OpenStack Lab on VMware Workstation – Getting Started

One of the hot topics these days is OpenStack. When I say hot topic, I should really say that it's a fast-growing ecosystem which has been, and will continue to gain traction in the cloud infrastructure world.

What makes OpenStack very interesting to me, and to many others in the virtualization community is that it can become a part of our core infrastructure without having to be a replacement. Most importantly, as a Systems Architect, I have a responsibility to understand where this can fit, and also where there may be challenges and limitations from the perspective of technology and business process.

Getting started with OpenStack

What is really cool about this system is that you can kick the tires on it with relative ease by utilizing some of the great how-to posts and deployment recipes out there in the blogosphere.

What this post is about

This post is what I call step 1. Just getting to the point where you are about to install OpenStack in your virtual lab using VMware Workstation. And I really mean step 1 because there will be lots more to do once we get started there.

I purposefully didn't call this Post 1 of "n" because it will be an ongoing set of an unknown number of posts as I bring you through the journey to getting your OpenStack lab going to test drive the core features. The goal is to learn and fully understand the concepts and practices for creating an OpenStack framework.

Truthfully this is the teaser to get you past the common first error to the point where you begin the installation. You will see what I mean as we get rolling. We are going to use the All-in-One deployment at first and then try out the multi-node deployment later once the concepts have been fully take in.

What do I need to get started?

There are a few key things that you will need to get started with demoing OpenStack:

- The Rackspace Private Cloud ISO – Thank you Rackspace for this one! http://www.rackspace.com/cloud/private/openstack_software/
- VMware Workstation 9 – This particular lab is built in Windows using VMware Workstation 9
- *nix command line comfort – you will need to be working in a Linux shell, so previous experience with some *nix OS will really help you going forward
- Lab time – physical or virtual, you will have lots of awesome lab time ahead of you
- Love of learning – This is important because this will be very new to many and is conceptually very cool but will require some serious grokking
- Persistence – It won't always go right. You will inevitably hit a few walls along the way as you experiment but that is what makes the whole experience better because we learn from those

moments when it goes a little off script

Starting your lab with VMware Workstation

In your VMware Workstation we get started by using the New Virtual Machine wizard to setup our virtual environment. I'm running this on a laptop with 8GB of RAM so performance isn't the goal, but at least we can get the deployment underway and build up or comfort before launching into a bigger environment.

Create your VM through the wizard by following the usual steps. Most importantly we choose Custom (advanced) during the first step to give us the option to set specifics in disk type:

Choose Workstation 9.0 for hardware compatibility:

This step is interesting because although we have our Alamo ISO image, we want to avoid the "Easy Install" which will be used because VMware Workstation recognizes the Ubuntu OS and wants to save us some effort. We like to take the extra steps during installation because this isn't just a traditional Ubuntu install.

Select **I will install the operating system later** in the wizard before moving to the next step:

Choose Linux and Ubuntu 64-bit for the version:

I've named mine OPENSTACK01 for my VM name:

Because I need to be able to carve off nodes within my OS, I've chosen to give 4 CPUs to the VM. The nested guests will then be given a subset of those resources during deployment after we are up and running:

I really wish I had more RAM at this point, but to keep the lights on for my little laptop I have decided to use 4GB of RAM for my guest:

This step will need revisiting later because there are some really funky networking requirements to build our OpenStack deployment. For our purposes to get the install started we set the first network as NAT and we will add other networks after the fact:

Let's take the recommended LSI Logic option for controller:

We will create a new disk for our VM:

We need to use SCSI for the disk type:

One thing you will need to have is some disk. Maybe not all at once, but as we deploy nodes into our OpenStack environment the parent machine needs space to work with. I've chosen to use a 200 GB virtual disk, but I'm using a thin deployment so that it will only use what is required as we grow.

We also want to select **Store virtual disk as a single file** unless we want to have a big pile of VMDK files to deal with later.

Nothing to change in our disk file name:

We have finished the wizard portion of the deployment, but let's click the **Customize Hardware** button before we go and attach our Alamo ISO file:

Click your **New CD/DVD** device, select **Use ISO image file** and point to your Alamo-v2.0.0.iso file:

Close up your deployment wizard and we are ready to start our VM. We are going to hit an error in a couple of steps, but I am doing this on purpose so that we learn why that error occurs.

You will now see your fancy new Rackspace Private Cloud installer begin:

Click inside the VM window and hit the Enter key to accept the EULA:

And suddenly after a few snazzy progress bars fly by, we hit the first wall! This was the error that I was mentioning earlier. In order to run OpenStack, the OS requires hardware virtualization enabled to be able to carve off resources to its own nodes in the OpenStack compute node environment.

When you see this unfortunate red screen below you can simply power down your VM and we are going to modify the settings to correct this and proceed:

In the Virtual Machine Settings, highlight **Processors** and in the right-hand pane you need to check off the **Virtualize Intel VT-x/EPT or AMD-V/RVI** option which simulates the appearance of a VT enabled BIOS. This is the same step required for nesting ESXi instances inside our VMware workstation.

Once you make that change, you can go back to your VM, start up the machine again and get back to

the installer just like we did in the previous steps.

As you can see, we have gotten past the issue where the installer needed hardware virtualization enabled to proceed.

And now we are ready to really get started!

This has gotten past the most common error that stops a lot of folks in deploying nested virtualized environments. There is much, much more to come with getting our OpenStack on VMware Workstation lab setup so stay tuned and I'll post the next steps in the coming days.

UPDATE: Here is the next step to get your first OpenStack All-In-One up and running!