LAB CYCLE 2

Date:

EXPERIMENT 4 : Transfer files from one VM to another

Theory :

When working with multiple virtual machines, it is often necessary to transfer files between them for testing, development, or data sharing. File transfer between VMs can be achieved using different methods such as shared folders, Secure Copy Protocol (SCP), Secure File Transfer Protocol (SFTP), or simple network sharing. In VirtualBox, shared folders allow files to be accessible from both VMs without duplication. Using SCP or SFTP requires enabling network connectivity between VMs, such as bridged networking or host-only adapters. Learning how to transfer files between VMs helps in building practical skills for real-world tasks like deploying software, sharing datasets, and collaborative development in isolated environments.

Question :

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

Steps :

I. *Copy & Paste Using Shared Clipboard*
   a. *Install Guest Additions on both virtual machines.*
   b. *In Settings > General > Advanced, set Shared Clipboard to Bidirectional.*
   c. *Restart both VMs.*
   d. *Copy content from one VM, paste into the host OS clipboard, then paste from the host OS into the second VM.*

II. *Drag and Drop*
   a. *In Settings > General > Advanced, set Drag and Drop to Bidirectional.*
   b. *Restart both VMs.*
   c. *Drag files directly from one VM to the host OS, then drop them into the second VM.*

III. *Using Shared Folders*
   a. *Install Guest Additions on both VMs.*

b. *In Settings → Shared Folders, add a directory from the host OS and enable Auto-mount and Make Permanent.*
c. *Use this shared folder as a common buffer to copy files.*
d. *Files placed in the shared folder from any VM or the host OS are instantly visible to all.*

Note: Differences in VM clock settings can sometimes cause timestamp issues.

IV. *Network-Based Transfer (SCP/SFTP)*
a. *Ensure both VMs have network connectivity (Internal Network or Host-Only Adapter).*
b. *On the receiving VM, enable the SSH server:*
`sudo apt-get install openssh-server`
c. *On the sending VM, transfer files using SCP:*

`scp abc.txt username@<target_VM_IP>:/path/to/destination`

d. *Authenticate with the target VM's credentials to complete the transfer.*

V. *Advanced File Sharing (NFS, SSHFS, Samba)*
a. *Configure file system sharing using NFS (Linux), SSHFS, or Samba.*
b. *Mount the shared directory on the other VM for direct file access.*

Note:
When working with multiple virtual machines in a network, remember that :

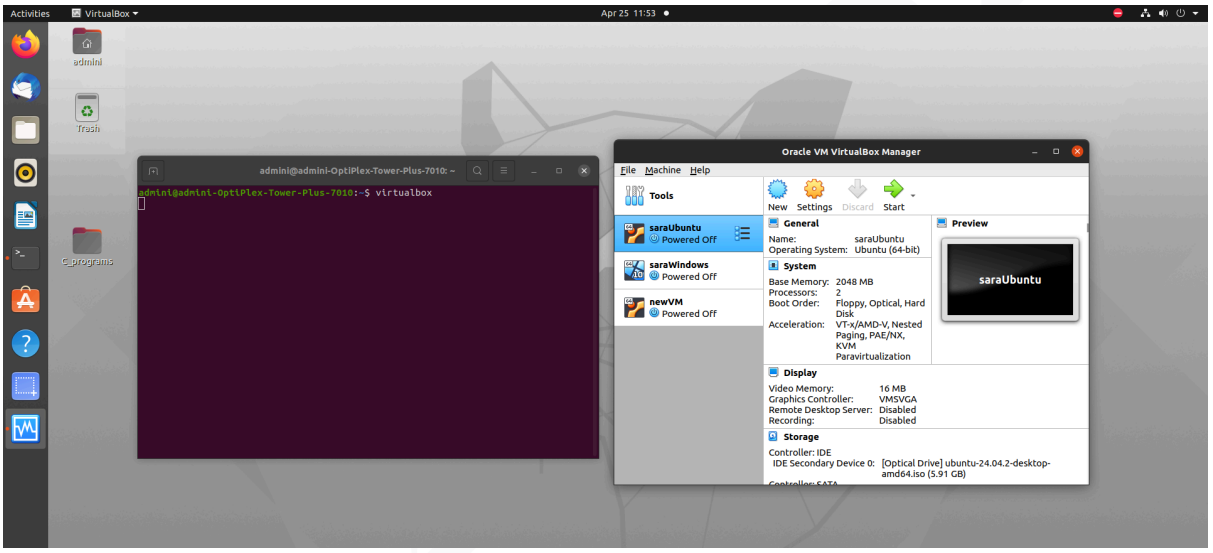
- *Each virtual machine runs its own operating system and behaves like a separate physical machine.*
- *Each virtual machine is an instance of a program owned by a specific user account on the host operating system, and therefore inherits that user's file access permissions.*

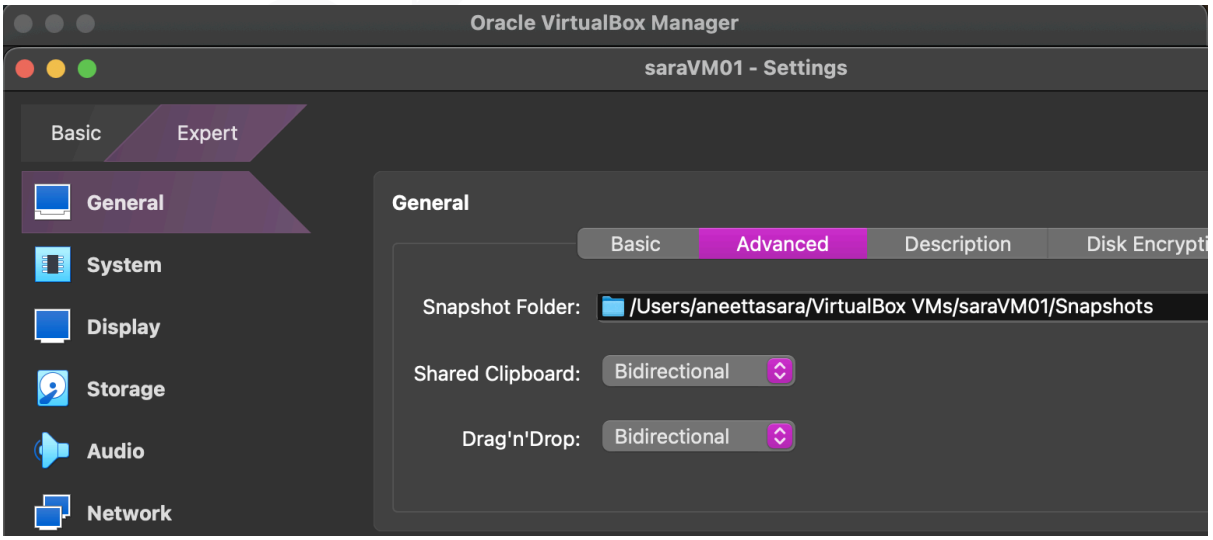
For example, suppose BVRITH and BVRITN are two different users on the host operating system. If neither user grants read/write/execute permissions to the other's directories, the virtual machines they run will also be restricted from accessing those directories. To allow file sharing between their VMs, you can :
Grant appropriate read/write/execute permissions to a specific directory, or
Choose a common directory where both users have full access rights.
For Windows-based virtual machines, enabling Drag & Drop can make file transfer more convenient. For Linux-based virtual machines, using Shared Folders is an efficient approach.

As you become more familiar with Linux, you may prefer using SSH tools for faster transfers. Generating SSH keys using: **`ssh-keygen`** *, will allow password-less authentication for copying files between machines. This also enables features like remote Bash auto-completion, improving workflow speed and efficiency.*

Conclusion :

This task helps to transfer files between virtual machines using different methods such as SCP and shared folders. I understood the importance of proper network configuration for file sharing and how VirtualBox features can simplify file exchange between isolated environments. This knowledge is valuable for collaborative work, testing, and simulating multi-system environments.





Date:

EXPERIMENT 5 : Launch Virtual Machine using OpenStack

Theory :

OpenStack is an open-source cloud computing platform for managing compute, storage, and networking resources in a data center. It allows users to launch and manage virtual machines (instances) using a web interface (Horizon dashboard) or command-line tools. TryStack was once an OpenStack-powered online demo platform for testing OpenStack's features, such as launching virtual machines (VMs). However, TryStack has not been active. DevStack is a lightweight way to install OpenStack on your local machine, mainly for development and testing. By using DevStack, we can simulate a real OpenStack cloud locally and practice creating and managing virtual machines without requiring external cloud access.

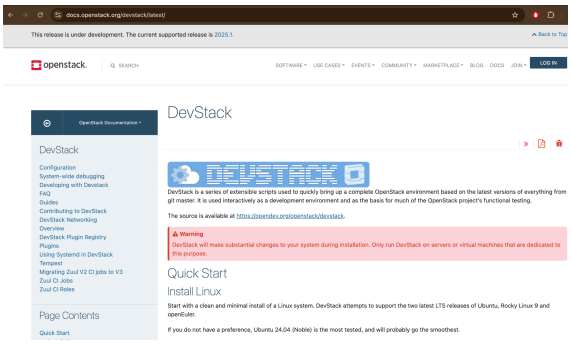
Question :

Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)

Steps :

I. *Prepare the Environment*
   a. *Install Ubuntu 20.04 or later on a dedicated system or a VirtualBox VM (minimum 4 GB RAM, 2 vCPUs, 20 GB disk).*
   b. *Update the package index :*
      ```
      sudo apt update && sudo apt upgrade
      ```
   c. *Install Git :*
      ```
      sudo apt install git
      ```

II. *Download DevStack*



Page No.:

a. *Clone the DevStack repository :*

```
git clone https://opendev.org/openstack/devstack
```

b. *Navigate to the DevStack directory :*

```
cd devstack
```

III. *Configure DevStack*

a. *Create a local.conf file containing credentials*

opendev.org/openstack/devstack/src/branch/master/samples/local....

```
26  # If the ``*_PASSWORD`` variables are not set here you will be prompted to enter
27  # values for them by ``stack.sh``and they will be added to ``local.conf``.
28  ADMIN_PASSWORD=nomoresecret
29  DATABASE_PASSWORD=stackdb
30  RABBIT_PASSWORD=stackqueue
31  SERVICE_PASSWORD=$ADMIN_PASSWORD
```

IV. *Install OpenStack via DevStack*

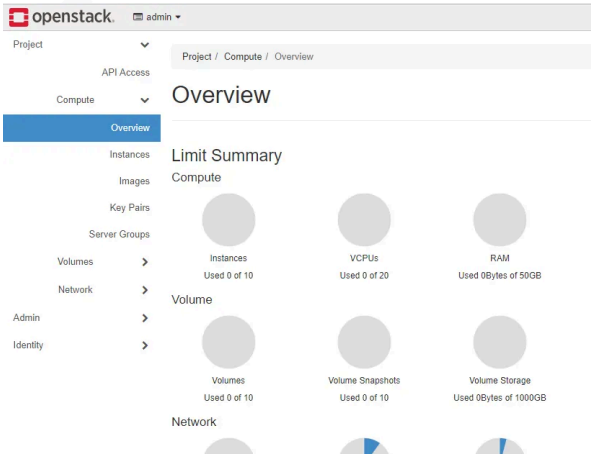
a. *Start the installation and wait until the script completes which may take 20 - 40 minutes :*

```
./stack.sh
```

V. *Access Horizon Dashboard*

a. *Once DevStack has finished, open a browser and go to http://localhost/dashboard.*

b. *Log in with the username admin and the password you set in the local.conf file.*



VI. *Launch a Virtual Machine in OpenStack*

a. *In Horizon, go to Project > Compute > Instances.*

b. *Click Launch Instance.*

c. *Fill in:*

i. *Instance Name (e.g., ECCTestVM)*

ii. *Image : Select an available OS image*

iii. *Flavor : Choose a size (e.g., m1.tiny).*

iv. *Network : Select the default network.*

d. *Click Launch.*
e. *Wait for the status to change to Active.*

VII. *Access the Instance*

a. *Use the Console tab in Horizon to log into the VM.*
b. *Or use SSH if network and key pairs are configured.*

Conclusion :

From this experiment, I learned how to install OpenStack locally using DevStack and use it to launch a virtual machine. I gained practical experience with the Horizon dashboard, understanding concepts like flavors, images, networks, and key pairs. This setup provided a real OpenStack environment for testing, without needing external cloud services, and helped me understand the cloud provisioning process end-to-end.

Date:

EXPERIMENT 6 : Install Hadoop Cluster and Run Word Count

Theory :

Hadoop is an open-source framework developed by the Apache Software Foundation for distributed storage and processing of large datasets. It uses a master-slave architecture, with HDFS (Hadoop Distributed File System) for storage and MapReduce for processing. In a single-node cluster setup, all Hadoop daemons (NameNode, DataNode, ResourceManager, NodeManager) run on a single machine, making it ideal for learning and testing. Running a WordCount program demonstrates the MapReduce processing model: the Map phase processes input data into key-value pairs, and the Reduce phase aggregates values for each key. Understanding Hadoop's setup and execution process is an essential skill in big data technologies.

Question :

Install Hadoop single node cluster and run simple applications like Word Count.

Steps :

PART A - *Java Installation*

I. *Open terminal and check Java version:*

```
java --version
```

If Java is not installed, type in terminal:

```
sudo apt update
sudo apt install default-jre
java --version
```

PART B - *Hadoop Installation*

II. *Download Hadoop from https://hadoop.apache.org/releases.html (latest stable release).*
Go to the latest version (As of 3.4.1), Click on binary and copy the first URL <https://dlcdn.apache.org.....

https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.4.1/hadoop-3.4.1.tar.gz

III. *Open terminal:*

```
wget https://dlcdn.apache.org/hadoop/common/hadoop-3.4.1/hadoop-3.4.1.tar.gz
tar -xvzf hadoop-3.4.1.tar.gz
mv hadoop-3.4.1 hadoop
sudo mv hadoop /usr/local/hadoop
```

IV. *Configure environment variables :*

```
nano ~/.bashrc
```

Copy and Paste the below content to the end of bash file

```
#Java
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/
#Hadoop
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

Save (Ctrl+X, then Y, then Enter) and reload :

```
source ~/.bashrc
```

V. *Close current terminal, Open new terminal and Verify installation :*

```
hadoop version
```

<u>PART C - *Hadoop Configuration*</u>

VI. *Open terminal and Edit Hadoop environment file:*

```
sudo nano /usr/local/hadoop/etc/hadoop/hadoop-env.sh
```

Copy and Paste the below content to the end of bash file

```
#Path to Java in Hadoop Configuration
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/
```

VII. *Open terminal:*

```
/usr/local/hadoop/bin/hadoop
```

OR

```
${HADOOP_HOME}/bin/hadoop
```

Output consisting of usage and commands means you have successfully configured Hadoop to run in stand-alone mode.

Configure a Hadoop environment by editing a set of configuration files

VIII. *Go to Hadoop Home via Terminal:*

```
cd ${HADOOP_HOME}
ls
```

IX. *Create data directories via Terminal:*

Create "data" folder inside Hadoop Home:

```
mkdir data
```

Create "datanode" , "namenode" folders inside ${HADOOP_HOME}/data:

```
mkdir datanode
mkdir namenode
```

X. *Go to etc folder to make the edit in config files:*

```
${HADOOP_HOME}/etc/hadoop
ls
```

1 => `sudo nano core-site.xml`

Copy and Paste the below content to the end of bash file

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

Ctrl + X and Press Y then press Enter

2 => `sudo nano mapred-site.xml`

Copy and Paste the below content to the end of bash file

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

Ctrl + X and Press Y then press Enter

3 => `sudo nano hdfs-site.xml`

Copy and Paste the below content to the end of bash file

```
<configuration>
```

```
            <property>
                <name>dfs.replication</name>
                <value>1</value>
            </property>
            <property>
                <name>dfs.namenode.name.dir</name>
                <value>${HADOOP_HOME}/data/namenode</value
            >
            </property>
            <property>
                <name>dfs.datanode.data.dir</name>
                <value>${HADOOP_HOME}/data/datanode</value
            >
            </property>
        </configuration>
```

Ctrl + X and Press Y then press Enter

4 => `sudo nano yarn-site.xml`

Copy and Paste the below content to the end of bash file

```
        <configuration>
            <property>
                <name>yarn.nodemanager.aux-services</name>
                <value>mapreduce_shuffle</value>
            </property>
            <property>
                <name>yarn.nodemanager.auxservices.mapredu
            ce.shuffle.class</name>
                <value>org.apache.hadoop.mapred.ShuffleHan
            dler</value>
            </property>
        </configuration>
```

Ctrl + X and Press Y then press Enter

PART D - *Format and Start Hadoop*

XI. *Close current terminal, Open new terminal to format NameNode :*

```
hdfs namenode -format
```

XII. *Go to Hadoop SBIN folder via Terminal to Start Hadoop services :*

```
cd ${HADOOP_HOME}/sbin
start-all.sh
```

Now, Type command to check/verify all the Hadoop daemons like NameNode, DataNode, ResourceManager, NodeManager etc. JPS (Java Virtual Machine Process Status Tool)

```
jps
```

XIII. *Open: http://localhost:8088 in any browser for ResourceManager*

XIV. *Open: http://localhost:50070 in any browser for NameNode*

<u>*PART E - Run WordCount Example*</u>

XV. *Create input directory in HDFS :*

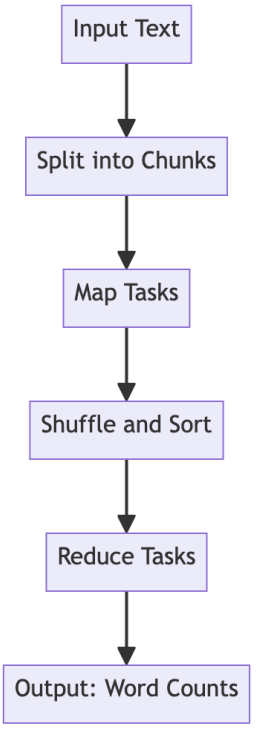
```
hdfs dfs -mkdir /input
```

XVI. *Upload a sample text file :*

```
echo "A teacher works tirelessly to help students succeed. A great teacher listens, understands, and encourages" > sample.txt
hdfs dfs -put sample.txt /input
```

XVII. *Run WordCount MapReduce example :*

```
hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar wordcount /input /output
```



XVIII. *View the output results*
(The output will display each unique word in your text along with its count) :

```
hdfs dfs -cat /output/part-r-00000
```

Output

teacher	2
understands,	1
encourages	1
and	1
great	1
students	1
A	2
works	1
to	1
succeed.	1
listens,	1
tirelessly	1
help	1

Conclusion :

This experiment taught me how to install and configure a Hadoop single-node cluster and execute a MapReduce application. I successfully ran the WordCount example, which processes text files and counts word occurrences in a distributed manner. This helped me understand the roles of HDFS, YARN, and MapReduce, as well as the workflow of big data processing in a Hadoop environment.