Ex.No: 6 TRANSFER THE FILES FROM ONE VIRTUAL MACHINE TO ANOTHER VIRTUAL MACHINE

Date:

AIM

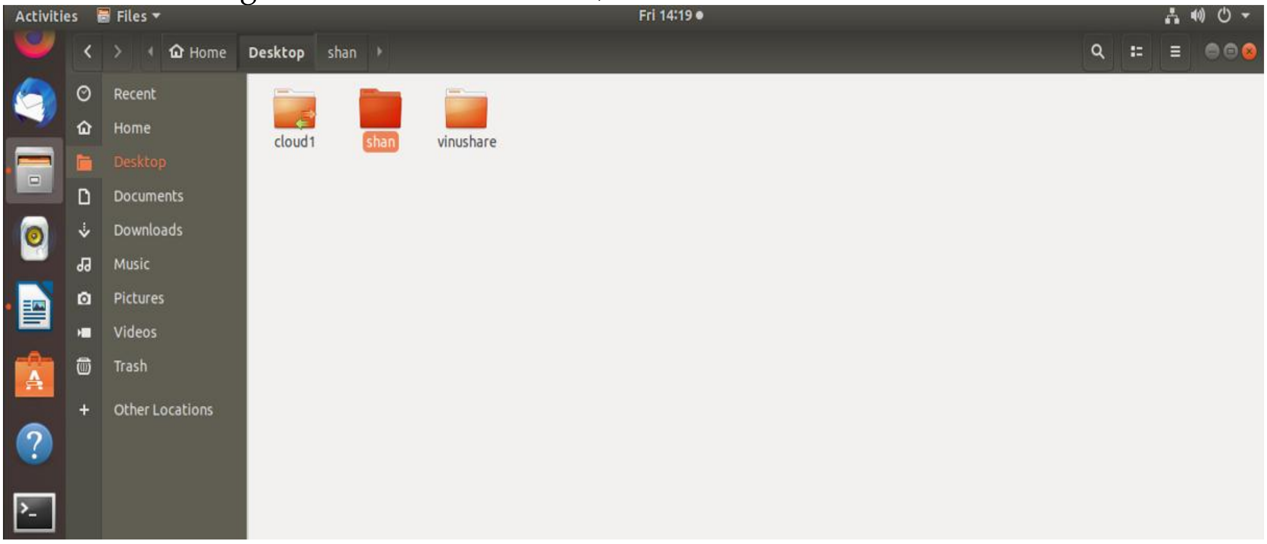
To find a procedure to transfer a file between Virtual machines using Shared folder in VMware Workstation and the host machine.

PRE-REQUISITES

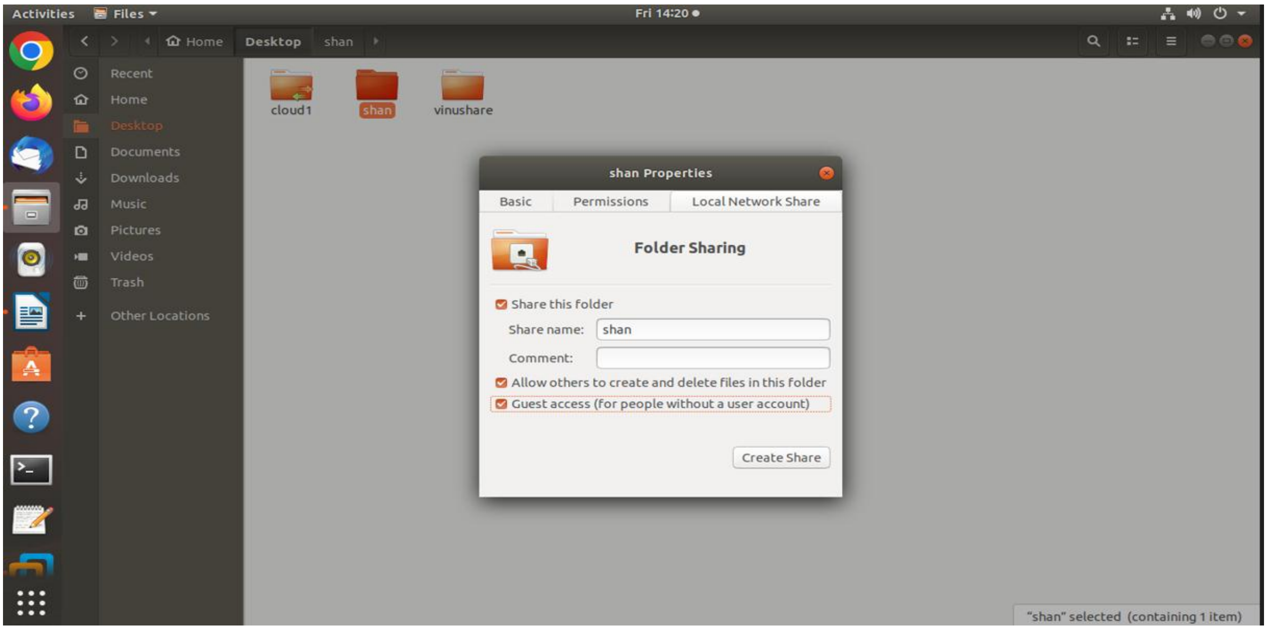
- VMware Workstation
- Host Machines,
- Text file.

PROCEDURE

- Before turning on the virtual machines, create a shared folder in the host OS.



- Choose the folder you want to share, right click and choose local network share or select properties and select the tab local network share.



shan Properties

Basic | Permissions | Local Network Share

Folder Sharing

☑ Share this folder

Share name: shan

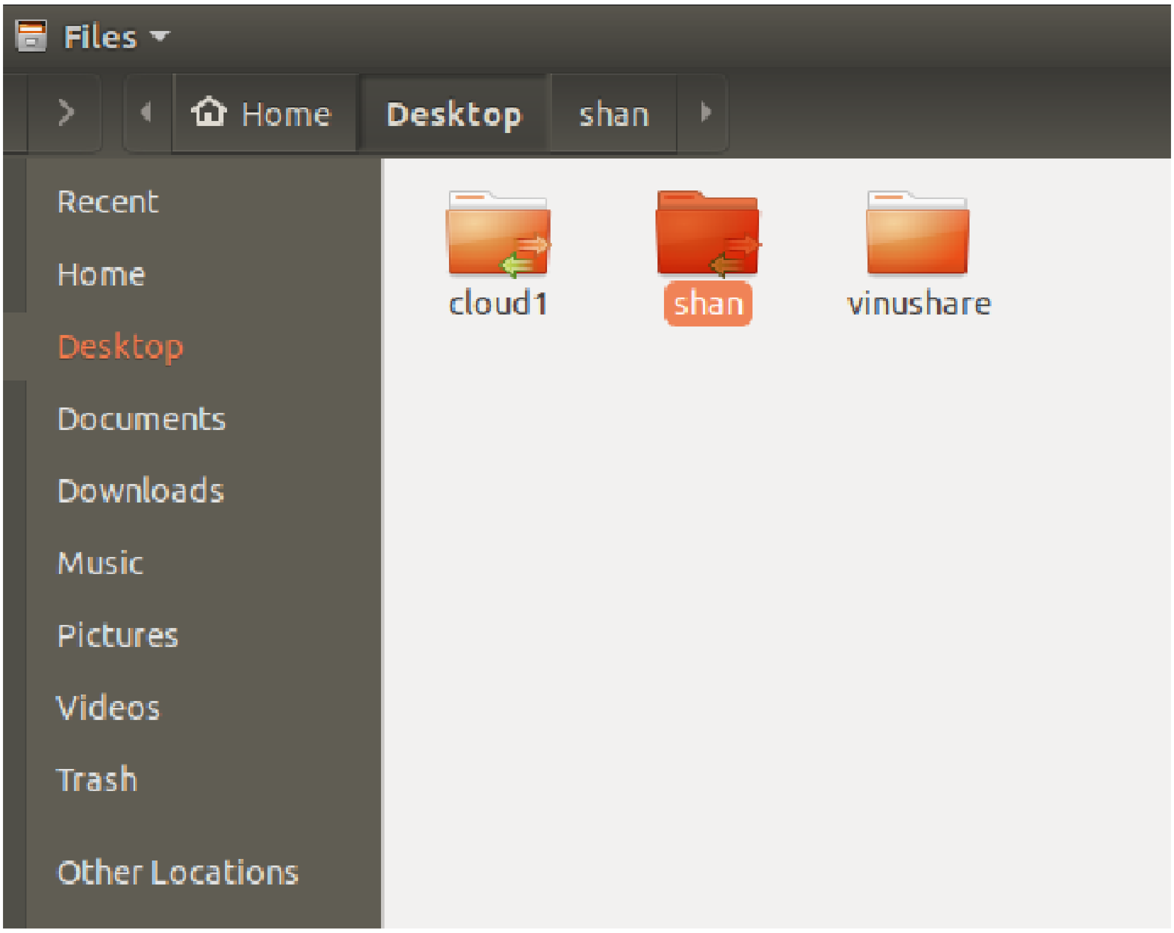
Comment:

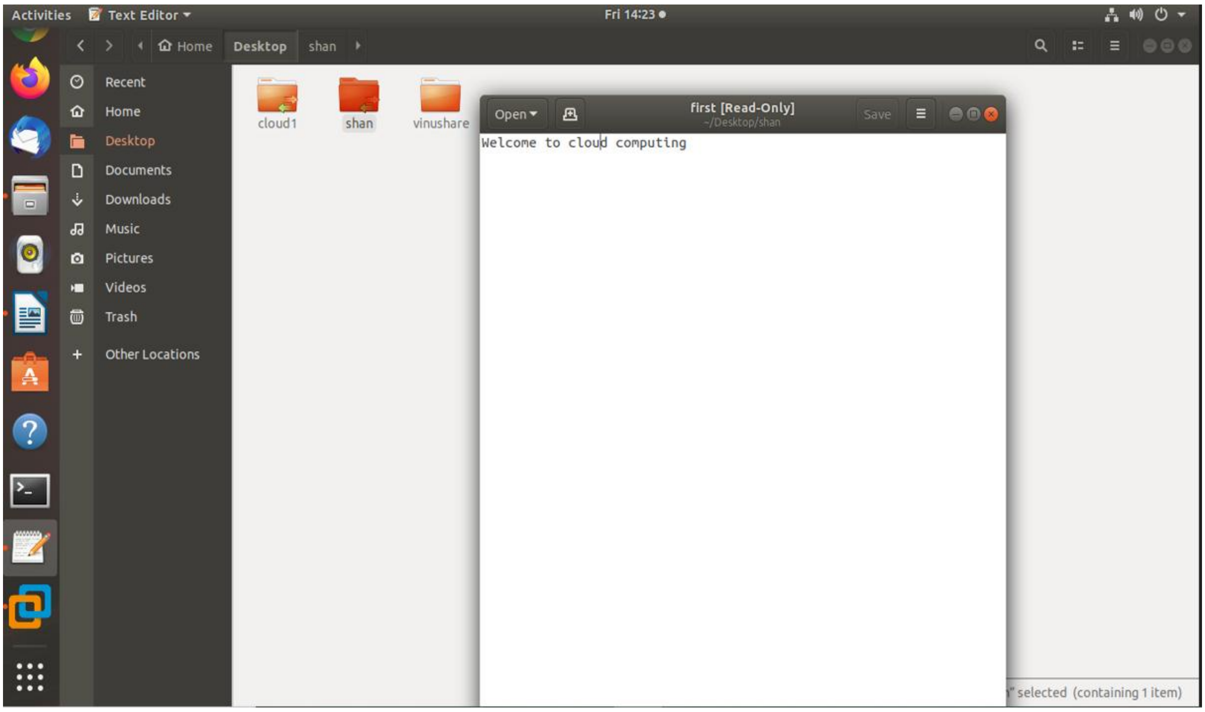
☑ Allow others to create and delete files in this folder

☑ Guest access (for people without a user account)

Create Share

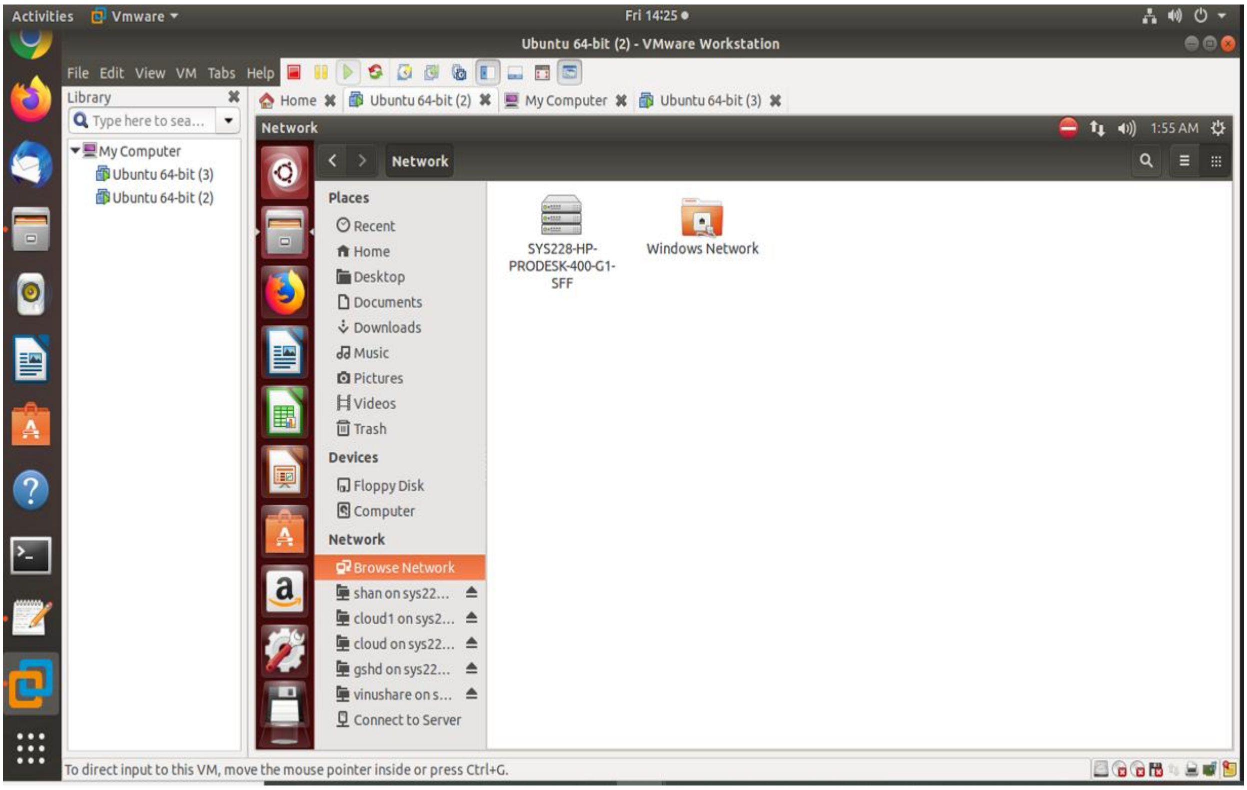
❖ Configure the Shared folder settings (select all the check boxes). Once settings are done the shared folder will be changed with a double arrow.





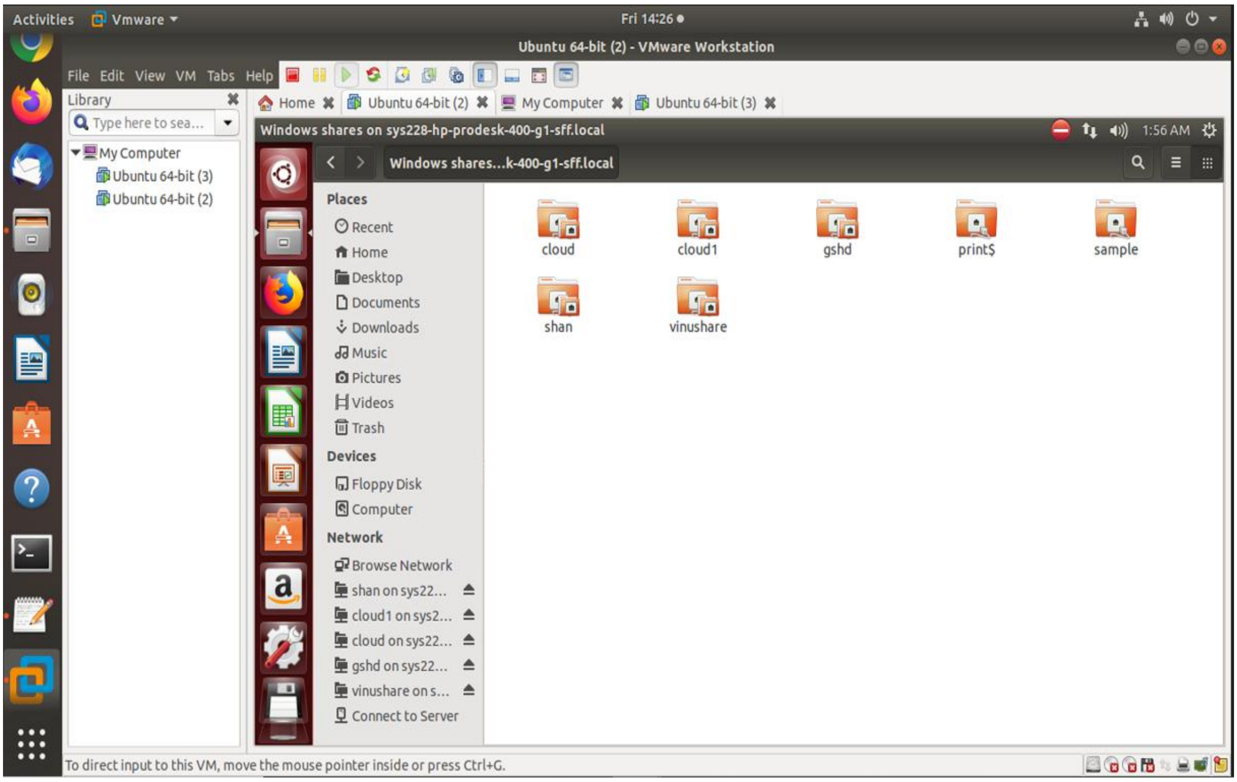
```
Welcome to cloud computing
```

❖ Click the Virtual machine, goto **Network** and choose **Browse Network** and click **SYS228-HP-PRODESK-400-G1-SFF named drive**.

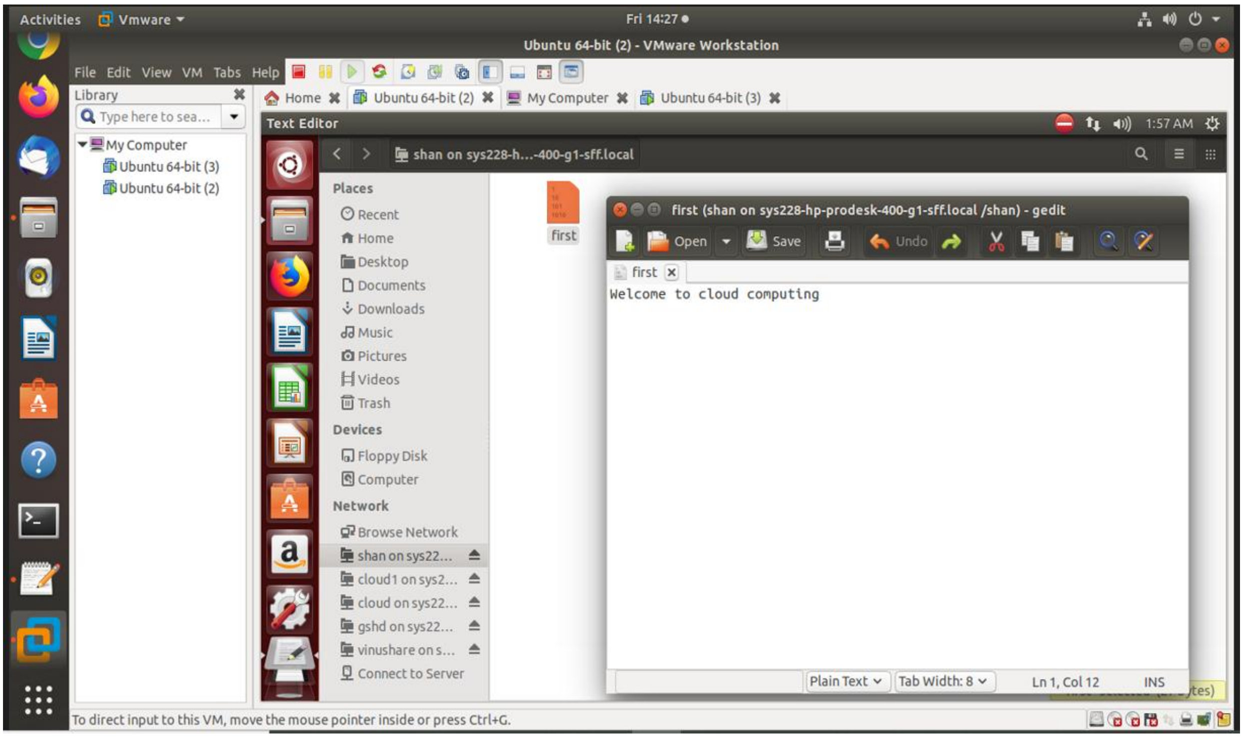


To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

❖ Shared folder is visible once the SYS drive is clicked. Open the shared folder.

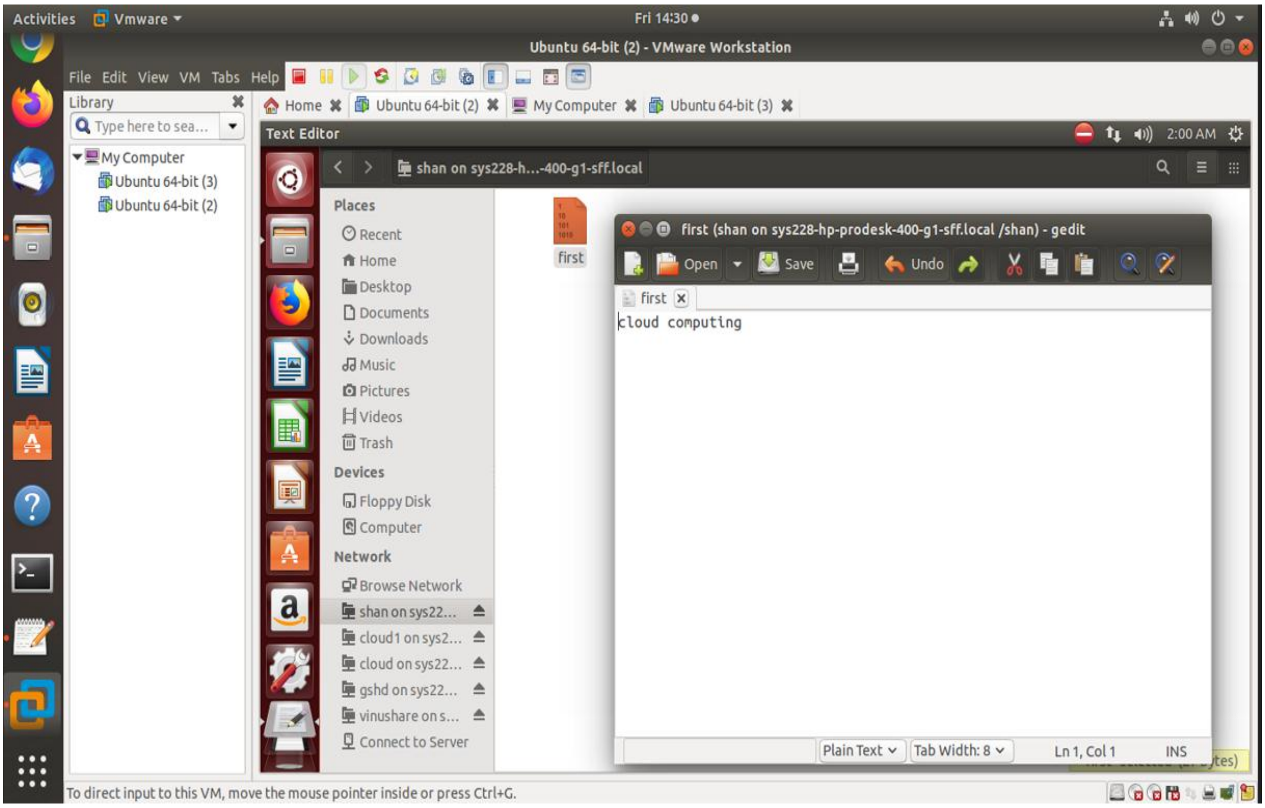


- ❖ Now open the file first.txt and edit it's contents.


Ubuntu 64-bit (2) - VMware Workstation
Text Editor
shan on sys228-h...-400-g1-sff.local
first
first (shan on sys228-hp-prodesk-400-g1-sff.local /shan) - gedit
Open
Save
Undo
Plain Text
Tab Width: 8
Ln 1, Col 12
INS
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

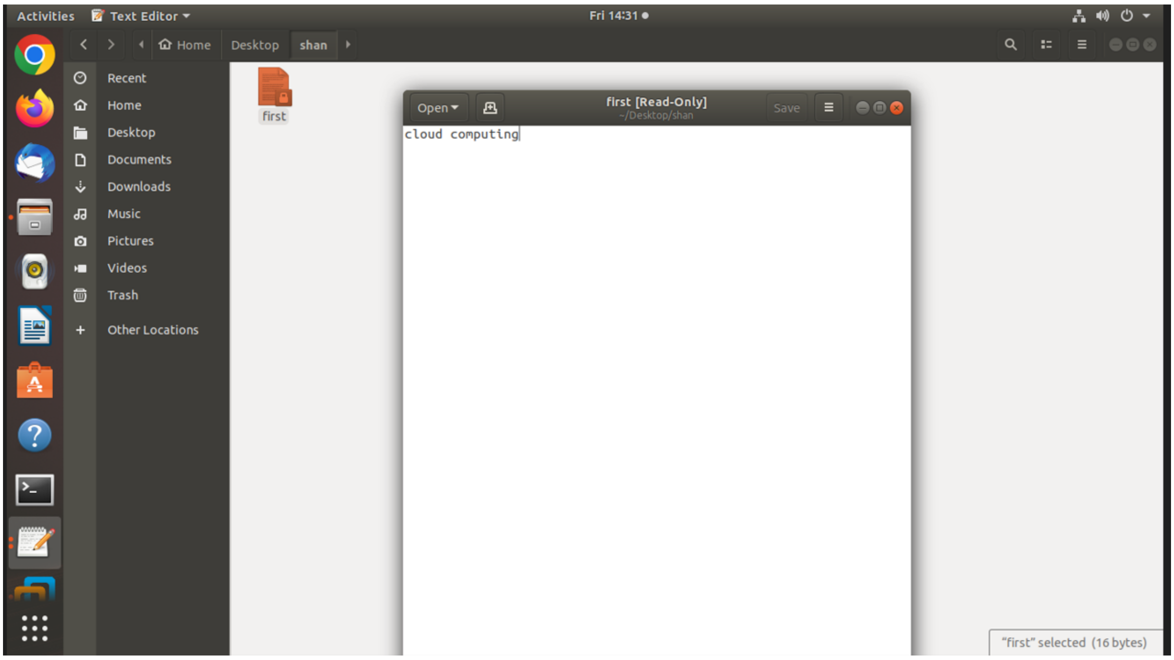
```
Welcome to cloud computing
```

- ❖ Save the file. Open the shared folder which you have seen in the Virtual machine. Open the folder and open the text file.



```
cloud computing
```

❖ Finally you can see that the file has been transferred from one host to virtual machine by using the shared folder method.



```
cloud computing
```

RESULT

Thus, the transfer of a file between virtual machines and the host was successfully implemented using the shared folder procedure as well as the correctness has been proved here successfully

EX NO: 7 **LAUNCH VIRTUAL MACHINE USING OPENSTACK**
DATE:

AIM:
To study and find a procedure to work with openstack

PROCEDURE:
OpenStack was introduced by Rackspace and NASA in July 2010. OpenStack is an Infrastructure as a Service known as Cloud Operating System, that take resources such as Compute, Storage, Network and Virtualization Technologies and control those resources at a data center level . The project is building an open source community - to share resources and technologies with the goal of creating a massively scalable and secure cloud infrastructure. The software is open source and limited to just open source APIs such as Amazon. It is modular architecture , Designed to easily scale out Based on (growing) set of core services ,The major components are 1. Keystone 2. Nova 3. Glance 4. Swift 5. Quantum 6. Cinder

- **Compute (Nova):** Compute is a controller that is used to manage resources in virtualized environments. It handles several virtual machines and other instances that perform computing tasks.
- **Object Storage (Swift):** To store and retrieve arbitrary data in the cloud, object storage is used. In Swift, it is possible to store the files, objects, backups, images, videos, virtual machines, and other unstructured data. Developers may use a special identifier for referring the file and objects in place of the path, which directly points to a file and allows the OpenStack to manage where to store the files.
- **Block Storage (Cinder):** This works in the traditional way of attaching and detaching an external hard drive to the OS for its local use. Cinder manages to add, remove, create new disk space in the server. This component provides the virtual storage for the virtual machines in the system.
- **Networking (Neutron):** This component is used for networking in OpenStack. Neutron manages all the network-related queries, such as IP address management, routers, subnets, firewalls, VPNs, etc. It confirms that all the other components are well connected with the OpenStack.
- **Dashboard (Horizon):** This is the first component that the user sees in the OpenStack. Horizon is the web UI (user interface) component used to access the other back-end services. Through individual API (Application programming interface), developers can access the OpenStack's components, but through the dashboard, system administrators can look at what is going on in the cloud and manage it as per their need.
- **Identity Service (Keystone):** It is the central repository of all the users and their permissions for the OpenStack services they use. This component is used to manage identity services like authorization, authentication, AWS Styles (Amazon Web Services) logins, token-based systems, and checking the other credentials (username & password).
- **Image Service (Glance):** The glance component is used to provide the image services to OpenStack. Here, image service means the images or virtual copies of hard disks. When we plan to deploy a new virtual machine instance, then glance allows us to use these

images as templates. Glance allows virtual box (VDI), VMware (VMDK, OVF), Raw, Hyper-V (VHD) and KVM (qcow2) virtual images.

- **Telemetry (Ceilometer):** It is used to meter the usage and report it to OpenStack's individual users. So basically, Telementry provides billing services to OpenStack's individual users.
- **Orchestration (Heat):** It allows the developers to store the cloud application's necessities as a file so that all-important resources are available in handy. This component organizes many complex applications of the cloud through the templates, via both the local OpenStack REST API and Query API.
- **Shared File System (Manila):** It offers storage of the file to a virtual machine. This component gives an infrastructure for managing and provisioning file shares.
- **Elastic Map-reduce (Sahara):** The Sahara component offers a simple method to the users to preplanned Hadoop clusters by referring to the multiple options such as the Hadoop version, cluster topology and hardware details of nodes and some more.

WORKING OF OPENSTACK

Basically, OpenStack is a series of commands which is called scripts. And these scripts are packed into packages, which are called projects that rely on tasks that create cloud environments. OpenStack relies on two other forms of software in order to construct certain environments:

- Virtualization means a layer of virtual resources basically abstracted from the hardware.
- A base OS that executes commands basically provided by OpenStack Scripts.

So, we can say all three technologies, i.e., virtualization, base operating system, and OpenStack must work together. As we know, the Horizon is an interface for the appliance environment. Anything that the user wants to do should use the Horizon (Dashboard). The Dashboard is a simple graphical user interface with multiple modules, where each module performs specific tasks.

All the actions in OpenStack work by the service API call. So, if you are performing any task, it means you are calling a service API. Each API call is first validated by Keystone. So, you will have to login yourself as a registered user with your login username and password before you enter the OpenStack dashboard.

Once you successfully log in to the OpenStack dashboard, you will get many options to create new instances, volumes, Cinder, and configure the network.

Instances are nothing but a virtual machine or environment. To generate a new VM, use the 'instances' option from the OpenStack dashboard. In these instances, you can configure your cloud. Instances can be RedHat, OpenSUSE, Ubuntu, etc. The formation of an instance is also an API call. You can configure network information in the instances. You can connect these instances to the cinder instance or volume to add more services.

After the successful creation of an instance, you can configure it, you can access it through CLI, and whatever data you want to add, you can do it. Even you can set up an instance to manage and store the snapshots for future reference or backup purposes.

BENEFITS OF OPENSTACK

There are a lot of benefits of OpenStack in the cloud computing platform. Let's see one by one :

1. Open Source

As we know, using the open-source environment, we can create a truly defined data center. OpenStack is the largest open-source platform. It offers the networking, computing, and storage subsystems in a single platform. Some vendors (such as RedHat) have developed and continue to support their own OpenStack distributions.

The two main advantages of the open-source OpenStack project is :

- ❖ OpenStack can be modified according to your rising demand - As per your requirement, you can add the extra features in OpenStack.
- ❖ It can be used without any limitations - Since OpenStack is a freely available project, so there are no limitations or restrictions to use it. You can use it as per your requirement. There are no limits for what purpose you use it, where you use it, or how long you use it.

2. Scalability

Scalability is the major key component of cloud computing. OpenStack offers better scalability for businesses. Through this feature, it allows enterprises to spin up and spin down servers on-demand.

3. Security

One of the significant features of OpenStack is security, and this is the key reason why OpenStack is so popular in the cloud computing world.

- ❖ With OpenStack, your data is always secure - When company owners want to move their IT infrastructure to the cloud, they always fear data loss. But there is no need to think about data loss with OpenStack. It offers the best security feature.
- ❖ OpenStack provides security professionals who are responsive to OpenStack's strong security.

4. Automation

Automation is one of the main keys selling points of OpenStack when compared to another option. The ease with which you can automate tasks makes OpenStack efficient. OpenStack comes with a lot of inbuilt tools that make cloud management much faster and easier. OpenStack provides its own API or Application Program Interface that helps other applications to have full control over the cloud. This function makes it easier to build your own apps that can communicate with OpenStack to perform tasks such as firing up VMs.

Development Support
Since OpenStack's source code is freely accessible, experts from all over the world can improve the platform. If a new feature is being designed, it can be built easily and professionally by a development team. OpenStack is like Linux in that have many distributions with different features but share the same component. Support from companies - For development, OpenStack gets support from IT founders, including Intel, IBM, AT&T, Cisco, Red Hat, Dell, Ubuntu, and so on. So, by leaps and bounds, it's changing, which is a massive benefit for you.Support from the developers' community - Many developers are working on the enhancement of OpenStack. They are continuously working hard to make the OpenStack better.

5. Easy to Access and Manage

We can easily access and manage OpenStack, which is the biggest benefit for you. OpenStack is easy to access and manage because of the following features :Command Line Tools - We can access the OpenStack using command-line tools.Dashboard - OpenStack offers users and administrators to access and manage various aspects of OpenStack using GUI (graphical user interface) based dashboard component. It is available as a web UI.APIs - There are a lot of APIs (Application Program Interface), which is used to manage OpenStack.

6. Services
OpenStack provides many services required for several different tasks for your public, private, and hybrid cloud.List of services - OpenStack offers a list of services or components such as the Nova, Cinder, Glance, Keystone, Neutron, Ceilometer, Sahara, Manila, Searchlight, Heat, Ironic, Swift, Trove, Horizon, etc.Each component is used for different tasks. Such as Nova provides computing services, Neutron provides networking services, Horizon provides a dashboard interface, etc.

7. Strong Community

OpenStack has many experts, developers, and users who love to come together to work on the product of OpenStack and enhance the feature of OpenStack.

8. Compatibility
Public cloud systems like AWS (Amazon Web Services) are compatible with OpenStack. Installation of OpenStack.In order to install the DevStack in a system, first, you have to create a Linux VM on your computer (such as using VirtualBox or VMware) or remotely in the cloud (such as using AWS). The VM must have at least 4GB of memory, and the proper internet connection is also important. Here, we are going to use one version of the ubuntu, i.e., 18.04.

Follow the following steps to install the OpenStack in your ubuntu virtual machine :

Step 1: Update Ubuntu System
Open the terminal and run the following command to ensure that the system is up to date :

```
$ sudo apt update
$ sudo apt -y upgrade
$ sudo apt -y dist-upgrade
```

Reboot the system after running the above command. To reboot the system, run the following command :

```
$ sudo reboot
```

Step 2: Create Stack User

It is important that the devstack must run as a regular user (non-root user) with the sudo enabled.

To keep this note in mind, let's create a new user with the name "stack" and assign the sudo permissions or privileges. To create a stack user, run the following command in your terminal:

```
$ sudo useradd -s /bin/bash -d /opt/stack -m stack
```

to assign the sudo privileges to the stack user, run the following command :

```
$ echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack
```

You can switch to the 'stack' user by running the following command:

```
$ sudo su - stack
```

Step 3: Install the Git

In Most of the ubuntu systems, git comes by default. But if git is missing on your system, then install it by running the following command:

```
$ sudo apt install git -y
```

Step 4: Download OpenStack

Once you install the git, use the git command to download the DevStack from Github.

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

Step 5: Create a DevStack Configuration File

First of all, go to the devstack directory by running the following command :

```
$ cd devstack
```

Now, create a local.conf file in which you have to enter the four passwords and the host IP address :

Copy the following line of content in the file :

```
[[local|localrc]]
# Password for KeyStone, Database, RabbitMQ and Service
ADMIN_PASSWORD=StrongAdminSecret
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD
# Host IP - To get your Server or VM IP, run the 'ip addr' or 'ifconfig' command
HOST_IP=192.168.56.103
```

Press the ESC, then wq to save and then exit from the local.conf file.

Here, ADMIN_PASSWORD is the password that we will use to log into the OpenStack login page. The default username for an OpenStack is 'admin'.

And HOST_IP is the IP address of your system. To get your Server or VM IP, run the 'ifconfig' or 'ip addr' command.

Step 6 : Install OpenStack with DevStack

To install and run the openstack, execute the following command :

```
$ ./stack.sh
```

The installation will take about 10-20 minutes, mostly depends on your internet speed.

At the very end of the installation, you will get the host's IP address, URL for managing it and the username and password to handle the administrative task.

Step 7: Accessing OpenStack on a browser

Copy the horizon URL given in the installation output and paste it into your browser :

http://<IP Address>/dashboard

To login to OpenStack with the default username - admin or demo and configured password - secret.

Once you login into the OpenStack, you will be redirected to the Dashboard of OpenStack. This dashboard screen is called the Openstack management web console.

RESULT

Thus the procedure to work with openstack was written successfully.

EX.NO: 8 **HADOOP SINGLE NODE CLUSTER**
Date:

AIM

To find procedure to set up the one node Hadoop cluster.

PROCEDURE

1. **Initial setup:** Set up a virtual machine (VM) with a fresh installation of Ubuntu 20.04.
2. **Installing hadoop dependencies, setting up the environment:** Hadoop requires Java 8 or 11 install default-jdk by executing the following command:
   - **$ sudo apt install default-jdk**

   The default JDK in Ubuntu 20.04 is 11, but you can also install JDK 11 explicitly by installing the openjdk-11-jdk package, as follows:
   - **$ sudo apt install openjdk-11-jdk**

3. Confirm that Java 11 is installed:
   - **$ java -version**

   openjdk version "11.0.11" 2021-04-20
   OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.20.04)
   OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.20.04, mixed mode, sharing).
4. **Now set up the environment variables:** Hadoop requires JAVA_HOME to be set, and variable HADOOP_HOME. Append the following lines to $HOME/.profile:

   - **export JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64"**
   - **export HADOOP_HOME="/opt/hadoop-3.3.1"**

   (or)
   - **$ echo"export JAVA_HOME=\"/usr/lib/jvm/java-11-openjdk-amd64\"" >> HOME/.profile**
   - **$ echo "export HADOOP_HOME=\"/opt/hadoop-3.3.1\"" >> $HOME/.profile**

5. **Source the file to apply the changes:**
   - **$ source $HOME/.profile**

6. **Confirm that the environment variables are correctly set:**
   - **$ echo $JAVA_HOME**
     /usr/lib/jvm/java-11-openjdk-amd64
   - **$ echo $HADOOP_HOME**
     /opt/hadoop-3.3.1
   - listing the files included in the openjdk-11-jdk package using dpkg:
     $ dpkg -L openjdk-11-jdk

7. **Installing Hadoop 3.3.1**

The de-facto way of installing Hadoop in 2021 is (still) directly fetching the compressed tarball from upstream. Fetch the upstream tarball for Hadoop 3.3.1 using wget:

$ wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.1/hadoop-3.3.1.tar.gz

8. **Unpack the tarball:**
    - **$ tar xvf hadoop-3.3.1.tar.gz**

9. **Now copy it to an appropriate location, e.g. $HADOOP_HOME:**
    - **$ sudo cp -r hadoop-3.3.1 "$HADOOP_HOME"**

This copies and places the Hadoop program files under /opt, which is where manually installed software should be placed as dictated by the FHS.

10. **Hadoop binaries are located under $HADOOP_HOME/bin. Add that to our PATH:**
    - **$ echo "export PATH=\"\$PATH:\$HADOOP_HOME/bin\"" >> $HOME/.profile**
    - **$ source $HOME/.profile**

If you did everything correctly, the hadoop command should now be available and print usage instructions:

- **$ hadoop**

11. **Fetch our text sample:**

$wget https://raw.githubusercontent.com/ErikSchierboom/sentencegenerator/master/samples/the-king-james-bible.txt

Now create an input directory and place the text sample there:

- **$ mkdir input**
- **$ mv the-king-james-bible.txt input**

My KJB copy is 4.2 megabytes in size:

12. **$ ls -lh input**

```
total 4.2M
-rw-rw-r-- 1 ubuntu ubuntu 4.2M AUG 14 10:20 the-king-james-bible.txt
```

13. **Finally we invoke the WordCount MapReduce program with input directory input and output directory output:**
    - **$ hadoop jar "$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.1.jar" wordcount input output**

14. **Confirm that the output has been successfully generated:**
    - **$ ls -lh output**

```
total 336K
-rw-r--r-- 1 ubuntu ubuntu      0 AUG 14 10:26 _SUCCESS
-rw-r--r-- 1 ubuntu ubuntu 334K AUG 14 10:26 part-r-00000
```

So our output fits in a single block (output/part-r-00000) as expected.

15. **Finally, let's take the last 10 lines of our output (or the first 10 lines, or whatever):**

$ tail output/part-r-00000

```
youth      7
youth      8
youth      2
youthful   1
youths     1
youths     1
```

```
zeal        13
zeal        3
zealous     8
zealously   2
```

16. **Remove the output directory:**
    - **$ rm -rf output**

RESULT

Thus the procedure to set up the one node Hadoop cluster has been developed.