**Building libvirt and QEMU from source**

**Overview**

This document describes the installation procedure of libvirt/QEMU from source.

**Version information**

Openstack Havana release is supported only on 12.04 release of Ubuntu. For libvirt and QEMU, the latest stable versions have been used, following are the version details for the host OS and the components:

**Host OS: Ubuntu 12.04 LTS**

**Openstack release: Havana**

**Libvirt: 1.2.1**

**QEMU: 1.7.0**

**Installation Procedure**

Install Ubuntu 12.04 LTS sever on the physical machine. After the installation is complete, add repository for Havana and update the system software.

**apt-get install python-software-properties**

**add-apt-repository cloud-archive:havana**

**apt-get update**

**apt-get upgrade**

**apt-get dist-upgrade**

After the upgrade is complete, reboot the system

**reboot**

Install the packages required to build QEMU

**apt-get install gcc libsdl1.2-dev zlib1g-dev libasound2-dev linux-kernel-headers pkg-config libgnutls-dev libpci-dev build-essential bzr bzr-builddeb cdbs debhelper devscripts dh-make diffutils dpatch fakeroot gnome-pkg-tools gnupg liburi-perl lintian patch patchutils pbuilder piuparts quilt ubuntu-dev-tools wget libglib2.0-dev libsdl1.2-dev libjpeg-dev libvde-dev libvdeplug2-dev libbrlapi-dev libaio-dev libfdt-dev texi2html texinfo info2man pod2pdf libnss3-dev libcap-dev libattr1-dev gcc-4.6-multilib libpixman-1-dev libxml2-dev wget subversion git**

Install additional packages required to build lbvirt

**apt-get install libyajl-dev libdevmapper-dev libpciaccess-dev libnl-dev**

Install the packages needed to run Libvirt/QEMU

**apt-get install bridge-utils dnsmasq pm-utils ebtables ntp chkconfig**

Start ntp service

**service ntp start**

**chkconfig ntp on**

Install python mysql connector, nova-compute and nova-network

**apt-get install python-mysqldb**

**apt-get install nova-compute-kvm nova-compute python-guestfs**

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

Reboot the system

**reboot**

Verify that kvm kernel module is loaded using this command

**lsmod |grep kvm**

If the module has been loaded then the output of the above command will look like this

**kvm\_intel 137721 0**

**kvm 415550 1 kvm\_intel**

If the output is blank then load the kernel module

**modprobe kvm**

Determine if the system is capable of running hardware accelerated KVM virtual machines

**kvm-ok**

The output should be like this

**INFO: /dev/kvm exists**

**KVM acceleration can be used**

Download version 1.7.0 of QEMU from the following link

**http://wiki.qemu-project.org/download/qemu-1.7.0.tar.bz2**

Extract, build and install QEMU (This will overwrite the existing installed version)

**tar -jxvf qemu-1.7.0.tar.bz2**

**cd qemu-1.7.0/**

**./configure --prefix=/usr --target-list=i386-softmmu,x86\_64-softmmu --enable-kvm --disable-werror --enable-debug**

**make**

**make install**

Verify the version of QEMU, it should show 1.7.0

**qemu-system-x86\_64 --version**

Download Libvirt 1.2.1 release form the following link

**http://libvirt.org/sources/libvirt-1.2.1.tar.gz**

Extract, build and install libvirt. This will overwrite the existing version

**tar -zxvf libvirt-1.2.1.tar.gz**

**cd libvirt-1.2.1/**

**./configure --prefix=/usr --localstatedir=/var --sysconfdir=/etc --with-xen=no --with-esx=no**

**make**

**make install**

Verify the version of Libvirt, it should show 1.2.1

**libvirtd --version**

Installation of QEMU and Libvirt is now complete. Now we need to make some configuration changes and install nova-compute on this host

Add the hostname entry of this machine to **/etc/hosts** if it is not already present, Assuming the IP is 10.35.34.14 and hostname is anarthita, add the following lines

**10.35.34.14 anarthita**

Update the network configuration in the file /etc/network/interfaces, put bridge information. Following is the sample configuration

**auto lo**

**iface lo inet loopback**

**auto eth0**

**iface eth0 inet dhcp**

**auto br100**

**iface br100 inet static**

**bridge\_ports eth0**

**bridge\_stp off**

**bridge\_maxwait 0**

**bridge\_fd 0**

**address 10.35.34.14**

**netmask 255.255.255.0**

**gateway 10.35.34.254**

**broadcast 10.35.34.255**

**dns-nameservers 192.168.0.4**

**dns-search gslab.com**

Set net.ipv4.ip\_forward=1 in **/etc/sysctl.conf**, then run this command

**sysctl -p**

Make the current kernel readable

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

To also enable this override for all future kernel updates, create the file **/etc/kernel/postinst.d/statoverride** with the following contents

**#!/bin/sh**

**version="$1"**

**# passing the kernel version is required**

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

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

Make the above file executable

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

Remove the SQLite database created by the packages

**rm /var/lib/nova/nova.sqlite**

Update **/etc/nova/nova.conf** and **/etc/nova/nova-compute.conf**. Please check Appendix of this document for sample configuration files. The configuration may differ based on Openstack controller configuration. Make sure that the files are owned by nova

**groupadd nova**

**usermod -g nova nova**

**chown -R nova:nova /etc/nova**

**chmod 640 /etc/nova/nova.conf**

Restart the services

**service nova-compute restart**

**service nova-network restart**

Add the following line to **/etc/rc.local** file

**chown -R nova:nova /var/run/libvirt/**

Openstack Havana code is not compatible with Libvirt version 1.2.1. To fix this issue open the file **/usr/share/pyshared/nova/virt/libvirt/driver.py** and go to the following function

**def \_get\_connection(self):**

and check for the line containing this condition

**if self.has\_min\_version(MIN\_LIBVIRT\_CLOSE\_CALLBACK\_VERSION):**

comment this line all the lines (8 lines) inside this condition and add the following lines below that, take care of leading spaces (there are 12 spaces before try:)

**try:  
                LOG.debug(\_("Registering for connection events: %s") %  
                          str(self))  
                wrapped\_conn.registerCloseCallback(self.\_close\_callback, None)  
            except (TypeError, AttributeError) as e:  
                LOG.debug(\_("The version of python-libvirt does not support "  
                            "registerCloseCallback or is too old: %s"), e)  
            except libvirt.libvirtError as e:  
                LOG.warn(\_("URI %(uri)s does not support connection"  
                           " events: %(error)s"),  
                         {'uri': self.uri(), 'error': e})**

Reboot the machine to make sure that nova-compute starts properly on reboot

**Reboot**

Go to Horizon UI and check the Hypervisors tab and verify that this machine is listed there. Now you can launch VMs on this host using Openstack.

**Appendix**

Sample **nova.conf** file

**[DEFAULT]**

**max\_kernel\_ramdisk\_size=2073741824**

**dhcpbridge\_flagfile=/etc/nova/nova.conf**

**dhcpbridge=/usr/bin/nova-dhcpbridge**

**logdir=/var/log/nova**

**state\_path=/var/lib/nova**

**lock\_path=/var/lock/nova**

**force\_dhcp\_release=True**

**iscsi\_helper=tgtadm**

**rootwrap\_config=/etc/nova/rootwrap.conf**

**verbose=True**

**ec2\_private\_dns\_show\_ip=True**

**api\_paste\_config=/etc/nova/api-paste.ini**

**volumes\_path=/var/lib/nova/volumes**

**enabled\_apis=metadata,ec2,osapi\_compute**

**#enabled\_apis=metadata**

**metadata\_host = 10.35.35.10**

**auth\_strategy=keystone**

**rpc\_backend = nova.rpc.impl\_kombu**

**rabbit\_host = 10.35.35.10**

**rabbit\_password = intelrp**

**my\_ip=10.35.34.14**

**vnc\_enabled=True**

**vncserver\_listen=0.0.0.0**

**vncserver\_proxyclient\_address=10.35.34.14**

**novncproxy\_base\_url=http://10.35.35.10:6080/vnc\_auto.html**

**glance\_host=10.35.35.10**

**glance\_api\_servers=10.35.35.10:9292**

**image\_service=nova.image.glance.GlanceImageService**

**compute\_scheduler\_driver=nova.scheduler.simple.SimpleScheduler**

**nova\_url=http://10.35.35.10:8774/v1.1/**

**rootwrap\_config=/etc/nova/rootwrap.conf**

**# Network**

**#dnsmasq\_config\_file=/etc/dnsmasq-nova.conf**

**network\_manager=nova.network.manager.FlatDHCPManager**

**dhcpbridge\_flagfile=/etc/nova/nova.conf**

**firewall\_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver**

**network\_size=253**

**allow\_same\_net\_traffic=True**

**multi\_host=True**

**send\_arp\_for\_ha=True**

**fixed\_range=''**

**share\_dhcp\_address=True**

**force\_dhcp\_release=True**

**flat\_network\_bridge=br100**

**flat\_interface=eth0**

**public\_interface=eth0**

**flat\_injected=False**

**network\_host=10.35.34.14**

**libvirt\_type=kvm**

**[database]**

**connection = mysql://nova:intelrp@10.35.35.10/nova**

Sample **nova-compute.conf** file

**[DEFAULT]**

**compute\_driver=libvirt.LibvirtDriver**

**libvirt\_type=kvm**

**libvirt\_cpu\_mode=none**

**instance\_name\_template=instance-%08x**

**instances\_path=/var/lib/nova/instances**