Working With openQRM

CPU and System details

cat /proc/cpuinfo | grep svm

flags : fpu de tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush mmx fxsr sse sse2 ht nx mmxext fxsr_opt lm 3dnowext 3dnow pni cx16 lahf_lm cmp_legacy svm extapic cr8legacy ts fid vid ttp tm stc

Linux fed08 2.6.21-2950.fc8xen #1 SMP Tue Oct 23 12:24:34 EDT 2007 i686 athlon i386 GNU/Linux

Installing openQRM :

Initial Requirements:

Checklist Xen rpms,openQRM rpms,php version.

PHP 5.2.4 (openQRM needs greater than PHP 5)

Installed Xen rpms:

kernel-xen-2.6.21-2950.fc8

xen-3.2.0-0xs.fc8

jaxen-1.1-1jpp.2.fc7

xen-libs-3.2.0-0xs.fc8

xen-devel-3.2.0-0xs.fc8

Installed openQRM rpms :

openqrm-server-4.4-fedora9

openqrm-plugin-image-shelf-4.4.1.3-fedora9

openqrm-plugin-dhcpd-4.4.1.2-fedora9

openqrm-plugin-xen-4.4.1.3-fedora9

openqrm-plugin-kvm-4.4.1.1-fedora9

openqrm-plugin-highavailability-4.4.1.2-fedora9

openqrm-plugin-nfs-storage-4.4.1.3-fedora9

openqrm-plugin-local-server-4.4.1.2-fedora9

openqrm-plugin-sshterm-4.4.1.3-fedora9

openqrm-plugin-tftpd-4.4.1.2-fedora9

openqrm-plugin-local-storage-4.4.1.3-fedora9

openqrm-plugin-lvm-storage-4.4.1.3-fedora9

openqrm-plugin-dns-4.4.1.1-fedora9

openqrm-plugin-cloud-4.4.1.3-fedora9

Installed PHP rpms:

php-pecl-apc-3.0.14-2.fc8

php-5.2.4-3

php-ldap-5.2.4-3

php-gd-5.2.4-3

php-mysql-5.2.4-3

php-cli-5.2.4-3

php-xml-5.2.4-3

php-pdo-5.2.4-3

php-mbstring-5.2.4-3

php-odbc-5.2.4-3

php-common-5.2.4-3

php-pgsql-5.2.4-3

phpldapadmin-1.0.1-1.fc6

Installed mysql rpms:

libdbi-dbd-mysql-0.8.2-1.2.fc8

mysql-devel-5.0.45-4.fc8

php-mysql-5.2.4-3

mod_auth_mysql-3.0.0-5

mysql-libs-5.0.45-4.fc8

mysql-connector-odbc-3.51.14r248-2.fc8

mysql-server-5.0.45-4.fc8

mysql-5.0.45-4.fc8

Installed gcc rpms

gcc-gfortran-4.1.2-33

gcc-c++-4.1.2-33

gcc-4.1.2-33

gcc-java-4.1.2-33

libgcc-4.1.2-33

How to Create Xen VM's using openQRM: (on single host)

Creating a VM involves 7 steps:

Make sure you have created LVM volume group called "vol".

Make sure you have added kernel to openqrm using command

/usr/lib/openqrm/bin/openqrm kernel add -n mykernel -v 2.6.18-6-686 -u openqrm -p openqrm -l / -i initramfs

Note: replace 2.6.18-6-686 with your kernel version.

Local host ipaddress : 10.2.3.4

Step 1: Go to Storage and click on new storage (where server template will stored later) .Now set the deployment type as Lvm Storage Server (Nfs) and choose your local openqrm server machine (10.2.3.4) from "Resource List". Fill the Storage name "Cloud_storage" and save.

->What Just Happened?

You have named a Storage location as "Cloud_storage" with type as LVM NFS on machine 10.2.3.4

Step 2: Go to LVM plug-in, Select "Cloud_storage" .

Now you can see table with Storage name as "Cloud_storage" and IPaddress and "Type". Click on refresh. In order to create logical volume,click on "Vol" (highlighted with black back ground) Now provide a logical volume name ("cloud_lv")and it's size and click on "Add"

->What Just Happened?

You have created a new logical volume (named "cloud_lv) on the Storage server.

You can verify this from terminal run following command

df -h

You can see "cloud_lv" mounted on "vol"

Step 3: Goto images select "New Image" Tab and chose "Cloud Storage", click on select to enter the image name. Enter image name as "Cloud_image", set root password and select root device from the drop down. (Set this root devices same as our logical location "/vol/cloud_lv")

Then deployment parameter as IMAGE_ROOT_DIR="/vol/cloud_lv"

-->What Just Happened?

Our logical volume "cloud_lv" mapped as root fs.

Now we have a empty logical volume that needs to be filled with server template (or virutal appliance)

Step 4: Now logical volume (/vol/cloud_lv) needs to be filled template.(virutal appliance) (we have downloaded debian4.0.gz template from openqrm.com) move this .gz file to /vol/cloud_lv and decompress it.

-->What Just Happened?

We have filled the logical volume with template.

Step 5: Now create Xen VM: Go to Appliance ->New Appliance. Choose local openqrm server (10.2.3.4) from the displayed Resource list. Provide appliance name ("cloud_appliance") and set resource type as "Xen Host" from the dropdown and save it.

-->What Just Happened?

You have created a Xen Host appliance to create Xen VM's

Step 6: Go to Xen-plug-in. Choose the Xen-host from the list and Now you can click on "+VM" And set VM name ("Cloud_VM"), disk, swap size and click create. Click refresh, you can see the "Cloud_VM" which can be start/reboot/stop/remove/delete.

Now Go to Resource, you will have "Idle1" resource, set this as Xen VM from dropdown and clickUpdate.

-->What Just Happened?

Xen VM’s are created. You can verify this by connecting via VNCviewer or xm list command.

Step 7: Now we need to deploy the image (cloud_image) created at location "logical_lv" to this new VM (cloud_VM).For that ,Goto appliance->New appliance and select the new "idle1" from resource list Now provide the appliance name (Debian_VM) chose kernel (mykernel) and image (cloud_image) and set resource as "xen VM" and save it. That's it.

Now goto "Resources" ,you can see new resource with it's new ipaddress (10.20.30,40) of Debian_VM.Goto ssh-terminal plugin,you can login via,available terminal for new Debian_VM.