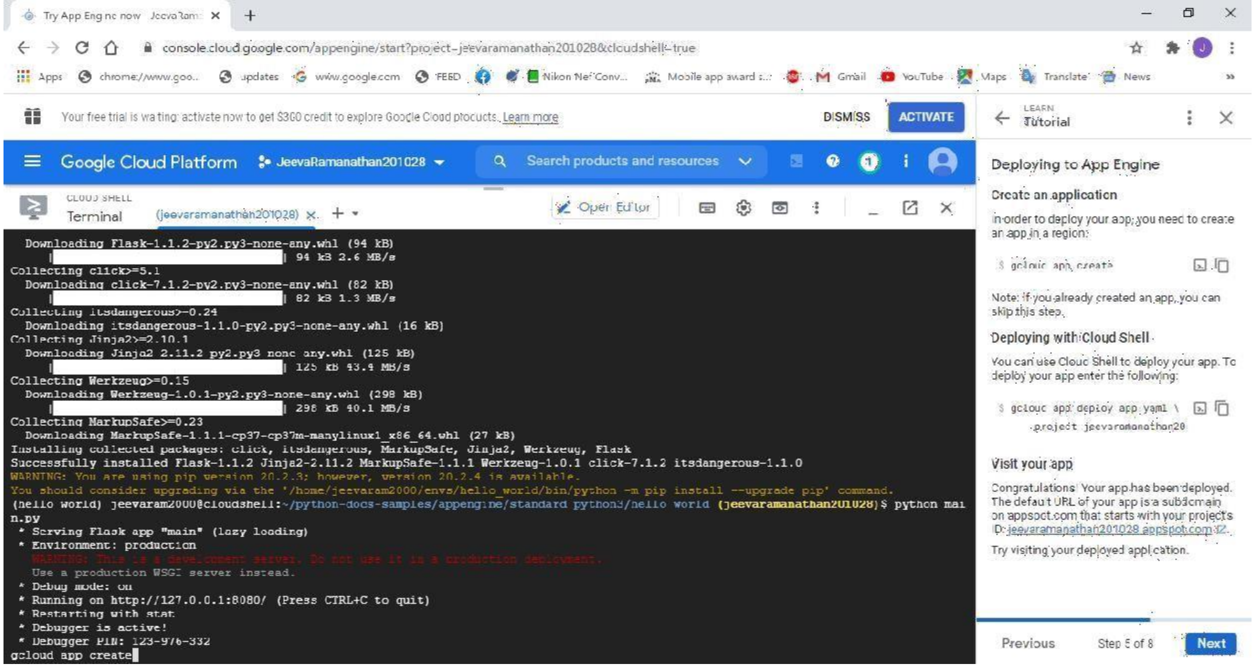
Use pip to install project dependencies. This "Hello World" app depends on the Flask microframework:

```
$ pip install -r requirements
```

Finally, run your app in Cloud Shell using the Flask development server.

```
$ python main.py
```

Step7: Create an application and deploy it in cloud shell.



```
  Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
     |████████████████████████████████| 94 kB 2.6 MB/s
Collecting click>=5.1
  Downloading click-7.1.2-py2.py3-none-any.whl (82 kB)
     |████████████████████████████████| 82 kB 1.3 MB/s
Collecting itsdangerous>=0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2>=2.10.1
  Downloading Jinja2 2.11.2 py2.py3 none any.whl (125 kB)
     |████████████████████████████████| 125 kB 43.4 MB/s
Collecting Werkzeug>=0.15
  Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
     |████████████████████████████████| 298 kB 40.1 MB/s
Collecting MarkupSafe>=0.23
  Downloading MarkupSafe-1.1.1-cp37-cp37m-manylinux1_x86_64.whl (27 kB)
Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, Flask
Successfully installed Flask-1.1.2 Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 click-7.1.2 itsdangerous-1.1.0
WARNING: You are using pip version 20.2.3; however, version 20.2.4 is available.
You should consider upgrading via the '/home/jeevaram2000/envs/hello_world/bin/python -m pip install --upgrade pip' command.
(hello world) jeevaram2000@cloudshell:~/python-docs-samples/appengine/standard python3/hello world (jeevaramanathan201028)$ python mai
n.py
 * 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: on
 * Running on http://127.0.0.1:8080/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 123-976-332
gcloud app create
```

Deploying to App Engine

Create an application

in order to deploy your app, you need to create an app in a region:

```
$ gcloud app create
```

Note: If you already created an app, you can skip this step.

Deploying with Cloud Shell

You can use Cloud Shell to deploy your app. To deploy your app enter the following:

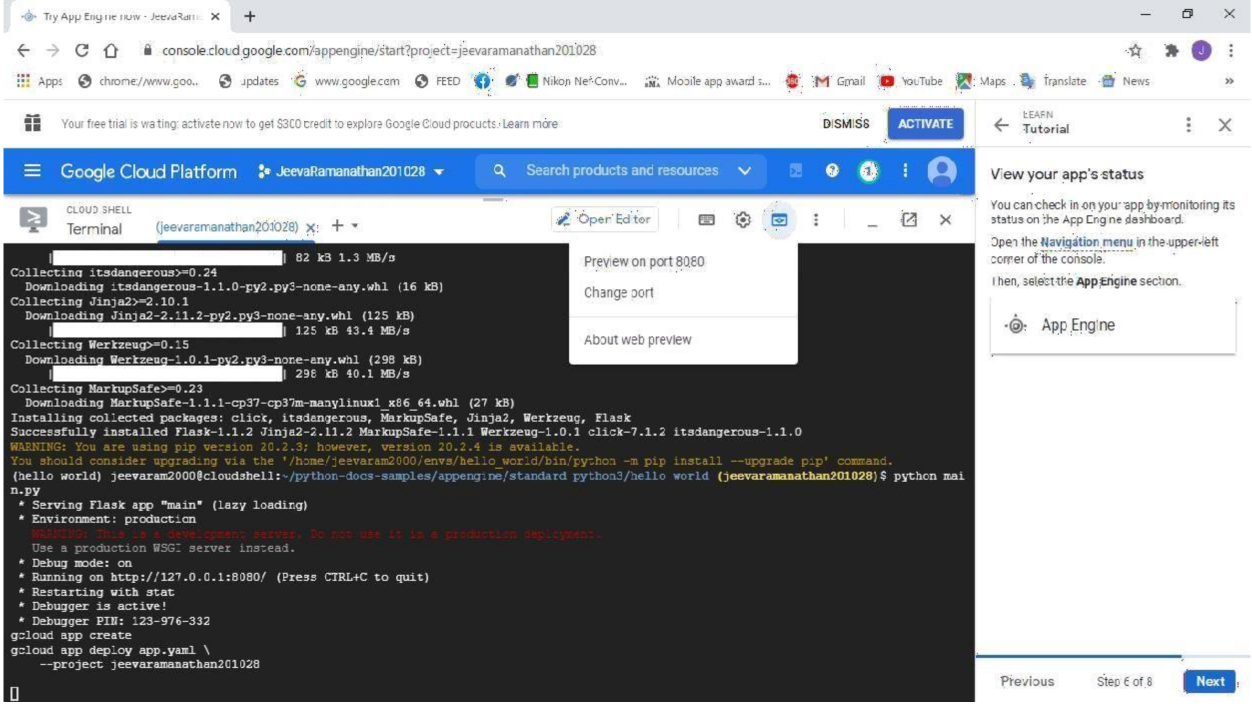
```
$ gcloud app deploy app.yaml \
    -project jeevaramanathan20
```

Visit your app

Congratulations! Your app has been deployed. The default URL of your app is a subdomain on appspot.com that starts with your project's ID: jeevaramanathan201028.appspot.com.

Try visiting your deployed application.

Step8: Click preview on port 8080 to see your deployed application



```
     |████████████████████████████████| 82 kB 1.3 MB/s
Collecting itsdangerous>=0.24
  Downloading itsdangerous-1.1.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2>=2.10.1
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
     |████████████████████████████████| 125 kB 43.4 MB/s
Collecting Werkzeug>=0.15
  Downloading Werkzeug-1.0.1-py2.py3-none-any.whl (298 kB)
     |████████████████████████████████| 298 kB 40.1 MB/s
Collecting MarkupSafe>=0.23
  Downloading MarkupSafe-1.1.1-cp37-cp37m-manylinux1_x86_64.whl (27 kB)
Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, Flask
Successfully installed Flask-1.1.2 Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 click-7.1.2 itsdangerous-1.1.0
WARNING: You are using pip version 20.2.3; however, version 20.2.4 is available.
You should consider upgrading via the '/home/jeevaram2000/envs/hello_world/bin/python -m pip install --upgrade pip' command.
(hello world) jeevaram2000@cloudshell:~/python-docs-samples/appengine/standard python3/hello world (jeevaramanathan201028)$ python mai
n.py
 * 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: on
 * Running on http://127.0.0.1:8080/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 123-976-332
gcloud app create
gcloud app deploy app.yaml \
    --project jeevaramanathan201028

▯
```

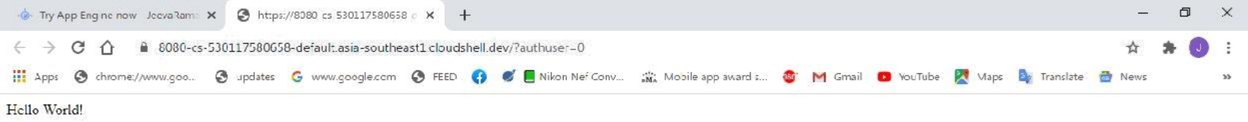
View your app's status

You can check in on your app by monitoring its status on the App Engine dashboard.

Open the **Navigation menu** in the upper-left corner of the console.

Then, select the **App Engine** section.

Step9: Finally the application is deployed and the output is seen.



8080-cs-530117580658-default.asia-southeast1.cloudshell.dev/?authuser=0

```
Hello World!
```

Result:

Thus a web application is launched by using the GAE launcher and the output is obtained successfully.