Building a Private Cloud
Cloud Infrastructure Using Opensource
Building a Private Cloud with Ubuntu Server 10.04 Enterprise Cloud (Eucalyptus)

OSCON 2010
(Note: Special thanks to Jim Beasley, my lead Cloud Ninja, for putting this document together!)

Introduction
In this demonstration we will show the steps required to build a private enterprise cloud. After the cloud has been built we will show how to manage images, security groups, monitor resources and deploy instances within the private cloud. We chose Ubuntu for this demonstration as it facilitates a rapid deployment of open source Amazon EC2 clone Eucalyptus.

Preparation
For our installation we will be using two servers, one as a cloud controller and the other as a cloud node. Cloud instances will be running on the node, so unless the systems are identical we will choose the system with more CPU cores and memory as our node controller. This allows us more room for growth in the cloud as we add instances.

We will be using the default network configuration of “Managed-NoVLAN” which provides dynamic IP assignment for VMs and allows us to control ingress traffic by building iptables profiles known as security groups. Note that another mode of network configuration known as “Managed” mode provides the additional feature of VM network isolation.

Additional Areas of Interest
Several areas we may explore if time permits include tweaking the /etc/eucalyptus/euca.conf file to multiplex several VMs per core, adding additional nodes to the cluster, VM to VM network access and/or isolation, custom image creation, etc.
Getting Started – Building the Cloud Controller

First we will build our cloud controller by booting from the Ubuntu 10.04 Server cdrom and selecting “Install Ubuntu Enterprise Cloud” from the menu:
After making the appropriate language, country and keyboard selections, we will be prompted to configure the network. For this lab we will be using eth0 for both the cloud and node controllers.
Next we will assign a hostname. We have chosen the name “cc” for our cloud controller:
Because we don’t already have a cloud controller installed on this network, we’ll select continue at this screen:

![Image showing a screen with instructions to select cloud installation mode.](image-url)
Here we must choose the role this server will play in our cloud. In larger and/or more complex installations, each of the functions shown in this menu may be divided onto separate physical servers. Eucalyptus private clouds will have a single cloud controller, but there may be multiple cluster controllers within the cloud, and multiple node controllers reporting to each cluster controller. Walrus is the data storage component of Eucalyptus, which is similar to Amazon’s Simple Storage Service (S3). For our simplified demonstration, we will use a single-cluster installation and accept the default cloud installation mode of Cloud controller, Walrus storage service, cluster controller, and storage controller.
In our lab we will use the eth0 interface to connect to the public network and to communicate with the node.
The next several screens show us accepting the default proposal for partitioning the disks:

The installer can guide you through partitioning a disk (using different standard schemes) or, if you prefer, you can do it manually. With guided partitioning you will still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.

Partitioning method:

- Guided - use entire disk
- Guided - use entire disk and set up LVM (Red)
- Guided - use entire disk and set up encrypted LVM
- Manual

<Go Back>

<Tab> moves; <Space> selects; <Enter> activates buttons
We have one disk to present to the Ubuntu installer, which is a RAID 1+0 array we built using the HP smart array bios:

![Partition disks](image)
If existing data is detected, you will notice a screen similar to this:

The selected device already contains the following LVM logical volumes, volume groups and physical volumes which are about to be removed:

Logical volume(s) to be removed: root, swap_1
Volume group(s) to be removed: cc
Physical volume(s) to be removed: /dev/cciss/c0d0p5

Note that this will also permanently erase any data currently on the logical volumes.

Remove existing logical volume data?

<Go Back>  <Yes>  <No>

<Tab> moves; <Space> selects; <Enter> activates buttons
Here we will accept the default partitioning and configure Linux Volume Manager (LVM):

![Partitioning and Configuration](image)

Before the Logical Volume Manager can be configured, the current partitioning scheme has to be written to disk. These changes cannot be undone.

After the Logical Volume Manager is configured, no additional changes to the partitioning scheme of disks containing physical volumes are allowed during the installation. Please decide if you are satisfied with the current partitioning scheme before continuing.

The partition tables of the following devices are changed:
SCSI.CCISS (-,0,0) (cciss/c0d0)

The following partitions are going to be formatted:
partition #1 of SCSI.CCISS (-,0,0) (cciss/c0d0) as ext2

Write the changes to disks and configure LVM?

<Yes> <No>

<Tab> moves; <Space> selects; <Enter> activates buttons
In our Lab we will keep it simple and use the entire volume group for guided partitioning:

The minimum size of the selected partitioning recipe is 596.0 MB (or 0%); please note that the packages you choose to install may require more space than this. The maximum available size is 146.5 GB.

Hint: "max" can be used as a shortcut to specify the maximum size, or enter a percentage (e.g. "200") to use that percentage of the maximum size.

Amount of volume group to use for guided partitioning:

146.5 GB

<Go Back> <Continue>

<Tab> moves; <Space> selects; <Enter> activates buttons
Here we will write the new changes to disk:

If you continue, the changes listed below will be written to the disks. Otherwise, you will be able to make further changes manually.

WARNING: This will destroy all data on any partitions you have removed as well as on the partitions that are going to be formatted.

The partition tables of the following devices are changed:
- LVM VG cc, LV root
- LVM VG cc, LV swap_1

The following partitions are going to be formatted:
- partition #1 of SCSI.CC1SS (-,0,0) (cciss/c0d0) as ext2
- LVM VG cc, LV root as ext4
- LVM VG cc, LV swap_1 as swap

Write the changes to disks?

<Yes>  <No>

<Tab> moves; <Space> selects; <Enter> activates buttons
Next, the installer will format the partitions and install the base system:
Here we are prompted to create a user account which will have sudo privileges. We chose “cladmin” as our username, with a password of “cloud9”: 
Although we aren’t using automatic updates in our demonstration, it is recommended:

Applying updates on a frequent basis is an important part of keeping your system secure.

By default, updates need to be applied manually using package management tools. Alternatively, you can choose to have this system automatically download and install security updates, or you can choose to manage this system over the web as part of a group of systems using Canonical’s Landscape service.

How do you want to manage upgrades on this system?

- No automatic updates
- Install security updates automatically
- Manage system with Landscape

<Tab> moves; <Space> selects; <Enter> activates buttons
After answering several email-related questions, we configure the name of the cluster:

![Image of configuring eucalyptus-cc]

Enter a name for this cluster. The name should contain only ASCII letters, digits, hyphens, and underscores. It will be shown to users as the name of an availability zone.

Eucalyptus cluster name:

(cluster1)

<Continue>
Here we provide a pool of addresses that will be automatically assigned to VMs as they are instantiated. These addresses will be automatically assigned to cloud instances to make them accessible from outside the cloud:

Eucalyptus requires a pool of IP addresses that can be dynamically assigned as the "public" IPs of virtual machines. These IPs must be unused within their Class C subnet, this system must have an interface configured with an address on this subnet, and your prospective users must be able to connect to these IPs from wherever they run the client tools.

Please specify one or more ranges of IP addresses, e.g.:
192.168.1.100-192.168.1.199
or
192.168.2.50-192.168.2.99 192.168.2.150-192.168.2.199

You may leave this blank if you have no IP addresses available, BUT you and your users must then request the private addressing scheme when starting a virtual machine instance. For ec2-run-instances and euca-run-instances, this is done with the option "--addressing private".

144.60.26.100-144.60.26.199

<Continue>
Next we install the Grand Unified Boot loader, GRUB:

It seems that this new installation is the only operating system on this computer. If so, it should be safe to install the GRUB boot loader to the master boot record of your first hard drive.

Warning: If the installer failed to detect another operating system that is present on your computer, modifying the master boot record will make that operating system temporarily unbootable, though GRUB can be manually configured later to boot it.

Install the GRUB boot loader to the master boot record?

<Go Back>  <Yes>  <No>

<Tab> moves; <Space> selects; <Enter> activates buttons
This completes the initial installation of the cloud controller.
Building the Node Controller

Now that our cloud controller (and cluster controller, walrus, storage controller) has been built, we will move on to the next server. To begin build our node controller we will boot from the Ubuntu 10.04 Server cdrom and select “Install Ubuntu Enterprise Cloud” from the menu:
After making the appropriate language, country, keyboard selections and network interface, we will be prompted for the hostname. We entered “nc” as the hostname of our node controller.

![Configure the network screen](image)

'hostname' is a single word that identifies your system to the network. If you don’t know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

nc

<Go Back>  <Continue>

Tab moves; Space selects; Enter activates buttons
The installer will detect the cluster controller already running on our network, and default to a cloud installation mode of “Node Controller” which we will accept:
After selecting the cloud installation mode, you might see a screen similar to this one if there is more than one cluster controller on the subnet:

A word of caution: we ran into some issues when installing more than one cloud on the same subnet, so beware!
The next several installation screens will present us with disk partitioning options, and we will use the same settings that were used for the cloud controller, then the installation will finish the node will be rebooted.

Now that our cloud controller and node controller have been installed, we are ready to configure administrative access to the cloud.

Please note that from here on, we may use the hostnames “cc” and “nc” in commands. If DNS is not configured on your network, you will need to specify the IP address instead of the hostname.

**Configuring Access for the Eucalyptus User**

NOTE: These steps are not needed if the node controller detected the cloud controller during installation.

**Step 1:**

Here we will set a temporary password for the eucalyptus account.

Login to the node controller as user “cladmin” password “cloud9”:

```
cladmin@nc:~$ sudo passwd eucalyptus
Type “cloud9” for the temporary password.
```

**Step 2:**

Here we will login to the cloud controller and copy the ssh public key for the eucalyptus user to the node controller:

```
cadmin@cc:~$ sudo -u eucalyptus ssh-copy-id -i ~eucalyptus/.ssh/id_rsa.pub
eucalyptus@nc
```

**Step 3:**

Now, from our node controller we’ll remove the temporary password:

```
cadmin@nc:~$ sudo passwd -d eucalyptus
```
Installing Cloud Administrative Credentials through the Eucalyptus Web Interface

Before we can use the Amazon EC2 command-line utilities to interact with the cloud, we will need to install credentials which consist of x.509 certificates and environment variables.

Step 1:

Browse to the URL https://cc:8443

Login with the default username and password of admin, admin.
Step 2:

Set a new password for the admin account and supply an email address. The cloud host IP is automatically filled in and is the public facing IP for the cloud controller:

![First-time Configuration Form](image)

**Administrator's new password:**

**The password, again:**

**Administrator's email address:** cladmin@labs.att.com

User signup requests will be sent to this address and messages from Eucalyptus will have this address in the **From:** field. (If you want to change this behavior, edit the appropriate values in the eucalyptus-web.properties file of your Eucalyptus installation. We still need an email address now, though.)

**Cloud Host:** 144.60.26.78

We made a guess about the external IP of the machine running the Cloud Controller. Please, make sure that it is correct, as this IP will be embedded in the credentials generated by Eucalyptus. Although it can be changed later, doing so will require that you perform extra tasks and may cause existing users from being unable to access the system.
Step 3:

Now we will download our credentials. The web front end of Eucalyptus is currently limited, so after the initial configuration much of the administration will be done from the command line using the Amazon EC2 tools. On Ubuntu the name of the package is “euca2ools” and is conveniently installed by default on our cloud controller, so we’ll be using the cloud controller as our command-line headquarters for managing the cloud later in this guide.

To download credentials, click the “Credentials” tab and click “Download Credentials”:

![Credentials Page]

**User account Information**

Login: admin  
Name:  
Email: cladmin@labs.att.com

Feel free to change the account information (except the login) and the password whenever you want. The cryptographic credentials for the Web services associated with this account, shown below, will not be affected by these changes.

[Edit Account Information]  
[Change Password]

**Credentials ZIP-file**

Click the button to download a ZIP file with your Eucalyptus credentials. Use the public/private key pair included therein with tools that require X.509 certificates, such as Amazon’s EC2 command-line tools.

[Download Credentials]
Step 4:

Copy the downloaded file euca2-admin-x509.zip to /home/cladmin folder on the cloud controller. You can use scp, ftp, sftp, or any other preferred method.

Step 5:

Now we will create a hidden folder on the cloud controller and extract the zip file to this folder:

```
claadmin@cc:~$ mkdir ~/.euca
claadmin@cc:~$ cd ~/.euca
claadmin@cc:~/.euca$ unzip ../euca2-admin-x509.zip
```

Step 6:

Because the credentials file contains information allowing administrative access to the cloud, it is recommended to remove the zip file and apply permissions to the .euca folder and its contents:

```
claadmin@cc:~/.euca$ rm ~/euca2-admin-x509.zip
claadmin@cc:~/.euca$ chmod 0700 ~/.euca
claadmin@cc:~/.euca$ chmod 0600 ~/.euca/
```

Step 7:

Next we will add a line to the ~/.bashrc file on the cloud controller to ensure the necessary environment variables are initialized upon login:

```
claadmin@cc:~/.euca$ echo ".
```

Step 8:

Next we will source the .bashrc file to ensure our settings take effect:

```
claadmin@cc:~/.euca$ source ~/.bashrc
```

You can log off and back on in order to ensure these settings are active.
Installing Cloud Images

The images tab will list any images that have been registered with the cloud. Each instance or VM running in the cloud is based on an image. No images exist by default after installation, so we’ll need to install them.

No images found
Step 1:

While it is possible to build custom images and bundle, upload and register them with the cloud, for the sake of time we will install an image from Canonical’s online cloud image store.

Clicking the “Store” tab in the web interface will show us the images that are available from Canonical over the internet. For our lab we will install the MediaWiki Demo Appliance image, which after downloading the image from Canonical it will be installed to the cloud:
Step 2:

After the image has been installed, we can click on the images tab to confirm it has been registered with the cloud:

![Image of Ubuntu Enterprise Cloud interface](image)

Make a note of the emi-xxxxxx under the Id column as it will be the identifier we use to run an instance. An emi file is the Eucalyptus equivalent of an Amazon Machine Image (AMI) file from Amazon web services, which consists of a raw disk image and a pointer to a kernel and optionally a ramdisk.
Running an Instance

Before we run an instance, we need to make sure there are sufficient resources available in the cloud (e.g. the nodes). We’ll use the euca-describe-availability-zones to show us all the available resources on our cloud nodes:

Step 1: Verifying Resources

```
cladmin@cc:~$ euca-describe-availability-zones verbose
AVAILABILITYZONE cluster1 144.60.26.85
AVAILABILITYZONE | - vm types free / max cpu ram disk
AVAILABILITYZONE | - m1.small 0016 / 0016 1 192 2
AVAILABILITYZONE | - c1.medium 0016 / 0016 1 256 5
AVAILABILITYZONE | - m1.large 0008 / 0008 2 512 10
AVAILABILITYZONE | - m1.xlarge 0008 / 0008 2 1024 20
AVAILABILITYZONE | - c1.xlarge 0004 / 0004 4 2048 20
```

These default availability zones can be modified under the “Administration” tab in the Eucalyptus administrative web interface.

Step 2: Checking Images

The command “euca-describe-images” is the command-line equivalent of clicking the “Images” tab in the Eucalyptus administrative web interface. This shows the emi-xxxxxx identifier for each image/bundle that will be used to run an instance.

```
cladmin@cc:~$ euca-describe-images
IMAGE emi-E088107E image-store-1276733586/image.manifest.xml admin available public x86_64machine eki-F6DD1103 eri-OB3E1166
IMAGE eri-OB3E1166 image-store-1276733586/ramdisk.manifest.xml admin available public x86_64ramdisk
IMAGE eki-F6DD1103 image-store-1276733586/kernel.manifest.xml admin available public x86_64kernel
```
Step 3: Checking Security Groups

Security groups are basically sets of iptables firewall rules that control connection requests originating from hosts outside the cloud and destined towards virtual instances running inside the cloud.

We can view the security groups within Eucalyptus by issuing the following command:

```
cladmin@cc:~$ euca-describe-groups
```

Because the security group “default” does not by default contain any rules allowing external access to cloud instances, we’ll need to either modify the default security group or create a new group and use it instead of the default group, and for this exercise we chose the latter, opting to create a new group called “wiki”:

```
cladmin@cc:~$ euca-add-group wiki -d wiki_demo_appliances
cladmin@cc:~$ euca-authorize wiki -P tcp -p 22 -s 0.0.0.0/0
cladmin@cc:~$ euca-authorize wiki -P tcp -p 80 -s 0.0.0.0/0
```

Running the `euca-describe-groups` command again should now show our newly built group.

Step 4: Installing a Keypair

We’ll need to build a keypair that will be injected into the instance allowing us to access it via ssh:

```
cladmin@cc:~$ euca-add-keypair mykey > ~/.euca/mykey.priv
cladmin@cc:~$ chmod 0600 ~/.euca/mykey.priv
```

Step 5: Running the instance

Now we are finally ready to begin running instances. We’ll start by creating an instance of our Mediawiki appliance and we’ll assign it to the wiki security group we built earlier so that inbound connections will be allowed on ports ssh and http:

```
cladmin@cc:~$ euca-run-instances -g wiki -k mykey -t c1.medium emi-xxxxx
```

Note that if a smaller availability zone was selected for our image, it would automatically terminate because of insufficient space. Checking the `/var/log/eucalyptus/nc.log` file on the node can provide useful clues in these cases.
Monitoring and Accessing Instances

After issuing the “euca-run-instances” command to run an instance, we can track its progress from pending to running state by using the euca-describe-instances command. We can also make a note of the public IP assigned so we can test accessing the instance from outside the cloud. Here we launch the euca-run-instances command in conjunction with the “watch” utility to view output every second:

```
cladmin@cc:~$ watch -n1 euca-describe-instances
```

It may be useful at times to see the console output of an instance. We can use the euca-get-console-output command for this task, where i-xxxxxx corresponds to the image ID listed by the “euca-describe-instances” command:

```
cladmin@cc:~$ euca-get-console-output i-xxxxxx
```

Because we allowed ssh in our security group, we can access the wiki via ssh using the key we specified when creating the instance:

```
cladmin@cc:~$ ssh -i ~/.euca/mykey.priv ubuntu@w.x.y.z
```

Using the public IP, we should also browse to the URL of the instance to ensure the wiki is available:

```
http://w.x.y.z/mediawiki
```
Maxing out the Cloud
To get a feel for the performance under load, we can spin up instances in all the remaining availability zones. First we’ll want to confirm what we have available:

```
cladmin@cc:~$ euca-describe-availability zones verbose
AVAILABILITYZONE    |              | 144.60.26.85
AVAILABILITYZONE    | - vm types   | free / max cpu  ram  disk
AVAILABILITYZONE    | - m1.small   | 0015 / 0016 1  192  2
AVAILABILITYZONE    | - c1.medium  | 0015 / 0016 1  256  5
AVAILABILITYZONE    | - m1.large   | 0007 / 0008 2  512 10
AVAILABILITYZONE    | - m1.xlarge  | 0007 / 0008 2 1024 20
AVAILABILITYZONE    | - c1.xlarge  | 0003 / 0004 4 2048 20
```

We can see how long it takes to spin up 15 instances of the wiki image on our DL380:
```
cladmin@cc:~$ euca-run-instances -g wiki -n xyz -k mykey -t c1.medium emi-xxxxx
cladmin@cc:~$ date
cladmin@cc:~$ watch -n2 euca-describe-instances
cladmin@cc:~$ date
```

Again, we can visit the URL of any of the new instances to see that the instance is up and running and responding to external connections.

Notes
The transient nature of cloud instances:
Once an instance is terminated, all data is lost. One way around this limitation is to configure Elastic Block Storage (EBS) and install the OS of the image inside a chroot environment on the EBS volume.

High Availability:
There isn’t much in the way of HA in a default installation of Eucalyptus, although the developers are almost certainly working on something in this department due to the demand. In the meantime there are probably a few Eucalyptus users out there who have either written scripts to detect an instance is no longer running and launch it on another node, or who are investigating something along those lines.

Q&A
For more information:
http://open.eucalyptus.com/forum
http://www.ubuntu.com/cloud/private
http://www.ubuntu.com/getubuntu/download-server