

Lab Manual For Cloud Computing:-----

EX.No:1

Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows

Aim:

To install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8 or 10.

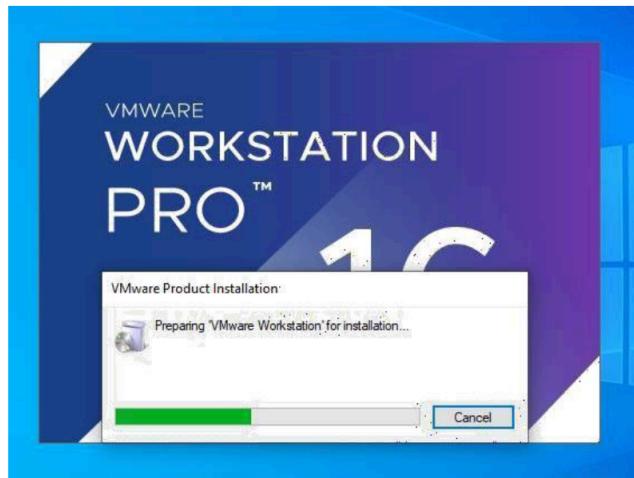
Procedure:

Downloading and installing VMware

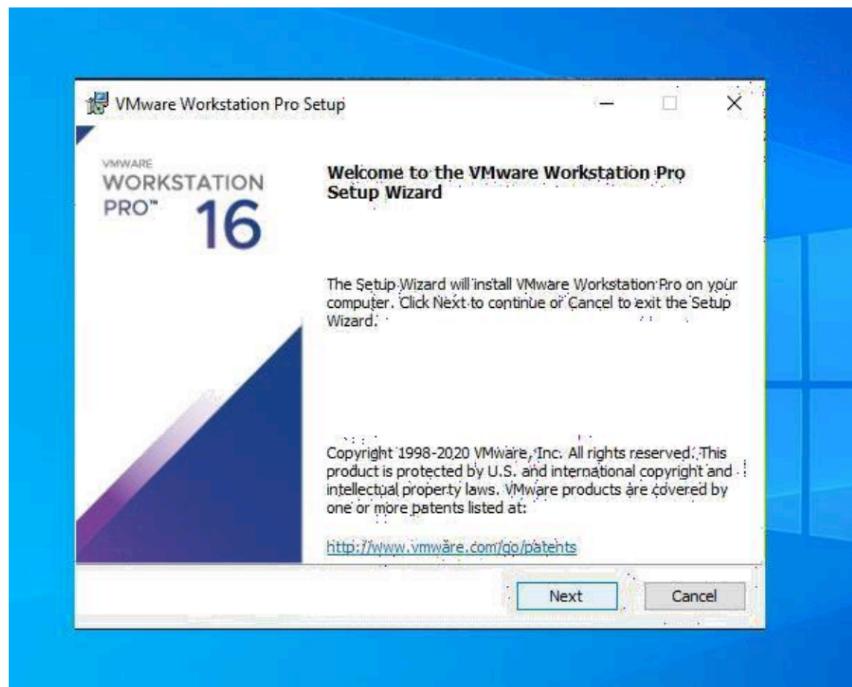
Step 1: Download VMware



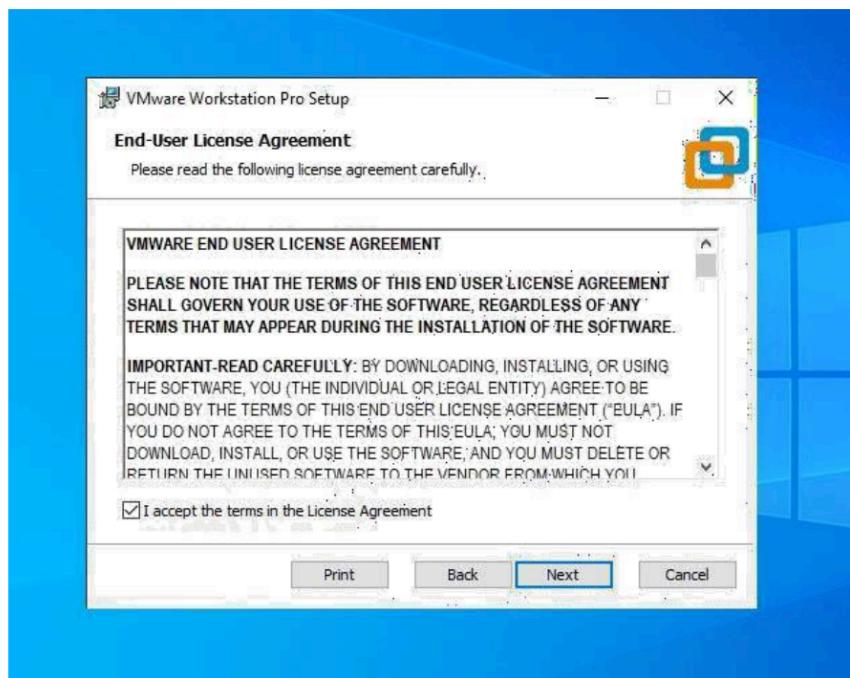
Step 2: Install the VMware Application



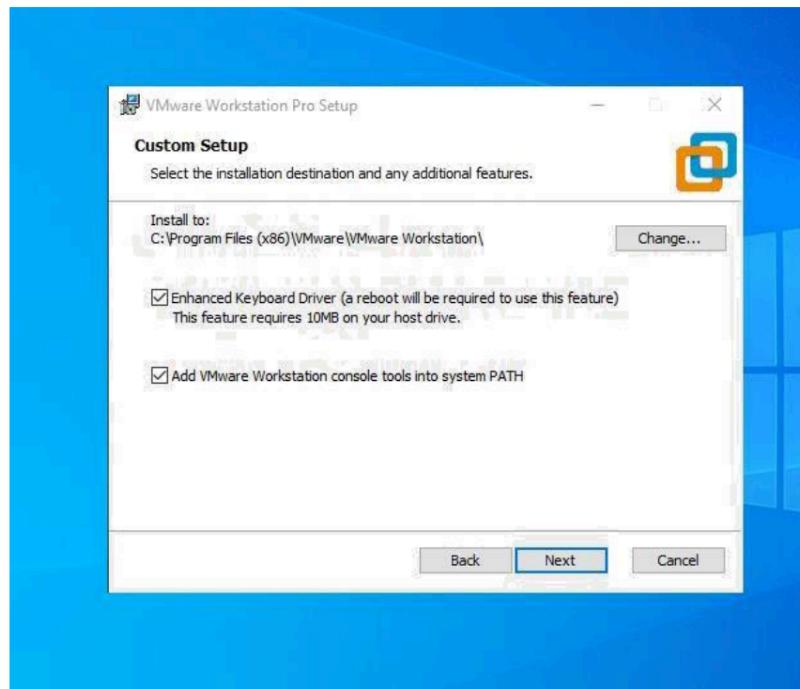
Step 3: Click Next



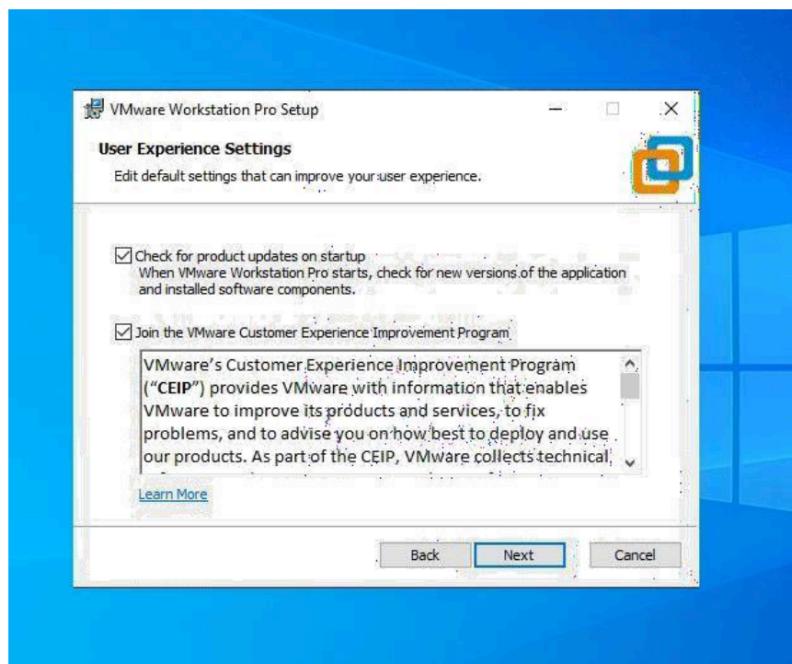
Step 4: Accept and Click Next



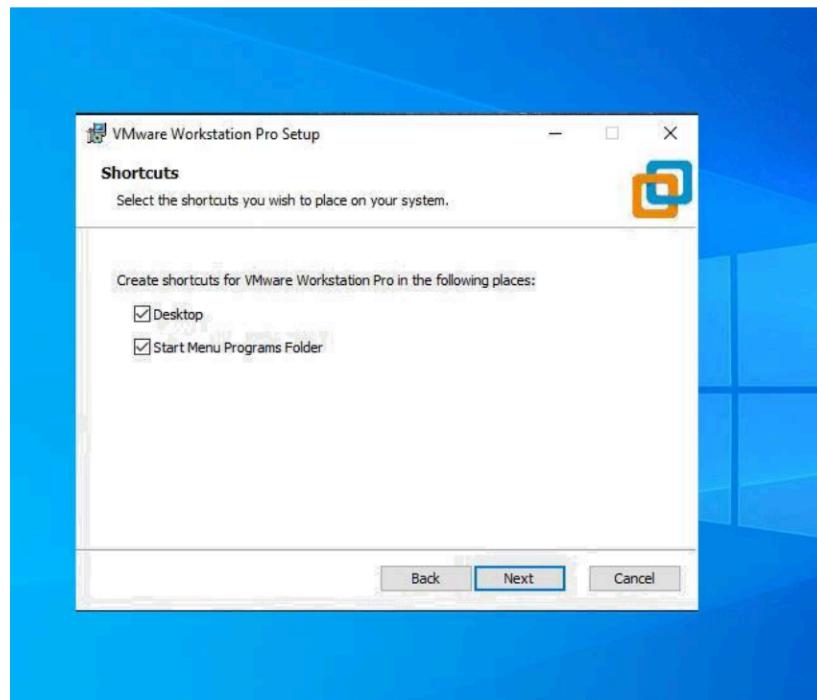
Step 5: Click Next



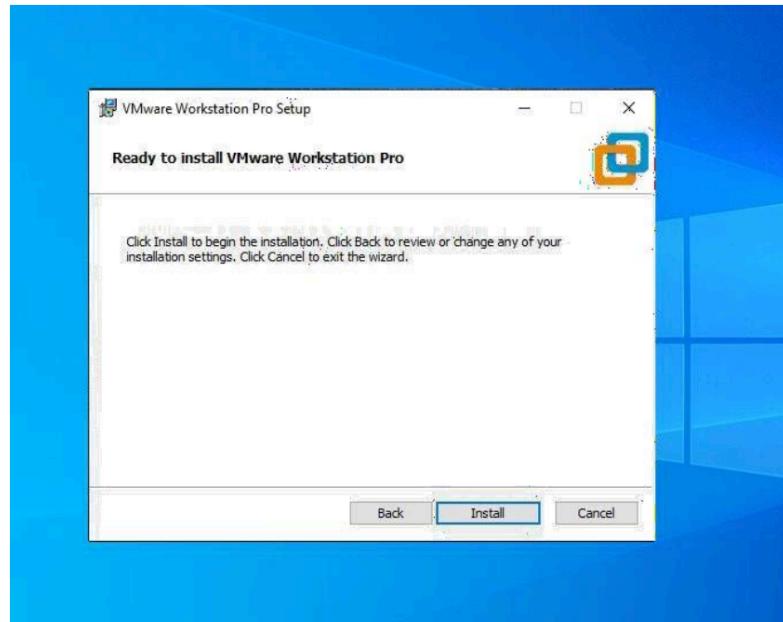
Step 6: Click Next



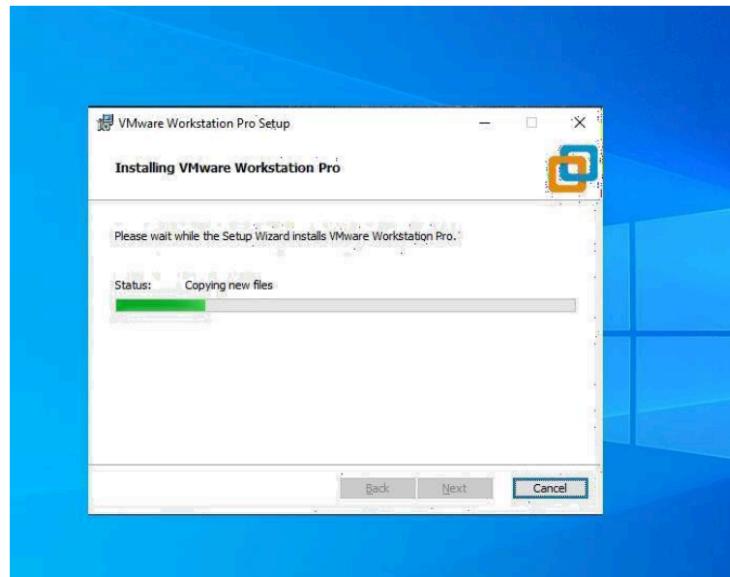
Step 7: Click Next



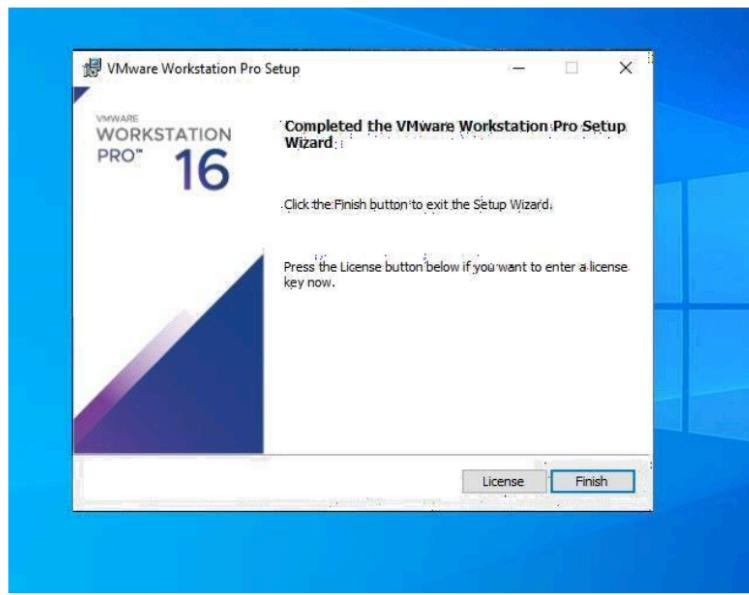
Step 8: Click Install



Step 9: Installing



Step 10: Click Finish



Downloading Ubuntu

Step 11: Download the Ubuntu OS



Ubuntu 20.04.1 LTS

Download the latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years, until April 2025, of free security and maintenance updates, guaranteed.

[Ubuntu 20.04 LTS release notes](#)

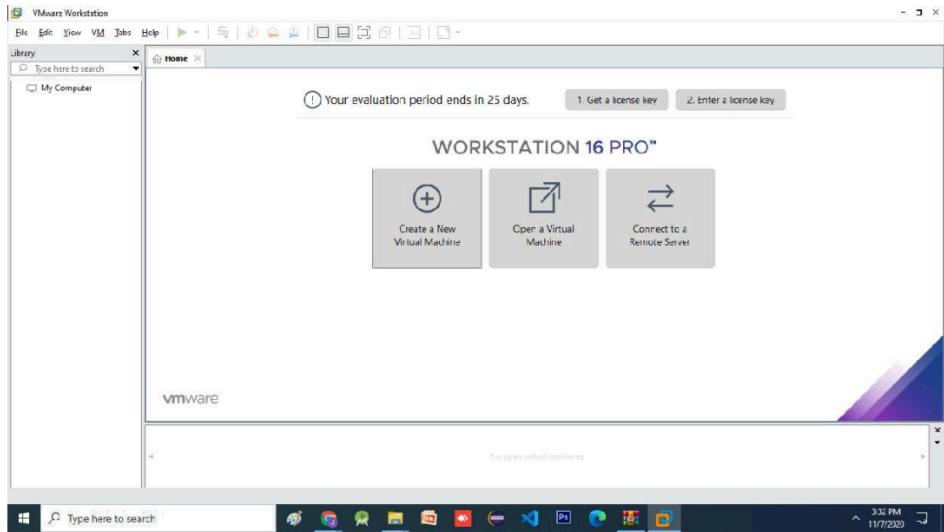
Recommended system requirements:

- 2 GHz dual core processor or better
- Internet access is helpful
- 4 GB system memory
- Either a DVD drive or a USB port for the installer media
- 25 GB of free hard drive space

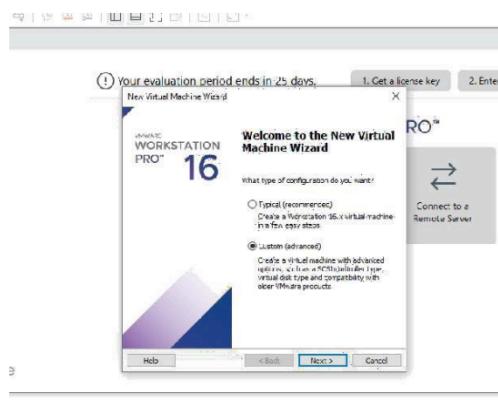
[Download](#)

For other versions of Ubuntu Desktop including torrents, the network installer, a list of legal mirrors, and beta releases see our [alternative downloads](#).

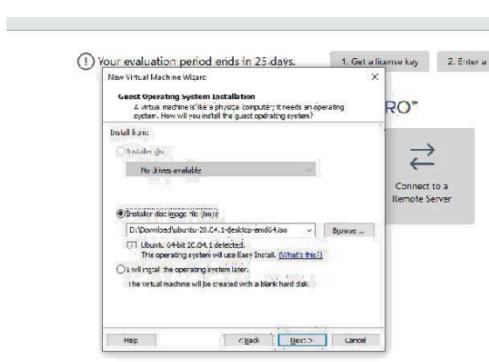
Step 12: Create new virtual machine



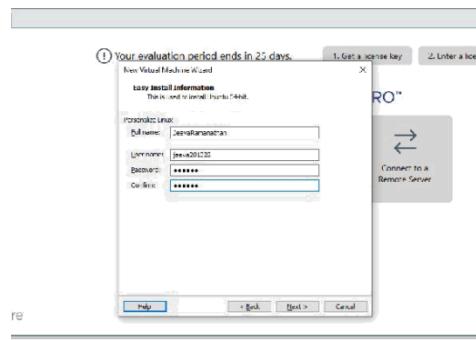
Step 13: Click Next



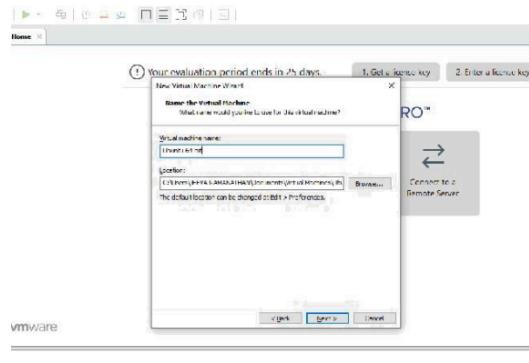
Step 14: Browse the downloaded Ubuntu file and click next



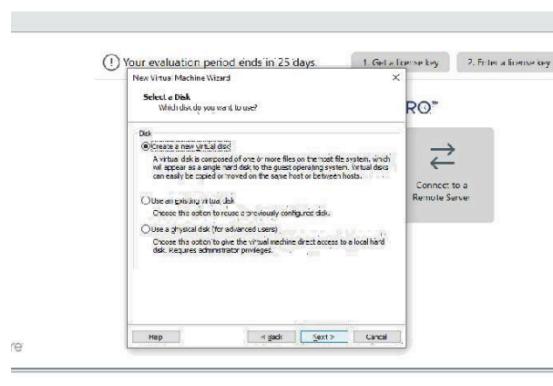
Step 15: Create an username and password and click next



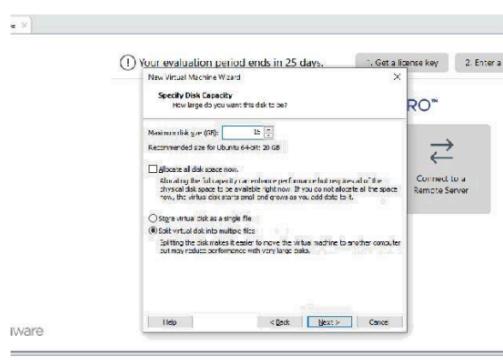
Step 16: Choose the location to use your virtual machine and click next



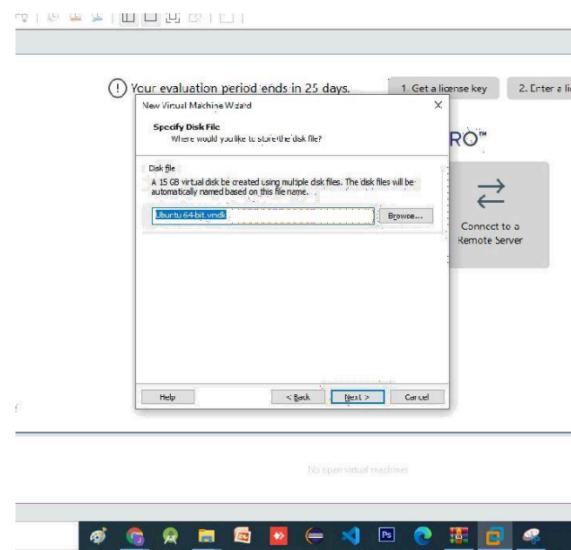
Step 17: Select create a new virtual disk and click next



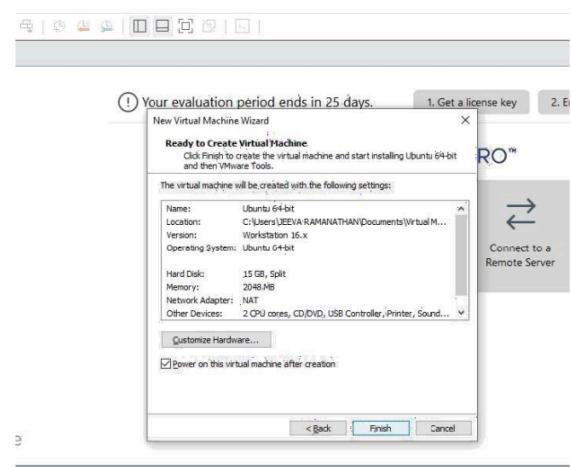
Step 18: Specify the disk size and click next



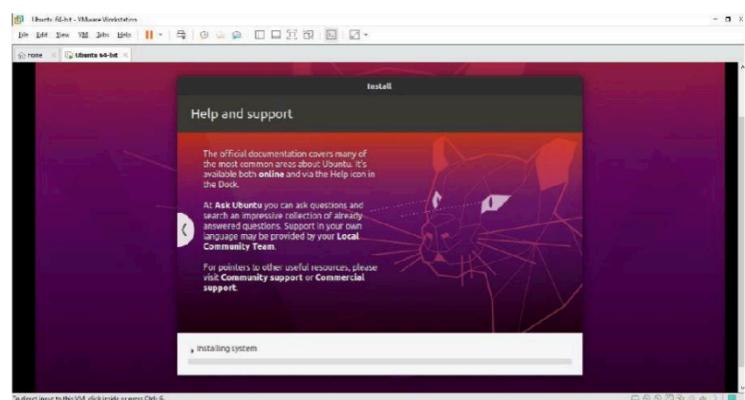
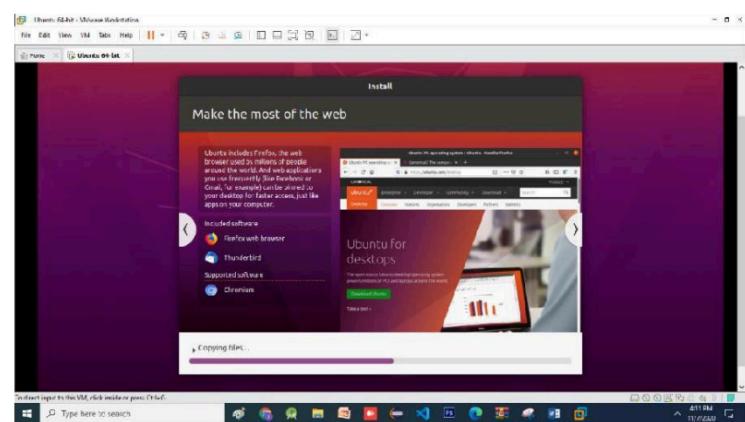
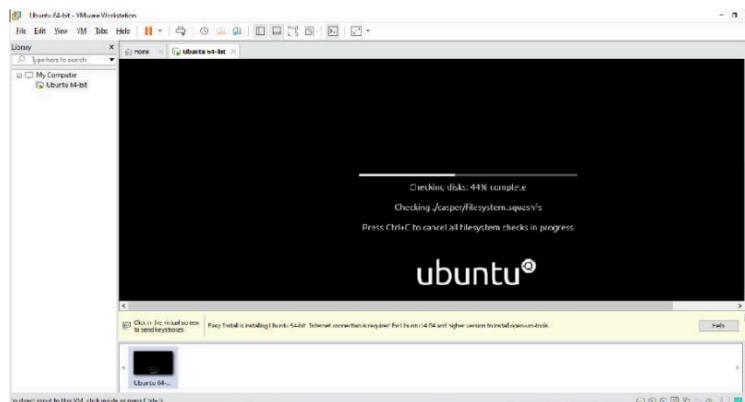
Step 19: Click Next



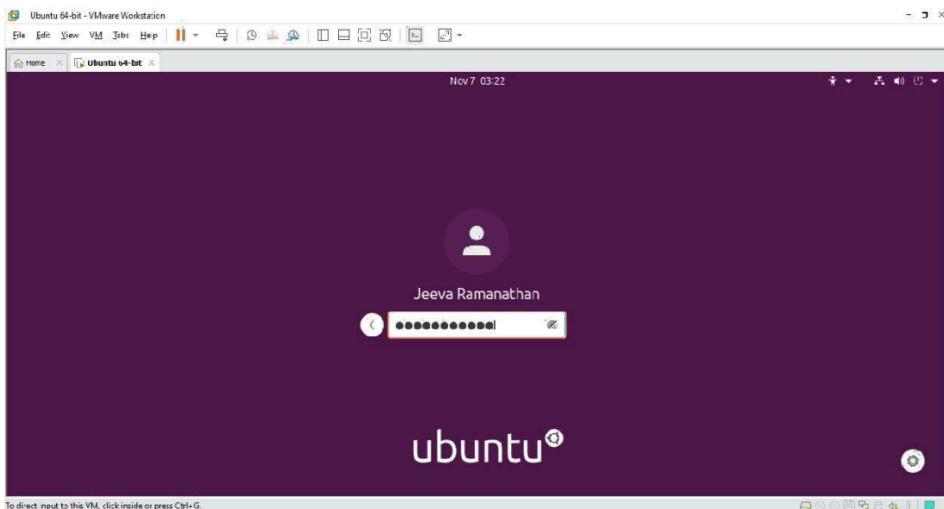
Step 20: Click Finish



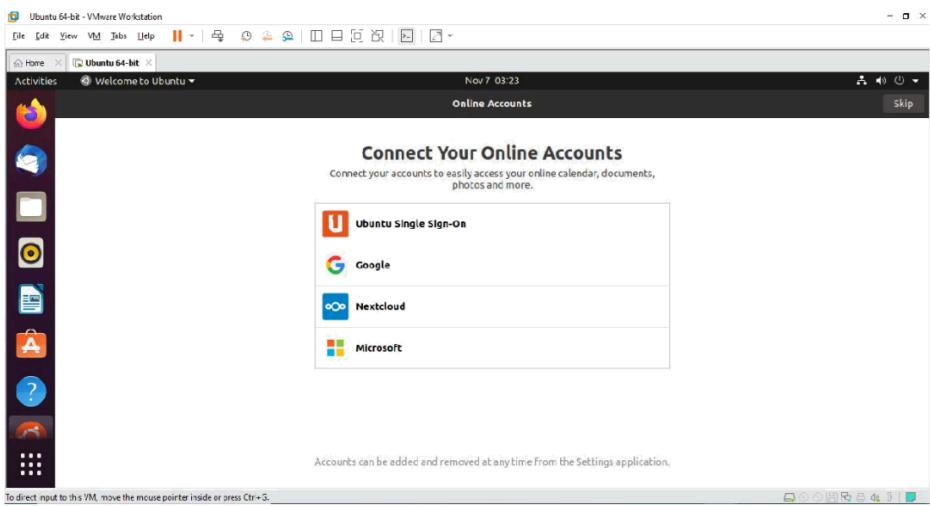
Step 21: Installing Ubuntu on VMWare and unzipping files



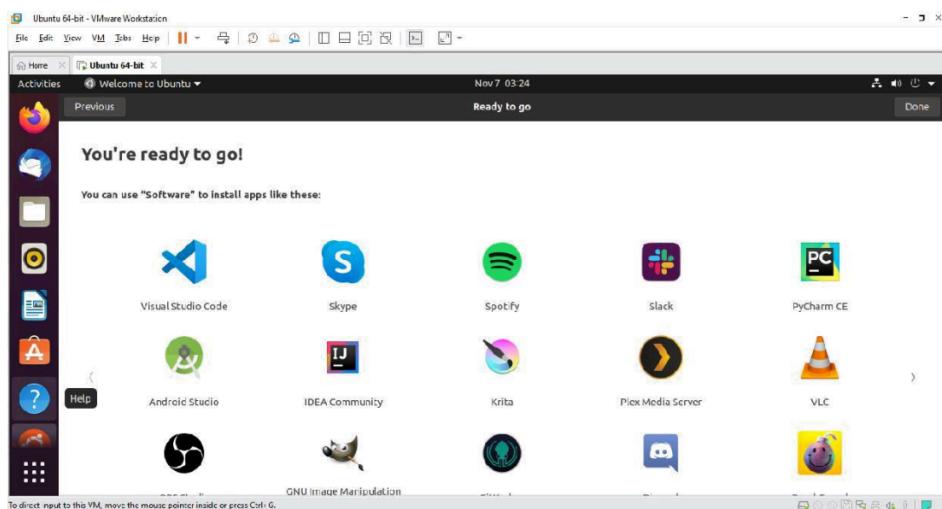
Step 22: Login to Ubuntu



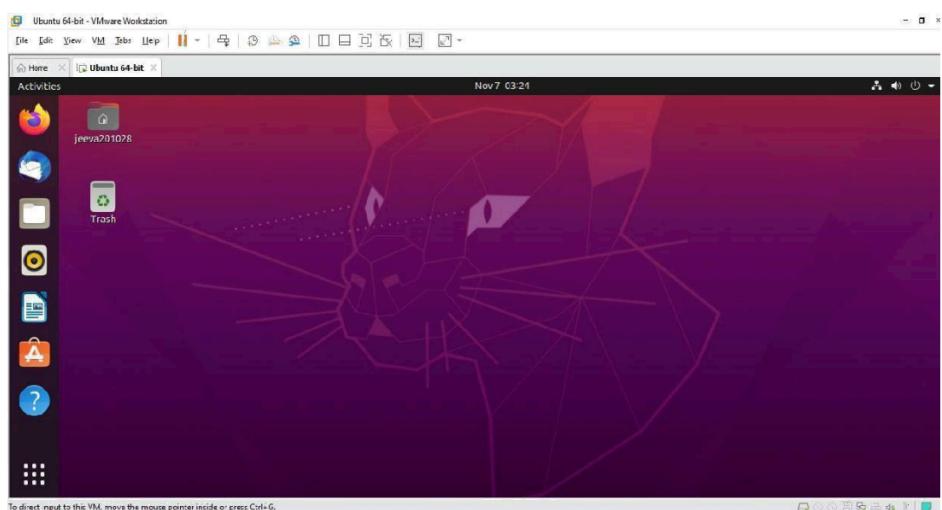
Step 23: Skip everything



Step 24: Click Done



Step 25: Thus we have installed VMware Workstation with different flavours of linux on top of windows



Result: Thus, VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8 or 10.has been successfully installed and executed.

Experiment 02

EX.No:5

Launch a web application using GAE launcher

Date:

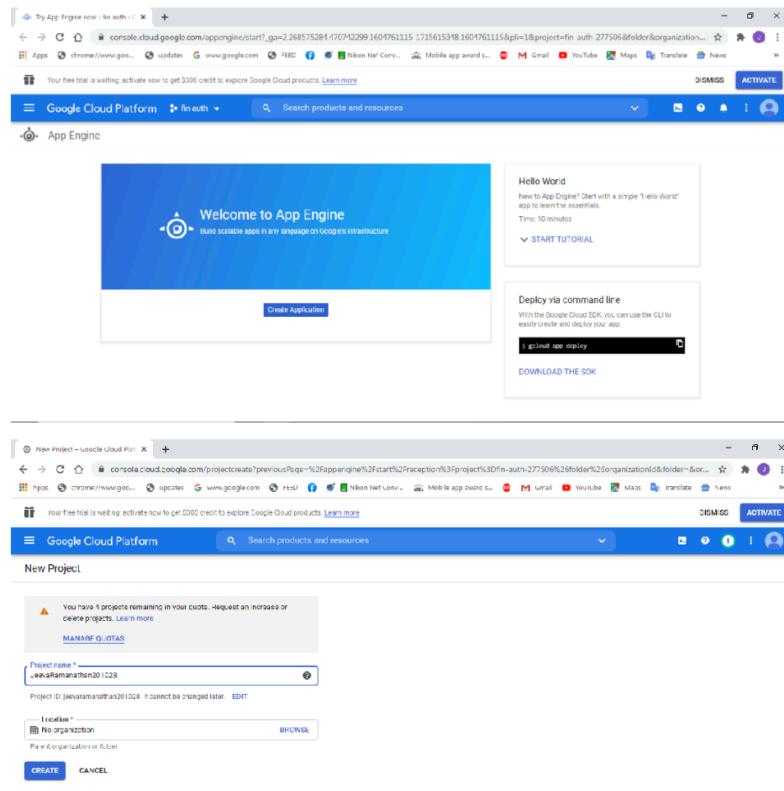
Aim:

To launch the web applications by using the GAE launcher.

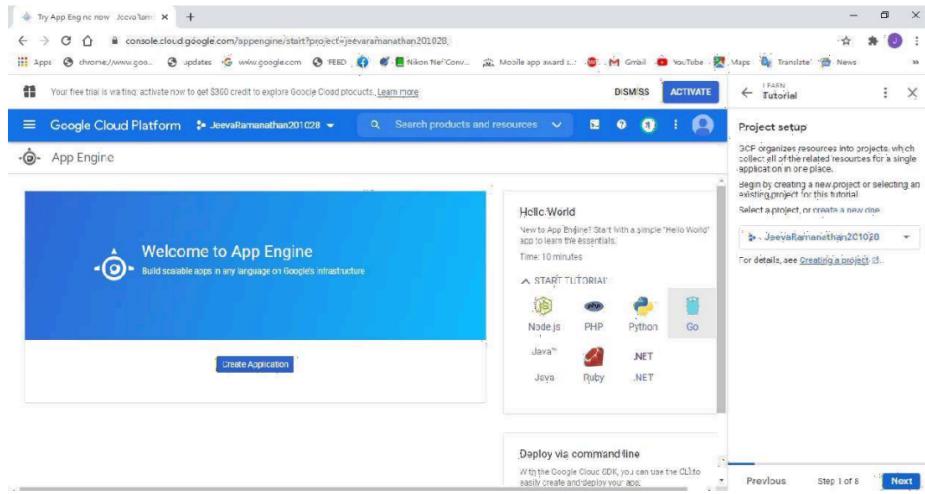
Procedure:

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.

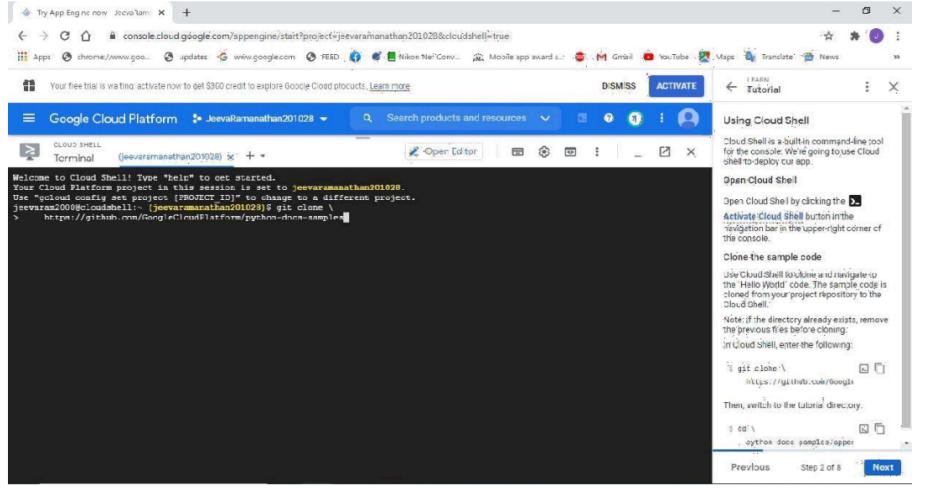


Step 2:Select python and click next.



Step 3:Open the cloud shell and follow the steps in the tutorial.

Clone the repository by using the given command



Step 4:Create the virtual environment

The screenshot shows a Google Cloud Platform Cloud Shell interface. On the left, a terminal window displays Python code for a 'Hello World' application. On the right, a sidebar provides instructions for testing the app on Cloud Shell, including commands to run the app and install dependencies.

```
# 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 one_python3_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=5000, debug=True)
# [END one_python3_app]
jeevaramanathan201028@jeevaramanathan201028:~/python-docs-samples/appengine/standard_python3/hello_world [jeevaramanathan201028]$ cat app.yaml
runtime: python38
jeevaramanathan201028@jeevaramanathan201028:~/python-docs-samples/appengine/standard_python3/hello_world [jeevaramanathan201028]# virtualenv --python python3
> ~/envs/hello_world

```

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/activate
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

Step 5:Activate your virtual environment.

The screenshot shows a Google Cloud Platform Cloud Shell interface. On the left, a terminal window displays the command to activate the virtual environment. On the right, a sidebar provides instructions for activating the virtual environment, including the command to run the app and install dependencies.

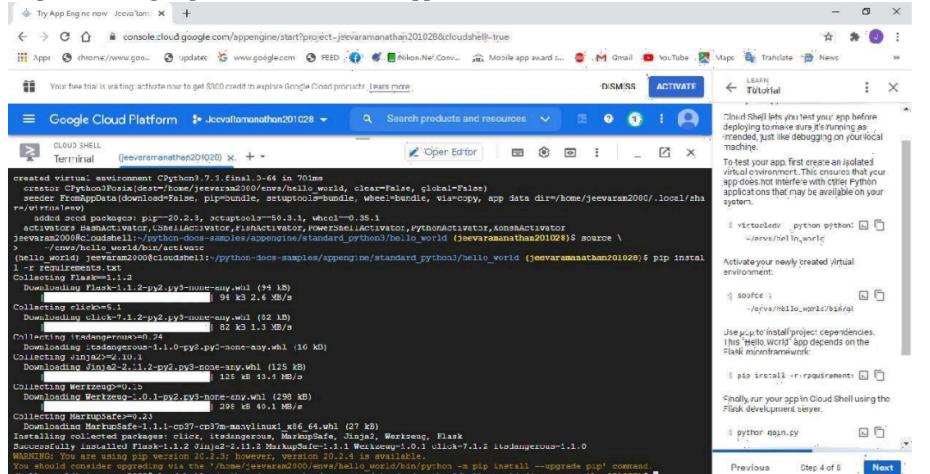
```
# If 'entrypoint' is not defined in app.yaml, App Engine will look for an app
# called 'app' in 'main.py'.
app = Flask(__name__)

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

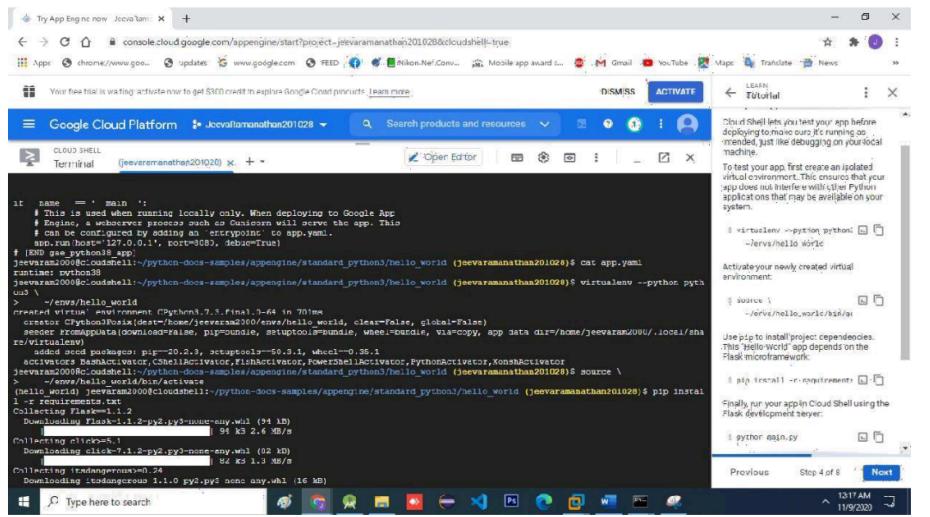
if __name__ == '__main__':
    # This is used when running locally only. When deploying to Google App
    # Engine, a webserver process such as Gunicorn will serve the app. This
    # can be configured by adding an 'entrypoint' to app.yaml.
    # app.run(host='127.0.0.1', port=5000, debug=True)
# [END one_python3_app]
jeevaramanathan201028@jeevaramanathan201028:~/python-docs-samples/appengine/standard_python3/hello_world [jeevaramanathan201028]$ cat app.yaml
jeevaramanathan201028@jeevaramanathan201028:~/python-docs-samples/appengine/standard_python3/hello_world [jeevaramanathan201028]# virtualenv --python python3
jeevaramanathan201028@jeevaramanathan201028:~/python-docs-samples/appengine/standard_python3/hello_world [jeevaramanathan201028]# source /envs/hello_world/bin/activate
(jeevaramanathan201028) jeevaramanathan201028:~/python-docs-samples/appengine/standard_python3/hello_world [jeevaramanathan201028]#
```

Code copied to clipboard

Step 6: Installing requirements and run the app.



```
created virtual environment CPython 3.6.4 final 0-64 in 701ms
creator: CPythonForUnix(dest='/home/jeevaramanathan201028/.envs/hello_world', clear=False, global=False)
seeder: FromPipData(downloaded=False, pip='bundle', setuptools='bundle', wheel='bundle', via='copy', app data dir='/home/jeevaramanathan201028/.local/share/virtualenvs')
added seed packages: pip==20.2.3, setuptools==56.3.1, wheel==0.35.1
activators: HamachiActivator, SshuttleActivator, FishActivator, PowershellActivator, PythonActivator, KonsoleActivator
jeevaramanathan201028@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ source venv/bin/activate
(jeevaramanathan201028) jeevaramanathan201028@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ pip install -r requirements.txt
Collecting Flask==1.1.2
  Downloading Flask-1.1.2-py2.py3-none-any.whl (44 kB)
Collecting click==5.1
  Downloading click_5.1-py2.py3-none-any.whl (52 kB)
Collecting itsdangerous==2.0.1
  Downloading itsdangerous-2.0.1-py2.py3-none-any.whl (16 kB)
Collecting Jinja2==2.11.2-py2.py3-none-any.whl (125 kB)
  Downloading Jinja2-2.11.2-py2.py3-none-any.whl (125 kB)
Collecting Werkzeug==0.13.0
  Downloading Werkzeug-0.13.0-py2.py3-none-any.whl (40.1 kB)
Collecting MarkupSafe==0.23
  Downloading MarkupSafe-0.23-py37-cp37m-manylinux_x86_64.whl (27 kB)
Installing collected packages: click, itsdangerous, MarkupSafe, Jinja2, Werkzeug, Flask
Successfully installed click==5.1 Jinja2==2.11.2 MarkupSafe==0.23 Werkzeug==0.13.0
WARNING: You are using pip version 20.2.3; however, version 20.2.4 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(jeevaramanathan201028) jeevaramanathan201028@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$
```



```
if name == '__main__':
    # This is used when running locally only. When deploying to Google App
    # Engine, Google App Engine will copy this file to /app/main.py
    # and set __name__ = '__main__'. It is recommended that you delete this
    # code for local development, as it will be copied by existing an entrypoint to app.yaml.
    app.run(host='127.0.0.1', port=8080, debug=True)

# END get_python3()
jeevaramanathan201028@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ cat app.yaml
runtime: python38
jeevaramanathan201028@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ virtualenv --python python > venv/hello_world
created virtual environment CPython 3.7.5 final 0-4 in 701ms
creator: CPythonForUnix(dest='/home/jeevaramanathan201028/.envs/hello_world', clear=False, global=False)
seeder: FromPipData(downloaded=False, pip='bundle', setuptools='bundle', wheel='bundle', via='copy', app data dir='/home/jeevaramanathan201028/.local/share/virtualenvs')
added seed packages: pip==20.2.3, setuptools==56.3.1, wheel==0.35.1
activators: HamachiActivator, SshuttleActivator, FishActivator, PowershellActivator, PythonActivator, KonsoleActivator
jeevaramanathan201028@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ source venv/bin/activate
(jeevaramanathan201028) jeevaramanathan201028@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$ pip install -r requirements.txt
Collecting Flask==1.1.2
  Downloading Flask-1.1.2-py2.py3-none-any.whl (44 kB)
Collecting click==5.1
  Downloading click_5.1-py2.py3-none-any.whl (52 kB)
Collecting itsdangerous==2.0.1
  Downloading itsdangerous-2.0.1-py2.py3-none-any.whl (16 kB)
(jeevaramanathan201028) jeevaramanathan201028@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (jeevaramanathan201028)$
```

Step 7:Create an application and deploy it in cloud shell.

```

Try App Engine now - JeevaRam
console.cloud.google.com/appengine/start?project=jeevaramanathan201028&clcloudshell=true

Your free trial is waiting activate now to get $300 credit to explore Google Cloud products. Learn more
DISMISS ACTIVATE
LEARN TUTORIAL

Google Cloud Platform Jeevaramanathan201028 Search products and resources
Terminal Jeevaramanathan201028 Open Editor
CLOUD SHELL
Collecting click<0.5
  Downloading click-0.5.1-py2.py3-none-any.whl (54 kB)
Collecting Jinja2<2.10.1
  Downloading Jinja2-2.10.1-py2.py3-none-any.whl (82 kB)
Collecting MarkupSafe<0.24
  Downloading itsdangerous-1.0.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2>2.10.1
  Downloading Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.2 click-7.1.2 itsdangerous-1.1.0
Collecting Werkzeug<0.15
  Downloading Werkzeug-0.15.0-py2.py3-none-any.whl (100 kB)
Collecting MarkupSafe<0.21
  Downloading MarkupSafe-0.21.0-py2.py3-none-anylinux_x86_64.whl (27 kB)
Successfully installed Flask-1.1.1-c037-c97e-maylinux_x86_64.whl (27 kB)
itsdangerous-1.0.0-py2.py3-none-any.whl (16 kB)
MarkupSafe-0.21.0-py2.py3-none-anylinux_x86_64.whl (27 kB)
Werkzeug-0.15.0-py2.py3-none-any.whl (100 kB)
You should consider upgrading via the '/home/jeevaramanathan201028/bin/python -m pip install --upgrade pip' command.
(hello world) jeevaramanathan201028$ python main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
* Debug mode: on
  Use a production WSGI server instead.
* Running on http://127.0.0.1:8080/ (Press CTRL+C to quit)
* Restarting with stat
  Debugger PIN: 123-976-332
gcloud app create
gcloud app deploy app.yaml \
--project jeevaramanathan201028

```

Deploying to App Engine

Create an application

In order to deploy your app, you need to create an app.yaml file:

gcloud app deploy app.yaml \ --project jeevaramanathan201028

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 jeevaramanathan201028

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 app's URL.

Previous Step 5 of 8 Next

Step 8:Click preview on port 8080 to see your deployed application

```

Try App Engine now - JeevaRam
console.cloud.google.com/appengine/start?project=jeevaramanathan201028

Your free trial is waiting activate now to get $300 credit to explore Google Cloud products. Learn more
DISMISS ACTIVATE
LEARN TUTORIAL

Google Cloud Platform Jeevaramanathan201028 Search products and resources
Terminal Jeevaramanathan201028 Open Editor
CLOUD SHELL
Collecting itsdangerous<0.24
  Downloading itsdangerous-1.0.0-py2.py3-none-any.whl (16 kB)
Collecting Jinja2<2.10.1
  Downloading Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.2 click-7.1.2 itsdangerous-1.1.0
Collecting Werkzeug<0.15
  Downloading Werkzeug-0.15.0-py2.py3-none-any.whl (100 kB)
Collecting MarkupSafe<0.21
  Downloading MarkupSafe-0.21.0-py2.py3-none-anylinux_x86_64.whl (27 kB)
Successfully installed Flask-1.1.1-c037-c97e-maylinux_x86_64.whl (27 kB)
itsdangerous-1.0.0-py2.py3-none-any.whl (16 kB)
MarkupSafe-0.21.0-py2.py3-none-anylinux_x86_64.whl (27 kB)
Werkzeug-0.15.0-py2.py3-none-any.whl (100 kB)
You should consider upgrading via the '/home/jeevaramanathan201028/bin/python -m pip install --upgrade pip' command.
(hello world) jeevaramanathan201028$ python main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
* Debug mode: on
  Use a production WSGI server instead.
* Running on http://127.0.0.1:8080/ (Press CTRL+C to quit)
* Restarting with stat
  Debugger PIN: 123-976-332
gcloud app create
gcloud app deploy app.yaml \
--project jeevaramanathan201028

```

View your app's status

You can check 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.

App Engine

Previous Step 6 of 8 Next

Step 9:Finally the application is deployed and the output is seen.

https://8380.cs.530117580568-default.usouth1.cloudshell.dev/?authuser=0

Hello World!

Result:

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