UC Cloud Summit 2011
UCLA
April 19,2011

Birds-of-a-Feather Session

Deploying private clouds as IaaS using Xen, KVM, Eucalyptus, Nimbus, OpenNebula and deploying images on Amazon EC2

Prakashan Korambath, UCLA

Running Hadoop on Hoffman2 cluster at UCLA

http://www.ats.ucla.edu/clusters/hoffman2/hadoop/default.htm

export HADOOP_HOME=$HOME/hadoop
export JAVA_HOME=/u/local/apps/java/jdk1.6.0_20
export PATH=$JAVA_HOME/bin:$HADOOP_HOME/bin:$PATH

Modify masters, slaves, core-site.xml, mapred-site.xml, hdfs-site.xml files
in $HADOOP_HOME/conf directory
mkdir input
cp conf/*.xml input
$HADOOP_HOME/bin/hadoop namenode -format
$HADOOP_HOME/bin/start-all.sh
netstat -plten | grep java
jps
$HADOOP_HOME/bin/hadoop fs -put conf input
$HADOOP_HOME/bin/hadoop jar hadoop-*-examples*.jar grep input output 'dfs[a-z.]+'
$HADOOP_HOME/bin/hadoop fs -get output output
cat output/*
$HADOOP_HOME/bin/stop-all.sh

Submitting Batch jobs to SGE

```
#!/bin/sh
#$ -o out.$JOB_ID
#$ -j y
#$ -cwd
#$ -l num_proc=8,hadoop
cd $HADOOP_HOME
$HADOOP_HOME/bin/hadoop namenode –format
$HADOOP_HOME/bin/start-dfs.sh
$HADOOP_HOME/bin/start-mapred.sh
$HADOOP_HOME/bin/hadoop fs -put $HADOOP_HOME/conf input
$HADOOP_HOME/bin/hadoop jar hadoop-0.20.2-examples.jar  grep input output 'dfs[a-z.]+’
$HADOOP_HOME/bin/hadoop fs -get output $HADOOP_HOME/output
$HADOOP_HOME/bin/stop-mapred.sh
$HADOOP_HOME/bin/stop-dfs.sh
```

qsub -v HADOOP_HOME=$HOME/hadoop-0.20.2 hdfs-sge.cmd

Prologs and Epilogs for Hadoop.q in SGE

prolog: hdfsstart.sh

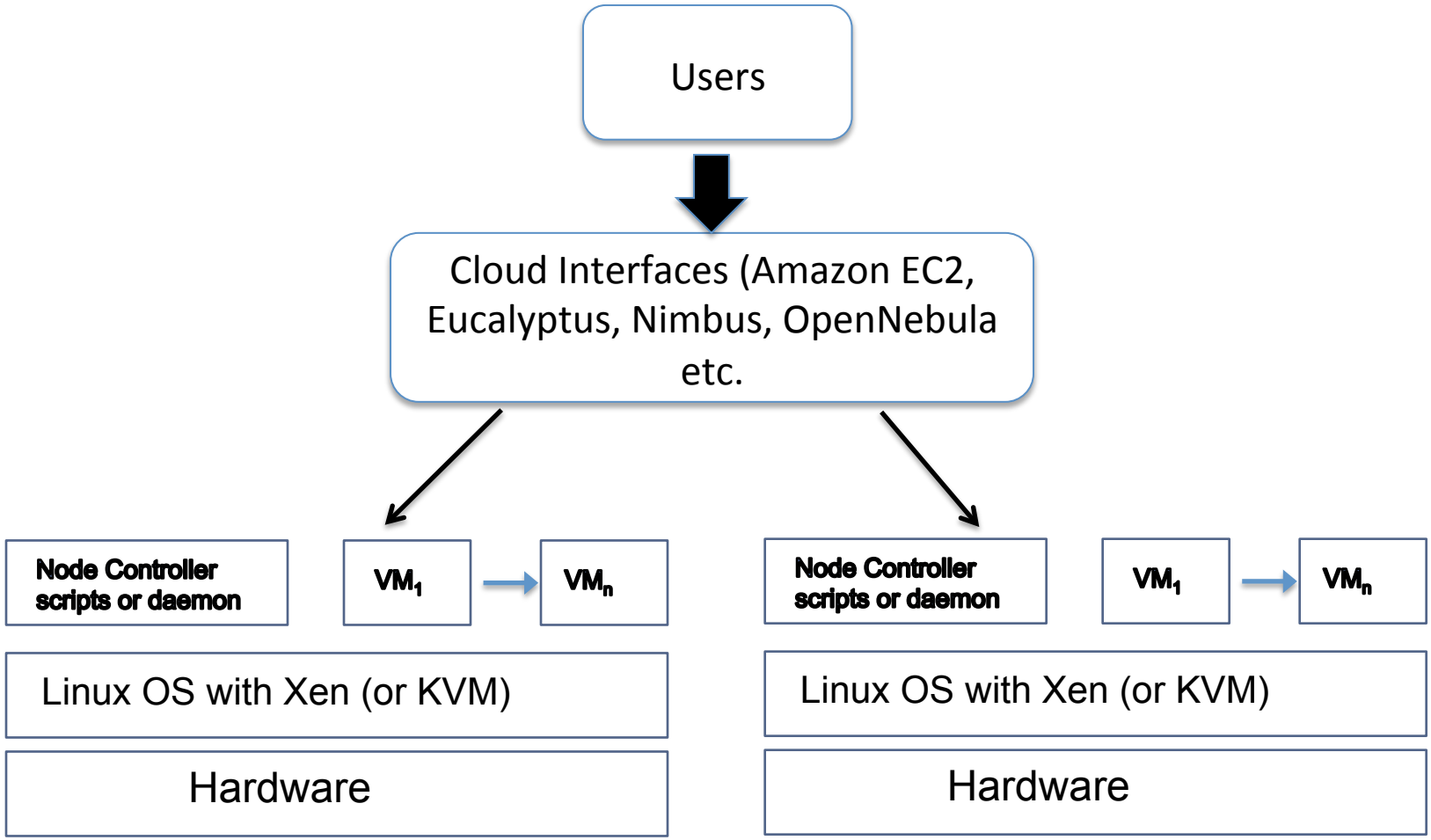
hdfsprolog.py $PE_HOSTFILE $TMPDIR $NHOSTS $HADOOP_HOME
http://www.ucgrid.org/hadoop/hdfsprolog.py

coresitefile=HADOOP_HOME+"/conf/core-site.xml”
mapredfile=HADOOP_HOME+"/conf/mapred-site.xml”
hdfssitefile=HADOOP_HOME+"/conf/hdfs-site.xml”
mastersfile=HADOOP_HOME+"/conf/masters”
slavesfile=HADOOP_HOME+"/conf/slaves"

epilog: hdfsstop.sh

$HADOOP_HOME/bin/stop-all.sh
rm -rf $TMPDIR

IaaS Overview



Deploying virtual OS using Xen

Requirements

- Xen Kernel-xen, xen (yum install)
- libvirt

Commands

- brctl show
- xm list
- xm console <#>
- xm create -c <configfile>
- xm help

Image creation process

- mkinitrd --omit-scsi-modules --with=xennet --with=xenblk --preload=xenblk /root/xen-image/centos-ramdisk.img `uname -r`
- dd if=/dev/zero of=centos-root.img bs=1M count=1999
- mount -o loop centos-root.img /root/xen-image/rootdisk/
- Create yum.conf file similar to the one you can find in a CentOS machine.
- yum –c yum-xen.conf --installroot=/root/xen-image/rootdisk -y groupinstall base
- chroot /root/xen-image/rootdisk
- Edit for passwd or any other files and unmount rootdisk

Deploying virtual OS using Xen – contd.

http://www.ucgrid.org/xen/uciaasimagecreate.py

Config file

```
kernel = "/boot/vmlinuz-2.6.18-128.1.14.el5xen"
ramdisk = "/root/xen-image/centos-ramdisk.img"
name = "centos53"
memory = "256"
disk = [ 'file:/root/xen-image/centos-root.img,sda1,w' ]
root = '/dev/sda1 ro'
vif = [ 'bridge=xenbr0', '']
vcpus=1
on_reboot = 'destroy'
on_crash = 'destroy'
```

- /usr/sbin/xm create -c centos53

Deploying Virtual OS using KVM

- egrep '(vmx|svm)' --color=always /proc/cpuinfo
- yum install kvm qemu libvirt python-virtinst qemu-kvm bridge-utils
- /etc/init.d/libvirtd start; /etc/init.d/network restart
- virsh -c qemu:///system list
- yum install virt-manager qemu-system-x86_64
- Create bridge network so that image can interact with outside network
  - cp ifcfg-eth0 ifcfg-br0 and change device to br0
  - Edit ifcfg-eth0 and add bridge=br0 and comment out IP/NM etc.

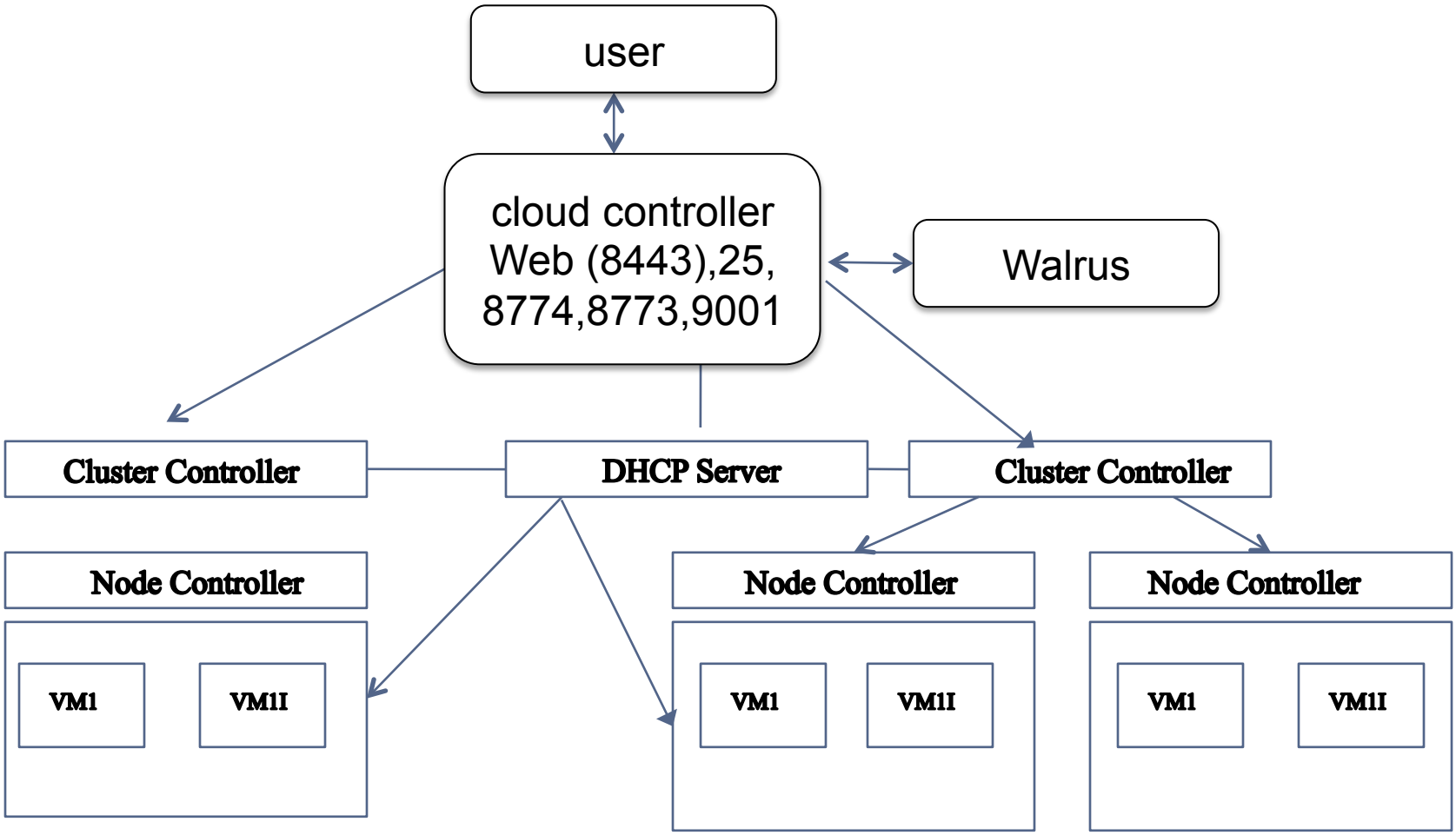
virt-install --connect qemu:///system -n vmwin7 -r 1024 --vcpus=2
–f /srv/vm/vmwin7.qcow2 -s 32 -c
/srv/vm/Win_Pro_7.ISO
--vnc --noautoconsole --os-type windows --os-variant win7
--accelerate --network=bridge:br0 --hvm

Run virt-manager to configure the image.

- virsh
- rdesktop ipaddress

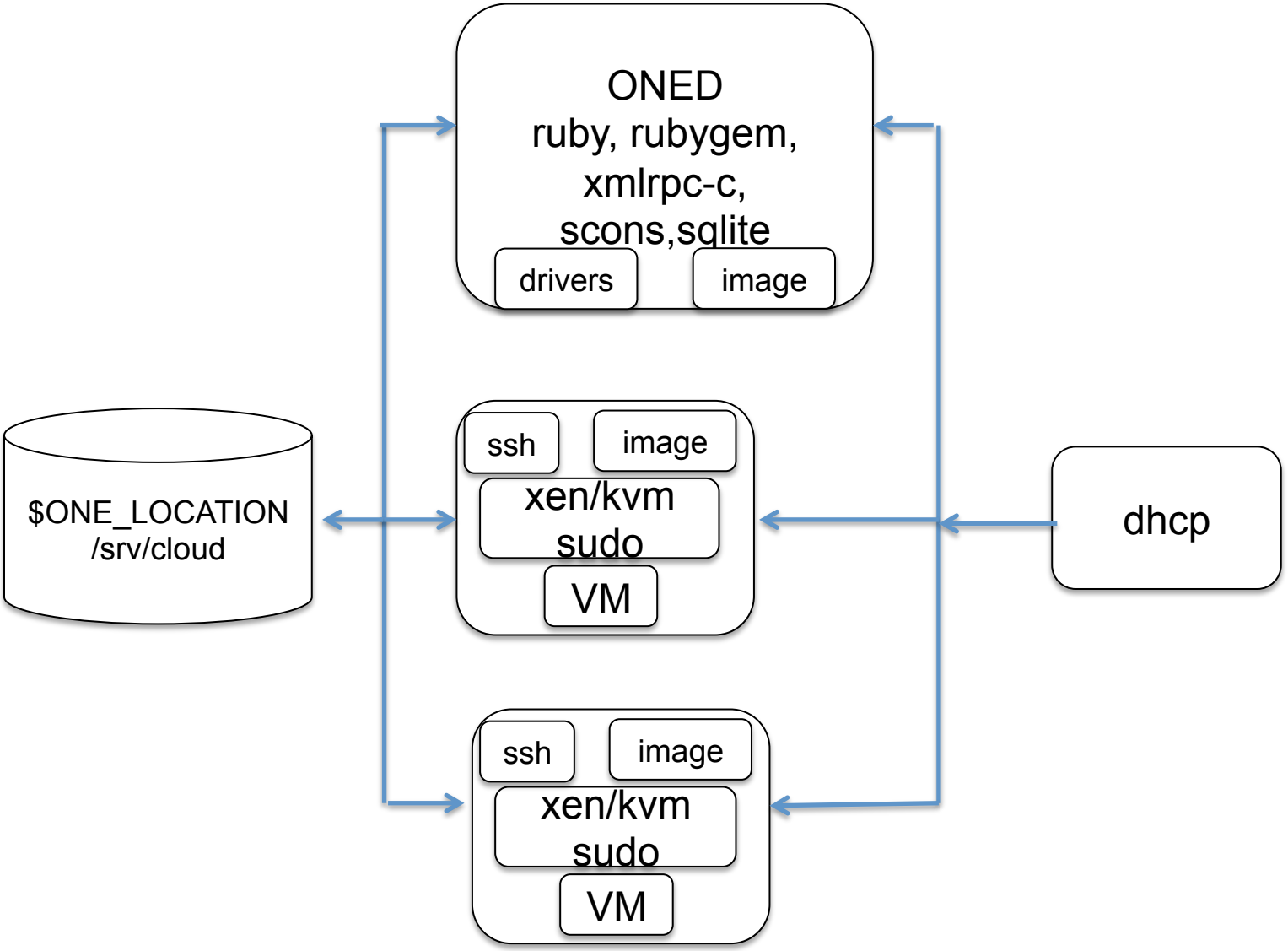
Eucalyptus Architecture

http://www.ucgrid.org/iaas/euca.html



Open Nebula Architecture

http://www.ucgrid.org/iaas/opennebula.html



ONE Commands

one start

onehost create host01 im_xen vmm_xen tm_nfs
onehost list
onehost show
onevnet create centos_network.net
onevnet list
onevm create centos.one
onevm list
onevm show Centos5
oneimage list
onevm delete 0

Contextualization

```
NAME   = Cs5
CPU    = 4.0
MEMORY = 1024
OS = [
  kernel     = "/srv/cloud/images/centos/vmlinuz-2.6.18-194.32.1.el5xen",
  initrd     = "/srv/cloud/images/centos/centos-ramdisk.img",
  root       = "sda1" ]
DISK   = [
  source   = "/srv/cloud/images/centos/centos-root.img",
  target   = "sda1",
  readonly = "no" ]
NIC = [ BRIDGE = "xenbr0", MAC = "A2:AA:BB:5B:7E:FD" ]
CONTEXT = [
    hostname    = "$NAME",
    ip_public   = "192.168.0.2",
    files       = "/home/gstuser/centos/init.sh /home/gstuser/centos/id_rsa.pub",
    target      = "sdc",
    root_pubkey = "id_rsa.pub",
    username    = ”gstuser",
    user_pubkey = "id_rsa.pub"
]
```

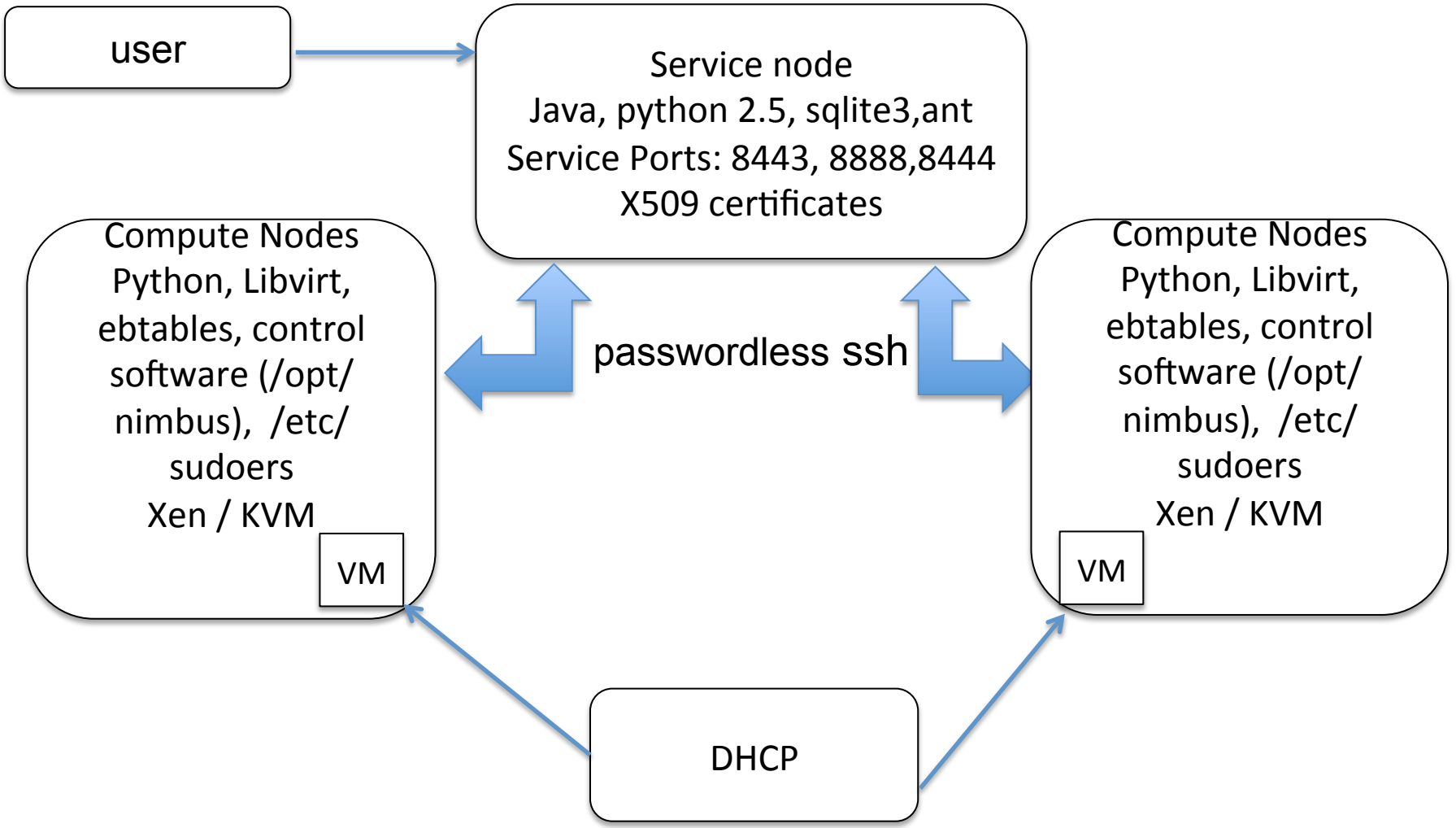
Initiation

```
mount -t iso9660 /dev/sdc /mnt
if [ -f /mnt/context.sh ]; then
  . /mnt/init.sh
fi
umount /mnt
exit 0
```

```
/bin/sh -x
if [ -f /mnt/context.sh ]; then
  . /mnt/context.sh
fi
hostname $HOSTNAME
ifconfig eth0 $IP_PUBLIC
useradd -m $USERNAME
mkdir -p /home/$USERNAME/.ssh
cat /mnt/id_rsa.pub >> /home/$USERNAME/.ssh/
authorized_keys
chown -R $USERNAME /home/$USERNAME
```

Nimbus Architecture

http://www.ucgrid.org/iaas/nimbus.html



Amazon EC2

- Educational grant
- Either use the images that are provided by Amazon or use your own image
- yum install process
- Download ec2-modules file and copy it to your image
- Copy the getssh file into /etc/init.d and chkconfig
- Make sure you have your ec2 certificates and keys
- ec2-bundle-image --image –prefix --privatekey –user –destination --arch
- ec2-upload-bundle --bucket --manifest --accesskey --secretkey
- ec2-register MANIFEST
- ec2-run-instances
- ec2-describe-instances