Cloud Test Service Setup Guide

Table of Contents

[Overview: 2](#_Toc345502975)

[Network Infrastructure 2](#_Toc345502976)

[Eucalyptus Cloud Setup 3](#_Toc345502977)

[Cloud Controller Setup 3](#_Toc345502978)

[Node Controller Setup 4](#_Toc345502979)

[OpenStack Cloud Setup 6](#_Toc345502980)

[Controller Node Setup 6](#_Toc345502981)

[Compute Node Setup 9](#_Toc345502982)

[Troubleshooting 11](#_Toc345502983)

[Build Guest Operating Systems 13](#_Toc345502984)

[Redhat/CentOS Linux (kernel version is earlier than 2.6.20) 13](#_Toc345502985)

[SuSE Linux (kernel version is earlier than 2.6.20) 13](#_Toc345502986)

[Linux (kernel version is later than 2.6.20) 14](#_Toc345502987)

[Windows XP and Windows Server 2003 14](#_Toc345502988)

[Windows Server 2008, and 2008 R2 16](#_Toc345502989)

[Synchronizing Time on Windows 19](#_Toc345502990)

[Register Images into Clouds 19](#_Toc345502991)

[Install Agent for Cloud Test Service 19](#_Toc345502992)

[Linux 20](#_Toc345502993)

[Windows 20](#_Toc345502994)

[Install Cloud Test Service 20](#_Toc345502995)

[Linux 20](#_Toc345502996)

[Windows 21](#_Toc345502997)

[Configurations 21](#_Toc345502998)

[Import Images to Cloud Test Service 21](#_Toc345502999)

# Overview:

Cloud Test Service is a dynamic task scheduling framework specially designed for software product development, testing and validation support. With the ability to talk with various public/private cloud solutions such as Amazon AWS, Eucalyptus, OpenStack, etc, Cloud Test Service can efficiently manage physical resource to support various software products in an agile way. Reload your tests to cloud, and share the great experience with Cloud Test Service now!

As a solution built on top of various clouds, Cloud Test Service requires at least 1 public or private cloud which supports EC2 API. Below sections will focus on how to setup such a private cloud (Eucalyptus 1.6.2 or OpenStack) for Cloud Test Service, and how to register such cloud instance into Cloud Test Service.

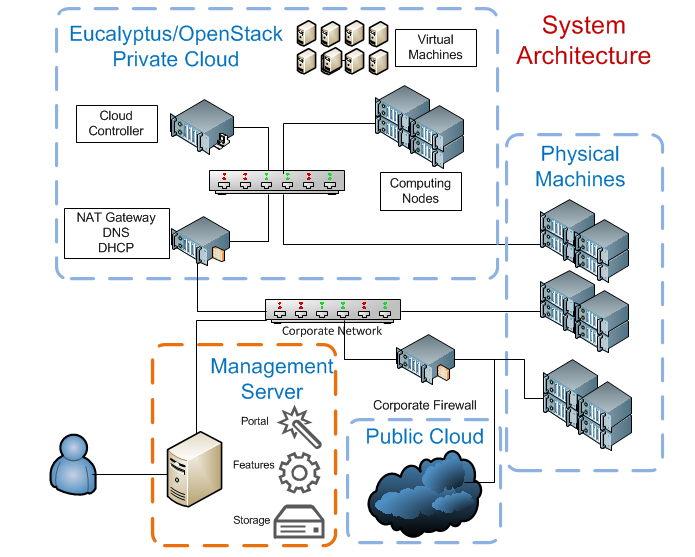
Both the Eucalyptus and OpenStack requires hardware virtualization for best performance. Please enable it in BIOS before installing the cloud management software.

# Network Infrastructure

Cloud Test Service supports multiple connectivity solutions for private and public clouds:

* Direct: Cloud Test Service will directly connect to the Cloud Service and launched Virtual Machines.
* HTTP Proxy: Cloud Test Service will connect to the Cloud Service and Virtual Machines via a configured HTTP Proxy. This mode is typically used for enterprise setup which uses a public cloud such as Amazon AWS, requires corporate proxy connectivity.
* NAT Gateway: Cloud Test Service will automatically manage the NAT Gateway (based on Linux iptables) when connecting to or from Cloud Service and Virtual Machines.

Typically, Cloud Test Service is deployed in the corporate network, and can be configured to manage both private and public clouds together with individual physical machines.



# Eucalyptus Cloud Setup

## Cloud Controller Setup

A typical Eucalyptus deployment is to install the following components: Cloud Controller, Cluster Controller, and Storage Controller on a server with high performance storage attached. The Node Controller as the actual computing node which hosts the virtual machines will be installed on the powerful machines with multi-cores and huge memory.

While Cloud Test Service was developed, Eucalyptus is still at version 1.6.2. Some patches were applied to work with Cloud Test Service and with the stability enhanced. To simplify the installation, a binary package is specially built for Redhat/CentOS 64bit edition. If other OS is required to be the Cloud Controller, please copy the patch source files: ChannelUtil.java, handlers\_kvm.c, and pools.groovy from [\\cloudtest.sh.intel.com\public\cloud\eucalyptus\patch](file:///\\cloudtest.sh.intel.com\public\cloud\eucalyptus\patch) , then replace the original Eucalyptus 1.6.2 source code with above files before building.

The steps to setup the Cloud Controller with the pre-built package is as below:

1. Install Redhat or CentOS 64bit edition. It’s also recommended to install libvirtd.
2. Install AOE and vblade: copy the files: aoe6-75.tar.gz, aoetools-30.tar.gz, vblade-19.tgz from [\\cloudtest.sh.intel.com\public\cloud\eucalyptus\](file:///\\cloudtest.sh.intel.com\public\cloud\eucalyptus\), extract them, and just *make && make install*
3. Install euca tools: copy the file euca2ools-1.2-centos-x86\_64.tar.gz from [\\cloudtest.sh.intel.com\public\cloud\eucalyptus\](file:///\\cloudtest.sh.intel.com\public\cloud\eucalyptus\), extract to a temporal folder, and *rpm -ivh \*.rpm*
4. (Optional) Setup NTP service to synchronize timestamp across the whole cloud. However, this is not a must step as Cloud Test Services has built in wall-clock synchronization mechanism for virtual machines.
5. Create a new user for Eucalyptus: *useradd eucalyptus*
6. Modify */etc/cron.daily/tmpwatch* to prevent removing some Eucalyptus files accidently:

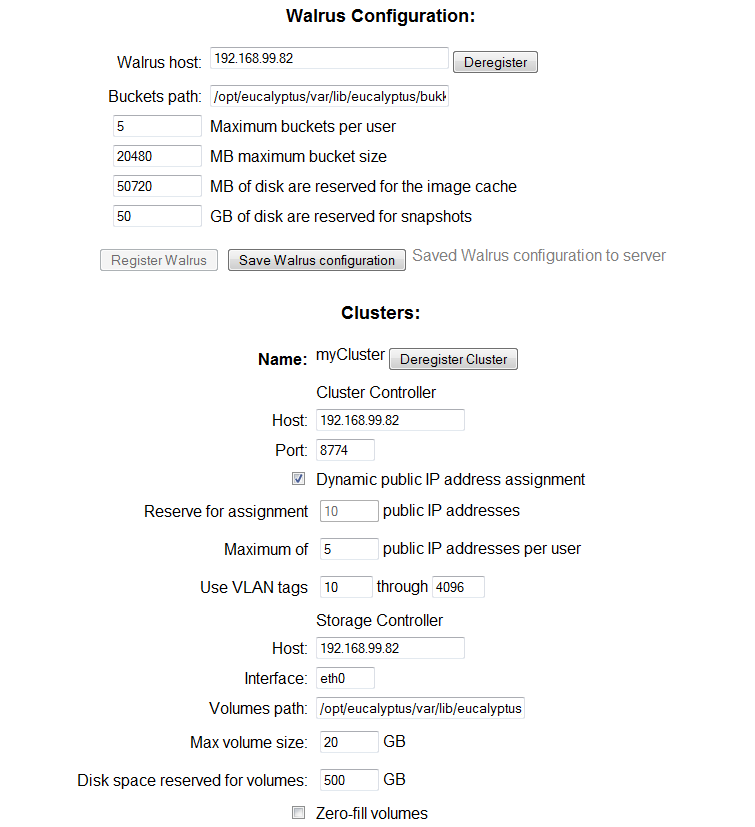
**/usr/sbin/tmpwatch "$flags" -x /tmp/.X11-unix -x /tmp/.XIM-unix \**

**-x /tmp/.font-unix -x /tmp/.ICE-unix -x /tmp/.Test-unix --exclude-user=eucalyptus 240 /tmp**

1. Copy the pre-built package from [\\cloudtest.sh.intel.com\public\cloud\eucalyptus\cedar\_euca1.6.2.tgz](file:///\\cloudtest.sh.intel.com\public\cloud\eucalyptus\cedar_euca1.6.2.tgz), and extract to */opt*
2. Add Eucalyptus executable to PATH environment by adding below to *~/.bashrc*: **export PATH=/opt/eucalyptus/usr/sbin:$PATH**
3. Perform the initial setup for Eucalyptus: *euca\_conf --setup*
4. Enable the single installation for CLC, CC,SC, and Walrus:

*euca\_conf --enable cloud && euca\_conf --enable walrus && euca\_conf –enable sc*

1. Start the Eucalyptus services: */opt/eucalyptus/etc/init.d/eucalyptus-cloud start && /opt/eucalyptus/etc/init.d/eucalyptus-cc start*
2. Wait for about 1 minute so that Eucalyptus administration web console is started up. Then navigate to <https://SERVER:8443>
3. Use ‘admin’ for both username and password to login, and change the default password by following the page.
4. Now switch to ‘Configuration’ tab, and fill the parameters as below example shows:



* Make sure set appropriate walrus and storage settings, e.g. max volume size is recommended to be 20GB
* The Host of Walrus, Storage Controller and Cluster Controller is the same as Cloud Controller
* The Interface of Storage Controller must be the actual interface which connects to the network

1. Copy the newly created cluster’s certificate to Eucalyptus installation folder:

*cp /opt/eucalyptus/var/lib/eucalyptus/keys/myCluster/\* /opt/eucalyptus/var/lib/eucalyptus/keys*

1. Now download the credential for euca tools, and check the Eucalyptus installation:

*cd ~*

*mkdir .euca*

*euca\_conf --get-credentials cred.zip*

*unzip cred.zip*

*source ~/.euca/eucarc*

*euca-describe-availablity-zones verbose*

## Node Controller Setup

Node Controller is the actual computing node which hosts all the virtual machines launched by Eucalyptus. After Cloud Controller is deployed and started successfully, we can install and add more Node Controller into Eucalyptus Cluster. Please follow below steps:

* 1. It’s recommended to use [Ubuntu Enterprise Server 10.04 LTS 64bit edition](http://www.ubuntu.com/start-download?distro=server&bits=64&release=lts), and select Eucalyptus Node Controller option to install.
  2. UEC 10.04 uses the EXT4 as default file system type. However, it’s confirmed that ext4 causes very bad IO performance together with KVM’s disk driver. We can select to use EXT3 while formatting the disk for installation.
  3. (Optional)To ease successive configuration steps, it’s recommended to reset the password for ‘root’ so that we can login Ubuntu with root:

*sudo password*

* 1. [Important]Disable auto-upgrading feature to prevent Eucalyptus broken issue: edit the file */etc/apt/apt.conf.d/10periodic* if exists to replace with below settings:

**APT::Periodic::Update-Package-Lists “0”;**

**APT::Periodic::Download-Upgradeable-Packages “0”;**

**APT::Periodic::AutocleanInterval “0”;**

And edit the file */etc/apt/apt.conf.d/20auto-upgrades* if exists to replace with below settings:

**APT::Periodic::Update-Package-Lists “0”;**

**APT::Periodic::Unattended-Upgrade “0”;**

* 1. After installation, modify the */etc/eucalyptus/eucalyptus.conf* to enable SYSTEM mode, disable ISCSI, and set log level to “ERROR”:

**DISABLE\_ISCSI=”Y”**

**VNET\_MODE=”SYSTEM”**

**LOGLEVEL=”ERROR”**

* 1. As the Cloud Controller is installed from source, we must create a symbolic to the Ubuntu’s eucalyptus installation, simply do below:

*mkdir –p /opt/eucalyptus/var/lib/eucalyptus*

*cd /opt/eucalyptus/var/lib/eucalyptus/*

*ln –sf /var/lib/eucalyptus/keys keys*

* 1. Copy the gen\_kvm\_libvirt\_xml from [\\cloudtest.sh.intel.com\public\cloud\eucalyptus\patch\ gen\_kvm\_libvirt\_xml](file:///\\cloudtest.sh.intel.com\public\cloud\eucalyptus\patch\libEucalyptusNC.so) to replace the original file in */usr/share/eucalyptus/*
  2. Copy the pre-built binary from [\\cloudtest.sh.intel.com\public\cloud\eucalyptus\patch\libEucalyptusNC.so](file:///\\cloudtest.sh.intel.com\public\cloud\eucalyptus\patch\libEucalyptusNC.so) to work around KVM driver issues:

*/etc/init.d/eucalyptus-nc stop*

*cp libEucalyptusNC.so /usr/lib/axis2/services/EucalyptusNC/*

*ln –sf /usr/lib/libcurl-gnutls.so.3 /usr/lib/libcurl.so.3*

*/etc/init.d/eucalyptus-nc start*

* 1. (Optional) As Eucalyptus keeps local virtual machine images on the Node Controller, we can maximize virtual machine startup performance by copying existent virtual machine images to the newly installed Node Controller. Execute below commands:

*scp -r OLD\_NODE\_CONTROLLER\_IP:/var/lib/eucalyptus/instances/eucalyptus/cache/\* /var/lib/eucalyptus/instances/eucalyptus/cache/*

*chown –R eucalyptus.eucalyptus /var/lib/eucalyptus/instances/eucalyptus/cache/\**

* 1. (Optional) Cloud Test Service is able to expose the graphical console of the launched virtual machine for users, in addition to the built-in terminal console output provided by Eucalyptus. Follow the [steps](#_Linux) to install and enable Agent software of Cloud Test Service on the Node Controller.
  2. Logon to the Cloud Controller machine, and use below command to register the newly installed Node Controller into the Cloud:

*euca\_conf –-register-nodes “NODE\_CONTROLLER\_IP”*

* 1. Wait for about 1 minute, execute *euca\_conf --list-nodes*, if above steps are correct, you can see that there’s 1 more node added to Eucalyptus cloud successfully. And also check the logs of Cloud Controller (typically in *$EUCALYPTUS\_ROOT/var/log/eucalyptus/cc.log*) and Node Controller (*typically in /var/log/eucalyptus/nc.log*) that there’re no configuration errors.
  2. Next step is to create guest OS image, and register to Eucalyptus cloud. For detailed instructions about registering kernel or image into Eucalyptus, please refer to the online manual. And some kernel and ramdisk files can be obtained from [\\cloudtest.sh.intel.com\public\cloud\eucalyptus](file:///\\cloudtest.sh.intel.com\public\cloud\eucalyptus) . Typically, these euca tools is used to register kernel, ramdisk and images: *euca-bundle-image, euca-upload-bundle, euca-register*

# OpenStack Cloud Setup

Cloud Test Service supports FlatNetwork mode of OpenStack, and can successfully manage any OpenStack setup with EC2 compatible layer enabled. Below sections illustrate the installation and configuration steps for a typical OpenStack (Essex release) on Fedora (release 17), and the Controller Node is configured to work as a Compute Node too.

All OpenStack Nodes are connected to 2 networks, one is the public network for management, and another is the private network for launched Virtual Machines. It’s not required to configure IP address for the Nodes’ interfaces in private network. OpenStack Network component will automatically manage the bridging for both Virtual Machines and Nodes.

For other Linux distributions, refer to the corresponding documents.

## Controller Node Setup

During installing Fedora 17, an LVM Volume Group called nova-volumes could be created, and it’s required for OpenStack volume service. This LVM Volume Group could be created on any storage device.

After Fedora 17 is installed, it’s recommended to disable both SELinux and Firewall. And perform below steps to accomplish Controller Node setup. (Or follow the instructions in [here](https://fedoraproject.org/wiki/Getting_started_with_OpenStack_on_Fedora_17), as reference)

1. Login and enable HTTP proxy if required(below commands assume login as root)
2. Configure Node hostname by editing /etc/sysconfig/network
3. Enable updates-testing repository

*vi /etc/yum.repos.d/fedora-updates-testing.repo*

[updates-testing]

name=Fedora $releasever - $basearch - Test Updates

failovermethod=priority

#baseurl=http://download.fedoraproject.org/pub/fedora/linux/updates/testing/$releasever/$basearch/

mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=updates-testing-f$releasever&arch=$basearch

enabled=1

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch

1. Configure NTP service

*yum –y install ntpdate ntp*

Comment out existent NTP server lines, and add below line in the file /etc/ntp.conf, this is to synchronize with an external NTP server

*server 192.168.10.1 iburst*

Enable NTP Service:

*systemctl enable ntpd.service && systemctl start ntpd.service*

1. Install OpenStack components:

*yum install --enablerepo=updates-testing openstack-utils openstack-nova openstack-glance openstack-keystone openstack-dashboard qpid-cpp-server-daemon*

1. Run the helper script to get MySQL configured for use with openstack. If mysql-server is not already installed, this script will install it for you:

*openstack-db --service nova –init*

*openstack-db --service glance --init*

1. Enable the start required services:

*systemctl start qpidd.service && systemctl enable qpidd.service*

*systemctl start libvirtd.service && systemctl enable libvirtd.service*

*for svc in api registry; do systemctl start openstack-glance-$svc.service; done*

*for svc in api registry; do systemctl enable openstack-glance-$svc.service; done*

*systemctl start tgtd.service && systemctl enable tgtd.service*

1. Enable and start OpenStack services

*openstack-config --set /etc/nova/nova.conf DEFAULT libvirt\_inject\_partition -1*

*openstack-config --set /etc/nova/nova.conf DEFAULT libvirt\_disk\_prefix hd*

*for svc in api objectstore compute network volume scheduler cert; do systemctl start openstack-nova-$svc.service; done*

*for svc in api objectstore compute network volume scheduler cert; do systemctl enable openstack-nova-$svc.service; done*

1. Configure quota limitation for OpenStack:

*openstack-config --set /etc/nova/nova.conf DEFAULT quota\_instances 30*

*openstack-config --set /etc/nova/nova.conf DEFAULT quota\_floating\_ips 40*

*openstack-config --set /etc/nova/nova.conf DEFAULT* *quota\_cores 100*

*openstack-config --set /etc/nova/nova.conf DEFAULT quota\_ram 65536*

1. Configure Keystone for OpenStack

*openstack-db --service keystone –init*

*cat > keystonerc <<EOF*

*export ADMIN\_TOKEN=$(openssl rand -hex 10)*

*export OS\_USERNAME=admin*

*export OS\_PASSWORD=changeit*

*export OS\_TENANT\_NAME=admin*

*export OS\_AUTH\_URL=http://127.0.0.1:5000/v2.0/*

*EOF*

*source ./keystonerc*

*openstack-config --set /etc/keystone/keystone.conf DEFAULT admin\_token $ADMIN\_TOKEN*

*systemctl start openstack-keystone.service && systemctl enable openstack-keystone.service*

*ADMIN\_PASSWORD=$OS\_PASSWORD SERVICE\_PASSWORD=servicepass openstack-keystone-sample-data*

*keystone user-list #test it*

1. Integrate Keystone with OpenStack

*openstack-config --set /etc/nova/api-paste.ini filter:authtoken admin\_tenant\_name service*

*openstack-config --set /etc/nova/api-paste.ini filter:authtoken admin\_user nova*

*openstack-config --set /etc/nova/api-paste.ini filter:authtoken admin\_password servicepass*

*openstack-config --set /etc/nova/nova.conf DEFAULT auth\_strategy keystone*

1. Enable and start KeyStone service

*for svc in api compute; do systemctl restart openstack-nova-$svc.service; done*

*nova flavor-list #test it*

1. Integrate Keystone with glance

*openstack-config --set /etc/glance/glance-api.conf paste\_deploy flavor keystone*

*openstack-config --set /etc/glance/glance-registry.conf paste\_deploy flavor keystone*

*openstack-config --set /etc/glance/glance-api-paste.ini filter:authtoken admin\_tenant\_name service*

*openstack-config --set /etc/glance/glance-api-paste.ini filter:authtoken admin\_user glance*

*openstack-config --set /etc/glance/glance-api-paste.ini filter:authtoken admin\_password servicepass*

*openstack-config --set /etc/glance/glance-registry-paste.ini filter:authtoken admin\_tenant\_name service*

*openstack-config --set /etc/glance/glance-registry-paste.ini filter:authtoken admin\_user glance*

*openstack-config --set /etc/glance/glance-registry-paste.ini filter:authtoken admin\_password servicepass*

*systemctl restart openstack-glance-api.service*

*systemctl restart openstack-glance-registry.service*

1. Configure networking for OpenStack, below command will create a FlatNetwork which can support 1024 instances.

*nova-manage network create novanet 172.16.0.0/22 1 1024 --bridge=novanetbr0* *--bridge\_interface=em2*

*openstack-config --set /etc/nova/nova.conf DEFAULT public\_interface em1*

*openstack-config --set /etc/nova/nova.conf DEFAULT auto\_assign\_floating\_ip True*

*systemctl restart openstack-nova-network.service*

*nova-manage floating create 192.168.200.0/22*

1. Enable VNC access

*openstack-config --set /etc/nova/nova.conf DEFAULT vnc\_enabled True*

*openstack-config --set /etc/nova/nova.conf DEFAULT novnc\_enabled True*

*openstack-config --set /etc/nova/nova.conf DEFAULT vncserver\_listen 0.0.0.0*

1. [Important]Copy over /usr/share/nova/libvirt.xml.template from [\\cloudtest.sh.intel.com\public\cloud\openstack\patch](file:///\\cloudtest.sh.intel.com\public\cloud\openstack\patch)
2. [Important]Copy over /usr/lib/python2.7/site-packages/nova/manager.py from [\\cloudtest.sh.intel.com\public\cloud\openstack\patch](file:///\\cloudtest.sh.intel.com\public\cloud\openstack\patch)\python\nova\manager.py
3. [Important]Copy over /usr/lib/python2.7/site-packages/nova/compute/manager.py from [\\cloudtest.sh.intel.com\public\cloud\openstack\patch](file:///\\cloudtest.sh.intel.com\public\cloud\openstack\patch)\python\nova\compute\manager.py
4. Restart all service:

*for svc in api objectstore compute network volume scheduler cert; do systemctl restart openstack-nova-$svc.service; done*

1. Create some flavors (Machine Types):

*nova flavor-create --ephemeral 0 c1.small 6 2048 20 2*

*nova flavor-create --ephemeral 10 c1.medium 7 2048 40 2*

*nova flavor-create --ephemeral 20 c1.large 8 4096 40 4*

*nova flavor-create --ephemeral 100 t1.xlarge 9 8192 20 4*

1. Add some firewall rules to OpenStack security group:

*ifconfig novanetbr0 promisc*

*nova secgroup-add-rule default tcp 1 65535 0.0.0.0/0*

*nova secgroup-add-rule default udp 1 65535 0.0.0.0/0*

## Compute Node Setup

After Fedora 17 is installed, it’s recommended to disable both SELinux and Firewall. And perform below steps to accomplish Compute Node setup. (Or follow the instructions in [here](https://fedoraproject.org/wiki/Getting_started_with_OpenStack_on_Fedora_17)). Here we assume Controller Node is os-controller, and its IP is 192.168.20.1, and below commands are executed as root.

To install the latest packages from external repository, configure the system as below:

* Login and enable HTTP and HTTPS proxy if required, e.g:

*export http\_proxy=http://proxy01.cd.intel.com:911*

*export https\_proxy=http://proxy01.cd.intel.com:911*

* Configure Node hostname by editing /etc/sysconfig/network
* Enable updates-testing repository

*vi /etc/yum.repos.d/fedora-updates-testing.repo*

[updates-testing]

name=Fedora $releasever - $basearch - Test Updates

failovermethod=priority

#baseurl=http://download.fedoraproject.org/pub/fedora/linux/updates/testing/$releasever/$basearch/

mirrorlist=https://mirrors.fedoraproject.org/metalink?repo=updates-testing-f$releasever&arch=$basearch

enabled=1

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$basearch

* *yum distribution-synchronization*

To install the cached packages from internal repository:

* Configure Node hostname by editing /etc/sysconfig/network
* Remove all existent .repo files in /etc/yum.repos.d/
* Create local.repo in /etc/yum.repos.d/ with below content:

[fedora]

name=Fedora $releasever - $basearch

baseurl=ftp://192.168.20.22/pub

enabled=1

gpgcheck=0

[local]

name=Fedora Local

baseurl=http://192.168.10.1/install/fedora17

enabled=1

gpgcheck=0

* *yum distribution-synchronization*

1. Configure NTP service

*yum -y install ntpdate ntp*

Comment out existent NTP server lines, and add below line in the file /etc/ntp.conf, this is to synchronize with the OpenStack Controller node.

*server os-controller iburst*

Enable NTP Service:

*systemctl enable ntpd.service && systemctl start ntpd.service*

1. Install OpenStack components:

*yum -y install --enablerepo=updates-testing openstack-nova*

*systemctl start libvirtd.service && systemctl enable libvirtd.service*

1. Configure OpenStack Compute on this node

*openstack-config --set /etc/nova/nova.conf DEFAULT qpid\_hostname os-controller*

*openstack-config --set /etc/nova/nova.conf DEFAULT sql\_connection mysql://nova:nova@os-controller/nova*

*openstack-config --set /etc/nova/nova.conf DEFAULT glance\_api\_servers os-controller:9292*

*openstack-config --set /etc/nova/nova.conf DEFAULT iscsi\_ip\_prefix 192.168.20.1*

*openstack-config --set /etc/nova/nova.conf DEFAULT libvirt\_inject\_partition -1*

*openstack-config --set /etc/nova/nova.conf DEFAULT libvirt\_disk\_prefix hd*

*openstack-config --set /etc/nova/nova.conf DEFAULT auth\_strategy keystone*

1. Enable VNC access

*openstack-config --set /etc/nova/nova.conf DEFAULT vnc\_enabled True*

*openstack-config --set /etc/nova/nova.conf DEFAULT novnc\_enabled True*

*openstack-config --set /etc/nova/nova.conf DEFAULT vncserver\_listen 0.0.0.0*

1. [Important]Copy over /usr/share/nova/libvirt.xml.template from [\\cloudtest.sh.intel.com\public\cloud\openstack\patch](file:///\\cloudtest.sh.intel.com\public\cloud\openstack\patch)
2. [Important]Copy over /usr/lib/python2.7/site-packages/nova/manager.py from [\\cloudtest.sh.intel.com\public\cloud\openstack\patch](file:///\\cloudtest.sh.intel.com\public\cloud\openstack\patch)\python\nova\manager.py
3. [Important]Copy over /usr/lib/python2.7/site-packages/nova/compute/manager.py from [\\cloudtest.sh.intel.com\public\cloud\openstack\patch](file:///\\cloudtest.sh.intel.com\public\cloud\openstack\patch)\python\nova\compute\manager.py
4. Update quota limitation with appropriate values by executing below commands on Controller Node, **and use appropriate limit values**:

*openstack-config --set /etc/nova/nova.conf DEFAULT quota\_instances 40*

*openstack-config --set /etc/nova/nova.conf DEFAULT quota\_floating\_ips 50*

*openstack-config --set /etc/nova/nova.conf DEFAULT* *quota\_cores 130*

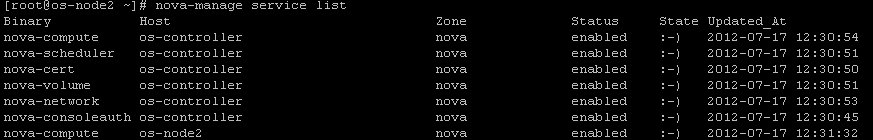
*openstack-config --set /etc/nova/nova.conf DEFAULT quota\_ram 90536*

**After updating limitation, the openstack api service must be restarted on Controller Node.**

1. Enable and start OpenStack Compute service

*systemctl enable openstack-nova-compute.service && systemctl start openstack-nova-compute.service*

1. Use “nova-manage service list” to check status:



## Troubleshooting

* To disable Firewall on Fedora, just use ‘setup’ tool. If it doesn’t exist, use below commands to install:

*yum -y install setuptool system-config-firewall-tui*

Another quick way is to disable below two lines in /etc/sysconfig/iptables and reboot



* When nova-network is hanging to startup, please check that there’s no \*.lock file in /var/lib/nova/tmp/
* To force renaming the network device to em1, and em2 (which are configured for OpenStack), create a file named 70-persistent-net.rules under /etc/udev/rules.d with below content:

SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?\*", ATTR{address}=="00:11:22:33:44:55", ATTR{type}=="1", KERNEL=="eth\*", NAME="em1"

SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?\*", ATTR{address}=="66:77:88:99:00:00", ATTR{type}=="1", KERNEL=="eth\*", NAME="em2"

* Due to an known OpenStack issue([Bug# 1014769](https://bugs.launchpad.net/nova/+bug/1014769)), Instances fail to launch with the error: No More Fixed Ip. To resolve this issue, apply the patch before restarting openstack-network-service. And perform below operations to recover from failure:

Login to mysql with nova/nova, and use the nova database, then:

>update fixed\_ips set instance\_id=null where allocated=0 and instance\_id is not null;

* When OpenStack is installed in a private NAT based network, it’s required to add a DNAT rule for the Controller Node such as:

*iptables -t nat -A PREROUTING -p tcp --dport 19999 -i eth1 -j DNAT --to 192.168.20.1:8773*

And below rules are also required (assuming eth1 is the public interface, and all Virtual Machines are assigned 192.168.x.x IP address):

*iptables -t nat -A POSTROUTING -s 192.168.0.0/16 -o eth1 -j MASQUERADE*

*echo 1 > /proc/sys/net/ipv4/ip\_forward*

* Sometimes there’s security enforced for enterprise that traffic is only allowed via HTTP proxy (assuming 10.x.x.x is only allowed network for all the Virtual Machines, and the private interface is eth0):

*iptables -A FORWARD -p tcp -m tcp -d 10.0.0.0/8 --dport 445 -j DROP*

*iptables -A FORWARD -i eth0 -d 10.0.0.0/8 -j ACCEPT*

*iptables -A FORWARD -i eth0 -j DROP*

*iptables -A OUTPUT -p tcp -m tcp -d 10.0.0.0/8 --dport 445 -j DROP*

*iptables -A OUTPUT -j ACCEPT -d 10.0.0.0/8*

*iptables -A OUTPUT -j ACCEPT -d 192.168.0.0/16*

*iptables -A OUTPUT -j DROP -d 0.0.0.0/0*

* The OpenStack controller node must be configured to open the 8773 port for EC2 compatible layer and 123 udp port for NTP. To simplify configuration, make sure there’s no REJCT rule as blow exists in iptables:

*REJECT all -- anywhere anywhere*

If it exists, just remove it by *iptables –D INPUT rule\_index*

* Configure Ganglia monitoring for cluster nodes

*yum install -y ganglia-gmond*

*chkconfig gmond on*

Modify /etc/ganglia/gmond.conf to contain below values:

send\_metadata\_interval = 10 /\*secs \*/

…..

cluster {

name = "Cloud Test Service"

owner = "IDH QA Team"

…..

#mcast\_join = 239.2.11.71

host = 192.168.10.1

…..

*/etc/init.d/gmond start*

Then check <http://cts-gateway.sh.intel.com/ganglia>

* Disaster Recovery Process can be found at[*http://docs.openstack.org/trunk/openstack-compute/admin/content/nova-disaster-recovery-process.html*](http://docs.openstack.org/trunk/openstack-compute/admin/content/nova-disaster-recovery-process.html)*)*

# Build Guest Operating Systems

To enable networking and volume attaching for Windows XP/2003/2008 RedHat and SuSE on IA for Eucalyptus 1.6.2, the e1000 and virtio block drivers are required.

Cloud Test Service has a specially designed agent application to accomplish powerful features such as task dispatch, dynamic resource balancing, and monitoring. It’s recommended to install this agent application after the guest operating system is installed.

Below sections listed the install and configuration steps for all supported OS with the agent application.

## Redhat/CentOS Linux (kernel version is earlier than 2.6.20)

After installing the base operating system, follow below steps to include acpiphp, and virtio\_blk in the ramdisk, and enable dynamic hostname assignment:

1. *#mkinitrd –v –f /boot/initrd-KERNEL-VERSION.img --with acpiphp --with virtio\_blk KERNEL-VERSION*
2. Use euca tools to register the kernel and initrd.
3. Remove HWADDR line in */etc/sysconfig/network-scripts/ifcfg-eth0*
4. (Optional)Install VNC server software, and configure to run at boot
5. [Follow](#_Linux) the steps to install the agent application for Cloud Test Service.
6. Copy rc.sysinit file from [\\cloudtest.sh.intel.com\public\support\_tools\RedHat\rc.sysinit](file:///\\cloudtest.sh.intel.com\public\support_tools\RedHat\rc.sysinit) to replace */etc/rc.d/rc.sysinit,* this is to automatically assign a unique hostname after Linux starts up. This is required if the launched Virtual Machine cannot obtain a hostname from DHCP.

## SuSE Linux (kernel version is earlier than 2.6.20)

After installing the base operating system, follow below steps to include acpiphp, and virtio\_blk in the ramdisk, and enable dynamic hostname assignment:

1. Modify /etc/sysconfig/kernel to append *acpiphp, virtio\_blk* to the MODULES, and execute *mkinitrd* to generate the ramdisk
2. Use euca tools to register the kernel and initrd.
3. (Optional)Install VNC server software, and configure to run at boot
4. [Follow](#_Linux) the steps to install the agent application for Cloud Test Service.
5. Copy boot.localnet file from [\\cloudtest.sh.intel.com\public\support\_tools\SuSE\boot.localnet](file:///\\cloudtest.sh.intel.com\public\support_tools\SuSE\boot.localnet) to replace */etc/rc.d/boot.localnet* , this is to automatically assign a unique hostname after Linux starts up. This is required if the launched Virtual Machine cannot obtain a hostname from DHCP.
6. The network persistent name udev rule must be disabled to avoid changing interface name when difference MAC address is assigned by Eucalyptus.
7. Rename the */etc/sysconfig/network/ifcfg-eth-XXX* to */etc/sysconfig/network/ifcfg-eth0*.

## Linux (kernel version is later than 2.6.20)

All Linux virtual machines which uses the kernel with the version later than 2.6.20 could share the same kernel and ramdisk. Before building Windows images, register the kernel and ramdisk to Eucalyptus Cloud. Please copy them from [\\cloudtest.sh.intel.com\public\cloud\eucalyptus\linux\_kernel](file:///\\cloudtest.sh.intel.com\public\cloud\eucalyptus\linux_kernel) and use euca tools to register them, then follow below instructions:

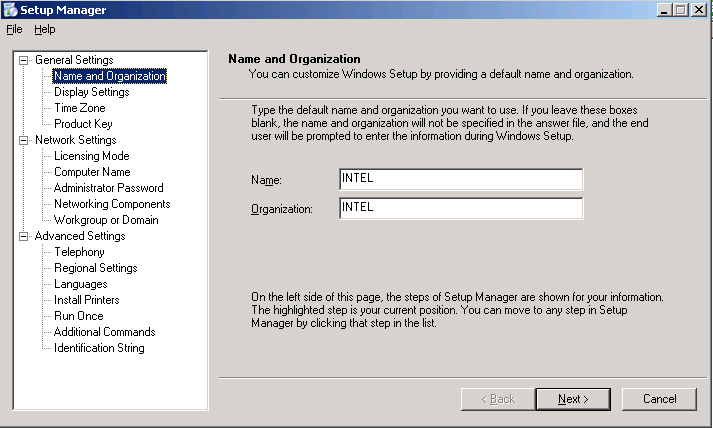
1. (Optional)Install VNC server software, and configure to run at boot
2. Remove HWADDR line in */etc/sysconfig/network-scripts/ifcfg-eth0* (This is for RedHat/CentOS, please refer to distribution document to accomplish this task for other Linux versions)
3. [Follow](#_Linux) the steps to install the agent application for Cloud Test Service.
4. Refer to rc.sysinit file from [\\cloudtest.sh.intel.com\public\support\_tools\RedHat\rc.sysinit](file:///\\cloudtest.sh.intel.com\public\support_tools\RedHat\rc.sysinit) to modify */etc/rc.d/rc.sysinit,* this is to automatically assign a unique hostname after Linux starts up. (This is for RedHat/CentOS, please refer to distribution document to accomplish this task for other Linux versions)
5. The network persistent name udev rule must be disabled by removing all items in */etc/udev/rules.d/70-persistent-net.rules* to avoid changing interface name when difference MAC address is assigned by Cloud.
6. Rename the */etc/sysconfig/network/ifcfg-eth-XXX* to */etc/sysconfig/network/ifcfg-eth0*. (This is for SuSE only)
7. Add firewall rules to allow VNC (5901), and CloudTestService agent(10614) ports

## Windows XP and Windows Server 2003

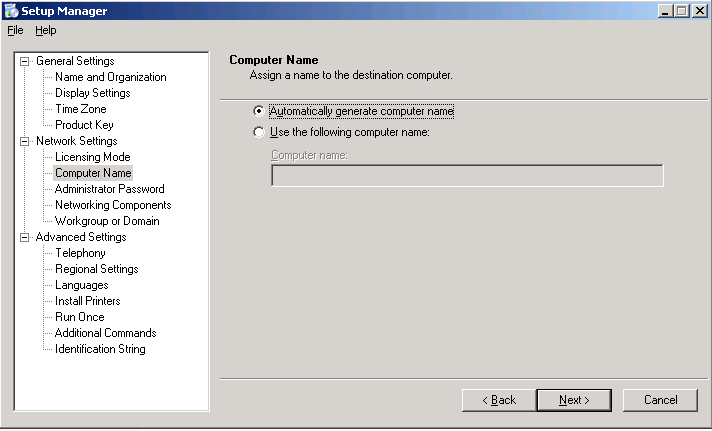
All the Windows virtual machines could share the same kernel and ramdisk. Before building Windows images, register the kernel and ramdisk to Eucalyptus Cloud. Please copy them from [\\cloudtest.sh.intel.com\public\cloud\eucalyptus\win\_kernel](file:///\\cloudtest.sh.intel.com\public\cloud\eucalyptus\win_kernel) and use euca tools to register them.

After installing the base operating system, follow below steps to install Intel e1000 networking driver and RedHat virtio\_block device driver, then generate a deployable image with sysprep tool:

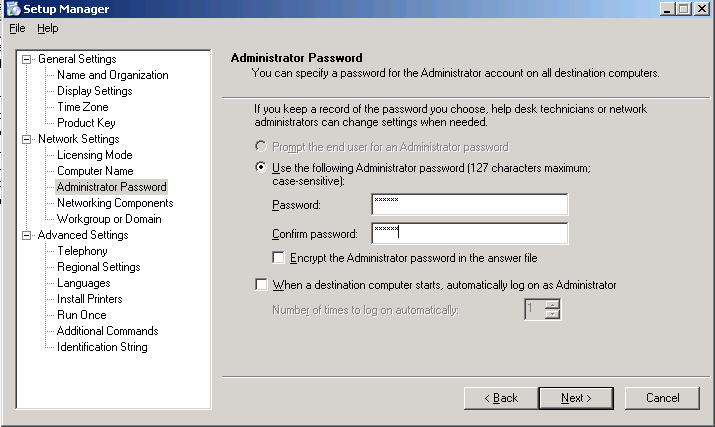
1. Install Intel e1000 networking driver from [\\cloudtest.sh.intel.com\public\support\_tools\Windows](file:///\\cloudtest.sh.intel.com\public\support_tools\Windows)
2. Install RedHat virtio\_block device driver from [\\cloudtest.sh.intel.com\public\support\_tools\RHEV\_Drivers](file:///\\cloudtest.sh.intel.com\public\support_tools\RHEV_Drivers)
3. (Optional)Enable Remote Desktop Connection
4. [Follow](#_Windows) the steps to install the agent application for Cloud Test Service, and configure exception rule (TCP: 10614) if Windows Firewall is enabled.
5. Install sysprep tool from Windows installation CD, and start the “Setup Manager”:



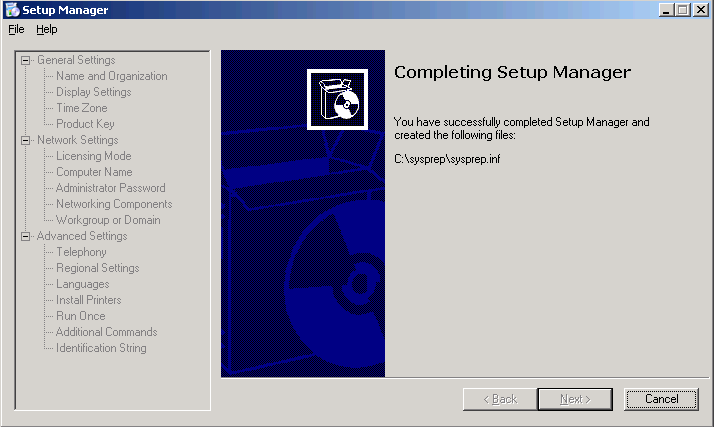
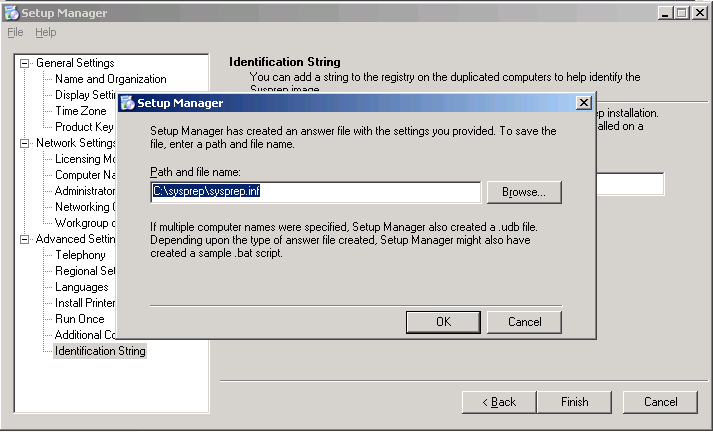
1. Fill the “Product Key”
2. Enable the option to automatically generate computer name:



1. Configure the password for Administrator:

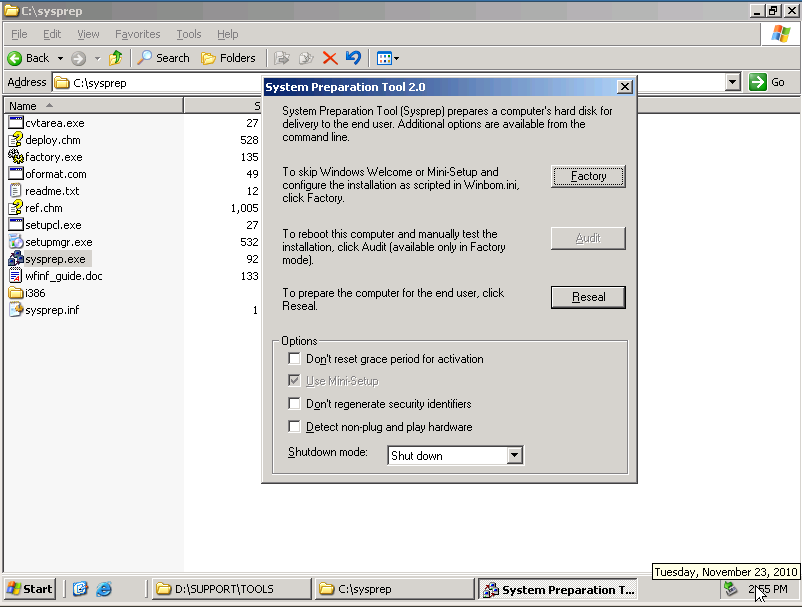
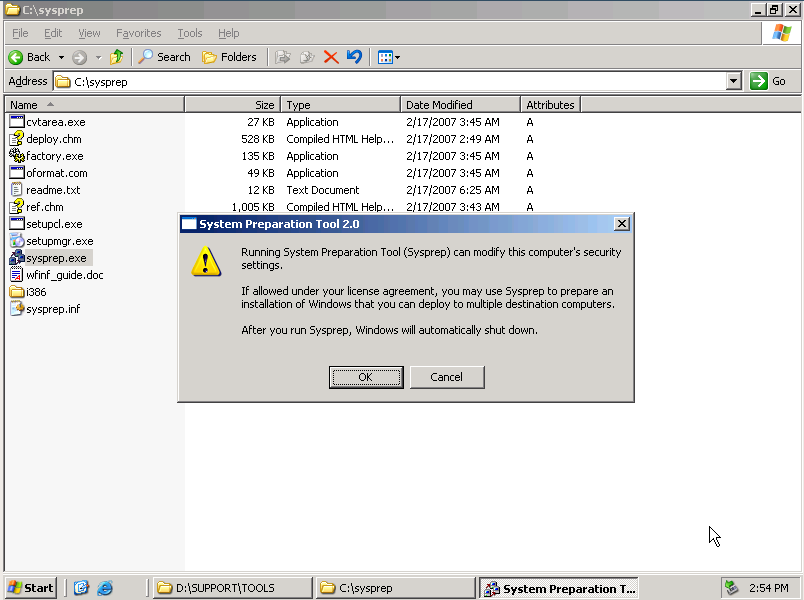


1. Then follow the default options for successive wizard pages:



Click OK to save the sysprep script, then click Cancel to finish.

1. Execute sysprep.exe to perform building:

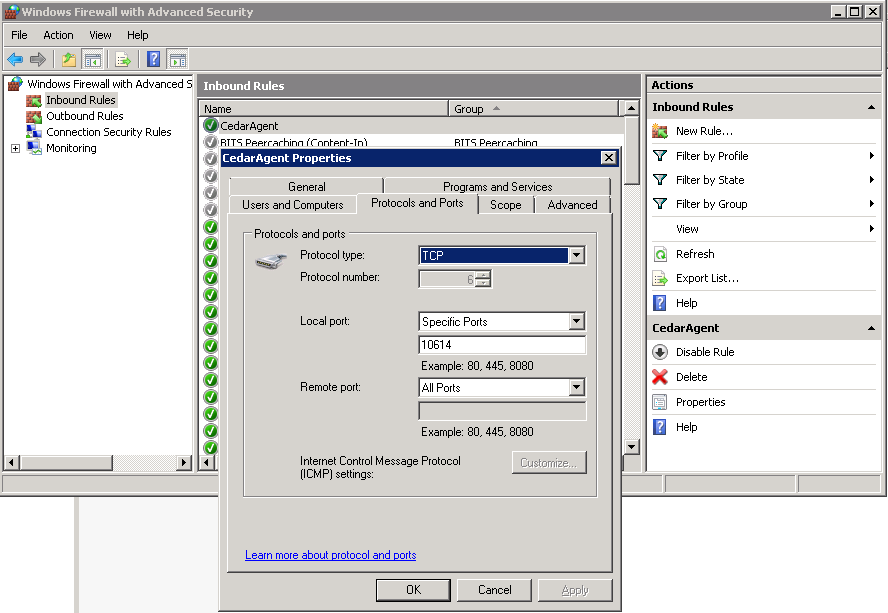


Just click “Reseal” and wait for Windows shutdown

## Windows Server 2008, and 2008 R2

After installing the base operating system, follow below steps to install RedHat virtio\_block device driver, then generate a deployable image with sysprep tool:

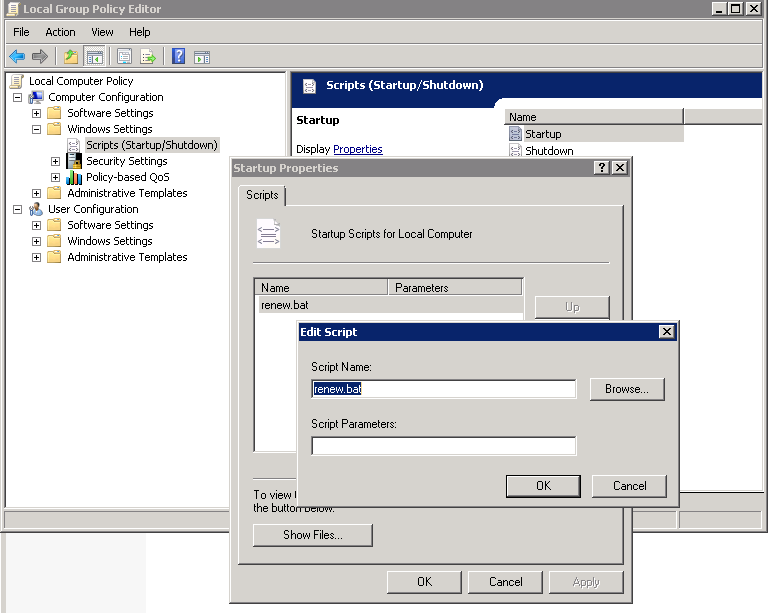
1. Install RedHat virtio\_block device driver from [\\cloudtest.sh.intel.com\public\support\_tools\RHEV\_Drivers](file:///\\cloudtest.sh.intel.com\public\support_tools\RHEV_Drivers)
2. (Optional)Enable Remote Desktop Connection
3. [Follow](#_Windows) the steps to install the agent application for Cloud Test Service, and assign inbound rule as Windows Firewall is enabled.



1. Create a renew.bat file in *C:\Windows\System32\GroupPolicy\Machine\Scripts\Startup* , use Notepad to write below command into it:

**ipconfig /renew**

And then configure it to automatically start when system is booted, type “*gpedit.msc*” in command line and configure as below:



Above step is to make sure that Eucalyptus could detect the virtual machine is started.

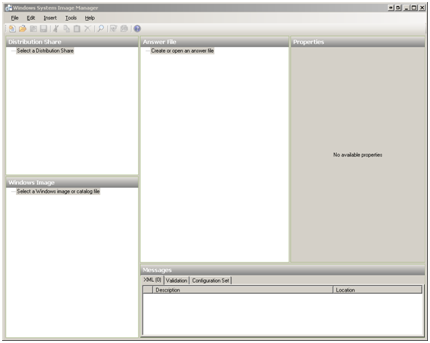
1. (Only Required for Windows 2008 R2) Relax SAN policy for volume functionality of Eucalyptus by executing below commands:

*diskpart*

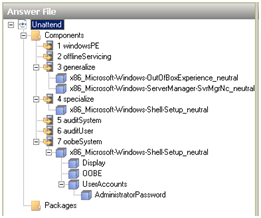
*DISKPART> san policy=OnlineAll*

*DiskPart successfully changed the SAN policy for the current operating system.*

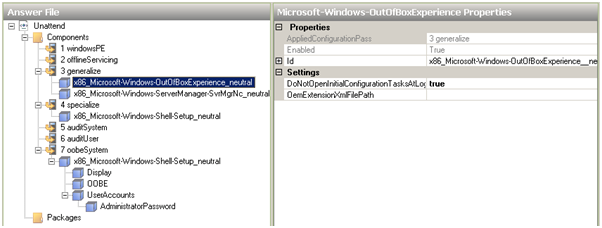
1. Use Windows Automated Installation Kit (WAIK) to create the image. You can get this package in ISO file format from Microsoft's website at <http://www.microsoft.com/downloads/details.aspx?FamilyID=94bb6e34-d890-4932-81a5-5b50c657de08&DisplayLang=en> . The download is about an 800MB install on a Windows Server 2003 SP2 machine.
2. The tool of choice for building your Sysprep configuration is the Windows System Image Manager (WSIM). When you start it you'll get a blank screen like this:



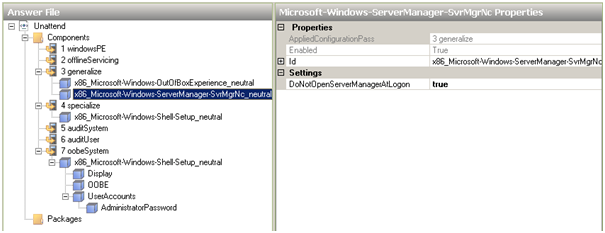
1. The first step is to catalog the image file. You can do this from Tools>Create Catalog, selecting your install.wim and then what image(s) to catalog. Note: You'll also need to copy the install.wim from your installation media DVD sources folder to the hard drive as the tool won't work with it if it doesn't have write access to the WIM file.
2. Create the answer file tree as below:



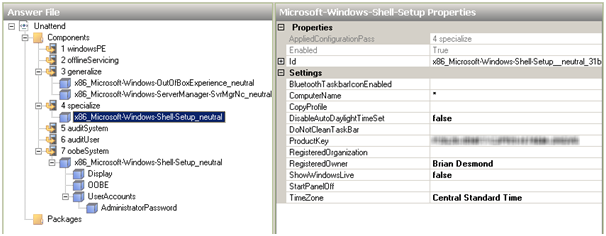
1. Disabling the initial configuration dialog:



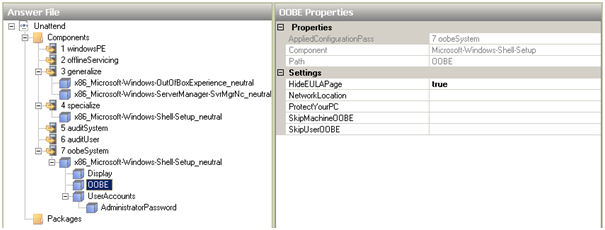
1. Disabling auto-starting the server manager application:



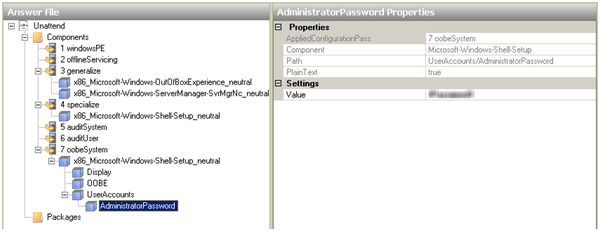
1. Setting my product key, timezone settings, and my name:



1. Configuring setup not to show me the EULA again:



1. Configuring setup to install a default local administrator account password:



1. Save the unattend file into *C:\Windows\System32\sysprep*
2. Run sysprep tool to generate the image:

*sysprep /generalize /oobe /shutdown /unattend:sysprep.xml*

## Synchronizing Time on Windows

net stop w32time

w32tm /config /syncfromflags:manual /manualpeerlist:192.168.10.1,0x8 /update /reliable:yes

net start w32time

# Register Images into Clouds

There’re different ways to register the virtual machine image templates to Clouds. For Eucalyptus, it’s simply to use the client tools (euca-tools) to register the created image file to Eucalyptus Cloud. For OpenStack, it’s recommended to use the native ‘glance’ client to register the images:

*glance add name=IMAGE\_NAME is\_public=true container\_format=bare disk\_format=raw < IMAGE\_FILE*

# Install Agent for Cloud Test Service

Agent is a software application running in privileged level when the virtual machine is started. The agent is controlled by Cloud Test Service to receive and execute tasks, and perform administration configurations such as change the password, and mount a volume automatically.

The agent application requires Java Runtime Environment. It’s recommended to install from [\\cloudtest.sh.intel.com\public\JDK1.6u22](file:///\\cloudtest.sh.intel.com\public\JDK1.6u22) on the guest operating system image. And below are the instructions to install agent application on different operating systems.

## Linux

Follow below steps to install the agent application on a Linux based system:

1. Copy cedar\_agent.tgz from [\\cloudtest.sh.intel.com\public\release](file:///\\cloudtest.sh.intel.com\public\release)
2. *mkdir /opt/cedar*
3. *cd /opt/cedar*
4. *tar zxf /tmp/cedar\_agent.tgz*
5. Consider JDK is installed at /opt, create a soft link to /opt/cedar, such as:

*ln -sf /opt/jdk/jre /opt/cedar/jre*

1. *cd /opt/cedar/startup*
2. *bash install\_agent.sh*
3. */etc/init.d/CedarAgent start*

## Windows

Follow below steps to install the agent application on Windows:

1. Copy cedar\_agent.tgz from [\\cloudtest.sh.intel.com\public\release](file:///\\cloudtest.sh.intel.com\public\release)
2. Create a folder named “cedar” in “C:\”
3. Extract cedar\_agent.tgz into “C:\cedar”
4. Add a system environment named JAVA\_HOME to point to the folder of JDK
5. Goto C:\cedar\startup, and execute “install\_agent.bat”
6. Now the agent is installed as a Windows service and will automatically start when system is booted.
7. To start agent, just type:

*net start CedarAgent*

# Install Cloud Test Service

Cloud Test Service application requires Java Runtime Environment. It’s recommended to install from [\\cloudtest.sh.intel.com\public\JDK1.6u22](file:///\\cloudtest.sh.intel.com\public\JDK1.6u22) on the server. And below are the instructions to install agent application on different operating systems.

## Linux

Follow below steps to install the Cloud Test Service on a Linux based system:

1. Copy cedar\_service.tgz from [\\cloudtest.sh.intel.com\public\release](file:///\\cloudtest.sh.intel.com\public\release)
2. *mkdir /opt/cedar*
3. *cd /opt/cedar*
4. *tar zxf /tmp/cedar\_service.tgz*
5. Consider JDK is installed at /opt, create a soft link to /opt/cedar, such as:

*ln -sf /opt/jdk/jre /opt/cedar/jre*

1. *cd /opt/cedar/startup*
2. *bash install\_service.sh*
3. */etc/init.d/CedarService start*

## Windows

Follow below steps to install Cloud Test Service on Windows:

1. Copy cedar\_service.tgz from [\\cloudtest.sh.intel.com\public\release](file:///\\cloudtest.sh.intel.com\public\release)
2. Create a folder named “cedar” in “C:\”
3. Extract cedar\_service.tgz into “C:\cedar”
4. Add a system environment named JAVA\_HOME to point to the folder of JDK
5. Goto C:\cedar\startup, and execute “install\_service.bat”
6. Now Cloud Test Service is installed as a Windows service and will automatically start when system is booted.
7. To start Cloud Test Service, just type:

*net start CedarService*

## Configurations

Cloud Test Service uses an ini style configuration file. Please modify the options in *cedar/conf/cedar.conf* before starting the service for the first time.

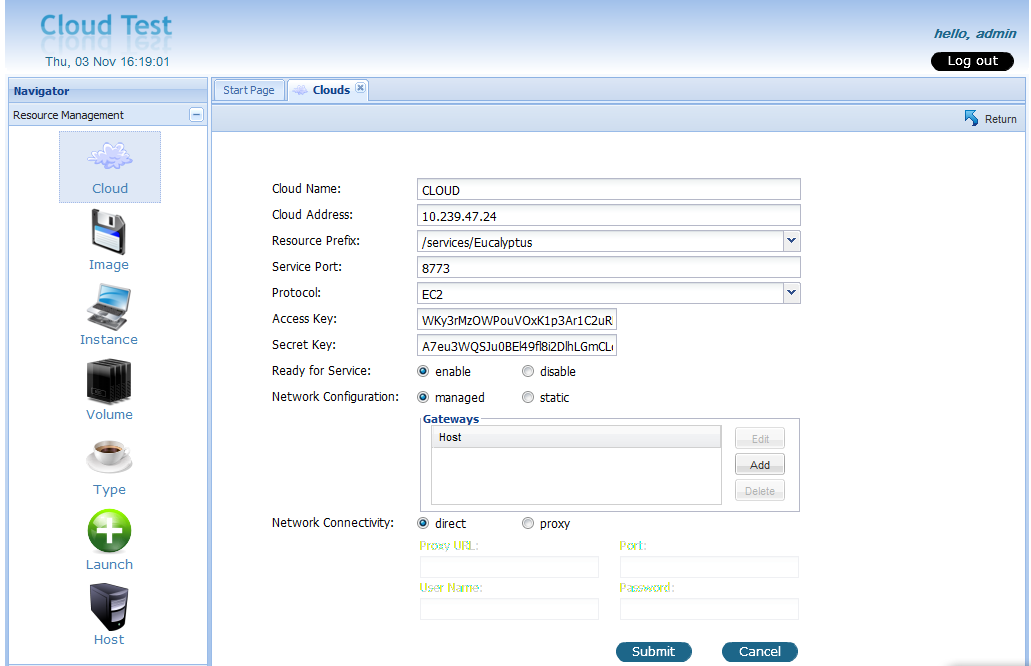
1. SMTPServer: the mail server for Cloud Test Service to send notification emails
2. EnableCloudService: true to allow launching VMs in registered Clouds.
3. EnableTaskService: true to enable Test Services Web UI, and Feature runtime support for Cloud Test Service.
4. MaxStandbyInstances: the maximum number of standby VMs available for Feature jobs.
5. MinAvailInstances: the minimum available instances for Cloud service.

# Import Images to Cloud Test Service

After Cloud Test Service is successfully started, we can login to the management console to import virtual machine images from Eucalyptus Cloud.

Considering Cloud Test Service is installed on SERVER, follow below steps to register a new Eucalyptus cloud and import its virtual machine images:

1. Navigate to <http://SERVER>
2. Use ‘admin’ as username, ‘passwd’ as password to login
3. Click  icon to open the view to manage clouds:

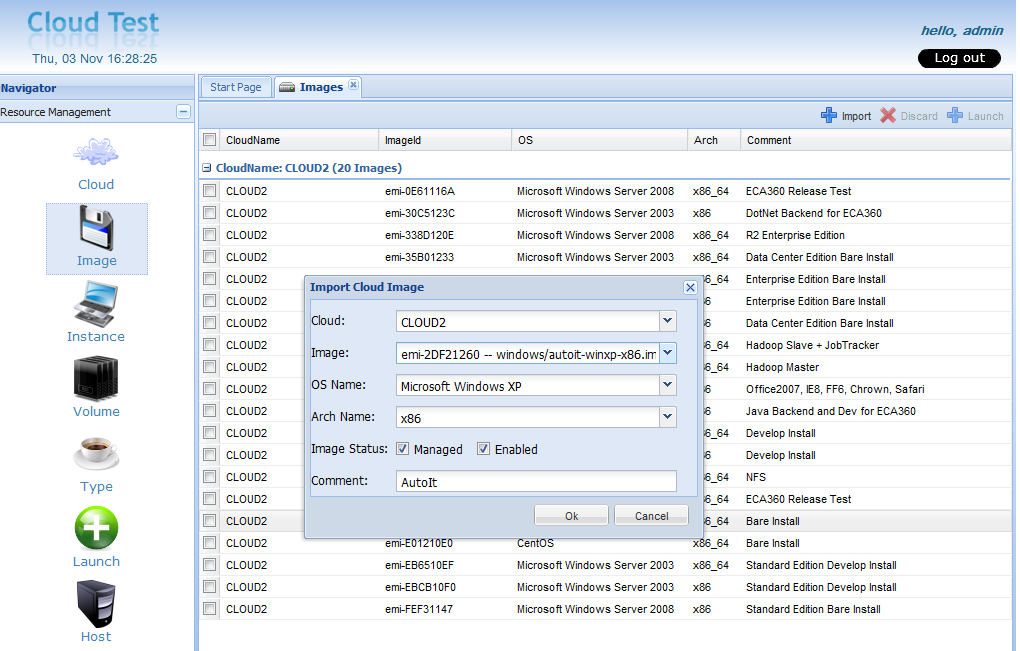


Click  to register a Eucalyptus cloud, in the prompted form, fill the address of the Eucalyptus cloud. For the “Access Key”, and “Secret Key”, they can be found by login to web console of Eucalyptus cloud.

If the SERVER is connected to the same network of Eucalyptus cloud, don’t add a gateway. Otherwise, there must be a gateway (it’s actually the machine which connects multiple networks and provides NAT port forwarding service) configured.

If required, configure the HTTP proxy for Cloud Test Service to connect to the Eucalyptus cloud.

1. Click “Submit” to register the cloud.
2. Click  to open the view to manage the virtual machine images of Eucalyptus cloud
3. Click  to import an existent virtual machine image from Eucalyptus cloud:



Enable “Managed” option only if the selected image has the agent application installed.

Click “OK” to import this image, so that users can start virtual machines of this image.