Procedure:
Before downloading and installing VMware Workstation:
Ensure that you are using a supported guest operating system

Downloading VMware Workstation
To download VMware Workstation:
Navigate to the VMware Workstation Download Center.

1. Based on your requirements, click **Go to Downloads** for VMware Workstation for Windows or VMware Workstation for Linux.
2. Click **Download Now**.
3. If prompted, log in to your My VMware profile. If you do not have a profile, create one. For more information, see How to create a My VMware profile (2007005).
4. Ensure that your profile is complete and enter all mandatory fields. For more information, see How to update your My VMware profile (2086266).
5. Review the End User License Agreement and click **Yes**.
6. Click **Download Now**.

If the installer fails to download during the download process:

- Delete the cache in your web browser. For more information, see:
- Mozilla Firefox: How to clear the Firefox cache
- Google Chrome: Delete your cache and other browser data
- Microsoft Internet Explorer: How to delete the contents of the Temporary Internet Files folder
- Disable the pop-up blocker in your web browser. For more information, see:
- Mozilla Firefox: How do I disable a Pop-up blocker?
- Google Chrome: Manage pop-ups
- Microsoft Internet Explorer: How to turn Internet Explorer Pop-up Blocker on or off on a Windows XP SP2-based computer
- Download using a different web browser application.
- Disable any local firewall software.
- Restart the virtual machine.
- Download the installer from a different computer or network.

Installing VMware Workstation Notes:

- You must have only one VMware Workstation installed at a time. You must uninstall previous version of VMware Workstation before installing a new version.
- If the installer reports an error when you run it, you must verify the download. For more information, see Verifying the integrity of downloaded installer files (1537).

To install VMware Workstation on a Windows host:

1. Log in to the Windows host system as the Administrator user or as a user who is a member of the local Administrators group.
2. Open the folder where the VMware Workstation installer was downloaded. The default location is the **Downloads** folder for the user account on the Windows host.
3. Right-click the installer and click Run as Administrator.
4. Select a setup option:
   Typical: Installs typical Workstation features. If the Integrated Virtual Debugger for Visual Studio or Eclipse is present on the host system, the associated Workstation plug-ins are installed.
   Custom: Lets you select which Workstation features to install and specify where to install them. Select this option if you need to change the shared virtual machines directory, modify the VMware Workstation Server port, or install the enhanced virtual keyboard driver. The enhanced virtual keyboard driver provides better handling of international keyboards and keyboards that have extra keys
5. Follow the on-screen instructions to finish the installation.
6. Restart the host machine.

To install VMware Workstation on a Linux host:

Note: VMware Workstation for Linux is available as a .bundle download in the VMware Download Center. The Linux bundle installer starts a GUI wizard on most Linux distributions. In some Linux distributions, the bundle installer starts a command-line wizard instead of a GUI wizard.

1. Log in to the Linux host with the user account that you plan to use with VMware Workstation.
2. Open a terminal interface. For more information, see Opening a command or shell prompt (1003892).
3. Change to root. For example: su root

Note: The command that you use depends on your Linux distribution and configuration.

4. Change directories to the directory that contains the VMware Workstation bundle installer file. The default location is the **Download** directory.
5. Run the appropriate Workstation installer file for the host system.

For example:

```
sh VMware-workstation-Full-xxxx xxxx.architecture.bundle [--option]
```

Where:

- *xxxx-xxxx* is the version and build numbers
- architect is a command line option.

This table describes the command line options:

Option	**Description**
--gtk	Opens the GUI-based VMware installer, which is the default option.
--console	Use the terminal for installation.
--custom	Use this option to customize the locations of the installation directories and set the hard limit for the number of open file descriptors.
--regular	Shows installation questions that have not answered before or are required. This is the default option.
--ignore-errors or -I	Allows the installation to continue even if there is an error in one of the installer scripts. Because the section that has an error does not complete, the component might not be properly configured.
--required	Shows the license agreement only and then proceeds to install Workstation

6. Accept the license agreement.

Note: If you are using the --console option or installing VMware Workstation on a Linux host that does not support the GUI

wizard, press Enter to scroll through and read the license agreement or type q to skip to the yes/no prompt.
7. Follow the on-screen instructions or prompts to finish the installation.
8. Restart the Linux host.

After installation

On Windows host systems:

- The installer creates a desktop shortcut, a quick launch shortcut, or a combination of these options in addition to a Start Menu item.
- To start VMware Workstation on a Windows host system, select **Start** > **Programs** >**VMware Workstation**.

On Linux host systems:

- VMware Workstation can be started from the command line on all Linux distributions.
- On some Linux distributions, VMware Workstation can be started in the GUI from the **System Tools** menu under **Applications**.
- To start VMware Workstation on a Linux host system from the command line, run the vmware & command in a terminal window. For more information, see Opening a command or shell prompt (1003892).

For example:

```
/usr/bin/vmware &
```

When you start the Workstation for the first time, Workstation prompts you to accept the **End User License Agreement**. After you start Workstation, the Workstation window opens.
To install VM on windows go to web site https://www.iitk.ac.in/nt/faq/vbox.htm

Simple Steps to a New Virtual Machine

By default, the new virtual machine uses an IDE disk for Windows 95, Windows 98, Windows Me, Windows XP, Windows Server 2003, NetWare and FreeBSD guests. The default for other guest operating systems is a SCSI disk.

Follow these steps to create a virtual machine using a virtual disk.

1. Start VMware Workstation.

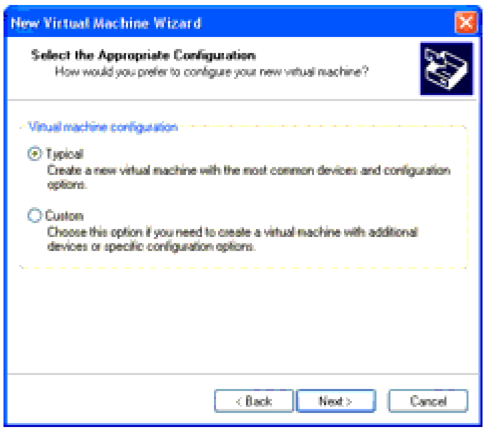
   Windows hosts: Double-click the VMware Workstation icon on your desktop or use the **Start** menu (**Start** > **Programs** > **VMware** > **VMware Workstation**).

Linux hosts: In a terminal window, enter the command vmware &

2. If this is the first time you have launched VMware Workstation and you did not enter the serial number when you installed the product (an option available on a Windows host), you are prompted to enter it. The serial number is on the registration card in your package. Enter your serial number and click **OK**.The serial number you enter is saved and VMware Workstation does not ask you for it again. For your convenience, VMware Workstation automatically sends the serial number to the VMware Web site when you use certain Web links built into the product (for example, **Help** > **VMware software on the Web** > **Register Now!** and **Help** > **VMware on the Web** > **Request Support**). This allows us to direct you to the correct Web page to register and get support for your product.
3. **Linux hosts:** If this is the first time you have launched VMware Workstation, a dialog box asks if you want to rename existing virtual disks using the new .vmdk extension. Click **OK** to search all local drives on the host computer and make this change. (On Windows hosts, you have a chance to rename virtual disk files when you are installing VMware Workstation.The converter also renames the files that store the state of a suspended virtual machine, if it finds them. It changes the old .std file extension to .vmss. However, it is best to resume and shut down all suspended virtual machines before you upgrade to Workstation 4.

   Besides renaming files, the converter updates the corresponding virtual machine configuration files so they identify the virtual disks using the new filenames.
   If you store your virtual disk files or suspended state files on a Windows XP or Windows Server 2003 host - or if you may do so in the future - it is important to convert the filenames to avoid conflicts with the System Restore feature of Windows XP and Windows Server 2003.
   Linux Hosts: One Chance to Rename Disk Files
   The Rename Virtual Disks dialog box appears only once. If you click **Cancel**, you will not have another opportunity to update the filenames and configuration files automatically.
4. Start the New Virtual Machine Wizard.
   When you start VMware Workstation, you can open an existing virtual machine or create a new one. Choose **File** > **New** > **New Virtual Machine** to begin creating your virtual machine.
5. The New Virtual Machine Wizard presents you with a series of screens that you navigate using the Next and Prev buttons at the bottom of each screen. At each screen, follow the instructions, then click **Next** to proceed to the next screen.

6. Select the method you want to use for configuring your virtual machine.


New Virtual Machine Wizard

Select the Appropriate Configuration

How would you prefer to configure your new virtual machine?

Virtual machine configuration

[✓ Typical]
Create a new virtual machine with the most common devices and configuration options.

Custom
Choose this option if you need to create a virtual machine with additional devices or specific configuration options.

If you select **Typical**, the wizard prompts you to specify or accept defaults for

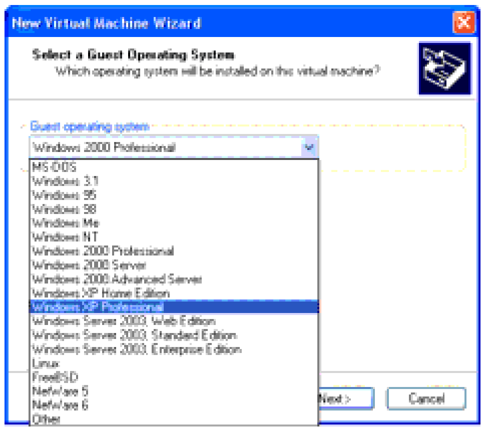
- The guest operating system
- The virtual machine name and the location of the virtual machine's files
- The network connection type

If you select **Custom**, you also can specify how to set up your disk - create a new virtual disk, use an existing virtual disk or use a physical disk - and specify the settings needed for the type of disk you select.

Select **Custom** if you want to

- Make a virtual disk larger or smaller than 4GB
- Store your virtual disk's files in a particular location
- Use an IDE virtual disk for a guest operating system that would otherwise have a SCSI virtual disk created by default
- Allocate all the space for a virtual disk at the time you create it
- Choose whether to split a virtual disk into 2GB files
- Use a physical disk rather than a virtual disk (for expert users)
- Set memory options that are different from the defaults.

7. Select a guest operating system.


New Virtual Machine Wizard

Select a Guest Operating System

Which operating system will be installed on this virtual machine?

Guest operating system

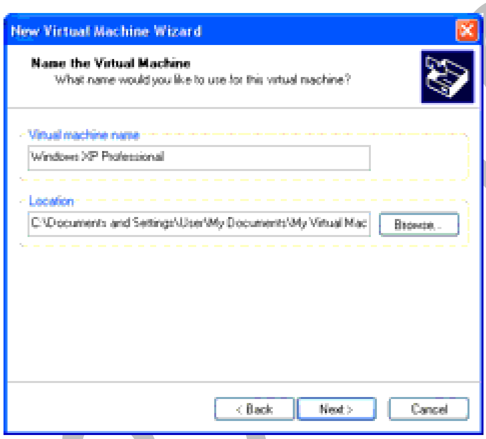
Windows 2000 Professional

MS-DOS
Windows 3.1
Windows 95
Windows 98
Windows Me
Windows NT
Windows 2000 Professional
Windows 2000 Server
Windows 2000 Advanced Server
Windows XP Home Edition
[✓ Windows XP Professional]
Windows Server 2003, Web Edition
Windows Server 2003, Standard Edition
Windows Server 2003, Enterprise Edition
Linux
FreeBSD
NetWare 5
NetWare 6
Other

This screen asks which operating system you plan to install in the virtual machine. The New Virtual Machine Wizard uses this information to select appropriate default values, such as the amount of memory needed. The wizard also uses this information when naming associated virtual machine files.If the operating system you are using is not listed, select **Other**.

The remaining steps assume you plan to install a Windows XP Professional guest operating system. You can find detailed installation notes for this and other guest operating systems in the *VMware Guest Operating System Installation Guide*, available from the VMware Web site or from the Help menu.

8. Select a name and folder for the virtual machine.


New Virtual Machine Wizard

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

Virtual machine name: Windows XP Professional

Location: C:\Documents and Settings\User\My Documents\My Virtual Mac

The name specified here is used if you add this virtual machine to the VMware Workstation Favorites list. This name is also used as the name of the folder where the files associated with this virtual machine are stored.

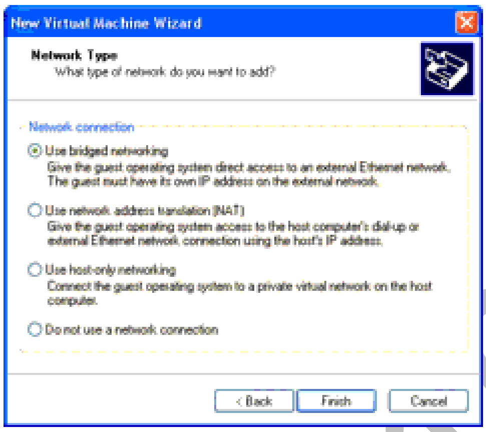
Each virtual machine should have its own folder. All associated files, such as the configuration file and the disk file, are placed in this folder.

Windows hosts: On Windows 2000, Windows XP and Windows Server 2003, the default folder for this Windows XP Professional virtual machine is C:\Documents and Settings\<username>\My Documents\My Virtual Machines\Windows XP Professional. On Windows NT, the default folder is C:\WINNT\Profiles\<username>\Personal\My Virtual Machines\Windows XP Professional.

Linux hosts: The default location for this Windows XP Professional virtual machine is <homedir>/vmware/winXPPro, where <homedir> is the home directory of the user who is currently logged on.

Virtual machine performance may be slower if your virtual hard disk is on a network drive. For best performance, be sure the virtual machine's folder is on a local drive. However, if other users need to access this virtual machine, you should consider placing the virtual machine files in a location that is accessible to them. For more information, see <u>Sharing Virtual Machines with Other Users.</u>

9. Configure the networking capabilities of the virtual machine.



New Virtual Machine Wizard

Network Type
What type of network do you want to add?

Network connection

(●) Use bridged networking
Give the guest operating system direct access to an external Ethernet network. The guest must have its own IP address on the external network.

( ) Use network address translation (NAT)
Give the guest operating system access to the host computer's dial-up or external Ethernet network connection using the host's IP address.

( ) Use host-only networking
Connect the guest operating system to a private virtual network on the host computer.

( ) Do not use a network connection

If your host computer is on a network and you have a separate IP address for your virtual machine (or can get one automatically from a DHCP server), select **Use bridged networking**.

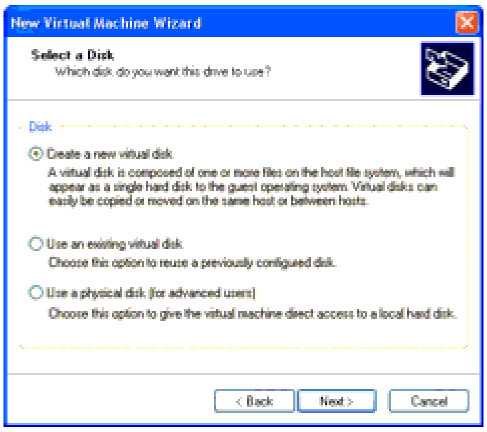
If you do not have a separate IP address for your virtual machine but you want to be able to connect to the Internet, select **Use network address translation (NAT)**. NAT is useful if you have a wireless network adapter on a Linux host (as bridged networking on wireless network adapters is supported only on Windows hosts). It also allows for the sharing of files between the virtual machine and the host operating system.

For more details about VMware Workstation networking options, see <u>Networking</u>.

10. If you selected **Typical** as your configuration path, click **Finish** and the wizard sets up the files needed for your virtual machine.

If you selected **Custom** as your configuration path, continue with the steps for configuring a disk for your virtual machine.

11. Select the disk you want to use with the virtual machine.



New Virtual Machine Wizard

Select a Disk
Which disk do you want this drive to use?

Disk

(●) Create a new virtual disk
A virtual disk is composed of one or more files on the host file system, which will appear as a single hard disk to the guest operating system. Virtual disks can easily be copied or moved on the same host or between hosts.

( ) Use an existing virtual disk
Choose this option to reuse a previously configured disk.

( ) Use a physical disk (for advanced users)
Choose this option to give the virtual machine direct access to a local hard disk.

Select **Create a new virtual disk**.

Virtual disks are the best choice for most virtual machines. They are quick and easy to set up and can be moved to new locations on the same host computer or to different host computers. By default, virtual disks start as small files on the host computer's hard drive, then expand as needed - up to the size you specify in the next step. The next step also allows you to allocate all the disk space when the virtual disk is created, if you wish.

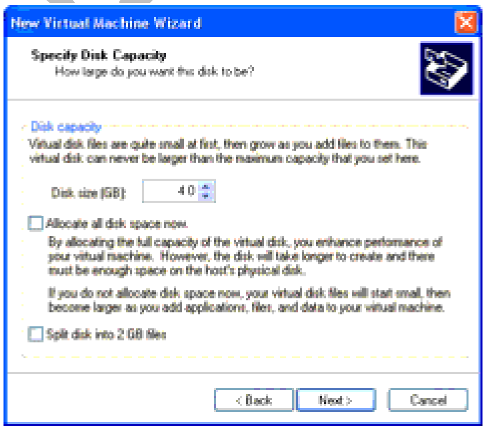
To use an existing operating system on a physical hard disk (a "raw" disk), read Configuring a Dual-Boot Computer for Use with a Virtual Machine. To install your guest operating system directly on an existing IDE disk partition, read the reference note Installing an Operating System onto a Raw Partition from a Virtual Machine.

Caution: Raw disk configurations are recommended only for expert users.

Caution: If you are using a Windows Server 2003, Windows XP or Windows 2000 host, see Do Not Use Windows 2000, Windows XP and Windows Server 2003 Dynamic Disks as Raw Disks.

To install the guest operating system on a raw IDE disk, select **Existing IDE Disk Partition**. To use a raw SCSI disk, add it to the virtual machine later with the Virtual Machine Control Panel. Booting from a raw SCSI disk is not supported. For a discussion of some of the issues involved in using a raw SCSI disk, see Configuring Dual- or Multiple-Boot SCSI Systems to Run with VMware Workstation on a Linux Host.

12. Specify the capacity of the virtual disk.


New Virtual Machine Wizard

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

Disk capacity
Virtual disk files are quite small at first, then grow as you add files to them. This virtual disk can never be larger than the maximum capacity that you set here.

Disk size (GB): 4.0

[ ] Allocate all disk space now.
By allocating the full capacity of the virtual disk, you enhance performance of your virtual machine. However, the disk will take longer to create and there must be enough space on the host's physical disk.

If you do not allocate disk space now, your virtual disk files will start small, then become larger as you add applications, files, and data to your virtual machine.

[ ] Split disk into 2 GB files

If you do not select this option, the virtual disk's files start small and grow as needed, but they can never grow larger than the size you set here.

You can set a size between 2GB and 256GB for a SCSI virtual disk or 128GB for an IDE virtual disk. The default is 4GB.

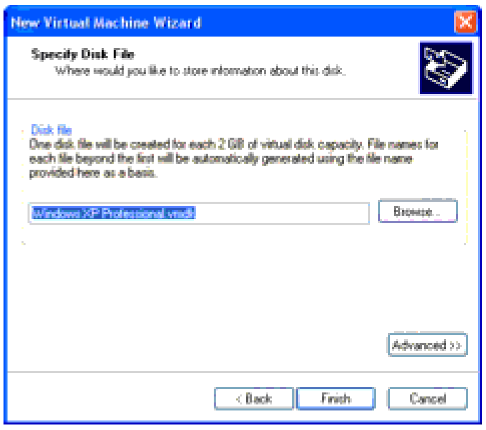
You may also specify whether you want the virtual disk created as one large file or split into a set of 2GB files.

Make the Virtual Disk Big Enough

The virtual disk should be large enough to hold the guest operating system and all of the software that you intend to install, with room for data and growth.
You cannot change the virtual disk's maximum capacity later.
You can install additional virtual disks using the Virtual Machine Control Panel.

For example, you need about 500MB of actual free space on the file system containing the virtual disk to install Windows Me and popular applications such as Microsoft Office inside the virtual machine. You can set up a single virtual disk to hold these files. Or you can split them up - installing the operating system on the first virtual disk and using a second virtual disk for applications or data files.

13. Specify the location of the virtual disk's files.

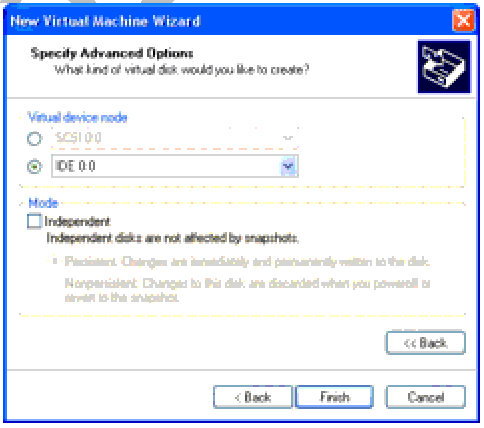


Specify Disk File
Where would you like to store information about this disk.

Disk file
One disk file will be created for each 2 GB of virtual disk capacity. File names for each file beyond the first will be automatically generated using the file name provided here as a basis.

Windows XP Professional.vmdk

On the advanced settings screen, you can also specify a disk mode. This is useful incertain special-purpose configurations in which you want to exclude disks from the snapshot. For more information on the snapshot feature, see using the Snapshot.



Specify Advanced Options
What kind of virtual disk would you like to create?

Virtual device node
SCSI 0:0
[✓ IDE 0:0]

Mode
Independent
Independent disks are not affected by snapshots.
Persistent. Changes are immediately and permanently written to the disk.
Nonpersistent. Changes to this disk are discarded when you poweroff or revert to the snapshot.

Normal disks are included in the snapshot. In most cases, this is the setting you want.
Independent disks are not included in the snapshot.
Caution: The independent disk option should be used only by advanced users who need it for special-purpose configurations.
You have the following options for an independent disk:

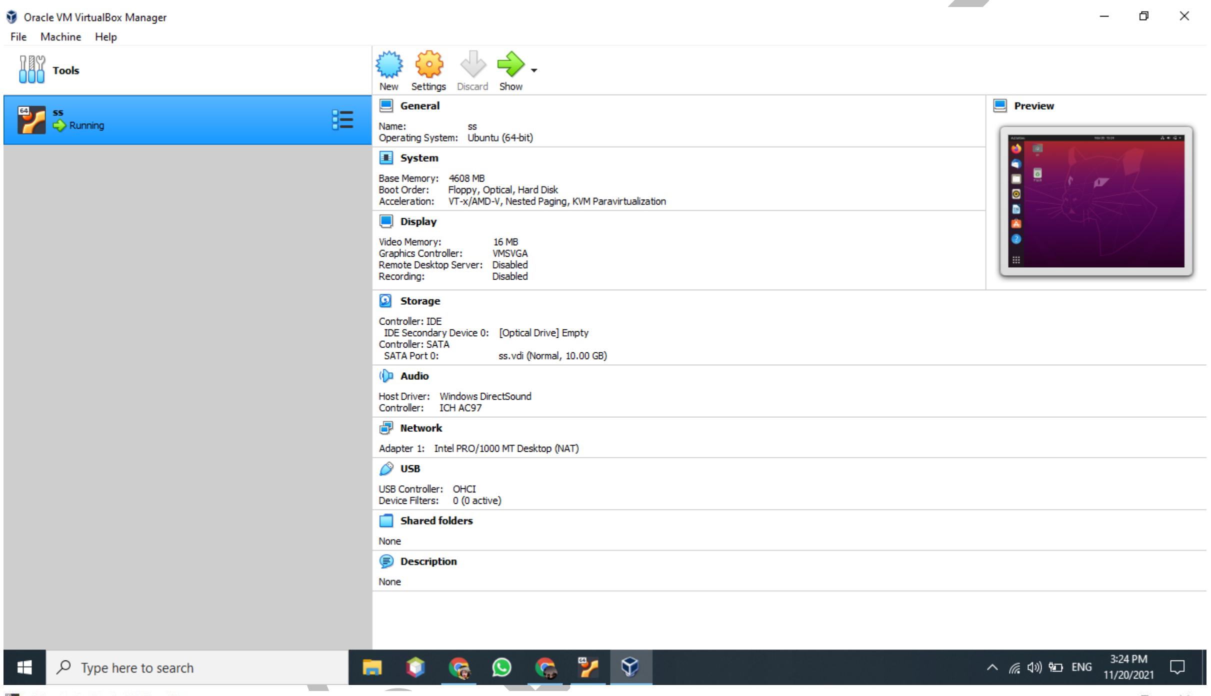
- **Persistent** - changes are immediately and permanently written to the disk.

- **Nonpersistent** - changes to the disk are discarded when you power off or revert to the snapshot.

When you have set the filename and location you want to use and have made any selections you want to make on the advanced settings screen, click **Finish**.

14. Click **Finish**. The wizard sets up the files needed for your virtual machine.

OUTPUT:



General
Name: ss
Operating System: Ubuntu (64-bit)

System
Base Memory: 4608 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display
Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage
Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: ss.vdi (Normal, 10.00 GB)

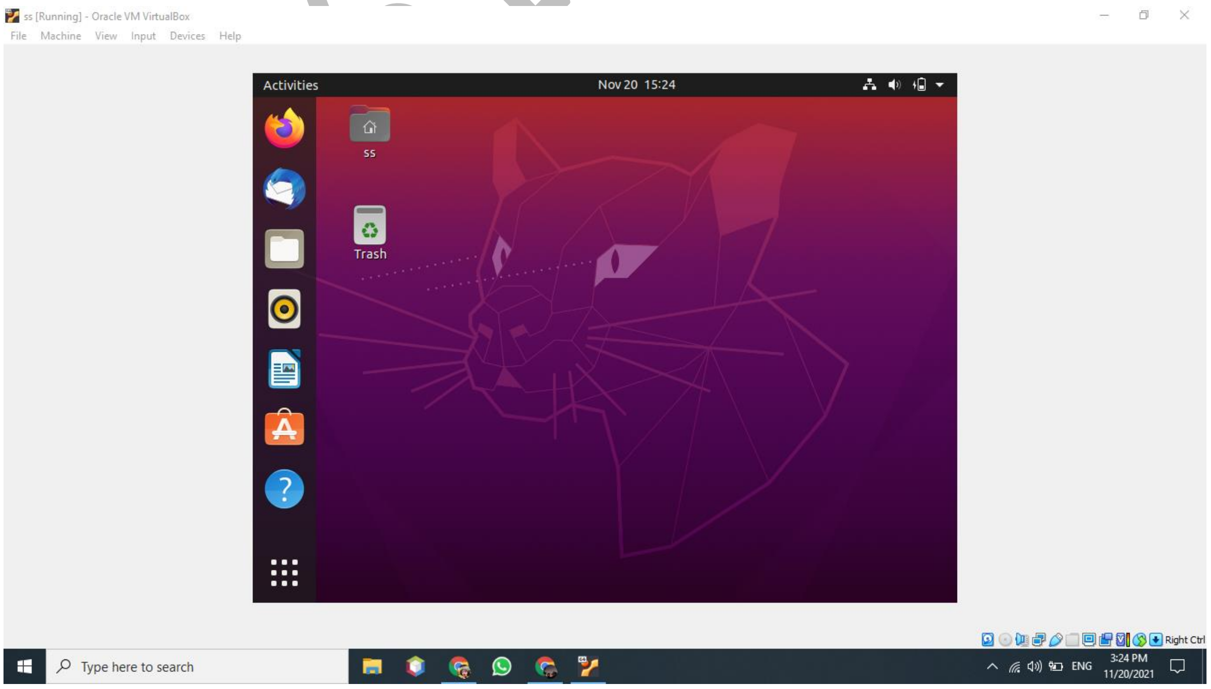
Audio
Host Driver: Windows DirectSound
Controller: ICH AC97

Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB
USB Controller: OHCI
Device Filters: 0 (0 active)

Shared folders
None

Description
None



Procedure:

Step1:

Install the centos or ubuntu in the opennebula as per previous commands.

Step 2:

Login into the VM of installed OS.

Step 3:

If it is ubuntu then, for gcc installation

```
$ sudo add-apt-repository ppa:ubuntu-toolchain-r/test
$ sudo apt-get update
$ sudo apt-get install gcc-6 gcc-6-base
```

Step 4:

Write a sample program like hello.c

```
#include<stdio.h>
int main()
{
Print("Hello, I am Surya Prakash");
return 0;
}
```

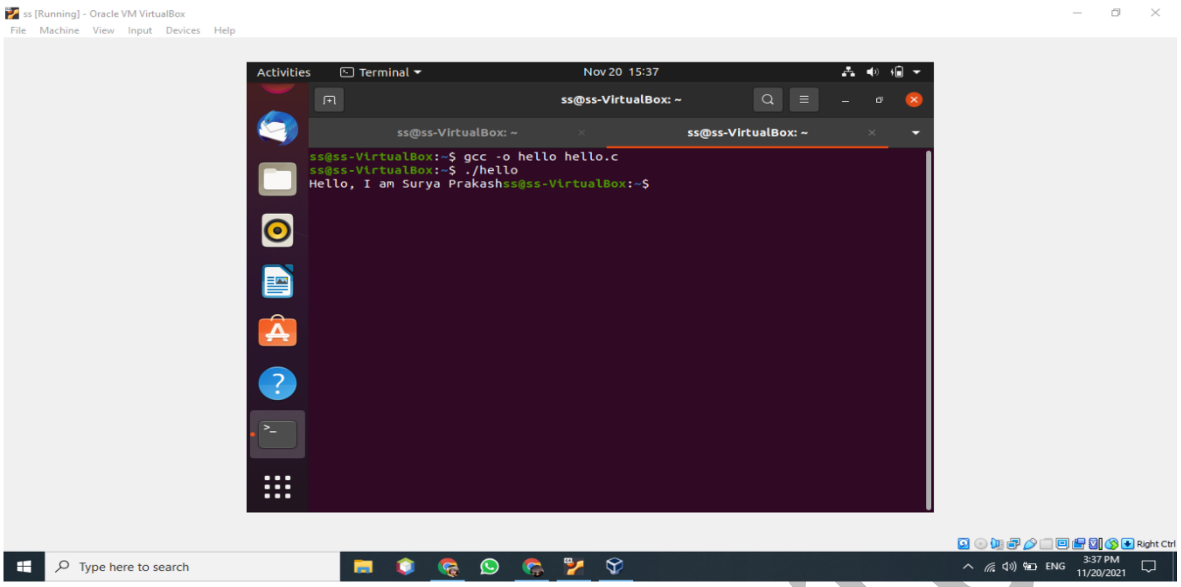
Step 5:

First we need to compile and link our program. Assuming the source code is saved in a file hello.c, we can do that using C compiler gcc, for example

```
gcc -o hello hello.c
```

And output can be executed by ./hello

OUTPUT:



```
ss@ss-VirtualBox:~$ gcc -o hello hello.c
ss@ss-VirtualBox:~$ ./hello
Hello, I am Surya Prakashss@ss-VirtualBox:~$
```

Procedure:
Use Eclipse to create a Google App Engine (GAE) Java project (hello world example), run it locally, and deploy it to Google App Engine account.

Tools used:

1. JDK 1.6
2. Eclipse 3.7 + Google Plugin for Eclipse
3. Google App Engine Java SDK 1.6.3.1

Note:

GAE supports Java 1.5 and 1.6.
P.S Assume JDK1.6 and Eclipse 3.7 are installed.

1. Install Google Plugin for Eclipse

   Read this guide – how to install Google Plugin for Eclipse. If you install the Google App Engine Java SDK together with "Google Plugin for Eclipse" then go to step 2, Otherwise, get the Google App Engine Java SDK and extract it.
2. Create New Web Application Project

   In Eclipse toolbar, click on the Google icon, and select "New Web Application Project…

Click finished, Google Plugin for Eclipse will generate a sample project automatically.
3. Hello World
The extra is this file "appengine-web.xml", Google App Engine need this to run and deploy the application.

File : appengine-web.xml

```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
  <application></application>
  <version>1</version>
  <!-- Configure java.util.logging -->
  <system-properties>
    <property name="java.util.logging.config.file" value="WEB-
    INF/logging.properties"/>
  </system-properties>
</appengine-web-app>
```

4. Run it local
Right click on the project and run as “**Web Application**“.

Eclipse console :

```
INFO: The server is running at http://localhost:8888/

30 Mac 2012 11:13:01 PM
com.google.appengine.tools.development.DevAppServerImpl start INFO:
The admin console is running at http://localhost:8888/_ah/admin
```

Access URL http://localhost:8888/, see output

and also the hello world servlet – http://localhost:8888/helloworld

5. Deploy to Google App Engine

Register an account on https://appengine.google.com/, and create an application ID for your web application.

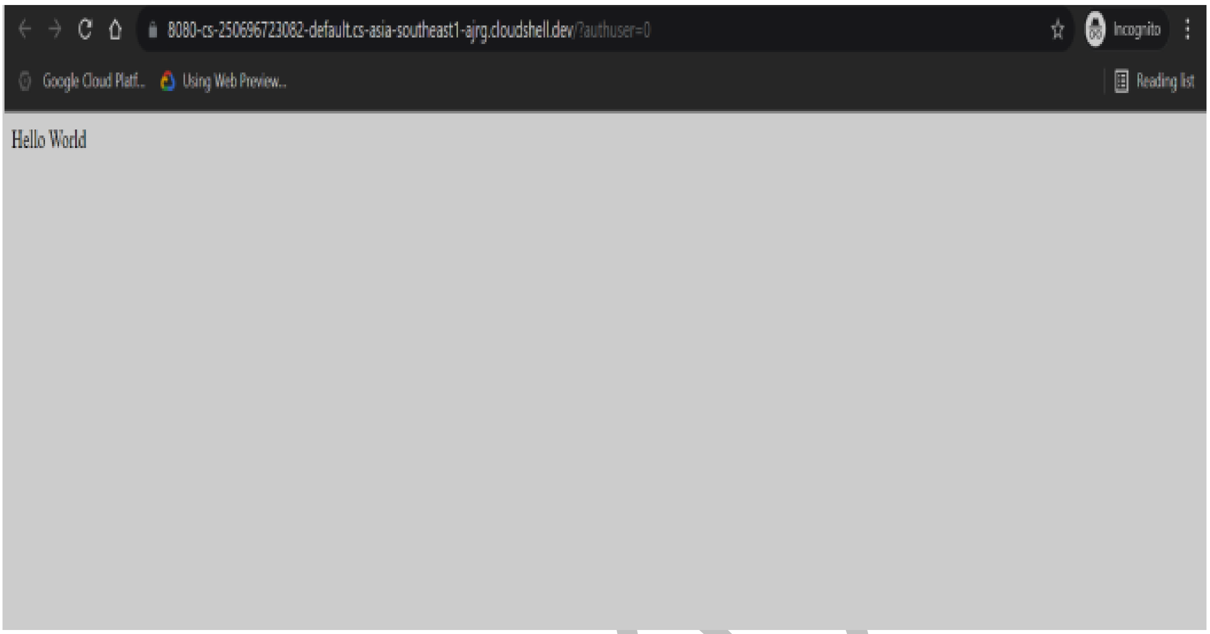
IIn this demonstration, I created an application ID, named “mkyong123”, and put it in appengine- web.xml.

```
<?xml version="1.0" encoding="utf-8"?>

<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">

<application>mkyong123</application>

<version>1</version>

<!-- Configure java.util.logging -->

<system-properties>

<property name="java.util.logging.config.file" value="WEB-
INF/logging.properties"/>

</system-properties>

</appengine-web-app>
```

To deploy, see following steps:
- Click on GAE deploy button on the toolbar.

– If everything is fine, the hello world web application will be deployed to this URL – http://mkyong123.appspot.com/

OUTPUT:



Hello World

Procedure:

You can use Google App Engine to host a static website. Static web pages can contain client-side technologies such as HTML, CSS, and JavaScript

Sites hosted on App Engine are hosted on the REGION_ID.r.appspot.com subdomain, such as [my-project-id].uc.r.appspot.com. \

Before you begin

1. Create a new Cloud Console project or retrieve the project ID of an existing project to use: Go to the Project page

Before you can host your website on Google App Engine:

Tip: You can retrieve a list of your existing project IDs with the gcloud command line tool.

2. Install and then initialize the Google Cloud SDK: Download the SDK

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.

Creating the app.yaml file:

The app.yaml file is a configuration file that tells App Engine how to map URLs to your static files. In the following steps, you will add handlers that will load www/index.html when someone visits your website, and all static files will be stored in and called from the www directory.

Create the app.yaml file in your application's root directory:

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:

```
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/(.*)
```

More reference information about the app.yaml file can be found in the app.yaml reference documentation.

Creating the index.html file

Create an HTML file that will be served when someone navigates to the root page of your website. Store this file in your www directory.

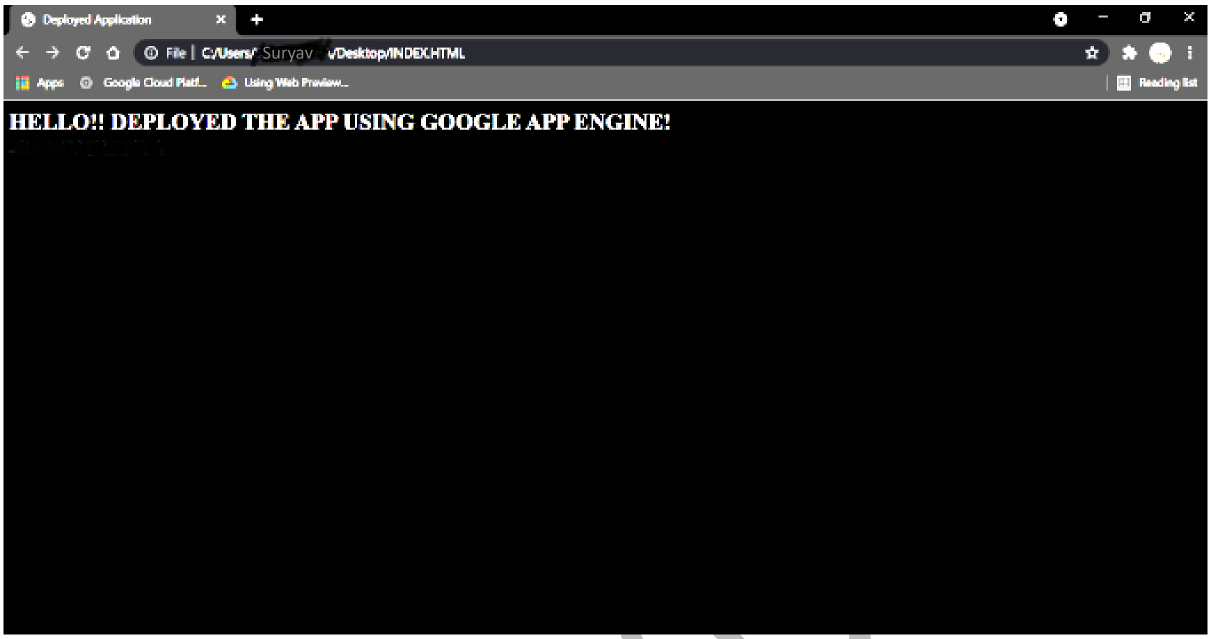
```
<html>
<head>
<title>Hello, world!</title>
<link rel="stylesheet" type="text/css" href="/css/style.css">
</head>
<body>
<h1>Hello, world!</h1>
<p>
This is a simple static HTML file that will be served
from Google App Engine.
</p>
</body>
</html>
```

Deploying your application to App Engine

When you deploy your application files, your website will be uploaded to App Engine. To deploy your app, run the following command from within the root directory of your application where the app.yaml file is located:

```
gcloud app browse
```

OUTPUT:


Deployed Application

File | C:/Users/ Suryav [redacted]/Desktop/INDEX.HTML

Apps Google Cloud Platf... Using Web Preview... Reading list

HELLO!! DEPLOYED THE APP USING GOOGLE APP ENGINE!

Procedure:

Basics of Scheduling

In computers, Scheduling is a process of arranging the submitted jobs/task into a very specific sequence of execution. It is an essential characteristic of any software operating environment, which is handled by a very special program known as a scheduler.

Space-shared: In this, the requested resources are allocated dedicatedly to the requesting workload for execution and will be released only on completion. **Space-shared is also known as a batch process scheduling.**

- **Time-shared:** In this, the requested resources would be shared among more than one workload(task). The sharing is done based on time-sliced allocation where each workload is allocated with a required resource for a defined time(e.g., 200 milliseconds). Once the defined time slice is over, the current workload execution paused, and the resource is released. The released resource gets allocated to the next workload for the same defined time slice, and this cycle goes on till the time all the workloads execution is over. Time- shared is also known as round-robin **scheduling.**

Scheduling in Cloud

As cloud computing is the virtualized operating environment, and the virtual machines are the primary computing component which is responsible for the execution of the workloads(tasks). The virtual machine(s) are powered by a physical server host machine (i.e.) hardware. Depending on the requirement of the Virtual Machine(VM) there could be ‘one to one’ or ‘many to one’ mapping between the VM and host machine. That means in cloud computing the scheduling is done at both the mapping levels that are:

- Virtual Machine to Host Machines
- Tasks to Virtual Machines

Both of VM to Host as well as Workload(task) to VM mappings may utilize space-share or time- shared or any other specialized scheduling algorithm.

Scheduling in Cloudsim

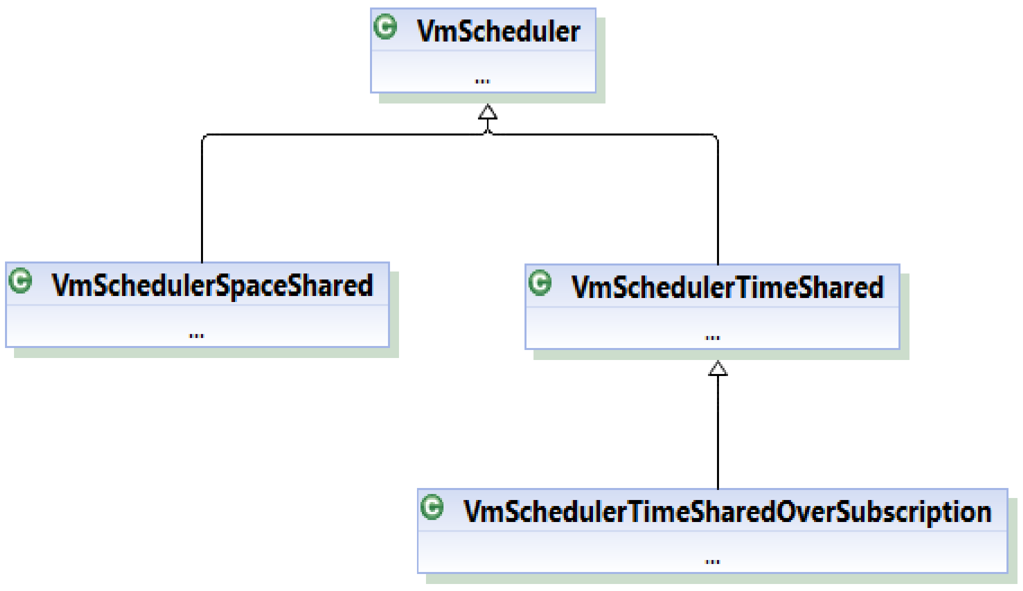
The Cloudsim simulation toolkit framework has effectively addressed the Scheduling scenario and implemented it as a set of the programmable class hierarchies with parent class as:

1. VmScheduler
2. CloudletScheduler

<u>Cloudsim Virtual Machine Scheduling</u>

The **VmScheduler** is an abstract class that defines and implements the policy used to share processing power among virtual machines running on a specified host. The hierarchy of the cloudsim virtual machine scheduler classes is as:

Cloudsim Virtual Machine Scheduler Class Hierarchy



These classes can be located in "*org.cloudbus.cloudsim*" package of cloudsim. The definition of this abstract class is extended to the following types of policies implemented as classes:

 VmSchedulerTimeShared: This class implements the VM scheduling policy that allocates one or more processing elements to a single Virtual machine and allows the sharing of processing elements by multiple virtual machines with a specified time slice. This class also considers the overhead of VM allocation switching(similar to context switching) in policy definition.

 VmSchedulerSpaceShared: This class implements the VM scheduling policy that allocates one or more processing elements to a single virtual machine, but this policy implementation does not support sharing of processing elements (i.e.) all the requested resources will be used by the allocated VM till the time the VM is not destroyed.

 VmSchedulerTimeSharedOverSubscription: This is an extended implementation of VMSchedulerTimeShared VM scheduling policy, which allows over-subscription of processing elements by the virtual machine(s) (i.e.) the scheduler still allows the allocation of VMs that require more CPU capacity that is available. And this oversubscription results in performance degradation.

Following is the code snippet used in CloudsimExample1.java from line number 160 to 174:

```
 int hostId = 0;
int ram = 2048; // host memory (MB)
long storage = 1000000; // host storage
int bw = 10000;
hostList.add(
            new Host(
                 hostId,
                 new RamProvisionerSimple(ram),
                         new
                         BwProvisionerSimple(bw),
                         storage,
                 peList,
                 new VmSchedulerTimeShared(peList)
      )
      ); //This is our machine
```

This is where the processing element list is passed as a parameter to the VmSchedulerTimeShared() class call and during the simulation, the cloudsim will simulate the timeshare behavior for the virtual machines. Also, in case you want to test other VmScheduler you may replace it with VmSchedulerTimeShared() call with appropriate parameters, this includes your own designed custom virtual machine scheduler.

Cloudsim Cloudlet Scheduling

The “***CloudletScheduler*”** is an abstract class that defines the basic skeleton to implement the policy to be used for cloudlet scheduling to be performed by a virtual machine.

Cloudlet Scheduler Class Hierarchy

These classes again exist in "*org.cloudbus.cloudsim*" package of cloudsim. The definition of this abstract class is extended as the following types of policies implemented as three individual classes in cloudsim:

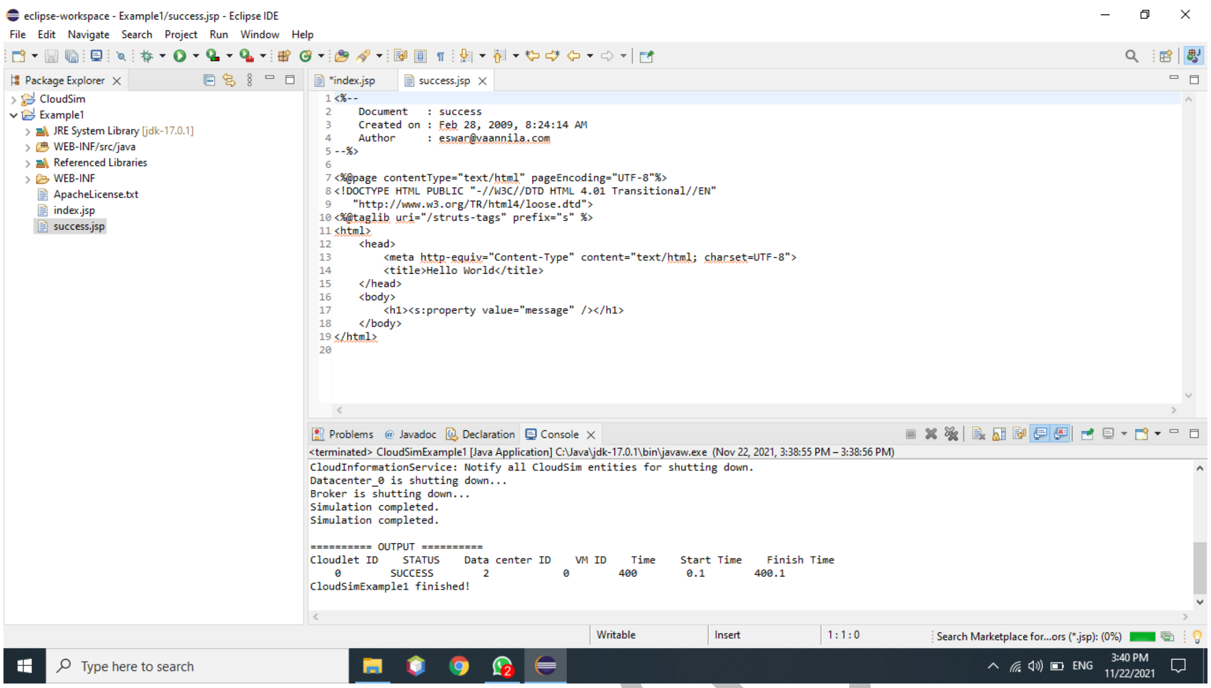
- **CloudletSchedulerSpaceShared**: This class implements a policy of scheduling for Virtual machine to execute cloudlet(s) in space shared environment (i.e.) only one cloudlet will be executed on a virtual machine at a time. It means cloudlets share the same queue and requests are processed one at a time per computing core. *Space-sharing* is similar to batch processing.
- **CloudletSchedulerTimeShared**: This class implements a policy of cloudlet scheduling for Virtual machines to execute cloudlets in a time-shared environment (i.e.) more than one cloudlet will be submitted to the virtual machine and each will get its specified share of time. It means several requests (cloudlets) are processed at once but they must share the computing power of that virtual machine(by simulating context switching), so they will affect each other's processing time. It basically influences the completion time of a cloudlet in CloudSim. Time-sharing is probably referring to the concept of sharing executing power (such as CPU, logical processor, GPU) and is commonly known as the round-robin scheduling.
- **CloudletSchedulerDynamicWorkload:** This implements a special policy of scheduling for virtual machine assuming that there is just one cloudlet which is working as an online service with a different requirement of workload as per the need of peak/offpeak user load at a specified period of time.

The application of the CloudletScheduler classes is while instantiating the Vm model. Following is the code snippet used in CloudsimExample1.java from line number 82 to 91:

```
int vmid = 0;
int mips = 1000;
long size = 10000; // image
size (MB) int ram = 512; //
vm memory (MB) long bw
= 1000;
int pesNumber = 1; // number
of cpus String vmm = "Xen";
// VMM name
Vm vm = new Vm(vmid, brokerId, mips, pesNumber,ram, bw, size, vmm,
        new CloudletSchedulerTimeShared()); // create VM
```

By instantiating the CloudletSchedulerTimeShared() class, the Virtual machine is decided to
follow the timeshare(round-robin) approach while simulation for scheduling & executing the Cloudlets. Also, in case you want to test other CloudletScheduler you may replace it with CloudletSchedulerTimeShared() call with appropriate parameters, this includes your own designed custom cloudlet scheduler.

OUTPUT:



```
<%--
    Document    : success
    Created on : Feb 28, 2009, 8:24:14 AM
    Author      : eswar@vaannila.com
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">
<%@taglib uri="/struts-tags" prefix="s" %>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Hello World</title>
    </head>
    <body>
        <h1><s:property value="message" /></h1>
    </body>
</html>
```

```
<terminated> CloudSimExample1 [Java Application] C:\Java\jdk-17.0.1\bin\javaw.exe  (Nov 22, 2021, 3:38:55 PM – 3:38:56 PM)
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    0         SUCCESS        2              0        400       0.1          400.1
CloudSimExample1 finished!
```

PROCEDURE:

How to Enable File sharing in VirtualBox.

Step 1. Install Guest Additions on the Guest machine.

Step 2. Configure File Sharing on VirtualBox.

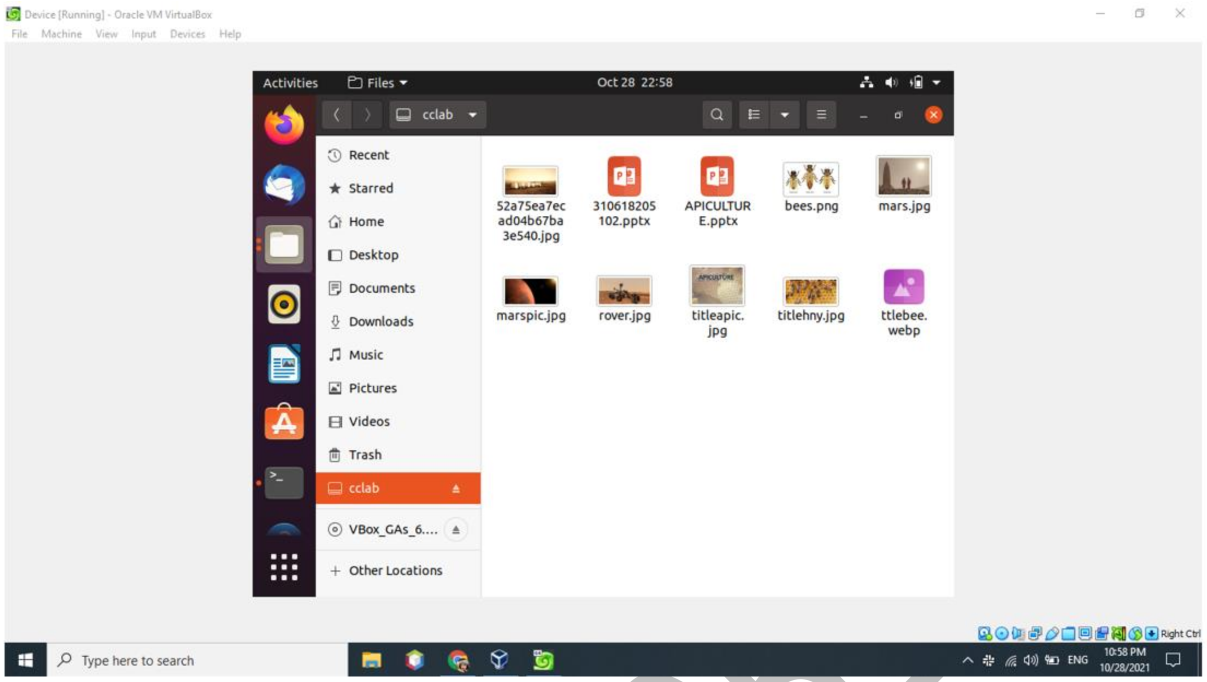
Step 1. Install Guest Additions on the Guest machine.

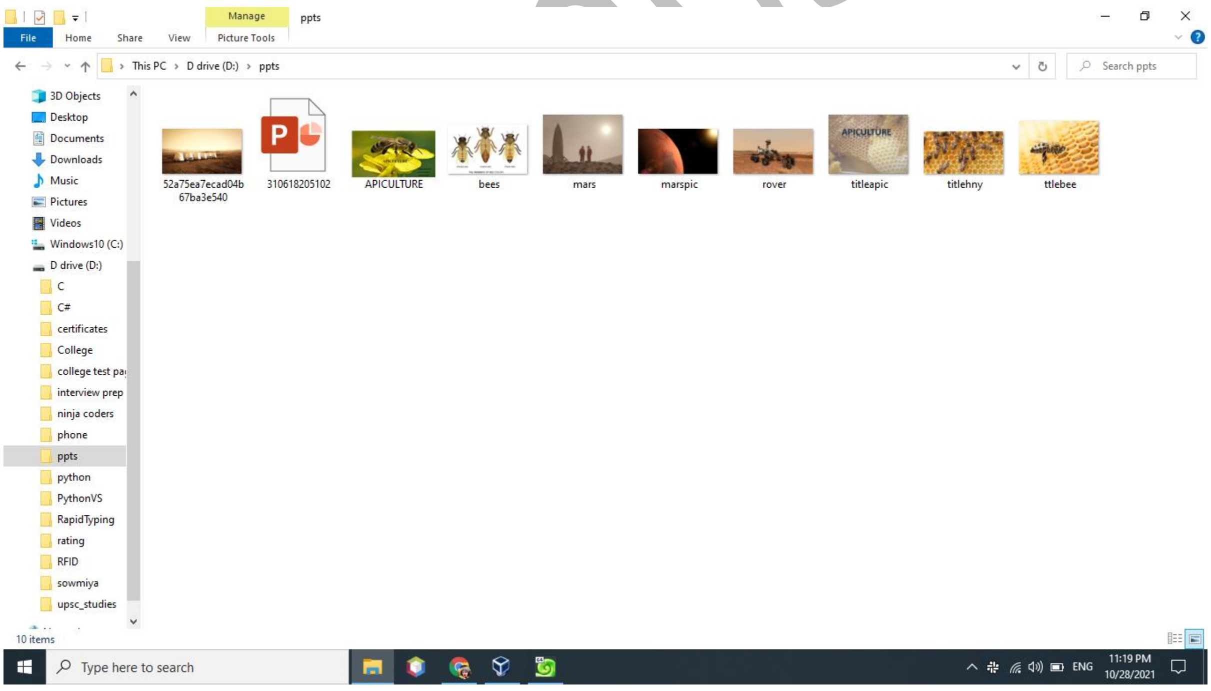
1. Start the Virtuabox Guest Machine (OS).
2. From Oracle's VM VirtualBox main menu, select **Devices** > **Install Guest Additions**
   a. Open Windows Explorer.
   b. Double click at the "CD Drive (X:) VirtualBox Guest additions" to explore its contents.
   c. Right click at "VBoxWindowsAdditions" application and from the pop-up menu, choose "**Run as administrator**
3. Press **Next** and then follow the on screen instructions to complete the Guest Additions installation.
4. When the setup is completed, choose **Finish** and **restart** the Virtuabox guest machine

Step 2. Setup File Sharing on VirtualBox Guest Machine.

1. From VirtualBox menu click **Devices** and choose **Shared Folders** -> **Shared Folder Settings.**
2. Click the **Add new shared folder**
3. Click the drop-down arrow and select **Other**.
4. Locate and highlight (from the Host OS) the folder that you want to share between the VirtualBox Guest machine and the Host and click Select Folder. *
5. Now, in the 'Add Share' options, type a name (if you want) at the 'Folder Name box, click the **Auto Mount** and the **Make Permanent** checkboxes and click **OK** twice to close the Shared Folder Settings
6. You 're done! To access the shared folder from the Guest OS, open Windows Explorer and under the 'Network locations' you should see a new network drive that corresponds to the shared folder on the Host OS

OUTPUT:


Device [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Files
Oct 28 22:58
cclab
Recent
Starred
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
cclab
VBox_GAs_6....
Other Locations
52a75ea7ecad04b67ba3e540.jpg
310618205102.pptx
APICULTURE.pptx
bees.png
mars.jpg
marspic.jpg
rover.jpg
titleapic.jpg
titlehny.jpg
ttlebee.webp
Type here to search
10:58 PM
10/28/2021


Manage
ppts
File Home Share View Picture Tools
This PC > D drive (D:) > ppts
Search ppts
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows10 (C:)
D drive (D:)
C
C#
certificates
College
college test pa
interview prep
ninja coders
phone
ppts
python
PythonVS
RapidTyping
rating
RFID
sowmiya
upsc_studies
52a75ea7ecad04b67ba3e540
310618205102
APICULTURE
bees
mars
marspic
rover
titleapic
titlehny
ttlebee
10 items
Type here to search
11:19 PM
10/28/2021

PROCEDURE:

Mandatory prerequisite:

1. Linux 64 bit Operating System (The commands mentioend are for Ubuntu Linux Operating System latest version).

Installing KVM (Hypervisor for Virtualization)

1. Check if the Virtualization flag is enabled in BIOS Run the command in terminal

```
egrep -c '(vmx|svm)' /proc/cpuinfo
```

If the result is any value higher than 0, then virtualization is enabled.
If the value is 0, then in BIOS enable Virtualization – Consult system administrator for this step.

2. To check if your OS is 64 bit, Run the command in terminal

```
uname -m
```

If the result is x86_64, it means that your Operating system is 64 bit Operating system.

3. Few KVM packages are availabe with Linux installation. To check this, run the command,

```
ls /lib/modules/{press tab}/kernel/arch/x86/kvm
nagarajan@JBL01:~$ ls /lib/modules/4.4.0-21-generic/kernel/arch/x86/kvm
```

The three files which are installed in your system will be displayed

```
kvm-amd.kokvm-intel.kokvm.ko
```

4. Install the KVM packages

- Switch to root (Administrator) user

```
sudo -i
export http_proxy=http://172.16.0.3:8080
```

- To install the packages, run the following commands,

```
t-get update
apt-get install qemu-kvm
apt-get install libvirt-bin
apt-get install bridge-util
```

5. To verify your installation, run the command

virsh -c qemu:///system list

it shows output

```
Id                    Name        State
        ----------------------------
```

If Vms are running, then it shows name of VM. If VM is not runnign, the system shows blank output, whcih means your KVM installation is perfect.

6. Run the command

virsh --connect qemu:///system list --all

7. Working with KVM

run the command

virsh

version (this command displays version of software tools installed)

nodeinfo (this command displays your system information)

quit (come out of the system)

8. To test KVM installation - we can create Virtual machines but these machines are to be done in manual mode. Skipping this, Directly install Openstack.

Installation of Openstack

1. add new user named stack – This stack user is the adminstrator of the openstack services. To add new user – run the command as root user.

adduser stack

2. run the command

apt-get install sudo -y || install -y sudo

3. Be careful in running the command – please be careful with the syntax. If any error in thsi following command, the system will crash beacause of permission errors

echo "stack ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers

4. Logout the system and login as stack user

5. Run the command (this installs git repo package)
Run in Root
export http_proxy=http://172.16.0.3:8080
sudo apt-get install git
6. Run the command (This clones updatesd version of dev-stack (which is binary auto-installer package of Openstack)
stack@JBL01:/$ **export http_proxy=http://172.16.0.3:8080** stack@JBL01:/$ **export https_proxy=http://172.16.0.3:8080** stack@JBL01:/$ **git config --global http.proxy $http_proxy** stack@JBL01**:/$ git config --global https.proxy $http_proxy**
git clone http://git.openstack.org/openstack-dev/devstack ls (this shows a folder named devstack)
cd devstack (enter into the folder)
7. create a file called **local.conf**. To do this run the command,
nano local.conf
stack@JBL01:/devstack$ **sudo nano local.conf**
8. In the file, make the following entry (Contact Your Network Adminstrator for doubts in these values)

```
[[local|localrc]]
FLOATING_RANGE=192.168.1.224/27
FIXED_RANGE=10.11.11.0/24
FIXED_NETWORK_SIZE=256
FLAT_INTERFACE=eth0
ADMIN_PASSWORD=root
DATABASE_PASSWORD=root
RABBIT_PASSWORD=root
SERVICE_PASSWORD=root
SERVICE_TOCKEN=root
```

9. Save this file
9.1stack@JBL01:/devstack$ **sudo gedit stackrc**
Save this file
Change File Permission: stack@JBL01:~$ chown stack * - R

10. Run the command (This installs Opentack)

./stack.sh

11. If any error occurs, then run the command for uninistallation

./unstack.sh

1.update the packages

apt-get update

2.Then reinstall the package

./stack.sh

12. Open the browser, http://IP address of your machine, you will get the openstack portal.
13. If you restart the machine, then to again start open stack

open terminal,

su stack

cd devstack

run ./rejoin.sh

14. Again you can access openstack services in the browser, http://IP address of your machine

<u>VIRTUAL MACHINE CREATION</u>

Launch an instance

1. Log in to the dashboard
2. Select the appropriate project from the drop down menu at the top left.
3. On the Project tab, open the Compute tab and click Instances category.

The dashboard shows the instances with its name, its private and floating IP addresses, size, status, task, power state, and so on.

4. Click Launch Instance.
5. In the Launch Instance dialog box, specify the following values:

Details tab

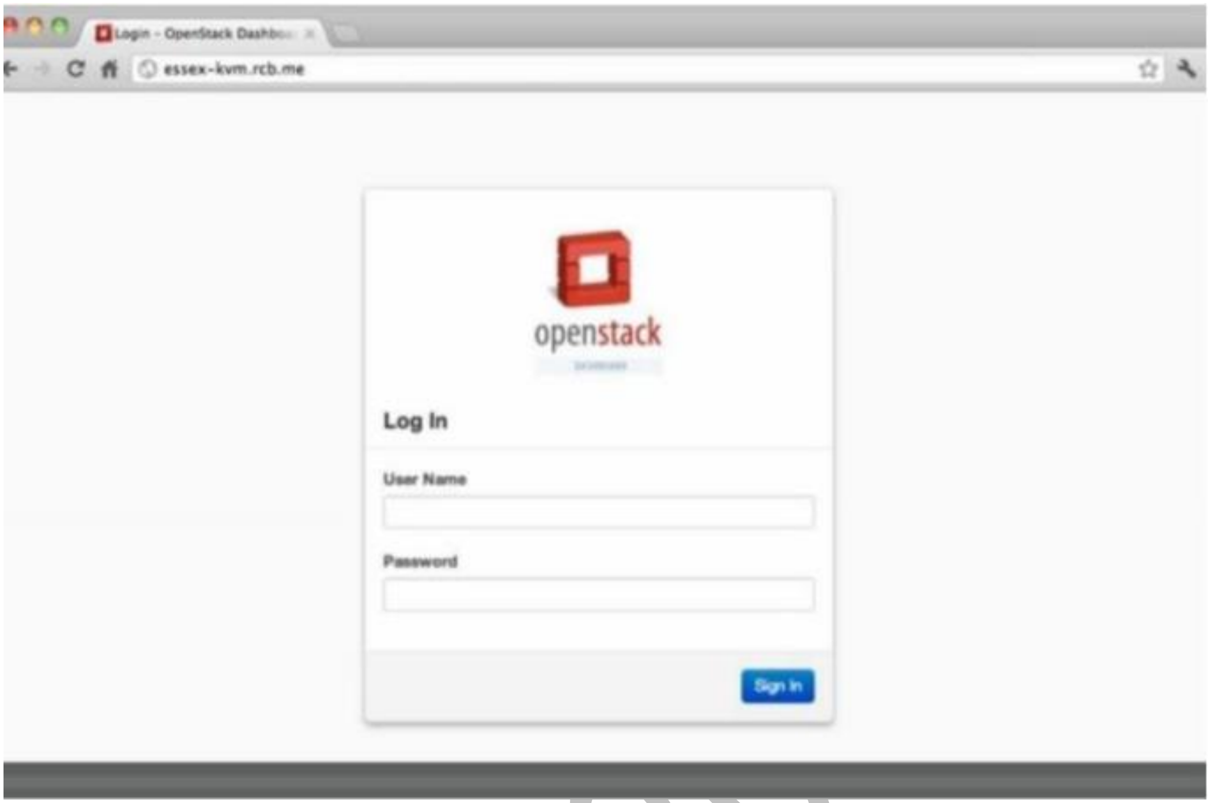
Availability Zone

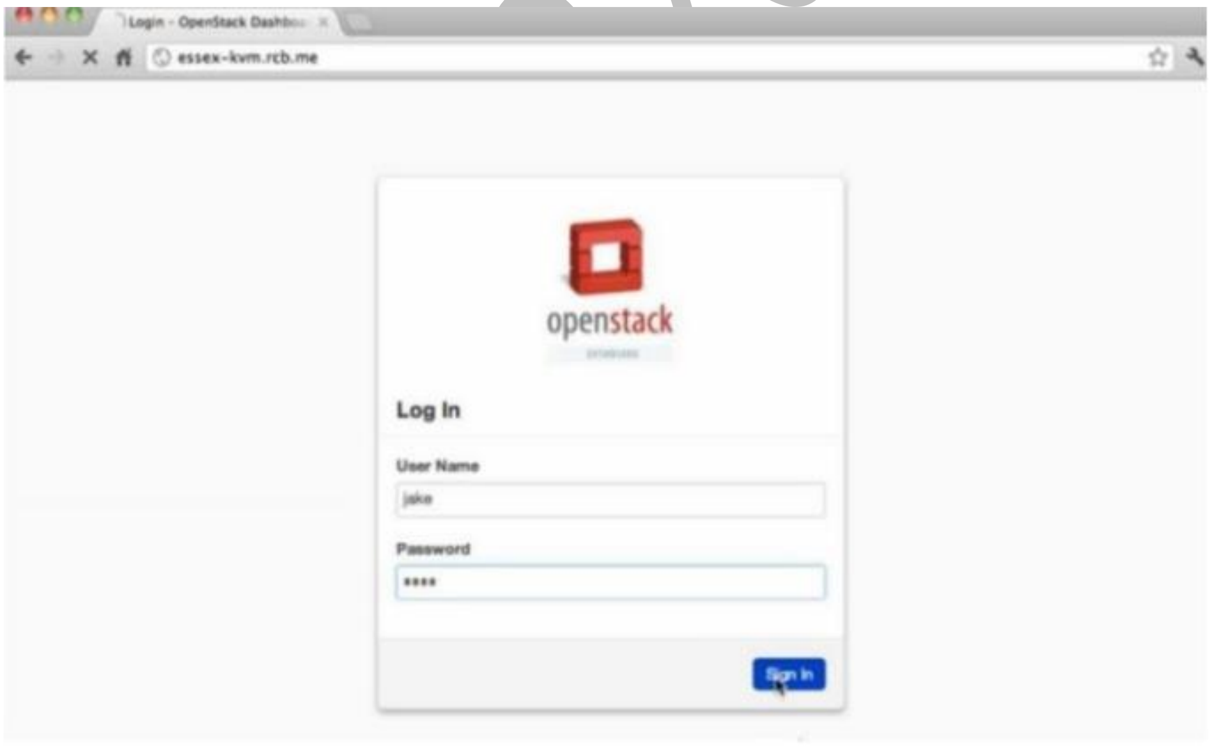
By default, this value is set to the availability zone given by the cloud provider

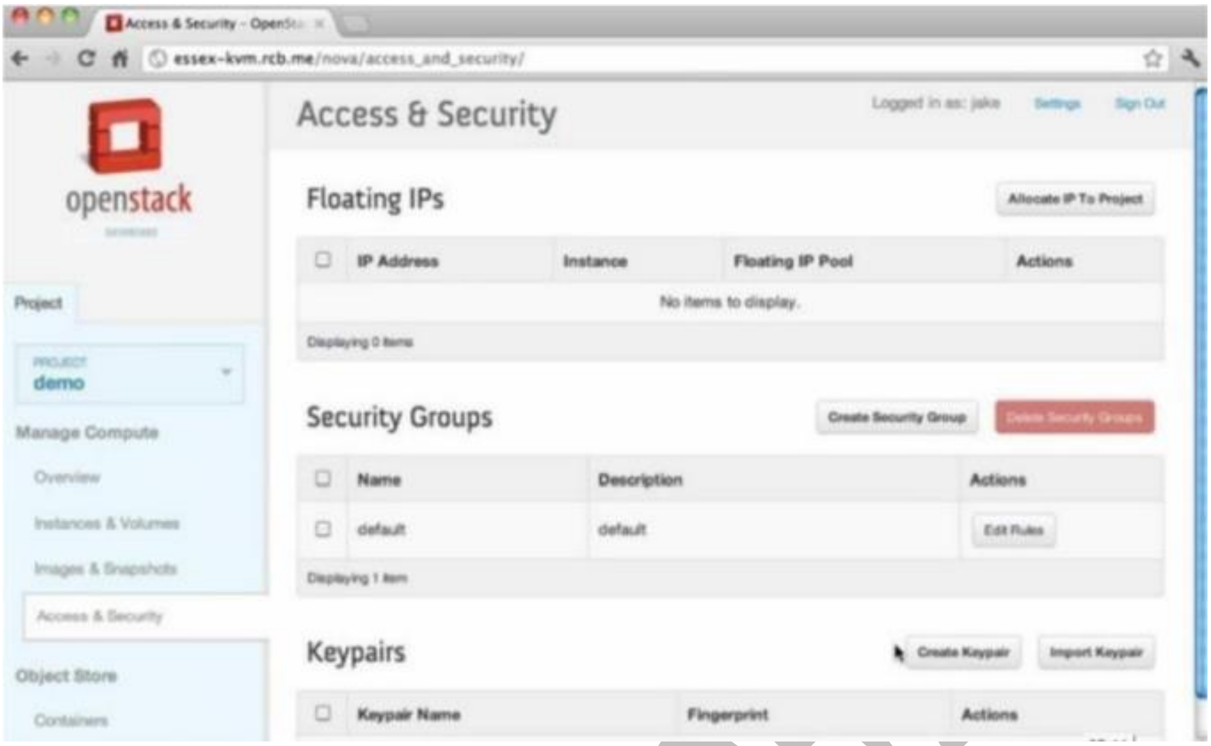
Instance Name

Assign a name to the virtual machine

OUTPUT:


Login - OpenStack Dashboard
essex-kvm.rcb.me
openstack
Log In
User Name
Password
Sign In


Login - OpenStack Dashboard
essex-kvm.rcb.me
openstack
Log In
User Name
jake
Password
****
Sign In


Access & Security - OpenSt...

essex-kvm.rcb.me/nova/access_and_security/

openstack

Project

PROJECT
demo

Manage Compute

Overview

Instances & Volumes

Images & Snapshots

Access & Security

Object Store

Containers

Access & Security

Logged in as: jake Settings Sign Out

Floating IPs

Allocate IP To Project

☐	IP Address	Instance	Floating IP Pool	Actions
No items to display.				

Displaying 0 items

Security Groups

Create Security Group   Delete Security Groups

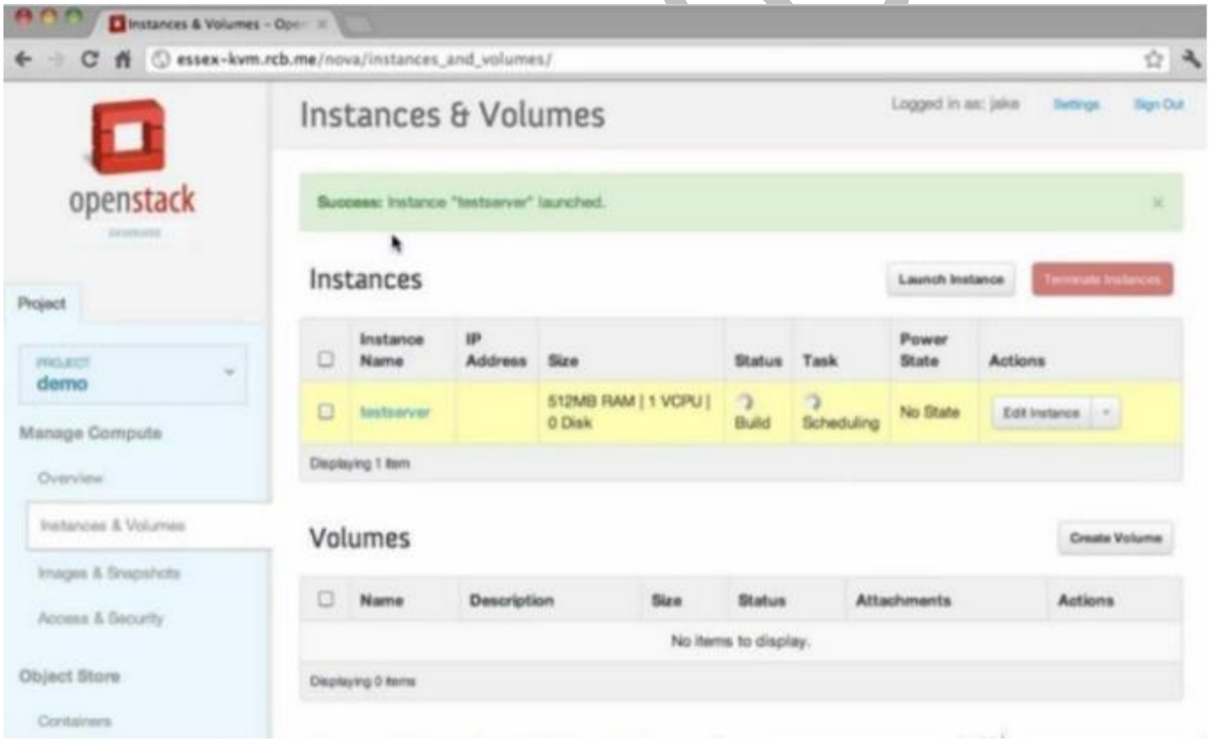
☐	Name	Description	Actions
☐	default	default	Edit Rules

Displaying 1 item

Keypairs

Create Keypair   Import Keypair

☐	Keypair Name	Fingerprint	Actions


Instances & Volumes - Ope...

essex-kvm.rcb.me/nova/instances_and_volumes/

openstack

Project

PROJECT
demo

Manage Compute

Overview

Instances & Volumes

Images & Snapshots

Access & Security

Object Store

Containers

Instances & Volumes

Logged in as: jake Settings Sign Out

Success: Instance "testserver" launched.

Instances

Launch Instance   Terminate Instances

☐	Instance Name	IP Address	Size	Status	Task	Power State	Actions
☐	testserver		512MB RAM | 1 VCPU | 0 Disk	Build	Scheduling	No State	Edit Instance

Displaying 1 item

Volumes

Create Volume

☐	Name	Description	Size	Status	Attachments	Actions
No items to display.						

Displaying 0 items

Procedure:

1) Installing Java:

Hadoop is a framework written in Java for running applications on large clusters of commodity hardware. Hadoop needs Java 6 or above to work.

Step 1: Download Jdk tar.gz file for linux-62 bit, extract it into "/usr/local"

```
boss@solaiv[]# cd /opt
boss@solaiv[]#      sudo  tar    xvpzf /home/itadmin/Downloads/jdk-8u5-linux-
x64.tar.gz boss@solaiv[]# cd /opt/jdk1.8.0_05
```

Step 2:

Open the "/etc/profile" file and Add the following line as per the version seta environment for Java

The 'profile' file contains commands that ought to be run for login shells

```
boss@solaiv[]# sudovi /etc/profile
#--insert JAVA_HOME
JAVA_HOME=/opt/jdk1.8.0_05
#--in PATH variable just append at the end of the line PATH=$PATH:$JAVA
_HOME/bin
#--Append JAVA_HOME at end of the export statement export PATH JAVA_HOME
```

save the file using by pressing "Esc" key followed by :wq!

Step 3: Source the /etc/profile boss@solaiv[]# source /etc/profile Step 4: Update the java alternatives

Step 4: Update the java alternatives

By default OS will have a open jdk. Check by "java -version". You will be prompt "openJDK"

If you also have openjdk installed then you'll need to update the java alternatives:

If your system has more than one version of Java, configure which one your system causes by entering the following command in a terminal window

By default OS will have a open jdk. Check by "java -version". You will be prompt "Java HotSpot(TM) 64-Bit Server"

```
boss@solaiv[]#      update-alternatives --install     "/usr/bin/java"     java
"/opt/jdk1.8.0_05/bin/java" 1
boss@solaiv[]# update- alternatives --config java --type selection number:
boss@solaiv[]# java -version
```

1) configure ssh

Hadoop requires SSH access to manage its nodes, i.e. remote machines plus your local machine if you want to use Hadoop on it (which is what we want to do in this short tutorial). For our single-node setup of Hadoop, we therefore need to configure SSH access to localhost

The need to create a Password-less SSH Key generation based authentication is so that the master node can then login to slave nodes (and the secondary node) to start/stop them easily without any delays for authentication

Generate an SSH key for the user. Then Enable password-less SSH access to yo sudo apt-get install openssh-server

--You will be asked to enter password, root@solaiv[]# sshlocalhost root@solaiv[]# ssh-keygenroot@solaiv[]# ssh-copy-id -i localhost

--After above 2 steps, You will be connected without password, root@solaiv[]# sshlocalhost

root@solaiv[]# exiT

2) Hadoop installation

Now Download Hadoop from the official Apache, preferably a stable release version of Hadoop 2.7.x and extract the contents of the Hadoop package to a location of your choice.

We chose location as "/opt/"

Step 1: Download the tar.gz file of latest version Hadoop(hadoop-2.7.x) from the official site .

Step 2: Extract(untar) the downloaded file from this commands to /opt/bigdata

```
root@solaiv[]# cd /opt
root@solaiv[/opt]#   sudo  tar    xvpzf /home/itadmin/Downloads/hadoop-2.7.0.tar.gz
root@solaiv[/opt]# cd hadoop-2.7.0/
```

Like java, update Hadop environment variable in /etc/profile

```
boss@solaiv[]# sudovi /etc/profile
#--insert HADOOP_PREFIX HADOOP_PREFIX=/opt/hadoop-2.7.0
#--in PATH variable just append at the end of the
   line PATH=$PATH:$HADOOP_PREFIX/bin

#--Append HADOOP_PREFIX at end of the exportstatement
   exportPATH JAVA_HOME HADOOP_PREFIX
```

save the file using by pressing “Esc” key followed by :wq!
Step 3: Source the /etc/profile boss@solaiv[]# source /etc/profile Verify Hadoop installation
boss@solaiv[]# cd $HADOOP_PREFIX boss@solaiv[]# bin/hadoop version
3.1 Modify the Hadoop Configuration Files
Add the following properties in the various hadoop configuration files which is available under $HADOOP_PREFIX/etc/hadoop/
core-site.xml, hdfs-site.xml, mapred-site.xml & yarn-site.xml
Update Java, hadoop path to the Hadoop environment file boss@solaiv[]# cd $HADOOP_PREFIX/etc/hadoop

boss@solaiv[]# vi hadoop-env.sh
Paste following line at beginning of the file export JAVA_HOME=/usr/local/jdk1.8.0_05 export HADOOP_PREFIX=/opt/hadoop-2.7.0
Modify the core-site.xml
boss@solaiv[]# cd $HADOOP_PREFIX/etc/hadoopboss@solaiv[]# vi core-site.xml
Paste following between <configuration> tags

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

Modify the hdfs-site.xml boss@solaiv[]# vi hdfs-site.xml
Paste following between <configuration> tags

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>
```

YARN configuration - Single Node modify the mapred-site.xml boss@solaiv[]# cpmapred-site.xml.template mapred-site.xml boss@solaiv[]# vi mapred-site.xml
Paste following between <configuration> tags

```
<configuration>
```

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

Modiy yarn-site.xml boss@solaiv[]# vi yarn-site.xml
Paste following between <configuration> tags

```
<property><name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value></property>
</configuration>
```

Formatting the HDFS file-system via the NameNode

The first step to starting up your Hadoop installation is formatting the Hadoop files system which is implemented on top of the local file system of our "cluster" which includes only our local machine. We need to do this the first time you set up a Hadoop cluster.

Do not format a running Hadoop file system as you will lose all the data currently in the cluster (in HDFS)
To stop the running process root@solaiv[]# sbin/stop-dfs.sh
To know the running daemons jut type jps or /usr/local/jdk1.8.0_05/bin/jps Start ResourceManager daemon and NodeManager daemon: (port 8088) root@solaiv[]# sbin/stop-yarn.sh

PROCEDURE:

Update APT

```
test@cs-88:~$ sudo apt-get update
E: Unable to lock directory /var/lib/apt/lists/
```

Check Backgroud Process :

```
test@cs-88:~$ ps -ef |grep apt
root   2292  228  0 11:11 ?    00:00:00 /bin/sh
                5                  /etc/cron.daily/apt
root   2612  229  0 11:30 ?    00:00:00 apt-get -qq -y update
                2
root   2615  261  0 11:30 ?    00:00:00
                2                  /usr/lib/apt/methods/http
```

```
              root   2616  261  0 11:30 ?     00:00:00
                            2                  /usr/lib/apt/methods/http
              root   2617  261  0 11:30 ?     00:00:00
                            2                  /usr/lib/apt/methods/http
              root   2619  261  0 11:30 ?     00:00:00
                            2                  /usr/lib/apt/methods/gpgv
              root   2627  261  0 11:30 ?     00:00:01
                            2                  /usr/lib/apt/methods/bzip2
test        2829  2813  0 11:36 pts/000:00:00 grep --color=auto apt
```

Kill Backgroud Process :

```
test@cs-88:~$ sudo kill -9 2292 2612 2615 2616 2617 2619 2627 2829
```

Updaet apt :

```
test@cs-88:~$ sudo apt-get update
```

git installation :

```
root@cs-88:~# sudo apt-get install git
```

Clone :

```
root@cs-88:~# git clone https://git.openstack.org/openstack-dev/devstack

root@cs-88:~# ls devstack
root@cs-88:~# cd devstack
root@cs-88:~/devstack# nano local.conf [[local|localre]]
HOST_IP=192.168.4.88
// FLOATING_RANGE=192.168.1.224/27 FIXED_RANGE=10.11.12.0/24
FIXED_NETWORK_SIZE=256 FLAT_RANGE=eth0 ADMIN_PASSWORD=linux
DATABASE_PASSWORD=linux RABBIT_PASSWORD=linux
SERVICE-TOKEN=linux
```

Save Nono file:
control+x

Hadoop Installation

```
stack@cs-88:~/Downloads$
sudo scp -r * /opt/
```

```
stack@cs-88:~/Downloads$ ls /opt/
test@cs-88:/opt$ ls
hadoop-2.7.0.tar.gz Hadoop Pseudo-Node.pdf HDFSCommands.pdf jdk-8u60-linux-
x64

test@cs-88:/opt$ ls total 383412
drwxr-xr-x  2 root root         4096 May 6 15:41 ./
drwxr-xr-x 23 root root         4096 May 6 16:34 ../
-rw-r--r-- 1 root root 210343364 May 6 15:41 hadoop-2.7.0.tar.gz
-rw-r--r--  1 root root 159315 May 6 15:41 Hadoop Pseudo-Node.pdf
-rw-r--r--  1 root root 43496 May 6 15:41 HDFSCommands.pdf
-rw-r--r-- 1 root root 181238643 May 6 15:41 jdk-8u60-linux-x64.gz
-rw-r--r--  1 root root 402723 May 6 15:41 mrsampledata(1).tar.gz
-rw-r--r--  1 root root 402723 May 6 15:41 mrsampledata.tar.gz Change root user to test
user:
test@cs-88:/opt$ sudo chown -Rh test:test /opt/ test@cs-88:/opt$ ll /* display list file
with permission total 383412
-rw-r--r-- 1 test test 181238643 May 6 15:41 jdk-8u60-linux-x64.gz
-rw-r--r--  1 test test   402723 May 6 15:41 mrsampledata(1).tar.gz
-rw-r--r--  1 test test   402723 May 6 15:41 mrsampledata.tar.gz test@cs-88:/opt$
```

Unzip JAVA

```
test@cs-88:/opt$ tar -zxvf jdk-8u60-linux-x64.gz

test@cs-88:/opt$ cd jdk1.8.0_60 test@cs-88:/opt/jdk1.8.0_60$ pwd
/opt/jdk1.8.0_60 test@cs-88:/
```

Set profiloe for JAVA

```
est@cs-88:/opt/jdk1.8.0_60$ sudo nano /etc/profile
JAVA_HOME=/opt/jdk1.8.0_60 HADOOP_PREFIX=/opt/hadoop-2.7.0

PATH=$PATH:$JAVA_HOME/bin PATH=$PATH:$HADOOP_PREFIX/bin

export PATH JAVA_HOME HADOOP_PREFIX
```

Save: Control +x Press   y

```
Press      Enterkey
test@cs-88:/opt/jdk1.8.0_60$ cd .. test@cs-88:/opt$ pwd
/opt
test@cs-88:/opt$ Unzip hadoop file
test@cs-88:/opt$tar -zxvf hadoop-2.7.0.tar.gz test@cs-88:/opt$ source /etc/profile
Java Version
test@cs-88:/opt$ java -version
test@cs-88:/$ source /etc/profile test@cs-88:/$ java -version
java version "1.8.0_60"
Java(TM) SE Runtime Environment (build 1.8.0_60-b27)
Java HotSpot(TM) 64-Bit Server VM (build 25.60-b23, mixed mode
SSH keygeneration
test@cs-88:/opt$ ssh-keygen Generating public/private rsa key pair.
Enter file in which to save the key (/home/test/.ssh/id_rsa): Created directory
'/home/test/.ssh'.
Enter passphrase (empty for no passphrase): Enter same passphrase again:
Configure Hadoop
Verify Hadoop installation

test@cs-88:/opt$ cd $HADOOP_PREFIX test@cs-88:/opt/hadoop-2.7.0$ pwd
/opt/hadoop-2.7.0
test@cs-88:/opt/hadoop-2.7.0$

test@cs-88:/opt/hadoop-2.7.0$ ls
bin include  libexec    NOTICE.txt  sbin
etc lib     LICENSE.txt README.txt share

test@cs-88:/opt/hadoop-2.7.0$ bin/hadoop version Hadoop 2.7.0
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r
d4c8d4d4d203c934e8074b31289a28724c0842cf
Compiled by jenkins on 2015-04-10T18:40Z Compiled with protoc 2.5.0
From source with checksum a9e90912c37a35c3195d23951fd18f
This command was run using /opt/hadoop-2.7.0/share/hadoop/common/hadoop-
common- 2.7.0.jar
test@cs-88:/opt/hadoop-2.7.0$
```

Word count program to demonstrate the use of Map and Reduce tasks Procedure:

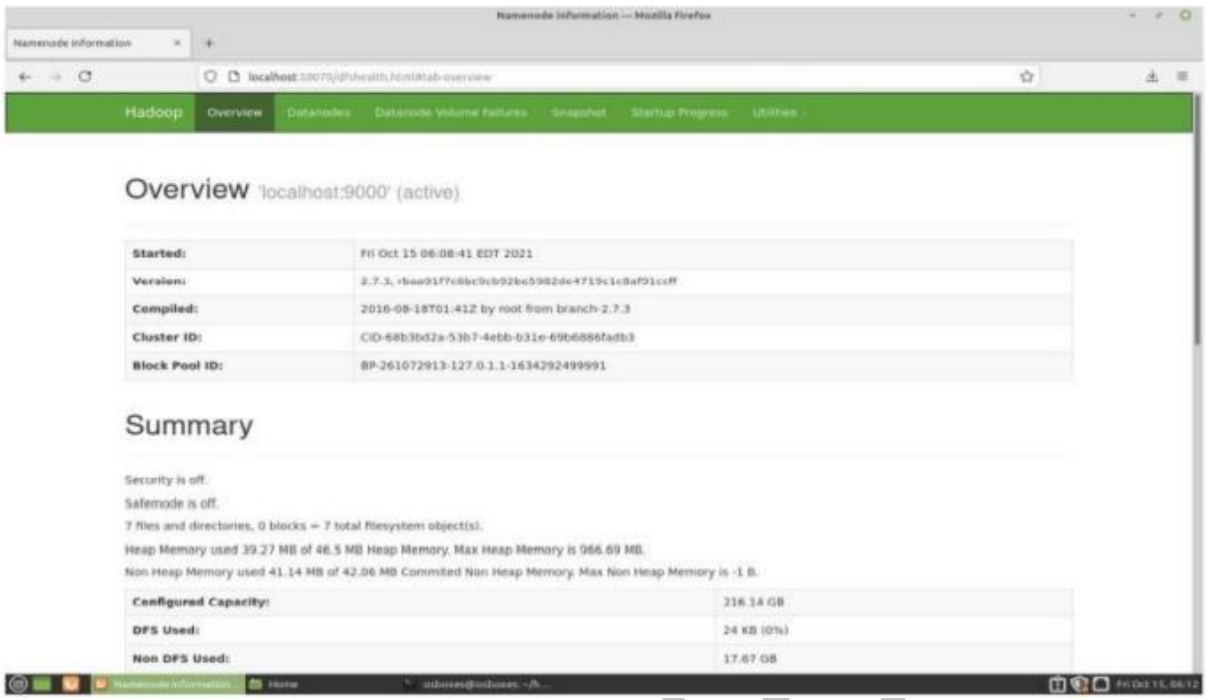
1. Format the path.
2. Start the dfs and check the no. of nodes running.
3. Start the yarn and check the no. of nodes running.
4. Open the browser and check whether the hadoop is installed correctly.
5. Add a file and check whether we can view the file.
6. Implement the grep command for the file added and see the result.
7. Implement the wordcount command for the file added and see the result.

After completing the process stop dfs and yarn properly.

PROGRAM:

```
packagehadoop; import java.util.* ;
importjava.io.IOException;
importjava.io.IOException;
importorg.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.* ;
import org.apache.hadoop.io.* ;
importorg.apache.hadoop.m apred.*
; import org.apache.hadoop.util.* ;
public class ProcessUnits
{
public static class E_EMapper extends MapReduceBaseimplements
Mapper<LongWritable Text,
Text, IntWritable>
{
public void m ap(LongWritable key, Text value,
OutputCollector<Text, IntWritable> output,
Reporter reporter) throws IOException
{
String line = value.toString(); String lasttoken = null;
StringTokenizer s = new StringTokenizer(line,"\t");
String year = s.nextToken();
while(s.hasMoreTokens())
{
lasttoken=s.nextToken();
}
intavgprice =
Integer.parseInt(lasttoken);
output.collect(new Text(year), new
IntWritable(avgprice));
}
}
public static class E_EReduce extends MapReduceBaseimplements
Reducer< Text, IntWritable, Text, IntWritable>
{
public void reduce( Text key, Iterator <IntWritable> values,
OutputCollector<Text, IntWritable> output, Reporter reporter) throws
IOException{
int m axavg=30; intval=Integer.MIN_VALUE; while
(values.hasNext())
{
if((val=values.next().get())>m axavg)
{
output.collect(key, new IntWritable(val));
}
}
}
}
public static void m ain(String args[])throws Exception
{
JobConfconf = new JobConf(Eleunits.class);
conf.setJobNam e("m ax_eletricityunits");
conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);
conf.setMapperClass(E_EMapper.class);
conf.setCombinerClass(E_EReduce.class);
conf.setReducerClass(E_EReduce.class);
conf.setInputForm at(TextInputFormat.class);
conf.setOutputForm at(TextOutputFormat.class);
FileInputFormat.setInputPaths(conf, new Path(args[0]));
FileOutputFormat.setOutputPath(conf, new
Path(args[1])); JobClient.runJob(conf);
}}}
```

OUTPUT:



Namenode information — Mozilla Firefox

localhost:50070/dfshealth.html#tab-overview

Hadoop | Overview | Datanodes | Datanode Volume Failures | Snapshot | Startup Progress | Utilities

Overview 'localhost:9000' (active)

Started:	Fri Oct 15 06:08:41 EDT 2021
Version:	2.7.3, rbaa91f7c6bc9cb92be5982de4719c1c8af91ccff
Compiled:	2016-08-18T01:41Z by root from branch-2.7.3
Cluster ID:	CID-68b3bd2a-53b7-4ebb-b31e-69b6886fadb3
Block Pool ID:	BP-261072913-127.0.1.1-1634292499991

Summary

Security is off.

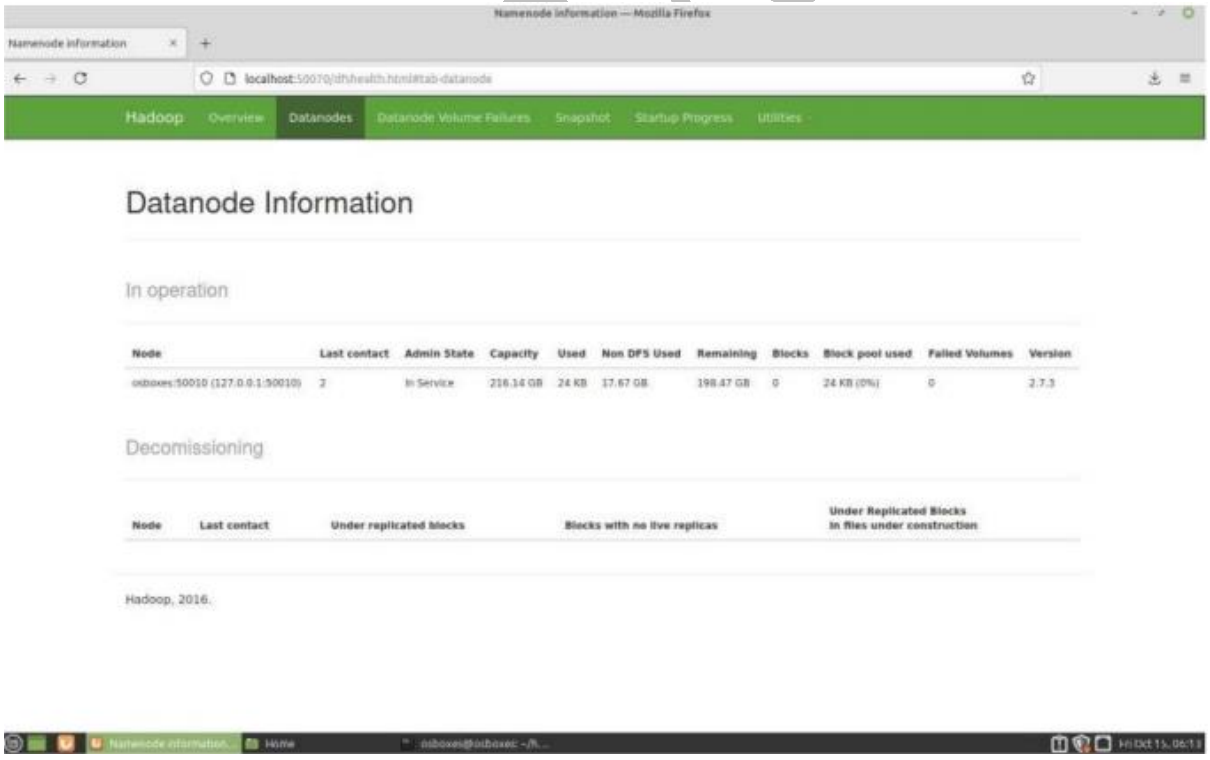
Safemode is off.

7 files and directories, 0 blocks = 7 total filesystem object(s).

Heap Memory used 39.27 MB of 46.5 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 41.14 MB of 42.06 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	216.14 GB
DFS Used:	24 KB (0%)
Non DFS Used:	17.67 GB



Namenode information — Mozilla Firefox

localhost:50070/dfshealth.html#tab-datanode

Hadoop | Overview | Datanodes | Datanode Volume Failures | Snapshot | Startup Progress | Utilities

Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
osboxes:50010 (127.0.0.1:50010)	2	In Service	216.14 GB	24 KB	17.67 GB	198.47 GB	0	24 KB (0%)	0	2.7.3

Decomissioning

Node	Last contact	Under replicated blocks	Blocks with no live replicas	Under Replicated Blocks In files under construction

Hadoop, 2016.

Procedure:

OpenNebula offers a simple but feature-rich and flexible solution to build and manage enterprise clouds and virtualized data centers. OpenNebula is designed to be simple. Simple to install, update and operate by the admins, and simple to use by end users.
Interfaces Provided by OpenNebula
OpenNebula provides many different interfaces that can be used to interact with the functionality offered to manage physical and virtual resources. There are four main different perspectives to interact with OpenNebula:

- Cloud interfaces for Cloud Consumers, like EC2 Query and EBS interfaces, and a simple Sunstone cloud user view that can be used as a self-service portal.
- Administration interfaces for Cloud Advanced Users and Operators, like a Unix- like command line interface and the powerful Sunstone GUI.
- Extensible low-level APIs for Cloud Integrators in Ruby, JAVA and XMLRPC API
- A Marketplace for Appliance Builders with a catalog of virtual appliances ready to run in OpenNebula environments.

OpenNebula Support to Cloud Consumers
OpenNebula provides a powerful, scalable and secure multi-tenant cloud platform for fast delivery and elasticity of virtual resources. Multi-tier applications can be deployed and consumed as pre-configured virtual appliances from catalogs.

- Image Catalogs: OpenNebula allows to store disk images in catalogs (termed datastores), that can be then used to define VMs or shared with other users. The images can be OS installations, persistent data sets or empty data blocks that are created within the datastore.
- Network Catalogs: Virtual networks can be also be organised in network catalogs, and provide means to interconnect virtual machines. This kind of resources can be defined as

IPv4, IPv6, or mixed networks, and can be used to achieve full isolation between virtual networks.

- VM Template Catalog: The template catalog system allows to register virtual machine definitions in the system, to be instantiated later as virtual machine instances.
- **Virtual Resource Control and Monitoring:** Once a template is instantiated to a virtual machine, there are a number of operations that can be performed to control lifecycle of the virtual machine instances, such as migration (live

and cold), stop, resume, cancel, poweroff, etc.

- **Multi-tier Cloud Application Control and Monitoring:** OpenNebula allows to define, execute and manage multi-tiered elastic applications, or services composed of interconnected Virtual Machines with deployment dependencies between them and auto- scaling rules

OpenNebula Support to Cloud Consumers

Users and Groups: OpenNebula features advanced multi-tenancy with powerful users and groups management, fine-grained ACLs for resource allocation, and resource quota management to track and limit computing, storage and networking utilization.

- **Virtualization**: Various hypervisors are supported in the virtualization manager, with the ability to control the complete lifecycle of Virtual Machines and multiple hypervisors in the same cloud infrastructure.
- **Hosts**: The host manager provides complete functionality for the management of the physical hosts in the cloud.
- **Monitoring**: Virtual resources as well as hosts are periodically monitored for key performance indicators. The information can then used by a powerful and flexible scheduler for the definition of workload and resource-aware allocation policies.
- **Accounting**: A Configurable accounting system to visualize and report resource usage data, to allow their integration with chargeback and billing platforms, or to guarantee fair share of resources among users.

Networking: An easily adaptable and customizable network subsystem is present in OpenNebula in order to better integrate with the specific network requirements of existing data centers and to allow full isolation between virtual machines that composes a virtualised service**.**

- **Storage**: The support for multiple datastores in the storage subsystem provides extreme flexibility in planning the storage backend and important performance benefits.
- **Security**: This feature is spread across several subsystems: authentication and authorization mechanisms allowing for various possible mechanisms to identify a authorize users, a powerful Access Control List mechanism allowing different role management with fine grain permission granting over any resource managed by OpenNebula, support for isolation at different levels...
- **High Availability**: Support for HA architectures and configurable behavior in the event of host or VM failure to provide easy to use and cost-effective failover

solutions.

- **Clusters**: Clusters are pools of hosts that share datastores and virtual networks. Clusters are used for load balancing, high availability, and high performance computing.
- **Multiple Zones**: The Data Center Federation functionality allows for the centralized management of multiple instances of OpenNebula for scalability, isolation and multiple- site support.
- **VDCs**. An OpenNebula instance (or Zone) can be further compartmentalized in Virtual Data Centers (VDCs), which offer a fully-isolated virtual infrastructure environment where a group of users, under the control of the group administrator, can create and manage compute, storage and networking capacity.
- **Cloud Bursting**: OpenNebula gives support to build a hybrid cloud, an extension of a private cloud to combine local resources with resources from remote cloud providers. A whole public cloud provider can be encapsulated as a local resource to be able to use extra computational capacity to satisfy peak demands.
- **App Market**: OpenNebula allows the deployment of a private centralized catalog of cloud applications to share and distribute virtual appliances across OpenNebula instances

OpenNebula Support to Cloud builders

OpenNebula offers broad support for commodity and enterprise-grade hypervisor, monitoring, storage, networking and user management services:

User Management: OpenNebula can validate users using its own internal user database based on passwords, or external mechanisms, like ssh, x509, ldap or Active Directory

Virtualization: Several hypervisor technologies are fully supported, like Xen, KVM and VMware.

Monitoring: OpenNebula provides its own customizable and highly scalable monitoring system and also can be integrated with external data center monitoring tools.

Networking:
Virtual networks can be backed up by 802.1QVLANs, ebtables, Open vSwitch or VMware networking.

Storage: Multiple backends are supported like the regular (shared or not) filesystem

datastore supporting popular distributed file systems like NFS, Lustre, GlusterFS, ZFS, GPFS, MooseFS...; the VMware datastore (both regular filesystem or VMFS based) specialized for the VMware hypervisor that handle the vmdk format; the LVM datastore to store disk images in a block device form; and Ceph for distributed block device.

Databases: Aside from the original sqlite backend, mysql is also supported.

OpenNebula 4.6 Installation on Ubuntu 14.04 and Kernel Virtual Machine (KVM) Hypervisor

OpenNebula installation using Ubuntu 14.04 as Operating system and KVM as the hypervisor. The two components are present in it are Frontend and Nodes. The Frontend server execute the OpenNebula services and the Nodes functionality is to execute virtual machines. The following steps to execute both components in a single server. The machine has to support virtualization (i.e.) VT enabled system. The command to test the virtualization support is as follows

```
grep -E 'svm|vmx'  /proc/cpuinfo
```

Package Layout

opennebula-common: Provides the user and common files

libopennebula-ruby: All ruby libraries

opennebula-node: Prepares a node as an opennebula-node **opennebula-sunstone:** OpenNebula Sunstone Web Interface **opennebula-tools:** Command Line interface

opennebula-gate: Gate server that enables communication between VMs and OpenNebula

opennebula-flow: Manages services and elasticity

opennebula: OpenNebula Daemon

Step 1. Installation in the Frontend

Step 1. Installation in the Frontend

1 Add the repository

```
sudo wget -q -O- http://downloads.opennebula.org/repo/Ubuntu/repo.key | apt-key add -
sudo    echo    "deb    http://downloads.opennebula.org/repo/Ubuntu/14.04   stable    opennebula"
> /etc/apt/sources.list.d/opennebula.list
```

2.Install the required packages

```
sudo apt-get update

sudo apt-get install opennebula opennebula-sunstone nfs-kernel-server
```

3 Configure and start the services

There are two main processes that must be started, the main OpenNebula daemon: **oned**, and the graphical user interface: **sunstone**.

Sunstone listens only in the loopback interface by default for security reasons. To change it edit /etc/one/sunstone-server.conf and

change :host: 127.0.0.1 to :host: 0.0.0.0. The command to restart the Sunstone:

```
sudo /etc/init.d/opennebula-sunstone restart
```

4. Configure Network File Service (NFS) (This is not needed if both Frontend and Nodes are in the same machine)

Export /var/lib/one/ from the frontend to the worker nodes. To do so add the following to the /etc/exports file in the frontend:

```
/var/lib/one/ *(rw,sync,no_subtree_check,root_squash)
```

5 Refresh the NFS export by the following command

```
sudo service nfs-kernel-server restart
```

6 Configure SSH public key

```
sudo su – oneadmin

cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
```

7 Add the following snippet to ~/.ssh/c

```
$ cat << EOT > ~/.ssh/config

Host *

    StrictHostKeyChecking no

    UserKnownHostsFile /dev/null
```

8 Restart the network

```
sudo /etc/init.d/networking restart
```

9 Configure NFS (This is not needed if both Frontend and Nodes are in the same machine) edit the file /etc/fstab as

```
<Frontend IP>:/var/lib/one/ /var/lib/one/ nfs soft,intr,rsize=8192,wsize=8192,noauto
```

10 Mount the NFS

```
sudo mount /var/lib/one
```

/* if there is any problem , then it will be firewall issue

1. Configure Qemu

oneadmin user must be able to manage libvirt as root

```
# cat << EOT > /etc/libvirt/qemu.conf
user = "oneadmin"

group = "oneadmin"
dynamic_ownership = 0
EOT
```

Restart libvirt

```
sudo service libvirt-bin restart
```

Step 3: Start the sunstone from the web browser

```
http://frontend:9869
```

oneadmin password is available in the file ~/.one/one_auth

1. From the command line, to login as oneadmin in the Frontend , execute

```
sudo su - oneadmin
```

2. Adding Host

To start running VMs, you should first register a worker node for OpenNebula.

Create resources in opennebula

```
$ onevnet create mynetwork.one
$ oneimage create --name "CentOS-6.5_x86_64" \
    --path "http://appliances.c12g.com/CentOS-6.5/centos6.5.qcow2.gz" \
    --driver qcow2 \
    --datastore default
$ onetemplate create --name "CentOS-6.5" --cpu 1 --vcpu 1 --memory 512 \
    --arch x86_64 --disk "CentOS-6.5_x86_64" --nic "private" --vnc \
```

Monitor the resources are created by running the command **oneimage list**

In order to dynamically add ssh keys to Virtual Machines we must add our ssh key to the user template, by editing the user template:

```
$ EDITOR = vi oneuser update oneadmin
```

Add a new line to the template (cat ~/.ssh/id_dsa.pub)

```
SSH_PUBLIC_KEY="ssh-dss AAAAB3NzaC1kc3MAAACBANBWTQmm4Gt..."
```

3. To run a virtual machine

```
$ onetemplate instantiate "Centos-6.5" --name "My Scratch VM"
```

To test, execute onevm list , VM going from Pending to Prolog to Running. If it fails, check the log as
/var/log/one/<VM_ID>/vm.log