**Steps to Create the ISO Image**

**Controller ISO:**

We are using the ubuntu-builder to create the custom-iso from the ubuntu-desktop iso.

For the controller iso, we are modifying the ‘ubuntu-12.04-desktop -amd64.iso’

1. Download the ubuntu-12.04-desktop -amd64.iso on the local disk
2. Install the ubuntu-builder

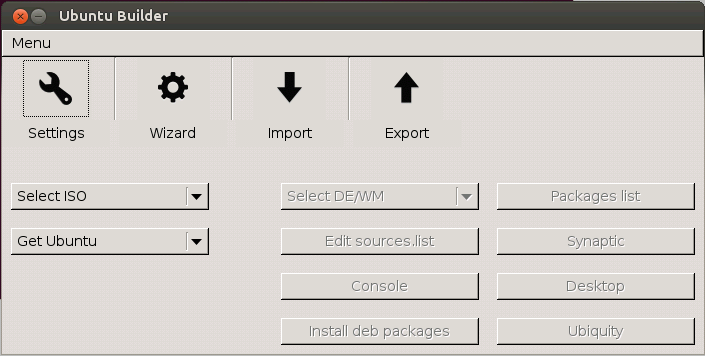
# sudo add-apt-repository ppa:f-muriana/ubuntu-builder

# sudo apt-get update

# sudo apt-get install ubuntu-builder

1. Start the ubuntu-builder

# ubuntu-builder



1. Select the iso (‘Select ISO’) by clicking on ‘From local disk’ option
2. Use the ‘Console’ button to access the command line, console gives you a CLI chroot
3. To install the controller packages, use the following commands

# apt-get install python-software-properties

# add-apt-repository cloud-archive:Havana (If you don’t see any instruction, press ENTER)

# apt-get update

# apt-get install python-mysqldb mysql-server openssh-server vim

# apt-get install rabbitmq-server

# apt-get install keystone

# apt-get install glance python-glanceclient python-cinderclient\* sheepdog\*

# apt-get install nova-novncproxy novnc nova-api nova-ajax-console-proxy nova-cert nova-conductor nova-consoleauth nova-doc nova-scheduler python-novaclient

# apt-get install memcached libapache2-mod-wsgi openstack-dashboard

# apt-get remove --purge openstack-dashboard-ubuntu-theme

1. Edit /etc/mysql/my.cnf and set the bind-address

bind-address = 0.0.0.0

1. Exit from the console

# exit

1. Click on the Build button from the ubuntu-builder UI
2. After completion of the build operation, exit from the ubuntu-builder, get the new modified iso at /home/ubuntu-builder/ Ubuntu-12.04-amd64.sio

**KVM Host ISO:**

For the kvm iso image, we are using the ubuntu-12.04-server-amd64.iso. Here, we are not installing any package manually, script will install all the required packages and will modify the configuration.

**Xen Dom0 ISO:**

For the xen-dom0 iso image, we are using the ubuntu-13.04-server-amd64.iso. Here, scripts will install all the required packages and will modify the configuration.

**Compute Image:**

We have installed VM on Xen and installed required compute packages on that VM. Then exported the VM using xe vm-export utility

Follow the below steps to create a VM using xe and install the compute packages

1. Create VM with Ubuntu Lucid template and later we will set it to raring

# xe vm-install template=Ubuntu\ Lucid\ Lynx\ 10.04\ \(64-bit\) sr-name-label=<sr-name> new-name-label=<vmname>

1. Get UUID of xenbr0 network

# xe network-list bridge=xenbr0 –minimal

1. Create two virtual interfaces with xenbr0 network

# xe vif-create mac=random device=0 network-uuid=<xenbr0-network-uuid> vm-uuid=<vm-uuid>

# xe vif-create mac=random device=1 network-uuid=<xenbr0-network-uuid> vm-uuid=<vm-uuid>

1. Set debian release to raring

# xe vm-param-set other-config:debian-release=raring uuid=<vm-uuid>

1. Set install-repository path

# xe vm-param-set other-config:install-repository=http://archive.ubuntu.com/ubuntu/ uuid=<vm-uuid>

1. Start VM

# xe vm-start uuid=<vm-uuid>

1. Access the VM console

# xe vm-console uuid=<vm-uuid>

1. Set the networking

# vi /etc/network/interfaces

---------------------------------------------------------------------------------------------

# This file describes the network interfaces available on your system

# and how to activate them. For more information, see interfaces(5).

# The loopback network interface

auto lo

iface lo inet loopback

#auto eth1

#iface eth1 inet dhcp

# The primary network interface

auto eth0

iface eth0 inet dhcp

auto eth1

iface eth1 inet manual

------------------------------------------------------------------

1. Install the following packages

# apt-get install nova-compute python-guestfs

# apt-get install nova-network nova-api-metadata

# dpkg-statoverride --update --add root root 0644 /boot/vmlinuz-$(uname -r)

# vi /etc/kernel/postinst.d/statoverride

#!/bin/sh

version="$1"

# passing the kernel version is required

[ -z "${version}" ] && exit 0

dpkg-statoverride --update --add root root 0644 /boot/vmlinuz-${version}

# chmod +x /etc/kernel/postinst.d/statoverride

# rm /var/lib/nova/nova.sqlite

# ip link set eth1 promisc on

1. Shutdown the domU (compute node) instance as follows:

# xe vm-shutdown uuid=<nova-compute-domU-uuid>

1. Export the image as follows:

# xe vm-export uuid=<nova-compute-domU-uuid> filename=/home/<filename> compress=true