Eucalyptus Hybrid Cloud Guide

2014-02-23 Eucalyptus Systems

Contents

Hybrid Cloud Overview

This topic presents a definition of hybrid cloud and some use cases for developing a hybrid cloud.

Welcome to the Eucalyptus Hybid Cloud Guide.

What is a Hybrid Cloud?

A *hybrid* cloud computing environment combines a variety of deployment scenarios. When application workloads can move between a public cloud and an on-premise cloud environment, it's a hybrid cloud. Amazon Web Services (AWS) used with Eucalyptus is a hybrid cloud.

In general, public clouds provide convenience of deployment, and private clouds provide control (control over expenses, control over configuration and performance, control over data and security). Together, in a hybrid cloud, they provide flexibility to run workloads wherever desired at any given moment. In financial terms, public clouds are expensive to operate for their users, and private clouds are skewed towards capital expenditures.

To enable workload mobility between the various clouds in a hybrid environment, APIs must be compatible. Eucalyptus was built from the ground up to be API-compatible with Amazon Web Services.

There are many use cases for hybrid clouds, including:

- Develop and test on a private cloud, then deploy for production on a public cloud for global scalability and elasticity.
- Develop and test on a public cloud, then deploy for production in the protection and security of a private cloud.
- Run the base load on a cost-efficient private cloud and use the public cloud for workload spikes (this is known as *cloudbursting*).
- Use hybrid cloud for backup and disaster recovery

Recommended Tools

We recommend that you use the following tools to perform the tasks explained in this guide.

- *Euca2ools*
- *AWS SDK for Java*
- *EucaLobo*

Hybrid Cloud Concepts

Concepts overview.

Availability Zone

Within each AWS *region*, there are multiple availability zones. Each availability zone is essentially a different data center.

Cloudbursting

Cloudbursting refers to the ability to move workloads from one cloud to another. Designing for cloudbursting allows an application that normally runs in a public cloud to also run in a private cloud. For example, a retail company that has a private cloud for day-to-day operations could use resources on a public cloud for peak sales periods.

Cluster

In Eucalyptus, a cluster is like an AWS *availability zone*. Each cluster in Eucalyptus has its own Cluster Controller and associated Node Controllers.

Data Gravity

Data gravity refers to the theory that as data increases, services and applications tend to draw closer to the data, alleviating latency. Data is larger and more dense (and so has greater 'gravity') than services or applications.

Hybrid Cloud

A hybrid cloud is an environment that contains distinct private and public cloud infrastructures. These clouds, usually private and public can share information and resources by using standardized or proprietary technology that enables data and application sharing.

Region

In AWS, a region acts as an independent cloud, located in geographical areas that help to reduce data latency in your applications. Each region is comprised of several *availability zones*.

Hybrid Cloud Tasks

Tasks overview.

Setup Euca2ools for AWS

Installing Euca2ools on RHEL 6 or Centos 6

Euca2ools is included with package installations of Eucalyptus. Please check with your administrator to confirm that Euca2ools is installed properly on your client machine.

To install Euca2ools 3.0.2 on RHEL 6 or CentOS 6 using Eucalyptus-provided packages:

1. Install the EPEL package repository:

```
yum install
http://downloads.eucalyptus.com/software/euca2ools/2.1/rhel/6/x86_64/epel-release-6.noarch.rpm
```

2. Install the euca2ools package repository:

```
yum install
http://downloads.eucalyptus.com/software/euca2ools/3.0/rhel/6/x86_64/euca2ools-release-3.0.noarch.rpm
```

3. Install the euca2ools package:

```
yum install euca2ools
```

Installing Euca2ools on Mac OS X

This tasks show you how to install Euca2ools on Mac OS X. NOTE: This section assumes you have read the "Working with Euca2ools Configuration Files" section of the Euca2ools guide.

To install Euca2ools 3.0.2 on RHEL 6 or CentOS 6 using Eucalyptus-provided packages:

1. Download the Euca2ools tar file from *http://downloads.eucalyptus.com/software/euca2ools/3.0/source/euca2ools-3.0.2.tar.gz*

```
curl -O
href="http://downloads.eucalyptus.com/software/euca2ools/3.0/source/euca2ools-3.0.2.tar.gz
```

2. Install Euca2ools:

```
sudo python setup.py install
```

Setup Euca2ools for both AWS and Eucalyptus

This tasks show you how setup a Euca2ools configuration file for both AWS and Eucalyptus clouds. For more information on Euca2ools configuration files, see the **Euca2ools Guide**.

To setup Euca2ools to quickly switch between your Eucalyptus and AWS clouds:

1. Open `/etc/euca2ools/euca2ools.ini`.

2. Add a `user` section with your Eucalyptus credentials. For example:

```
[user euca-user]
key-id=YOUR-KEY-ID
secret-key=YOUR-SECRET-KEY
```

3. Add a `user` section with your AWS credentials. For example:

```
[user aws-user]
key-id=YOUR-KEY-ID
secret-key=YOUR-SECRET-KEY
```

4. Add a `region` section with the appropriate Eucalyptus cloud service endpoints, with a user entry pointing to the user you defined in the previous step that contains your AWS credentials.:

```
[region euca]
autoscaling-url=http://128.0.0.1:8773/services/AutoScaling
ec2-url=http://128.0.0.1:8773/services/Eucalyptus
elasticloadbalancing-url = http://128.0.0.1:8773/services/LoadBalancing
iam-url=http://128.0.0.1:8773/services/Euare
monitoring-url=http://128.0.0.1:8773/services/Monitoring
s3-url=http://128.0.0.1:8773/services/Walrus
user=euca-user
```

5. Add a region section with the appropriate AWS cloud service endpoints, with a user entry pointing to the user you defined in the previous step that contains your AWS credentials. For example:

```
[region aws]
autoscaling-url = https://autoscaling.us-east-1.amazonaws.com/
ec2-url = https://ec2.us-east-1.amazonaws.com/
elasticloadbalancing-url =
https://elasticloadbalancing.us-east-1.amazonaws.com/
iam-url = https://iam.amazonaws.com/
monitoring-url = https://monitoring.us-east-1.amazonaws.com/
s3-url = https://s3.amazonaws.com/
user=aws-user
```

6. Add a `global` section with an entry specifying your default region - this will be the region that's used by Euca2ools if you do not specify a --region parameter on the command line:

```
[general]
default-region=euca
```

You've now configured Euca2ools to easily switch between your AWS and your Eucalyptus clouds. To specify a region that you've defined in the configuration file, simply use the --region parameter when running Euca2ools commands. For example:

```
euca-describe-instances --region aws
```

Prepare a Linux Image for Eucalyptus

This section explains how to prepare an image before importing it for use in Eucalyptus.

1. **Install cloud software and drivers:**

a) Make sure Virtio drivers are installed if the image is going to be run in a KVM cluster which has virtio enabled, and verify use if possible (ie. set disks and network interface in hypervisor, try hot plug in for disks, etc). For most recent Linux distributions nothing is needed to be done.

b) Make sure Vmware tools are installed if the image is going to be run in a vmware cluster. There are several options for Linux-based images, including RPM, yum installations, or through Vmware if the guest is running in Vmware.

c) Make sure appropriate init scripts are in place; for example: cloud-init packages (if appropriate), and rc.local or similar scripts to prepare new instances at boot time utilizing user/meta-data.

d) Install cloud-init:

Note: For more information on cloud-init, go to *https://help.ubuntu.com/community/CloudInit*

For Ubuntu images:

```
sudo apt-get install cloud-init
```

For Red Hat, Fedora, and CentOS EL5:

```
rpm -Uvh
http://download.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm
```

For Red Hat, Fedora, and CentOS EL6:

```
rpm -Uvh
http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
yum install cloud-init
```

e) Install and configure ssh:

For Ubuntu:

```
sudo apt-get install openssh-server
```

For Red Hat, Fedora, and CentOS:

```
yum install openssh-server
chkconfig sshd on
```

f) Install Euca2ools:

For Ubuntu:

```
sudo apt-get install euca2ools
```

For Red Hat, Fedora, and CentOS:

```
yum install euca2ools
```

g) Optionally, update existing packages.

For Ubuntu:

```
sudo apt-get update
sudo apt-get upgrade
```

For Red Hat, Fedora, and CentOS:

```
yum update
```

2. **Prepare the network:**

a) Disable the firewall. It is recommended that the firewall is disabled and network rules are instead enforced in the security-group the instances run in. If the guest's firewall is not disabled, review the existing rules and make sure they are appropriate for the guest's future use within a cloud environment.

b) Clear or disable iptable rules:

Save the rules in case you want to restore them later:

```
sudo iptables-save > /root/firewall.rules
```

Clear the rules:

```
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
```

For Red Hat, Fedora, and CentOS, you can disable iptables via service scripts. For example:

```
/etc/init.d/iptables stop
(or...)
service iptables stop
(then use...)
chkconfig iptables off  (to disable at boot time as well)
```

c) Disable selinux.

In some distributions, the selinux configuration file can be found in the following locations:

```
/etc/sysconfig/selinux
/etc/selinux/config
```

The following is an example selinux config file:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - SELinux is fully disabled.
SELINUX=disabled
# SELINUXTYPE= type of policy in use. Possible values are:
#       targeted - Only targeted network daemons are protected.
#       strict - Full SELinux protection.
SELINUXTYPE=targeted
```

Note: Some distributions may not have the selinux config and may need a flag set in the booter. For example /boot/grub/grub.conf may need 'enforcing=0' added to the 'kernel' configuration line.

d) Make sure there is only a single primary network interface.

Check the configuration for:

- Enabled on boot (`ONBOOT="yes"`)
- IP provisioning is done via DHCP (`BOOTPROTO="dhcp"`)
- MAC address is commented out (for example:`#HWADDR="AA:BB:CC:DD:EE:FF"`).

Note: If the system is rebooted after you've commented out the MAC address, the MAC address may be restored and will need to be commented out again.

For Red Hat, Fedora, and CentOS images, the configuration for the default network interface can usually be found in the following file:

```
/etc/sysconfig/network-scripts/ifcfg-eth0
```

The following is an example of an `ifcfg-eth0` configuration file:

```
DEVICE=eth0
ONBOOT=yes
#THE HWADDR LINE MUST BE COMMENTED OUT OR REMOVED
#HWADDR=AA:BB:CC:DD:EE:FF
TYPE=Ethernet
BOOTPROTO=dhcp
PERSISTENT_DHCLIENT=yes
```

e) Remove persistent udev rules:

```
echo "" > /etc/udev/rules.d/70-persistent-net.rules
echo "" > /lib/udev/rules.d/75-persistent-net-generator.rules
```

f) On CentOS and Red Hat, disable zeroconf by adding an entry to the `/etc/sysconfig/network` file:

```
NETWORKING=yes
NOZEROCONF=yes
```

3. Clean the image:

a) We recommend that you remove all non-root, non-administrator users before bundling the image.

b) Remove root/Administrator password. We recommend that you remove root's password for Linux systems (for windows, use sysprep (see Administrators guide for Windows Integration tool).

Note: Once these passwords are removed, access to this system will be limited or blocked until this image is recreated as a cloud instance. SSH host and authorization keys for Linux (or dynamically created passwords for Windows sysprep) will be used going forward.

c) For larger Windows images, we recommend that you use the tool of your choice to zero out unused disk space.

d) Remove any unwanted programs.

4. Configure a serial port by adding an option to the end of the `/boot/grub/menu.lst` file:

```
console=ttyS0
```

You've now prepared your instance for image creation. ** add 'continue to...' link to image creation topic here **

Migrate a Linux Image from AWS to Eucalyptus

You can migrate an S3-backed image from AWS to Eucalyptus.

Note: This topic assumes you are migrating an S3-backed Amazon Machine Image (AMI) that you own. For instructions on creating an S3-backed AMI from an existing AMI, see *Creating an Instance Store-Backed AMI From an Existing AMI*.

Note: Specific examples may vary depending on the distro running on the image that you want to migrate.

1. Set up your Euca2ools configuration to work with both Eucalyptus and Amazon Web Services. For more information, see .
2. Check ownership of the AMI that you want to migrate to Eucalyptus by using the euca-describe-images command. For example:

```
euca-describe-images --region us-east-1 --owner 999999999999
IMAGE ami-e1a1e888 999999999999/precise-test 999999999999 available private
 x86_64 machine aki-88aa75e1   instance-store paravirtual xen
 BLOCKDEVICEMAPPING EPHEMERAL sda2 ephemeral0
```

3. Run the AWS instance.

```
euca-run-instance ami-e1a1e888 --region us-east-1
```

4. Install Euca2ools on the instance. For instructions, see the Euca2ools Guide.
5. Make sure that you have enough space on a volume to hold the bundle that we will create.
6. Bundle the running AWS instance. For example:

```
sudo -s
ec2-bundle-vol -b testbucket -d /mnt -u 4299-4227-3585 -k my-aws.pem -c
my-aws-cert.pem -r x86_64 -s 2048
Copying / into the image file /mnt/image...
Excluding:
  /dev/pts
  /
  /sys
  /proc
  /proc/sys/fs/binfmt_misc
  /dev
  /media
  /mnt
  /proc
  /sys
  /mnt/image
  /mnt/img-mnt
1+0 records in
1+0 records out
1048576 bytes (1.0 MB) copied, 0.00209056 s, 502 MB/s
mke2fs 1.42.3 (14-May-2012)
Bundling image file...
Splitting /mnt/image.tar.gz.enc...
Created image.part.00
Created image.part.01
[example truncated]
Created image.part.30
Generating digests for each part...
Digests generated.
Unable to read instance meta-data for ancestor-ami-ids
Unable to read instance meta-data for ramdisk-id
Unable to read instance meta-data for product-codes
Creating bundle manifest...
ec2-bundle-vol complete.
```

7. Switch the Euca2ools configuration file to use Eucalyptus. You can do this by specifying the Eucalyptus region as defined in your Euca2ools configuration file by specifying the `--region` parameter on the command line, or by changing the `default-region` option in the Euca2ools configuration file:

```
default-region = euca-release
```

8. Download the bundle from the AWS S3 bucket:

```
euca-download-bundle --bucket testbucket --directory /tmp/aws-image/ --region
 us-east-1
```

9. Unbundle the AWS instance bundle:

```
euca-unbundle --manifest /tmp/aws-image/image.manifest.xml --source
/tmp/aws-image/ --destination /tmp/aws-image/ --region us-east-1
```

10. Run some checks to make sure that the image can be used with Eucalyptus:

    a) Mount the image via loopback:

```
#  sudo mkdir /mnt/aws-image
 #  sudo mount -o loop /tmp/aws-image/image /mnt/aws-image
 #  df -ah
 ……
 /tmp/aws-image/image
                      9.9G  1.1G  8.3G  12% /mnt/aws-image
```

    b) Make sure that the distro repositories in the image do not point to EC2-specific repositories.

    c) Install a non-Xen kernel into the image from distro and make sure VirtIO modules are added.

    d) Extract the ramdisk and kernel to be bundled, uploaded and registered as ERI and EKI files. In this example, initrd.img-3.2.0-53-virtual and vmlinuz-3.2.0-53-virtual will be copied from /mnt/aws-image/boot to the /tmp/aws-image directory:

```
# sudo cp /mnt/aws-image/boot/initrd.img-3.2.0-53-virtual /tmp/aws-image/.
# sudo cp /mnt/aws-image/boot/vmlinuz-3.2.0-53-virtual /tmp/aws-image/.
```

    e) Make sure that the file system of the image is either ext2, ext3, or ext4 by using the file command. For example:

```
# file /tmp/aws-image/image
/tmp/aws-image/image: Linux rev 1.0 ext4 filesystem data (extents) (large
 files) (huge files)
```

11. Bundle the image using euca-bundle-image:

```
euca-bundle-image -i /tmp/aws-image/image
```

12. Upload the AWS bundled instance to Eucalyptus.

```
euca-upload-bundle -b hybrid-guide-sample-bucket -m
/tmp/aws-image/image.manifest.xml --access-key myaccesskey --secret-key
mysecretkey
```

13. Test the new uploaded image.

```
euca-run-instance emi-a6e15bcf
```

Migrate a Linux Image from Eucalyptus to AWS

To migrate an image from Eucalyptus to AWS, perform the following steps.

Note: These are high-level guidelines for moving an instance from Eucalyptus to AWS. Specific examples will vary depending on the distro running on the image.

1. Run an instance from the image you chose.

```
euca-run-instance emi-1A6338AE
```

2. SSH into the instance and verify that the instance is valid for use with AWS:
   a) Download the latest ec2-modules from *http://s3.amazonaws.com/ec2-downloads* and put them into the `/lib/modules` directory on the instance.
   b) Copy the AWS EC2 certificate and private key from the AWS instance to your local workstation.
   c) Shut down unneeded services on the AWS instance (for example, Apache and MySQL).
   d) Clear out log files and bash history files.
   e) Remove your ssh keys from the instance.
   f) Reset passwords for the instance, and for any services that maintain their own password database.
   g) Clear out any temporary directories.
3. Install Euca2ools on the instance.

```

```

4. Mount a volume that is at least 1.5 times as large as the entire instance.
5. Bundle the running instance.

```
euca-bundle-instance
```

6. Switch the Euca2ools configuration file to use Eucalyptus

```
default-region = euca-release
```

7. Upload the AWS bundled instance to Eucalyptus.

```
euca-upload-bundle -b bucket_name -m manifest_file
```

8. Test the new uploaded image.

```
euca-run-instance emi-a6e15bcf
```

Hybrid Cloud Use Cases

Eucalyptus addresses the following use cases for using a hybrid cloud.

- Cloud bursting or resource expansion: In this use case, you would need to create additonal resources for your cloud, using resources from another cloud.
- Migrating your cloud: In this use case, you would need to export data (images, volumes, configuration, etc.) to another cloud.

Hybrid Cloud Best Practices

This section contains an overview of best practices for creating and maintaining a hybrid cloud.

- Hybrid image/volume management
- Kernels, ramdisks, pvgrub, hypervisor independence, and related issues
- Image import/export tools/svcs
- Minimal image + (puppet/chef/etc.) versus full baked image: getting the best of both worlds
- Using tagging to keep images in sync across regions/clouds
- Access
- Sharing scaling policies across regions/clouds
- Tying DNS together across public/private clouds (maybe use Route 53, if you have external access)

Index