Vidyavardhini's
College of Engineering & Technology
Vasai Road (W)

Department of Computer Engineering

Laboratory Manual
(Student Copy)

Semester	VI	**Class**	TE
Course Code	CSL605		
Course Name	SBL-Cloud Computing Laboratory		

Vidyavardhini's College of Engineering & Technology

Vision

To be a premier institution of technical education; always aiming at becoming a valuable resource for industry and society.

Mission

- To provide technologically inspiring environment for learning.
- To promote creativity, innovation and professional activities.
- To inculcate ethical and moral values.
- To cater personal, professional and societal needs through quality education.

Department Vision:

To evolve as a center of excellence in the field of Computer Engineering to cater to industrial and societal needs.

Department Mission:

- To provide quality technical education with the aid of modern resources.
- Inculcate creative thinking through innovative ideas and project development.
- To encourage life-long learning, leadership skills, entrepreneurship skills with ethical & moral values.

Program Education Objectives (PEOs):

PEO1: To facilitate learners with a sound foundation in the mathematical, scientific and engineering fundamentals to accomplish professional excellence and succeed in higher studies in Computer Engineering domain
PEO2: To enable learners to use modern tools effectively to solve real-life problems in the field of Computer Engineering.
PEO3: To equip learners with extensive education necessary to understand the impact of computer technology in a global and social context.
PEO4: To inculcate professional and ethical attitude, leadership qualities, commitment to societal responsibilities and prepare the learners for life-long learning to build up a successful career in Computer Engineering.

Program Specific Outcomes (PSOs):

PSO1: Analyze problems and design applications of database, networking, security, web technology, cloud computing, machine learning using mathematical skills, and computational tools.

PSO2: Develop computer-based systems to provide solutions for organizational, societal problems by working in multidisciplinary teams and pursue a career in the IT industry.

Program Outcomes (POs):

Engineering Graduates will be able to:

- **PO1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- **PO2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- **PO3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- **PO4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- **PO5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- **PO6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- **PO7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- **PO8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- **PO9. Individual and teamwork:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- **PO10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- **PO11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- **PO12. Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Course Objectives

Sr. No.	The course will help the learners

1	To make students familiar with key concepts of virtualization.
2	To make students familiar with various deployment models of cloud such as private, public ,hybrid and community so that they start using and adopting appropriate types of cloud for their application.
3	To make students familiar with various service models such as IaaS, SaaS, PaaS, Security as a Service (SECaaS) 12.and Database as a Service
4	To make students familiar with security and privacy issues in cloud computing and how to address them.

Course Outcomes

CO	At the end of course students will be able to:	**Action verbs**	**Bloom's Level**
CSL605.1	Create virtual machines using open source technology.	Create	Create (Level 6)
CSL605.2	Compare cloud computing services SaaS/PaaS/IaaS for a given application	Compare	Analyze (Level 4)
CSL605.3	Design and develop real world web applications and deploy them on commercial clouds.	Design Develop	Create (Level 6)
CSL605.4	Deploy cloud services to address security issues .	Deploy	Apply (Level 3)
CSL605.5	Identify commercially available cloud services and recommend the appropriate one for the given application.	Identify Recommend	Evaluate (Level 5)
CSL605.6	Implement the concept of containerization.	Implement	Apply (Level 3)

Mapping of Experiments with Course Outcomes

Course Modules	Course Outcomes

	CSL605.1	CSL605.2	CSL605.3	CSL605.4	CSL605.5	CSL605.6
To study introduction and overview of cloud computing	3	—	—	—	—	—
To study and implement Hosted Virtualization using VirtualBox& KVM.	3	—	—	—	—	—
To study and Implement Bare-metal Virtualization using Xen, HyperV or VMware Esxi.	3	—	—	—	—	—
To study and Implement Infrastructure as a Service using AWS/Microsoft Azure.	—	3	—	—	—	—
To study and Implement Platform as a Service using AWS Elastic Beanstalk/ Microsoft Azure App Service.	—	3	—	—	—	—
To study and Implement Storage as a Service using OwnCloud/ AWS S3, Glaciers/ Azure Storage.	—	3	—	—	—	—
To study and Implement Security as a Service on AWS/Azure	—	—	—	3	—	—
To study and implement virtual own cloud.	—	3	—	—	—	—

To deploy different cloud web services on created virtual own cloud	—	3	—	—	—	—
To study and Implement Containerization using Docker	—	—	—	—	—	3
Mini-project: Design a Web Application hosted on public cloud platform.[It should cover the concept of IaaS, PaaS, DBaaS, Storage as a Service, Security as a Service etc.	—	—	3	—	3	—
Assignment based on selection of suitable cloud platform solution based on requirement analysis considering given problem statement	—	—	—	—	3	—
Assignment on recent trends in cloud computing and related technologies	—	—	—	—	3	—

Enter correlation level 1, 2 or 3 as defined below
1: Slight (Low) 2: Moderate (Medium) 3: Substatial (High)
If there is no correlation put "—".

INDEX

Sr.No.	Name of Experiment	D.O.P.	D.O.C	Page No.	Remark
1	To study introduction and overview of cloud computing				
2	To study and implement Hosted Virtualization using VirtualBox& KVM.				

3	To study and Implement Bare-metal Virtualization using Xen, HyperV or VMware Esxi.				
4	To study and Implement Infrastructure as a Service using AWS/Microsoft Azure.				
5	To study and Implement Platform as a Service using AWS Elastic Beanstalk/ Microsoft Azure App Service.				
6	To study and Implement Storage as a Service using OwnCloud/ AWS S3, Glaciers/ Azure Storage.				
7	To study and Implement Security as a Service on AWS/Azure				
8	To study and implement virtual own cloud.				
9	To deploy different cloud web services on created virtual own cloud				
10	To study and Implement Containerization using Docker				
11	Mini-project: Design a Web Application hosted on public cloud platform.[It should cover the concept of IaaS, PaaS, DBaaS, Storage as a Service, Security as a Service etc.]				
Assignment No. 1	Assignment based on selection of suitable cloud platform solution based on requirement analysis considering given problem statement				
Assignment No. 2	Assignment on recent trends in cloud computing and related technologies				

D.O.P: Date of performance

D.O.C : Date of correction

Experiment No.1
Study of NIST model of cloud computing.
Date of Performance:
Date of Submission:

Aim: To study introduction and overview of cloud computing.

Objective: Understand deployment models, service models, advantages of cloud computing..

Theory:

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

This cloud model is composed of five essential characteristics, three service models, and four deployment models.

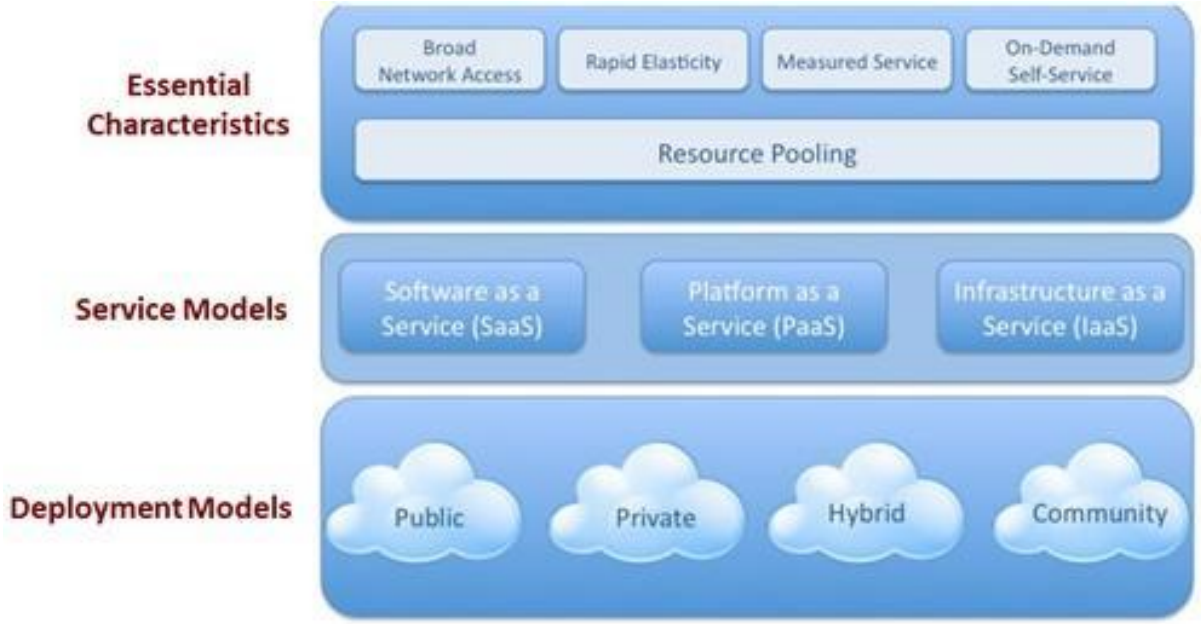


Figure 1.1 Cloud model

Essential Characteristics:

1. On-demand self-service. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.
2. Broad network access. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

3. Resource pooling. The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, and network bandwidth.
4. Rapid elasticity. Capabilities can be rapidly and elastically provisioned, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
5. Measured Service. Cloud systems automatically control and optimize resource use by leveraging a metering capability1 at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Service Models:

1. Cloud Software as a Service (SaaS).

   The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a Web browser (e.g., Web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.
2. Cloud Platform as a Service (PaaS).

   The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or -acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or

storage, but has control over the deployed applications and possibly application hosting environment configurations.

3. Cloud Infrastructure as a Service (IaaS).

   The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications; and possibly limited control of select networking components (e.g., host, firewalls).

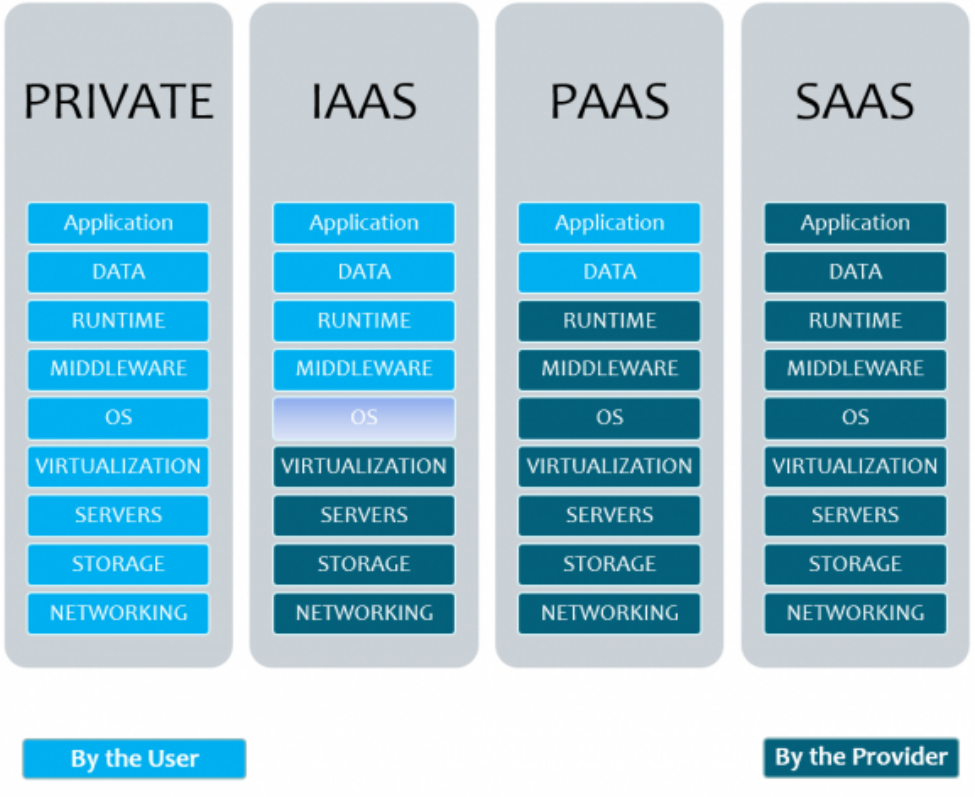


Figure 1.2 Cloud Service Model

Deployment Models:

1. Private cloud. The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.
2. Community cloud. The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

3. Public cloud. The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.
4. Hybrid cloud. The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds)."

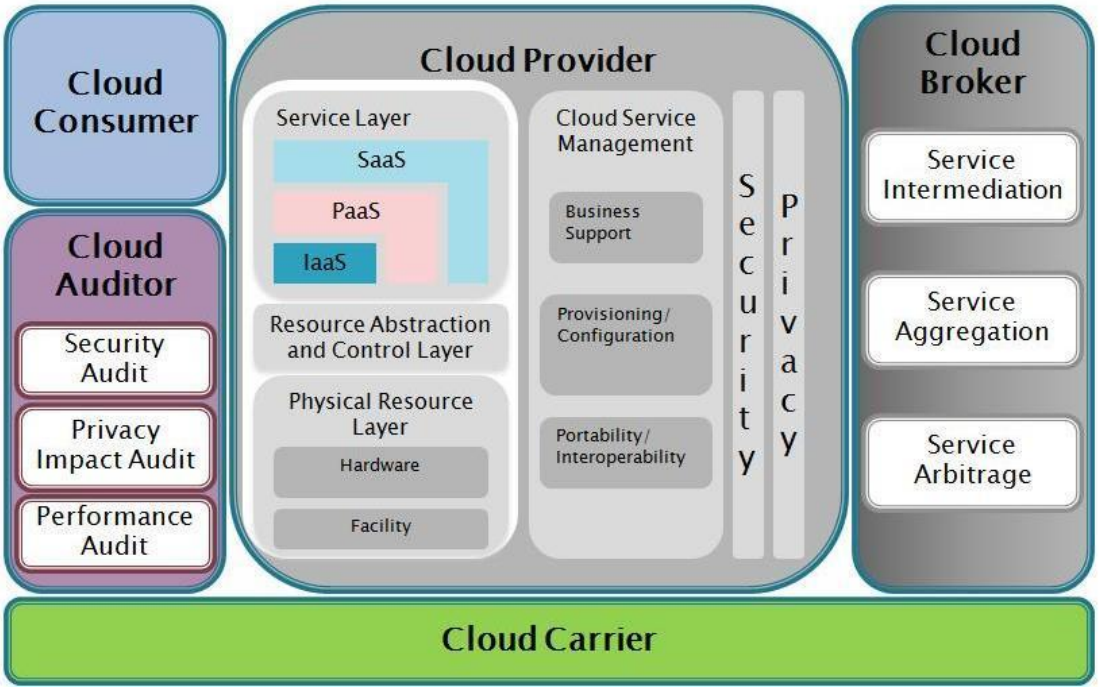


Figure 1.3 NIST Reference Architecture of Cloud Computing

Conclusion:

Comment on NIST Cloud computing architecture, cloud services and types of cloud.

Experiment No. 2
To study and implement Hosted Virtualization using VirtualBox& KVM.
Date of Performance:
Date of Submission:

Aim To study and implement Hosted Virtualization using VirtualBox& KVM.

Objective: Understand different types of virtualizations, Host and bare metal hypervisors and implement horizontal scalability.

Technology: XEN/ Vmwares EXSi

Hosted Virtualization on Oracle Virtual Box Hypervisor

Hosted Virtualization on KVM Hypervisor

Theory:

In computing, virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, operating systems, storage devices, and computer network resources.

Why is virtualization useful?

The techniques and features that VirtualBox provides are useful for several scenarios:

- **Running multiple operating systems simultaneously.** VirtualBox allows you to run more than one operating system at a time. Since you can configure what kinds of "virtual" hardware should be presented to each such operating system, you can install an old operating system such as DOS or OS/2 even if your real computer's hardware is no longer supported by that operating system.

- **Easier software installations.** Software vendors can use virtual machines to ship entire software configurations. For example, installing a complete mail server solution on a real machine can be a tedious task. With VirtualBox, such a complex setup (then often called an "appliance") can be packed into a virtual machine. Installing and running a mail server becomes as easy as importing such an appliance into VirtualBox.
- **Testing and disaster recovery.** Once installed, a virtual machine and its virtual hard disks can be considered a "container" that can be arbitrarily frozen, woken up, copied, backed up, and transported between hosts. On top of that, with the use of another VirtualBox feature called "snapshots", one can save a particular state of a virtual machine and revert back to that state, if necessary. This way, one can freely experiment with a computing environment. If something goes wrong (e.g. after installing misbehaving software or infecting the guest with a virus), one can easily switch back to a previous snapshot and avoid the need of frequent backups and restores. Any number of snapshots can be created, allowing you to travel back and forward in virtual machine time. You can delete snapshots while a VM is running to reclaim disk space.
- **Infrastructure consolidation.** Virtualization can significantly reduce hardware and electricity costs. Most of the time, computers today only use a fraction of their potential power and run with low average system loads. A lot of hardware resources as well as electricity is thereby wasted. So, instead of running many such physical computers that are only partially used, one can pack many virtual machines onto a few powerful hosts and balance the loads between them.

Hypervisor

A hypervisor or virtual machine monitor (VMM) is a piece of computer software, firmware or hardware that creates and runs virtual machines. It allows multiple operating systems to share a single hardware host. Each operating system appears to have the host's processor, memory, and other resources all to itself. However, the hypervisor is actually controlling the host processor and resources, allocating what is needed to each operating system in turn and making sure that the guest operating systems (called virtual machines) cannot disrupt each other.

There are two types of hypervisors: Type 1 and Type 2.

- Type 1 hypervisor (also called a bare metal hypervisor) is installed directly on physical host server hardware just like an operating system. Type 1 hypervisors run on dedicated hardware. They require a management console and are used in data centers. Examples include Oracle OVM for SPARC, ESXi, Hyper-V and KVM.
- Type 2 hypervisors support guest virtual machines by coordinating calls for CPU, memory, disk, network and other resources through the physical host's operating system. This makes it easy for an end user to run a virtual machine on a personal computing device. Examples include VMware Fusion, Oracle Virtual Box, Oracle VM for x86, Solaris Zones, Parallels and VMware Workstation.

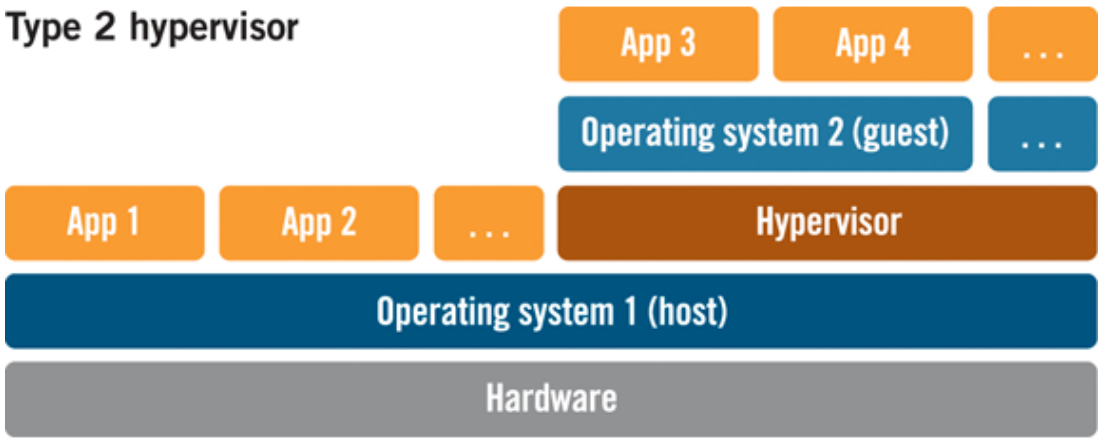


Figure 1. A Type 2 hypervisor runs as an application on a host operating system.

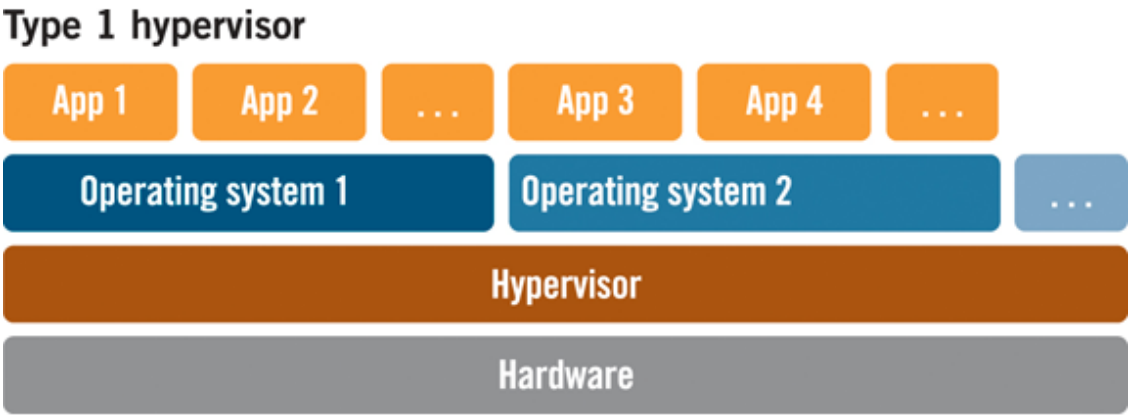


Figure 2. A Type 1 or bare-metal hypervisor sits directly on the host hardware.

Terminology

Host operating system (host OS).

This is the operating system of the physical computer on which VirtualBox was installed. There are versions of VirtualBox for Windows, Mac OS X, Linux and Solaris hosts.

Guest operating system (guest OS).

This is the operating system that is running inside the virtual machine. Theoretically, VirtualBox can run any operating system (DOS, Windows, OS/2, FreeBSD, OpenBSD).

Virtual machine (VM).

This is the special environment that VirtualBox creates for your guest operating system while it is running. In other words, you run your guest operating system "in" a VM. Normally, a VM will be shown as a window on your computer's desktop, but depending on which of the various frontends of VirtualBox you use, it can be displayed in full screen mode or remotely on another computer.

In a more abstract way, internally, VirtualBox thinks of a VM as a set of parameters that determine its behavior. They include hardware settings as well as state information. These settings are mirrored in the VirtualBox Manager window as well as the VBoxManage command line program.

Guest Additions.

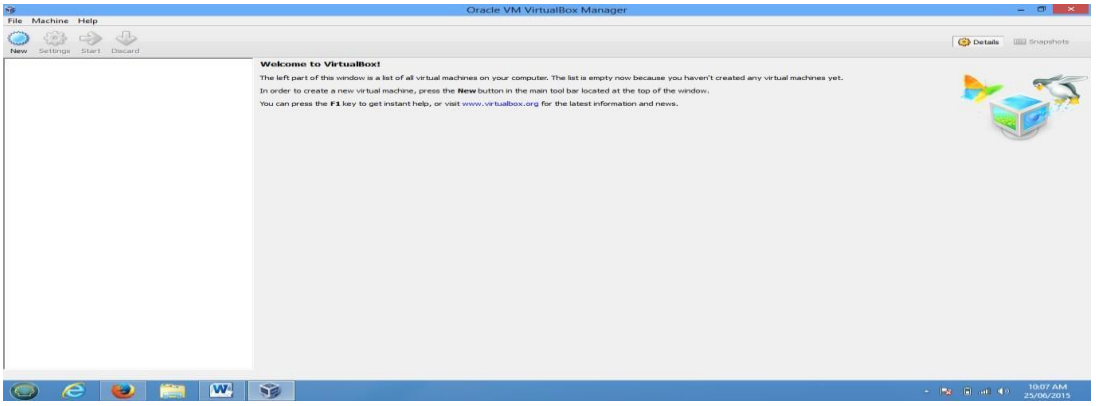
This refers to special software packages which are shipped with VirtualBox but designed to be installed *inside* a VM to improve performance of the guest OS and to add extra features.

Procedure :

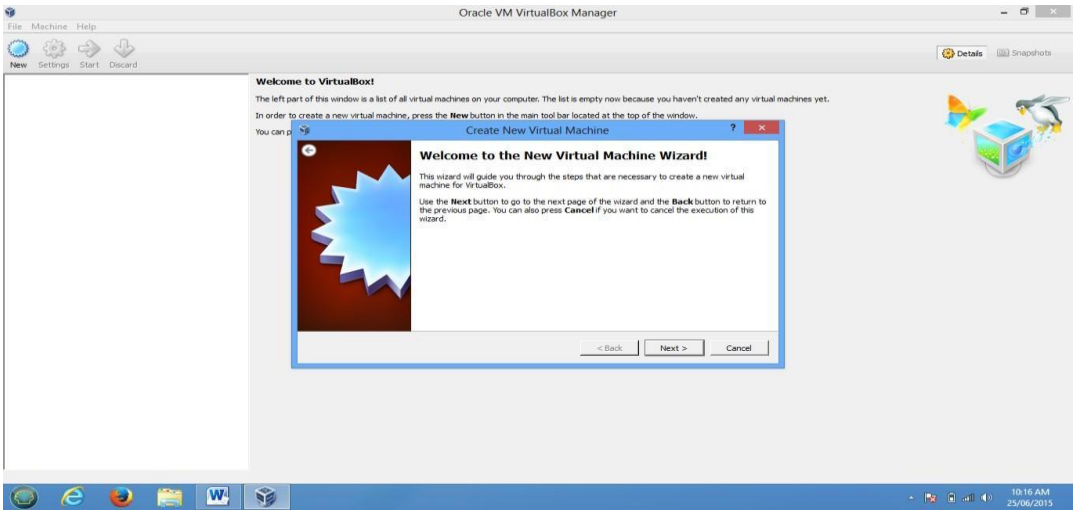
A) Hosted Virtualization on Oracle Virtual Box Hypervisor

Step 1: Download Oracle Virtual box from https://www.virtualbox.org/wiki/Downloads

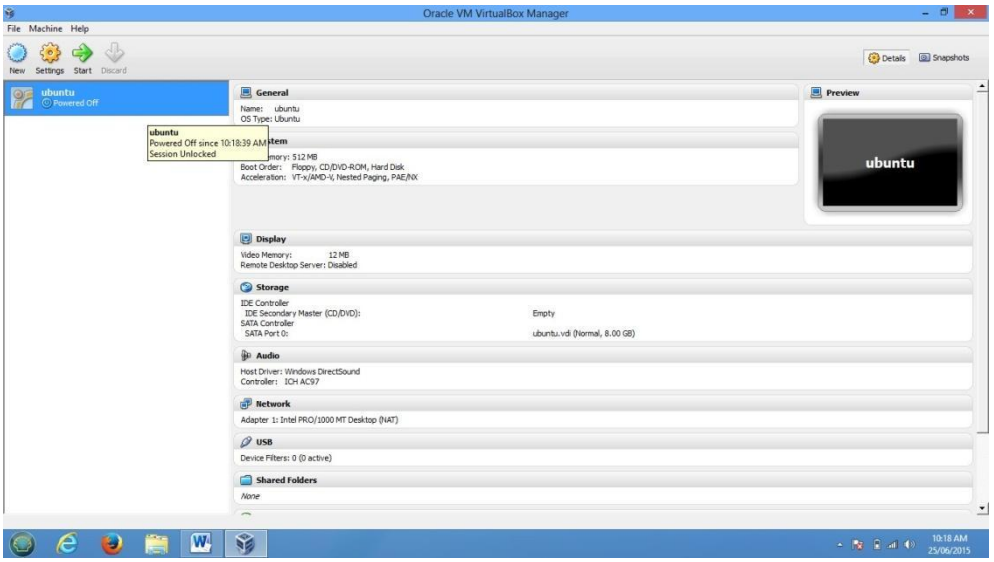
Step 2: Install it in Windows, Once the installation has done open it.



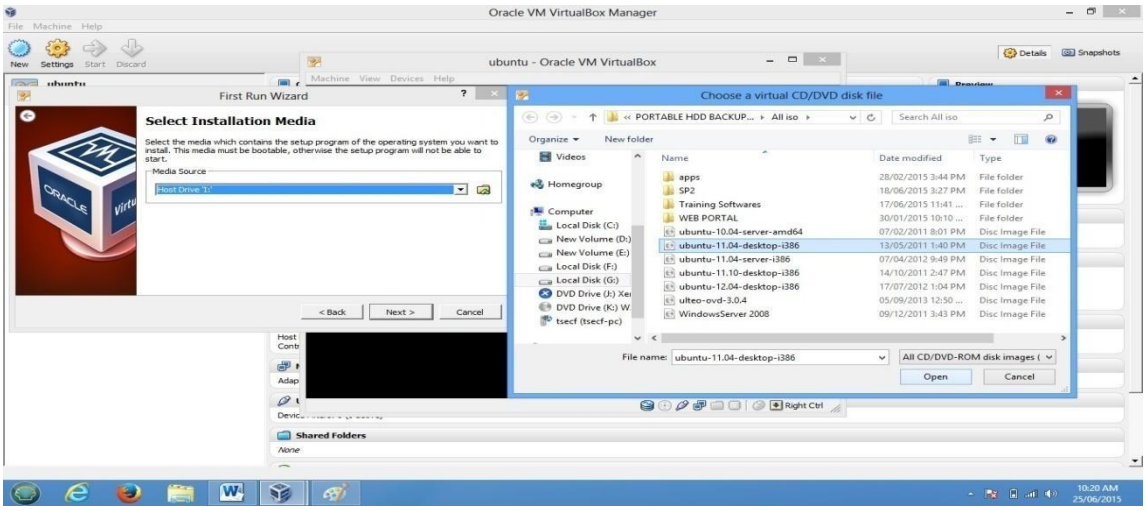
Step 3:-: Create Virtual Machine by clicking on New



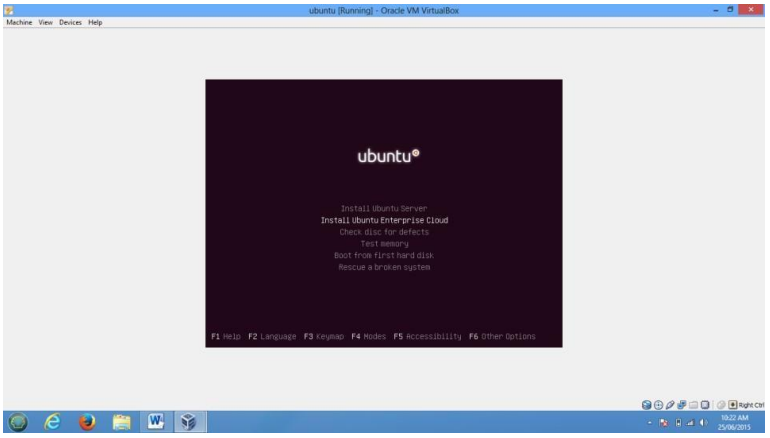
Step 4-: Specify RAM Size, HDD Size, and Network Configuration and Finish the wizard



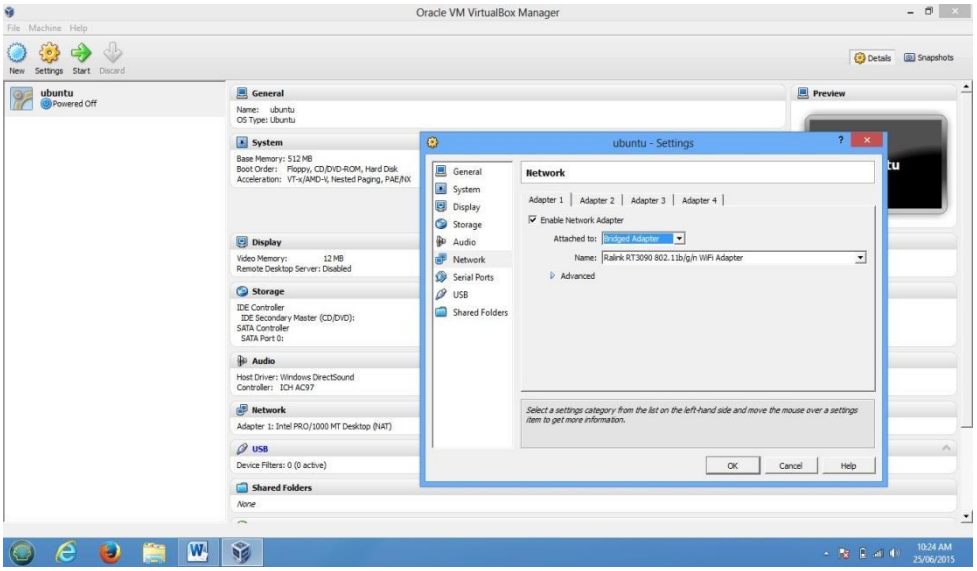
Step 5-: To Select the media for installation Click on start and browse for iso file



Step 6:Complete the Installation and use it.



Step 7: To Connect OS to the network change network Mode to Bridge Adaptor



B) Hosted Virtualization on KVM Hypervisor

The Steps to Create and run Virtual machines in KVM are as follows

1) Check whether CPU has hardware virtualization support.

KVM only works if your CPU has hardware virtualization support – either Intel VT-x or AMD-V. To determine whether your CPU includes these features, run the following command:

#sudo grep -c "svm\|vmx" /proc/cpuinfo

A 0 indicates that your CPU doesn't support hardware virtualization, while a 1 or more indicates that it does.

2) Install KVM and supporting packages.

Virt-Manager is a graphical application for managing your virtual machines.you can use the kvm command directly, but libvirt and Virt-Manager simplify the process.

#sudo apt-get install qemu-kvm libvirt-bin bridge-utils virt-manager

3) Create User.

Only the root user and users in the libvirtd group have permission to use KVM virtual machines. Run the following command to add your user account to the libvirtd group:

#sudo adduser sakec

#sudo adduser sakec libvirtd

After running this command, log out and log back in as sakec

4) Check whether everything is working correctly.

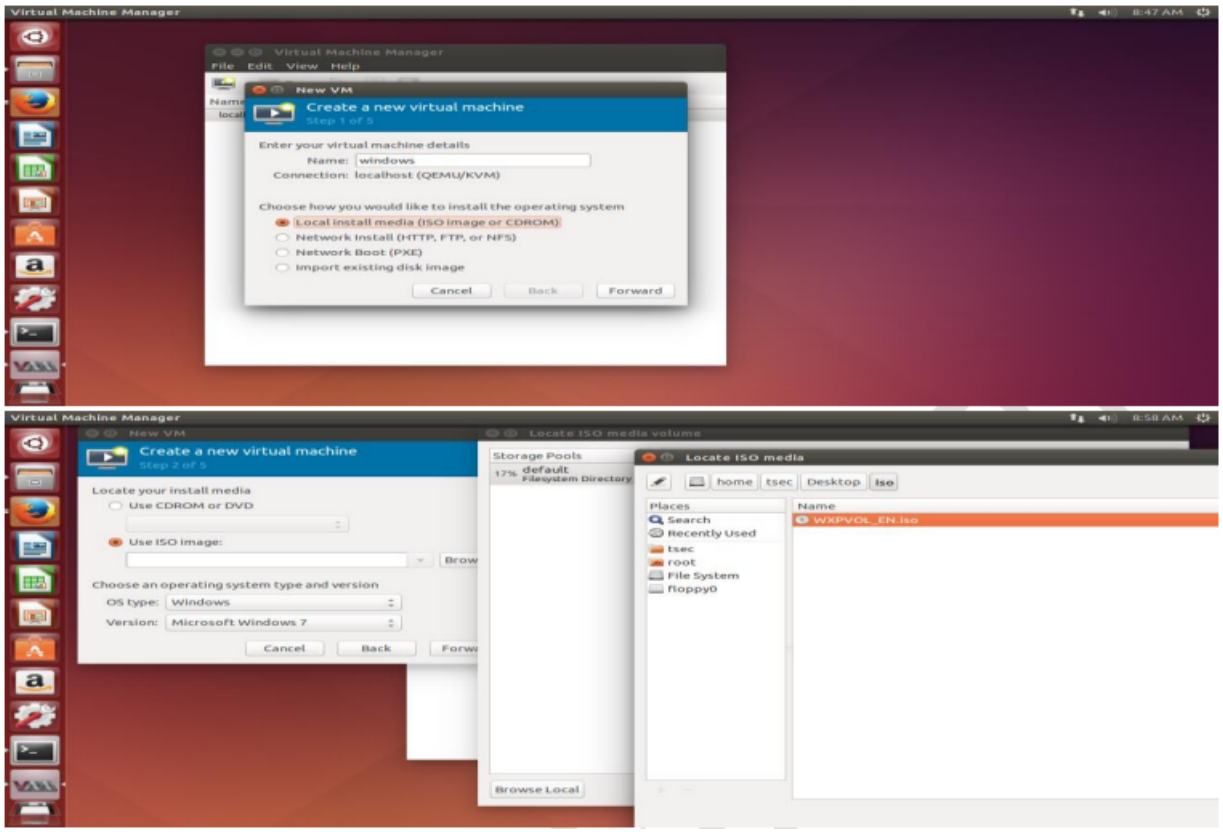
Run following command after logging back in as sakec and you should see an empty list of virtual machines. This indicates that everything is working correctly.

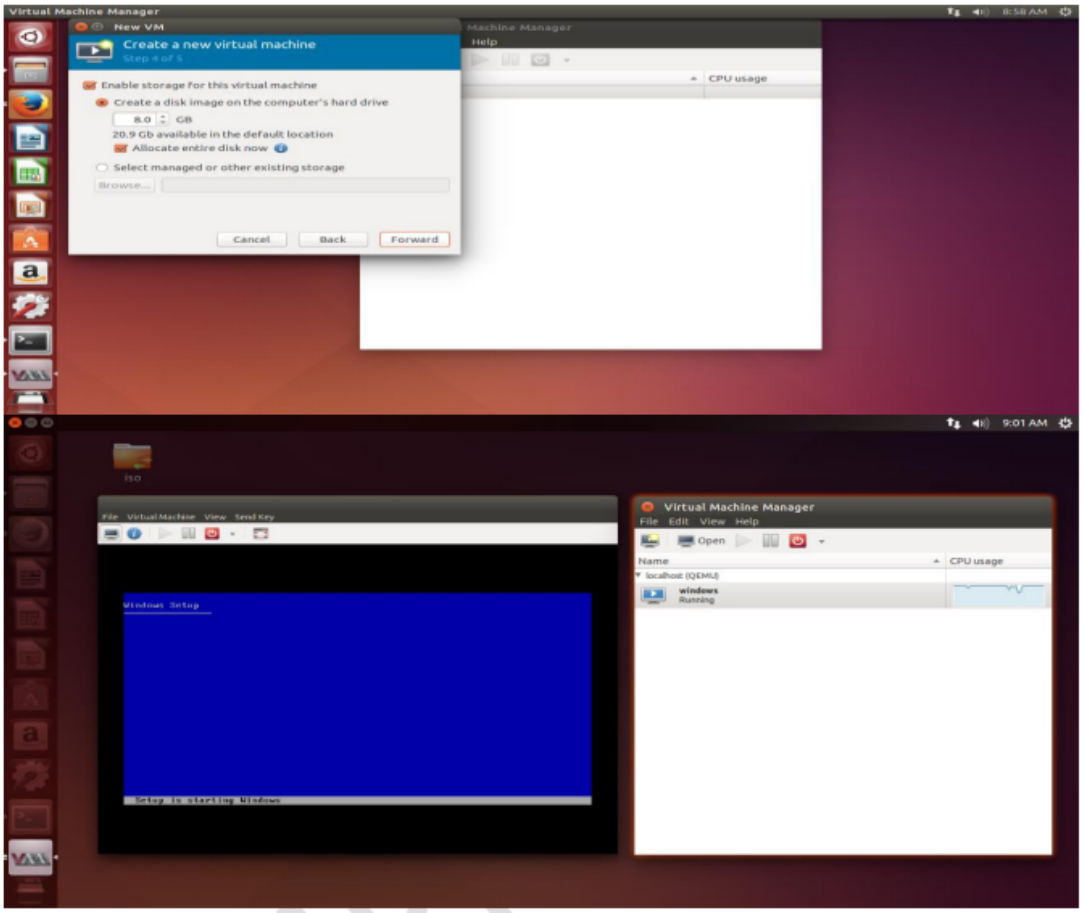
#virsh -c qemu:///system list

5) Open Virtual Machine Manager application and Create Virtual Machine

#virt-manager

6) Create and run Virtual Machines


Virtual Machine Manager
Create a new virtual machine
Enter your virtual machine details
Name: windows
Connection: localhost (QEMU/KVM)
Choose how you would like to install the operating system
Local install media (ISO image or CDROM)
Network Install (HTTP, FTP, or NFS)
Network Boot (PXE)
Import existing disk image
Cancel
Back
Forward
Create a new virtual machine
Locate your install media
Use CDROM or DVD
Use ISO image:
Choose an operating system type and version
OS type: Windows
Version: Microsoft Windows 7
Locate ISO media
Browse Local




```
$ sudo virsh list --all
 Id    Name                           State
----------------------------------------------------
 1     ubuntu-vm                      running
```

```
$ virsh
Welcome to virsh, the virtualization interactive terminal.

Type:  'help' for help with commands
       'quit' to quit

virsh #
virsh #
virsh # list --all
 Id    Name                           State
----------------------------------------------------
 1     Windows                        running
```

Conclusion:

Comment on Virtual Machine you created on the software tool used.

Experiment No. 3
To study and Implement Bare-metal Virtualization using Xen, HyperV or VMware Esxi.
Date of Performance:
Date of Submission:

Aim To study and Implement Bare-metal Virtualization using Xen, HyperV or VMware Esxi.

Objective: Understand different types of virtualizations, Host and bare metal hypervisors and implement horizontal scalability.

Technology: XEN/ Vmwares EXSi

Theory:

Creating and running virtual machines on Bare-Metal Hypervisor Xen Server.

Step 1: Install Xen Server

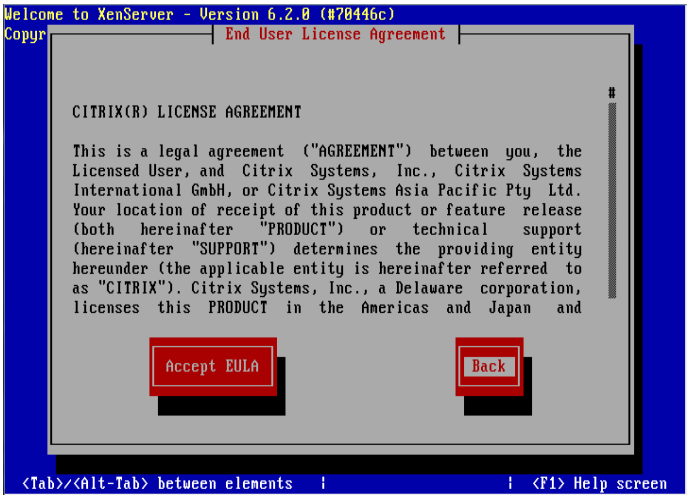
Step i-: Insert Bootable Xen Server CD into CDROM and Make first boot device as a CDROM from BIOS

Step ii-: press F2 to see the advanced options, otherwise press Enter to start installation

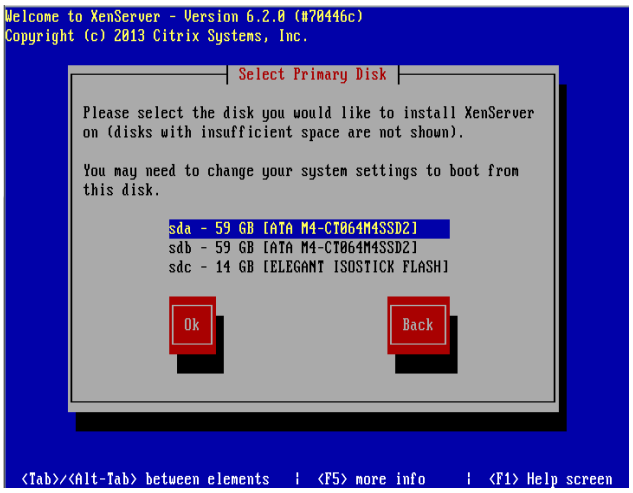
Step iii -: Select Keyboard Layout

Step iv -:Press Enter to load Device Drivers

Step v -:Press Enter to Accept End user license Agreement



Step vi -:Select Appropriate disk on which you want to install Xen server



Step vii -:Select Appropriate installation Media (LOCAL Media)

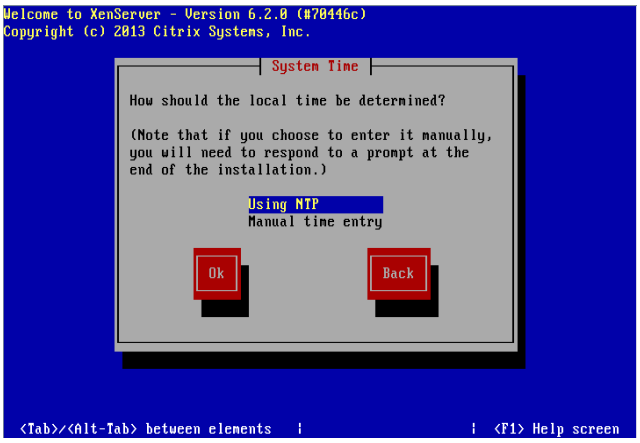
Step viii -:Select Additional Packages for installation

Step ix-: Specify Root password

Step x -: Specify IP Address to a Xen Server

Step xi-:Select Time Zone

Step xii-:Specify NTP Servers address or use manual time entry then start installation.

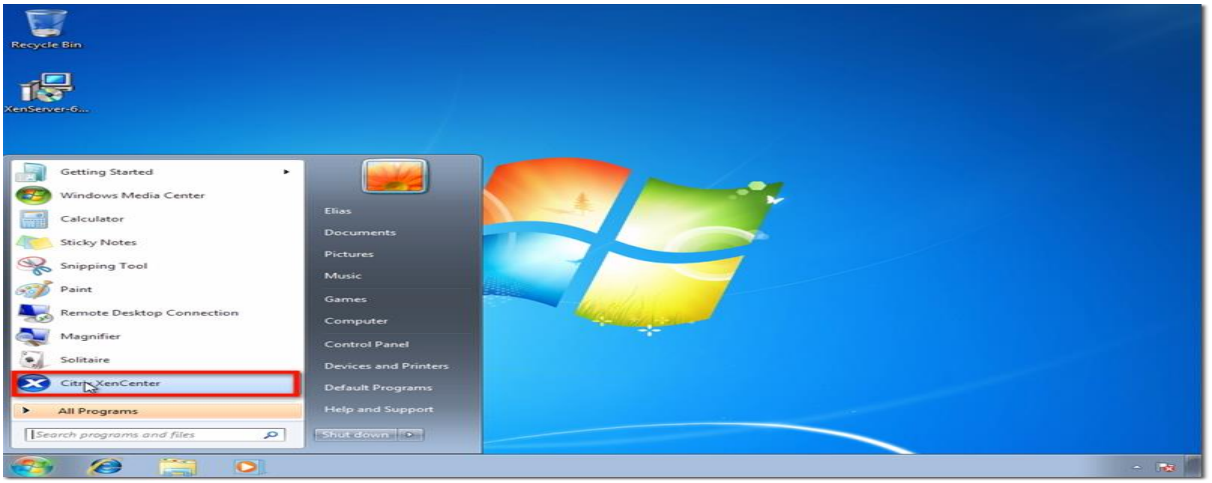


Once installation is done you will see the final screen shown below.

Xen Server Final Screenshot

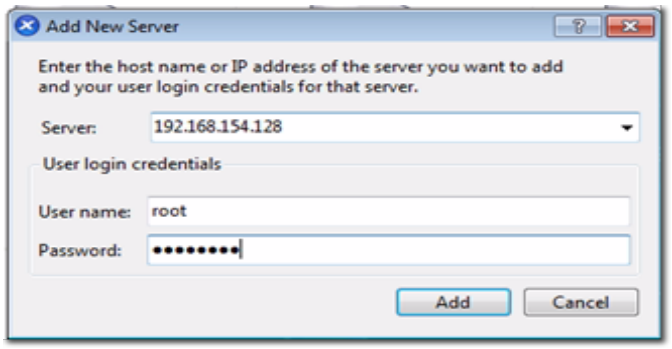
Step 2:Connect Xen Server to Xen Center

Firstly, download the xen center a management utily from xen server by opening the xen severs IP address as a URL on browser. Once Xen center is downloaded, install it.Open Xen center from start menu of Windows.

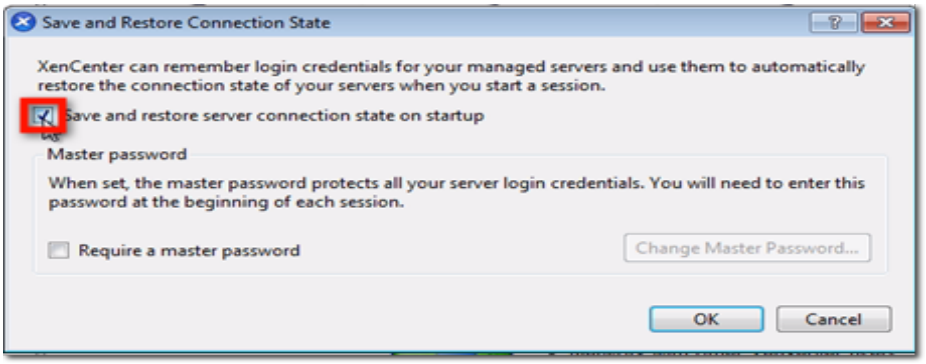


Here's how XenCenter looks like (see screenshot below) before any hosts, resource pools, and so on, are added to it. To connect to the XenServer host you configured earlier, click Add a server.

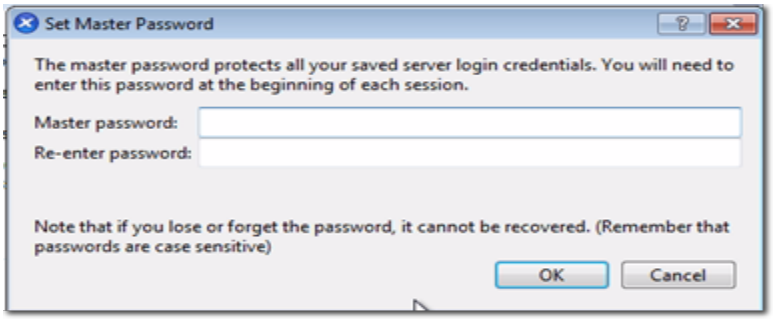
Enter the IP address I asked you to take note of earlier. Also enter the password you assigned for your root account. Click Add.



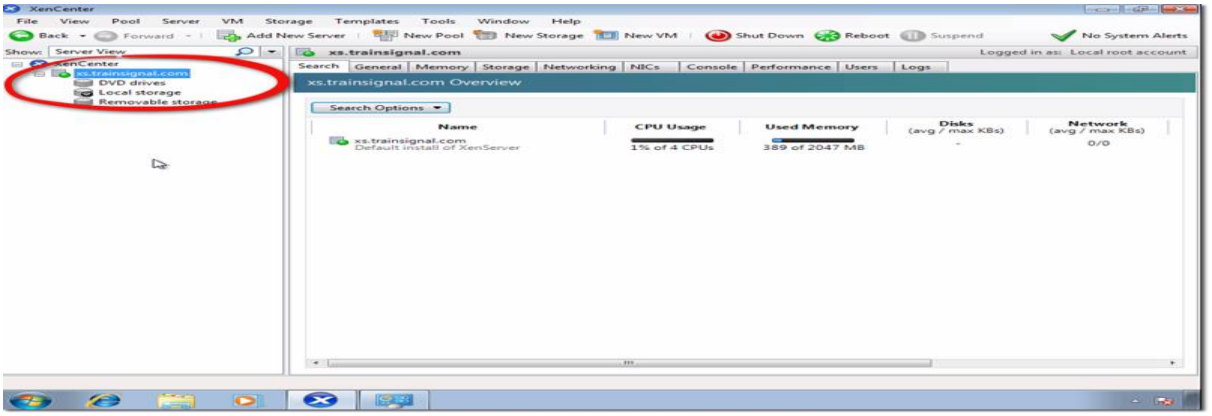
One of the first things you want to make sure as you're adding a new XenServer to XenCenter is to save and restore the server connection state on startup. Check the box that will do just that.



Once you do that, you will be allowed to configure a master password for all the XenServers you'll be associating with this XenCenter. Click the Require a master password checkbox if that's what you want to do, and then enter your desired master password in the fields provided.



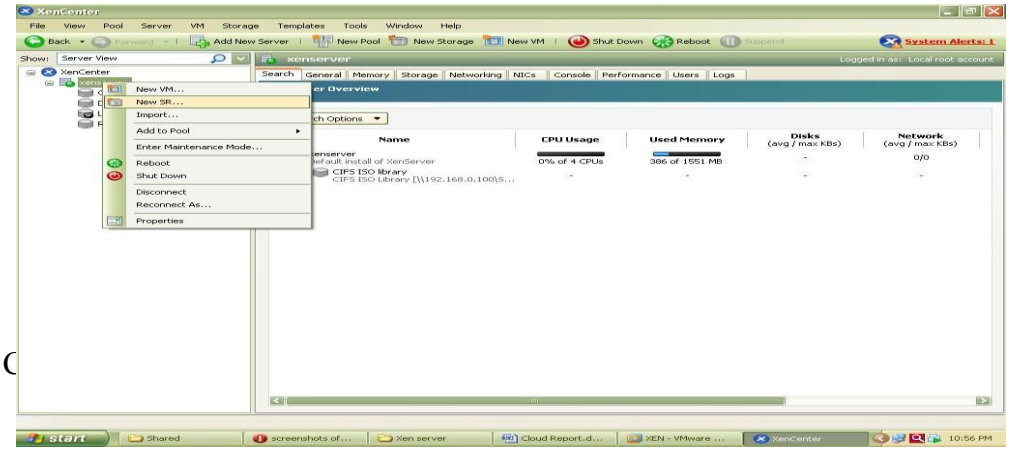
After you click OK, you'll be brought back to the main screen, where you'll see your XenServer already added to XenCenter.



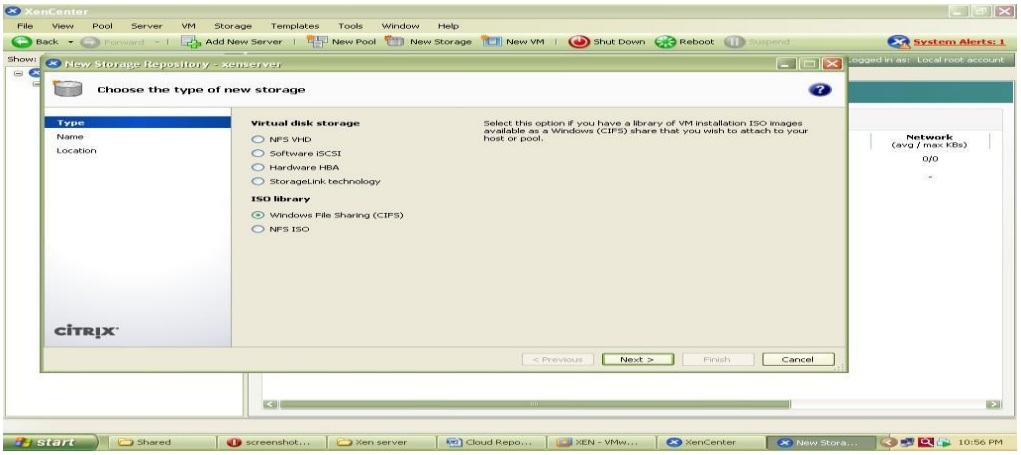
Step-:3 Create Storage Repository and Installing VM

Now Before Creating VM we have to Create Storage Repository first which is nothing but shared directory on Xen Center which holds all iso files and which is required to install Operating system on Xen Server its steps are as follows.Right click on Xenserver icon on xen center and click on New SR

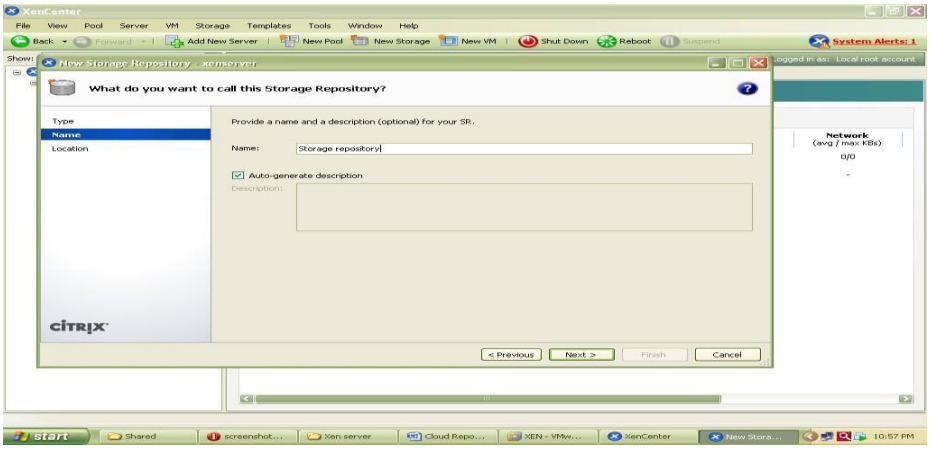
C



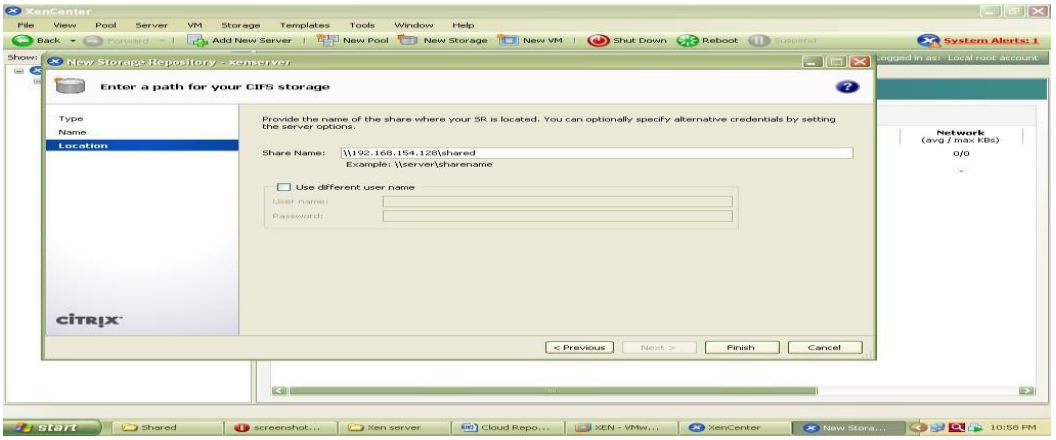
Now Select Windows CIFS library



Specify Storage Repository Name



Now specify path of shared folder at client side which holds all iso files of os or VM which we are going to install on Xen Server.

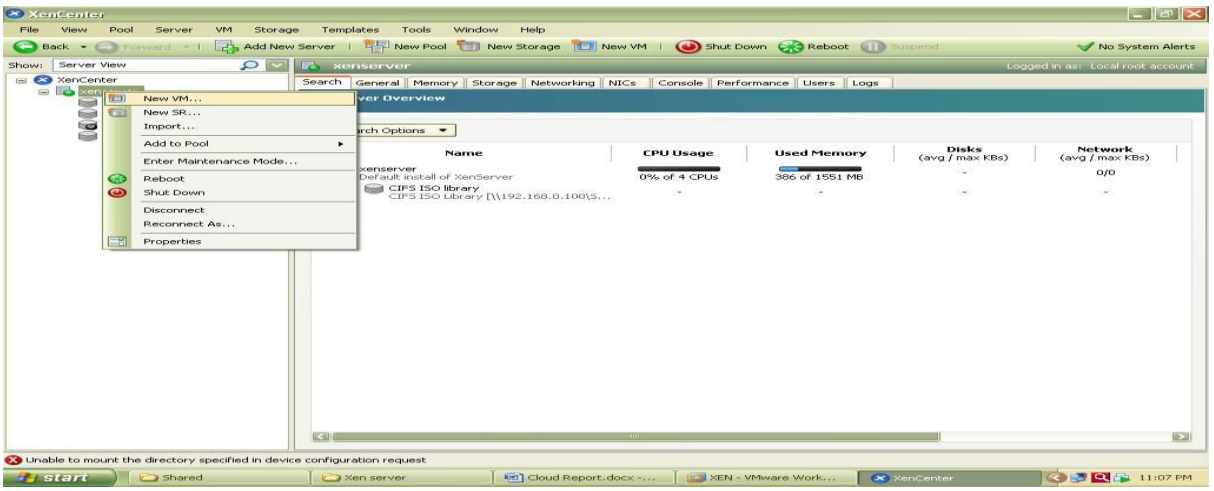


At the end Click on finish to create SR.

To check all iso files click on CIFS library and select storage this will show you all iso files.

Installation of UBUNTU Server on Xen Server

Step 1 -: Right click on Xenserver icon on xen center and select New VM

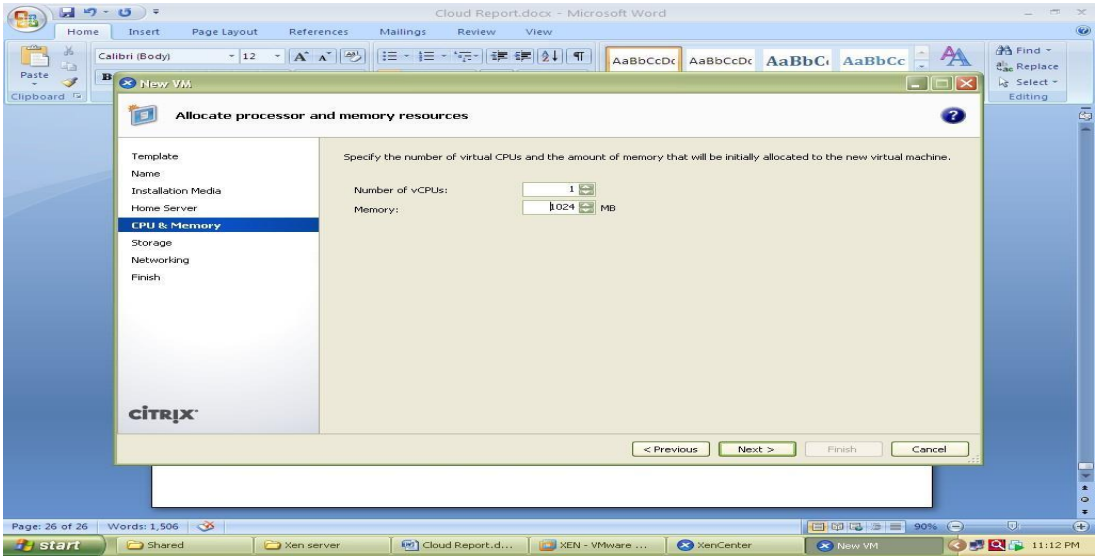


Now select an Operating System to be install here select Ubuntu Lucid Lynx and click on next

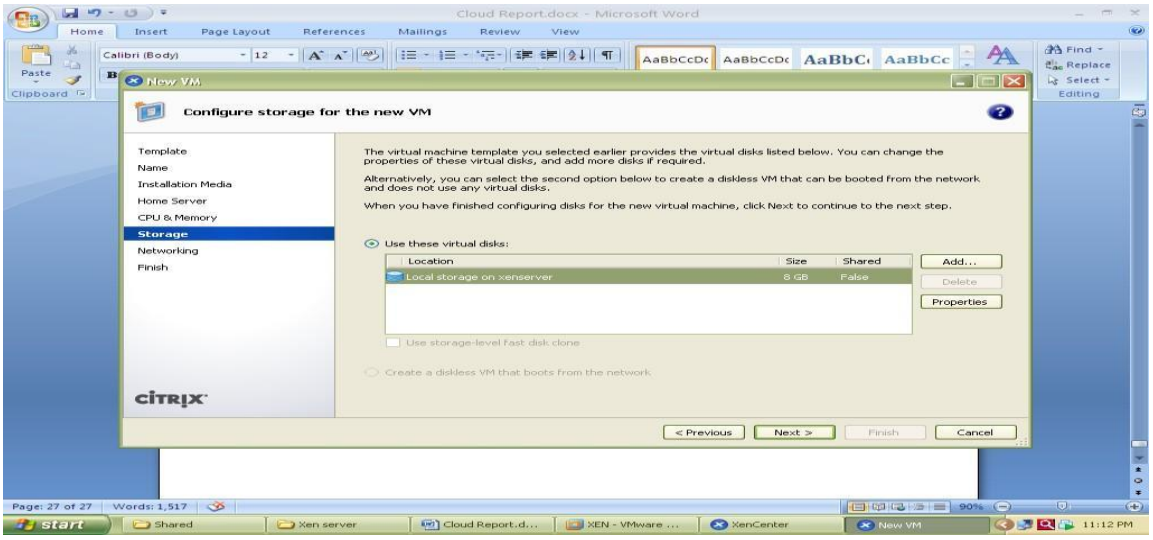
Now specify Instance Name as ubuntu server

Select iso file of Ubuntu server 10.10 to be install

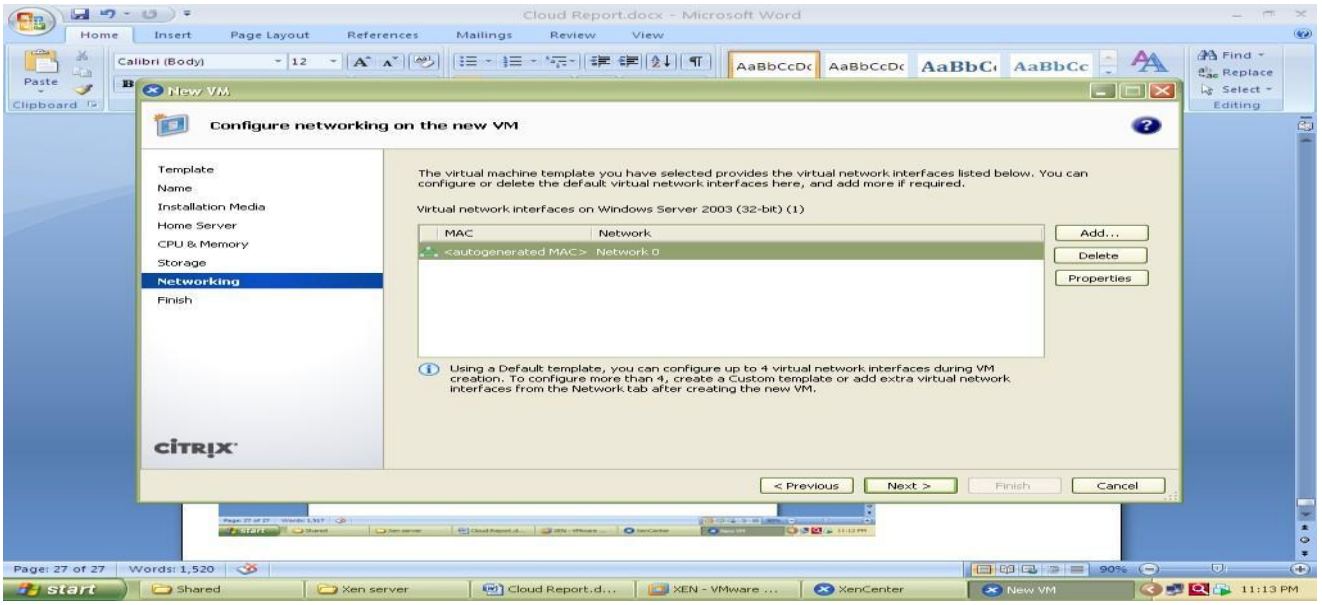
Now select hardware for vm i.e. no. of cpu's and memory



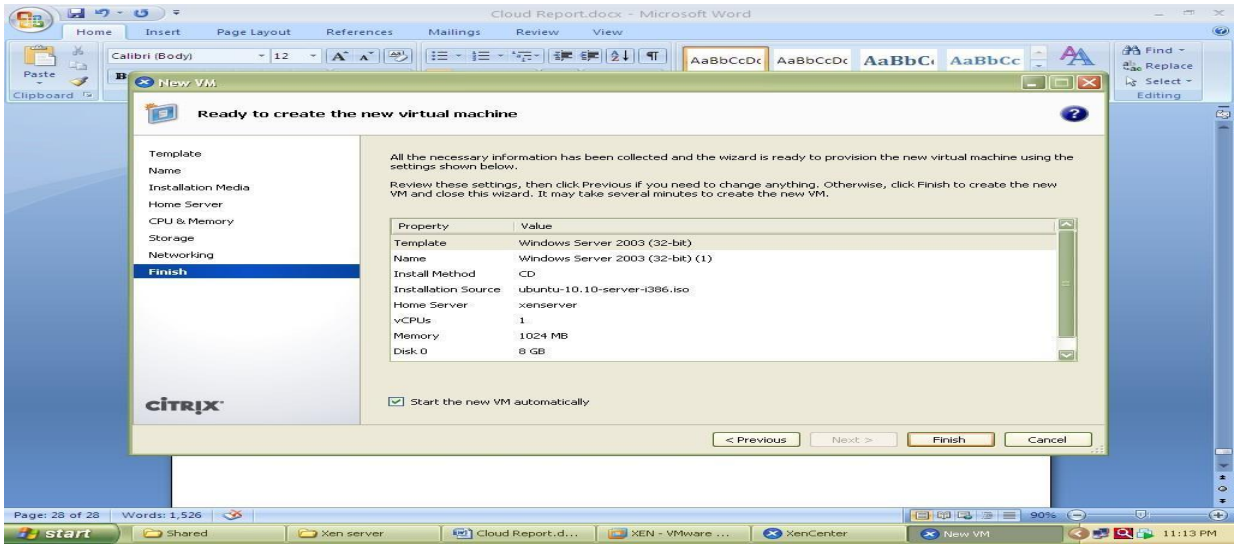
Select local storage



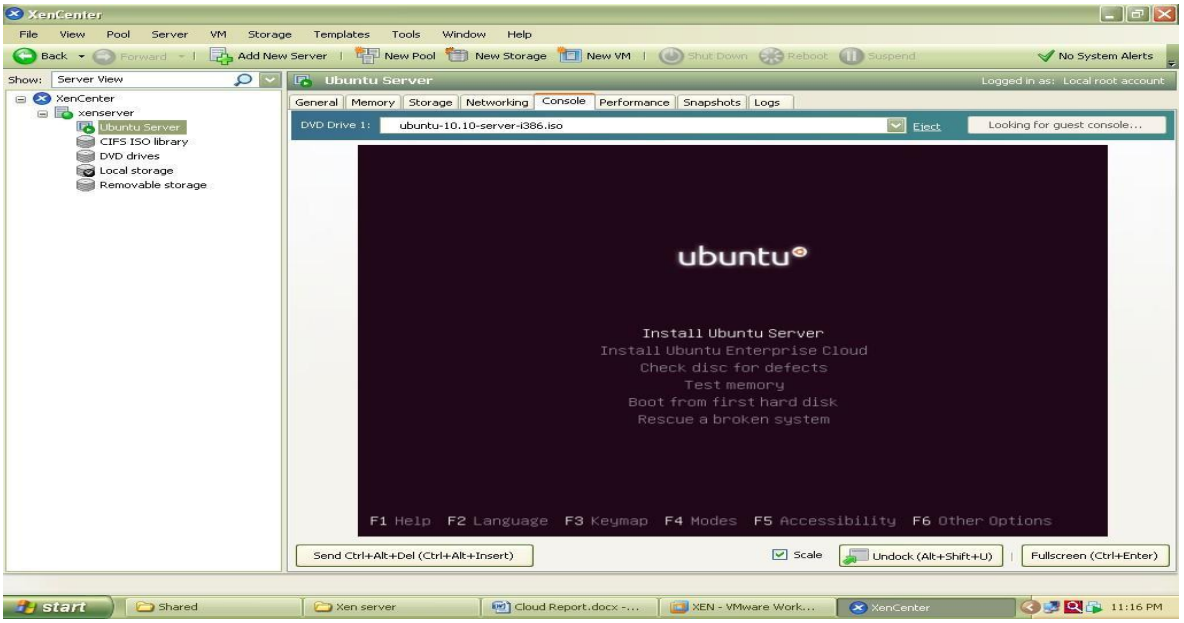
Select network



And click on finish

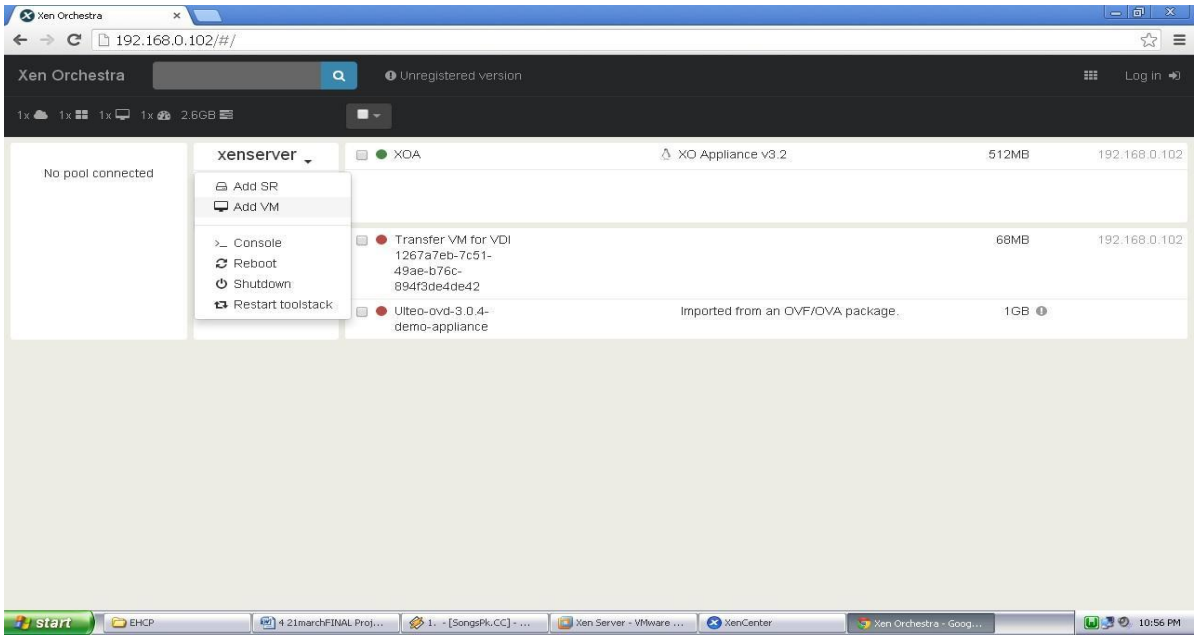


Now go to Console tab to install ubuntu and follow installation Steps.

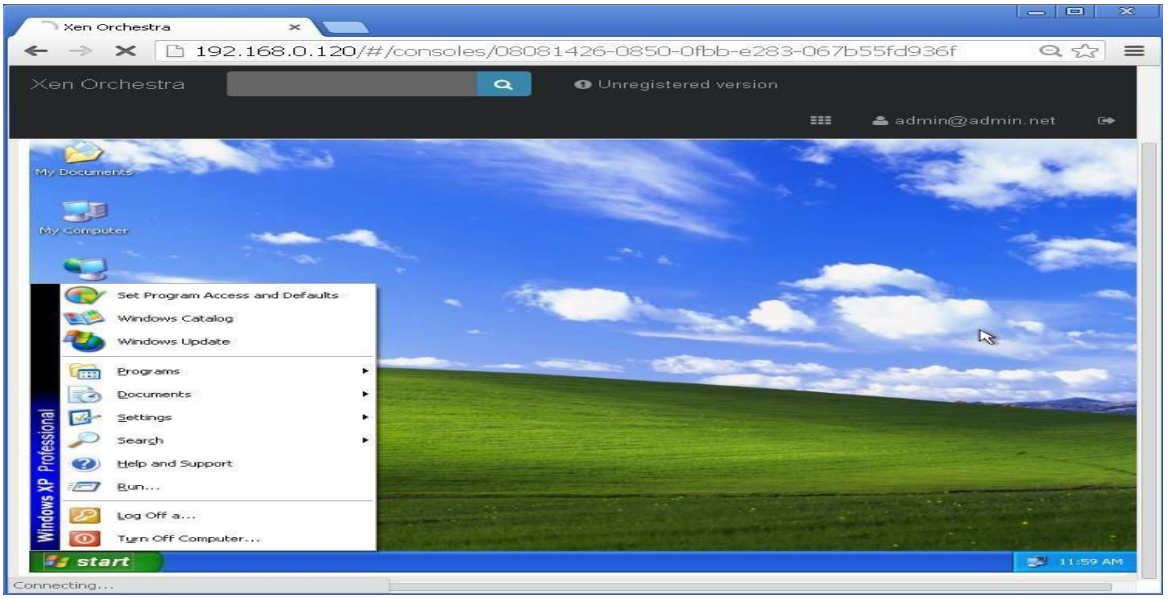


The Xen orchestra provides web based functionality of Xen Center.it provides access to all the VMs with their lifecycle management which are installed over Xen Server shown in figure

Xen Orchestra (XOA) Portal



The Windows XP image running on Xen Orchestra over Google chrome web browser is shown in following screenshot



Windows XP running on Xen orchestra (XOA)

Conclusion:

Comment on Virtual Machine you created on the software tool used.

Experiment No.4
Infrastructure as a Service(IaaS) using Open stack
Date of Performance:
Date of Submission:

Aim: To study and Implement Infrastructure as a Service using AWS/Microsoft Azure

Objective: To study and Implement IaaS using your resources..

Theory:

DevStack Minimum Requirements

- Fresh installation of Ubuntu 18.04
- Minimum memory of 4 GB
- At least 2 vCPU
- The storage capacity of 10 GB should be sufficient.
- Internet connection
- User with sudo

Step 1: Update Ubuntu system

Login to your Ubuntu system – Can be Desktop or VM in the Cloud and update it.

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

Reboot it after an upgrade.

sudo reboot

Step 2: Add Stack User

Devstack should be run as a non-root user with sudo enabled. If you're running your instance in the cloud, standard logins to cloud images such as "**centos**" or "**ubuntu**" or "**cloud-user**" are usually fine.

For other installations of Ubuntu 18.04, run the commands below to create DevStack deployment user.

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

Enable sudo privileges for this user without need for a password.

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

Switch to `stack` user to test.

jmutai@devstack:~$ sudo su - stack

stack@devstack:~$ sudo su -

root@devstack:~#

Step 3: Download DevStack

Clone Destack deployment code from Github.

su - stack

sudo apt -y install git

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

Create a `local.conf` file with 4 passwords and Host IP address.

cd devstack

nano local.conf

Add:

[[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 - get your Server/VM IP address from ip addr command

HOST_IP=192.168.10.100

Step 4: Start Openstack Deployment on Ubuntu 18.04 with DevStack

Now that you've configured the minimum required config to get started with DevStack, start the installation of Openstack.

cd devstack

./stack.sh

DevStack will install;

- **Keystone** – Identity Service
- **Glance** – Image Service

- **Nova** – Compute Service
- **Placement** – Placement API
- **Cinder** – Block Storage Service
- **Neutron** – Network Service
- **Horizon** – Openstack Dashboard

This will take a 15 – 20 minutes, largely depending on the speed of your internet connection. At the end of the installation process, you should see output like this:

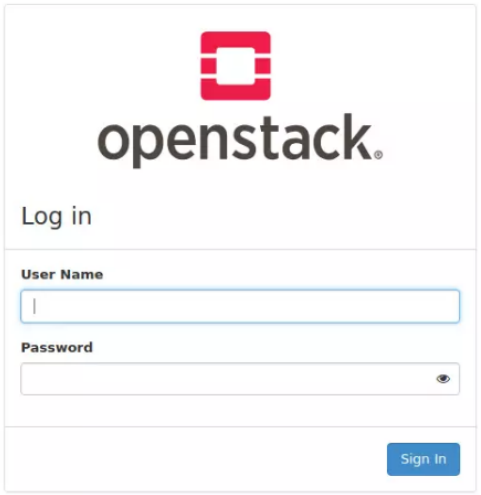
```
This is your host IP address: 192.168.10.100
This is your host IPv6 address: 2a01:4f8:c2c:308e::1
Horizon is now available at http://192.168.10.100/dashboard
Keystone is serving at http://192.168.10.100/identity/

The default users are: admin and demo
The password: StrongAdminSecret

WARNING:
Using lib/neutron-legacy is deprecated, and it will be removed in the future

Services are running under systemd unit files.
For more information see: https://docs.openstack.org/devstack/latest/systemd.html
DevStack Version: stein
Change: 8bdbf850967b90ebdca428247bb93ad2eb6478c0 Merge "Set ownership of /etc/pki/
files for TLS" 2019-03-26 08:07:26 +0000
OS Version: Ubuntu 18.04 bionic
2019-03-26 21:32:56.743 | stack.sh completed in 1761 seconds.
```

Step 5: Access OpenStack Dashboard

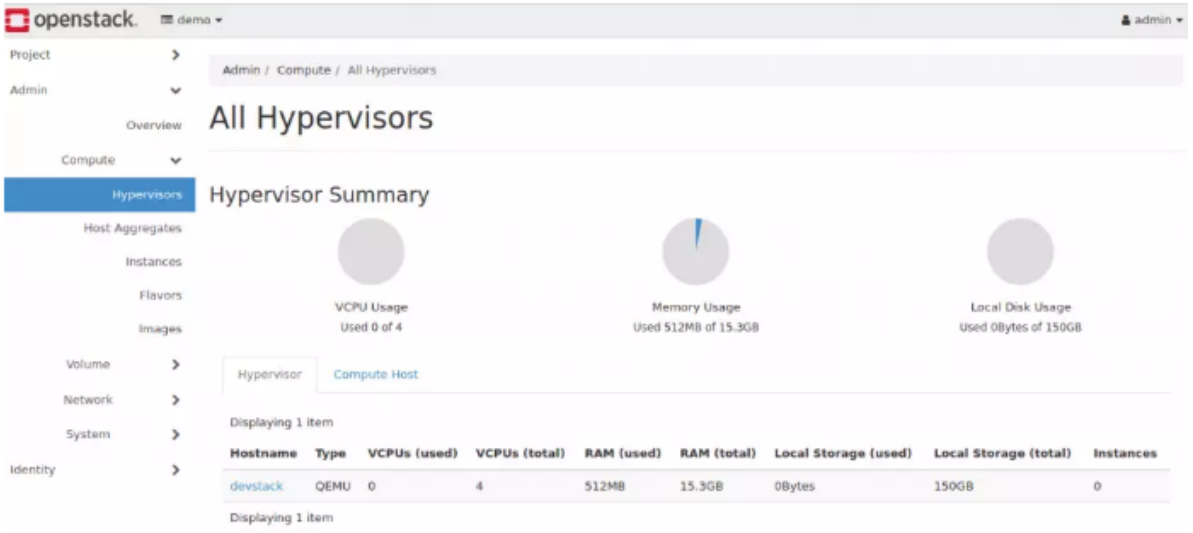
Copy the Horizon URL shown on the installation output and paste it into your web browser:

http://192.168.10.100/dashboard

Use the default users demo or admin and configured password to login.



You should see Openstack Management Web console after logging in.



If you want to use Openstack command-line tool to manage your devstack.

You've to source openrc in your shell.

source openrc

You can now add instance images to Glance to use when creating Virtual Machines with Nova.

We have a comprehensive guide on Adding images to Openstack Glance.

Adding Cirros Test Image

Download Virtual Image.

wget http://download.cirros-cloud.net/0.4.0/cirros-0.4.0-x86_64-disk.img

Upload it to Glance.

openstack image create \

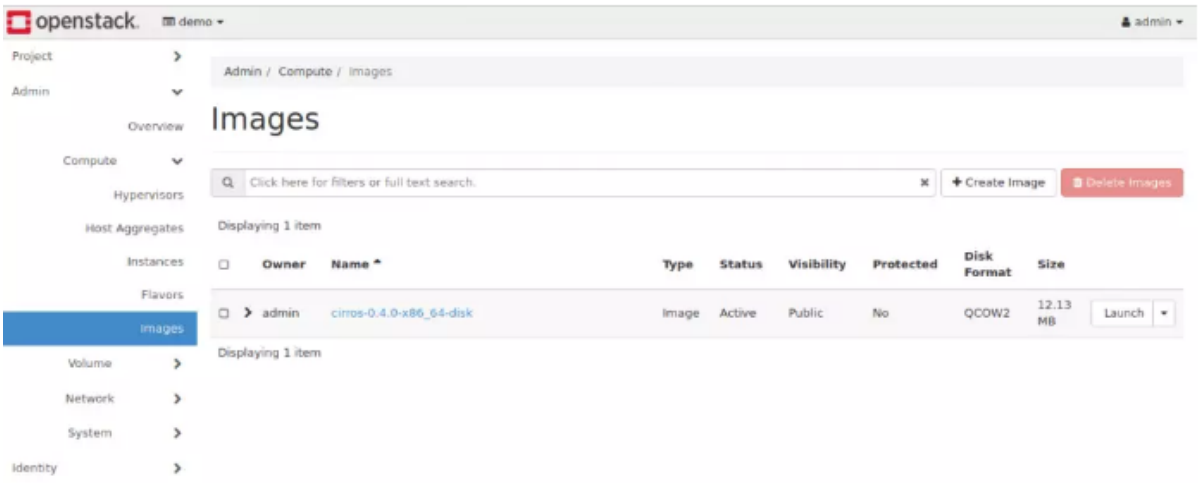
--container-format bare \

```
openstack image create \
  --container-format bare \
  --disk-format qcow2 \
  --file cirros-0.4.0-x86_64-disk.img \

    Cirros-0.4.0-x86_64
```

Confirm image upload.

```
$ openstack image list
+--------------------------------------+---------------------+--------+
| ID                                   | Name                | Status |
+--------------------------------------+---------------------+--------+
| 3c1e6e1e-a997-4f1a-a70d-06c4a6dafc37 | cirros-0.4.0-x86_64-disk | active |
+--------------------------------------+---------------------+--------+
```

It should be visible from the Web UI



Conclusion:

Comment on installation of OpenStack and services provided by OpenStack such as IaaS.

Experiment No.5
Implement Platform as a Service using AWS
Date of Performance:
Date of Submission:

Aim: To study and Implement Platform as a Service using OwnCloud/ AWS S3, Glaciers/ Azure Storage.

Objective: To learn concept of PaaS and implement using OwnCloud which gives universal access to files through a web interface.

Theory:

Conclusion: Comment on installation of PaaS in cloud computing.

Experiment No.6
Storage as a Service (SaaS) using OwnCloud.
Date of Performance:
Date of Submission:

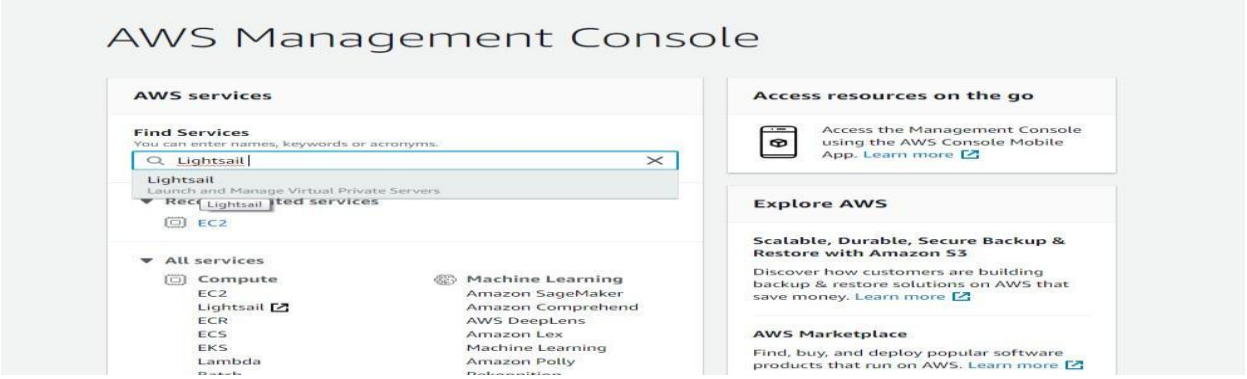
Aim: To study and Implement Storage as a Service using OwnCloud/ AWS S3, Glaciers/ Azure Storage.

Objective: To learn concept of SaaS and implement using OwnCloud which gives universal access to files through a web interface.

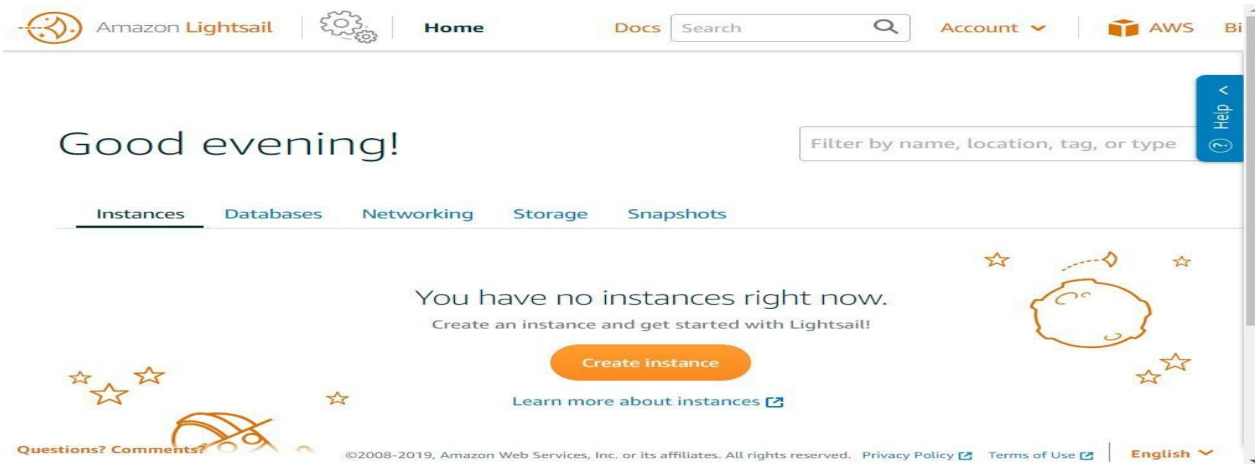
Theory:

To create simple word press app using Light sail service in AWS

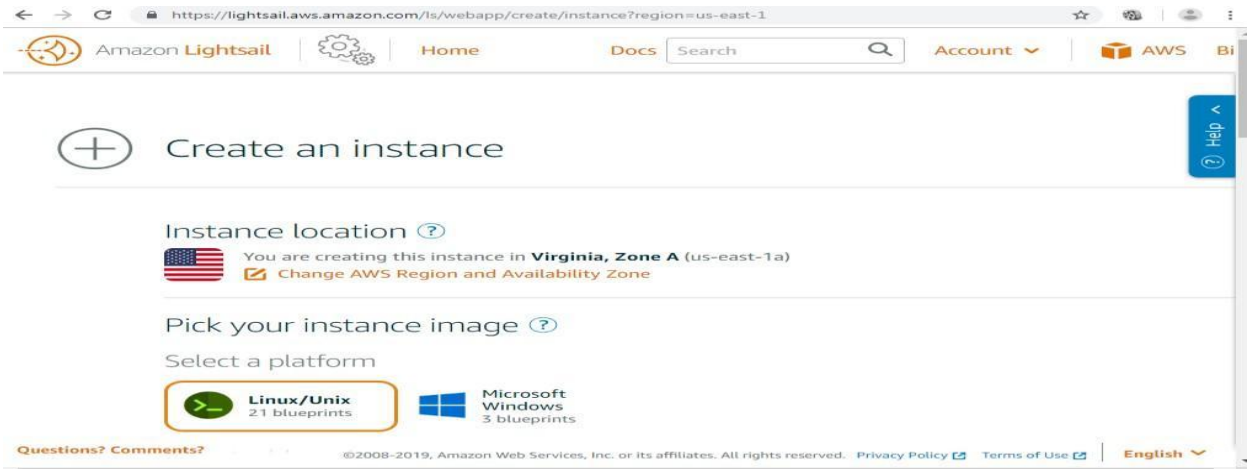
(SAAS) Step 1-: Open Admin console of AWS and select light sail service



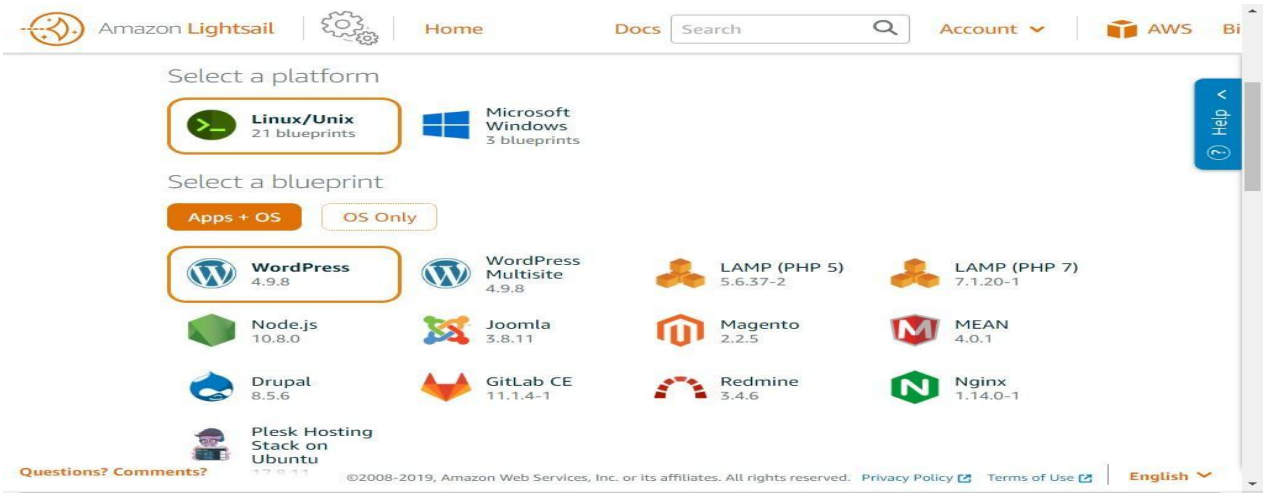
Step 2-: Select Create instance option



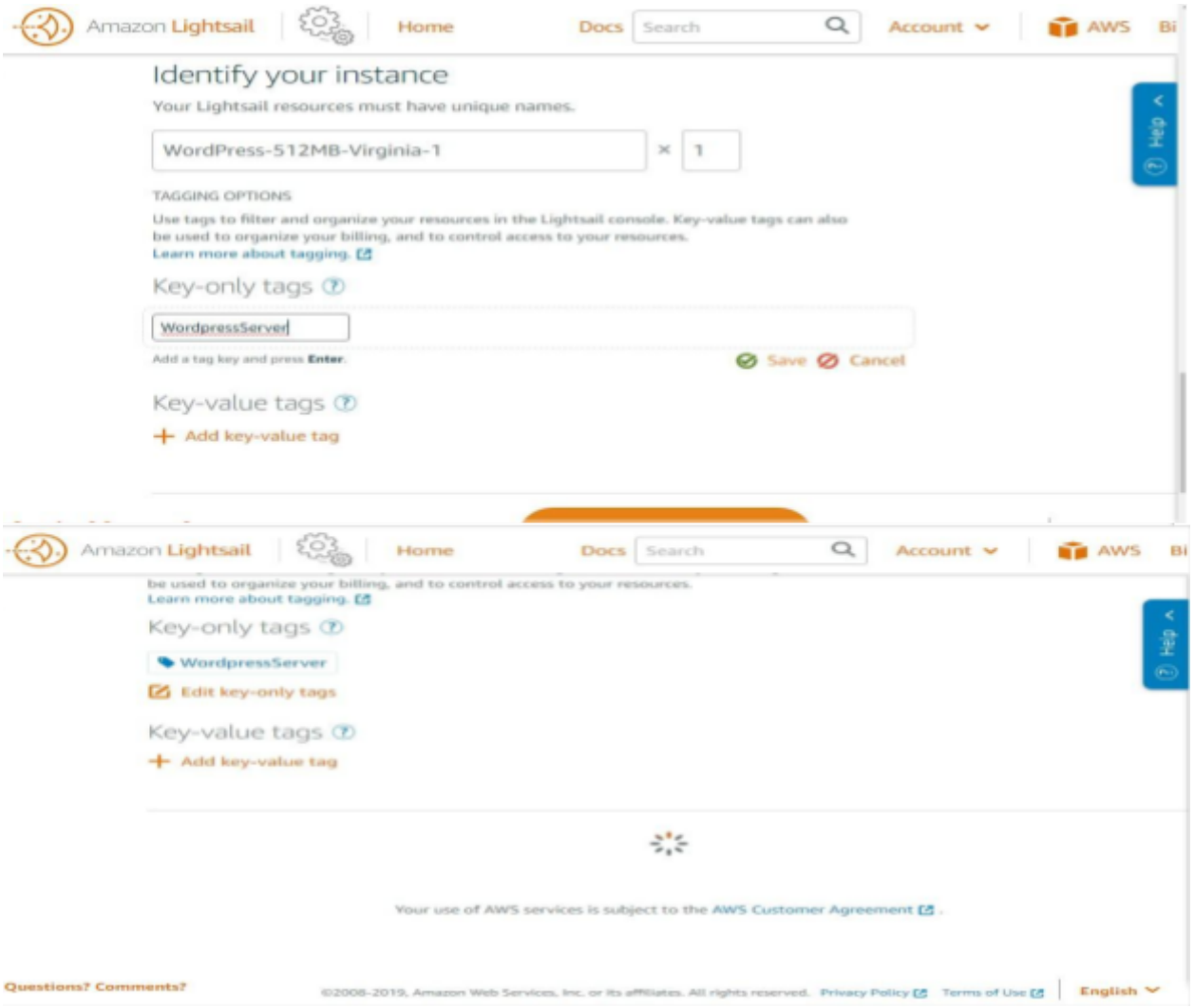
Step 3 -: Select the Linux hosting instance



Step 4 -: Select word press hosing

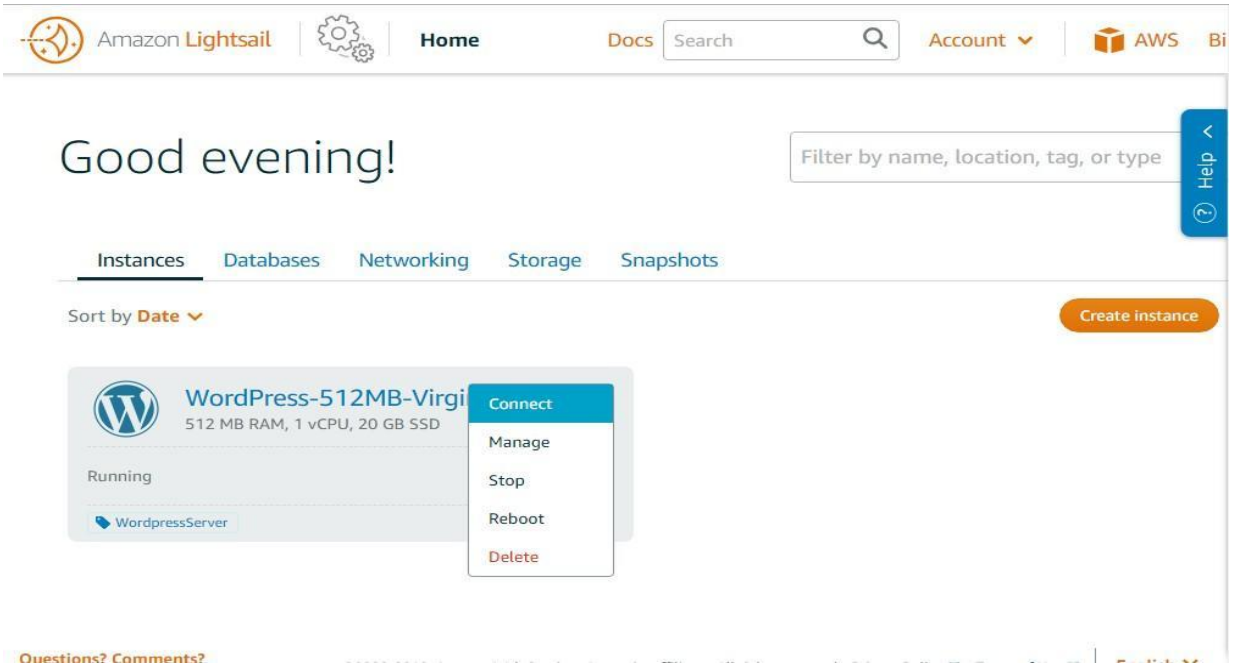


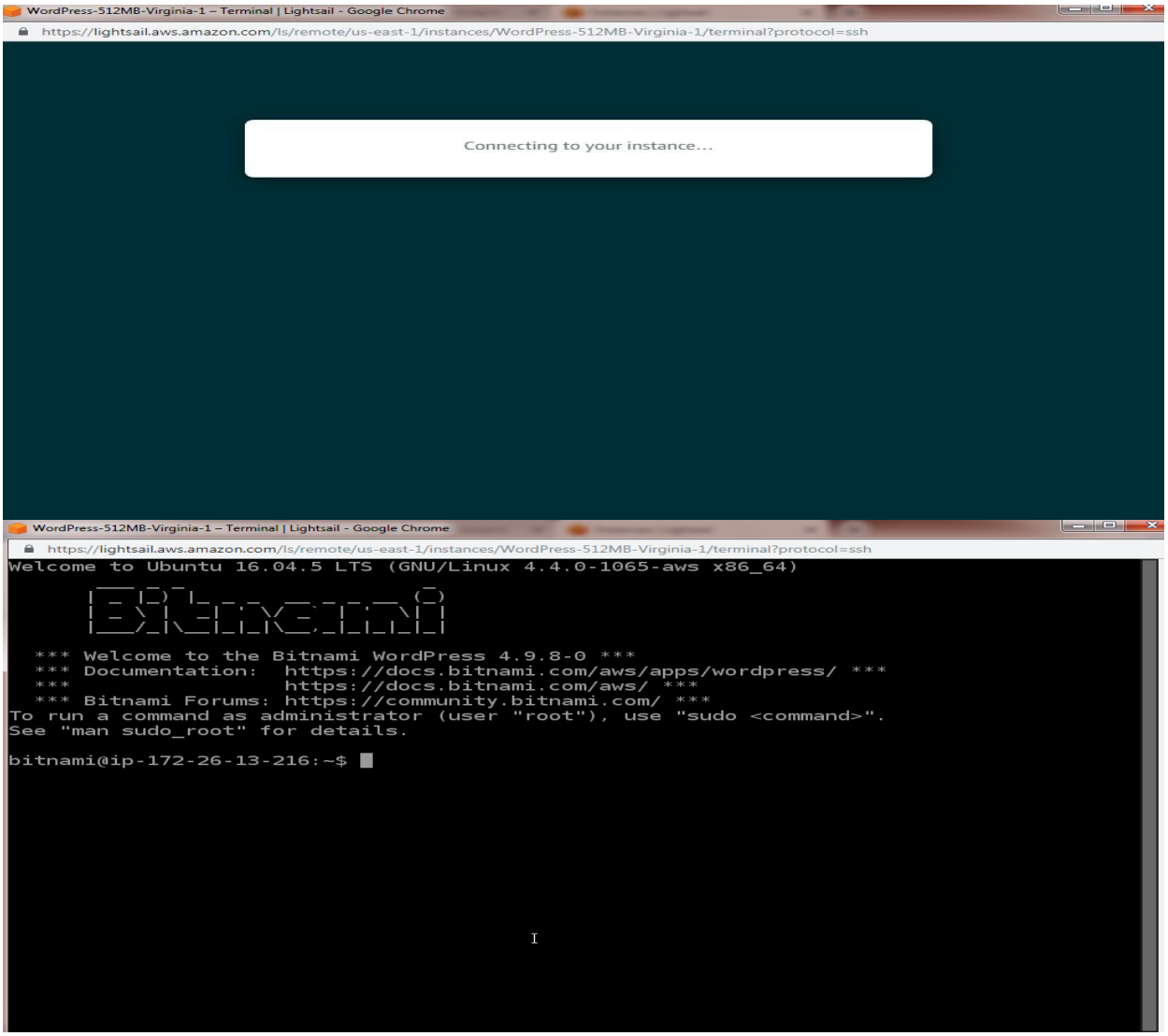
Step 5 -: Specify name to the instance



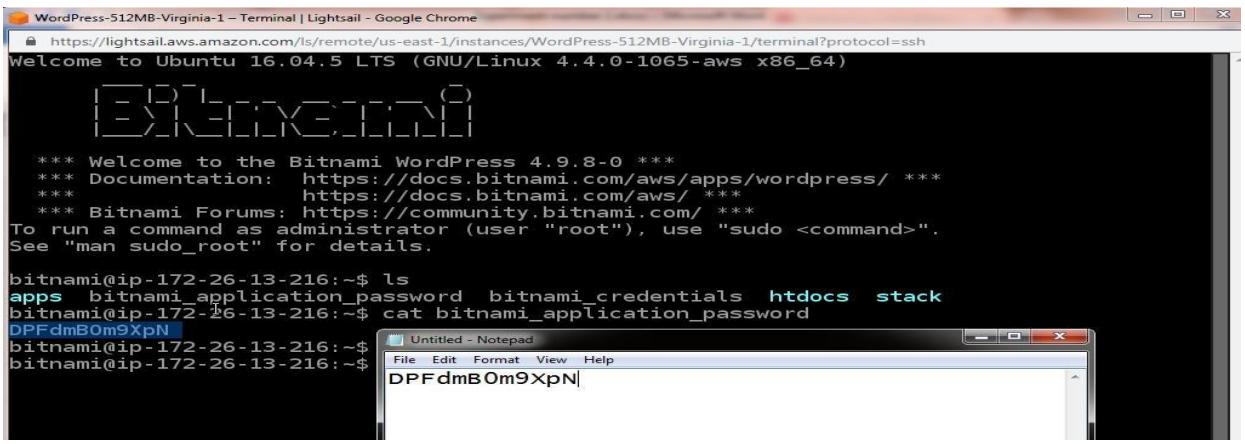
Step 6-: Now click on create to launch the instance Step

Step 7 -: Click on connect instance to get the password for word press

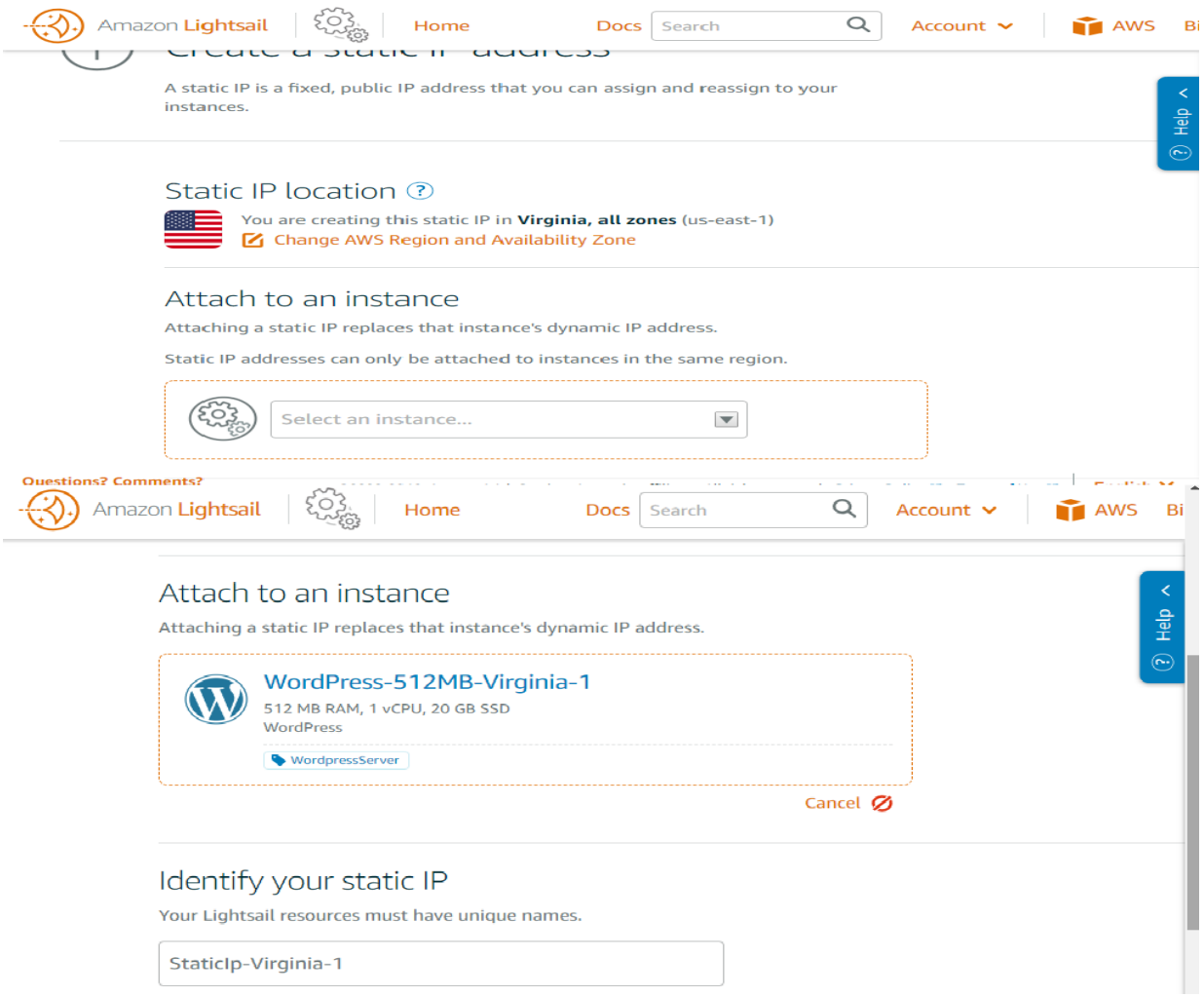

Amazon Lightsail
Home
Docs
Search
Account
AWS
Good evening!
Filter by name, location, tag, or type
Help
Instances
Databases
Networking
Storage
Snapshots
Sort by Date
Create instance
WordPress-512MB-Virgi
512 MB RAM, 1 vCPU, 20 GB SSD
Running
WordpressServer
Connect
Manage
Stop
Reboot
Delete
Questions? Comments?

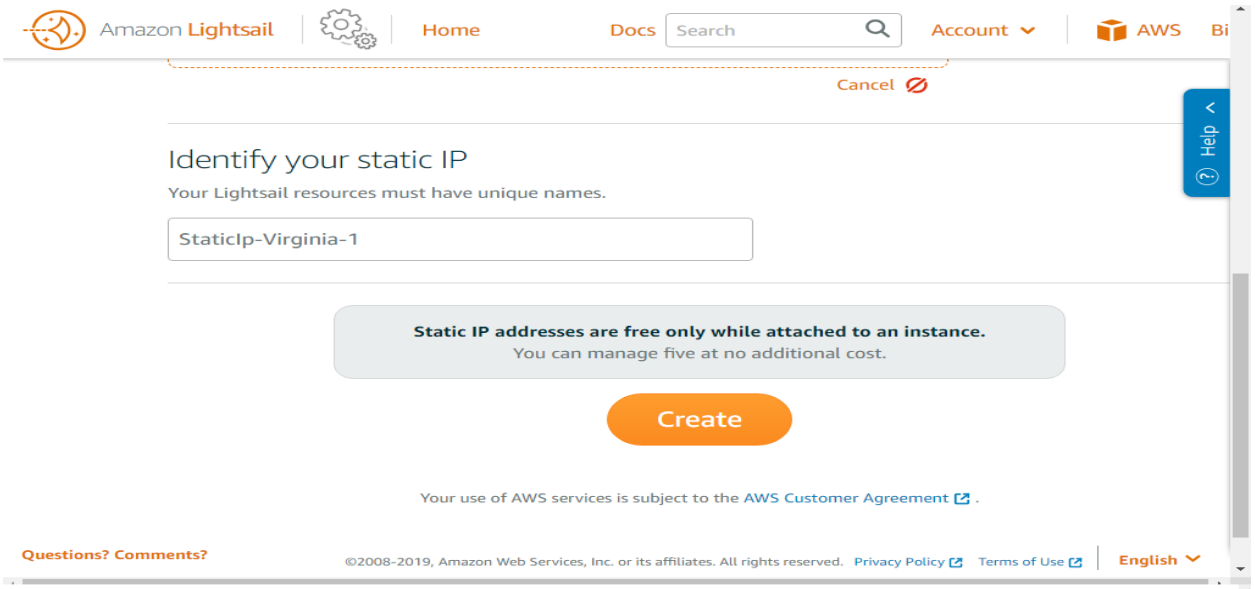


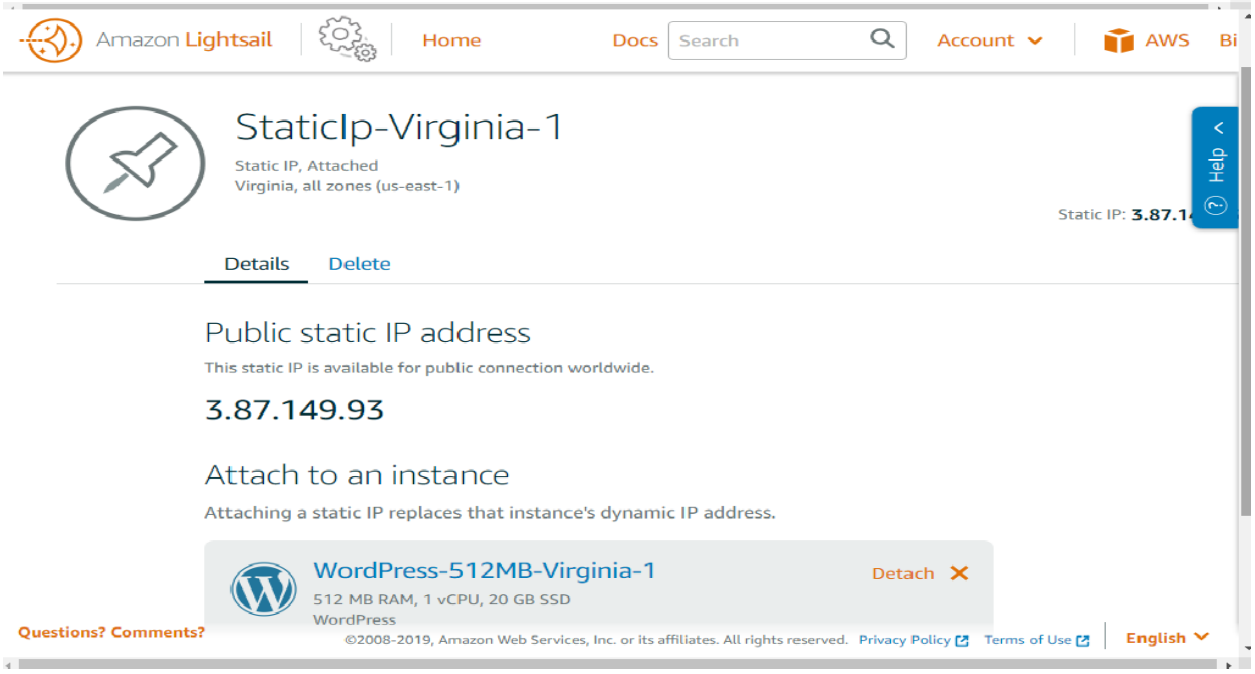
Step 8-: Now open bitnami_application_password file to get the admin password. So copy it and use over admin console



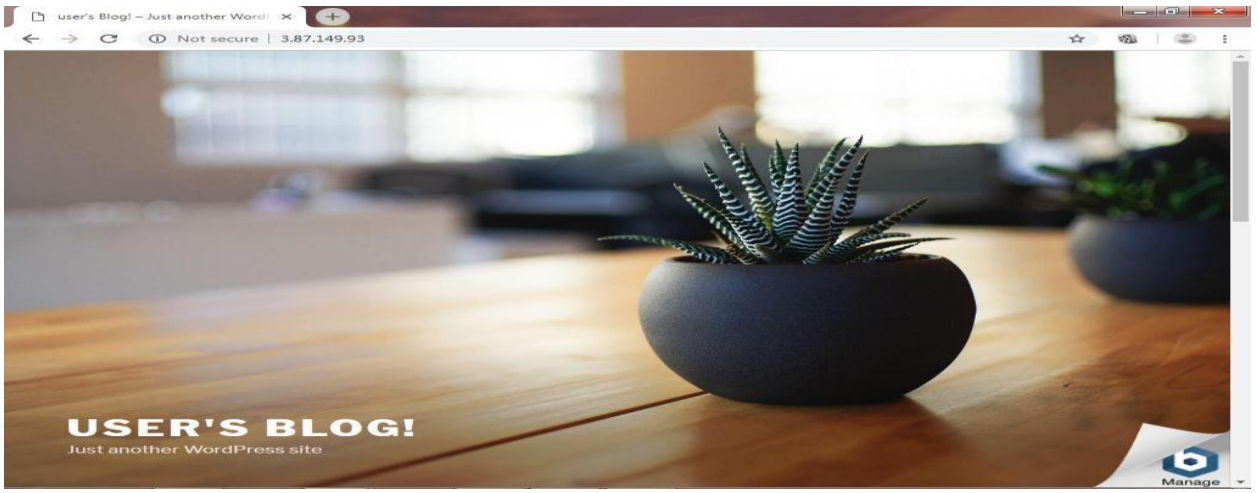
Step 9 -: Now researve static ip by selecting network option and creating static IP



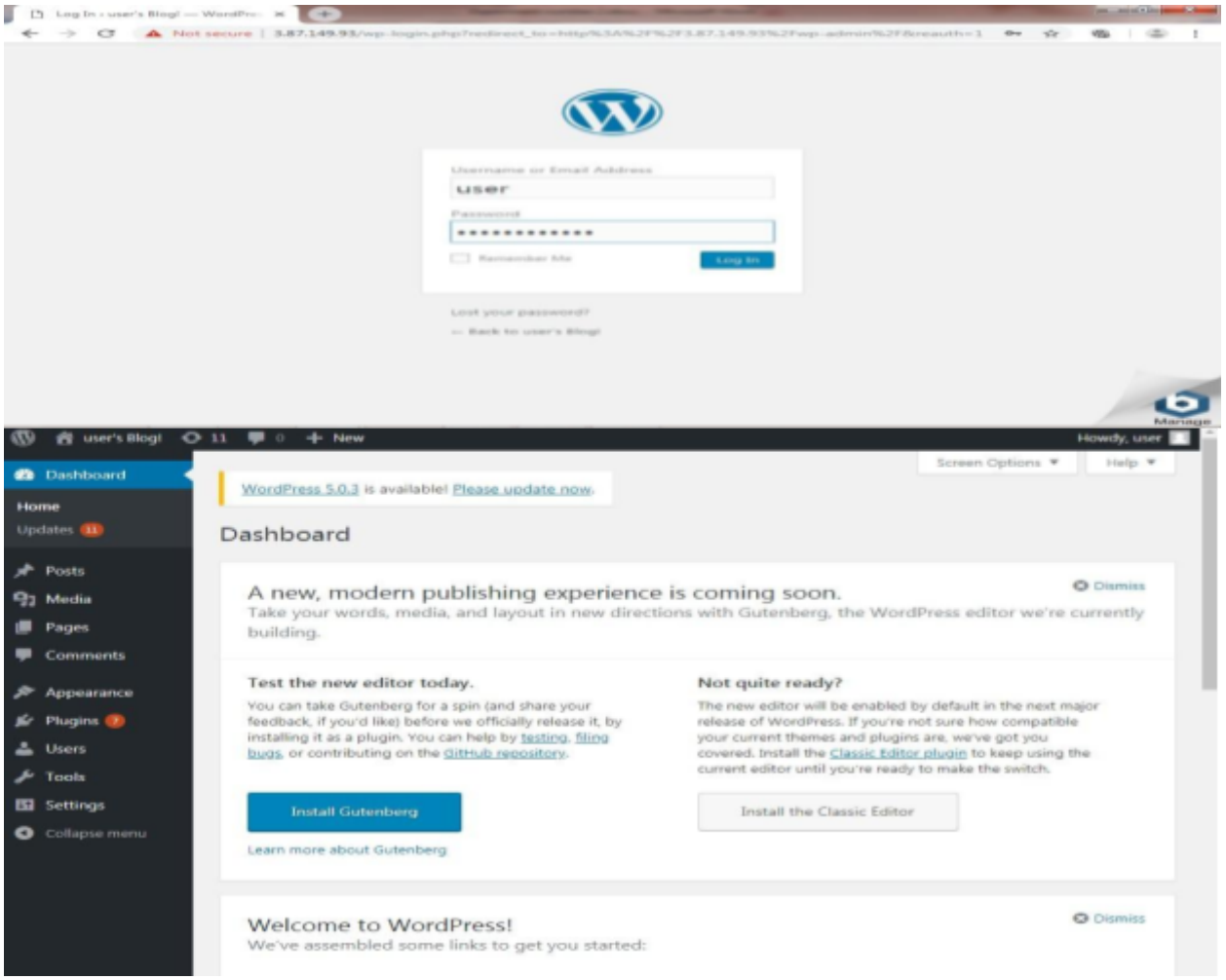




Once static IP is allocated then open that ip on browser to see Word press Website



Open admin console of Word press and use password obtained in step 8 to open word press site builder. Now you can develop a complete Word press website and use that.



Conclusion:

Comment on storage as a service in Amazon Web Services (AWS).

Experiment No.7
Cloud Security using OwnCloud.
Date of Performance:
Date of Submission:

Aim: Cloud Security using ownCloud.

Objective: To Understand security of web server and data directory.

Theory:

As the use of file sharing increases across the industry, more attention is being paid to the inherent security of these solutions and the need for corporations to provide enterprise file sync and share (EFSS) solutions that meet IT's security parameters.

Security Features of ownCloud

ownCloud's drivers for continued security improvement is to not only fix individual symptoms (e.g. the single bugs), but to also focus on identifying and resolving the root cause to prevent whole categories of vulnerabilities. ownCloud internal security processes and secure software development lifecycle aligns with industry standards such as ISOs 29147, 30111 and 27304.

- **Strict Content Security Policy :** Content Security Policy (CSP) is one of the most useful and powerful web security features introduced in recent years. With CSP, applications can instruct the browser to follow a specified security model, including instructions to not execute any inline scripts or load remote resources.

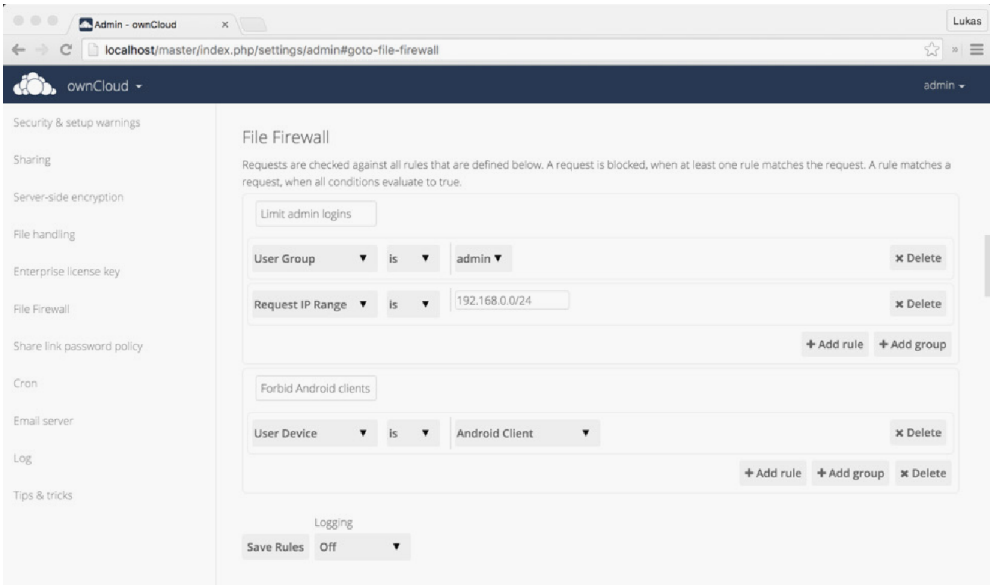
- **Data in Session is Stored Encrypted:** PHP stores session related data within sessions. These are usually small files on the server containing data such as the login state or the username. We have hardened the PHP session storage in such a way that the ownCloud server can only read session data at the same time the user is using ownCloud. This is done by encrypting the stored session data with an encryption key stored in another cookie. If the userrequests a page on ownCloud the encryption cookie will be sent by the sync clients or the web browser. Only with this cookie (which is not stored on the disk of the application server) can the session content be decrypted.

- **Secure by Default Model:** New ownCloud code uses the so called "ownCloud App Framework", a modern MVC-like framework to develop code for ownCloud. Code relying on this framework uses a lot of secure defaults such as requiring CSRF (another specific kind of web vulnerability caused by the original design of the web) and authentication checks being opt-out rather than the more common (and less safe) opt-in. The default mode for every critical security feature in ownCloud is "on", and requires the developer to deliberately "opt-out" of these security checks. These secure defaults are part of ownCloud's secure software development lifecycle. Secure

defaults make it more difficult to accidentally trigger a security vulnerability Instead, it allows internal and external security professionals to easily assess the overall security of an ownCloud component.

- **Strict Comparison in PHP Technically Enforced:** PHP has some peculiarities such as "Type Juggling". This means that it will automatically try to convert data types when applicable such as in comparisons. An example would be the following comparison: "0 == false" where PHP will try to convert both values (integer and Boolean value) into a comparable state and thus, will return true. This can lead to unexpected behavior and potential security bugs if developers don't take this into consideration. ownCloud forces PHP to confirm that the data is exactly the same type by verifying the data type using strict comparisons as a best practice.

- **File Firewall :** Using the internal File Firewall of ownCloud's Enterprise Edition, enterprises can limit access to sensitive data even further. File Firewall is an application-level firewall that inspects all incoming ownCloud requests and evaluates them based on rules set by the administrator to only allow through *"approved"* requests for a finely granular level of control. Administrators can, for example, limit administrative logins to a pre-defined internal network to enhance security or allow access to shared folders only from a specific location to implement internal security guidelines.



There have been several reports of various cloud storage services getting hacked of late. You can actually create your own cloud storage solution that you can control with an open source service called ownCloud. It is a simple way to set up your own syncing and Dropbox-like cloud storage system on your own server or website. It is robust, quick and easy to set up, and does not require much advanced technical knowledge. Let's have a quick look at how you can set up a private cloud using ownCloud.

Prerequisites

You do not really need much to get started with setting up ownCloud, but should have the following handy before you start.

1. **Any Web host that supports PHP5 and MySQL (or SQLite):** You should be signed up for a service like Dreamhost. If you already have a domain name (like *http://www.name.com*) through a Web host, you can install ownCloud in a couple of minutes. You will not have to deal with things like MySQL and PHP for installing ownCloud. All of these are taken care of automatically. You just need to make sure that your hosting service supports them.

2. **A copy of ownCloud Server 5:** You can actually install ownCloud in different ways, but let's just stick to the simplest method, i.e., using the Web installer. If you know how to put a file onto your website, you can install this.

3. **A URL for remote access:** Since you would prefer to tap into ownCloud from anywhere, you will need a URL to do so. If you do not already have a domain name, you can buy one; but if you do, it's very easy to set up ownCloud in a sub-directory of your site.

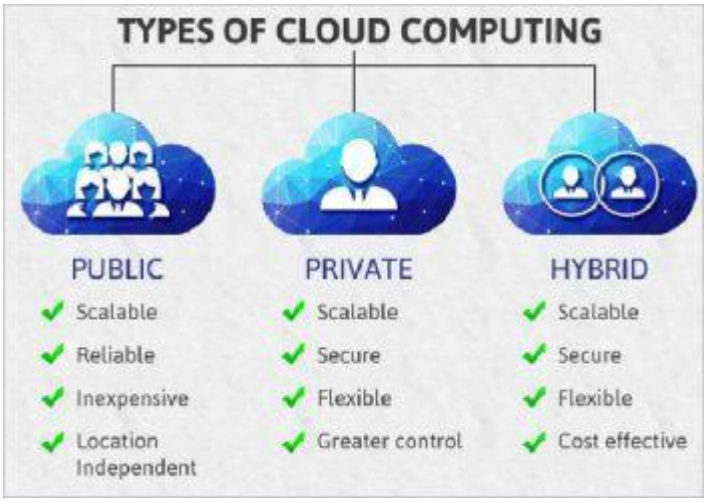


Figure : Different types of cloud storage systems (Image source: googleimages.com)

Initial setup and installation

As was decided earlier, we will use the Web installer, with the help of which ownCloud automatically creates everything you need so that no special skills are required to get it set up. If you have multiple users wanting to access ownCloud, it is recommended that you manually create a database. Here is how you can install ownCloud using the Web installer.

1. Download and save the Web installer to the computer.

2. Upload the setup file – owncloud.php – to the Web space using the host's Web interface or an FTP application.

3. Enter the URL of the setup file into the Web browser. It should be something like ***http://www.yourdomainname.com/setup-owncloud.php*.**

4. Follow the given on-screen instructions to install ownCloud. After a few minutes, you will be redirected to the login page.

The setup for desktop and mobile sync

Now that you have ownCloud installed on your Web server, it's time to set up desktop sync so that the files in ownCloud are the same format as those on your computer. For this, you need to install the desktop client (Linux, Windows or Mac).

From here onwards, the setup is pretty simple:

1. Open up ownCloud software on the computer, and select *Configure*.
2. Add the URL of the ownCloud server along with the login credentials.

3. Now, you need to select the folders and files that you want to sync. Click *Add folder* and select a folder that needs to be synced on the computer. All the files present in this folder will be uploaded and synced automatically to ownCloud. Similarly, you can add as many folders as you like.

As with Dropbox, you can simply drag files into the Web interface so that they can be uploaded and synchronised both locally as well as in the cloud, and you can also share files with friends by selecting the Share option when you hover the mouse over a file.

Syncing the calendar, address book and music

Syncing calendars: If you use a calendar app that supports CalDAV, you just need to point it to your ownCloud installation.

1. Click the *Calendar* icon on the right side.
2. Click the gear icon present in the top right.
3. Copy down the URL for the calendar (most of the calendars can access simple URLs, but iOS requires a slightly different URL).

Now, simply open the settings of your favourite calendar app, and add your account in the CalDAV section. All your appointments will be directly dumped into ownCloud and will be automatically synced across any other device you connect to it.

Sync your contacts: Similar to the calendar, you can easily import and sync your address book with ownCloud.

1. Export all the Contacts from the address book into a VCF file.
2. From ownCloud, select the Contacts sidebar, and then click the gear present in the bottom left corner.
3. Click Import and then select the VCF file to be uploaded.

It will take a few minutes to get all the contacts uploaded, but once they are up, you can synchronise them with any address book that supports CardDAV.

Add apps and extend ownCloud's power

It is even possible to extend ownCloud's functionality with the help of additional applications. If you click your user name from the ownCloud Web interface and select Apps, you are taken to a list of the installable applications. You can also browse through a few more such applications over here. To install any of the external applications, you just need to select the app, and click Enable. After a few moments, it will be installed and you will find a new icon on the right panel.

The benefits of setting up your own cloud storage system are many, some of which are:

1. It helps to access the data present on the cloud from anywhere; hence, it's flexible.
2. It is highly scalable.
3. As it is a private cloud, it has got higher security.
4. It is easy to sync files from your desktop to the cloud.
5. It enables you to sync ownCloud with almost any desktop or mobile calendar and contacts application.

Conclusion:

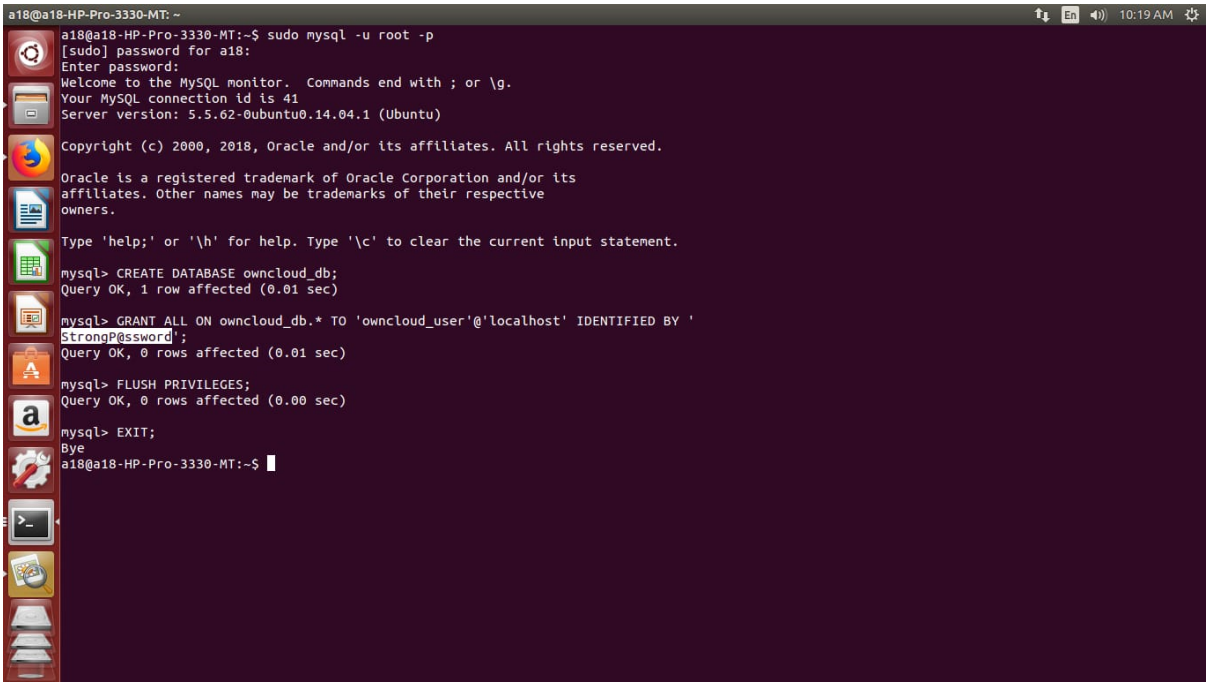
Comment on installation of OwnCloud and various security features such as Strict Content Security Policy, Data in Session is Stored Encrypted etc.

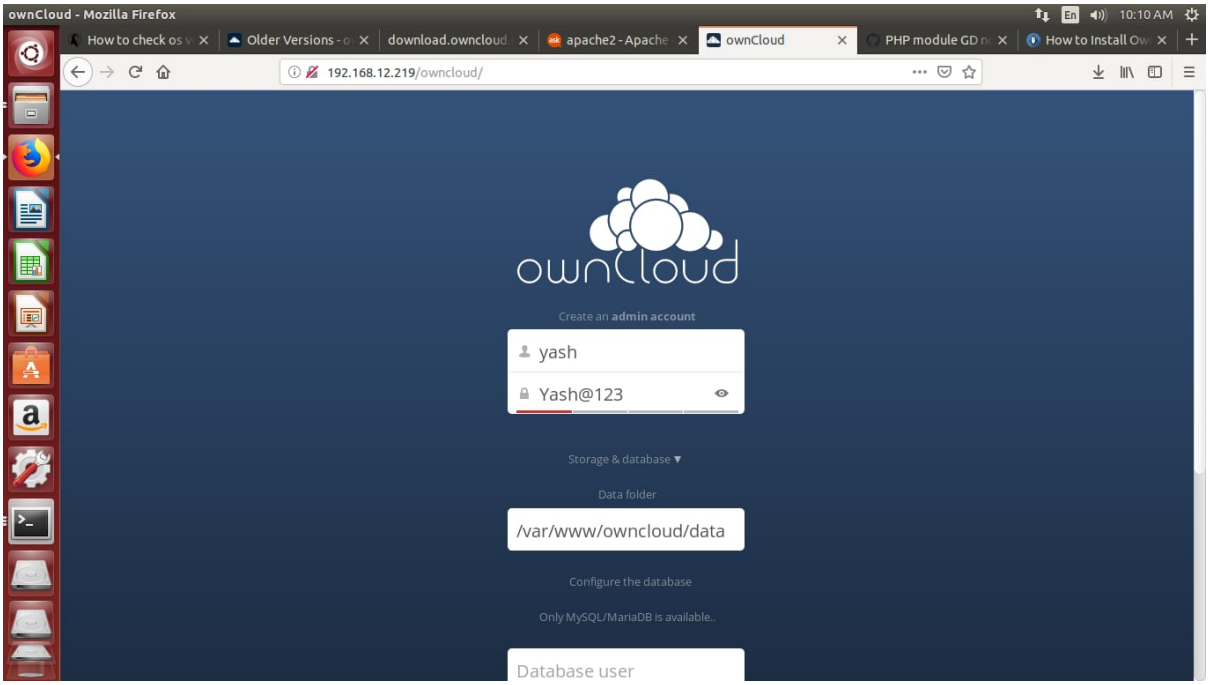
Experiment No.8
To study and implement virtual own cloud.
Date of Performance:
Date of Submission:

Aim: To study and implementation of creation of virtual own cloud

Objective: To Understand virtual cloud imlementation.

Theory:





Conclusion: comment on Steps performed to create virtual own cloud

Experiment No.9
To deploy different cloud web services on created virtual own cloud
Date of Performance:
Date of Submission:

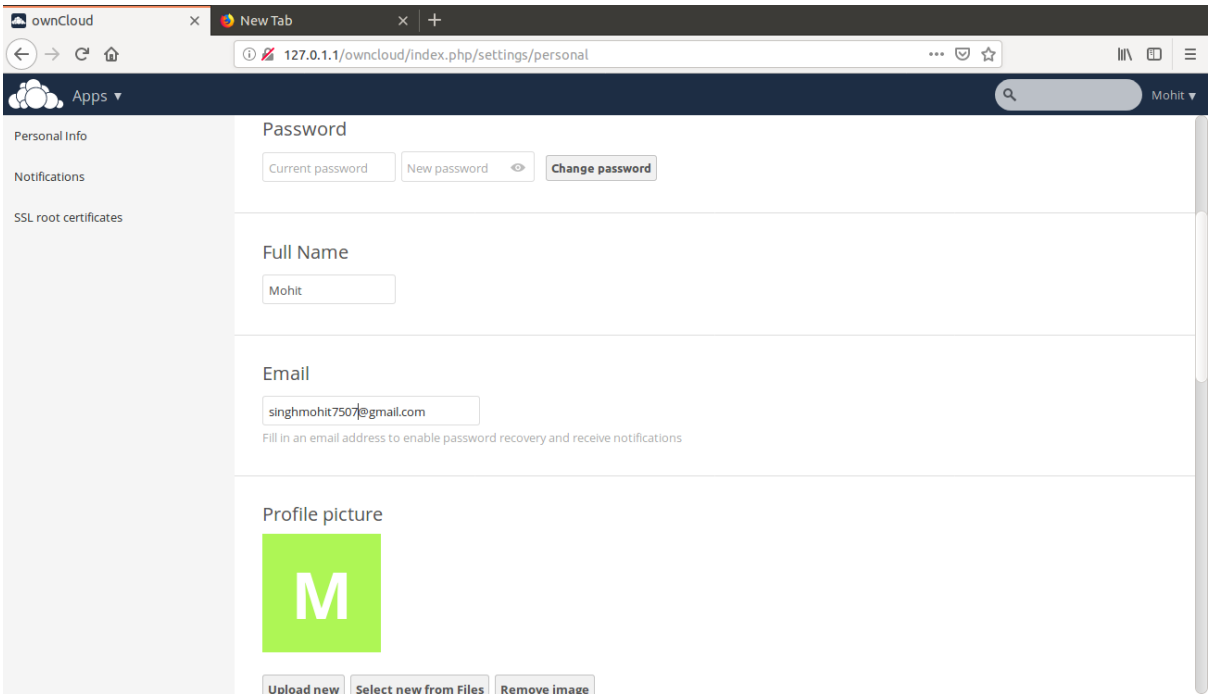
Aim: To deploy different cloud web services on created virtual own cloud

Objective: To Understand cloud web services .

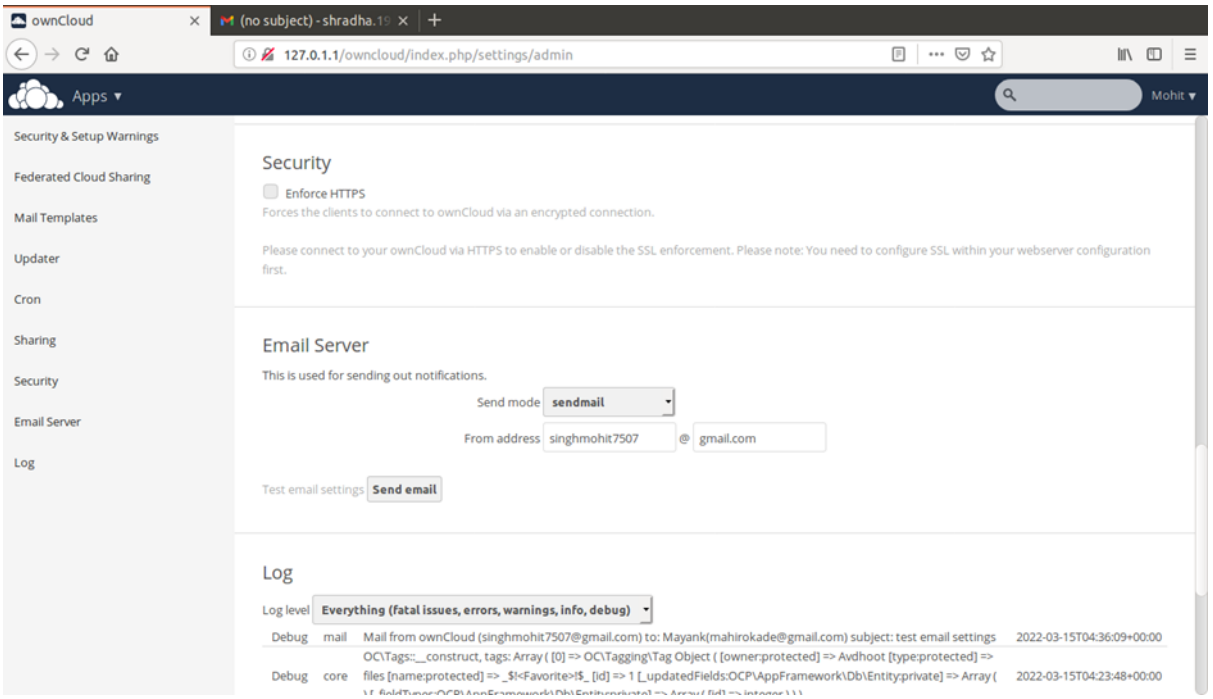
Theory:

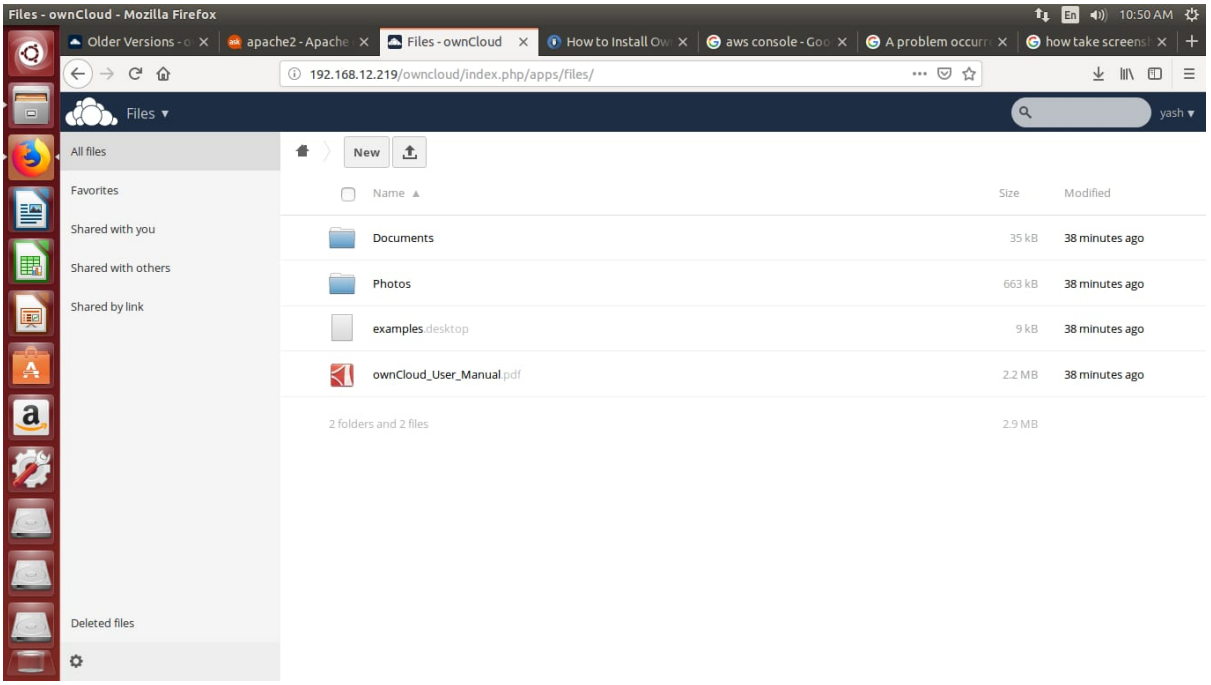
Steps performed on created virtual own cloud are as follows:

1: Login system



2: configuration:


ownCloud
(no subject) - shradha.19
127.0.1.1/owncloud/index.php/settings/admin
Apps
Mohit
Security & Setup Warnings
Federated Cloud Sharing
Mail Templates
Updater
Cron
Sharing
Security
Email Server
Log
Security
Enforce HTTPS
Forces the clients to connect to ownCloud via an encrypted connection.
Please connect to your ownCloud via HTTPS to enable or disable the SSL enforcement. Please note: You need to configure SSL within your webserver configuration first.
Email Server
This is used for sending out notifications.
Send mode sendmail
From address singhmohit7507 @ gmail.com
Test email settings Send email
Log
Log level Everything (fatal issues, errors, warnings, info, debug)
Debug mail Mail from ownCloud (singhmohit7507@gmail.com) to: Mayank(mahirokade@gmail.com) subject: test email settings 2022-03-15T04:36:09+00:00
Debug core 2022-03-15T04:23:48+00:00


Files - ownCloud - Mozilla Firefox
10:50 AM
Older Versions
apache2 - Apache
Files - ownCloud
How to Install
aws console - Goo
A problem occurr
how take screens
192.168.12.219/owncloud/index.php/apps/files/
Files
yash
All files
Favorites
Shared with you
Shared with others
Shared by link
Deleted files
New
Name
Size
Modified
Documents 35 kB 38 minutes ago
Photos 663 kB 38 minutes ago
examples.desktop 9 kB 38 minutes ago
ownCloud_User_Manual.pdf 2.2 MB 38 minutes ago
2 folders and 2 files 2.9 MB

Conclusion: Comment on Steps performed on created virtual own cloud

Experiment No.10
To study and Implement Containerization using Docker
Date of Performance:
Date of Submission:

Aim: To study and Implement Containerization using Docker

Objective: To understand the use of containerization.

Theory:

Containerization allows developers to create and deploy applications faster and more securely. With traditional methods, code is developed in a specific computing environment which, when transferred to a new location, often results in bugs and errors .

Version of Docker Server Engine running on the Host:

```
[root@docker-mast /]# docker --version
Docker version 18.09.5, build e8ff056
[root@docker-mast /]# docker version
Client:
 Version:           18.09.5
 API version:       1.39
 Go version:        go1.10.8
 Git commit:        e8ff056
 Built:             Thu Apr 11 04:43:34 2019
 OS/Arch:           linux/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          18.09.5
  API version:      1.39 (minimum version 1.12)
  Go version:       go1.10.8
  Git commit:       e8ff056
  Built:            Thu Apr 11 04:13:40 2019
  OS/Arch:          linux/amd64
  Experimental:     false
[root@docker-mast /]#
```

Containers are running on host:

```
[root@docker-mast /]# docker ps
```

```
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS
NAMES
[root@docker-mast /]#
```

Images are available on host:

```
[root@docker-mast /]# docker images
REPOSITORY   TAG   IMAGE ID   CREATED   SIZE
[root@docker-mast /]#
```

Run a container using the ubuntu image which was just downloaded:

```
[root@docker-mast /]# docker run ubuntu
[root@docker-mast /]# docker ps
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS
NAMES
[root@docker-mast /]# docker ps -a
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS
NAMES
314a7fe862a9   ubuntu   "/bin/bash"   54 seconds ago   Exited (0) 14 seconds ago
festive_mclean
[root@docker-mast /]#
```

Conclusion: Comment on use of containerization using Docker .