Using Juju Charm to Configure and Deploy an HPCC on AWS

Part I

INTRODUCTION

This is part one of a two part series on Using Juju Charm to Configure and Deploy an HPCC on AWS (Part I) and Understanding Juju Charm and the Scripts that Configure and Deploy an HPCC on AWS (Part II).

The main reason for presenting Juju Charm is because it gives you flexibility in configuring and deploying HPCC Systems without requiring you to know a lot more than you currently do.

What is HPCC?

HPCC (High Performance Computing Cluster) is a massive parallel-processing computing platform that solves Big Data problems. The platform is Open Source!

The HPCC Systems architecture incorporates the Thor (data refinement) and Roxie (data delivery) clusters as well as common middleware components, an external communications layer, client interfaces which provide both end-user services and system management tools, and auxiliary components to support monitoring and to facilitate loading and storing of file system data from external sources. An HPCC environment can include only Thor clusters, or both Thor and Roxie clusters.

What is Juju charm?

Juju is a next generation service orchestration framework. It has been likened to APT for the cloud. With Juju, different authors are able to create service formulas, called charms, independently, and make those services coordinate their communication and configuration through a simple protocol.

For this presentation, the only parts of Juju that you will interact are:

1. The file ~/.juju/environments.yaml which sets cloud provider specific parameters.
2. The file ~/juju-charms/precise/hpcc/config.yaml which contains parameters for the HPCC System we deployed.
3. The Juju commands used to configure and deploy the HPCC on AWS.

This presentation assumes that you will run Juju charm on an Ubuntu 12.04 linux machine. If you don't have an Ubuntu 12.04 linux machine then you can download and install a VMware image of one and run it on a VMware Player, which can also be installed. You will find instructions in Appendix A.

In what follows, commands given in an Ubuntu terminal window are shown with a gray background. For instance.

```
sudo apt-get install git
```

In addition, throughout this presentation, for screenshots, I will mark lines being discussed with a green arrow, like this one.

	When you are done with your deployed HPCC System shut it down so AWS charges don't continue to accumulate.

SETUP FOR USING JUJU CHARM ON UBUNTU 12.04 LINUX

Installing the HPCC Juju Charm

The HPCC Juju Charm enables one to configure and deploy an HPCC System, with both THOR and ROXIE clusters, to Amazon's AWS.

If you don't have git installed on your Ubuntu linux machine them the following command does that.

```
sudo apt-get install git
```

	If you followed the instructions of Appendix A to build a virtual Ubuntu machine, then after entering the above sudo command, you will be asked to enter a password. Enter the word "password" (without quotes).

Next install the HPCC Juju charm by cloning it from github.com.

```
git clone https://github.com/xwang2713/juju-charms.git
```

Here is a screenshot of the directory structure created by the above clone.

```
ubuntu: ~                                    ✉ ⇅ ◀ɪ) 3:32 PM ⚊ tlhumphrey2 ⚙
user@ubuntu:~$ tree juju-charms
juju-charms
└── precise
    └── hpcc
        ├── bin
        │   ├── config_hpcc.sh
        │   ├── get_url.sh
        │   ├── parse_config.py
        │   └── parse_status.py
        ├── config.yaml
        ├── copyright
        ├── dependencies
        │   ├── el6
        │   ├── precise
        │   ├── saucy
        │   └── trusty
        ├── hooks
        │   ├── config-changed
        │   ├── hpcc-cluster-relation-changed
        │   ├── hpcc-cluster-relation-departed
        │   ├── hpcc-cluster-relation-joined
        │   ├── hpcc-common
        │   ├── hpcc-relation-changed
        │   ├── install
        │   ├── install~
        │   ├── start
        │   └── stop
        ├── icon.svg
        ├── metadata.yaml
        ├── README.md
        └── revision
└── README.md
```

	The command, tree Juju-charms, will create the above directory structure. If you don't have tree then you can download it with the following command: sudo apt-get install tree

Installing Juju

To install Juju, you simply need to grab the latest juju-core package from the PPA, update the packages on your Ubuntu system and apt-get install juju-core.

```
sudo add-apt-repository ppa:juju/stable
sudo apt-get update
sudo apt-get install juju-core
```

To configure and deploy an HPCC System to Amazon's AWS, you only need to install juju-core. But, installing juju-local gives you the capability to configure and install an HPCC System on your local Ubuntu machine and thereby not incur costs associated with Amazon's AWS. The following command installs juju-local.

```
sudo apt-get install juju-local
```

	The installation of both juju-core and juju-local will take a few minutes and generate a lot of output line.

You can check if juju-core and juju-local were installed by doing the following command:

```
dpkg -l|egrep "juju-core|juju-local"
```

Here is a screenshot of what you should see:

Configuring Juju Charm

So Juju is almost ready to configure and deploy an HPCC System to Amazon's AWS. There are two files that need to be changed -- config.yaml and environments.yaml.

First, use your favorite editor to update the config.yaml file with the name of the version of the HPCC Platform to deploy and its checksum. The following command places config.yaml in vi.

```
vi ~/juju-charms/precise/hpcc/config.yaml
```

The following is a screenshot of config.yaml in the vi editor. The two green arrows point to the lines that need to be changed to configure and deploy a different HPCC Platform. Change the version number, in the top line, to the version you want to deploy. Change the checksum in the 2nd line to the checksum of the version you want to deploy.

Here is a list of HPCC Platform versions, with their checksums, that can currently be configured and deployed to Amazon's AWS. There will be other versions deployable soon. Please watch our web site, hpccsystems.com.

AWS Deployable versions of HPCC	Checksum
4.2.0-4	bc2fb0ad5f9fa1eaf7a33483ec31bfa3

5.0.0-1	2ce1265e564a2ff3d99c51c87ea57fe1
5.0.0-3	a1a1b06886e9229cb2d01bdf8c9945e4

Also, you need to generate and edit the environments.yaml file, which will live in your ~/.juju/ directory. You can generate the environments.yaml file manually, but Juju also includes a boilerplate configuration option that will flesh out most of the file for you and minimize the amount of work (and potential errors).

To generate an initial environments.yaml file, you simply need to run:

```
juju generate-config
```

The environments.yaml file contains sample profiles for different types of cloud services (including Amazon's). You need to edit the file to provide specific information for AWS. For now, you only need to make two changes to the file, i.e. insert your AWS access key and secret key. So, to bring up environments.yaml in vi, do the following:

```
vi ~/.juju/environments.yaml
```

Here, is a screenshot of environments.yaml in vi. It is opened to the amazon section where you will insert your AWS access key and secret key (where the green arrows are pointing). For both lines, you replace <secret> with your aws access-key and secret-key. Make sure you uncomment both lines.

	In the above screenshot there is also a place where you can change the AWS region that you deploy to. It is currently commented out, i.e. the line with "region: us-east-1" on it. If you want to change the region, this is the place to do it.

CONFIGURE AND DEPLOY AN HPCC ON AWS

Now, Juju and the HPCC Juju charm are ready for you to configure and deploy an HPCC on AWS.

Because Juju can talk to any number of CLOUD systems, including your local machine, you must first make sure that Juju is setup to talk to AWS. The following command checks to see which CLOUD system Juju's commands are currently directed to:

```
juju switch
```

The following is a screenshot of what is returned if Juju is setup to talk to amazon's AWS.

```
user@ubuntu:~$ juju switch
amazon
user@ubuntu:~$ 
```

If Juju is not currently talking to Amazon, then you can switch over with the following command.

```
juju switch amazon
```

```
user@ubuntu:~$ juju switch amazon
amazon -> amazon
user@ubuntu:~$ 
```

You start the deployment process by issuing a bootstrap command.

```
juju bootstrap
```

You can watch the bootstrapped instance being deployed by going to console.aws.amazon.com to the Instances page of the EC2 service section. In the following screenshot, the green arrow points to the instance being bootstrapped.
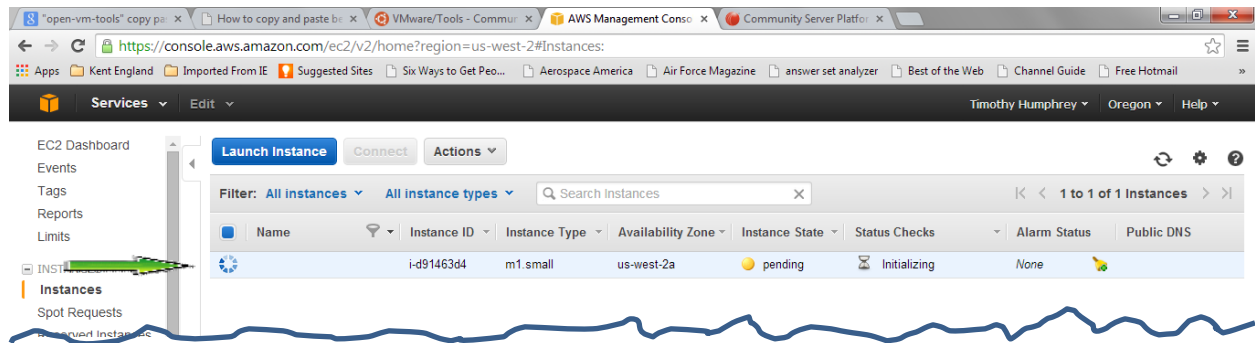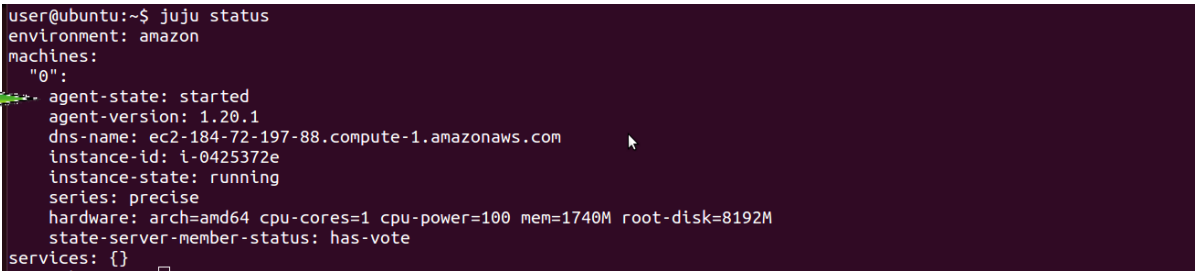
⚠	You may have to hit refresh several times before the deployment starts.

The bootstrap will take a few minutes. And, it outputs several lines to the Ubuntu terminal window. Once the bootstrap is done, the command prompt is displayed and you can enter the status command.

```
juju status
```

The following screenshot shows what the juju status command outputs to the terminal once the bootstrapped instance has started.


```
user@ubuntu:~$ juju status
environment: amazon
machines:
  "0":
    agent-state: started
    agent-version: 1.20.1
    dns-name: ec2-184-72-197-88.compute-1.amazonaws.com
    instance-id: i-0425372e
    instance-state: running
    series: precise
    hardware: arch=amd64 cpu-cores=1 cpu-power=100 mem=1740M root-disk=8192M
    state-server-member-status: has-vote
services: {}
```

⚠	It is important that you procedure only after the state of the instance is "started" (see green arrow below).

To deploy the HPCC System, you issue a juju deploy command. But, first cd into the juju-charms directory. Then issue the deploy command. The following juju deploy command deploys the HPCC System to AWS instances of type m3.large (default is m1.small). The command says that the instructions (scripts and configuration files) are in the sub-directory precise/hpcc. The name of our deployed HPCC System will be tlh-hpcc (this name can be any legal name). And, the number of nodes of the THOR deployed is 2, one master and one slave (if n was larger than 2 then there would be one master and the rest slaves). Also, deployed is a Roxie with n-1 nodes or in this case, 1.

```
cd ~/juju-charms
juju deploy --constraints="instance-type=m3.large" --repository=. local:precise/hpcc tlh-hpcc -n 2
```

⚠	Copying the above two commands and then pasting them into your Ubuntu terminal window may not work and possibly may cause errors in the deployment.

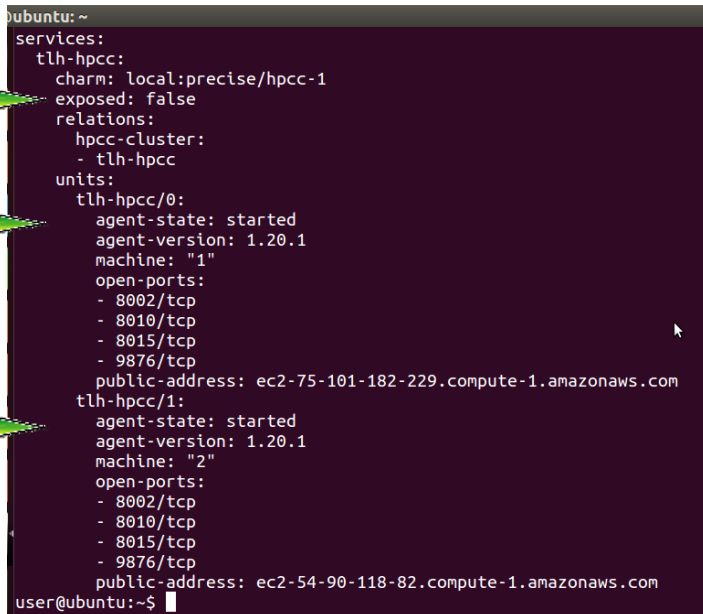| ⚠	If juju gives you an error when the above juju deploy command is issued, then you just remove any files in the directory ~/.juju/environments before trying again.

The following screenshot shows what the deploy command outputs to the Ubuntu terminal.

After the above deploy has displayed the command prompt, the HPCC system has not completed configuring and deploying. Use the juju status command to determine when the system is completely deployed.

```
juju status
```

The "services" section of the status output looks like the following screenshot. The 2nd and 3rd green arrows show where the state of the nodes are reported. In this case, the state is "started" which means the HPCC System is completely deployed.

```
ubuntu: ~
 services:
   tlh-hpcc:
     charm: local:precise/hpcc-1
     exposed: false
     relations:
       hpcc-cluster:
       - tlh-hpcc
     units:
       tlh-hpcc/0:
         agent-state: started
         agent-version: 1.20.1
         machine: "1"
         open-ports:
         - 8002/tcp
         - 8010/tcp
         - 8015/tcp
         - 9876/tcp
         public-address: ec2-75-101-182-229.compute-1.amazonaws.com
       tlh-hpcc/1:
         agent-state: started
         agent-version: 1.20.1
         machine: "2"
         open-ports:
         - 8002/tcp
         - 8010/tcp
         - 8015/tcp
         - 9876/tcp
         public-address: ec2-54-90-118-82.compute-1.amazonaws.com
user@ubuntu:~$
```
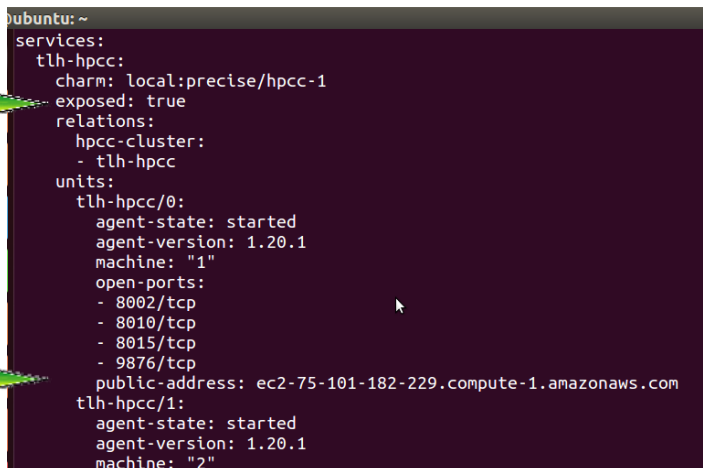
You will notice in the above screenshot that exposed is false (line pointed to by 1st green arrow). This means public IP addresses cannot be used by you to access say the ECL Watch or the deployed HPCC System.

The following juju command exposes public IP addresses.

```
juju expose tlh-hpcc
```

Notice in the following screenshot that exposed is now true (line pointed to by the top green arrow).

```
ubuntu: ~
 services:
   tlh-hpcc:
     charm: local:precise/hpcc-1
     exposed: true
     relations:
       hpcc-cluster:
       - tlh-hpcc
     units:
       tlh-hpcc/0:
         agent-state: started
         agent-version: 1.20.1
         machine: "1"
         open-ports:
         - 8002/tcp
         - 8010/tcp
         - 8015/tcp
         - 9876/tcp
         public-address: ec2-75-101-182-229.compute-1.amazonaws.com
       tlh-hpcc/1:
         agent-state: started
         agent-version: 1.20.1
         machine: "2"
```

USING DEPLOYED HPCC SYSTEM

To use ECL Watch you must get the public IP address of the deployed HPCC's ESP. You can get it by doing the following.

```
cd ~/juju-charms/precise/hpcc/bin
./get_url.sh
```

The following is a screenshot of the output of get_url.sh.

So, the ESP public IP address is 75.101.182.229.

ECL Watch

To display ECL Watch for this deployed HPCC System, you would enter into your browser address box http://75.101.182.229:8010, where 8010 is the port number for ECL Watch.
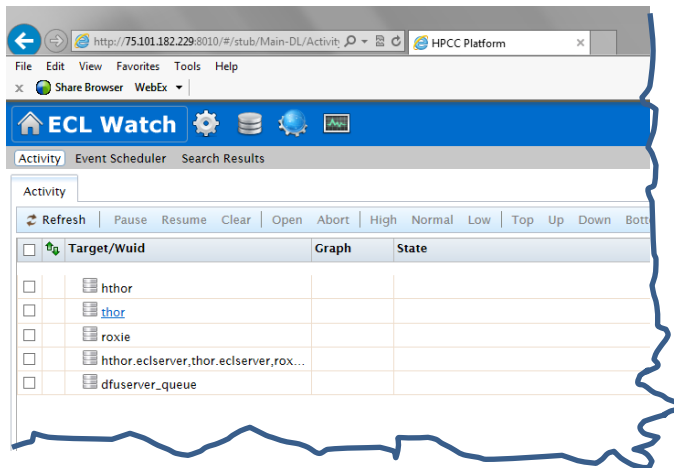
Setting up and Using ECL IDE

You must setup your ECL IDE to use your deployed HPCC System. In the Running ECL section of the following document: http://cdn.hpccsystems.com/pdf/RunningHPCCwithinAWS_EC2_Beta1.pdf, are

detail how to do this. But, if you have used ECL IDE before and have it setup for other clusters then all you need to do, under Preferences, is:

1. Create a New Configuration. I called my AWS_ 75_101_182_229, where the numbers are the IP address.
2. Place the IP address in the Server box then click OK.

My Preference popup looks like the following screenshot.
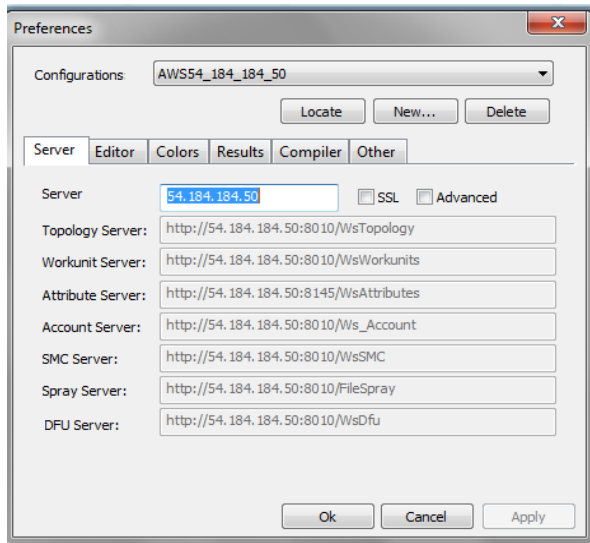
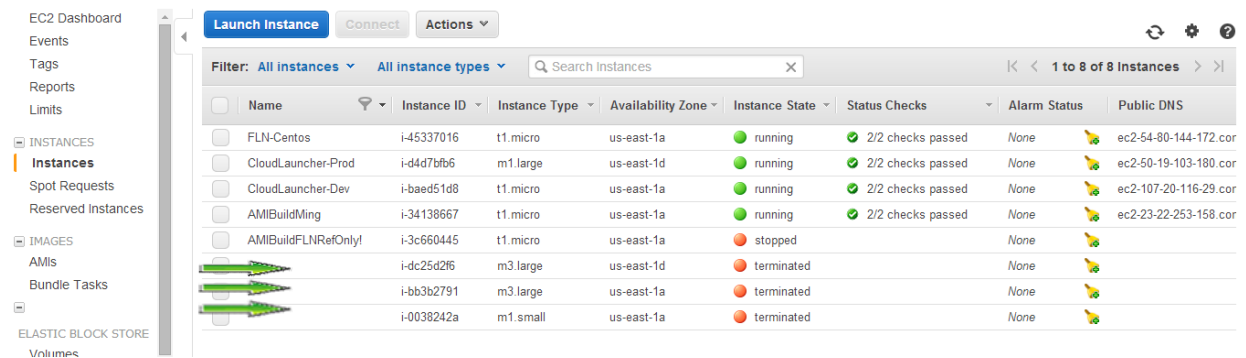To login use hpccdemo as the user id and the password. You are ready to go.

Final Words

When you are done with your deployed HPCC System shut it down so AWS charges don't continue to accumulate. You can do this with the following command.

```
juju destroy-environment amazon
```

The following screenshot shows what the terminal looks like. Notice that after you have entered the above command, you need to confirm that you want the HPCC System terminated by typing y, for yes.

After your HPCC System has terminated, the Instances page of console.aws.amazon.com will look like the following screenshot. The green arrows point to the instances that were terminated.

Appendix A. Setting Up Ubuntu 12.04 Linux VMware Machine

Download and Install VMware Player

First, download a VMware Player at https://my.vmware.com/web/vmware/free.

As you can see from the following screenshot of the site, there are many products that you can download. The one that you want to download is marked with a green arrow, below, i.e. VMware Player.
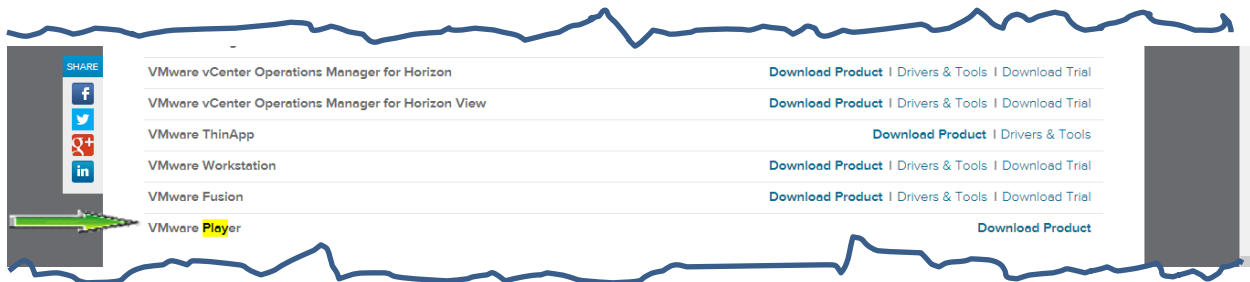
You will get the following web page after clicking on the VMware Player link shown above.
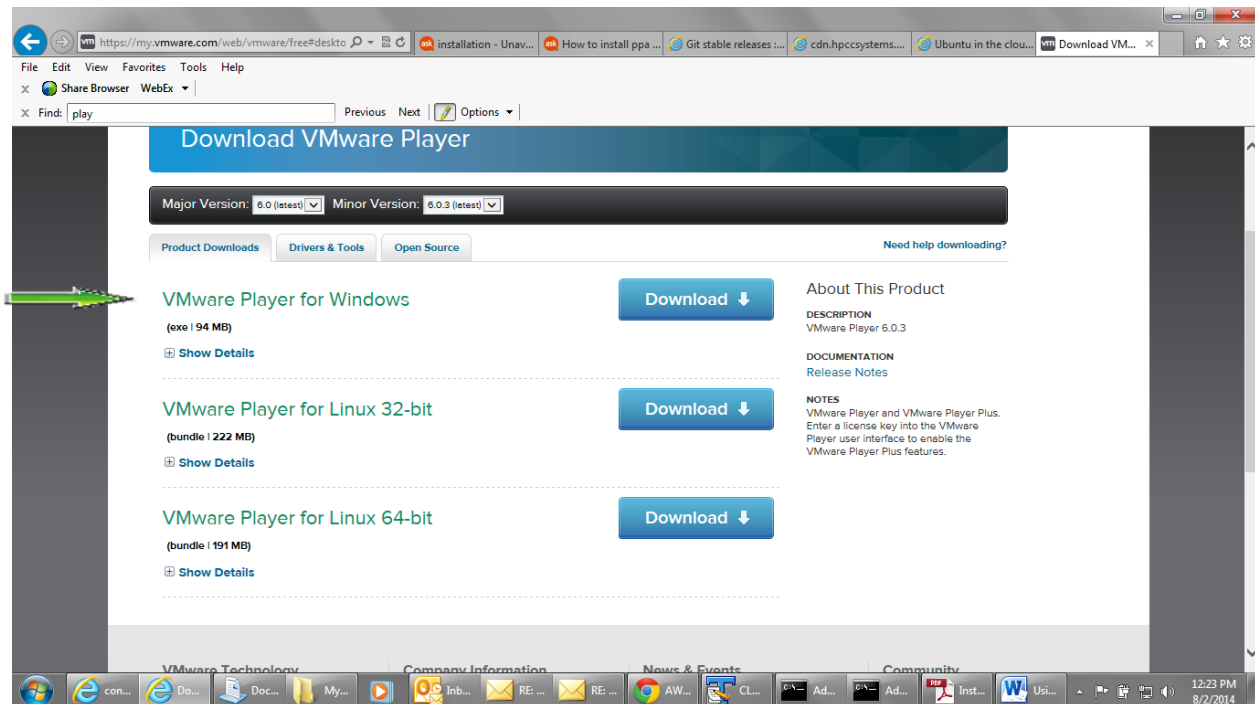
Click on the Windows VMware Player download button, marked with the green arrow, i.e. VMware Player for Windows. The download takes a few minutes.

Download and Install Ubuntu 12.04 VM Image with VMware Tools.

Secondly, download the VMware image for Ubuntu 12.04 Linux machine with VMware Tools at
http://www.traffictool.net/vmware/ubuntu1204t.html

When you load this site into your browser, it should look like the following:

To download Ubuntu 12.04 with the VMware Tools, click on the link pointed to by the green arrow, above. Make sure you save the downloaded zip file where you can find it. The download may take several minutes.

After the download is complete, extract the contents of the zip file and place it in a folder where you can find it. I've stored mine in Documents\MyVMImages.

Setup Ubuntu 12.04 and VMware Tools on the VMware Player

Open VMware Player and click on Open a Virtual Machine (marked with a green arrow below).
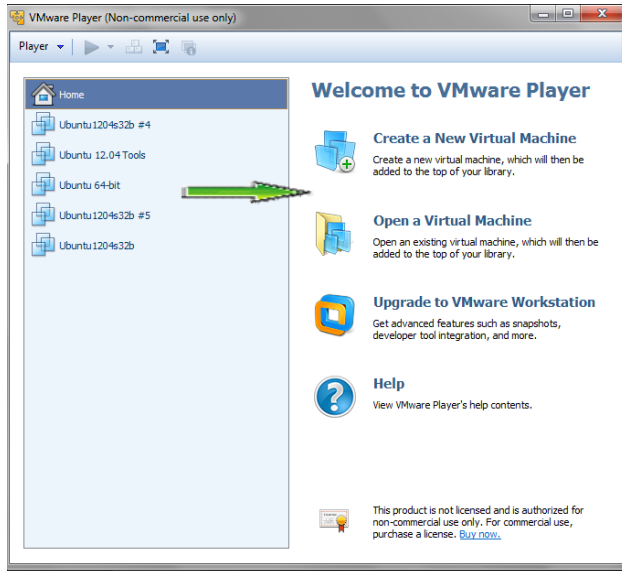

This will open the Windows Explorer so you can open the folder containing the Ubuntu 12.04 VM image. Once you have found it, you should see the file Ubuntu.vmx (see below screenshot).
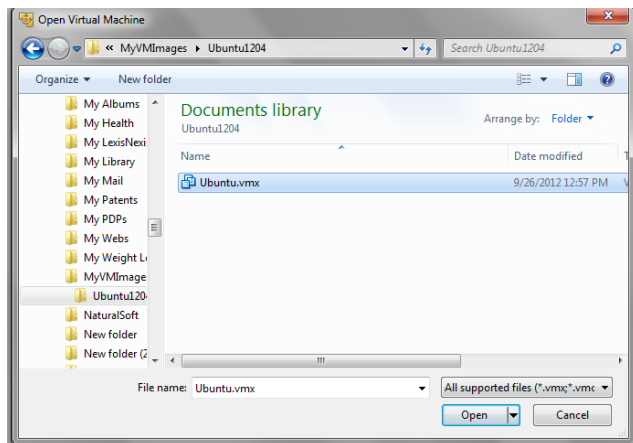

Click on it so it will be listed in the VMware Player's list of virtual machines (see the top green arrow in the following screenshot). As you can see, it has already been selected. So, you can click on Play virtual machine to start the virtual machine (see bottom green arrow in the following screenshot).

	When the machine starts, you may see a popup asking to check for updates. Click the X to ignore it.
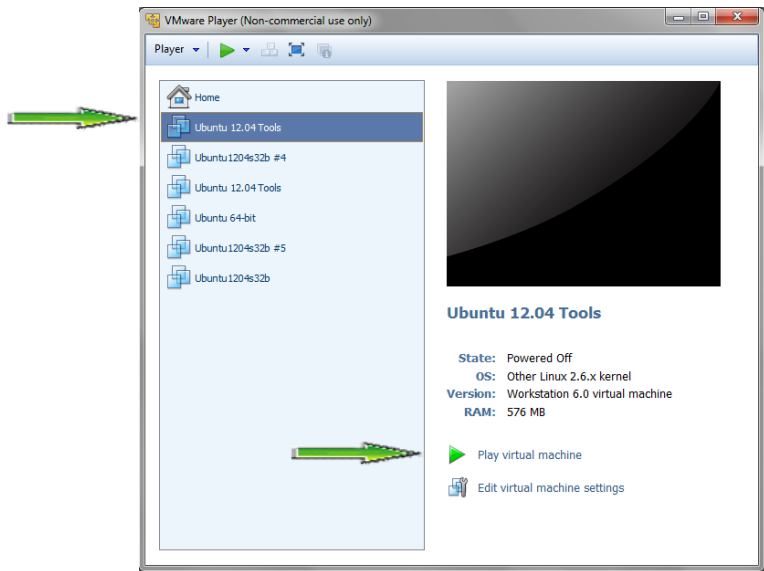
When the machine is ready for use, its screen should look like the following screenshot.
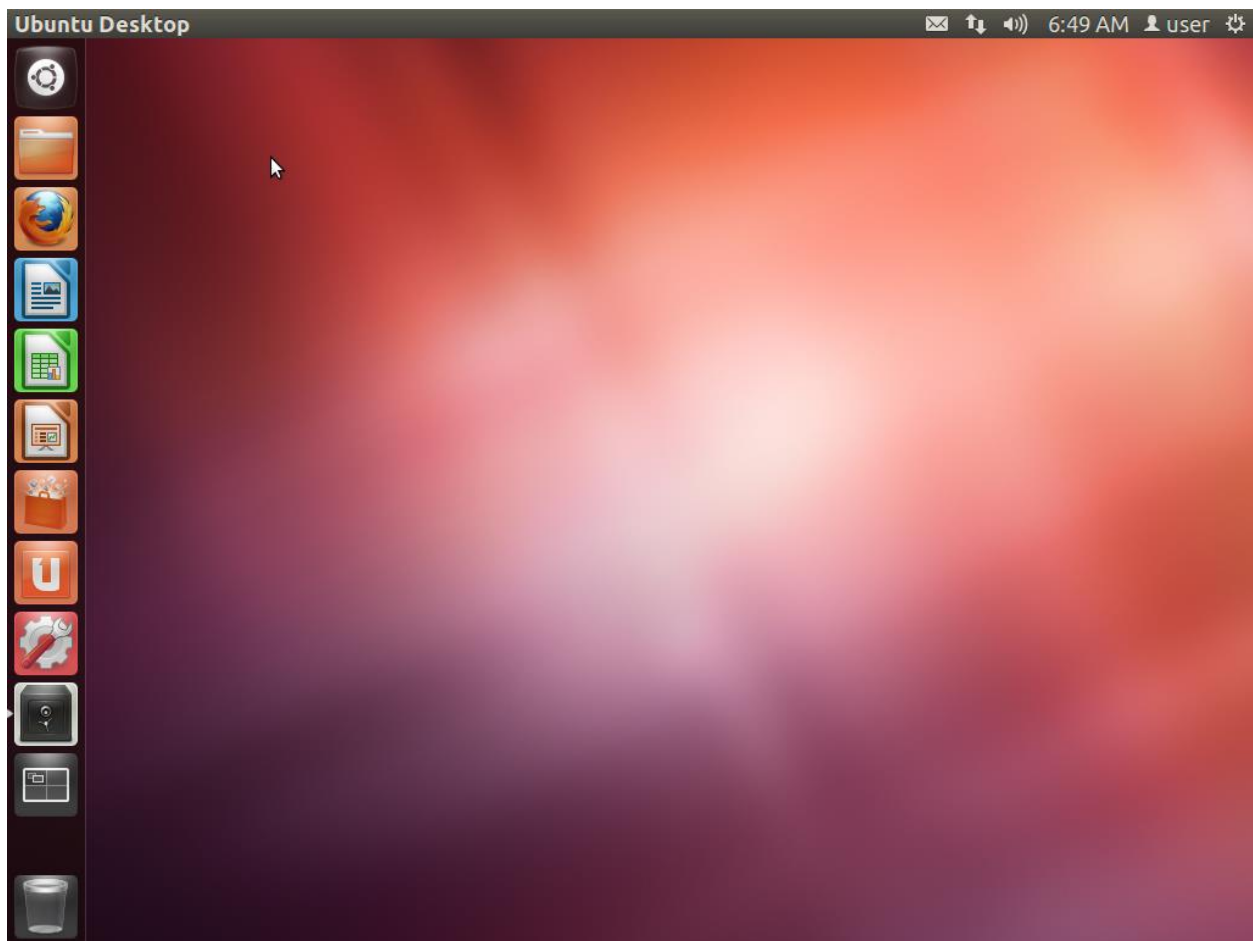
So, at this point, you can open an Ubuntu Terminal window with CRTL-ALT-T or by selecting in from the Dashboard (top icon on the left). The machine with an open Terminal Windows looks like the following.

user@ubuntu: ~

user@ubuntu:~$