

# OpenStack Havana All-in-One lab on VMware Workstation

With all of the popularity of OpenStack in general, and specifically with my other posts on deploying the Rackspace Private Cloud lab on VMware Workstation, I think it's time for me to update with a new lab build for everyone.

This lab build process will be done using VMware Workstation, an Ubuntu 12.04 LTS ISO file, and a few install shell scripts that I've created for you to pull from Github.

**NOTE: This looks like a lot of work because it's a long post, but the actual steps are quick and I've tried to do the heavy lifting with the install scripts, so hopefully this will only take about 20-30 minutes for you to be fully up and running.**

This post will run through the use of 3 scripts that are used to set up:

- OpenStack pre-requisites - MySQL, RabbitMQ, NTP, Ubuntu Cloud Keyring, and other useful services like OpenSSH server for remote access into your VM
- OpenStack Identity (Keystone)
- OpenStack Image Service (Glance)
- OpenStack Compute (Nova)
- OpenStack Dashboard (Horizon)

There will be more posts coming after this to add additional features to illustrate the entire set of OpenStack Havana core projects, but this is your starting point to get you up on your first OpenStack Havana lab.

## What is different with this install?

In my previous posts, I showed you how to use the Rackspace Private Cloud deployment as a simple recipe-oriented build that was self-installing with some input. Those builds were based on the Folsom release. This time around, I wanted to show you how to run a simple shell script with minimal input, and use the native build of OpenStack Havana so that you can test the waters on a more self-directed deployment.

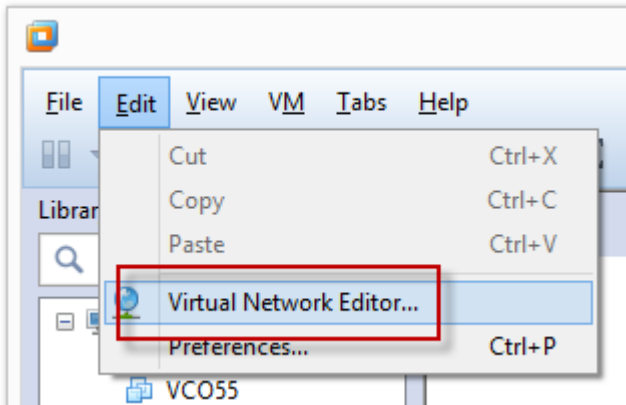
The use of a shell script will give newcomers a little bit of insight in how to deploy using the command structure without using Puppet or Chef. Now, why would we do this when a more efficient way is available using those tools, right? It's all about the learning process.

I want to bring more folks to the first steps of OpenStack deployment and management so that we can all grow our comfort with the process. Don't worry, there will be lots of fully orchestrated posts coming, but we have to walk before we run.

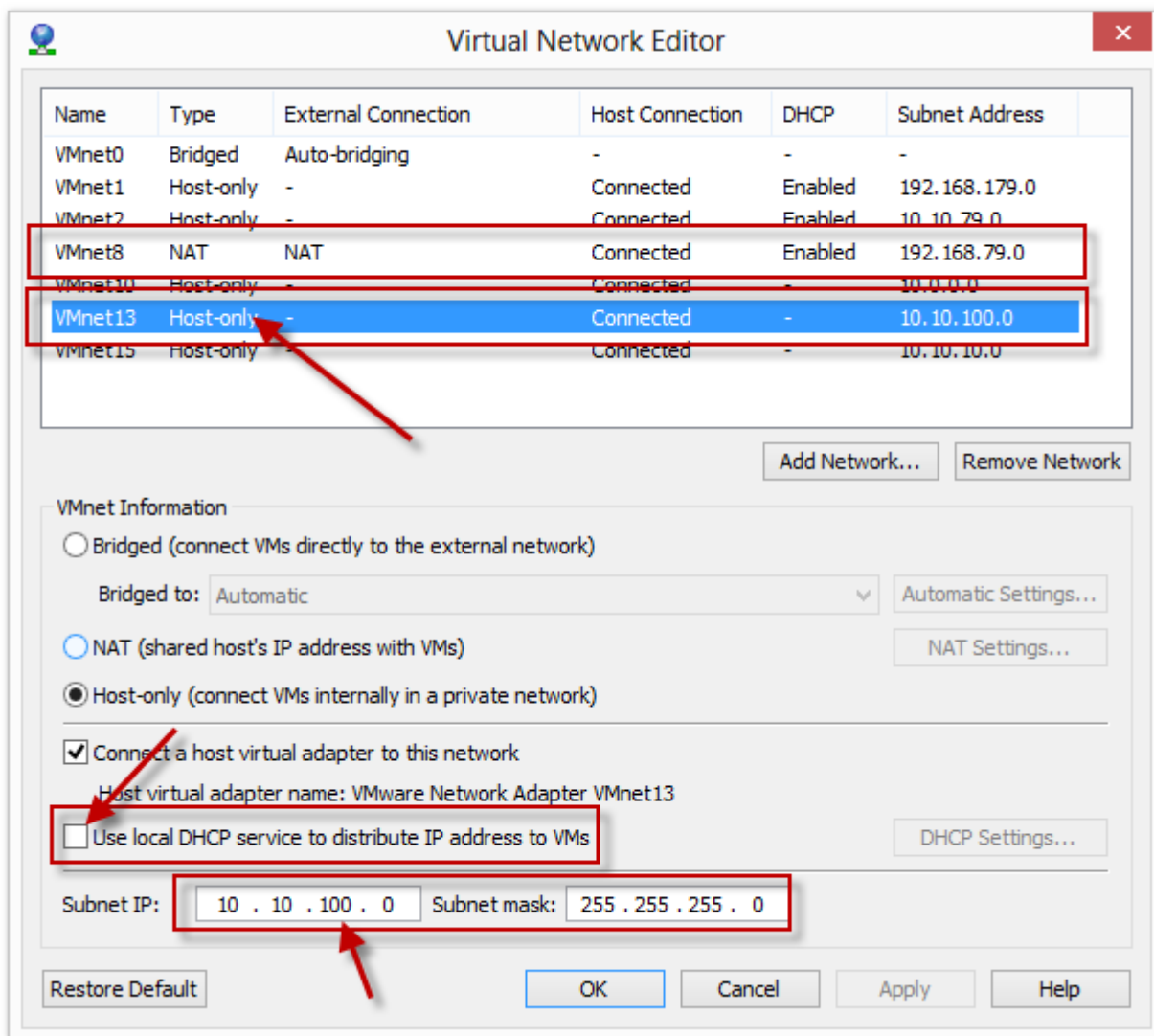
## Creating your Virtual Networks

This VM will require 2 networks: a public network that has internet access to pull down the installation components, and a private network that is where we launch our nested virtual guests into.

In VMware Workstation, go to your **Virtual Network Editor**:



Once the editor launches, you will see a variety of different networks. Your internal network will differ, but I've built the process around a private network of **10.10.100.0/24** which in my case is **VMNet13**. Be sure to uncheck the **Use local DHCP service to distribute IP address to VMs** and be sure that your Subnet IP matches the image below:



You will need to know your IP range of the NAT network, so please write down an IP address that is on your NAT range. In the case of mine, I've used 192.168.79.50/24. That will be used as we launch our script once the VM is built.

## Download your Ubuntu 12.04 LTS ISO file

To download your Ubuntu install image, go to <http://www.ubuntu.com/download/server> and choose the 12.04.03 LTS 64-bit, and keep track of where you store the ISO file once it is downloaded.

**For extended support,**  
choose Ubuntu Server 12.04.3 LTS

Ubuntu 12.04.3 LTS is a long-term support release, with support guaranteed until April 2017. Havana is available via the [Ubuntu Cloud Archive](#).

[Read the full installation instructions >](#)

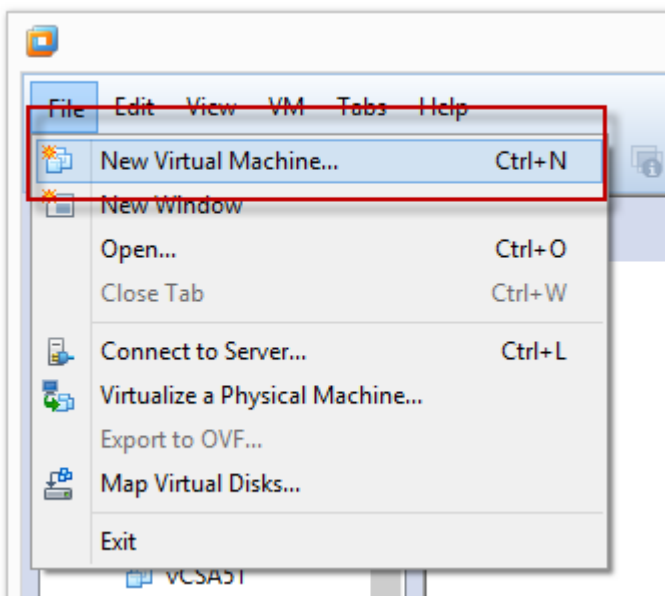
Choose your flavour  
64-bit (recommended) ▾

**Get Ubuntu 12.04 LTS**

Once you have your ISO file, and your VM network information, we are ready to get started.

## Building your All-in-One VM with VMware Workstation

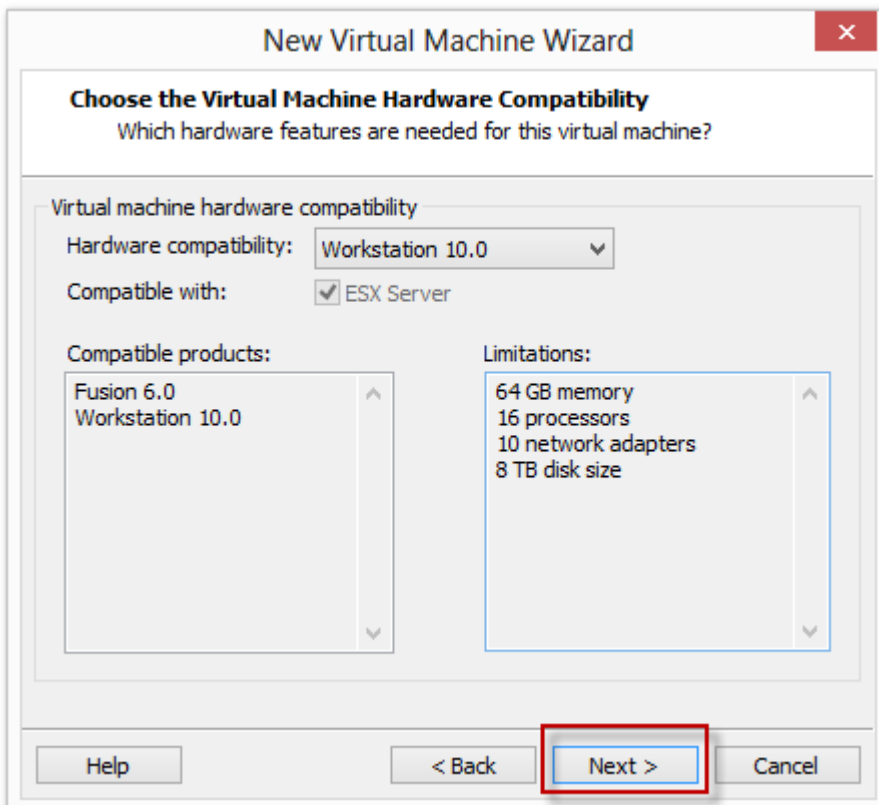
This is a step-by-step of the process to deploy a new Ubuntu Server VM guest on VMware Workstation 10 using the Easy Install wizard. You may already be quite familiar with the process, but be sure to double-check your steps as there are some key points which could impact the build later.



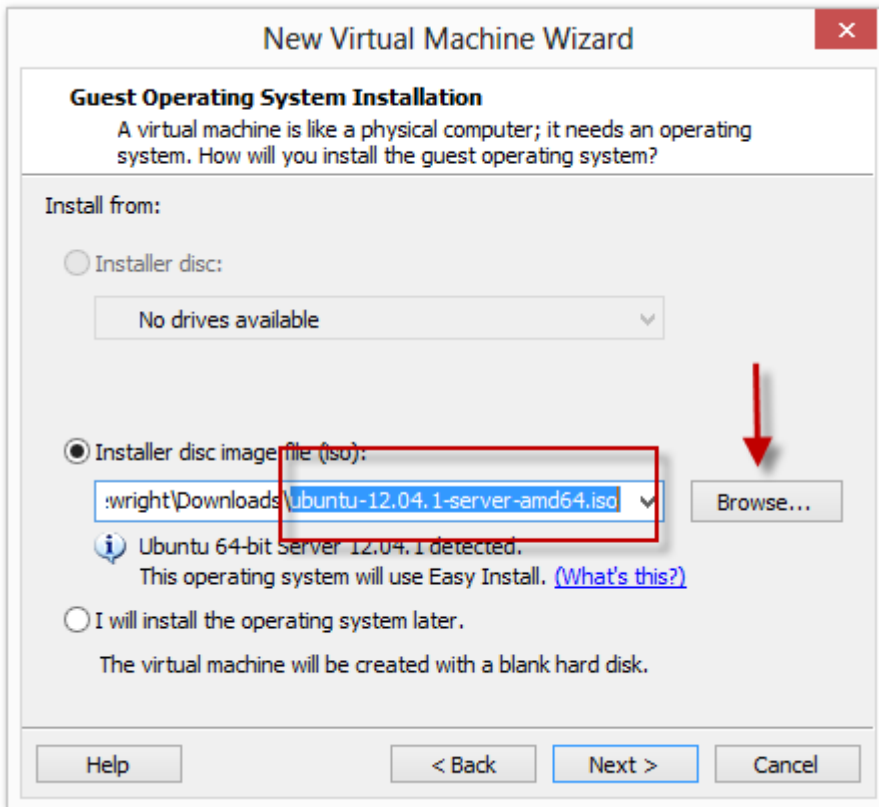
This will launch the New Virtual Machine Wizard. **Choose Custom (advanced)** and click **Next**:



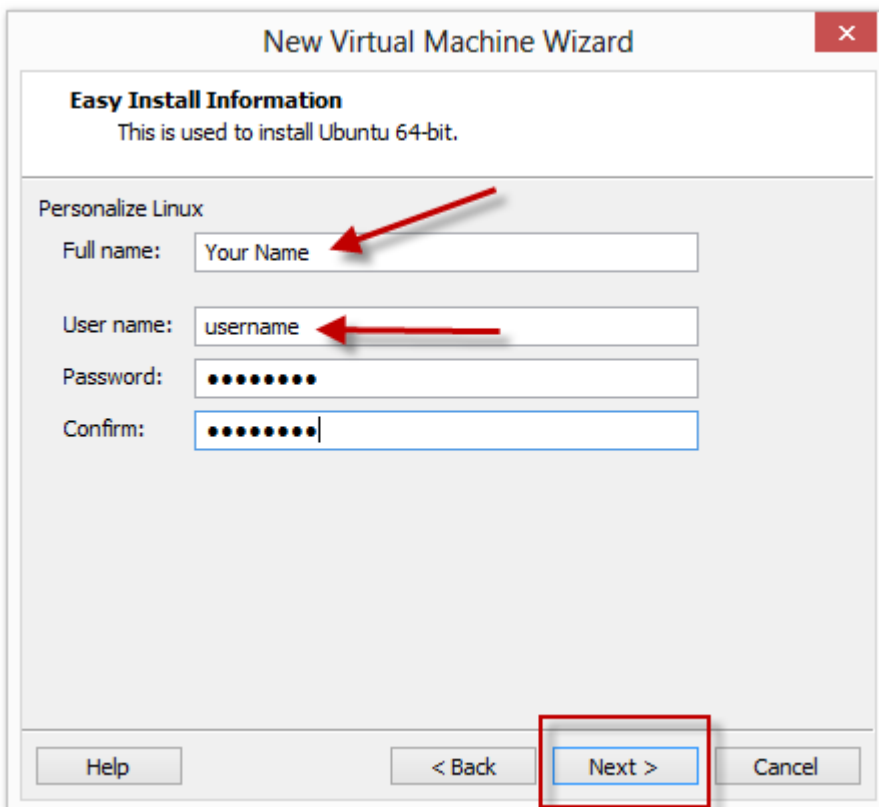
Take the default setting of **Workstation 10.0** for your Hardware Compatibility:



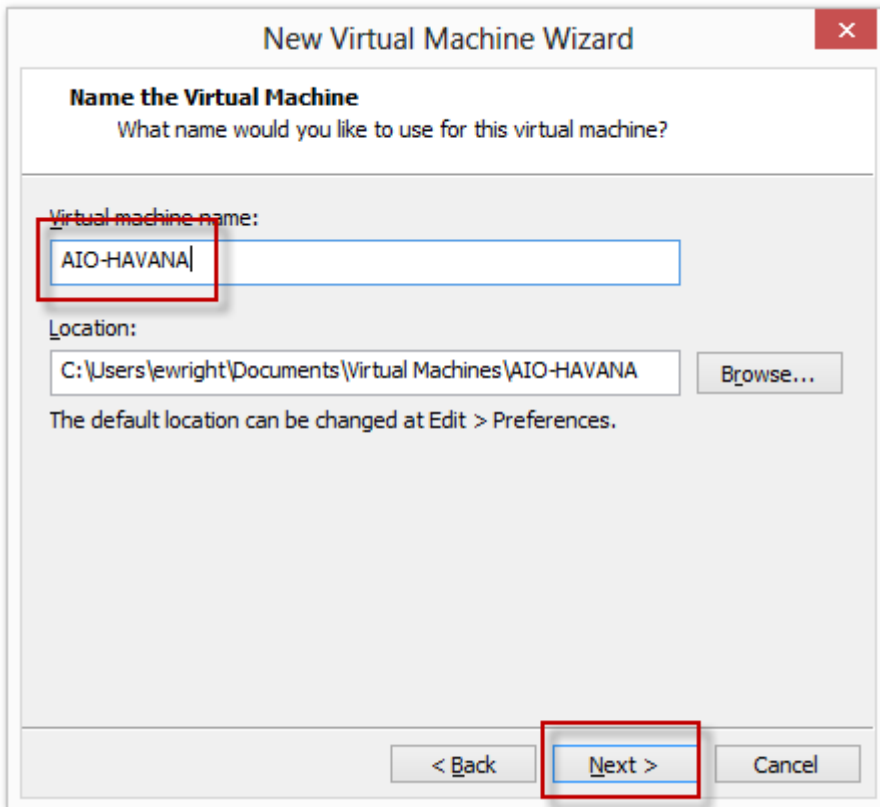
Click the **Browse** button and locate your Ubuntu Server ISO install file then click Next:



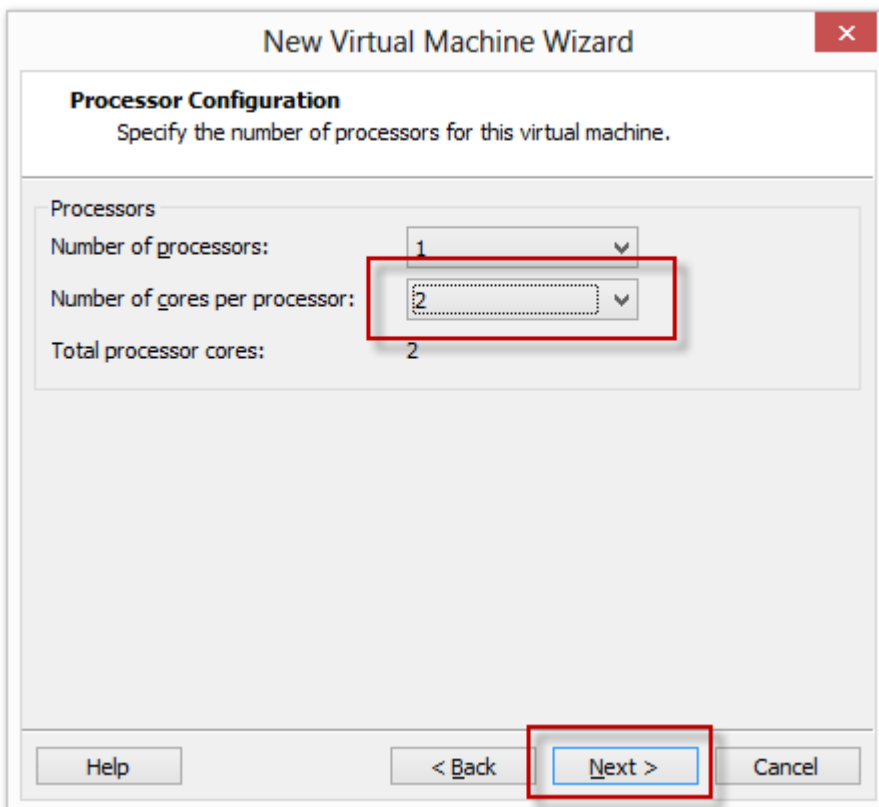
Because we are using the Easy Install wizard, you can type in the username that you will use for your server image. Type in your **full name**, then your **username** (which is your login name), and type your password, then click **Next**:



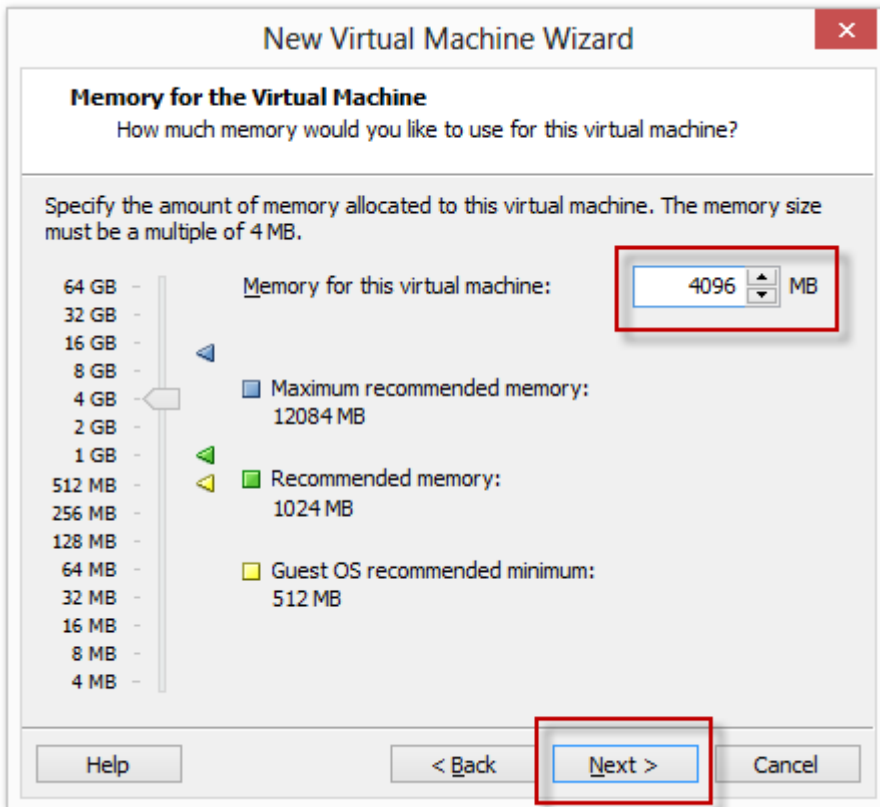
For the machine name, you can choose anything but I recommend that you call it **AIO-HAVANA** which is what the hostname will be after the install script is run. Then click **Next**:



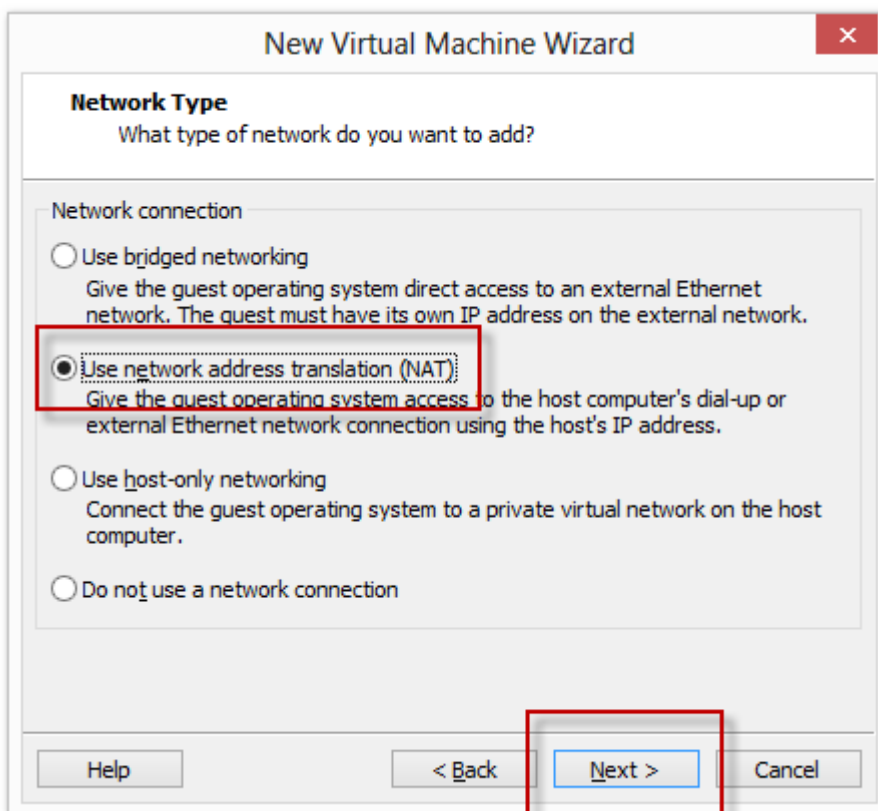
For our processor configuration, you will want to add some cores. The smallest build that I'd recommend is a 1 CPU with 2 cores. You can choose more virtual CPU or cores per processor depending on your physical hardware. Pick the appropriate sizing and click **Next**:



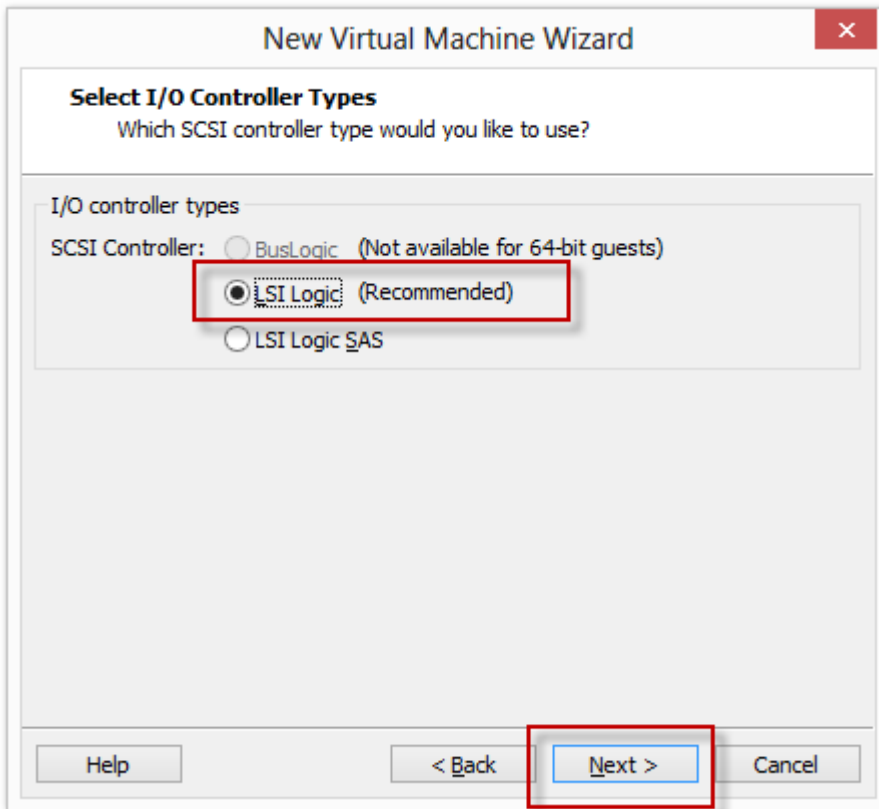
For memory, choose as much as you can spare. The minimum I will recommend is 4 GB, but if you want to do much more guest deployments and testing, more is always better. Pick the amount which suits your lab capabilities and click **Next**:



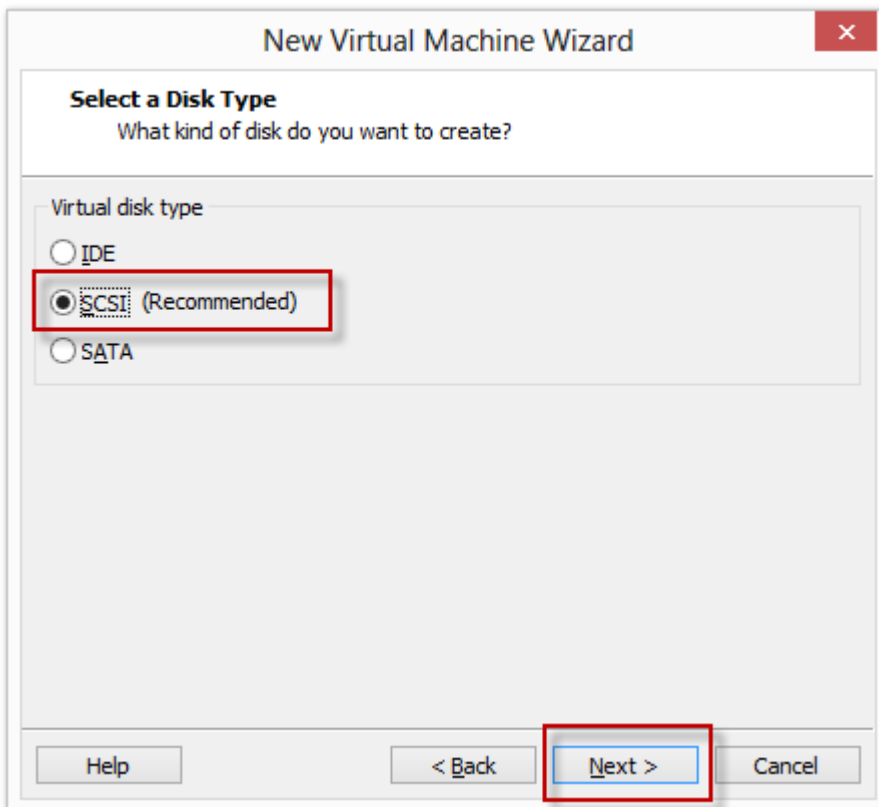
Choose **Use network address translation (NAT)** for your NIC because this is your public facing interface that we will download the OpenStack packages with. Click **Next**:



Take the default I/O controller (LSI Logic) and click **Next**:

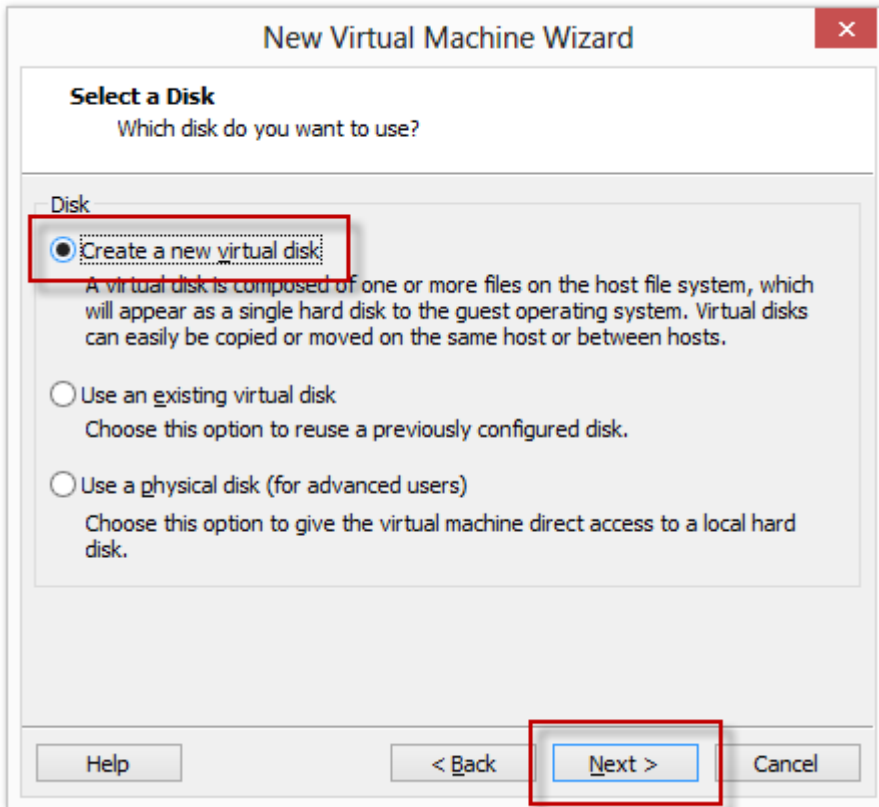


Choose the default (SCSI) and click **Next**:



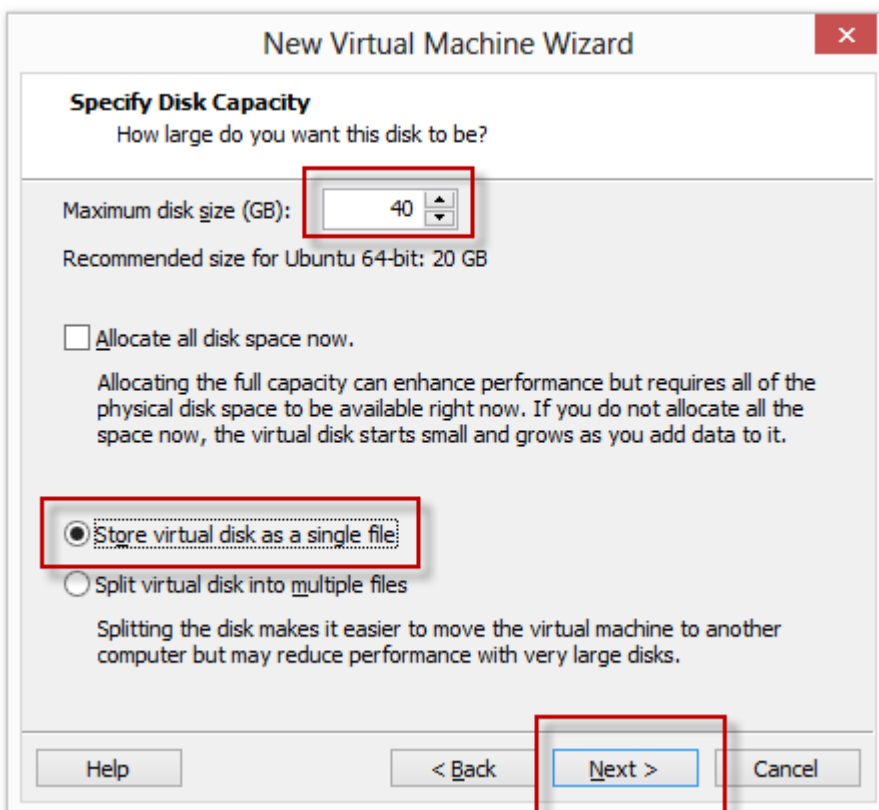
Choose **Create a new virtual disk** and click **Next**:



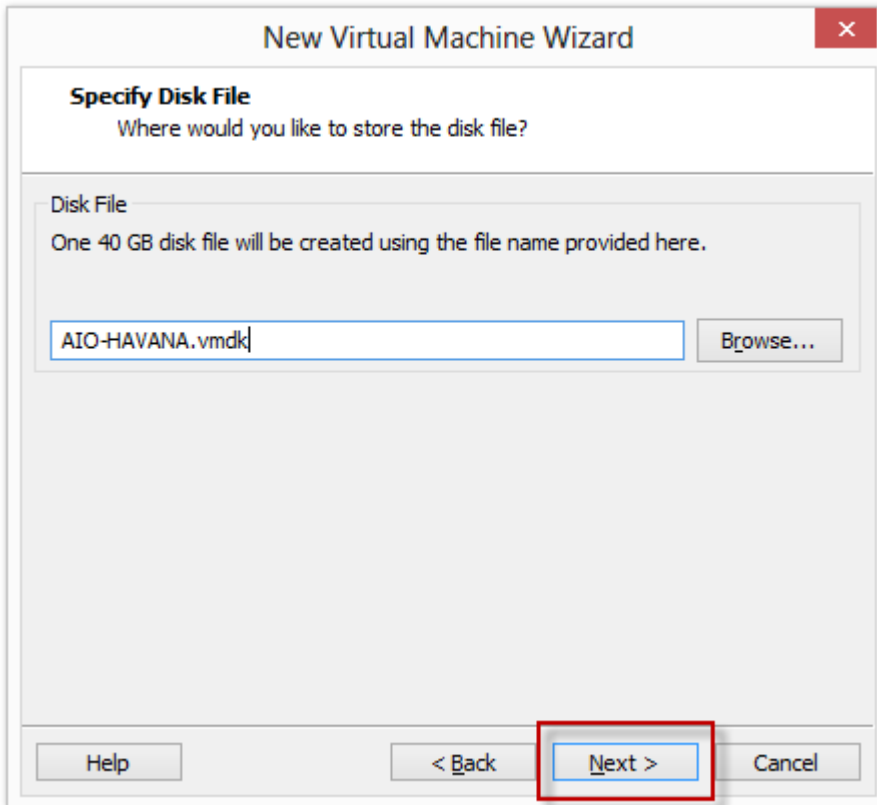


The default is 20 GB, which may work, but I recommend a minimum of 40 GB. Remember that this is going to be Thin Provisioned, which means that it will only grow as the data size inside your VM guest grows.

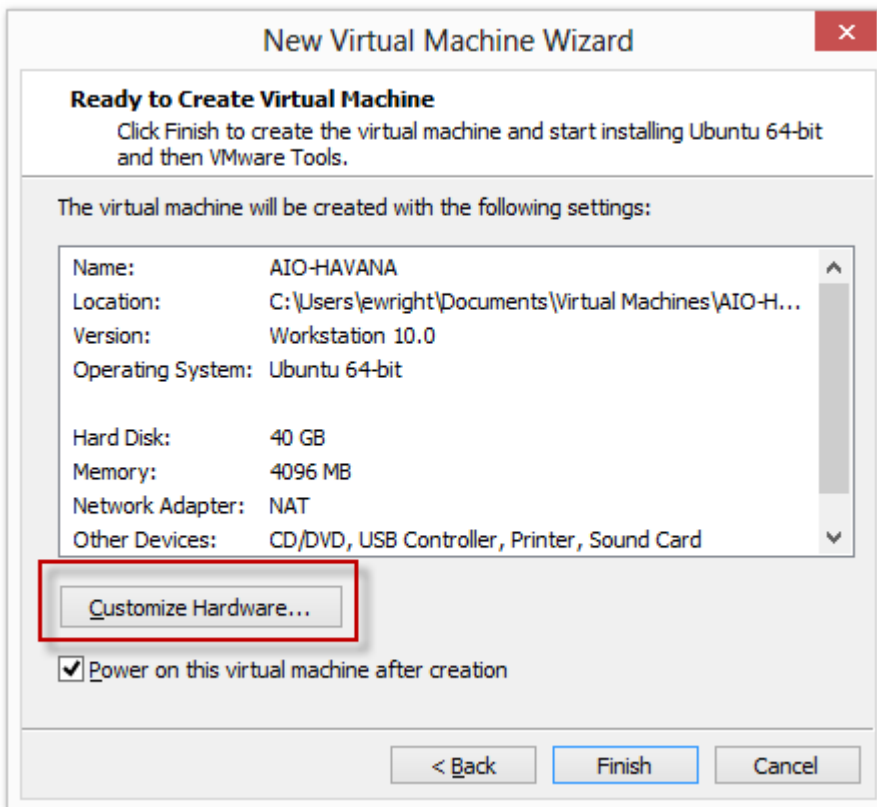
For ease of movement and management, I always like to select **Store virtual disk as a single file**. Once that has been selected, click **Next**:



Choose the default name and click **Next**:

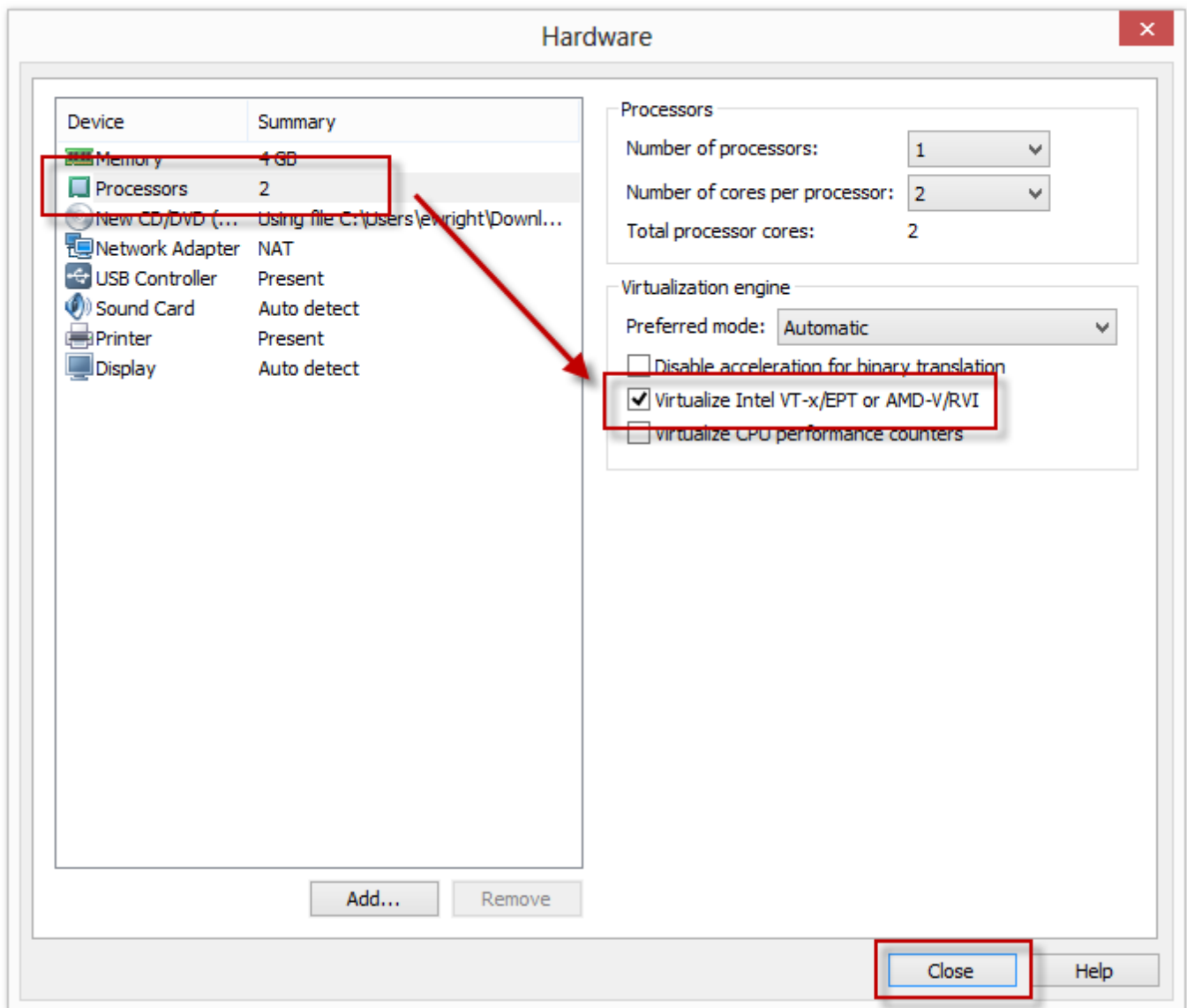


DON'T CLICK FINISH YET!! Before we do that, we have to modify our CPU configuration to set the CPU virtualization option. Click on **Customize Hardware**:

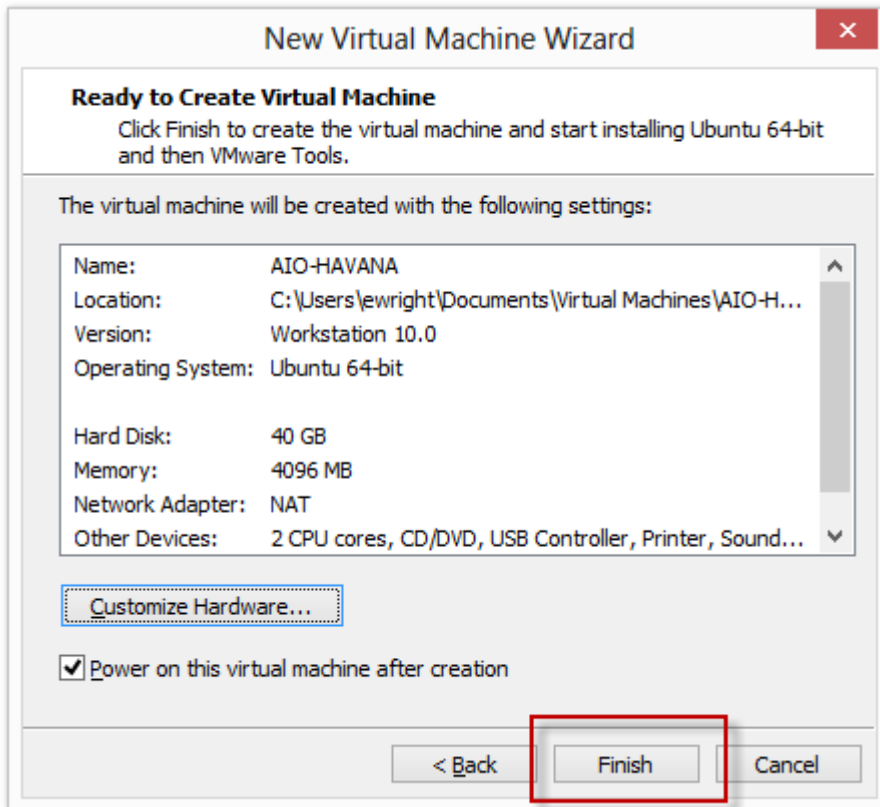


In the left hand pane, click **Processors** and then check the **Virtualize Intel VT-x/EPT or AMD-**

V/RVI checkbox and then click **Close**:

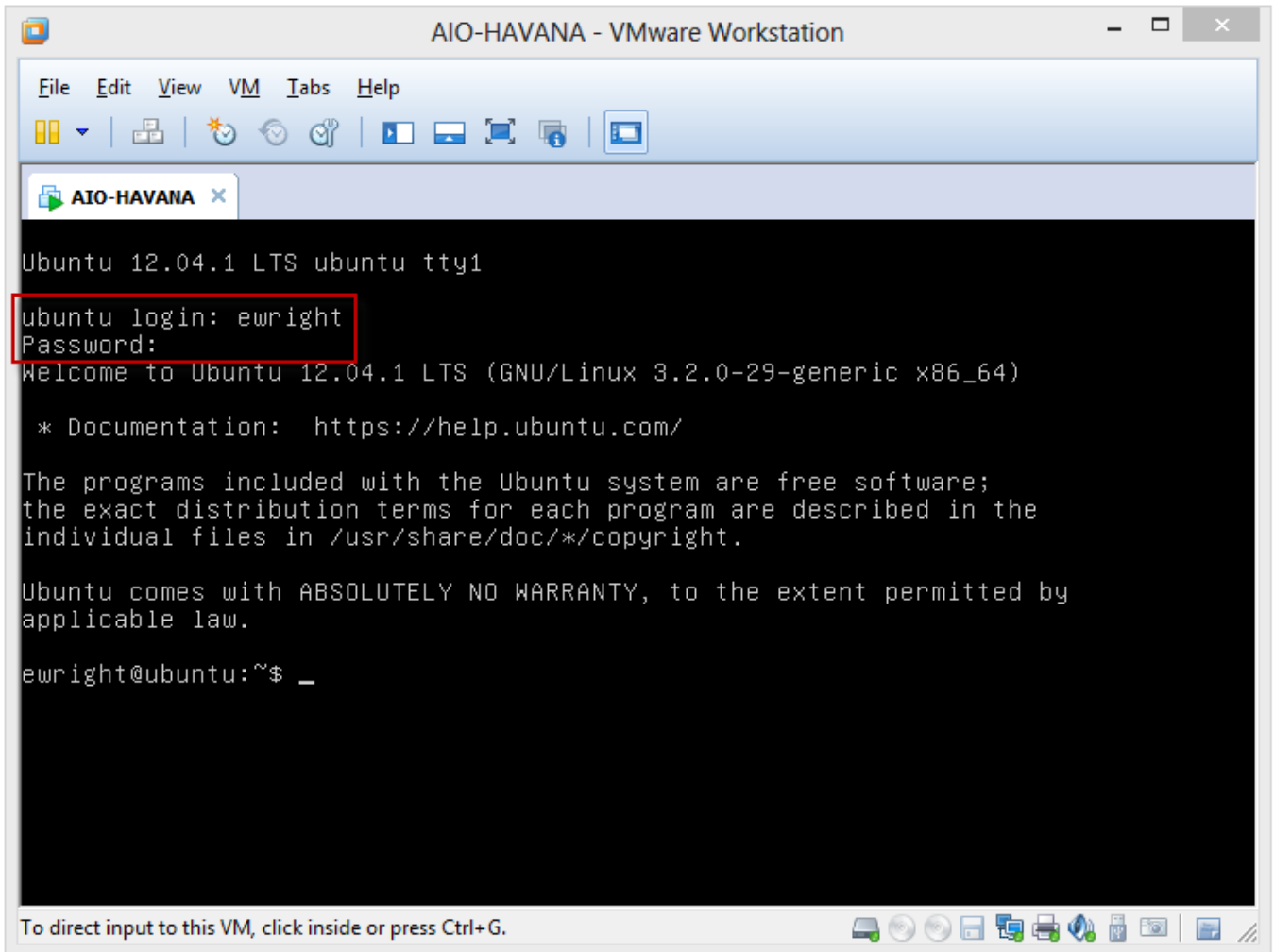


Now we are good to go! Close so click Finish and the VM build will begin:

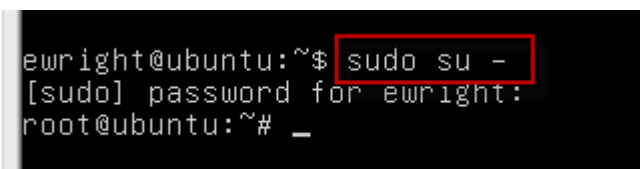


The installation will take anywhere from 5-20 minutes depending on hardware and other factors. Once you are done you will be presented with the login prompt as shown below.

Log in with the username and password that you set during the Easy Install wizard:



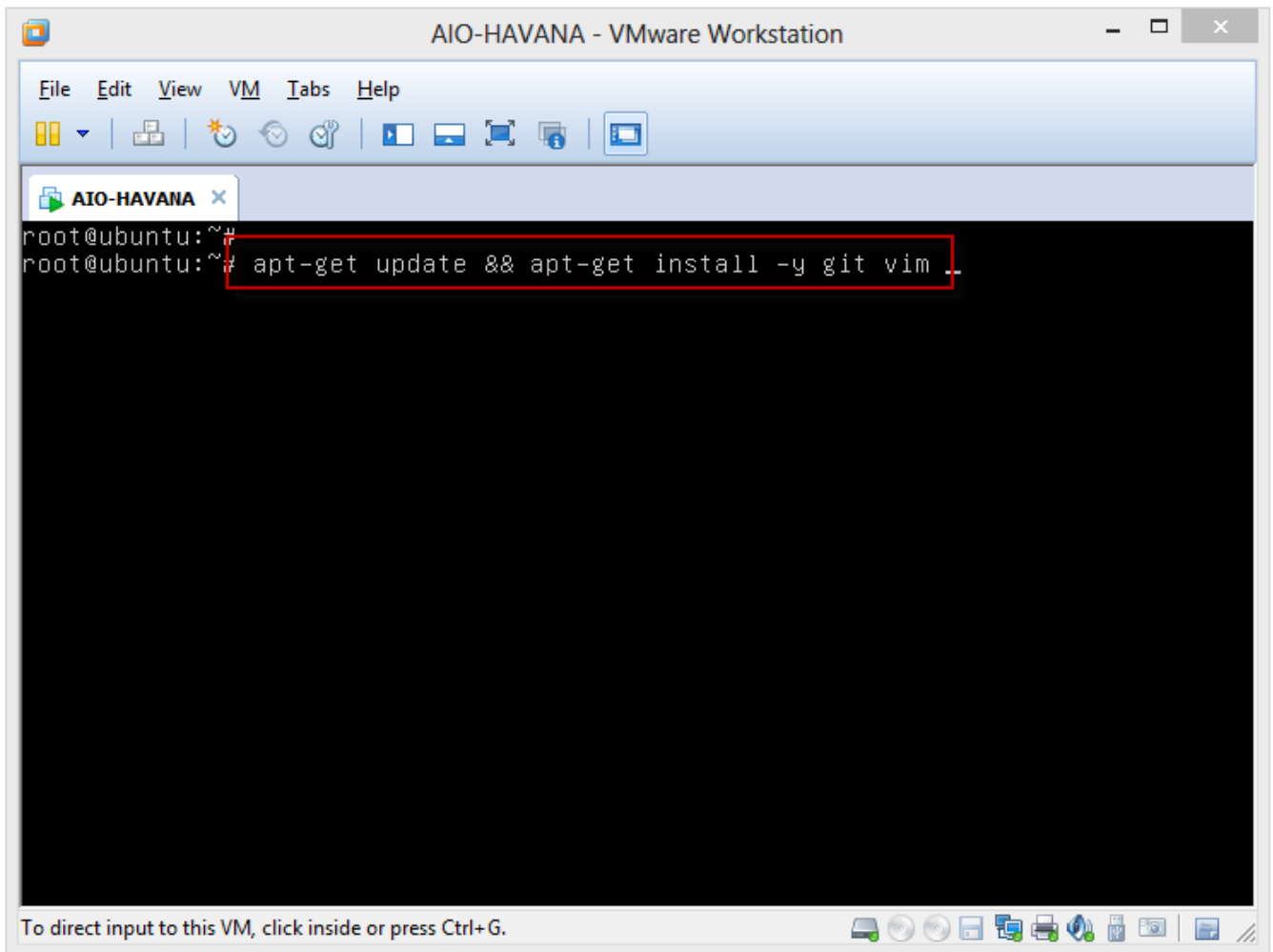
Now we are going to do something naughty and type **sudo su -** which elevates our privileges for the preparation script.



Now we will run the basic package updates and install the git and vim packages. We really only need git, but if you're feeling like you'd like to dig around, you can edit some files using vim if you feel comfortable with that.

Type the following command:

```
apt-get update && apt-get install -y git vim
```



Once that completes, you will be back at the root@ubuntu prompt and you can clone the Git repo which contains the scripts for installing. Type the following command to download the files:

```
git clone https://github.com/discoposse/OpenStack-All-in-One-Havana.git
```

Once the files have been cloned, change directory using the following command (Remember that these are case sensitive!):

```
cd OpenStack-All-in-One-Havana
```

```
root@ubuntu:~#
root@ubuntu:~# git clone https://github.com/discoposse/OpenStack-All-in-One-Havana.git
Cloning into 'OpenStack-All-in-One-Havana'...
remote: Counting objects: 14, done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 14 (delta 4), reused 14 (delta 4)
Unpacking objects: 100% (14/14), done.
root@ubuntu:~#
root@ubuntu:~# cd OpenStack-All-in-One-Havana/
root@ubuntu:~/OpenStack-All-in-One-Havana#
root@ubuntu:~/OpenStack-All-in-One-Havana# _
```

Now we will set some system variables which are used by the preparation and install scripts. The information that you need to know is the IP address that you will use, and the gateway address.

Type the following two commands using your IP info where the xx.xx.xx.xx is listed (see image below):

```
export INTERNAL_IP=xx.xx.xx.xx
```

```
export INTERNAL_GW=xx.xx.xx.xx
```

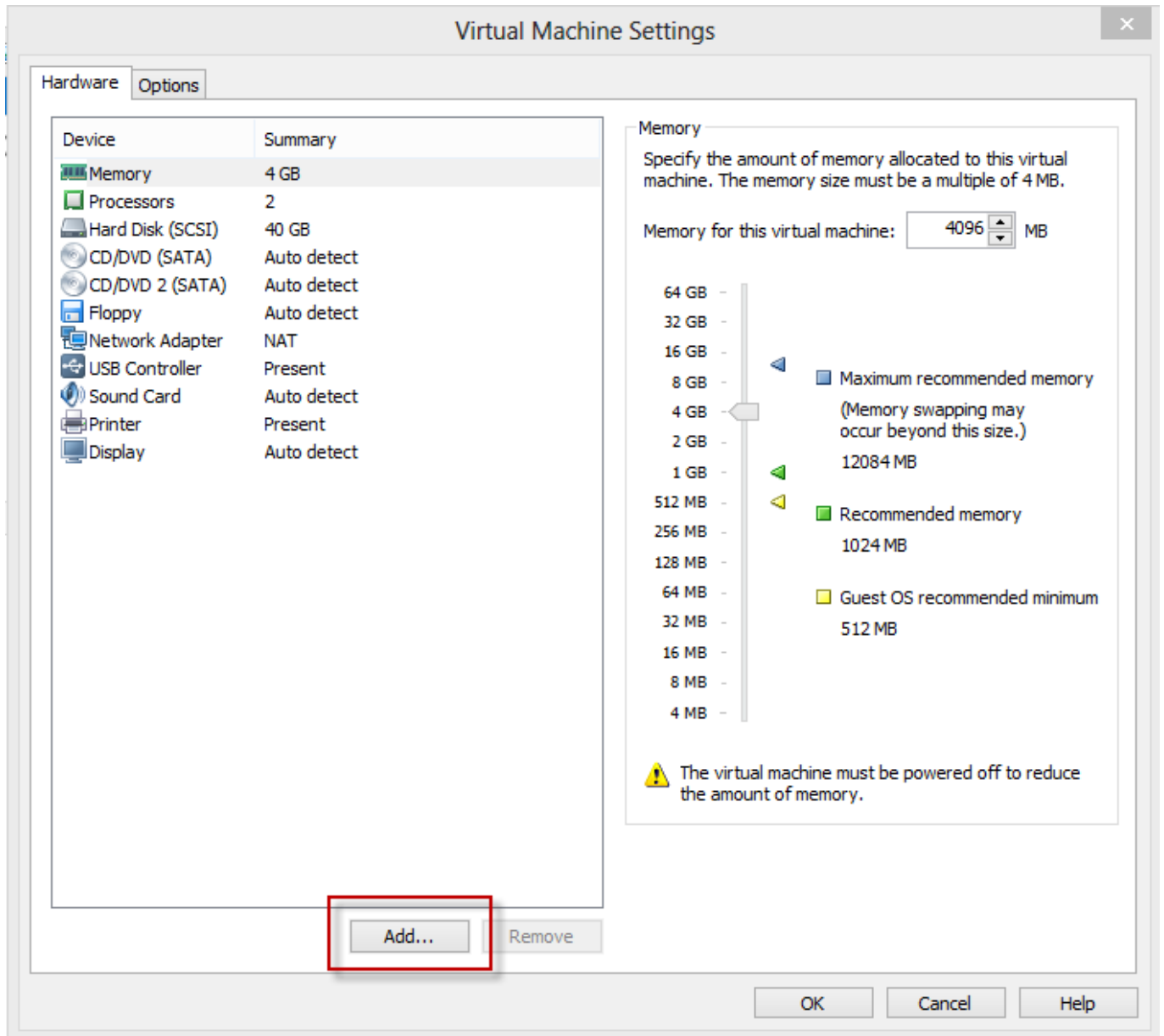
Once those variables are set, we will run the preparation script using this command. Note that the preparation script will set the IP network information and then shut down your VM which we need to do for the step following this:

```
sh prepare.sh
```

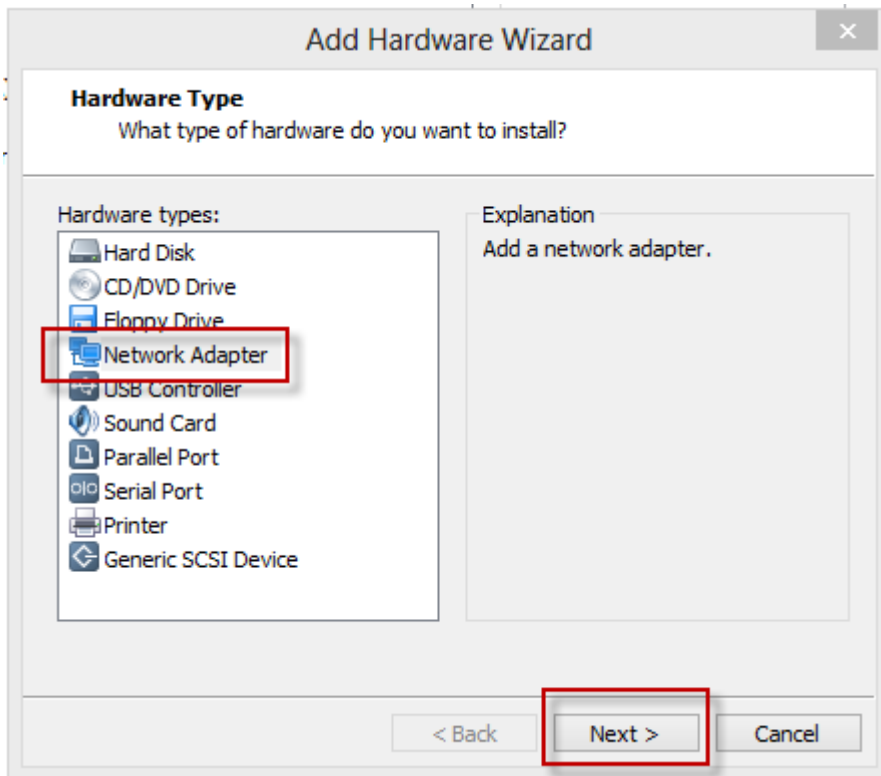
```
root@localhost:~/OpenStack-All-in-One-Havana#
root@localhost:~/OpenStack-All-in-One-Havana#
root@localhost:~/OpenStack-All-in-One-Havana# export INTERNAL_IP=192.168.79.50
root@localhost:~/OpenStack-All-in-One-Havana# export INTERNAL_GW=192.168.79.2
root@localhost:~/OpenStack-All-in-One-Havana#
root@localhost:~/OpenStack-All-in-One-Havana# sh prepare.sh _
```

When the preparation script completes the shutdown of your VM, edit the hardware settings and click the **Add** button:

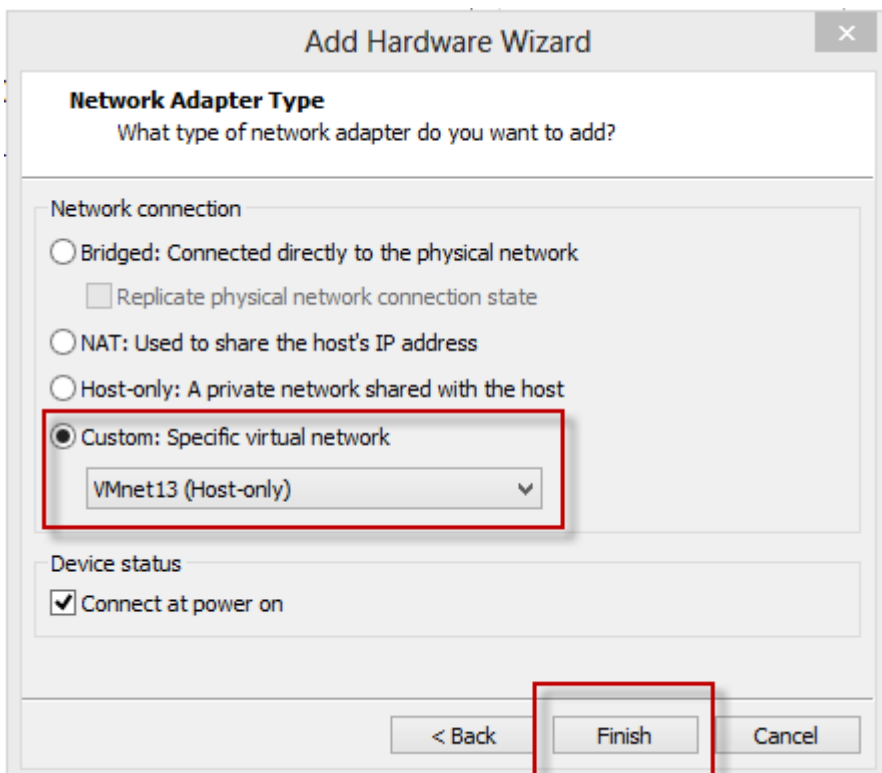




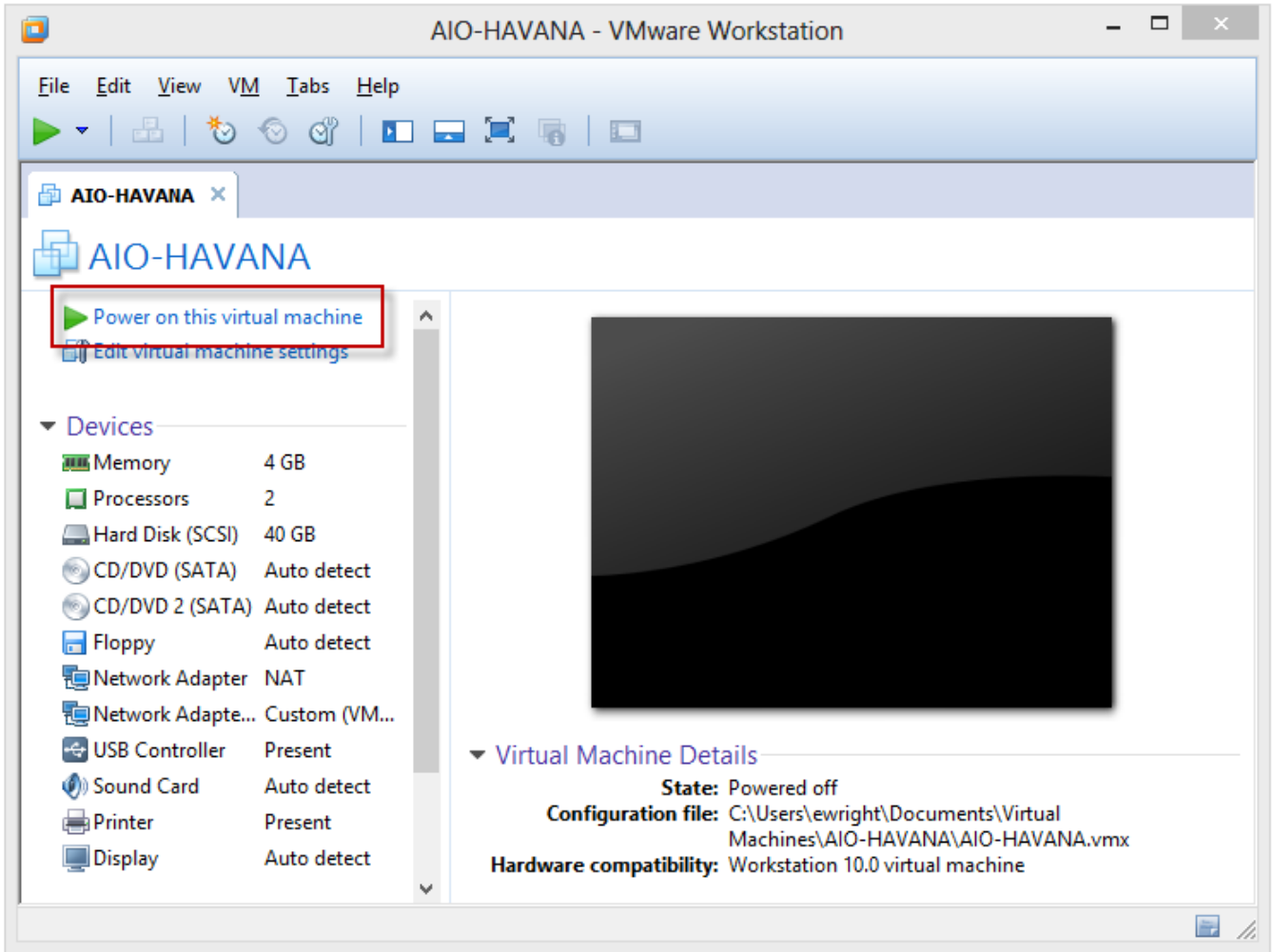
Select **Network Adapter** as the hardware type and click **Next**:



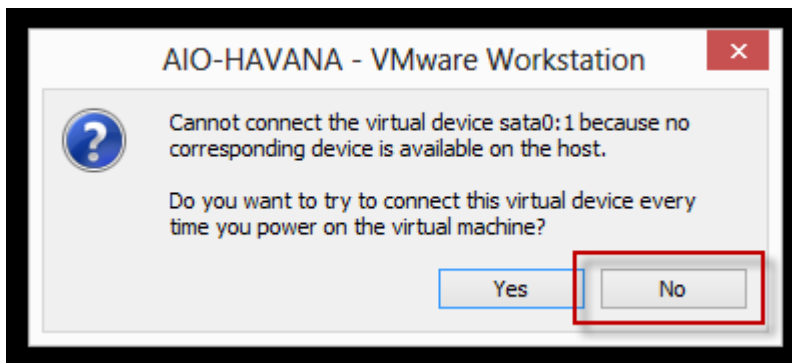
For the Network Adapter Type, choose **Custom: Specific virtual network** and pick the network from the drop list that is our 10.10.100.0/24 segment and then click **Finish**:



Now that we have added our spare network adapter, power on the VM again:



You will be prompted about hardware devices that are not available. This is because we aren't mounting the CD-ROM ISO file. There will be either one, or two prompts. Click **No** for each and then the machine will boot up:



Now we will see the login prompt which shows the hostname is **aio-havana**. Log in with the credentials you used in the first part of the build and we will start the OpenStack install script next.

## Half way there!

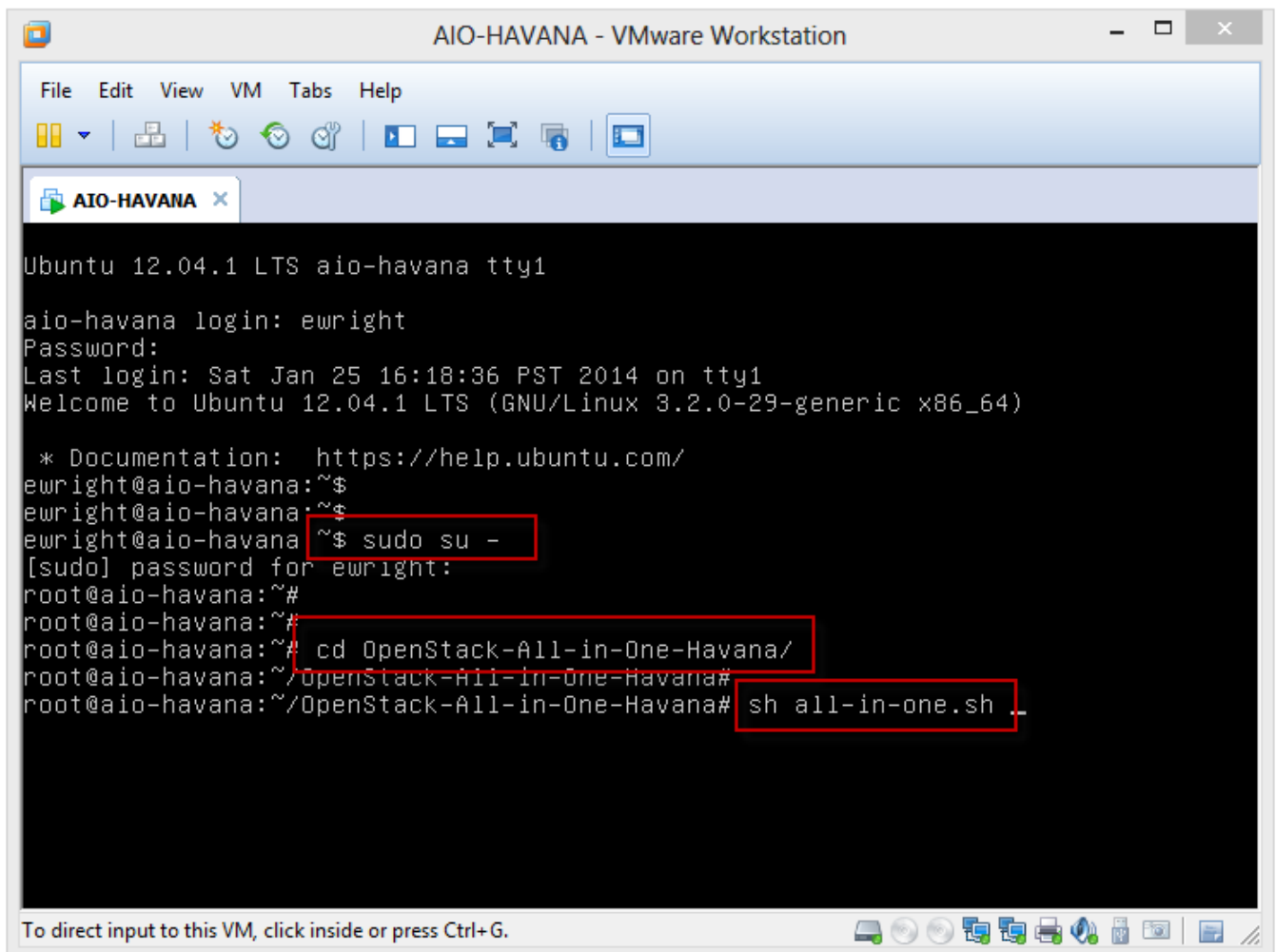
You've done the first, very important steps to prepare your VM environment. Take a breather, and get ready for the next stage of the installation. Now we will log back into our VM and initialized the build script which was downloaded from the Git repository during the preparation steps.

Log on to your VM with your credentials, and we will run the following commands:

```
sudo su -
```

```
cd OpenStack-All-in-One-Havana
```

```
sh all-in-one.sh
```



The screenshot shows a VMware Workstation window titled "AIO-HAVANA - VMware Workstation". Inside, a terminal window titled "AIO-HAVANA" displays the following text:

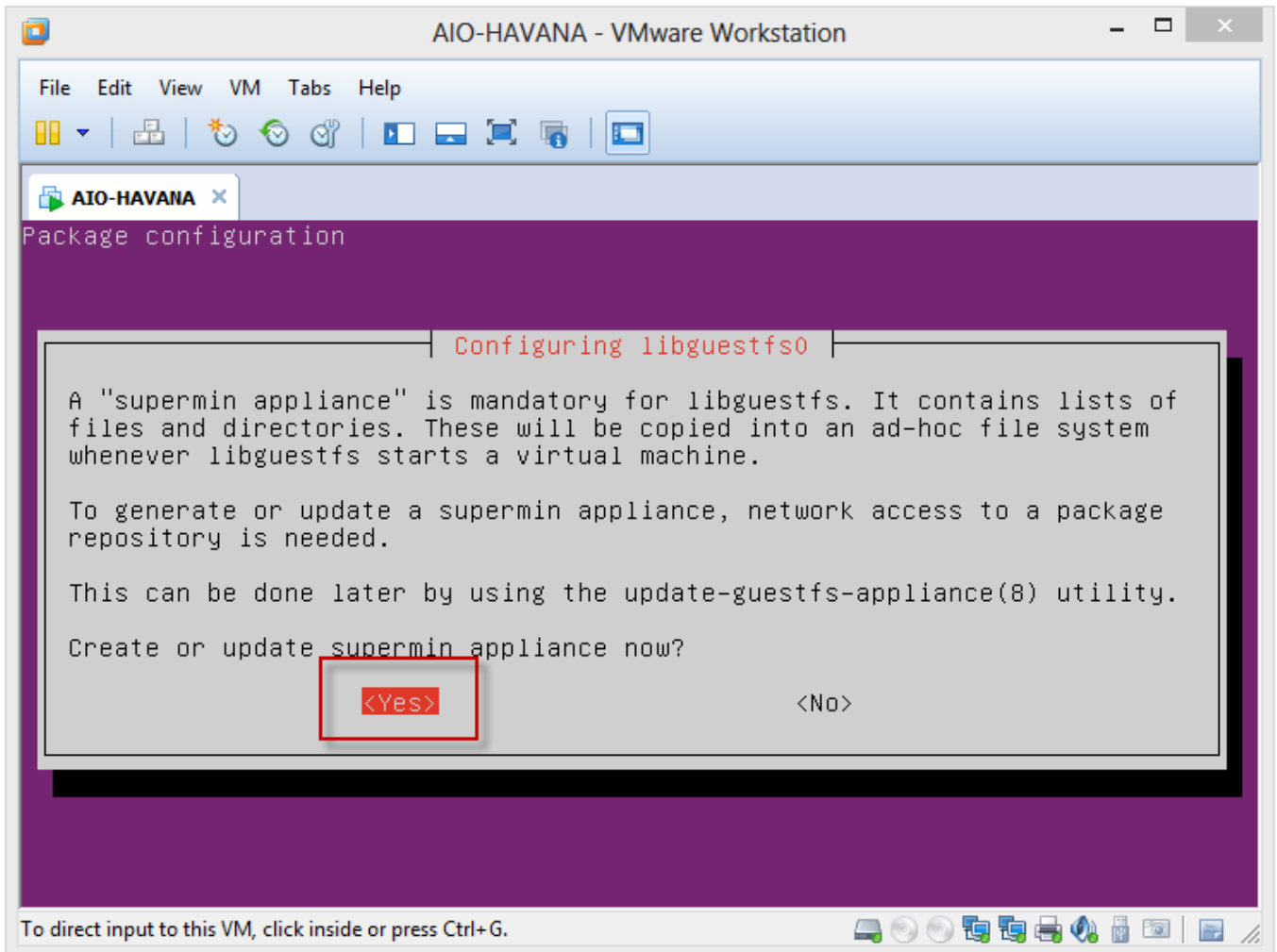
```
Ubuntu 12.04.1 LTS aio-havana tty1
aio-havana login: ewright
Password:
Last login: Sat Jan 25 16:18:36 PST 2014 on tty1
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-generic x86_64)

* Documentation:  https://help.ubuntu.com/
ewright@aio-havana:~$
ewright@aio-havana:~$
ewright@aio-havana:~$ sudo su -
[sudo] password for ewright:
root@aio-havana:~#
root@aio-havana:~#
root@aio-havana:~# cd OpenStack-All-in-One-Havana/
root@aio-havana:~/OpenStack-All-in-One-Havana#
root@aio-havana:~/OpenStack-All-in-One-Havana# sh all-in-one.sh
```

Red boxes highlight the commands: `sudo su -`, `cd OpenStack-All-in-One-Havana/`, and `sh all-in-one.sh`.

The install script will start, and lots of exciting OpenStack goodness will be scrolling by for the next 5-15 minutes depending on the speed of your hardware and network hosting your VM.

At some point you will see this screen prompting for what is called a "supermin appliance". Just click into the console, use the arrow key to move the option to Yes and press Enter:



The install script will keep working away for another few minutes and will reboot when it is completed. Once that has happened, you should have a fully deployed OpenStack Havana environment to start your lab work with.

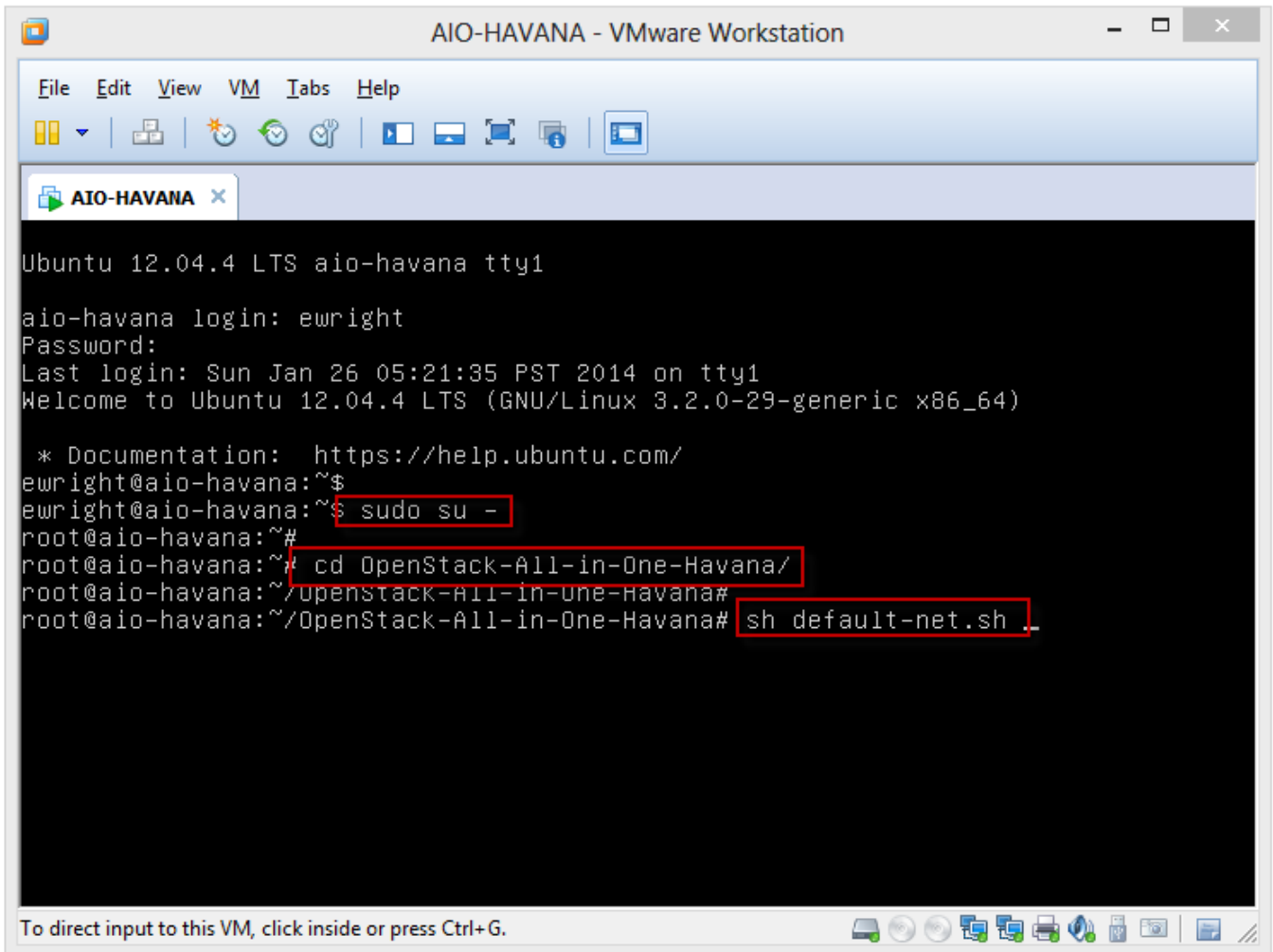
## Configuring your OpenStack Havana Install

A few more steps are required to configure OpenStack to allow for your first instance to be booted. Log in with your credentials and run the following commands:

```
sudo su -
```

```
cd OpenStack-All-in-One-Havana
```

```
sh default-net.sh
```



That script does the following:

- Create a flat DHCP network for our guest VMs (10.10.100.0/24)
- Add a firewall rule to allow telnet (TCP 22) from all source IP addresses
- Add a firewall rule to allow ping (ICMP) from all source IP addresses

When you run the script, you will see this output:

```
+-----+-----+-----+-----+-----+
| IP Protocol | From Port | To Port | IP Range | Source Group |
+-----+-----+-----+-----+-----+
| tcp         | 22        | 22      | 0.0.0.0/0 |               |
+-----+-----+-----+-----+-----+
| IP Protocol | From Port | To Port | IP Range | Source Group |
+-----+-----+-----+-----+-----+
| icmp        | -1       | -1     | 0.0.0.0/0 |               |
+-----+-----+-----+-----+-----+
```

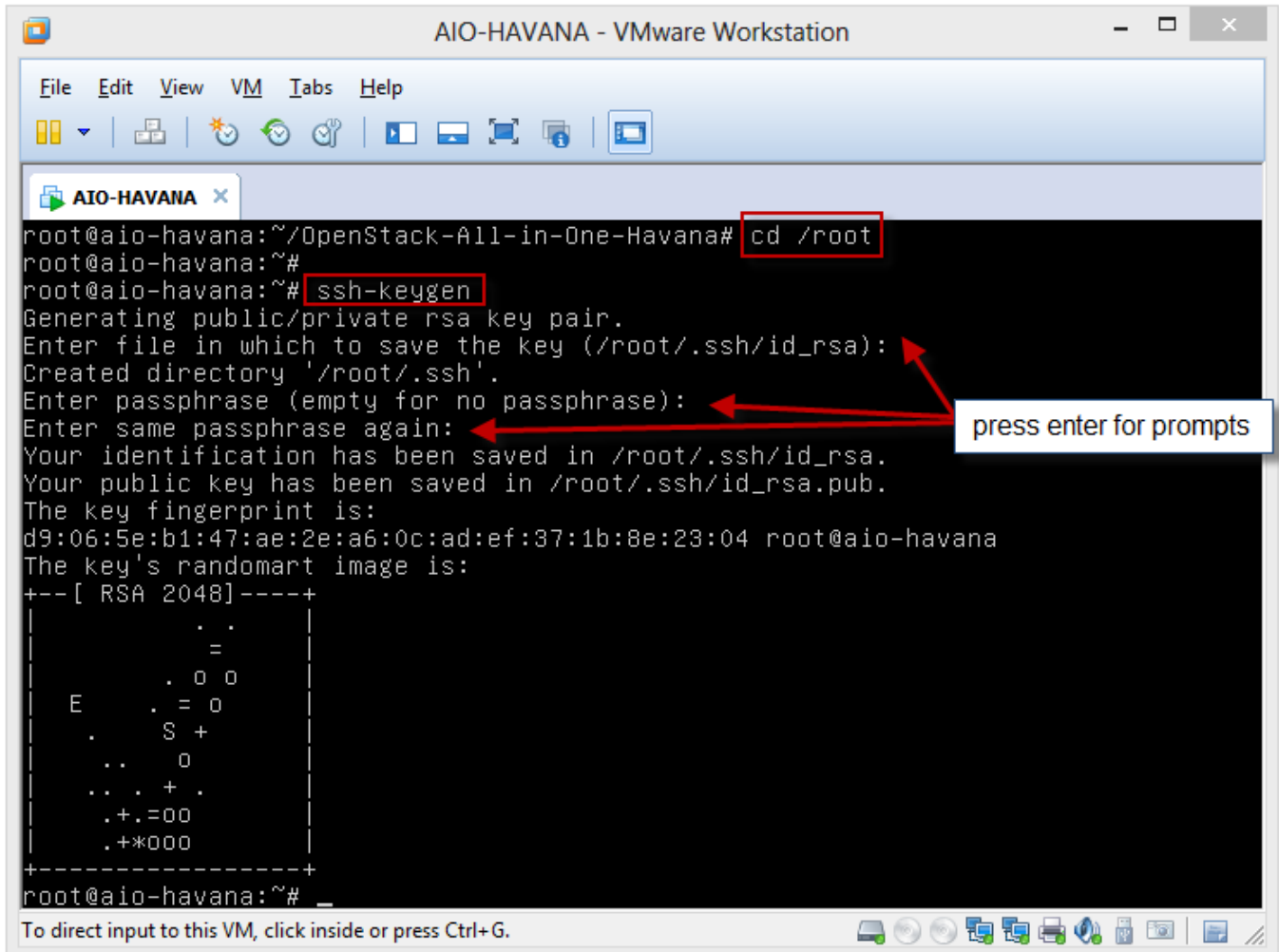
Once that is done we are ready to do the final pre-launch step which is to create an SSH keypair on our host and load it into Nova.

## Creating your SSH Keypair

Generating an SSH key is quite simple. We will run the following commands to do that:

```
cd /root
ssh-keygen
```

When we are prompted by the SSH keygen for questions, just press Enter to take the defaults for file name and location:



```
AIO-HAVANA - VMware Workstation
File Edit View VM Tabs Help
AIO-HAVANA x
root@aio-havana:~/OpenStack-All-in-One-Havana# cd /root
root@aio-havana:~#
root@aio-havana:~# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
d9:06:5e:b1:47:ae:2e:a6:0c:ad:ef:37:1b:8e:23:04 root@aio-havana
The key's randomart image is:
+--[ RSA 2048 ]-----+
|          .          |
|         =          |
|        . 0 0       |
|       . = 0        |
|      S +          |
|     . 0           |
|    . . + .        |
|   .+. = 00        |
|  .+*000           |
+-----+
root@aio-havana:~# _
To direct input to this VM, click inside or press Ctrl+G.
```

This creates a hidden directory (`/root/.ssh`) where it stores our file that we will now load named `id_rsa.pub` in the new directory. We will name it **mykey** just for an easy name to remember.

To import this keypair for use in OpenStack, we run the following commands:

```
cd .ssh
nova keypair-add -pub_key id_rsa.pub mykey
nova keypair-list
```

```
root@aio-havana:~#
root@aio-havana:~# cd .ssh
root@aio-havana:~/ssh# nova keypair-add --pub_key id_rsa.pub mykey
root@aio-havana:~/ssh#
root@aio-havana:~/ssh#
root@aio-havana:~/ssh# nova keypair-list
+-----+
| Name | Fingerprint |
+-----+
| mykey | d9:06:5e:b1:47:ae:2e:a6:0c:ad:ef:37:1b:8e:23:04 |
+-----+
root@aio-havana:~/ssh# _
```

The **nova keypair-list** command confirms that our key has been created and loaded into Nova. We finally have our lab ready for use.

Coming up next in a series of posts are:

- [Booting your first Nova instance](#)
- [Using the Horizon Dashboard](#)
- [Adding OpenStack Block Storage \(Cinder\)](#)

Stay tuned, and I hope that this will be helpful to get you on your journey to discovering OpenStack!

---

## [Open Source - Free Market Capitalism of Technology](#)

✘ Before I get started, I want to say that I am a huge supporter of open source technology, open standards and I firmly believe that using open source to innovate is one of the most advantageous ways to advance a software or infrastructure product.



As I bring more and more people into great open source technology projects such as Puppet and OpenStack, there are a few challenges that can become a barrier to certain individuals, and their organizations, in adopting the tools. Most often, the issue comes with support and the development roadmap. What comes to the surface quickly in these cases is that the openness does not automatically imply support and adaptability.

With proprietary, closed source products (e.g. vSphere, vCenter Orchestrator, SCCM, Hyper-V) we have a fairly fixed set of support and the drivers are listed for known hardware and software compatibility. Additional support can be requested as a feature request, but generally the packaged product is a WYSIWYG (What You See is What You Get) technology and we work within its limitations.

The hot term lately is the “one throat to choke”. It’s the idea of a vendor model where you have a single place to contact for purchase and support for major portions of your infrastructure. This is usually only available in closed source, or what are called “walled garden” types of products and services.

## **Open Innovation**

Using open source technologies, you have the ability to use the known drivers and interfaces, plus there is the bonus of being able to develop the product and advance the technology as you need to. But the challenge is that word “develop”.

For many people and organizations, developing drivers, shims and other enhancements into a product may not be in their capability. These skills are over and above what many can effectively do.



The community is active, but there is no guarantee that the particular feature or driver you are in need of is in active development. A road paved with hope is a dirt road, so there is no way to force the integration of the product feature that you are after, and there may even be risk of regression of current support and compatibility as the core product changes.

## **Free Market isn't the Panacea**

In the economic world, there is a strong support for the use of free market capitalism to grow the economy. This is absolutely the best method to significantly advance the economy and flourish throughout the world...in theory. In practice however, the idea that if we reduce taxes on high earners to allow them to put money back into the system for all to use is flawed by the human condition.

The proven history shows that the money doesn't flow down in the way that the theory has dictated it should, and of course, there are all sorts of people to blame for this from all sides.

On the other side of it, the goal of free markets is that they can also freely fail. It is entirely possible for the economy to tumble for a number of reasons, and the conundrum is built where the owners of the wealth want to be protected from failure.

How does this compare to open source development though?

## **It's your fault the book isn't finished**

Imagine a book that was given to you with 200 pages in it, but only 140 of them are completed and the story has missing components that you would have thought to be required. You approach the author on the issue and in response, they hand you a pen and say that you have to finish the book, and it's your fault that it isn't good right now because you haven't put the time in to finish it.

Even better, you write the finale to the book, and then a sea of complaints on your choice of writing come in to your inbox. The open source methodology has now made you a part of the system; but just like free market capitalism, it may not always work in your favor.

Quite often we find that there is a "works in my environment, so it must be localized to your configuration" statement, and you are left searching for answers to make the product fit correctly for your needs.

So in choosing a product based on its potential to be grown, adapted and advanced, you ultimately are still left with the basic abilities of the core and unless you take on the task, or find someone who can do it for you. As a result, the product will not innovate in the way that we have been led to believe that it can go.

## **Fragmentation Nightmares**

The Android OS is a prime example of one of the key challenges in an open source environment. Each telco has branched off a piece of the code and is deviating from the core in order to support their very specific needs and lock you into the version in order to reduce their support impact.

The very thing that we loved about open source has been broken in this case. The openness has been relinquished for vendor lock-in.

## **The Forgotten Ones**

Another challenge is that you are working with a plug-in, module or drive that had community development, but you look at the last commit date in GitHub: December 2011. Much has happened since then, and with some core system update this custom update no longer works, but the project owner has abandoned it.

## **What are we getting to with this?**

To summarize, I wanted to highlight the highs and lows of open source products, and what the impact may be in your environment. I am pretty sure I'll take a beating from a few with some of the more general statements made here, but it is important that you, as an architect, are aware of risk factors.

The FUD flies both ways of course, and the walled garden is no panacea either. So when you make your decision, just flavor to taste and be aware.

---

# [Is OpenStack a cloud? Thoughts on Subbu Allamaraju's post](#)

As you may have seen from Tweets floating around today, there was an interesting post written by Subbu Allamaraju, Chief Engineer for the cloud program at eBay, titled "OpenStack is not cloud" - <http://www.subbu.org/blog/2013/07/openstack-is-not-cloud>

This article brings up some very interesting questions about how we define the litmus test for a cloud, and how OpenStack is measured against these criteria.

## **AWS is a cloud**

This is a true statement. We use AWS as a benchmark for most other public cloud offerings mostly. The AWS ecosystem has grown and matured over the years to be a fully-featured environment which has expanded to cover every component of the \*aaS model.

If we think about how AWS came to be, it is not unlike what OpenStack is, but just a few years older. AWS grew out of the internal infrastructure that Amazon used to provision to its own organization. The choice came to bring the product to the public in 2004, and with the launch of the SQS (Simple Queue Service) the first AWS public offering was in production.

## **OpenStack 2013 is AWS 2006**

Recall that AWS is now nearly 10 years in existence, so is it really proper for us to measure OpenStack, a 3 year old product, against AWS in its current form? I think that we need to make sure that we measure the current OpenStack release (Grizzly) against the 2006 version of AWS to really get a sense of the comparison.

Also, remember that AWS has different origins. AWS is a walled garden in a sense because we (community developers) do not contribute to the product. It is, and only will be developed by Amazon, and we are consumers of the service in the same way that I don't know how my shirt was sewn, but I know I can go to any American Eagle store and pick one up.

## **Comparing Apples to Apple Sauce**

If we compare AWS to OpenStack, it really isn't a fair comparison in a lot of ways. We (you and I as consumers) don't build AWS. We consume AWS. The key difference is that for us to adopt OpenStack, we are building the infrastructure ourselves. For me to use AWS, it is already prepared. In effect, it is the apple sauce. The product has already been processed for consumption.

When I am deploying OpenStack, I am picking apples. The community has built the tree, and I consume the product at that layer. I then take the apples and prepare the product, or in this case the cloud, and build my solution to present to my consumer.

Is there a lot of work there? Yes. Is there a well documented upgrade path to the next revision of OpenStack? No. Was there for the internal AWS team in 2006? Great question. This is why this is an important discussion, and why we need to see what OpenStack is meant to deliver, and what they deliver today. As mentioned in the article, there are clearly some services that are needed in OpenStack in its current form to make it more effective.

## **Rackspace is cloud**

Subbu summed things up very well, and I agree with a lot of the detail that he posted. But I think the article could really be titled “OpenStack is not cloud. Rackspace is cloud”. The only true comparison to AWS in relation to what drives is to view Rackspace as the consumable product built on OpenStack.

We will look towards Eucalyptus to bring the private cloud comparison, but again this is a difficult A/B test because there are clear and fundamental differences to what is being brought to the table by OpenStack versus what Eucalyptus is doing with AWS compatibility.

## **Thank you Subbu!**

At risk of sounding wishy-washy, I have to say that I both agree with Subbu’s post, and disagree at the same time. In spirit, he highlights some of the shortcomings of what OpenStack is when we install it ourselves for public or private cloud deployments. But we have to view all of the factors that draw us to those conclusions.

OpenStack is a positive disruption at the very least. The potential is great, and although it won’t be the “AWS killer”, nothing else will either. That being said, I encourage you to give it a try, and grow along with it as OpenStack matures. Remember that the Havana release is coming in the fall, and along with it some significant growth in the service offering.

Thanks to Subbu for sparking some great conversation, and I look forward to seeing how this all goes in the coming months!

---

## **VMware or OpenStack? Don’t choose one; choose both!**

✘ In the ever evolving world of virtualization, we are faced with new and exciting changes all the time. As a long time VMware customer and advocate, I am obviously very attached to the entire VMware ecosystem and the community that surrounds it.

Recently I’ve been doing a significant amount of OpenStack work, and I’m even writing a book on the subject, so that should indicate just how committed I’ve become to that environment. OpenStack is dubbed the “open cloud” and is clearly gaining recognition by the virtualization community as an upcoming player for public and private cloud deployments.

So the question has come up a few times to me: “Are you switching from VMware to OpenStack?”. I have to say no, but at the same time I want to be clear that the question shouldn’t have been “Are you switching?”.

## **Co-opetition**

A popular word that’s been used with this sort of situation is co-opetition. This is a cross between co-operation and competition. What OpenStack and VMware do may be similar, but it is also very

different in a lot of ways.



Two different views, but not that different after all.

What I'm doing by diving into the OpenStack world is to get a deeper view of how it can compare and contrast to VMware technologies in the same way that I study Citrix and Microsoft virtualization systems. It is part of our role as architects to be able to fully understand what each part of the product means to a current or potential customer.

## **OpenStack Challenges**

There are a few distinct challenges in working with OpenStack. I like to say challenges, because they are solvable, or sometimes a non-issue altogether depending on your situation. These are simply talking points that get brought up most often when I'm discussing OpenStack with people.

- Volatility - Although it is in its 7th release with Grizzly this spring, there are concerns (some genuine) that the environment is volatile. What does volatile mean though? We will discuss below.
- Host OS support - Sorry WIntel folks, this one only works on Linux. Unlike vSphere though, which is also Linux based, this requires administration of the OS independent of the product and hypervisor.
- High Availability and Disaster Recovery - There are deployment methods to provide HA and DR scenarios, but many of the techniques are lacking full support, and require the use of other third party plugins and products to architect always-on solutions in certain scenarios.
- Vendor Support - This is the classic. Whether hardware vendor, software vendor, or consultant support, the ever wanted "five 9s" up-time and a 1-800 number to get 24/7/365 access for support makes some folks nervous about putting their production workloads in an OpenStack environment that they have to manage.

Again, these are only challenges for certain customers. Plus, we also said the same thing about VMware ESX early on, and the same stood for Hyper-V in its first iterations.

## **Why choose?**

This is the real question that needs to be asked. And the answer is easy: "Choose both!" There are probably use-cases for both products in your environment. And if there isn't today, you may find that in your current role, or in your future role you will inevitably be involved with looking at OpenStack somewhere along the way.

I myself am a big cloud proponent, but I am also fully aware of where cloud products are not the appropriate product for certain use-cases. In the same way, I will be doing similar evaluations of OpenStack, and just like other cloud products, it may not be the right fit.

The key to take away from this is that you don't need to have a vested interest in the product, but you should have a vested interest in your knowledge base to be able to offer the best solutions to your employer or customer for a virtualization product.

## **Don't Hesitate, Orchestrate**

The side-effect of dabbling with technologies like OpenStack, vCloud, vCloud Hybrid Service and other great products, is that we begin to do exciting things internally to prepare ourselves to consume our IaaS environment. In order for us to get the most out of cloud deployment methodologies, we have to first build in the orchestration and automation processes that feed our cloud service.

Even if you aren't on the fast track to a cloud, whether public or private, you should be evaluating how you can improve your operational model in IT. You don't necessarily have to be a full DevOps environment, but making the move towards orchestrated builds will be the stepping stone to your successful cloud adoption when it does come.

## **Is OpenStack the <insert cloud vendor here> Killer?**

As we've learned from the mobile phone market, the long sought after "iPhone killer" has never come. What has come though is a relatively strong share of the mobile marketplace being pared down and split among other strong competitive vendors.

So the questions coming in about whether OpenStack is the AWS killer, or the vCloud killer, are really all just noise which creates headlines. Will OpenStack cut into the market share of those, and other vendors? Absolutely. Will it be a significant impact? We have to wait and see, but all signs point to this being a strong player in both the public and private cloud space.

It certainly creates buzz when we see posts like the open letter from Randy Bias at Cloudscaling come out calling for OpenStack to adopt AWS and other comparative compute engines as the core for public cloud deployment. That letter (<http://www.cloudscaling.com/blog/cloud-computing/openstack-aws/>) tells you that this is a disruptive movement for sure. But, disruptive in this sense is a very positive move.

## **Get Knowledge, Make an Informed Decision**

Now is the time to get involved, and to dig into the technology so that when the time does come to evaluate OpenStack or vCloud that you have all the information needed to make an informed decision.

A great resource for you to get started is the Couch to OpenStack series offered by the #vBrownbag community. The weekly Podcast run live on Tuesday evenings at 8PM Central time are doing a ground up installation of an OpenStack private cloud and you can view the archived sessions as they are uploaded in case you can't attend the live sessions: <http://openstack.prov12n.com/vbrownbag-podcast-couch-to-openstack/>

I look forward to sharing the journey with vCloud, OpenStack and many other exciting products over the next few months. If you have any questions or comments, feel free to drop me a line at [twitter.com/DiscoPosse](https://twitter.com/DiscoPosse) or through the comments on this site.