SRI SIVASUBRAMANIYA NADAR COLLEGE OF ENGINEERING

(AFFILIATED TO ANNA UNIVERSITY, CHENNAI)

RAJIV GANDHI SALAI (OMR), KALAVAKKAM – 603110

LABORATORY RECORD

IT8711-FOSS and Cloud Computing Laboratory

Name	: ....Harshavarthini K...............................
Reg. No.	: ....312217205035...................................
Dept.	: ....IT............. Sem. : ......VII...... Sec. : ...A......

ANNA UNIVERSITY

BONAFIDE CERTIFICATE

Certified that this is the bonafide record of

the practical work done for the

........**IT8711-FOSS and Cloud Computing Laboratory**......... lab

by

Name ....Harshavarthini K...............................

Register Number ....312217205035........................

of Department of ....INFORMATION TECHNOLOGY.............................

Sri Sivasubramaniya Nadar College of Engineering,

Kalavakkam, Chennai

during the academic year ....2020-2021..

Staff In-Charge   Head

Department of ....IT.............

Submitted for the University Examination held at

Sri Sivasubramaniya Nadar College of Engineering on ....

Internal Examiner   External Examiner

INDEX

Name: _____Harshavarthini K_______ Reg. No. : ___312217205035_______

Sem. : ________VII_________________ Sec. : _____A_____________________

Ex: 1 Make File Creation

Date : 19.08.2020

Objective:

To understand the Cygwin software and to understand the working of make command.

Requirement:

Cygwin software.

Theory:

Make:

make is typically used to build executable programs and libraries from source code. Generally speaking, make is applicable to any process that involves executing arbitrary commands to transform a source file to a target result. For example, make could be used to detect a change made to an image file (the source) and the transformation actions might be to convert the file to some specific format, copy the result into a content management system, and then send e-mail to a predefined set of users that the above actions were performed.

make is invoked with a list of target file names to build as command-line arguments:

make [target]

Without arguments, make builds the first target that appears in its makefile, which is traditionally a target named all.Make decides whether a target needs to be regenerated by comparing file modification times. This solves the problem of avoiding the building of files that are already up to date, but it fails when a file changes but its modification time stays in the past. Such changes could be caused by restoring an older version of a source file, or when a network filesystem is a source of files and its clock or timezone is not synchronized with the machine running make. The user must handle this situation by forcing a complete build. Conversely, if a source file's modification time is in the future, it may trigger unnecessary rebuilding.

Makefiles:

make searches the current directory for the makefile to use. GNU make searches files for a file named one of GNUmakefile, makefile, and then Makefile, and runs the specified target(s) from that file.

The makefile language is similar to declarative programming, in which necessary end conditions are described but the order in which actions are to be taken is not important. This may be confusing to programmers used to imperative programming, which explicitly describes how the end result will be reached.

One problem in build automation is the tailoring of a build process to a given platform. For instance, the compiler used on one platform might not accept the same options as the one used on another. This is not well handled by make on its own. This problem is typically handled by generating separate platform-specific build instructions, which in turn may be processed by make. Common tools for this process are autoconf and cmake.

Procedure and implementation:

- Install Cygwin software.
- Install GCC command.
- Create head.h, fun1.c, main.c and Makefile

head.h:

```
void print()
```

fun1.c

```
#include<stdio.h>

#include "head.h"

void print(){

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

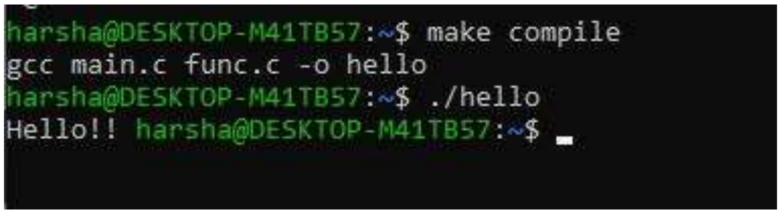
main.c

```
#include<stdio.h>

#include "head.h"

int main(){

        print();}
```

Makefile

```
all:

        gcc main.c fun1.c –o hello
```

- ❖ Run using make command.



```
harsha@DESKTOP-M41TB57:~$ make compile
gcc main.c func.c -o hello
harsha@DESKTOP-M41TB57:~$ ./hello
Hello!! harsha@DESKTOP-M41TB57:~$ 
```

Results:

The make command has been successfully executed using cygwin.

Ex: 2 **Maintaining repositories using Version Control System**

Date : 26.08.2020

Objective:

To get familiar working with Git and Github through Git GUI.

Requirement:

- ❖ Git GUI.
- ❖ Google Chrome.
- ❖ Github account.

Theory:

Version Control System:

Version control systems are a category of software tools that helps record changes to files by keeping a track of modifications done to the code.

Use of Version Control System:

- **A repository:** It can be thought as a database of changes. It contains all the edits and historical versions (snapshots) of the project.
- **Copy of Work (sometimes called as checkout):** It is the personal copy of all the files in a project. You can edit to this copy, without affecting the work of others and you can finally commit your changes to a repository when you are done making your changes.

Purpose of Version Control:

- Multiple people can work simultaneously on a single project. Everyone works on and edits their own copy of the files and it is up to them when they wish to share the changes made by them with the rest of the team.
- It also enables one person to use multiple computers to work on a project, so it is valuable even if you are working by yourself.
- It integrates the work that is done simultaneously by different members of the team. In some rare case, when conflicting edits are made by two people to the same line of a file, then human assistance is requested by the version control system in deciding what should be done.

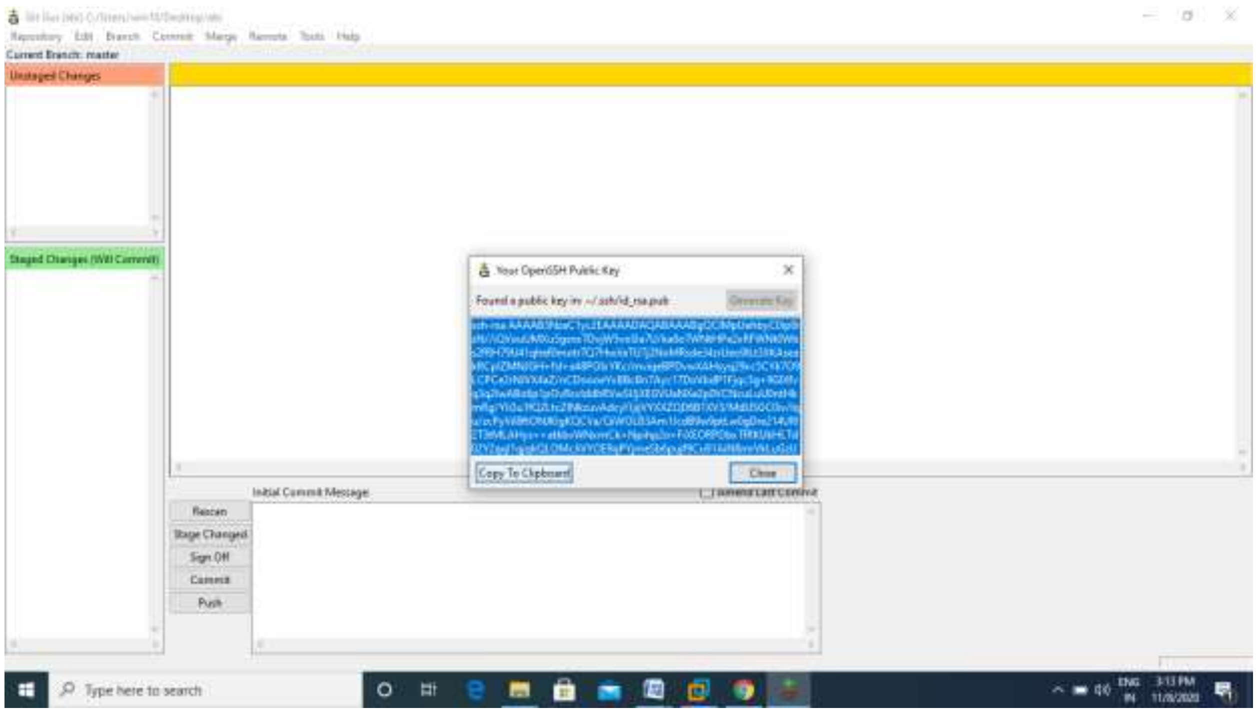
- Version control provides access to the historical versions of a project. This is insurance against computer crashes or data loss. If any mistake is made, you can easily roll back to a previous version. It is also possible to undo specific edits that too without losing the work done in the meanwhile. It can be easily known when, why, and by whom any part of a file was edited.

GIT:

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning fast performance .

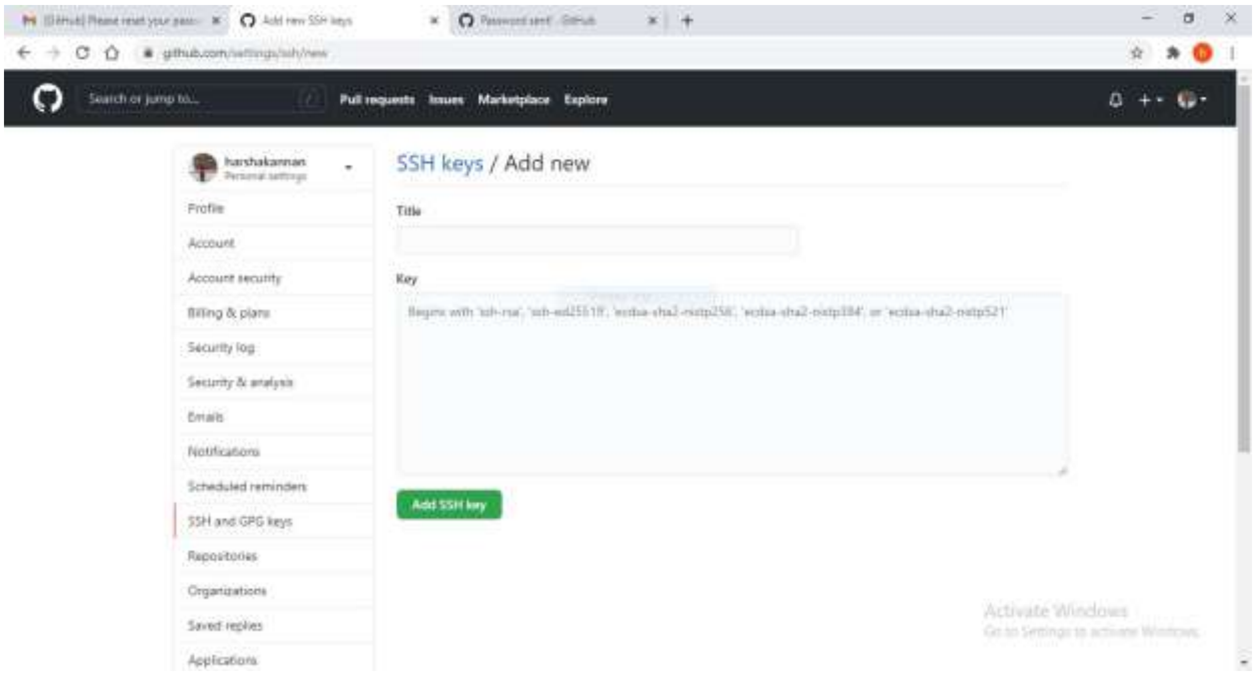
Procedure and implementation:

- The first two things you'll want to do are install git and create a free GitHub account.
- Install Git
- First, you need to install on windows; to do so, follow the below steps.
- Download and install the latest version of https://github.com/git-forwindows/git/releases/download/v2.28.0.windows.1/Git-2.28.0-64-bit.exe
- Use the default options for each step in the installation
- Remove Git Bash Desktop Icon
- Go to Start -> All Programs -> Git -> Git GUI and make a Desktop Shortcut.
- Now click on Show SSH Key under the Help Menu.
- It's possible that there is already a SSH key on your computer; it's best to remove or backup the key if you do not know where it came from.
- To do so, simply remove all files within: C:\Users\\.ssh. Be sure to replace with your Windows username. You can generate a SSH key by clicking on the Generate Key button.
- When you do so, you will need to supply a passphrase for security purposes. Remember this passphrase; you will need to use it later.

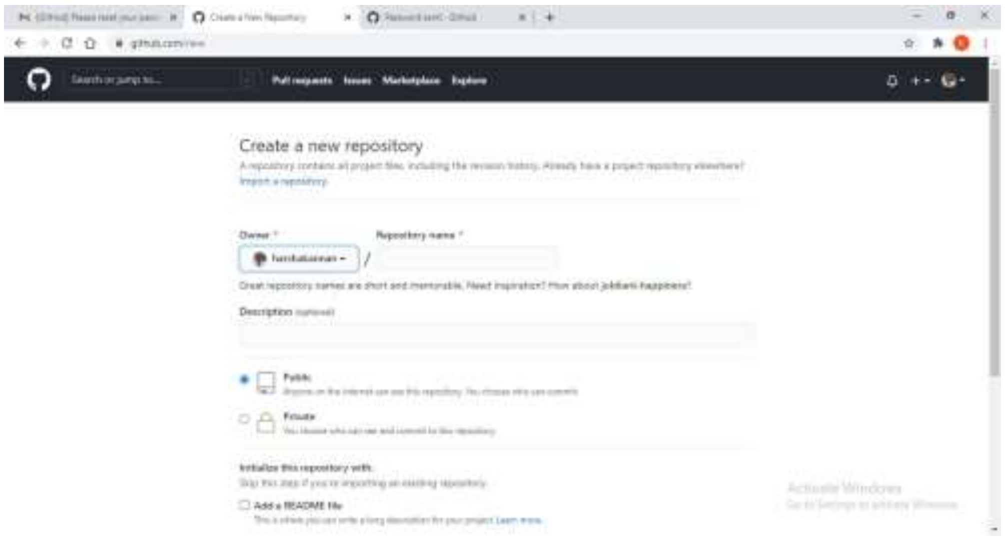


❖ click the "Copy to Clipboard" button.

❖ Next, you need to provide your hosted repo service with your public SSH key. Similar to Github, most of these sites usually have a tab, called "SSH Keys". Click the tab and add your SSH key to the website.



❖ create new remote repository on Github



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
Import a repository.

Owner * Repository name *

harshakannan /

Great repository names are short and memorable. Need inspiration? How about [illegible]

Description (optional)

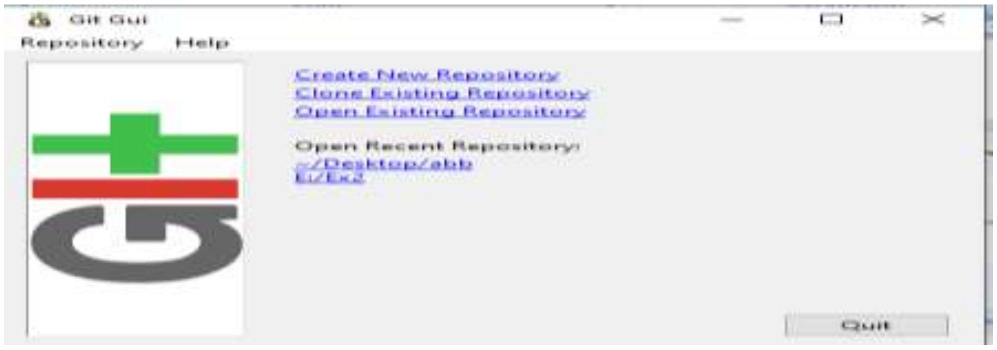
● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

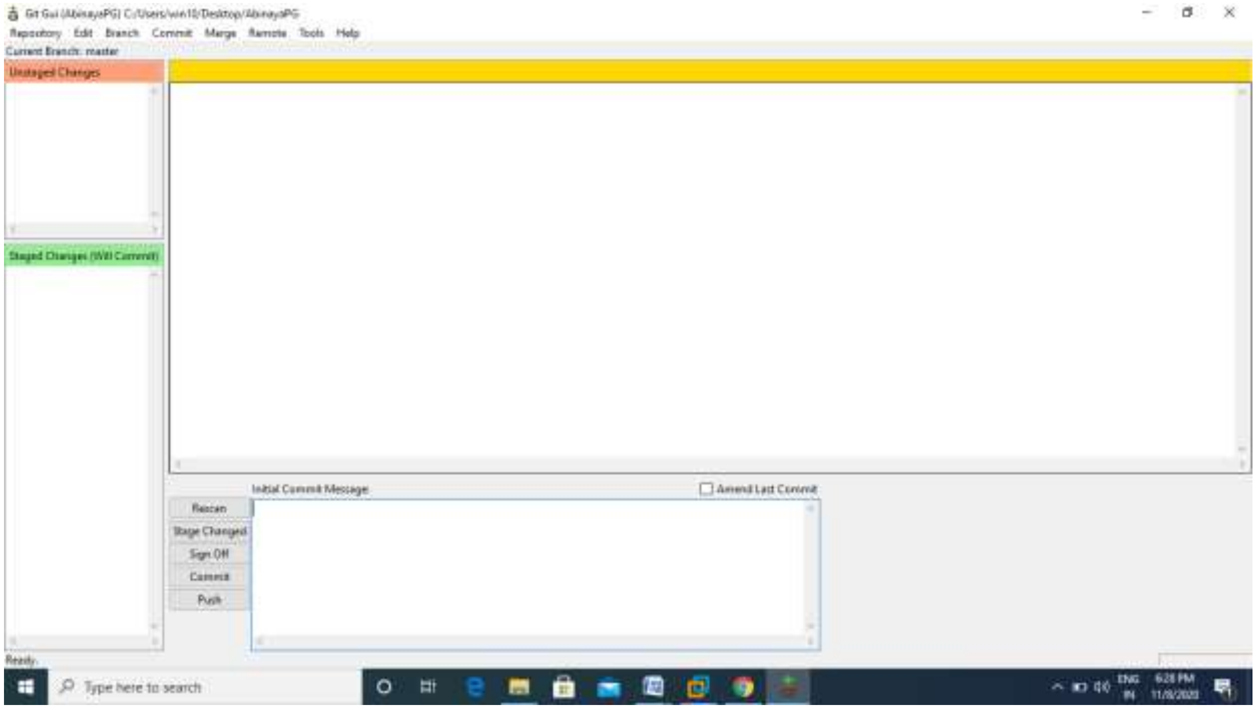
Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ Add a README file
This is where you can write a long description for your project. Learn more.

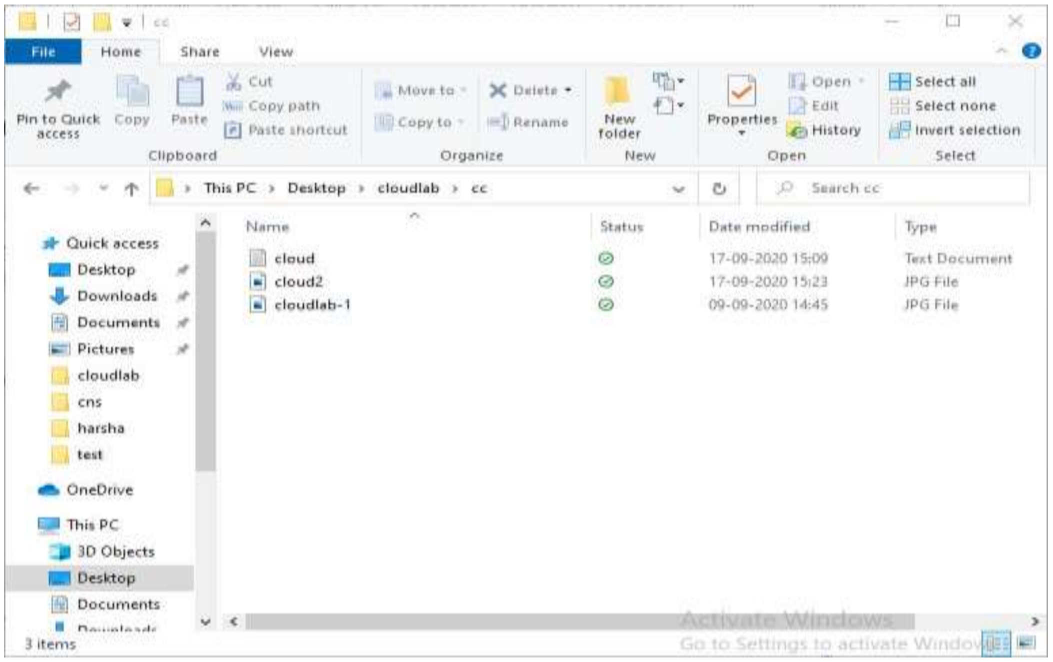
❖ Create a Local Repository In the Git GUI; clicking on "Create New Repository".



❖ Create new repository

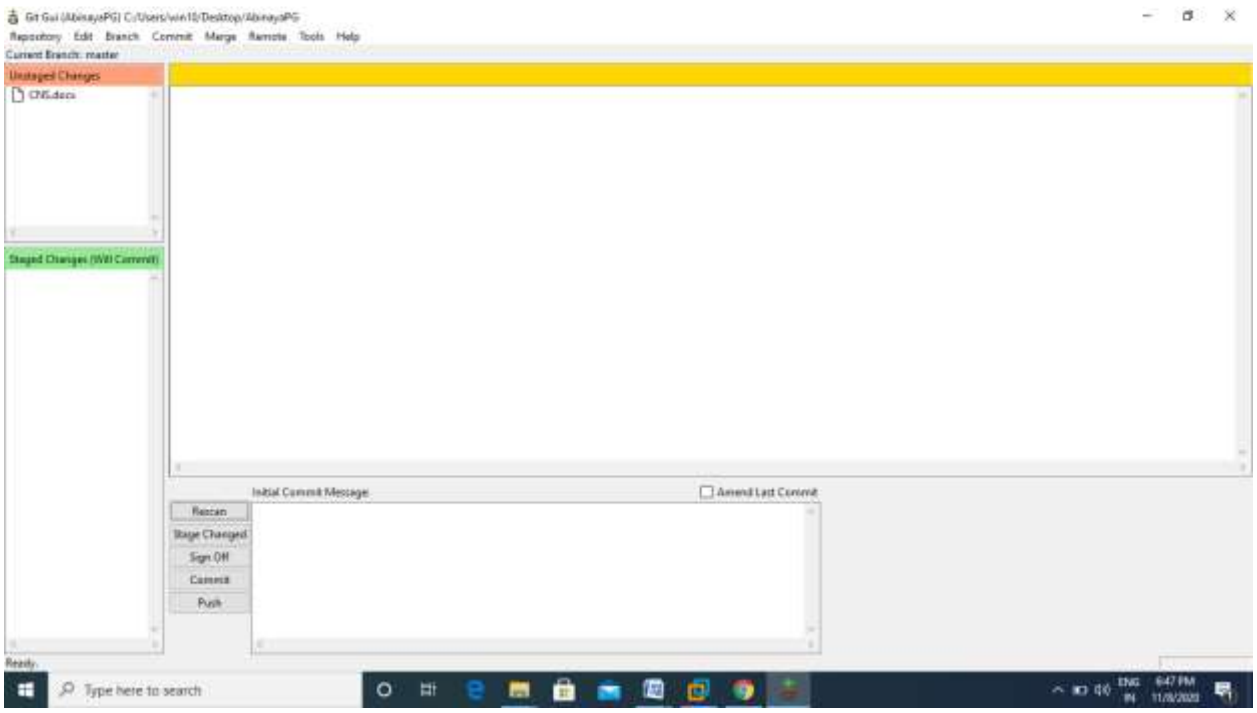

Git Gui (AbinayaPG) C:/Users/win10/Desktop/AbinayaPG
Repository Edit Branch Commit Merge Remote Tools Help
Current Branch: master
Unstaged Changes
Staged Changes (Will Commit)
Initial Commit Message
Amend Last Commit
Rescan
Stage Changed
Sign Off
Commit
Push
Ready.

❖ Add a file in that folder.

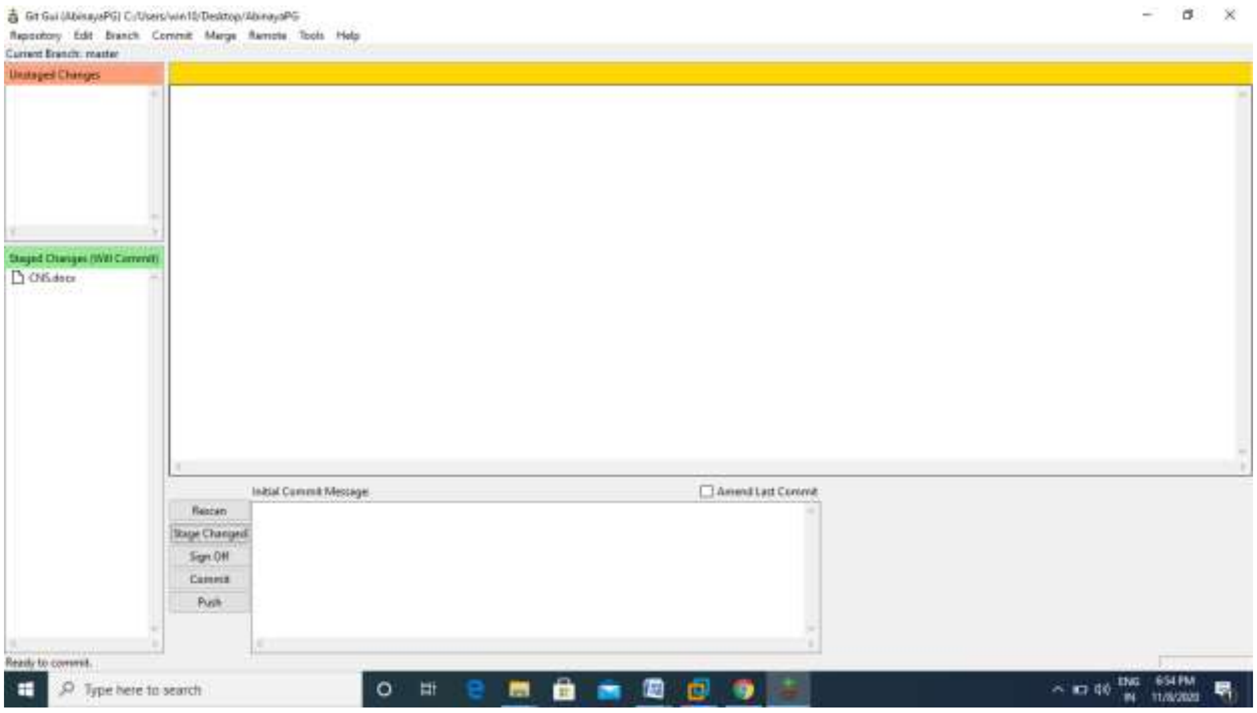


Name	Status	Date modified	Type
cloud	✓	17-09-2020 15:09	Text Document
cloud2	✓	17-09-2020 15:23	JPG File
cloudlab-1	✓	09-09-2020 14:45	JPG File

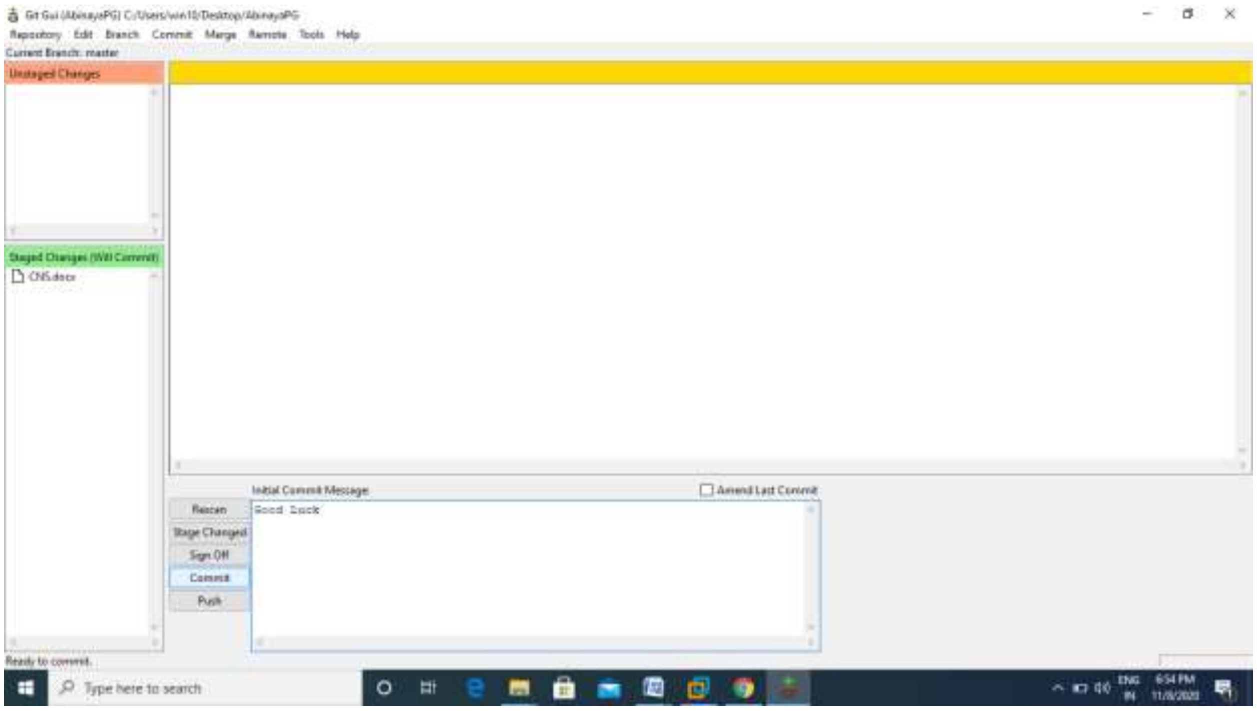
❖ Click rescan.


Git Gui (AbinayaPG) C:/Users/win10/Desktop/AbinayaPG
Repository Edit Branch Commit Merge Remote Tools Help
Current Branch: master
Unstaged Changes
CNS.docx
Staged Changes (Will Commit)
Initial Commit Message
Amend Last Commit
Rescan
Stage Changed
Sign Off
Commit
Push
Ready.

❖ Click Stage Changed.



❖ Type message and click commit.


Git Gui (AbinayaPG) C:/Users/win10/Desktop/AbinayaPG
Repository Edit Branch Commit Merge Remote Tools Help
Unstaged Changes
Staged Changes (Will Commit)
Rescan
Stage Changed
Sign Off
Commit
Push
Amend Last Commit

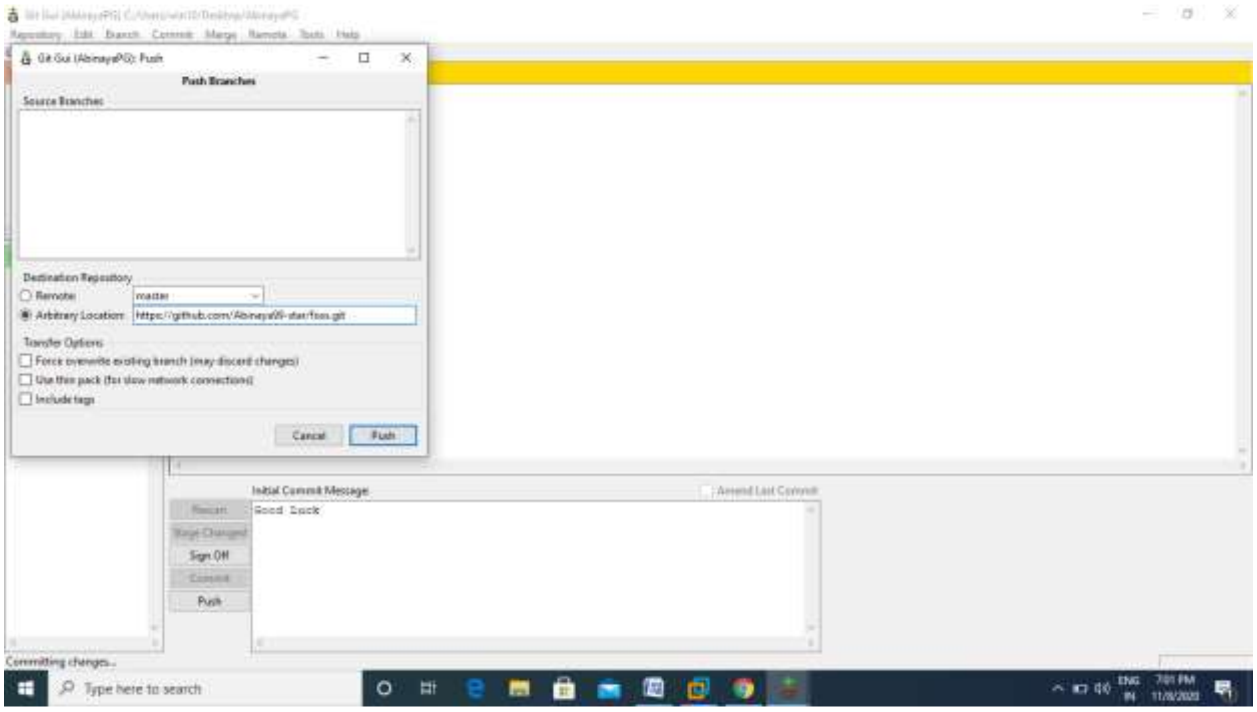
Current Branch: master

Staged Changes (Will Commit): CNS.docx

Initial Commit Message: Good luck

Ready to commit.

❖ Click Push and add arbitrary location as github repositary link.


Git Gui (AbinayaPG): Push
Push Branches
Source Branches
Destination Repository
Transfer Options
Force overwrite existing branch (may discard changes)
Use thin pack (for slow network connections)
Include tags
Cancel
Push

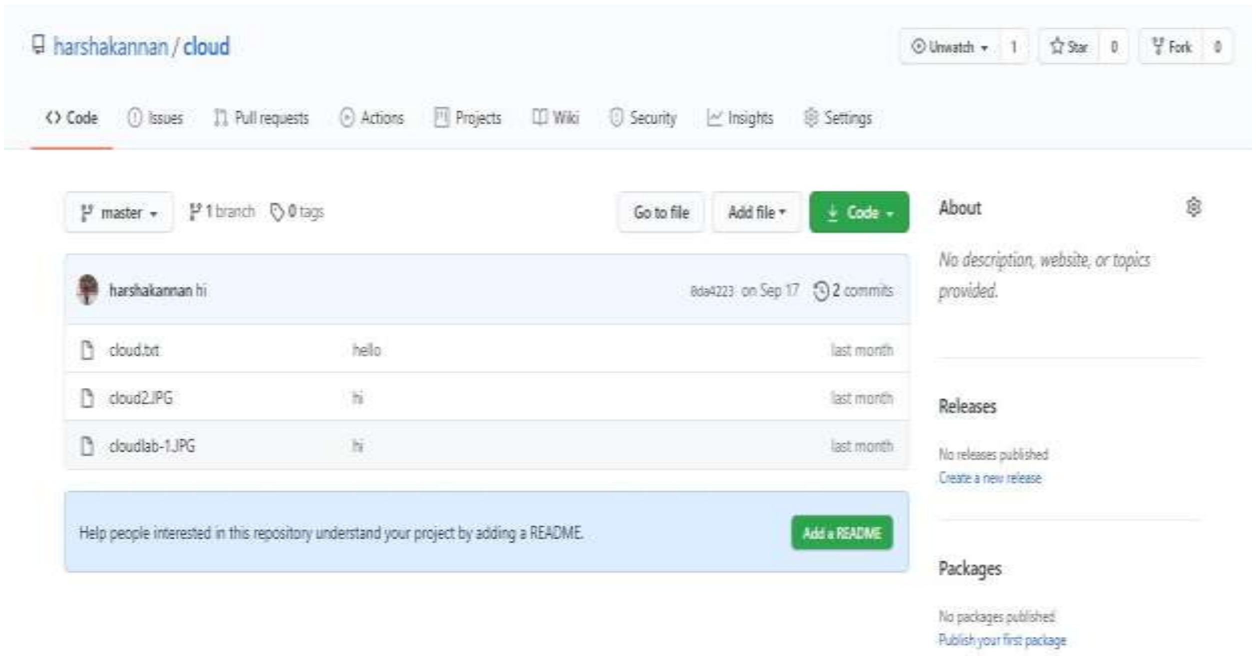
Remote: master

Arbitrary Location: https://github.com/Abinaya09-star/foss.git

Initial Commit Message: Good luck

Committing changes...

❖ Now the files are copied to github repository.



harshakannan / cloud

Unwatch 1 | Star 0 | Fork 0

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

master | 1 branch | 0 tags | Go to file | Add file | Code

harshakannan hi — 8da4223 on Sep 17 — 2 commits

cloud.txt	hello	last month
cloud2.JPG	hi	last month
cloudlab-1.JPG	hi	last month

Help people interested in this repository understand your project by adding a README. Add a README

About
No description, website, or topics provided.

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

Results:

Thus creating and maintaining repositories using Version control system has been successfully done.

Ex: 3 Virtualbox /VMware Workstation

Date : 02.09.2020

Objective:

To understand the working of WMware Workstation and to understand the installation procedure and creation of VMs.

Requirement:

- ❖ Google Chrome (or) Related Internet Explorer.
- ❖ Ubuntu iso file.

Theory:

Architecture:

VMware Workstation virtualizes I/O devices using a novel design called the *Hosted Virtual Machine Architecture*. The primary feature of this design is that it takes advantage of a pre-existing operating system for I/O device support and still achieves near native performance for CPU-intensive workloads.

VMware Workstation installs like a normal application on an operating system, known as the host operating system. When run, the application portion (**VMApp**) uses a driver loaded into the host operating system (**VMDriver**) to establish the privileged virtual machine monitor component (**VMM**) that runs directly on the hardware. From then on, a given physical processor is executing either the host world or the VMM world, with the VMDriver facilitating the transfer of control between the two worlds. A world switch between the VMM and the host worlds involves saving and restoring *all* user and system visible state on the CPU, and is thus more heavyweight than a normal process switch.

In this architecture, the CPU virtualization is handled by the VMM. A guest application or operating system performing pure computation runs just like a traditional mainframe-style virtual machine system. However, whenever the guest performs an I/O operation, the VMM will intercept it and switch to the host world rather than accessing the native hardware directly. Once in the host world, the VMApp will perform the I/O on behalf of the virtual machine through appropriate system calls. For example, an attempt by the guest to fetch sectors from its disk will become a read() issued to the host for the corresponding data. The VMM also yields control to the host OS upon receiving a hardware interrupt. The hardware interrupt is reasserted in the host world so that the host OS will process the interrupt as if it came directly from hardware.

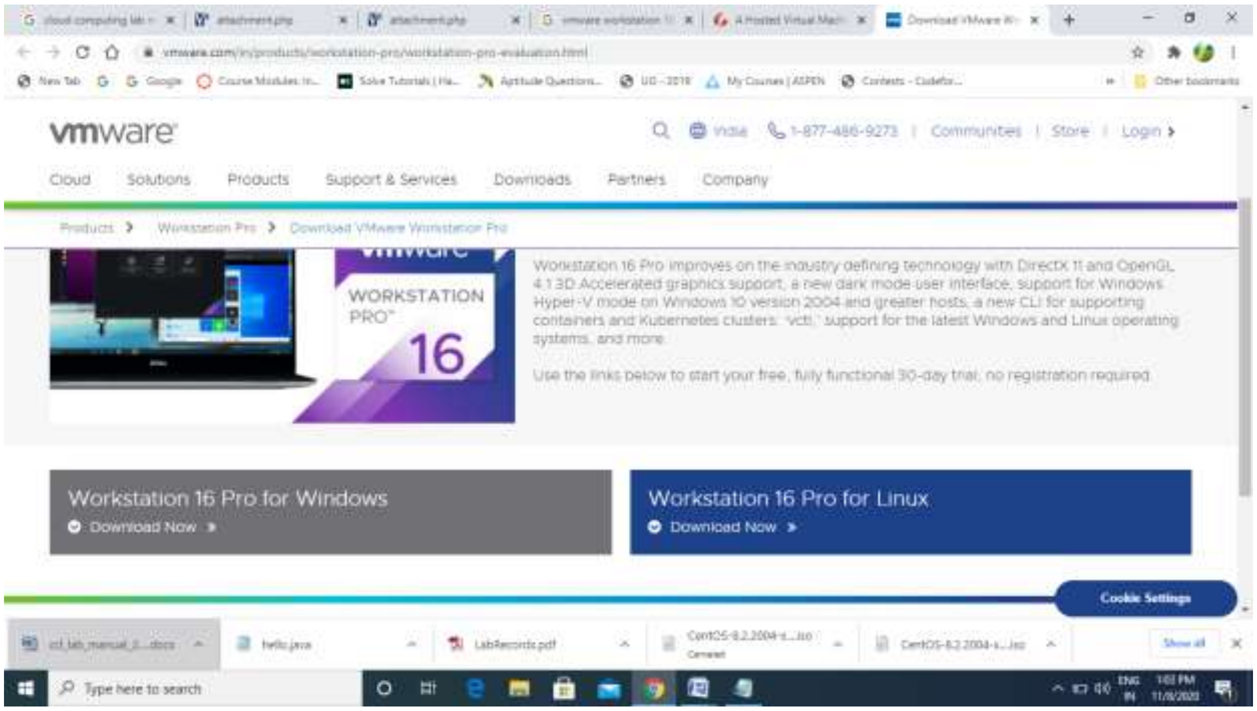
The hosted architecture is a powerful way for a PC-based virtual machine monitor to cope with the vast array of available hardware. One of the primary purposes of an operating system is to present applications with an abstraction of the hardware that allows hardware-independent code to access the underlying devices. For example, a program to play audio CD-ROMs will work on both IDE and SCSI CD-ROM drives because operating systems provide an abstract CD-ROM interface. VMware Workstation takes advantage of this generality to run on whole classes of hardware without itself needing special device drivers for each possible device.

The most significant trade-off of a hosted architecture is in potential I/O performance degradation. Because I/O emulation is done in the host world, a virtual machine executing an I/O intensive workload can accrue extra CPU time switching between the VMM and host worlds, as well as significant time in the host world performing I/O to the native hardware. This increases the CPU overhead associated with any I/O operation.

Another trade-off of the hosted architecture is that the host OS is in full control of machine resources. Even though the VMM has full system and hardware privileges, it behaves cooperatively and allows the host OS to schedule it. The host OS can also page out the memory allocated to a particular virtual machine except for a small set of pages that the VMM has pinned on behalf of the virtual machine. This allows VMware Workstation to be treated by the host OS like a regular application, but occasionally at the expense of performance if the host OS makes poor resource scheduling choices for the virtual machine.

Procedure and implementation:

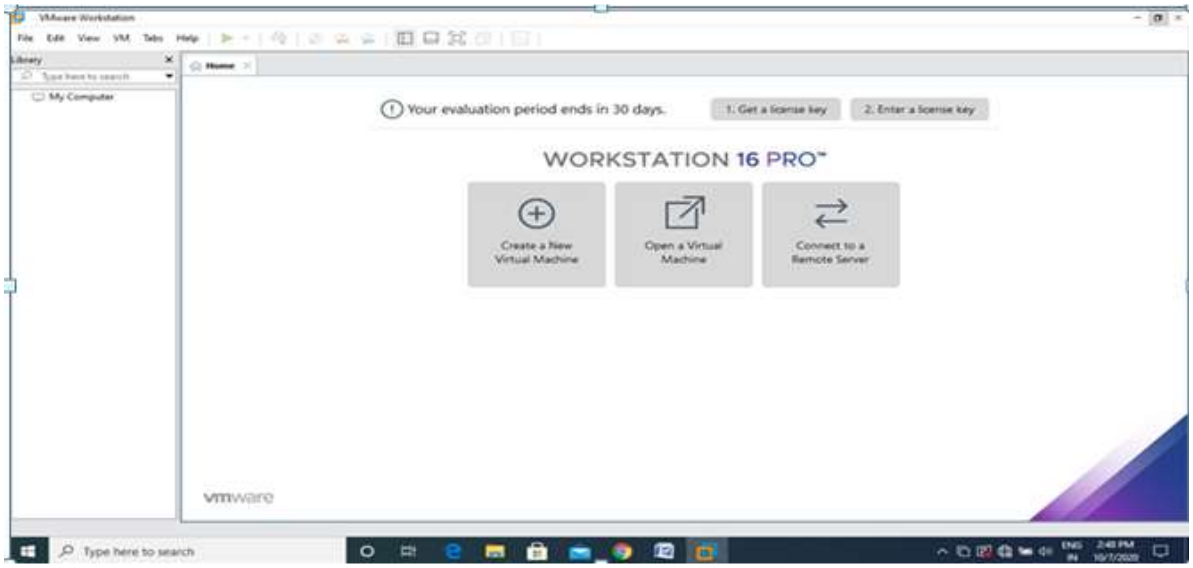
- ❖ Go to https://www.vmware.com/in/products/workstation-pro/workstation-pro-evaluation.html



Workstation 16 Pro improves on the industry defining technology with DirectX 11 and OpenGL 4.1 3D Accelerated graphics support, a new dark mode user interface, support for Windows Hyper-V mode on Windows 10 version 2004 and greater hosts, a new CLI for supporting containers and Kubernetes clusters: 'vctl,' support for the latest Windows and Linux operating systems, and more.

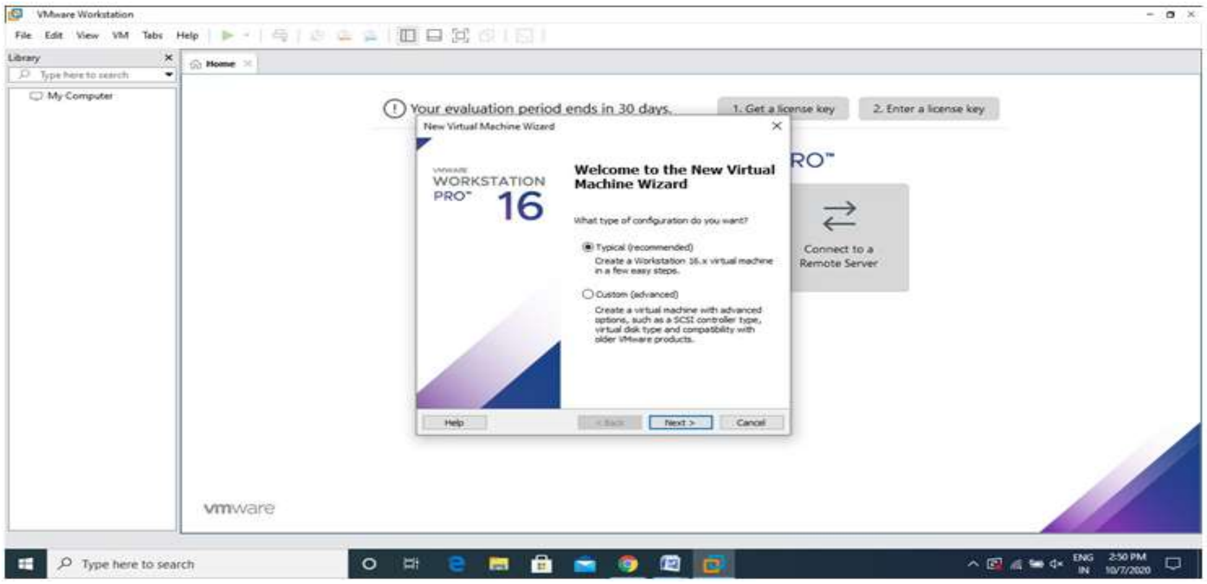
Use the links below to start your free, fully functional 30-day trial; no registration required.

❖ Download workstation 16 Pro for Windows.
❖ Once downloaded install it.



Your evaluation period ends in 30 days.

❖ Create a new Virtual Machine and use ubuntu iso file for creation of ubuntu VM.



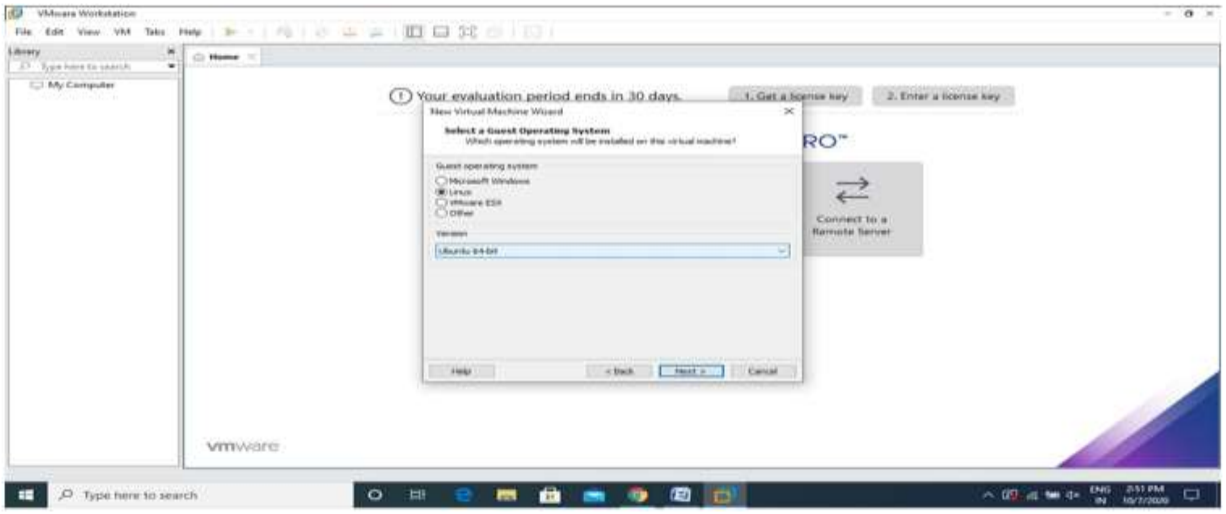
Your evaluation period ends in 30 days.

Welcome to the New Virtual Machine Wizard

What type of configuration do you want?

(●) Typical (recommended)
Create a Workstation 16.x virtual machine in a few easy steps.

( ) Custom (advanced)
Create a virtual machine with advanced options, such as a SCSI controller type, virtual disk type and compatibility with older VMware products.



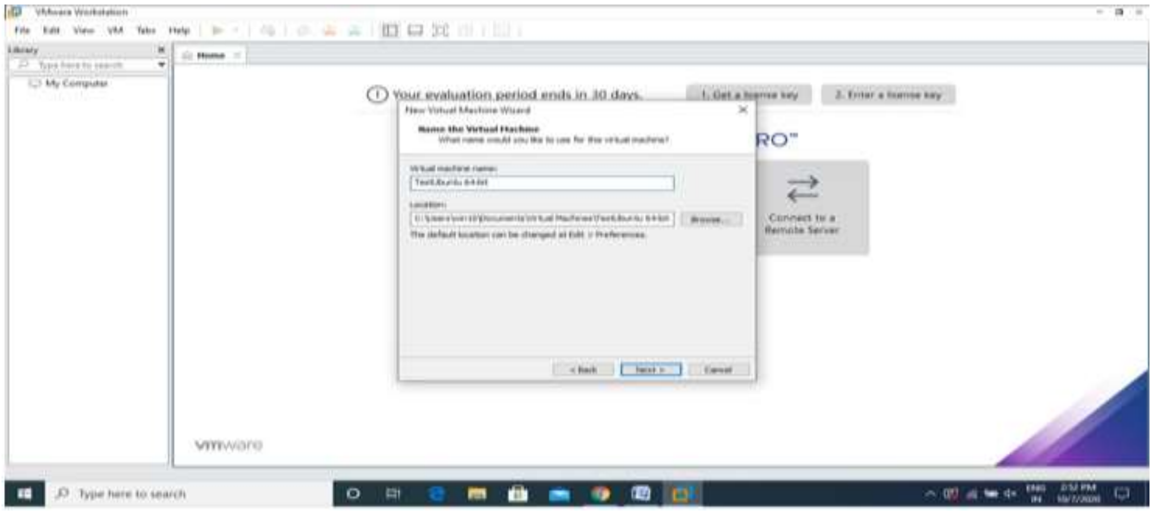
Your evaluation period ends in 30 days.

Select a Guest Operating System
Which operating system will be installed on this virtual machine?

Guest operating system
( ) Microsoft Windows
(●) Linux
( ) VMware ESX
( ) Other

Version
Ubuntu 64-bit

Harshavarthini K
312217205035



Your evaluation period ends in 30 days.

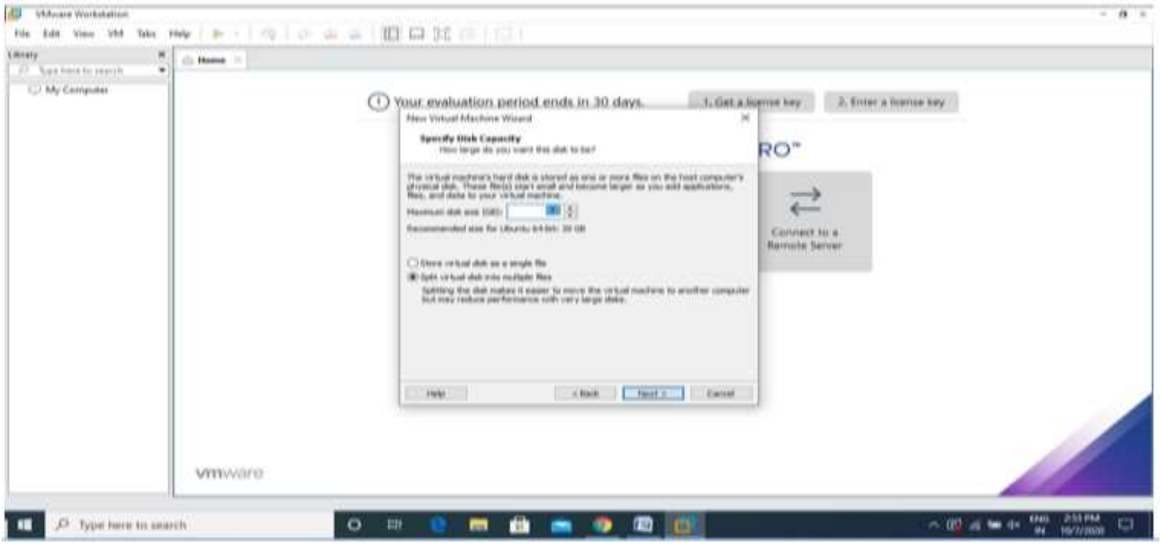
New Virtual Machine Wizard

Name the Virtual Machine
What name would you like to use for the virtual machine?

Virtual machine name:
TestUbuntu 64-bit

Location:
[illegible]\Documents\Virtual Machines\TestUbuntu 64-bit

The default location can be changed at Edit > Preferences.



Your evaluation period ends in 30 days.

New Virtual Machine Wizard

Specify Disk Capacity
How large do you want this disk to be?

The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

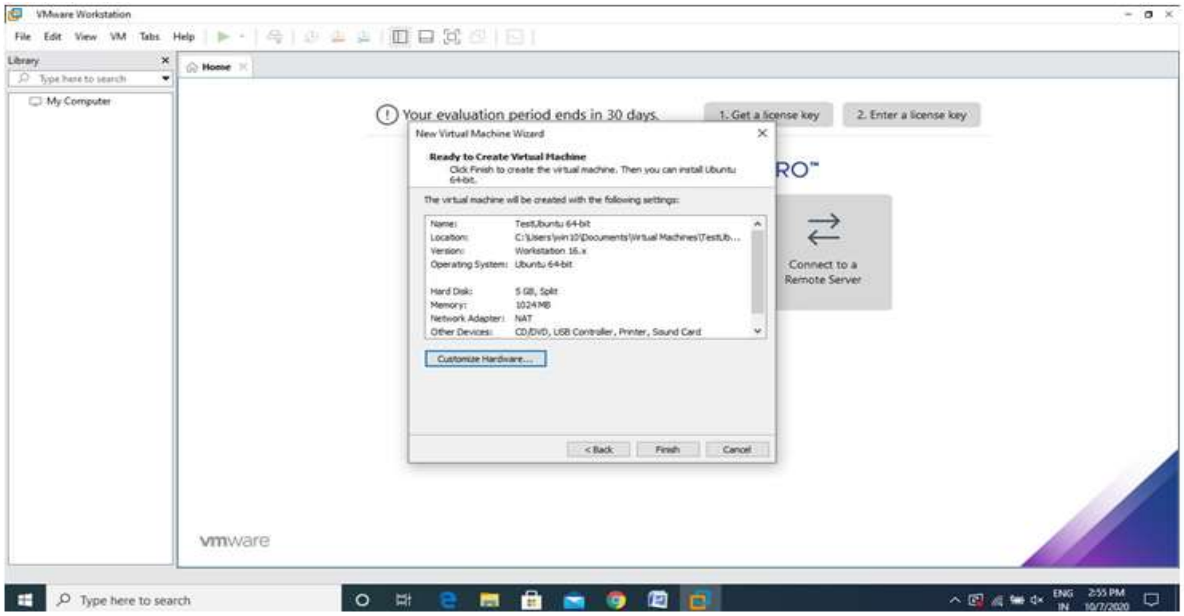
Maximum disk size (GB): 30

Recommended size for Ubuntu 64-bit: 20 GB

( ) Store virtual disk as a single file
(●) Split virtual disk into multiple files

Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

Harshavarthini K
312217205035



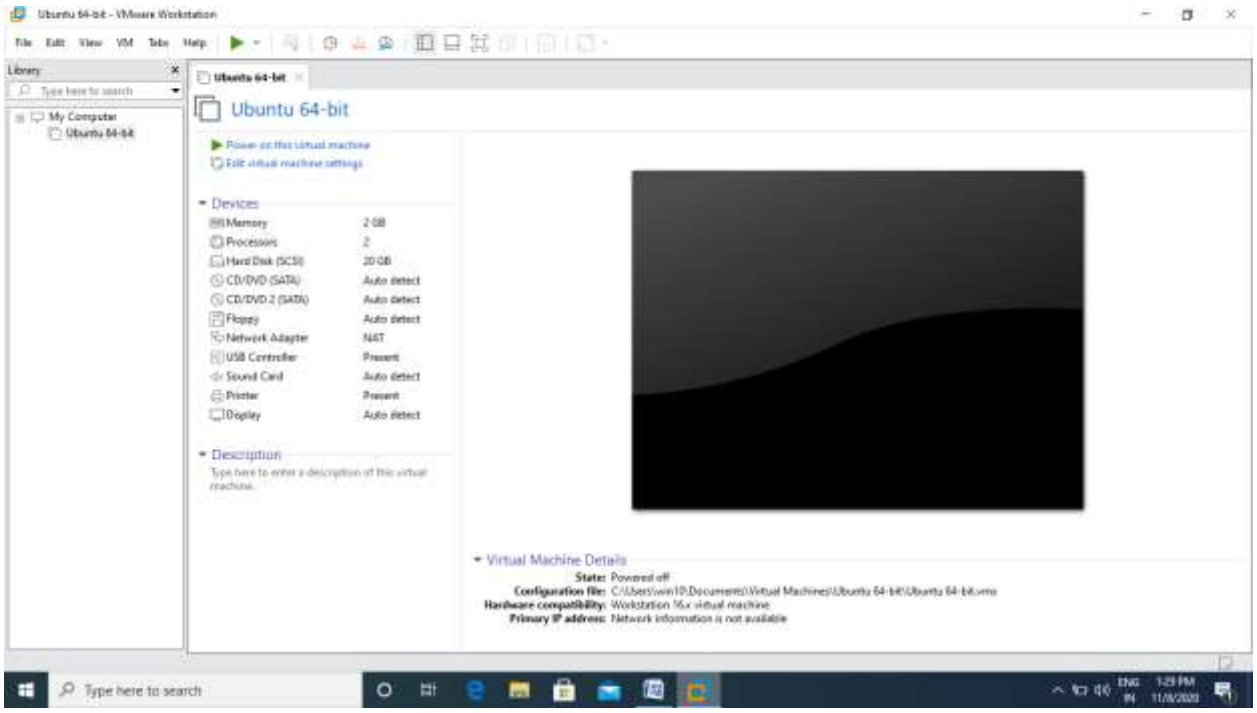
Your evaluation period ends in 30 days.

New Virtual Machine Wizard

Ready to Create Virtual Machine
Click Finish to create the virtual machine. Then you can install Ubuntu 64-bit.

The virtual machine will be created with the following settings:

Name:	TestUbuntu 64-bit
Location:	C:\Users\win 10\Documents\Virtual Machines\TestUb...
Version:	Workstation 16.x
Operating System:	Ubuntu 64-bit
Hard Disk:	5 GB, Split
Memory:	1024 MB
Network Adapter:	NAT
Other Devices:	CD/DVD, USB Controller, Printer, Sound Card


Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Ubuntu 64-bit
Ubuntu 64-bit
Power on this virtual machine
Edit virtual machine settings
Type here to search

Ubuntu 64-bit

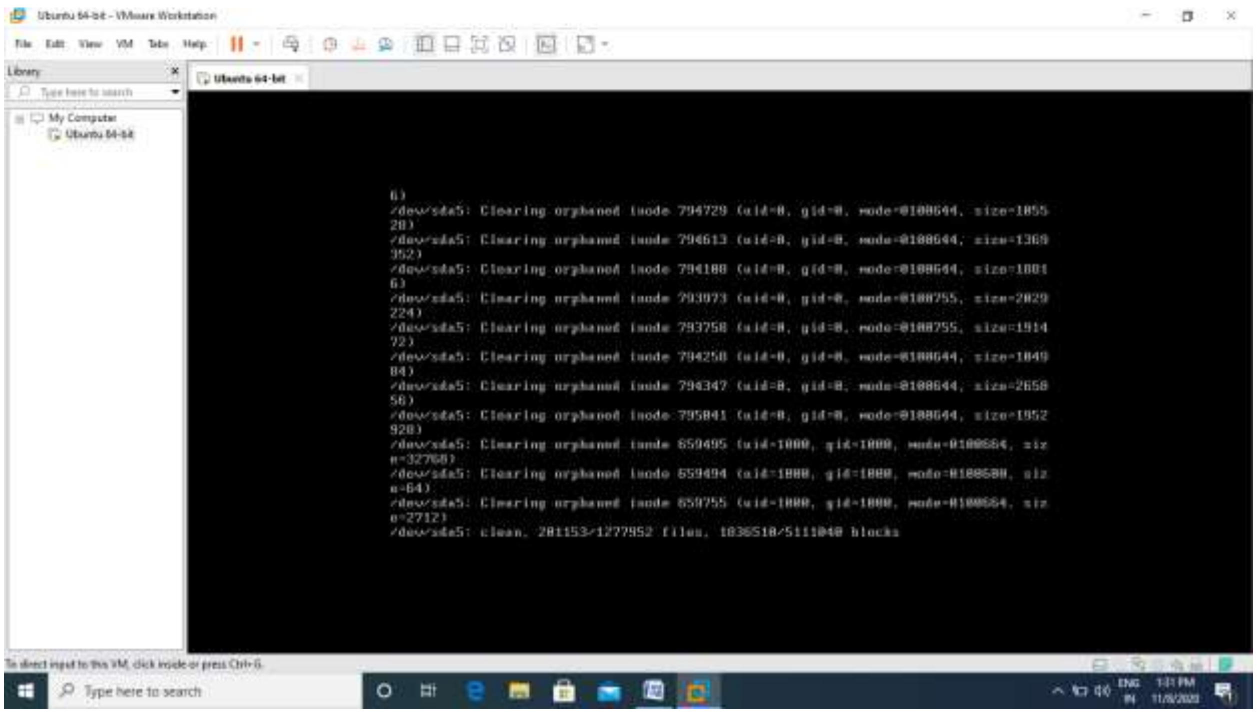
Devices

Memory	2 GB
Processors	2
Hard Disk (SCSI)	20 GB
CD/DVD (SATA)	Auto detect
CD/DVD 2 (SATA)	Auto detect
Floppy	Auto detect
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

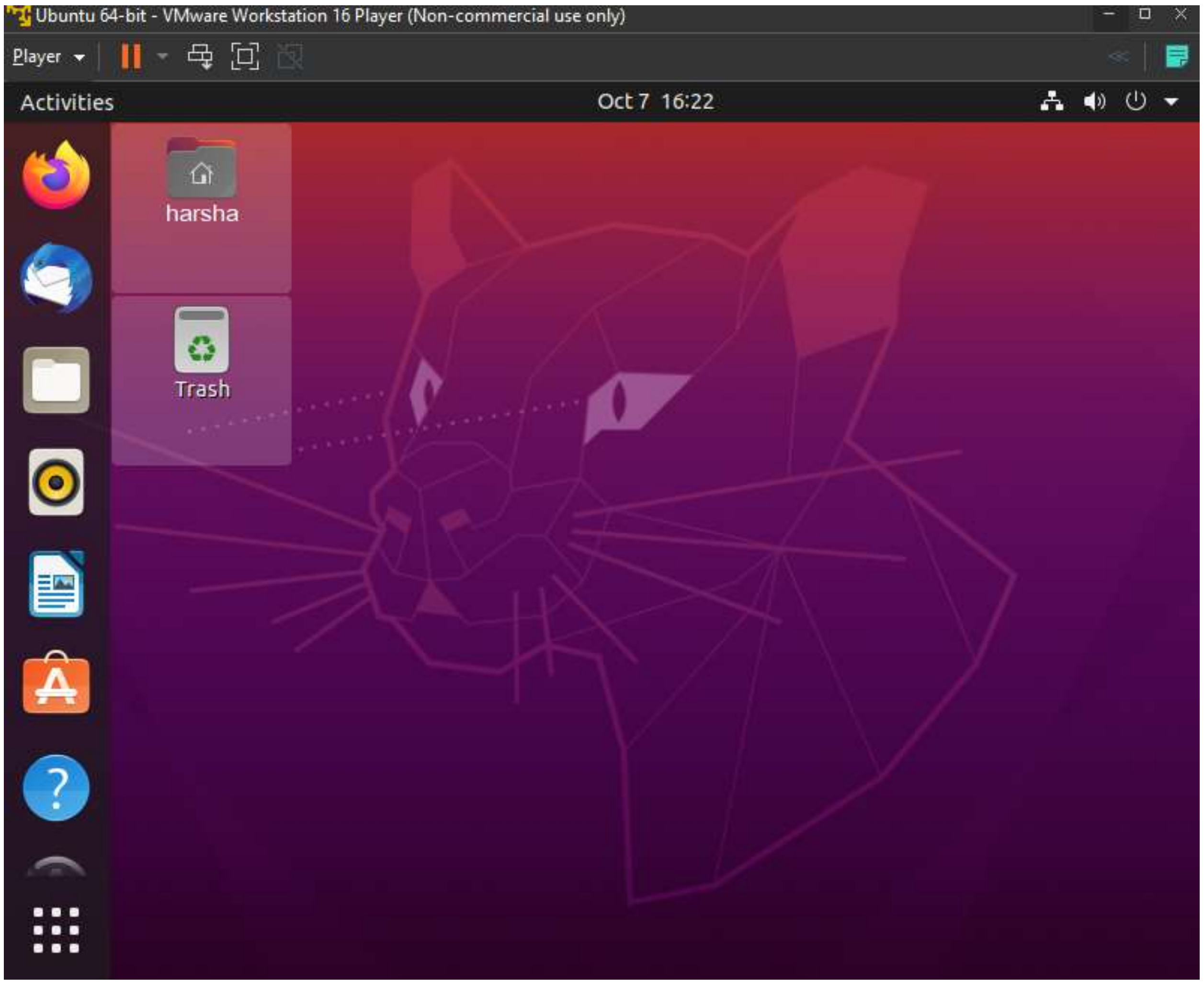
Description
Type here to enter a description of this virtual machine.

Virtual Machine Details

State:	Powered off
Configuration file:	C:\Users\win10\Documents\Virtual Machines\Ubuntu 64-bit\Ubuntu 64-bit.vmx
Hardware compatibility:	Workstation 16.x virtual machine
Primary IP address:	Network information is not available



```
6)
/dev/sda5: Clearing orphaned inode 794729 (uid=0, gid=0, mode=0100644, size=1055
28)
/dev/sda5: Clearing orphaned inode 794613 (uid=0, gid=0, mode=0100644, size=1369
352)
/dev/sda5: Clearing orphaned inode 794188 (uid=0, gid=0, mode=0100644, size=1801
6)
/dev/sda5: Clearing orphaned inode 793973 (uid=0, gid=0, mode=0100755, size=2029
224)
/dev/sda5: Clearing orphaned inode 793758 (uid=0, gid=0, mode=0100755, size=1914
72)
/dev/sda5: Clearing orphaned inode 794258 (uid=0, gid=0, mode=0100644, size=1049
84)
/dev/sda5: Clearing orphaned inode 794347 (uid=0, gid=0, mode=0100644, size=2658
56)
/dev/sda5: Clearing orphaned inode 795841 (uid=0, gid=0, mode=0100644, size=1952
928)
/dev/sda5: Clearing orphaned inode 659495 (uid=1000, gid=1000, mode=0100664, siz
e=32768)
/dev/sda5: Clearing orphaned inode 659494 (uid=1000, gid=1000, mode=0100600, siz
e=64)
/dev/sda5: Clearing orphaned inode 659755 (uid=1000, gid=1000, mode=0100664, siz
e=2712)
/dev/sda5: clean, 281153/1277952 files, 1836518/5111040 blocks
```


Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Player
Activities
Oct 7 16:22
harsha
Trash

Results:

The VMware workstation 16 Pro has been successfully installed and Ubuntu VM has been successfully created.

Ex: 4 Programming with VMs

Date : 09.09.2020

Objective:

To understand the installation procedure of GCC and to program with VMs.

Requirement:

- ❖ VMware workstation 16 Pro,
- ❖ Ubuntu OS.

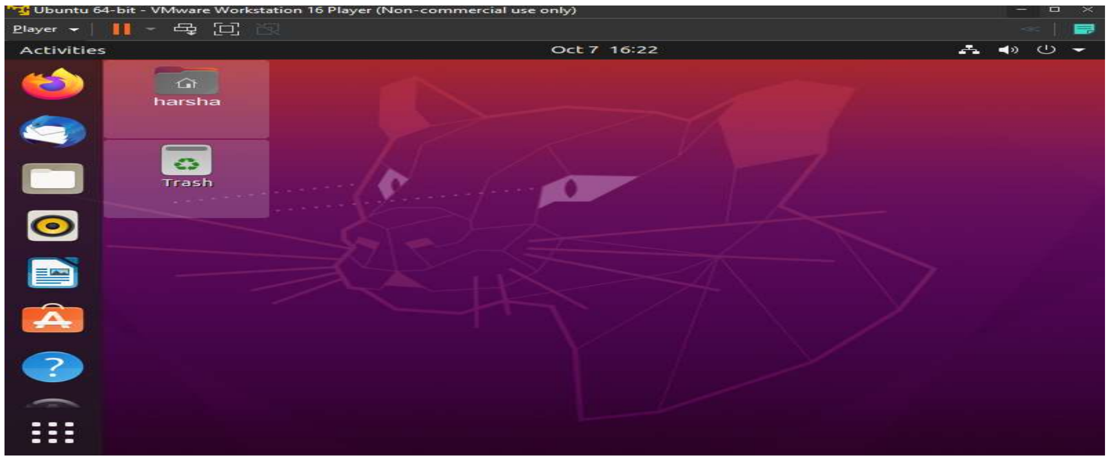
Theory:

Virtual Machine:

With the introduction of new technologies and newer research models, a lot number of hardware and software products are being launched. Many of the software are platform-dependent hence it is sometimes difficult to debug or check them because of the limited hardware resources.A VM (virtual machine) is an emulation of a computer system, where these machines use computer architectures to provide the functionality of a physical computer. The physical device on which virtual machines work is known as Host, whereas the virtual machines are known as Guest. A single host can have multiple numbers of guests.

Procedure and implementation:

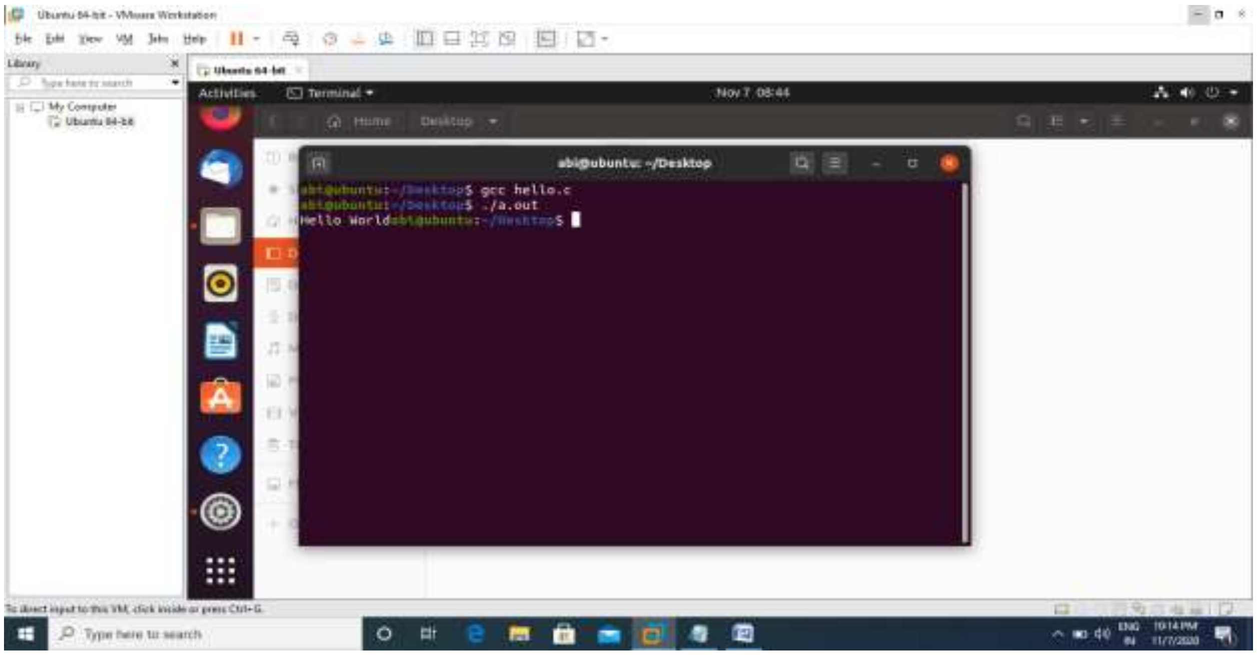
- ❖ Open Ubuntu VM in VMware workstation Pro.


Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)
Player
Activities
Oct 7 16:22
harsha
Trash

❖ Install gcc using the following command.

sudo apt install gcc

❖ Create hello.c file.
❖ Compile and Run the hello.c file.



```
abi@ubuntu:~/Desktop$ gcc hello.c
abi@ubuntu:~/Desktop$ ./a.out
Hello Worldabi@ubuntu:~/Desktop$
```

Results:

GCC has been successfully installed and the above program has been successfully executed in Ubuntu Virtual Machine in VMware workstation 16 Pro.

Ex: 5 Google App Engine installation

Date : 16.09.2020

Objective:

To understand the working of Google App Engine and to learn the installation procedure of Google App Engine.

Requirement:

- ❖ Google Chrome (or) related internet explorer
- ❖ JRE 11 (or) higher

Theory:

Architecture:

The Google App Engine (GAE) is Google`s answer to the ongoing trend of Cloud Computing offerings within the industry. In the traditional sense, GAE is a web application hosting service, allowing for development and deployment of web-based applications within a predefined runtime environment. Unlike other cloud-based hosting offerings such as Amazon Web Services that operate on an IaaS level, the GAE already provides an application infrastructure on the PaaS level. This means that the GAE abstracts from the underlying hardware and operating system layers by providing the hosted application with a set of application-oriented services. While this approach is very convenient for developers of such applications, the rationale behind the GAE is its focus on scalability and usage-based infrastructure as well as payment.

Costs:

Developing and deploying applications for the GAE is generally free of charge but restricted to a certain amount of traffic generated by the deployed application. Once this limit is reached within a certain time period, the application stops working. However, this limit can be waived when switching to a billable quota where the developer can enter a maximum budget that can be spent on an application per day. Depending on the traffic, once the free quota is reached the application will continue to work until the maximum budget for this day is reached. Table 1 summarizes some of the in our opinion most important quotas and corresponding amount per unit that is charged when free resources are depleted and additional, billable quota is desired.

Features:

With a Runtime Environment, the Data store and the App Engine services, the GAE can be divided into three parts. Runtime Environment The GAE runtime environment presents itself as the place where the actual application is executed. However, the application is only invoked once an HTTP request is processed to the GAE via a web browser or some other interface, meaning that the application is not constantly running if no invocation or processing has been done. In case of such an HTTP request, the request handler forwards the request and the GAE selects one out of many possible Google servers where the application is then instantly deployed and executed for a certain amount of time (8). The application may then do some computing and return the result back to the GAE request handler which forwards an HTTP response to the client. It is important to understand that the application runs completely embedded in this described sandbox environment but only as long as requests are still coming in or some processing is done within the application. The reason for this is simple: Applications should only run when they are actually computing, otherwise they would allocate precious computing power and memory without need. This paradigm shows already the GAE‘s potential in terms of scalability. Being able to run multiple instances of one application independently on different servers guarantees for a decent level of scalability. However, this highly flexible and stateless application execution paradigm has its limitations. Requests are processed no longer than 30 seconds after which the response has to be returned to the client and the application is removed from the runtime environment again (8). Obviously this method accepts that for deploying and starting an application each time a request is processed, an additional lead time is needed until the application is finally up and running. The GAE tries to encounter this problem by caching the application in the server memory as long as possible, optimizing for several subsequent requests to the same application. The type of runtime environment on the Google servers is dependent on the programming language used. For Java or other languages that have support for Java-based compilers (such as JRuby, Rhino and Groovy) a Java-based Java Virtual Machine (JVM) is provided. Also, GAE fully supports the Google Web Toolkit (GWT), a framework for rich web applications. For Python and related frameworks a Python-based environment is used.

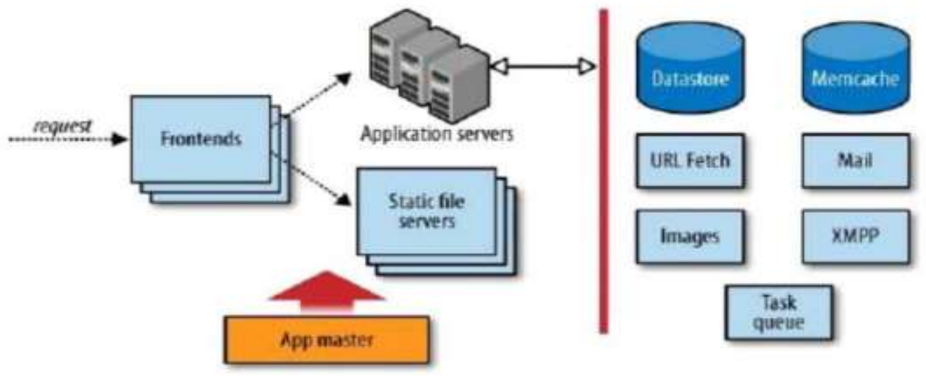
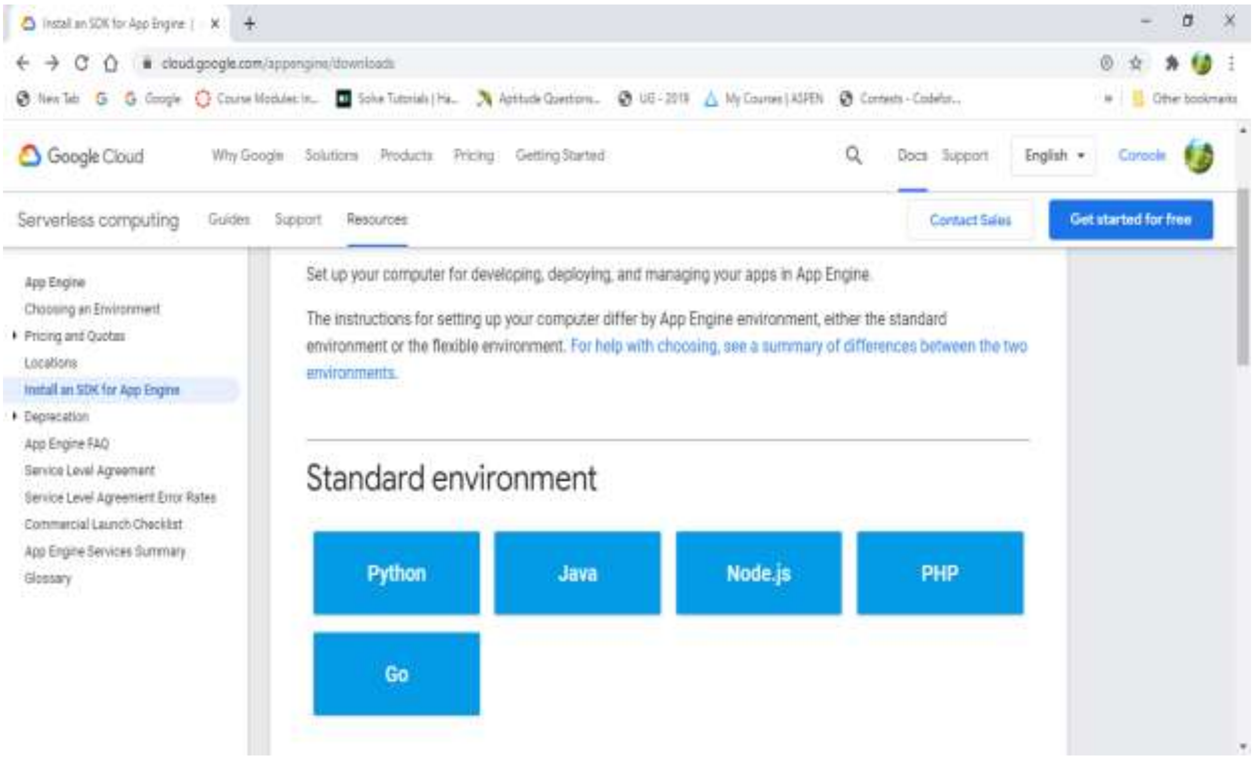


FIGURE 4: STRUCTURE OF GOOGLE APP ENGINE (13)

Procedure and implementation:

- ❖ Go to https://cloud.google.com/appengine/downloads and select java as standard environment.

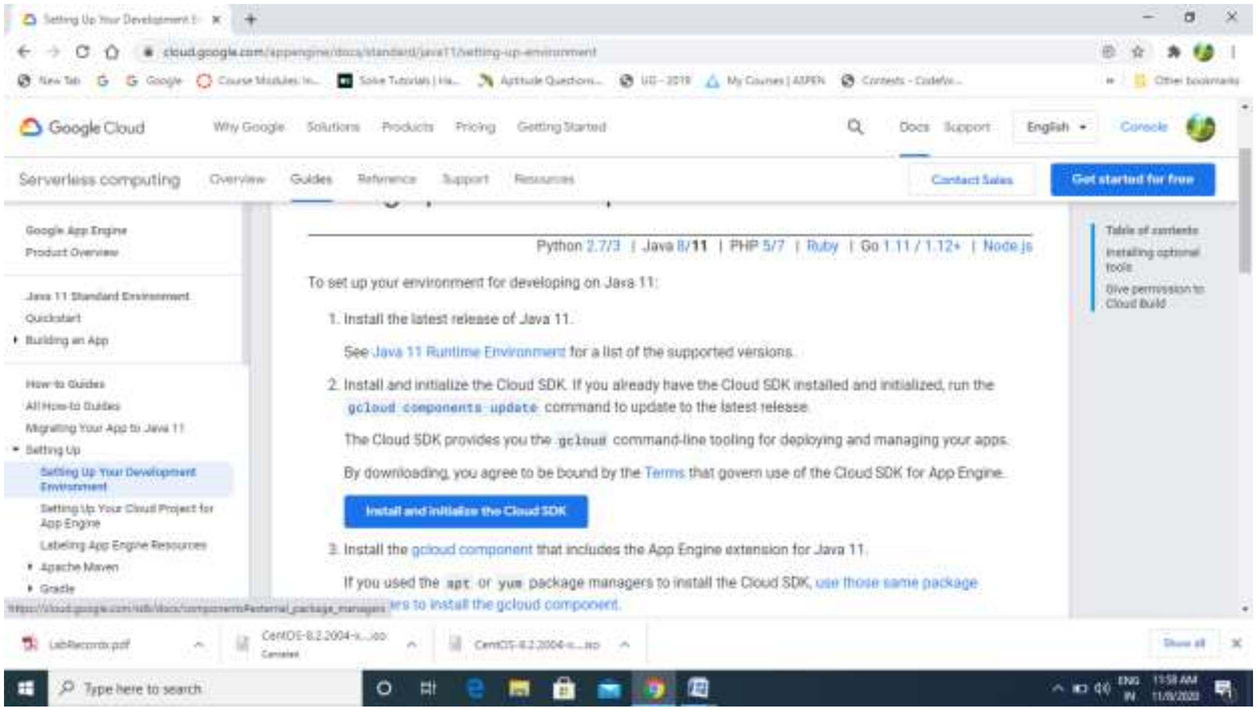


Set up your computer for developing, deploying, and managing your apps in App Engine.

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

Standard environment

❖ Install and Initialize the cloud SDK.



Python 2.7/3 | Java 8/11 | PHP 5/7 | Ruby | Go 1.11 / 1.12+ | Node.js

To set up your environment for developing on Java 11:

1. Install the latest release of Java 11.

   See Java 11 Runtime Environment for a list of the supported versions.

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

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

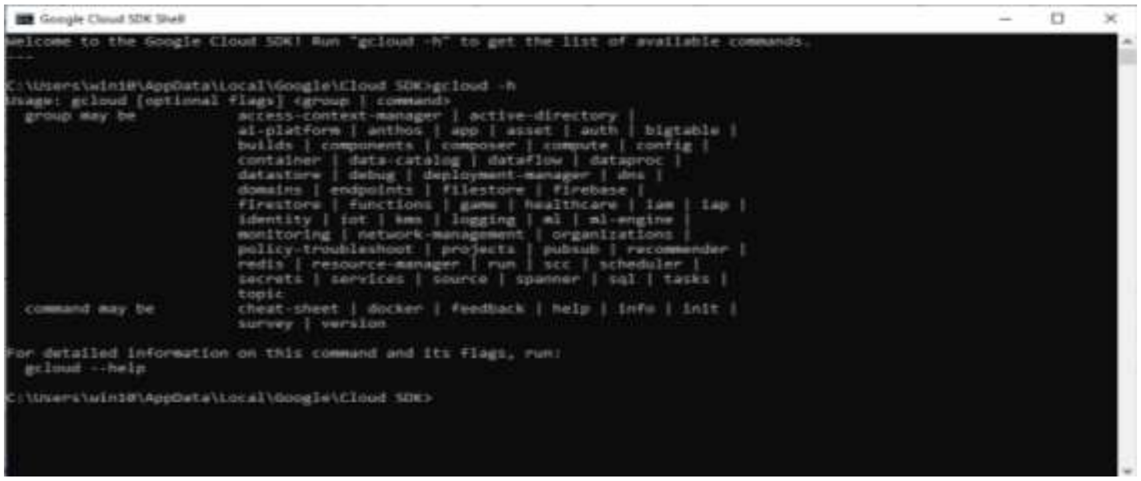
   By downloading, you agree to be bound by the Terms that govern use of the Cloud SDK for App Engine.

3. Install the gcloud component that includes the App Engine extension for Java 11.

   If you used the `apt` or `yum` package managers to install the Cloud SDK, use those same package managers to install the gcloud component.

❖ The google cloud SDK shell will be opened as soon as the installation is over.

❖ Use *gcloud –h* to view the list of commands.



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

C:\Users\win10\AppData\Local\Google\Cloud SDK>gcloud -h
Usage: gcloud [optional flags] <group | command>
  group may be           access-context-manager | active-directory |
                         ai-platform | anthos | app | asset | auth | bigtable |
                         builds | components | composer | compute | config |
                         container | data-catalog | dataflow | dataproc |
                         datastore | debug | deployment-manager | dns |
                         domains | endpoints | filestore | firebase |
                         firestore | functions | game | healthcare | iam | iap |
                         identity | iot | kms | logging | ml | ml-engine |
                         monitoring | network-management | organizations |
                         policy-troubleshoot | projects | pubsub | recommender |
                         redis | resource-manager | run | scc | scheduler |
                         secrets | services | source | spanner | sql | tasks |
                         topic
  command may be         cheat-sheet | docker | feedback | help | info | init |
                         survey | version

For detailed information on this command and its flags, run:
  gcloud --help

C:\Users\win10\AppData\Local\Google\Cloud SDK>
```

Results:

The Google App Engine has been successfully installed.

Ex: 6 **Build and Deploy Applications in GAE**

Date : 23.09.2020

Objective:

To understand and work with building and deployment of Applications in Google App Engine.

Requirement:

Google App Engine

Theory:

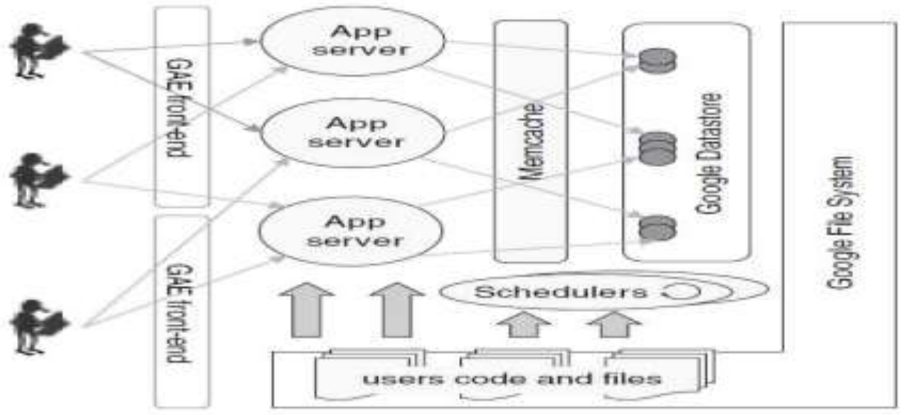
APPLICATION DEVELOPMENT USING GOOGLE APP ENGINE:

General Idea In order to evaluate the flexibility and scalability of the GAE we tried to come up with an application that relies heavily on scalability, i.e. collects large amounts of data from external sources. That way we hoped to be able to test both persistency and the gathering of data from external sources at large scale. Therefore our idea has been to develop an application that connects people`s delicious bookmarks with their respective Facebook accounts. People using our application should be able to see what their Facebook friends' delicious bookmarks are, provided their Facebook friends have such a delicious account. This way a user can get a visualization of his friends' latest topics by looking at a generated tag cloud giving him a clue about the most common and shared interests.

PLATFORM AS A SERVICE:

GOOGLE APP ENGINE:

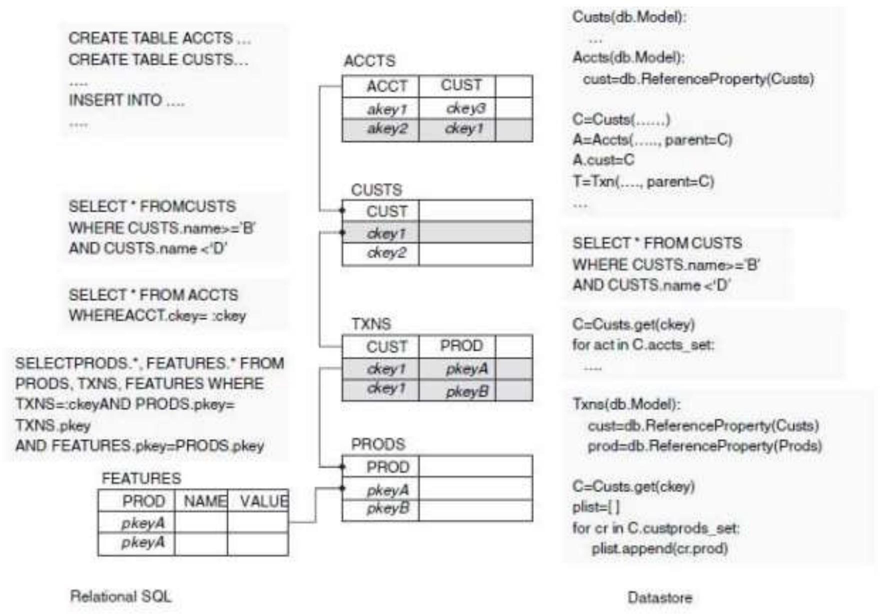
The Google cloud, called Google App Engine, is a _platform as a service' (PaaS) offering. In contrast with the Amazon infrastructure as a service cloud, where users explicitly provision virtual machines and control them fully, including installing, compiling and running software on them, a PaaS offering hides the actual execution environment from users. Instead, a software platform is provided along with an SDK, using which users develop applications and deploy them on the cloud. The PaaS platform is responsible for executing the applications, including servicing external service requests, as well as running scheduled jobs included in the application. By making the actual execution servers transparent to the user, a PaaS platform is able to share application servers across users who need lower capacities, as well as automatically scale resources allocated to applications that experience heavy loads. Below Figure depicts a user view of Google App Engine. Users upload code, in either Java or Python, along with related files, which are stored on the Google File System, a very large scale fault tolerant and redundant storage system. It is important to note that an application is immediately available on the internet as soon as it is successfully uploaded (no virtual servers need to be explicitly provisioned as in IaaS).



Resource usage for an application is metered in terms of web requests served and CPUhours actually spent executing requests or batch jobs. Note that this is very different from the IaaS model: A PaaS application can be deployed and made globally available 24×7, but charged only when accessed (or if batch jobs run); in contrast, in an IaaS model merely making an application continuously available incurs the full cost of keeping at least some of the servers running all the time. Further, deploying applications in Google App Engine is free, within usage limits; thus applications can be developed and tried out free and begin to incur cost only when actually accessed by a sufficient volume of requests. The PaaS model enables Google to provide such a free service because applications do not run in dedicated virtual machines; a deployed application that is not accessed merely consumes storage for its code and data and expends no CPU cycles. GAE applications are served by a large number of web servers in Google‘s data centers that execute requests from end-users across the globe. The web servers load code from the GFS into memory and serve these requests. Each request to a particular application is served by any one of GAE‘s web servers; there is no guarantee that the same server will serve requests to any two requests, even from the same HTTP session. Applications can also specify some functions to be executed as batch jobs which are run by a scheduler.

Google Datastore:

Applications persist data in the Google Datastore, which is also (like Amazon SimpleDB) a nonrelational database. The Datastore allows applications to define structured types (called _kinds‘) and store their instances (called _entities‘) in a distributed manner on the GFS file system. While one can view Datastore _kinds‘ as table structures and entities as records, there are important differences between a relational model and the Datastore, some of which are also illustrated in Figure.



```
CREATE TABLE ACCTS ...
CREATE TABLE CUSTS...
....
INSERT INTO ....
....
```

```
SELECT * FROMCUSTS
WHERE CUSTS.name>='B'
AND CUSTS.name <'D'
```

```
SELECT * FROM ACCTS
WHEREACCT.ckey= :ckey
```

```
SELECTPRODS.*, FEATURES.* FROM
PRODS, TXNS, FEATURES WHERE
TXNS=:ckeyAND PRODS.pkey=
TXNS.pkey
AND FEATURES.pkey=PRODS.pkey
```

```
Custs(db.Model):
  ...
Accts(db.Model):
  cust=db.ReferenceProperty(Custs)

C=Custs(......)
A=Accts(....., parent=C)
A.cust=C
T=Txn(...., parent=C)
...
```

```
SELECT * FROM CUSTS
WHERE CUSTS.name>='B'
AND CUSTS.name <'D'
```

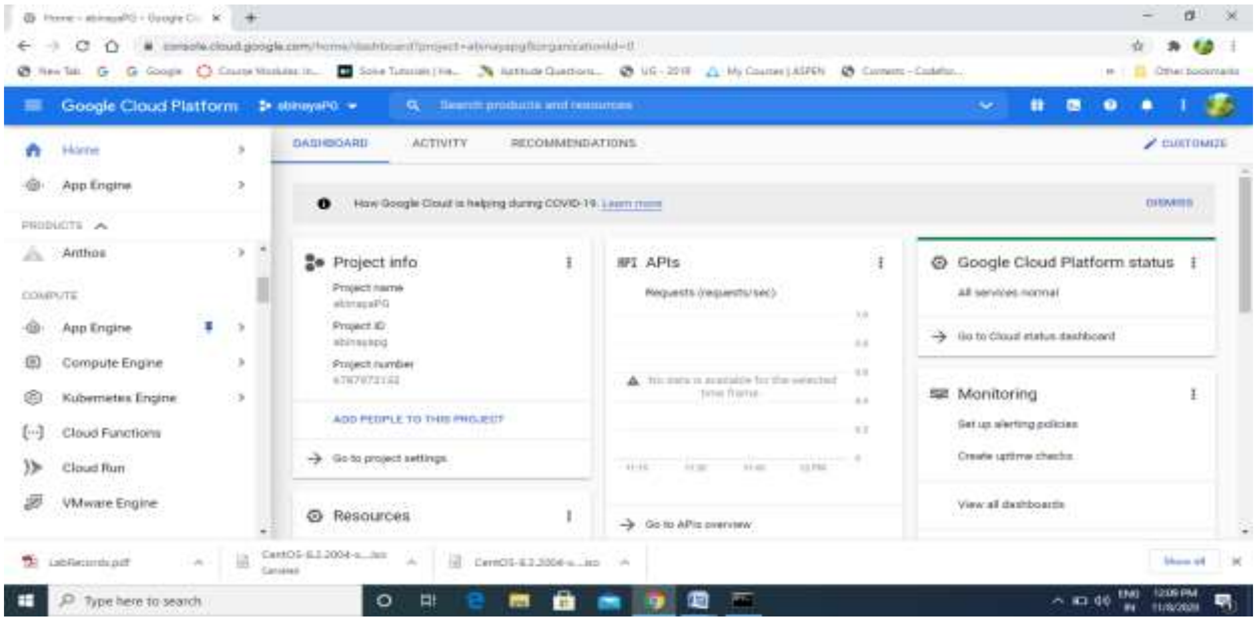
```
C=Custs.get(ckey)
for act in C.accts_set:
  ....
```

```
Txns(db.Model):
   cust=db.ReferenceProperty(Custs)
   prod=db.ReferenceProperty(Prods)

C=Custs.get(ckey)
plist=[ ]
for cr in C.custprods_set:
    plist.append(cr.prod)
```

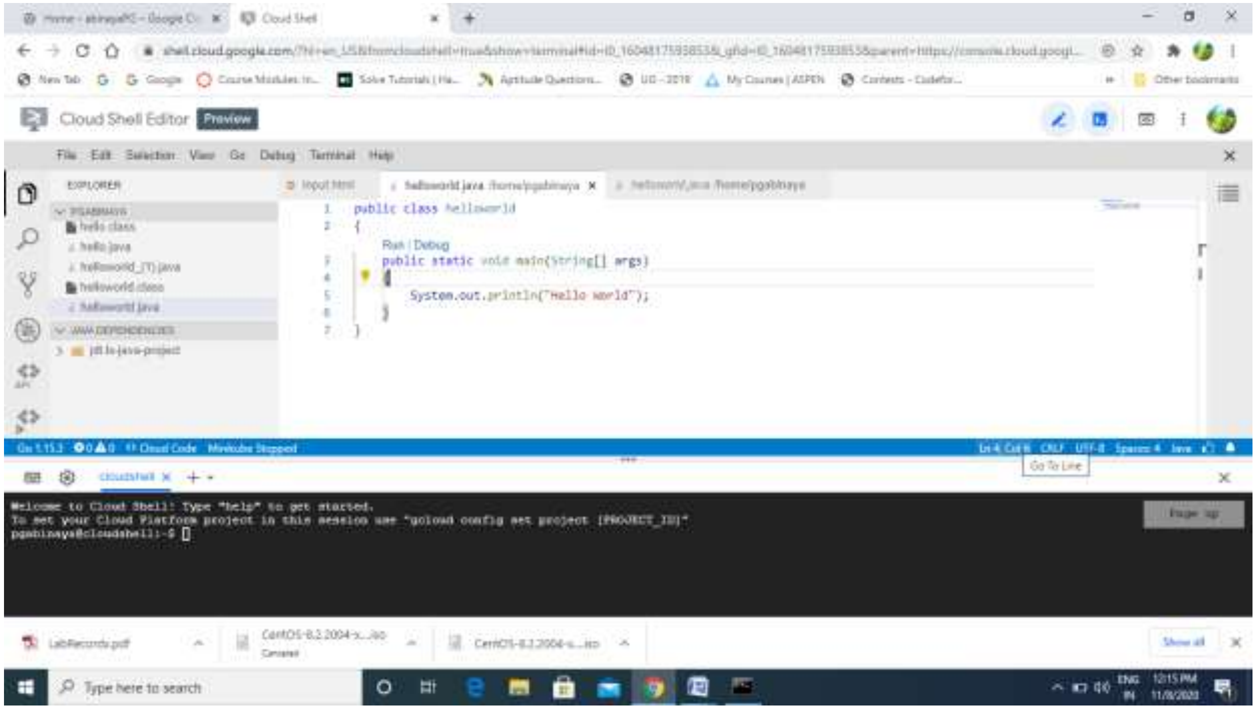
Procedure and implementation:

❖ Go to https://cloud.google.com/appengine/docs/standard/java11/setting-up-environment and go to console.

Harshavarthini K
312217205035



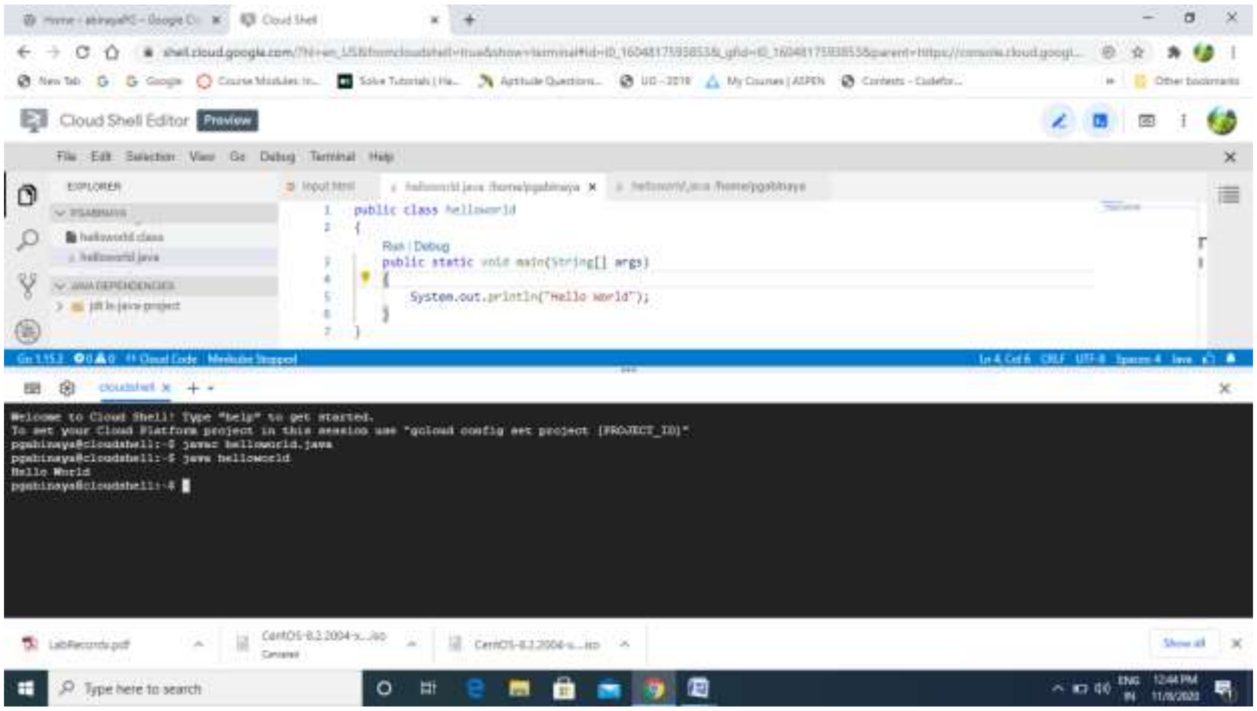
❖ Activate cloud shell and type the application in the editor.



```
public class helloworld
{
    Run | Debug
    public static void main(String[] args)
    {
        System.out.println("Hello world");
    }
}
```

```
Welcome to Cloud Shell! Type "help" to get started.
To set your Cloud Platform project in this session use "gcloud config set project [PROJECT_ID]"
pgabinaya@cloudshell:~$ 
```

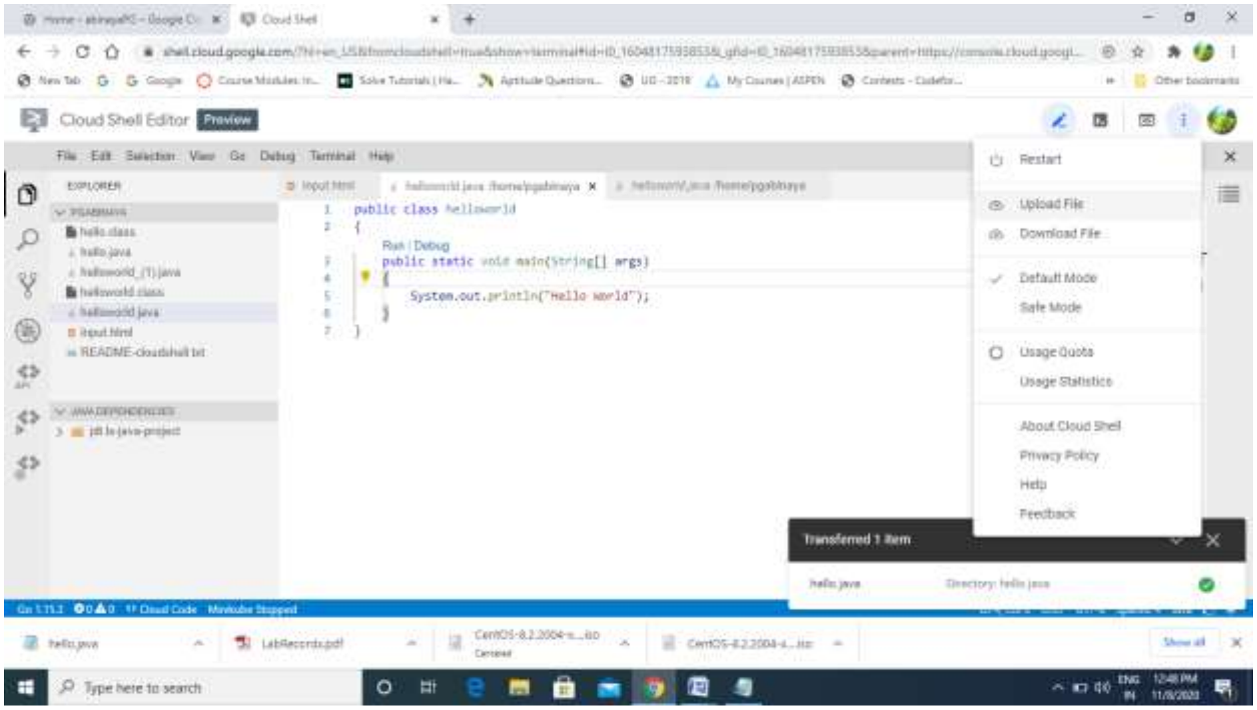
❖ Run/Deploy the application.



```
public class helloworld
{
    public static void main(String[] args)
    {
        System.out.println("Hello World");
    }
}
```

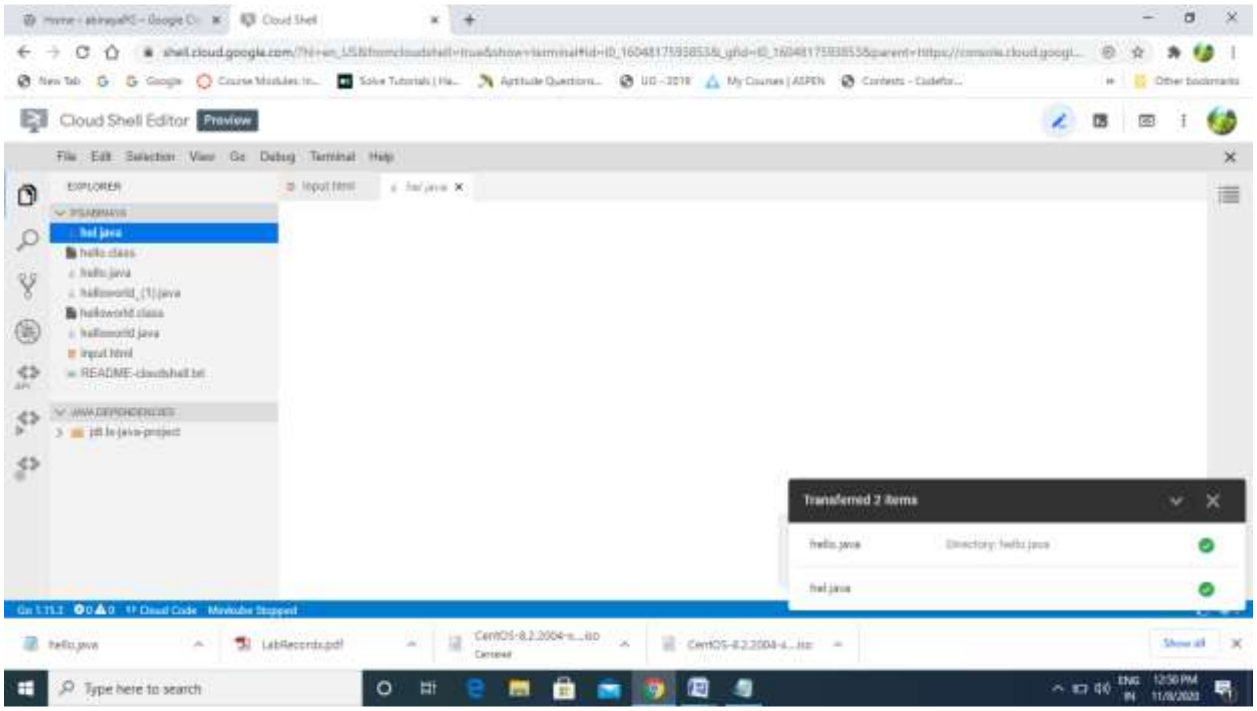
```
Welcome to Cloud Shell! Type "help" to get started.
To set your Cloud Platform project in this session use "gcloud config set project [PROJECT_ID]"
pgabinaya@cloudshell:~$ javac helloworld.java
pgabinaya@cloudshell:~$ java helloworld
Hello World
pgabinaya@cloudshell:~$
```

❖ Upload any file from Desktop to GAE.

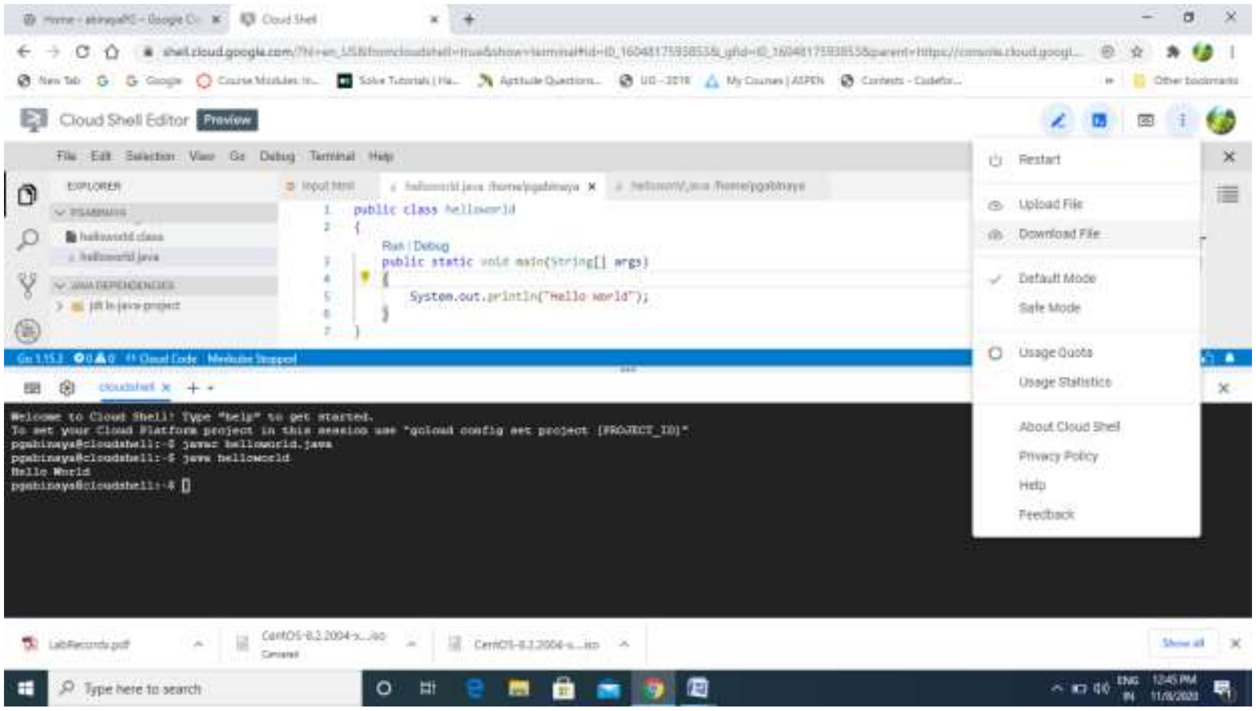


```
public class helloworld
{
    public static void main(String[] args)
    {
        System.out.println("Hello World");
    }
}
```

Harshavarthini K
312217205035

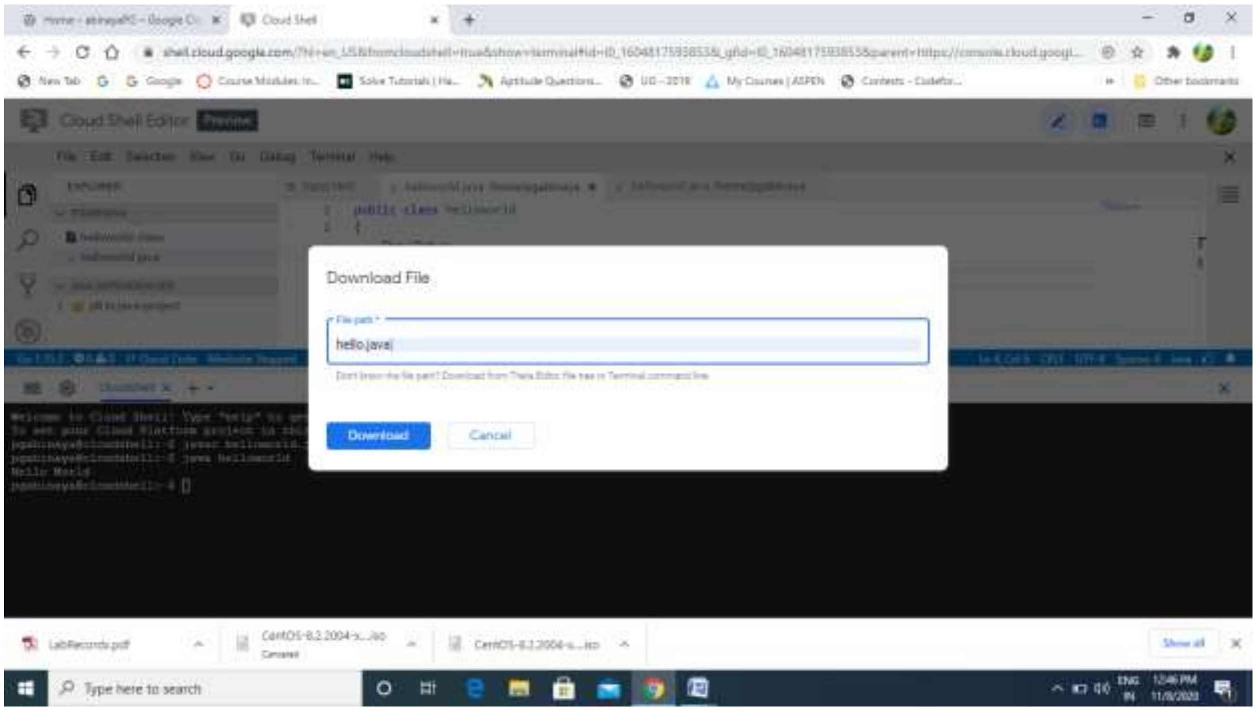


❖ Download the file from GAE to Desktop.

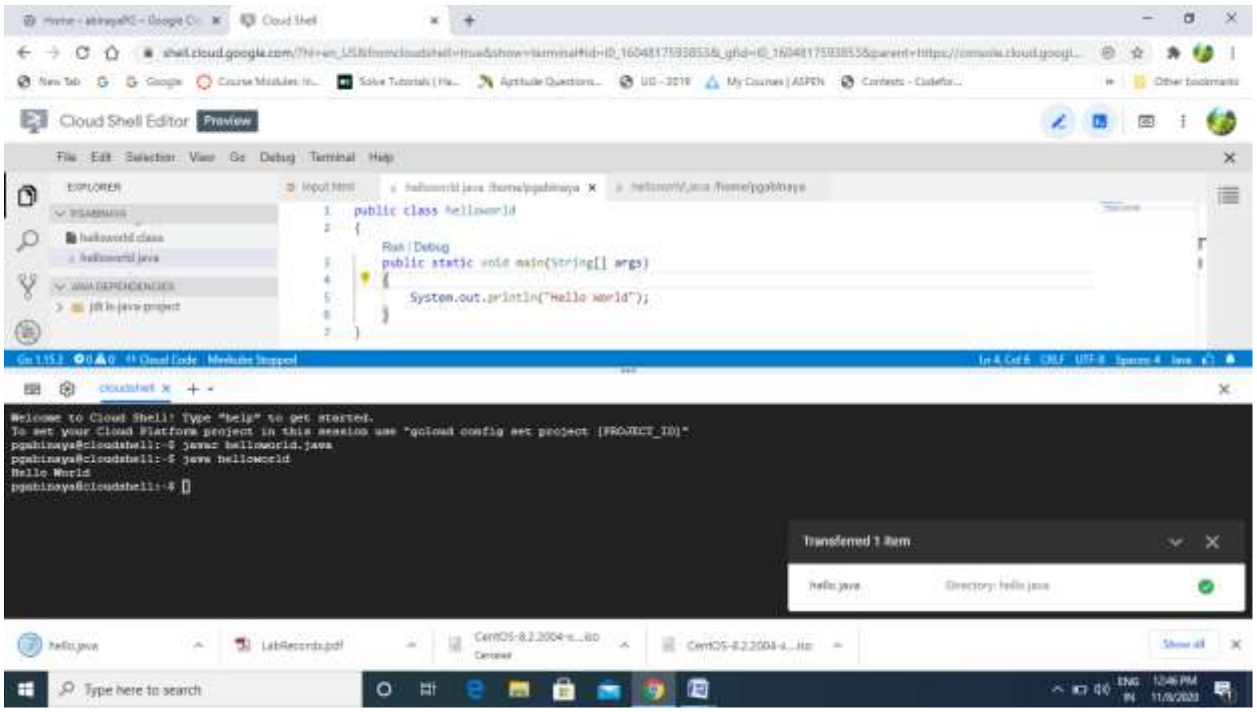


```
public class helloworld
{
    public static void main(String[] args)
    {
        System.out.println("Hello World");
    }
}
```

```
Welcome to Cloud Shell! Type "help" to get started.
To set your Cloud Platform project in this session use "gcloud config set project [PROJECT_ID]"
pgabinaya@cloudshell:~$ javac helloworld.java
pgabinaya@cloudshell:~$ java helloworld
Hello World
pgabinaya@cloudshell:~$ 
```



```
hello.java
```



```java
public class helloworld
{
    public static void main(String[] args)
    {
        System.out.println("Hello World");
    }
}
```

```
Welcome to Cloud Shell! Type "help" to get started.
To set your Cloud Platform project in this session use "gcloud config set project [PROJECT_ID]"
pgabinaya@cloudshell:~$ javac helloworld.java
pgabinaya@cloudshell:~$ java helloworld
Hello World
pgabinaya@cloudshell:~$
```

Results:

The above application has been successfully build and deployed in Google App Engine.

Ex: 7 Build and run the scheduling in CloudSim

Date : 30.09.2020

Objective:

To understand the working of cloudsim and to implement scheduling algorithm using cloudsim in eclipse.

Requirement:

- ❖ Eclipse (latest version)
- ❖ Cloudsim folder
- ❖ Common math folder

Theory:

CloudSim is a simulation toolkit that supports the modeling and simulation of the core functionality of cloud, like job/task queue, processing of events, creation of cloud entities(datacenter, datacenter brokers, etc), communication between different entities, implementation of broker policies, etc.

This toolkit allows to:

- ❖ Test application services in a repeatable and controllable environment.
- ❖ Tune the system bottlenecks before deploying apps in an actual cloud.
- ❖ Experiment with different workload mix and resource performance scenarios on simulated infrastructure for developing and testing adaptive application provisioning techniques

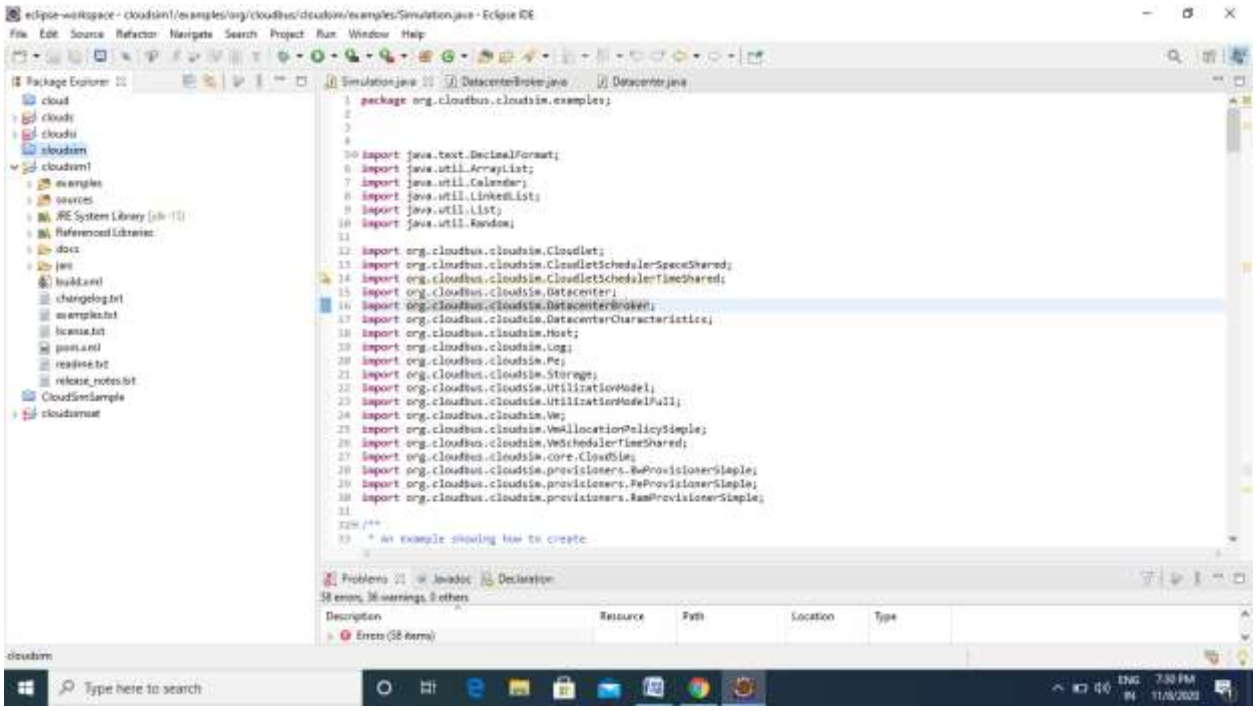
Core features :

- ❖ The Support of modeling and simulation of large scale computing environment as federated cloud data centers, virtualized server hosts, with customizable policies for provisioning host resources to virtual machines and energy-aware computational resources
- ❖ It is a self-contained platform for modeling cloud’s service brokers, provisioning, and allocation policies.
- ❖ It supports the simulation of network connections among simulated system elements.

- ❖ Support for simulation of federated cloud environment, that inter-networks resources from both private and public domains.
- ❖ Availability of a virtualization engine that aids in the creation and management of multiple independent and co-hosted virtual services on a data center node.
- ❖ Flexibility to switch between space shared and time shared allocation of processing cores to virtualized services.

Procedure and implementation:

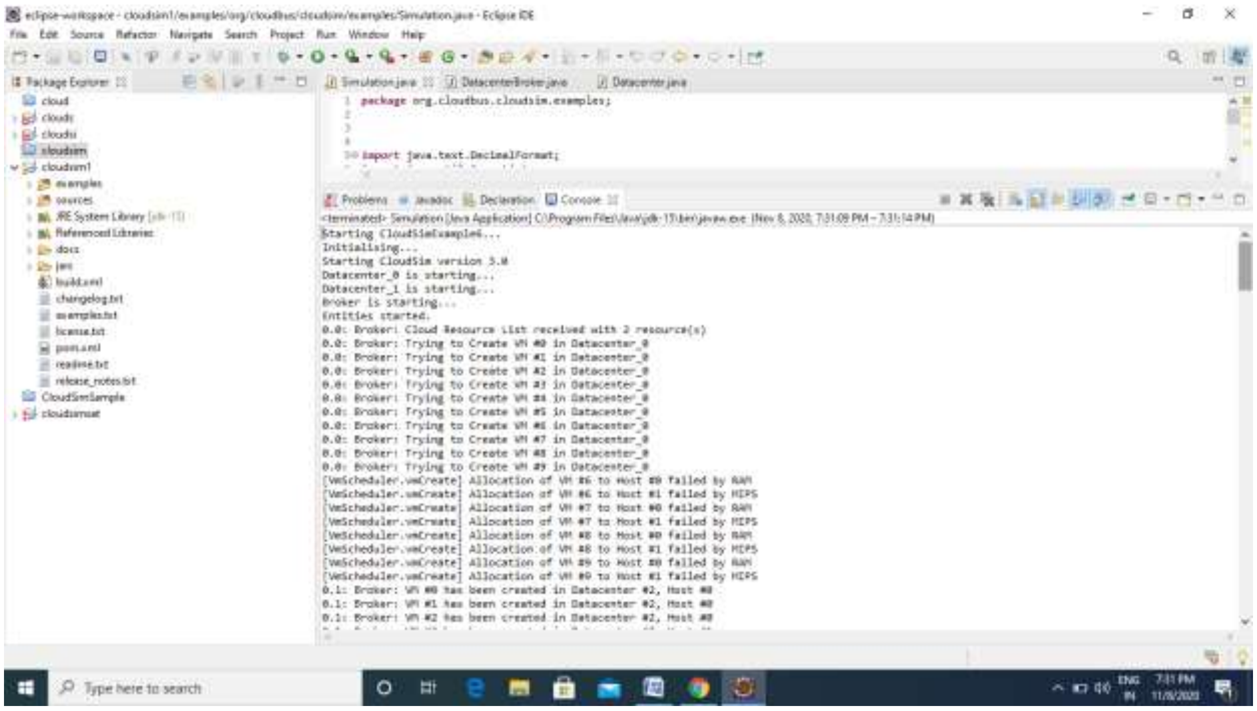
- ❖ Create a folder and paste the cloudsim and common math folder in that.
- ❖ Open eclipse.
- ❖ File->New->java project.
- ❖ Browse cloudsim folder and click finish
- ❖ Create Simulation.java and update DataCenterBroker.java



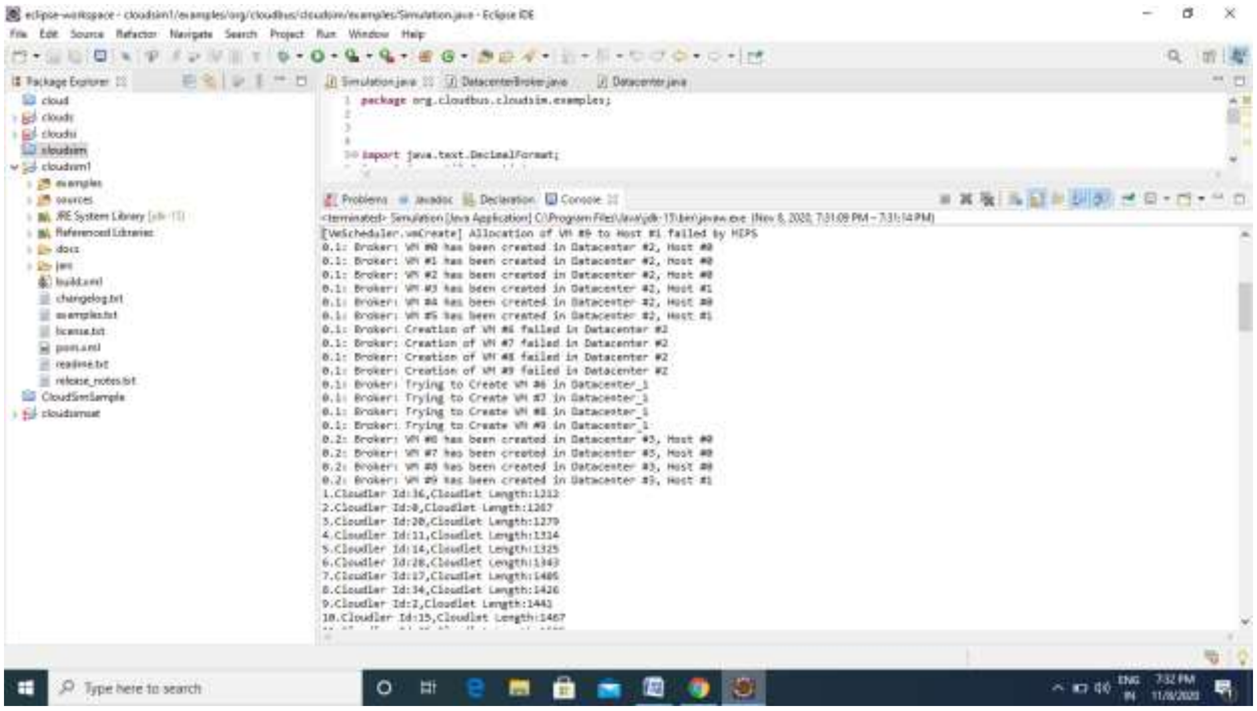
```
package org.cloudbus.cloudsim.examples;



import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.LinkedList;
import java.util.List;
import java.util.Random;

import org.cloudbus.cloudsim.Cloudlet;
import org.cloudbus.cloudsim.CloudletSchedulerSpaceShared;
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;

/**
 * An example showing how to create
```

```
58 errors, 36 warnings, 0 others
Errors (58 items)
```

- ❖ Run the program

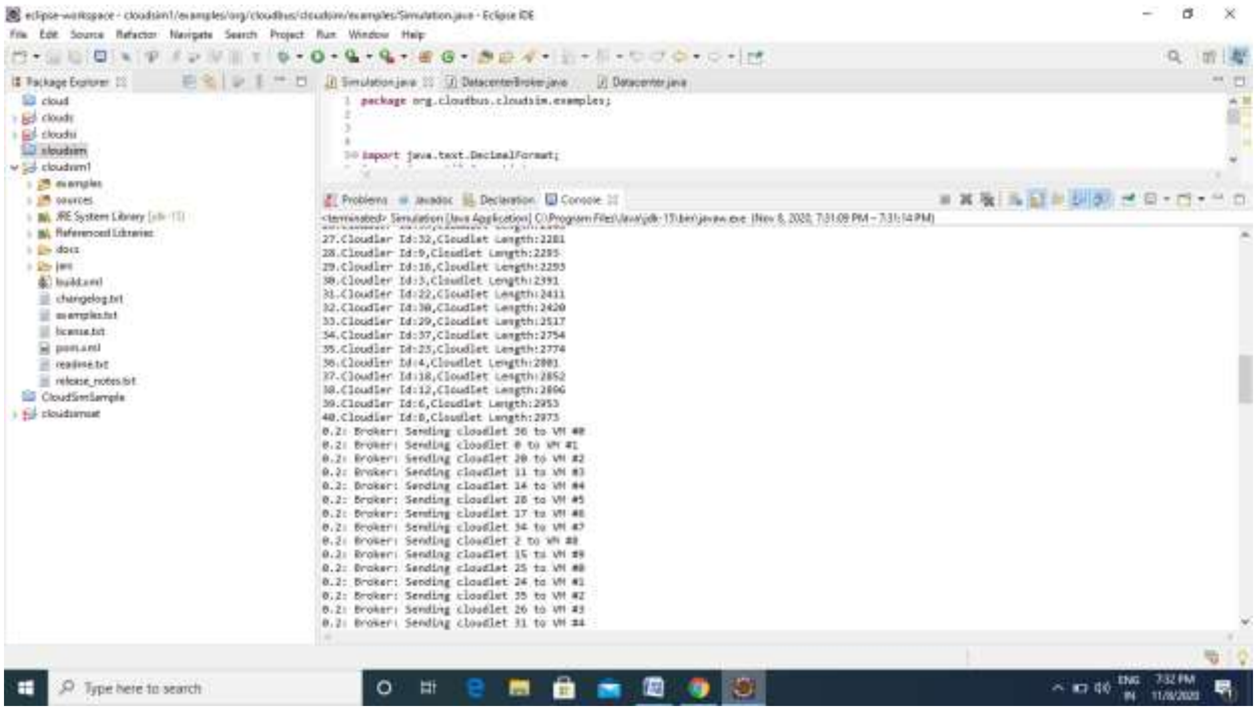


```
<terminated> Simulation [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (Nov 8, 2020, 7:31:09 PM – 7:31:14 PM)
Starting CloudSimExample6...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.0: Broker: Trying to Create VM #3 in Datacenter_0
0.0: Broker: Trying to Create VM #4 in Datacenter_0
0.0: Broker: Trying to Create VM #5 in Datacenter_0
0.0: Broker: Trying to Create VM #6 in Datacenter_0
0.0: Broker: Trying to Create VM #7 in Datacenter_0
0.0: Broker: Trying to Create VM #8 in Datacenter_0
0.0: Broker: Trying to Create VM #9 in Datacenter_0
[VmScheduler.vmCreate] Allocation of VM #6 to Host #0 failed by RAM
[VmScheduler.vmCreate] Allocation of VM #6 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #7 to Host #0 failed by RAM
[VmScheduler.vmCreate] Allocation of VM #7 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #8 to Host #0 failed by RAM
[VmScheduler.vmCreate] Allocation of VM #8 to Host #1 failed by MIPS
[VmScheduler.vmCreate] Allocation of VM #9 to Host #0 failed by RAM
[VmScheduler.vmCreate] Allocation of VM #9 to Host #1 failed by MIPS
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #2, Host #0
0.1: Broker: VM #2 has been created in Datacenter #2, Host #0
```



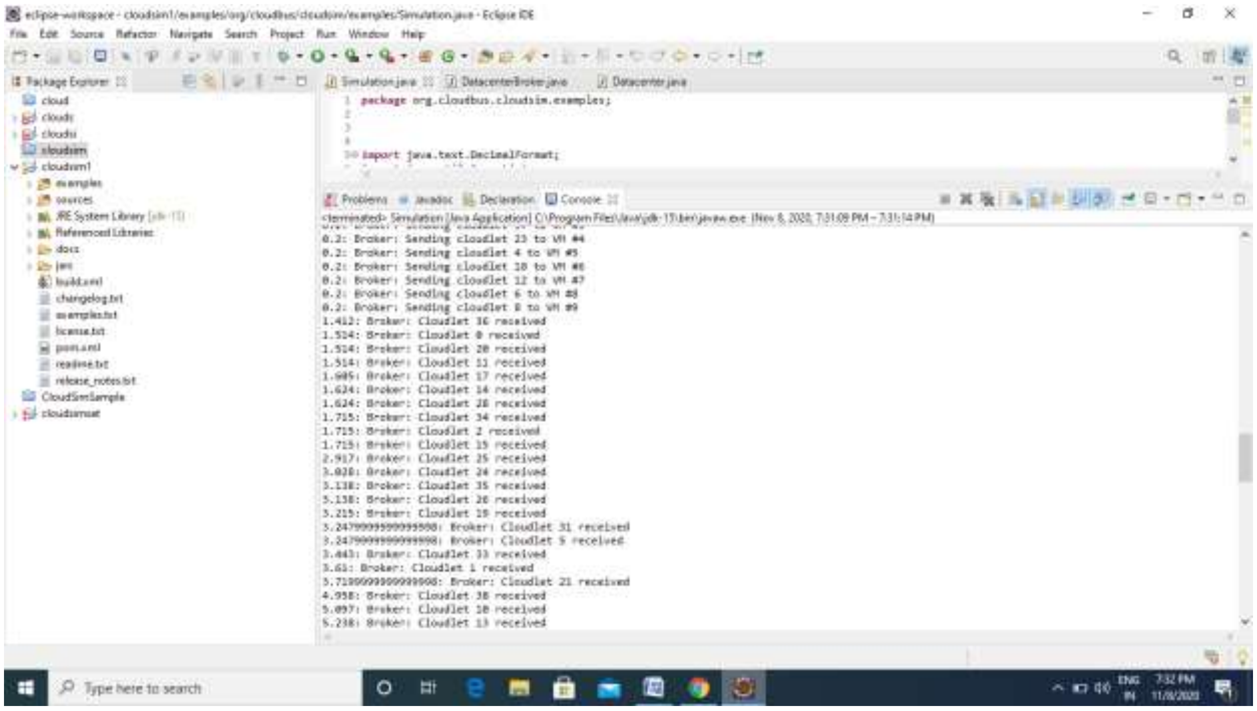
```
<terminated> Simulation [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (Nov 8, 2020, 7:31:09 PM – 7:31:14 PM)
[VmScheduler.vmCreate] Allocation of VM #9 to Host #1 failed by MIPS
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #2, Host #0
0.1: Broker: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker: VM #3 has been created in Datacenter #2, Host #1
0.1: Broker: VM #4 has been created in Datacenter #2, Host #0
0.1: Broker: VM #5 has been created in Datacenter #2, Host #1
0.1: Broker: Creation of VM #6 failed in Datacenter #2
0.1: Broker: Creation of VM #7 failed in Datacenter #2
0.1: Broker: Creation of VM #8 failed in Datacenter #2
0.1: Broker: Creation of VM #9 failed in Datacenter #2
0.1: Broker: Trying to Create VM #6 in Datacenter_1
0.1: Broker: Trying to Create VM #7 in Datacenter_1
0.1: Broker: Trying to Create VM #8 in Datacenter_1
0.1: Broker: Trying to Create VM #9 in Datacenter_1
0.2: Broker: VM #6 has been created in Datacenter #3, Host #0
0.2: Broker: VM #7 has been created in Datacenter #3, Host #0
0.2: Broker: VM #8 has been created in Datacenter #3, Host #0
0.2: Broker: VM #9 has been created in Datacenter #3, Host #1
1.Cloudler Id:36,Cloudlet Length:1212
2.Cloudler Id:0,Cloudlet Length:1267
3.Cloudler Id:20,Cloudlet Length:1279
4.Cloudler Id:11,Cloudlet Length:1314
5.Cloudler Id:14,Cloudlet Length:1325
6.Cloudler Id:28,Cloudlet Length:1343
7.Cloudler Id:17,Cloudlet Length:1405
8.Cloudler Id:34,Cloudlet Length:1426
9.Cloudler Id:2,Cloudlet Length:1441
10.Cloudler Id:15,Cloudlet Length:1467
```

Harshavarthini K
312217205035



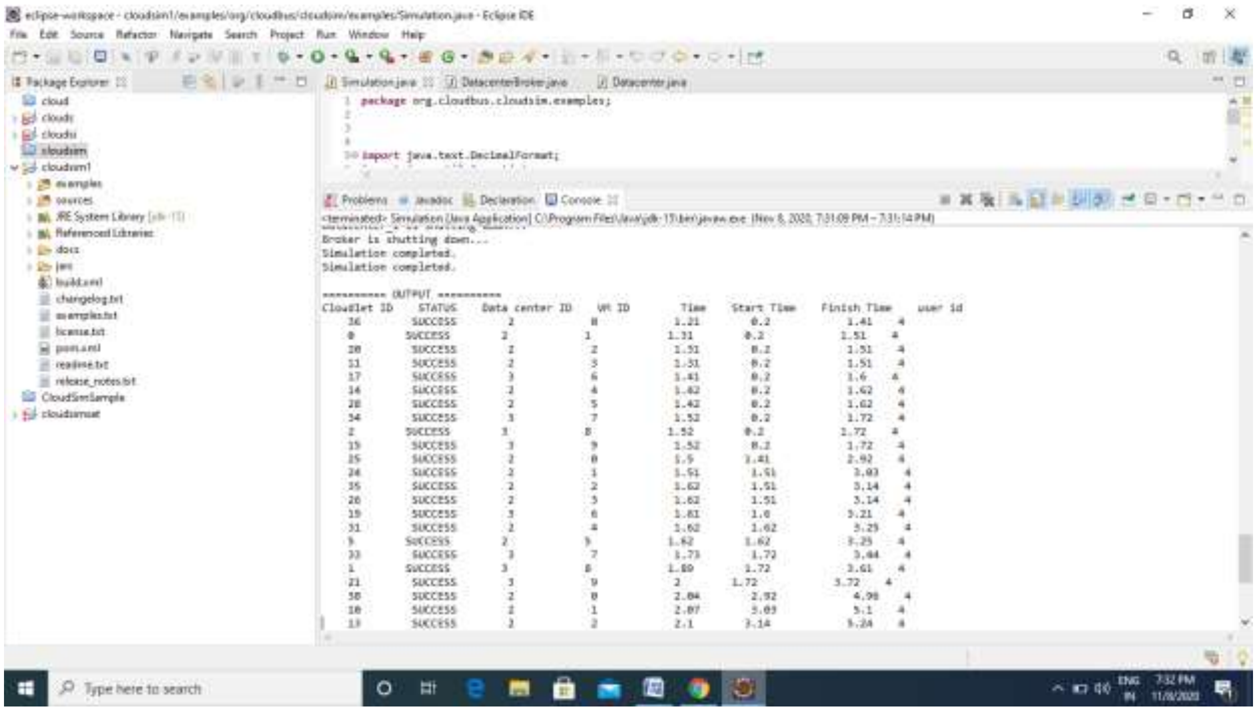
```
package org.cloudbus.cloudsim.examples;



import java.text.DecimalFormat;
```

```
<terminated> Simulation [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (Nov 8, 2020, 7:31:09 PM – 7:31:14 PM)
27.Cloudler Id:32,Cloudlet Length:2281
28.Cloudler Id:9,Cloudlet Length:2285
29.Cloudler Id:16,Cloudlet Length:2293
30.Cloudler Id:3,Cloudlet Length:2391
31.Cloudler Id:22,Cloudlet Length:2411
32.Cloudler Id:30,Cloudlet Length:2420
33.Cloudler Id:29,Cloudlet Length:2517
34.Cloudler Id:37,Cloudlet Length:2754
35.Cloudler Id:23,Cloudlet Length:2774
36.Cloudler Id:4,Cloudlet Length:2801
37.Cloudler Id:18,Cloudlet Length:2852
38.Cloudler Id:12,Cloudlet Length:2896
39.Cloudler Id:6,Cloudlet Length:2953
40.Cloudler Id:8,Cloudlet Length:2973
0.2: Broker: Sending cloudlet 36 to VM #0
0.2: Broker: Sending cloudlet 0 to VM #1
0.2: Broker: Sending cloudlet 20 to VM #2
0.2: Broker: Sending cloudlet 11 to VM #3
0.2: Broker: Sending cloudlet 14 to VM #4
0.2: Broker: Sending cloudlet 28 to VM #5
0.2: Broker: Sending cloudlet 17 to VM #6
0.2: Broker: Sending cloudlet 34 to VM #7
0.2: Broker: Sending cloudlet 2 to VM #8
0.2: Broker: Sending cloudlet 15 to VM #9
0.2: Broker: Sending cloudlet 25 to VM #0
0.2: Broker: Sending cloudlet 24 to VM #1
0.2: Broker: Sending cloudlet 35 to VM #2
0.2: Broker: Sending cloudlet 26 to VM #3
0.2: Broker: Sending cloudlet 31 to VM #4
```



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



import java.text.DecimalFormat;
```

```
<terminated> Simulation [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (Nov 8, 2020, 7:31:09 PM – 7:31:14 PM)
0.2: Broker: Sending cloudlet 23 to VM #4
0.2: Broker: Sending cloudlet 4 to VM #5
0.2: Broker: Sending cloudlet 18 to VM #6
0.2: Broker: Sending cloudlet 12 to VM #7
0.2: Broker: Sending cloudlet 6 to VM #8
0.2: Broker: Sending cloudlet 8 to VM #9
1.412: Broker: Cloudlet 36 received
1.514: Broker: Cloudlet 0 received
1.514: Broker: Cloudlet 20 received
1.514: Broker: Cloudlet 11 received
1.605: Broker: Cloudlet 17 received
1.624: Broker: Cloudlet 14 received
1.624: Broker: Cloudlet 28 received
1.715: Broker: Cloudlet 34 received
1.715: Broker: Cloudlet 2 received
1.715: Broker: Cloudlet 15 received
2.917: Broker: Cloudlet 25 received
3.028: Broker: Cloudlet 24 received
3.138: Broker: Cloudlet 35 received
3.138: Broker: Cloudlet 26 received
3.215: Broker: Cloudlet 19 received
3.2479999999999998: Broker: Cloudlet 31 received
3.2479999999999998: Broker: Cloudlet 5 received
3.443: Broker: Cloudlet 33 received
3.61: Broker: Cloudlet 1 received
3.7199999999999998: Broker: Cloudlet 21 received
4.958: Broker: Cloudlet 38 received
5.097: Broker: Cloudlet 18 received
5.238: Broker: Cloudlet 13 received
```



```
<terminated> Simulation [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (Nov 8, 2020, 7:31:09 PM – 7:31:14 PM)
Broker is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID        Time    Start Time    Finish Time    user id
    36          SUCCESS        2             0          1.21      0.2           1.41    4
    0          SUCCESS        2             1          1.31      0.2           1.51    4
    20          SUCCESS        2             2          1.31      0.2           1.51    4
    11          SUCCESS        2             3          1.31      0.2           1.51    4
    17          SUCCESS        3             6          1.41      0.2           1.6    4
    14          SUCCESS        2             4          1.42      0.2           1.62    4
    28          SUCCESS        2             5          1.42      0.2           1.62    4
    34          SUCCESS        3             7          1.52      0.2           1.72    4
    2          SUCCESS        3             8          1.52      0.2           1.72    4
    15          SUCCESS        3             9          1.52      0.2           1.72    4
    25          SUCCESS        2             0          1.5      1.41           2.92    4
    24          SUCCESS        2             1          1.51      1.51           3.03    4
    35          SUCCESS        2             2          1.62      1.51           3.14    4
    26          SUCCESS        2             3          1.62      1.51           3.14    4
    19          SUCCESS        3             6          1.61      1.6           3.21    4
    31          SUCCESS        2             4          1.62      1.62           3.25    4
    5          SUCCESS        2             5          1.62      1.62           3.25    4
    33          SUCCESS        3             7          1.73      1.72           3.44    4
    1          SUCCESS        3             8          1.89      1.72           3.61    4
    21          SUCCESS        3             9          2      1.72           3.72    4
    30          SUCCESS        2             0          2.04      2.92           4.96    4
    10          SUCCESS        2             1          2.07      3.03           5.1    4
    13          SUCCESS        2             2          2.1      3.14           5.24    4
```

Results:

Thus the scheduling algorithm using cloudsim has been successfully executed.

EX NO:8 Sharing files between VMs

DATE: 06.10.2020

Objective:

To share a file between two virtual machines using virtual box.

Requirement:

- ❖ Two Virtual Environments - VM1, VM2
- ❖ Open SSH Client.

Theory:

Transferring files between Ubuntu and Windows can be done in many ways, but in this tutorial, we are going to discuss the following popular methods:

1. Transfer files between Ubuntu and Windows via the SSH.
2. Transfer files between Ubuntu and Windows using Samba.
3. Transfer files between Ubuntu and Windows using Shared Network Folders.

Procedure and implementation:

I. Create two VMs and install Ubuntu 16.04 server for AMD64 architecture. Name them as VM1 & VM2.

Install open ssh-server while installing OS.

]$ sudo apt-get install open ssh-server]$ sudo apt-get install open ssh-client

II. Ping VM1 from VM2 and vice versa.
III.Generate ssh public and private key pairs in VM1 & VM2 using the command]$

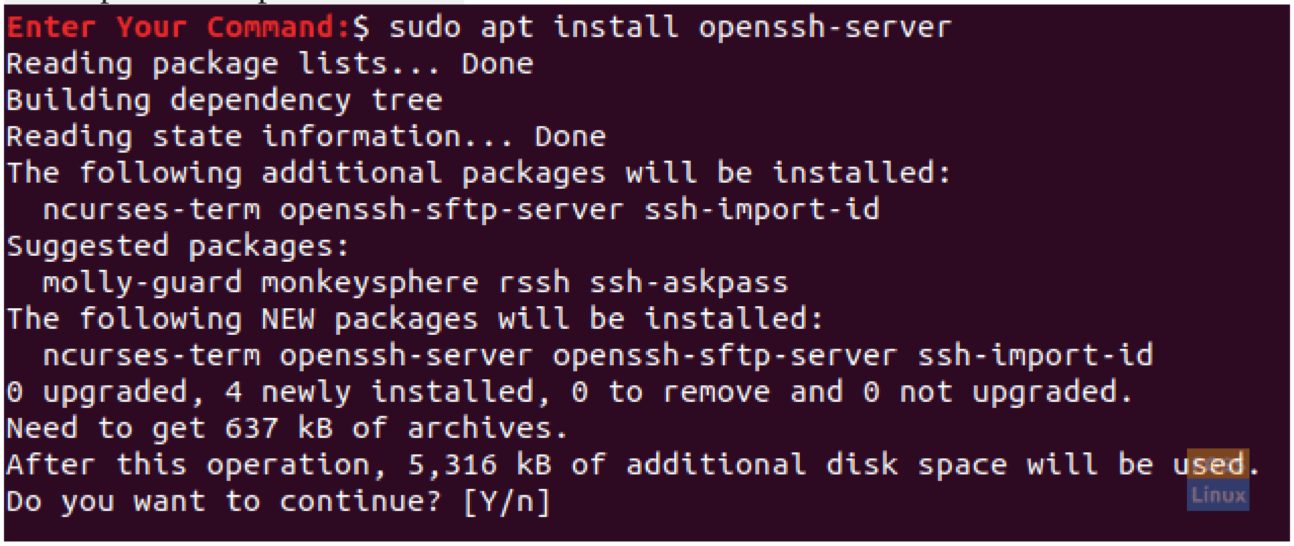
IV. Exchange public keys between two VMs using ssh-copy-id command. V. File Transfer between VMs on same Host and different Hosts

1. Transfer File / Directory from VM1 to VM2 using Secure copy command (scp). 2. Transfer File / Directory from VM2 to VM1 using Secure copy

command (scp). 3. Receive a file / directory from VM1 to VM2 using Secure copy command (scp). 4. Try to transfer a file from physical machine (Windows as host OS) to VM1.

5. Try to transfer a file from VM1 to physical machine (Windows as host OS).

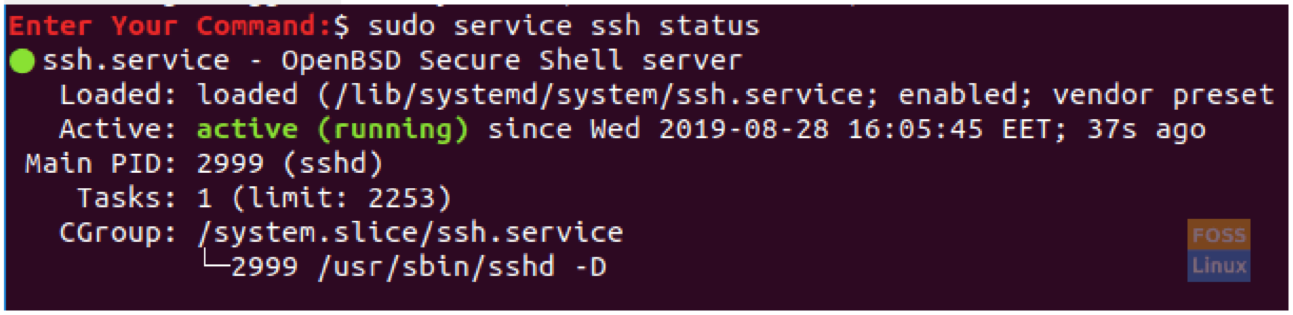
Step 1. Install the SSH package on Ubuntu using the next command.
sudo apt install openssh-server



```
Enter Your Command:$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere rssh ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 637 kB of archives.
After this operation, 5,316 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Install The Open SSH Package On Ubuntu

Step 2. After installation finishes successfully, you can check the SSH service status using the following command.
sudo service ssh status



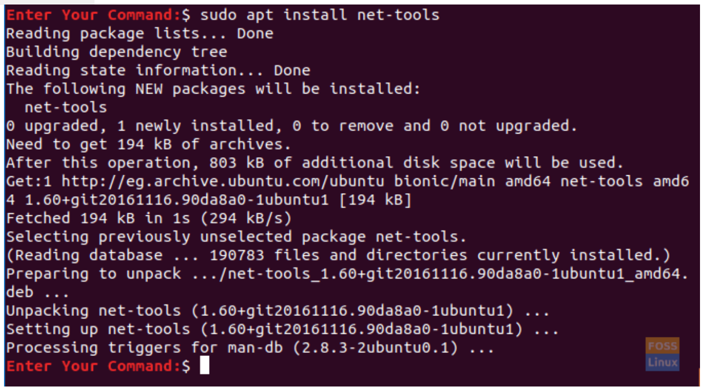
```
Enter Your Command:$ sudo service ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset
   Active: active (running) since Wed 2019-08-28 16:05:45 EET; 37s ago
 Main PID: 2999 (sshd)
    Tasks: 1 (limit: 2253)
   CGroup: /system.slice/ssh.service
           └─2999 /usr/sbin/sshd -D
```

Check The SSH Service Status

If the service is not running, you can use the following command to start it:
sudo service ssh start
sudo service ssh enable

Step 3. Install the net-tools package.

sudo apt install net-tools

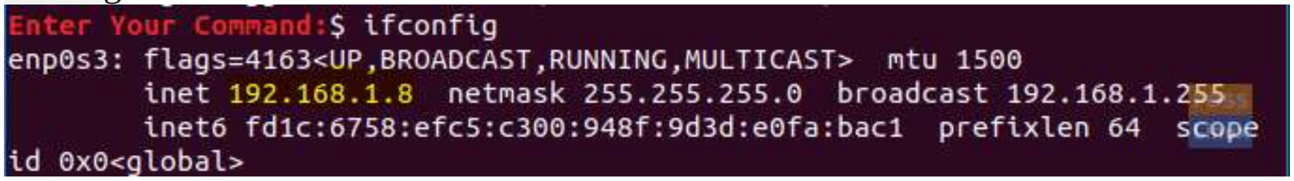


```
Enter Your Command:$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 194 kB of archives.
After this operation, 803 kB of additional disk space will be used.
Get:1 http://eg.archive.ubuntu.com/ubuntu bionic/main amd64 net-tools amd6
4 1.60+git20161116.90da8a0-1ubuntu1 [194 kB]
Fetched 194 kB in 1s (294 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 190783 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20161116.90da8a0-1ubuntu1_amd64.
deb ...
Unpacking net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Setting up net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Enter Your Command:$
```

Install net-tools package

Step 4. Execute the next command to get your Ubuntu machine IP.

ifconfig



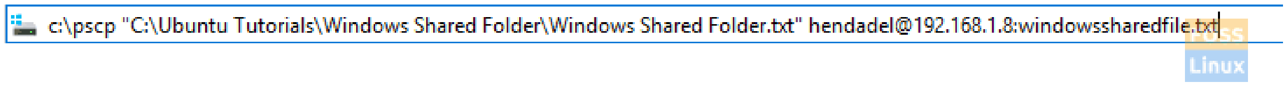
```
Enter Your Command:$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.8  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fd1c:6758:efc5:c300:948f:9d3d:e0fa:bac1  prefixlen 64  scope
id 0x0<global>
```

Ubuntu Machine IP

Step 5. From your Windows machine, we need an SSH client (which is Putty) and a PSCP. PSCP is considered a secure copy tool that is used alongside with Putty to transfer files securely over a network.

You can download both (Putty and PSCP) from the Putty official website.Kindly note that Putty will need to be installed while the PSCP won’t. You have to put the PSCP.exe in the “C:\” drive to be used, as you should see shortly.
Step 6. Now open the file explorer and use the next command to start transferring your files from your Windows machine to Ubuntu.

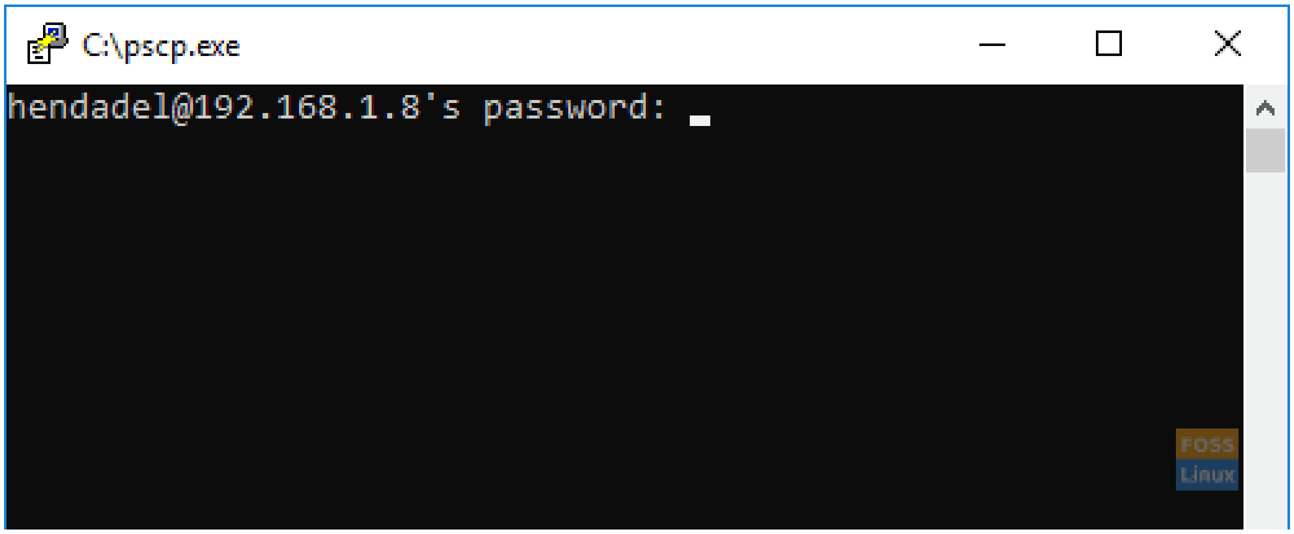
c:\pscp "C:\Ubuntu Tutorials\Windows Shared Folder\Windows Shared Folder.txt" hendadel@192.168.1.8:windowssharedfile.txt



```
c:\pscp "C:\Ubuntu Tutorials\Windows Shared Folder\Windows Shared Folder.txt" hendadel@192.168.1.8:windowssharedfile.txt
```

Copy File From Windows To Ubuntu Via SSH

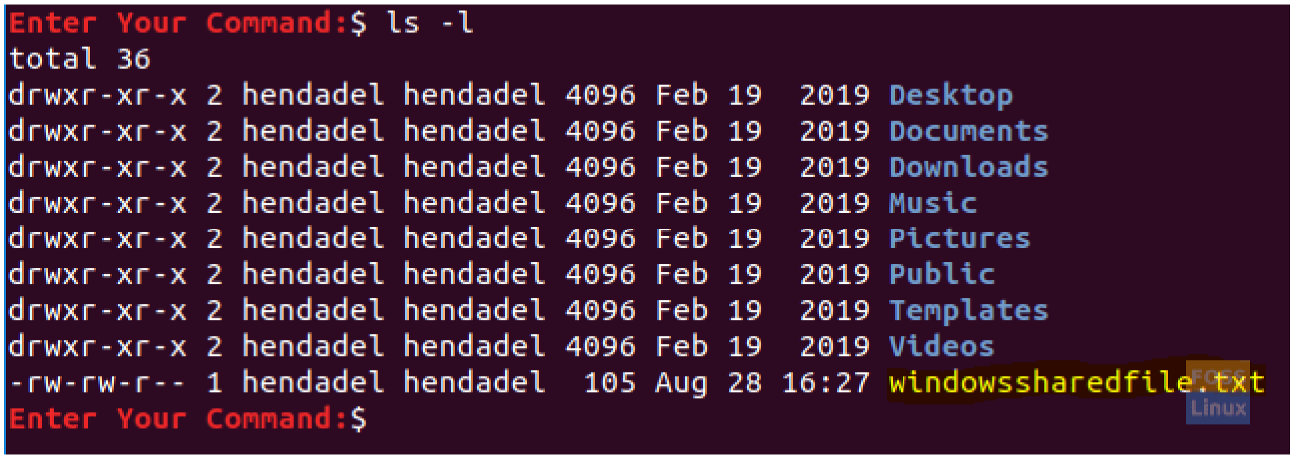
Step 7. Next, you should enter your Ubuntu password.



```
hendadel@192.168.1.8's password: _
```

Enter Your Ubuntu Password

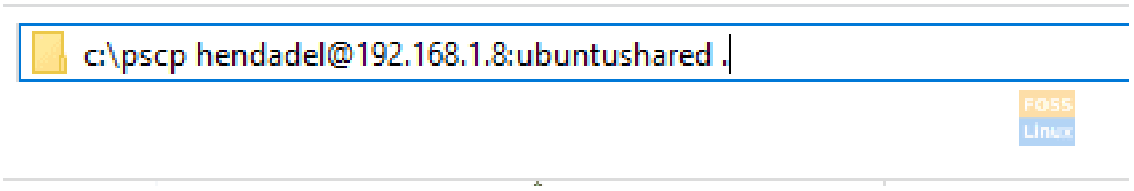
Step 8. After entering the password successfully, the command shall copy the text file from your Windows machine to your Ubuntu home. You can check your home directory now, and you should find the file there.



```
Enter Your Command:$ ls -l
total 36
drwxr-xr-x 2 hendadel hendadel 4096 Feb 19  2019 Desktop
drwxr-xr-x 2 hendadel hendadel 4096 Feb 19  2019 Documents
drwxr-xr-x 2 hendadel hendadel 4096 Feb 19  2019 Downloads
drwxr-xr-x 2 hendadel hendadel 4096 Feb 19  2019 Music
drwxr-xr-x 2 hendadel hendadel 4096 Feb 19  2019 Pictures
drwxr-xr-x 2 hendadel hendadel 4096 Feb 19  2019 Public
drwxr-xr-x 2 hendadel hendadel 4096 Feb 19  2019 Templates
drwxr-xr-x 2 hendadel hendadel 4096 Feb 19  2019 Videos
-rw-rw-r-- 1 hendadel hendadel  105 Aug 28 16:27 windowssharedfile.txt
Enter Your Command:$
```

Check The Copied File

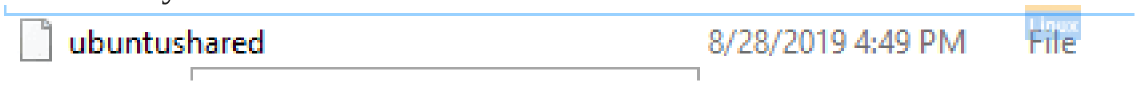
Step 9. Now in case you need to transfer your files from Ubuntu machine to Windows, open the file explorer from your Windows machine and use the next command. Be careful, and do not forget the dot at the end of the command.
c:\pscp hendadel@192.168.1.8:ubuntushared .



```
c:\pscp hendadel@192.168.1.8:ubuntushared .
```

Copy File From Ubuntu To Windows Via SSH

Step 10: By executing the previous command, you should enter the password of your Ubuntu machine. Next, the file gets transferred from Ubuntu to the Windows current directory.



```
ubuntushared                         8/28/2019 4:49 PM     File
```

File Copied Successfully

Results:

Thus the file sharing between two virtual machines has been implemented successfully using open SSH client.

EX NO:9

Trystack Virtual Machine

DATE: 13.10.2020

Objective:

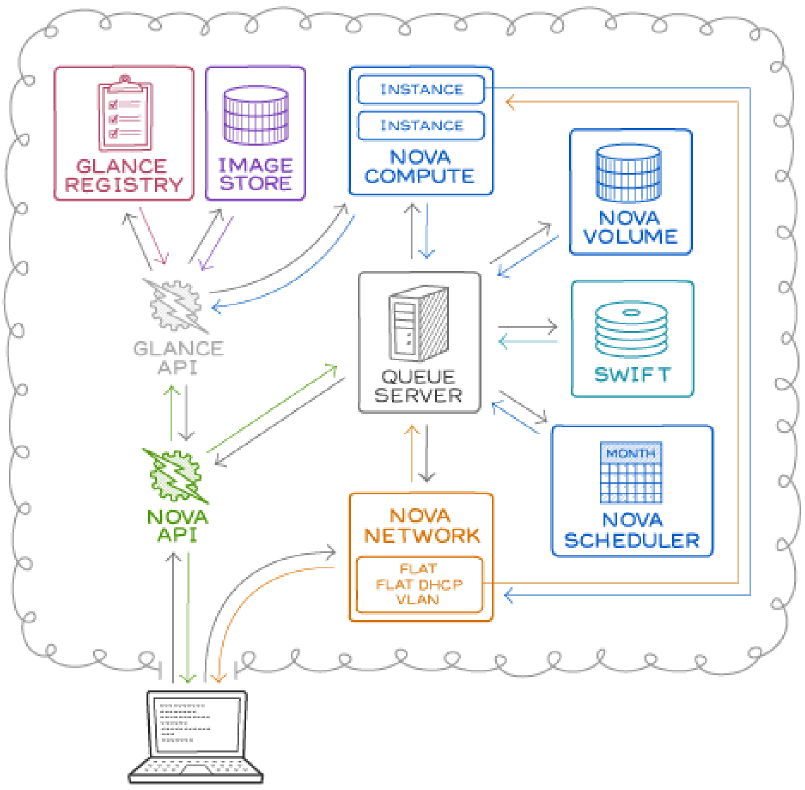
To explore trystack/openstack platform and create a virtual machine on platform managed by Openstack sandbox.

Requirements:

- Web browser
- Open stack Sandbox

Theory:

OpenStack is an open source platform, which offers powerful virtual servers and required services for cloud computing. It is mostly deployed as Infrastructure-as-a-service (IaaS), which aims to provide hardware tools and components for processing, storage, and networking resources throughout a data center.



OpenStack can be understood as a software platform that uses pooled virtual resources to build and manage clouds, both public and private ones.
By default, OpenStack offers a couple of cloud-related services like networking, storage, image services, identity, etc., and can be clubbed with a few more to get a customized cloud optimization to support the cloud-native apps.

OpenStack community has declared around 9 components to be an integral part of OpenStack.

They are:
• Nova: This is the fundamental computing engine of OpenStack. It manages a large number of Virtual machines and other instances, which handle computing tasks.

• Swift: Swift is the storage system of OpenStack. It is used to store the objects and files. Instead of referring to the file and objects through the path, developers can instead refer to them through a unique identifier, which points to a file or piece of information and thereby allow the OpenStack to manage where to store the files. This reduces the effort of the developers to understand and worry about storage distribution. This also ensures that the data is being backed up, if in case of any failure of the machine or network loss.

• Cinder: Cinder is known as the block storage component of OpenStack. This functions in a way, analogous to the traditional ways of locating and accessing specific locations on a disk or a drive.
• Neutron: As the name suggests, Neutron is the component that enables networking in OpenStack. It ensures that each component within the OpenStack is well connected with other components, to establish good communication amongst them.

• Horizon: Horizon is the dashboard of the OpenStack system. It provides all the possibilities for the system administrators to access and manage the cloud. This is the first component that everyone "sees" upon starting to use the OpenStack. Developers will be able to access and deal with all the components through the Application Programming Interface (API) also, while Horizon is the only place through which the system admins will be interacting with the OpenStack architecture.

• Keystone: Keystone is the component that provides the identity services for all the users. It basically contains a central list of all the users of the OpenStack cloud, mapped to the accessible services of the OpenStack. It provides a way for multiple accesses by allowing the developers to map their existing user access methods to the Keystone.

• Glance: Glance provides the image services in OpenStack, where images refer to the virtual copies of the hard disks. Glance helps in allocating these images to be used as templates while assigning new virtual machine instances.

• Ceilometer: Ceilometer provides telemetry services to its users. It performs a close regulation of each user's cloud components' usage and provides a bill for the services used. Think of Ceilometer as a component to meter the usage and report the same to individual users.

• Heat: Heat is that component of the OpenStack which allows developers to store the requirements of a cloud application in a file so that all the resources necessary for a program are available at hand. It thus provides an infrastructure to manage a cloud application. It is an orchestration instrument of OpenStack.

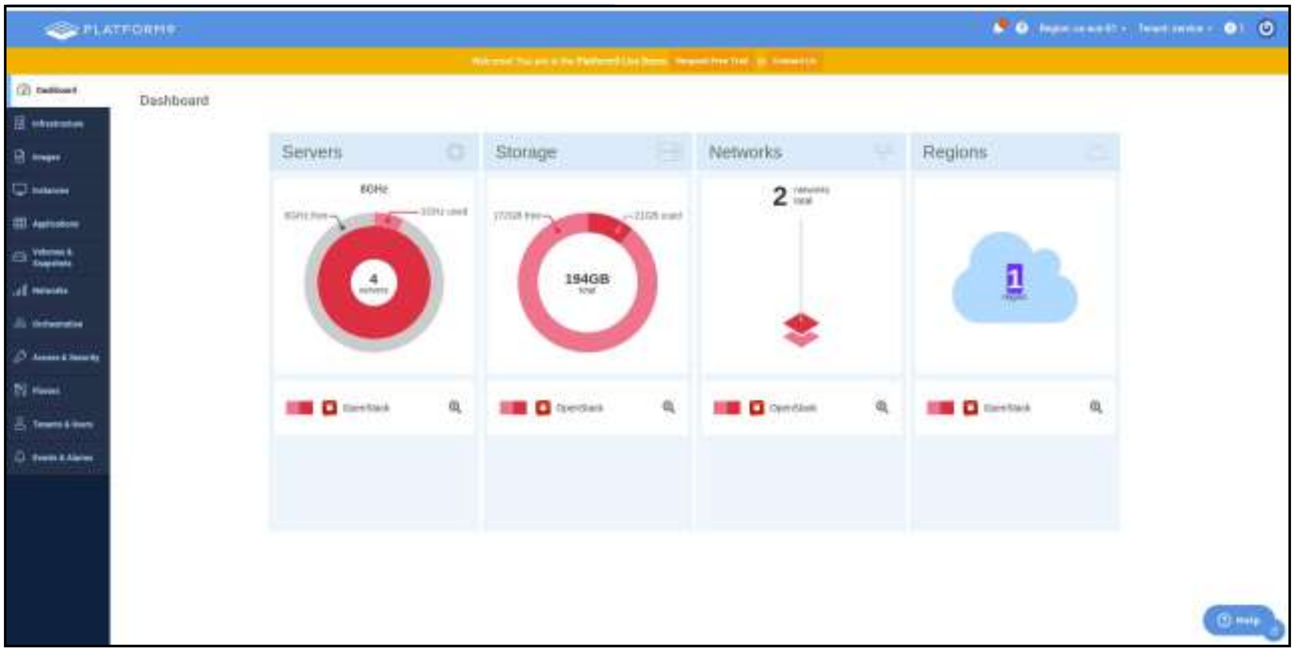
Procedure and implementation:

1. Setting up an account:

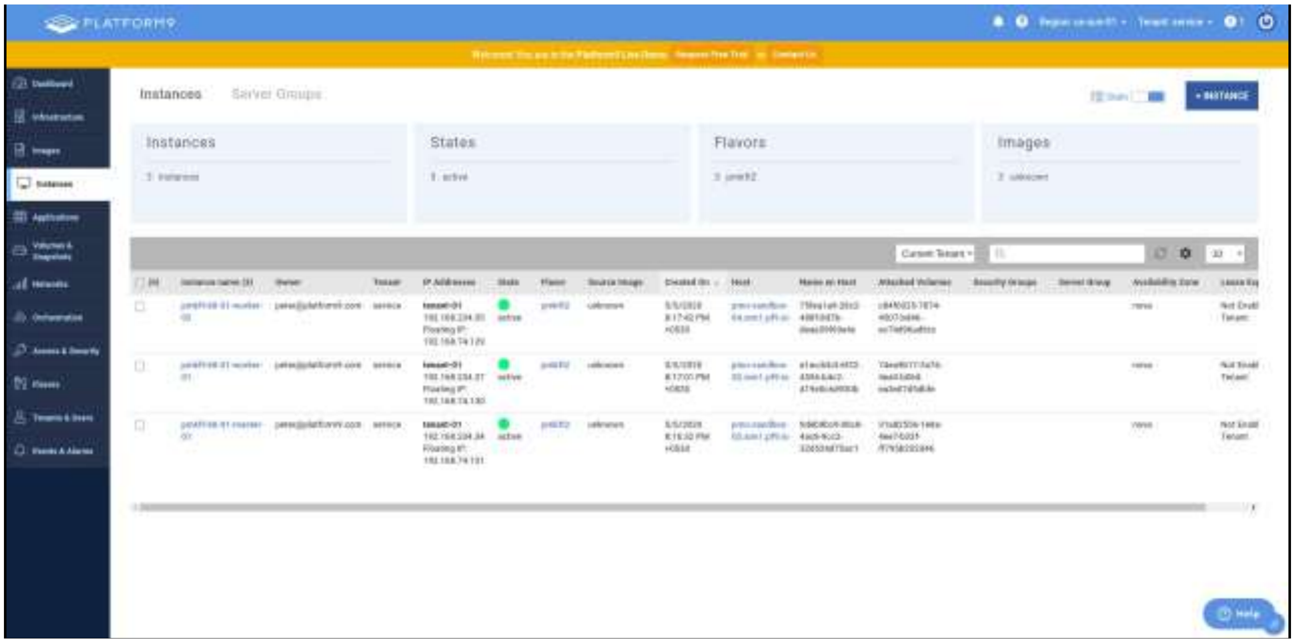
1.1. Visit https://platform9.com/managed-openstack/ and create a free account. 1.2. Login to the account and move to the sandbox mode.

2. Creating a VM instance:

2.1. Login and move to the Dashboard



2.2 Click on the instances Tab to view current instances and choose on “create a new instance”.

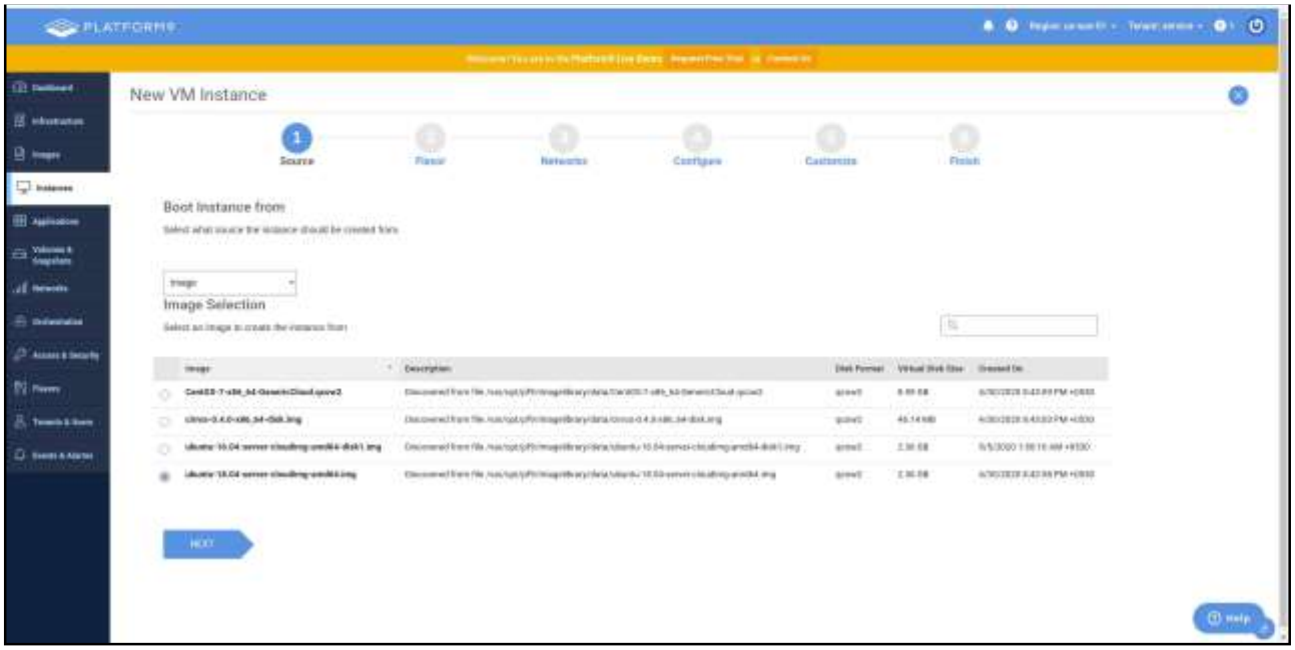


Instances | Server Groups

Instances	States	Flavors	Images
3 instances	3 active	3 [illegible]	2 [illegible]

Instance name	Owner	Tenant	IP Addresses	State	Flavor	Source Image	Created On	Host	Name on Host	Attached Volumes	Security Groups	Server Group	Availability Zone	Lease Exp
[illegible] worker 02	[illegible]@platform9.com	service	tenant-01 [illegible] Floating IP: 192.168.74.129	active	[illegible]	unknown	5/5/2020 8:17:42 PM +0530	[illegible]	[illegible]	[illegible]			none	Not Enabled Tenant
[illegible] worker 01	[illegible]@platform9.com	service	tenant-01 [illegible] Floating IP: 192.168.74.130	active	[illegible]	unknown	5/5/2020 8:17:00 PM +0530	[illegible]	[illegible]	[illegible]			none	Not Enabled Tenant
[illegible] master 01	[illegible]@platform9.com	service	tenant-01 [illegible] Floating IP: 192.168.74.131	active	[illegible]	unknown	5/5/2020 8:16:32 PM +0530	[illegible]	[illegible]	[illegible]			none	Not Enabled Tenant

2.3 Choose an OS DISK Image to use.



New VM Instance

1 Source · 2 Flavor · 3 Networks · 4 Configure · 5 Customize · 6 Finish

Boot Instance from

Select what source the instance should be created from

Image

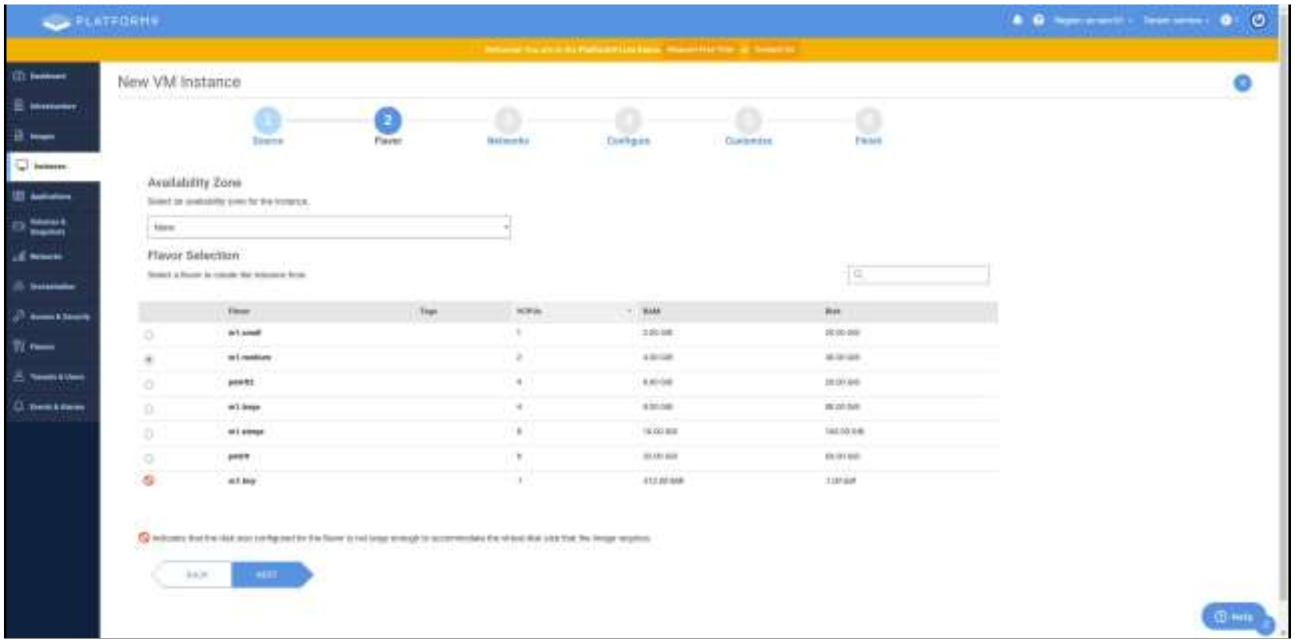
Image Selection

Select an image to create the instance from

	Image	Description	Disk Format	Virtual Disk Size	Created On
○	CentOS-7-x86_64-GenericCloud.qcow2	Discovered from file [illegible]	qcow2	[illegible] GB	[illegible]
○	cirros-0.4.0-x86_64-disk.img	Discovered from file [illegible]	qcow2	46.14 MB	[illegible]
○	ubuntu-16.04-server-cloudimg-amd64-disk1.img	Discovered from file [illegible]	qcow2	2.30 GB	5/5/2020 [illegible]
●	ubuntu-18.04-server-cloudimg-amd64.img	Discovered from file [illegible]	qcow2	2.30 GB	[illegible]

NEXT

2.4 Next Choose the type of VM based on the CPU and Memory Requirements.



New VM Instance

Availability Zone

[illegible]

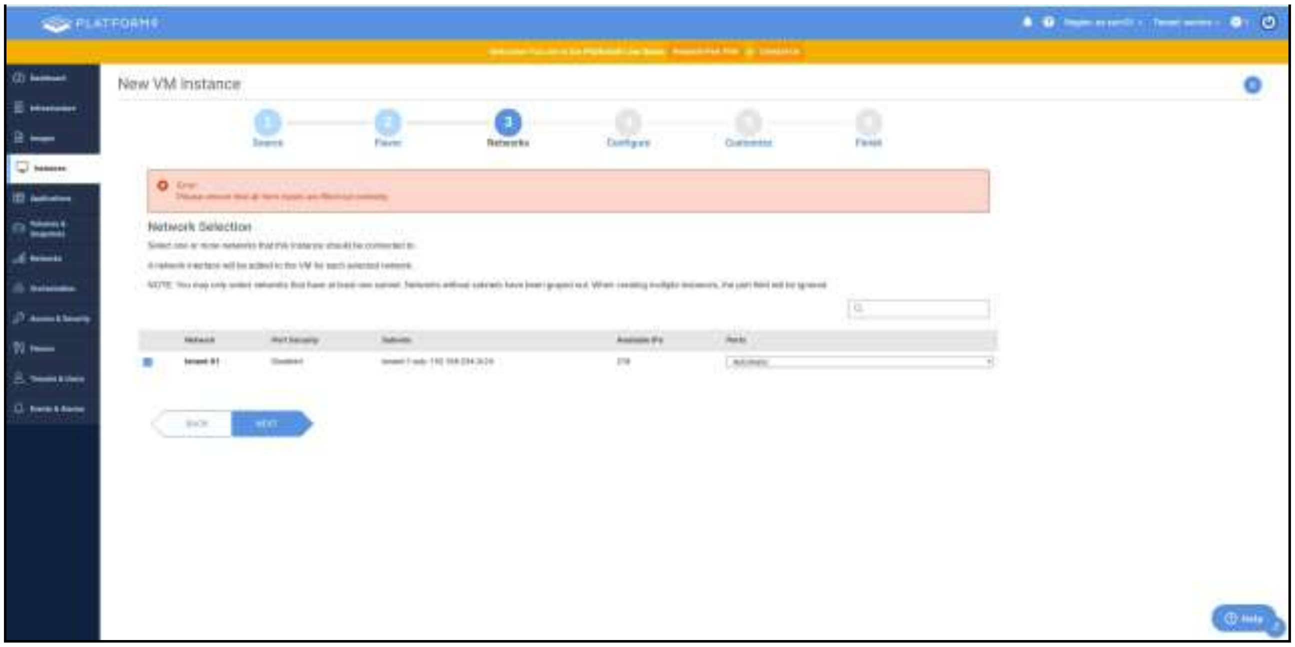
Flavor Selection

[illegible]

	Flavor	Tags	VCPUs	RAM	Disk
○	m1.small		1	2.00 GB	20.00 GB
●	m1.medium		2	4.00 GB	40.00 GB
○	[illegible]		4	8.00 GB	20.00 GB
○	m1.large		4	8.00 GB	80.00 GB
○	m1.xlarge		8	16.00 GB	160.00 GB
○	[illegible]		8	[illegible]	[illegible]
🚫	m1.tiny		1	512.00 MB	1.00 GB

🚫 [illegible]

2.5 Set/Select Network configuration for the VM.



New VM Instance

Error
Please ensure that all form inputs are filled out correctly

Network Selection

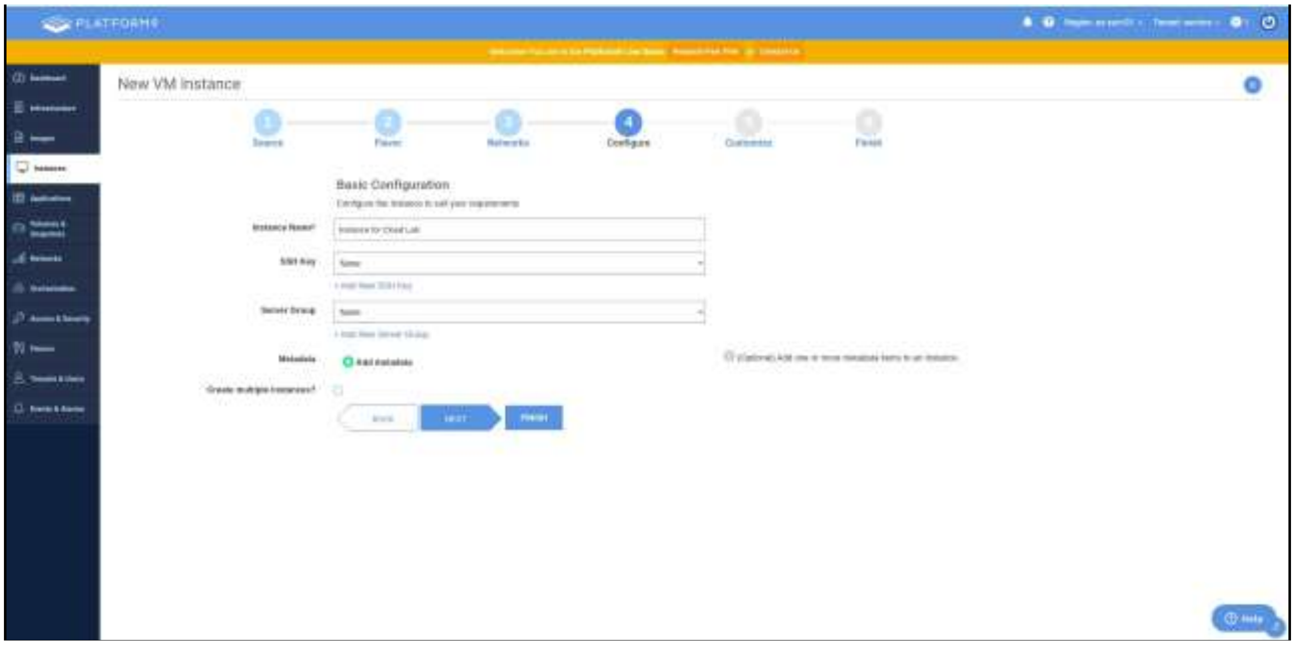
Select one or more networks that this instance should be connected to.

A network interface will be added to the VM for each selected network.

NOTE: You may only select networks that have at least one subnet. Networks without subnets have been grayed out. When creating multiple instances, the port field will be ignored.

	Network	Port Security	Subnets	Available IPs	Ports
✓	[illegible]	Disabled	[illegible] 192.168.254.0/24	[illegible]	Automatic

2.6 Set a name for the instance, then choose/create an ssh keypair for connecting to the VM also choose the number of such VMs to be created.



New VM Instance

Source · Flavor · Networks · Configure · Customize · Finish

Basic Configuration

Configure the instance to suit your requirements

Instance Name*: Instance for Cloud Lab

SSH Key: None

Server Group: None

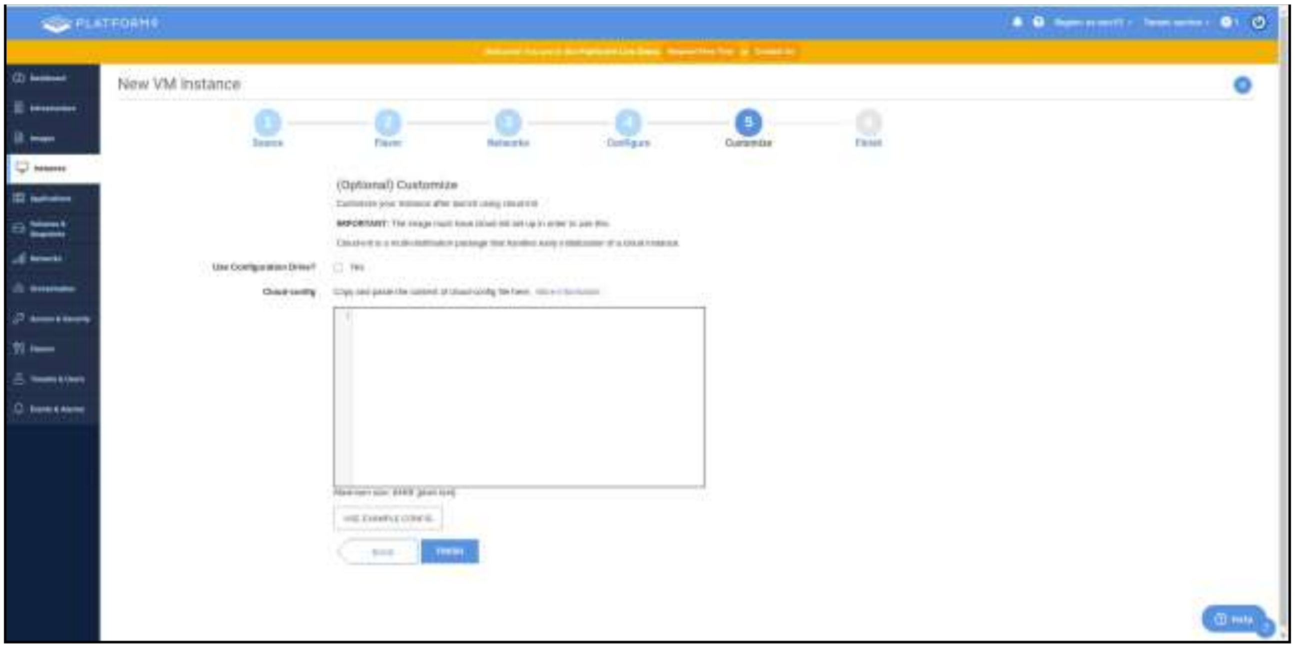
Metadata: Add metadata

(Optional) Add one or more metadata items to an instance.

Create multiple instances?

BACK · NEXT · FINISH

2.7 Optionally add any additional configuration for the VM.



New VM Instance

Source · Flavor · Networks · Configure · Customize · Finish

(Optional) Customize

Customize your instance after launch using cloud-init

IMPORTANT: The image must have cloud-init set up in order to use this.

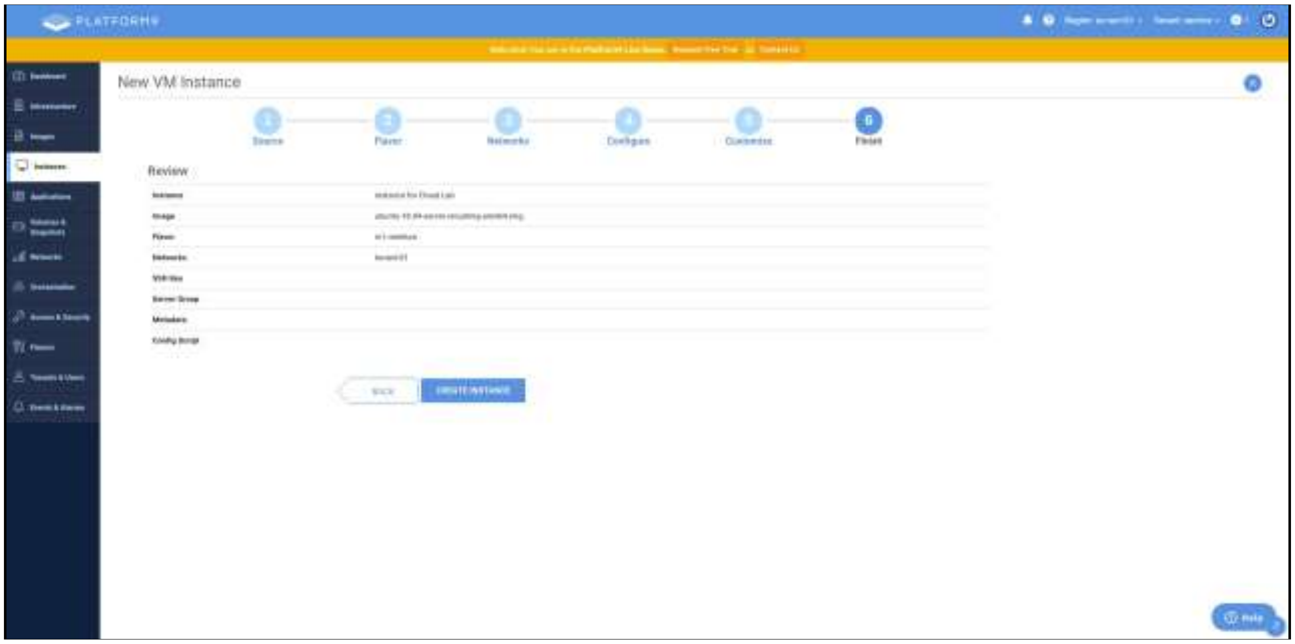
Use Configuration Drive? Yes

Cloud-config: Copy and paste the content of cloud-config file here.

USE EXAMPLE CONFIG

BACK · FINISH

2.8 Finally, Click on “create instance” to successfully create the VM.



Result:

Trystack/Openstack was explored and a VM was successfully created on Platform9 Managed Openstack sandbox.

EX NO:10 **Hadoop and Word Count**

DATE: 20.10.2020

Objective:

To understand the features of Hadoop and to implement a word count program.

Requirement:

1. OperatingSystem:Ubuntu16.04LTSDesktop(64-Bits only)OS
2. InstallationMode: Installing GuestOS(Note:Virtual Machine can be slower to work when Hadoop Cluster starts)
3. Java: JDK 1.8
4. Download latest version of Apache Hadoop package
5. Eclipse Luna 64-bit for Linux (Can be downloaded from SSN Intranet – Tech Support). Install Eclipse. (If Necessary)
6. Create a Virtual Machine and install Ubuntu 16.04 Desktop AMD 64.iso in VM. Name the VM as HadoopVM and set network configuration.

Theory:

The Apache™ Hadoop project develops open-source software for reliable, scalable, distributed computing.

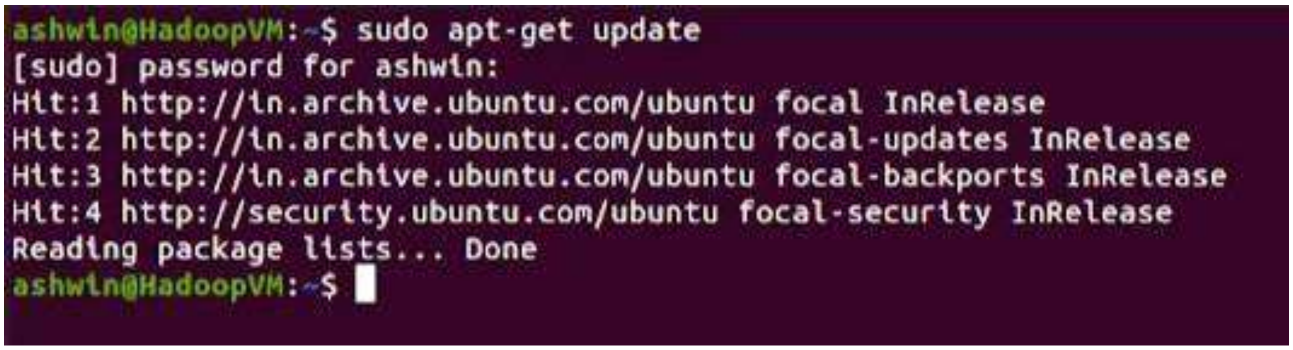
The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

Apache Hadoop is an open source framework that is used to efficiently store and process large datasets ranging in size from gigabytes to petabytes of data. Instead of using one large computer to store and process the data, Hadoop allows clustering multiple computers to analyze massive datasets in parallel more quickly.

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs).
Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

Procedure and Implementation:

HadoopVM :



```
ashwin@HadoopVM:~$ sudo apt-get update
[sudo] password for ashwin:
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
ashwin@HadoopVM:~$
```

➔ Create a New Ubuntu 20.04 Desktop VM .

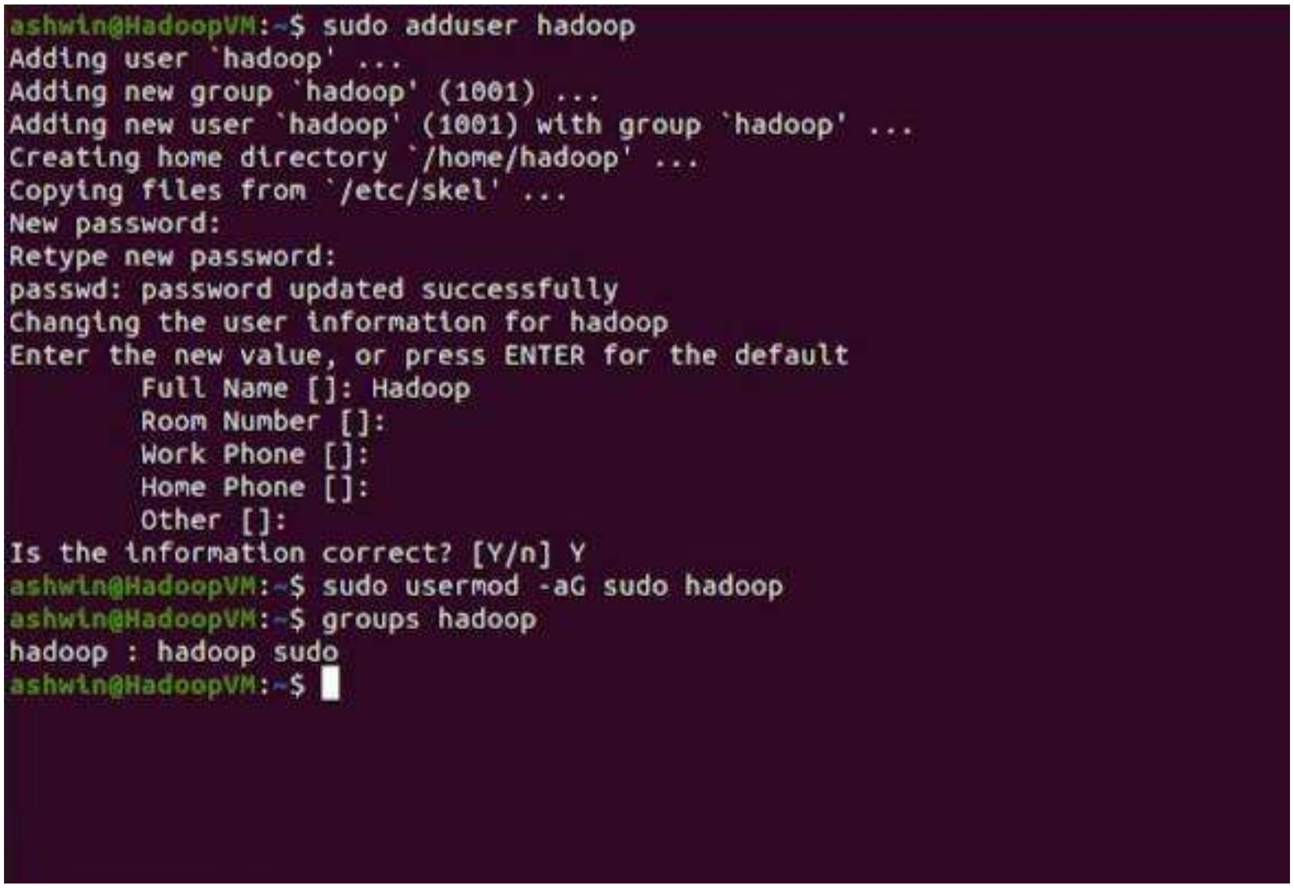
➔ Provide 1Gb Ram and 20 Gb Disk space .

➔ Launch the VM .

Hadoop Installation in HadoopVM :

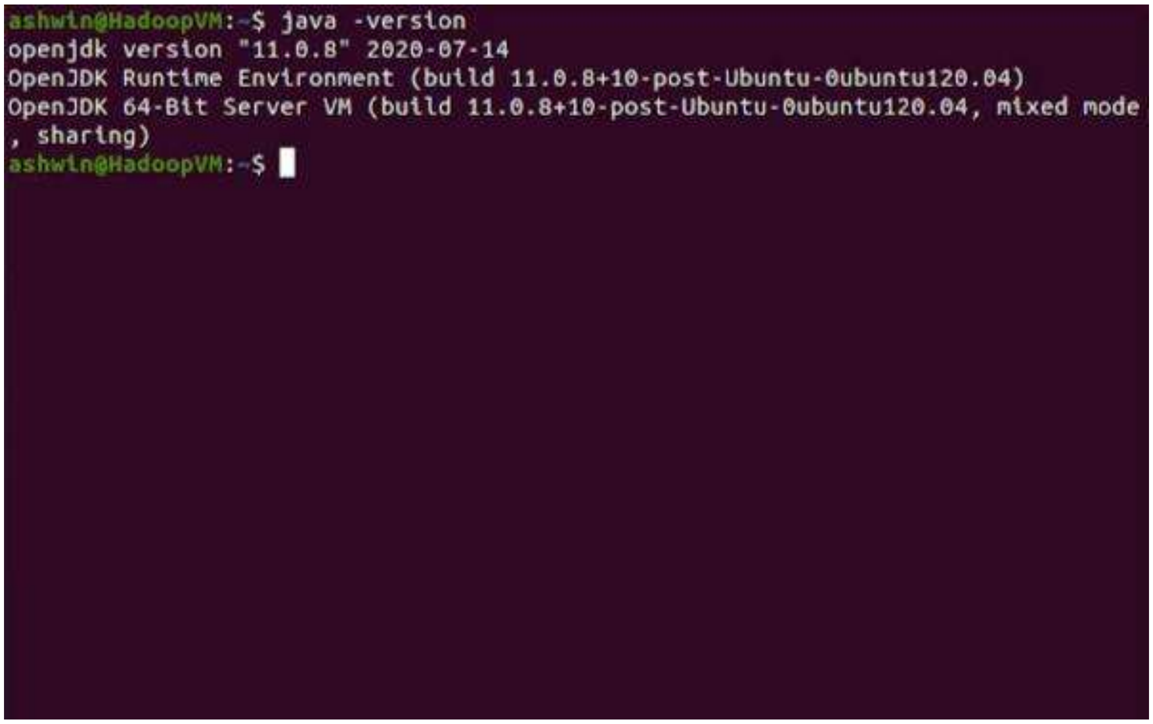
➔ Update Ubuntu using sudo apt-get update .

➔ Create a New User ‘Hadoop ’and add it to the sudo users list .



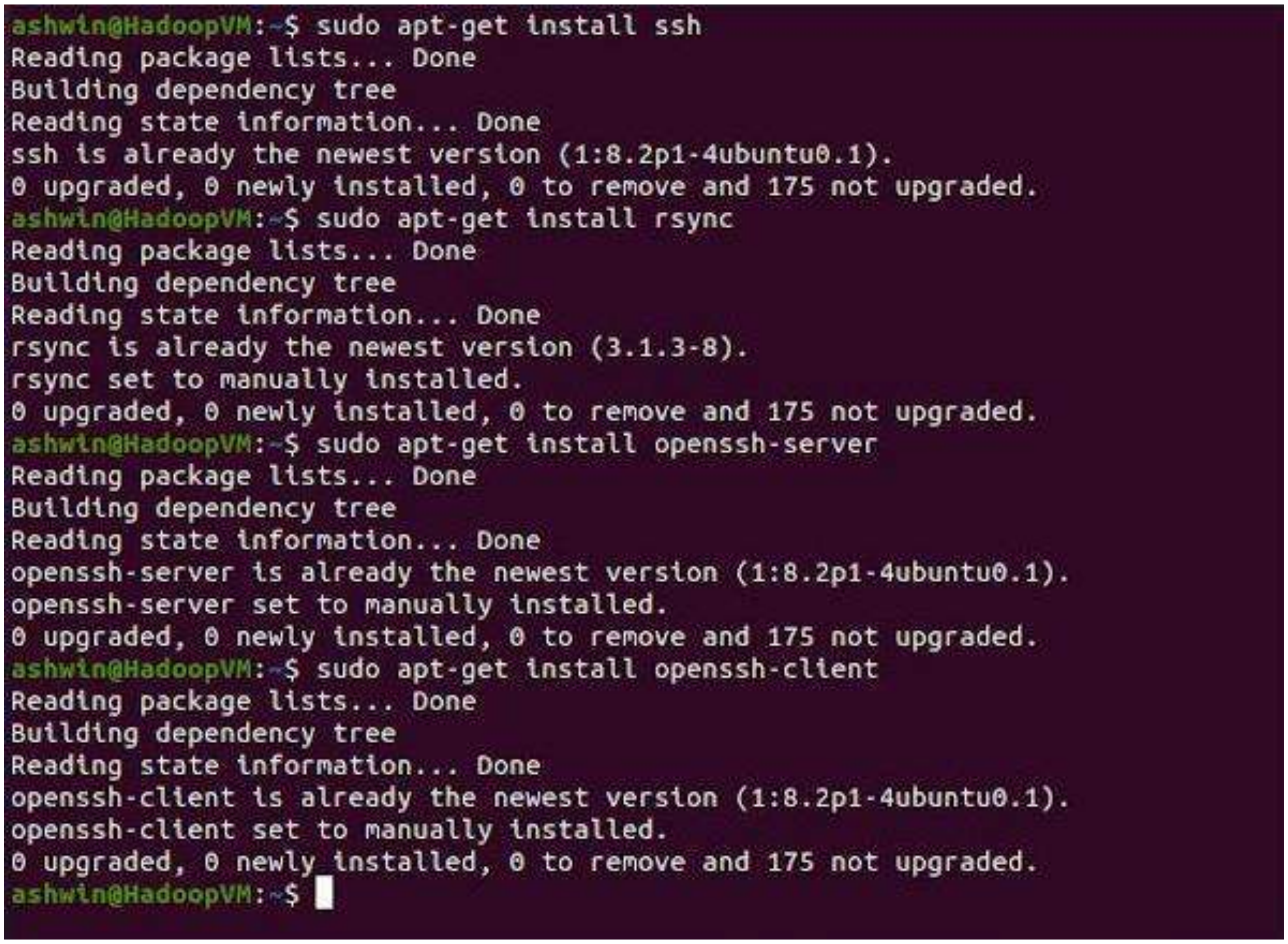
```
ashwin@HadoopVM:~$ sudo adduser hadoop
Adding user `hadoop' ...
Adding new group `hadoop' (1001) ...
Adding new user `hadoop' (1001) with group `hadoop' ...
Creating home directory `/home/hadoop' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for hadoop
Enter the new value, or press ENTER for the default
        Full Name []: Hadoop
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
ashwin@HadoopVM:~$ sudo usermod -aG sudo hadoop
ashwin@HadoopVM:~$ groups hadoop
hadoop : hadoop sudo
ashwin@HadoopVM:~$ 
```

➔ Install java using sudo apt-get install default-jdk default-jre and check java -version .



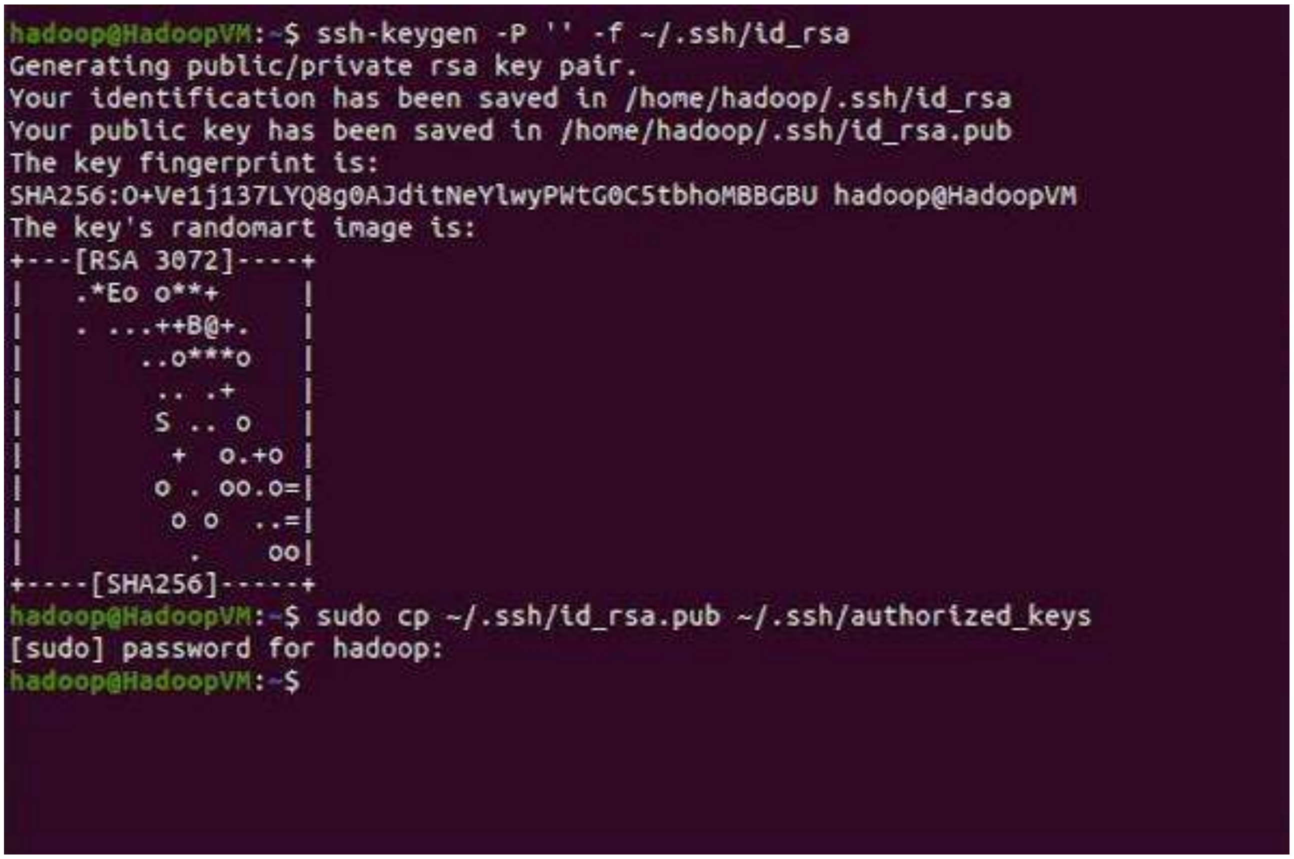
```
ashwin@HadoopVM:~$ java -version
openjdk version "11.0.8" 2020-07-14
OpenJDK Runtime Environment (build 11.0.8+10-post-Ubuntu-0ubuntu120.04)
OpenJDK 64-Bit Server VM (build 11.0.8+10-post-Ubuntu-0ubuntu120.04, mixed mode
, sharing)
ashwin@HadoopVM:~$
```

➔ Login to New User and install SSH .



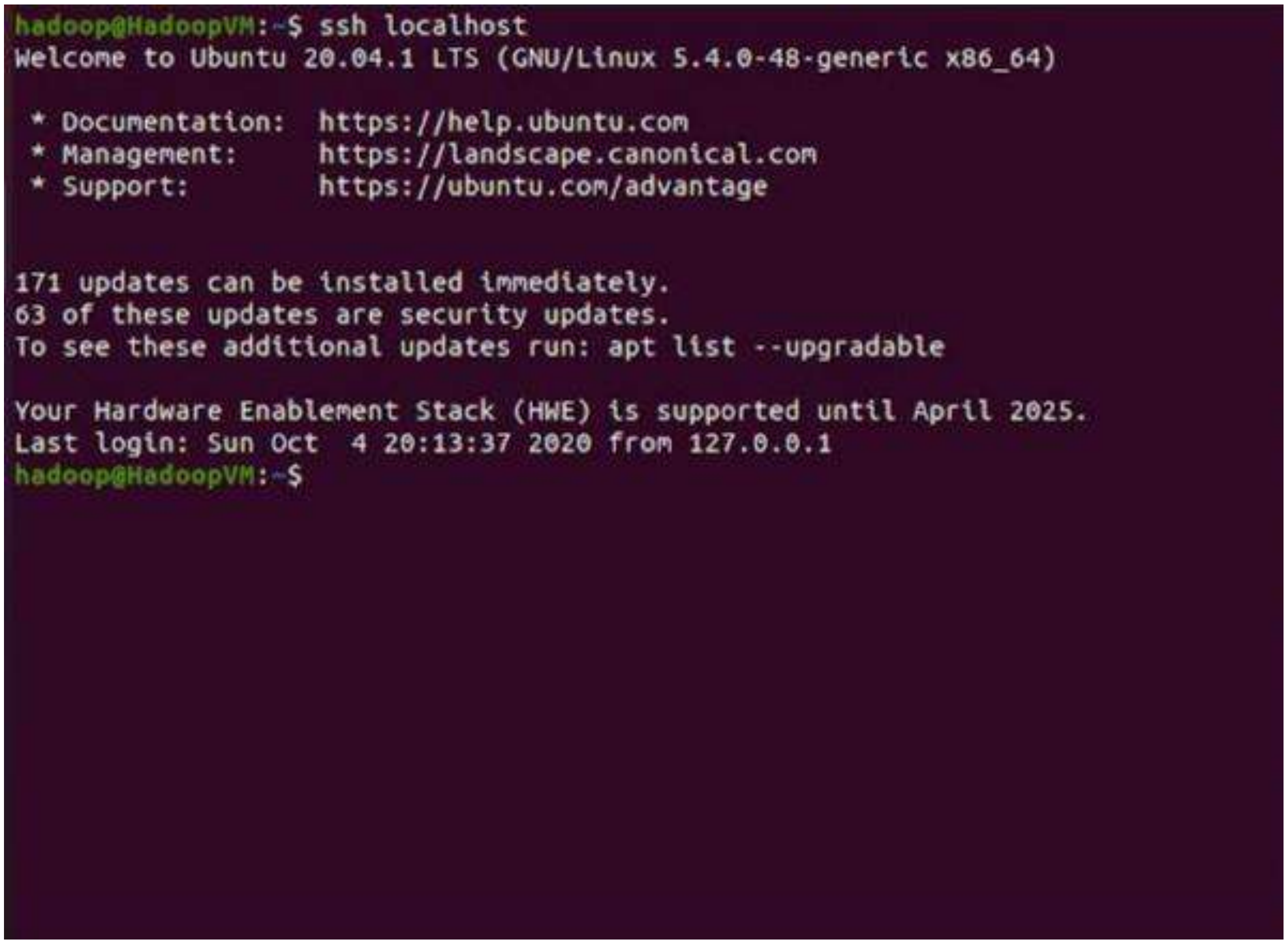
```
ashwin@HadoopVM:~$ sudo apt-get install ssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
ssh is already the newest version (1:8.2p1-4ubuntu0.1).
0 upgraded, 0 newly installed, 0 to remove and 175 not upgraded.
ashwin@HadoopVM:~$ sudo apt-get install rsync
Reading package lists... Done
Building dependency tree
Reading state information... Done
rsync is already the newest version (3.1.3-8).
rsync set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 175 not upgraded.
ashwin@HadoopVM:~$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:8.2p1-4ubuntu0.1).
openssh-server set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 175 not upgraded.
ashwin@HadoopVM:~$ sudo apt-get install openssh-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-client is already the newest version (1:8.2p1-4ubuntu0.1).
openssh-client set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 175 not upgraded.
ashwin@HadoopVM:~$
```

➔ Create SSH Key Pair and move it to Authorized_Keys.



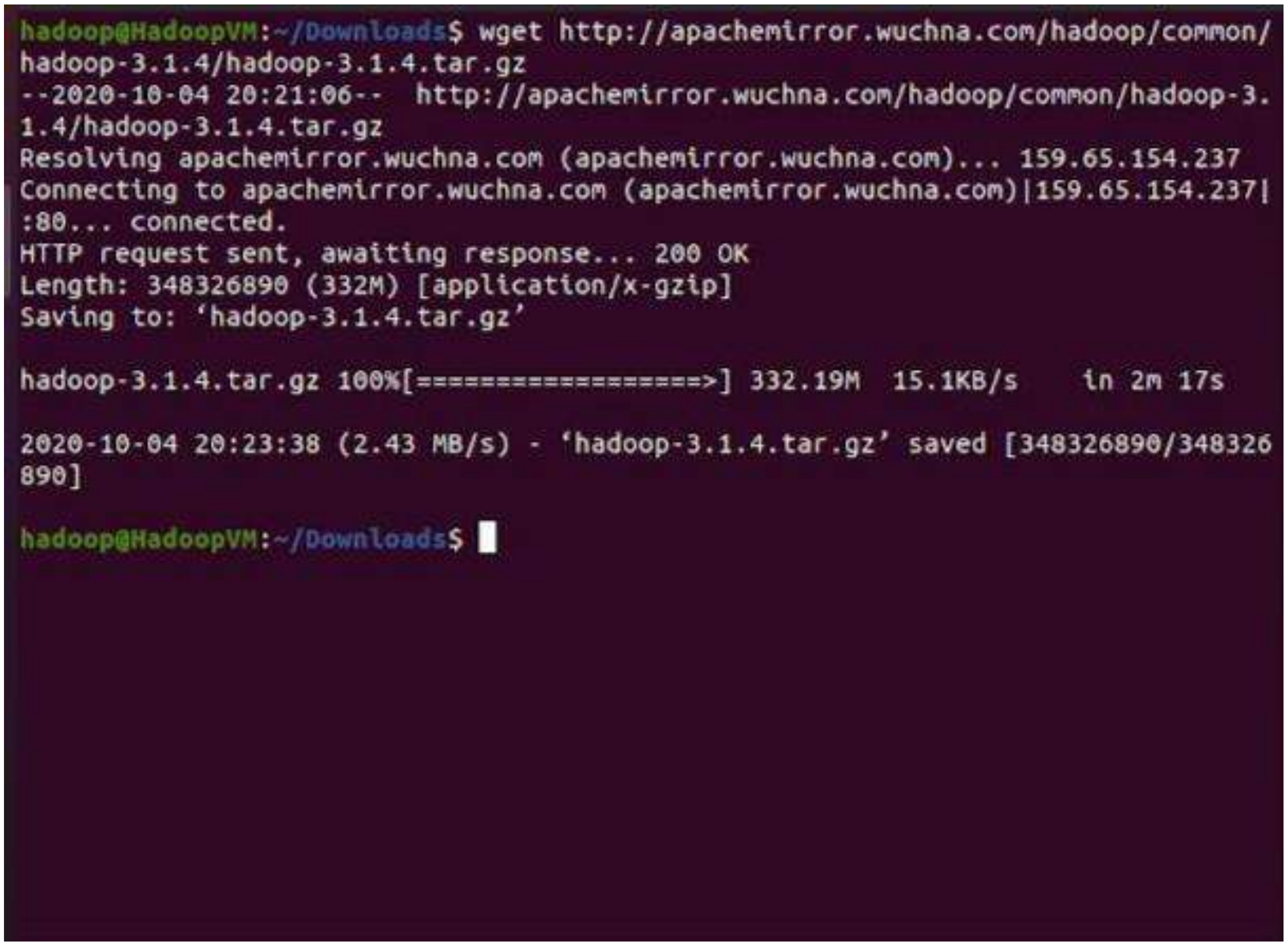
```
hadoop@HadoopVM:~$ ssh-keygen -P '' -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
Your identification has been saved in /home/hadoop/.ssh/id_rsa
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:O+Ve1j137LYQ8g0AJditNeYlwyPWtG0C5tbhoMBBGBU hadoop@HadoopVM
The key's randomart image is:
+---[RSA 3072]----+
|    .*Eo o**+    |
|    . ...++B@+.  |
|        ..o***o  |
|         .. .+   |
|         S .. o  |
|          +  o.+o |
|         o . oo.o=|
|          o o  ..=|
|           .    oo|
+----[SHA256]-----+
hadoop@HadoopVM:~$ sudo cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
[sudo] password for hadoop:
hadoop@HadoopVM:~$
```

➔ Verify SSH Login using ssh localhost .

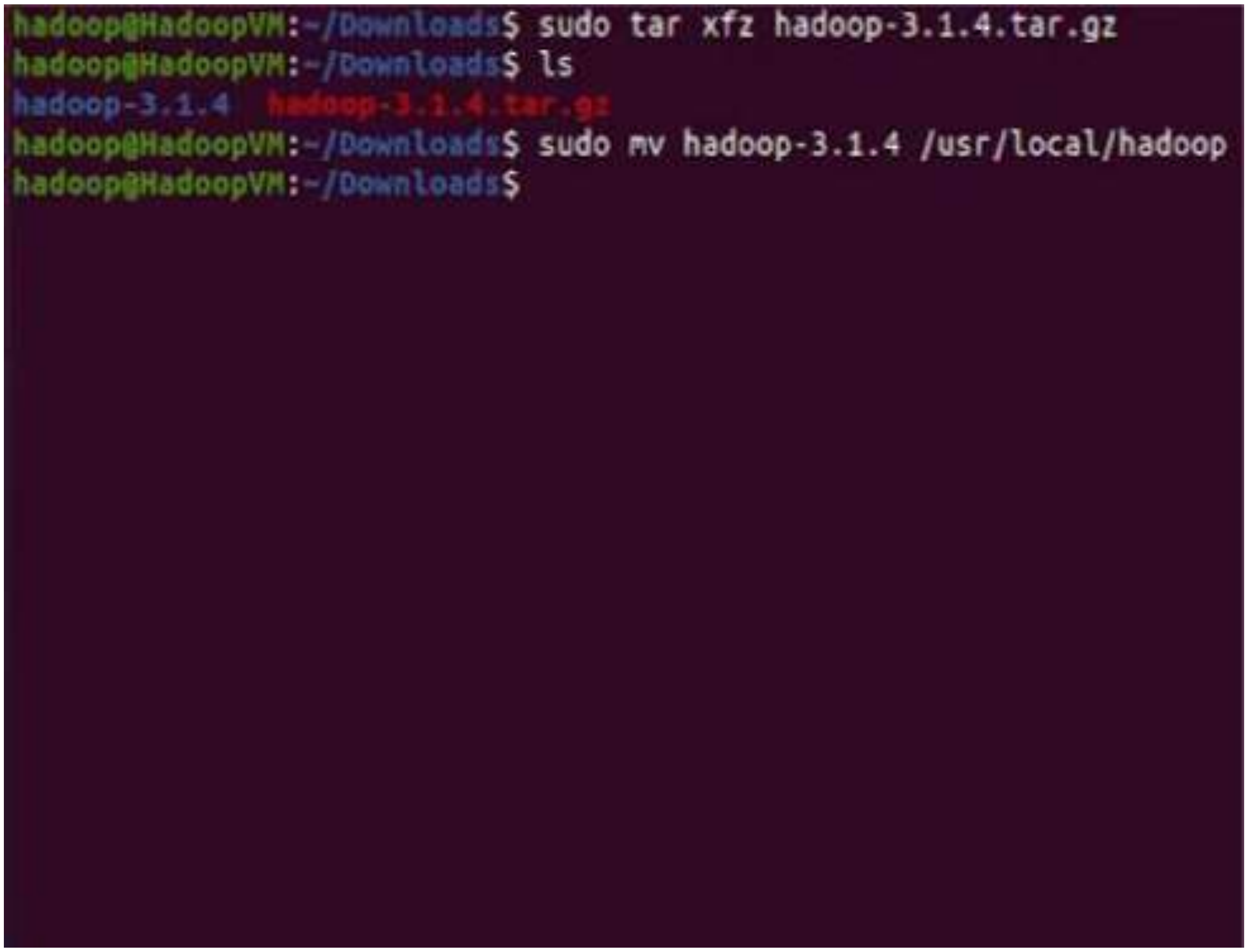


```
hadoop@HadoopVM:~$ ssh localhost
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-48-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


171 updates can be installed immediately.
63 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun Oct  4 20:13:37 2020 from 127.0.0.1
hadoop@HadoopVM:~$
```

➔ Download Hadoop Files and move them to Hadoop Directory .

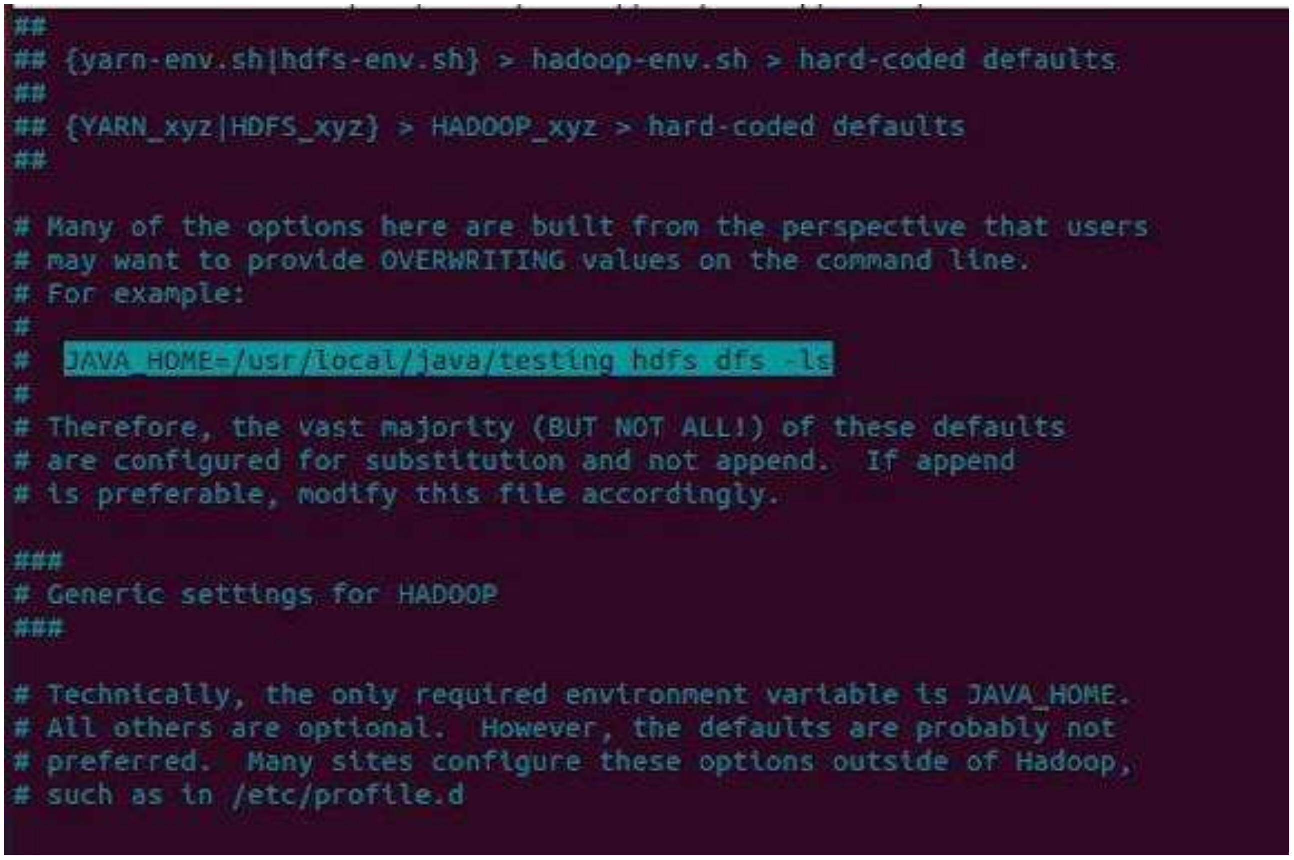


```
hadoop@HadoopVM:~/Downloads$ wget http://apachemirror.wuchna.com/hadoop/common/
hadoop-3.1.4/hadoop-3.1.4.tar.gz
--2020-10-04 20:21:06--  http://apachemirror.wuchna.com/hadoop/common/hadoop-3.
1.4/hadoop-3.1.4.tar.gz
Resolving apachemirror.wuchna.com (apachemirror.wuchna.com)... 159.65.154.237
Connecting to apachemirror.wuchna.com (apachemirror.wuchna.com)|159.65.154.237|
:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 348326890 (332M) [application/x-gzip]
Saving to: ‘hadoop-3.1.4.tar.gz’

hadoop-3.1.4.tar.gz 100%[==================>] 332.19M  15.1KB/s    in 2m 17s

2020-10-04 20:23:38 (2.43 MB/s) - ‘hadoop-3.1.4.tar.gz’ saved [348326890/348326
890]

hadoop@HadoopVM:~/Downloads$ 
```



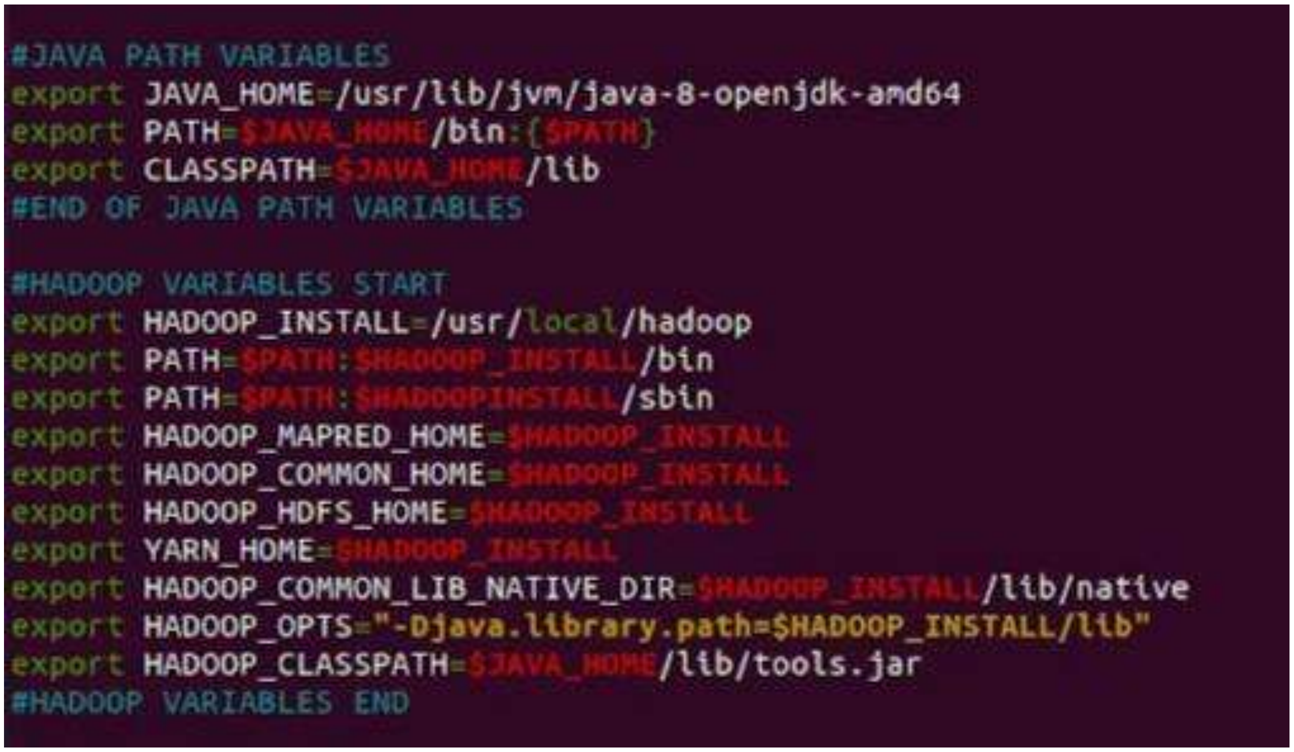
```
hadoop@HadoopVM:~/Downloads$ sudo tar xfz hadoop-3.1.4.tar.gz
hadoop@HadoopVM:~/Downloads$ ls
hadoop-3.1.4  hadoop-3.1.4.tar.gz
hadoop@HadoopVM:~/Downloads$ sudo mv hadoop-3.1.4 /usr/local/hadoop
hadoop@HadoopVM:~/Downloads$
```

➔ Include Java_Home variable to hadoop_env.sh .



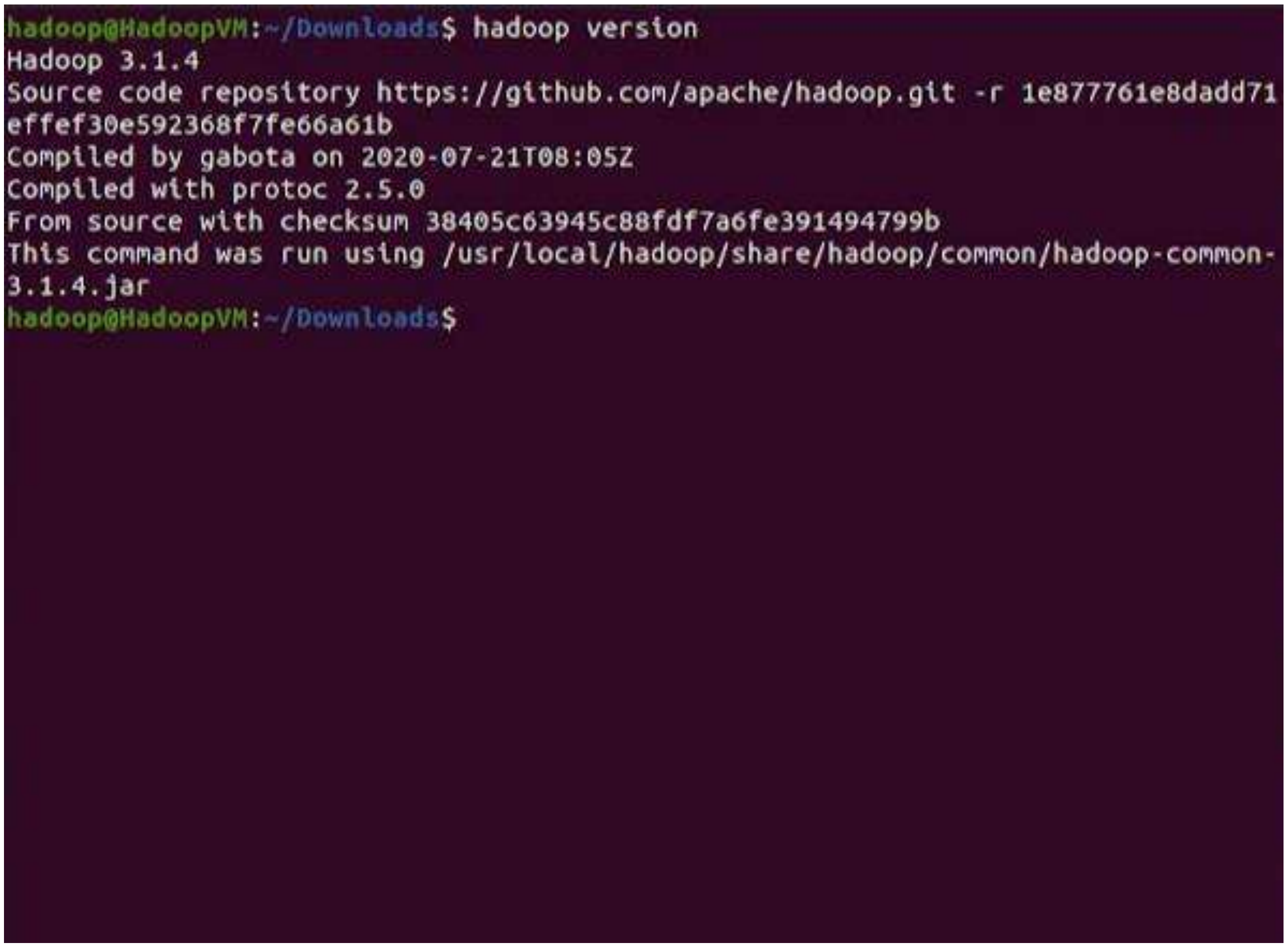
```
##
## {yarn-env.sh|hdfs-env.sh} > hadoop-env.sh > hard-coded defaults
##
## {YARN_xyz|HDFS_xyz} > HADOOP_xyz > hard-coded defaults
##

# Many of the options here are built from the perspective that users
# may want to provide OVERWRITING values on the command line.
# For example:
#
#  JAVA_HOME=/usr/local/java/testing hdfs dfs -ls
#
# Therefore, the vast majority (BUT NOT ALL!) of these defaults
# are configured for substitution and not append.  If append
# is preferable, modify this file accordingly.

###
# Generic settings for HADOOP
###

# Technically, the only required environment variable is JAVA_HOME.
# All others are optional.  However, the defaults are probably not
# preferred.  Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d
```

➔ Include Java Path variables and Hadoop variables to bashrc .



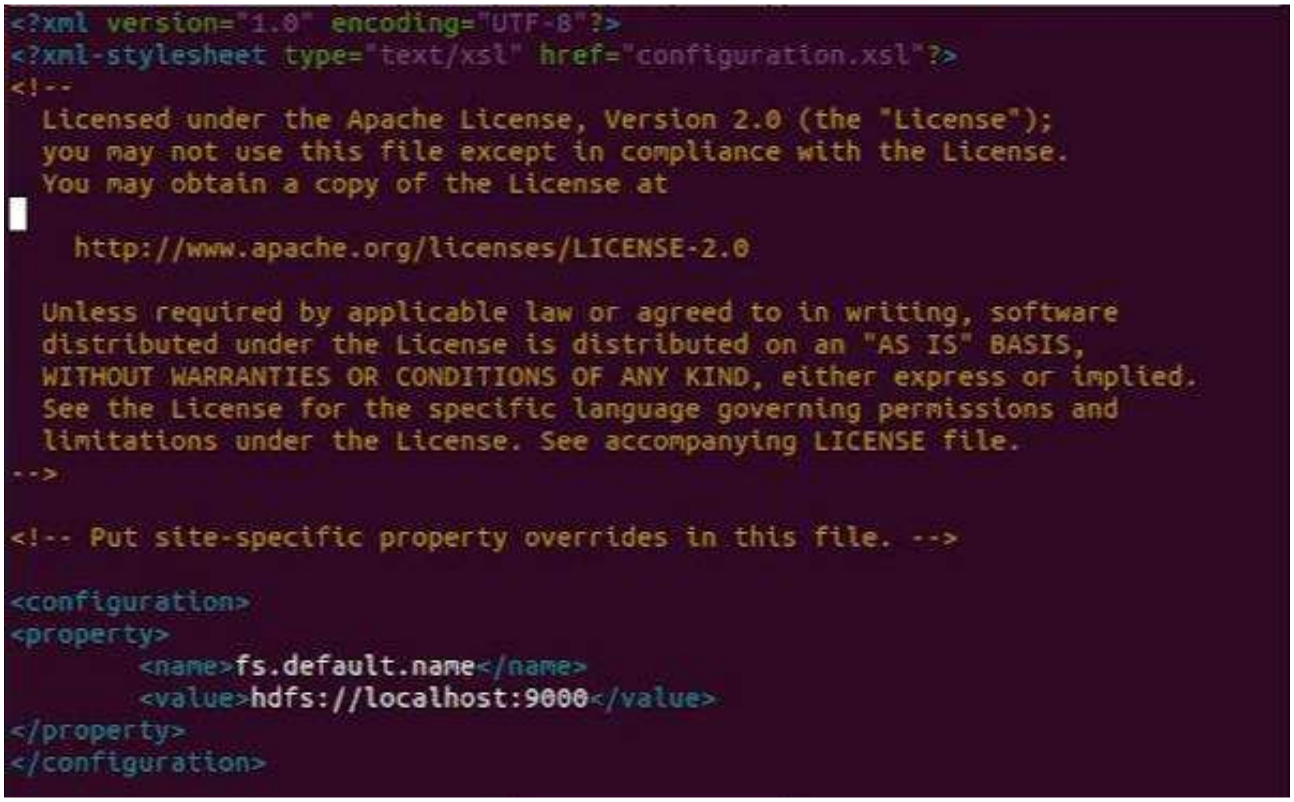
```
#JAVA PATH VARIABLES
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export PATH=$JAVA_HOME/bin:{$PATH}
export CLASSPATH=$JAVA_HOME/lib
#END OF JAVA PATH VARIABLES

#HADOOP VARIABLES START
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOPINSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
export HADOOP_CLASSPATH=$JAVA_HOME/lib/tools.jar
#HADOOP VARIABLES END
```

➔ Check Hadoop Version .



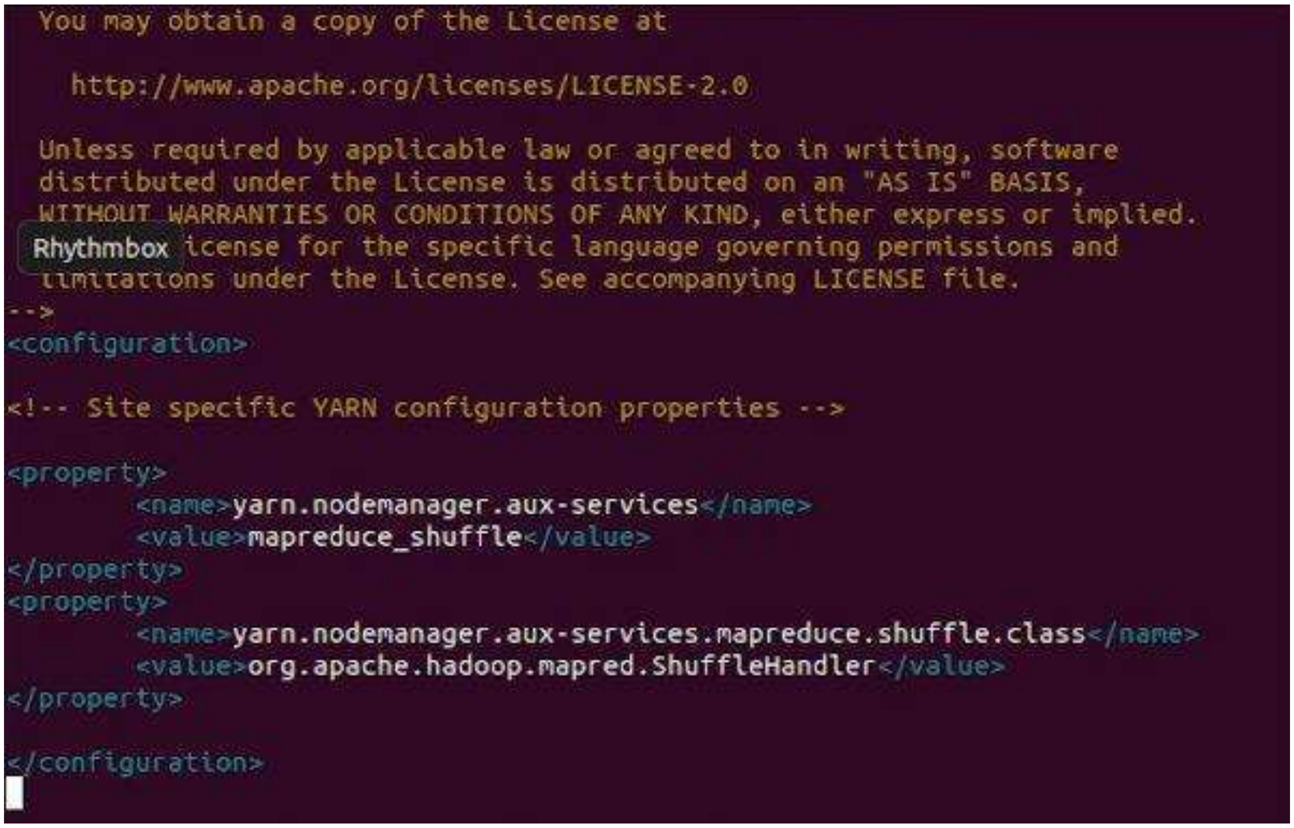
```
hadoop@HadoopVM:~/Downloads$ hadoop version
Hadoop 3.1.4
Source code repository https://github.com/apache/hadoop.git -r 1e877761e8dadd71
effef30e592368f7fe66a61b
Compiled by gabota on 2020-07-21T08:05Z
Compiled with protoc 2.5.0
From source with checksum 38405c63945c88fdf7a6fe391494799b
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-
3.1.4.jar
hadoop@HadoopVM:~/Downloads$
```

➔ Include properties to core-site.xml configuration .



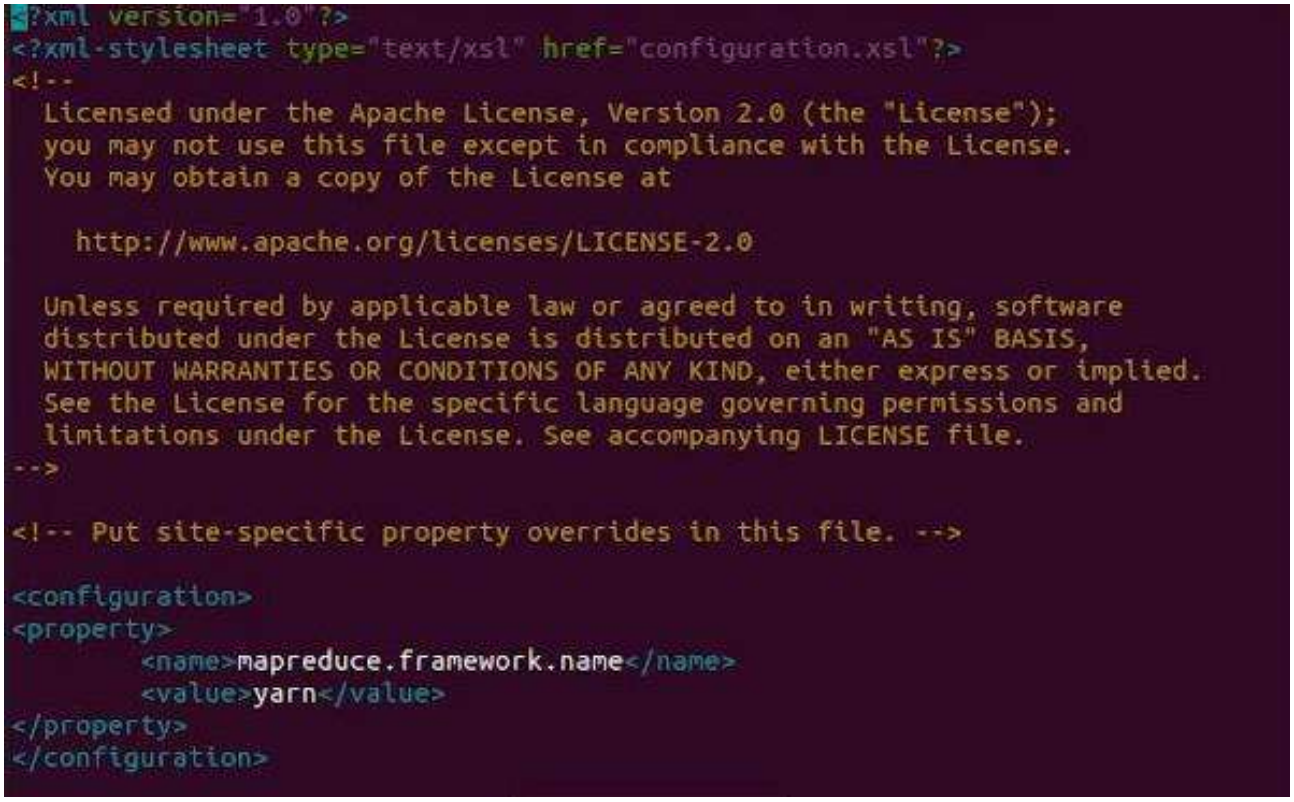
```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    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. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

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

➔ Include properties to yarn-site.xml configuration .



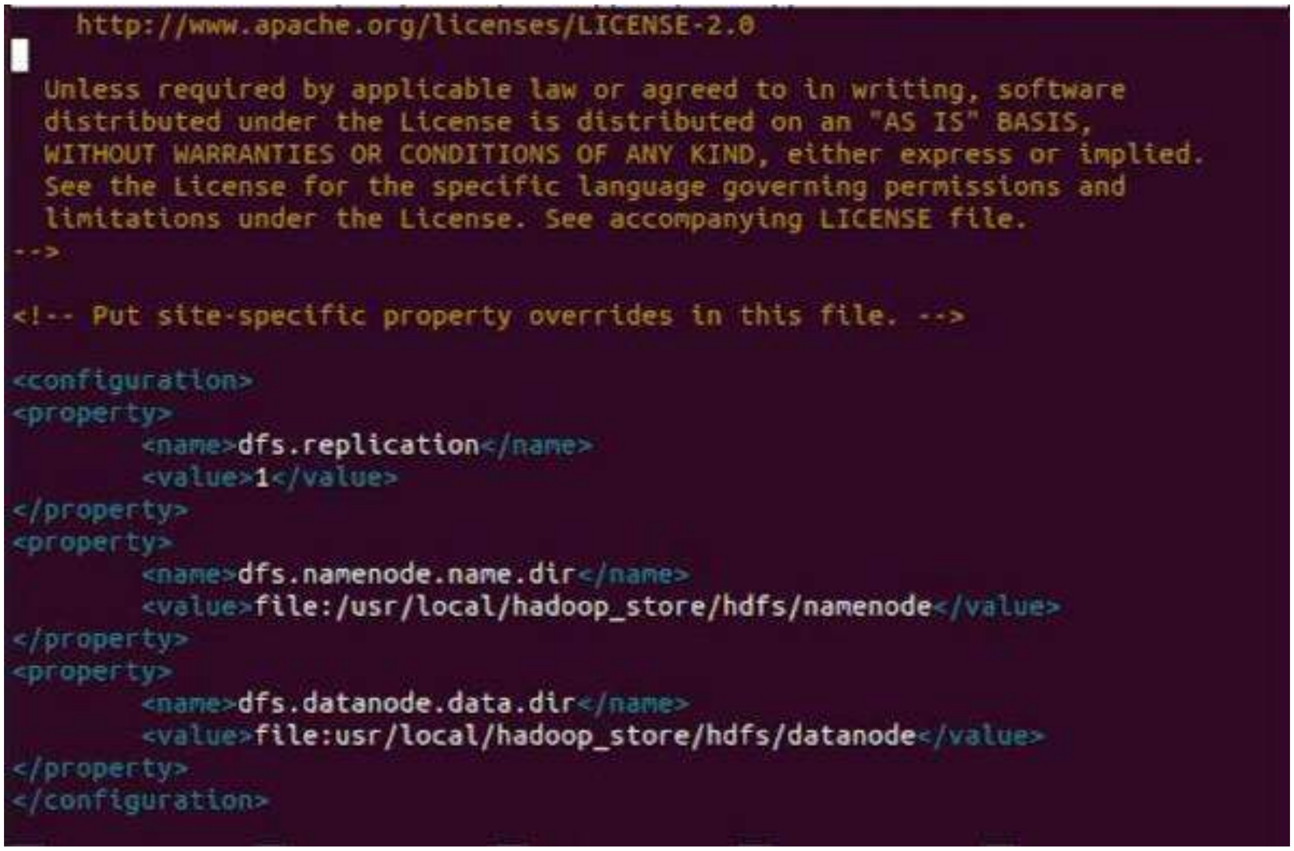
```
  You may obtain a copy of the License at

    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.
  Rhythmbox icense for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>

<!-- Site specific YARN configuration properties -->

<property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
</property>
<property>
        <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
        <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>

</configuration>
```

➔ Include properties to mapred-site.xml configuration .



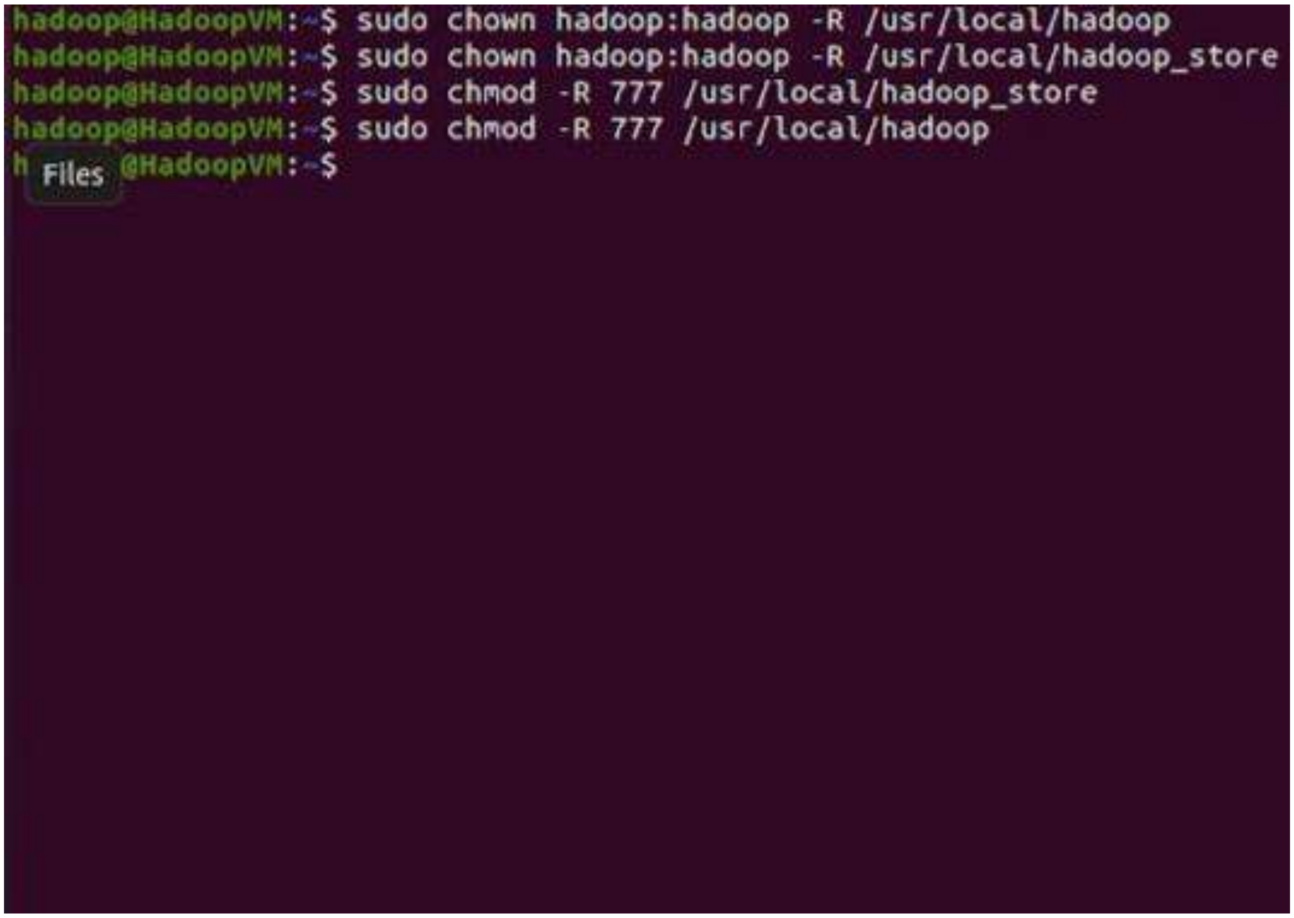
```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    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. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

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

➔ Include properties to hdfs-site.xml configuration .



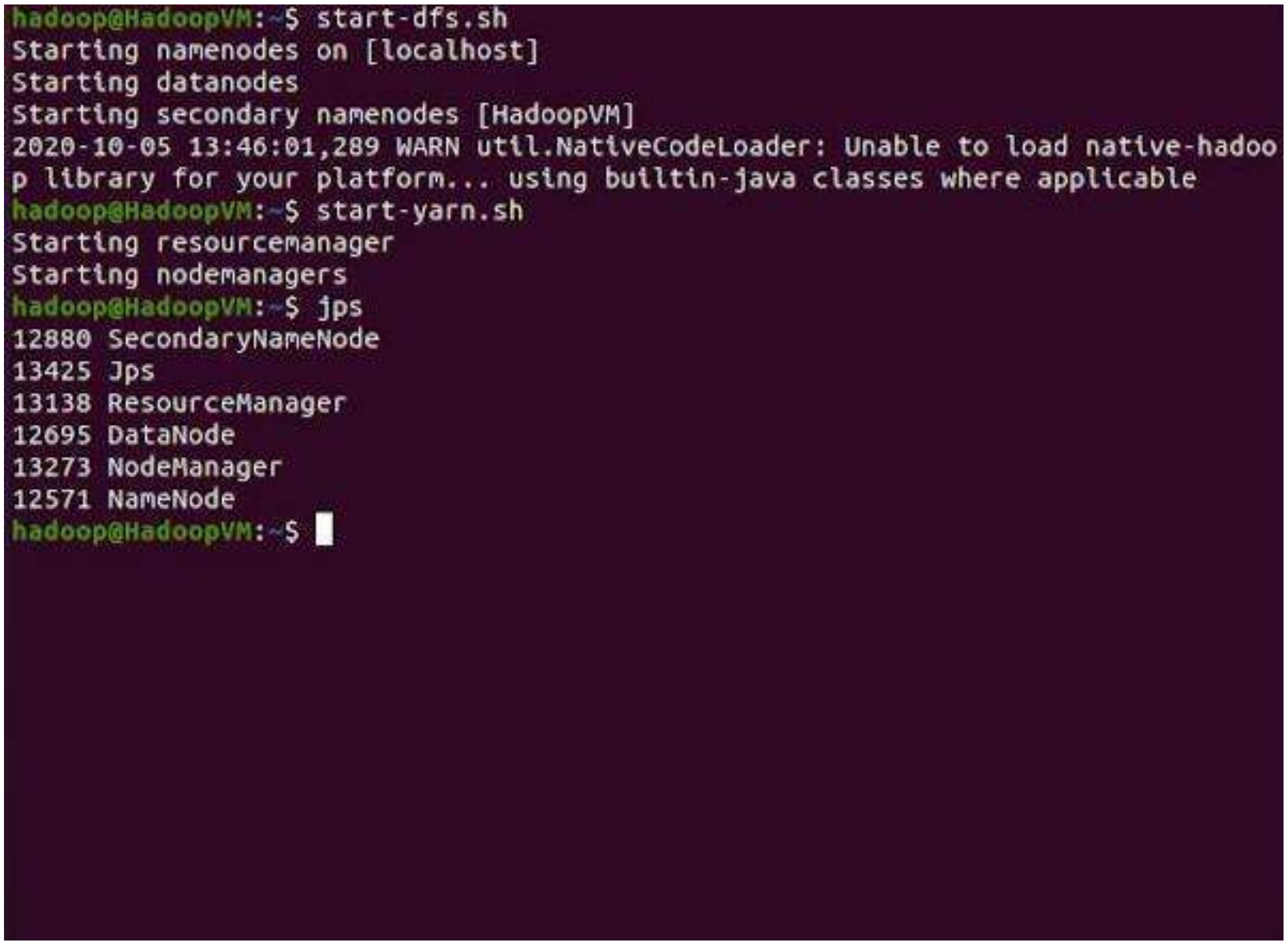
```
    http://www.apache.org/licenses/LICENSE-2.0

  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. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
        <name>dfs.replication</name>
        <value>1</value>
</property>
<property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/usr/local/hadoop_store/hdfs/namenode</value>
</property>
<property>
        <name>dfs.datanode.data.dir</name>
        <value>file:usr/local/hadoop_store/hdfs/datanode</value>
</property>
</configuration>
```

➔ Change Ownership and Mode for the hdfs directories .



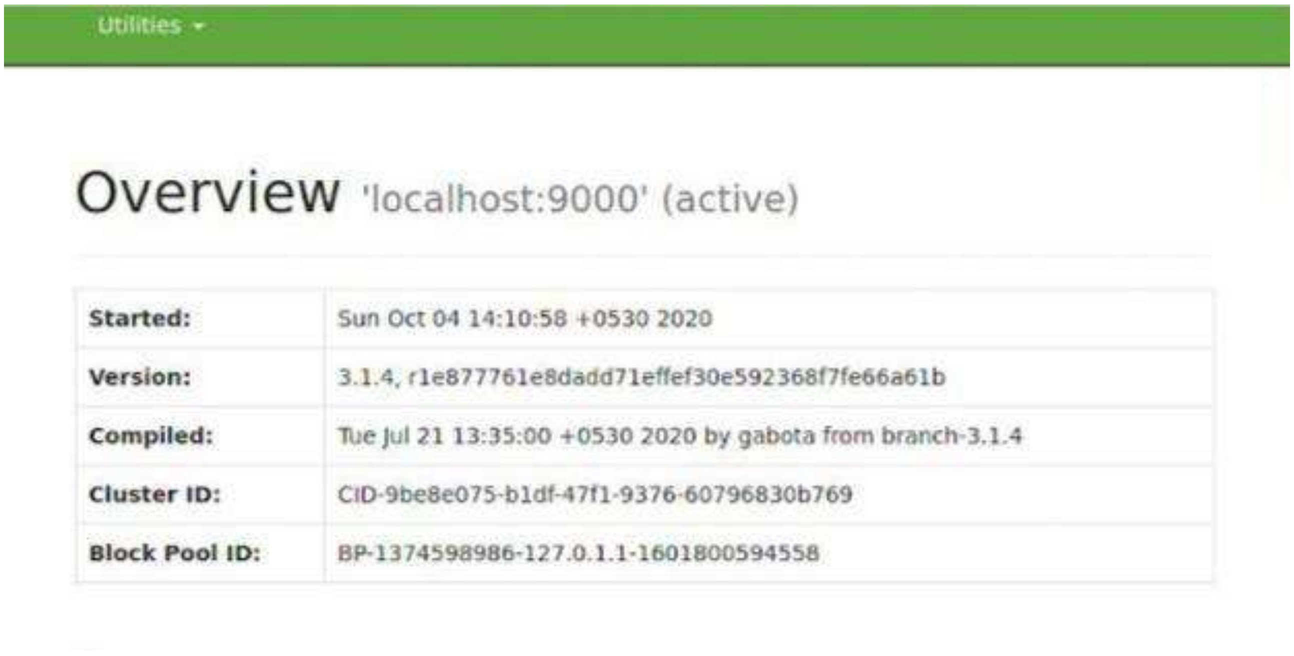
```
hadoop@HadoopVM:~$ sudo chown hadoop:hadoop -R /usr/local/hadoop
hadoop@HadoopVM:~$ sudo chown hadoop:hadoop -R /usr/local/hadoop_store
hadoop@HadoopVM:~$ sudo chmod -R 777 /usr/local/hadoop_store
hadoop@HadoopVM:~$ sudo chmod -R 777 /usr/local/hadoop
h Files @HadoopVM:~$
```

➔ Start the Hadoop services . Run jps to view the running process and id .



```
hadoop@HadoopVM:~$ start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [HadoopVM]
2020-10-05 13:46:01,289 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
hadoop@HadoopVM:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hadoop@HadoopVM:~$ jps
12880 SecondaryNameNode
13425 Jps
13138 ResourceManager
12695 DataNode
13273 NodeManager
12571 NameNode
hadoop@HadoopVM:~$
```

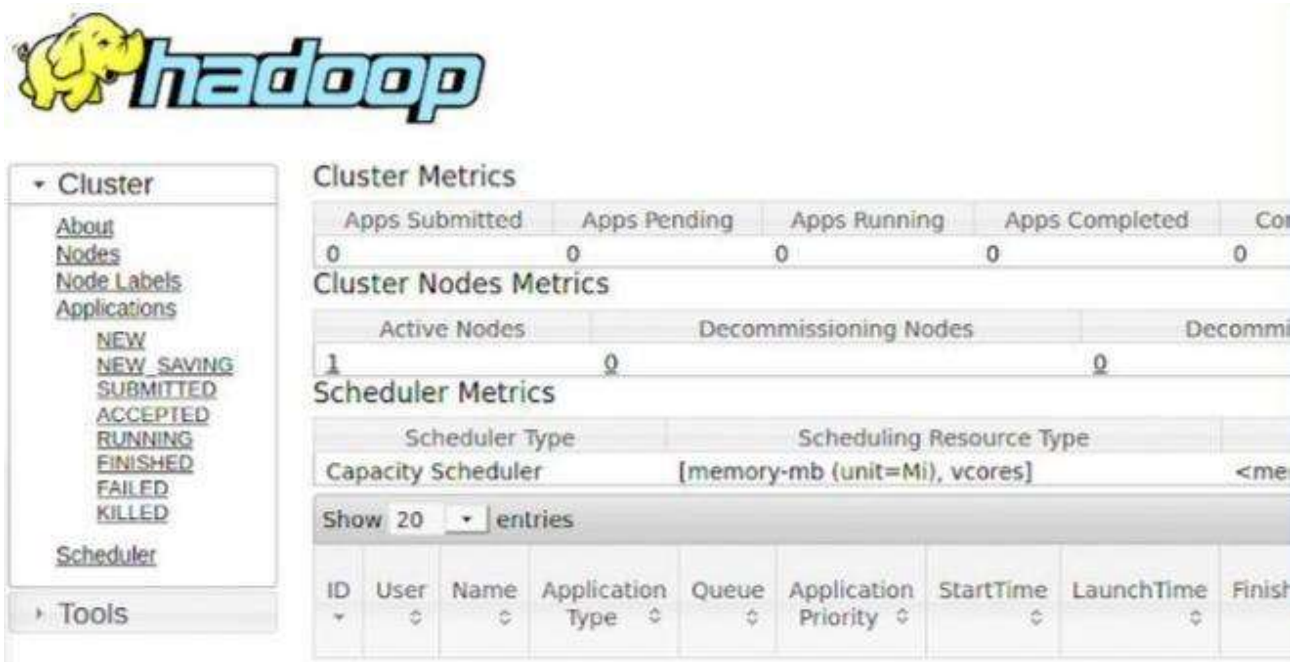
➔ Check the running services in Web Interface . ➢ Port 9870 :



Overview 'localhost:9000' (active)

Started:	Sun Oct 04 14:10:58 +0530 2020
Version:	3.1.4, r1e877761e8dadd71effef30e592368f7fe66a61b
Compiled:	Tue Jul 21 13:35:00 +0530 2020 by gabota from branch-3.1.4
Cluster ID:	CID-9be8e075-b1df-47f1-9376-60796830b769
Block Pool ID:	BP-1374598986-127.0.1.1-1601800594558

➢ Resource Manager :



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Co
0	0	0	0	0

Cluster Nodes Metrics

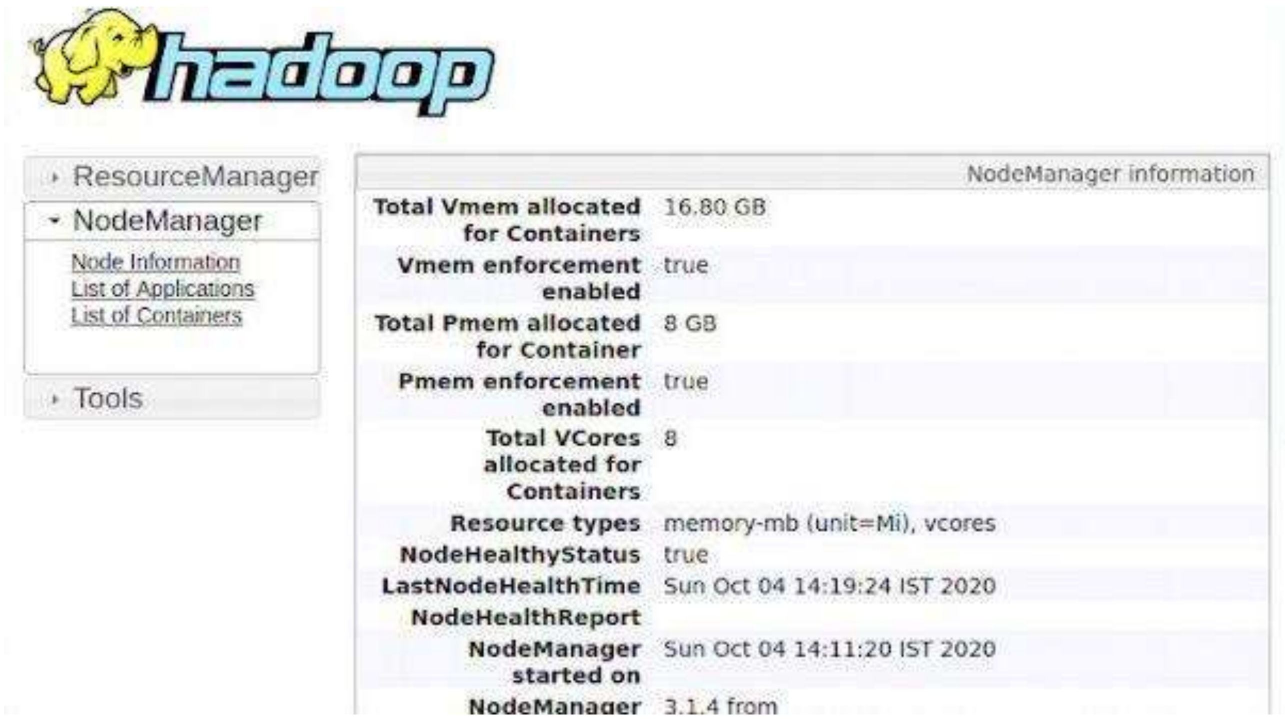
Active Nodes	Decommissioning Nodes	Decommi
1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<me

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	Finish

➢ Node Manager :



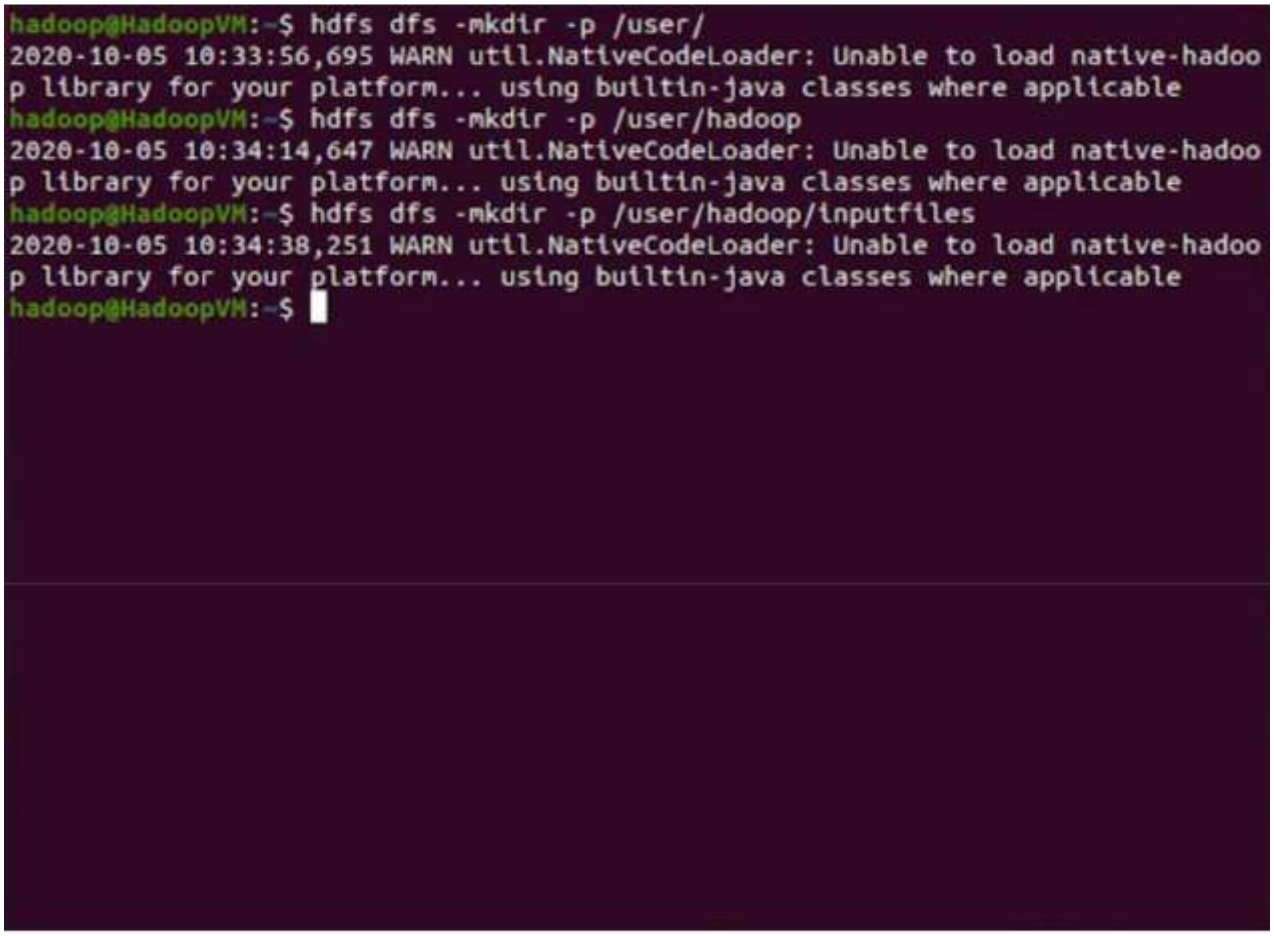
NodeManager information

Total Vmem allocated for Containers	16.80 GB
Vmem enforcement enabled	true
Total Pmem allocated for Container	8 GB
Pmem enforcement enabled	true
Total VCores allocated for Containers	8
Resource types	memory-mb (unit=Mi), vcores
NodeHealthyStatus	true
LastNodeHealthTime	Sun Oct 04 14:19:24 IST 2020
NodeHealthReport	
NodeManager started on	Sun Oct 04 14:11:20 IST 2020
NodeManager	3.1.4 from

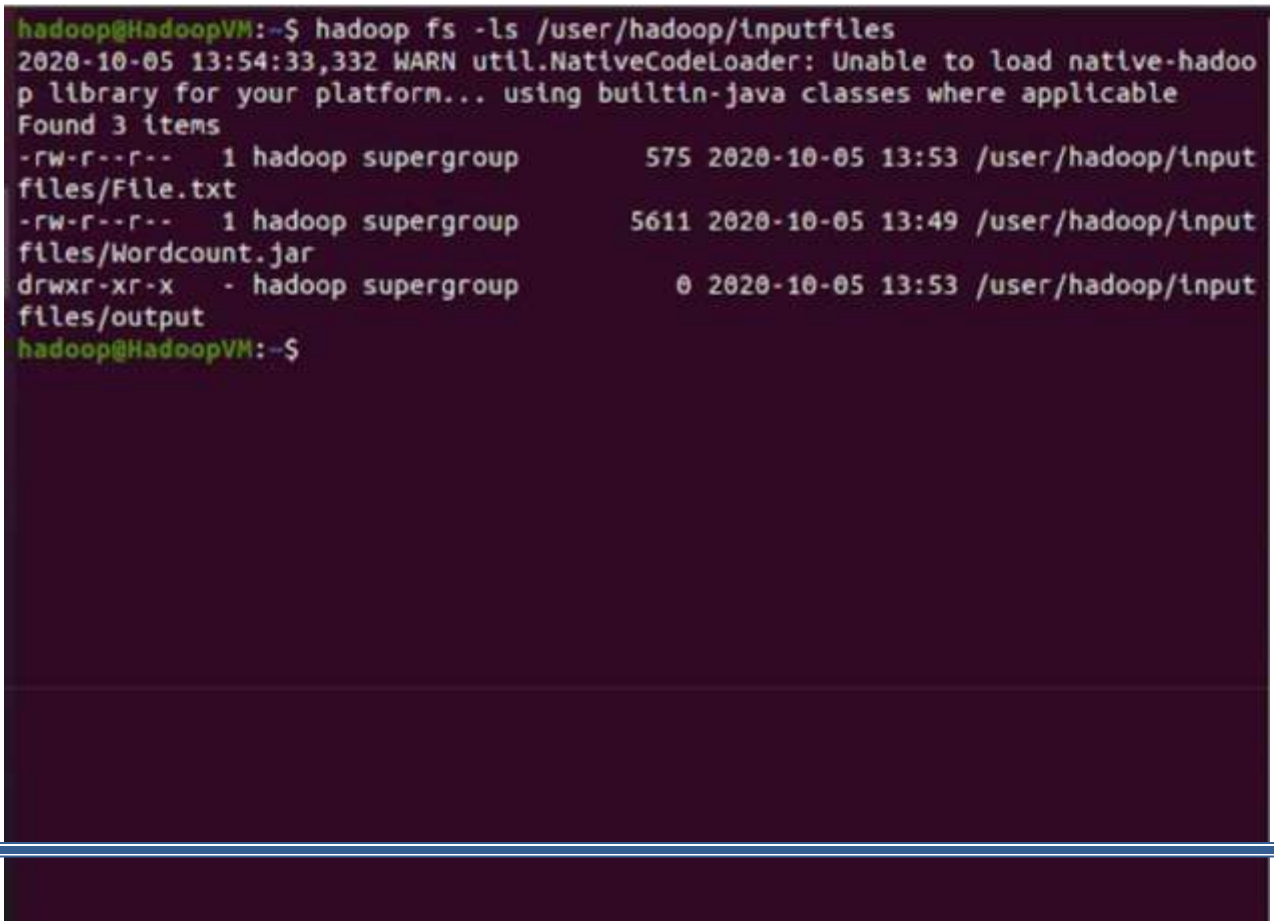
❖ Running Word Count program in Hadoop :

➔ Create Directories for Input , Output and Upload the Input and Text File .

Harshavarthini K
312217205035

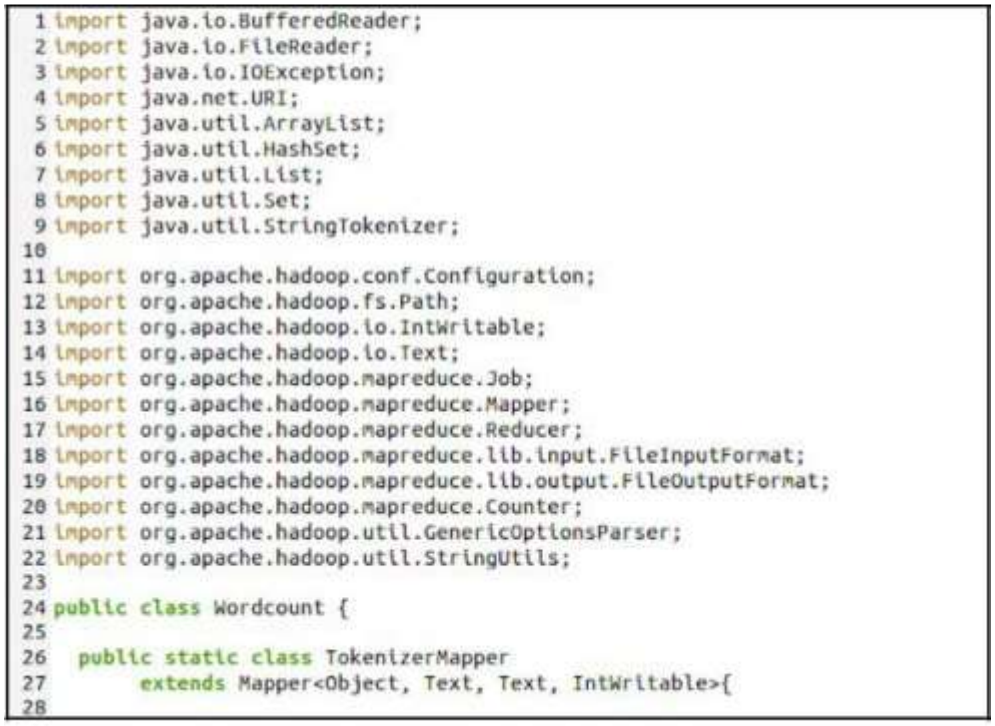


```
hadoop@HadoopVM:~$ hdfs dfs -mkdir -p /user/
2020-10-05 10:33:56,695 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
hadoop@HadoopVM:~$ hdfs dfs -mkdir -p /user/hadoop
2020-10-05 10:34:14,647 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
hadoop@HadoopVM:~$ hdfs dfs -mkdir -p /user/hadoop/inputfiles
2020-10-05 10:34:38,251 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
hadoop@HadoopVM:~$
```

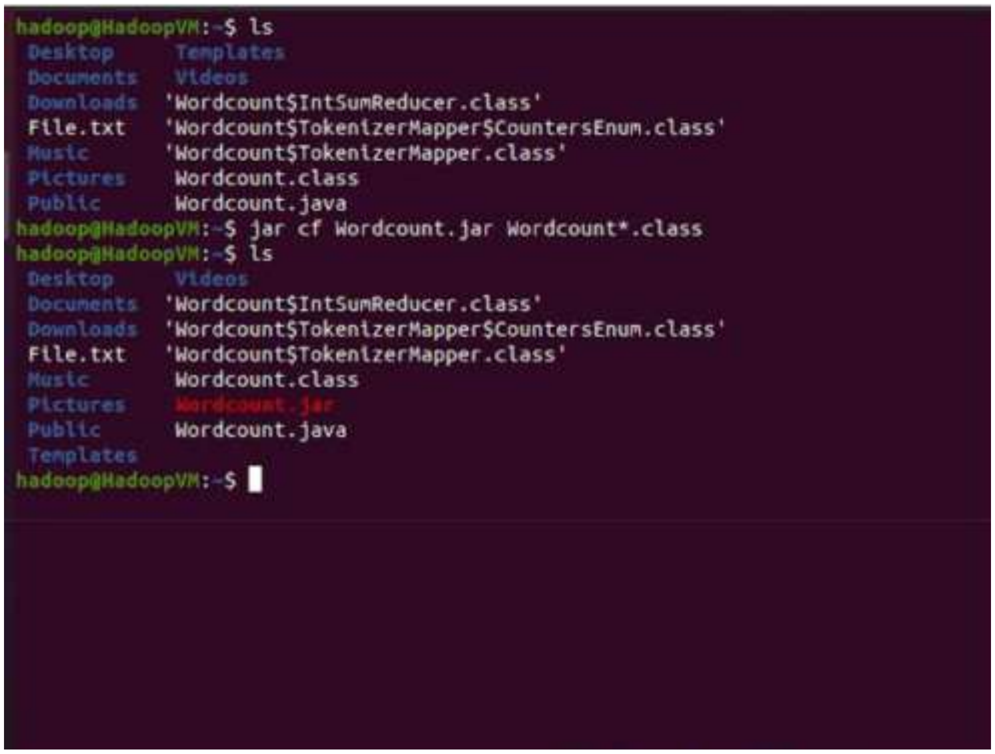


```
hadoop@HadoopVM:~$ hadoop fs -ls /user/hadoop/inputfiles
2020-10-05 13:54:33,332 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
Found 3 items
-rw-r--r--   1 hadoop supergroup        575 2020-10-05 13:53 /user/hadoop/input
files/File.txt
-rw-r--r--   1 hadoop supergroup       5611 2020-10-05 13:49 /user/hadoop/input
files/Wordcount.jar
drwxr-xr-x   - hadoop supergroup          0 2020-10-05 13:53 /user/hadoop/input
files/output
hadoop@HadoopVM:~$
```

➔ Create a Word Count program and compile it to create a jar file .



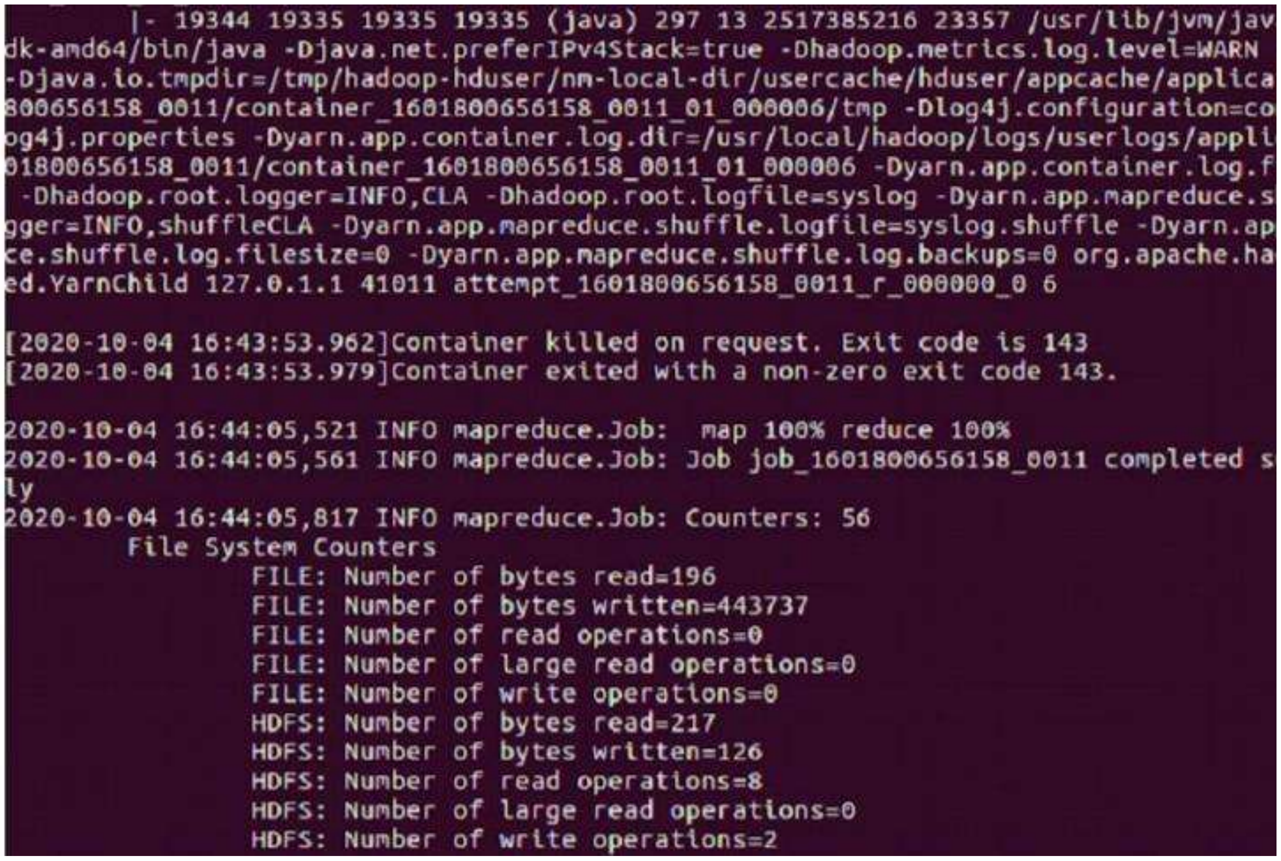
```java
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.net.URI;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.Counter;
import org.apache.hadoop.util.GenericOptionsParser;
import org.apache.hadoop.util.StringUtils;

public class Wordcount {

  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

```



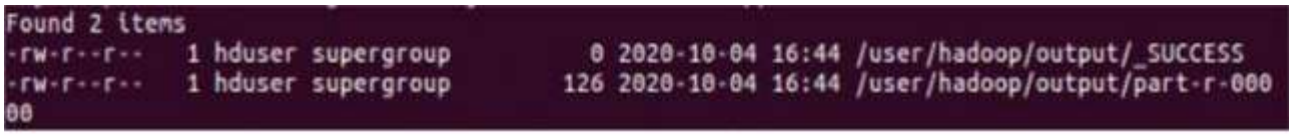
```
hadoop@HadoopVM:~$ ls
 Desktop      Templates
 Documents    Videos
 Downloads   'Wordcount$IntSumReducer.class'
 File.txt    'Wordcount$TokenizerMapper$CountersEnum.class'
 Music       'Wordcount$TokenizerMapper.class'
 Pictures     Wordcount.class
 Public       Wordcount.java
hadoop@HadoopVM:~$ jar cf Wordcount.jar Wordcount*.class
hadoop@HadoopVM:~$ ls
 Desktop      Videos
 Documents   'Wordcount$IntSumReducer.class'
 Downloads   'Wordcount$TokenizerMapper$CountersEnum.class'
 File.txt    'Wordcount$TokenizerMapper.class'
 Music        Wordcount.class
 Pictures     Wordcount.jar
 Public       Wordcount.java
 Templates
hadoop@HadoopVM:~$ 
```

➔ Run the jar using hadoop command hadoop jar <jar> <classfile> <inputfilepath> <outputfilepath> .

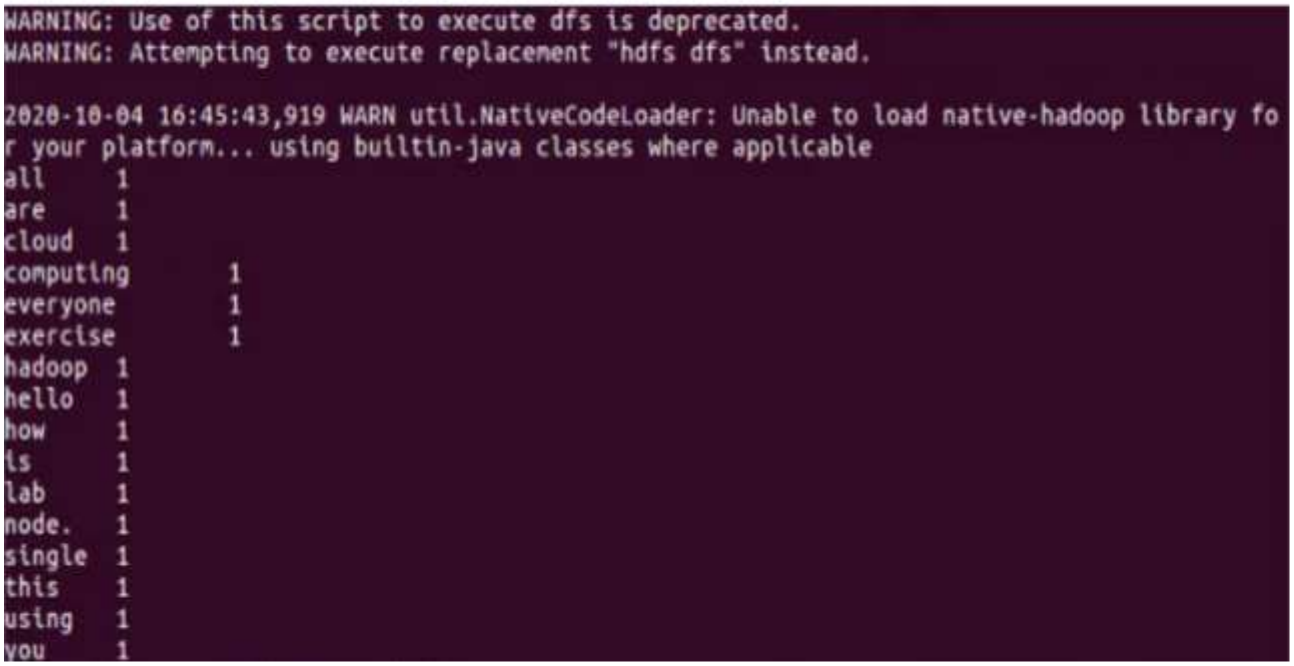
➔ View the contents of the output file , it should contain words and their frequency in the Text File .



```
        |- 19344 19335 19335 19335 (java) 297 13 2517385216 23357 /usr/lib/jvm/jav
dk-amd64/bin/java -Djava.net.preferIPv4Stack=true -Dhadoop.metrics.log.level=WARN
-Djava.io.tmpdir=/tmp/hadoop-hduser/nm-local-dir/usercache/hduser/appcache/applica
800656158_0011/container_1601800656158_0011_01_000006/tmp -Dlog4j.configuration=co
og4j.properties -Dyarn.app.container.log.dir=/usr/local/hadoop/logs/userlogs/appli
01800656158_0011/container_1601800656158_0011_01_000006 -Dyarn.app.container.log.f
 -Dhadoop.root.logger=INFO,CLA -Dhadoop.root.logfile=syslog -Dyarn.app.mapreduce.s
gger=INFO,shuffleCLA -Dyarn.app.mapreduce.shuffle.logfile=syslog.shuffle -Dyarn.ap
ce.shuffle.log.filesize=0 -Dyarn.app.mapreduce.shuffle.log.backups=0 org.apache.ha
ed.YarnChild 127.0.1.1 41011 attempt_1601800656158_0011_r_000000_0 6

[2020-10-04 16:43:53.962]Container killed on request. Exit code is 143
[2020-10-04 16:43:53.979]Container exited with a non-zero exit code 143.

2020-10-04 16:44:05,521 INFO mapreduce.Job:  map 100% reduce 100%
2020-10-04 16:44:05,561 INFO mapreduce.Job: Job job_1601800656158_0011 completed s
ly
2020-10-04 16:44:05,817 INFO mapreduce.Job: Counters: 56
        File System Counters
                FILE: Number of bytes read=196
                FILE: Number of bytes written=443737
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=217
                HDFS: Number of bytes written=126
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
```

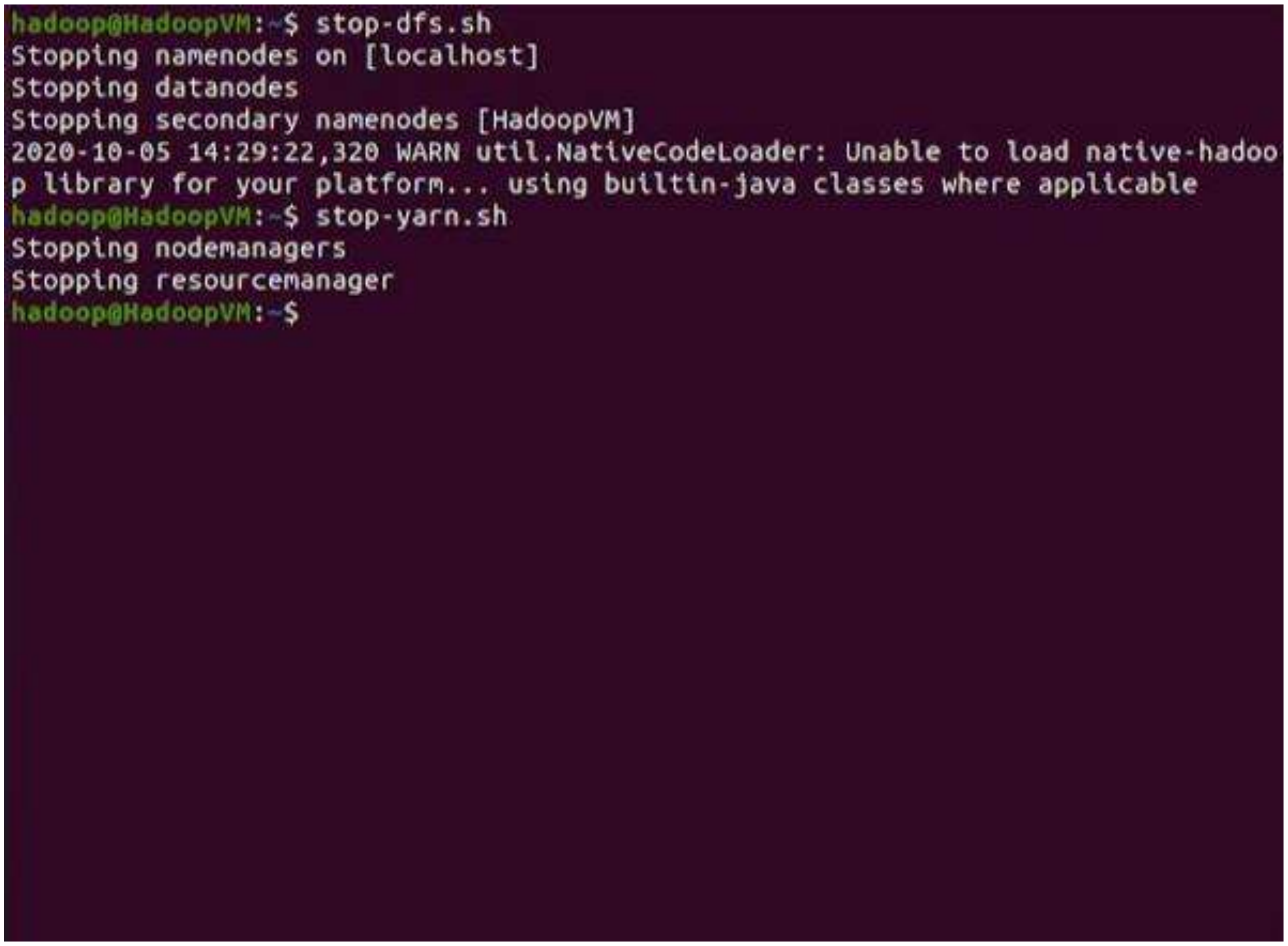


```
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2020-10-04 16:44 /user/hadoop/output/_SUCCESS
-rw-r--r--   1 hduser supergroup        126 2020-10-04 16:44 /user/hadoop/output/part-r-000
00
```



```
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

2020-10-04 16:45:43,919 WARN util.NativeCodeLoader: Unable to load native-hadoop library fo
r your platform... using builtin-java classes where applicable
all     1
are     1
cloud   1
computing       1
everyone        1
exercise        1
hadoop  1
hello   1
how     1
is      1
lab     1
node.   1
single  1
this    1
using   1
you     1
```

➔ Stop all Hadoop services .



```
hadoop@HadoopVM:~$ stop-dfs.sh
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [HadoopVM]
2020-10-05 14:29:22,320 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
hadoop@HadoopVM:~$ stop-yarn.sh
Stopping nodemanagers
Stopping resourcemanager
hadoop@HadoopVM:~$
```

Result:

Thus the Single node Hadoop platform has been installed and Word count program has been implemented successfully.