@@@@@ LW Cloud Computing @@@@@

Cloud computing, also known as on-demand computing, is a kind of internet-based computing, where shared resources and information are provided to computers and other devices on-demand.
Cloud computing is a model for enabling ubiquitous, 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

Deployment models

1. Private cloud

Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party, and hosted either internally or externally.[4] Undertaking a private cloud project requires a significant level and degree of engagement to virtualize the business environment, and requires the organization to reevaluate decisions about existing resources.

2. Public cloud

A cloud is called a "public cloud" when the services are rendered over a network that is open for public use. Public cloud services may be free.[77] Technically there may be little or no difference between public and private cloud architecture, however, security consideration may be substantially different for services (applications, storage, and other resources) that are made available by a service provider for a public audience and when communication is effected over a non-trusted network. Generally, public cloud service providers like Amazon AWS, Microsoft and Google own and operate the infrastructure at their data center and access is generally via the Internet.

3. Hybrid Cloud

Hybrid cloud is a composition of two or more clouds (private, community or public) that remain distinct entities but are bound together, offering the benefits of multiple deployment models. Hybrid cloud can also mean the ability to connect collocation, managed and/or dedicated services with cloud resources

Major Types of cloud

There is some most usable and Demanding Cloud Platform

1. **SAAS** (Software as a Service)
2. **StAAS** (Storage as a service)
3. **IAAS** (Infrastructure as a service)
4. **Pass** (Plateform as a Service)

Let's Start with SAAS (software as a service)

SAAS: [Software as a Service]

This type of cloud service means the client doesn't have to worry about any
Particular software installation or license charges and system compatibility
Means client can use any desired software without installing in their own system they client need to pay as per there requirement like per month or per hours basis.

Note: Some SAAS based companies are listed here

i) Antenna Software
ii) Cloud9 Analytics
iii) CVM Solutions
iv) Exoprise Systems
v) GageIn

How to setup SAAS cloud in single Node system

A general method for making your system to serve as **SAAS**

Requirement

i) A Linux based any GUI installed system
ii) ssh server with X-window system running
iii) IP and other networking things must be pre configured

Note: i am using redhat 6/7 to perform all this

For installing SSH server

A)
[root@desktop57 Desktop]# **yum install openssh-server**

b) configure the ssh server for gui support

[root@desktop57 Desktop]#**vim /etc/ssh/sshd_config**
X11Forwarding yes

c) start the service for server and make it persistent

[root@desktop57 Desktop]#**service sshd restart**
Stopping sshd: [OK]
Starting sshd: [OK]
[root@desktop57 Desktop]# **chkconfig sshd on**

Important:

Now your ssh server with X-windows system is ready Now you can install the software's which you want to server to clients

LinuxWorld, 5, Krishna Tower, Next to Triveni Nagar Flyover, Gopalpura Bypass, JAIPUR-15. Ph: 0141-3224438
Website : www.linuxworldindia.org E-mail : training@lwindia.com

Client:
=======

Client can have any kind of operating system

Case 1: You have client with Linux based system
==========

Note: Install a software "openssh-clients"

[root@desktop57 Desktop]# **yum install openssh-clients**

Now you can access the desired software

Method:

ssh -X root@IP softwarename

Note: Here IP is Server IP address

Example:

ssh -X root@192.168.0.100 firefox

Case 2: You have client based on Windows
==========

There is client software putty or plink

##########

LinuxWorld, 5, Krishna Tower, Next to Triveni Nagar Flyover, Gopalpura Bypass, JAIPUR-15. Ph: 0141-3224438
Website : www.linuxworldindia.org E-mail : training@lwindia.com

StAAS (Storage As A Service)

The One of the most amazing service of cloud computing service for providing Cyber Storage or Online Remote storage
which is independent of Client system and its plateform.

Benefits:
=========

Easy to Access and Manage
Very cheap
Desired size

Type of Storage
==========

There are two type of Remote Storage generally provided by Cloud Storage

1. Object Storage
2. Block Storage

Object Storage

This is the kind of storae where client only can store its data like (Docs,Pdf,text,Movies) etc
you can not change FileSystem of the provided storage and also you can not create any partitions.

Example of Object Storage:
==========

Amazon: - S3 (simple storage service)
Google: - Gdrive
DropBox: - free storage
Openstack: - Swift

Here we are going to create own Object Storage Service

Requirement:

i) Redhat / Centos {Operating system/Server}
ii) Free space in / Mount point Or Space in Hard Disk
iii) Any Protocol that can share storage like (NFS,SMB,SSHFS,Glusterfs).. etc
iv) IP address and LAN or WAN connectivity with Clinets

Important: Storage Server have IP address (192.168.0.254)
==========

Note: You can also use your external hard disk

Share Your Storage Over the Network Using NFS (Network File System)
==========

Case 1: You are going share Storage from (/) partition of Linux

Step 1: Create A Directory and assign some specific permission
---

```
[root@desktop57 Desktop]# mkdir /mydata

[root@desktop57 Desktop]# chmod o+w  /mydata
```

Note: You can also create partition or any type like LVM or General then mount it to share

Step 2: Configure NFS server and manage Services and Security
---

```
[root@desktop57 Desktop]#yum install nfs-utils

[root@desktop57 Desktop]#echo "/mydata
192.168.0.10(rw,sync,no_root_squash)"   >>/etc/exports
```

Note: Here 192.168.0.10 is the Ip address of Client who can access storage
==========

If You have more than one client to share the same storage then you can add more IP address or complete subnet mask like given below :

```
[root@desktop57]#echo "/mydata  192.168.0.10(rw,sync)
192.168.0.20(rw,sync,no_root_squash)"  >>/etc/exports
```

OR

For complet subnet
==========

```
[root@desktop57]#echo "/mydata  192.168.0.0/24(rw,sync,no_root_squash)"
>>/etc/exports
```

OR

Whole World
==========

```
[root@desktop57]#echo "/mydata   *(rw,sync,no_root_squash)"
>>/etc/exports
```

Important: Manage all required service of NFS
==========

```
[root@desktop57 Desktop]#service rpcbind restart

[root@desktop57 Desktop]#chkconfig rpcbind on

[root@desktop57 Desktop]#service nfs restart

[root@desktop57 Desktop]#chkconfig nfs on
```

Important: Maintain security and firewall
==========

```
[root@desktop57 Desktop]# setenforce   0   (Off selinux security)
[root@desktop57 Desktop]# iptables -F       (flush all firewall rules)
```

Now turn for Client

Case 1: Linux based Client

Means client also using Linux based O.S

To access Shared storage with NFS

i) Install the required software

[root@desktop57 Desktop]#**yum install nfs-utils**

ii) Check connectivity with server

[root@desktop57 Desktop]#**showmount -e 192.168.0.254**

iii) Creating Mount points to access cloud storage

[root@desktop57 Desktop]# **mkdir /media/mystore**

iv) Mount the Cloud storage

[root@desktop57 Desktop]#**mount 192.168.0.254:/mydata/media/mystore**

Important: Now you have a cloud storage on Your Desktop check this out by own

Note:

If you Reboot you Clinet machine then you need to remount the storage so make it persistant you need to do a specific change in your client machine

[root@desktop57]#**echo "192.168.0.254:/mydata /media/mystore nfs** _netdev,defaults 0 0" >>/etc/fstab

Note: Via NFS there is very low level or not any support with every windows version so we will access via SMB

Case 2: Storage Sharing with CIFS / SMB

Step 1: Create A Directory and assign some specific permission

---

```
[root@desktop57 Desktop]# mkdir /mydata

[root@desktop57 Desktop]# chmod o+w  /mydata
```

Note: You can also create partition or any type like LVM or General then mount it to share

Step 2: Configure Samba server and manage Services and Security

---

i) Installing required software's

```
[root@desktop57 Desktop]#yum install samba samba-client
```

OR

```
[root@desktop57 Desktop]#yum install samba*
```

ii) Configuring samba server

==========

Note: creating samba user and assigning the password

```
[root@desktop57 Desktop]# useradd   -s  /sbin/nologin   vimal
[root@desktop57 Desktop]# smbpasswd -a  vimal
```

Open configuration file and go to last line then made some given entry

---

```
[root@desktop57 Desktop]#vim /etc/samba/smb.conf
```

```
        [publicdata]
path=/mydata
hosts allow = 192.168.0.0/24
writable = yes
valid users = vimal
browseable = yes
```

iii) Start the Service
==========

```
[root@desktop57 Desktop]# service smb restart
[root@desktop57 Desktop]# chkconfig smb on
```

IV) Manage security and firewall

```
[root@desktop57 Desktop]# setenforce   0
[root@desktop57 Desktop]# iptables -F
```

##########

Now How to access storage from client Side

I) From Windows client
==========

Go to windows run method and type

\\192.168.0.254

Here: 192.168.0.254 is IP of Server it will also ask for username and password

II) From Linux Client
==========

a) Install the Software

```
[root@desktop57 Desktop]#yum install cifs-utils samba-client
```

b) Create Mount Point

```
[root@desktop57 Desktop]#mkdir /media/data
```

c) Mount the Storage

```
[root@desktop57 Desktop]# mount  -o username=vimal
//192.168.0.254/publicdata  /media/data
```

Note: Here //ServerIP/Sharename (you can find from configuration file)

Important: like NFS you cal also make this storage Persistance

==========

```
echo "//192.168.0.254/publicdata  /media/data cifs
_netdev,username=vima,password=redhat" 0  0 >>/etc/fstab
```

##########$$$$$$$$$$$$$$$$

LinuxWorld, 5, Krishna Tower, Next to Triveni Nagar Flyover, Gopalpura Bypass, JAIPUR-15. Ph: 0141-3224438
Website : www.linuxworldindia.org E-mail : training@lwindia.com

Block Storage

we have already discussed about Block storage the type of storage which provide raw storage over network
this is the method where you can share real block storage over network using real time os

Note: For sharing block storage we can use ISCSI or FC protocol

==========

Here we are going to use ISCSI protocol

==========

More about ISCSI

==========

iSCSI, which stands for Internet Small Computer System Interface, works on top of the Transport Control Protocol (TCP) and allows the SCSI command to be sent end-to-end over local-area networks (LANs), wide-area networks (WANs) or the Internet. IBM developed iSCSI as a proof of concept in 1998, and presented the first draft of the iSCSI standard to the Internet Engineering Task Force (IETF) in 2000. The protocol was ratified in 2003.

How ISCSI works:

==========

iSCSI works by transporting block-level data between an iSCSI initiator on a server and an iSCSI target on a storage device. The iSCSI protocol encapsulates SCSI commands and assembles the data in packets for the TCP/IP layer. Packets are sent over the network using a point-to-point connection. Upon arrival, the iSCSI protocol disassembles the packets, separating the SCSI commands so the operating system (OS) will see the storage as a local SCSI device that can be formatted as usual. Today, some of iSCSI's popularity in small to midsize businesses (SMBs) has to do with the way server virtualization makes use of storage pools. In a virtualized environment, the storage pool is accessible to all the hosts within the cluster and the cluster nodes nodes communicate with the storage pool over the network through the use of the iSCSI protocol.

Note: past the given Image here name of Image is Iscsi_initaitor.png

Steps:
=====

1) First we need to create a partition general or LVM type
==========

Here we are going to create LVM partition

```
[root@desktop57 Desktop]# fdisk -cu /dev/sda

Command (m for help): p

Disk /dev/sda: 160.0 GB, 160000000000 bytes
255 heads, 63 sectors/track, 19452 cylinders, total 312500000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0007456d

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048     1050623      524288   83  Linux
/dev/sda2         1050624   154650623    76800000   8e  Linux LVM
/dev/sda3       154650624   155699199      524288   82  Linux swap /
Solaris
/dev/sda4       155699200   312499999    78400400    5  Extended
/dev/sda5       155701248   281530367    62914560   8e  Linux LVM
/dev/sda6       281532416   289921023     4194304   83  Linux
/dev/sda7       289923072   290127871      102400   83  Linux

Command (m for help): n
First sector (290129920-312499999, default 290129920):
Using default value 290129920
Last sector, +sectors or +size{K,M,G} (290129920-312499999, default
312499999): +3G

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or
resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
```

2) Update partition table using partx command

```
[root@desktop57 Desktop]# partx -a /dev/sda
BLKPG: Device or resource busy
error adding partition 1
BLKPG: Device or resource busy
error adding partition 2
BLKPG: Device or resource busy
error adding partition 3
BLKPG: Device or resource busy
error adding partition 4
BLKPG: Device or resource busy
error adding partition 5
BLKPG: Device or resource busy
error adding partition 6
BLKPG: Device or resource busy
error adding partition 7
```

3) Create Physical volume

```
[root@desktop57 Desktop]# pvcreate /dev/sda7

 Physical volume "/dev/sda7" successfully created
```

4) Create Volume Group

```
[root@desktop57 Desktop]# vgcreate  mystorage  /dev/sda7

  Volume group "mystorage" successfully created
```

5) Now time for creating Logical Volume of desired size example (2GB)

```
[root@desktop57 Desktop]# lvcreate --name  myfirlv --size 2G mystorage
```

6) Configure the IScSi protocol to share the created LVM

a) Installing the software

```
[root@desktop57 Desktop]#yum install scsi-target-utils
```

b) Open the configuration file and make entry

```
[root@desktop57 Desktop]#vim /etc/tgt/targets.conf

<target myiqn>
    backing-store /dev/mystorage/myfirlv
</target>
```

c) Start the service and manage security

```
[root@desktop57 Desktop]# service tgtd  restart
[root@desktop57 Desktop]# chkconfig tgtd on
[root@desktop57 Desktop]# iptables -F
```

Now we are done with ISCSI server configuration it's time to connect from clients

Case 1: when client is Linux based system
==========

i) Installing the software

[root@desktop57 Desktop]#**yum install iscsi-initiator-utils**

Important: there are some certains steps to connect
==========

a) Send a discover packet to iscsi target server to find the IQN number of shared block storage

[root@desktop57 Desktop]# **iscsiadm --mode discoverydb --type sendtargets --portal 192.168.0.254 --discover**

Note: it will give you the Shared IQN for next step

b) Login to target server with Received IQN to access the hard disk

iscsiadm --mode node --targetname iqn.2001-05.com.doe:test --portal 192.168.1.1:3260 --login

Now you check the hard with fdisk command

##########@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

IAAS (Infrastructure as a Service)

This is the most usable and powerful cloud computing service where you can access any operating system with your hardware required without using your own hardware

Infrastructure as a Service (IaaS) is a form of cloud computing that provides virtualized computing resources over the Internet.

Infrastructure as a Service (IaaS) is a form of cloud computing that provides virtualized computing resources over the Internet. IaaS is one of three main categories of cloud computing services, alongside Software as a Service (SaaS) and Platform as a Service (PaaS).

In an IaaS model, a third-party provider hosts hardware, software, servers, storage and other infrastructure components on behalf of its users. IaaS providers also host users' applications and handle tasks including system maintenance, backup and resiliency planning.

IaaS platforms offer highly scalable resources that can be adjusted on-demand. This makes IaaS well-suited for workloads that are temporary, experimental or change unexpectedly.

Other characteristics of IaaS environments include the automation of administrative tasks, dynamic scaling, desktop virtualization and policy-based services.

IaaS customers pay on a per-use basis, typically by the hour, week or month. Some providers also charge customers based on the amount of virtual machine space they use. This pay-as-you-go model eliminates the capital expense of deploying in-house hardware and software. However, users should monitor their IaaS environments closely to avoid being charged for unauthorized services.

Examples:
========

Some most amazing examples of IAAS CLoud are:

Amazon EC2, Redhat Openstack, Microsoft azure etc.

so like these IAAS cloud we can create our own

Requirement:
==========

i) any hypervisor for distributing resources like cpu/ram

ii) Storage Device for installing os

Note:

We are working with Redhat linux Operating system so we will use (QEMU-KVM) type-2 hypervisor for managing CPU and RAM to VM.

Important: You need to check in your system that your BIOS/UEFI is able to support Virtualization
or Below given command to check.

so in linux based os there is a command to check that your CPU is virtualized supported or not

For Intel Cup:
==========

[root@desktop57 Desktop]# **grep -i vmx /proc/cpuinfo**

if any output shows in terminal then it is supported

For AMD CPU:
==========

[root@desktop57 Desktop]# **grep -i svm /proc/cpuinfo**

if any output is shown on the terminal then it is supported

Now Installing KVM virtual machine in Your Redhat OS

[root@desktop57 Desktop]# **yum install libvirt qemu-kvm virt-manager**

[root@desktop57 Desktop]# **service libvritd restart**

Note: Now you can use virtual machine in redhat
==========

Installation of O.S Inside virtual Machine

There are three types of method for installing os upon your KVM based Virtual machine

A) By CD/DVD or ISO file
B) By Networking protocol like (ftp/http/nfs)
C) By kick start File method

Note: In kvm based Virtual machine you can two the method for installing os

By using GUI method:
==========

Everything using graphical interface:

Step 1
======
Launch Virtual Machine Manager by either selecting in Applications or running virt-manager in the command line.

Step 2
======

In the VM Manager window, create a new virtual machine by selecting New

Step 3
======

Follow the installation steps of creating a new VM. Specify the following:

Location of the guest operating system file
Operating system type and version
Memory and CPU Settings
Disk Image (either new or existing) and disk image size
Architecture

Step 4
======

Select "Finish" and installation of the VM will begin.

Important:
=========
You can make your installing very fast using GUI and In Real Industry
Your server is generally configured with Command Line .

By Using Command Line
======================

Case 1: When we are using ISO file to install os.
=======

```
[root@desktop57 Desktop]# virt-install  --name  rhel71 --ram  2048  --
vcpu 1  --cdrom /rhel-server-7.0-x86_64-dvd.iso  --disk
path=/var/lib/libvirt/images/rhel71.img,size=10  --os-type linux --os-
variant  rhel7  --hvm --extra-args=" ip=192.168.0.155
netmask=255.255.255.0 gateway=192.168.0.254 dns=192.168.0.254 noipv6"
```

This is command seems to be very long but this is very easy in reality
and i am going to explain what is written above in command line interface

Note: virt-install command is used to install virtual machine inside KVM
and there options are explained below

virt-install: options
======================

```
--name  "any name for os that you want to access"
--ram   "amount of RAM you want to give to your os from your main system"
--vcpu  "number of virtual cpu to be given"
--cdrom  "parameter of assigning ISO image location"
--disk   "location where .img file of  this particular vm is going to
store"
```

Case 2: When you are using Network Installation
=======

```
[root@desktop57 Desktop]# virt-install  --name  rhel71 --ram  2048  --
vcpu 1  --location ftp://192.168.0.254/pub/LinuxWorld-Extra/rhel6_4/  --
disk path=/var/lib/libvirt/images/rhel71.img,size=10  --os-type linux --
os-variant  rhel7  --hvm --extra-args=" ip=192.168.0.155
netmask=255.255.255.0 gateway=192.168.0.254 dns=192.168.0.254 noipv6"
```

Note:
=====

Here we replace --cdrom with --location

--location "URL of that place where your DVD is copied"

Case 3: when You are Using Kickstar File with Network Installation
=======

```
[root@desktop57 Desktop]# virt-install  --name  rhel71 --ram  2048  --
vcpu 1  --location ftp://192.168.0.254/pub/LinuxWorld-Extra/rhel6_4/  --
disk path=/var/lib/libvirt/images/rhel71.img,size=10  --os-type linux --
os-variant  rhel7  --hvm --extra-args=" ip=192.168.0.155
netmask=255.255.255.0 gateway=192.168.0.254 dns=192.168.0.254 noipv6
ks=nfs:192.168.0.254:/ks.cfg"
```

Case 4: Enabling VNC port of VM during Installation
=======

```
[root@desktop57 Desktop]# virt-install  --vnc --vncport=5911 --
vnclisten=0.0.0.0 --name  rhel71 --ram  2048  --vcpu 1  --location
ftp://192.168.0.254/pub/LinuxWorld-Extra/rhel6_4/  --disk
path=/var/lib/libvirt/images/rhel71.img,size=10  --os-type linux --os-
variant  rhel7  --hvm --extra-args=" ip=192.168.0.155
netmask=255.255.255.0 gateway=192.168.0.254 dns=192.168.0.254 noipv6
ks=nfs:192.168.0.254:/ks.cfg"
```

Note:
====

```
--vnc  "For enabling  VNC server"
--vncport   "deciding port number"
--vnclisten   "making server listen on any IP of IAAS Cloud Server"
```

Important:
==========

For accessing VM's from Client Side we can use many protocol's there are two way of accessing VM's

i) VIA GUI:

We need to user VNC,RDP,XRDP,SPICE,TEAM_VEIWER etc.

ii) VIA Command Line

Note: This is only for Linux/Unix based Operating system

Important:
==========

Live VM Mirgration
==========

This is the most advanced features IAAS cloud because If your One VM is needed to shift / Migrate from one IAAS cloud plateform to another Cloud Plateform then you can Migrate your Live OS without any down time upon all the services that you are running under VM

How to Migrate VM from One system to another System
==========

When you install OS inside KVM there are two places you need to remember

i) **XML format file of that os**
==========

To check this info:
==========

```
[root@desktop67 ~]# cd /etc/libvirt/qemu
[root@desktop67 qemu]# ls
autostart      networks   redhat6.xml
```

ii) documentroot where .img file of VM is stored

To check that info:
==========

```
[root@desktop67 qemu]# cd /var/lib/libvirt/images/
[root@desktop67 images]# ls
redhat6.img
```

Now SETUP LIVE MIGRATIOn :

System 1: (From this system 1 we need to migrate VM to system 2)
==========

Step 1: check that vm is running
=======

IP address: 192.168.0.67
running VM : 1 (redhat 6)

Note: you can check using below command
=====

```
[root@desktop67 ~]# virsh list
 Id    Name                           State
----------------------------------------------------
 1     redhat6                    running
```

Step 2: share .img file with NFS to system 2
=======

```
[root@desktop67 ~]#yum install nfs-utils
```

Export file should be look like this

```
[root@desktop67 ~]#cat /etc/exports
/var/lib/libvirt/images/           *(rw,no_root_squash)

[root@desktop67 ~]# chmod o+w   /var/lib/libvirt/images/redhat6.img

[root@desktop67 ~]# service nfs  restart
[root@desktop67 ~]# setenforce 0
[root@desktop67 ~]# iptables -F
```

System 2:
=======

Now go to system 2 and mount the shared directory on the same location

IP address: 192.168.0.83
==========

```
[root@desktop83 ~]# showmount -e  192.168.0.67
Export list for 192.168.0.67:
/var/lib/libvirt/images *
```

Mounting share directory on the same place of your system
==========

```
[root@desktop67 ~]# mount   192.168.0.67:/var/lib/libvirt/images/
/var/lib/libvirt/images/
```

Now check for VM in system 2:

```
[root@desktop67 ~]# virsh list
 Id    Name                           State
----------------------------------------------------
```

Note: as we can check there is no VM
=====

Final Point:
==========

Go to system and Migrate VM from one system to another

[root@desktop67 images]# **virsh migrate --live redhat6 qemu+ssh://192.168.0.80/system**

Note: it will ask for the root password of system 2 type it now it will take 1 mint to migrate in system now go to system 2 and check

[root@desktop80 ~]# **virsh list**

```
 Id    Name                           State
----------------------------------------------------
 1     redhat6                    running
```

Enjoy the CLoud Migration: