EUCALYPTUS

Image Management Guide BETA PREVIEW

2014-11-05 Eucalyptus Systems

Contents

Image Management Overview

Welcome to the Eucalyptus Image Management Guide. This guide contains conceptual overview about what images are, as well as best practices and common tasks for using images in your cloud.

Note: This guide assumes a moderately high level of expertise with the Linux command line.

Note: Because of the wide variety of Linux distributions (and other operating systems) that may be running on an instance, specific examples may vary.

Document version: Build 2436 (2014-11-05 15:24:12)

Image Overview

An image defines what will run on a guest instance in your Eucalyptus cloud. An image contains everything necessary to boot and run an operating system: either one of the Linux distributions – CentOS, Fedora, Ubuntu, Debian, etc – or one of the supported Windows server versions.

Images of two types are supported by Eucalyptus:

- **HVM images** are raw disks that can boot independently. They can contain Linux or Windows operating systems. HVM stands for Hardware-assisted Virtual Machine because such images can only run efficiently on hardware that supports virtualization. When an HVM image is uploaded and registered, it becomes a Eucalyptus Machine Image (EMI) of type "hvm", with a unique ID.
- **Paravirtual images** are Linux images that can boot if they are paired with a kernel and ramdisk that are compatible with the host's hypervisor. Currently only root file system images are supported (on AWS, a paravirtual image can be a file system or a full disk). When a paravirtual image is uploaded and registered, it also becomes an EMI of "paravirtual" type, which needs to be paired with a kernel (EKI) and ramdisk (ERI) images to be usable. EKI contains a kernel (i.e., the 'vmlinuz' file typically found in the /boot directory of a Linux system). ERI contains the kernel modules (i.e., the 'initrd' file from the /boot directory).

Depending on the method used for upload, an instance's disk will reside on one of two types of storage:

- **Instance Store -** volumes are located on temporary disk space that is destroyed when instances shut down. These volumes are based on a template residing in Object Storage. Instance Store can host either HVM or paravirtual images.
- **Elastic Block Store (EBS) -** volumes are disks with lifetimes that can be independent of instances. These volumes are based on snapshots of EBS volumes (instead of templates in Object Storage). Only HVM images can be deployed on EBS in Eucalyptus (in AWS, EBS can also host paravirtual images).

To help get you started, Eucalyptus Systems provides pre-packaged virtual machine images that are ready to run in your cloud. You can download them from the *Eucalyptus Machine Images* page. Both HVM and paravirtual images are available there. Each paravirtual image comes bundled with a correspoding EKI and ERI.

If you find that the pre-packaged images don't meet your needs, you can migrate an image from another cloud system (such as vSphere or Amazon Web Services) or create your own image. See the rest of this guide for more information.

Tip: For a list of supported guest operating systems, go to *Eucalyptus Cloud Compatibility Matrix*.

Image Tasks

An image is the basis for instances that you spin up for your computing needs. This section explains how to create or acquire an image and then add it to your Eucalyptus cloud.

There are a few ways you can create or acquire an image for use in a Eucalyptus cloud:

- Add an image based on an existing image. Eucalyptus has stock images available to help you get started right away. For more information, see *Browse and Download from Eucalyptus*.
- Add an image that you create from scratch, by installing an operating system into a virtual machine. This is the only way to deploy a Windows instance, since Eucalyptus Systems does not distribute such images. For more information, see the *Create a New Image* section.
- Migrate an image or an instance from another cloud, such as Amazon Web Services (AWS) or a virtualization platform. For more information, see the *Migrate an Image* section.

Browse and Download from Eucalyptus

Eucalyptus Systems provides a few prebuilt "starter" images that you can download and install. This is one of the simplest ways to get started with Eucalyptus. Many other organizations, including Linux distributions, publish starter images for use on cloud systems, which can also be installed on Eucalyptus using the steps in this guide.

1. With your Web browser, explore the image catalog at *http://emis.eucalyptus.com/*
2. Download an image of your choice (we suggest starting with images that have "hvm" as their Virtualization Type, as they are easier to install and work for both Instance Store and EBS-backed instances).
3. Optionally, if the image is compressed (has a .tgz or .gz extension), you will have to uncompress it.

If you obtained an HVM image, you now have a raw disk file that can be installed using instructions in the *Install an HVM image* section. If you obtained a paravirtual image with kernel and ramdisk, you can install them with instructions in the *Install a paravirtual image* section.

Install an HVM Image

HVM (Hardware-assisted Virtual Machine) images are sequences of raw sectors of a complete bootable disk, including a boot loader and one or more partitions with an operating system. Such images are self-contained and can be booted on hardware that supports virtualization or on a full software emulation of hardware (QEMU).

At run time, HVM images can be deployed to disks local to the hypervisor (so-called Instance Store images) or they can run off of dynamic volumes on EBS (so-called EBS-backed instances or boot-from-EBS instances). This topic covers methods for installing an HVM image into either type of EMI.

Install an HVM image as an Instance Store EMI

Bundle, upload, and register the HVM image. All of this can be accomplished with a single command.

- For Linux images, the required options are:

```
euca-install-image -i /path/to/hvm-image -n name-of-the-image -r x86_64
--virtualization-type hvm -b bucket-name-for-image
```
- When installing a Windows image, an additional flag is necessary:

```
euca-install-image -i /path/to/hvm-image -n name-of-the-image -r x86_64
--virtualization-type hvm -b bucket-name-for-image --platform=windows
```

Install an HVM image as an EBS-Backed EMI

An EBS-backed image (sometimes referred to as a "bfEBS" image) is an image with a root device that is an EBS volume created from an EBS snapshot. An EBS-backed image has a number of advantages (over an Instance Store image), including:

- Faster boot time
- Larger volume size limits
- Changes to the data on the image persist after instance termination

Option A: Install an EBS image with euca-import-vol

With this method, a single command uploads image data into Object Store and triggers a conversion task, which copies the data into an EBS volume. The volume will be the source of the snapshot behind the EBS-backed EMI.

1. Import the HVM image file into Object Storage and then copy it into EBS. For example:

```
euca-import-volume /path/to/hvm-image --format raw --availability-zone AZ-NAME
--bucket bucket-for-hvm-image --owner-akid $EC2_ACCESS_KEY --owner-sak
$EC2_SECRET_KEY --prefix image-name-prefix --description "textual description"
```

> **Note:**
>
> If the volume import process was interrupted, import can be resumed (within a one-week deadline) with the following command:
>
> ```
> euca-resume-import -t import-vol-fae1e94d /path/to/hvm-image
> ```

2. After the volume data has been uploaded to Object Store, an internal conversion task will copy the data into a new EBS volume. You can query the progress of this task with the following command:

```
euca-describe-conversion-tasks import-vol-fae1e94d
```

3. Once the task has finished successfully and the volume is available, you can take a snapshot of the volume:

```
euca-create-snapshot <volume-id>
```

4. You can now register the snapshot:

```
euca-register --name <image-name> --snapshot <snapshot-id> --root-device-name
<device>
```

You've now created a EBS-backed image. To maintain data persistence, be sure to use `euca-stop-instances` and `euca-start-instances` to stop and start your EBS-backed instance.

Option B: Install an EBS image with euca-import-instance

With this method, a single command uploads image data into the Object Store and triggers a conversion task; this conversion task copies the data into an EBS volumetakes a snapshot of the volume, registers the snapshot as an EMI, and deploys the EMI as an instance.

1. Turn an HVM image into a running EBS instance with a single command:

> **Note:** Just as with euca-run-instance, one can specify the Instance Type for the instance that will run, as well as the SSH key. For example:

```
euca-import-instance /root/paravirtualimage/ubuntuJune06.img --format raw
--architecture x86_64 --platform Linux --availability-zone EDU --bucket
import_instance --owner-akid $EC2_ACCESS_KEY --owner-sak $EC2_SECRET_KEY
--prefix eduii --description "import instance" --key admin --instance-type
m1.small
```

> **Note:**
>
> If volume import process was interrupted (by an interruption of the command or due to a network mishap, import can be resumed (within a one-week deadline).

```
euca-resume-import -t import-i-b393f3f6 /path/to/hvm-image
```

2. After the volume data has been uploaded to Object Store, an internal conversion task will copy the data into a new EBS volume. You can query the progress of this task with the following command:

```
euca-describe-conversion-tasks import-i-b393f3f6
```

Once the conversion task has finished successfully, you should see a running instance. You may log into that instance if you specified an SSH key. To maintain data persistence, be sure to use `euca-stop-instances` and `euca-start-instances` to stop and start your EBS-backed instance.

Option C: Install an EBS image using a helper instance

This is a more manual method for installing an HVM image into an EBS-backed EMI. Instead of importing the image data into an EBS volume directly, a helper instance is used to copy data from the image file into the volume. The helper instance can be either an instance store or an EBS-backed instance, and can be deleted when finished. The volume will be the source of the snapshot behind the EBS-backed EMI.

1. Create and launch a help instance.

2. Create a volume big enough to hold the bootable image file:

```
euca-create-volume -z <cluster_name> -s <size_in_GB>
```

3. Attach the volume to the helper instance:

```
euca-attach-volume <volume-id> -i <instance-id> -d <device>
```

4. Log in to the instance and copy the bootable image to the attached volume by performing one of the following steps:

   • If the bootable image is saved in an http or ftp repository, use `curl` or `wget` to download the .img file to the attached volume. For example:

   ```
   curl <path_to_bootable_image> > <device>
   ```

   • If the bootable image is from a source other than an http or FTP repository, copy the bootable image from its source to the helper instance, and then copy it to the attached volume using the `dd` command. For example:

   ```
   dd if=<path_to_bootable_image> of=<device> bs=1M
   ```

5. Detach the volume from the instance:

```
euca-detach-volume <volume-id>
```

6. Create a snapshot of the volume:

```
euca-create-snapshot <volume-id>
```

7. Register the snapshot:

```
euca-register --name <image-name> --snapshot <snapshot-id> --root-device-name
<device>
```

You've now created a EBS-backed image. To maintain data persistence, be sure to use `euca-stop-instances` and `euca-start-instances` to stop and start your EBS-backed instance.

Create an HVM Image

This section covers how to create a new image for use in your Eucalyptus cloud.

Create an HVM Image from a Linux Distribution ISO (KVM)

This topic shows how to install a Linux distribution ISO and prepare an image for registration with Eucalyptus.

1. Use the QEMU disk utility to create a disk image. In the following example, we create a 5GB disk image:

```
qemu-img create -f raw centos6.img 5G
```

2. Use the `parted` utility to set the disk label.

    ```
    parted centos6.img mklabel msdos
    ```

3. Use `virt-install` to start a new virtual machine installation, as in the following example (note the example has been broken into multiple lines for formatting purposes):

    ```
    virt-install --name centos6 --ram 1024 --os-type linux --os-variant rhel6
     -c /tmp/CentOS-6.3-x86_64-bin-DVD1.iso --disk
    path=/tmp/centos6.img,device=disk,bus=virtio
     --graphics vnc,listen=0.0.0.0 --force
    ```

4. Use the VNC client of your choice to connect to the new virtual machine and complete the installation.

5. Modify the following `libvirt.xml` template to create the VM and run `virsh create <libvirt.xml>`".

    ```
    <domain type='kvm'>
        <name>eucalyptus-centos</name>
        <os>
            <type>hvm</type>
        </os>
        <features>
            <acpi/>
        </features>
        <memory>1073741</memory>
        <vcpu>1</vcpu>
        <devices>
            <!--<emulator>/usr/bin/kvm</emulator>-->
            <disk type='file'>
                <source file='/tmp/centos6.img'/>
                <target dev='hda'/>
            </disk>
          <interface type='bridge'>
             <source bridge='br0'/>
             <model type='virtio'/>
          </interface>
            <graphics type='vnc' port='-1' autoport='yes' listen='0.0.0.0'/>
        </devices>
    </domain>
    ```

6. Connect to the virtual machine using your VNC client of choice and make the following configuration changes:

    a) Modify the `/etc/sysconfig/network-scripts/ifcfg-eth0` file and set the `ONBOOT` option to "yes". For example:

    ```
    DEVICE="eth0"
    BOOTPROTO="dhcp"
    #HWADDR="B8:AC:6F:83:1C:45"
    IPV6INIT="yes"
    MTU="1500"
    NM_CONTROLLED="yes"
    ONBOOT="yes"
    TYPE="Ethernet"
    UUID="499c07cc-4a53-408c-87d2-ce0db991648e"
    PERSISTENT_DHCLIENT=1
    ```

    b) Enable a serial console on the virtual machine by adding the following option to the end of the /boot/grub/menu.lst:

    ```
    console=ttyS0
    ```

    c) Remove the `quiet` option from the kernel parameters and the grub menu splash image in the `/boot/grub/menu.lst` file. For example:

    ```
    # grub.conf generated by anaconda
    #
    # Note that you do not have to rerun grub after making changes to this file
    # NOTICE:  You have a /boot partition.  This means that
    ```

```
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/sda2
#          initrd /initrd-[generic-]version.img
#boot=/dev/sda
default=0
timeout=5
#splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Eucalyptus (2.6.32-358.18.1.el6.x86_64)
 root (hd0,0)
 kernel /vmlinuz-2.6.32-358.18.1.el6.x86_64 ro
root=UUID=062b9c31-95f3-424f-8b47-35107cfdfc08 rd_NO_LUKS rd_NO_LVM
LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=auto
KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb
 initrd /initramfs-2.6.32-358.18.1.el6.x86_64.img
```

d) Add the following line to the `/etc/sysconfig/network` file to disable the zeroconf route, which can interfere with access to the metadata service:

`NOZEROCONF=yes`

e) Edit the `/etc/udev/rules.d/70-persistent-net.rules` file and remove all entries for the existing network interface. Make sure that you delete all entries before terminating the virtual machine and registering it with Eucalyptus.

7. Eucalyptus instances use cloud-init to specify actions to run on your instance at boot time, which can be passed using the userdata parameter. To install and configure `cloud-init` on your instance:

a) Install cloud-init:

```
# rpm -Uvh
http://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
# yum install cloud-init
```

b) By default, cloud-init uses ec2-user as the log-in user. Add ec2-user to your instance and give it appropriate sudo permissions:

```
# adduser ec2-user
# passwd ec2-user
```

c) Run visudo and add the following entries at the bottom of the sudoers file:

```
## Allow root to run any commands anywhere
    root    ALL=(ALL)       ALL
    ec2-user        ALL=(ALL)       NOPASSWD: ALL
```

d) You can optionally copy Eucalyptus `rc.local` file (available at *https://github.com/eucalyptus/Eucalyptus-Scripts/blob/master/rc.local*) to the `/etc` directory on your virtual machine and modify it as needed.

You now have a raw disk file that can be installed using instructions in the *Install an HVM image* section.

Create an HVM image from Windows Installation Media

This section details the tasks necessary to create a Windows image.

We recommend that you perform this task on a node controller (NC), or a host running the same Linux distributions and hypervisors as your NCs. If you are creating the Windows image on a machine currently running as a NC, terminate all running instances and stop the NC. To stop the NC, enter:

```
service eucalyptus-nc stop
```

A template file that closely matches those that Eucalyptus generates at VM instantiation time is located at `/usr/share/eucalyptus/doc/libvirt-kvm-windows-example.xml`. We recommend that you review

the file to acquaint yourself with its contents, noting required files, bridges, and resources. For more information about configuring the libvirt.xml file, go to the *Domain XML format* page in the libvirt documentation.

To create an image from a Windows OS in VMware you will need one network interface and one disk.

> **Note:** If you are using VMware, make sure that the Windows VM uses the LSI Logic Parallel driver as the SCSI controller. For some Windows versions, this is not the default SCSI controller in the VM setting.

Install Base Windows OS

The first task for creating a Windows image is installing a base Windows operating system (OS). To install a base Windows OS using KVM:

1. Log in to the stopped NC server or a host that runs the same hypervisor as the NCs.
2. Create a blank disk file. Specify your Windows VM image name using the parameter `of`.

   ```
   dd if=/dev/zero of=windows.<image_name>.img bs=1M count=1 seek=16999
   ```

   > **Important:** Your image name must start with the word, `windows` (all lower-case).

3. Create a floppy and secondary blank disk to be attached to the image later, in order to test paravirtualization drivers

   ```
   dd if=/dev/zero of=floppy.img \
   bs=1k count=1474
   dd if=/dev/zero of=secondary.img \
   bs=1M count=1 seek=1000
   ```

4. Copy all of the .img and .iso files to the `/var/lib/libvirt/images/` directory.
5. Copy the `libvirt-kvm-windows-example.xml` file to your working directory and rename it to `libvirt-kvm-windows.xml`.

   ```
   cp /usr/share/eucalyptus/doc/libvirt-kvm-windows-example.xml
   /var/lib/libvirt/images/libvirt-kvm-windows.xml
   ```

   or

   ```
   cp /usr/share/eucalyptus/doc/libvirt-xen-windows-example.xml
   /var/lib/libvirt/images/libvirt-xen-windows.xml
   ```

6. Open the new `libvirt-kvm-windows.xml` file and provide fully qualified paths to the VM image file and iso. Make sure that the name of the bridge is the same as the one used by the hypervisor on which you are creating the Windows image.

   Your file should look similar to the following example:

   ```
   <domain type='kvm'>
       <name>your-domain-name-here</name>
        <os>
        <type>hvm</type>
        <boot dev='cdrom'/>
        </os>
        <features>
            <acpi/>
        </features>
        <memory>524288</memory>
        <vcpu>1</vcpu>
        <devices>
            <emulator>/usr/libexec/qemu-kvm</emulator>
            <disk type='file'>
                <source file='/var/lib/libvirt/images/windows_2003.img'/>
                <target dev='hda'/>
            </disk>
            <!-- <disk type='file' device='disk'>
   ```

```
              <source file='fully_qualified_path_to_secondary_disk'/>
              <target dev='vda' bus='virtio'/>
        </disk>
        <disk type='file' device='floppy'>
              <source file='fully_qualified_path_to_floppy_disk'/>
              <target dev='fda'/>
        </disk> -->
        <disk type='file' device='cdrom'>
              <source
file='/var/lib/libvirt/images/en_win_srv_2003_r2_enterprise_with_sp2_cd1_x13-05460.iso'/>

              <target dev='hdc'/>
              <readonly/>
        </disk>
        <interface type='bridge'>
              <source bridge='br0'/>
              <model type='rtl8139'/>
        </interface>
        <!--<interface type='bridge'>
              <source bridge='br0'/>
              <model type='virtio'/>
        </interface> -->
        <graphics type='vnc' port='-1' autoport='yes' listen='0.0.0.0'/>
    </devices>
</domain>
```

7. Start the VM.

```
cd /var/lib/libvirt/images
virsh create libvirt-kvm-windows.xml
```

8. Connect to the virtual console using the VNC client of your choice. On the NC, check the display number that has been allocated by looking at the process table (`ps axw | grep vnc`). For example, if the display number is 0, then connect to the NC using the VNC client:

```
vinagre <machine-hosting-vm>:0
```

9. Follow the standard Windows installation procedure until the VM has completed installing Windows.

> **Tip:** On some hosts, the VNC's display number will change when an image restarts. Use `ps` to find the current number.

10. Run `virsh list` to display the domain name.

11. Shut down the Windows VM you have just created. The easiest way to shutdown your VM is to use the `virsh destroy` command, as shown:

```
virsh destroy <domain_name>
```

To install the base Windows operating system using VMware, create a new VM using the VMware vSphere Client. Install Windows on the VM following standard VMware procedures, and install VMware Tools.

Install Eucalyptus Windows Integration

To install the Eucalyptus Windows Integration Service:

1. **Note:** The following step assumes you're running a recent version of Windows. If you're running Windows Server 2003, you'll need to download the legacy version of the Windows prep tool from: *http://downloads.eucalyptus.com/software/tools/windows-prep/*

If you're running a version of Windows more recent than Windows Server 2003, download the most recent version of the Windows Image Preparation Tool from *http://downloads.eucalyptus.com/software/tools/windows-prep/* to `/var/lib/libvirt/images` on your NC or on the host running the vSphere client.

2. Open the libvirt-kvm-windows-example.xml file you used in the previous section and make the following edits:

- Comment out the lines of XML code directing the hypervisor to boot the Windows image from the CDROM.
- Change the text so that `windows-prep-tools-latest.iso` replaces the Windows .iso image and is mounted as `cdrom`.
- Enter the path to the secondary disk file you created in the previous task.
- Uncomment the lines that direct attachment of a floppy disk, secondary disk, and secondary network interface.

> **Tip:** If you plan on using virtio networking for instances (via USE_VIRTIO_NET option on node controllers), uncommenting the virtio interface in the xml is mandatory

Your finished file should look similar to the following example:

```
<domain type="kvm">
   <name>eucalyptus-windows</name>
   <os>
   <type>hvm</type>
   <!-- <boot dev='cdrom'/> -->
   </os>
   <features>
       <acpi/>
   </features>
   <memory>524288</memory>
   <vcpu>1</vcpu>
   <devices>
       <emulator>/usr/libexec/qemu-kvm</emulator>
       <disk type='file'>
           <source file='/var/lib/libvirt/images/windows_2003.img'/>
           <target dev='hda'/>
       </disk>
       <disk type='file' device='disk'>
            <source file='/var/lib/libvirt/images/secondary.img'/>
            <target dev='vda' bus='virtio'/>
       </disk>
       <disk type='file' device='floppy'>
            <source file='/var/lib/libvirt/images/floppy.img'/>
            <target dev='fda'/>
       </disk>
       <disk type='file' device='cdrom'>
            <source
file='/var/lib/libvirt/images/windows-preps-tools-latest.iso'/>
            <target dev='hdc'/>
            <readonly/>
       </disk>
       <interface type='bridge'>
            <source bridge='br0'/>
            <model type='rtl8139'/>
       </interface>
       <interface type='bridge'>
            <source bridge='br0'/>
            <model type='virtio'/>
       </interface>
       <graphics type='vnc' port='-1' autoport='yes' listen='0.0.0.0'/>
   </devices>
</domain>
```

3. Start the VM.

```
cd /var/lib/libvirt/images/
virsh create libvirt-kvm-windows-example.xml
```

4. For VMware, use the VMware vSphere client to upload the ISO file to the VSphere datastore. Attach the `windows-prep-tools-latest.iso` to the Windows VM.

5. Log in to Windows and find the Eucalyptus installation files in the CDROM drive.

   • For Windows Server 2003 R2, run `setup.exe`. This automatically installs the .NET framework 2.0, which is not bundled in Server 2003 R2.
   • For all other versions, run `EucalyptusWindowsIntegration.msi`. (setup.exe will automatically install .NET framework 2.0, which is not bundled in Server 2003 R2).

6. In the **Choose your hypervisor** step, select **KVM** and then click **Next**.
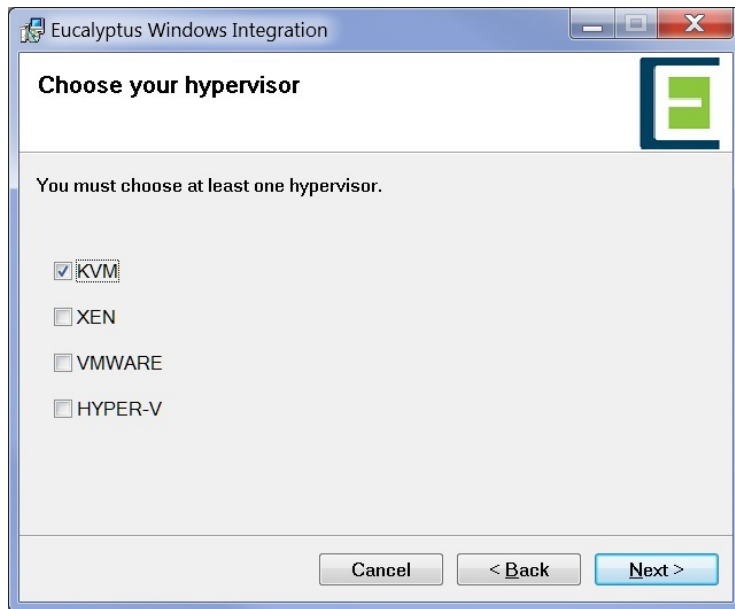
   Click **Next** and continue until the end of installation.

7. Reboot the Windows VM

8. Open the Windows device manager and check that the following drivers are found for each device.

   • Floppy disk drive
   • Disk drivers: Red Hat VirtIO SCSI Disk Device
   • SCSI and RAID controllers: Red Hat VirtIO SCSI controller
   • Network adapters: Red Hat VirtIO Ethernet Adapter

   If the correct drivers are not found, question marks display on the devices. To install the devices, do the following:

   • Right-click on the devices in question and select **Update Drivers** to open the New Hardware Wizard.
   • When the new hardware wizard asks if Windows update is to be connected, click **No, not this time**.
   • Choose **Install software automatically (recommended)**.
   • If a confirmation popup message displays, click **Continue**.

Configure Active Directory

The Eucalyptus Integration service lets an enterprise with existing Active Directory domains attach Windows instances to the domains and control access to these instances using the existing AD user database. Users can log into the instance either using their domain credentials or the Administrator's password generated with the `euca-get-password` command.

Because AD technology is tightly integrated with domain name service (DNS), the default name server contacted by the instance must be able to resolve the AD address as a proper domain controller. You can do this for all networking modes except System, by configuring the following line the CC's eucalyptus.conf file:

```
VNET_DNS=<domain_controller_IP_address>
```

If there is no such pre-existing DNS set-up or your networking mode is System, you might need to change the VM's network interface so that the preferred DNS server points to the domain controller.
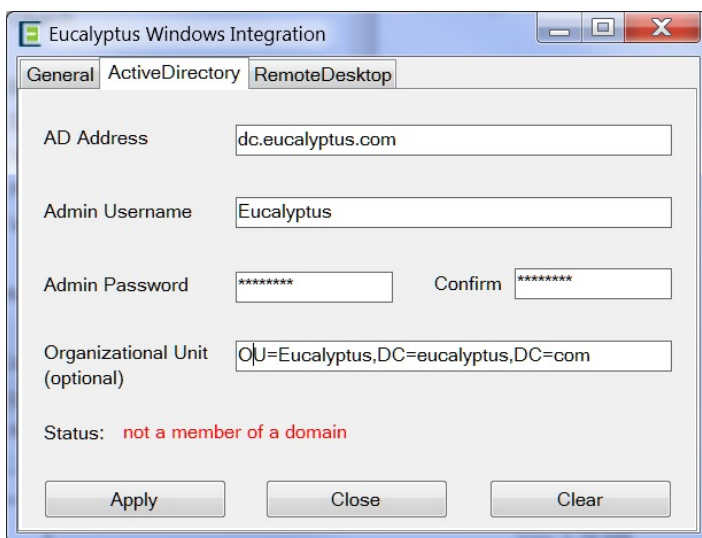
To set up Active Directory:

1. Click **Windows Programs** > **Eucalyptus** > **Eucalyptus Setup**.
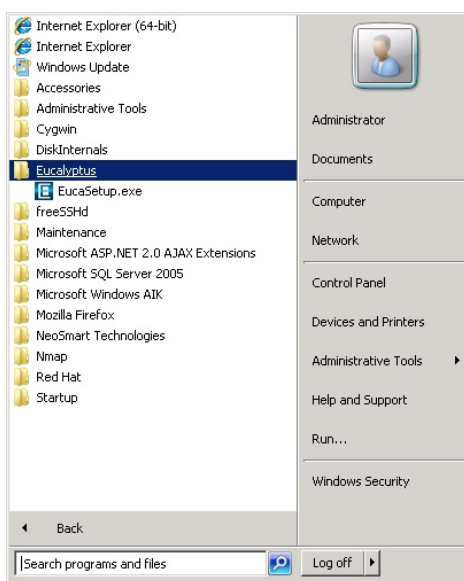
    The **Eucalyptus Windows Integration** popup displays.

2. Click the **Active Directory** tab in the Eucalyptus Windows Integration window and enter the following information:

    -   Enter the domain name of the existing Active Directory domain controller in the **AD Address** field.
    -   Enter the administrator username in the **Admin Username** field. We recommend using a generic user account that has permission to join a computer to a domain or a specific organizational unit.
    -   Enter and confirm the password in the **Admin Password** field. Note that the Admin username and password are required to join an instance to an Active Directory. When launched in Eucalyptus, these properties will be deleted as soon as the instance joins (or fails to join) the domain.
    -   Optionally, enter an organizational unit in the **Organizational Unit** field. This specifies a container that the instances launched from this image will be attach to.

    > **Tip:** If the values entered in this section are incorrect, the launched instances will fail to join the domain. We recommend that you verify the information by manually joining a computer to a domain using the same

information that you entered in this step. You may first log in the launched instance using the administrator password (`euca-get-passcword`) and manually join the domain for verification.

3. Click **Apply**.

Configure Remote Desktop

Domain users or groups require remote desktop permission to log into an instance. By default, only the local administrator has the remote desktop permission. The Eucalyptus Integration Service provides a way to grant remote desktop permission to additional domain users or groups.

To configure remote desktop permission:

1. Open the **Eucalyptus Windows Integration** popup (**Windows Programs** > **Eucalyptus** > **Eucalyptus Setup**).

2. Click the **RemoteDesktop** tab
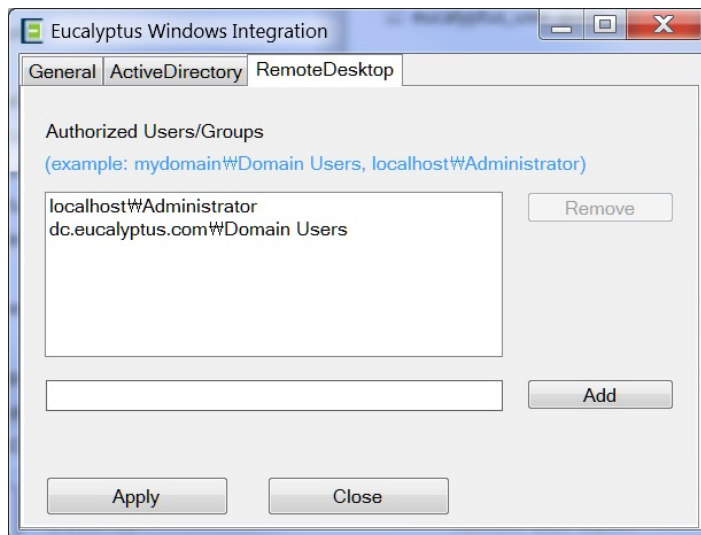
The names of authorized domain users and groups display in the **Authorized User/Groups** field. By default, only the local administrator is listed as authorized.

3. In the text field below the list, enter a user and group account name in the format `[DOMAIN]\[USER or GROUP]`. If you add a new local user or local group, prepend the account name `localhost\` instead of the domain name.
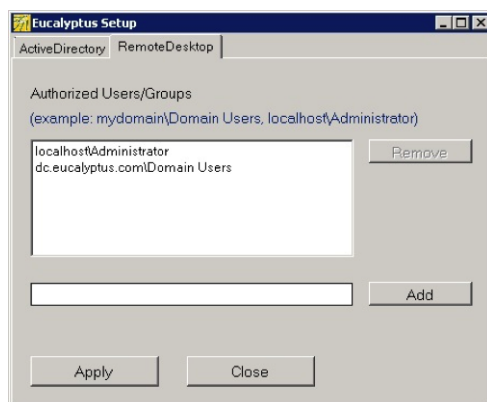
4. Click **Add**.

5. Repeat for all user/groups that you want to add.

6. Click **Apply**.
   When the instance launches, the members of the groups you added can log in to the instance through remote desktop.

Run Sysprep

Sysprep is a Microsoft tool for deploying multiple Windows operating systems in an enterprise. Running Sysprep removes system-specific information such as security ID (SID) from the Windows OS before you clone an image. Sysprep then re-initializes the OS after the image is cloned and started on multiple computers. Use Sysprep to prepare images when you use Microsoft Key Management Service to activate license keys. Also, use Sysprep when your Windows systems are attached to Active Directory.

In Eucalyptus, you can run Sysprep before you bundle images with the `euca-bundle-image` or `euca-bundle-instance` commands.

> **Note:** The Eucalyptus Integration Service supports Sysprep for Windows Server 2008, Windows Server 2008 R2, and Windows 7.

To configure and run Sysprep:

1. Open the **Eucalyptus Windows Integration** popup (**Windows Programs** > **Eucalyptus** > **Eucalyptus Setup**).
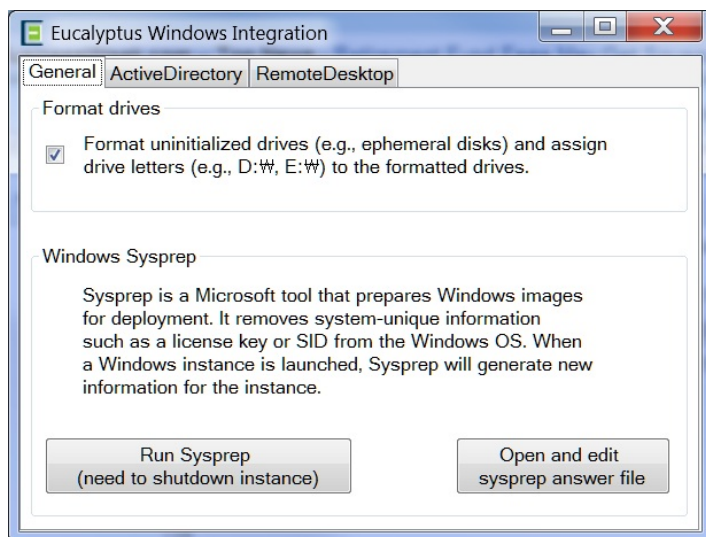
2. Click the **General** tab.
3. Ensure that **Format uninitialized drives** is checked
4. If you want to edit the Sysprep answer file, click the **Open and change** button in the General tab. Otherwise skip this step.
5. Click the **Run Sysprep** button.
   Sysprep starts.
6. After Sysprep is complete, close the application and shutdown the Windows VM using the Windows Programs menu.

Add a Windows Image to Eucalyptus

After the Windows virtual machine has been shut down, its entire disk can be uploaded to Eucalyptus as an HVM image.

1. Extract the disk from the hypervisor into a file.

    - With KVM, the disk may already be a file in raw format. If it's a file in a different format, convert it into raw disk format. If it's a block device, you can use 'dd' utility to copy contents of the block device into a file.
    - With VMware, the disk resides in VMDK format on a possibly remote datastore. A number of VMware tools can be used to extract the disk from a datastore and convert it into raw format. Follow instructions in the *Extract a disk from a vSphere virtual machine* to obtain a raw disk.

2. Upload the raw disk file using the instructions in the *Install an HVM image* section.

Migrate an Image

This section covers migrating images to and from your Eucalyptus cloud.

Create an Image from an Existing EBS-Backed Instance

A common way to create a new image is to customize an existing instance. This topic describes how to create a new image by modifying an existing EBS-backed instance.

This example shows how to create a new EBS-backed Eucalyptus image based on an existing EBS-backed Eucalyptus instance.

1. Log on to an existing Eucalyptus EBS-backed instance and customize the instance.
2. Prepare the image. See *Prepare a Linux System for Eucalyptus* for instructions.
3. Create a new image based on the image you just modified by using the eucalyptus-create-image command, specifying a name, a description, and the instance ID of the Eucalyptus instance you customized in the previous step. For example:

```
euca-create-image -name "mynewimage" -d "This is my new custom recycled image"
i-1ABCDEFF
```

This command will show the ID of the new machine image and exit immediately. In the background, Eucalyptus will begin the process of creating a new image based on the instance you supplied.

You can monitor the status of the image using the euca-describe-images command, supplying the ID of the image returned from the euca-create-image command. For example:

```
euca-describe-images i-e12398ab
```

Create an Image from an Existing Instance-Store Instance

A common way to create a new image is to customize an existing instance.

This example shows how to create a new instance store Eucalyptus image based on an existing instance store Eucalyptus instance. This topic describes how to create a new image by modifying an existing instance-store instance.

1. Log on to an existing Eucalyptus isntance-store instance and customize the instance. See *Prepare a Linux System for Eucalyptus* for instructions.
2. Create a new image based on the image you just modified by using the euca-bundle-instance command, specifying a name, a description, and the instance ID of the Eucalyptus instance you customized in the previous step. For example:

```
euca-bundle-instance -b mybundle -p mycentos6 -o $EC2_ACCESS_KEY -w
$EC2_SECRET_KEY i-96154365
BUNDLE     bun-96154365     i-96154365     mybundle     mycentos
62013-11-05T21:37:23.469Z2013-11-05T21:37:23.469Z     pending     0
```

This command will the bundle task ID and exit immediately. In the background, Eucalyptus will begin the bundling process. Depending on the size of the instance, it can take several minutes for the bundling task to complete.

You can monitor the status of the bundle task using the euca-describe-bundle-tasks command, supplying the ID of the image returned from the eucalyptus-bundle-instance command. For example:

```
euca-describe-bundle-tasks
BUNDLE     bun-96154365     i-96154365     mybundle     mycentos6
2013-11-05T21:37:23.469Z2013-11-05T21:37:58.446Z     storing     0
```

3. Once the bundle task is complete, you can register thebundle as an instance-store image using the euca-register command, specifying the path to the bundle manifest in the format [bucket]/[prefix].manifest.xml. For example:

```
euca-register mybundle/mycentos6.manifest.xml.
```

Your new image is now ready for use in your Eucalyptus cloud.

Prepare a Linux System for Eucalyptus

This section explains how to prepare a running Linux system (cloud instance, virtual machine, or a system running on bare metal) 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.

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.

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

This topic describes how to migrate an image from Eucalyptus to AWS.

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
```

Extract a disk from a vSphere virtual machine

To migrate a disk in VMDK format from vSphere to Eucalyptus, perform the following steps.

1. Download the Eucalyptus Image Tools from github:

```
yum install python-pip git
pip-python install -U psphere
git clone https://github.com/eucalyptus/eucalyptus-image-tools.git
cd eucalyptus-image-tools
```

2. Download the disk image from vSphere, supplying your vSphere IP address, login, password, and VM name. For example:

```
python tools/exporters/export_vmdk.py --ip 10.111.1.133 \
 --user your-vsphere-login --password your-vsphere-password \
 --vmName your-vm-name --workDir /path.to/yourworkdir
Downloading 490 bytes
 File saved to /path.to/yourworkdir/your-vm-name.vmdk
Downloading 5473598976 bytes
 File saved to /path.to/yourworkdir/your-vm-name-flat.vmdk
```

Note: The command will fail if the VM has more than one disk or NIC.

3. Convert the VMDK into a raw disk image:

```
cd /path.to/yourworkdir
sudo qemu-img convert -O raw your-vm-name.vmdk disk.raw
```

This command will create a raw disk image which can then be bundled, uploaded, and registered with Eucalyptus.

Note: When registering the converted disk image, you must specify the `--virtualization-type=hvm` option when running the `euca-register` command line tool.)

Install a paravirtual image

Note: As of Eucalyptus version 4.0, it is now required to pass a Eucalyptus Kernel Image (EKI) and a Eucalyptus Ramdisk Image (ERI) when uploading and registering a paravirtual Eucalyptus Machine Image (EMI) using the `euca-bundle-image`, `euca-bundle-and-upload-image`, and `euca-register` command line tools.

Once you've customized or acquired a paravirtual image to use with Eucalyptus, you can enable the image as an executable entity with the following steps:

1. Unless a suitable kernel is already registered, bundle the kernel, upload it to Object Storage, and register it as a new EKI.
2. Unless a suitable ramdisk is already registered, bundle the ramdisk, upload it to Object Storage, and register it as a new ERI

3. Bundle the root disk image, which must be a Linux partition, requesting the kernel and ramdisk that you desire, upload the bundle to Object Storage, and register it as a new EMI.

> ⭐ **Important:** Note that while all users can bundle, upload and register images, only users under the `administrator` account have the required permissions to upload and register kernels and ramdisks.

Once you have an image that meets your needs, perform the tasks listed in this section to add the image to your cloud.

Add a Kernel

When you add a kernel to Walrus, you bundle the kernel file, upload the file to a bucket in Walrus that you name, and then register the kernel with Eucalyptus.

To add a kernel to Walrus:

Use the following three commands:

```
euca-bundle-image -i <kernel_file> --kernel true --arch <architecture>
euca-upload-bundle -b <kernel_bucket> -m /tmp/<kernel_file>.manifest.xml
euca-register <kernel_bucket>/<kernel_file>.manifest.xml -a x86_64 -n
mynewkernel
```

For example:

```
euca-bundle-image -i
euca-fedora-10-x86_64/xen-kernel/vmlinuz-2.6.27.21-0.1-xen --kernel
 true --arch x86_64
...
Generating manifest /tmp/vmlinuz-2.6.27.21-0.1-xen.manifest.xml

euca-upload-bundle -b example_kernel_bucket -m
/tmp/vmlinuz-2.6.27.21-0.1-xen.manifest.xml
...
Uploaded image as
example_kernel_bucket/vmlinuz-2.6.27.21-0.1-xen.manifest.xml

euca-register
example_kernel_bucket/vmlinuz-2.6.27.21-0.1-xen.manifest.xml -a
x86_64 -n mynewkernel
IMAGE eki-XXXXXXXX
```

Where the returned value `eki-XXXXXXXX` is the unique ID of the registered kernel image.

Add a Ramdisk

When you add a ramdisk to Walrus, you bundle the ramdisk file, upload the file to a bucket in Walrus that you name, and then register the ramdisk with Eucalyptus.

To add a ramdisk to Walrus:

Use the following three commands:

```
euca-bundle-image -i <ramdisk_file> --ramdisk true -r x86_64
euca-upload-bundle -b <ramdisk_bucket> -m /tmp/<ramdisk_file>.manifest.xml
euca-register <ramdisk_bucket>/<ramdisk_file>.manifest.xml -n <name_of_ramdisk>
```

For example:

```
euca-bundle-image -i
euca-fedora-10-x86_64/xen-kernel/initrd-2.6.27.21-0.1-xen
--ramdisk true -r x86_64
```

```
...
Generating manifest /tmp/initrd-2.6.27.21-0.1-xen.manifest.xml

euca-upload-bundle -b example_rd_bucket -m
/tmp/initrd-2.6.27.21-0.1-xen.manifest.xml
...
Uploaded image as
example_rd_bucket/initrd-2.6.27.21-0.1-xen.manifest.xm

euca-register example_rd_bucket/initrd-2.6.27.21-0.1-xen.manifest.xml
 -n mynewramdisk
IMAGE eri-XXXXXXXX
```

Where the returned value `eri-XXXXXXXX` is the unique ID of the registered ramdisk image.

Add a Root Filesystem

When you add a root filesystem to Walrus, you bundle the root filesystem file, upload the file to a bucket in Walrus that you name, and then register the root filesystem with Eucalyptus. The bundle operation can include a registered ramdisk (ERI ID) and a registered kernel (EKI ID). The resulting image will associate the three images.

You can also bundle the root file system independently and associate the ramdisk and kernel with the resulting EMI at run time.

To add a root filesystem to Walrus:

Use the following three commands:

```
euca-bundle-image -i <root_filesystem_file> -r <architecture>
euca-upload-bundle -b <root_filesystem_file_bucket> -m
/tmp/<root_filesystem_file>.manifest.xml
euca-register <root_filesystem_file_bucket>/<root_filesystem_file>.manifest.xml
 -n <rootfs_name> -a <architecture>
```

For example:

```
euca-bundle-image -i euca-fedora-10-x86_64/fedora.10.x86-64.img
--ramdisk eri-722B3CBA --kernel eki-5B3D3859 -r x86_64
...
Generating manifest /tmp/fedora.10.x86-64.img.manifest.xml

euca-upload-bundle -b example_rf_bucket -m
/tmp/fedora.10.x86-64.img.manifest.xml
...
Generating manifest /tmp/fedora.10.x86-64.img.manifest.xml

euca-register example_rf_bucket/fedora.10.x86-64.img.manifest.xml
-n example_rf -a x86_64
IMAGE   emi-XXXXXXXX
```

Where the returned value `emi-XXXXXXXX` is the unique ID of the registered machine image.

Remove an Image

When a new image is uploaded to Eucalyptus, Eucalyptus saves the bundle and the image manifest to a bucket in Walrus. This bucket is set in the Eucalyptus property `walrus.storagedir`. The default value for this property is `/var/lib/eucalyptus/bukkits`.

When you register an image, Eucalyptus creates the actual image file. Both the image manifest and the bundle must remain intact to run as an instance.

Eucalyptus stores images in the path set

To delete an image fully, you must deregister the image and delete the bundle. To successfully remove an image and associated bundle files:

1. Find the image you want to remove.

    ```
    euca-describe-images
    IMAGE   emi-E533392E    alpha/centos.5-3.x86-64.img.manifest.xml    965590394582
    available   public   i386    machine eki-345135C9    eri-C4F135BC
    instance-store
    IMAGE   emi-623C38B0  alpha/ubuntu.9-04.x86-64.img.manifest.xml   965590394582

    available   public   i386    machine eki-E6B13926    eri-94DB3AB9
    instance-store
    ```

2. Note the image file name (for example, emi-623C38B0).

3. Deregister the image.

    ```
    euca-deregister emi-623C38B0
    IMAGE   emi-623C38B0
    ```

4. Delete the bundle.

    ```
    euca-delete-bundle -b alpha -p ubuntu.9-04.x86-64.img
    ```

    > **Tip:** If you accidentally try to delete a bundle for a second time, you might see an error message: `problem parsing: /tmp/centos.5-3.x86-64.img.manifest.xml`. This error only displays if you try to delete a bundle that no longer exists.

When you have finished these steps, display all images to confirm that the image was removed.

```
euca-describe-images
IMAGE   emi-E533392E    alpha/centos.5-3.x86-64.img.manifest.xml    965590394582
available   public   i386    machine eki-345135C9    eri-C4F135BC  instance-store
```

Index