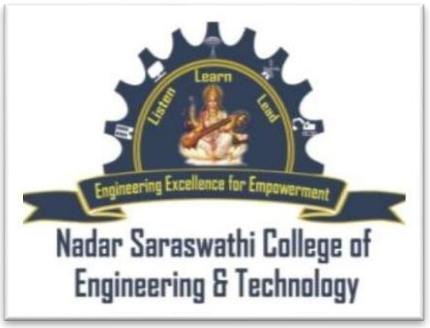
NADAR SARASWATHI COLLEGE OF ENGINEERING AND TECHNOLOGY

VADAPUTHUPATTI, THENI



DEPARTMENT OF IT , AI&DS

LAB MANUAL

CCS335/CLOUD COMPUTING

(Regulation 2021)

THENI MELAPETTAI HINDU NADARGAL URAVINMURAI

NADAR SARASWATHI COLLEGE OF ENGINEERING & TECHNOLOGY

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai
ISO 9001 : 2015 Certified Institution
Vadapudupatti, Annanji (po), Theni - 625 531,
Tamilnadu, India.

Department of IT , AI&DS

LAB MANUAL

SUBJECT CODE	:	CCS335
SUBJECT NAME	:	CLOUD COMPUTING
YEAR	:	III
DEPARTMENT	:	CSE
SEMESTER	:	V
REGULATION	:	2021
PREPARED BY	:	S Abirami Kayathiri ASSISTANT PROFESSOR/AI&DS

DEPARTMENT OF IT , AI&DS

COLLEGE VISION

To establish ourselves as a leading technological institution.

COLLEGE MISSION

1. To provide professional, constructive and learner centered education.
2. To make learners contribute to the development of the nation through academic and industrial excellence.
3. To encourage learners, involve in innovative researches with ethics.
4. To produce competitive and confident graduates to face the ever-growing challenges of the labour market.

DEPARTMENT OF IT , AI&DS

VISION

To become a leading hub in the field of Computer Engineering.

MISSION

1. To provide a strong theoretical and practical knowledge emphasizing on software developments.
2. To encourage autonomous learning, foster interactions and establish partnership with renowned software industries.
3. To inculcate soft skills, leadership qualities and innovative research skills with ethical values.

NADAR SARASWATHI COLLEGE OF ENGINEERING & TECHNOLOGY

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai
ISO 9001 : 2015 Certified Institution
Vadapudupatti, Annanji (po), Theni - 625 531,
Tamilnadu, India.

DEPARTMENT OF IT , AI&DS

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

I. Apply their technical competence in computer science to solve real world problems, with technical and people leadership.

II. Conduct cutting edge research and develop solutions on problems of social relevance.

III. Work in a business environment, exhibiting team skills, work ethics, adaptability and lifelong learning.

PROGRAMME OUTCOMES (POs):

PO	Graduate Attribute	Programme Outcome
1.	Engineering knowledge	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2.	Problem analysis	Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3.	Design/development of solutions	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4.	Conduct investigations of complex problems	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5.	Modern tool usage	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6.	The Engineer and society	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional

		engineering practice.
7.	Environment and sustainability	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8.	Ethics	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9.	Individual and team work	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10.	Communication	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions
11.	Project management and finance	Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments
12.	Life-long learning	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

On completion of Computer Science and Engineering program, the student will have the following Program Specific Outcomes.

1. Exhibit design and programming skills to build and automate business solutions using cutting edge technologies
2. Strong theoretical foundation leading to excellence and excitement towards research, to provide elegant solutions to complex problems
3. Ability to work effectively with various engineering fields as a team to design, build and develop system applications.

THENI MELAPETTAI HINDU NADARGAL URAVINMURAI
NADAR SARASWATHI COLLEGE OF ENGINEERING & TECHNOLOGY
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai
ISO 9001 : 2015 Certified Institution
Vadapudupatti, Annanji (po), Theni - 625 531,
Tamilnadu, India.

DEPARTMENT OF IT , AI&DS

CCS335 CLOUD COMPUTING **L T P C**

2 0 2 3

SYLLABUS

Sl. No	LIST OF EXPERIMENTS
1.	Install VirtualBox/VMware/ Equivalent open-source cloud Workstation with different flavours of Linux or Windows OS on top of windows 8 and above
2.	Install a C compiler in the virtual machine created using a virtual box and execute Simple Programs
3.	Install Google App Engine. Create a hello world app and other simple web applications using python/java.
4.	Use the GAE launcher to launch the web applications.
5.	Simulate a cloud scenario using Cloud Sim and run a scheduling algorithm that is not present in Cloud Sim.
6.	Find a procedure to transfer the files from one virtual machine to another virtual machine.
7.	Install Hadoop single node cluster and run simple applications like wordcount.
8.	Creating and Executing Your First Container Using Docker.
9.	Run a Container from Docker Hub

TOTAL: 30 PERIODS

NADAR SARASWATHI COLLEGE OF ENGINEERING & TECHNOLOGY

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai
ISO 9001 : 2015 Certified Institution
Vadapudupatti, Annanji (po), Theni - 625 531,
Tamilnadu, India.

DEPARTMENT OF IT , AI&DS

CLOUD COMPUTING

1. Definition: Cloud computing is a paradigm shift in computing technology that enables users to access and utilize computing resources, such as servers, storage, databases, networking, and software applications, over the internet, on a pay-as-you-go basis.

2. On-demand Service: Cloud computing provides on-demand access to a shared pool of configurable computing resources, allowing users to rapidly provision and release resources as needed without the hassle of managing physical infrastructure.

3. Scalability: Cloud computing offers scalability, allowing users to easily scale resources up or down based on demand. This elasticity enables organizations to handle fluctuating workloads efficiently without over-provisioning or under-provisioning resources.

4. Resource Pooling: Cloud computing pools computing resources across multiple users and organizations, enabling efficient resource utilization and maximizing the economies of scale. Users typically access resources via web-based interfaces or APIs.

5. Service Models: Cloud computing offers different service models, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), catering to different levels of abstraction and management responsibilities.

6. IaaS: Infrastructure as a Service provides users with virtualized computing resources, such as virtual machines, storage, and networking, allowing them to deploy and manage their own applications and operating systems.

7. PaaS: Platform as a Service abstracts away the underlying infrastructure and provides a platform for developing, deploying, and managing applications. PaaS offerings typically include development tools, middleware, and runtime environments.

8. SaaS: Software as a Service delivers software applications over the internet on a subscription basis, eliminating the need for users to install, maintain, and update software locally. Common examples include email services, office productivity suites, and customer relationship management (CRM) software.

9. Cost-effectiveness: Cloud computing offers cost-effectiveness by shifting capital expenses (CapEx) to operational expenses (OpEx), as users only pay for the resources they consume on a usage-based billing model.

10. Flexibility and Agility: Cloud computing enables organizations to innovate and respond to changing business requirements quickly by providing flexible, agile, and scalable IT resources on-demand.

11. Global Reach: Cloud computing providers operate data centers worldwide, allowing users to deploy applications and services globally with low latency and high availability.

12. Security and Compliance: Cloud computing providers implement robust security measures and compliance standards to protect data privacy, confidentiality, and integrity. However, users must also ensure proper data governance and security practices.

DEPARTMENT OF IT , AI&DS

CCS335- CLOUD COMPUTING

COURSE OBJECTIVES

OB1: To understand the principles of cloud architecture, models and infrastructure.
OB2: To understand the concepts of virtualization and virtual machines.
OB3: To gain knowledge about virtualization Infrastructure.
OB4: To explore and experiment with various Cloud deployment environments.
OB5: To learn about the security issues in the cloud environment.

.

COURSE OUTCOME

COs	Outcomes	Bloom's Taxonomy	BTS
CO1	Understand the design challenges in the cloud.	**Understand**	**2**
CO2	Apply the concept of virtualization and its types.	**Apply**	**3**
CO3	Experiment with virtualization of hardware resources and Docker.	**Apply**	**3**
CO4	Develop and deploy services on the cloud and set up a cloud environment.	**Apply, Evaluate**	**3,5**
CO5	Explain security challenges in the cloud environment.	**Creating**	**6**

PO - CO Correlation Matrix

CO-PO,PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	1	1	-	-	-	2	3	1	3	2	1	3
CO2	3	1	2	2	1	-	-	-	1	2	1	3	2	2	1
CO3	2	3	2	3	1	-	-	-	3	1	1	3	1	1	1
CO4	1	2	3	3	3	-	-	-	3	3	1	2	1	3	3
CO5	2	3	3	1	3	-	-	-	2	2	1	2	2	2	3
Avg	2.2	2.2	2.2	2.0	1.8	-	-	-	2.2	2.2	1	2.6	1.6	1.8	2.2

THENI MELAPETTAI HINDU NADARGAL URAVINMURAI
NADAR SARASWATHI COLLEGE OF ENGINEERING & TECHNOLOGY
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai
ISO 9001 : 2015 Certified Institution
Vadapudupatti, Annanji (po), Theni - 625 531,
Tamilnadu, India.

DEPARTMENT OF IT , AI&DS

CO – PO MAPPING WITH EXPERIMENTS

S.NO	TITLE OF EXPERIMENT	CO	PO
1.	Install Virtualbox/VMware/ Equivalent open source cloud Workstation with different flavours of Linux or Windows OS on top of windows 8 and above	CO1	1,2,3,4,5,9 ,10,11,12
2.	Install a C compiler in the virtual machine created using a virtual box and execute Simple Programs	CO1	1,2,3,4,5,9 ,10,11,12
3.	Install Google App Engine. Create a hello world app and other simple web applications using python/java.	CO1	1,2,3,4,5,9 ,10,11,12
4.	Use the GAE launcher to launch the web applications.	CO2	1,2,3,4,5,9 ,10,11,12
5.	Simulate a cloud scenario using Cloud Sim and run a scheduling algorithm that is not present in Cloud Sim.	CO3	1,2,3,4,5,9 ,10,11,12
6.	Find a procedure to transfer the files from one virtual machine to another virtual machine.	CO1	1,2,3,4,5,9 ,10,11,12
7.	Install Hadoop single node cluster and run simple applications like wordcount.	CO1	1,2,3,4,5,9 ,10,11,12
8.	Creating and Executing Your First Container Using Docker.	CO4	1,2,3,4,5,9 ,10,11,12
ADVANCED EXPERIMENTS			
9.	Run a Container from Docker Hub	CO3	1,2,3,4,5,9 ,10,11,12

LIST OF HARDWARE/COMPONENTS

Standalone desktops for all students

THENI MELAPETTAI HINDU NADARGAL URAVINMURAI

NADAR SARASWATHI COLLEGE OF ENGINEERING & TECHNOLOGY

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai
ISO 9001 : 2015 Certified Institution
Vadapudupatti, Annanji (po), Theni - 625 531,
Tamilnadu, India.

DEPARTMENT OF IT , AI&DS

INDEX

S.NO	EXPERIMENT	MARKS	SIGN
	Install VirtualBox/VMware/ Equivalent open-source cloud		
1.	Install VirtualBox/VMware/ Equivalent open-source cloud Workstation with different flavours of Linux or Windows OS on top of windows 8 and above		
2.	Install a C compiler in the virtual machine created using a virtual box and execute Simple Programs		
3.	Install Google App Engine. Create a hello world app and other simple web applications using python/java.		
4.	Use the GAE launcher to launch the web applications.		
5.	Simulate a cloud scenario using Cloud Sim and run a scheduling algorithm that is not present in Cloud Sim.		
6.	Find a procedure to transfer the files from one virtual machine to another virtual machine.		
7.	Install Hadoop single node cluster and run simple applications like wordcount.		
8.	Creating and Executing Your First Container Using Docker.		
9.	Run a Container from Docker Hub		

CCS335 - CLOUD COMPUTING LABORATORY

INDEX

EX.N0	NAME OF THE EXPERIMENT	PAGE NO	SIGNATURE
1	INSTALL VIRTUAL BOX/VMWARE WORKSTATION		
2	INSTALL A C COMPILER IN THE VIRTUAL MACHINE CREATED USING VIRTUAL BOX AND EXECUTE A SIMPLEPROGRAMS		
3	INSTALL GOOGLE APP ENGINE .CREATE HELLO WORLD APP AND OTHER SIMPLE WEB APPLICATIONS USING PYTHON/JAVA.		
4	USE GAE LAUNCHER TO LAUNCH THEWEB APPLICATION		
5	SIMULATE A CLOUD SCENARIO USINGCLOUDSIM AND RUN A SCHEDULING ALGORITHM THAT IS NOT PRESENT INCLOUDSIM		
6	FIND A PROCEDURE TO TRANSFER FILES FROM ONE VIRTUAL MACHINE TO ANOTHER VITUAL MACHINE		
7	INSTALL HADOOP SINGLE NODE CLUSTERAND RUN A SIMPLE APPLICATION LIKE WORDCOUNT		

8	CREATING AND EXECUTING YOUR FIRSTCONTAINER USING DOCKER.		
9	RUN A CONTAINER FROM DOCKER HUB		

EXP NO :1 DATE:	INSTALL VIRTUALBOX/VMware WORKSTATION

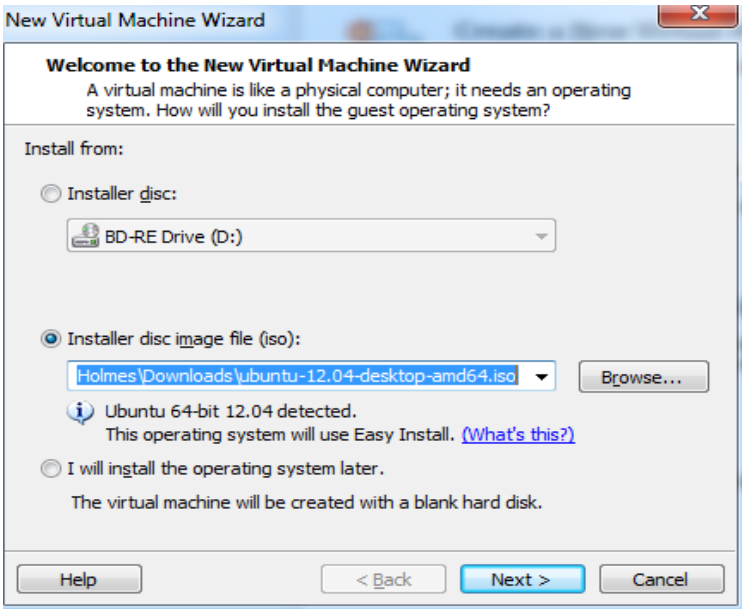
AIM:

Find procedure to Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.

PROCEDURE:

1. Download the Ubuntu iso (desktop not server) and the free VMware Player.
2. Install VMware Player and run it, you'll see something like this:

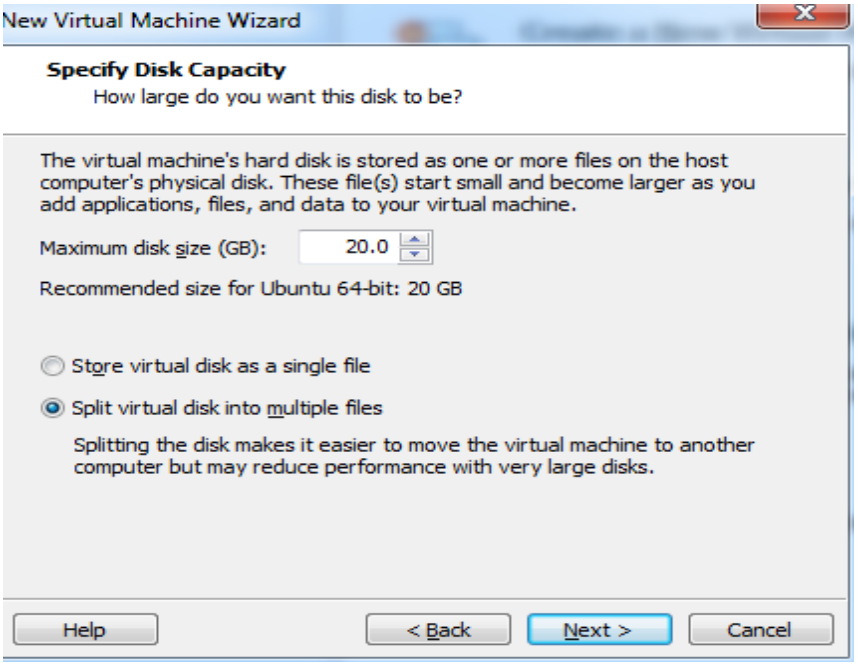
3. Select "Create a New Virtual Machine"
4. Select "Installer disc image file" and browse to the Ubuntu iso you downloaded.



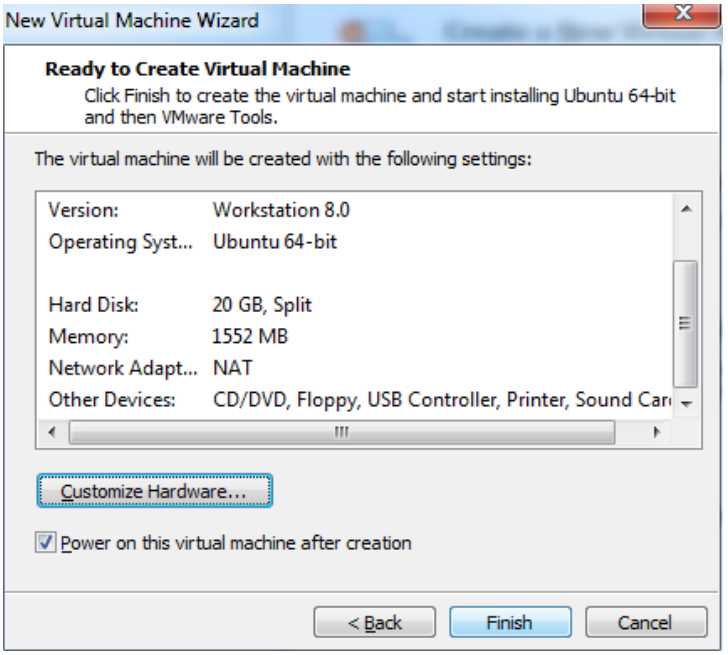
You should see that it will use Easy Install – this takes care of most of the hard work for you. Click next

5. Enter your full name, username and password and hit next

6. Select the maximum disk size and type. Unless you're planning on some really CPU intensive work inside the VM, select the "Split virtual disk into multiple files" option. Hit next when you're happy with the settings.

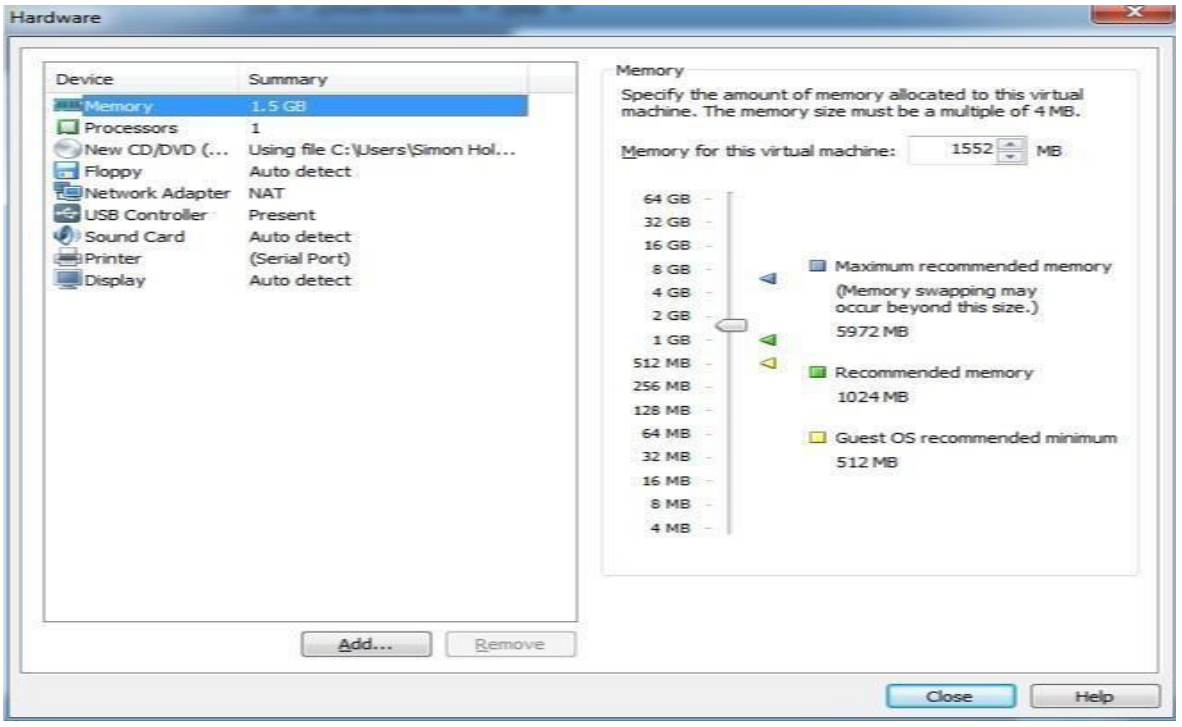


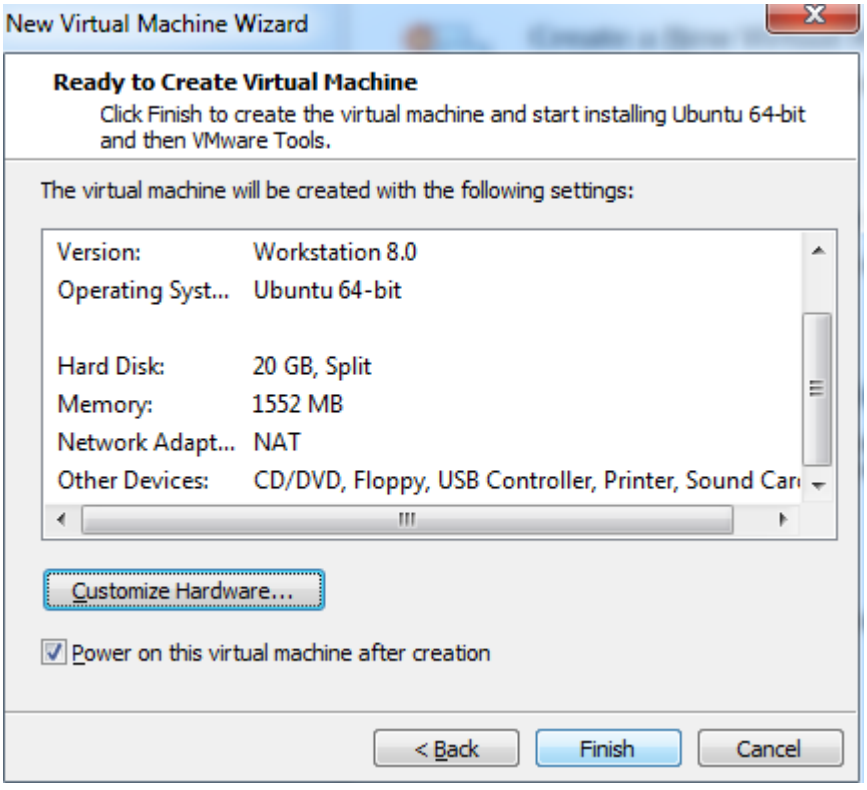
7. This brings you to the confirmation page. Click "Customize Hardware"



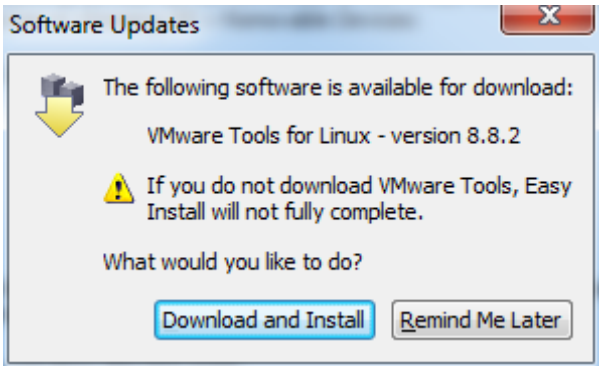
8. In the hardware options section select the amount of memory you want the VM to use. In this instance I've gone for 1.5GB out of the 8GB installed in my laptop. Leave everything else as it is and click Close.

9. This brings you back to the confirmation page. Click Finish this time



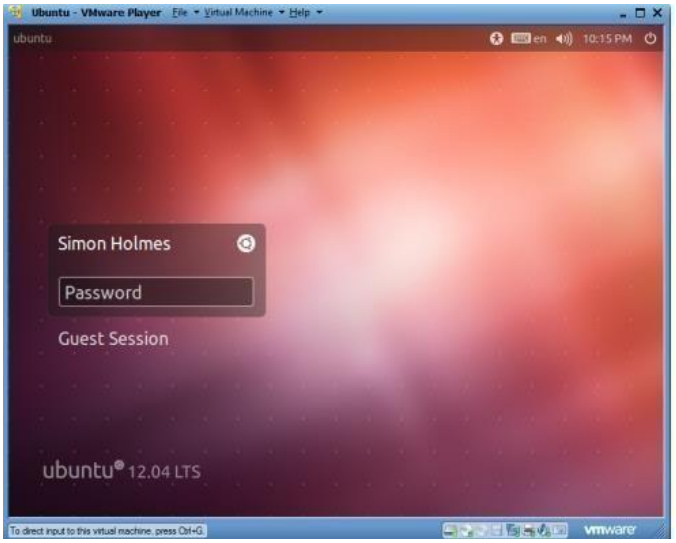


10. You will probably be prompted to download VMware Tools for Linux. Click “Download and Install” to continue

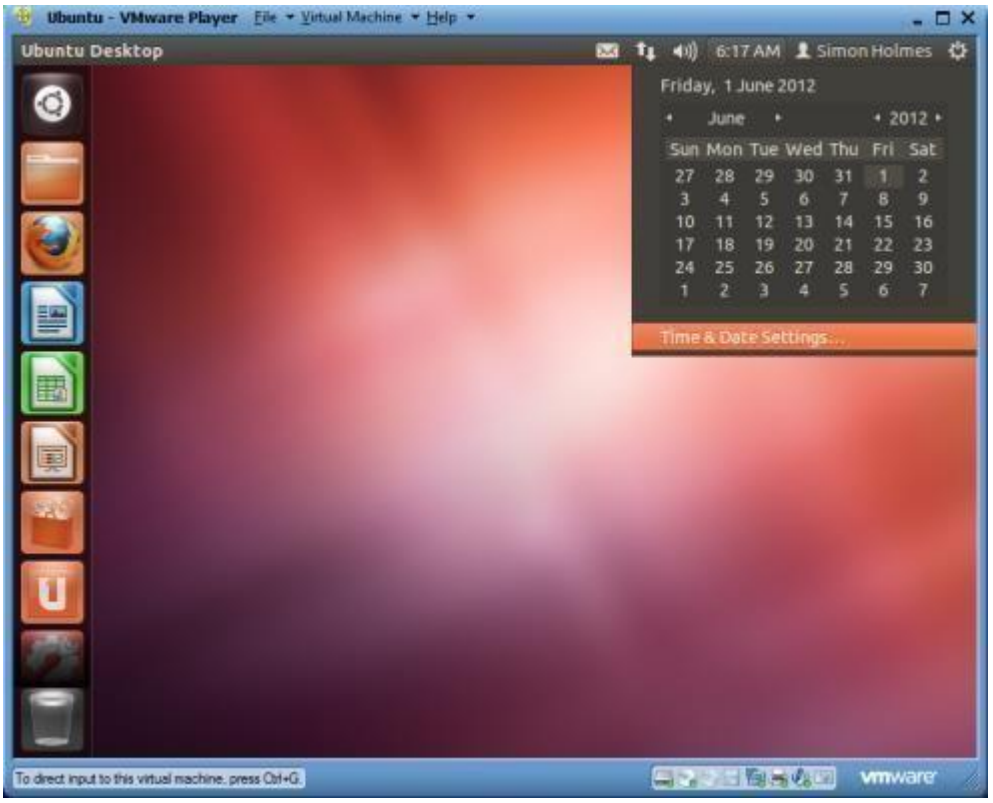


11. Ubuntu will then start to install

12. When all is done you'll be presented with the Ubuntu login screen. So enter your password and you're on your way.



13. Click the clock in the top right to set your time and date settings



15. Once you've set that up, you're up and running with Ubuntu in VMware Player on your Windows machine.

RESULT:

Ubuntu on virtual machine is installed successfully.

EXP NO :2 DATE:	INSTALL A C COMPILER ON A VIRTUAL MACHINE CREATED USING VIRTUAL BOX AND EXECUTE SIMPLE PROGRAM

AIM:

To Install a C compiler in the virtual machine created using virtual box and execute Simple Program.

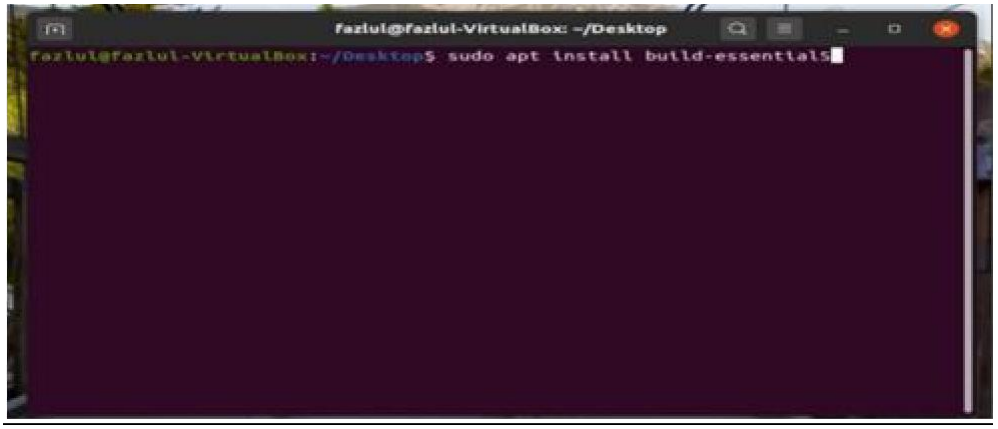
PROCEDURE:

Step 1: Install vmware.

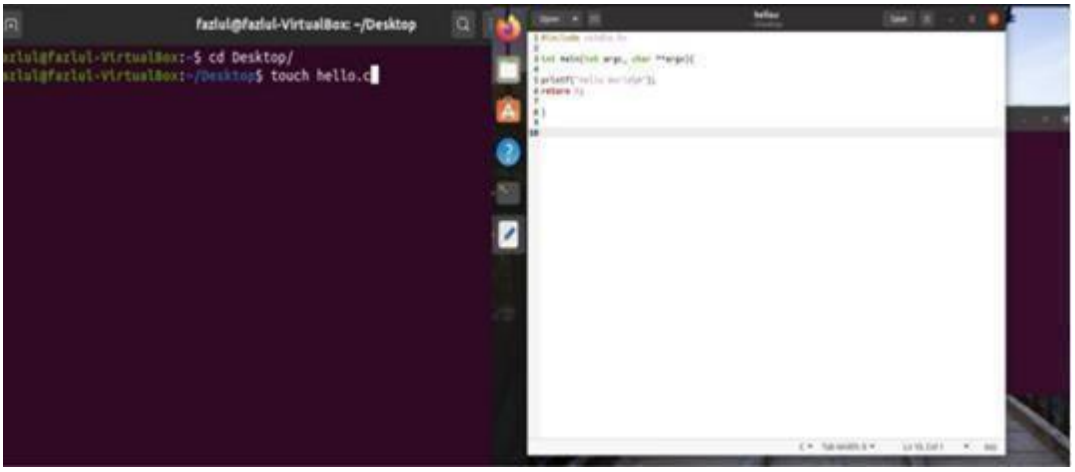
Step 2: After completing the virtual box installation ,Install Ubuntu

https://ubuntu.com/download/desktop

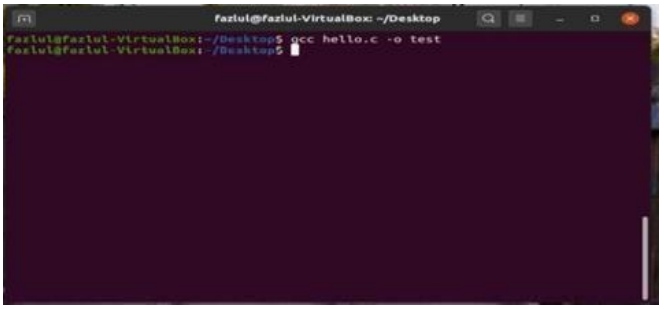
Step 3: Install C compiler in ubuntu.



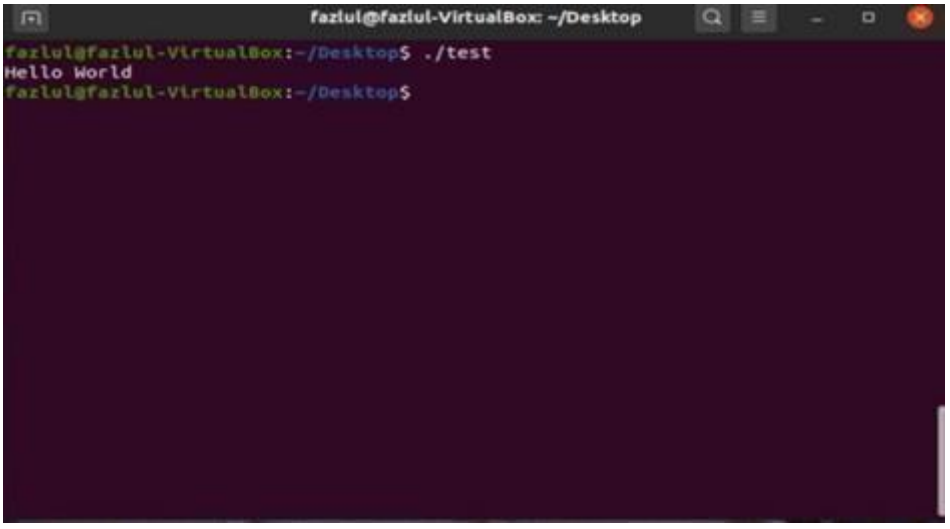
Step 4: Create a C file using terminal



Step 5 : Create a test file to view the output from the terminal using the command **$ gcc hello.c –o test**



Step 6 : Compile the test file using this command **$./test** to get the output.



RESULT:

C compiler in virtual machine is installed successfully.

EXP NO : 3 DATE:	INSTALL GOOGLE APP ENGINE.CREATE HELLO WORLD APP AND OTHER SIMPLE WEB APPLICATIONS USING PYTHON/JAVA

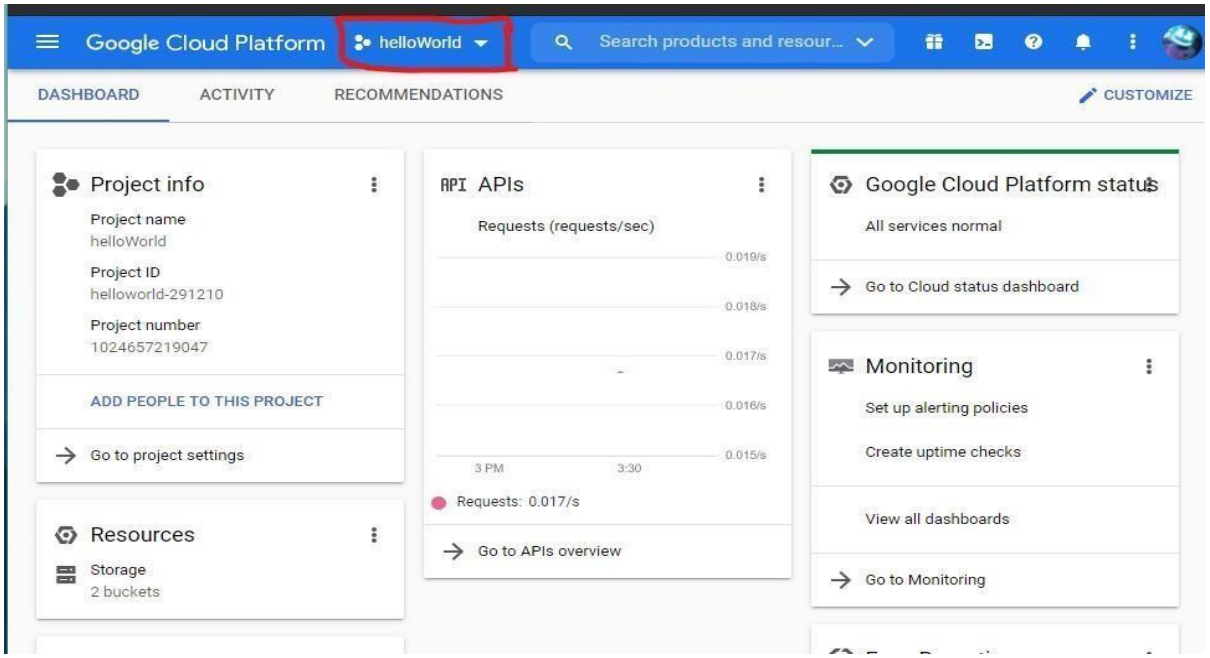
AIM:

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

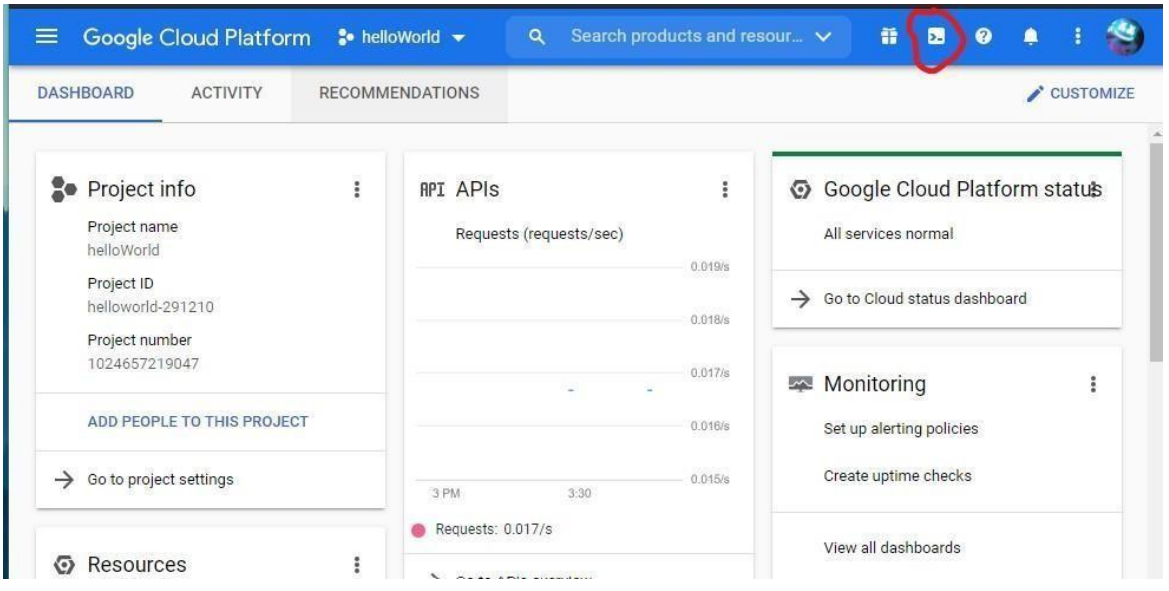
PROCEDURE :

Step 0: visit https://console.cloud.google.com/

Step 1: Create the new project and select the project



Step 3: Click the cloud shell icon on TOP RIGHT CORNOR



Step 4: Once the shell is open, enter the following commands:

- gcloud app create
- git clone https://github.com/GoogleCloudPlatform/python-docs-samples
- cd python-docs-samples/appengine/standard_python3/hello_world
- virtualenv --python python3 ~/envs/hello_world
- source ~/envs/hello_world/bin/activate
- pip install -r requirements.txt
- python main.py

Step 5: Then click “web preview” and select “preview on port 8080”.

RESULT:

Install Google App Engine. Create hello world app and other simple web applications using python/java is executed successfully and the result is obtained successfully.

EXP NO : 4 DATE:	USE GAE LAUNCHER TO LAUNCH WEB APPLICATIONS

AIM:

To Use GAE launcher to launch the web applications.

PROCEDURE:

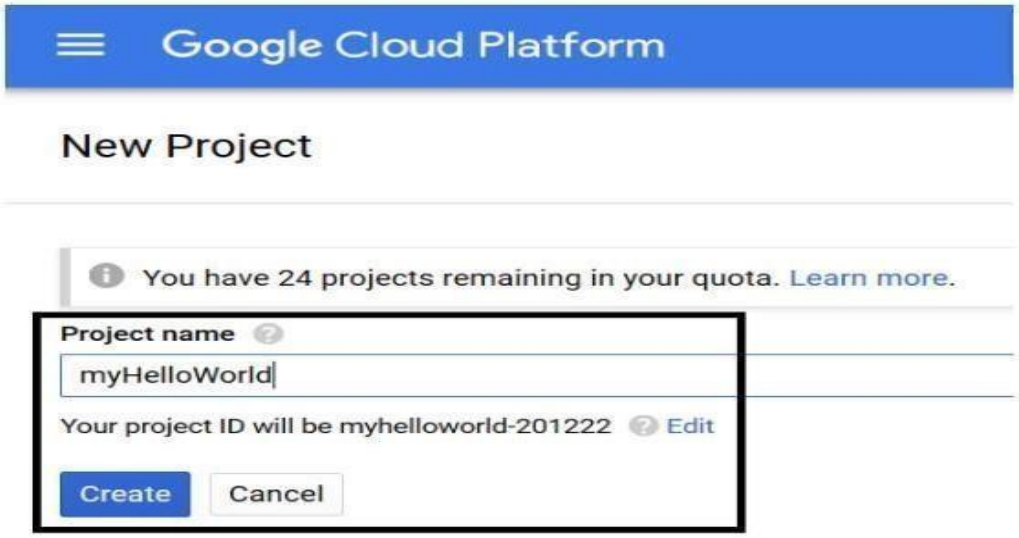
STEP 0 : Visit https://console.cloud.google.com/

STEP 1: Create a new Cloud Console project or retrieve the project ID of an existing project to use:

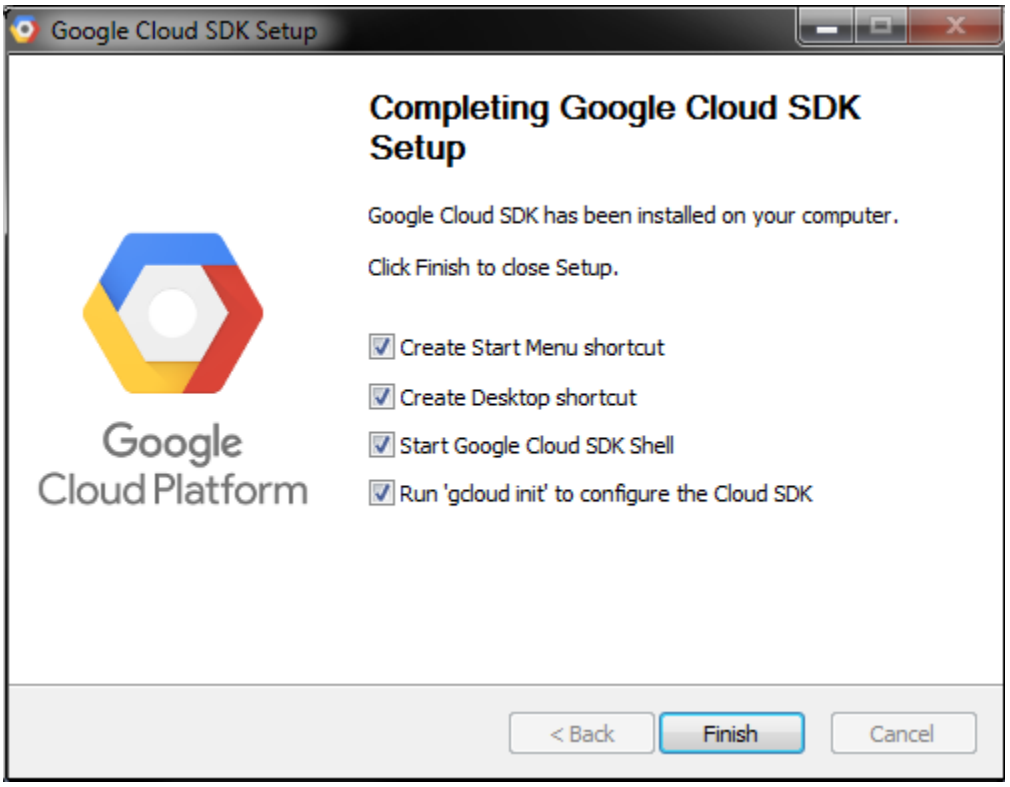
Go to the project page

Image courtesy. https://console.cloud.google.com/home

Image courtesy https://console.cloud.google.com/home



STEP 2: Install and then initialize the Google Cloud SDK



STEP3: Creating a website to host on Google App Engine.

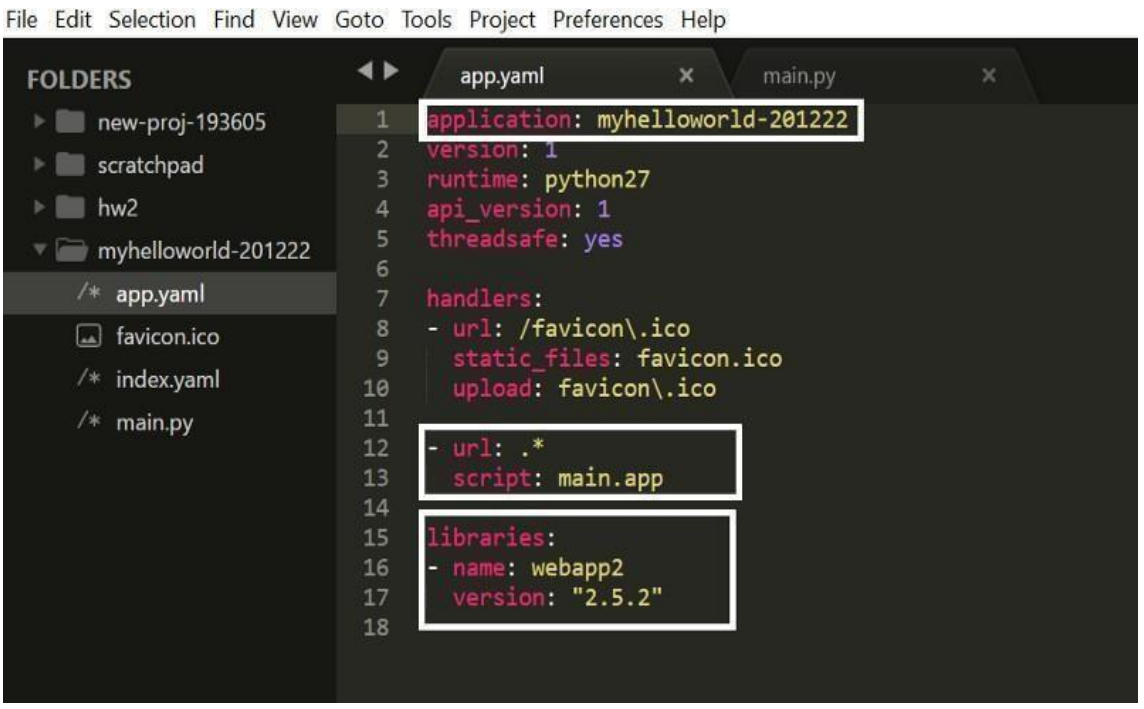
Basic structure for the project

This guide uses the following structure for the project:

•app.yaml: Configure the settings of your App Engine application.

- www/: Directory to store all of your static files, such as HTML, CSS, images, and JavaScript.
- css/: Directory to store stylesheets.
- style.css: Basic stylesheet that formats the look and feel of your site.
- images/: Optional directory to store images.
- index.html: An HTML file that displays content for your website.
- js/: Optional directory to store JavaScript files.
- Other asset directories.

STEP 4: Creating the app.yaml file



1. Create a directory that has the same name as your project ID.

You can find your project ID in the console

2. In directory that you just created, create a file named app.yaml.

3.Edit the app.yaml file and add the following code to the file:

4.runtime: python27

api_version: 1

threadsafe: true

handlers:

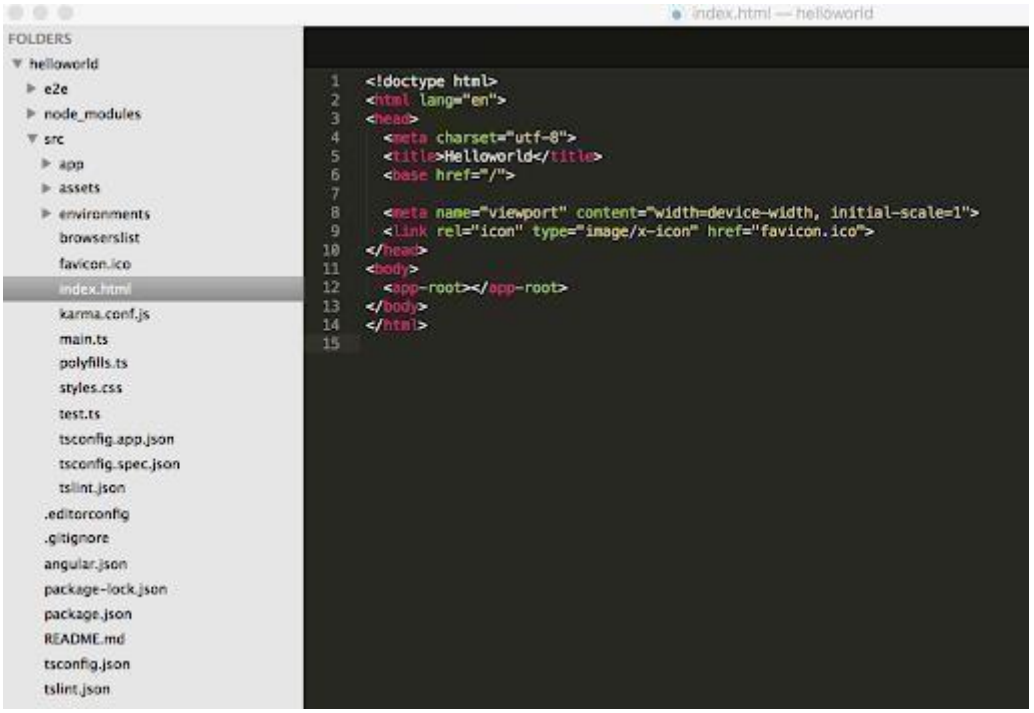
- url: /

static_files: www/index.html upload: www/index.html

- url: /(.*)

static_files: www/\1 upload: www/(.*)

STEP 5: Creating the index.html file



<html>

<head>

<title>Hello, world!</title>

<link rel="stylesheet" type="text/css" href="/css/style.css">

</head>

<body>

<h1>Hello, world!</h1>

This is a simple static HTML file that will be served from Google App Engine.

</body>

</html>

STEP 6: Deploying your application to App Engine

```
(env) pavanraonavule@cloudshell:~/fastapi (fastapi-tutlinks-demo)$ gcloud app deploy app.yaml
Services to deploy:

descriptor:      [/home/pavanraonavule/fastapi/app.yaml]
source:          [/home/pavanraonavule/fastapi]
target project:  [fastapi-tutlinks-demo]
target service:  [default]
target version:  [20200326t204745]
target url:      [https://fastapi-tutlinks-demo.appspot.com]


Do you want to continue (Y/n)?  Y

Beginning deployment of service [default]...
Created .gcloudignore file. See `gcloud topic gcloudignore` for details.
╔═══ Uploading 1029 files to Google Cloud Storage ═══╗

File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://fastapi-tutlinks-demo.appspot.com]

You can stream logs from the command line by running:
  $ gcloud app logs tail -s default

To view your application in the web browser run:
  $ gcloud app browse
(env) pavanraonavule@cloudshell:~/fastapi (fastapi-tutlinks-demo)$
```

STEP 7: Once the shell is open, enter the following commands:

1.gcloud app create

2. git clone https://github.com/GoogleCloudPlatform/python-docs-samples

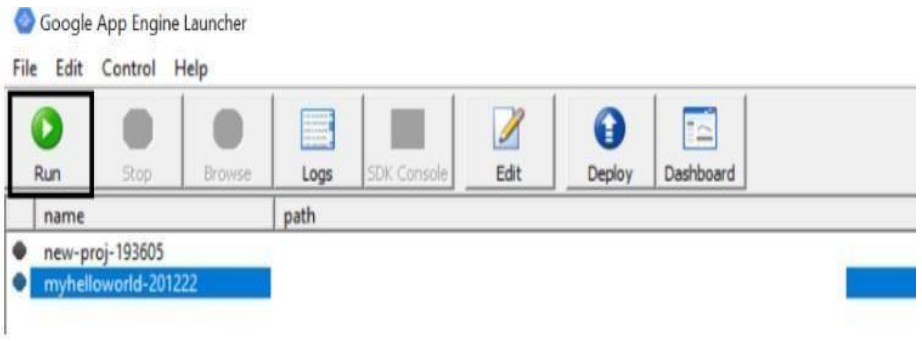
3.cd python-docs-samples/appengine/standard_python3/hello_world

4.virtualenv --python python3 ~/envs/hello_world

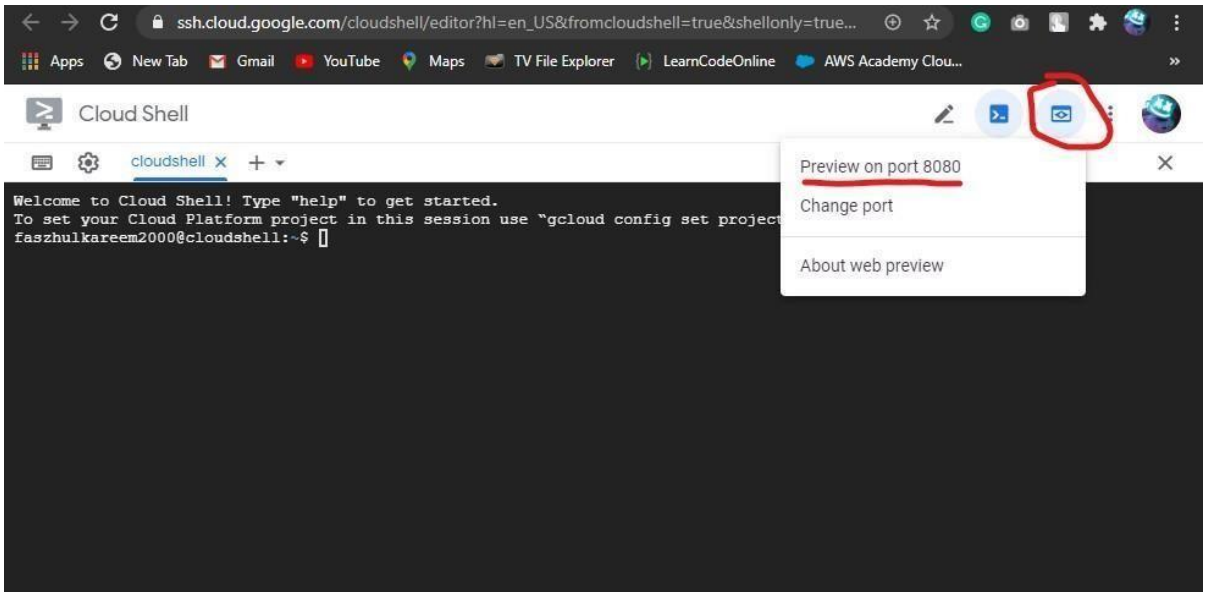
5.source ~/envs/hello_world/bin/activate

6.pip install -r requirements.txt

7.python main.py



STEP 8: Then click “web preview” and select “preview on port 8080.



Step 6: To Deploy the program, enter the following command:

gcloud app deploy app.yaml

RESULT:

Use GAE launcher to launch the web applications is successfully executed.

EXP NO :5 DATE:	**SIMULATE A CLOUD SCENARIO USING CLOUDSIM AND RUN A SCHEDULING ALGORITHM THAT IS NOT PRESENT IN CLOUDSIM**

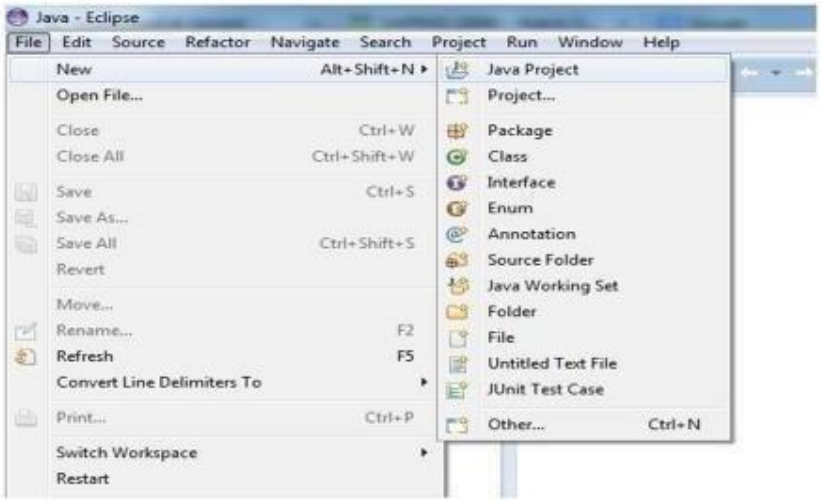
AIM :

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

PROCEDURE:

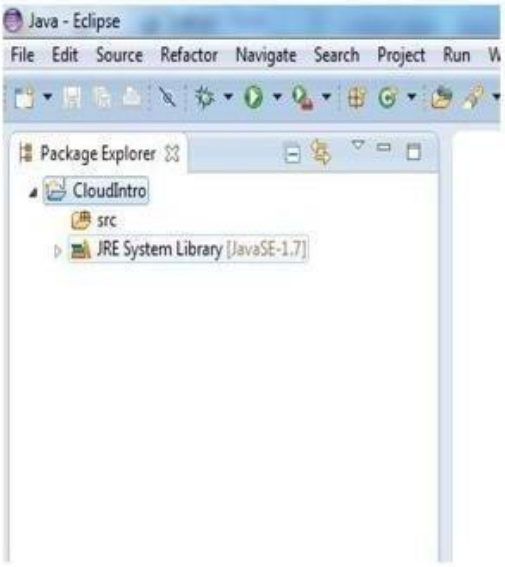
STEP BY STEP INSTALLATION OF CLOUD SIM INTO ECLIPSE

1. Open up Eclipse and go to Menu Section, then click File, keep on clicking New and finally select java project.

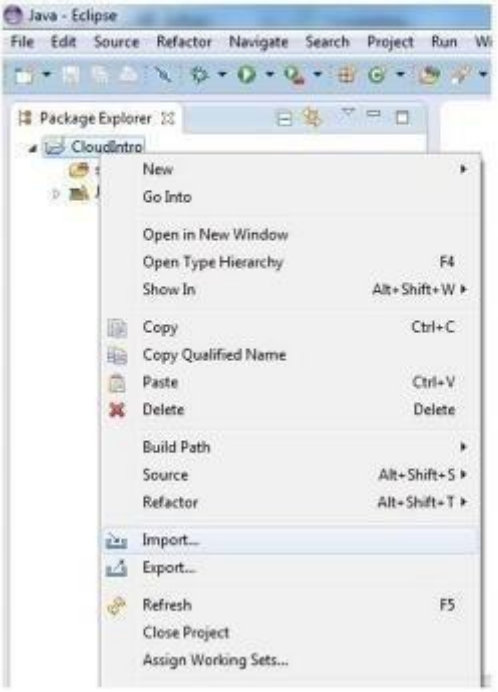


2. A new window will get open. Put a foot on to the following steps:-
   2.1.Enter project name. (I have named it as CloudIntro)
   2.2 In the next line you will see the path where your project will be created.
   2.3 Next You need to select the JRE environment.
   2.4 Finally Click Finish.

3. Once you hit finish. An empty project named CloudIntro will be created in the project List.

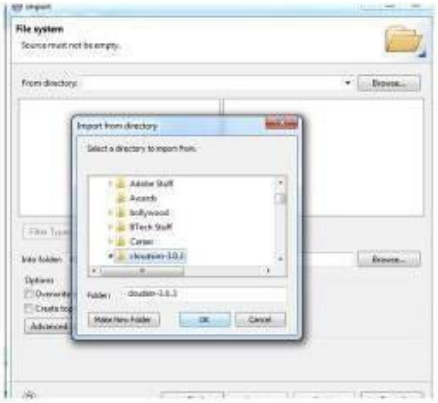


4. Next step is to go to the project CloudIntro, right click on it. Click Import as shown.



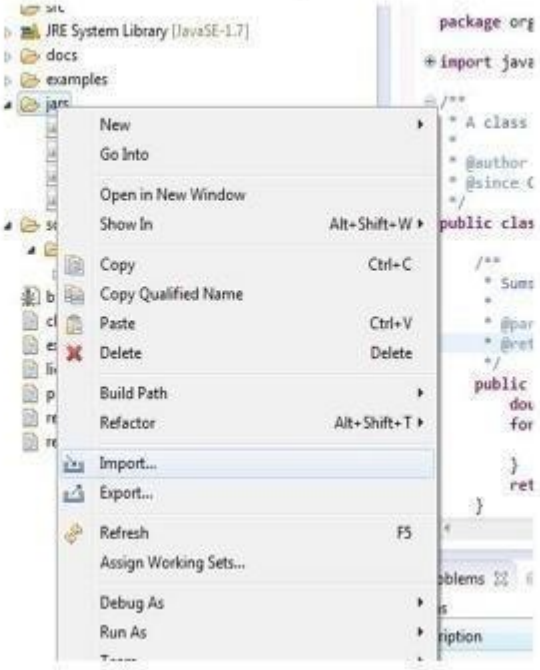
5. A new window will get open, now click File System as demonstrated in the Figure.
6. Next Step is to go to the directory where you have extracted your cloudsim tool. Figure6 is shown to guide you to get into the directory where your cloudsim folder is located.

7. Select the cloudsim and click Finish as shown in the Figure



8. Now go to the link http://commons.apache.org/proper/commonsmath/download_math.cgi. Download the file named as "commons-math3-3.4.1-bin.zip". Unzip this file. We need jar files for math functions.

9. Now go to the left side of the eclipse tool in the project bar. Go to jar and right click on it. Click import as shown in the Figure.



10. Now go to the folder where you have placed the downloaded and extracted file as described by point 8. Then all you have to do is select that jar file and hit finish as shown by the Figure.

11. Finally the cloud sim is installed into your Eclipse environment.

RESULT:

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

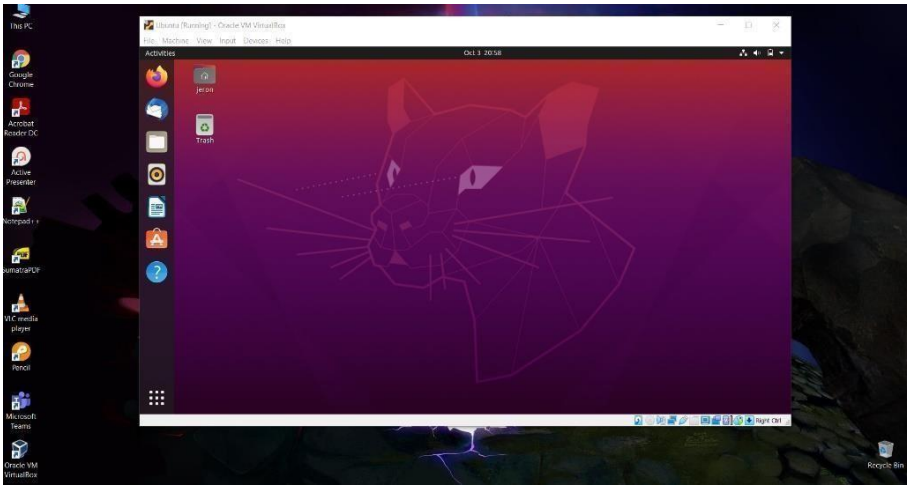
EXP NO :6 DATE:	FIND A PROCEDURE TO TRANSFER FILES FROM ONE VIRTUAL MACHINE TO ANOTHER VIRTUAL MACHINE

AIM:

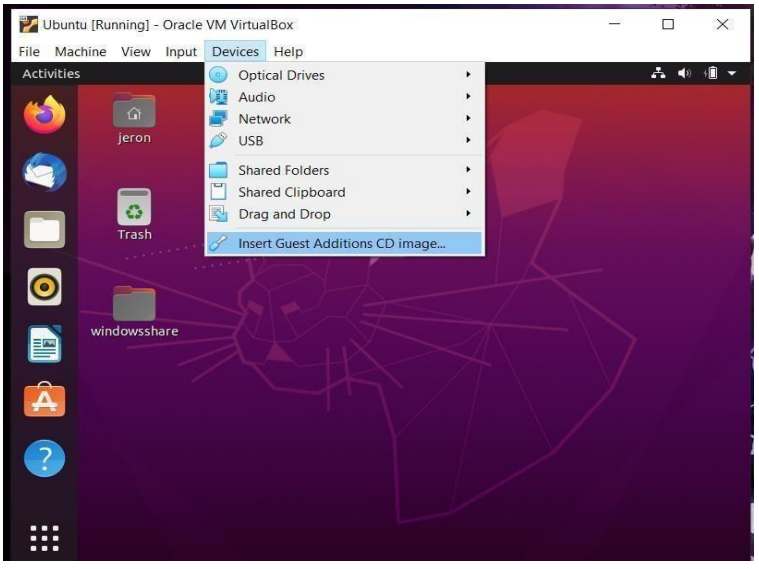
To Find a procedure to transfer the files from one virtual machine to another virtual machine

PROCEDURE:

1. Download the virtual Box in your Host OS and Install the Guest OS in the virtual machine
2. First we will share a folder from host machine to one vm. Create a folder in the Host Machine.

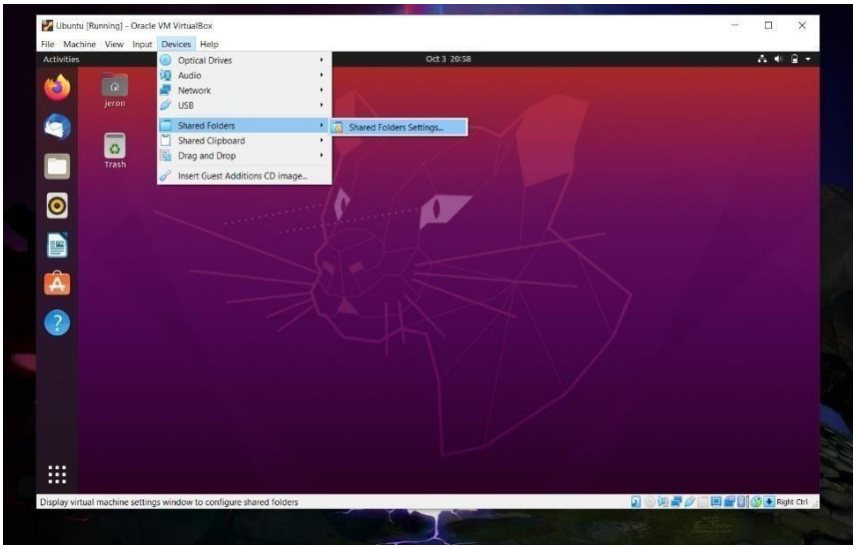


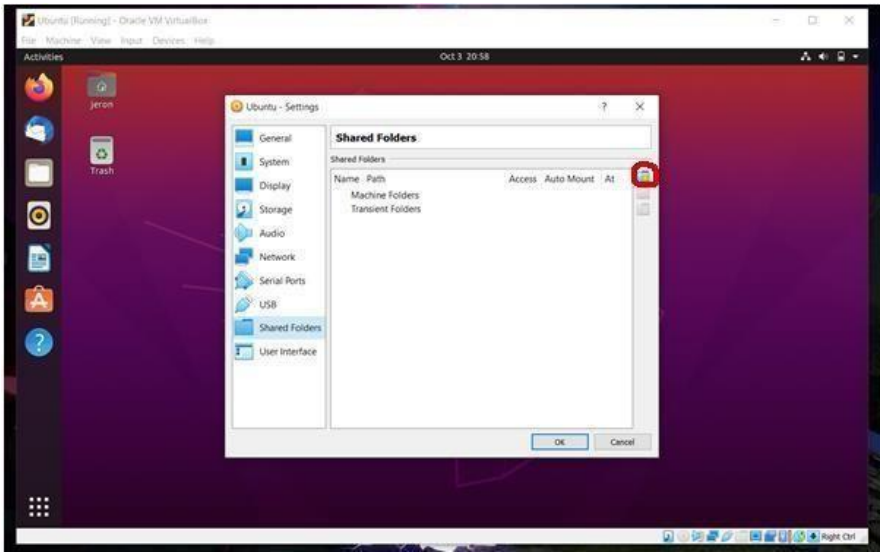
3. Install the vm and download the necessary files and then install another vm say vm2 and download the necessary files.
4. Now run the vm and Go to **Device->Insert Guest additional CD image** option and install the CD image.



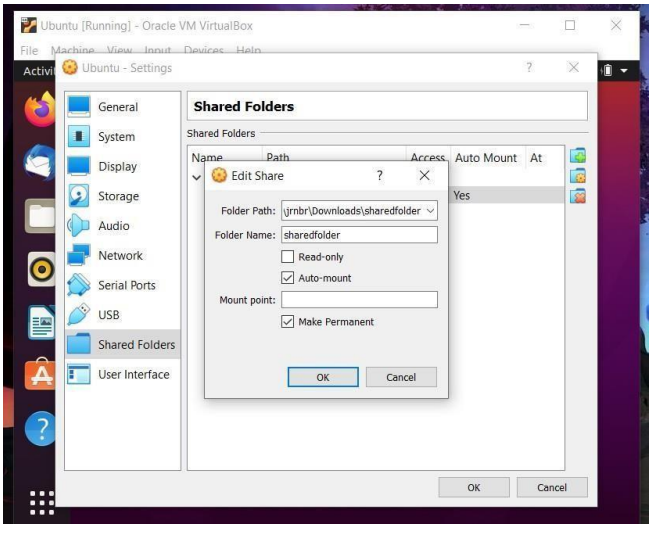
and eject the CD image this helps in the access of file from one vm to another.

5. Go to **Device->Shared Folder->Shared Folder Settings** and a dialog box appears.

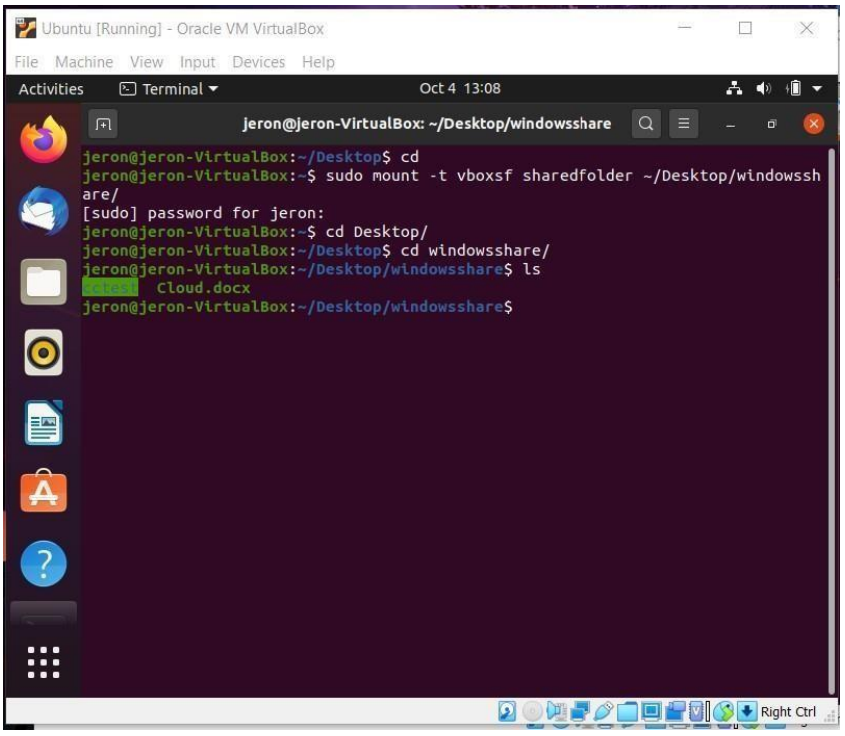




6. Click on the AddFileas marked in the above picture and give the directory of a sharefolder in the Host Machine and click ok



7. Now open the terminal and create a folder of your choice and command sudomkdir ~/Desktop/ windowshare in the vm folder with windowsshare will be created in the vm desktop.And type sudomount -t vboxf “folder name” ~/Desktop/ windowshare and enter.

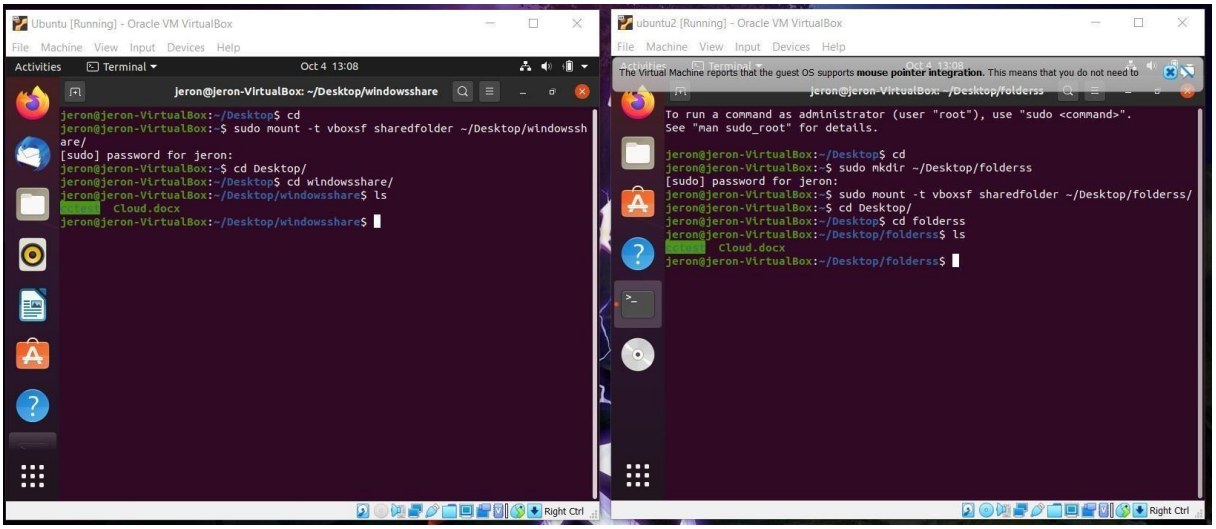


8. Now, we can access the shared file in the vm. Now to share folder from one vm another vm

9.Create a file inside the folder windowsshare of vm and Run the vm2 and do step 4 to 6.

10. Now open the **terminal** in vm2 and create a folder of your choice as step 7.

11.Now type **sudo munt -t vboxsf “filename” ~/Desktop/”folderinvm2”**



12.Now enter and view the folder and the folder in the vm 1 will be seen in vm2

RESULT:

Transfer the files from one virtual machine to another virtual machine is executed successful

EXP NO :7 DATE:	INSTALL HADOOP SINGLE NODE CLUSTER AND RUN SIMPLE APPLICATIONS LIKE WORDCOUNT

AIM:

To Install Hadoop single node cluster and run simple applications like wordcount.

PROCEDURE:

Install virtual box

https://download.virtualbox.org/virtualbox/6.1.14/VirtualBox-6.1.14-140239-Win.exe

After completing the virtual box installation ,Install Ubuntu

https://ubuntu.com/download/desktop

Click to download Hadooop:

https://hadoop.apache.org/releases.html

STEPS:

Step 1: Install Java

Command: sudo apt install openjdk-8-jdk

cd /usr/lib/jvm

Step 2:Openbashrc file

Command: sudo nano ~/.bashrc

Location Path of the Environment variable:

Past these:

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export PATH=$PATH:export PATH=$PATH:/usr/lib/jvm/java-8-openjdk-amd64/bin

export HADOOP_HOME=~/Downloads/hadoop-3.2.1/

export PATH=$PATH:$HADOOP_HOME/bin

export PATH=$PATH:$HADOOP_HOME/sbin

export HADOOP_MAPRED_HOME=$HADOOP_HOME

export YARN_HOME=$HADOOP_HOME

export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop

export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native

export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"

export

HADOOP_STREAMING=$HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.2.1.jar

export HADOOP_LOG_DIR=$HADOOP_HOME/logs

export PDSH_RCMD_TYPE=ssh

export HIVE_HOME=~/Downloads/apache-hive-3.1.2-bin

export PATH=$PATH:~/Downloads/apache-hive-3.1.2-bin/bin

Then come out of the file.

Step 3:Install Secure shell

Command: sudo apt-get install ssh

Step 4: Install parallel shell

Command: sudo apt-get install pdsh

Step 5: Extract Hadoop tar file:

Command: tar -zxvf ~/Downloads/hadoop-3.2.1.tar.gz

Go to Hadoop directory

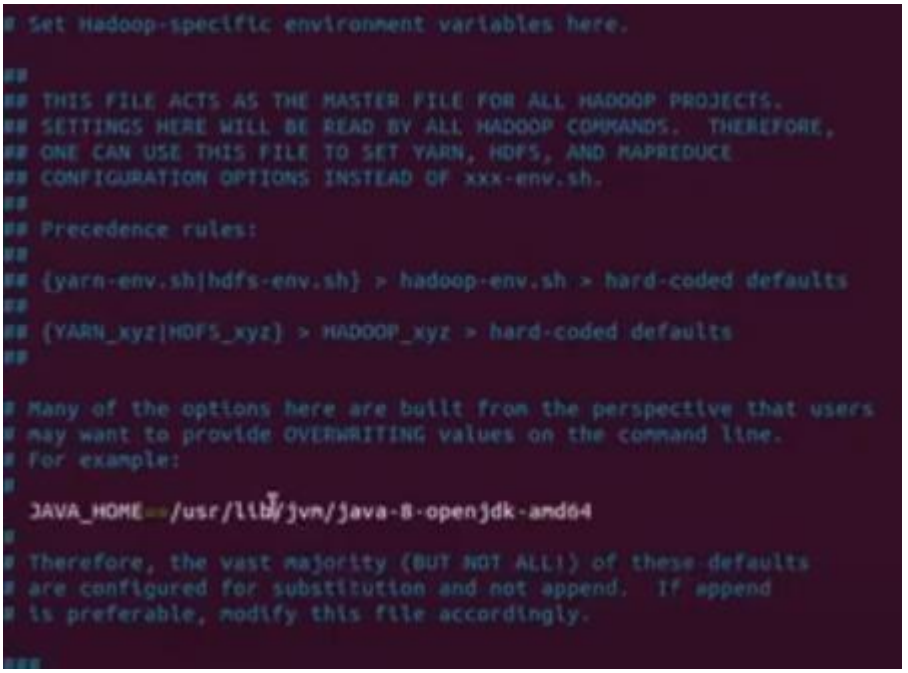
Command: cd hadoop-3.2.1/etc/Hadoop

Step 6: Open Hadoop-env.h

Command: sudo nano hadoop-env.h

JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

(Set the path for JAVA_HOME)



Step 7: Go to coresite.xml

Command: sudo nano coresite.xml

Paste these:

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

<property>

<name>hadoop.proxyuser.dataflair.groups</name>

<value>*</value>

</property>

<property>

<name>hadoop.proxyuser.dataflair.hosts</name>

<value>*</value>

</property>

<property>

<name>hadoop.proxyuser.server.hosts</name>

<value>*</value>

</property>

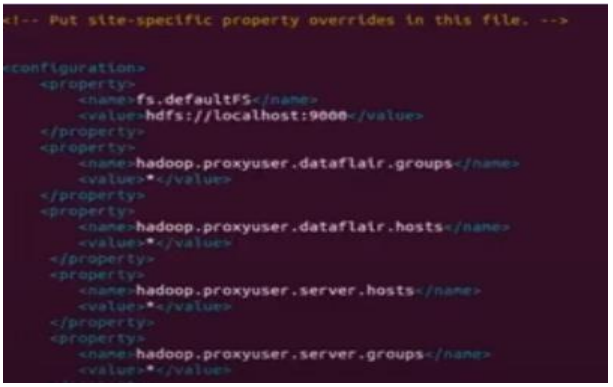
<property>

<name>hadoop.proxyuser.server.groups</name>

<value>*</value>

</property>

</configuration>



Step 8: Go to hdfs-site.xml

Command: sudo nano hdfs-site.xml

Paste these:

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

</configuration>

Step 9: Go to mapred-site.xml

Command: sudo nano mapred-site.xml

Paste these:

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

<property>

<name>mapreduce.application.classpath</name>

<value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoo

p/mapreduce/lib/*</value>

</property>

</configuration>

Step 10: Go to yarn-site.xml

Command: sudo nano yarn-site.xml

Paste these:

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.env-whitelist</name>

<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREP

END_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>

</property>

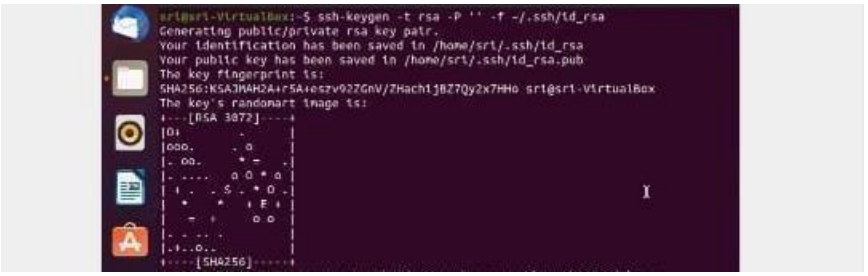
</configuration>

Step11:Establish Localhost:

Command: ssh localhost

Step 12: Key generation:

Command: ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa



Step 13: Granting Permission to the file

Command: chmod 0600 ~/.ssh/authorized_keys

Command:~/Downloads/hadoop-3.2.1/bin/hdfsnamenode–format

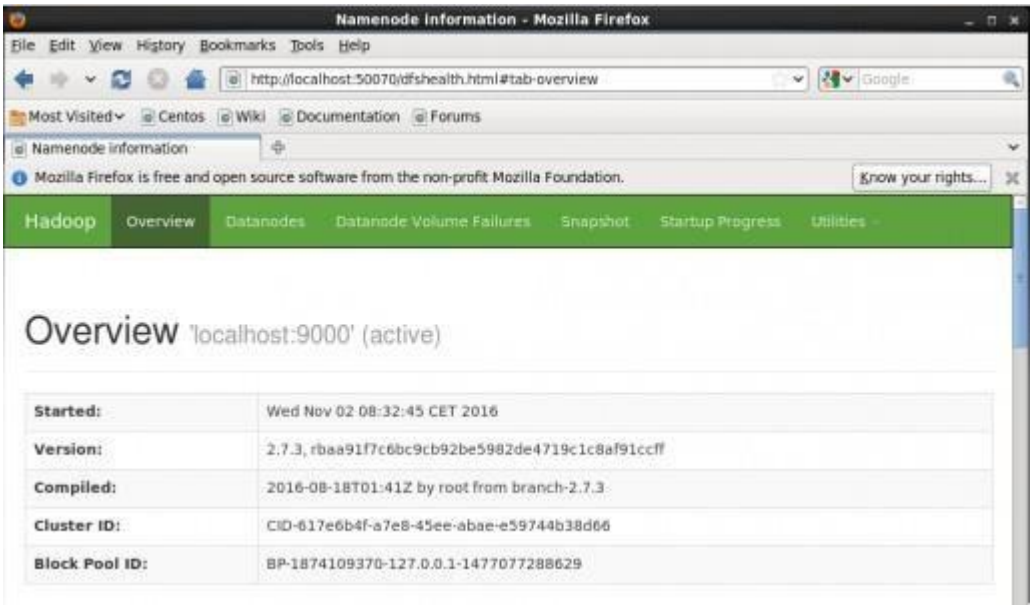
Step 14: Export pdsh

Command: export PDSH_RCMD_TYPE=ssh

Step 15:dfs node:

Command:~/Downloads/hadoop-3.2.1/sbin/start-dfs.sh

Step 16: Open the browser: Type **localhost:9870**



Wordcount program:

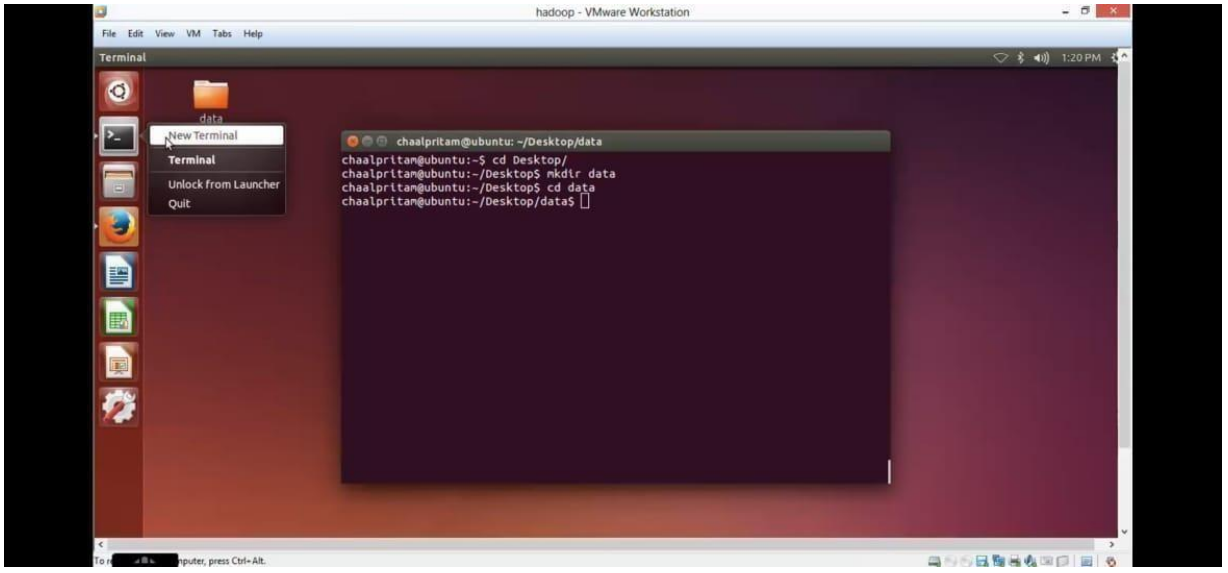
Step 1: Creating a working directory for your data

user@ubuntu:~$ cd Desktop/

user@ubuntu:~/ Desktop$ mkdir data

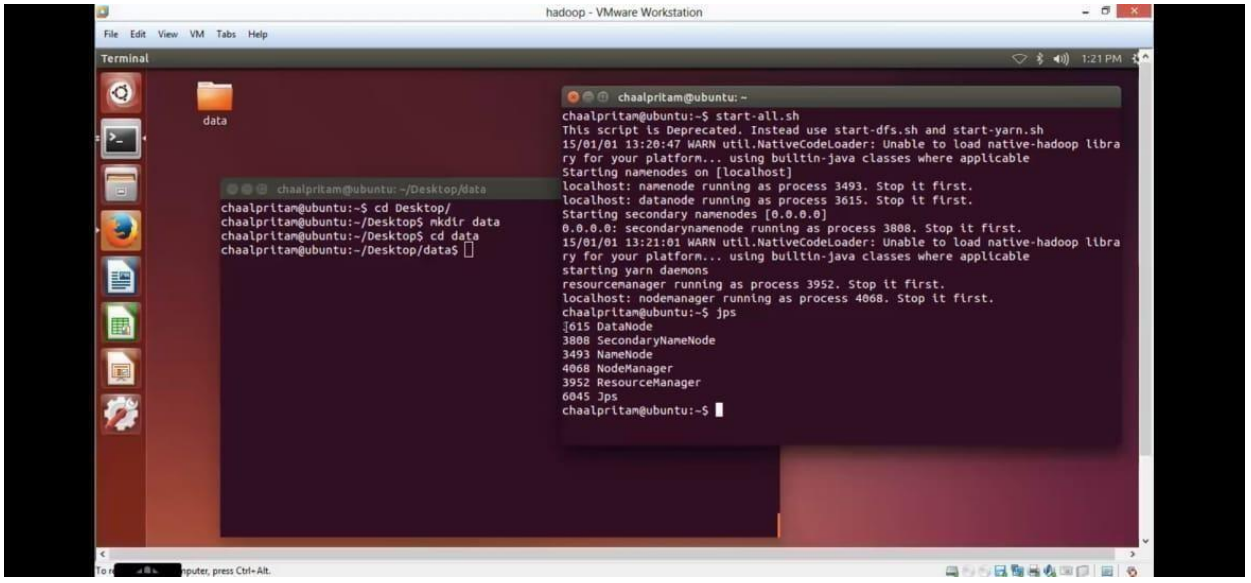
user@ubuntu:~/ Desktop$ cd data

user@ubuntu:~/ Desktop/ data$



Step 2:

user@ubuntu:~ start-all.sh



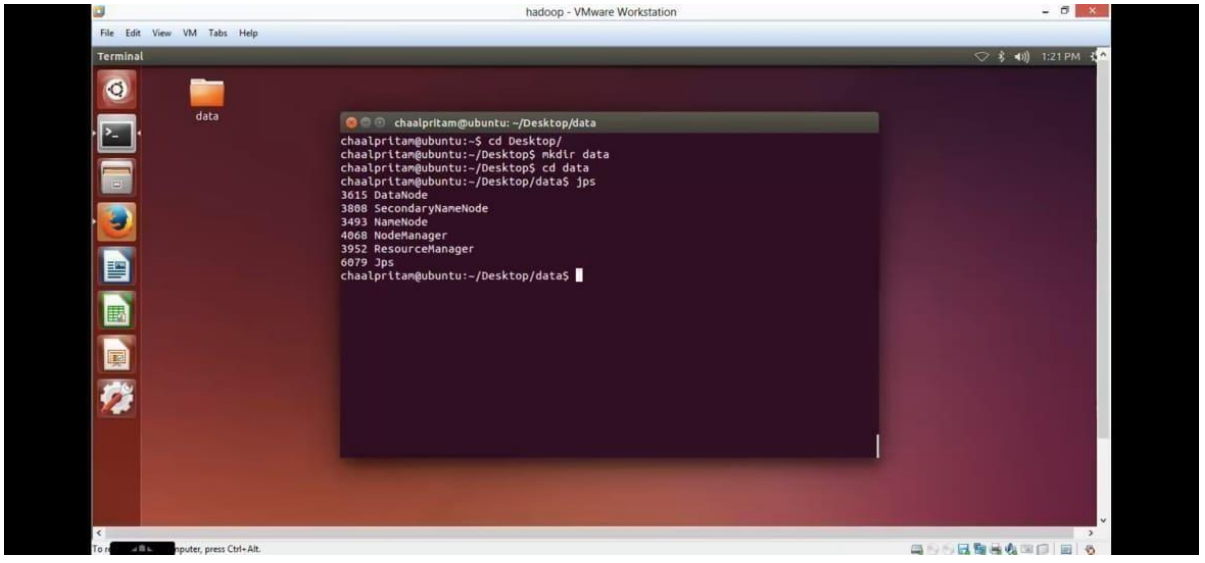
Step 3:

user@ubuntu:~$ cd Desktop/

user@ubuntu:~/ Desktop$ mkdir data

user@ubuntu:~/ Desktop$ cd data

user@ubuntu:~/ Desktop/ data$ jps

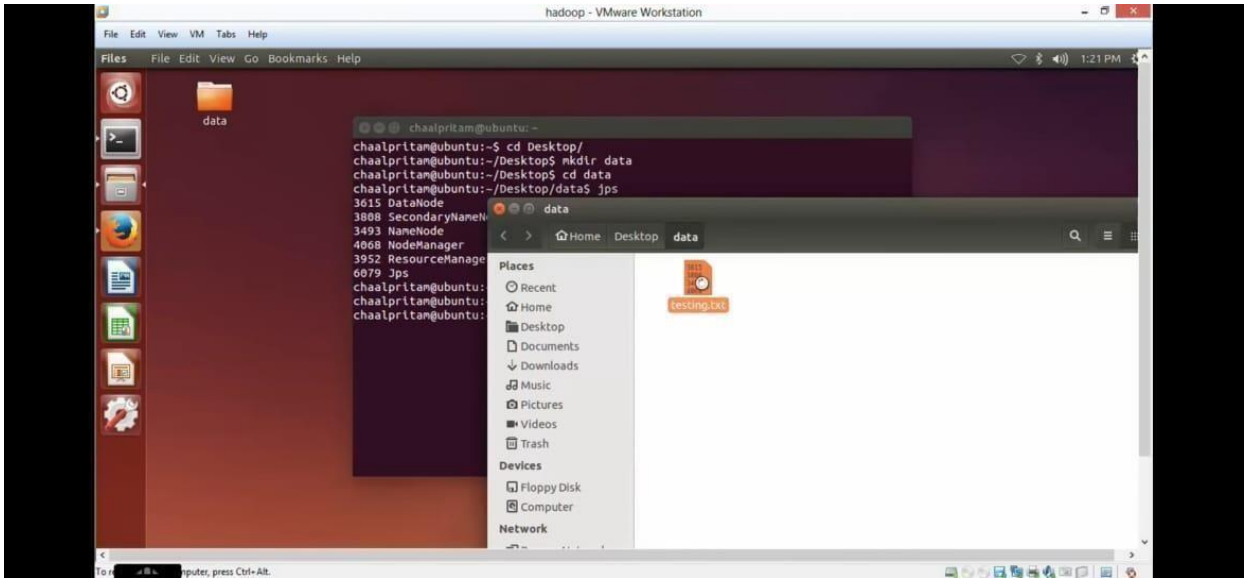


Step 4:

user@ubuntu:~/ Desktop/ data$ jps>> testing.txt

user@ubuntu:~/ Desktop/ data$ cd

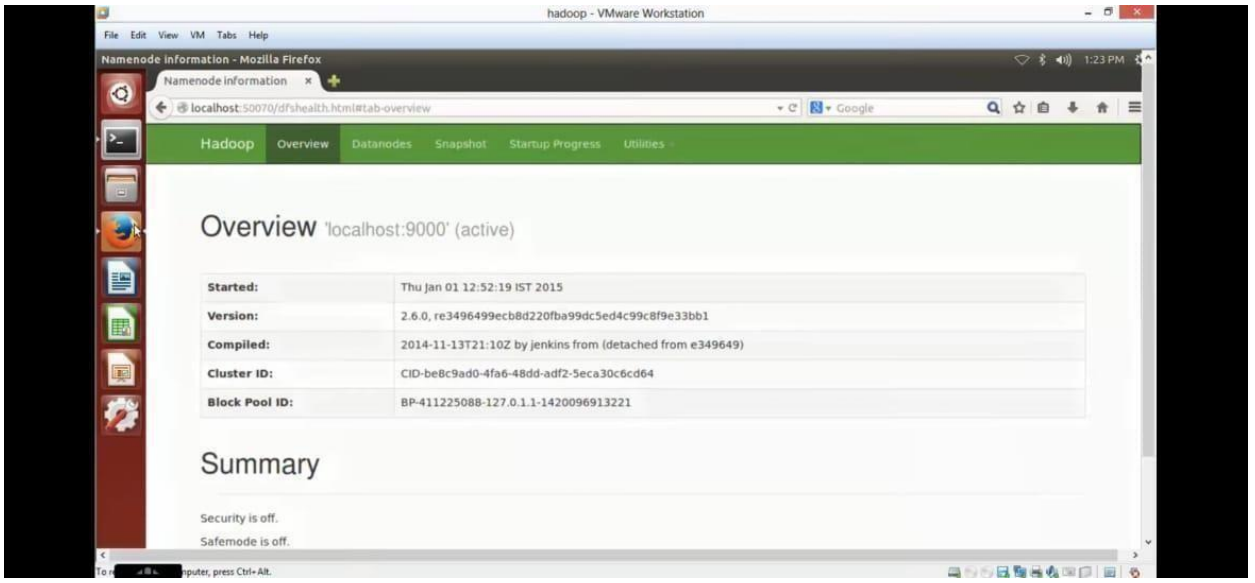
user@ubuntu:~$

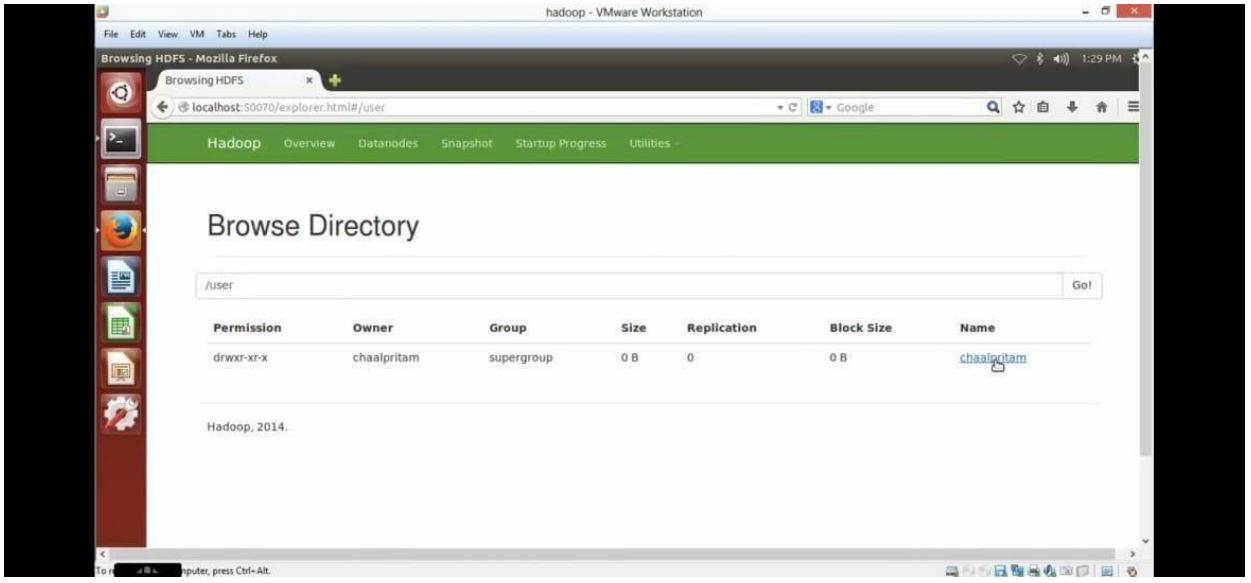


Step 5:

user@ubuntu:~$ cd /user /local /hadoop/

user@ubuntu: /user /local /hadoop$ bin/hdfsdfs -mkdir /user

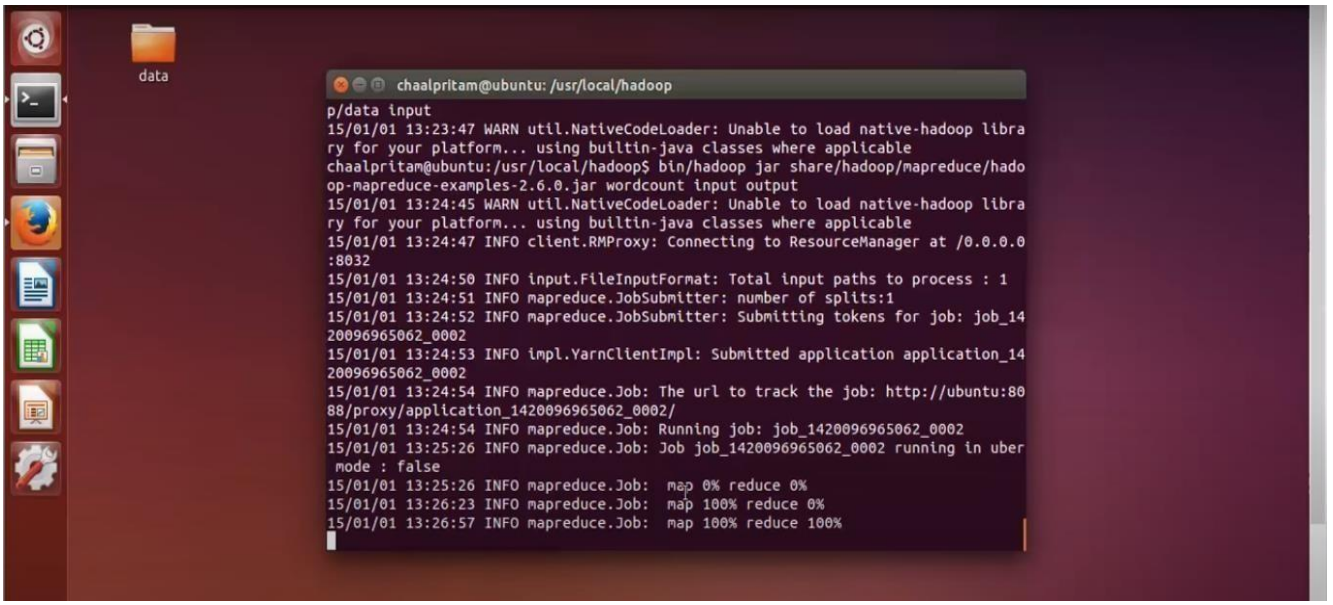


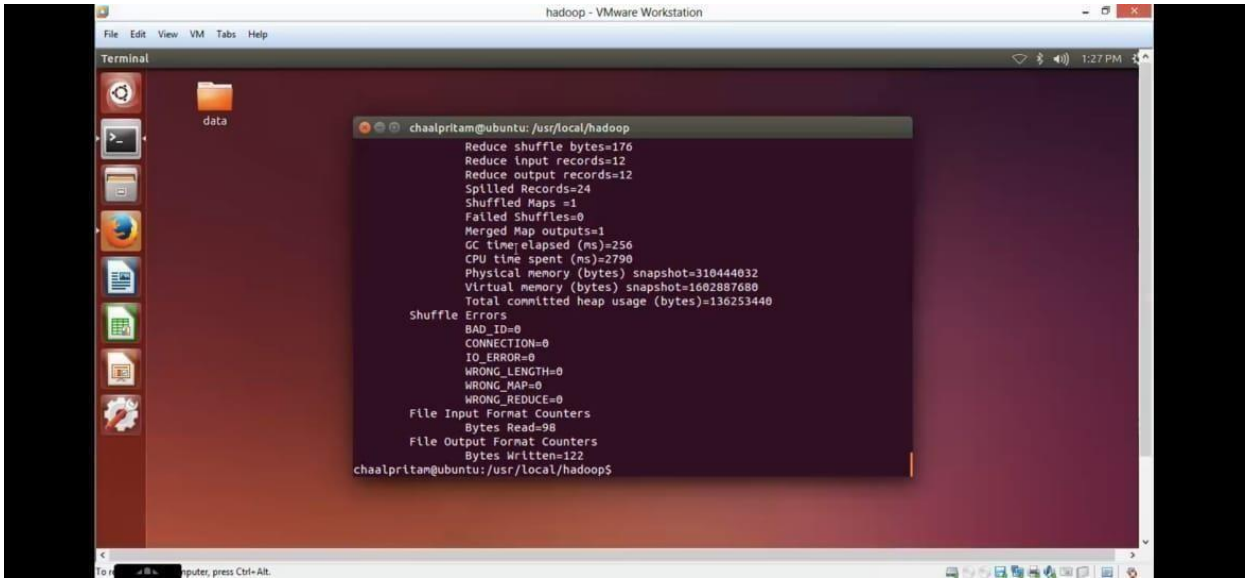


Step 6:

user@ubuntu: /user /local /hadoop$ bin/hdfsdfs -put /home /user /Desktop /data input

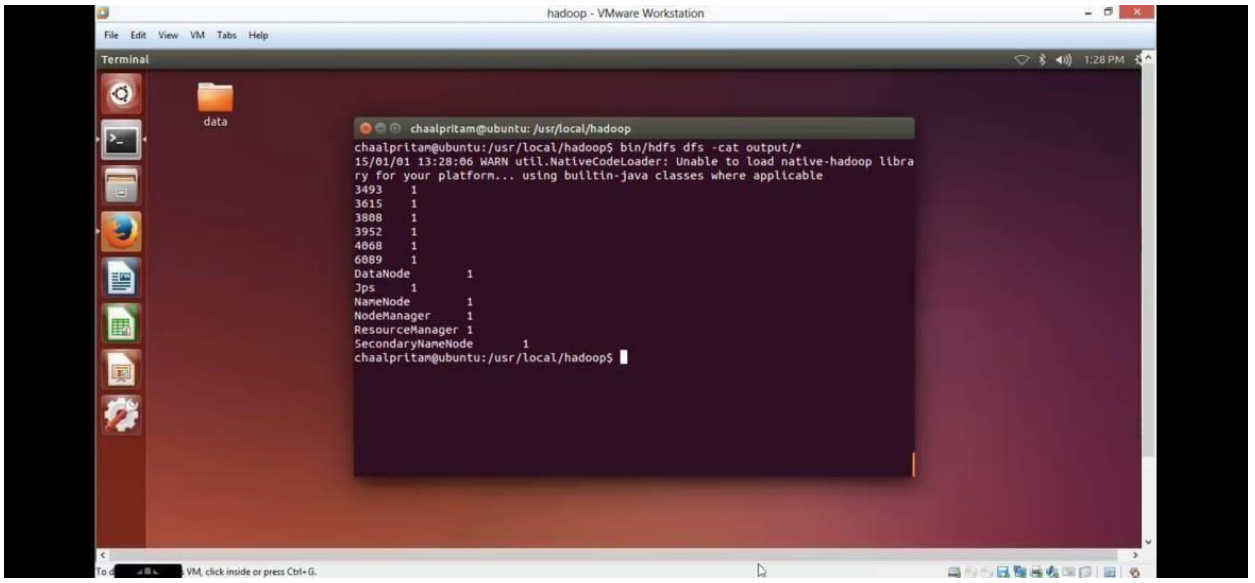
user@ubuntu: /user /local /hadoop$ bin/hadoop jar share/hadoop /mapreduce /hadoop-mapreduce-examples-2.6.0. jar wordcount input output



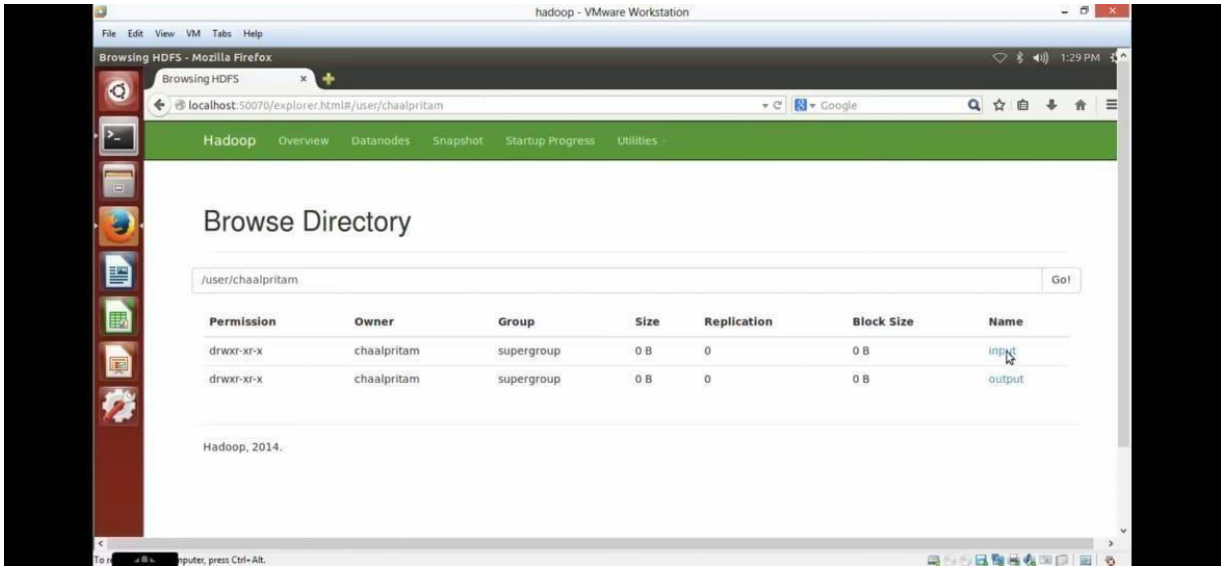


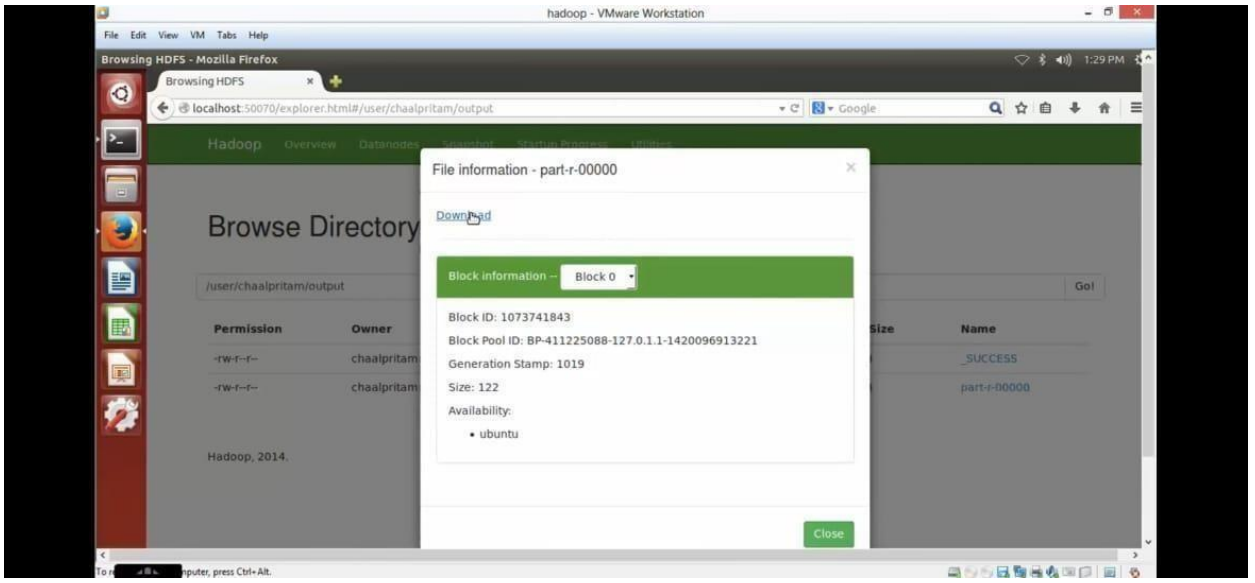
Step 7:

user@ubuntu: /user /local /hadoop$ bin/hdfsdfs -cat output /*

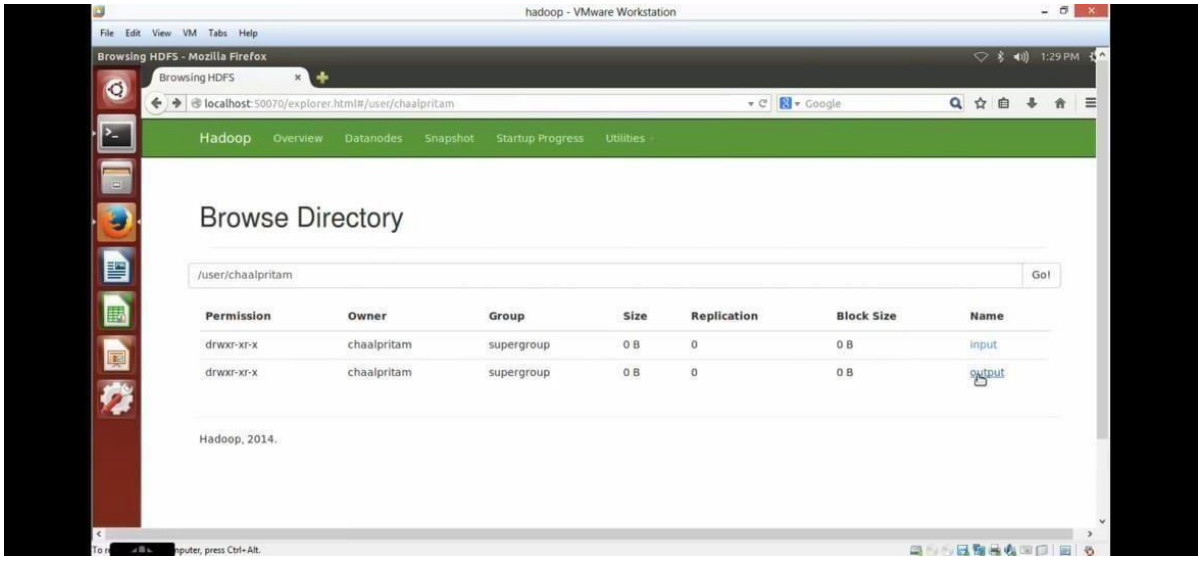

hadoop - VMware Workstation
File Edit View VM Tabs Help
Terminal
1:28 PM
data
chaalpritam@ubuntu: /usr/local/hadoop
chaalpritam@ubuntu:/usr/local/hadoop$ bin/hdfs dfs -cat output/*
15/01/01 13:28:06 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
3493 1
3615 1
3808 1
3952 1
4068 1
6089 1
DataNode 1
Jps 1
NameNode 1
NodeManager 1
ResourceManager 1
SecondaryNameNode 1
chaalpritam@ubuntu:/usr/local/hadoop$

Step 8:

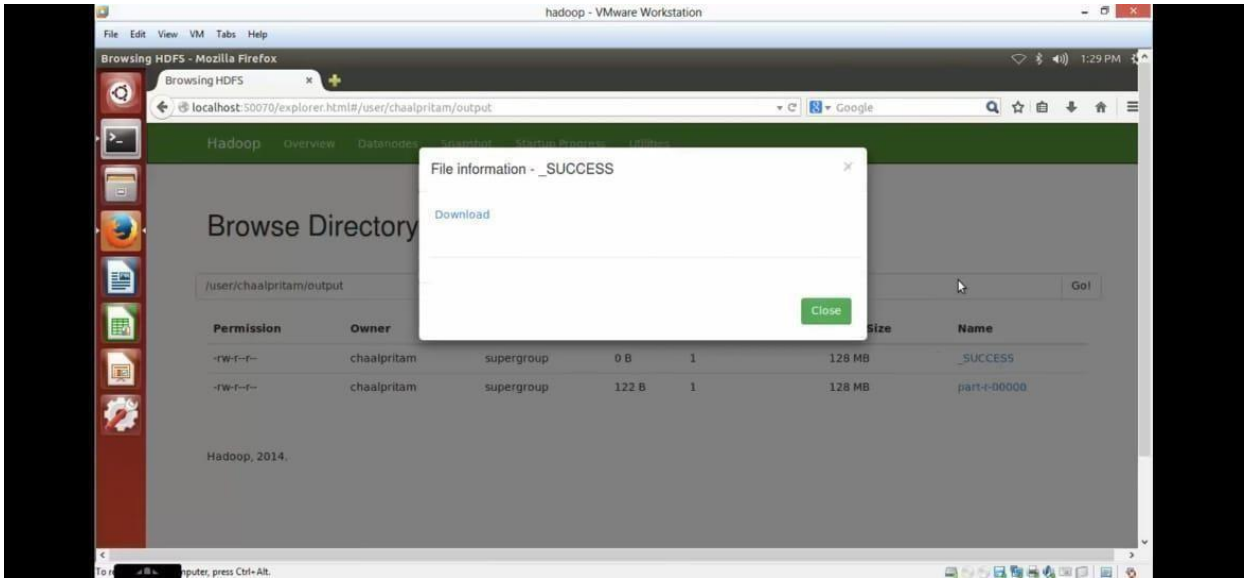

hadoop - VMware Workstation
File Edit View VM Tabs Help
Browsing HDFS - Mozilla Firefox
1:29 PM
Browsing HDFS
localhost:50070/explorer.html#/user/chaalpritam
Google
Hadoop
Overview
Datanodes
Snapshot
Startup Progress
Utilities
Browse Directory
/user/chaalpritam
Go!
Permission
Owner
Group
Size
Replication
Block Size
Name
drwxr-xr-x
chaalpritam
supergroup
0 B
0
0 B
input
drwxr-xr-x
chaalpritam
supergroup
0 B
0
0 B
output
Hadoop, 2014.


hadoop - VMware Workstation
File Edit View VM Tabs Help
Browsing HDFS - Mozilla Firefox
Browsing HDFS
localhost:50070/explorer.html#/user/chaalpritam/output
Hadoop
Browse Directory
/user/chaalpritam/output
Permission
Owner
Name
-rw-r--r--
chaalpritam
_SUCCESS
part-r-00000
Hadoop, 2014.
File information - part-r-00000
Download
Block information --
Block 0
Block ID: 1073741843
Block Pool ID: BP-411225088-127.0.1.1-1420096913221
Generation Stamp: 1019
Size: 122
Availability:
ubuntu
Close

Step 9:


hadoop - VMware Workstation
File Edit View VM Tabs Help
Browsing HDFS - Mozilla Firefox
Browsing HDFS
localhost:50070/explorer.html#/user/chaalpritam
Hadoop
Overview
Datanodes
Snapshot
Startup Progress
Utilities
Browse Directory
/user/chaalpritam
Go!
Permission
Owner
Group
Size
Replication
Block Size
Name
drwxr-xr-x
chaalpritam
supergroup
0 B
0
0 B
input
output
Hadoop, 2014.

Step10:



RESULT:

Install Hadoop single node cluster and run simple applications like wordcount is executedsuccessfully.

EXP NO :8 DATE:	INSTALL HADOOP SINGLE NODE CLUSTER AND RUN SIMPLE APPLICATIONS LIKE WORDCOUNT

AIM:

Install hadoop single node cluster and run simple applications like wordcount

PROCEDURE:

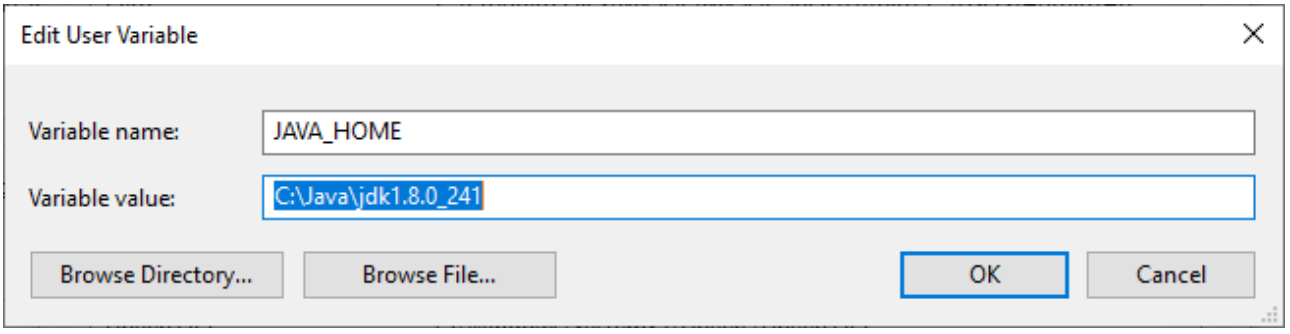
1. Install Java
2. Configure and install hadoop
3. Test hadoop installation
4. Create wordcount program
5. Input file to mapreduce
6. Display the output

I. JAVA Installation

1. Go to official Java Downloading page

https://www.oracle.com/java/technologies/javase-jre8-downloads.html

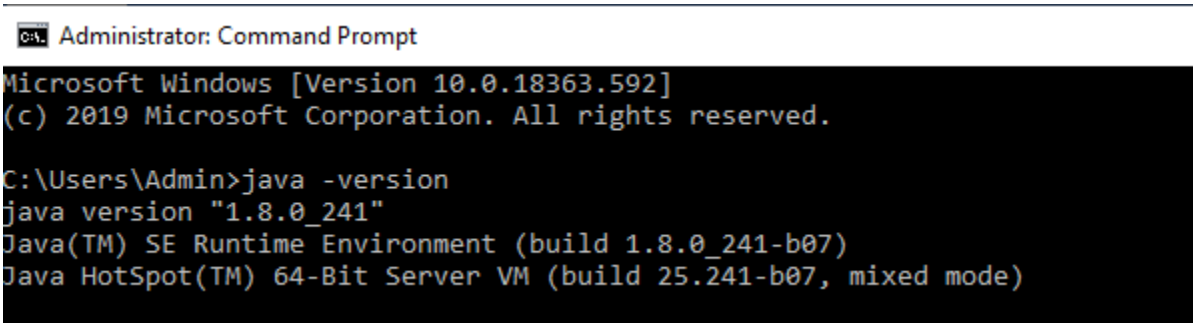
1. After downloading java, run the **jdk-8u241-windows-x64.exe** file
2. Follow the instructions and click next.
3. After finishing the installation it is need to set Java environment variable
4. Go to Start->Edit the System environment variable->Environment variable
5. Then Click new and enter variable name as “JAVA_HOME”
6. In the value field Enter the java path such as “C:\Java\jdk1.8.0_241”(Consider your installation folder)



8. Go to path and click edit then type “%JAVA_HOME%\bin”

9 . Then click Ok and Go to Command Prompt

10. Type “Java -version”. If it prints the installed version of java, now java successfully installed in your System.



II Configuring And Installing Hadoop

1. Download Hadoop 2.8.0 from http://archive.apache.org/dist/hadoop/core//hadoop-2.8.0/hadoop-2.8.0.tar.gz)

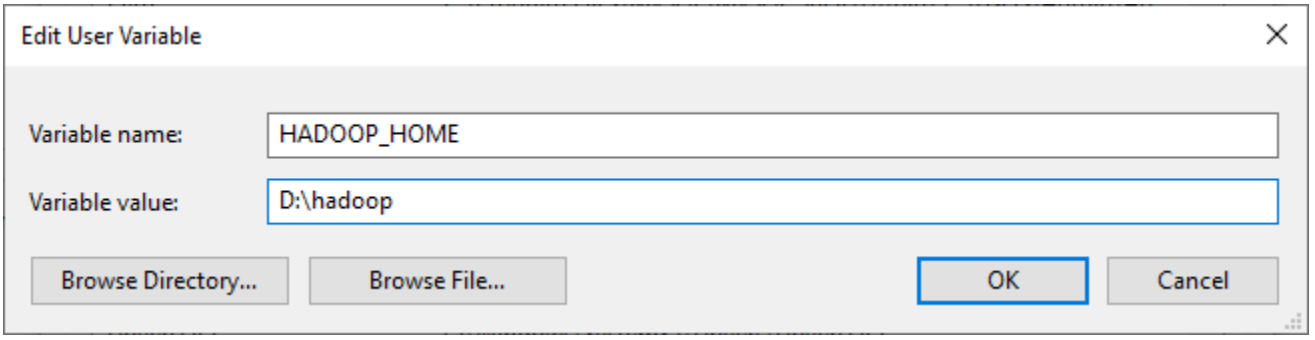
2. Extract the tar file (in my case I used **7-zip** to extract the file and I stored the extracted file in the **D:\hadoop**)

3. After finishing the extraction it is need to set Hadoop environment variable

4. Go to Start->Edit the System environment variable->Environment variable

5. Then Click new and enter variable name as “HADOOP_HOME”

6. In the value field Enter the java path such as “D:\hadoop”(Consider your installation folder)



7. Go to path and click edit then type “%HADOOP_HOME%\bin”

8. Now we have to configure the hadoop.

9. Go to D:/hadoop/etc/hadoop/.. folder, find the below mentioned files and paste the following.

i. core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

</configuration>

ii. Rename "**mapred-site.xml.template**" to "**mapred-site.xml**" and edit this fileD:/Hadoop/etc/hadoop/mapred-site.xml, paste below xml paragraph and save this file.

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

iii. Create folder "data" under "D:\Hadoop"

- Create folder "datanode" under "D:\Hadoop\data"
- Create folder "namenode" under "D:\Hadoop\data" data

iv. hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property

<name>dfs.namenode.name.dir</name>

<value>D:\hadoop\data\namenode</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>D:\hadoop\data\datanode</value>

</property>

</configuration>

v. yarn-site.xml

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

</configuration>

vi. Edit file D:\Hadoop\etc\hadoop**hadoop-env**.cmd by closing the command line "JAVA_HOME=%JAVA_HOME%" instead of set "JAVA_HOME= C:\Java\jdk1.8.0_241" (if your java file in Program Files the instead of give **Progra~1** otherwise you will get JAVA_HOME incorrectly set error)

vii. Download file Hadoop Configuration.zip **https://github.com/Prithiviraj2503/hadoop-installation-windows**

viii. Delete file bin on D:\Hadoop\bin and replace it by the bin file of Downloaded configuration file (from Hadoop Configuration.zip).

ix. Open cmd and typing command "**hdfs namenode –format**" .You will see through command prompt which tasks are processing, after competeation you will get a massage like namenode format succesfully and shutdown message

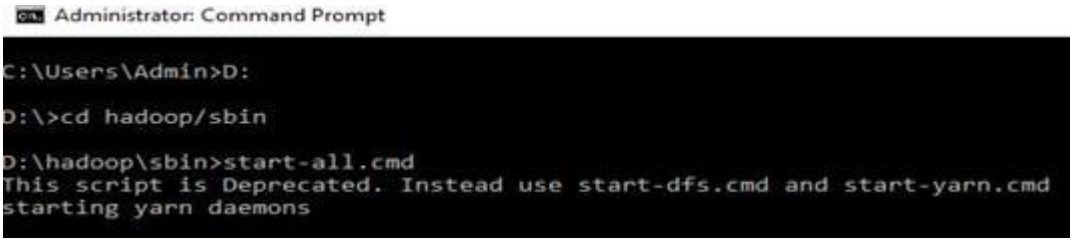
hdfs namenode –format

III. Testing Hadoop Installation

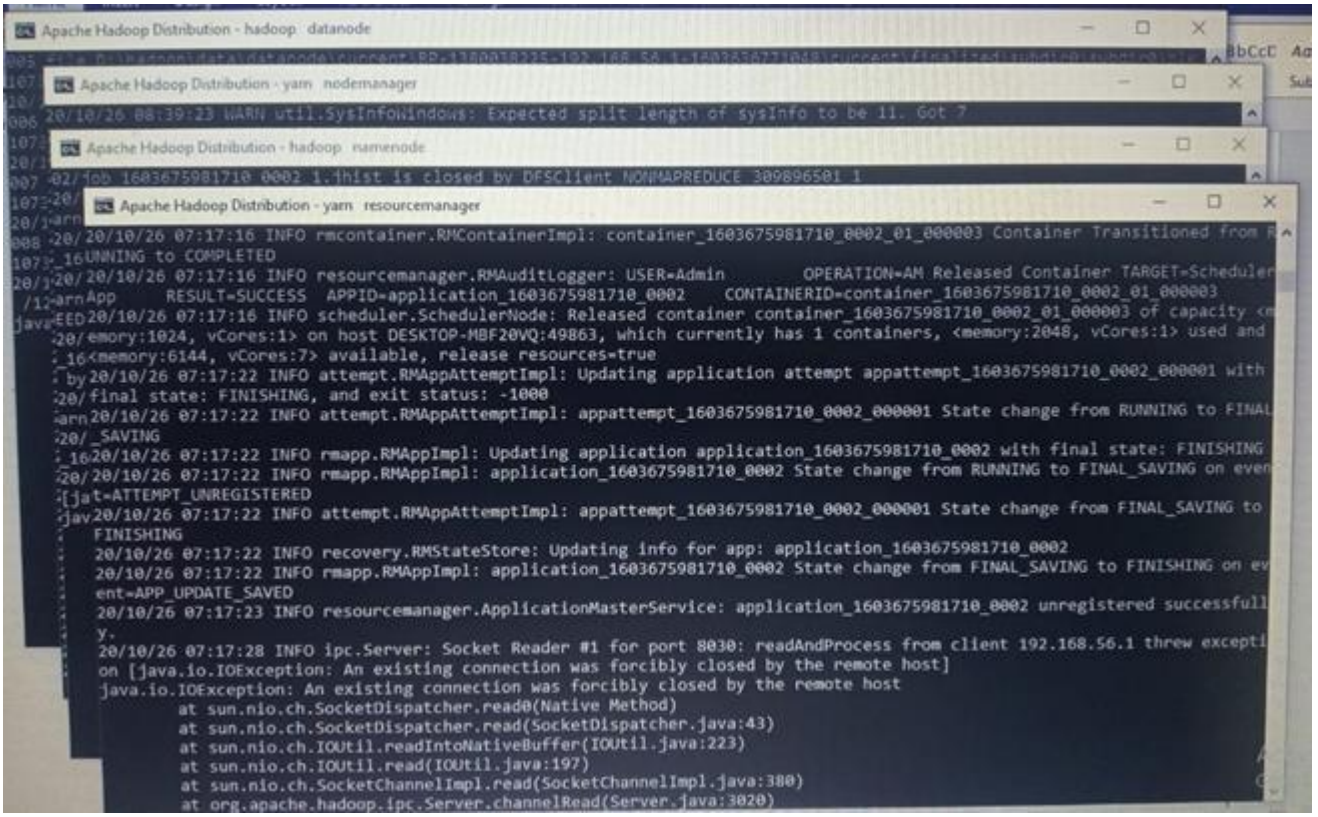
1. Open Cmd and type the following “Hadoop -version”

```
C:\Users\Admin>hadoop -version
java version "1.8.0_241"
Java(TM) SE Runtime Environment (build 1.8.0_241-b07)
Java HotSpot(TM) 64-Bit Server VM (build 25.241-b07, mixed mode)
```

2. To start the hadoop locate to “D:\hadoop\sbin” via command prompt and press **start-all.cmd**



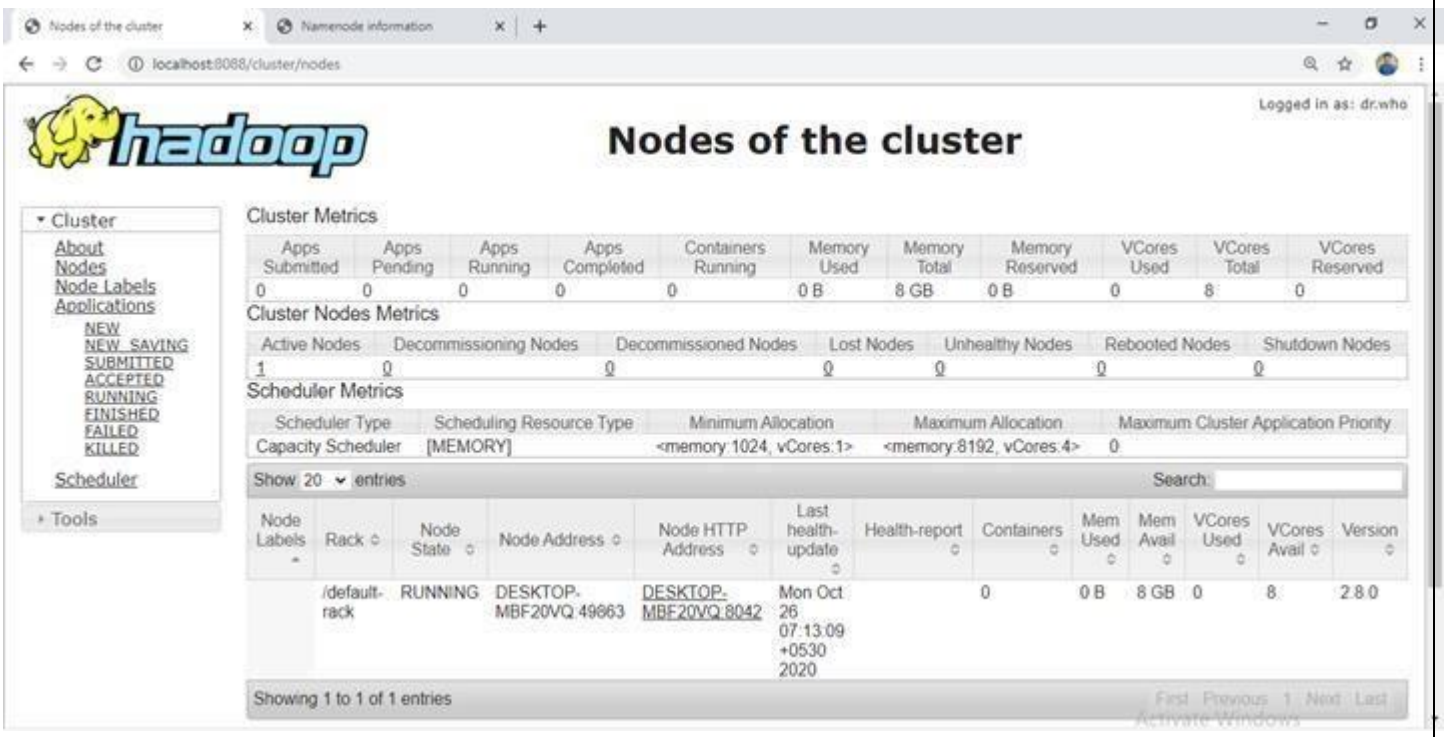
Now, you can see the namenode, datanode and yarn engines getting start,



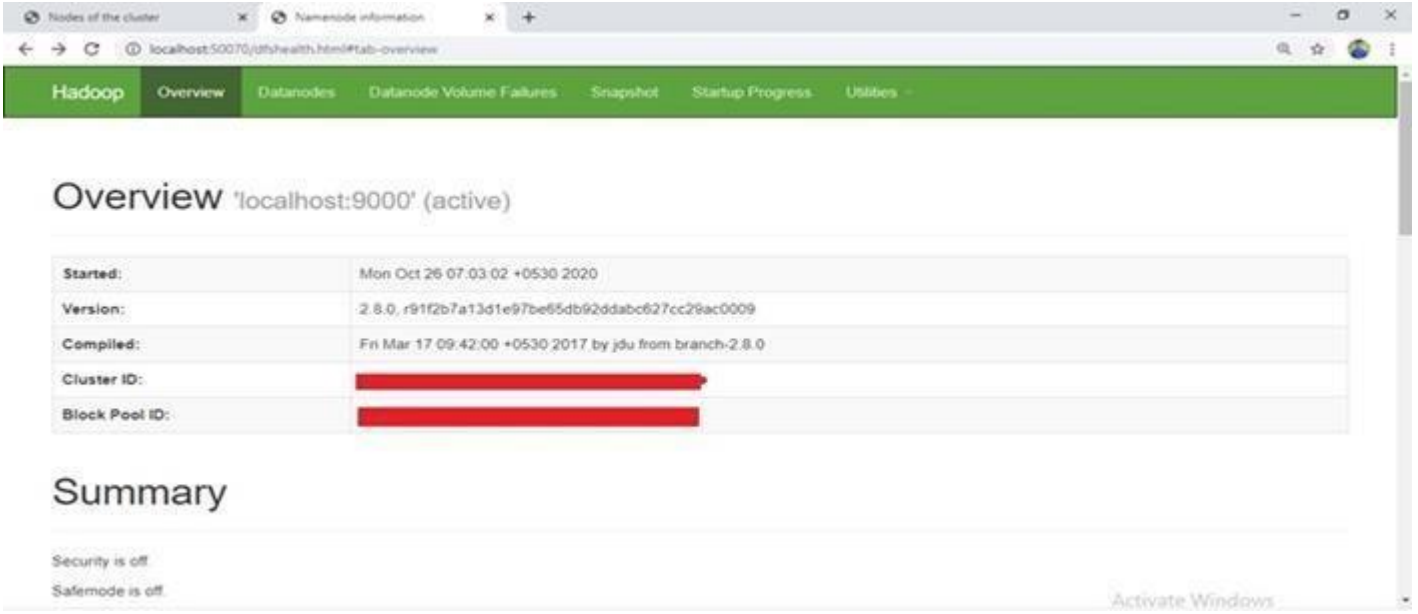
3. Now type “jps”. JPS (Java Virtual Machine Process Status Tool) is a command is used to check all the Hadoop daemons like NameNode, DataNode, ResourceManager, NodeManager etc.

```
D:\hadoop\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

D:\hadoop\sbin>jps
5296 NameNode
2372 Jps
9192 ResourceManager
10140 NodeManager
9420 DataNode
```

4. Open: **http://localhost:8088** in any browser



5. Open: **http://localhost:50070** in any browser



Now hadoop succesfully installed in your System.

IV. Simple WordCount Program

1) After successful hadoop installation we need to create an directory in the hadoop file system

2) Start the hadoop via command prompt **$ start-all.cmd**

3) By using **$jps** command Ensure hadoop nodes are running

4) To create a directory, use: **$ hadoop fs –mkdir /inputdir**

5) To input a file within a directory, use: **$ hadoop fs –put D:/input_file.txt/inputdir**

6) To ensure wether your file succesfully imported, use: **$ hadoop fs –ls /inputdir/**

7) To view the content of the file, use: **$ hadoop dfs –cat /inputdir/input_file.txt**

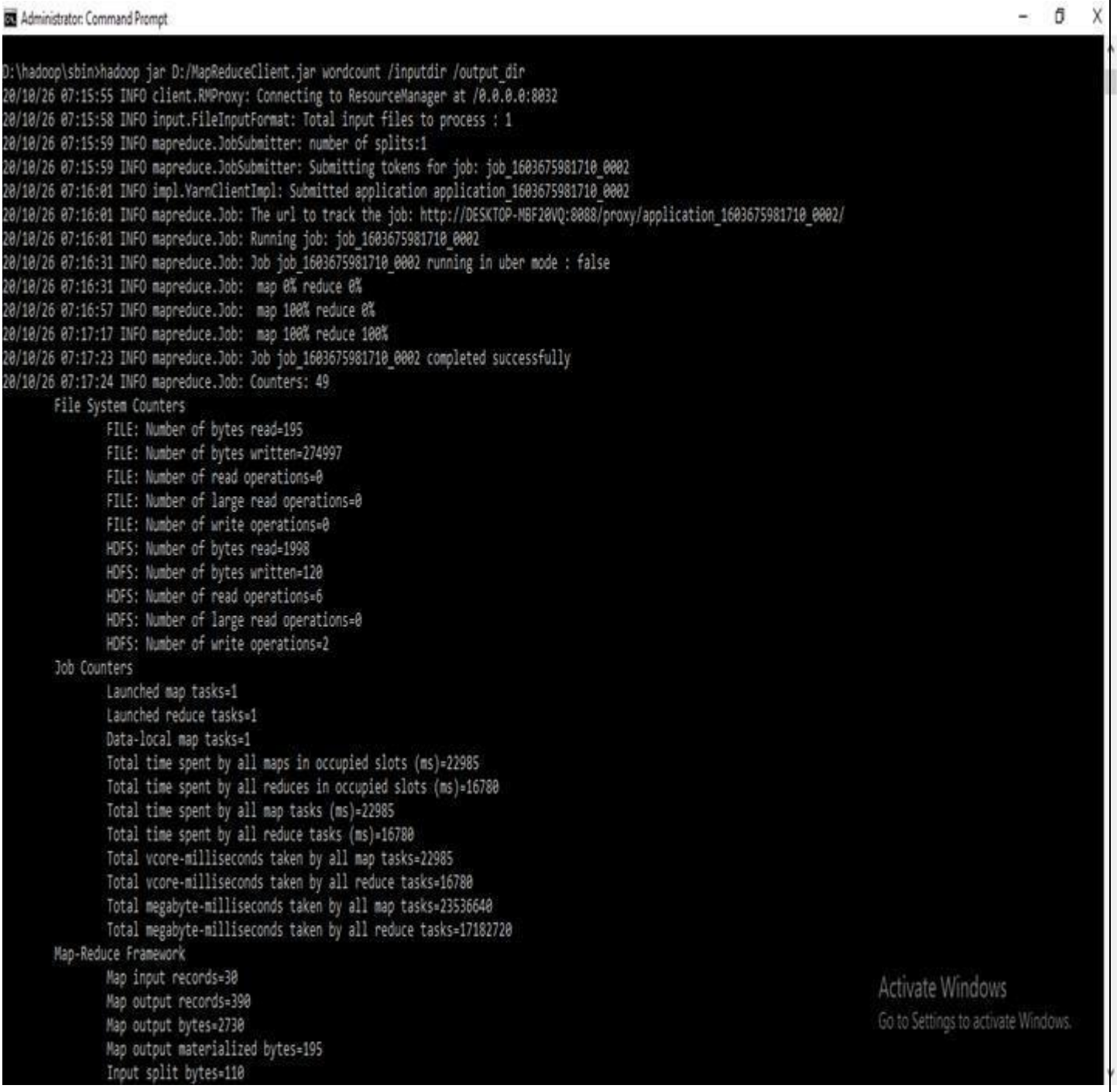
Link for input file : **https://github.com/Prithiviraj2503/hadoop-installation-windows**

Administrator: Command Prompt

```
D:\hadoop\sbin>hadoop fs -mkdir /inputdir

D:\hadoop\sbin>hadoop fs -put D:/input_file.txt /inputdir

D:\hadoop\sbin>hadoop fs -ls /inputdir/
Found 1 items
-rw-r--r--   1 Admin supergroup       1888 2020-10-26 07:10 /inputdir/input_file.txt

D:\hadoop\sbin>hadoop dfs -cat /inputdir/input_file.txt
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
23   23   27  43   24   25   26   26   26   26   25   26  25
26   27   28  28   28   30   31   31   31   30   30   30  29
31   32   32  32   33   34   35   36   36   34   34   34  34
39   38   39  39   39   41   42   43   40   39   38   38  40
38   39   39  39   39   41   41   41   28   40   39   39  45
23   23   27  43   24   25   26   26   26   26   25   26  25
26   27   28  28   28   30   31   31   31   30   30   30  29
31   32   32  32   33   34   35   36   36   34   34   34  34
39   38   39  39   39   41   42   43   40   39   38   38  40
38   39   39  39   39   41   41   41   28   40   39   39  45
23   23   27  43   24   25   26   26   26   26   25   26  25
26   27   28  28   28   30   31   31   31   30   30   30  29
31   32   32  32   33   34   35   36   36   34   34   34  34
39   38   39  39   39   41   42   43   40   39   38   38  40
38   39   39  39   39   41   41   41   28   40   39   39  45
23   23   27  43   24   25   26   26   26   26   25   26  25
26   27   28  28   28   30   31   31   31   30   30   30  29
31   32   32  32   33   34   35   36   36   34   34   34  34
39   38   39  39   39   41   42   43   40   39   38   38  40
38   39   39  39   39   41   41   41   28   40   39   39  45
23   23   27  43   24   25   26   26   26   26   25   26  25
26   27   28  28   28   30   31   31   31   30   30   30  29
31   32   32  32   33   34   35   36   36   34   34   34  34
39   38   39  39   39   41   42   43   40   39   38   38  40
38   39   39  39   39   41   41   41   28   40   39   39  45
23   23   27  43   24   25   26   26   26   26   25   26  25
26   27   28  28   28   30   31   31   31   30   30   30  29
31   32   32  32   33   34   35   36   36   34   34   34  34
39   38   39  39   39   41   42   43   40   39   38   38  40
38   39   39  39   39   41   41   41   28   40   39   39  45
D:\hadoop\sbin>hadoop jar D:/MapReduceClient.jar wordcount /input_dir /output_dir
20/10/26 07:15:19 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
20/10/26 07:15:22 INFO mapreduce.JobSubmitter: Cleaning up the staging area /tmp/hadoop-yarn/staging/Adm
```

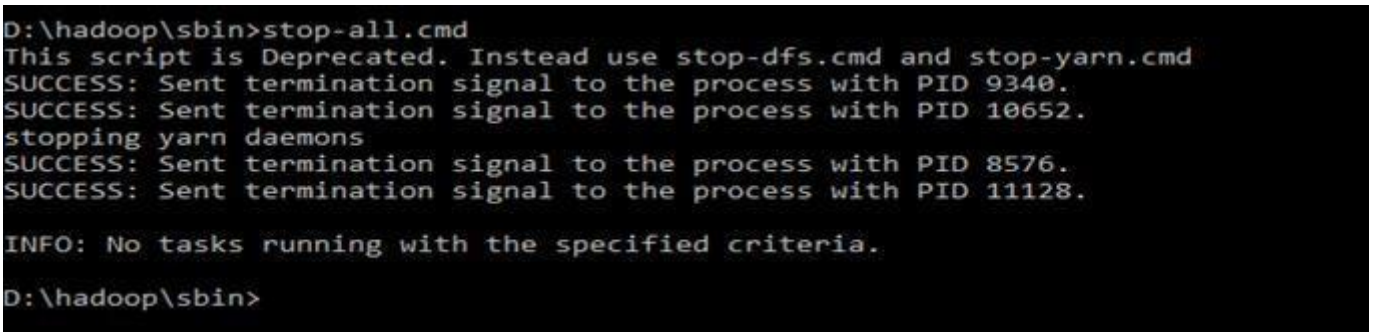
8) Now appy mapreduce program to the input file. We have a **mapReduceClient.jar** which contain java mapper and reducer programs. After applying the jar file you can see the task performed in the mapreduce phase.All the resuts of completed tasks will be printed in the command prompt.

Link for mapReduceClient.jar : **https://github.com/Prithiviraj2503/hadoop-installation-windows**



9) After completed the mapreduce tasks the output will be stored in the **output_dir** directory To see the output, use: **$ hadoop dfs –cat /output_dir**/

10) To stop the hadoop type **$stop-all.cmd**



Now the hadoop single node cluster was installed succesfully and the simple word count program were executed succesfully in your windows system.

RESULT:

Installation of hadoop and executed the simple word count program .

EXP NO :9 DATE:	CREATING AND EXECUTING YOUR FIRST CONTAINER USING DOCKER

AIM:

Creating and executing your first container using docker

PROCEDURE:

Containers are just a process (or a group of processes) running in isolation. Isolation is achieved via linux namespaces, control groups (cgroups), seccomp and SELinux. Note that linux namespaces and control groups are built into the linux kernel! Other than the linux kernel itself, there is nothing special about containers.

What makes containers useful is the tooling that surrounds it. For these labs, we will be using Docker, which has been a widely adopted tool for using containers to build applications. Docker provides developers and operators with a friendly interface to build, ship and run containers on any environment with a Docker engine. Because Docker client requires a Docker engine, an alternative is to use Podman, which is a deamonless container engine to develop, manage and run OCI containers and is able to run containers as root or in rootless mode. For those reasons, we recommend Podman but because of adoption, this lab still uses Docker.

The first part of this lab, we will run our first container, and learn how to inspect it. We will be able to witness the namespace isolation that we acquire from the linux kernel.

After we run our first container, we will dive into other uses of containers. You can find many examples of these on the Docker Store, and we will run several different types of containers on the same host. This will allow us to see the benefit of isolation- where we can run multiple containers on the same host without conflicts.

We will be using a few Docker commands in this lab. For full documentation on available commands check out the official documentation.

PREREQUISITES

Completed Lab 0: You must have access to a docker client, either on localhost, use a terminal from `Theia - Cloud IDE` at https://labs.cognitiveclass.ai/tools/theiadocker or be using Play with Docker for example.

Get Started¶

Run `docker -h`,

`$` docker -h
Flag shorthand -h has been deprecated, please use --help

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

...

Management Commands:
builder Manage builds

```
  config      Manage Docker configs
  container   Manage containers
  engine      Manage the docker engine
  image       Manage images
  network     Manage networks
  node        Manage Swarm nodes
  plugin      Manage plugins
  secret      Manage Docker secrets
  service     Manage services
  stack       Manage Docker stacks
  swarm       Manage Swarm
  system      Manage Docker
  trust       Manage trust on Docker images
  volume      Manage volumes
```

The Docker command line can be used to manage several features of the Docker Engine. In this lab, we will mainly focus on the `container` command.

If `podman` is installed, you can run the alternative command for comparison.

sudo podman -h

You can additionally review the version of your Docker installation,

docker version

```
Client:
  Version:      19.03.6
  ...

Server: Docker Engine - Community
  Engine
    Version:    19.03.5
    ...
```

sudo podman version --events-backend=none

```
Version:      2.1.1
API Version:  2.0.0
Go Version:  go1.15.2
Built:        Thu Jan  1 00:00:00 1970
OS/Arch:      linux/amd64
```

Step 1: Run your first container¶

We are going to use the Docker CLI to run our first container.

1. Open a terminal on your local computer
2. Run `docker container run -t ubuntu top`

   Use the `docker container run` command to run a container with the ubuntu image using the `top` command. The `-t` flags allocate a pseudo-TTY which we need for the `top` to work correctly.

   ```
   $ docker container run -it ubuntu top
   Unable to find image 'ubuntu:latest' locally
   latest: Pulling from library/ubuntu
   aafe6b5e13de: Pull complete
   0a2b43a72660: Pull complete
   ```

18bdd1e546d2: Pull complete
8198342c3e05: Pull complete
f56970a44fd4: Pull complete
Digest: sha256:f3a61450ae43896c4332bda5e78b453f4a93179045f20c8181043b26b5e79028
Status: Downloaded newer image for ubuntu:latest

The `docker run` command will result first in a `docker pull` to download the ubuntu image onto your host. Once it is downloaded, it will start the container. The output for the running container should look like this:

top - 20:32:46 up 3 days, 17:40, 0 users, load average: 0.00, 0.01, 0.00
Tasks: 1 total, 1 running, 0 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.1 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 2046768 total, 173308 free, 117248 used, 1756212 buff/cache
KiB Swap: 1048572 total, 1048572 free, 0 used. 1548356 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
1 root 20 0 36636 3072 2640 R 0.3 0.2 0:00.04 top

3. Inspect the container with `docker container exec`

The `docker container exec` command is a way to "enter" a running container's namespaces with a new process.

Open a new terminal. On cognitiveclass.ai, select `Terminal` > `New Terminal`.

Using play-with-docker.com, to open a new terminal connected to node1, click "Add New Instance" on the lefthand side, then ssh from node2 into node1 using the IP that is listed by 'node1 '. For example:

[node2] (local) root@192.168.0.17 ~
$ ssh 192.168.0.18
[node1] (local) root@192.168.0.18 ~
$

In the new terminal, use the `docker container ls` command to get the ID of the running container you just created.

$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
b3ad2a23fab3 ubuntu "top" 29 minutes ago Up 29 minutes goofy_nobel

Then use that id to run `bash` inside that container using the `docker container exec` command. Since we are using bash and want to interact with this container from our terminal, use `-it` flags to run using interactive mode while allocating a psuedo-terminal.

$ docker container exec -it b3ad2a23fab3 bash
root@b3ad2a23fab3:/#

From the same termina, run `ps -ef` to inspect the running processes.

root@b3ad2a23fab3:/# ps -ef
UID PID PPID C STIME TTY TIME CMD
root 1 0 0 20:34 ? 00:00:00 top
root 17 0 0 21:06 ? 00:00:00 bash
root 27 17 0 21:14 ? 00:00:00 ps -ef

For comparison, exit the container, and run `ps -ef` or `top` on the host. These commands will work on linux or mac. For windows, you can inspect the running processes using `tasklist`.

root@b3ad2a23fab3:/# exit
exit
$ ps -ef

Lots of processes!

4. Clean up the container running the `top` processes by typing: `<ctrl>-c`, list all containers and remove the containers by their id.
5. docker ps -a
6.
7. docker rm <CONTAINER ID>

Step 2: Run Multiple Containers¶

1. Explore the Docker Hub

   The Docker Hub is the public central registry for Docker images, which contains community and official images.

   . Run an Nginx server

   Let's run a container using the official Nginx image from the Docker Hub.

   $ docker container run --detach --publish 8080:80 --name nginx nginx
   Unable to find image 'nginx:latest' locally
   latest: Pulling from library/nginx
   36a46ebd5019: Pull complete
   57168433389f: Pull complete
   332ec8285c50: Pull complete
   Digest: sha256:c15f1fb8fd55c60c72f940a76da76a5fccce2fefa0dd9b17967b9e40b0355316
   Status: Downloaded newer image for nginx:latest
   5e1bf0e6b926bd73a66f98b3cbe23d04189c16a43d55dd46b8486359f6fdf048

   Nginx is a lightweight web server. You can access it on port 8080 on your localhost.

2. Access the nginx server on localhost:8080.
3. curl localhost:8080

   will return the HTML home page of Nginx,

   ```
   <!DOCTYPE html>
   <html>
   <head>
   <title>Welcome to nginx!</title>
   <style>
       body {
           width: 35em;
           margin: 0 auto;
           font-family: Tahoma, Verdana, Arial, sans-serif;
       }
   </style>
   </head>
   <body>
   <h1>Welcome to nginx!</h1>
   ```

   4. If you are using play-with-docker, look for the `8080` link near the top of the page, or if you run a Docker client with access to a local browser,

1. Run a mongo DB server

   Now, run a mongoDB server. We will use the official mongoDB image from the Docker Hub. Instead of using the `latest` tag (which is the default if no tag is specified), we will use a specific version of the mongo image: 4.4.

   $ docker container run --detach --publish 8081:27017 --name mongo mongo:4.4
   Unable to find image mongo:4.4 locally

4.4: Pulling from library/mongo
d13d02fa248d: Already exists
bc8e2652ce92: Pull complete
3cc856886986: Pull complete
c319e9ec4517: Pull complete
b4cbf8808f94: Pull complete
cb98a53e6676: Pull complete
f0485050cd8a: Pull complete
ac36cdc414b3: Pull complete
61814e3c487b: Pull complete
523a9f1da6b9: Pull complete
3b4beaef77a2: Pull complete
Digest: sha256:d13c897516e497e898c229e2467f4953314b63e48d4990d3215d876ef9d1fc7c
Status: Downloaded newer image for mongo:4.4
d8f614a4969fb1229f538e171850512f10f490cb1a96fca27e4aa89ac082eba5

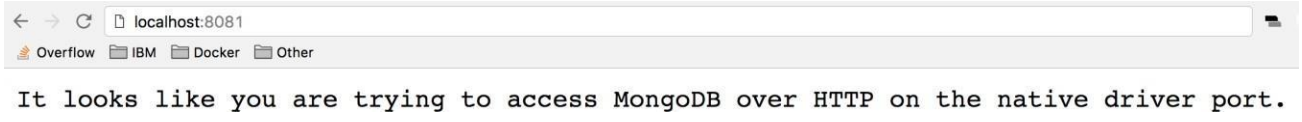
Again, since this is the first time we are running a mongo container, we will pull down the mongo image from the Docker Store. We are using the `--publish` flag to expose the 27017 mongo port on our host. We have to use a port other than 8080 for the host mapping, since that port is already exposed on our host. Again refer to the official docs on the Docker Hub to get more details about using the mongo image.

2. Access localhost:8081 to see some output from mongo.
3. curl localhost:8081

   which will return a warning from MongoDB,

   It looks like you are trying to access MongoDB over HTTP on the native driver port.

4. If you are using play-with-docker, look for the `8080` link near the top of the page.



5. Check your running containers with `docker container ls`
6. $ docker container ls
7. CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8. d6777df89fea nginx "nginx -g 'daemon ..." Less than a second ago Up 2 seconds 0.0.0.0:8080->80/tcp nginx
9. ead80a0db505 mongo "docker-entrypoint..." 17 seconds ago Up 19 seconds 0.0.0.0:8081->27017/tcp mongo
10. af549dccd5cf ubuntu "top" 5 minutes ago Up 5 minutes priceless_kepler

Step 3: Clean Up¶

Completing this lab results in a bunch of running containers on your host. Let's clean these up.

1. First get a list of the containers running using `docker container ls`.
2. $ docker container ls
3. CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
4. d6777df89fea nginx "nginx -g 'daemon ..." 3 minutes ago Up 3 minutes 0.0.0.0:8080->80/tcp nginx
5. ead80a0db505 mongo "docker-entrypoint..." 3 minutes ago Up 3 minutes 0.0.0.0:8081->27017/tcp mongo

6. af549dccd5cf ubuntu "top" 8 minutes ago Up 8 minutes priceless_kepler
7. Next, run `docker container stop [container id]` for each container in the list. You can also use the names of the containers that you specified before.
8. $ docker container stop d67 ead af5
9. d67
10. ead
11. af5

    Note: You only have to reference enough digits of the ID to be unique. Three digits is almost always enough.

12. Remove the stopped containers

    `docker system prune` is a really handy command to clean up your system. It will remove any stopped containers, unused volumes and networks, and dangling images.

    $ docker system prune
    WARNING! This will remove:
    - all stopped containers
    - all volumes not used by at least one container
    - all networks not used by at least one container
    - all dangling images

    Are you sure you want to continue? [y/N] y
    Deleted Containers:

RESULT:

Thus the docker was created and executed successfully.