################ PART I. Pre-requisite

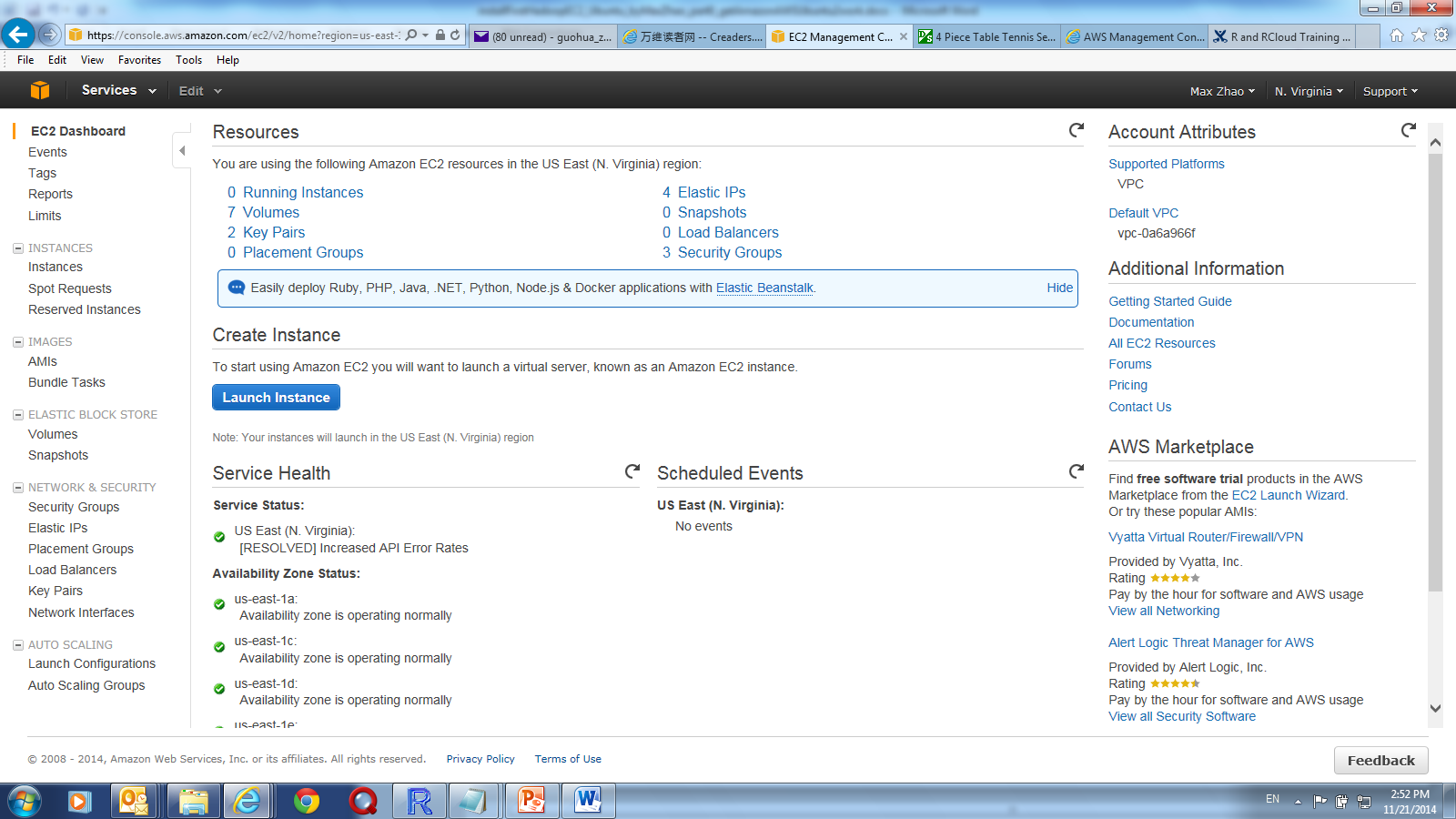
####################### Step 0 Need the following software installed:

1. PuTTY: Source: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
2. PuttyGEN: Source: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
3. WinSCP Source: http://winscp.net/eng/download.php
4. SuperPuTTY Source:http://code.google.com/p/superputty/downloads/list

####################### PART I. Do the following 3 times to start 3 Linux Ubuntu instances

####################### Step 1. Generate the Amazon EC2 Ubuntu instances. Most of the steps are straightforward. Step 9 and Step 13 are 2 steps that needs to be carefully setup.

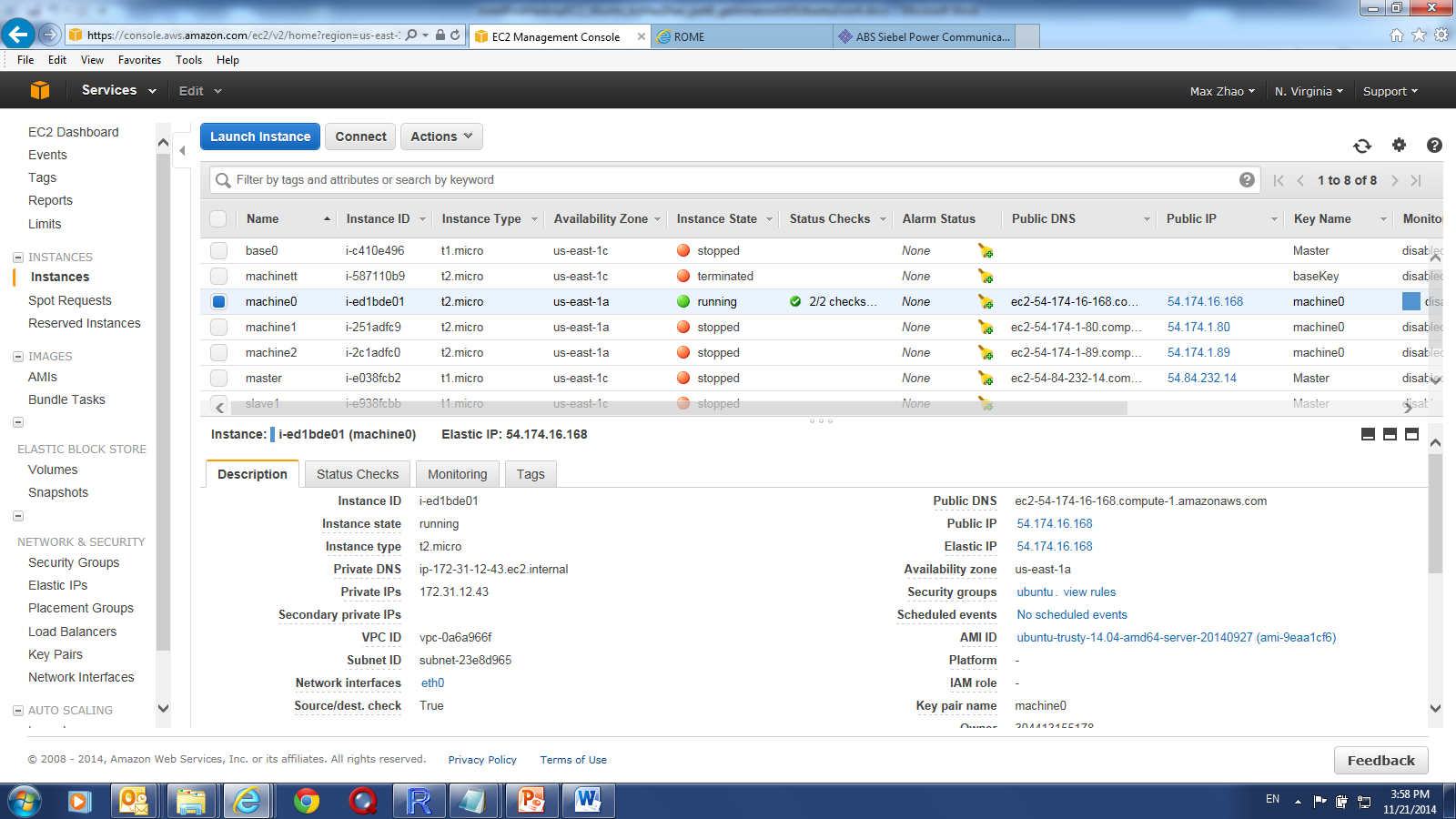
1. Go to the website: <https://console.aws.amazon.com/console/>
2. Pick EC2 and you should see a screen like mine.



1. Click “Launch Instance” to next screen labelled “Step 1: Choose an Amazon Machine Image (AMI)”
2. Pick the 4th one “Ubuntu” to the next screen “ Step 2: Choose an Instance Type”
3. Pick the first image “General Use”. And Click bottom right tab, ”Next Step: Configure Instance” to the screen “Step 3 Configure Instance Details”
4. Do nothing here and click “Next Step: Add Storage”
5. Pick the maximum hard drive of “30G” and click “Next”
6. Add the machine name here: “machine0” for first and “machine1” and “machine2” for the other 2 machines. The name needs to be just like ‘machine0’, ‘machine1’ and ‘machine2’ since they will be used and referred to in the setup files. Click next
7. On the screen “Step 6: Configure Security Group”. You will need to create a security group to use. Basically this is to define how the machine may be accessed by other machines. So Outgoing should be “All”. Incoming needs to be defined. You can click “Add Rule” to add the definitions. One example of incoming definition is like below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Protocol** | **Port Range** | **Source** |
| Custom TCP Rule | TCP | 0 | Your IP Home Address 1 |
| Custom TCP Rule | TCP | 0 | Your IP Home Address 2 |
| Custom TCP Rule | TCP | 8000 - 8900 | 0.0.0.0 |
| Custom TCP Rule | TCP | 9000 | 0.0.0.0 |
| Custom TCP Rule | TCP | 9001 | 0.0.0.0 |
| Custom TCP Rule | TCP | 50000-50100 | 0.0.0.0 |
| HTTP | TCP | 80 | 0.0.0.0 |
| HTTPS | TCP | 443 | 0.0.0.0 |
| SSH | TCP | 22 | 0.0.0.0 |

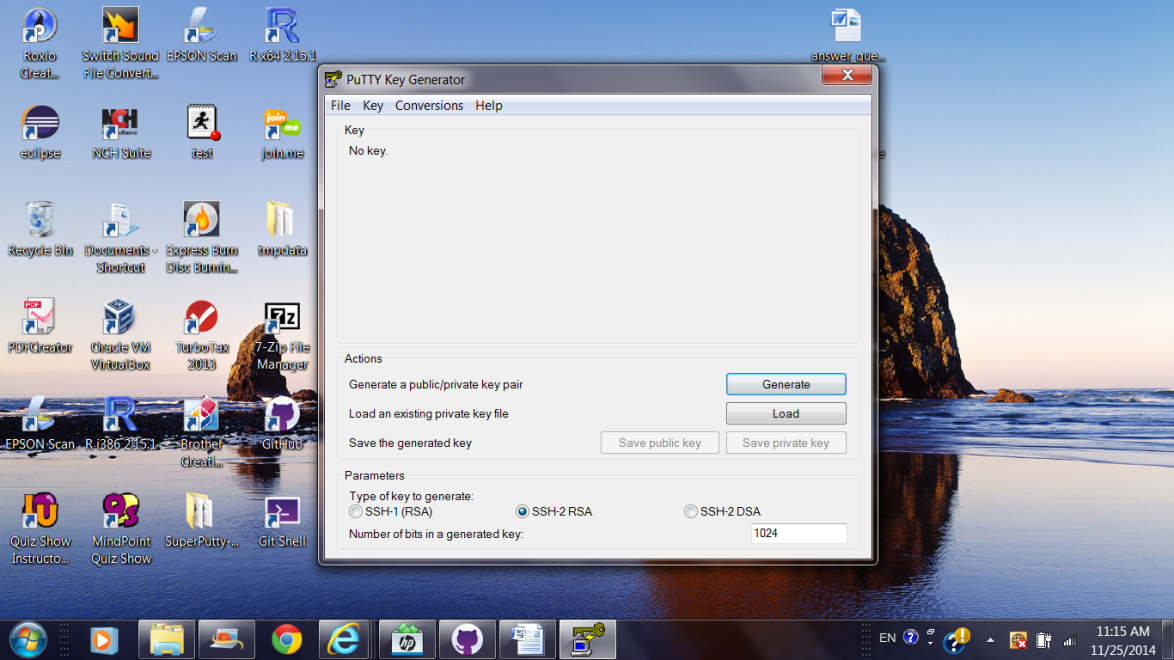
1. After creating a security group, you can attach the Security Group to the instance.
2. Click “Review and Launch”
3. Click “Launch”
4. At this time, you will need a Security Key Pair. Pick “Create a new Key pair”, give the name machine0, and download and save the file at your local drive. The file name will be **machine0.pem**.
5. Click the “View Your Instance” on the bottom right of the screen. You should see the machine name ‘machine0’ with status as ‘running’ after a while. The screen shot is like the below:



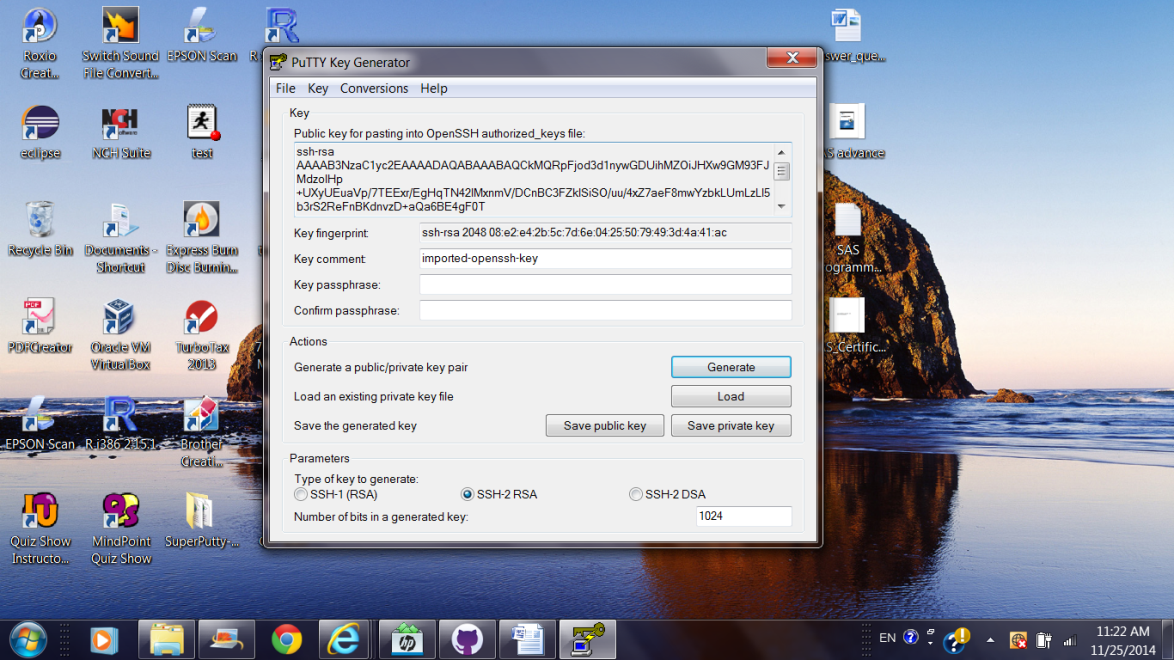
1. If you highlight the machine0 you just created, you will see below lots of info about this instance of Linux machine, including IP address, public DNS etc.
2. Now do the same process to generate the other 2 machines, using the Security Group and the same key pair you have generated when generating the first Ubuntu instance. Write down the IP Addresses of the 3 machines.

####################### Step 2. Generate the private key from the key pair from amazon AWS.

1. Open up the PuTTYGEN, Click the “conversion” and pick “import key”

 and click “

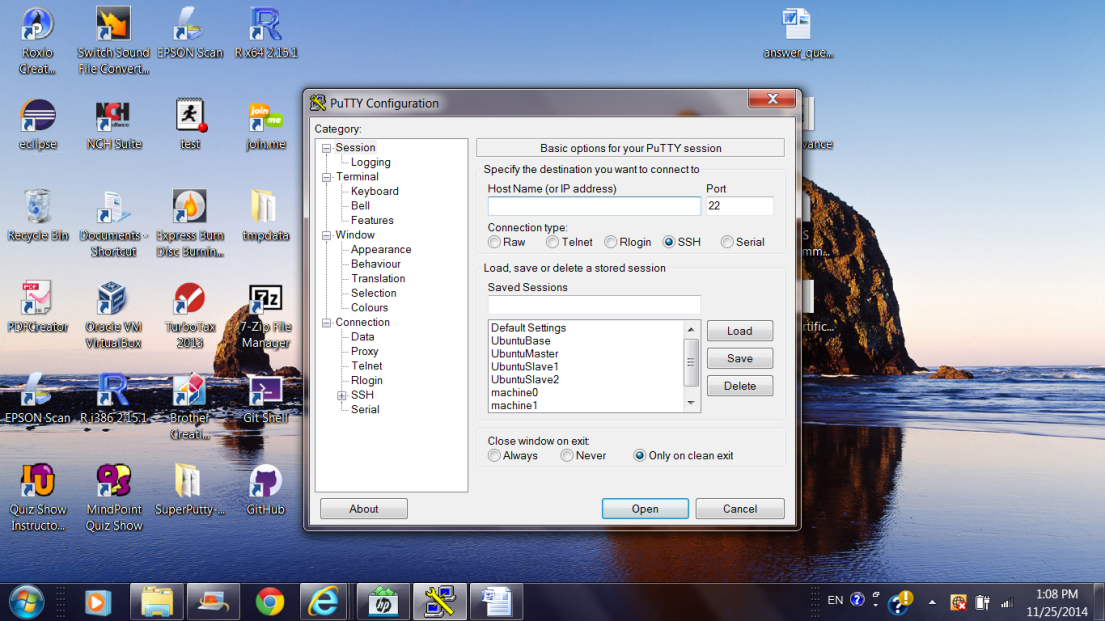
1. Browse to the machine0.pem you have downloaded and choose the file. It should give you the following screen to the left.



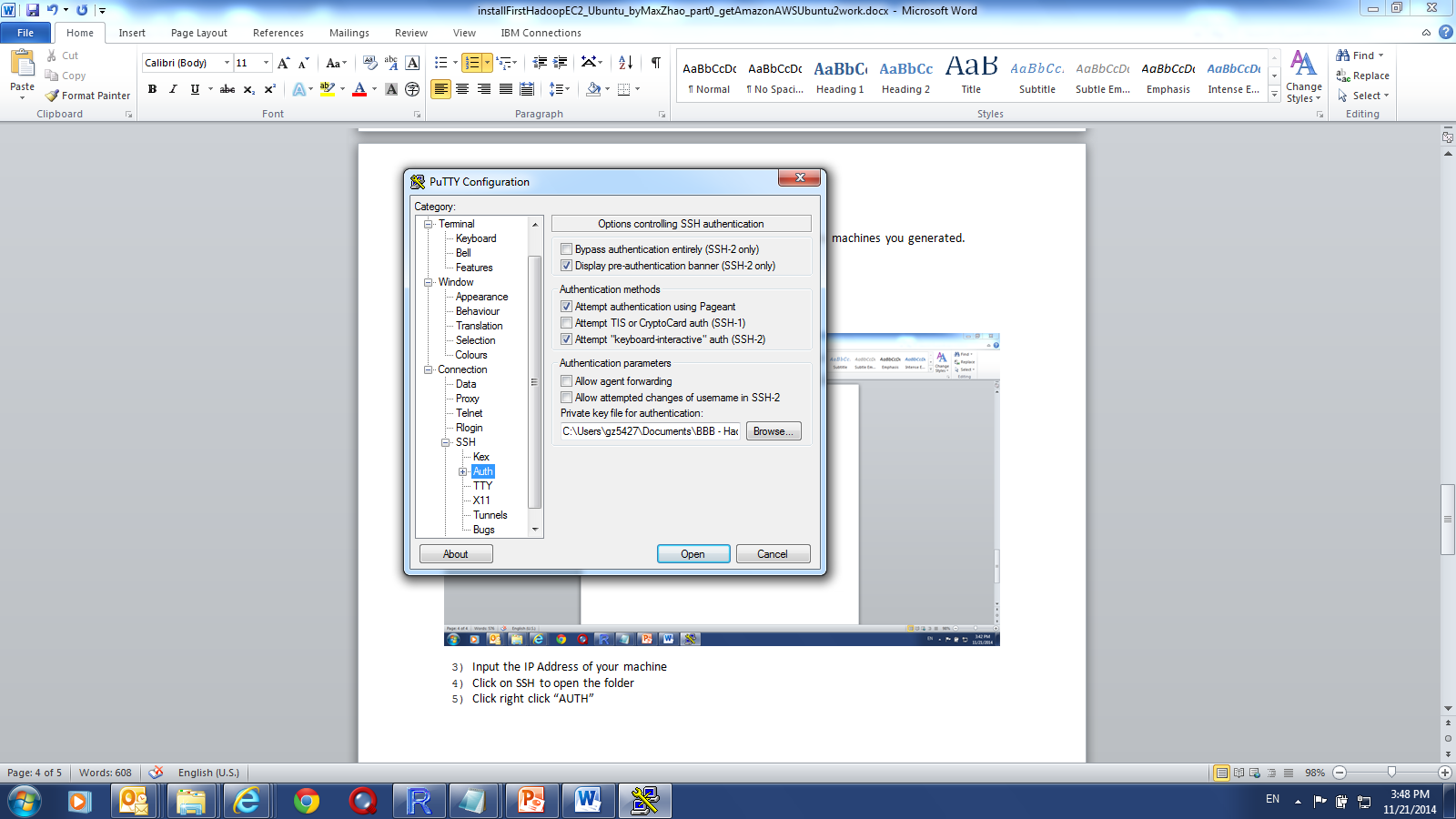
1. You can choose to input to key passphraise or ignore it.
2. Click “Save Public Key” to save it to the local file.
3. Click “Save Private Key” to save it to the local file such as: machine0.ppk. ---- Important: the private key file is the one you use when you try to access the Ubuntu instance using PuTTY.

####################### Step 3 How to access the Amazon Ubuntu machines you generated.

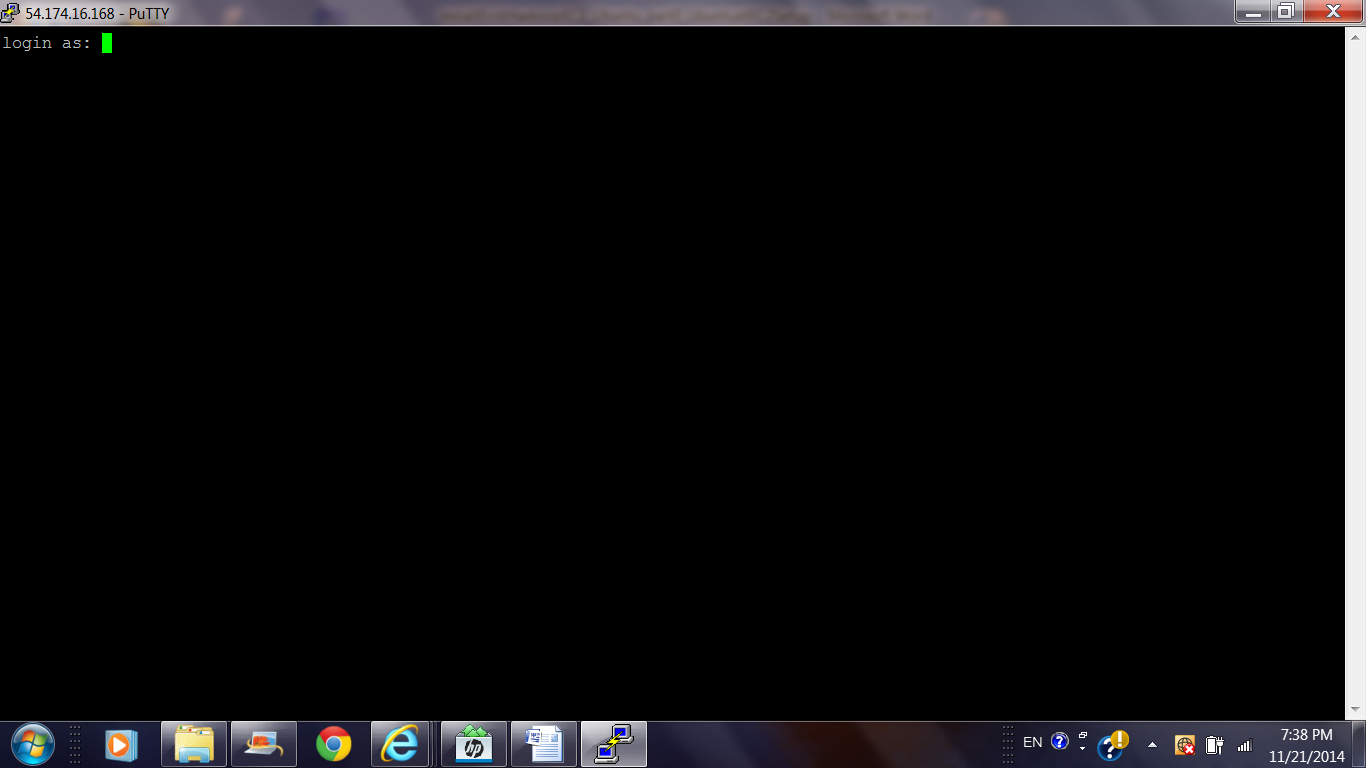
1. Open the PuTTY, you should see screen like this:



1. Input the IP Address of your machine
2. Click on SSH to open the folder
3. Click right click “AUTH”, you should see something like this:



1. Browse to the private key (machine0.ppk) you have saved, and then go back to top and click “session”. You can save the session labeled it as machine0
2. Click “Open” and A window should open up like the following:



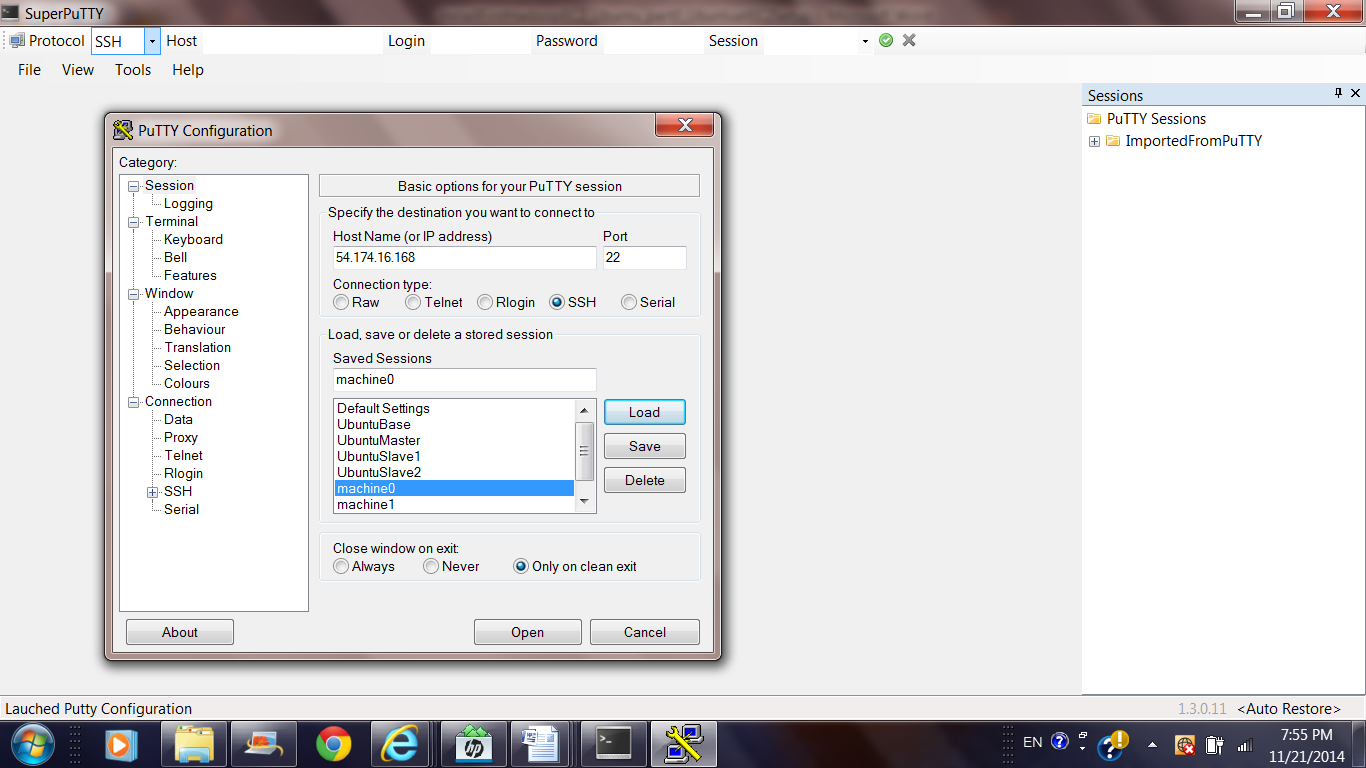
1. Input “ubuntu” as the user at the prompt and you will be logged on!!!.
2. Now you should be able to get into the Ubuntu machine you just created. You are the admin and the superuser!!!!
3. You can run mkdir, ls, or the first thing you can do to update the system such as:

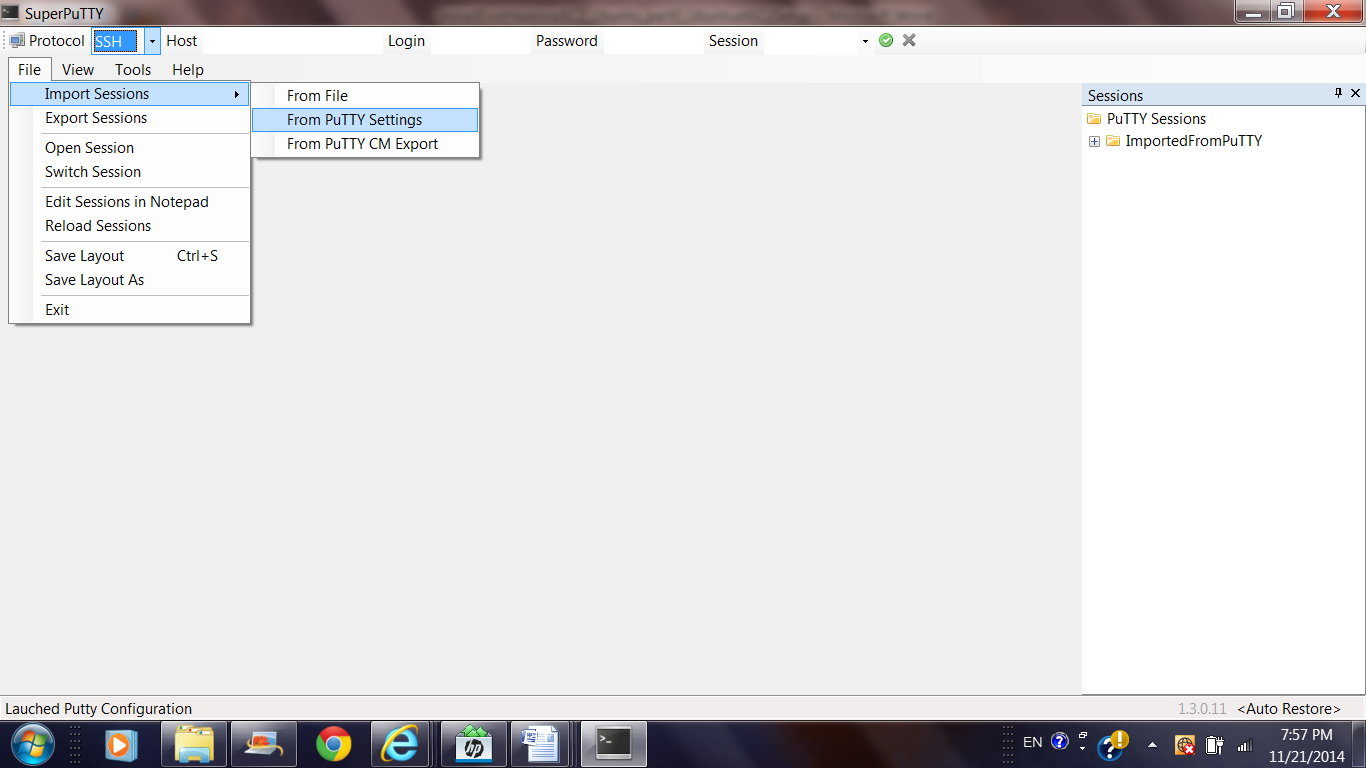
sudo apt-get update ## Run the update needed.

sudo dpkg-reconfigure tzdata ## This is to setup your local time.

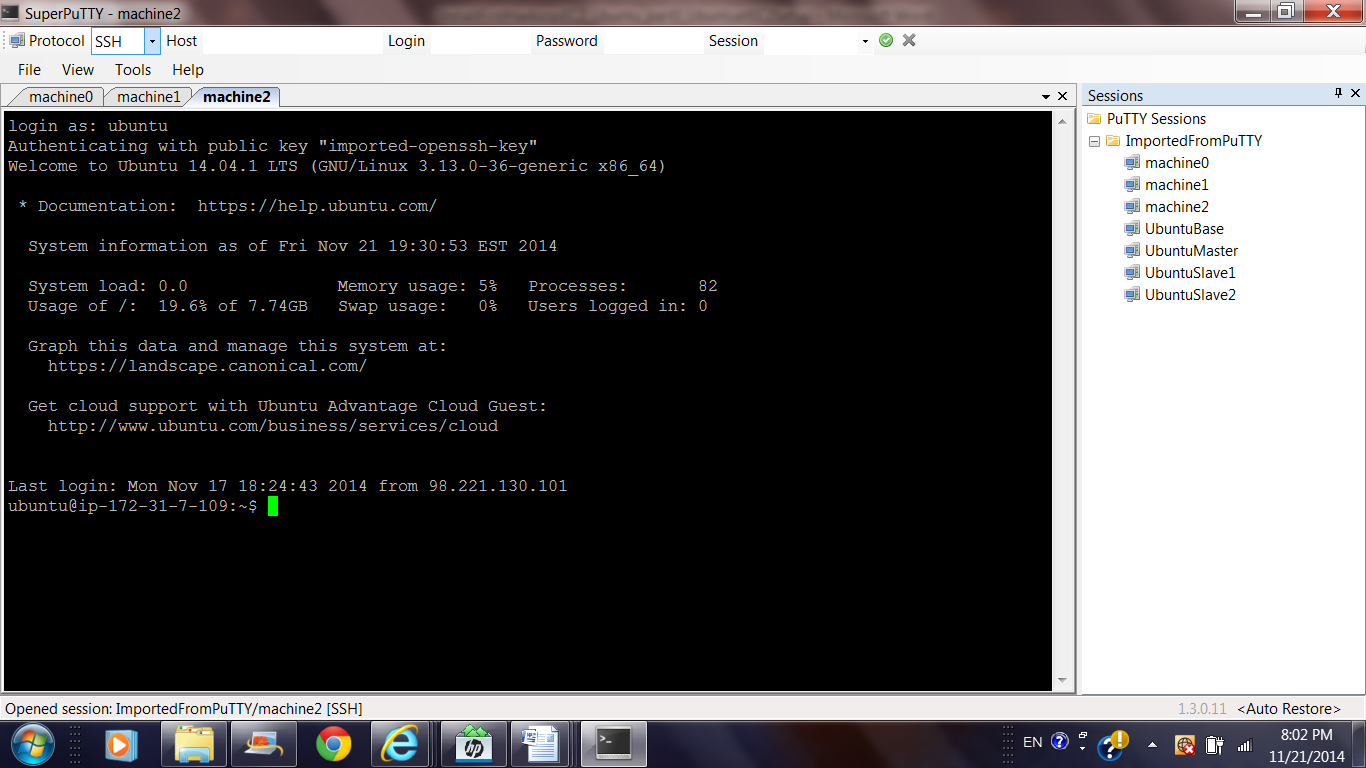
wget <http://apache.mirrors.lucidnetworks.net/hadoop/common/hadoop-2.4.0//hadoop-2.4.0.tar.gz> ## Download something from the web.

####################### Step 4 Using of SuperPuTTY

1. Open the SuperPutty
2. Pick Tool PuTTY Configuration. This is the same PuTTY window.
3. Input the IP and point AUTH to the private key location
4. Go to the session, and input the machine name in the “Saved Session” and click save 
5. Do this for the other 2 machines and save the logon info as well and save the sessions.
6. Close the PuTTY Windows and open it up again.
7. Go to File -> Import Sessions -> From PuTTY Sessions. You will see windows like this



1. Click to expand “ImportedFromPuTTY” to see all the saved sessions. And Click each of the machines: machine0, machine1 and machine2.
2. After logon to all three machines, you can see the following screen:



1. Now you see the 3 tabs, one for each machines. And you can easily switch between 3 machines.