Ex 1 **Create an application using the make command**
25.08.22

Aim:
To use gcc to compile c-programs. Split the programs to different modules and create an application using make command.

Algorithm:

1. Create a file with .c extension in ubuntu system
2. Add your program to the file
3. Create two more c files and a makefile

Program:
Sample.c

```
#include<stdio.h>
void main()
{
int a=5,b=3,c;
c=a+b;
printf("\n addition =%d",c);
}
```

Makefile By Example

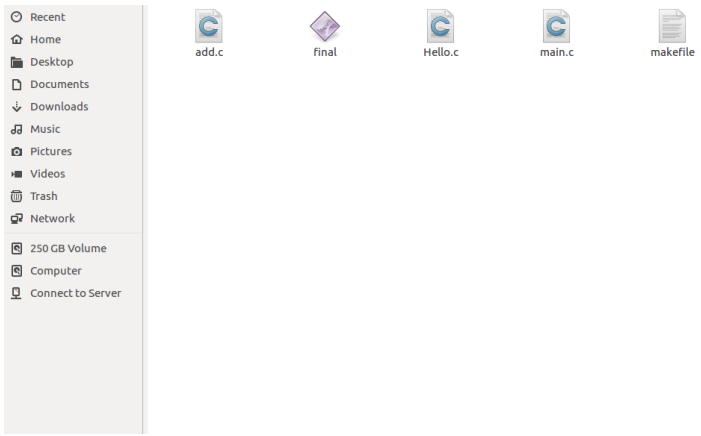
```
#include<stdio.h>
void main()
{
int a=5,b=3,c;
c=a+b;
printf("\n addition =%d",c);
}
```

To Create a file makefile
Syntax:
target: pre-req-1 pre-req-2 ...
command
The target and pre-requisites are separated by a colon (:). The command must be preceded by a tab

```
all: sample.exe
sample.exe: sample.o
gcc -o sample.exe sample.o
```

sample.o: sample.c
gcc -c sample.c
clean:
rm sample.o sample.exe
When make is asked to evaluate a rule, it begins by finding the files in the prerequisites. If any of the prerequisites have an associated rule, make attempts to update those first.
You can then run the make without a target, which is the same as "make all".
> make clean
> make

Output:




```
student@student-ThinkCentre-M71e:~$ gedit makefile
student@student-ThinkCentre-M71e:~$ make final
gcc Sample.c Hello.c add.c -o final
Sample.c: In function ‘main’:
Sample.c:6:1: warning: implicit declaration of function ‘hello’ [-Wimplicit-func
tion-declaration]
 hello();
 ^
Sample.c:8:1: warning: implicit declaration of function ‘add’ [-Wimplicit-functi
on-declaration]
 add(3,6);
 ^
student@student-ThinkCentre-M71e:~$ ./final

 I am in main.c

 calling hello function
Iam in hello.c
PID of hello.c is 3436/n
 calling add function

 addition =9
 back in main.c
```

Result:
Thus an application was created using make command

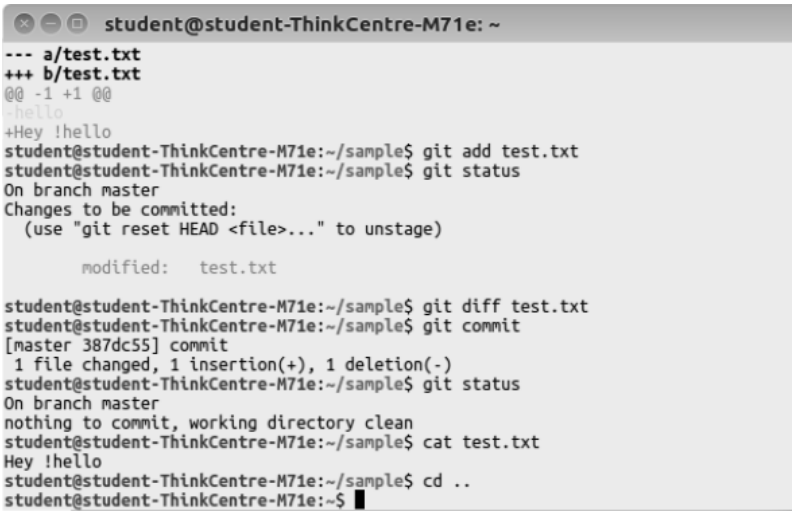
Ex 2 **Version control systems command**
30.08.22

Aim:
To use version control systems command to clone, commit, push, fetch, pull, checkout, reset, and delete repositories.

Algorithm:

1. Install git and initialize empty git repository
2. Create Repository and move your local folder to the repository
3. Create file and check status of the file
4. Commit the changes of the file and push it to the repository
5. To create a new repository as repository in terminal:
6. Click + plus on the right side of your GITHUB profile -> Click new repository -> type new
   repository name as repository name in your terminal -> give some data in the description field
   eg(test) - > click create repository

Output:



sudo apt-get install git
[sudo] password for student:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
libllvm4.0

awdasdsdasdads Zack

```
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-arch git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 3,939 kB of archives.
After this operation, 25.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu xenial/main amd64 liberror-perl all 0.17-1.2 [19.6 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu xenial-updates/main amd64 git-man all
1:2.7.4-0ubuntu1.10 [737 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu xenial-updates/main amd64 git amd64
1:2.7.4-0ubuntu1.10 [3,183 kB]
Fetched 3,939 kB in 3s (1,281 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 211317 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17-1.2_all.deb ...
Unpacking liberror-perl (0.17-1.2) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.7.4-0ubuntu1.10_all.deb ...
Unpacking git-man (1:2.7.4-0ubuntu1.10) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.7.4-0ubuntu1.10_amd64.deb ...
Unpacking git (1:2.7.4-0ubuntu1.10) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up liberror-perl (0.17-1.2) ...
Setting up git-man (1:2.7.4-0ubuntu1.10) ...
Setting up git (1:2.7.4-0ubuntu1.10) ...
student@student-ThinkCentre-M71e:~$ which git
/usr/bin/git
student@student-ThinkCentre-M71e:~$ git init myRepo
Initialized empty Git repository in /home/student/myRepo/.git/
student@student-ThinkCentre-M71e:~$ ls
add.c  Downloads     Hello.c  myRepo       Sample.c
Desktop         examples.desktop  makefile  Pictures  Templates
Documents  final                 Music  Public  Videos
student@student-ThinkCentre-M71e:~$ cd myRepo
student@student-ThinkCentre-M71e:~/myRepo$ ls -l .git
total 32
```

```
drwxrwxr-x 2 student student 4096 Aug 25 15:18 branches
-rw-rw-r-- 1 student student   92 Aug 25 15:18 config
-rw-rw-r-- 1 student student   73 Aug 25 15:18 description
-rw-rw-r-- 1 student student   23 Aug 25 15:18 HEAD
drwxrwxr-x 2 student student 4096 Aug 25 15:18 hooks
drwxrwxr-x 2 student student 4096 Aug 25 15:18 info
drwxrwxr-x 4 student student 4096 Aug 25 15:18 objects
drwxrwxr-x 4 student student 4096 Aug 25 15:18 refs
student@student-ThinkCentre-M71e:~/myRepo$ ls -la
total 12
drwxrwxr-x  3 student student 4096 Aug 25 15:18 .
drwxr-xr-x 18 student student 4096 Aug 25 15:18 ..
drwxrwxr-x  7 student student 4096 Aug 25 15:18 .git
student@student-ThinkCentre-M71e:~/myRepo$ nano hello.txt
student@student-ThinkCentre-M71e:~/myRepo$ ls
hello.txt
student@student-ThinkCentre-M71e:~/myRepo$ git status
On branch master
Initial commit
Untracked files:
  (use "git add <file>..." to include in what will be committed)

hello.txt

nothing added to commit but untracked files present (use "git add" to track)
student@student-ThinkCentre-M71e:~/myRepo$ git add hello.txt
student@student-ThinkCentre-M71e:~/myRepo$ git status
On branch master
Initial commit
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
new file:   hello.txt
student@student-ThinkCentre-M71e:~/myRepo$ git config --global user.name "student"
student@student-ThinkCentre-M71e:~/myRepo$ git config --global user.email "student"
student@student-ThinkCentre-M71e:~/myRepo$ git commit -m "initial commit"
[master (root-commit) 1fbcf2c] initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 hello.txt

student@student-ThinkCentre-M71e:~/myRepo$ git status
On branch master
nothing to commit, working directory clean
```

awdasdsdasdads Zack

```
student@student-ThinkCentre-M71e:~/myRepo$ git diff hello.txt
diff --git a/hello.txt b/hello.txt
index 3b18e51..d75e656 100644
--- a/hello.txt
+++ b/hello.txt
@@ -1 +1 @@
-hello world
+hello world, welcome

student@student-ThinkCentre-M71e:~/myRepo$ git add hello.txt
student@student-ThinkCentre-M71e:~/myRepo$ git commit -m "second commit"
[master 36f647c] second commit
 1 file changed, 1 insertion(+), 1 deletion(-)
student@student-ThinkCentre-M71e:~/myRepo$ cat hello.txt
hello world, welcome
student@student-ThinkCentre-M71e:~/myRepo$ cd ..

student@student-ThinkCentre-M71e:~$ git clone
https://github.com/elizabeth-d20/UnitConvertorTest.git
Cloning into 'UnitConvertorTest'...
Username for 'https://github.com': elizabeth-d20
Password for 'https://elizabeth-d20@github.com':
remote: Enumerating objects: 75, done.
remote: Counting objects: 100% (75/75), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 75 (delta 2), reused 75 (delta 2), pack-reused 0
Unpacking objects: 100% (75/75), done.
Checking connectivity... done.

student@student-ThinkCentre-M71e:~/UnitConvertorTest$ ls
app  build.gradle  gradle  gradle.properties  gradlew  gradlew.bat  settings.gradle
student@student-ThinkCentre-M71e:~/UnitConvertorTest$ cd .git
student@student-ThinkCentre-M71e:~/UnitConvertorTest/.git$ ls
branches  config  description  HEAD  hooks  index  info  logs  objects  packed-refs  refs

student@student-ThinkCentre-M71e:~/myRepo$ git remote add origin
https://github.com/elizabeth-d20/myRepo.git
student@student-ThinkCentre-M71e:~/myRepo$ git push origin master
Username for 'https://github.com': elizabeth-d20
Password for 'https://elizabeth-d20@github.com':
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
```

awdasdsdasdads Zack

```
Writing objects: 100% (6/6), 420 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/elizabeth-d20/myRepo.git
 * [new branch]          master -> master

student@student-ThinkCentre-M71e:~/myRepo$ git pull origin master
From https://github.com/elizabeth-d20/myRepo
 * branch         master          -> FETCH_HEAD
Already up-to-date.

student@student-ThinkCentre-M71e:~/myRepo$ git log
commit 36f647c5f69c1befa9a0235c1daf0449dd2cf444
Author: student <student>
Date:   Thu Aug 25 15:32:38 2022 +0530

        second commit

commit 1fbcf2c15628c762768e96a8031c49f6fccded07
Author: student <student>
Date:   Thu Aug 25 15:26:56 2022 +0530
initial commit
```

Result:

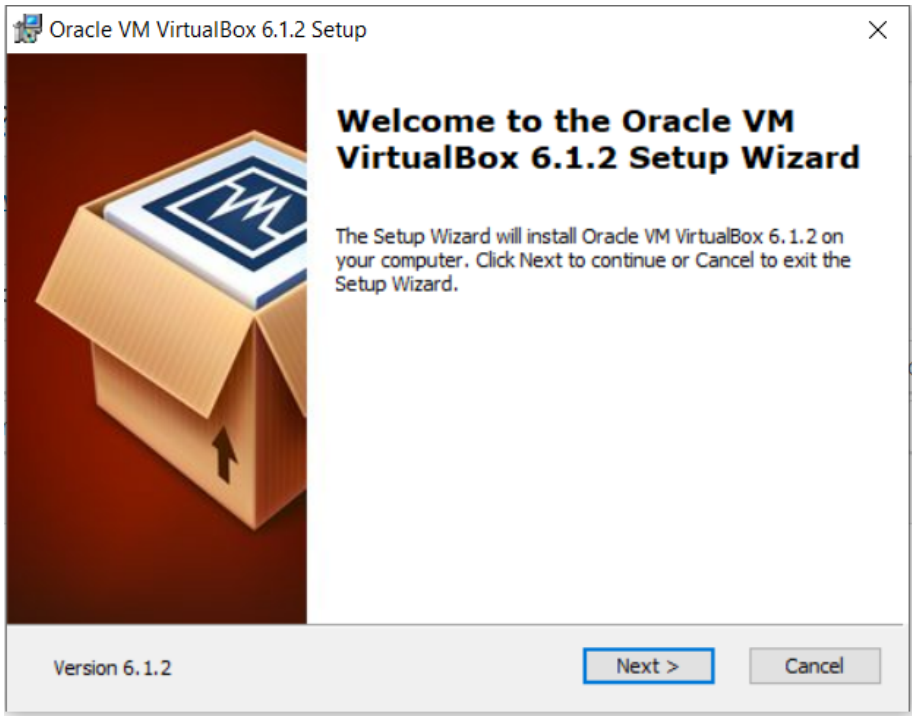
Thus, version control systems commands were used to clone, commit, push, fetch, pull, checkout, reset, and delete repositories.

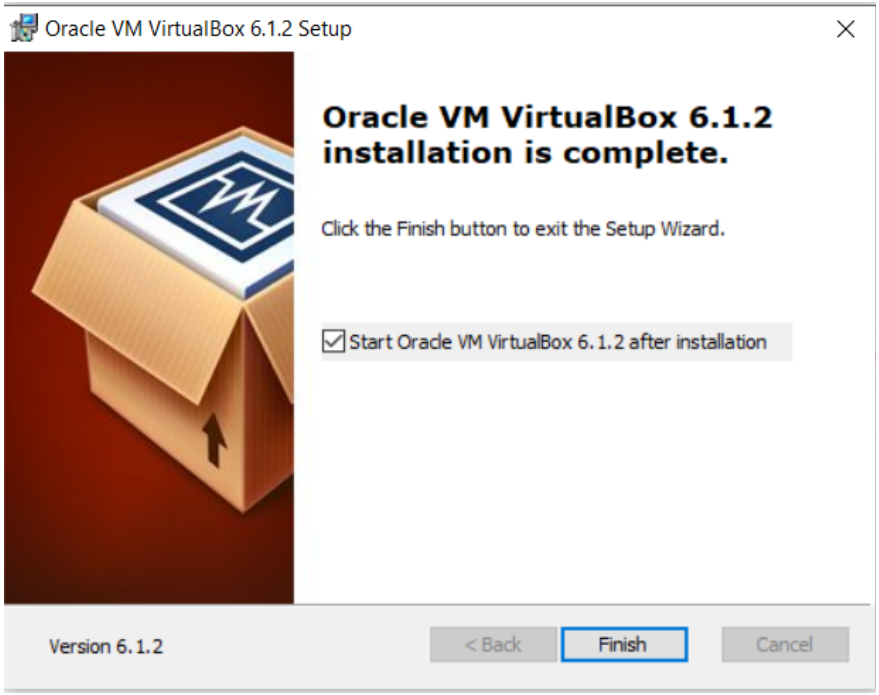
Ex 3 **Install Virtualbox/VMware Workstation**
08.09.22

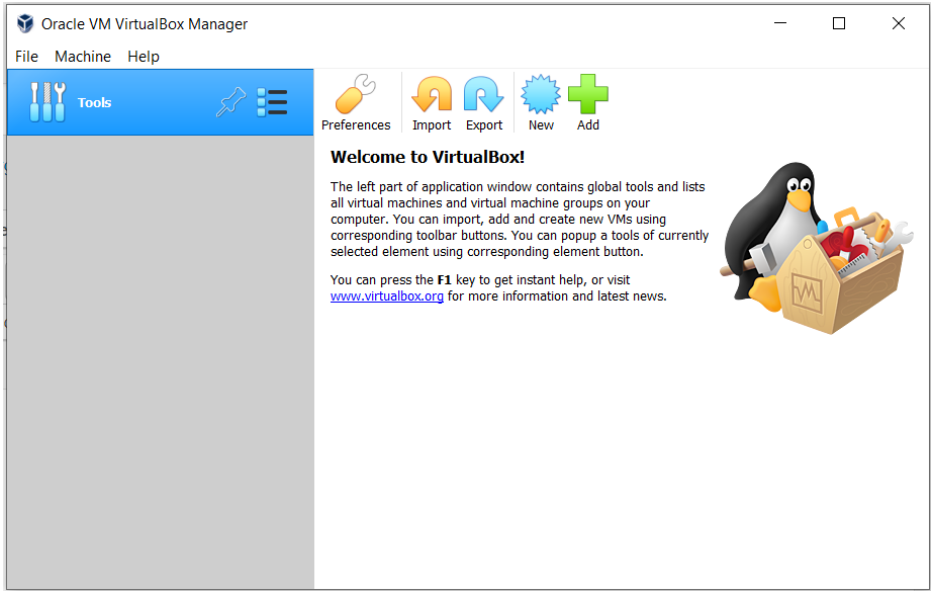
Aim:
To install Virtualbox/VMware Workstation with different flavors of linux or windows OS on top of windows7 or 8.

Procedure:
Step 1- Download link and the software for windows.
Step 2- Download the installer file
Step 3- Select Installation Location:
Step 4- Give install option
Step 4- Grant admin permission to install
Step 5- Open the installed virtual box
Step 6- Create a VM and add ubuntu in it

Output:







Result:

Thus, virtualbox/VMware Workstation is installed successfully.

Ex 4 **Setting up C Programming Environment**

08.09.22

Aim:

To install a C compiler in the virtual machine created using virtual box and execute Simple Programs.

Procedure:

1. Open Terminal (Applications-Accessories-Terminal)
2. Open gedit by typing “gedit &” on terminal and type the code in the editor
3. Type “ls” on Terminal to see all files under current folder
4. Confirm that “palindrome.c” is in the current directory. If not, type cd DIRECTORY_PATH to go to the directory that has “palindrome.c”
5. Type “gcc palindrome.c” to compile, and type “ls” to confirm that a new executable file “a.out” is created
6. Type “./a.out” on Terminal to run the program
7. If you see your output on the screen, on the next line, you just successfully ran your first C program!

Output:

Palindrome.c

```
#include<stdio.h>
int main()
{ int n,r,sum=0,temp;
scanf(“%d”,&n);
temp=n;
while(temp>0) {
 r=temp%10;
 sum=sum+r;
 temp%=10;  }
 if(num==n)
 printf(“Palindrome”);
 else
 printf(“Not a palindrome”);  }
```

Fibonacci.c

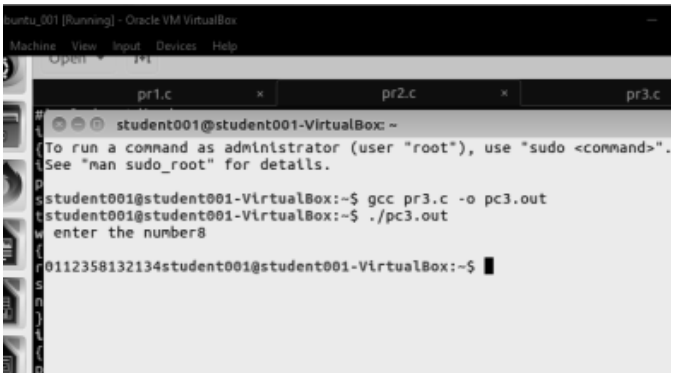
```
#include<stdio.h>
int main() {
```

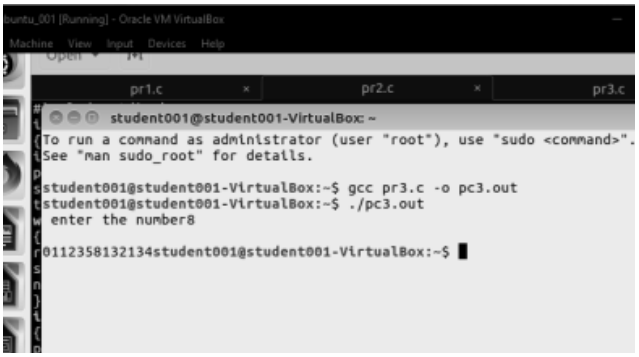
```
int n1=0,n2=1,n3,i,number;
printf("Enter the number of elements:");
scanf("%d",&number); printf("\n%d %d",n1,n2);
for(i=2;i<number;++i) {
n3=n1+n2;
printf(" %d",n3); n1=n2;
n2=n3; } }
```

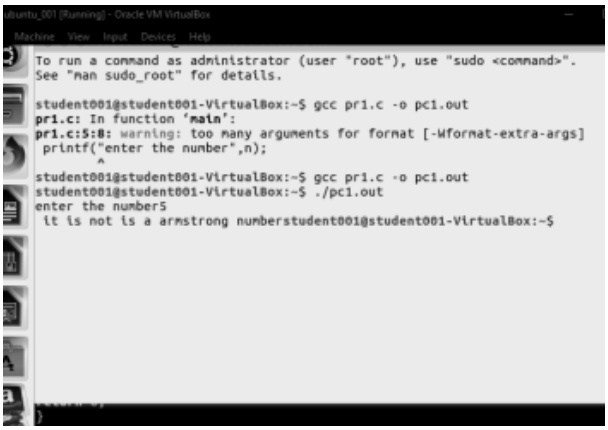
Armstrong.c

```
#include<stdio.h>
int main() {
int n,r,sum=0,temp;
printf("enter the number=");
scanf("%d",&n);
temp=n;
while(n>0) {
r=n%10;
sum=sum+(r*r*r);
n=n/10; }
if(temp==sum)
printf("armstrong number ");
else
printf("not armstrong number");  }
```

Output:







Result:

Thus, C compiler was installed in the VM and simple programs were executed successfully

Ex-5 Google App Engine Installation and Simple Programming in GAE

15.09.22

Aim:

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

Algorithm

Step 1: Install the cloud SDK in the specified link: https://cloud.google.com/sdk/docs/install

Step 2: Once the installation is done open the installed Cloud shell

Step 3: It will prompt you to login with email id of any google account, so sign in with your google account

Step 4: Choose a new project to be created by typing any name

Step 5: Create a new folder inside C drive (eg. cclabex6) Now move into that folder by the command
Cd cclabex6

Upload the code to your github repository or clone the existing repository of helloworld from the link: https://github.com/GoogleCloudPlatform/python-docs-samples/

Step 6: Change the directory which contains hello world code as:

cd C:\cclabex6\python-docs-samples\appengine\standard_python3\hello_world

Step 7: Install the requirements like flask needed for the application using the command: pip install -r requirements.txt

Step 8: Provided python is preinstalled in the system execute the command: python main.py

Step 9: This will make the application run in the localhost port :
http://127.0.0.1:8080/

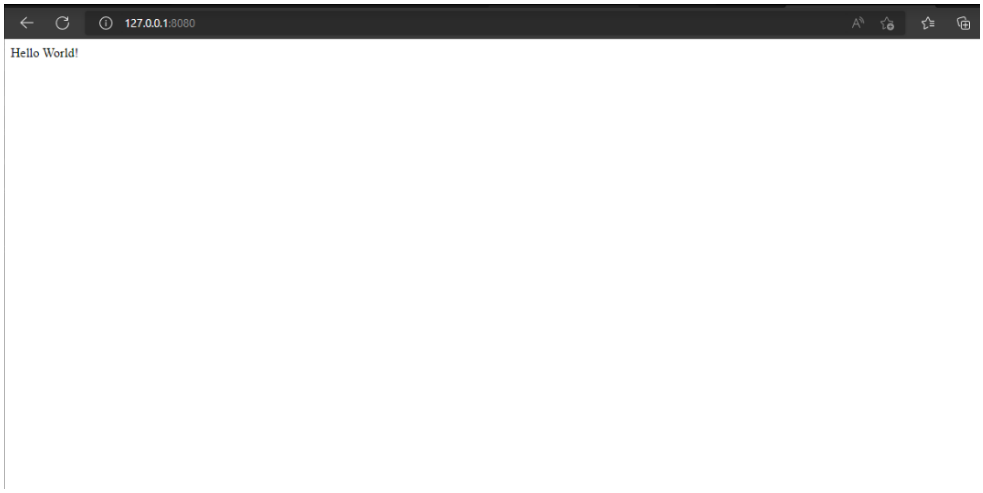
Step 10: Record the displayed webapp's output.

Output:

```
Welcome to the Google Cloud CLI! Run "gcloud -h" to get the list of available commands.
---

C:\Users\ctadmin\AppData\Local\Google\Cloud SDK>cd c:/

c:\>cd gae

c:\gae>git clone https://github.com/GoogleCloudPlatform/python-docs-
Cloning into 'python-docs-'...

c:\gae>git clone https://github.com/GoogleCloudPlatform/python-docs-samples/
Cloning into 'python-docs-samples'...
remote: Enumerating objects: 66657, done.
remote: Counting objects: 100% (368/368), done.
remote: Compressing objects: 100% (210/210), done.
remote: Total 66657 (delta 190), reused 283 (delta 151), pack-reused 66289
Receiving objects: 100% (66657/66657), 109.09 MiB | 6.72 MiB/s, done.
Resolving deltas: 100% (38107/38107), done.
Updating files: 100% (2406/2406), done.

c:\gae>cd C:\gae\python-docs-samples\appengine\standard_python3\hello_world

C:\gae\python-docs-samples\appengine\standard_python3\hello_world>pip install -r requirements.txt
'pip' is not recognized as an internal or external command,
operable program or batch file.

C:\gae\python-docs-samples\appengine\standard_python3\hello_world>python

C:\gae\python-docs-samples\appengine\standard_python3\hello_world>py
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug  1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

```
C:\Users\ctadmin\AppData\Local\Google\Cloud SDK>cd c:/

c:\>cd C:\gae\python-docs-samples\appengine\standard_python3\hello_world

C:\gae\python-docs-samples\appengine\standard_python3\hello_world>pip install -r requirements.txt
Collecting Flask==2.1.0
  Downloading Flask-2.1.0-py3-none-any.whl (95 kB)
     ---------------------------------------- 95.2/95.2 kB 1.1 MB/s eta 0:00:00
Collecting Werkzeug>=2.0
  Downloading Werkzeug-2.2.2-py3-none-any.whl (232 kB)
     ---------------------------------------- 232.7/232.7 kB 1.8 MB/s eta 0:00:00
Collecting itsdangerous>=2.0
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting Jinja2>=3.0
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
     ---------------------------------------- 133.1/133.1 kB 2.6 MB/s eta 0:00:00
Collecting click>=8.0
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
     ---------------------------------------- 96.6/96.6 kB 1.8 MB/s eta 0:00:00
Requirement already satisfied: colorama in c:\users\ctadmin\appdata\local\programs\python\python310\lib\site-packages (from click>=8.0->Flask==2.1.0->-r requirements.tx
t (line 1)) (0.4.4)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.1-cp310-cp310-win_amd64.whl (17 kB)
Installing collected packages: MarkupSafe, itsdangerous, click, Werkzeug, Jinja2, Flask
Successfully installed Flask-2.1.0 Jinja2-3.1.2 MarkupSafe-2.1.1 Werkzeug-2.2.2 click-8.1.3 itsdangerous-2.1.2

C:\gae\python-docs-samples\appengine\standard_python3\hello_world>python main.py
Python was not found; run without arguments to install from the Microsoft Store, or disable this shortcut from Settings > Manage App Execution Aliases.

C:\gae\python-docs-samples\appengine\standard_python3\hello_world>py main.py
 * Serving Flask app 'main' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
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: 105-800-528
127.0.0.1 - - [05/Sep/2022 16:58:12] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [05/Sep/2022 16:58:12] "GET /favicon.ico HTTP/1.1" 404 -
```

Activate Windows
Go to Settings to activate Windows.



Result:

Thus the Google App Engine is installed and a simple hello world app is created using python in cloud shell

Ex 6 **GAE Launcher**

22.09.22

Aim:

To use GAE launcher to launch the web applications.

Procedure:

1. Download google app engine sdk and the windows installer
2. Make a folder for your Google App Engine applications
3. Open a text editor and enter your program

Program:

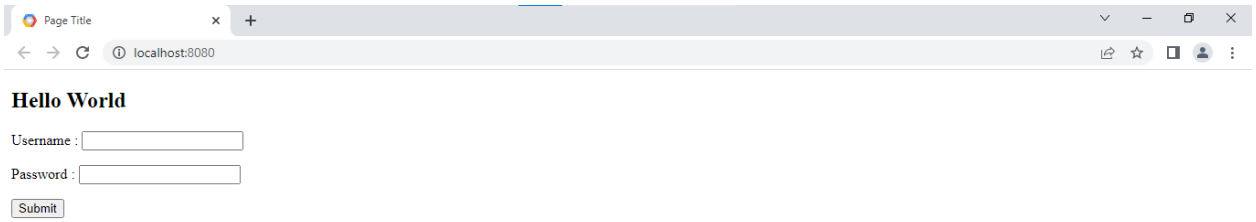
index.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"><html><head><META http-equiv="Content-Type" content="text/html; charset=utf-8"><style></style></head><body><u></u>
<div>
<h2>Hello World</h2>
<form target="_blank">
<p>
        <label>Username : <input type="text"></label>
        </p>
<p>
        <label>Password : <input type="password"></label>
        </p>
<p>
        <button type="submit">Submit</button>
        </p>
</form>
</div>
</body></html>
```

main.py

```
#!/usr/bin/env python
#
# Copyright 2007 Google Inc.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
# See the License for the specific language governing permissions
and
# limitations under the License.
#
import os
import webapp2
from google.appengine.ext.webapp import template



class MainPage(webapp2.RequestHandler):
    def get(self):
        template_values = {}
        path = os.path.join(os.path.dirname(__file__),
'index.html')
        self.response.out.write(template.render(path,
template_values))



app = webapp2.WSGIApplication([('/', MainPage)], debug=True)
```

Output:



Result:

Thus a web application was successfully launched using GAE

Exp 7 **File transfer from one virtual machine to another**

29.09.22

Aim:

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

Procedure:

1.Create two virtual Machines.

2.Follow the steps to Create NATNetwork .

File->preferences->Network-> NATNetwork -> Add NATNetwork

3. Select a NATNetwork for created VMs

Ex:VM1->settings->Network->Attached to: NATNetwork

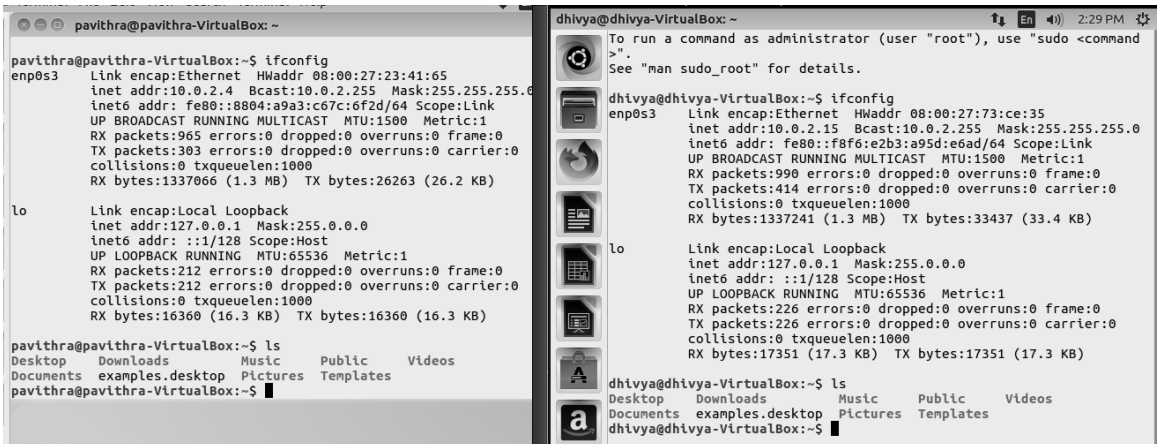
->Choose your created NATNetwork name

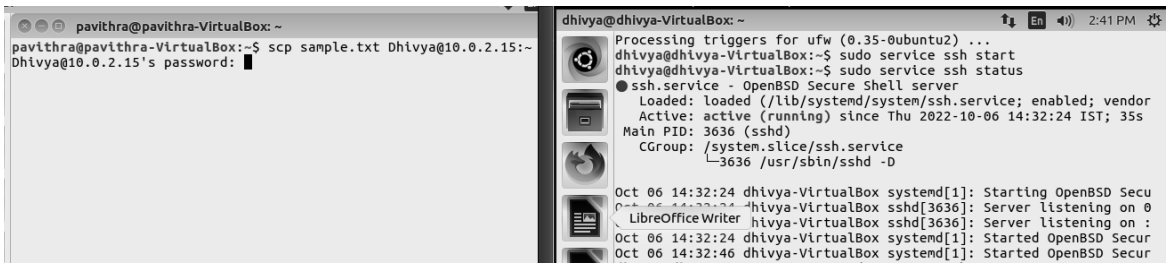
Click OK

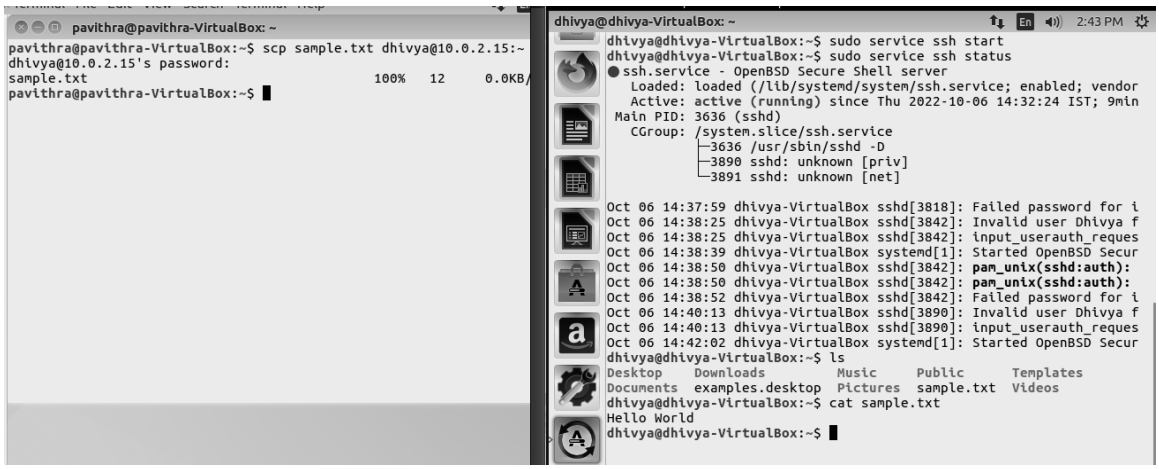
4. Open the two VMs

5. Open Terminal on your VMs

Output:







Result:

Thus, files were transferred from one virtual machine to another virtual machine successfully

Exp 8 **Hadoop Single Node Cluster Installation**
01.11.22

AIM:

To install Hadoop Single node cluster and run a simple application

ALGORITHM:

Step 1: Start
Step 2: If Java is not installed in your system then first download and install java under “C:\Java”
Step 3: Check whether Java 1.8.0 is installed on your system, use “Javac -version” to check.
Step 4: Extract file Hadoop 2.8.0.tar.gz or Hadoop-2.8.0.zip and place under “C:\hadoop-2.8.0”.
Step 5: Set the path (System Variable) HADOOP_HOME Environment variable on windows 10
Step 6: Set the path (System Variable) JAVA_HOME Environment variable on windows 10
Step 7: Set the Hadoop bin directory path and JAVA bin directory path (User variables).

Configuration

Step 1: Edit file C:/hadoop-2.8.0/etc/hadoop/core-site.xml, paste below xml paragraph and save this file.

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

Step 2: Rename “mapred-site.xml.template” to “mapred-site.xml” and edit this file C:/hadoop-2.8.0/etc/hadoop/mapred-site.xml, paste below xml paragraph and save this file.

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

Step 3: Create folder “data” under “C:\hadoop-2.8.0”

Create folder “datanode” under “C:\hadoop-2.8.0\data”

Create folder “namenode” under “C:\hadoop-2.8.0\data”

Step 4: Edit file C:\hadoop-2.8.0/etc/hadoop/hdfs-site.xml, paste below xml paragraph and save this file.

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>/hadoop-2.8.0/data/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>/hadoop-2.8.0/data/datanode</value>
</property>
</configuration>
```

Step 5: Edit file C:/hadoop-2.8.0/etc/hadoop/yarn-site.xml, paste below xml paragraph and save this file.

```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shu
ffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value
>
</property>
</configuration>
```

Step 6: Edit file C:/hadoop-2.8.0/etc/hadoop/hadoop-env.cmd by closing the command line “JAVA_HOME=%JAVA_HOME%” instead of set “JAVA_HOME=C:\Java” (On C:\java this is path to file jdk.18.0)

Hadoop Configuration

1. Download folder from the below link and extract it
https://drive.google.com/drive/folders/1qJxErQfEzVQBzOECcvzLRb8VXosfK0f1
2. Delete the folder bin on C:\hadoop-2.8.0\bin, replaced by folder bin on file just download (from Hadoop Configuration.zip).
3. Open command prompt and type the command “hdfs namenode –format”.

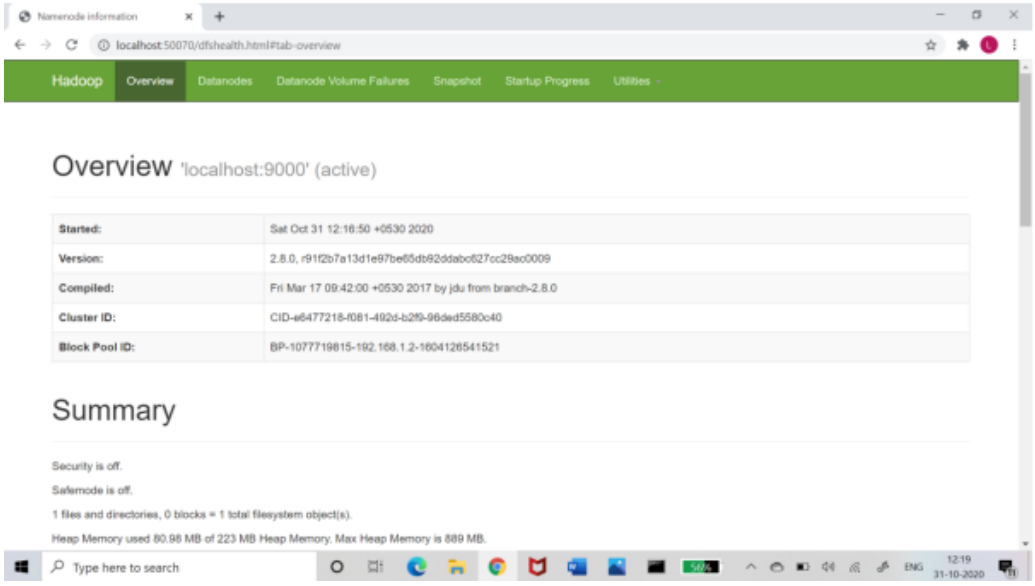
Testing

Step1: Open cmd and change directory to “C:\hadoop-2.8.0\sbin” and type “start-dfs.cmd” and “start-yarn.cmd” to start apache.

Step 2: Make sure these apps are running

- Hadoop Namenode
- Hadoop datanode
- YARN Resourc Manager
- YARN Node Manager

Step 3: Open: http://localhost:50070



RESULT:

Thus Hadoop Single node cluster is installed successfully

EX No: 9 **CloudSim Programming**

DATE:08.11.22

AIM:

To Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

PROCEDURE:

1. Go to this website and download the cloudsim simulator “ http://www.cloudbus.org/cloudsim/ ”.

a. This link would open a github account and then open the required version.

b. Download the zip file and extract it.

2. Open the netbeans create a new project

a. Choose java -> choose java application

b. Give name and then click finish

3. Click on libraries and add a library of cloudsim to the application.

4. Type the program you want to run.

5. Run the application and see the output in the output area.

PROGRAM:

```
package cloudsim;
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.LinkedList;
import java.util.List;
import org.cloudbus.cloudsim.Cloudlet;
import org.cloudbus.cloudsim.CloudletSchedulerTimeShared;
import org.cloudbus.cloudsim.Datacenter;
import org.cloudbus.cloudsim.DatacenterBroker;
import org.cloudbus.cloudsim.DatacenterCharacteristics;
import org.cloudbus.cloudsim.Host;
import org.cloudbus.cloudsim.Log;
import org.cloudbus.cloudsim.Pe;
import org.cloudbus.cloudsim.Storage;
```

```
import org.cloudbus.cloudsim.UtilizationModel;
import org.cloudbus.cloudsim.UtilizationModelFull;
import org.cloudbus.cloudsim.Vm;
import org.cloudbus.cloudsim.VmAllocationPolicySimple;
import org.cloudbus.cloudsim.VmSchedulerTimeShared;
import org.cloudbus.cloudsim.core.CloudSim;
import org.cloudbus.cloudsim.provisioners.BwProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.PeProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.RamProvisionerSimple;
/**
* A simple example showing how to create a data center with one host and run one cloudlet on
it.
*/
public class Cloudsim {
/** The cloudlet list. */
private static List&lt;Cloudlet&gt; cloudletList;
/** The vmlist. */
private static List&lt;Vm&gt; vmlist;

/**
* Creates main() to run this example.
*
* @param args the args
*/
@SuppressWarnings(&quot;unused&quot;)
public static void main(String[] args) {
Log.printLine(&quot;Starting CloudSimExample1...&quot;);
try {
// First step: Initialize the CloudSim package. It should be called

before creating any entities.

int num_user = 1; // number of cloud users
Calendar calendar = Calendar.getInstance(); // Calendar whose fields
```

have been initialized with the current date and time.
boolean trace_flag = false; // trace events
/* Comment Start - Dinesh Bhagwat
* Initialize the CloudSim library.
* init() invokes initCommonVariable() which in turn calls initialize()

(all these 3 methods are defined in CloudSim.java).

* initialize() creates two collections - an ArrayList of SimEntity

Objects (named entities which denote the simulation entities) and

* a LinkedHashMap (named entitiesByName which denote the LinkedHashMap

of the same simulation entities), with name of every SimEntity as the key.

* initialize() creates two queues - a Queue of SimEvents (future) and

another Queue of SimEvents (deferred).

* initialize() creates a HashMap of of Predicates (with integers as
keys) - these predicates are used to select a particular event from the deferred queue.
* initialize() sets the simulation clock to 0 and running (a boolean

flag) to false.

* Once initialize() returns (note that we are in method

initCommonVariable() now), a CloudSimShutDown (which is derived from SimEntity) instance is
created

* (with numuser as 1, its name as CloudSimShutDown, id as -1, and

state as RUNNABLE). Then this new entity is added to the simulation

* While being added to the simulation, its id changes to 0 (from the
earlier -1). The two collections - entities and entitiesByName are updated with this
SimEntity.

* the shutdownId (whose default value was -1) is 0
* Once initCommonVariable() returns (note that we are in method init()
now), a CloudInformationService (which is also derived from SimEntity) instance is created
* (with its name as CloudInformatinService, id as -1, and state as

RUNNABLE). Then this new entity is also added to the simulation.

* While being added to the simulation, the id of the SimEntitiy is

changed to 1 (which is the next id) from its earlier value of -1.

* The two collections - entities and entitiesByName are updated with

this SimEntity.

* the cisId(whose default value is -1) is 1
* Comment End - Dinesh Bhagwat
*/
CloudSim.init(num_user, calendar, trace_flag);
// Second step: Create Datacenters
// Datacenters are the resource providers in CloudSim. We need at
// list one of them to run a CloudSim simulation
Datacenter datacenter0 = createDatacenter("Datacenter_0");
// Third step: Create Broker
DatacenterBroker broker = createBroker();
int brokerId = broker.getId();
// Fourth step: Create one virtual machine

```
vmlist = new ArrayList&lt;Vm&gt;();
// VM description
int vmid = 0;

int mips = 1000;
long size = 10000; // image size (MB)
int ram = 512; // vm memory (MB)
long bw = 1000;
int pesNumber = 1; // number of cpus
String vmm = &quot;Xen&quot;; // VMM name
// create VM
Vm vm = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new

CloudletSchedulerTimeShared());

// add the VM to the vmList
vmlist.add(vm);
// submit vm list to the broker
broker.submitVmList(vmlist);
// Fifth step: Create one Cloudlet
cloudletList = new ArrayList&lt;Cloudlet&gt;();
// Cloudlet properties
int id = 0;
long length = 400000;
long fileSize = 300;
long outputSize = 300;
UtilizationModel utilizationModel = new UtilizationModelFull();
Cloudlet cloudlet =

new Cloudlet(id, length, pesNumber, fileSize,
outputSize, utilizationModel, utilizationModel,
utilizationModel);
cloudlet.setUserId(brokerId);
cloudlet.setVmId(vmid);
```

```
// add the cloudlet to the list
cloudletList.add(cloudlet);
// submit cloudlet list to the broker
broker.submitCloudletList(cloudletList);
// Sixth step: Starts the simulation
CloudSim.startSimulation();
CloudSim.stopSimulation();
//Final step: Print results when simulation is over
List&lt;Cloudlet&gt; newList = broker.getCloudletReceivedList();
printCloudletList(newList);
Log.printLine(&quot;CloudSimExample1 finished!&quot;);
} catch (Exception e) {
e.printStackTrace();
Log.printLine(&quot;Unwanted errors happen&quot;);
}
}
/**
* Creates the datacenter.
*
* @param name the name
*
* @return the datacenter
*/
private static Datacenter createDatacenter(String name) {
// Here are the steps needed to create a PowerDatacenter:
// 1. We need to create a list to store

// our machine
List&lt;Host&gt; hostList = new ArrayList&lt;Host&gt;();
// 2. A Machine contains one or more PEs or CPUs/Cores.
// In this example, it will have only one core.
List&lt;Pe&gt; peList = new ArrayList&lt;Pe&gt;();
int mips = 1000;
// 3. Create PEs and add these into a list.
```

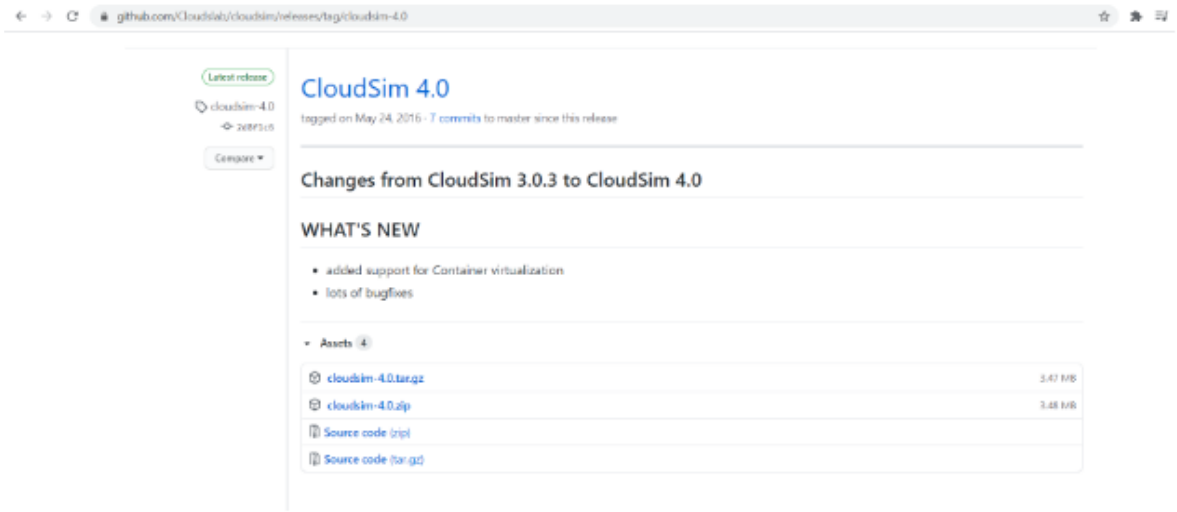
```
peList.add(new Pe(0, new PeProvisionerSimple(mips))); // need to store Pe id

and MIPS Rating

// 4. Create Host with its id and list of PEs and add them to the list
// of machines
int hostId = 0;
int ram = 2048; // host memory (MB)
long storage = 1000000; // host storage
int bw = 10000;
hostList.add(
new Host(
hostId,
new RamProvisionerSimple(ram),
new BwProvisionerSimple(bw),
storage,
peList,
new VmSchedulerTimeShared(peList)
)
); // This is our machine
// 5. Create a DatacenterCharacteristics object that stores the
// properties of a data center: architecture, OS, list of
// Machines, allocation policy: time- or space-shared, time zone
// and its price (G$/Pe time unit).
String arch = &amp;quot;x86&amp;quot;; // system architecture
String os = &amp;quot;Linux&amp;quot;; // operating system
String vmm = &amp;quot;Xen&amp;quot;;
double time_zone = 10.0; // time zone this resource located
double cost = 3.0; // the cost of using processing in this resource
double costPerMem = 0.05; // the cost of using memory in this resource
double costPerStorage = 0.001; // the cost of using storage in this
// resource
double costPerBw = 0.0; // the cost of using bw in this resource
LinkedList&amp;lt;Storage&amp;gt; storageList = new LinkedList&amp;lt;Storage&amp;gt;(); // we are not
```

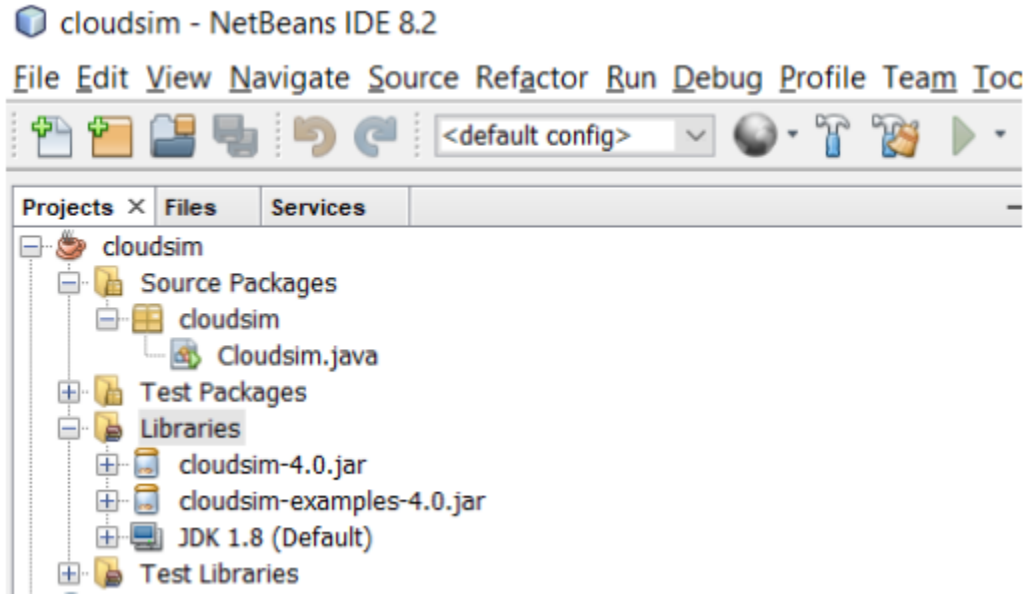
```
adding SAN
// devices by now
DatacenterCharacteristics characteristics = new DatacenterCharacteristics(
arch, os, vmm, hostList, time_zone, cost, costPerMem,
costPerStorage, costPerBw);

// 6. Finally, we need to create a PowerDatacenter object.
Datacenter datacenter = null;
try {
datacenter = new Datacenter(name, characteristics, new

VmAllocationPolicySimple(hostList), storageList, 0);

} catch (Exception e) {
e.printStackTrace();
}
return datacenter;
}
// We strongly encourage users to develop their own broker policies, to
// submit vms and cloudlets according

// to the specific rules of the simulated scenario
/**
* Creates the broker.
*
* @return the datacenter broker
*/
private static DatacenterBroker createBroker() {
DatacenterBroker broker = null;
try {
broker = new DatacenterBroker(&amp;quot;Broker&amp;quot;);
} catch (Exception e) {
e.printStackTrace();
```

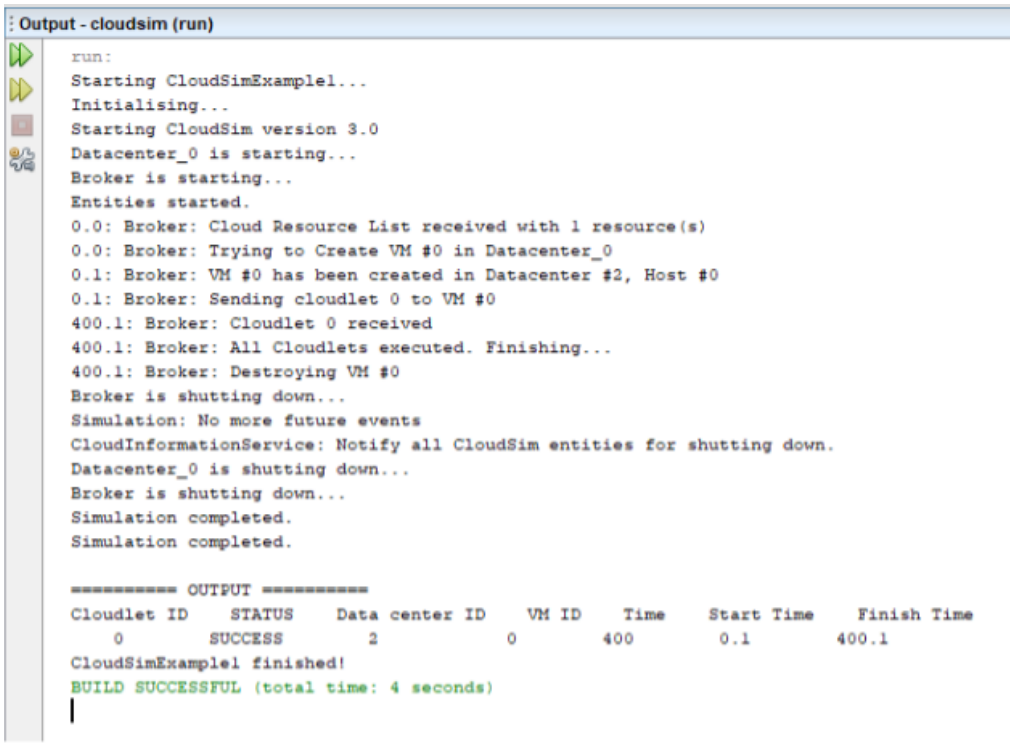
```
return null;
}
return broker;
}
/**
* Prints the Cloudlet objects.
*
* @param list list of Cloudlets
*/
private static void printCloudletList(List&lt;Cloudlet&gt; list) {
int size = list.size();
Cloudlet cloudlet;
String indent = &quot; &quot;;
Log.printLine();
Log.printLine(&quot;========== OUTPUT ==========&quot;);
Log.printLine(&quot;Cloudlet ID&quot; + indent + &quot;STATUS&quot; + indent

+ &quot;Data center ID&quot; + indent + &quot;VM ID&quot; + indent + &quot;Time&quot; +
indent
+ &quot;Start Time&quot; + indent + &quot;Finish Time&quot;);
DecimalFormat dft = new DecimalFormat(&quot;###.##&quot;);
for (int i = 0; i &lt; size; i++) {
cloudlet = list.get(i);
Log.print(indent + cloudlet.getCloudletId() + indent + indent);
if (cloudlet.getCloudletStatus() == Cloudlet.SUCCESS) {
Log.print(&quot;SUCCESS&quot;);
Log.printLine(indent + indent + cloudlet.getResourceId()
+ indent + indent + indent + cloudlet.getVmId()
+ indent + indent
+ dft.format(cloudlet.getActualCPUTime()) +

indent

+ indent +
```

```
dft.format(cloudlet.getExecStartTime())

+ indent + indent
+ dft.format(cloudlet.getFinishTime()));

}
}
}
}
```

OUTPUT:







RESULT:

The Simulation of a cloud scenario using CloudSim and running a scheduling algorithm that is not present in CloudSim was done and verified.

awdasdsdasdads Zack

EX No: 10 **Openstack Installation**

DATE:08.11.22

AIM:

To install openstack with devstack

ALGORITHM:

Step 1: Start

Step 2: Log into your Ubuntu 18.04 system using SSH protocol and update & upgrade system repositories using the following command.

apt update -y && apt upgrade -y

Step 3:Reboot the system using the command.

sudo reboot

Step 4: To create stack user execute

sudo adduser -s /bin/bash -d /opt/stack -m stack

Step 5: Run the command below to assign sudo privileges to the user

echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee

/etc/sudoers.d/stack

Step 6: Install git and download DevStack

su - stack

sudo apt install git -y

clone devstack's git repository as shown.

git clone https://git.openstack.org/openstack-dev/devstack

Step 7: Create devstack configuration file

cd devstack

Then create a local.conf configuration file.

vim local.conf

Paste the following content

[[local|localrc]]

Password for KeyStone, Database, RabbitMQ and Service

ADMIN_PASSWORD=StrongAdminSecret

DATABASE_PASSWORD=$ADMIN_PASSWORD

RABBIT_PASSWORD=$ADMIN_PASSWORD

SERVICE_PASSWORD=$ADMIN_PASSWORD

Host IP - get your Server/VM IP address from ip addr

command

HOST_IP=10.208.0.10

Save and exit the text editor.

Step 8: Install OpenStack with Devstack

Step 9: Accessing OpenStack on a web browser

https://server-ip/dashboard

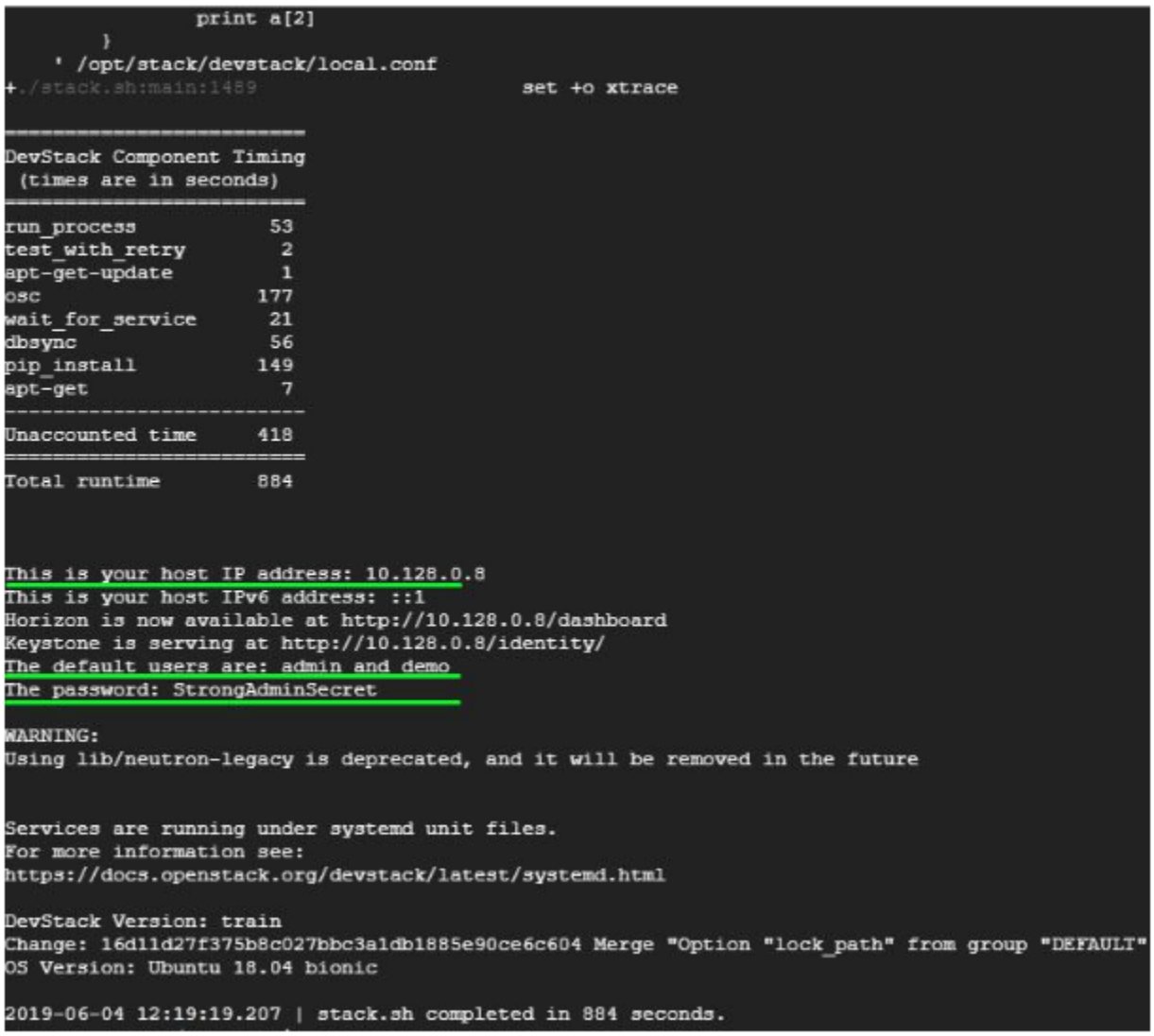
OUTPUT:

```
root@ubuntu:/# apt update -y && apt ugrade -y
Hit:1 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:5 http://archive.canonical.com/ubuntu bionic InRelease [10.2 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:7 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:8 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:9 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:10 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [627 kB]
```

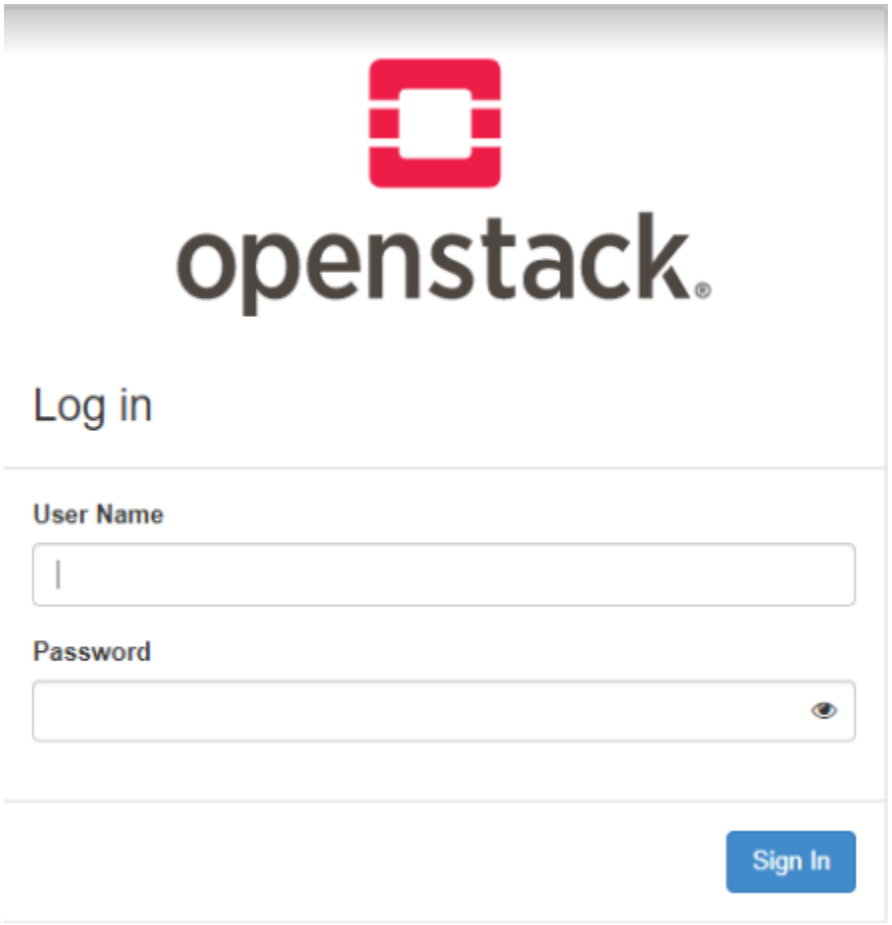
```
root@ubuntu:/# sudo useradd -s /bin/bash -d /opt/stack -m stack
root@ubuntu:/#
root@ubuntu:/# echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack
stack ALL=(ALL) NOPASSWD: ALL
root@ubuntu:/#
```

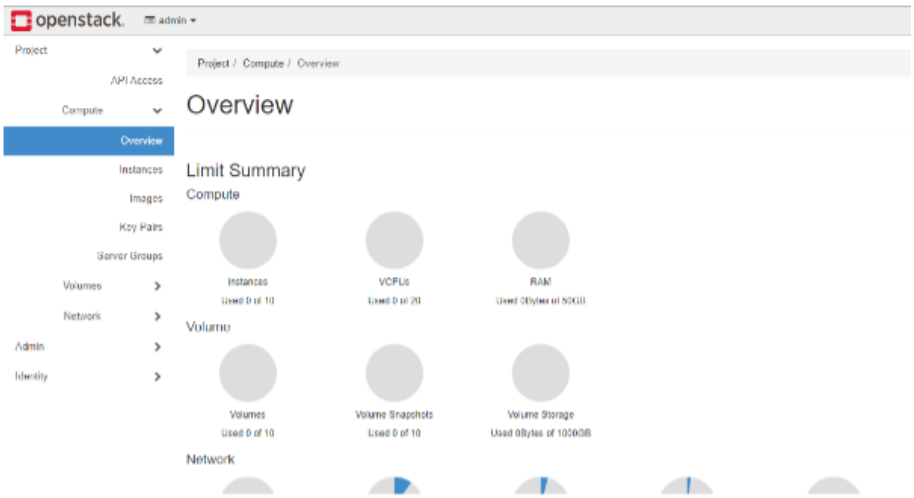
```
root@ubuntu:~# su - stack
stack@ubuntu:~$
stack@ubuntu:~$ sudo apt install git -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.17.1-1ubuntu0.4).
The following packages were automatically installed and are no longer required:
  grub-pc-bin libnuma1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

```
stack@ubuntu:~$ git clone https://git.openstack.org/openstack-dev/devstack
Cloning into 'devstack'...
warning: redirecting to https://opendev.org/openstack/devstack/
remote: Enumerating objects: 43615, done.
remote: Counting objects: 100% (43615/43615), done.
remote: Compressing objects: 100% (12575/12575), done.
remote: Total 43615 (delta 31152), reused 42370 (delta 30360)
Receiving objects: 100% (43615/43615), 8.27 MiB | 24.61 MiB/s, done.
Resolving deltas: 100% (31152/31152), done.
stack@ubuntu:~$
stack@ubuntu:~$ ls
devstack
stack@ubuntu:~$
```



awdasdsdasdads Zack





RESULT:

Thus openstack with devstack is installed successfully.