**EXERCISE 1:**

**Installing Guest OS on Virtual Machine**

**PROCEDURAL STEPS**

1. Download and Install Oracle Virtual Box latest version & Extension Package.

**Link 1**: <https://www.virtualbox.org/wiki/Downloads>

**Link 2:** <https://www.oracle.com/in/virtualization/technologies/vm/downloads/virtualboxdownloads.html>

1. Download Ubuntu 14.4 OVA (Open Virtual Appliance)

Link: https://www.osboxes.org/ubuntu/

1. The files are downloaded in Local Machine ->Click the Oracle VM VirtualBox 6.0.8 & Setup Wizard Move to run time environment (Open Terminal)
2. Custom Setup -> Check VirtualBox Application
3. Custom Setup -> Select the features to be installed
4. Warning: Network Interfaces -> Click „Yes‟ & Proceed to install
5. Ready to install -> Click „Install‟
6. Oracle VM VirtualBox 6.1.22 installing
7. To install device software -> Click „Install‟
8. Oracle VM VirtualBox 6.1.22 installation is complete -> Click „Finish‟
9. Import the Oracle VM Virtual Extension Pack into Oracle VirtualBox
10. Click „Install‟ Oracle VM Virtual Extension Pack
11. VirtualBox License -> Click „I Agree‟
12. Extension Pack installation successful -> Click „OK‟
13. To install Virtual Machine: File -> Import Appliance
14. Select Ubuntu 14.4 OVA from directory
15. Appliance Settings -> Choose „RAM‟ Size -> Click „Import‟
16. Importing the Virtual Disk Image
17. Guest OS „Ubuntu 14.4‟ is installed successfully and Click „Start‟ button to lunch the virtual machine
18. Login to Ubuntu 14.4 Login Details:

User name: Hadoop

Password: Test1234

**EXERCISE 2:**

**Installing C Complier on Guest OS**

**PROCEDURAL STEPS**

1. To download package information from all configured sources -> $ sudo apt-get update

2. To install C Compiler on Ubuntu 14.4 ◊ $ sudo apt-get install gcc

3. To install C++ Compiler on Ubuntu 14.4 ◊ $ sudo apt-get install g++

4. Create a file to write C program ◊ $ sudo gedit <>.c

To compile C program -> $ gcc <>.c

To run C program -> $ ./a.out

**EXERCISE 4:**

**Simulation of VM Scheduling Using CloudSim**

**PROCEDURAL STEP**

1. Download NetBeans and install it.

**Link** : <https://netbeans-ide.informer.com/versions/>

2. Download CloudSim 4.0 (jar and source code), extract and place it in respective folders.

**Link:** <https://github.com/Cloudslab/cloudsim/releases/tag/cloudsim-4.0>

3.Open NetBeans & Create a project “CloudSim”

Check “Use dedicated folder for Storing Libraries” -> Click “Finish”

4. Add jar file “cloudsim-4.0”

5. Add “cloudsim examples” in CloudSim Project

6. Run the CloudSim examples

7. Output for CloudSimExample2.java

**EXERCISE 5:**

**Installation of Single Node Hadoop Cluster**

**PROCEDURAL STEP**

**Step 1:** Download the following Packages

1. hadoop-3.2.2.tar.gz

**Link:** https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop3.2.2/hadoop-3.2.2.tar.gz

1. 7zip to unzip the tar.gz file

**Link:** <https://www.7-zip.org/download.html>

1. Java 8 (JDK-8U333 & JRE-8U333)

**Link:** https://www.oracle.com/java/technologies/javase/javase8u211-later-archivedownloads.html

1. Hadoop dll Files & hadoop-hdfs-3.2.2.jar

**Link:** <https://github.com/paranikumar/Hadoop-Dll-JAR-Files.git>

**Step 2:** Install / Extract the following Packages:

1. Uninstall the existing java and Install Java 8 in the location C:\JAVA. Before installing create two folders namely “jdk1.8.0\_333” and “jre1.8.0\_333” inside the “C:\JAVA” Folder.

2. Create a folder “hadoop-env” in D:\

3. Paste the downloaded “hadoop-3.2.2.tar.gz” into “d:\ hadoop-env”

4. Unzip the “hadoop-3.2.2.tar.gz” using 7zip (do the unzip function two times)

5. Copy the downloaded Hadoop dll files to the location “D:\hadoop-env\hadoop3.2.2\bin”

6. Copy the downloaded hadoop-hdfs-3.2.2.jar to “D:\hadoop-env\hadoop3.2.2\share\hadoop\hdfs”

7. Edit the JAVA\_HOME = C:\JAVA\jdk1.8.0\_333 in the path “D:\hadoopenv\hadoop-3.2.2\etc\hadoop\hadoop-env.cmd

**Step 3:** Set the path for Java and Hadoop in system environment variables To edit environment variables, go to Control Panel -> System and Security ->System (or) right-click “This PC” -> Properties (My Computer icon) and click on the “Advanced system settings” link.

When the “Advanced system settings” dialog appears, go to the “Advanced” tab and click on the “Environment variables” button located on the bottom of the dialog.

In the “Environment Variables” dialog, press the “New” button to add a new variable.

Now, Edit the PATH variable to add the Java and Hadoop binaries paths as shown in the following screenshots.

**Step 4:** Open Command Prompt as Administrator and run the following command:

**hadoop –version**

**Step 5:** Create the following folders

D:\hadoop-env\hadoop-3.2.2\data\dfs\namenode

* D:\hadoop-env\hadoop-3.2.2\data\dfs\datanode

**Step 6:** Configuring Hadoop cluster

There are four files to configure Hadoop cluster:

Location of the File: “D:\hadoop-env\hadoop-3.2.2\etc\hadoop”

**File Name: “hdfs-site.xml”**

Add the following properties within the <configuration></configuration> element:

<property>

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

<value>yarn</value>

<description>MapReduce framework name</description>

</property>

<value>1</value>

</property>

<property>

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

<value>file:///D:/hadoop-env/hadoop-3.2.2/data/dfs/namenode</value>

</property>

<property>

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

<value>file:///D:/hadoop-env/hadoop-3.2.2/data/dfs/datanode</value>

</property>

**File Name: “core-site.xml”**

Add the following properties within the <configuration></configuration> element:

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9820</value>

</property>

**File Name: “mapred-site.xml”**

Add the following properties within the <configuration></configuration> element:

<property>

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

<value>yarn</value>

<description>MapReduce framework name</description>

</property>

**File Name: “yarn-site.xml”**

Add the following properties within the <configuration></configuration> element:

<property>

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

<value>mapreduce\_shuffle</value>

<description>Yarn Node Manager Aux Service</description>

</property>

**Step 7:** Format the namenode

After finishing the configuration, format the name using the following command hdfs namenode –format

**Step 8:** Start the Hadoop nodes

Now navigate to the location “D:\hadoop-env\hadoop-3.2.2\sbin” in powershell and then run the following command to start the Hadoop nodes: .\start-dfs.cmd

Two command prompt windows will open (one for the namenode and one for the datanode) as follows:

Next, Start the Hadoop Yarn service using the following command: ./start-yarn.cmd

Next, Start the Hadoop Yarn service using the following command: ./start-yarn.cmd

To make sure that all services started successfully, Run the following command:

**jps**

It should display the following services:

14560 DataNode

4960 ResourceManager

5936 NameNode

768 NodeManager

14636 Jps

**EXERCISE 6:**

**Word Count Program using MapReduce**

**PROCEDURAL STEPS**

**Step 1**: Create a text file **“D:/data.txt”**

**Step 2**: Create a directory in HDFS, where to kept text file.

**hdfs dfs -mkdir /user**

**Step 3:** Upload the data.txt file on HDFS in the specific directory

**hdfs dfs -put D:/data.txt /user**

**Step 4:** List the files or directories in

**hdfs hdfs dfs -ls /user/**

**Step 5**: To view the content of the file “/user/data.txt”

**hdfs dfs -cat /user/data.txt**

**Step 6:** Run the jar file hadoop jar

**D:/hadoop-env/hadoop-3.2.2/share/hadoop/mapreduce/hadoopmapreduce-examples-3.2.2.jar wordcount /user /out**

**Step 7**: To view the output in “/out/\*”

**hadoop fs -cat /out/\***

**EXERCISE 6:**

**Create Virtual Machine in OpenStack Cloud Platform**

**PROCEDURAL STEPS**

**Step 1:** **Switch to “root” user**

user@vmub-hadoop1:~$ sudo su –

root@vmub-hadoop1:~# sudo apt-get update

**Step 2: Install “git” and logout from “root user”**

root@vmub-hadoop1:~# sudo apt-get install git

root@vmub-

hadoop1:~# exit

logout

user@vmub-hadoop1:~$

**Step 3: Create/Add a user “stack” with “no password”**

user@vmub-hadoop1:~$ sudo useradd -s /bin/bash -d /opt/stack -m stack

user@vmub-hadoop1:~$ echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack stack ALL=(ALL) NOPASSWD: ALL

**Step 4: Move to “stack” user**

user@vmub-hadoop1:~$ sudo su – stack

**Step 5:** **Clone the “devstack” from the specified link**

stack@vmub-hadoop1:~$ git clone <https://github.com/openstack-dev/devstack>

Cloning into 'devstack'...

remote: Enumerating objects: 49330, done.

remote: Counting objects: 100% (2556/2556), done.

remote: Compressing objects: 100% (836/836), done.

remote: Total 49330 (delta 1831), reused 2226 (delta 1709), pack-reused 46774

Receiving objects: 100% (49330/49330), 15.56 MiB | 2.33 MiB/s, done.

Resolving deltas: 100% (34462/34462), done

. Checking connectivity... done.

**Step 6:** **Move to “/devstack/samples”**

stack@vmub-hadoop1:~$ cd devstack

stack@vmub-hadoop1:~/devstack$ls

clean.sh doc functions gate lib openrc roles stackrc tools

CONTRIBUTING.rst extras.d functions-common HACKING.rst LICENSE playbooks

run\_tests.sh stack.sh tox.ini

data files FUTURE.rst inc Makefile README.rst samples tests

unstack.sh

stack@vmub-hadoop1:~/devstack$ cd samples

stack@vmub-hadoop1:~/devstack/samples$ ls

local.conf local.sh

**Step 7: Copy “local.conf” file to “devstack”**

stack@vmub-hadoop1:~/devstack/samples$ cp local.conf ../

stack@vmub-hadoop1:~/devstack/samples$ cd ..

stack@vmub-hadoop1:~/devstack$ls

clean.sh doc functions gate lib Makefile README.rst samples

tests unstack.sh

CONTRIBUTING.rst extras.d functions-common HACKING.rst LICENSE openrc

roles stackrc tools

data files FUTURE.rst inc local.conf playbooks run\_tests.sh

stack.sh tox.ini

**Step 8: Open “local.conf” and edit the lines**

stack@vmub-hadoop1:~/devstack$ nano local.conf

ADMIN\_PASSWORD=p1

DATABASE\_PASSWORD=p1

RABBIT\_PASSWORD=p1

SERVICE\_PASSWORD=p1

HOST\_IP=10.0.2.4

FLOATING\_RANGE=10.0.2.224/27

**Step 9: Remove the following files to Lock the error**

stack@vmub-hadoop1:~/devstack$ sudo rm /var/lib/dpkg/lock

stack@vmub-hadoop1:~/devstack$ sudo rm /var/lib/apt/lists/lock

stack@vmub-hadoop1:~/devstack$ sudo rm /var/cache/apt/archives/lock

stack@vmub-hadoop1:~/devstack$ sudo rm -rf /var/lib/apt/lists/\*

**Step 10: To install stack**

stack@vmub-hadoop1:~/devstack$ FORCE=yes./stack.sh Accessing OpenStack on a web browser

To access OpenStack via a web browser browse your Ubuntu‟s IP address as shown.

https://server-ip/dashboard

This directs you to a login page as shown.