

EX.NO: 1

DATE:

INSTALL VIRTUALBOX WITH LINUX OS ON TOP OF WINDOWS

AIM:

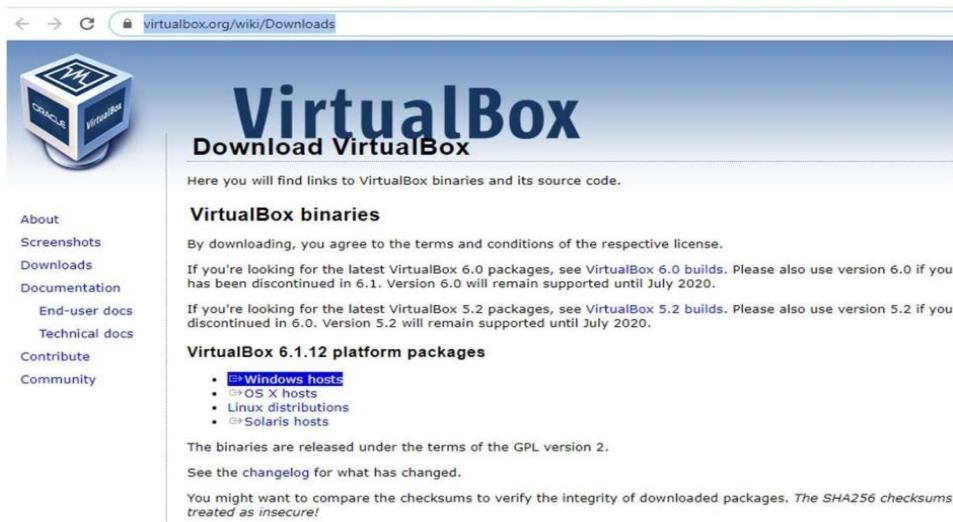
To install Virtualbox with Ubuntu OS on top of windows host operating system.

PROCEDURE:

Steps to install VirtualBox:

1. Download VirtualBox installer for windows.
2. The installer can be downloaded from the link

<https://www.virtualbox.org/wiki/Downloads>



3. Click “Windows host” to download the binary version for windows host.

4. The installer file downloaded will have the file name format like

VirtualBox-VersionNumber-BuildNumber-Win.exe.

Example: VirtualBox-6.1.12-139181-Win.exe.

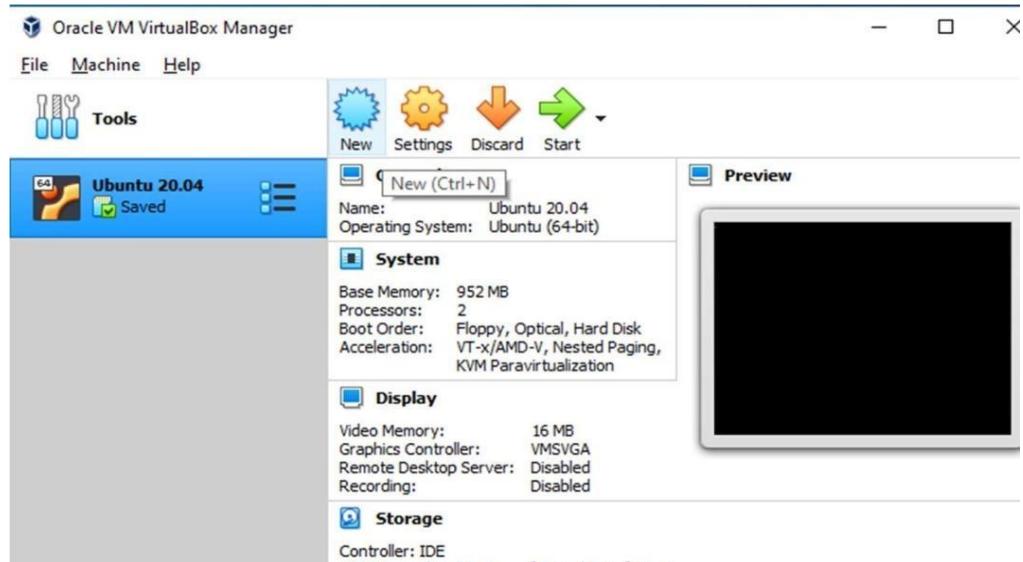
5. Double click on the installer to launch the setup Wizard. Click on Next to continue.



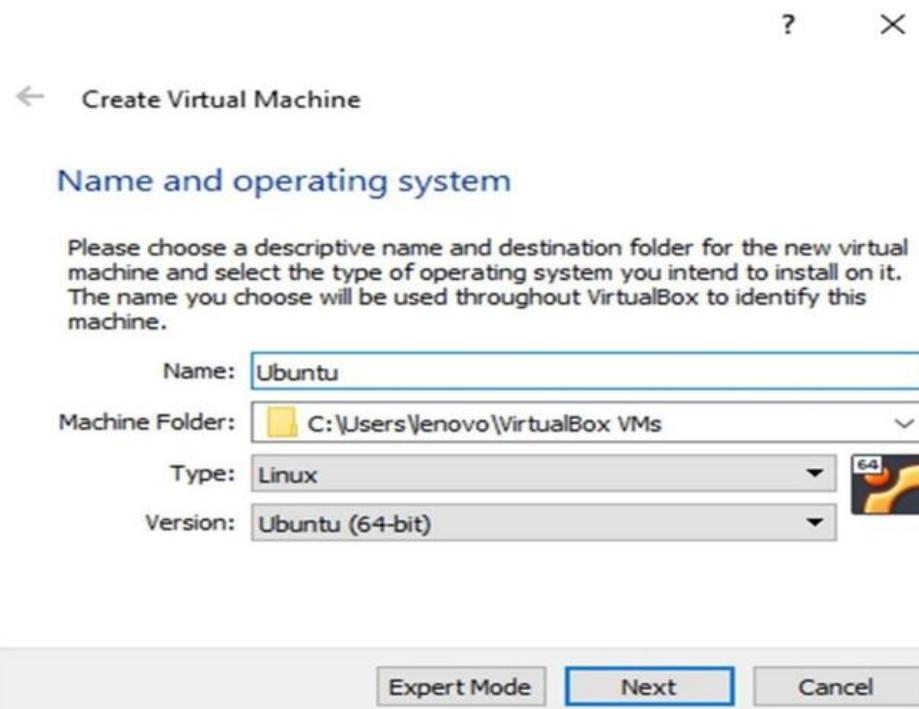
6. Custom setup dialog box will be opened. Accept the default settings and click next.
7. Select the way you want the features to be installed. You can accept the default and clicknext.
8. A dialog box opens with Network Interfaces warning. Click Yes to proceed.
9. Click install to begin the installation process.
10. When prompted with a message to install (Trust) Oracle Universal Serial Bus, click Install to continue.
11. After the installation completes, click finish to exit the setup wizard.
12. Launch the Oracle VM VirtualBox.

Steps to create a virtual machine [Ubuntu] in VirtualBox:

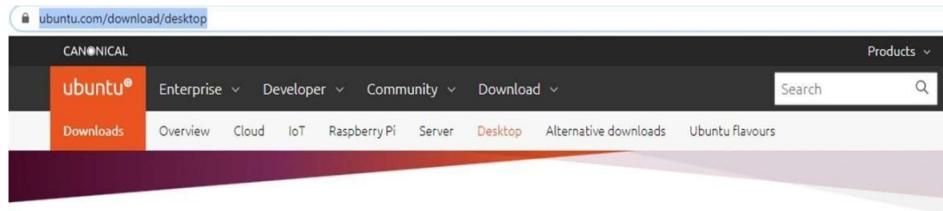
1. Open the Oracle VM VirtualBox.
2. Click New icon or „Ctrl + N“ to create a new virtual machine.



3. Enter a name for the new virtual machine. Choose the Type and Version. Note that VirtualBox automatically changes 'Type' to Linux and 'Version' to 'Ubuntu (64 bit)' if the name is given as „Ubuntu“. Click Next.



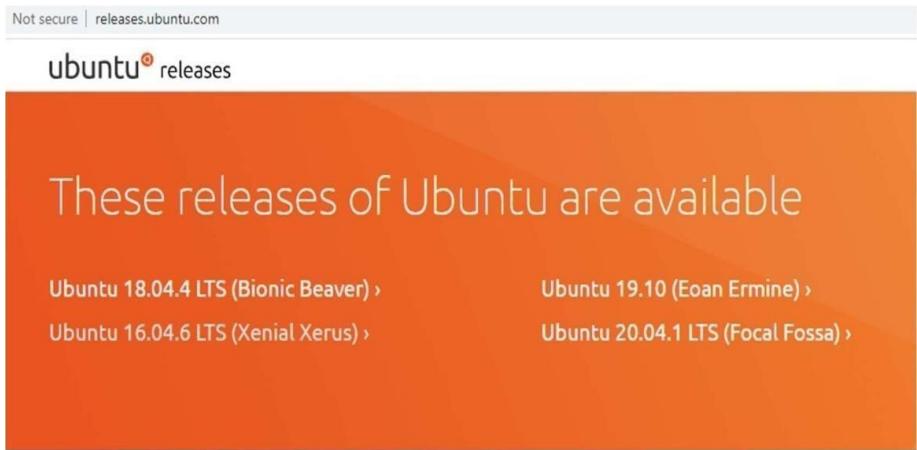
4. Select the amount of RAM to use. The ideal amount of RAM will automatically be selected. Do not increase the RAM into the red section of the slider; keep the slider in the green section.
5. Accept the default 'Create a virtual hard drive now' and click 'Create' button.
6. Choose the hard disk file type as VDI (VirtualBox Disk Image). Click Next.
7. Click Next to accept the default option „Dynamically allocated“ for storage on physical harddrive.
8. Select the size of the virtual hard disk and click create.
9. The newly created virtual machine will be displayed in the dashboard.
10. Download the ISO file [Ubuntu disk image file]. Latest version of Ubuntu iso file can be downloaded from the link <https://ubuntu.com/download/desktop>. Click Download button.



Download Ubuntu Desktop

A screenshot of the Ubuntu 20.04.1 LTS download page. The page title is "Ubuntu 20.04.1 LTS". It includes a brief description of what LTS means: "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." Below this is a link to "Ubuntu 20.04 LTS release notes". On the right side, there is a large green "Download" button and a note: "For other versions of Ubuntu Desktop including torrents, the network installer, a list of local mirrors, and more, visit [Ubuntu Desktop](#)".

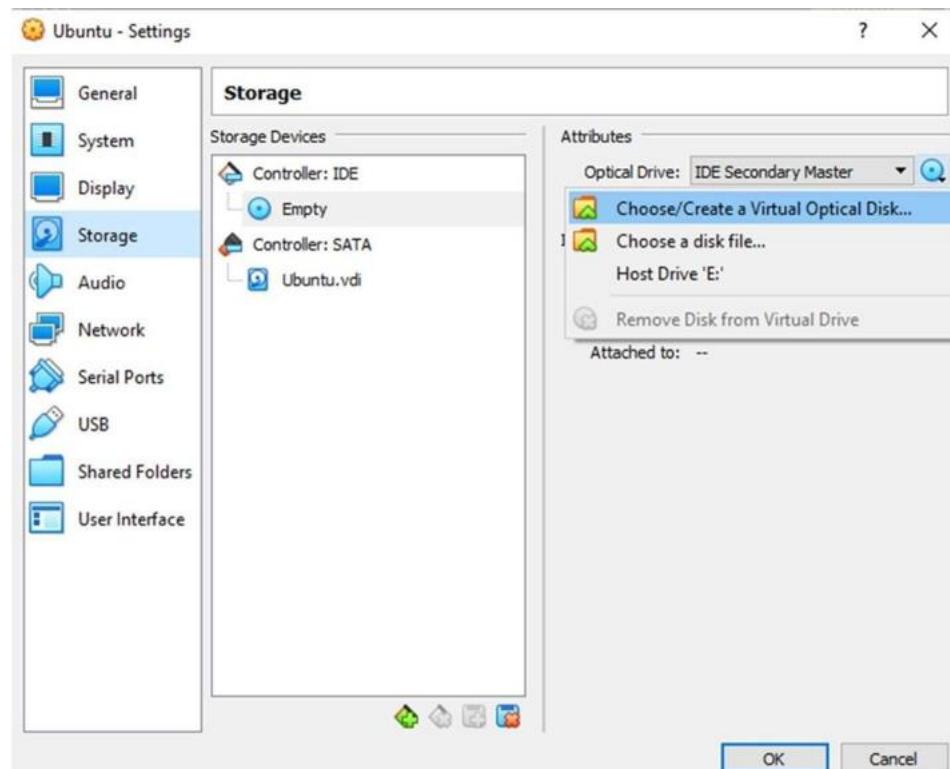
11. For previous versions, goto <http://releases.ubuntu.com>. Choose the preferred version of Ubuntu and download the iso file.



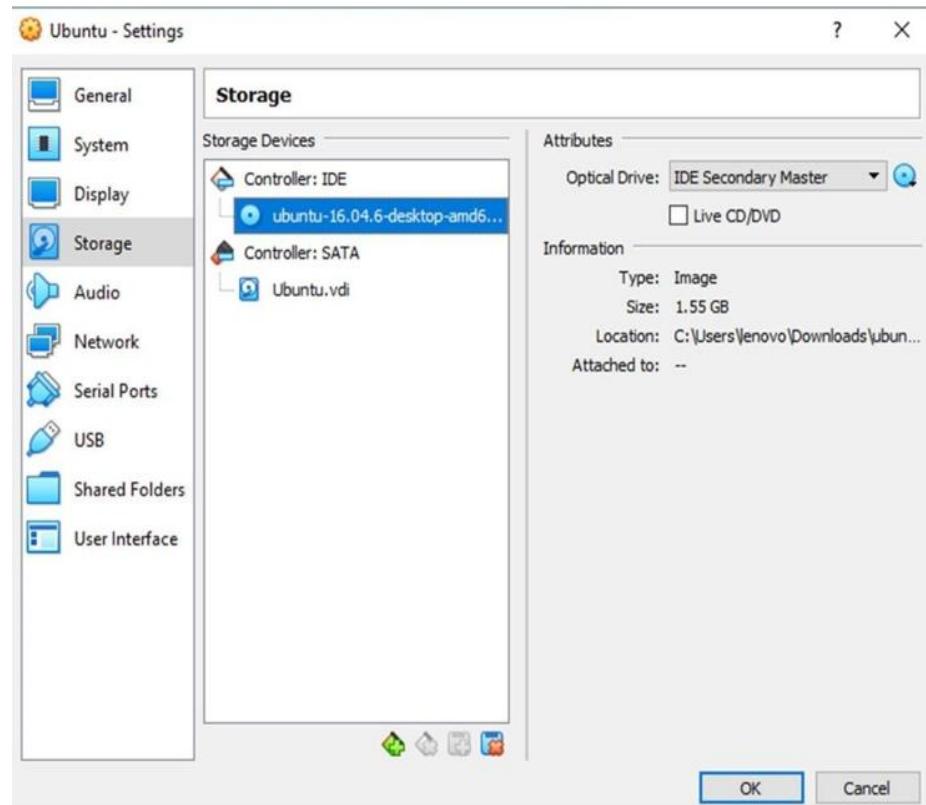
These older Ubuntu releases are now in Extended Maintenance (ESM):

- [Ubuntu 14.04.6 LTS \(Trusty Tahr\) >](#)
- [Ubuntu 12.04.5 LTS \(Precise Pangolin\) >](#)

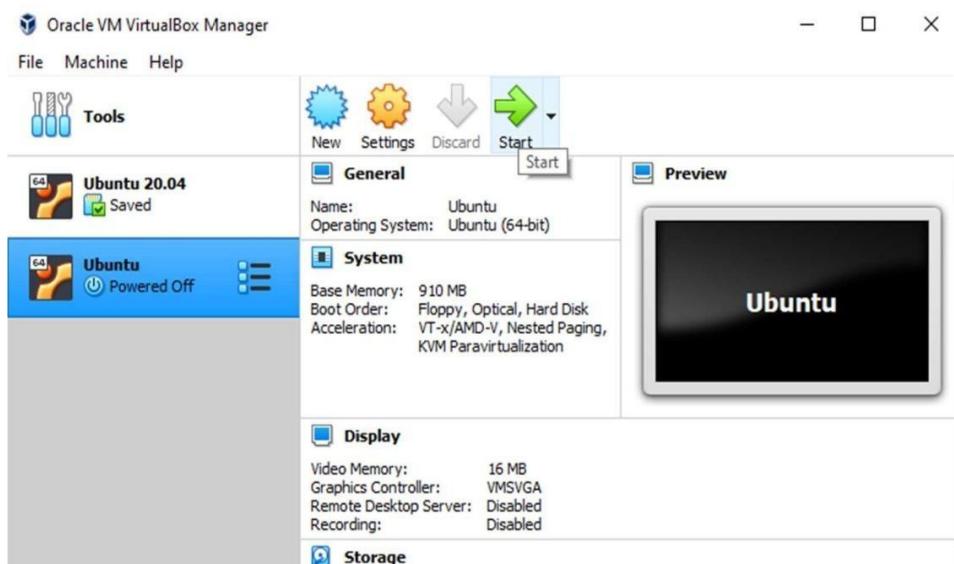
12. To setup the Ubuntu disk image file (iso file) goto settings.
13. Click Storage. Under „Storage Devices“ section click „Empty“.
14. In Attributes section, click the disk image and then "Choose Virtual Optical Disk File".



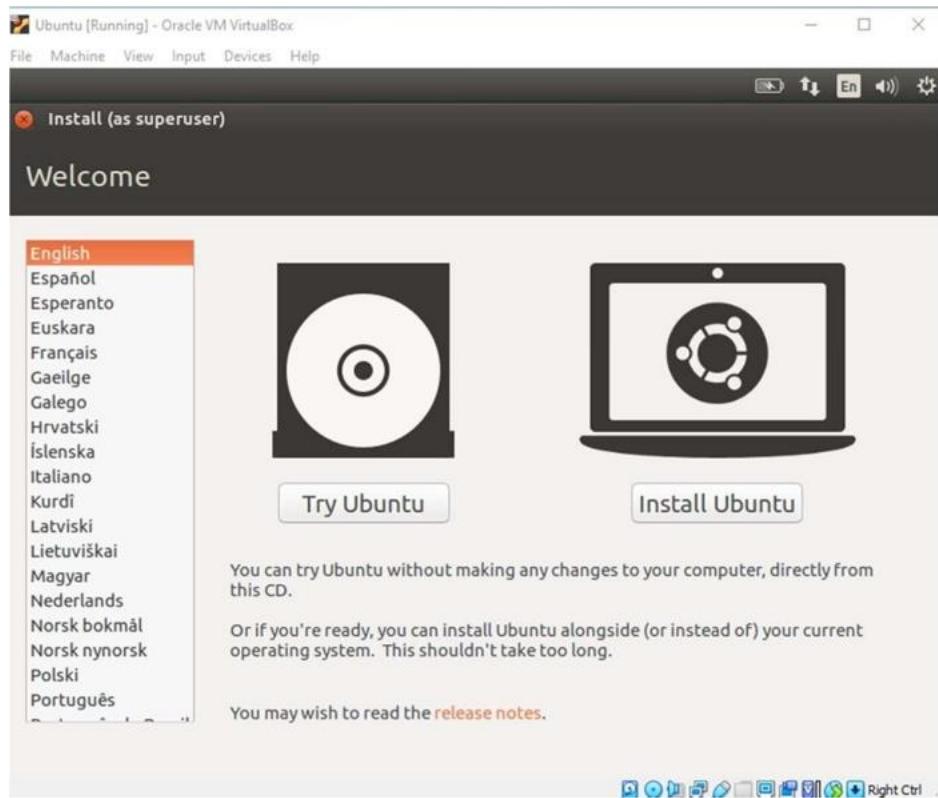
15. Browse and select the downloaded iso file. Click ok.



16. Select the newly created virtual machine in the dashboard and click start button.



17. In the welcome screen, click „Install Ubuntu“ button.



18. Click 'Continue' button.

19. Make sure 'Erase disk and install Ubuntu' option is selected and click 'Install Now' button.

20. Choose the default and click continue.

21. Setup up your profile by creating username and password.

22. After installation is complete, click 'Restart Now' button and follow the instructions.

23. The Ubuntu OS is ready to use. Login with the username and password.

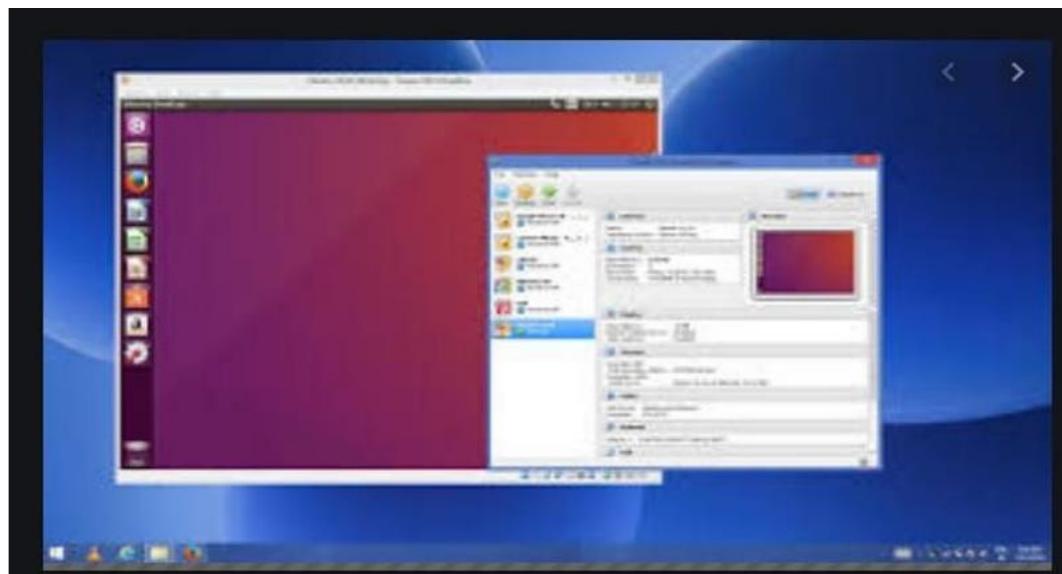
OUTPUT 1:

Virtualbox on top of windows.



OUTPUT 2:

Installation of Virtual box with **Linux OS (Guest OS/VM)** on top of windows Host.



RESULT:

The virutalbox installation is completed and the Virtual machine is created on top of windows host operating system

EX.NO: 2

DATE:

INSTALL A C COMPILER IN THE VIRTUAL MACHINE

AIM:

To install a C compiler in the virtual machine and execute a sample program.

PROCEDURE:

1. Launch the virtual box and open the virtual machine (Ubuntu).
2. Run the following command in the virtual machine terminal.

\$ sudo apt update

\$ sudo apt install gcc

It will install all the necessary packages for gcc complier.

3. Type the C program in the text editor and save the file with .c extension.

//demo.c

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    printf("Hello World");
```

```
    return
```

```
    0;
```

```
}
```

4. Compile and Run the C Program.

cc demo.c

./a.out

OUTPUT:

Installation of a C compiler in the virtual machine and executing a sample program.



RESULT:

Thus a C compiler is installed on a Ubuntu Virtual Machine on top of Windows Host and Executed a C program on a virtual machine.

EX.NO:3

DATE:

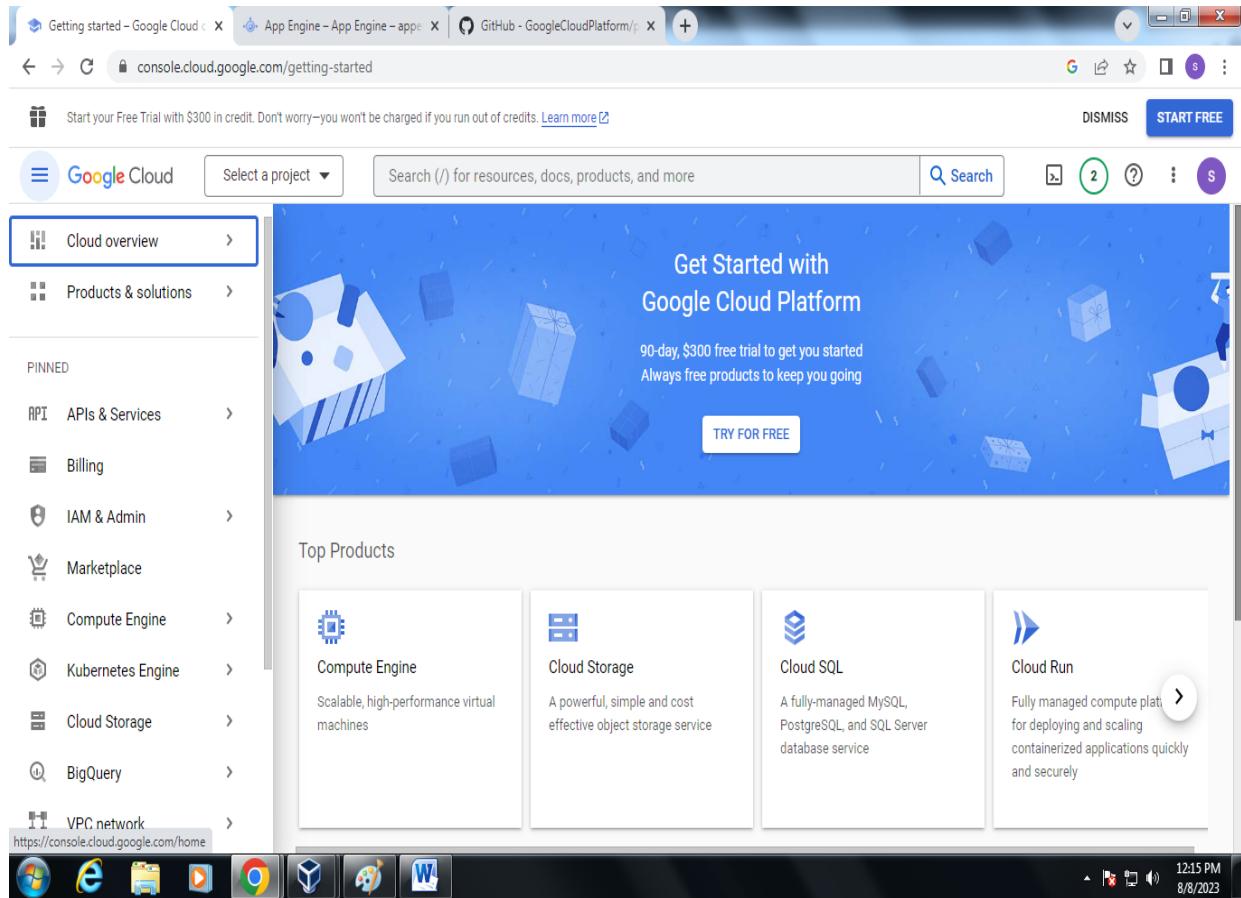
INSTALL GOOGLE APP ENGINE CREATE HELLO WORLD APP AND OTHER SIMPLE WEB APPLICATION USING PYTHON/JAVA.

AIM:

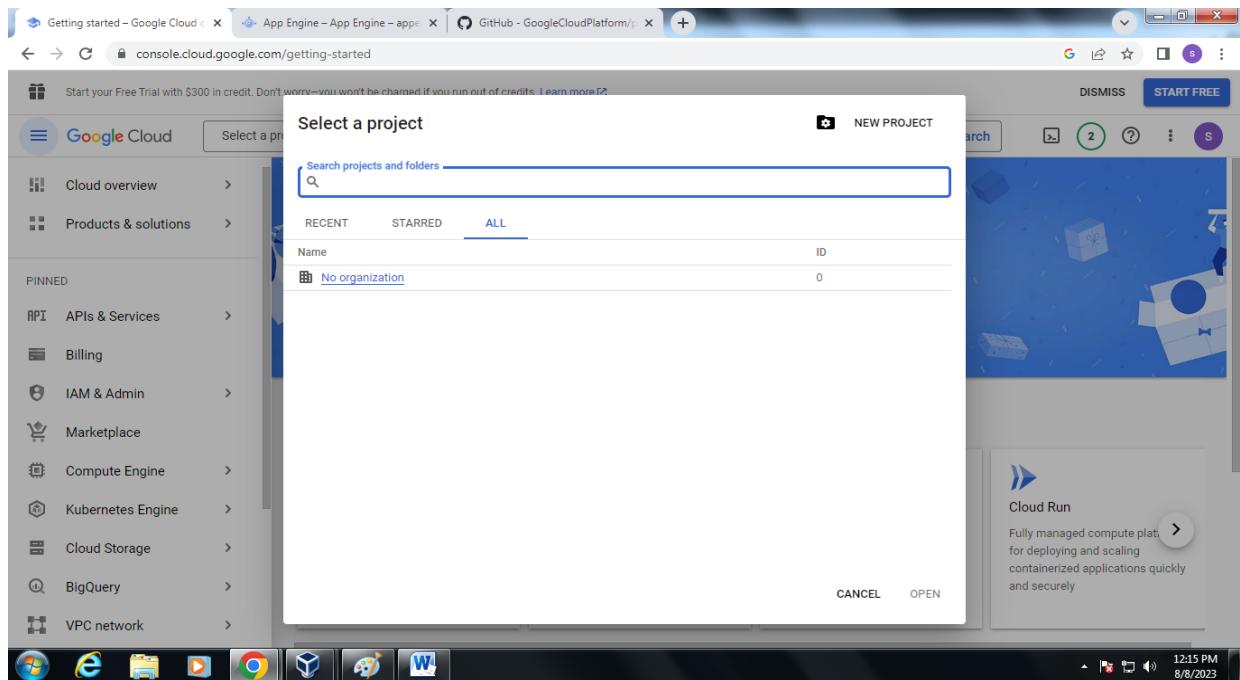
To run the python sample program to display Helloworld in Google App Engine in google cloud console.

PROCEDURE:

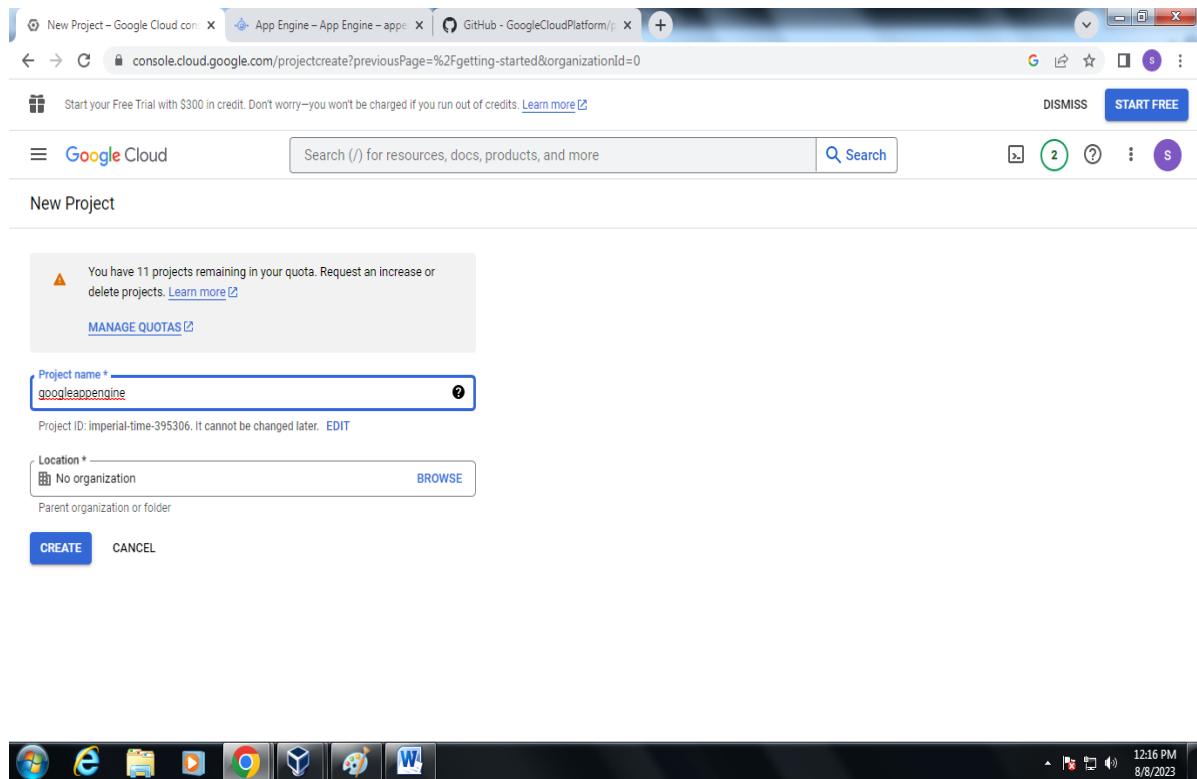
- 1.Go to the link in chrome “console.cloud.google.com”.
- 2.Select the project.



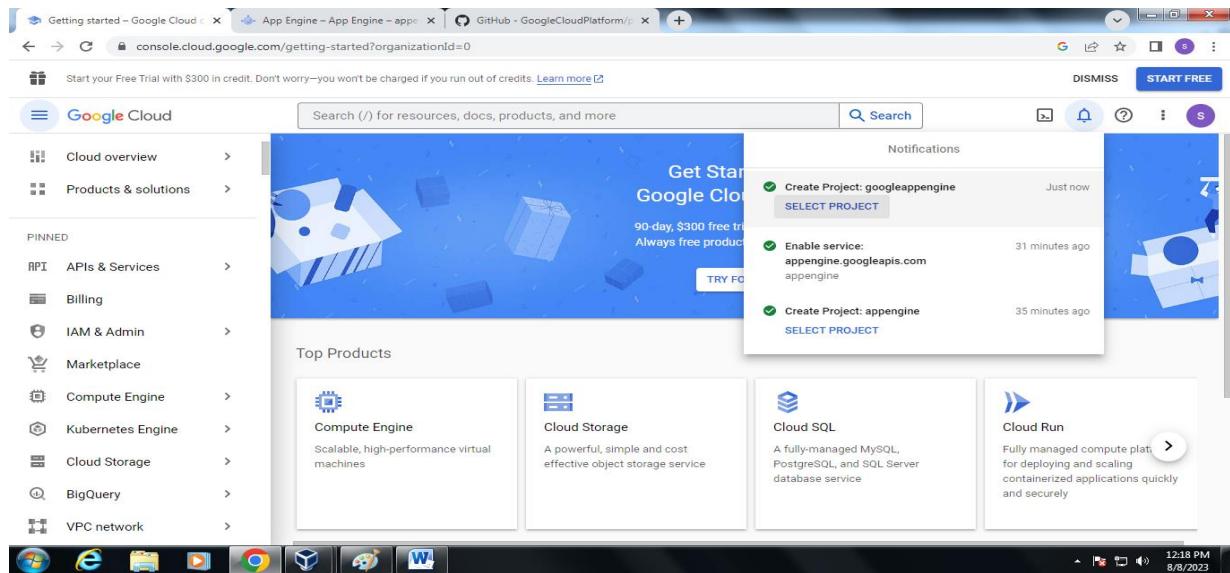
3. Select new project.



4. Give project name with a string of characters that uniquely identifies your project. For example, google appengine.

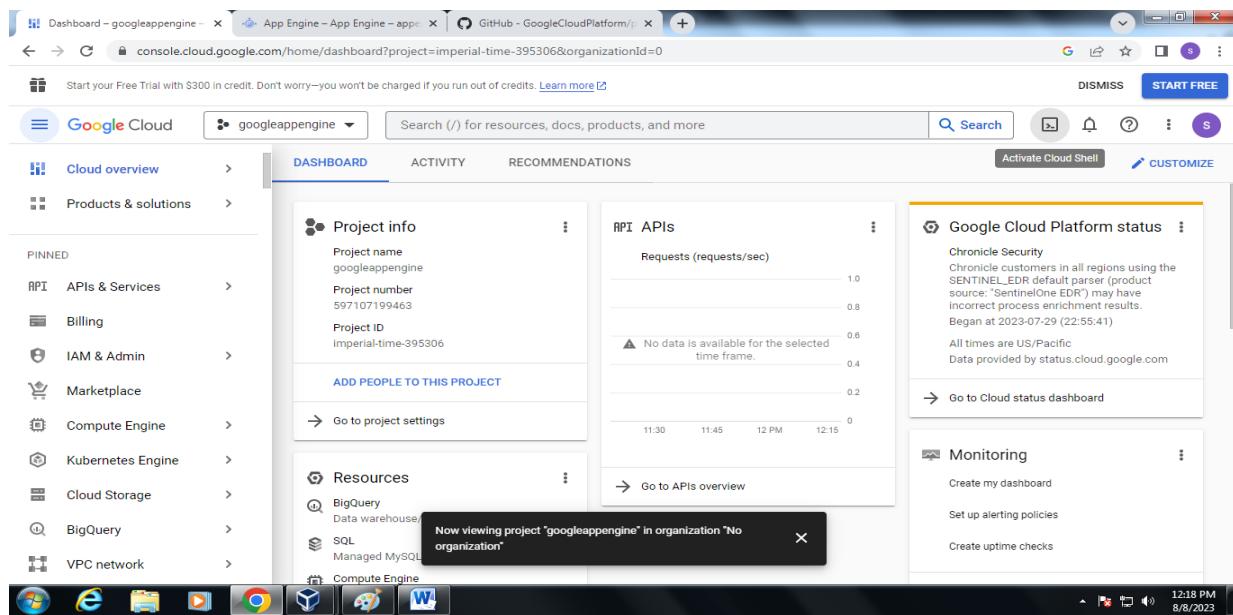


5.Select the created project – googleappengine.



Now the project is created with the project number and ID.

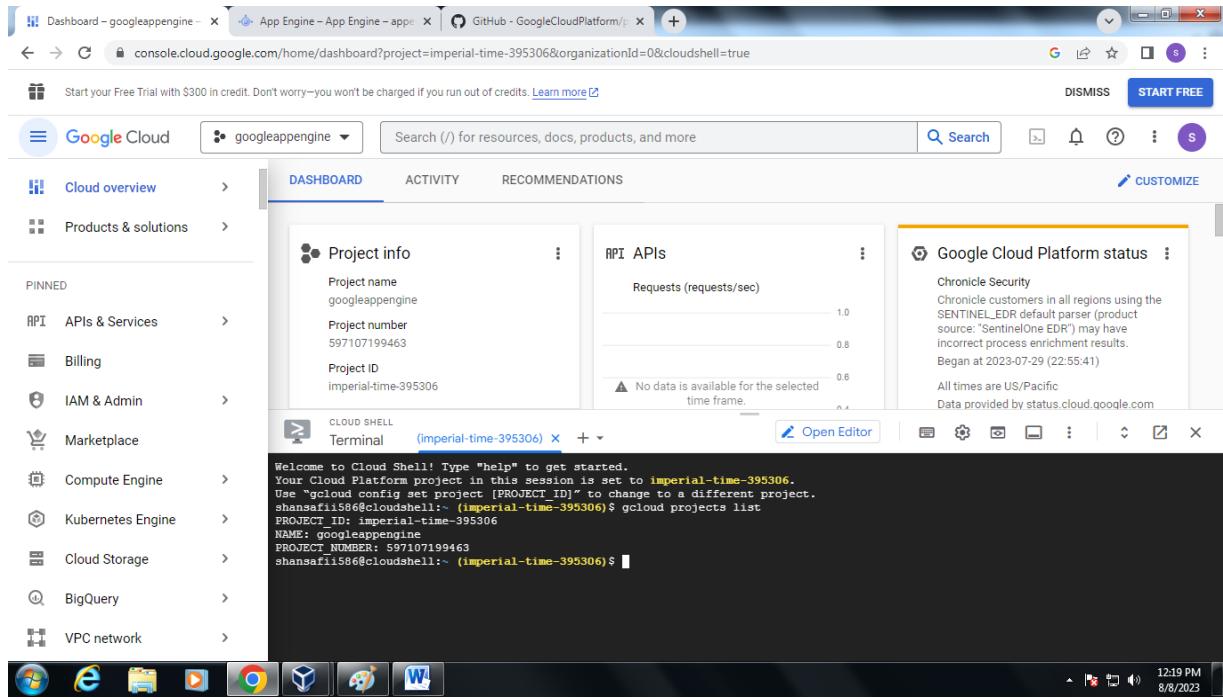
6.Activate Cloud Shell.



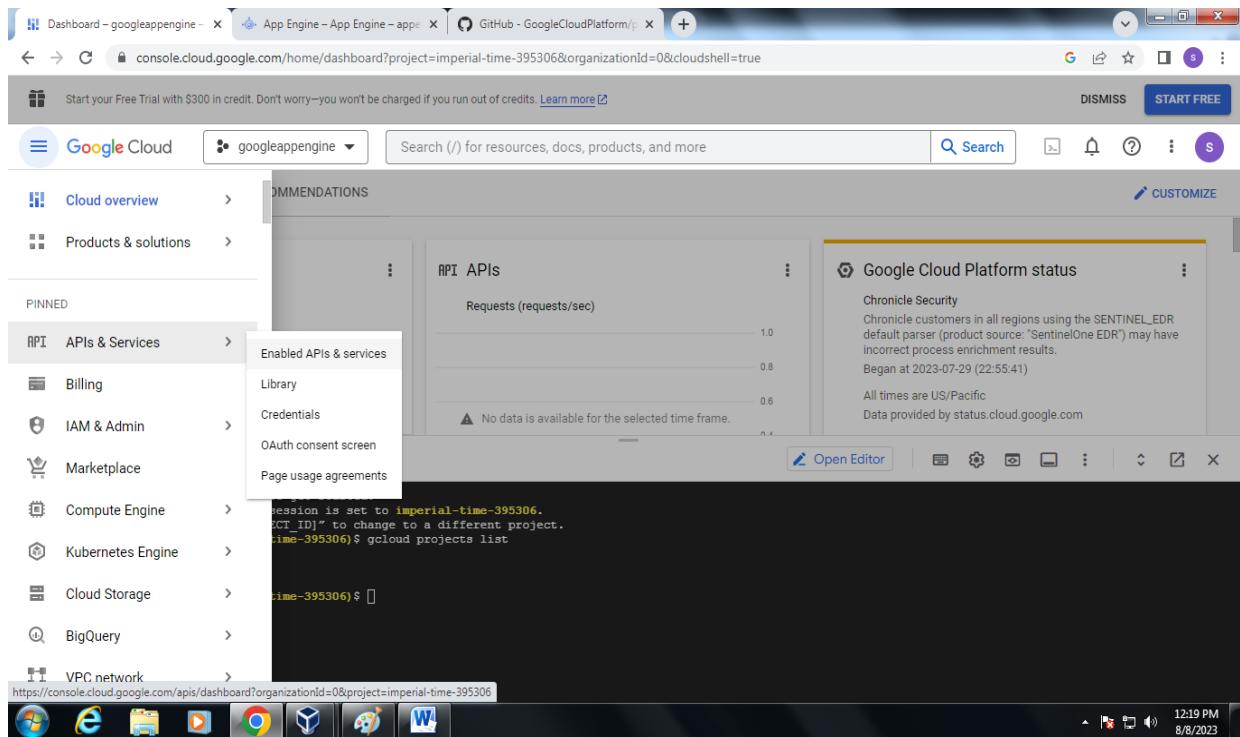
7.In Cloud Shell, type the command.

```
gcloud projects list
```

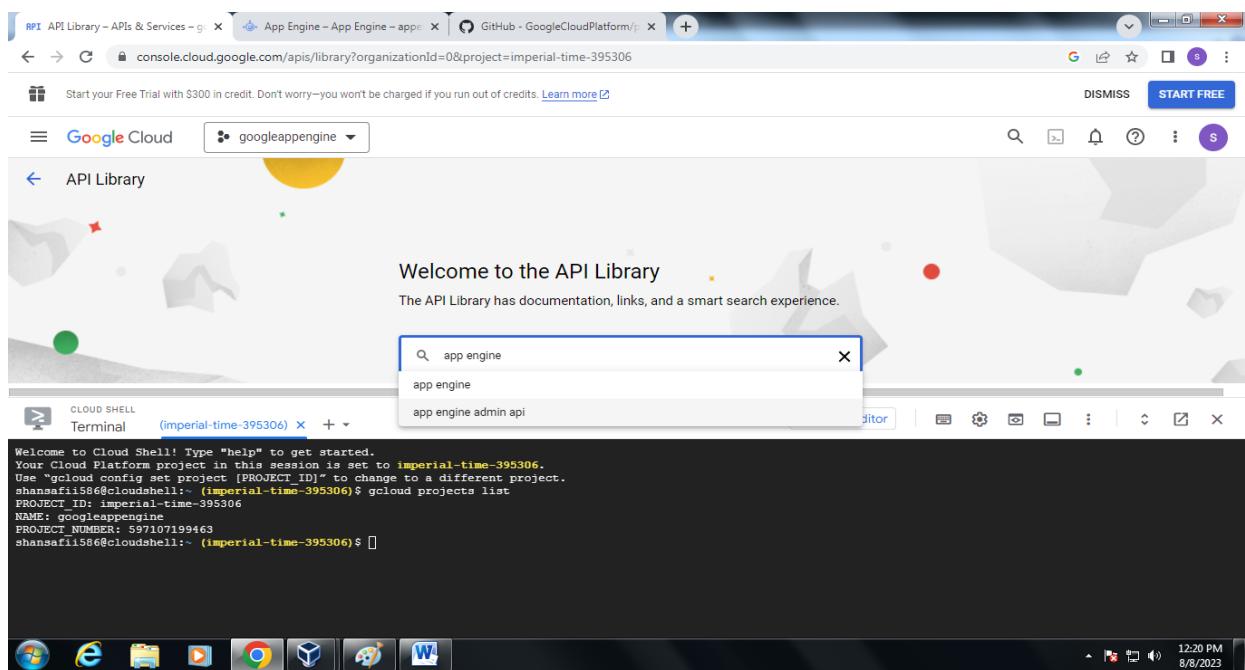
Project details will be listed in cloud shell.



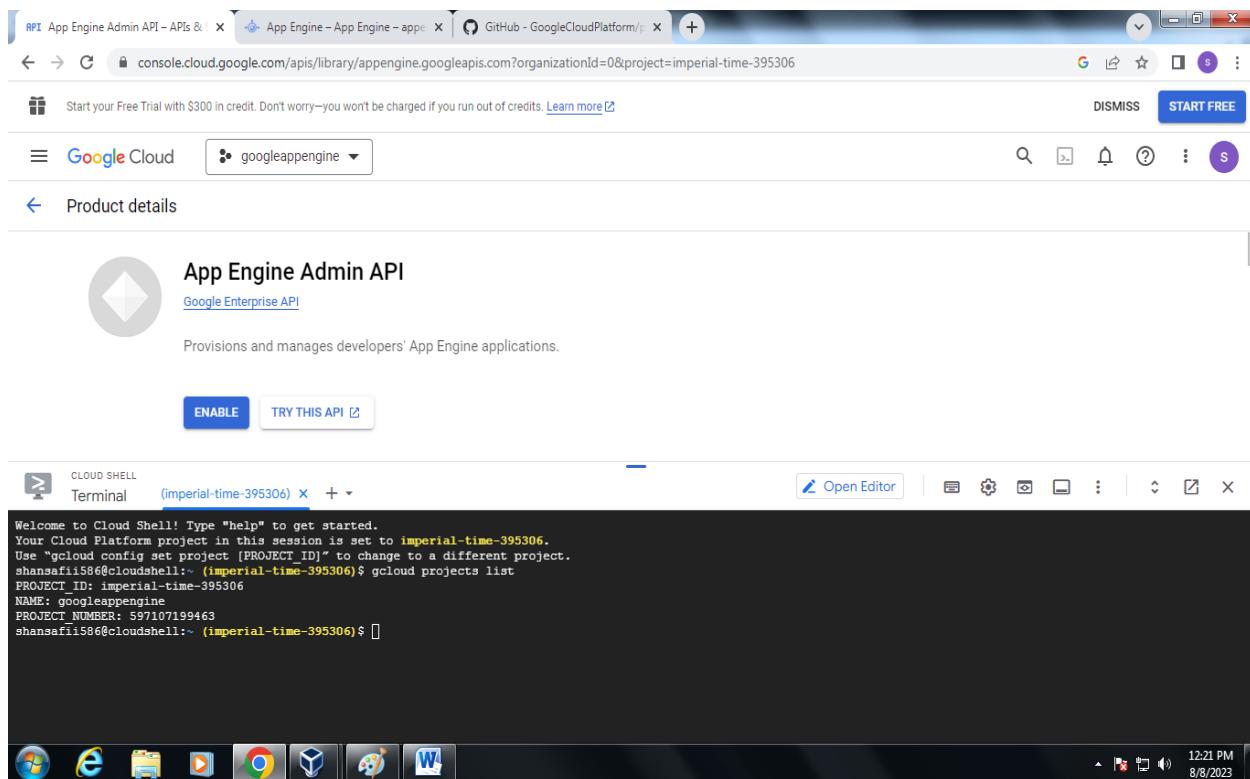
8. Go to menu select “APIs & Services” then select “Enabled APIs & Services”



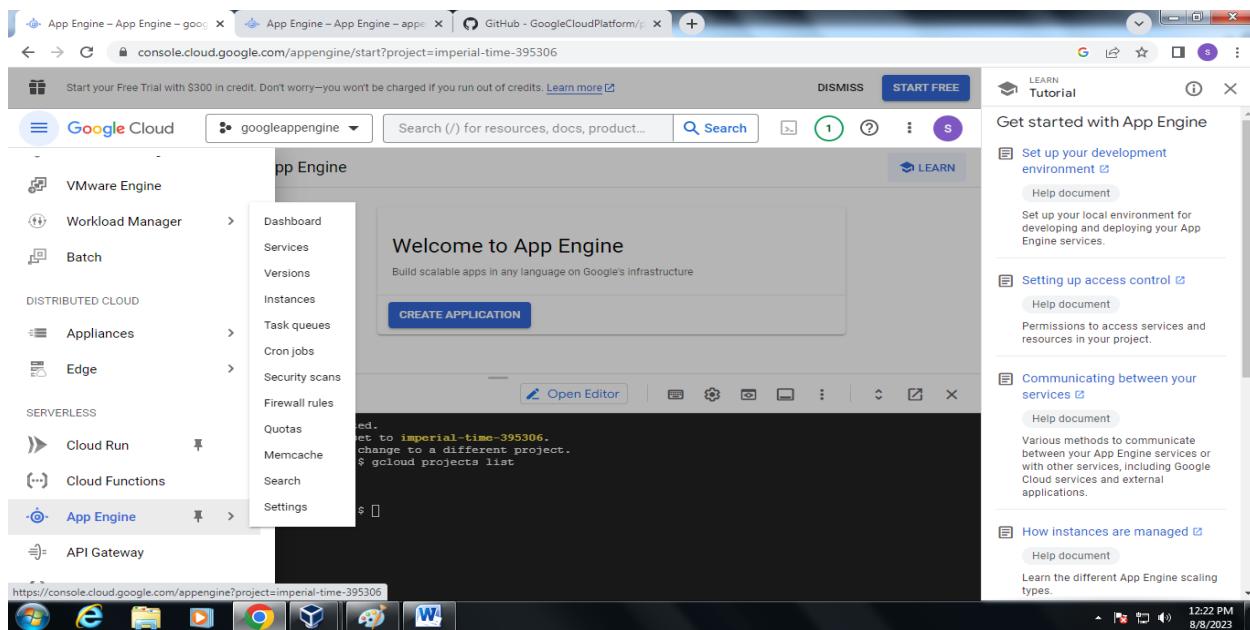
9. Search box will appear. Type “app engine” and select “App Engine Admin API”.



10. Click enable.

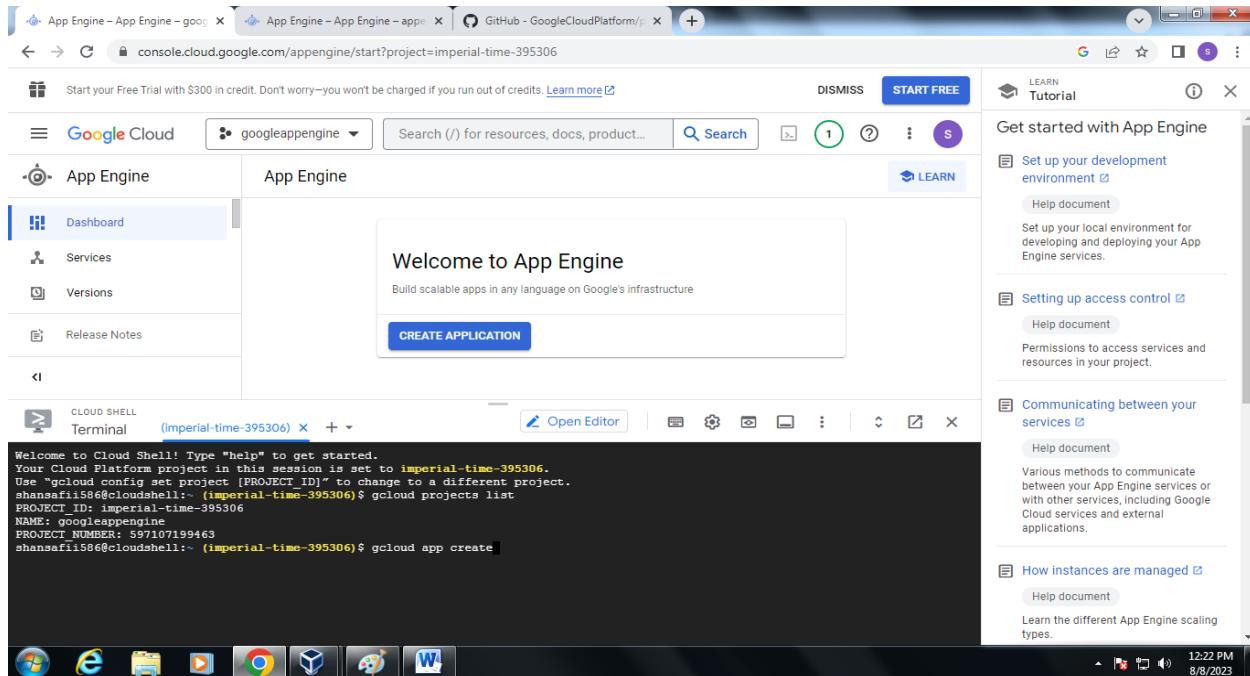


11. Goto menu and select “App Engine”.

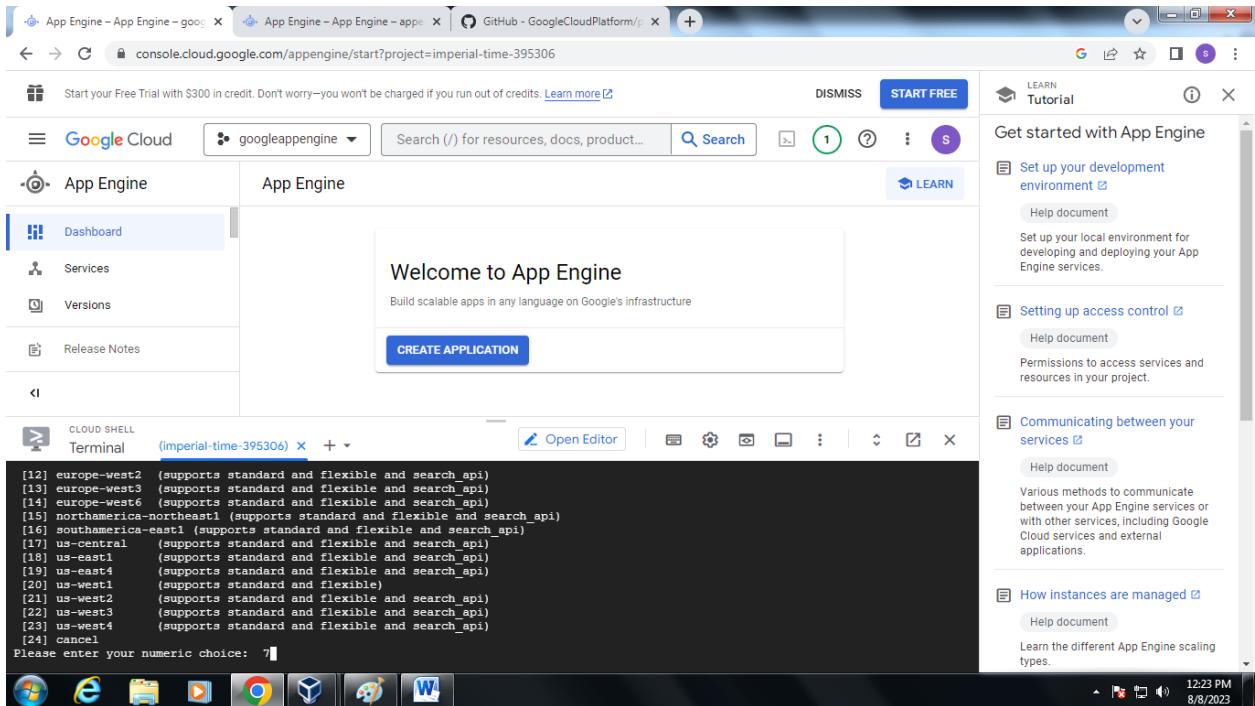


12. In cloud shell type the command to create app

```
gcloud app create
```



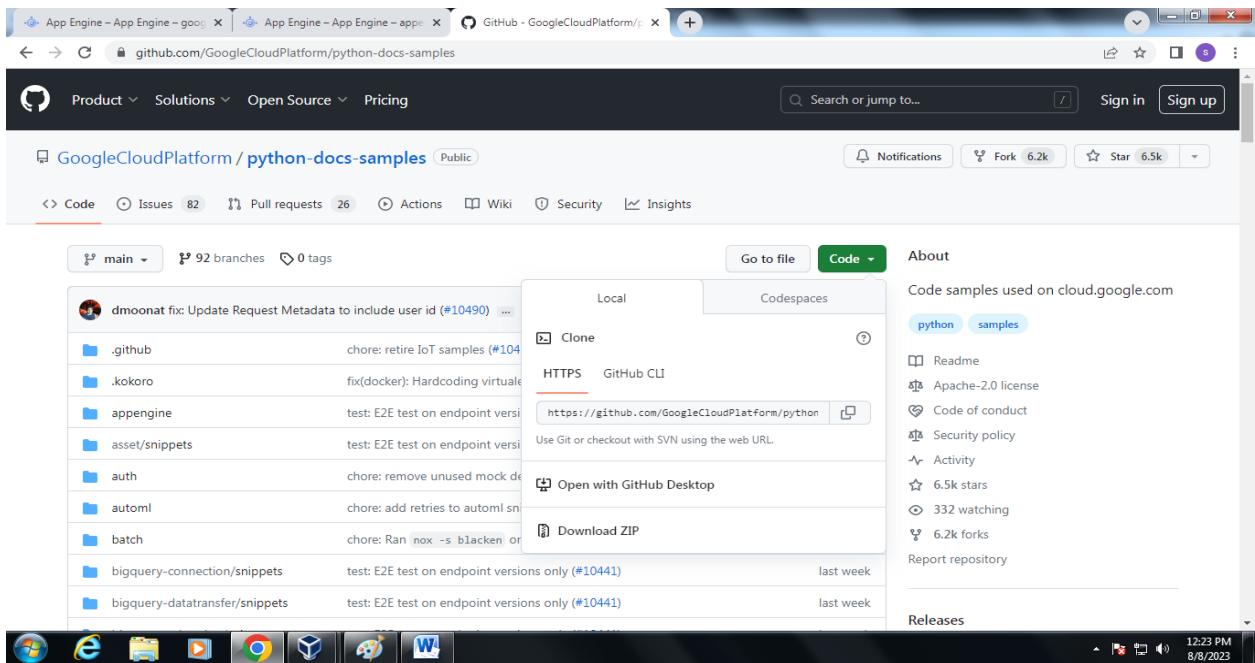
13. Select region 7(asia south east 1)

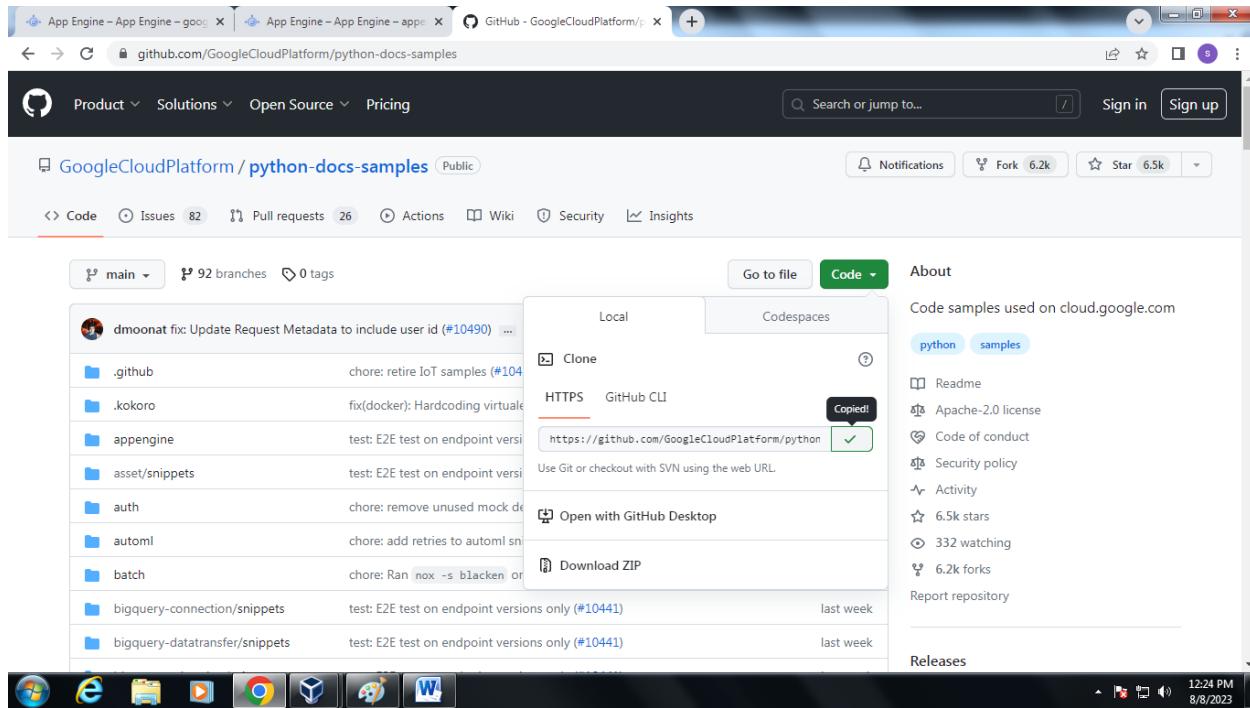


14. goto the drive repository to get the python file

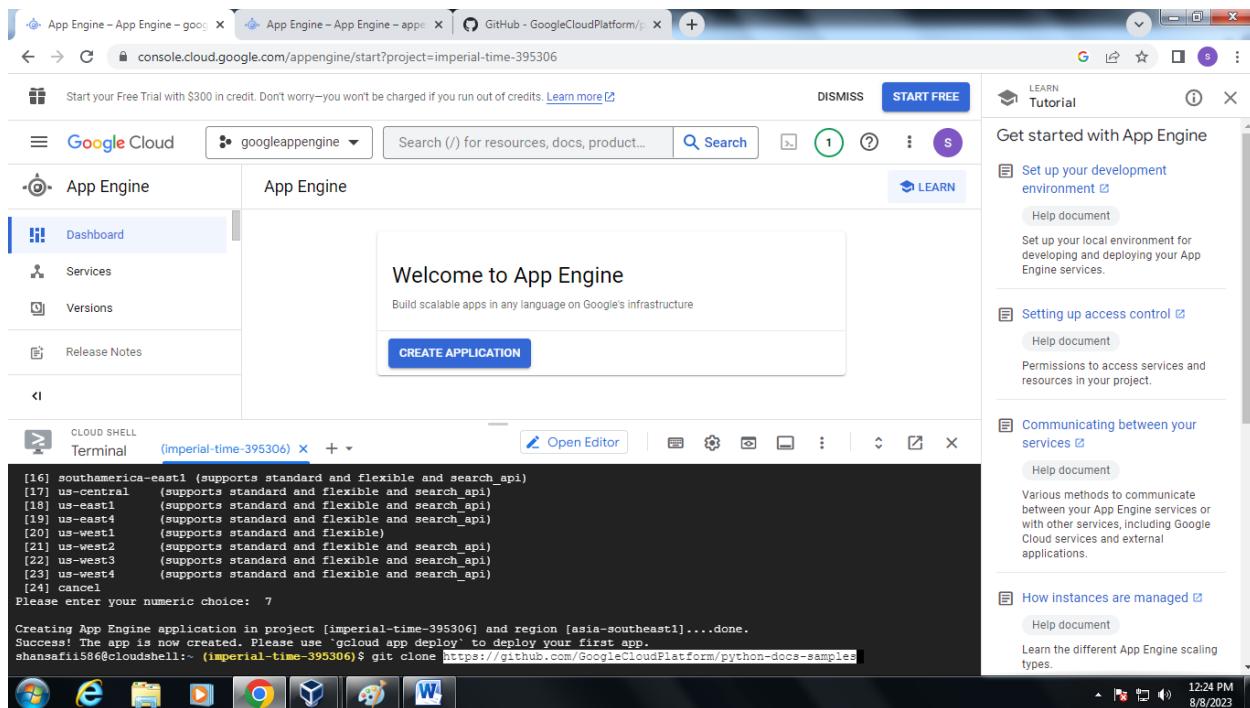
github.com/GoogleCloudPlatform/python-docs-samples

select code and copy the link.





15.Paste the link in cloud shell.



16.In cloud shell, type the command ls.

- cd python-docs-samples.
- cd appengine.

- cd standard_python3.
- cd hello_world.
- cat main.py (which contains helloworld program).
- python3 main.py.
- click URL <http://127.0.0.1:8080>.

The screenshot shows a Google Chrome window with three tabs: 'App Engine - App Engine - app', 'https://8080-cs-90a3912e-ef86...', and 'GitHub - GoogleCloudPlatform/p...'. The main content is a 'Google Cloud' interface with a 'Terminal' tab open. The terminal window displays Python code for a 'hello_world' application. It includes a docstring for the 'hello()' function and logic to check if the script is run as the main module. The terminal then runs the command `python3 main.py` and shows the resulting WSGI logs. A blue callout box points to the terminal window with the text: 'Click here to see details about your Cloud Shell session and usage quota' and 'Got it!'. To the right, a sidebar titled 'Get started with App Engine' lists several sections: 'Set up your development environment', 'Setting up access control', 'Communicating between your services', and 'How instances are managed'. The status bar at the bottom right shows the time as 12:27 PM and the date as 8/8/2023.

```

app.route("/")
def hello():
    """Return a friendly HTTP greeting.

    Returns:
        A string with the words 'Hello World!'.

    """
    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. You
    # can configure startup instructions by adding 'entrypoint' to app.yaml.
    app.run(host="127.0.0.1", port=8080, debug=True)
# [END gae_python3_app]
# [END gae_python38_app]
sharmub394@cloudshell:/python-docs-samples/appengine/standard_python3/hello_world (virtual-firefly-395306)$ python3 main.py
* Serving Flask app 'main'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:8080
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 143-582-884
127.0.0.1 -- [08/Aug/2023 06:40:53] "GET /authuser=0&redirectedPreviously=true HTTP/1.1" 200 -
127.0.0.1 -- [08/Aug/2023 06:40:53] "GET /favicon.ico HTTP/1.1" 404 -
[]

```

This screenshot is nearly identical to the one above, showing the same terminal session and sidebar. The only difference is the timestamp in the status bar, which has changed from 12:27 PM to 12:28 PM, and the date, which remains at 8/8/2023.

```

app.route("/")
def hello():
    """Return a friendly HTTP greeting.

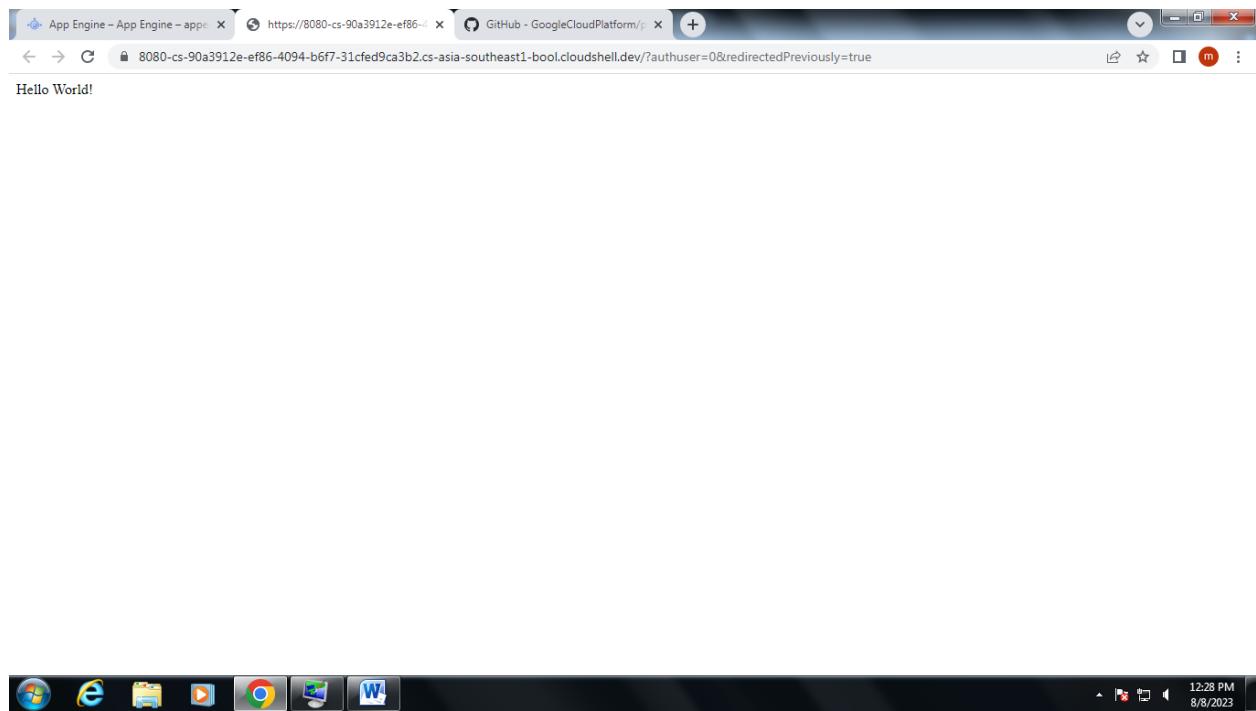
    Returns:
        A string with the words 'Hello World!'.

    """
    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. You
    # can configure startup instructions by adding 'entrypoint' to app.yaml.
    app.run(host="127.0.0.1", port=8080, debug=True)
# [END gae_python3_app]
# [END gae_python38_app]
sharmub394@cloudshell:/python-docs-samples/appengine/standard_python3/hello_world (virtual-firefly-395306)$ python3 main.py
* Serving Flask app 'main'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:8080
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 143-582-884
127.0.0.1 -- [08/Aug/2023 06:40:53] "GET /authuser=0&redirectedPreviously=true HTTP/1.1" 200 -
127.0.0.1 -- [08/Aug/2023 06:40:53] "GET /favicon.ico HTTP/1.1" 404 -
[]

```

17.The Hello world message from the sample app is displayed on the web page.



RESULT:

Thus the project to run the python sample program to display Helloworld in Google App Engine in google cloud console is executed successfully.

EX.NO: 4

DATE:

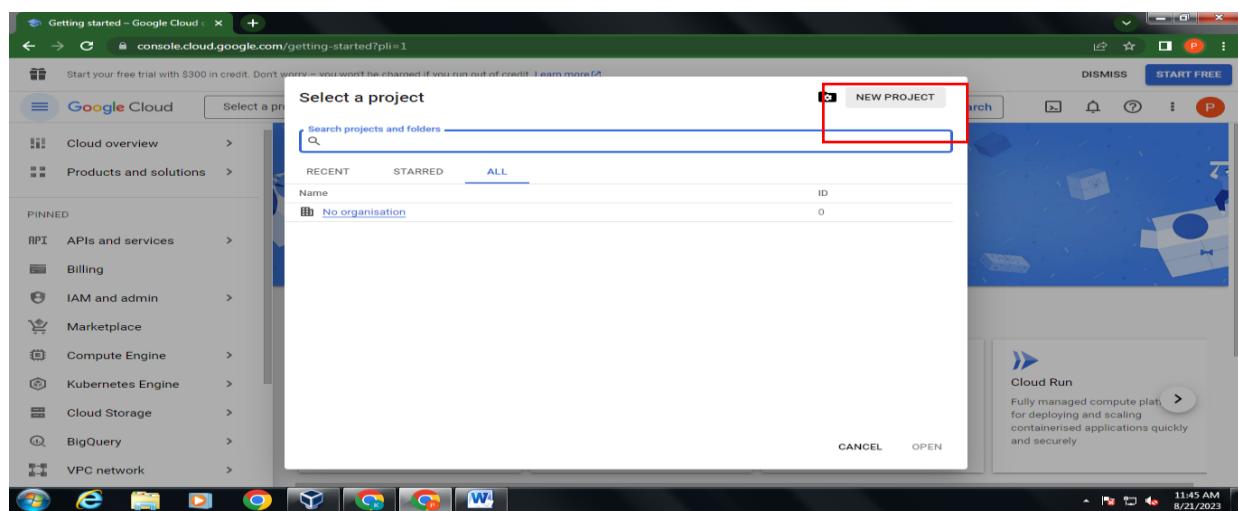
USE THE GAE LAUNCHER TO LAUNCH THE WEB APPLICATIONS

AIM:

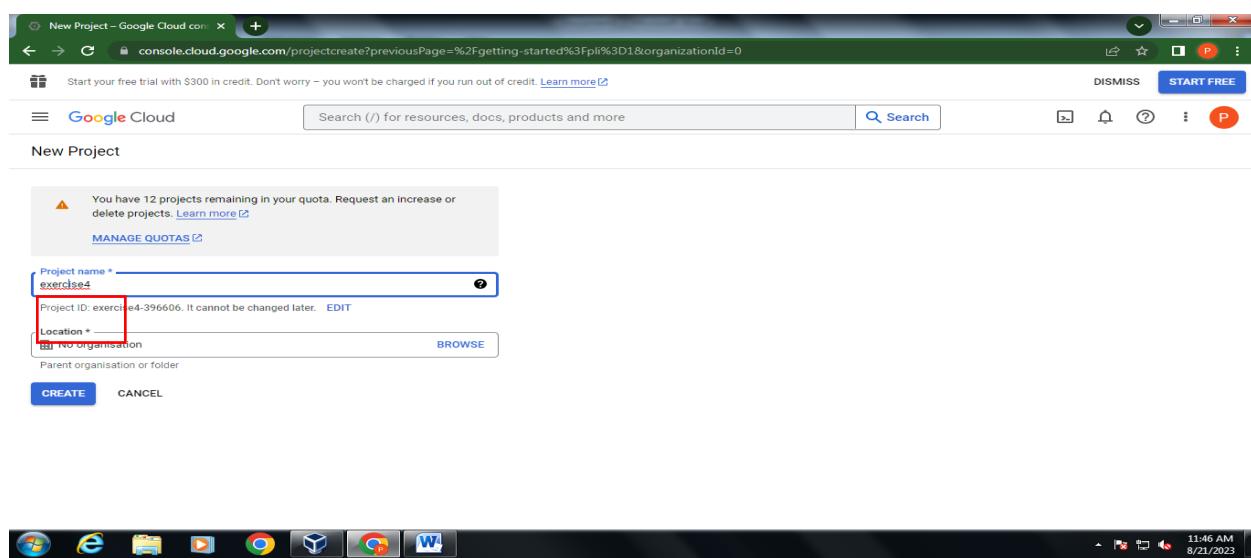
To create a project with the help of GAE launcher to launch web application Helloworld in java.

PROCEDURE:

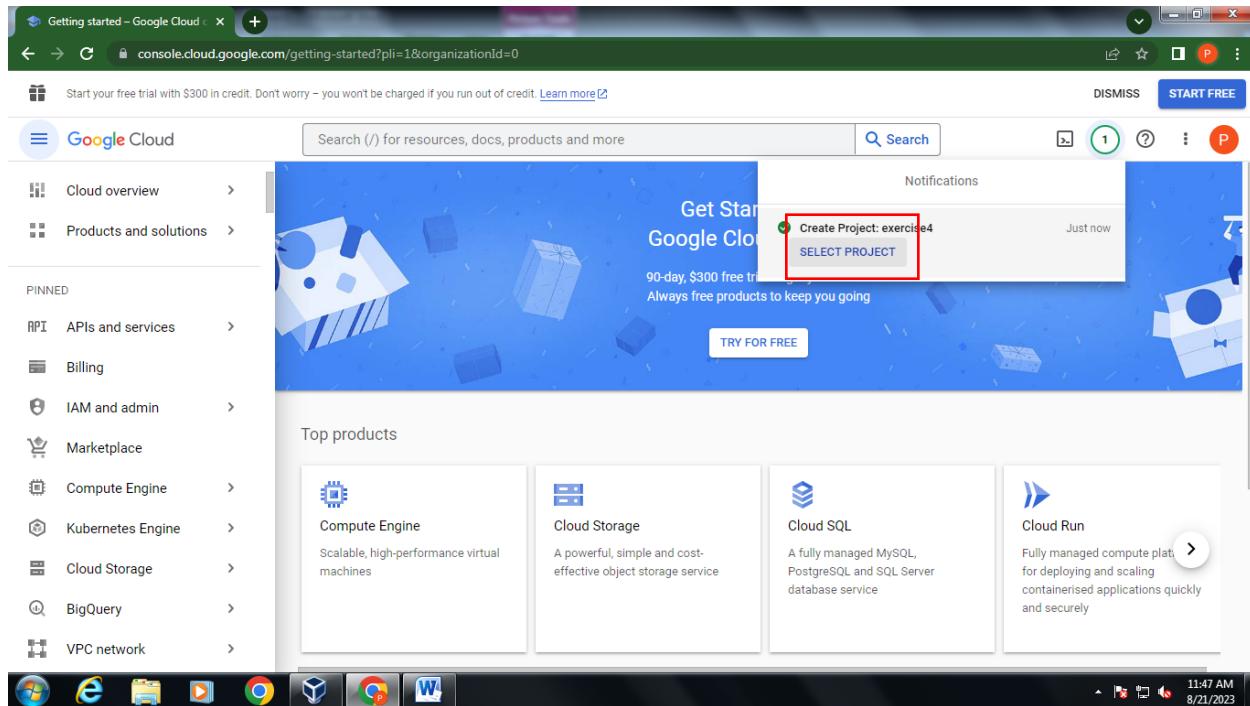
1.Create new project



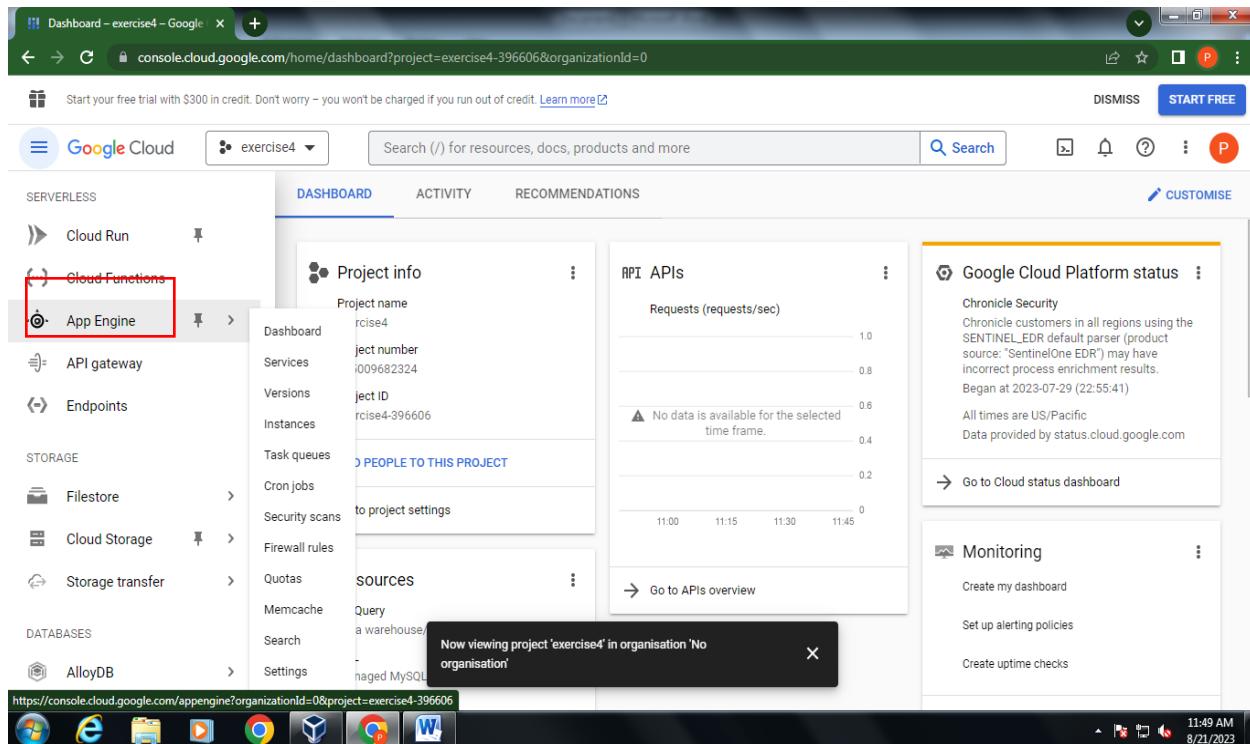
2.Give project name



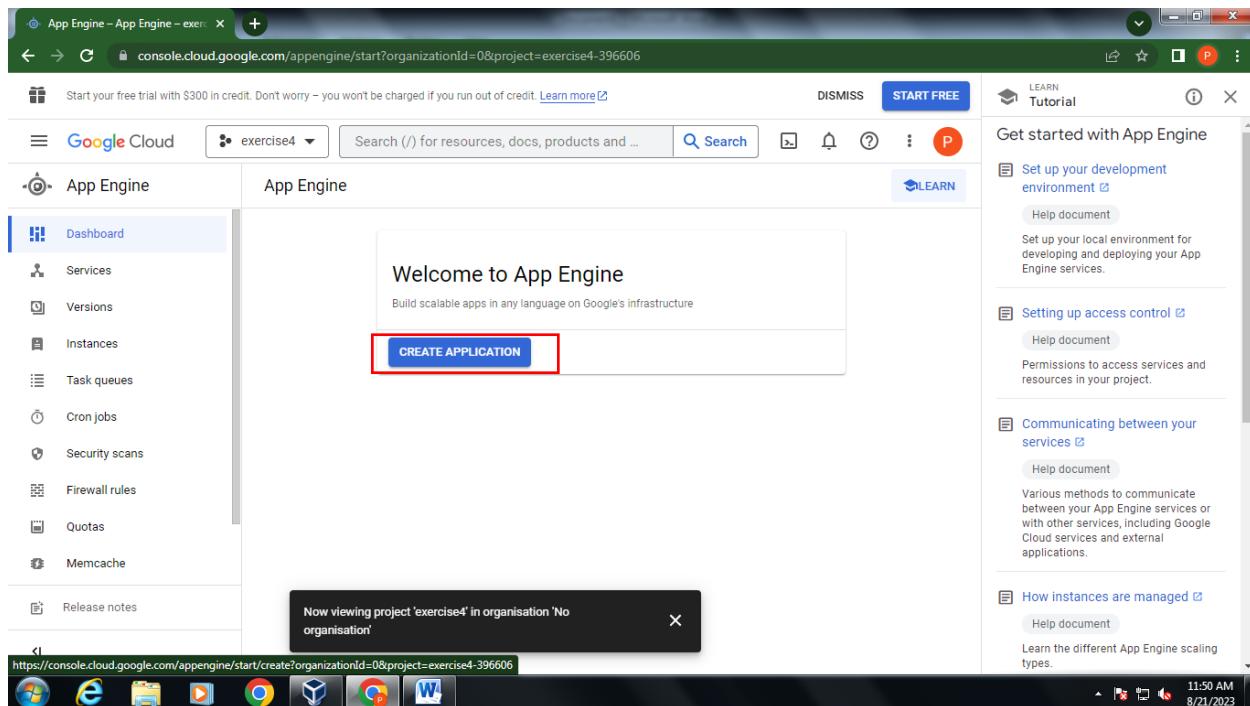
3.select that created project



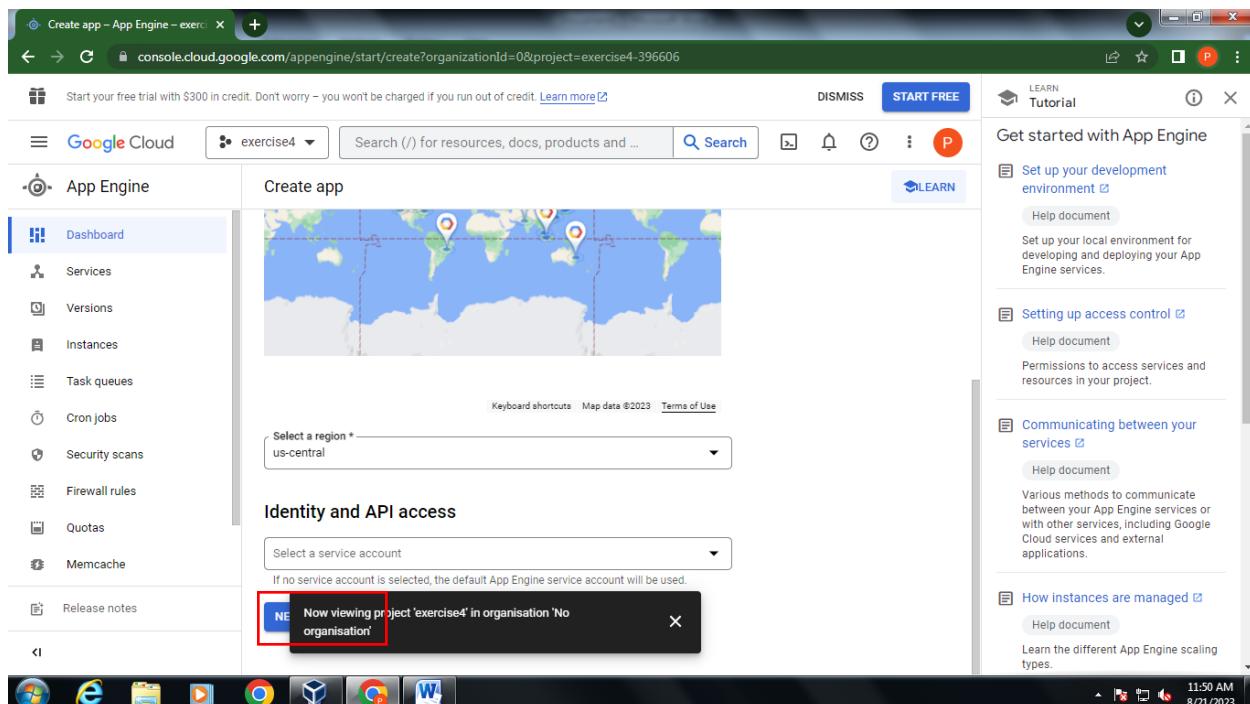
4.In menu goto appengine

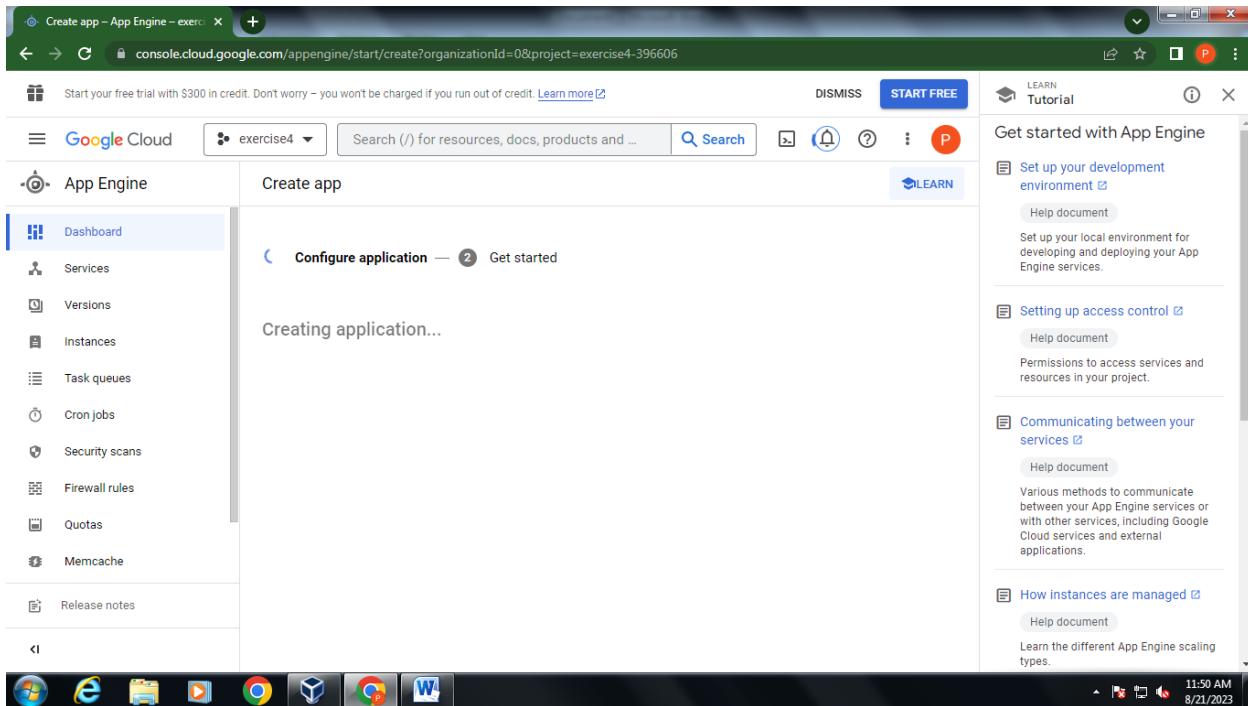


5.select “create application”

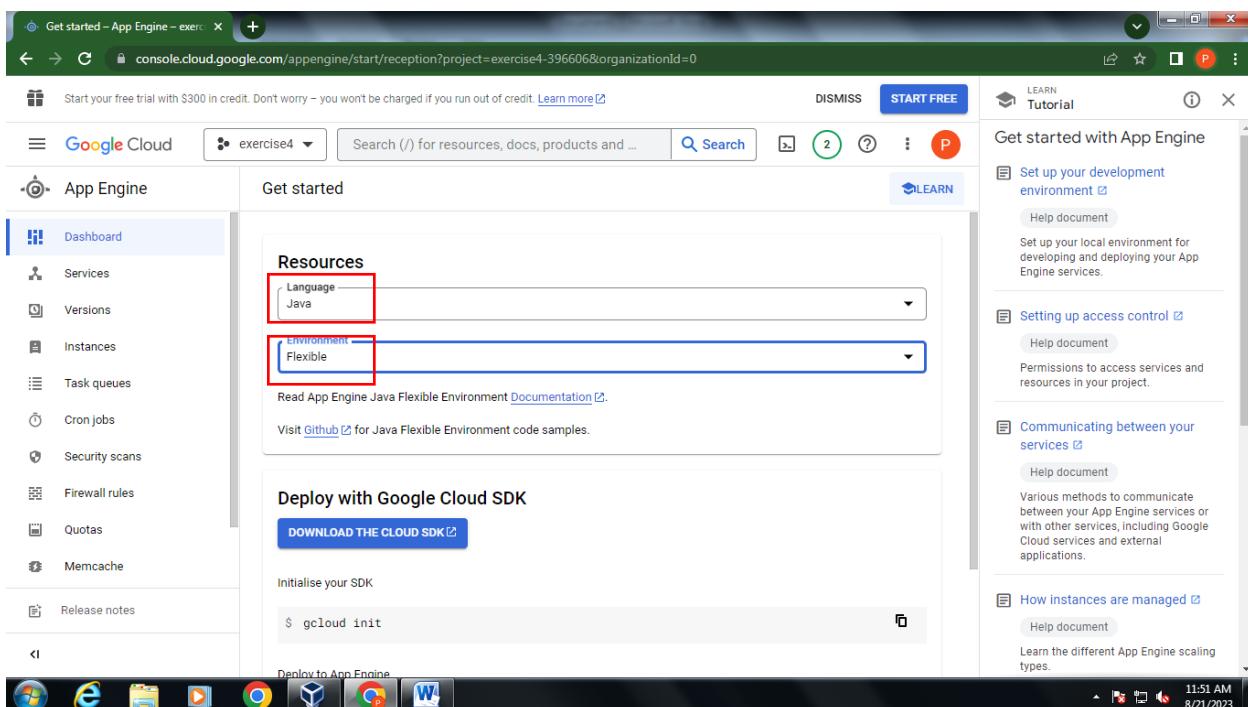


6.Select region us-central and give ok





7. Select language-java and Environment-flexible



8.Select GITHUB repository for java application

The screenshot shows the Google Cloud Platform App Engine dashboard. On the left, there's a sidebar with options like Dashboard, Services, Versions, Instances, Task queues, Cron jobs, Security scans, Firewall rules, Quotas, Memcache, and Release notes. The main area is titled 'Get started' for the Flexible environment. It includes a link to 'Read App Engine Java Flexible Environment Documentation' and a link to 'Github' (which is highlighted with a red box). Below that is a section titled 'Deploy with Google Cloud SDK' with a 'DOWNLOAD THE CLOUD SDK' button and some terminal command examples: '\$ gcloud init', 'Deploy to App Engine', and '\$ gcloud app deploy'. A 'I'LL DO THIS LATER' button is also present. To the right, there's a 'LEARN' section with links to 'Tutorial', 'Get started with App Engine', 'Set up your development environment', 'Setting up access control', 'Communicating between your services', and 'How instances are managed'. The bottom status bar shows the URL <https://github.com/GoogleCloudPlatform/java-docs-samples/tree/master/appengine-java8>, the date 8/21/2023, and the time 11:51 AM.

The screenshot shows a GitHub repository page for 'java-docs-samples/appengine-java8'. The URL in the address bar is highlighted with a red box. The page has a dark theme. At the top, there are navigation links for Product, Solutions, Open Source, and Pricing, along with a search bar and sign-in options. The repository name 'GoogleCloudPlatform / java-docs-samples' is shown, along with a 'Code' tab (which is selected), 'Issues' (12), 'Pull requests' (11), 'Actions', 'Security', and 'Insights' tabs. On the left, there's a file tree showing directories like '.github', '.kokoro', 'accessapproval', 'aiplatform', 'appengine-java11-bundled-se...', 'appengine-java11', 'appengine-java17-bundled-se...', 'appengine-java8', 'analytics', and 'appidentity'. The main area displays a commit history for the 'appengine-java8' branch. The first commit is from 'renovate-bot' with the message 'chore(deps): update dependency gradle to v8.3 (#8558)' and a timestamp of 'a59e97c - 4 days ago'. Below it is a table of commits:

Name	Last commit message	Last commit date
..	chore(deps): update appengine packages to v2.0.17 (#8550)	5 days ago
analytics	chore(deps): update appengine packages to v2.0.17 (#8550)	5 days ago
appidentity	chore(deps): update appengine packages to v2.0.17 (#8550)	5 days ago
bigrquery	chore(deps): update appengine packages to v2.0.17 (#8550)	5 days ago
bigtable	chore(deps): update dependency gradle to v8.3 (#8558)	4 days ago
cloudsql	Remove old GAE to Cloud SQL samples and point to new ones instead...	3 years ago
datastore-indexes-explosive	chore(deps): update appengine packages to v2.0.17 (#8550)	5 days ago

The bottom status bar shows the URL <https://github.com/GoogleCloudPlatform/java-docs-samples/tree/main/appengine-java8>, the date 8/21/2023, and the time 11:53 AM.

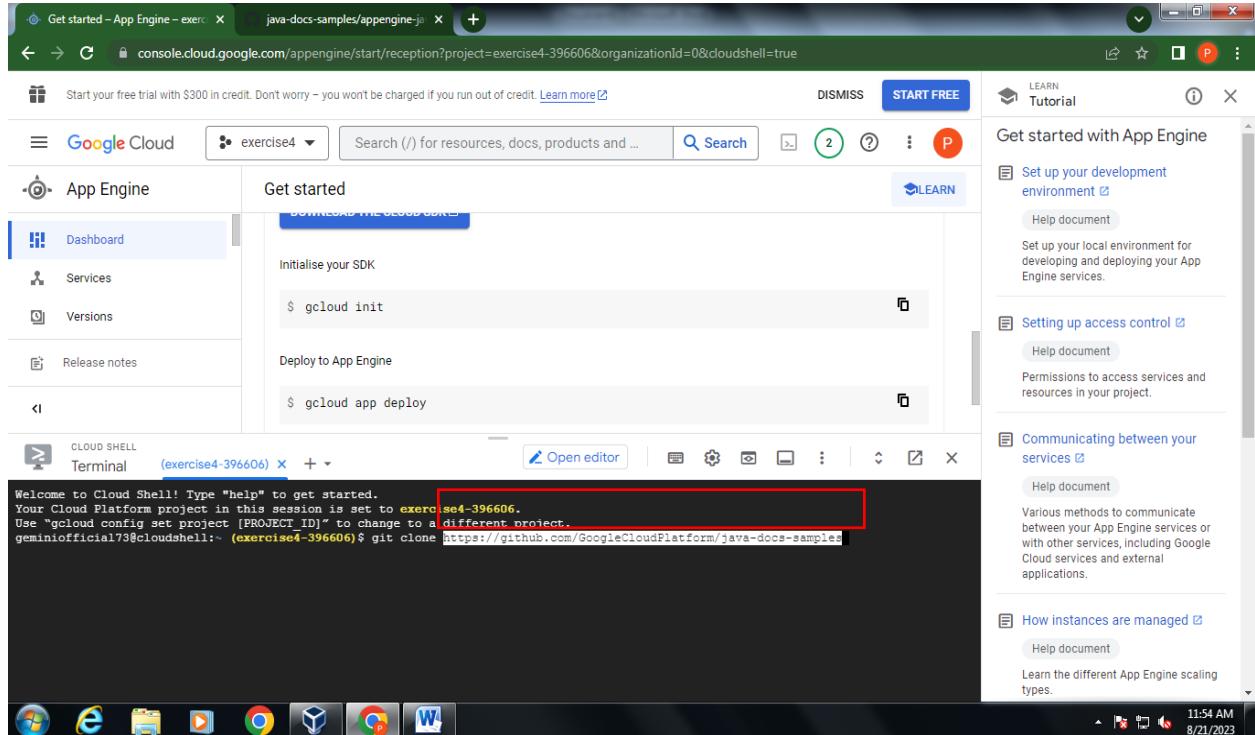
9. Activate Cloud shell

The screenshot shows the Google Cloud Platform console with the URL `console.cloud.google.com/appengine/start/reception?project=exercise4-396606&organizationId=0&cloudshell=false`. The left sidebar is open, showing the 'App Engine' section with various options like Dashboard, Services, Versions, Instances, Task queues, Cron jobs, Security scans, Firewall rules, Quotas, Memcache, and Release notes. The main content area is titled 'Get started' for the Flexible environment. It includes links to 'Documentation' and 'Github'. Below this is a section titled 'Deploy with Google Cloud SDK' with a 'DOWNLOAD THE CLOUD SDK' button. It shows two code snippets: '\$ gcloud init' under 'Initialise your SDK' and '\$ gcloud app deploy' under 'Deploy to App Engine'. At the bottom, there's a link 'I'LL DO THIS LATER'. In the top right, there are 'DISMISS' and 'START FREE' buttons, along with a 'Cloud Shell' button which is highlighted with a red box. The right side of the screen features a 'LEARN Tutorial' sidebar with sections like 'Get started with App Engine', 'Setting up access control', 'Communicating between your services', and 'How instances are managed'. The system tray at the bottom right shows the date and time as 11:53 AM 8/21/2023.

This screenshot is identical to the one above, but the 'Cloud Shell' button in the top right is now greyed out, indicating it has been activated. A terminal window titled 'Terminal (exercise4-396606)' is open at the bottom. The terminal displays the following text:
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to `exercise4-396606`.
Use "`gcloud config set project [PROJECT_ID]`" to change to a different project.
geminiofficial73@cloudshell:~ (exercise4-396606)\$

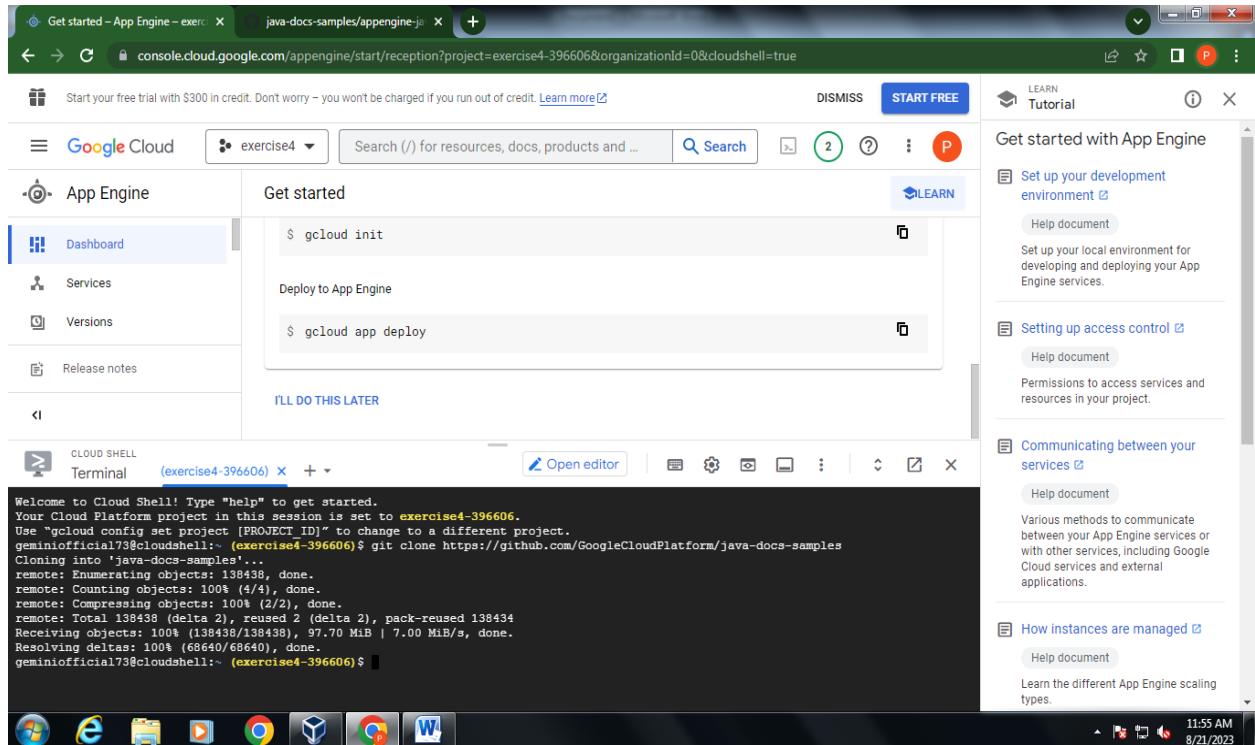
10.Paste the github link in cloud shell

github.com/GoogleCloudPlatform/java-docs-samples



The screenshot shows the Google Cloud Platform App Engine dashboard. In the terminal window, the command `git clone https://github.com/GoogleCloudPlatform/java-docs-samples` is being typed. A red box highlights the URL in the command line. The terminal also displays a welcome message and some initial deployment steps.

11.Project is authorized in cloud platform



The screenshot shows the Google Cloud Platform App Engine dashboard. The terminal window now shows the output of cloning the repository: "Cloning into 'java-docs-samples'...". The right sidebar indicates that the project is now authorized.

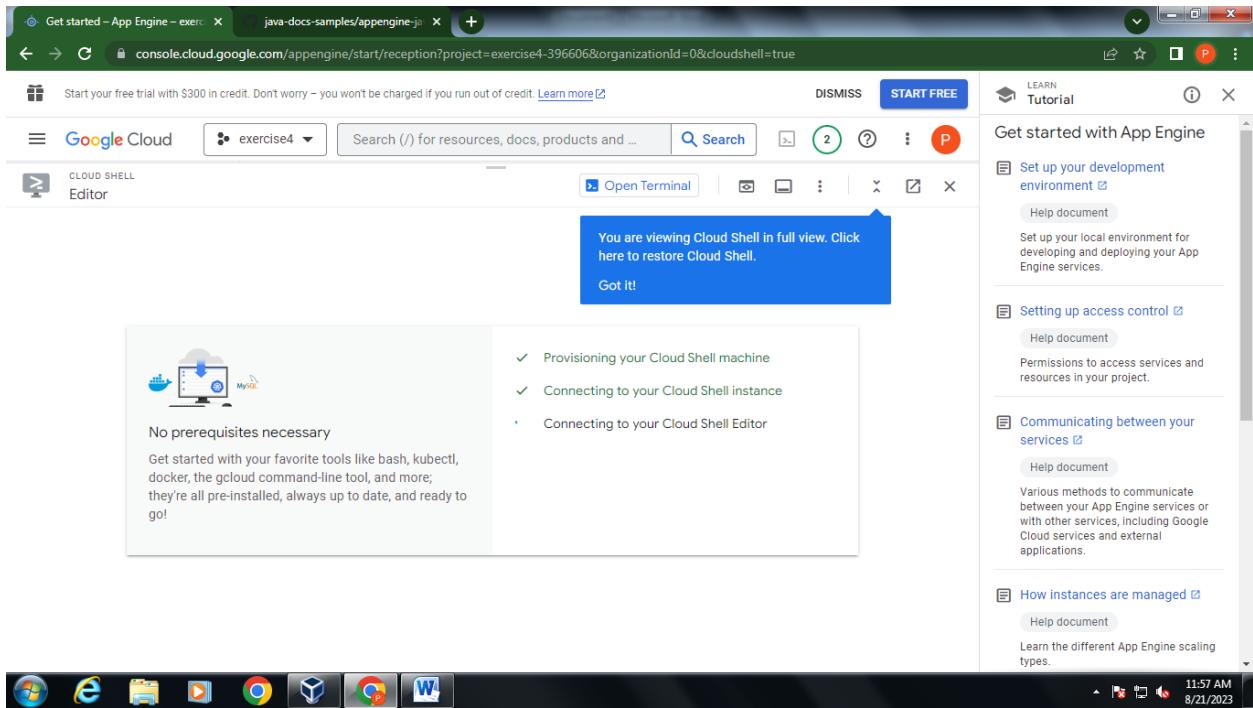
12.Type the commands in cloud shell.

cd java-docs-samples

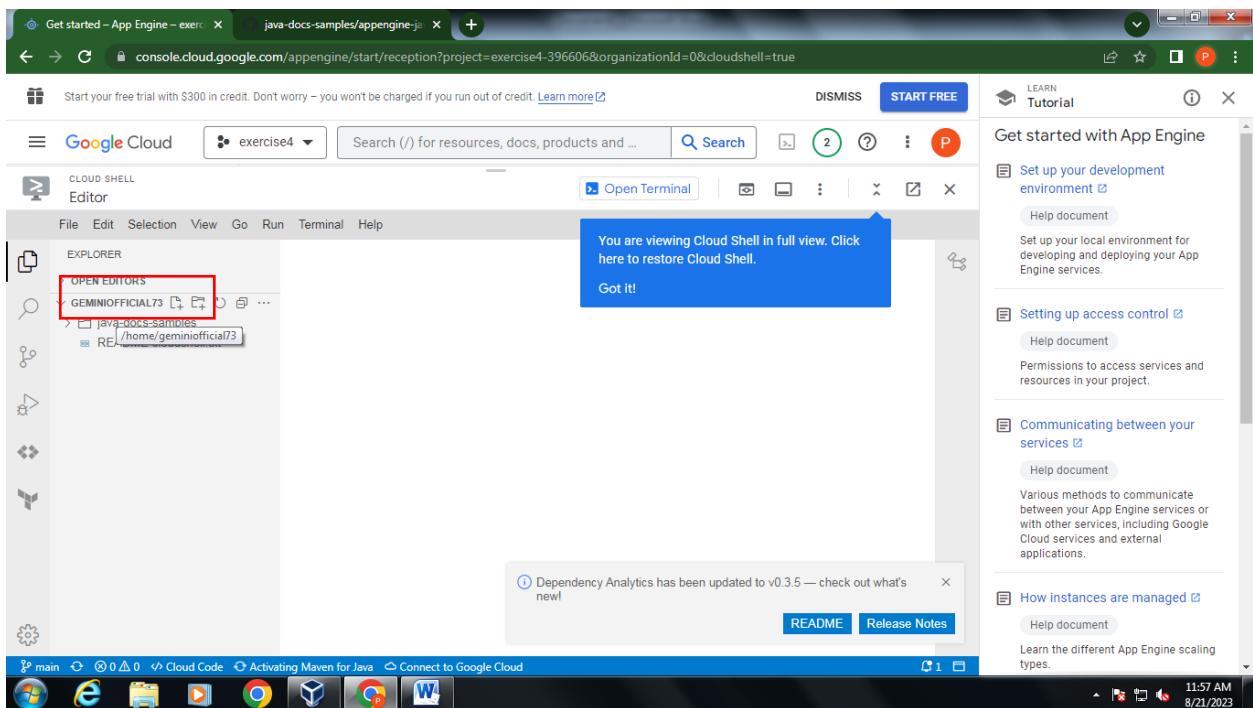
The screenshot shows the Google Cloud Platform App Engine dashboard. On the left, there's a sidebar with 'App Engine' selected, showing 'Dashboard', 'Services', 'Versions', and 'Release notes'. The main area is titled 'Get started' with two code snippets: '\$ gcloud init' and '\$ gcloud app deploy'. Below these are two buttons: 'I'LL DO THIS LATER' and 'CLOUD SHELL'. The 'CLOUD SHELL' button is highlighted with a red box. A terminal window titled '(exercise4-396606)' is open, showing a git clone command for 'java-docs-samples'. The command 'cd java-docs-samples' is being typed and is highlighted with a red box. The terminal output shows the cloning process and the current directory as 'java-docs-samples'. The status bar at the bottom right shows the date and time: '11:56 AM 8/21/2023'.

13.open editor.

This screenshot is similar to the previous one, showing the App Engine dashboard with the 'CLOUD SHELL' button highlighted. However, the 'Open editor' button in the terminal bar is now highlighted with a red box. The terminal window content and status bar are identical to the previous screenshot.



14. Check whether you are working in your google account (account name).



15.Goto cloud shell

The screenshot shows the Google Cloud Platform interface with the 'Cloud Shell' tab selected. A terminal window titled '(exercise4-396606)' is open, displaying a command-line session. The user has run 'git clone' to clone a GitHub repository into their project directory. The terminal output includes details about the cloning process and lists various Google services available in the project. A red box highlights the command 'cd java-docs-samples' which the user has just entered. A blue callout box says 'You are viewing Cloud Shell in full view. Click here to restore Cloud Shell.' with a 'Got it!' button.

```
Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to exercise4-396606.  
Use "gcloud config set project [PROJECT_ID]" to change to a different project.  
geminiofficial73@cloudshell:~ (exercise4-396606)$ git clone https://github.com/  
Cloning into 'java-docs-samples'...  
remote: Enumerating objects: 138438, done.  
remote: Counting objects: 100% (4/4), done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 138438 (delta 2), reused 2 (delta 2), pack-reused 138434  
Receiving objects: 100% (138438/138438), 97.70 MiB | 7.00 MiB/s, done.  
Resolving deltas: 100% (68640/68640), done.  
geminiofficial73@cloudshell:~ (exercise4-396606)$ cd ^C  
geminiofficial73@cloudshell:~ (exercise4-396606)$ cd java-docs-samples  
geminiofficial73@cloudshell:~/java-docs-samples (exercise4-396606)$ ls  
accessapproval      compute     eventarc    monitoring   session-handling  
aiplatform          contact-center-insights flexible    opencensus  spanner  
appengine-javall    container-registry  functions   optimization speech  
appengine-javall-bundled-services content-warehouse  healthcare privateca storage  
appengine-javall7-bundled-services CONTRIBUTING.md    iam        pubsub    storageinsights  
appengine-java8      dataproc     jobs       privateca  storage-transfer  
asset                dataflow     iot        pubsublite  talents  
auth                datalabeling  kms       reCAPTCHA_enterprise tasks  
automacl           dataproc     language   run        texttospeech  
batch               dialogflow   dialogflow-cx  LICENSE    retail  
bigquery            discoveryengine  discoveryengine  media  renovate.json translate  
bigtable             dip         dip        mediacdn   run  
build.gradle        document-ai  endpoints   memorystore security-command-center trace  
cdn                 endpoints   erroreporting  mlenGINE  SECURITY.md vision  
cloud-sql            endpoints   erroreporting  mlengIne  servicedirectory webrisk  
CODE_OF_CONDUCT.md  
geminiofficial73@cloudshell:~/java-docs-samples (exercise4-396606)$
```

16.type the commands in shell.

```
cd appengine-java8
```

```
ls
```

The screenshot shows the Google Cloud Platform interface with the 'Cloud Shell' tab selected. A terminal window titled '(exercise4-396606)' is open, displaying a command-line session. The user has run 'git clone' to clone a GitHub repository into their project directory. The terminal output includes details about the cloning process and lists various Google services available in the project. A red box highlights the command 'cd appengine-java8' which the user has just entered. A blue callout box says 'You are viewing Cloud Shell in full view. Click here to restore Cloud Shell.' with a 'Got it!' button.

```
Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to exercise4-396606.  
Use "gcloud config set project [PROJECT_ID]" to change to a different project.  
geminiofficial73@cloudshell:~ (exercise4-396606)$ git clone https://github.com/  
Cloning into 'java-docs-samples'...  
remote: Enumerating objects: 138438, done.  
remote: Counting objects: 100% (4/4), done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 138438 (delta 2), reused 2 (delta 2), pack-reused 138434  
Receiving objects: 100% (138438/138438), 97.70 MiB | 7.00 MiB/s, done.  
Resolving deltas: 100% (68640/68640), done.  
geminiofficial73@cloudshell:~ (exercise4-396606)$ cd ^C  
geminiofficial73@cloudshell:~ (exercise4-396606)$ cd java-docs-samples  
geminiofficial73@cloudshell:~/java-docs-samples (exercise4-396606)$ ls  
accessapproval      compute     eventarc    monitoring   session-handling  
aiplatform          contact-center-insights flexible    opencensus  spanner  
appengine-javall    container-registry  functions   optimization speech  
appengine-javall-bundled-services content-warehouse  healthcare privateca storage  
appengine-javall7-bundled-services CONTRIBUTING.md    iam        pubsub    storageinsights  
appengine-java8      dataproc     jobs       privateca  storage-transfer  
asset                dataflow     iot        pubsublite  talents  
auth                datalabeling  kms       reCAPTCHA_enterprise tasks  
automacl           dataproc     language   run        texttospeech  
batch               dialogflow   dialogflow-cx  LICENSE    retail  
bigquery            discoveryengine  discoveryengine  media  renovate.json translate  
bigtable             dip         dip        mediacdn   run  
build.gradle        document-ai  endpoints   memorystore security-command-center trace  
cdn                 endpoints   erroreporting  mlenGINE  SECURITY.md vision  
cloud-sql            endpoints   erroreporting  mlengIne  servicedirectory webrisk  
CODE_OF_CONDUCT.md  
geminiofficial73@cloudshell:~/java-docs-samples (exercise4-396606)$ cd appengine-java8
```

The screenshot shows a Google Cloud Shell terminal window. The command `cd java-docs-samples` has been run, and the contents of the directory are listed:

```
geminiofficial73@cloudshell:~ (exercise4-396606)$ cd java-docs-samples
geminiofficial73@cloudshell:~/java-docs-samples (exercise4-396606)$ ls
accessapproval           compute          eventarc
aiplatform                contact-center-insights flexible
appengine-javall           container-registry   functions
appengine-javall-bundled-services content-warehouse healthcare
appengine-javall-bundled-services CONTRIBUTING.md    iam
asset                     datacatalog      iap
auth                      dataflow        iot
automa                    datalabeling     jobs
batch                     dialogflow      language
bigquery                  dialogflow-cx    LICENSE
bigtable                  discoveryengine media
build.gradle              dip             media_cdn
cdn                       document-ai      mediatranslation
cloud-sql                 endpoints       memorystore
CODE_OF_CONDUCT.md        errorreporting  mlenGINE
geminiofficial73@cloudshell:~/java-docs-samples (exercise4-396606)$ cd appengine-java8 (exercise4-396606)$ ls
analytics                endpoints-v2-guice  helloworld  README.md
appidentity               endpoints-v2-migration  iap         remote-client
bigquery                  endpoints-v2-README.md  images     remote-README.md
bigtable                  endpoints-v2-skeleton  mail       remote-server
cloudsql                 firebase-backend  mailgun    requests
datastore                firebase-event-listener-python  mailjet   search
datastore-indexes         firebase-event-proxy  metadata  memcache
datastore-indexes-exploding firebase-event-proxy-README.md  multitenancy  sendgrid
datastore-indexes-perfect firebase-tictactoe  spanner   sparkjava-helloworld
datastore-schedule-export gaainfo          oauth2    static-files
endpoints-v2-backend     guestbook-cloud-datastore  pubsub   users
geminiofficial73@cloudshell:~/java-docs-samples/appengine-java8 (exercise4-396606)$
```

A blue tooltip box is overlaid on the right side of the terminal window, containing the text: "You are viewing Cloud Shell in full view. Click here to restore Cloud Shell." with a "Got it!" button.

17.cd helloworld

The screenshot shows a Google Cloud Shell terminal window. The command `cd appengine-java8` has been run, and the contents of the directory are listed:

```
geminiofficial73@cloudshell:~ (exercise4-396606)$ cd appengine-java8 (exercise4-396606)$ ls
analytics                endpoints-v2-guice  helloworld  README.md
appidentity               endpoints-v2-migration  iap         remote-client
bigquery                  endpoints-v2-README.md  images     remote-README.md
bigtable                  endpoints-v2-skeleton  mail       remote-server
cloudsql                 firebase-backend  mailgun    requests
datastore                firebase-event-listener-python  mailjet   search
datastore-indexes         firebase-event-proxy  metadata  memcache
datastore-indexes-exploding firebase-event-proxy-README.md  multitenancy  sendgrid
datastore-indexes-perfect firebase-tictactoe  spanner   sparkjava-helloworld
datastore-schedule-export gaainfo          oauth2    static-files
endpoints-v2-backend     guestbook-cloud-datastore  pubsub   users
geminiofficial73@cloudshell:~/java-docs-samples/appengine-java8 (exercise4-396606)$ cd helloworld
```

A red rectangular box highlights the command `cd helloworld` at the bottom of the terminal window.

18.type for Maven clean

mvn clean

The screenshot shows a Google Cloud Shell terminal window titled "Terminal (exercise4-396606)". The terminal output shows the user navigating to the "appengine-java8" directory and running the command "mvn clean". A red box highlights the command "mvn clean". The terminal window has a blue tooltip at the top right stating: "You are viewing Cloud Shell in full view. Click here to restore Cloud Shell." The status bar at the bottom right shows the time as 11:59 AM and the date as 8/21/2023.

```
iplatform contact-center-insights flexible op
appengine-javall container-registry functions op
appengine-javall-bundled-services healthcare
appengine-javall-bundled-services CONTRIBUTING.ad iam pt
appengine-java8 datacatalog iap pv
asset dataflow iot re
anth datalabeling jobs renovate.json textspeech
dataproc kms language retail trace
batch dialogflow langnage run translate
bigquery discoveryengine media SAMPLE_FORMAT.md unittests
bigtable dip media_cdn secretmanager video
build.gradle endpoints memorystore SECURITY.md webrisk
cdn document-ai mediatranslation security-command-center workflows
CODE_OF_CONDUCT.md endpoints mlengine servicedirectory
cloud-sql endpoints mlengine
geminiofficial73@cloudshell:~/java-docs-samples (exercise4-396606)$ cd appengine-java8
geminiofficial73@cloudshell:~/java-docs-samples/appengine-java8 (exercise4-396606)$ ls
analytics endpoints-v2-guide helloworld README.md
appidentity endpoints-v2-migration iap remote-client taskqueues-deferred
bigquery endpoints-v2-README.md images remote-README.md taskqueues-pull
bigtable endpoints-v2-skeleton mail remote-server tasks
cloudsql firebase-backend mailgun requests translate-push
datastore firebase-event-listener-python mailjet search twilio
datastore-indexes firebase-event-proxy README.md metadata spanner urlfetch
datastore-indexes-exploiting firebase-event-proxy-README.md multitenancy users
datastore-indexes-perfect firebase-tictactoe
datastore-schedule-export gaainfo oauth2 springboot-helloworld
endpoints-v2-backend guestbook-cloud-datastore pubsub static-files
geminiofficial73@cloudshell:~/java-docs-samples/appengine-java8 (exercise4-396606)$ cd helloworld
geminiofficial73@cloudshell:~/java-docs-samples/appengine-java8/helloworld (exercise4-396606)$ ls
build.gradle gradlew gradlew.bat pom.xml README.md settings.gradle src
geminiofficial73@cloudshell:~/java-docs-samples/appengine-java8/helloworld (exercise4-396606)$ mvn clean
```

The screenshot shows a Google Cloud Shell terminal window titled "Terminal (exercise4-396606)". The terminal output shows the user navigating to the "appengine-java8" directory and running the command "mvn clean". A red box highlights the command "mvn clean". The terminal window has a blue tooltip at the top right stating: "You are viewing Cloud Shell in full view. Click here to restore Cloud Shell." The status bar at the bottom right shows the time as 12:00 PM and the date as 8/21/2023.

```
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ helloworld ---
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.pom (1.5 kB at 45 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-2.0.6/maven-2.0.6.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/5/maven-parent-5.pom (9.0 kB at 185 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/5/maven-parent-5.pom (15 kB at 726 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/3/apache-3.pom (3.4 kB at 149 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.pom (4.1 kB at 169 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/sonatype/spice/spice-parent/16/spice-parent-16.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/spice/spice-parent-16.pom (8.4 kB at 597 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/sonatype/forge/forge-parent/5/forge-parent-5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/forge/forge-parent/5/forge-parent-5.pom (8.4 kB at 335 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.jar (13 kB at 146 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.jar (13 kB at 69 kB/s)
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 3.055 s
[INFO] Finished at: 2023-08-21T06:30:11Z
[INFO]
geminiofficial73@cloudshell:~/java-docs-samples/appengine-java8/helloworld (exercise4-396606)$
```

19.In cloud shell.type commands to download maven packages

mvn package

mvn package appengine:run

Get started – App Engine – exercise4 java-docs-samples/appengine-jar +

console.cloud.google.com/appengine/start/reception?project=exercise4-396606&organizationId=0&cloudshell=true

Start your free trial with \$300 in credit. Don't worry – you won't be charged if you run out of credit. [Learn more](#)

DISMISS START FREE

Google Cloud exercise4 Search (/) for resources, docs, products and ... 2 P

CLOUD SHELL Terminal (exercise4-396606) Open editor

```
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ helloworld ---
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.pom (1.9 kB at 45 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven/2.0.6/maven-2.0.6.pom (9.0 kB at 185 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/5/maven-parent-5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/5/maven-parent-5.pom (15 kB at 726 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/3/pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/3/pom (3.4 kB at 149 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.pom (4.1 kB at 1 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/sonatype/spice/spice-parent/16/spice-parent-16.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/spice/spice-parent/16/spice-parent-16.pom (8.4 kB at 597 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/sonatype/forge/forge-parent/5/forge-parent-5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/forge/forge-parent/5/forge-parent-5.pom (0.4 kB at 335 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.jar
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-api/2.0.6/maven-plugin-api-2.0.6.jar (13 kB at 146 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0/plexus-utils-3.0.jar (226 kB at 696 kB/s)
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.055 s
[INFO] Finished at: 2023-08-21T06:30:11Z
[INFO] -----
```

geminiofficial73@cloudshell:~/java-docs-samples/appengine-java8/helloworld (exercise4-396606)\$ mvn package

12:00 PM 8/21/2023

Get started – App Engine – exercise4 java-docs-samples/appengine-jar +

console.cloud.google.com/appengine/start/reception?project=exercise4-396606&organizationId=0&cloudshell=true

Start your free trial with \$300 in credit. Don't worry – you won't be charged if you run out of credit. [Learn more](#)

DISMISS START FREE

Google Cloud exercise4 Search (/) for resources, docs, products and ... 2 P

CLOUD SHELL Terminal (exercise4-396606) Open editor

```
6.jar (85 kB at 185 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-filtering/3.3.1/maven-filtering-3.3.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-compress/1.23.0/commons-compress-1.23.0.jar (1.1 MB at 2.1 MB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/sonatype/plexus/plexus-build-api/0.0.7/plexus-build-api-0.0.7.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/plexus/plexus-build-api/0.0.7/plexus-build-api-0.0.7.jar (8.5 kB at 16 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-mapping/3.0.0/maven-mapping-3.0.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-filtering/3.3.1/maven-filtering-3.3.1.jar (55 kB at 99 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-mapping/3.0.0/maven-mapping-3.0.0.jar (11 kB at 19 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/plexus/plexus-utils/3.5.1/plexus-utils-3.5.1.jar (269 kB at 398 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/commons-io/commons-io/2.13.0/commons-io-2.13.0.jar (484 kB at 643 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/com/github/luben/zstd-jni/1.5.5-2/zstd-jni-1.5.5-2.jar (5.9 MB at 6.7 MB/s)
[INFO] Packaging webapp
[INFO] Assembling webapp [helloworld] in [/home/geminiofficial73/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/geminiofficial73/java-docs-samples/appengine-java8/helloworld/src/main/webapp]
[INFO] Building war: /home/geminiofficial73/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 39.139 s
[INFO] Finished at: 2023-08-21T06:31:38Z
[INFO] -----
```

geminiofficial73@cloudshell:~/java-docs-samples/appengine-java8/helloworld (exercise4-396606)\$ mvn package appengine:run

12:01 PM 8/21/2023

Get started – App Engine – exercise4 java-docs-samples/appengine-jar

console.cloud.google.com/appengine/start/reception?project=exercise4-396606&organizationId=0&cloudshell=true

Start your free trial with \$300 in credit. Don't worry – you won't be charged if you run out of credit. [Learn more](#)

DISMISS START FREE

Google Cloud exercise4 Search (/) for resources, docs, products and ...

CLOUD SHELL Terminal (exercise4-396606) Open editor

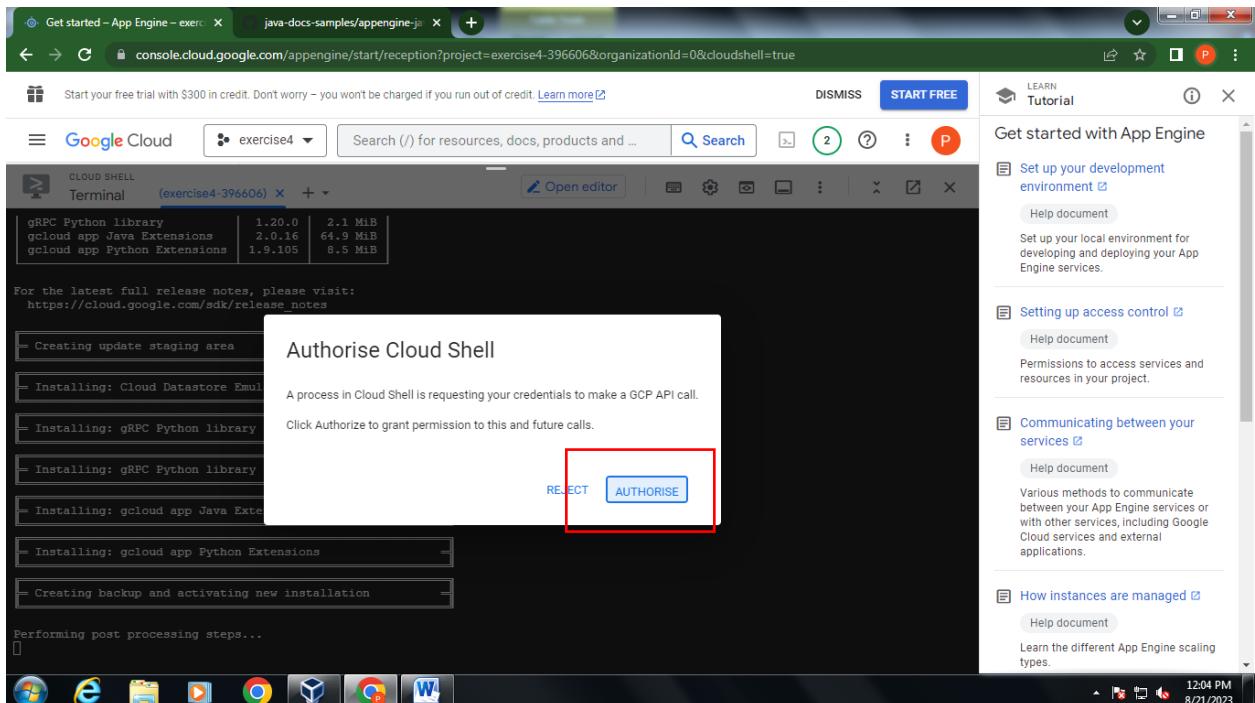
```

6.jar (85 kB at 185 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/shared/maven-filtering/3.3.1/maven-filtering-3.3.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-compress/1.23.0/commons-compress-1.23.0.jar (1.1 MB at 2.1 MB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/sonatype/plexus/plexus-build-api/0.0.7/plexus-build-api-0.0.7.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/plexus/plexus-build-api/0.0.7/plexus-build-api-0.0.7.jar (8.5 kB at 16 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-mapping/3.0.0/maven-mapping-3.0.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-filtering/3.3.1/maven-filtering-3.3.1.jar (55 kB at 99 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-mapping/3.0.0/maven-mapping-3.0.0.jar (1 kB at 19 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.5.1/plexus-utils-3.5.1.jar (269 kB at 398 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/commons-io/2.13.0/commons-io-2.13.0.jar (484 kB at 643 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/com/github/luben/zstd-jni/1.5.5-2/zstd-jni-1.5.5-2.jar (5.9 MB at 6.7 MB/s)
[INFO] Packaging webapp
[INFO] Assembling webapp [helloworld] in [/home/geminiofficial73/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/geminiofficial73/java-docs-samples/appengine-java8/helloworld/src/main/webapp]
[INFO] Building war: /home/geminiofficial73/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT.war
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 39.139 s
[INFO] Finished at: 2023-08-21T06:31:38Z
[INFO]
```

geminiofficial73@cloudshell:~/java-docs-samples/appengine-java8/helloworld (exercise4-396606) \$ mvn package appengine:run

12:02 PM 8/21/2023

20.click authorize to get an authorization



```

[INFO] GCloud: 2023-08-21 06:35:08.265:WARN:oeja.AnnotationParser:qtp770911223-14: javax.annotation.meta.TypeQualifierNickname scanned from multiple locations: jarfile:///home/geminiofficial173/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/jar305-3.0.2.jar!/javax/annotation/meta/TypeQualifierNickname.class, jarfile:///home/geminiofficial173/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/appengine-api-1.0-sdk-2.0.17.jar!/javax/annotation/meta/TypeQualifierNickname.class
[INFO] GCloud: 2023-08-21 06:35:08.265:WARN:oeja.AnnotationParser:qtp770911223-14: javax.annotation.meta.TypeQualifierValidator scanned from multiple locations: jarfile:///home/geminiofficial173/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/jar305-3.0.2.jar!/javax/annotation/meta/TypeQualifierValidator.class, jarfile:///home/geminiofficial173/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/appengine-api-1.0-sdk-2.0.17.jar!/javax/annotation/meta/TypeQualifierValidator.class
[INFO] GCloud: 2023-08-21 06:35:08.265:WARN:oeja.AnnotationParser:qtp770911223-14: javax.annotation.meta.TypeQualifierNickname When scanned from multiple locations: jarfile:///home/geminiofficial173/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/jar305-3.0.2.jar!/javax/annotation/meta/When.class, jarfile:///home/geminiofficial173/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/appengine-api-1.0-sdk-2.0.17.jar!/javax/annotation/meta/When.class
[INFO] GCloud: 2023-08-21 06:35:08.297:INFO:oeja.AnnotationConfiguration:main: Scanning elapsed time=3096ms
[INFO] GCloud: 2023-08-21 06:35:08.990:INFO:oejh.ContextHandler:main: Started c.g.a.t.d.DevAppEngineWebAppContext@68ead591{/home/geminiofficial173/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT}
[INFO] GCloud: 2023-08-21 06:35:08.993:INFO:oejs.AbstractConnector:main: Started NetworkTrafficSelectChannelConnector@482cd91f{HTTP/1.1, (http/1.1)}(localhost:8080)
[INFO] GCloud: 2023-08-21 06:35:09.014:INFO:oejs.Server:main: Started #6981ms
[INFO] GCloud: Aug 21, 2023 6:35:09 AM com.google.appengine.tools.development.AbstractModule startup
[INFO] GCloud: INFO: Module instance default is running at http://localhost:8080/
[INFO] GCloud: Aug 21, 2023 6:35:09 AM com.google.appengine.tools.development.AbstractModule startup
[INFO] GCloud: INFO: The admin console is running at http://localhost:8080/ ah/admin
[INFO] GCloud: Aug 21, 2023 6:35:09 AM com.google.appengine.tools.development.DevAppServerImpl doStart
[INFO] GCloud: INFO: Dev App Server is now running

```

21. Now the device server from Maven is running

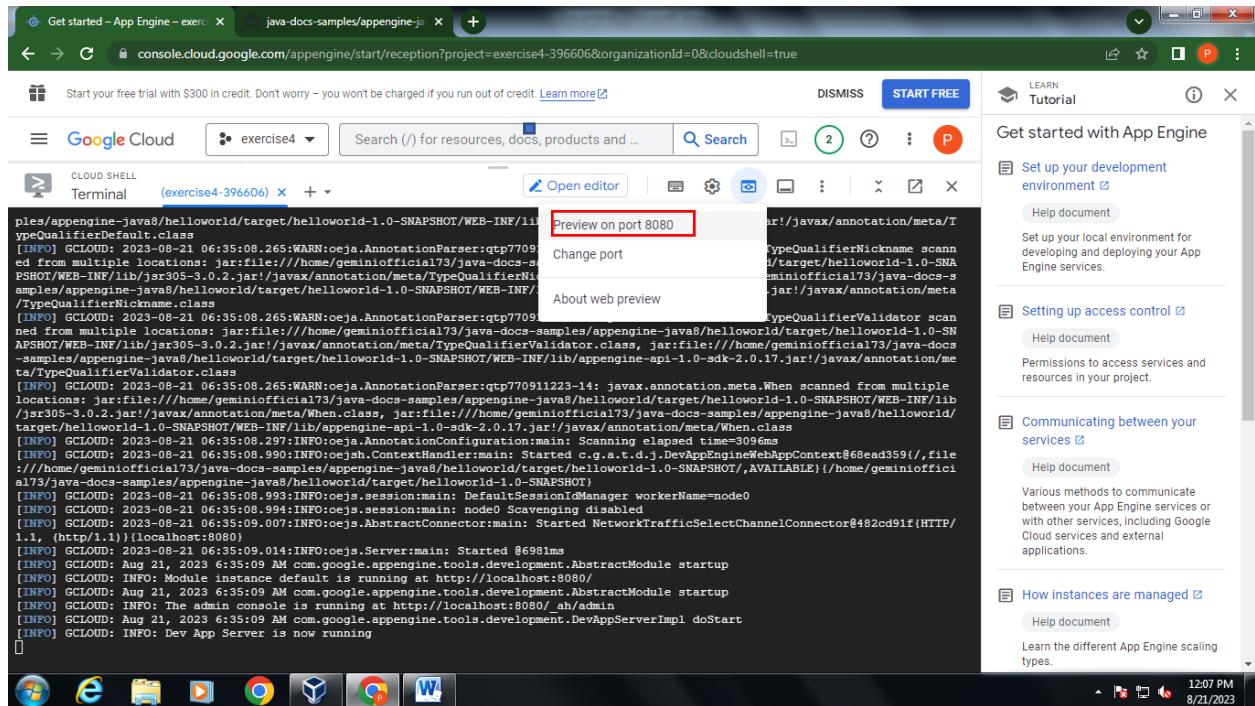
The screenshot shows the same terminal output as the previous one, but with two red boxes highlighting specific areas. One box highlights the 'Web preview' button in the terminal toolbar, and the other box highlights the URL 'http://localhost:8080' in the browser's address bar.

```

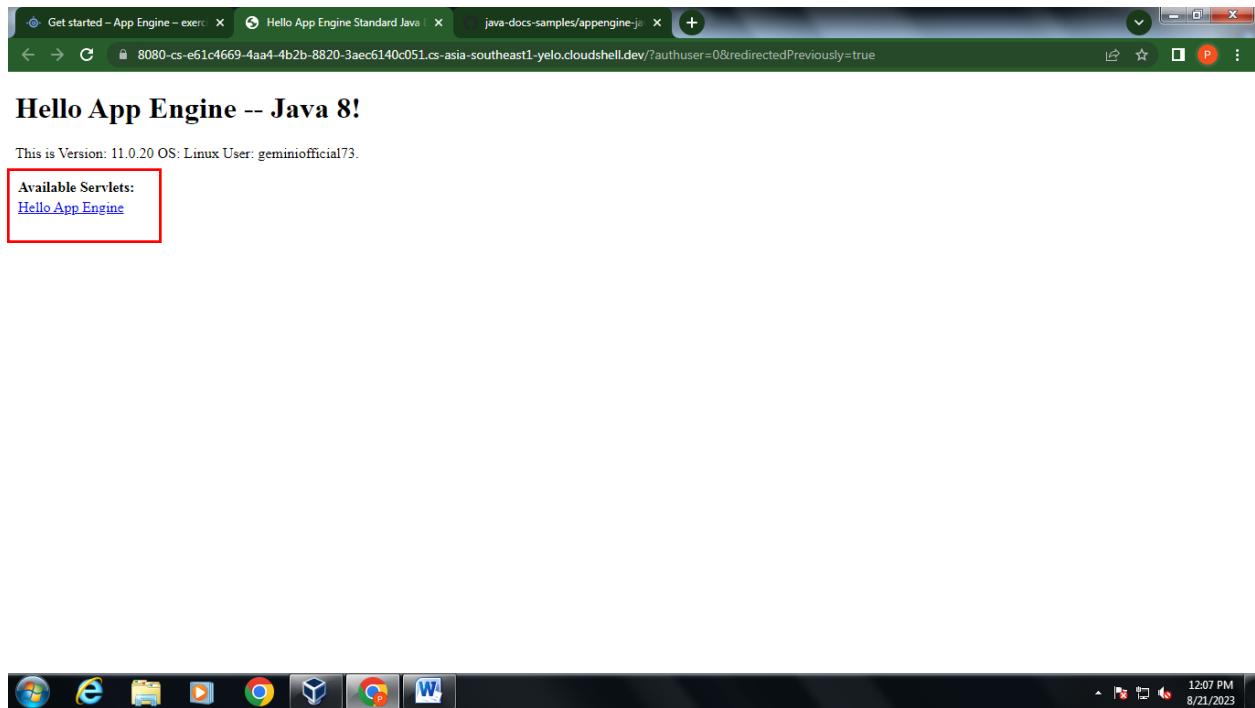
[INFO] GCloud: 2023-08-21 06:35:08.265:WARN:oeja.AnnotationParser:qtp770911223-14: javax.annotation.meta.TypeQualifierNickname scanned from multiple locations: jarfile:///home/geminiofficial173/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/jar305-3.0.2.jar!/javax/annotation/meta/TypeQualifierNickname.class, jarfile:///home/geminiofficial173/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/appengine-api-1.0-sdk-2.0.17.jar!/javax/annotation/meta/TypeQualifierNickname.class
[INFO] GCloud: 2023-08-21 06:35:08.265:WARN:oeja.AnnotationParser:qtp770911223-14: javax.annotation.meta.TypeQualifierValidator scanned from multiple locations: jarfile:///home/geminiofficial173/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/jar305-3.0.2.jar!/javax/annotation/meta/TypeQualifierValidator.class, jarfile:///home/geminiofficial173/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/appengine-api-1.0-sdk-2.0.17.jar!/javax/annotation/meta/TypeQualifierValidator.class
[INFO] GCloud: 2023-08-21 06:35:08.265:WARN:oeja.AnnotationParser:qtp770911223-14: javax.annotation.meta.TypeQualifierNickname When scanned from multiple locations: jarfile:///home/geminiofficial173/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/jar305-3.0.2.jar!/javax/annotation/meta/When.class, jarfile:///home/geminiofficial173/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/appengine-api-1.0-sdk-2.0.17.jar!/javax/annotation/meta/When.class
[INFO] GCloud: 2023-08-21 06:35:08.297:INFO:oeja.AnnotationConfiguration:main: Scanning elapsed time=3096ms
[INFO] GCloud: 2023-08-21 06:35:08.990:INFO:oejh.ContextHandler:main: Started c.g.a.t.d.DevAppEngineWebAppContext@68ead591{/home/geminiofficial173/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT}
[INFO] GCloud: 2023-08-21 06:35:08.993:INFO:oejs.AbstractConnector:main: Started NetworkTrafficSelectChannelConnector@482cd91f{HTTP/1.1, (http/1.1)}(localhost:8080)
[INFO] GCloud: 2023-08-21 06:35:09.014:INFO:oejs.Server:main: Started #6981ms
[INFO] GCloud: Aug 21, 2023 6:35:09 AM com.google.appengine.tools.development.AbstractModule startup
[INFO] GCloud: INFO: Module instance default is running at http://localhost:8080/
[INFO] GCloud: Aug 21, 2023 6:35:09 AM com.google.appengine.tools.development.AbstractModule startup
[INFO] GCloud: INFO: The admin console is running at http://localhost:8080/ ah/admin
[INFO] GCloud: Aug 21, 2023 6:35:09 AM com.google.appengine.tools.development.DevAppServerImpl doStart
[INFO] GCloud: INFO: Dev App Server is now running

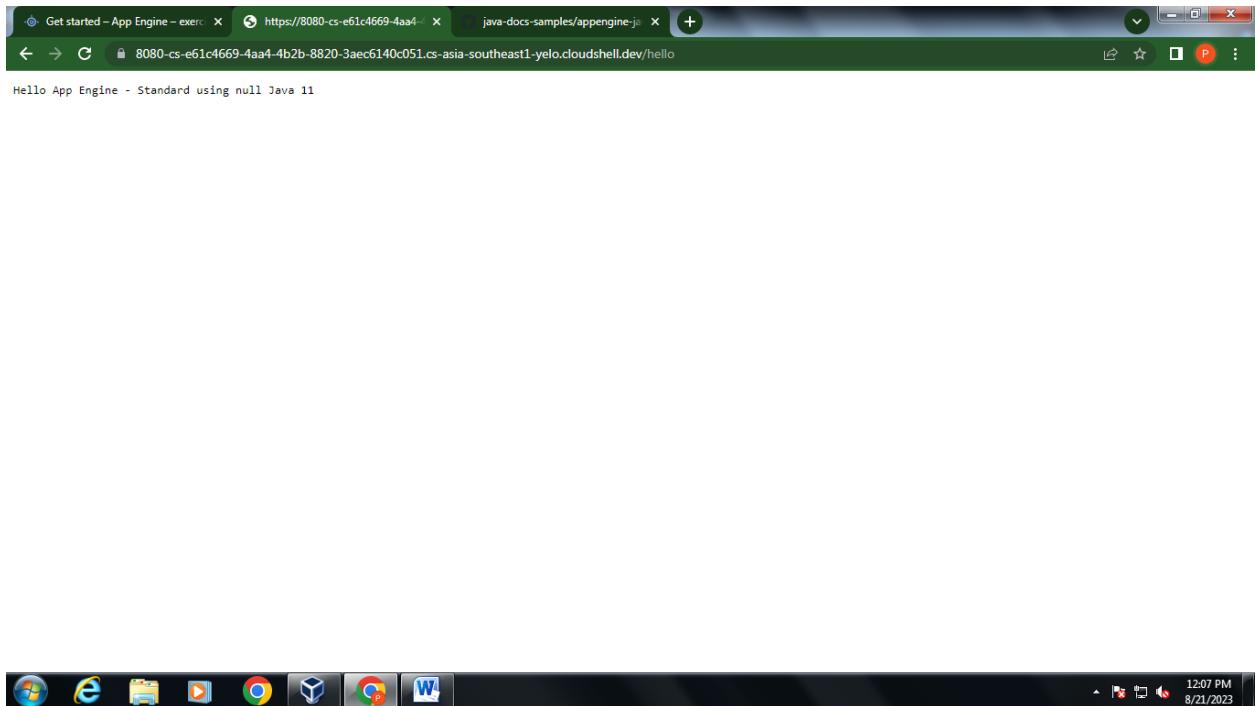
```

22.Click web preview and select “ preview on port 8080”



23.The java web application is launched in cloud platform successfully





RESULT:

Thus the project with the help of GAE launcher to launch web application Helloworld in java is executed successfully.

EX.NO: 5

DATE:

SIMULATE A CLOUD SCENARIO USING CLOUDSIM AND RUN A SCHEDULING ALGORITHM THAT IS NOT PRESENTIN CLOUDSIM.

AIM:

To execute the program by simulate a cloud scenario using cloudsim and run a scheduling program that is not present in cloudsim.

PROCEDURE:

Step 1:

Java installation

Install java setup

C:\Program Files\Java\jdk-13.0.2

Set path variable for java

Goto ->Edit the System Environment variables ->select Environment variable
-> select path in system variable -> edit ->new -> C:\Program Files\Java\jdk-13.0.2\bin -> ok

Step 2:

Download CloudSim 3.0.3

Goto the link

<https://github.com/Cloudslab/cloudsim/releases>

<https://github.com/Cloudslab/cloudsim/releases/download/cloudsim-3.0.3/cloudsim-3.0.3.zip>

Step 3:

Eclipse IDE installation

<https://www.eclipse.org/downloads/>

<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2023-06/R/eclipse-inst-jre-win64.exe>

Step 4:

Run Cloudsim in Eclipse

Build a java project(Scheduling algorithm) with CloudSiM 3.0.3 Folder.

ECLIPSE IDE INSTALLATION STEPS:

1. Download the Eclipse Installer.

Download Eclipse Installer from <http://www.eclipse.org/downloads>

Eclipse is hosted on many mirrors around the world. Please select the one closest to you and start to download the Installer.

2. Start the Eclipse Installer executable.

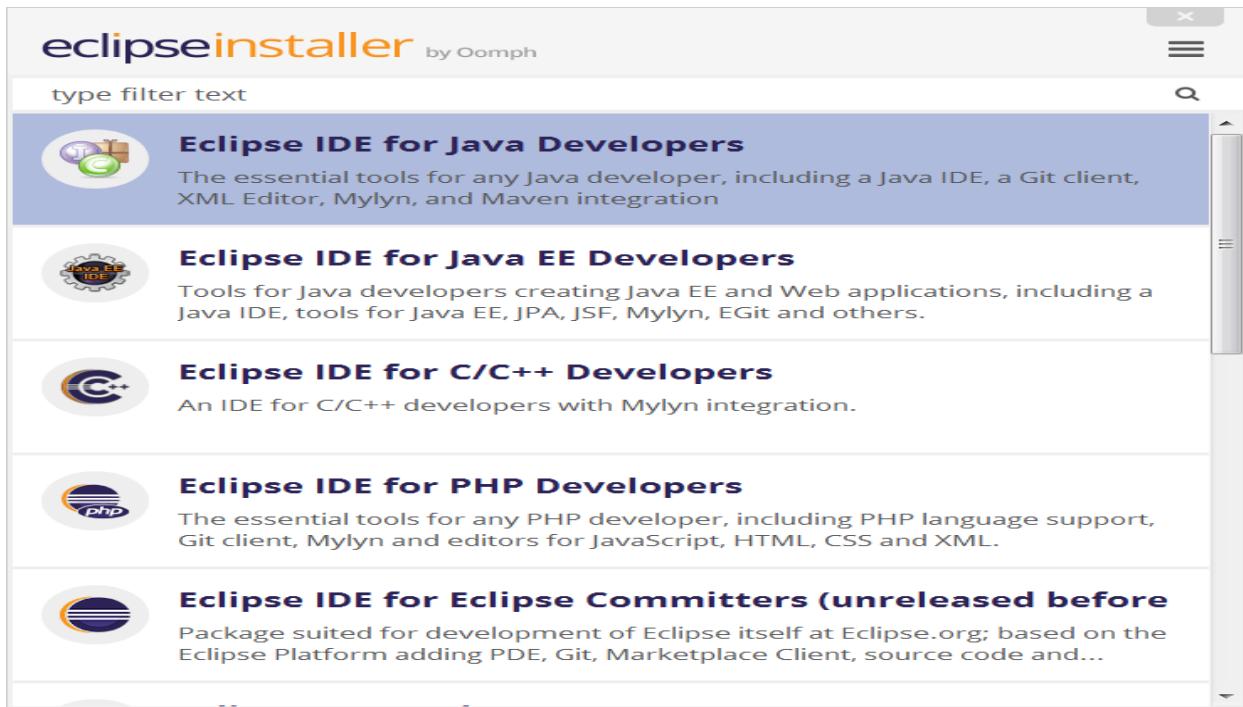
For Windows users, after the Eclipse Installer executable has finished downloading it should be available in your download directory. Start the Eclipse Installer executable. You

may get a security warning to run this file. If the Eclipse Foundation is the Publisher, you are good to select Run.



3. Select the package to install.

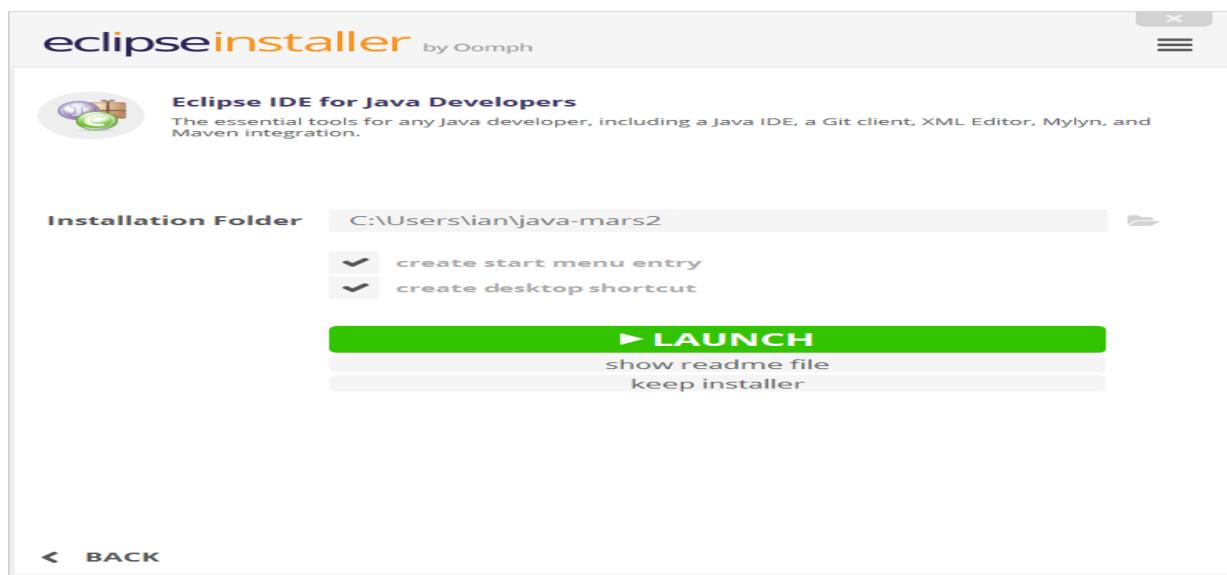
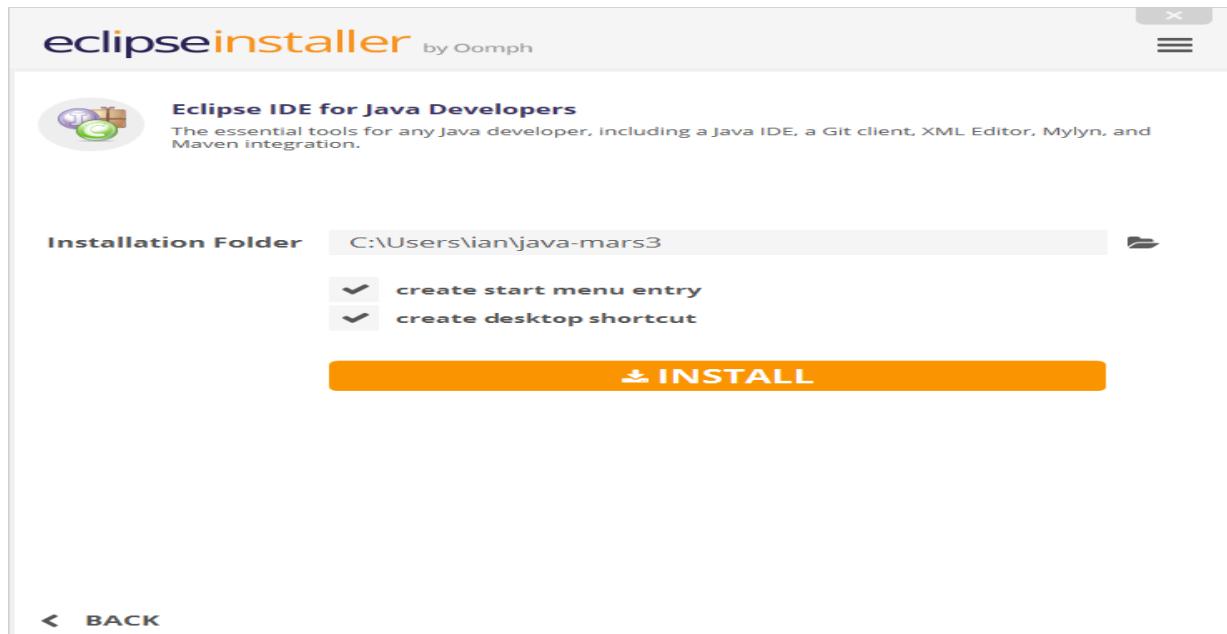
The new Eclipse Installer shows the packages available to Eclipse users. You can search for the package you want to install or scroll through the list.
Select and click on the package you want to install.



4. Select your installation folder

Specify the folder where you want Eclipse to be installed. The default folder will be in your User directory.

Select the 'Install' button to begin the installation.

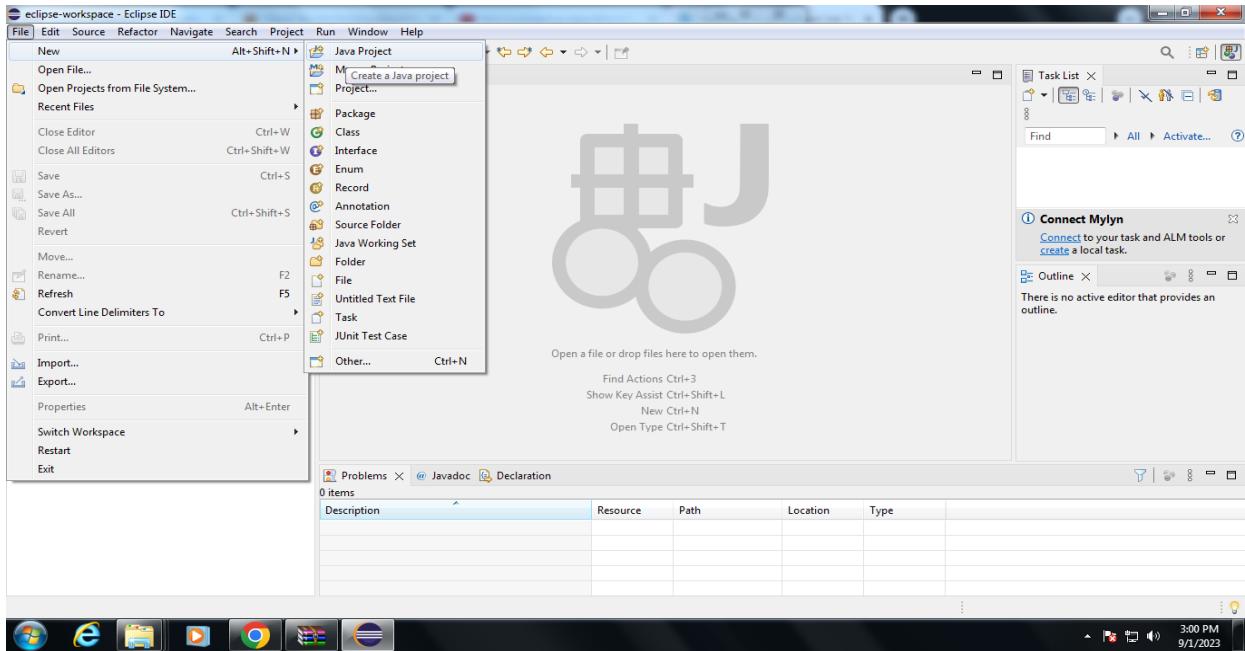


5. Launch Eclipse.

Once the installation is complete you can now launch Eclipse. The Eclipse Installer has done its work.

Run cloudsim in Eclipse:

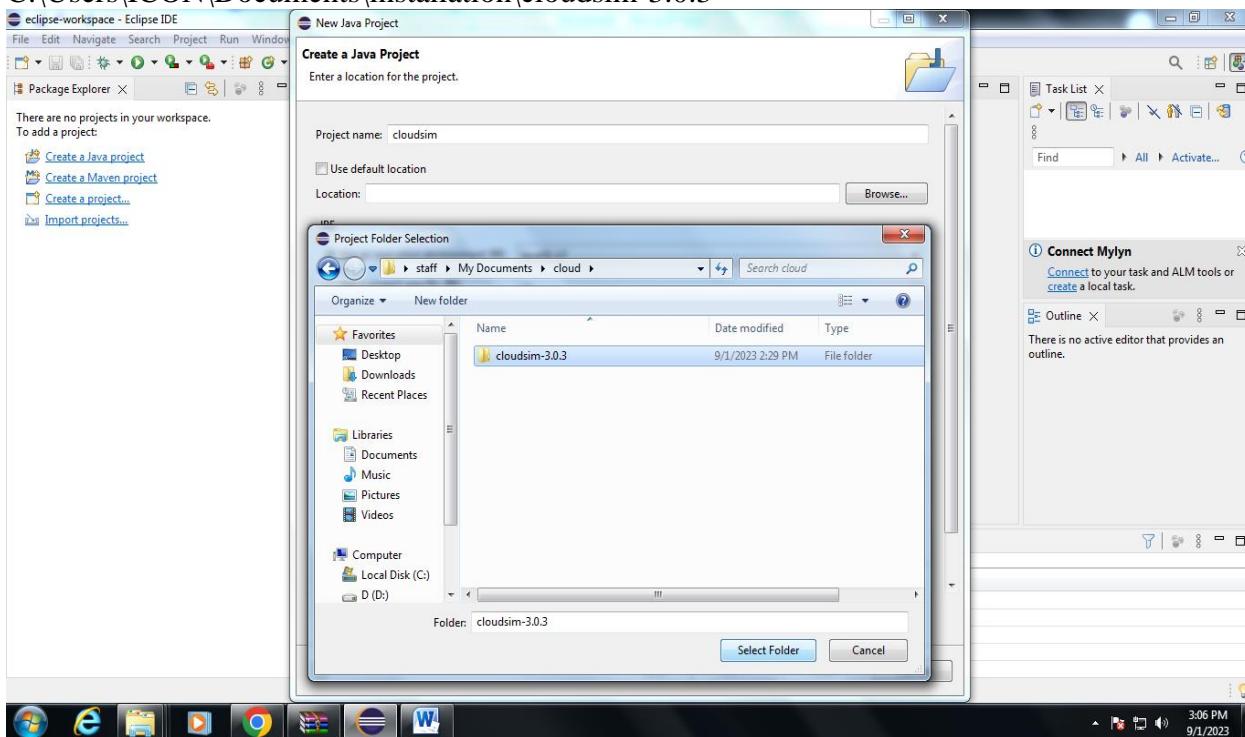
File -> new -> java project

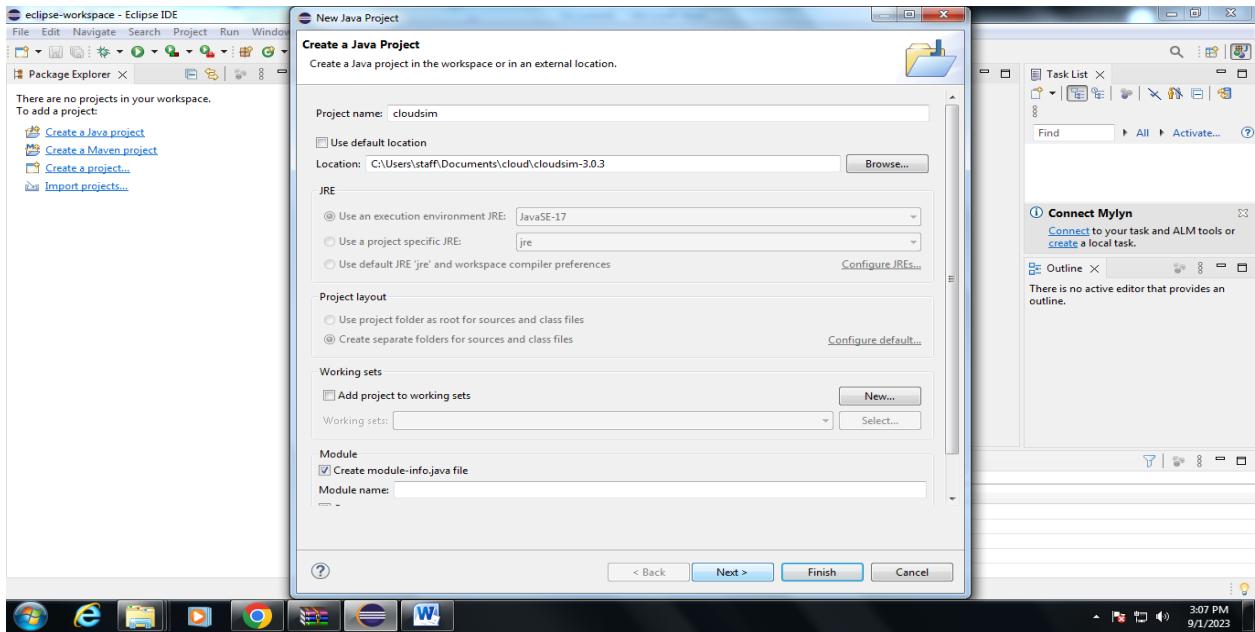


Select location

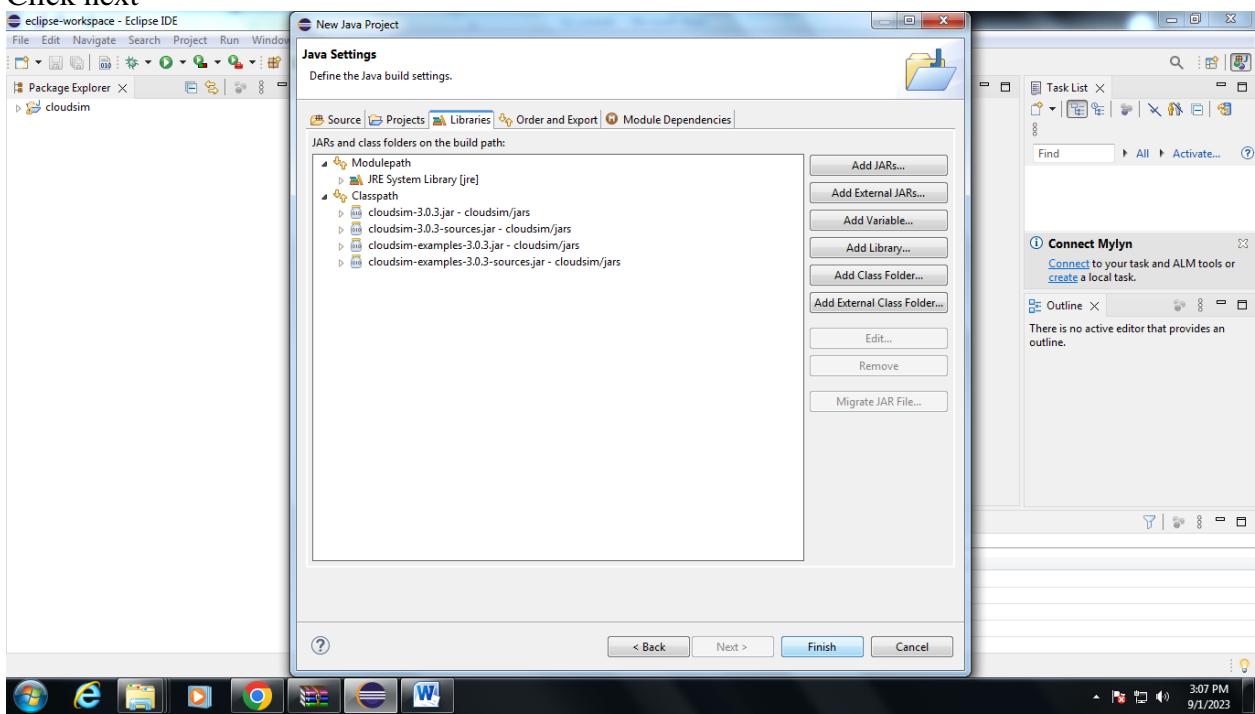
Set this path

C:\Users\ICON\Documents\installation\cloudsim-3.0.3





Click next

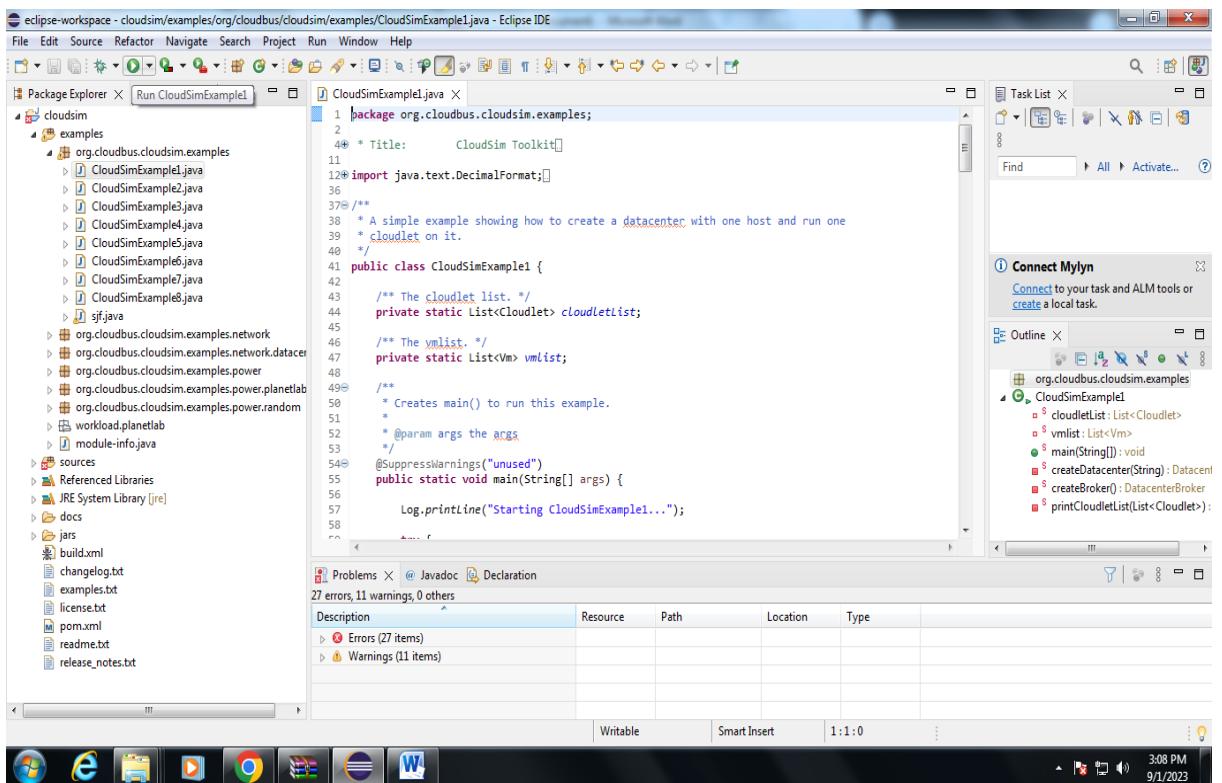
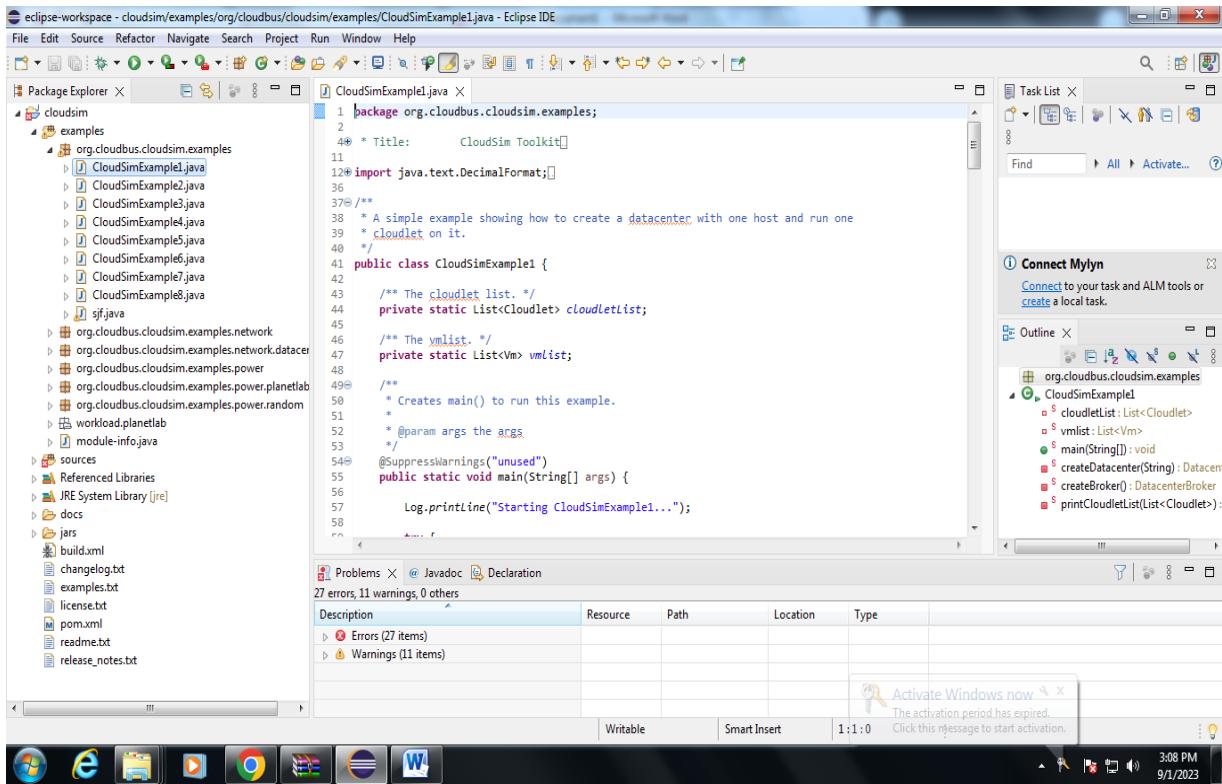


Cloudsim libraries are added over here

Click finish

In example of cloudsim folder, run the first program which is present in cloudsim

CloudSimExample1.java



OUTPUT: (PROGRAM ALREADY IN CLOUDSIM)

```
CloudSimExample1.java
40
41 public class CloudSimExample1 {
42     /* The cloudlet list */
43     private static List<Cloudlet> cloudletList;
44
45     /* The vmlist */
46     private static List<Vm> vmList;
47
48
49     /**
50      * @param args
51      */
52     public static void main(String[] args) {
53         // Create Broker
54         Broker broker = new Broker();
55
56         // Create Datacenter
57         Datacenter datacenter = new Datacenter();
58
59         // Create VM
60         Vm vm = new Vm("VM #0");
61
62         // Create Cloudlet
63         Cloudlet cloudlet = new Cloudlet("Cloudlet #0");
64
65         // Add cloudlet to cloudlet list
66         cloudletList.add(cloudlet);
67
68         // Add VM to vmlist
69         vmList.add(vm);
70
71         // Start simulation
72         broker.start();
73
74         // Execute cloudlets
75         broker.execute();
76
77         // Shut down simulation
78         broker.shutdown();
79
80         System.out.println("Simulation completed.");
81     }
82 }
```

Output Log:

```
CloudSimExample1 [Java Application] C:\Users\staff\p2\pool\plugins\org.eclipse.jst.java.core\1.17.0.v20230831-1047\jre\bin\javaw.exe (Sep 1, 2023, Starting CloudSimExample1...
Initialising...
Starting CloudSimExample1...
Datacenter #1 is starting...
Broker is starting...
Entities started.
Broker received Resource List received with 1 resource(s)
0:0: Broker: Trying to Create VM #0 in Datacenter #0
0:1: Broker: VM #0 has been created in Datacenter #2, Host #0
0:2: Broker: Sensor Cloudlet #0 to VM #0
400:1: Broker: Cloudlet #0 received
400:1: Broker: All Cloudlets executed. Finishing...
400:1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter #1 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

=====
Cloudlet ID STATUS Data center ID VM ID Time Start Time Finish Time
----- -----
CloudSimExample1 finished!
```

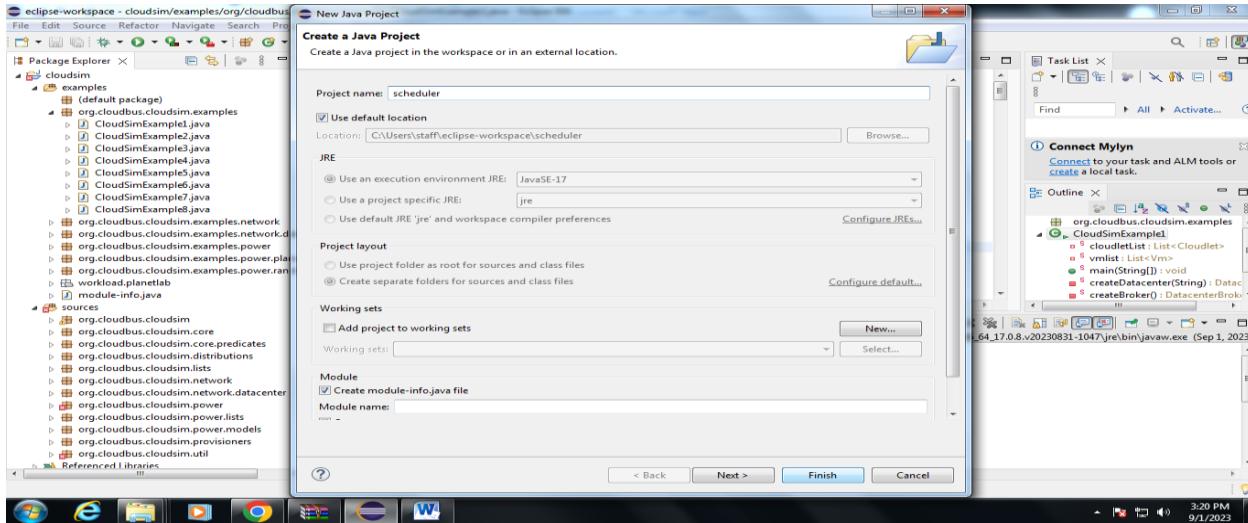
=====

CloudSimExample1 finished!

CloudSimExample1.java successfully executed

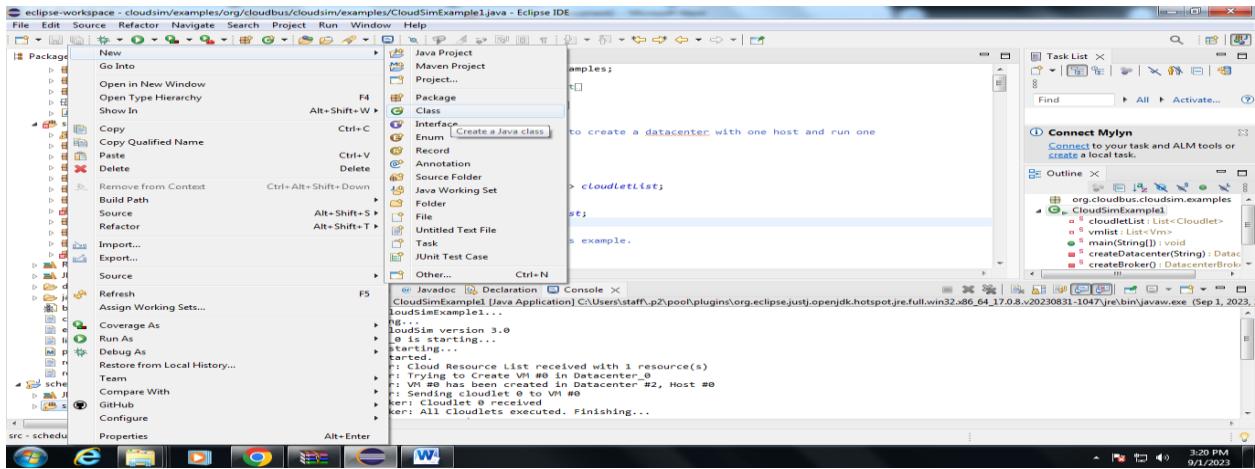
Next we have to add and execute scheduling program in java which is not already present in cloudsim folder

Goto -> file -> new -> java project -> give the project name as scheduler



Click finish

Next we have to create a java class for scheduler program under src in that scheduler java project

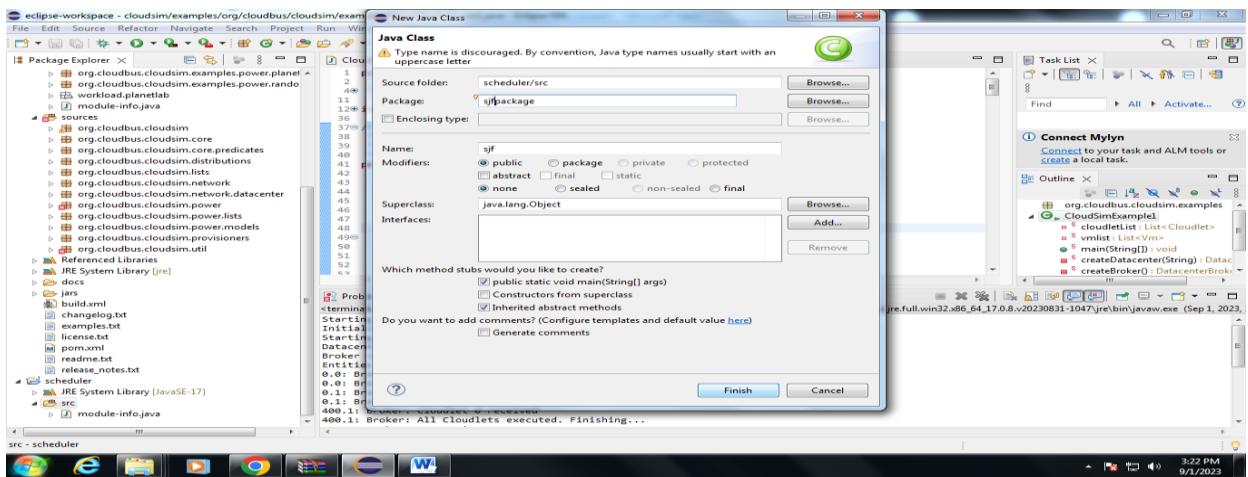


Here we should give package name as sjfpackage

Give the java program name “ sjf ”

select the option “ public static void main(String{} args) ”

Click finish



Open that java class program sjf.java which is created

Type the scheduler program in that file

Run that program

sjf.java:

```
import java.util.*;
public class sjf {
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println ("enter no of process:");
        int n = sc.nextInt();
```

```

int pid[] = new int[n];
int at[] = new int[n]; // at means arrival time
int bt[] = new int[n]; // bt means burst time
int ct[] = new int[n]; // ct means complete time
int ta[] = new int[n]; // ta means turn around time
int wt[] = new int[n]; // wt means waiting time
int f[] = new int[n]; // f means it is flag it checks process is completed or not
int st=0, tot=0;
float avgwt=0, avgta=0;

for(int i=0;i<n;i++)
{
    System.out.println ("enter process " + (i+1) + " arrival time:");
    at[i] = sc.nextInt();
    System.out.println ("enter process " + (i+1) + " brust time:");
    bt[i] = sc.nextInt();
    pid[i] = i+1;
    f[i] = 0;
}
boolean a = true;
while(true)
{
    int c=n, min=999;
    if (tot == n) // total no of process = completed process loop will be terminated
        break;
    for (int i=0; i<n; i++)
    {
    /*
     * If i'th process arrival time <= system time and its flag=0 and burst<min
     * That process will be executed first
     */
    if ((at[i] <= st) && (f[i] == 0) && (bt[i]<min))
    {
        min=bt[i];
        c=i;
    }
    }
    /* If c==n means c value can not updated because no process arrival time< system time so we
    increase the system time */
    if (c==n)
        st++;
    else
    {
        ct[c]=st+bt[c];
        st+=bt[c];
        ta[c]=ct[c]-at[c];
        wt[c]=ta[c]-bt[c];
        f[c]=1;
    }
}

```

```

tot++;
}
}
System.out.println("\npid arrival brust complete turn waiting");
for(int i=0;i<n;i++)
{
avgwt+= wt[i];
avgta+= ta[i];
System.out.println(pid[i]+\t"+at[i]+\t"+bt[i]+\t"+ct[i]+\t"+ta[i]+\t"+wt[i]);
}
System.out.println ("\naverage tat is "+(float)(avgta/n));
System.out.println ("average wt is "+(float)(avgwt/n));
sc.close();
}
}

```

OUTPUT 1: (WHICH IS NOT ADDED IN CLOUDSIM)

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure under "scheduler/src/sjfpackage/sjf.java".
- CloudSimExample.java:** The code is as follows:


```

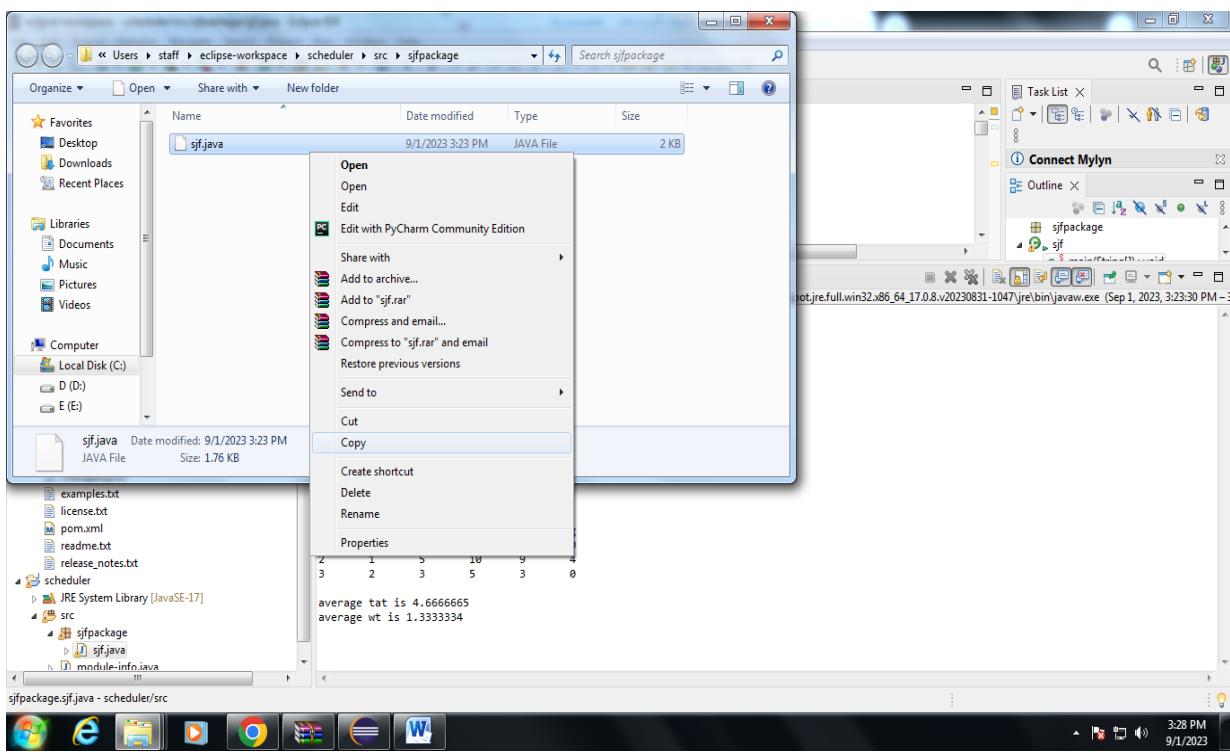
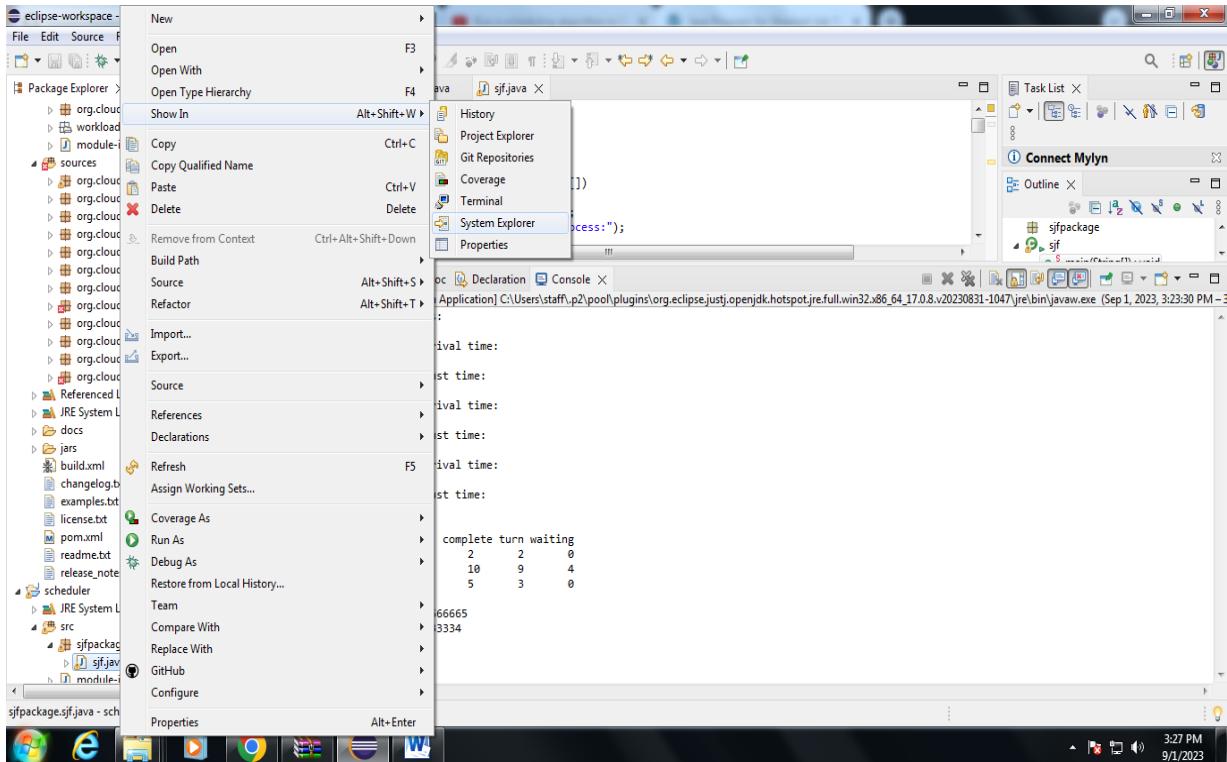
1 package sjfpackage;
2 import java.util.*;
3
4 public class sjf {
5     public static void main(String args[]) {
6         Scanner sc = new Scanner(System.in);
7         System.out.println ("enter no of process:");
8         int n = sc.nextInt();
9         for (int i=1; i<=n; i++) {
10             System.out.print("enter process "+i+" arrival time: ");
11             int arrTime = sc.nextInt();
12             System.out.print("enter process "+i+" brust time: ");
13             int brustTime = sc.nextInt();
14             System.out.println();
15         }
16         System.out.println("pid arrival brust complete turn waiting");
17         for (int i=1; i<=n; i++) {
18             int pid = i;
19             int arrival = 0;
20             int brust = 0;
21             int complete = 0;
22             int turnWaiting = 0;
23             if (i == 1) {
24                 arrival = arrTime;
25                 brust = brustTime;
26                 complete = arrival + brust;
27                 turnWaiting = 0;
28             } else {
29                 arrival = arrTime;
30                 brust = brustTime;
31                 complete = arrival + brust;
32                 turnWaiting = brust;
33             }
34             System.out.print(pid + "\t" + arrival + "\t" + brust + "\t" + complete + "\t" + turnWaiting);
35         }
36         System.out.println();
37         System.out.println("average tat is "+(float)(avgta/n));
38         System.out.println("average wt is "+(float)(avgwt/n));
39     }
40 }
      
```
- Console View:** Displays the program's output:


```

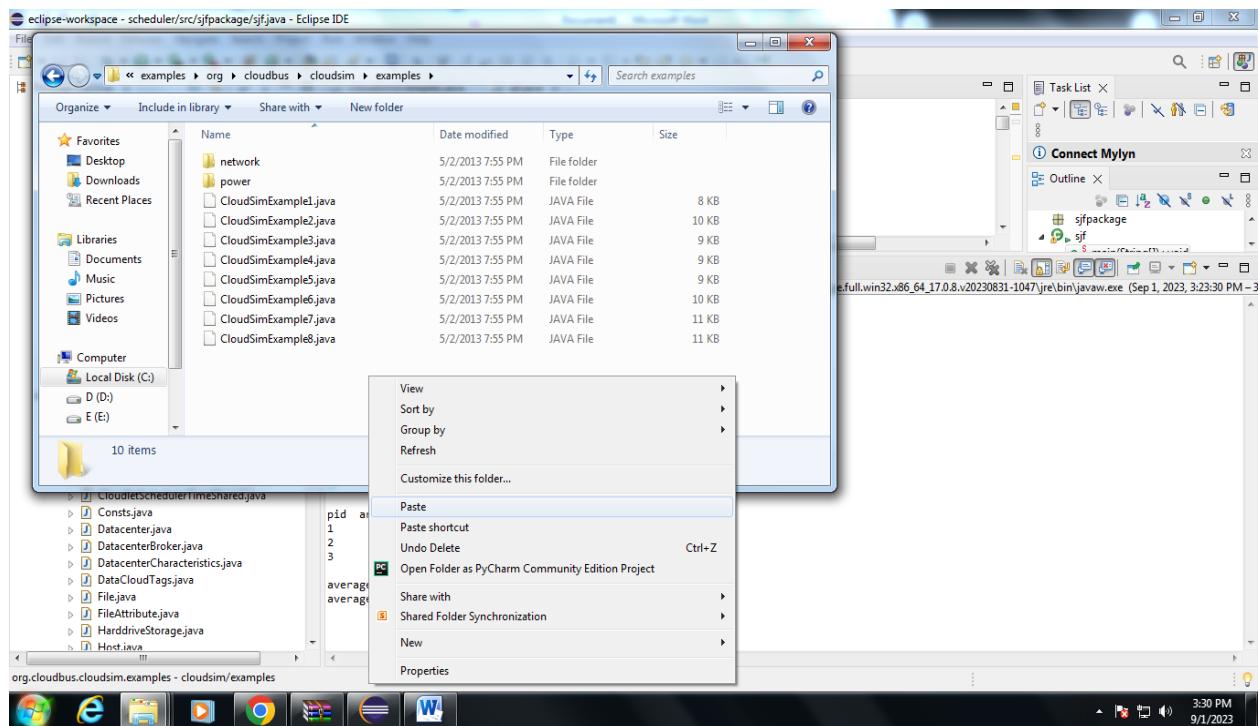
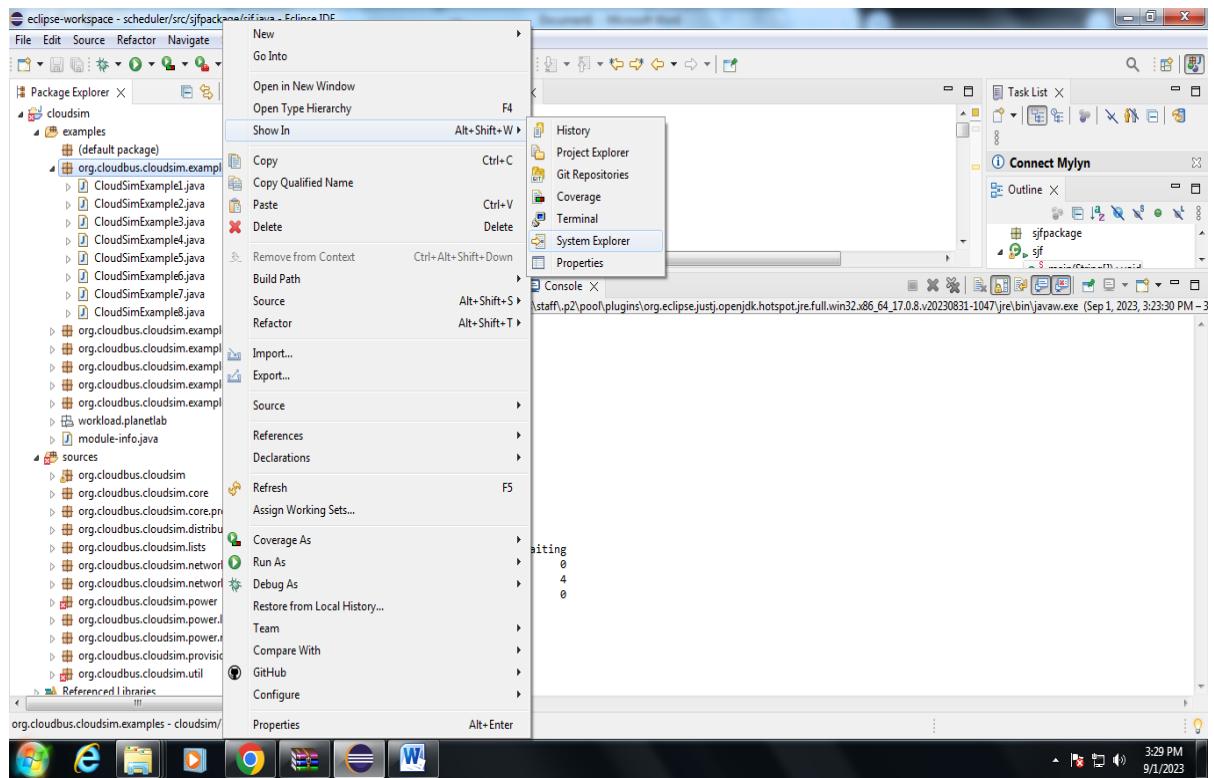
enter no of process:
3
enter process 1 arrival time:
1
enter process 1 brust time:
2
enter process 2 arrival time:
1
enter process 2 brust time:
5
enter process 3 arrival time:
2
enter process 3 brust time:
3
pid arrival brust complete turn waiting
1 0 2 2 0
2 1 5 10 9 4
3 2 3 5 3 0
average tat is 4.6666665
average wt is 1.3333334
      
```
- Task List:** Shows a single task named "Connect Mylyn".
- Bottom Status Bar:** Shows the date and time as "3:26 PM 9/1/2023".

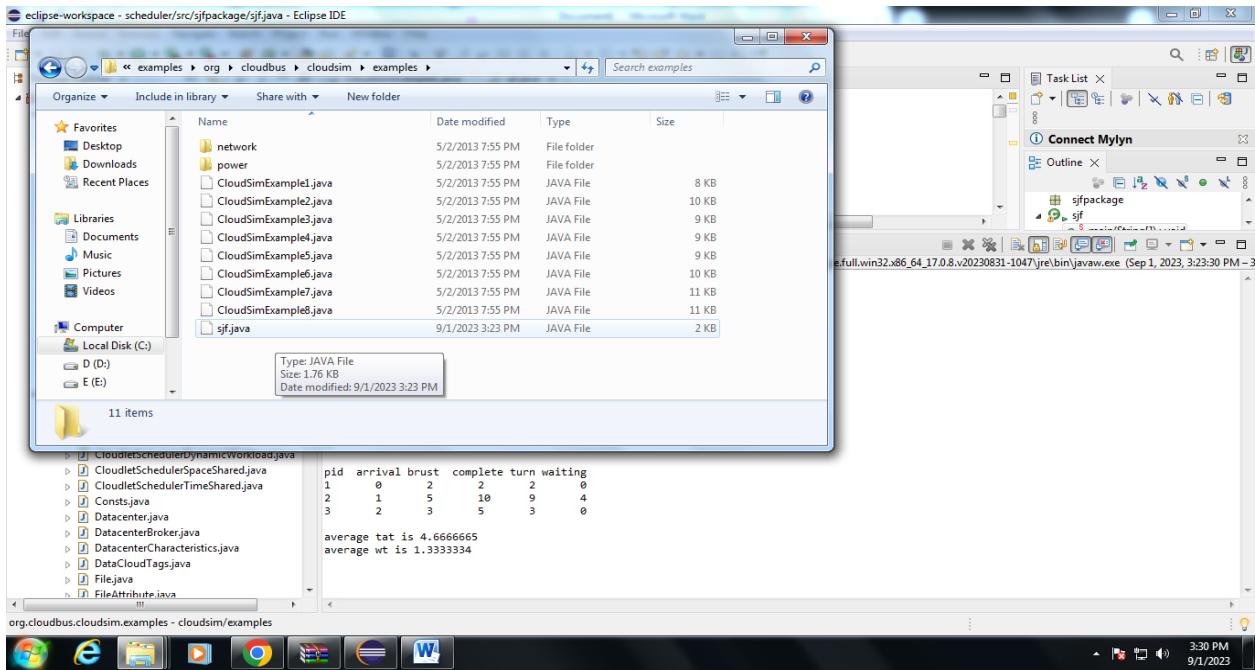
The scheduler program we have to run inside the cloudsim so open that folder by clicking

Show in -> system explorer.



Copy that sjf.java and paste under the cloudsim folder, example packages.





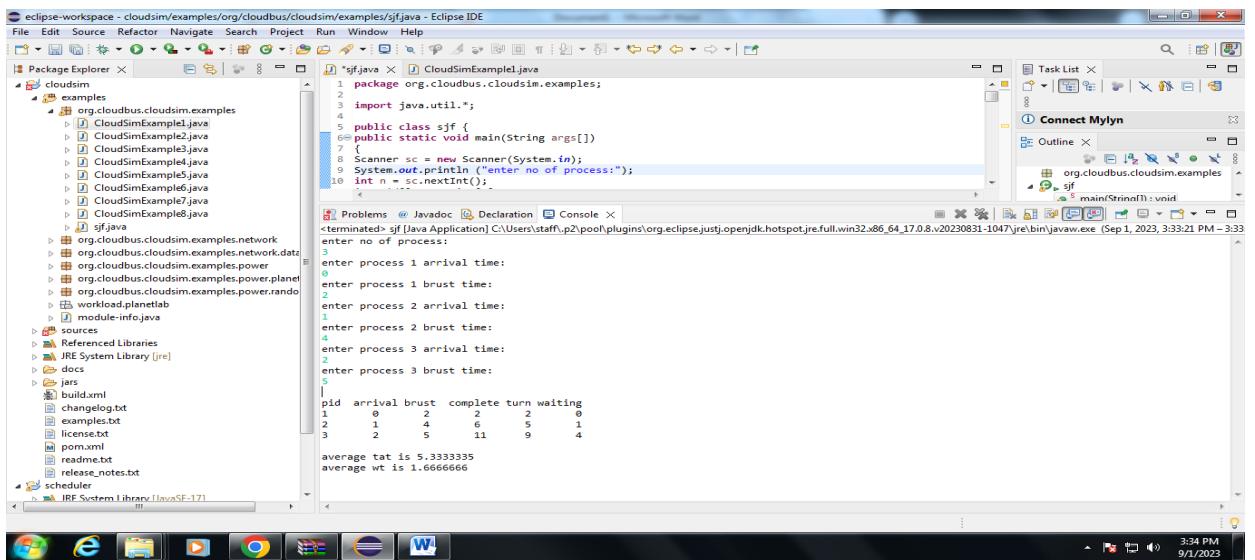
Open sjf.java

Change the package:

```
package org.cloudbus.cloudsim.examples;
```

Run the scheduling program in cloudsim

OUTPUT(WHICH IS ADDED IN CLOUDSIM FOLDER)



RESULT:

Thus the simulate a cloud scenario using cloudsim and run a scheduling algorithm that is not present in cloudsim was successfully completed.

EX.NO: 6

DATE:

FIND A PROCEDURE TO TRANSFER THE FILES FROM ONE VIRTUAL MACHINE TO ANOTHER VIRUTAL MACHINE.

AIM:

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

PROCEDURE:

Steps:

1. Select Oracle virtual box
2. Go to file and import Appliance
3. Select Open-nebula sandbox
4. Open Browser, type localhost:9869
5. Login using username: oneadmin, password: opennebula
6. Then follow the steps to migrate VMs
 - a. Click on infrastructure
 - b. Select clusters and enter the cluster name
 - c. Then select host tab, and select all host
 - d. Then select Vnets tab, and select all Vnet
 - e. Then select datastores tab, and select all datastores
 - f. And then choose host under infrastructure tab
 - g. Click on + symbol to add new host, name the host then click on create.
7. On instances, select VMs to migrate then follow the steps
 - a. Click on 8 th icon, the drop-down list display
 - b. Select migrate on that, the popup window display
 - c. On that select the target host to migrate then click on migrate.

Before migration

Host: one-sandbox

The screenshot shows the OpenNebula Sunstone web interface. The left sidebar contains navigation links for Dashboard, Instances (VMs, Services, Virtual Routers), Templates, Storage, Network, Infrastructure (Clusters, Hosts, Zones), System, Settings, Support (Not connected), and Upgrade Available. The main content area is titled "Host one-sandbox". It features tabs for Info, Graphs, VMs (selected), VMs, and Zombies. A table lists two VMs: vm6 and vm7, both running on host "one-sandbox". The table includes columns for ID, Owner, Group, Name, Status, Host, and IPs. A search bar and pagination controls are at the bottom.

The screenshot shows the "Migrate Virtual Machine" dialog box. It displays two notifications: "VM 6 vmB is currently running on Host one-sandbox" and "VM 7 vmB is currently running on Host one-sandbox". Below this, a section titled "Select a Host" asks to "Please select a Host from the list". A table lists three hosts: raa, naveenkumar, and one-sandbox. The "one-sandbox" host is selected. The table columns are ID, Name, Cluster, RVMs, Allocated CPU, Allocated MEM, and Status. A "Search" bar and pagination controls are present. At the bottom right of the dialog is a green "Migrate" button.

After Migration:

The screenshot shows the OpenNebula Sunstone interface for managing Virtual Machines (VMs). The left sidebar contains navigation links for Dashboard, Instances (VMs, Services, Virtual Routers), Templates, Storage, Network, Infrastructure (Clusters, Hosts, Zones), System, and Settings. A support section indicates 'Not connected' and shows a 'Sign in' button. The main panel displays a table titled 'VMs' with columns: ID, Owner, Group, Name, Status, Host, and IPs. The table lists 8 entries, with rows 6 and 7 highlighted. Row 6 has status 'SAVE' and row 7 has status 'FAILURE'. Below the table, it says 'Showing 1 to 8 of 8 entries' and provides 'Previous' and 'Next' buttons. At the bottom, it shows statistics: 8 TOTAL, 2 ACTIVE, 0 OFF, 0 PENDING, and 6 FAILED.

The screenshot shows the OpenNebula Sunstone interface for managing Hosts. The left sidebar contains navigation links for Dashboard, Instances (VMs, Services, Virtual Routers), Templates, Storage, Network, Infrastructure (Clusters, Hosts, Zones), System, and Settings. A support section indicates 'Not connected' and shows a 'Sign in' button. The main panel displays a table titled 'Hosts' with columns: ID, Name, Cluster, RVMs, Allocated CPU, Allocated MEM, and Status. The table lists 3 entries, with rows 1 and 2 highlighted. Row 1 has status 'ERROR' and row 2 has status 'ERROR'. Below the table, it says 'Showing 1 to 3 of 3 entries' and provides 'Previous' and 'Next' buttons. At the bottom, it shows statistics: 3 TOTAL, 1 ON, 0 OFF, and 2 ERROR.

Host:one-sandbox

The screenshot shows the OpenNebula Sunstone web interface. The left sidebar has a tree view with categories like Dashboard, Instances, Templates, Storage, Network, Infrastructure, System, and Support. The 'Hosts' node under Infrastructure is expanded. The main content area is titled 'Host one-sandbox'. It features tabs for Info, Graphs, VMs (selected), Wlids, and Zombies. A search bar is at the top right. Below is a table header with columns: ID, Owner, Group, Name, Status, Host, and IPs. A large 'Info' icon is centered, and a message says 'There is no data available'. At the bottom, there's a dropdown menu set to '10' and buttons for 'Previous' and 'Next'. The status bar at the bottom right shows '2:37 PM 8/23/2016'.

RESULT:

Thus the file transfer between VM was successfully completed.

EX.NO: 7

DATE:

INSTALL HADOOP SINGLE NODE CLUSTER AND RUN SIMPLE APPLICATIONS LIKE WORD COUNT

AIM:

To install hadoop single node cluster and run simple applications like word count.

PROCEDURE:

1. Download Cloudera Quickstart VM and start it up in VirtualBox or VMWare Fusion.
2. Open the terminal of cloudera VM
3. Before running WordCount example, we need to create some input text file, then move it to HDFS. First, create an input test file in your local file system.
4. Next, we need to move this file into HDFS. The following commands are the most basic HDFS commands to manage files in HDFS. In order of appearance below, we create a folder, copy the input file from local filesystem to HDFS, and list the content on HDFS.
5. It should be noted that for a fresh Cloudera VM, there is a “/user” folder in HDFS but not in the local filesystem. This example illustrates that local file system and HDFS are separate, and the Linux’s “ls” and HDFS’s “ls” interact with those independently.
6. Next, we want to run some MapReduce example, such as WordCount. The WordCount example is commonly used to illustrate how MapReduce works. The example returns a list of all the words that appear in a text file and the count of how many times each word appears. The output should show each word found and its count, line by line.
7. We need to locate the example programs on the sandbox VM. On Cloudera Quickstart VM, they are packaged in this jar file “hadoop-mapreduce-examples.jar”. Running that jar file without any argument will give you a list of available examples.
8. To run the WordCount example using the input file that we just moved to HDFS, use the following command:

```
[cloudera@quickstart temp]$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar wordcount /user/cloudera/input/wordcount.txtt  
/user/cloudera/output  
/user/cloudera/output
```
9. The output folder is specified as “/user/cloudera/output” in the above command.
Finally, check the output of WordCount example in the output folder.

COMMANDS:

```
[cloudera@quickstart ~]$ mkdir temp
[cloudera@quickstart ~]$ cd temp
[cloudera@quickstart temp]$ total 0 ls -ltr
[cloudera@quickstart temp]$ echo "this is the file which is used to count the word for the program which is simulated for the hadoop exercise" > wordcount.txt
[cloudera@quickstart temp]$ hdfs dfs -mkdir /user/cloudera/input
[cloudera@quickstart temp]$ hdfs dfs -ls /user/cloudera/input
[cloudera@quickstart temp]$ hdfs dfs -put /home/cloudera/temp/wordcount.txt /user/cloudera/input

put: `/home/cloudera/temp/wordcount.txt/user/cloudera/input': Nosuch file or directory
[cloudera@quickstart temp]$ hdfs dfs -put /home/cloudera/temp/wordcount.txt /user/cloudera/input

put: `/temp/wordcount.txt/user/cloudera/input': No such file or directory
[cloudera@quickstart temp]$ hdfs dfs -put /home/cloudera/temp/wordcount.txt /user/cloudera/input

put: `/cloudera/input': No such file or directory [cloudera@quickstart
temp]$ hdfs dfs -put /home/cloudera/temp/wordcount.txt /user/cloudera/input
[cloudera@quickstart temp]$ hdfs dfs -ls /user/cloudera/input
[cloudera@quickstart temp]$Found 1 items
-rw-r--r-- 1 cloudera cloudera 109 2017-09-23 01:20
  /user/cloudera/input/wordcount.txt
[cloudera@quickstart temp]$ls -ltr /usr/lib/hadoop-mapreduce/
total 29424
-rw-r--r-- 1 root root 94672 Jun 29 04:59 xz-1.0.jar
-rw-r--r-- 1 root root 15010 Jun 29 04:59 xmlenc-0.52.jar
-rw-r--r-- 1 root root 23346 Jun 29 04:59 stax-api-1.0-2.jar
-rw-r--r-- 1 root root 995968 Jun 29 04:59 snappy-java-1.0.4.1.jar
-rw-r--r-- 1 root root 105112 Jun 29 04:59 servlet-api-2.5.jar
-rw-r--r-- 1 root root 533455 Jun 29 04:59 protobuf-java-2.5.0.jar
-rw-r--r-- 1 root root 29555 Jun 29 04:59 paranamer-2.3.jar
-rw-r--r-- 1 root root 64661 Jun 29 04:59 okio-1.4.0.jar
-rw-r--r-- 1 root root 319099 Jun 29 04:59 okhttp-2.4.0.jar
-rw-r--r-- 1 root root 141986 Jun 29 04:59 mockito-all-9
  1.8.5.jar
-rw-r--r-- 1 root root 582965 Jun 29 04:59 microsoft-windowsazure-storage-sdk-0.6.0.jar
-rw-r--r-- 1 root root 85448 Jun 29 04:59 metrics-core-3.0.2.jar
-rw-r--r-- 1 root root 489884 Jun 29 04:59 log4j-1.2.17.jar
-rw-r--r-- 1 root root 245039 Jun 29 04:59 junit-4.11.jar
-rw-r--r-- 1 root root 33031 Jun 29 04:59 jsr305-3.0.0.jar
-rw-r--r-- 1 root root 100636 Jun 29 04:59 jsp-api-2.1.jar
-rw-r--r-- 1 root root 185746 Jun 29 04:59 jsch-0.1.42.jar
-rw-r--r-- 1 root root 177702 Jun 29 04:59 jetty-util-6.1.26.cloudera.4.jar
```

-rw-r--r--	1	root	root	147952	Jun 29 04:59	jersey-json-1.9.jar
-rw-r--r--	1	root	root	458739	Jun 29 04:59	jersey-core-1.9.jar
-rw-r--r--	1	root	root	890168	Jun 29 04:59	jaxb-impl-2.2.3-1.jar
-rw-r--r--	1	root	root	105134	Jun 29 04:59	jaxb-api-2.2.2.jar
-rw-r--r--	1	root	root	18490	Jun 29 04:59	java-xmlbuilder-0.4.jar
-rw-r--r--	1	root	root	76844	Jun 29 04:59	jasper-runtime-5.5.23.jar
-rw-r--r--	1	root	root	408133	Jun 29 04:59	jasper-compiler-5.5.23.jar
-rw-r--r--	1	root	root	32353	Jun 29 04:59	jackson-xc-1.8.8.jar
-rw-r--r--	1	root	root	668564	Jun 29 04:59	jackson-mapper-asl-1.8.8.jar
-rw-r--r--	1	root	root	17884	Jun 29 04:59	jackson-jaxrs-1.8.8.jar
-rw-r--r--	1	root	root	865838	Jun 29 04:59	jackson-databind-2.2.3.jar
-rw-r--r--	1	root	root	227500	Jun 29 04:59	jackson-core-asl-1.8.8.jar
-rw-r--r--	1	root	root	192699	Jun 29 04:59	jackson-core-2.2.3.jar
-rw-r--r--	1	root	root	33483	Jun 29 04:59	jackson-annotations-2.2.3.jar
-rw-r--r--	1	root	root	227708	Jun 29 04:59	httpcore-4.2.5.jar
-rw-r--r--	1	root	root	433368	Jun 29 04:59	httpclient-4.2.5.jar
-rw-r--r--	1	root	root	148510	Jun 29 04:59	htrace-core4-4.0.1-2
						incubating.jar
-rw-r--r--	1	root	root	45024	Jun 29 04:59	hamcrest-core-1.3.jar
-rw-r--r--	1	root	root	110259	Jun 29 04:59	hadoop-streaming-2.6.0-cdh5.12.0.jar
-rw-r--r--	1	root	root	118860	Jun 29 04:59	hadoop-sls-2.6.0-cdh5.12.0.jar
-rw-r--r--	1	root	root	282578	Jun 29 04:59	hadoop-rumen-2.6.0-cdh5.12.0.jar
-rw-r--r--	1	root	root	120983	Jun 29 04:59	hadoop-openstack-2.6.0-cdh5.12.0.jar
-rw-r--r--	1	root	root	276385	Jun 29 04:59	hadoop-mapreduce-540685 Jun 29 04:59 jetty-6.1.26.cloudera.4.jar
-rw-r--r--	1	root	root	67758	Jun 29 04:59	jettison-1.1.jar
-rw-r--r--	1	root	root	539735	Jun 29 04:59	jets3t-0.9.0.jar
-rw-r--r--	1	root	root	713089	Jun 29 04:59	jersey-server-1.9.jar

examples-2.6.0-cdh5.12.0.jar						
-rw-r--r-- 1 root root	56385	Jun 29 04:59	hadoop-mapreduce-			
client-shuffle-2.6.0-cdh5.12.0.jar						
-rw-r--r-- 1 root root	91100	Jun 29 04:59	hadoop-mapreduce-			
client-nativetask-2.6.0-cdh5.12.0.jar						
-rw-r--r-- 1 root root 1507341 Jun 29 04:59	hadoop-mapreduce-client-jobclient-2.6.0-					
cdh5.12.0-tests.jar						
-rw-r--r-- 1 root root	46074	Jun 29 04:59	hadoop-mapreduce-			
client-jobclient-2.6.0-cdh5.12.0.jar						
 cdh5.12.0.jar						
-rw-r--r-- 1 root root 2.6.0-cdh5.12.0.jar	19108	Jun 29 04:59	hadoop-datajoin-			
-rw-r--r-- 1 root root cdh5.12.0.jar	139930	Jun 29 04:59	hadoop-azure-2.6.0-			
-rw-r--r-- 1 root root cdh5.12.0.jar	75907	Jun 29 04:59	hadoop-auth-2.6.0-			
-rw-r--r-- 1 root root 2.6.0-cdh5.12.0.jar	26305	Jun 29 04:59	hadoop-archives-			
-rw-r--r-- 1 root root 2.6.0-cdh5.12.0.jar	24438	Jun 29 04:59	hadoop-archive-logs-			
-rw-r--r-- 1 root root cdh5.12.0.jar	14258	Jun 29 04:59	hadoop-ant-2.6.0-			
-rw-r--r-- 1 root root 0	164820	Jun 29 04:59	guava-11.0.2.jar			
-rw-r--r-- 1 root root	190432	Jun 29 04:59	gson-2.2.4.jar			
-rw-r--r-- 1 root root 2.7.1.jar	270342	Jun 29 04:59	curator-recipes-			
-rw-r--r-- 1 root root 2.7.1.jar	186273	Jun 29 04:59	curator-framework-			
-rw-r--r-- 1 root root 2.7.1.jar	69500	Jun 29 04:59	curator-client-			
-rw-r--r-- 1 root root 273370 Jun 29 04:59	commons-net-3.1.jar					
-rw-r--r-- 1 root root 159962 Jun 29 04:59	commons-math3-					
3.1.1.jar	7					
-rw-r--r-- 1 root root 1.1.3.jar	62050	Jun 29 04:59	commons-logging-			
-rw-r--r-- 1 root root 284220 Jun 29 04:59	commons-lang-2.6.jar					
-rw-r--r-- 1 root root 185140 Jun 29 04:59	commons-io-2.4.jar					
-rw-r--r-- 1 root root 305001 Jun 29 04:59	commons-httpclient-					
3.1.jar						
-rw-r--r-- 1 root root 112341 Jun 29 04:59	commons-el-1.0.jar					
-rw-r--r-- 1 root root 143602 Jun 29 04:59	commons-digester-					
1.8.jar						
-rw-r--r-- 1 root root configuration-1.6.jar	298829	Jun 29 04:59	commons-			
-rw-r--r-- 1 root root 1.4.1.jar	241367	Jun 29 04:59	commons-compress-			
-rw-r--r-- 1 root root 3.2.2.jar	588337	Jun 29 04:59	commons-collections-			
-rw-r--r-- 1 root root 1.4.jar	58160	Jun 29 04:59	commons-codec-			
-rw-r--r-- 1 root root core-1.8.0.jar	41123	Jun 29 04:59	commons-cli-1.2.jar			
-rw-r--r-- 1 root root commons-beanutils-	206035	Jun 29 04:59	core-1.8.0.jar			

```

-rw-r--r-- 1 root root 233859 Jun 29 04:59 commons-beanutils-
1.9.2.jar
-rw-r--r-- 1 root root 43398 Jun 29 04:59 asm-3.2.jar
-rw-r--r-- 1 root root 79912 Jun 29 04:59 api-util-1.0.0-
-rw-r--r-- 1 root root 10531 Jun 29 04:59 hadoop-mapreduce-
client-hs-plugins-2.6.0-cdh5.12.0.jar
-rw-r--r-- 1 root root 177983 Jun 29 04:59 hadoop-mapreduce-client-hs-2.6.0-
cdh5.12.0.jar
-rw-r--r-- 1 root root 1552892 Jun 29 04:59 hadoop-mapreduce-client-core-2.6.0-
cdh5.12.0.jar
-rw-r--r-- 1 root root 21993 Jun 29 04:59 hadoop-gridmix-
2.6.0-cdh5.12.0.jar
-rw-r--r-- 1 root root 66640 Jun 29 04:59 hadoop-extras-2.6.0-
cdh5.12.0.jar
-rw-r--r-- 1 root root 19660 Jun 29 04:59 hadoop-distcp-2.6.0-
59
-rw-r--r-- 1 root root
755220 Jun 29 04:59 hadoop-
mapreduce- client-common-
2.6.0-cdh5.12.0.jar
-rw-r--r-- 1 root root
530889 Jun 29 04:59 hadoop-
mapreduce- client-app-2.6.0-
cdh5.12.0.jar
codec-2.0.0-M15.jar

```

```

M20.jar
-rw-r--r-- 1 root root 16560 Jun 29 04:59 api-asn1-api-1.0.0-
M20.jar
-rw-r--r-- 1 root root 691479 Jun 29 04:59 apacheds-kerberos-
/usr/lib/avro/avro.jar
drwxr-xr-x 2 root root 4096 Jul 19 05:27 cloudera
drwxr-xr-x 2 root root 4096 Jul 19 05:27 bin
lrwxrwxrwx 1 root root 30 Jul 19 05:27 hadoop-ant.jar ->hadoop-
ant-2.6.0-cdh5.12.0.jar
lrwxrwxrwx 1 root root 35 Jul 19 05:27 hadoop-archives.jar
->hadoop-archives-2.6.0-cdh5.12.0.jar
lrwxrwxrwx 1 root root 39 Jul 19 05:27 hadoop-archive- logs.jar -
> hadoop-archive-logs-2.6.0-cdh5.12.0.jar lrwxrwxrwx 1 root root 31 Jul 19 05:27
hadoop-auth.jar ->hadoop-auth-2.6.0-cdh5.12.0.jar
lrwxrwxrwx 1 root root 35 Jul 19 05:27 hadoop-datajoin.jar
->hadoop-datajoin-2.6.0-cdh5.12.0.jar
lrwxrwxrwx 1 root root 32 Jul 19 05:27 hadoop-azure.jar ->hadoop-
azure-2.6.0-cdh5.12.0.jar
lrwxrwxrwx 1 root root 33 Jul 19 05:27 hadoop-distcp.jar ->hadoop-
distcp-2.6.0-cdh5.12.0.jar
lrwxrwxrwx 1 root root 33 Jul 19 05:27 hadoop-extras.jar ->hadoop-
extras-2.6.0-cdh5.12.0.jar
lrwxrwxrwx 1 root root 34 Jul 19 05:27 hadoop-gridmix.jar -
> hadoop-gridmix-2.6.0-cdh5.12.0.jar

```

```

lrwxrwxrwx 1 root root          47 Jul 19 05:27 hadoop-mapreduce-
client-app.jar -> hadoop-mapreduce-client-app-2.6.0- cdh5.12.0.jar
lrwxrwxrwx 1 root root          50 Jul 19 05:27 hadoop-mapreduce-
client-common.jar -> hadoop-mapreduce-client-common-2.6.0- cdh5.12.0.jar
lrwxrwxrwx 1 root root          48 Jul 19 05:27 hadoop-mapreduce-
client-core.jar -> hadoop-mapreduce-client-core-2.6.0- cdh5.12.0.jar
lrwxrwxrwx 1 root root          54 Jul 19 05:27 hadoop-mapreduce-
client-hs-plugins.jar -> hadoop-mapreduce-client-hs-plugins-2.6.0-cdh5.12.0.jar
lrwxrwxrwx 1 root root          46 Jul 19 05:27 hadoop-mapreduce- client-
hs.jar -> hadoop-mapreduce-client-hs-2.6.0-cdh5.12.0.jar lrwxrwxrwx 1 root root  59 Jul
19 05:27 hadoop-mapreduce- client-jobclient-tests.jar -> hadoop-mapreduce-client-
jobclient-2.6.0-cdh5.12.0-tests.jar
lrwxrwxrwx 1 root root          53 Jul 19 05:27 hadoop-mapreduce- client-
jobclient.jar -> hadoop-mapreduce-client-jobclient-2.6.0-cdh5.12.0.jar
lrwxrwxrwx 1 root root          51 Jul 19 05:27 hadoop-mapreduce-
client-shuffle.jar -> hadoop-mapreduce-client-shuffle-2.6.0-cdh5.12.0.jar
lrwxrwxrwx 1 root root          54 Jul 19 05:27 hadoop-mapreduce-
client-nativetask.jar -> hadoop-mapreduce-client-nativetask-2.6.0-cdh5.12.0.jar
lrwxrwxrwx 1 root root          45 Jul 19 05:27 hadoop-mapreduce-
examples.jar -> hadoop-mapreduce-examples-2.6.0-cdh5.12.0.jar lrwxrwxrwx 1 root root
36 Jul 19 05:27 hadoop-openstack.jar
-> hadoop-openstack-2.6.0-cdh5.12.0.jar

lrwxrwxrwx 1 root root          32 Jul 19 05:27 hadoop-rumen.jar ->
hadoop-rumen-2.6.0-cdh5.12.0.jar
lrwxrwxrwx 1 root root          30 Jul 19 05:27 hadoop-sls.jar -> hadoop-
sls-2.6.0-cdh5.12.0.jar
lrwxrwxrwx 1 root root          36 Jul 19 05:27 hadoop-streaming.jar
-> hadoop-streaming-2.6.0-cdh5.12.0.jar
drwxr-xr-x 2 root root          4096 Jul 19 05:27 lib
lrwxrwxrwx 1 root root          26 Jul 19 05:27 logs ->
/var/log/hadoop-mapreduce/
drwxr-xr-x 2 root root          4096 Jul 19 05:27 sbin
lrwxrwxrwx 1 root root          32 Jul 19 05:27 zookeeper.jar ->
/usr/lib/zookeeper/zookeeper.jar [cloudera@quickstart temp]$ [cloudera@quickstart temp]$ 
[cloudera@quickstart temp]$ [cloudera@quickstart temp]$ [cloudera@quickstart temp]$ 
[cloudera@quickstart temp]$ [cloudera@quickstart temp]$ 
[cloudera@quickstart temp]$ had bash: had:
command not found [cloudera@quickstart temp]$ 
[cloudera@quickstart temp]$
```

An example program must be given as the first argument. Valid program names are:

aggregatewordcount: An Aggregate based map/reduce program that counts the words in the input files.

aggregatewordhist: An Aggregate based map/reduce program that computes the histogram of the words in the input files.

bbp: A map/reduce program that uses Bailey-Borwein-Plouffe to compute exact digits of Pi.

dbcount: An example job that counts the pageview counts from a database.

distbbp: A map/reduce program that uses a BBP-type formula to compute exact bits of Pi.

grep: A map/reduce program that counts the matches of a regex in the input.

join: A job that effects a join over sorted, equally partitioned datasets

multifilewc: A job that counts words from several files. pentomino: A map/reduce tile laying program to find solutions

to pentomino problems.

pi: A map/reduce program that estimates Pi using a quasi-MonteCarlo method.

randomtextwriter: A map/reduce program that writes 10GB of random textual data per node.

randomwriter: A map/reduce program that writes 10GB of random data per node.

secondarysort: An example defining a secondary sort to the reduce.

sort: A map/reduce program that sorts the data written by the random writer.

sudoku: A sudoku solver.

teragen: Generate data for the terasort terasort: Run the terasort

teravalidate: Checking results of terasort

wordcount: A map/reduce program that counts the words in the input files.

wordmean: A map/reduce program that counts the average length of the words in the input files.

wordmedian: A map/reduce program that counts the median length of the words in the input files.

wordstandarddeviation: A map/reduce program that counts the standard deviation of the length of the words in the input files.

```
[cloudera@quickstart temp]$ hadoop jar /usr/lib/hadoop-
mapreduce/hadoop-mapreduce-examples.jar wordcount
/usr/cloudera/input/wordcount.txt /user/cloudera/output
17/09/23 01:26:03 INFO client.RMProxy: Connecting to ResourceManager
at /0.0.0.0:8032
17/09/23 01:26:04 INFO input.FileInputFormat: Total input paths to process : 1
17/09/23 01:26:04 INFO mapreduce.JobSubmitter: number of splits:1
17/09/23 01:26:04 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1506154437319_0001
17/09/23 01:26:05 INFO impl.YarnClientImpl: Submitted application
application_1506154437319_0001
17/09/23 01:26:05 INFO mapreduce.Job: The url to track the job:
http://quickstart.cloudera:8088/proxy/application_1506154437319_0001/
17/09/23 01:26:05 INFO mapreduce.Job: Running job:
job_1506154437319_0001
```

17/09/23 01:26:15 INFO mapreduce.Job: Job job_1506154437319_0001 running in uber mode : false
17/09/23 01:26:15 INFO mapreduce.Job: map 0% reduce 0%
17/09/23 01:26:23 INFO mapreduce.Job: map 100% reduce 0%
17/09/23 01:26:31 INFO mapreduce.Job: map 100% reduce 100% 17/09/23 01:26:31 INFO mapreduce.Job: Job job_1506154437319_0001 completed successfully
17/09/23 01:26:31 INFO mapreduce.Job: Counters: 49 File System Counters
FILE: Number of bytes read=180 FILE:
Number of bytes written=250871 FILE:
Number of read operations=0
FILE: Number of large read operations=0 FILE:
Number of write operations=0 HDFS: Number of bytes read=239
HDFS: Number of bytes written=114 HDFS:
Number of read operations=6 HDFS: Number of large read operations=0 HDFS: Number of write operations=2
Job Counters
Launched map tasks=1
Launched reduce tasks=1
Data-local map tasks=1
Total time spent by all maps in occupied slots
(ms)=5352
Total time spent by all reduces in occupied slots
(ms)=5376
Total time spent by all map tasks (ms)=5352 Total time spent by all reduce tasks (ms)=5376
Total vcore-milliseconds taken by all map tasks=5352
Total vcore-milliseconds taken by all reduce tasks=5376 Total megabyte-milliseconds taken by all map tasks=5480448
Total megabyte-milliseconds taken by all reduce tasks=5505024

Map-Reduce Framework

```
Map input records=1 Map
output records=21 Map
output bytes=193
Map output materialized bytes=180 Input
split bytes=130
Combine input records=21
Combine output records=15
Reduce input groups=15
Reduce shuffle bytes=180
Reduce input records=15
Reduce output records=15
Spilled Records=30 Shuffled
Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=206
CPU time spent (ms)=1330
Physical memory (bytes) snapshot=359673856 Virtual
memory (bytes) snapshot=3015249920 Total committed
heap usage (bytes)=226627584
```

Shuffle Errors

```
BAD_ID=0
CONNECTION
=0
IO_ERROR=0
WRONG LENG
TH=0
WRONG_MAP
=0
WRONG_REDU
CE=0
```

File Input Format Counters

```
Bytes Read=109
```

File Output Format Counters

```
Bytes Written=114
```

```
[cloudera@quickstart temp]$ hdfs dfs -ls /user/cloudera/output/
[cloudera@quickstart temp]$ Found 2 items
-rw-r--r-- 1 cloudera cloudera 0 2017-09-23 01:26
/user/cloudera/output/_SUCCESS
-rw-r--r-- 1 cloudera cloudera 114 2017-09-23 01:26
/user/cloudera/output/part-r-00000
```

```
[cloudera@quickstart temp]$ hdfs dfs -ls
```

```
/user/cloudera/output/part-r-00000
```

```
-rw-r--r-- 1 cloudera cloudera 114 2017-09-23 01:26
```

```
/user/cloudera/output/part-r-00000
```

```
[cloudera@quickstart temp]$ hdfs dfs -cat
```

```
/user/cloudera/output/part-r-00000
```

```
count 1
exercise 1
file1
for 2
hadoop 1
```

```
is          2
iss 1
program      1
simulated 1
the 4

this 1
to      1
used 1
which      2
word 1
[cloudera@quickstart temp]$
```

RESULT:

Thus the installation of single node Hadoop cluster and executing the wordcount application is successfully completed.

EX.NO: 8

DATE:

CREATING AND EXECUTING YOUR FIRST CONTAINER USING DOCKER

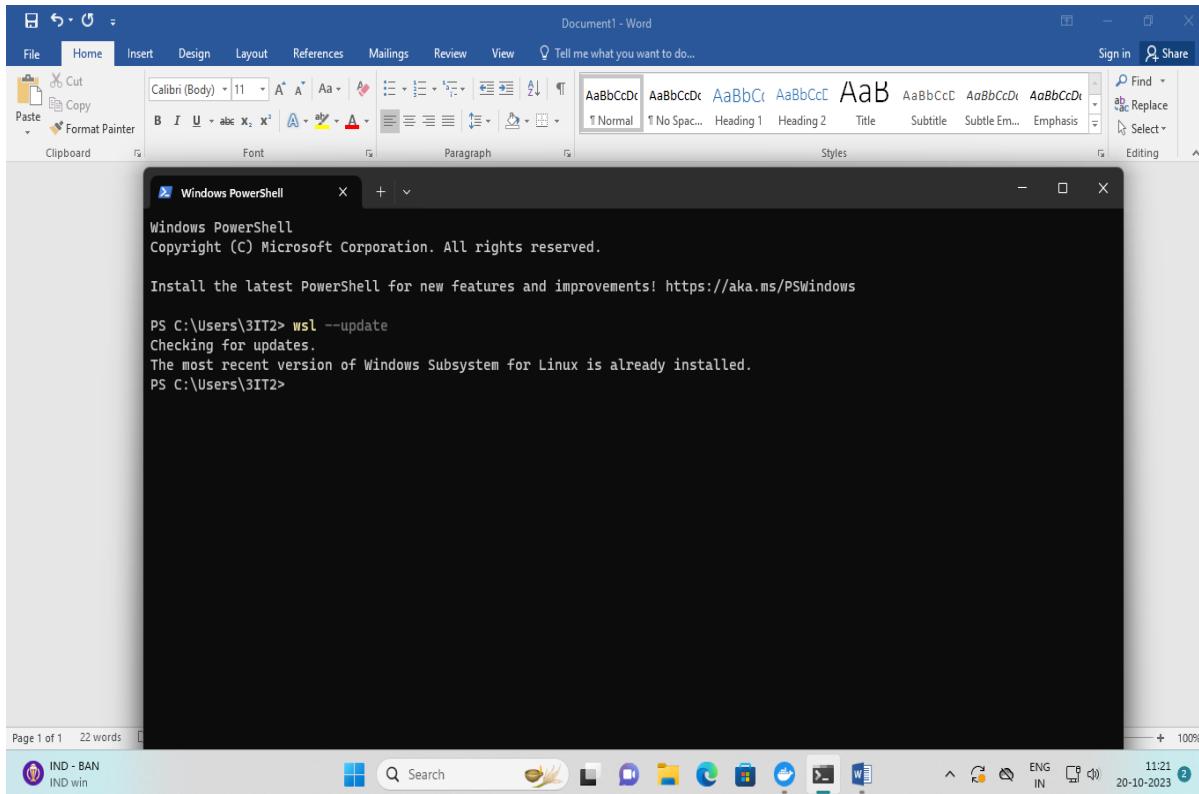
AIM:

To create and execute your first container using Docker.

PROCEDURE:

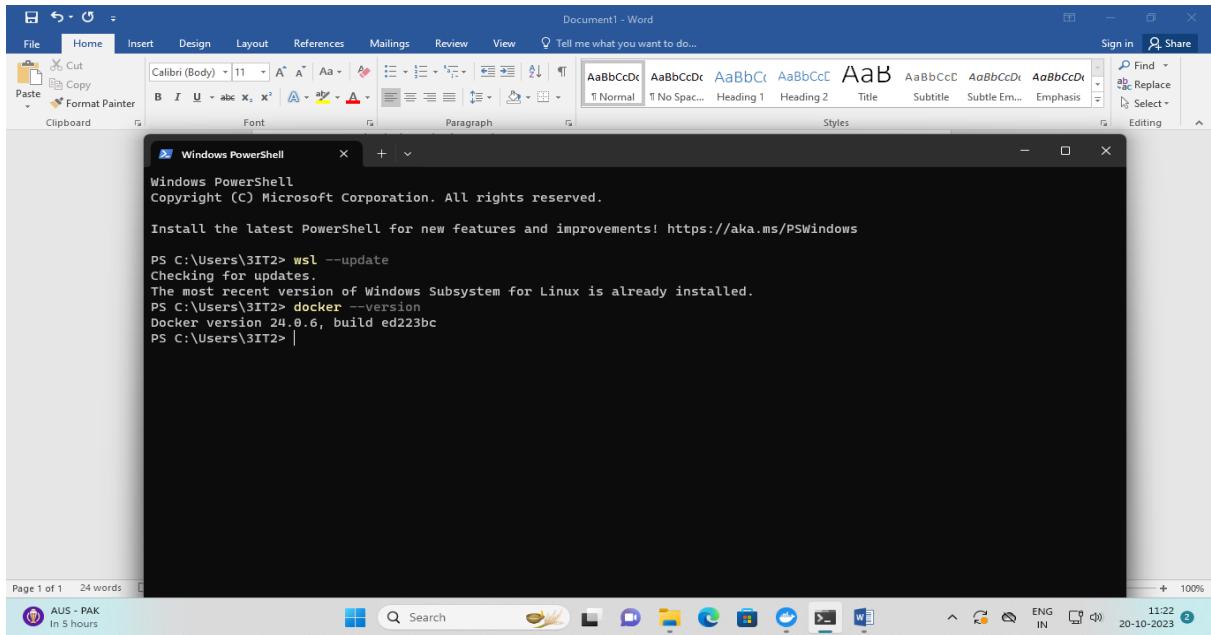
1. Install docker desktop for windows .
2. Open the docker and run as administrator.
3. Run the docker in background.
4. Open power shell and type the following commands.

wsl -- update



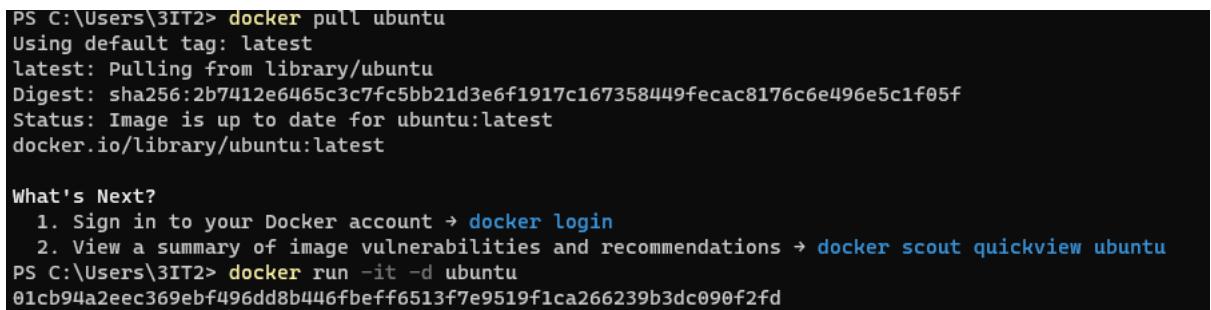
5. Check the docker version using this command.

docker --version



docker pull ubuntu

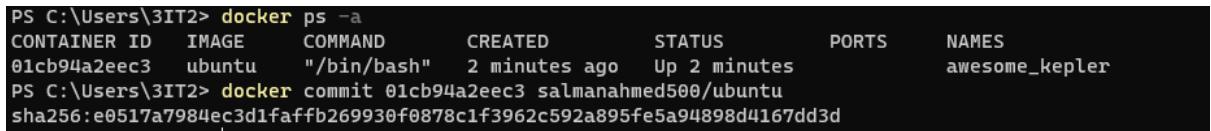
docker run -it -d ubuntu



6. Create an account in docker repository.

7. Create your own image using the Ubuntu container using the below command.

Docker commit “id” username/ubuntu



8. Create your own container using the Ubuntu by the below command.

Docker run -it -d username/ubuntu

```
PS C:\Users\3IT2> docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
salmanahmed500/ubuntu    latest    e0517a7984ec  2 minutes ago  77.8MB
<none>            <none>    95d7c58a2f43  28 minutes ago  77.8MB
ubuntu              latest    e4c58958181a  2 weeks ago   77.8MB
hello-world         latest    9c7a54a9a43c  5 months ago   13.3kB
PS C:\Users\3IT2> docker run -it -d salmanahmed500/ubuntu
c17cce45e0cd89bd908f8b93eb10286643e4fba94fbd12ad2011ce03b9eb808c
PS C:\Users\3IT2> docker ps -a
CONTAINER ID   IMAGE           COMMAND      CREATED       STATUS        PORTS     NAMES
c17cce45e0cd   salmanahmed500/ubuntu   "/bin/bash"   7 seconds ago   Up 6 seconds
01cb94a2eec3   ubuntu          "/bin/bash"   5 minutes ago   Up 5 minutes

```

9. Now the new container is created eg-“c17cce45e0cd” .

RESULT:

Thus the creating and executing your first container using docker is successfully completed.

EX.NO: 9

DATE:

RUN A CONTAINER FROM DOCKER HUB

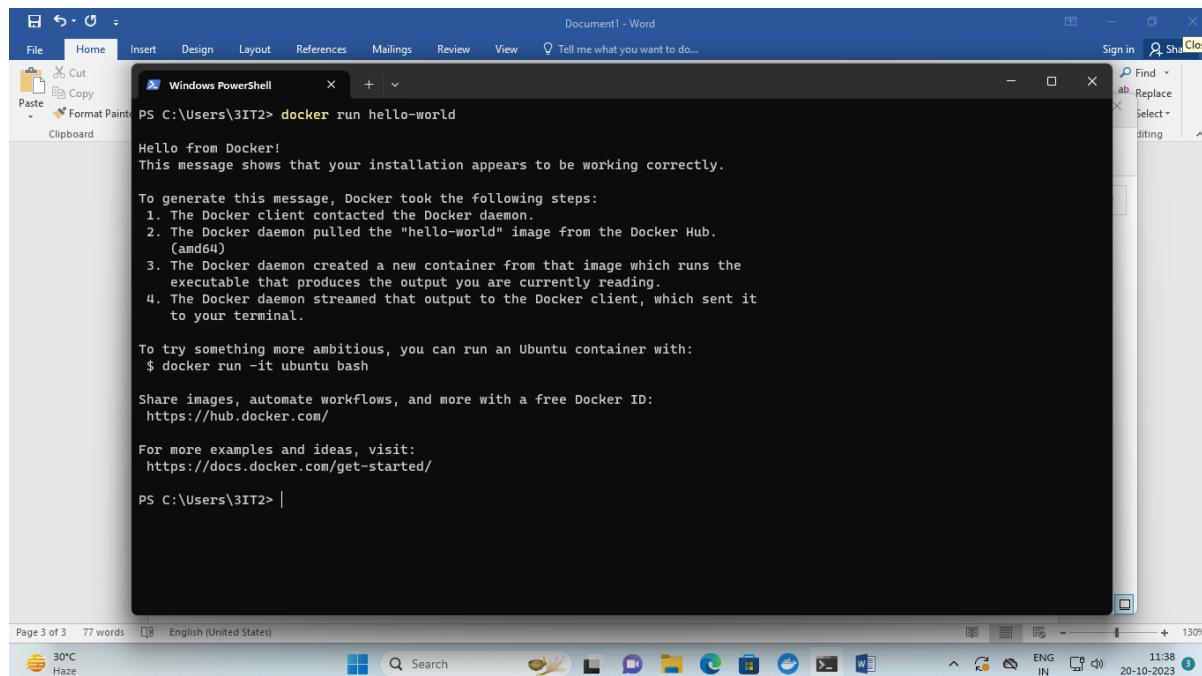
AIM:

To run a container from docker hub.

PROCEDURE:

1. To run a container from docker hub, we need to run the image from docker hub
2. Run the image hello-world form the docker hub using the below command

docker run hello-world



The screenshot shows a Microsoft Word document titled "Document1 - Word". Inside the document, there is a Windows PowerShell window. The command "PS C:\Users\3IT2> docker run hello-world" is entered, and the output is displayed:
Hello from Docker!
This message shows that your installation appears to be working correctly.
To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.
To try something more ambitious, you can run an Ubuntu container with:
\$ docker run -it ubuntu bash
Share images, automate workflows, and more with a free Docker ID:
<https://hub.docker.com/>
For more examples and ideas, visit:
<https://docs.docker.com/get-started/>
PS C:\Users\3IT2> |

3. Now the image is created
4. list the image using the below command

docker images

5. The container is created with the image hello-world
6. list the container using the below command

docker ps -a

```
PS C:\Users\3IT2> docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
salmanahmed500/ubuntu    latest   e0517a7984ec  12 minutes ago  77.8MB
<none>              <none>   95d7c58a2f43  38 minutes ago  77.8MB
ubuntu               latest   e4c58958181a  2 weeks ago   77.8MB
hello-world          latest   9c7a54a9a43c  5 months ago   13.3kB
PS C:\Users\3IT2> docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
0a87564a6f3a        hello-world        "/hello"           5 minutes ago     Exited (0) 5 minutes ago
c17cce45e0cd        salmanahmed500/ubuntu  "/bin/bash"       10 minutes ago    Up 10 minutes
01cb94a2eec3        ubuntu             "/bin/bash"       15 minutes ago    Up 15 minutes
```

7. Now the container is created from the image hello-world Eg-id “ 0a87564a6f3a”

RESULT:

Thus the run a container from docker hub is successfully completed.

