A technical whitepaper describing steps to setup a Private Cloud using the Eucalyptus Private Cloud Software and Xen hypervisor.

Vivek Juneja
Cloud Computing COE
Torry Harris Business Solutions
INDIA
Contents

Introduction to Private Cloud, Eucalyptus and Xen 3
Setting up Eucalyptus Cloud on Xen 4
Hardware and Software 5
Eucalyptus Front End 6
Eucalyptus Node Controller 7
Registering Node Controller on Front End 11
Adding VM Images 12
Configure Eucalyptus Tools 15
References 19
**What is Private Cloud?**

A System in which Infrastructure is dedicatedly allocated to an individual or an enterprise and provides cloud based services: Dynamic Provision, Elastic Resources, Chargeback and Automation. A Private cloud provides shared hosting environment for multiple applications and environments owned or managed by an Enterprise or individual. It’s a preferred platform for most enterprises who fear of security, compliance regulations and ownership of data problems with Public cloud and yet need the elastic power of computing to be available on-demand.

**What is Eucalyptus?**

Eucalyptus is an infrastructure software that enables enterprises and government agencies to establish their own cloud computing environments. With Eucalyptus, customers make more efficient use of their computing capacity, thus increasing productivity and innovation, deploying new applications faster, and protecting sensitive data, while reducing capital expenditure. It provides APIs compatible to the popular Amazon Web Services offerings: EC2, S3 and EBS, thus allowing access to wide variety of cloud tools and option of building hybrid clouds.

**What is Xen?**

Xen is the most popular Open Source Virtualization software that allows multiple OS to run on the same computer hardware concurrently, thereby improving the effective usage and efficiency of the underlying hardware. It benefits the enterprises with the power of consolidation, increased utilization and rapid provisioning.
Setting up Eucalyptus Cloud on Xen :-

In any Eucalyptus Cloud Installation, there are 2 top-level components: *Cloud Controller* (CLC) and *Walrus*. These 2 components manage the various clusters, where cluster is a set of physical machines that host the Virtual Instances. In each cluster, there are 2 components that interact with the high level components: *Cluster Controller* (CC) and *Storage Controller* (SC). CC and SC are cluster-level components. Each cluster is composed of various Nodes, or physical machines. Each Node will run a *Node Controller* (NC) that will control the hypervisor for managing the Virtual Instances.

For this setup, We have implemented a *Single-Cluster Installation*, where all the components except NC are co-located on one machine. As per Eucalyptus documentation, this co-located system is called: front-end. So in a gist, We have 1 physical machine which hosts CLC, Walrus, CC, SC, and 5 other machines that hosts NC each.

The Node Controller uses Xen as a hypervisor. The NC service runs on Domain-0 kernel in the Xen Setup.
Hardware :-

We used 1 Admin Machine with the config: *Intel Core 2 Duo Processor 1.8 GHz, with 1 GB RAM, 160 GB HDD*, and 5 Node Controllers each with the config: *Intel Core 2 Duo Processor 1.8 Ghz (VT enabled) with 2 GB RAM, 160 GB HDD*.

The recommended configuration for the Admin and the Node Controller machine can be obtained from the following link :-

https://help.ubuntu.com/community/UEC/CDInstall

Software :-

*Eucalyptus 1.6.1* running on *Ubuntu 9.04 (Jaunty)*, *Xen 3.3* hypervisor.

For dependencies, in case of installing from source, please follow the link:- [http://open.eucalyptus.com/wiki/EucalyptusInstallationUbuntuJaunty_v1.6](http://open.eucalyptus.com/wiki/EucalyptusInstallationUbuntuJaunty_v1.6)

The above link mentions the dependencies for both Admin and Node Controller environments.
Eucalyptus Front End :-

The Eucalyptus Front End hosts the Cloud Controller, Storage Controller and the Cluster Controller services. It exposes AWS compatible WS (Web Services) interfaces.

Add the following line in /etc/apt/sources.list file,

```
deb http://www.eucalyptussoftware.com/downloads/repo/eucalyptus/1.6.1/ubuntu jaunty
```

Then update it, using

```
apt-get update
```

We have setup a static IP address for the Front-end, and for reference, I have the network interfaces setting as following :-

```
bash> cat /etc/network/interface

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
transition 192.168.1.26
netmask 255.255.255.0
network 192.168.1.0
```

Then, We installed Eucalyptus CLC, CC, Walrus, SC using the following :-

```
sudo apt-get install eucalyptus-cc eucalyptus-cloud eucalyptus-common eucalyptus-walrus eucalyptus-sc
```

Install the euca2ools for managing the Cloud :-

```
sudo apt-get install euca2ools
```
Eucalyptus Node Controller :-

A Node Controller will host the Virtual Machines. Hence it must have a hypervisor installed into it.

We first installed the Ubuntu 9.04 Server Edition on the bare machine. To this, we installed the Debian Domain 0 Kernel, Xen Hypervisor, and Xen utils.

Installing Debian Domain 0 Kernel :-
We were able to find the Debian Kernel for Domain 0 :-

Kernel:

Modules:

We installed it :-

```bash
```

Once the Domain 0 kernel is installed, the utility grub-update can automatically create the menu entry in the GRUB like as following :-

```
hash> cat /boot/grub/menu.lst

uuid        23c57099-3cee-4c9c-ae22-579b6128bac3
kernel      /vmlinuz-2.6.26-2-xen-686 root=/devmapper/ubuntu--cloud-root ro conso
module      /initrd.img-2.6.26-2-xen-686
quiet
```

Installing Xen hypervisor and tools:

```bash
sudo apt-get install xen-hypervisor-3.3 xen-utils-3.3
```

Now you should remove the Ubuntu Network Manager by :

```bash
sudo apt-get remove network-manager
```

update the /etc/network/interfaces with this :-

```
iface lo inet loopback
auto lo
iface eth0 inet dhcp
```

```
The above settings are for individuals who have a DHCP server running like mine, but in case you are on a fixed IP, then you can go for something like this below:-

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.1.26
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
```

Now boot the machine and select the Xen Domain 0 kernel from the GRUB.

Once the Xen Domain 0 kernel was installed, we rebooted the machine. On GRUB, we selected the new Dom 0 Kernel which got installed. In case the Xen Dom 0 kernel entry is not visible in the GRUB Menu, we need to boot into Ubuntu 9.04 and edit the menu.lst file to have the entry for the same. We then use grubupdate to update the GRUB which will take the new Menu listing. The last procedure differs in case you are using the GRUB 2. In this case then, we would need to change the grub.cfg file with the same changes.

Once we have logged in the Xen Dom 0 kernel from the GRUB, we need to install the Eucalyptus Node Controller component on it. But before that, it is essential that we test that we are able to create a new Xen Domain from the Xen tools at the command prompt.

Please install the xen tools to create a new domain :-

```
sudo apt-get install xen-tools
```

Now, you can edit the file `/etc/xend/xend-config.sxp` and check if you have the following settings :

```
(xend-http-server yes)
(xend-unix-server yes)
(xend-unix-path /var/lib/xend/xend-socket)
(xend-address localhost)
(network-script network-bridge)
(vif-script vif-bridge)
(dom0-min-mem 196)
(dom0-cpus 0)
```
We can follow the below mentioned link to test the Xen tools and create a new test domain. If the domain is created successfully, and we are able to do console login to the running domain, we can confirm that Xen is working fine.

Creating a Xen Domain U :-
http://www.infohit.net/blog/post/installing-and-running-xen-domu-jaunty-on-dom0-ubuntu-jaunty.html

After successful testing, stop the running domain, and remove it using the xen tools CLI.

We can then check for the bridge setup, by issuing the following command. The output is as shown :

```
bash> brctl show
<table>
<thead>
<tr>
<th>bridge name</th>
<th>bridge id</th>
<th>STP enabled</th>
<th>interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0</td>
<td>8000.001a6b4684e5</td>
<td>no</td>
<td>peth0</td>
</tr>
<tr>
<td>virbr0</td>
<td>8000.000000000000</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>
```

Please note : Against the bridge name eth0, the interface is peth0 in the above console output.

After this we must install the dependencies for the Eucalyptus Node Controller including Apache Axis2/c, Rampart, LibVirt etc. as per the documentation for the Eucalyptus Installation.

Once the dependencies are installed, we must install the eucalyptus-nc service. Once the eucalyptus-nc service is installed, we must edit the file eucalyptus.conf usually located under /etc/eucalyptus folder. This file holds all the critical settings for the eucalyptus-nc to work correctly. We must first change the hypervisor setting, and set it to xen instead of KVM which comes by default. Also, we must set the INSTANCE_PATH setting to the convenient path where we want to have the Xen Domain Images stored. This folder must be allocated space that makes it sufficient to contain large Virtual Machine images.

Next, we must configure the VNET_PRIVINTERFACE settings. In my environment, following was the assignment. Please refer to the section in this document where we mentioned about the xen bridge and interfaces. :-

```
VNET_PRIVINTERFACE="peth0"
```

The VNET_BRIDGE setting must be pointing to the name of the bridge that Xen has configured. In my case it was eth0.

In my environment, the need was to have Eucalyptus not to manage the VM network. This was because the network in which Eucalyptus was installed was not isolated and self-contained. So, the VNET_MODE setting in the same file was set to SYSTEM. For more extensive details on the Eucalyptus Network settings, please refer to the Eucalyptus Documentation.
So, now the Eucalyptus configuration for the Node Controller is done. We will now start the node controller service. In my environment, the Eucalyptus Node Controller was started using the command:

```
/etc/init.d/eucalyptus-nc restart
```

Please refer to the Eucalyptus Log files, usually located under `/var/log/eucalyptus/` directory. The log file is `nc.log`. Check this file if the eucalyptus-nc service shows any failure.

Remember as obvious, we must restart the eucalyptus-nc service on any changes done to the `eucalyptus.conf` file.
Registering Node Controller on Front End:

Now, we will start configuring the Cloud Controller, and allow it to register the new Node that we have setup. For this, we will login to the Eucalyptus Cloud Controller box, and use the euca_conf application to register the new Node.

Use the following command:

```
euca_conf --register-nodes "192.168.1.101"
```

The IP address mentioned in the above command refers to the machine that had the Eucalyptus Node Controller service running.

Now we need to obtain the credentials from the command line of the Cloud Controller by doing the following:

```
mkdir -p ~/.euca
chmod 700 ~/.euca
cd ~/.euca
sudo euca_conf --get-credentials mycreds.zip
unzip mycreds.zip
ln -s ~/.euca/eucarc ~/.eucarc
cd -
```

Eucalyptus Cloud Controller will attempt to register the new Node, and we can check for successful registration by following command:

```
. ~/.euca/eucarc
euca-describe-availability-zones verbose
```

The above command produces the following console output in my setup:

<table>
<thead>
<tr>
<th>AVAILABILITYZONE</th>
<th>vm types</th>
<th>free / max</th>
<th>cpu</th>
<th>ram</th>
<th>disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVAILABILITYZONE</td>
<td>m1.small</td>
<td>0002 / 0002</td>
<td>1</td>
<td>128</td>
<td>2</td>
</tr>
<tr>
<td>AVAILABILITYZONE</td>
<td>c1.medium</td>
<td>0002 / 0002</td>
<td>1</td>
<td>256</td>
<td>5</td>
</tr>
<tr>
<td>AVAILABILITYZONE</td>
<td>m1.large</td>
<td>0001 / 0001</td>
<td>1</td>
<td>512</td>
<td>20</td>
</tr>
<tr>
<td>AVAILABILITYZONE</td>
<td>m1.xlarge</td>
<td>0000 / 0000</td>
<td>2</td>
<td>1024</td>
<td>40</td>
</tr>
<tr>
<td>AVAILABILITYZONE</td>
<td>c1.xlarge</td>
<td>0000 / 0000</td>
<td>4</td>
<td>2048</td>
<td>60</td>
</tr>
</tbody>
</table>

If the above command works out properly, we are sure that Eucalyptus is working fine. Now we will proceed to run a new instance of the Cloud.
Adding VM Images :-

Adding a VM image to the Eucalyptus Cloud requires :-

a) Download a VM Image
b) Add the root disk image, kernel / ramdisk pair to the Walrus (Storage Service)
c) Register the image with Eucalyptus

First, we downloaded an image from http://uec-images.ubuntu.com/releases/ , which in this case is : http://uec-images.ubuntu.com/releases/9.10/rc/ubuntu-9.10-rc-uec-i386.tar.gz

We will now bundle the Kernel, Initrd and the OS Image :-

1) Unpack the Downloaded image from the tarball :

```bash
tar -S -xzf ubuntu-9.10-rc-uec-i386.tar.gz
```

2) We bundle the kernel

```bash
euca-bundle-image i karmic-uec-i386-vmlinuz-virtual --kernel
```

3) We upload the kernel bundle

```bash
TIMESTAMP=$(date +%Y%m%d%H%M%S)
BUCKET_KERNEL="k-$TIMESTAMP"
euca-upload-bundle -b $BUCKET_KERNEL -m /tmp/karmic-uec-i386-vmlinuz-virtual.manifest.xml

EKI=$(euca-register $BUCKET_KERNEL/$UEC_KERNEL.manifest.xml | grep "^IMAGE" | awk '{print $2}') && echo $EKI
```

4) Register the kernel bundle with Eucalyptus

5) We bundle the ramdisk

```bash
BUCKET_INITRD="r-$TIMESTAMP"
euca-bundle-image -i karmic-uec-i386-initrd-virtual -r i386
euca-upload-bundle -b $BUCKET_INITRD -m /tmp/karmic-uec-i386-initrd-virtual.manifest.xml

ERI=$(euca-register $BUCKET_INITRD/karmic-uec-i386-initrd-virtual.manifest.xml| grep "^IMAGE" | awk '{print $2}') && echo $ERI
```

6) We upload the ramdisk bundle

7) Register the ramdisk bundle with Eucalyptus
8) We bundle the image

```bash
BUCKET_IMAGE="i-$TIMESTAMP"
euca-bundle-image -i karmic-uec-i386.img -r i386 --kernel $EKI $[ERI:+--ramdisk $ERI]
```

The above step will take a little time, depending on the size of the image.

9) Upload the image bundle

```bash
BUCKET_IMAGE="i-$TIMESTAMP"
euca-upload-bundle -b $BUCKET_IMAGE -m /tmp/$UEC_IMG.img.manifest.xml
```

10) We register the image with Eucalyptus

```bash
EMI=$(euca-register $BUCKET_IMAGE/karmic-uec-i386.img.manifest.xml | grep "^IMAGE" | awk '{print $2}') && echo $EMI
```

Once the above steps have been performed, the image will be ready to run.

Execute the following command to check if the image is available :-

```bash
bash>
euca-describe-image
```

```
<table>
<thead>
<tr>
<th>IMAGE</th>
<th>emi-242F1C0</th>
<th>1-20100322194920/karmic-uec-i386.img.manifest.xml</th>
<th>admin</th>
<th>available</th>
<th>public</th>
<th>i386</th>
<th>machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAGE</td>
<td>eki-41A11D5</td>
<td>k-20100322194920/karmic-uec-i386-vmlinuz-virtual.manifest.xml</td>
<td>admin</td>
<td>available</td>
<td>public</td>
<td>i386</td>
<td>kernel</td>
</tr>
<tr>
<td>IMAGE</td>
<td>eri-24EA1645</td>
<td>r-20100322194920/karmic-uec-i386-initrd-virtual.manifest.xml</td>
<td>admin</td>
<td>available</td>
<td>public</td>
<td>i386</td>
<td>ramdisk</td>
</tr>
</tbody>
</table>
```

Now, we need to create the keypair (ssh key) that can be used to log in to the new VM instance.

```bash
if [ ! -e ~/.euca/cloud.priv ];
then mkdir -p ~/.euca/cloud.priv
chmod 0600 ~/.euca/cloud.priv
fi
```

```bash
euca-add-keypair cloud > ~/.euca/cloud.priv
```

In the above command, `cloud` is the name of the keypair.

We must also allow access to port 22 for FTP to the VM Instances :

```bash
euca-authorize default -P tcp -p 22 -s 0.0.0.0/0
```

Now, we create instance of the image that we registered in the above instructions

```bash
euca-run-instances $EMI -k mykey -t m1.small
```
It will take sometime for the instance to be ready, and we can use the below command to monitor the state of the instance:

```
euca-describe-instances 
```

Once the above command shows the image to be in running state, then we need to check for the IP Address of the new Instance:

```
IPADDR=$(euca-describe-instances | grep $EMI | grep running | tail -n1 | awk
```

Once the IP address is available, we can access the running instance, by using the below command:

```
ssh -i ~/.euca/mykey.priv ubuntu@$IPADDR
```
Configure Eucalyptus Tools :-

One of the popular tools to manage both Amazon and Eucalyptus EC2 instances is HybridFox. It is a Mozilla Firefox Plugin, and integrates well with the Eucalyptus Cloud. It allows to manage the EC2 instances and EBS Volumes. The user can create / stop/ start instances, attach EBS volumes and even take EBS snapshots. We have covered HybridFox in brief through the series of screenshots.

We can download HybridFox from http://code.google.com/p/hybridfox/downloads/list
To install, just drag and drop the HybridFox.xpi file into Mozilla Firefox browser. Once installed, the plugin can be accessed from the Tools menu of the browser.

We will now cover the configuration part of the HybridFox. The user must click on the Regions button as shown below, and enter the Region Name and EndPoint URL. The EndPoint URL must point to the IP address of the Cloud Controller. Region Name can be anything that the user likes.

Once added, we must select the added region from the Regions Drop down.

Now we need to get the credentials that are required for HybridFox to make secure Web Service calls to the Cloud Controller. For retrieving the Credentials, we must log in to the Eucalyptus Admin Tool as shown in the next screenshot.
We will use the Query ID and Secret Key available from the Credentials Tab of the Admin Console.

The User must now get back to HybridFox, and click on Credentials button. The Query ID obtained from above will be the AWS Access Key, and the Secret Key will be the AWS Secret Access Key. The account name can be anything that the user wants.
Once the credentials are added, the user must select the added credentials from drop down, and hit refresh for the browser. This will allow HybridFox to access the Cloud Controller with the config.

The screenshot below shows the list of Images available. The user can select any EMI and launch an instance for the same. Subsequent screenshot shows the running instance.
The following screenshot shows the security groups created, and the permission assigned to the groups.

EBS Volumes and Snapshots management is shown below.
References :-

http://www.ubuntu.com/cloud/private
http://www.infohit.net/blog/post/running-xen-on-ubuntu-intrepid-and-jaunty.html
http://xen.org/