

# Setting Up a Fast HPCC on AWS

---

Timothy Humphrey

3/21/2015

## Table of Contents

INTRODUCTION .....	3
What is an HPCC System? .....	3
About This Document .....	3
Task 0. Setup/Install Code Needed on Your Ubuntu Linux Machine .....	4
Cloning BestHPCCoAWS .....	4
Installing aws cli .....	6
Task 1. Launch All Instances of Your HPCC System from the AWS Console .....	7
Launch Instance .....	7
Step 1. Choose an Amazon Machine Image (AMI).....	8
Step 2. Select the Instance Type You Want .....	8
Step 3. The Configure Instance Details screenshot .....	9
Step 4. Add Storage.....	10
Step 6. Configure Security Group.....	12
Step 7. Review Instance Launch.....	13
Task 2. Fill in Configuration File, cfg_BestHPCC.sh .....	15
Task 3. Get Instance IDs of All Launched Instances .....	17
Task 4. Run getPublicAndPrivateIPsOfAllInstances.pl to Get Public & Private IPs .....	18
Task 5. Run cpServerFilesToAllInstances.pl to Copy Files to Instances .....	19
Task 6. Run setupDisksOnAllInstances.pl to Raid the Disks Etc. on All Instances.....	19
Task 7. Run installHPCCOnAllInstancesAndStart.pl to Install the HPCC on All Instances .....	20
Task 8. Make Sure the System is Running By Bringing Up ECL Watch.....	20
Task 9. Run configureHPCC_multislaves_per_instance.pl.....	20
Task 10. Make Sure the System is Running By Bringing Up ECL Watch.....	21
Task 11. Mount S3 Bucket on Dropzone (if needed) .....	22
Task 12. Run updateSystemFilesOnAllInstances.pl.....	22
Task 13. Setup ECL IDE to Work with the Deployed HPCC System on AWS .....	23
Appendix A. Setting Up Ubuntu 12.04 Linux VMware Machine .....	28
Download and Install VMware Player.....	28
Download and Install Ubuntu 12.04 VM Image with VMware Tools. ....	29
Setup Ubuntu 12.04 and VMware Tools on the VMware Player.....	31
Appendix B. Setting up a Security Group For Your HPCC System on AWS .....	34

Security Group .....	34
Appendix C. Setting up a Virtual Private Cloud (VPC) .....	38

## INTRODUCTION

This document describes in detail how to configure and deploy a fast executing HPCC System on Amazon's Cloud Service (AWS). The software for doing this as well as this document is in the following github repository: <https://github.com/tlhumphrey2/BestHPCCoAWS>.

### What is an HPCC System?

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.


### About This Document

In what follows, commands given in an Ubuntu terminal window are shown with a gray background, like the following example:

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

In addition 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.
-------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------


	As you use this document to configure and deploy an HPCC System to AWS, if you get errors that I haven't talked about here, please take a screenshot showing the error and email it to me: <a href="mailto:timothy.humphrey@lexisnexis.com">timothy.humphrey@lexisnexis.com</a>
-------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Table 1, below, summaries all the tasks that you do to configure and deploy an HPCC System that performs well on AWS. There are 13 tasks.

All except tasks 1, 8, 10 and 13 are done in an ubuntu linux machine. If you don't have an ubuntu linux machine, Appendix A. Setting Up Ubuntu 12.04 Linux VMware Machine gives detailed instructions for downloading and installing both a VMWare Player and an Ubuntu 12.04 VM Image with VMware tools.

Task 1 is the only task that uses the aws console. And, you do Task 13 in your ECL IDE. So, be sure you download the it at <http://hpccsystems.com/download/free-community-edition/ecl-ide>.

All tasks that require running a program must be executed while in the BestHPCCoAWS directory of your ubuntu machine.

**Table 1. All Tasks to Configure & Deploy an HPCC System on AWS**

Task	Task Description
0	Setup/Install code needed on your Ubuntu Linux machine.
1	Launch all instances of your HPCC System from the AWS console.
2	Setup the configuration file, <code>cfg_BestHPCC.sh</code> (in the <code>instance_files</code> directory).
3	Get instance IDs of all launched instances from Instances page of aws console
4	Run <code>getPublicAndPrivateIPsOfAllInstances.pl</code> to get private and public IPs into the files, <code>private_ips.txt</code> and <code>public_ips.txt</code> , respectively.
5	Run <code>cpServerFilesToAllInstances.pl</code> to copy software to instances that will be ran there.
6	Run <code>setupDisksOnAllInstances.pl</code> to setup disk for good performance (raid, mount unmounted disks, etc.)
7	Run <code>installHPCCOnAllInstancesAndStart.pl</code> to install HPCC and start it (this will be the 1st minimal system with 1 slave node per instance)
8	Go into ECL Watch just to make sure the system is up and running.
9	Run <code>configureHPCC_multislaves_per_instance.pl</code> to reconfigure HPCC System so there are multiple slave nodes per instance.
10	Go into ECL Watch just to make sure the system is up and running. Try running on thor the code in playground.
11	If you need data in an S3 bucket, ssh into master instance and run <code>mountS3Bucket.sh</code> to mount your S3 bucket on the dropzone.
12	Run <code>updateSystemFilesOnAllInstances.pl</code> to update system files that enable the Linux system to handle the traffic from many slave nodes per instance.
13	Setup ECL IDE to work with the deployed HPCC System on AWS.

## Task 0. Setup/Install Code Needed on Your Ubuntu Linux Machine

First, all the commands I give in this section should be run in your ubuntu linux home directory (~). When you first login, this should be the directory you are in.

The only software you need is BestHPCCoAWS which you clone from its github repository and the aws cli (command line interface).

### Cloning BestHPCCoAWS

You will need git to clone BestHPCCoAWS. Use the following command to install git on your ubuntu linux machine.

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



If you get errors when you attempt to install git, i.e. the above command, then try doing an update followed by an upgrade then try the install of git again.

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install git
```

If “install git” still doesn’t work then try the following.

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

If you get lots of errors during all these attempts your ubuntu vmware is probably not connected to the internet.

Next install the BestHPCCoAWS scripts cloning them from github.com. From your home directory, enter the following command.

```
git clone https://github.com/tlhumphrey2/BestHPCCoAWS.git
```

The result of the above should be a directory structure that looks like the following:

```
BestHPCCoAWS
├── change_credentials.pl
├── configureHPCC_multislaves_per_instance.pl
├── cpServerFilesToAllInstances.pl
├── getConfigurationFile.pl
├── getInstanceIDs.pl
├── getPublicAndPrivateIPsOfAllInstances.pl
├── installHPCCOnAllInstancesAndStart.pl
├── instance_files
│   ├── cfg_BestHPCC.sh
│   ├── change_credentials.sh
│   ├── configureHPCC.sh
│   ├── install_hpcc.sh
│   ├── mountS3Bucket.sh
│   ├── push2hpcc.sh
│   ├── setup_zz_zNxlarge_disks.pl
│   └── updateSystemFilesForHPCC.pl
├── instance_ids.txt
├── mountS3BucketOntoDropZone.pl
├── push2AllHPCCInstances.pl
├── README.md
├── SetupBestHPCCoAWS.pdf
├── setupDisksOnAllInstances.pl
├── startHPCCOnAllInstances.pl
├── stopHPCCOnAllInstances.pl
└── updateSystemFilesOnAllInstances.pl
```



The command, `tree BestHPCCoAWS`, 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
```

Next, make sure that all programs in the `BestHPCCoAWS` and `BestHPCCoAWS/instance_files` directories are executable, by doing the following two commands.

```
chmod 777 ~/BestHPCCoAWS/*.pl
chmod 777 ~/BestHPCCoAWS/instance_files/*
```

### Installing aws cli

The `aws cli` (command line interface) is used to get: 1) a list of instance IDs for instances that has been launched in the region you are launching your HPCC, and 2) a list of public and private IPs for the instance IDs in the `instance_ids.txt` file. Here is how you install and configure it.

First, install `python-pip` by doing the following command. `Python-pip` is used to install Python packages.

```
sudo apt-get install python-pip
```

Then, install the Python package `awscli` with the following command.

```
sudo pip install awscli
```

To configure the `aws cli`, do the following command which will prompt you for your `aws` access and secret keys like the following screenshot.

```
aws configure
```

```
al
File Edit View Search Terminal Help
tlhumphrey2@ubuntu1204s32b:~$ aws configure
AWS Access Key ID [*****LCYQ]:
```

You respond by entering your `AWS Access Key` and hit enter. Then you are prompted for your `AWS Secret Key`. And, you respond by entering your secret key and hitting enter. After these you will be prompted for your default region and default output format. For both of these, just hit enter.

## Task 1. Launch All Instances of Your HPCC System from the AWS Console



Before starting Task 1, make sure you have setup a Virtual Private Cloud (VPC). There are two places in Task 1 that you need the same VPC – Step 3 and Step 6. In Figure 4 of Step 3 you need to select a VPC for Network. And, for Step 6 you will need to select or create a security group which has the same VPC as the Network selected in Figure 4 of Step 3. Detailed instructions for setting up a VPC are in Appendix C.

Also, before you get started with this launching procedure, if you don't currently have a security group for your HPCC System, Appendix B has detailed instructions for setting up one.

If your THOR master's instance type is different than your THOR slaves' instance types (as suggested above), the following procedure is done twice to launch all instances of your HPCC System. This procedure is done on the AWS console.



For a high performing HPCC THOR cluster on AWS, we suggest you launch 7 instances – one for the THOR master and the other 6 for the THOR slaves.

For all instances that will have THOR slave nodes, we suggest you use instance type i2.8xlarge. But, you won't need such a high performance instance type as i2.8xlarge for your THOR master instance. And, since the i2.8xlarge instance type currently cost \$6.82 per hour, we suggest you use something less expensive for the THOR master instance, say the c3.4xlarge instance type which currently cost \$0.84 per hour.



Because the i2.8xlarge instance cost so much, we suggest that while you are learning how to use this procedure, you use instance type r3.8xlarge instead and only launch 2 of these for THOR slave nodes.

### Launch Instance

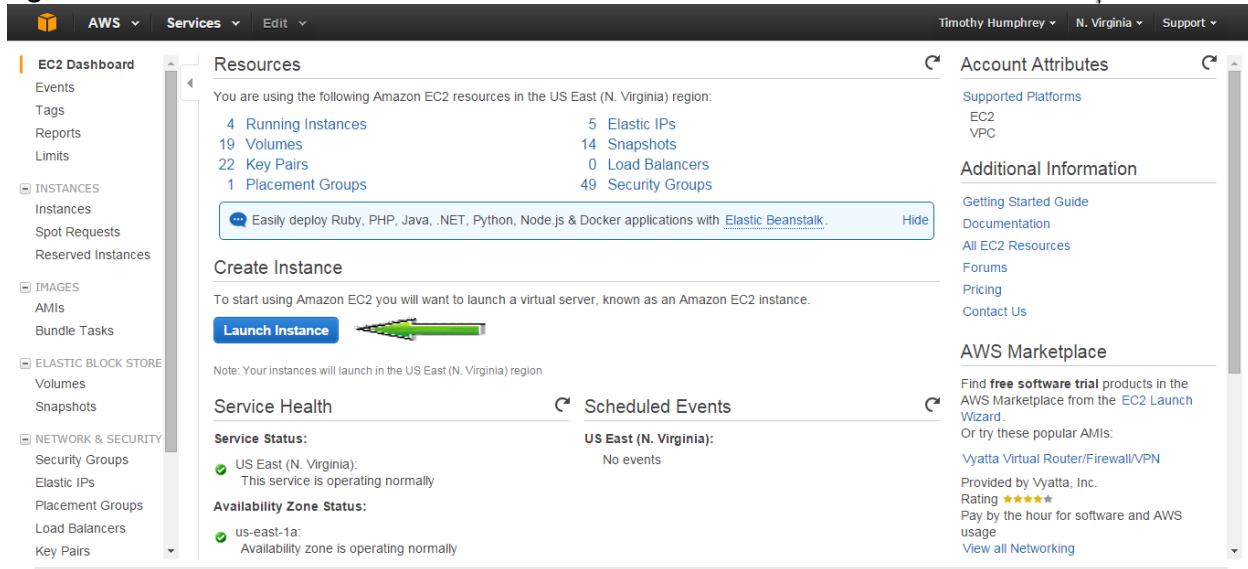
As mentioned above, this launching procedure is done on the aws console. So, in your browser, go to <http://console.aws.amazon.com>, login and go to the EC2 Dashboard, which is shown in Figure 1, below.



A good region to launch your instances is us-west-2 because it has newer hardware than some of the other regions. You can change the region by clicking on the current region (see the downward pointing green arrow Figure 1, below) and selecting a different region from the dropdown menu (us-west-2 is Oregon).

From the EC2 Dashboard web page, shown in the following screenshot, click on "Launch Instance" (pointed to by green arrow in the middle of the screenshot).



**Figure 1. EC2 Dashboard Screenshot 1**

This click will bring up the next web page which lets you pick an AMI for the instance(s) you are about to launch. Figure 2 is a screenshot of that page.

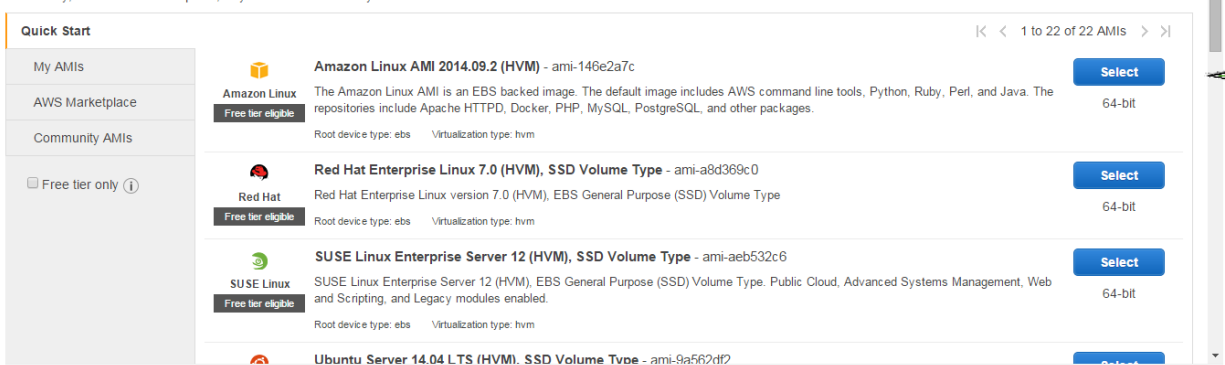
### Step 1. Choose an Amazon Machine Image (AMI)

Select the first AMI on this page, the Amazon Linux AMI, by clicking the “Select” button (pointed to by the green arrow in the right margin). This AMI has Centos Linux. You will choose this AMI for all instances launched.

**Figure 2. Choose An Amazon Machine Image (AMI) Screenshot**

#### Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace, or you can select one of your own AMIs.



### Step 2. Select the Instance Type You Want

When you select the AMI you want, the next web page you see will be the “Choose an Instance Type” web page. Figure 3 is a screenshot of that page. Scroll down this page until you see the instance type you want. Then, click the check-box on the left to select it (shown in Figure 3 by the green arrow in the left margin).

This screenshot indicates that I am choosing the c3.xlarge instance type which will be the instance type for my THOR master node.

**Figure 3. Choose an Instance Type Screenshot**

Step 2: Choose an Instance Type

<input type="checkbox"/>	Compute optimized	c4.xlarge	16	30	EBS only	Yes	High
<input type="checkbox"/>	Compute optimized	c4.xlarge	36	60	EBS only	Yes	10 Gigabit
<input type="checkbox"/>	Compute optimized	c3.large	2	3.75	2 x 16 (SSD)	-	Moderate
<input type="checkbox"/>	Compute optimized	c3.xlarge	4	7.5	2 x 40 (SSD)	Yes	Moderate
<input type="checkbox"/>	Compute optimized	c3.2xlarge	8	15	2 x 80 (SSD)	Yes	High
<input type="checkbox"/>	Compute optimized	c3.4xlarge	16	30	2 x 160 (SSD)	Yes	High
<input type="checkbox"/>	Compute optimized	c3.8xlarge	32	60	2 x 320 (SSD)	-	10 Gigabit
<input type="checkbox"/>	GPU instances	g2.2xlarge	8	15	1 x 60 (SSD)	Yes	High
<input type="checkbox"/>	Memory optimized	r3.large	2	15	1 x 32 (SSD)	-	Moderate
<input type="checkbox"/>	Memory optimized	r3.xlarge	4	30.5	1 x 80 (SSD)	Yes	Moderate

Cancel Previous **Review and Launch** Next: Configure Instance Details

After selecting your instance type, click on “Next: Configure Instance Details” (pointed to by the right margin green arrow in Figure 3). The next web you see is the “Configure Instance Details” page, Figure 4.

### Step 3. The Configure Instance Details screenshot

Below is a screenshot for the “Configure Instance Details” web page, Figure 4.

**Figure 4. Configure Instance Details Screenshot**

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances

Purchasing option ☐ Request Spot Instances

Network  [Create new VPC](#)

Subnet  [Create new subnet](#)  
251 IP Addresses available

Auto-assign Public IP

Placement group

IAM role

Shutdown behavior

Enable termination protection ☐ Protect against accidental termination


Cancel Previous **Review and Launch** Next: Add Storage

For your master instance, the “Number of instances”, in the above screenshot, will be 1, which is already set. There are three other places that may need changing (where the 3 left margin green arrows are pointing in Figure 4, above):

1. For “Network”, select a vpc type network. At the beginning of this section, Task1, we mentioned that you would need a vpc at this point. So, you should have one in the Network dropdown menu. If you don’t then the detailed instructions of Appendix C to make one.

- For “Auto-assign Public IP”, it should be set to “Use subnet setting (Enable)”. If it isn’t then select it from the dropdown menu.
- For “Placement group”, select from the dropdown menu a placement group that you have named. Do not select “no placement group”.

In Figure 4, above, I already had a placement group named “best-hpcc-pg”. If you don’t have one, select “New placement group” from the dropdown menu. Then give it a name by typing the name into the box called “New placement group name”.



The VPC selected above, for “Network”, should be the same as the one selected for VPC of Figure 17 when you are creating a security group (in Appendix B).









Once you have made these changes, click “Next: Add Storage” (where the right margin green arrow is pointing in Figure 4, above).

#### Step 4. Add Storage

The Add Storage page looks different depending on the instance type you choose. For some instance types, the SSD storage that comes with that instance is already setup. For example, the following screenshot, Figure 5, is for the i28xlarge instance type. It shows Instance SSD Store 0 through 7 are already setup.

**Figure 5. Add Storage Screenshot 1**

Step 4: Add Storage

Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Delete on Termination ⓘ	Encrypted ⓘ
Root	/dev/xvda	snap-f518b274	8	General Purpose (SSD) ▼	24 / 3000	<input checked="" type="checkbox"/>	Not Encrypted
Instance Store 0 ▼	/dev/sdb ▼	N/A	N/A	N/A	N/A	N/A	Not Encrypted 
Instance Store 1 ▼	/dev/sdc ▼	N/A	N/A	N/A	N/A	N/A	Not Encrypted 
Instance Store 2 ▼	/dev/sdd ▼	N/A	N/A	N/A	N/A	N/A	Not Encrypted 
Instance Store 3 ▼	/dev/sde ▼	N/A	N/A	N/A	N/A	N/A	Not Encrypted 
Instance Store 4 ▼	/dev/sdf ▼	N/A	N/A	N/A	N/A	N/A	Not Encrypted 
Instance Store 5 ▼	/dev/sdg ▼	N/A	N/A	N/A	N/A	N/A	Not Encrypted 
Instance Store 6 ▼	/dev/sdh ▼	N/A	N/A	N/A	N/A	N/A	Not Encrypted 
Instance Store 7 ▼	/dev/sdi ▼	N/A	N/A	N/A	N/A	N/A	Not Encrypted 

[Cancel](#)
[Previous](#)
[Review and Launch](#)
[Next: Tag Instance](#)

But, for other instances types, e.g. c3.4xlarge, the available SSD storage that comes with the instance type is not setup. So, the Add Storage page will look similar to the following screenshot, Figure 6.

**Figure 6. Add Storage Screenshot 2****Step 4: Add Storage**

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Delete on Termination ⓘ	Encrypted ⓘ
Root	/dev/xvda	snap-f518b274	8	General Purpose (SSD) ▾	24 / 3000	<input checked="" type="checkbox"/>	Not Encrypted

 **Add New Volume**

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Tag Instance](#)

So, to setup the SSD storage that comes with the instance type, you must click on the “Add New Volume” (shown in the above screenshot, Figure 6, by the left margin green arrow).

When you click on “Add New Volume”, a new row of storage will be added to the page. So, the web page will look like the following screenshot, Figure 7. (the newly added row of storage is pointed to by the left margin green arrow ). To add a new SSD store, click on the down arrow just to the right of EBS in the newly added row. The dropdown menu will show what storage can be added, see Figure 8.

**Figure 7. Add Storage Screenshot 3****Step 4: Add Storage**

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.


Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Delete on Termination ⓘ	Encrypted ⓘ
Root	/dev/xvda	snap-f518b274	8	General Purpose (SSD) ▾	24 / 3000	<input checked="" type="checkbox"/>	Not Encrypted
EBS ▾	/dev/sdb ▾	Search (case-insensitive)	8	General Purpose (SSD) ▾	24 / 3000	<input type="checkbox"/>	<input type="checkbox"/>

 **Add New Volume**

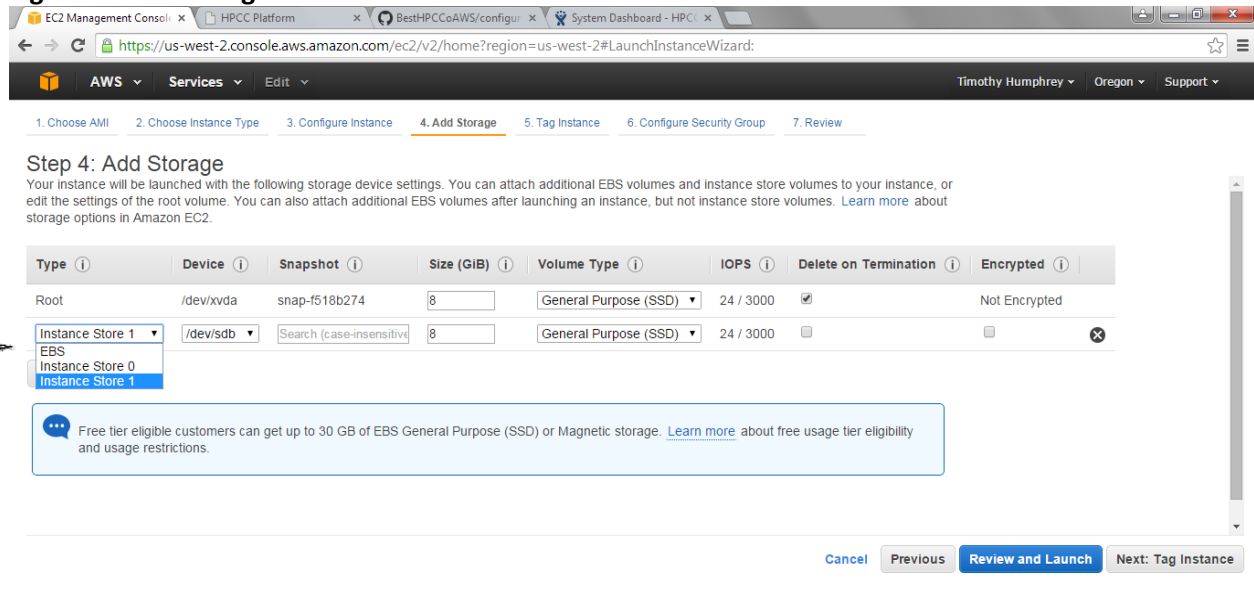
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Tag Instance](#)

The dropdown will look similar to the following screenshot, Figure 8 (the dropdown menu is pointed to by the left margin green arrow). The dropdown menu in Figure 8 shows the storage units available are: an EBS unit and two SSD storage units that show up as Instance Store 0 and Instance Store 1. The number of SSD storage units you see will depend on the instance type you choose. For the following screenshot the instance type was c3.4xlarge which has two 160 GB SSDs that come with it. So, you see two SSD storage units in the dropdown. If the instance type you selected does not come with any SSD storage units, the dropdown menu will only have EBS storage unit on it.

	<p>Be sure to select an SSD storage unit instead of an EBS storage unit (Why? One reason is because the EBS storage units only have 8 GB of storage. Plus, there is a difference in the cost of EBS and SSD storage units).</p>
-----------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Figure 8. Add Storage Screenshot 4**



**Step 4: Add Storage**  
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/xvda	snap-f518b274	8	General Purpose (SSD)	24 / 3000	<input checked="" type="checkbox"/>	Not Encrypted
Instance Store 1	/dev/sdb	Search (case-insensitive)	8	General Purpose (SSD)	24 / 3000	<input type="checkbox"/>	<input type="checkbox"/>

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous **Review and Launch** Next: Tag Instance

Keep adding SSD storage units until all the SSD units are off the dropdown menu. So, this process is: 1) click on “Add New Volume” then click on the dropdown menu arrow and select an “Instance Store” from the menu, if one exists.

When you are finishing adding SSD storage units, click on “Next: Tag Instance” (pointed to by the right margin green arrow in Figure 8, above).


When the “Tag Instance” web page comes up, you don’t make any changes. Just click on the “Next: Configure Security Group” button in the lower right corner of the “Tag Instance” web page.

### Step 6. Configure Security Group

When this page comes-up, it looks like the following screenshot, Figure 9, where the “Create a new security group” button is already selected (see top green arrow, below).

**Figure 9. Configure Security Group Screenshot 1****Step 6: Configure Security Group**

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group  ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere 0.0.0.0/0

[Add Rule](#)

**Warning**

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

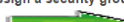
[Cancel](#)
[Previous](#)
[Review and Launch](#)

So, the first thing you do is click on the “Select an existing security group” button, just below the “Create a new security group” button. This will cause the web page to change to look something like the following screenshot, Figure 10, where “Create a new security group” is no longer selected. Instead the “Select an existing security group” button is selected (see top green arrow, in Figure 10).

You will also see a list of existing security groups. In this screenshot, Figure 10, we have 2: one whose name is “default” and another whose name is “tlh-best-security-group”, which is a security group I setup earlier for HPCC Systems (to learn how to setup a security group for an HPCC System, go to Appendix B. Setting up a Security Group For Your HPCC System on AWS. Click the check box (see lower left margin green arrow of Figure 10) to select the security group, “tlh-best-security-group”.

**Figure 10. Configure Security Group Screenshot 2****Step 6: Configure Security Group**


A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☐ Create a new security group  ☒ Select an existing security group

Filter: VPC security groups

Security Group ID	Name	Description	Actions
<input type="checkbox"/> sg-b5a5c7d0	default	default VPC security group	<a href="#">Copy to new</a>
<input checked="" type="checkbox"/> sg-65b0e600	tlh-best-security-group	This security group modeled after Hpcc-SBJU (non-VPC)	<a href="#">Copy to new</a>

Select a security group above to view its inbound rules.

[Cancel](#)
[Previous](#)
[Review and Launch](#) 

Once you have selected your security group, you click on the “Review and Launch” button (see right margin green arrow of Figure 10), which gets you over to “Review Instance Launch” web page which will look something like the following screenshot, Figure 11.

**Step 7. Review Instance Launch**

After you have scrolled through this page to review what you have selected then click on the “Launch” button (pointed to by the right margin green arrow in the following screenshot, Figure 11).

**Figure 11. Review Instance Launch Screenshot****Step 7: Review Instance Launch**

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**⚠ Your instance configuration is not eligible for the free usage tier**

To launch an instance that's eligible for the free usage tier, check your AMI selection, instance type, configuration options, or storage devices. Learn more about [free usage tier](#) eligibility and usage restrictions. [Don't show me this again](#)

**⚠ Improve your instances' security. Your security group, `tlh-best-security-group`, is open to the world.**

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

▼ AMI Details [Edit AMI](#)

**Amazon Linux AMI 2014.09.2 (HVM) - ami-dfc39aef**

**Free tier eligible**

The Amazon Linux AMI is an EBS backed image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Apache HTTPD, Docker, PHP, MySQL, PostgreSQL, and other packages.

Root Device Type: ebs Virtualization type: hvm

[Cancel](#) [Previous](#) [Launch](#)

When you click on “Launch”, you will see a pop-up like the following screenshot, Figure 12. This pop-up lets you select or make a key pair that can be used to ssh into any of the instances. I always use the same key pair for all instances of the HPC System I’m deploying.

There are two dropdown menus in this pop-up. The top one (pointed to by the top left margin green arrow in Figure 12) is set to “Choose an existing key pair”. So, if you already have an existing key pair that you want to use, you select it from the 2<sup>nd</sup> dropdown menu (pointed to by the top green arrow in the right margin of Figure 12). After you have selected the key pair, you have to click on a check box that says you acknowledge that you have saved and have access to the key pair you have selected (pointed to by bottom left margin green arrow)s. Then, you click on “Launch Instances” (pointed to by the bottom green arrow in the right margin).

**Figure 12. Select Existing Key Pair Screenshot 1**

**Select an existing key pair or create a new key pair** ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair ▼

Select a key pair ▼


AWS-AMI-KEY

☐ I acknowledge that I have access to the selected private key file (AWS-AMI-KEY.pem), and that without this file, I won't be able to log into my instance.

[Cancel](#) [Launch Instances](#)

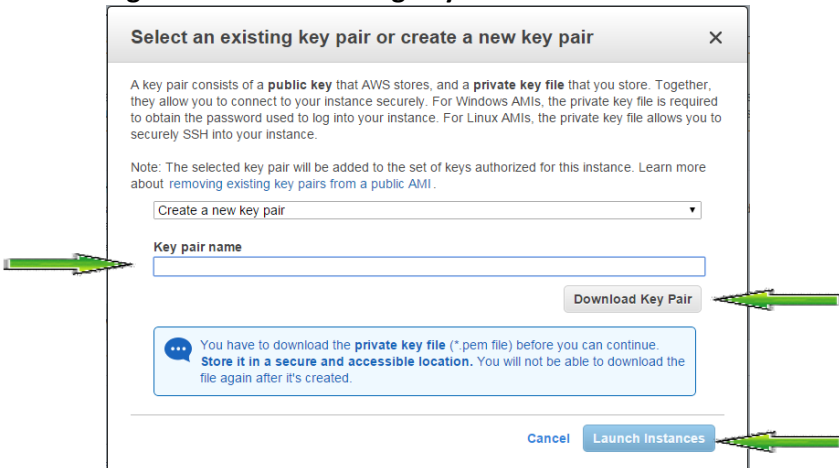
If you want to create a new key pair, then select “create a new key pair” from the top dropdown menu (pointed to by the top left margin green arrow of Figure 12). When you do, the pop-up will change to look like the following screenshot, Figure 13, where the left margin green arrow points to the “key pair name” text box, where you enter the name of the new key pair. Then, you click “Download key pair” (pointed to by the top green arrow in the right margin). This will cause a file to be downloaded

containing the key pair. Save this file where it can be accessed when you desire to ssh into an instance of your HPCC System.

	<p>Since my laptop is Windows 7 and my ubuntu machine is a VMware image, I setup a directory that is shared between the 2 machines. That way, after the key pair is downloaded onto my Windows 7 machine, I can put it in the shared directory and thereby make it available to ssh done on my ubuntu machine.</p>
-----------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Once, your newly created key pair has been downloaded and saved, click the “Launch Instances” button (pointed to by the bottom green arrow in the right margin of Figure 13).

**Figure 13. Select Existing Key Pair Screenshot 2**



**Select an existing key pair or create a new key pair** X

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name

Download Key Pair

You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel Launch Instances

To see the instance(s) you have launched as they are being launched, click on “Instances” in the menu on the left side of the very first screenshot shown above, Figure 1, i.e. the EC2 Dashboard web page screenshot.

## Task 2. Fill in Configuration File, `cfg_BestHPCC.sh`

The following screenshot shows the contents of the configuration file, `cfg_BestHPCC.sh`, as it came from the github repository. And, the table that followings this screenshot, gives a short description of each configuration variable in the file.



```

# cfg_BestHPCC.sh
user=ec2-user
private_ips=private_ips.txt
public_ips=public_ips.txt
created_environment_file=/etc/HPCCSystems/source/newly_created_environment.xml
supportnodes=1
non_support_instances=6
roxienodes=0
slavesPerNode=16
hpcc_platform=hpccsystems-platform_community-with-plugins-5.0.0-3.el6.x86_64.rpm
hpcc_full_path=http://cdn.hpccsystems.com/releases/CE-Candidate-5.0.0/bin/platform/$hpcc_platform
S3_ACCESS_KEY=<your aws access key>
S3_SECRET_KEY=<your aws secret key>
bucket_name=<your bucket>
system_username=<your system username>
system_password=<your system password>
slave_instance_type=i2.8xlarge
master_instance_type=c3.4xlarge
pem=<your pem file name>
infolder=instance_files
instance_ids=instance_ids.txt
region=<your region>

```

Table 2. cfg\_BestHpcc.sh Configuration File Variables

Cfg_BestHPCC.sh Configuration File Variables		
Var #	Variable Name	Explanation
1	user	User name for logging into launched EC2 instances
2	private_ips	Name of file containing private IPs of all launched instances.
3	public_ips	Name of file containing public IPs of all launched instances.
4	created_environment_file	Name of file containing environment.xml file created by envgen.
5	supportnodes	Number of support instances (should be 1)
6	non_support_instances	Number of instances containing thor slave nodes.
7	roxienodes	Number of roxie nodes
8	slavesPerNode	Number of thor slave nodes per instance.
9	hpcc_platform	Name of HPCC Platform that is placed on your HPCC System
10	hpcc_full_path	The full path to the HPCC Platform named above.
11	S3_ACCESS_KEY	Your AWS access key
12	S3_SECRET_KEY	Your AWS secret key
13	bucket_name	Name of S3 bucket to be mounted to dropzone (if none, omit)
14	system_username	Your username for ECL IDE and ECL Watch
15	System_password	Your password for ECL IDE and ECL Watch
16	slave_instance_type	Instance type of instances having thor slave nodes
17	master_instance_type	Instance type of instances having the support components (master, etc).
18	pem	Key pair (public and private) for secure ssh into launched instances.
19	infolder	Directory containing all files copied to launched instances.
20	instance_ids	Name of file containing instance ids for all instances launched (master instance must always be first in file)

21	region	AWS region where all instances are launched.
----	--------	----------------------------------------------



If you specify a `system_username` and `system_password` in the `cfg_BestHPCC.sh` configuration file then you will have to log into ECL Watch using these credentials. But because there is a bug in ECL Watch, you may be told that your password has expired. Ignore this or any related message(s) by clicking on “X”.

Most of the variables in this file don’t need to be changed. The only variables whose values must change are those with values contained in angled brackets. These variables are: `S3_ACCESS_KEY`, `S3_SECRET_KEY`, `bucket_name`, `system_username`, `system_password`, `pem`, and `region`. But, if you don’t have an S3 bucket to mount to the dropzone then you can remove `S3_ACCESS_KEY`, `S3_SECRET_KEY`, and `bucket_name` from the configuration file. Also, if you don’t want a `system_username` and `system_password` then remove these or comment them out.



Read [this](#) to learn how to get your access and secret keys.

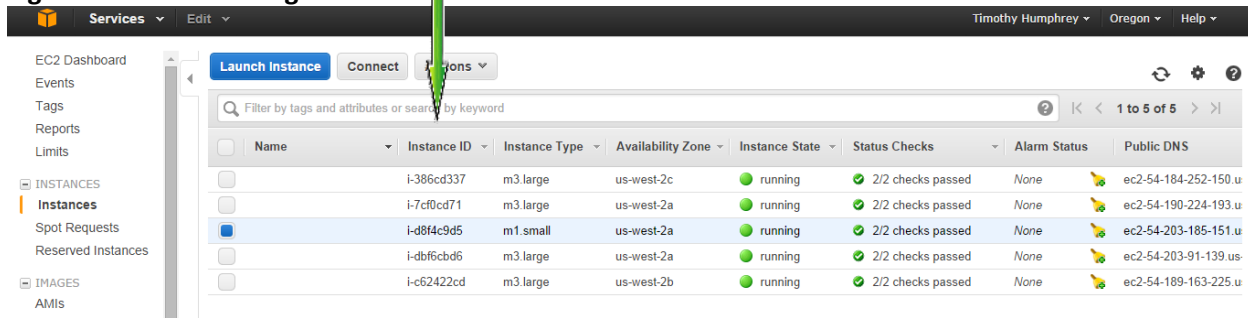
One scenario where the variable, `non_support_instances`, must change, is the case where you don’t have 7 instances launched (the initial `cfg_BestHPCC.sh` configuration file assumes you will launch 7 instances – 1 for the THOR master and 6 for THOR slave nodes). But, if you launch a number different than 7 then the variable, `non-support_instances`, should be 1 minus the number of instances launched (except the in case where the number of instances launched is just one then it is one).

Also, if you want a different number of THOR slave nodes per instance, then the variable, `slavesPerNode` must be the number you want.

### Task 3. Get Instance IDs of All Launched Instances

The instance IDs for all launched instances of your HPCC System have to be placed in the file, `instance_ids.txt` (or the file name that is the value for the configuration file variable, “`instance_ips`”).

Normally, how I get the Instance ids for all launched instances is to 1) go to the EC2 Dashboard, see Figure 1 or Figure 14 above, and 2) click on Instances in the left side menu. The page that comes up should look something like the following screenshot, Figure 19. The green arrow points to the column containing the instance ids for the instances launched. So, you can look at this page and copy to the file, `instance_ids.txt`, the instance IDs shown in the Instance ID column of Figure 19 .

**Figure 14. Instances Page**


Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-386cd337	m3.large	us-west-2c	running	2/2 checks passed	None	ec2-54-184-252-150.u
	i-7cf0cd71	m3.large	us-west-2a	running	2/2 checks passed	None	ec2-54-190-224-193.u
	i-d8f4c9d5	m1.small	us-west-2a	running	2/2 checks passed	None	ec2-54-203-185-151.u
	i-dbf6cbd6	m3.large	us-west-2a	running	2/2 checks passed	None	ec2-54-203-91-139.us
	i-c62422cd	m3.large	us-west-2b	running	2/2 checks passed	None	ec2-54-189-163-225.u

Another method of getting the instance IDs is to execute the perl script, `getInstanceIDs.pl`, as so.

```
./getInstanceIDs.pl
```

This script creates the file, `instance_ids.txt`. Its content looks like the following screenshot.

```
user@ubuntu:~/hpcc-20140819/BestHoA$ cat instance_ids.txt
i-488a9a44
i-498a9a45
i-448a9a48
user@ubuntu:~/hpcc-20140819/BestHoA$
```

If you run the above perl script, there are two concerns. First, if there are instances launched in your region other than those you launched for your HPCC, their instance IDs will show up in the list shown in the above screenshot. So, make sure their instance IDs are not put in the file, `instance_ids.txt`. Second, the instance ID for the THOR master must be the first ID in the file, `instance_ids.txt`. So, verify that the first instance ID is that of the master's by looking at the instances on the Instances page of the aws console (e.g. Figure 14, above).

#### Task 4. Run `getPublicAndPrivateIPsOfAllInstances.pl` to Get Public & Private IPs

Once you have the instance IDs in the file referenced in `cfg_BestHPCC.sh` as "`instance_ids`", with the ID of the master instance as the first ID in the file then, use the following command to get the private and public IPs for all instances.

```
./getPublicAndPrivateIPsOfAllInstances.pl 1> ~/getIPs.log 2> ~/getIPs.err
```



Notice that when the program is ran, its output and stderr go to files, `getIPs.log` and `getIPs.err`, respectively, in my home directory. I do this so I can look to see if everything ran as I expected. I do the same thing with other commands below.

This command uses the aws cli to get the public and private IPs. It puts the public IPs in a file of the current directory where this command is executed and the private IPs in a file of the sub-directory, instance\_files.

## Task 5. Run cpServerFilesToAllInstances.pl to Copy Files to Instances

```
./cpServerFilesToAllInstances.pl 1> ~/getServerFiles.log 2> ~/getServerFiles.err
```



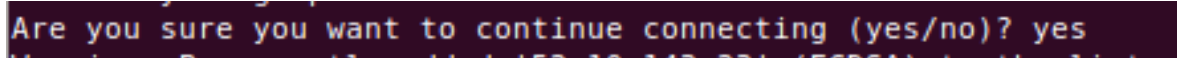
Before you run the above command or any of the other commands that follow, make sure the .pem file containing the ssh keys you selected in Figure 12 of Task 1 are in the BestHPCCoAWS directory. If the .pem file exists on your ubuntu linux machine then you can use the 'cp' command to copy the .pem file to the BestHPCCoAWS directory.

If you have a vmware ubuntu linux machine that is hosted on a windows machine and you have downloaded your .pem file to a windows directory then create a shared windows folder that can be accessed by your vmware machine. Instructions for doing so are in the pdf, "Getting Started with VMware Player". Here is its link:

[https://www.vmware.com/pdf/desktop/vmware\\_player50.pdf](https://www.vmware.com/pdf/desktop/vmware_player50.pdf). In this pdf, go to Using Virtual Machines > Transferring Files And Text > Using Shared Folders, which is on page 42.

This command copies all files of the instance\_files directory to all of the launched instances. These are files needed to complete Tasks 6 through 13.

When you execute the above perl code, since it is the first time you are ssh'ing into each of the launched instances, you will get the following prompt. So, answer with "yes". As shown in the following screenshot.



```
Are you sure you want to continue connecting (yes/no)? yes
```

## Task 6. Run setupDisksOnAllInstances.pl to Raid the Disks Etc. on All Instances

```
./setupDisksOnAllInstances.pl 1> ~/setupDisks.log 2> ~/setupDisks.err
```

This command sets up SSD disks that come with the instance types launched. The setup process, raids all SSD, makes them as one drive, /dev/md0, and then mounts them onto /var/lib/HPCCSystems. So, this directory has a lot of space.



Make sure you are in the directory, BestHPCCoAWS, when you execute any of the commands given in Tasks 4 through 12.

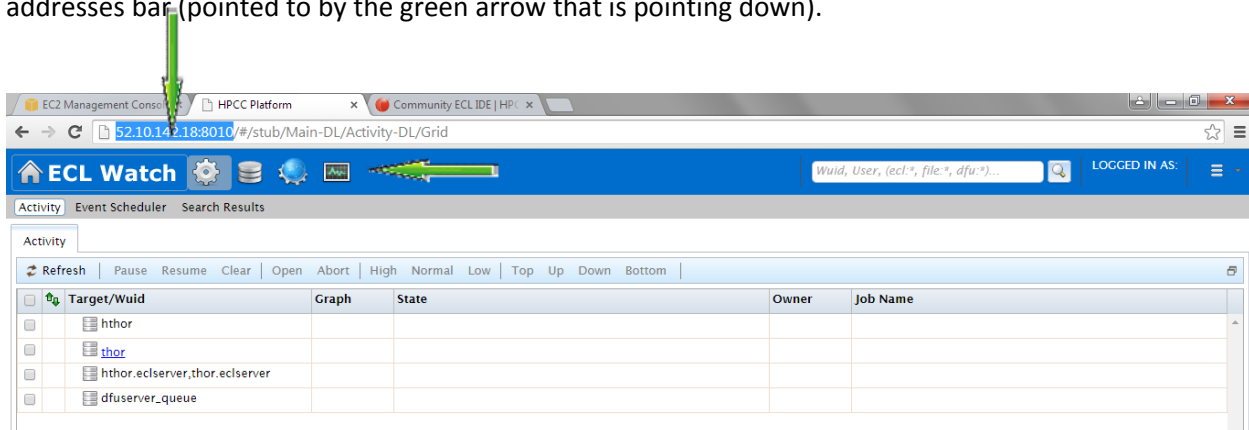
## Task 7. Run installHPCCOnAllInstancesAndStart.pl to Install the HPCC on All Instances

```
./installHPCCOnAllInstancesAndStart.pl 1> ~/installHPCC.log 2> ~/installHPCC.err
```

This command does an initial installation of the HPCC System on each launched instance. The result of this installation is an HPCC System that has 1 THOR slave node on each launched instance.

## Task 8. Make Sure the System is Running By Bringing Up ECL Watch

To check to see if the HPCC System was installed, get the first public IP address, which is the master's, from the file, public\_ips.txt. Put it in your browser's address box with :8010 just to the right of it. Then, when you click on Enter, ECL Watch for your deployed HPCC System should come-up (see the following screenshot for an example). The THOR master's public IP with the ECL Watch are highlighted in the addresses bar (pointed to by the green arrow that is pointing down).



## Task 9. Run configureHPCC\_multislaves\_per\_instance.pl

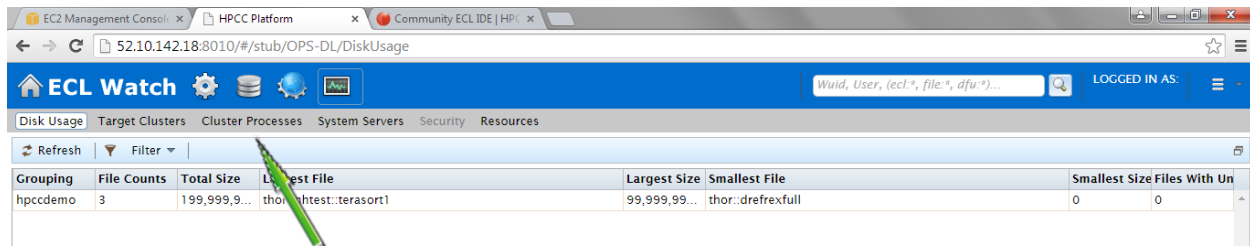
```
./configureHPCC_multislaves_per_instance.pl 1> ~/cfgMSPI.log 2> ~/cfgMSPI.err
```

This command re-configures the HPCC System so it has more than 1 THOR slave node per instance. The number of THOR slave nodes per instance is given by the configuration variable, slavesPerNode, in the configuration file, cfg\_BestHPCC.sh (the performing HPCC THOR has 16 slave nodes per instance).

## Task 10. Make Sure the System is Running By Bringing Up ECL Watch

Do the same thing you did in Task 8 to make sure the newly configured HPCC System is running. Also, to make sure the correct number of THOR slave nodes per instance exists, click on the Operations icon of ECL Watch (this icon is pointed to by the left pointing green arrow in the above ECL Watch screenshot).

When you click on the Operations icon, you will get a web page that looks like the following screenshot.



EC2 Management Console | HPCC Platform | Community ECLIDE | HPCC

52.10.142.18:8010/#/stub/OPS-DL/DiskUsage

ECL Watch

Wuid, User, (ecl-\*, file-\*, dfu-\*)...

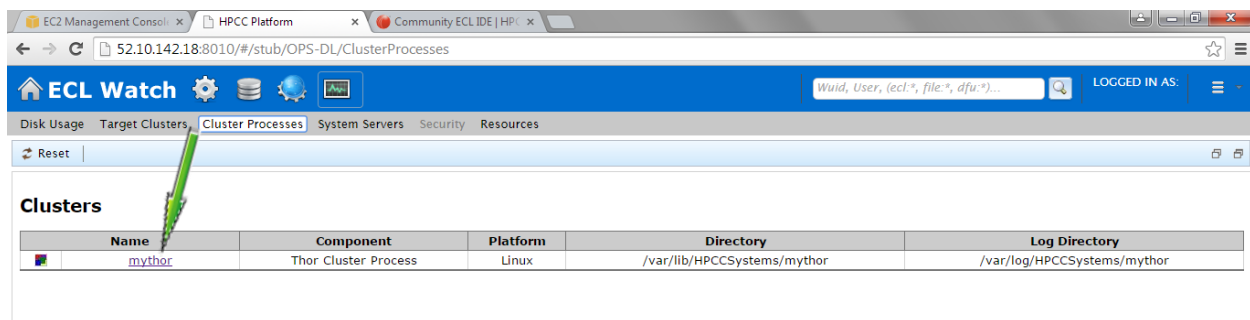
LOGGED IN AS: [User]

Disk Usage | Target Clusters | Cluster Processes | System Servers | Security | Resources

Refresh Filter

Grouping	File Counts	Total Size	Largest File	Largest Size	Smallest File	Smallest Size	Files With Un
hpccdemo	3	199,999,9...	thor::hptest::terasort1	99,999,99...	thor::drefrexfull	0	0

On this page, click on “Cluster Processes” which is pointed to by the green arrow in the above screenshot. This brings up a web page that looks like the following screenshot.



EC2 Management Console | HPCC Platform | Community ECLIDE | HPCC

52.10.142.18:8010/#/stub/OPS-DL/ClusterProcesses

ECL Watch

Wuid, User, (ecl-\*, file-\*, dfu-\*)...

LOGGED IN AS: [User]

Disk Usage | Target Clusters | Cluster Processes | System Servers | Security | Resources

Reset

Clusters

Name	Component	Platform	Directory	Log Directory
mythor	Thor Cluster Process	Linux	/var/lib/HPCCSystems/mythor	/var/log/HPCCSystems/mythor

Then, click on “mythor” which is pointed to by the green arrow of the above screenshot. This will bring up a web page that looks like the following screenshot. And, this page shows the nodes of your deployed THOR. You can scroll down to determine how slave nodes were configured. And, the Network Address, pointed to by the green arrow, below, gives the private IP of each node. So, you can determine how many slave nodes are on each instance (indicated by the private IP).

**Thor Cluster 'mythor'**

	Name	Network Address	Component	Slave Number	Domain	Platform
<input checked="" type="checkbox"/>	node000113	10.0.0.113	Thor Master		localdomain	Linux
<input checked="" type="checkbox"/>	node000236	10.0.0.236	Thor Slave	1	localdomain	Linux
<input checked="" type="checkbox"/>	node000237	10.0.0.237	Thor Slave	2	localdomain	Linux
<input checked="" type="checkbox"/>	node000238	10.0.0.238	Thor Slave	3	localdomain	Linux
<input checked="" type="checkbox"/>	node000236	10.0.0.236	Thor Slave	4	localdomain	Linux
<input checked="" type="checkbox"/>	node000237	10.0.0.237	Thor Slave	5	localdomain	Linux
<input checked="" type="checkbox"/>	node000238	10.0.0.238	Thor Slave	6	localdomain	Linux
<input checked="" type="checkbox"/>	node000236	10.0.0.236	Thor Slave	7	localdomain	Linux
<input checked="" type="checkbox"/>	node000237	10.0.0.237	Thor Slave	8	localdomain	Linux
<input checked="" type="checkbox"/>	node000238	10.0.0.238	Thor Slave	9	localdomain	Linux
<input checked="" type="checkbox"/>	node000236	10.0.0.236	Thor Slave	10	localdomain	Linux
<input checked="" type="checkbox"/>	node000237	10.0.0.237	Thor Slave	11	localdomain	Linux
<input checked="" type="checkbox"/>	node000238	10.0.0.238	Thor Slave	12	localdomain	Linux
<input checked="" type="checkbox"/>	node000236	10.0.0.236	Thor Slave	13	localdomain	Linux
<input checked="" type="checkbox"/>	node000237	10.0.0.237	Thor Slave	14	localdomain	Linux
<input checked="" type="checkbox"/>	node000238	10.0.0.238	Thor Slave	15	localdomain	Linux
<input checked="" type="checkbox"/>	node000236	10.0.0.236	Thor Slave	16	localdomain	Linux
<input checked="" type="checkbox"/>	node000237	10.0.0.237	Thor Slave	17	localdomain	Linux
<input checked="" type="checkbox"/>	node000238	10.0.0.238	Thor Slave	18	localdomain	Linux

## Task 11. Mount S3 Bucket on Dropzone (if needed)

```
./mountS3BucketOntoDropZone.pl 1> ~/mountS3.log 2> ~/mountS3.err
```

The above command will mount an S3 bucket on your deployed HPCC's dropzone. The S3 bucket that is mounted is the one you have named in the configuration file, `cfg_BestHPCC.sh`.

So, you can use the files in the S3 bucket that is mounted, you must make sure all files that you want to use has permissions of 777. One way to assure they do is to ssh into the master instance, which is where the dropzone is at. Then do the following command.

```
sudo chmod 777 /var/lib/HPCCSystems/mydropzone/*
```

This command makes sure that all files in your S3 bucket has the permissions, 777. Plus, it only needs to be done once, because these permissions are stored in the S3 bucket. So, the next time you mount this bucket, you don't have to this `chmod` command again.



If the files in your S3 bucket are large or there are many of them, it will take a while for the `chmod` to complete its job.

## Task 12. Run `updateSystemFilesOnAllInstances.pl`

```
./updateSystemFilesOnAllInstances.pl 1> ~/updateSysFiles.log 2> ~/updateSysFiles.err
```

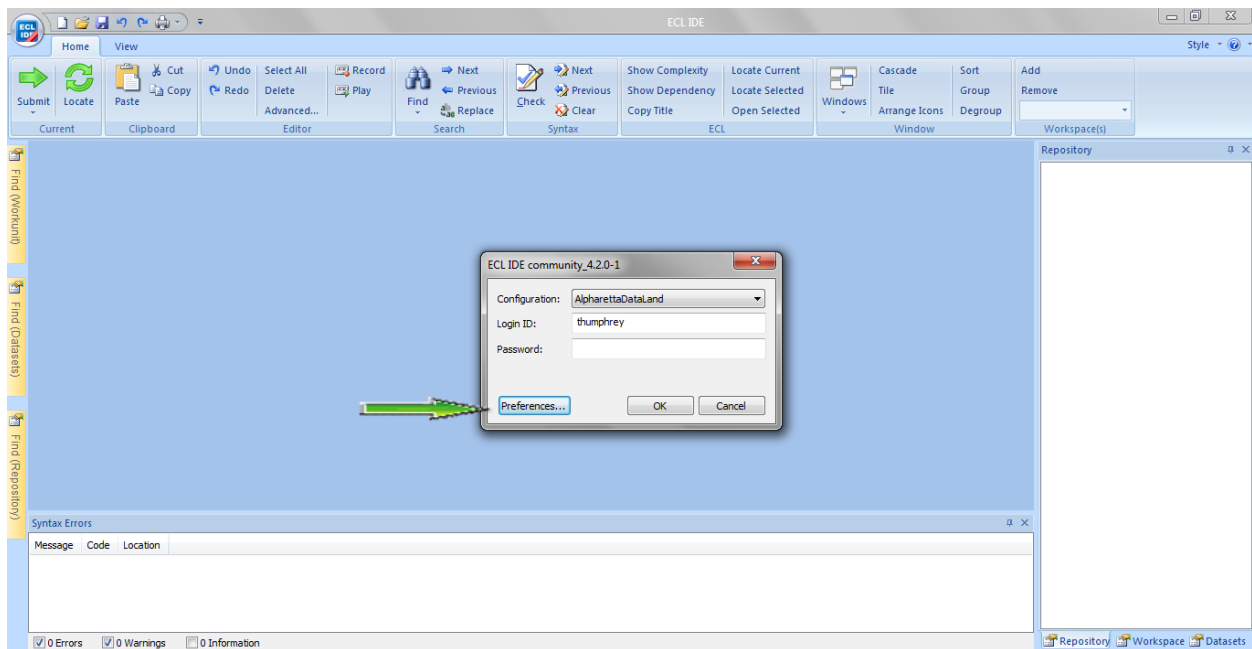
This command, changes some settings in 3 of the linux system files because some of default settings are too low for an HPCC System that has so many THOR slave nodes per instance.

### Task 13. Setup ECL IDE to Work with the Deployed HPCC System on AWS

Now that your HPCC System is running on AWS, it is time to setup the ECL IDE so you can do real work with it. First, you must download the ECL IDE at <http://hpccsystems.com/download/free-community-edition/ecl-ide>.

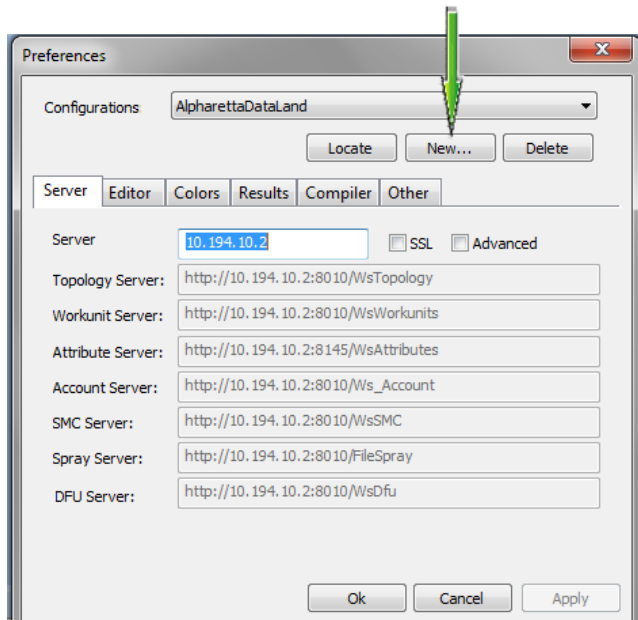
There are two places that need to be setup, both of which are in Preferences: 1) A new configuration with the IP of your HPCC System's ESP must be created, and 2) some changes need to be made under the Compiler tab.

When ECL IDE first opens, it will look like the following screenshot with a popup for logging-in that has a button for going to Preferences popup (pointed to by the green arrow).

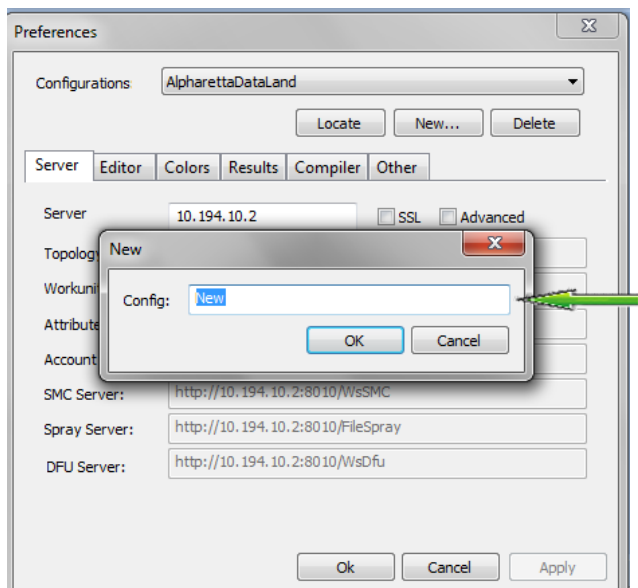


Click on the Preferences button to go to the Preferences popup which will look like the following screenshot.

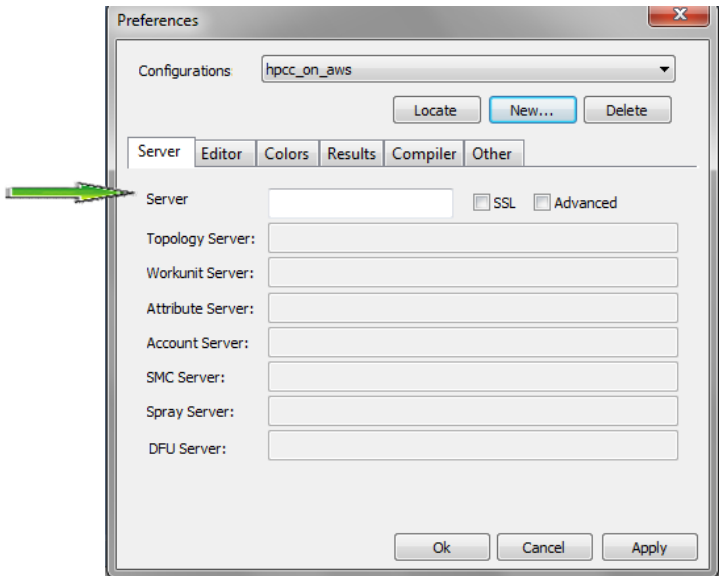




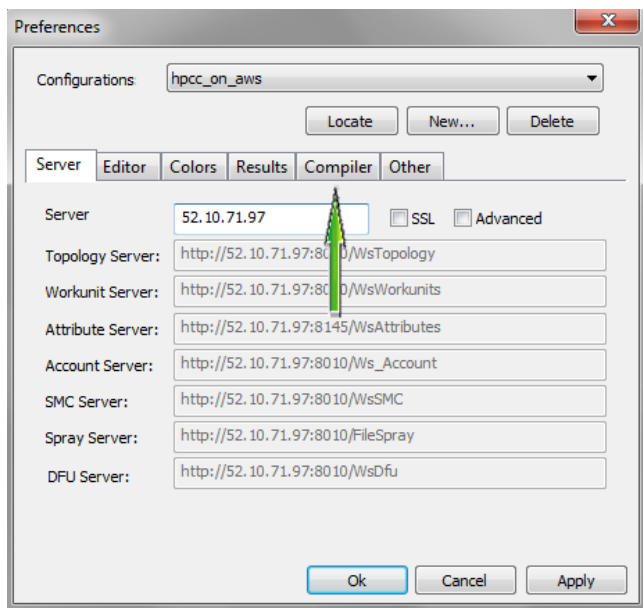
Next, click on the “New” button (pointed to by the green arrow in the above screenshot) which lets you setup a new configuration. Clicking on the “New” button causes another popup to come-up that lets you enter a name for the new configuration ( See the following screenshot). Enter in a name for your new configuration in the Config text box (pointed to by the green arrow in the following screenshot).



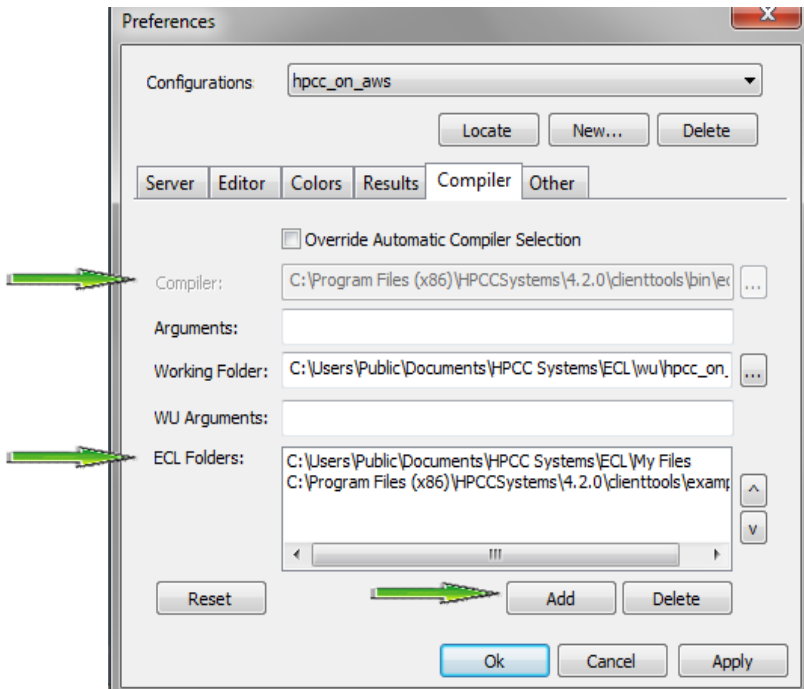
After entering in a name, I called mine “hpcc\_on\_aws”, then, click OK which causes the “New” popup to go away and a blank “Preferences” popup to be displayed like the following screenshot.



So, in the “Server” text box (pointed to by the green arrow in the above screenshot), enter in the THOR master’s public IP address (this is also the public IP of the ESP) which is the first IP address of the file, public\_ips.txt. For me, that public IP was 52.10.71.97. After entering in this IP, the Preferences popup will change to look similar to the following screenshot, where all the text boxes below the “Server” text box, as well as the “Server” text box, will be filled-in like the following screenshot.



Next, click on the “Compiler” tab (pointed to by the green arrow in the above screenshot). This causes the Preferences popup to display the contents of the “Compiler” tab, which looks like the following screenshot.



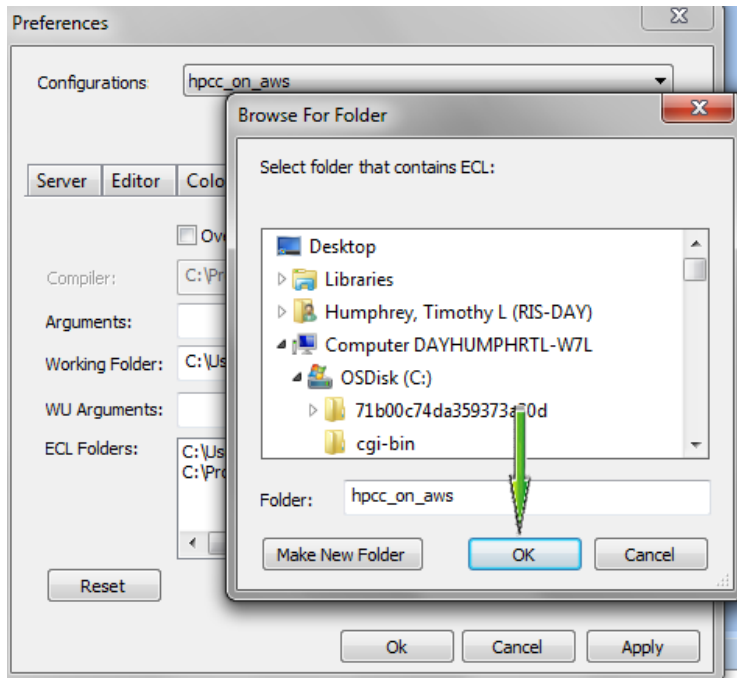
There are a couple text boxes that you may have to change on the “Compiler” tab contents popup: 1) the “Compiler” text box (whose name is grayed out in the above screenshot and is pointed to by the top left margin green arrow), and 2) the “ECL Folders” text box (pointed to by the bottom left margin green arrow).

You may want to change the compiler, whose full path is in the “Compiler” text box (in the above screenshot that is version 4.2.0). This compiler is only used to do syntax checks on local repositories of ECL code. So, if your ECL code contains keywords that aren’t known to compiler 4.2.0 then you will get syntax errors.

If you want to change the compiler whose full path is in the “Compiler” text box, first click on the “Override Automatic Compiler Selection” check box that is just above the “Compiler” text box. This enables the contents of the “Compiler” text box to be changeable. So, second, you change the full path of this text box to that for the compiler you want (click [here](#) to download various versions of the clienttools (which includes the ecl compiler).

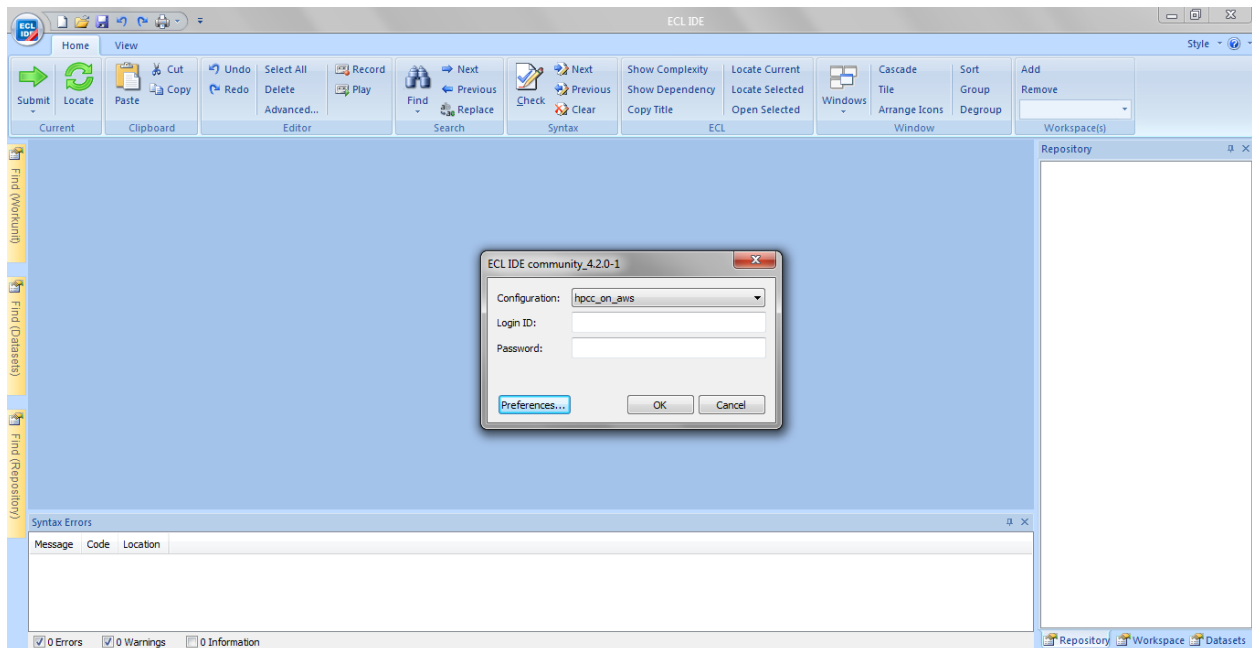
If you have ecl programs that you want to run in your ECL IDE, then a full path to their directory must be in the list of full paths of the “ECL Folders” text box. Here is how you add another full path to the “ECL Folders” text box.

First, click on the “Add” button just below the “ECL Folders” text box (pointed to by a green arrow in the above screenshot). This causes another popup to appear, called “Browse For Folder”, that lets you find the folder you want to add (it looks like the following screenshot).



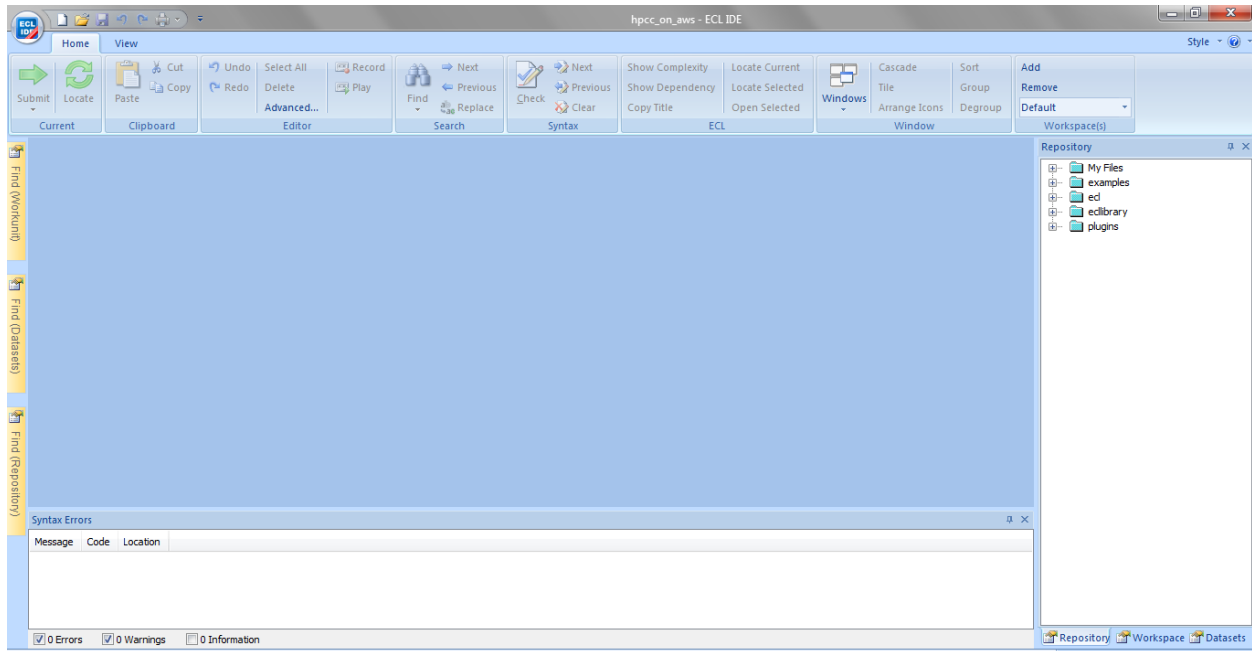
Once you find the folder containing the ECL programs you want, you click on “OK” (pointed to by the green arrow in the above screenshot). Then, the new path is added to the “ECL Folders” text box.

Once, you have completed making changes to the contents of the “Compiler” tab text boxes, then click on “Apply” and “OK” to have your ECL IDE updated with all your changes. And, after clicking on “OK”, what you see next should look like the following screenshot.



To complete the process of setting up and logging into your ECL IDE, enter “hpcddemo” (without the quotes) for both the Login ID and Password and then click on “OK”.

The next screen you see should look something like the following screenshot. So, you are ready to use the HPCC System that you configured and deployed on AWS.

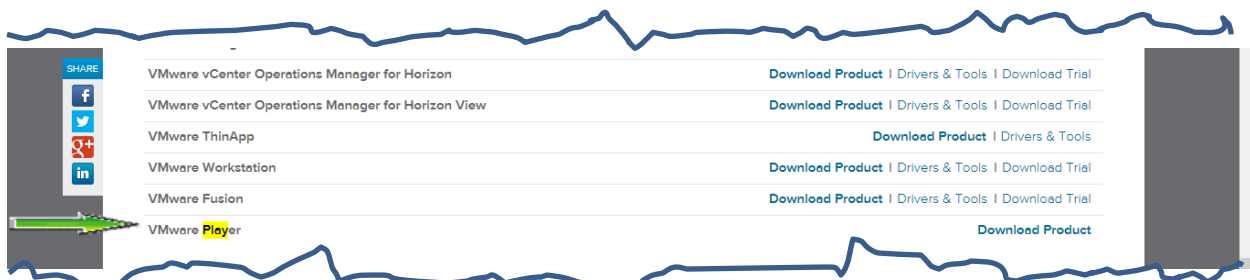


## Appendix A. Setting Up Ubuntu 12.04 Linux VMware Machine

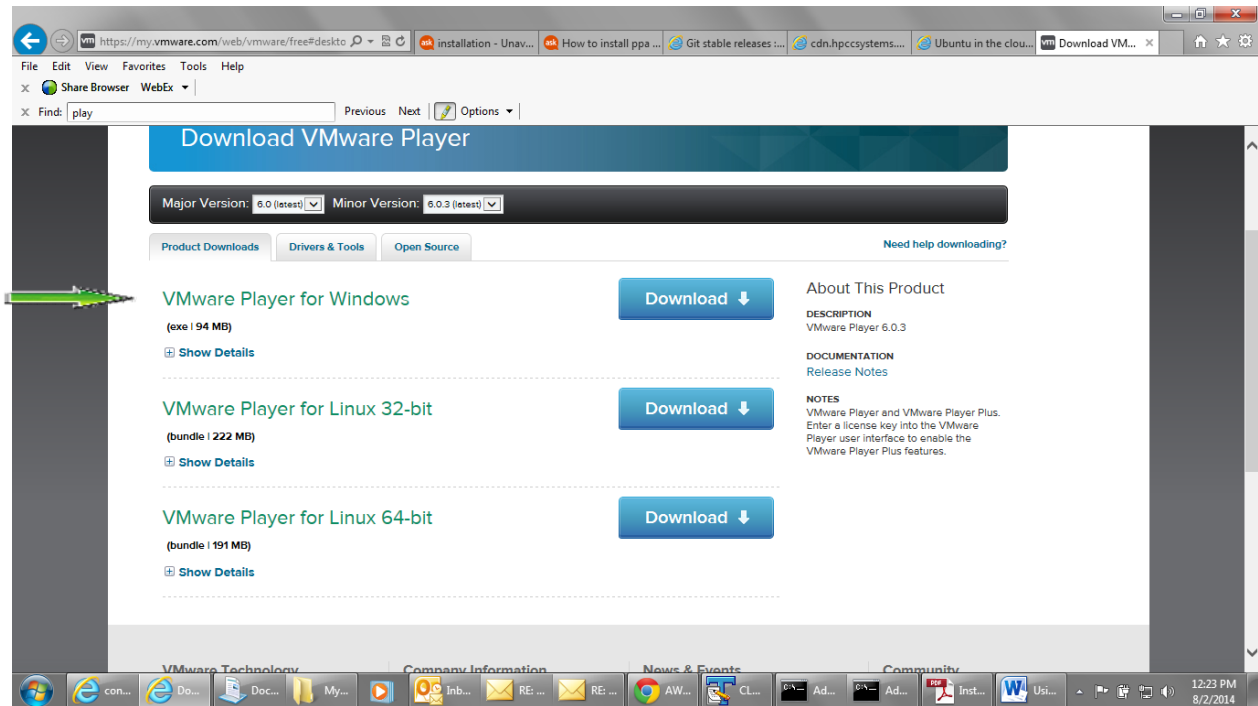
### Download and Install VMware Player

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:

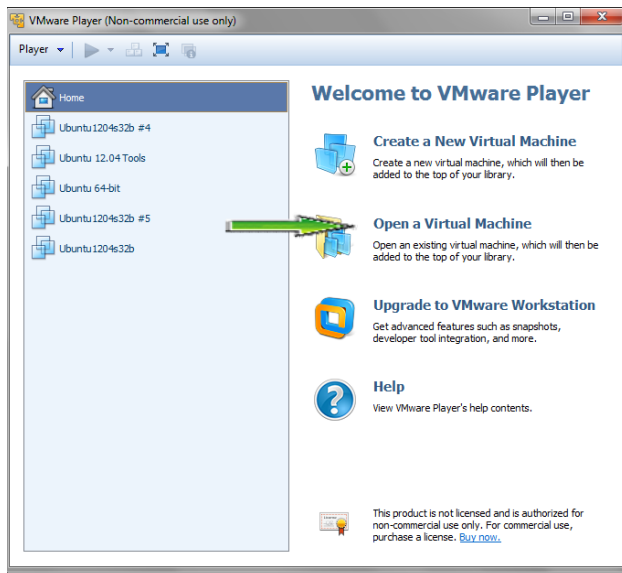


To download Ubuntu 12.04 with the VMware Tools, click on the link pointed to by the green arrow, above. Save the downloaded zip file where you can find it. The download may take several minutes.

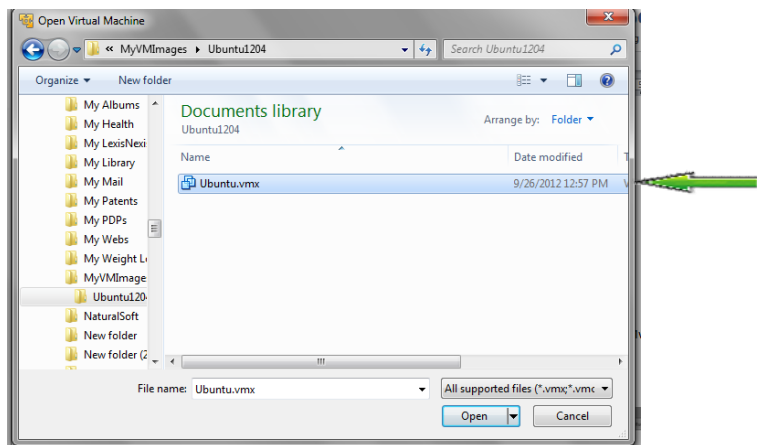
After the download is complete, extract the contents of the zip file and place it in a folder where you can find it. I've stored mine in Documents\MyVMImages.

## 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).



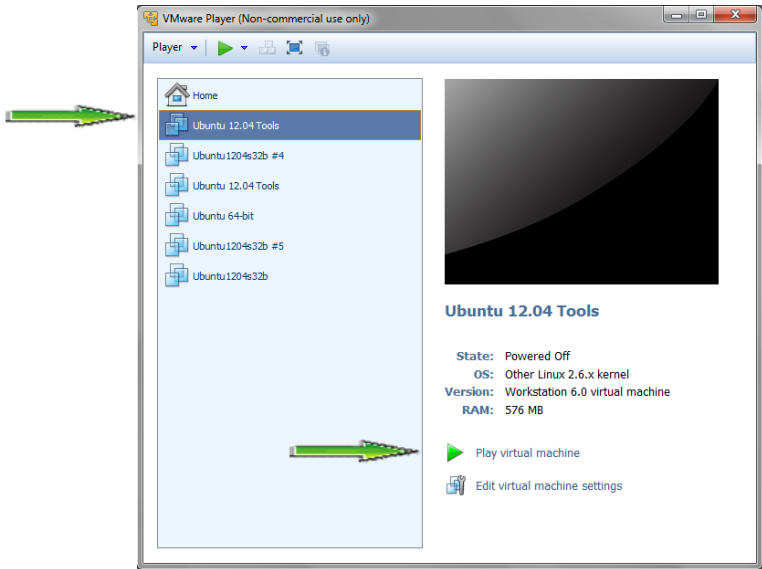
This will open the Windows Explorer so you can open the folder containing the Ubuntu 12.04 VM image. Once you have found it, you should see the file Ubuntu.vmx (see below screenshot).



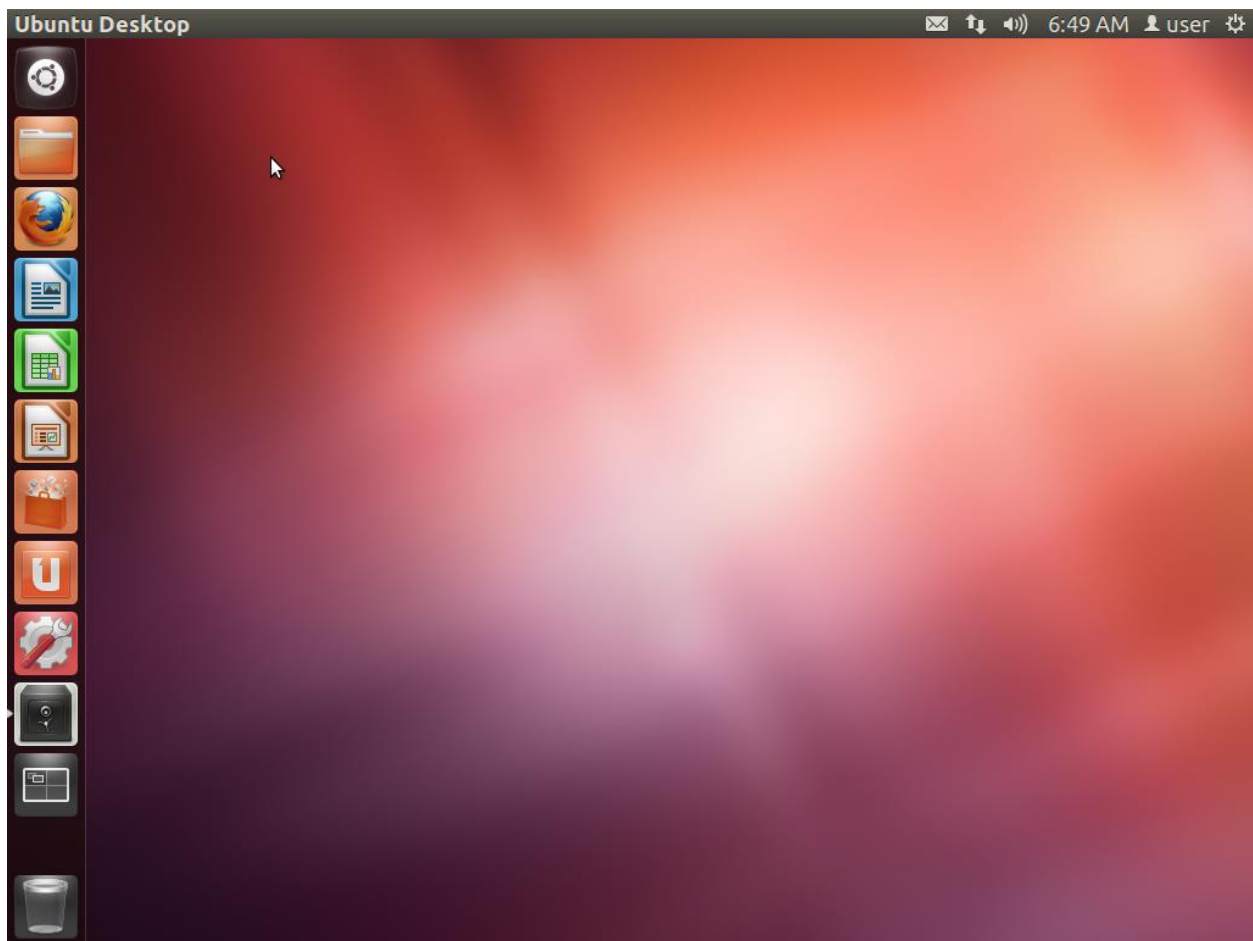
Click on it so it will be listed in the VMware Player's list of virtual machines (see the top green arrow in the following screenshot). As you can see, it has already been selected. So, you can click on Play virtual machine to start the virtual machine (see bottom green arrow in the following screenshot).

	<p>When the machine starts, you may see a popup asking to check for updates. Click the X to ignore it.</p>
--	------------------------------------------------------------------------------------------------------------

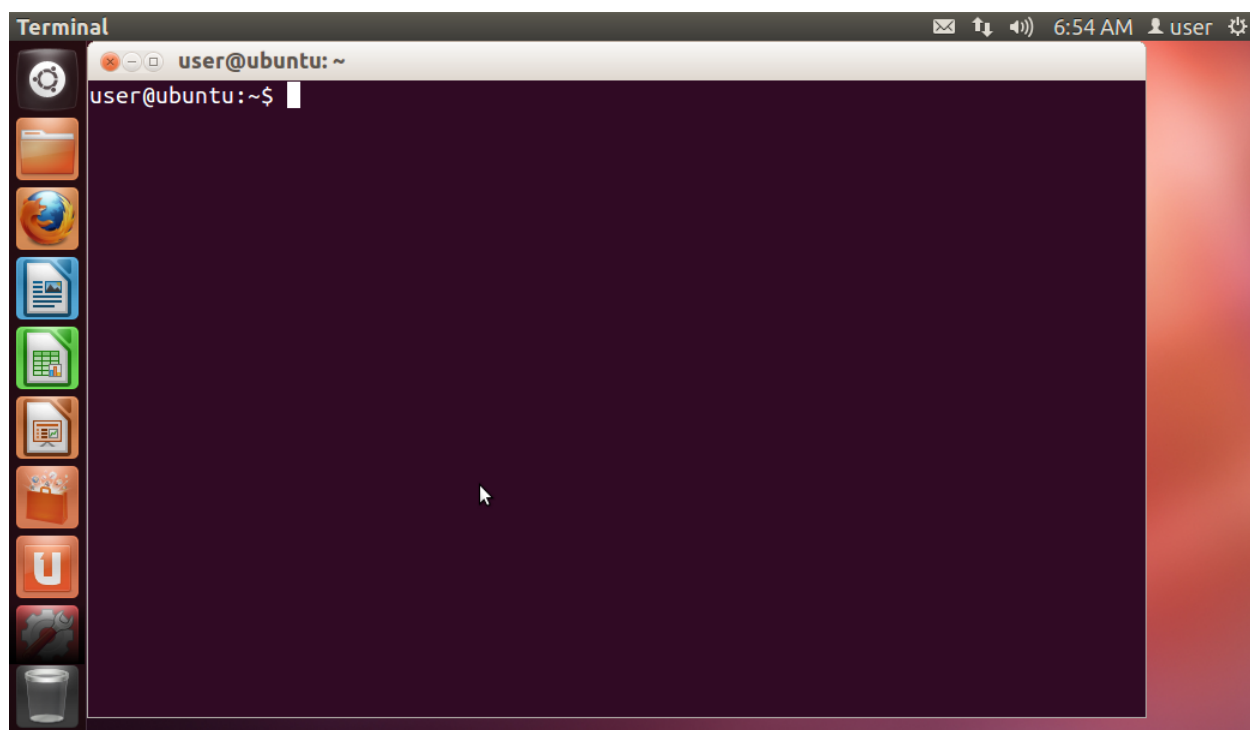




When the machine is ready for use, its screen should look like the following screenshot.



So, at this point, you can open an Ubuntu Terminal window with CTRL-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.



## Appendix B. Setting up a Security Group For Your HPCC System on AWS

On the EC2 Dashboard web page which is Figure 1 above and which I duplicated below, from the list on the left of the EC2 Dashboard, select “Security Groups” (pointed to by the left margin green arrow, below).

Figure 15. EC2 Dashboard Screenshot 2

The screenshot shows the AWS EC2 Dashboard. On the left, the navigation menu is expanded to 'NETWORK & SECURITY', and 'Security Groups' is highlighted with a green arrow. The main content area displays 'Resources' for the US East (N. Virginia) region, listing various EC2 resources. Below the resource list, there's a 'Create Instance' section and a 'Service Health' section indicating that the service is operating normally.

### Security Group

The Security Group web page looks something like the following screenshot which lists the security groups that already exists. And, at the top is the “Create Security Group” which you click on to make a new security group (the top green arrow points at this button).

Figure 16. Security Group Screenshot

The screenshot shows the AWS Security Groups page. At the top, there's a 'Create Security Group' button highlighted with a green arrow. Below this is a search bar and a table of existing security groups. The table has columns for Name, Group ID, Group Name, VPC ID, and Description. It lists several security groups, including 'tlh-best-security-group' and 'Thor-tim1'.

Name	Group ID	Group Name	VPC ID	Description
<input type="checkbox"/>	sg-65b0e600	tlh-best-security-group	vpc-dbe731be	This security group modeled after Hpcc-SBJU (non-VPC)
<input type="checkbox"/>	sg-3ed3160d	Thor-tim1		Visit http://hpccsystems.com for more information on Th...
<input type="checkbox"/>	sg-70933143	launch-wizard-9		launch-wizard-9 created 2014-12-04T15:12:26.285-05:00
<input type="checkbox"/>	sg-62923051	launch-wizard-8		launch-wizard-8 created 2014-12-04T14:54:28.307-05:00
<input type="checkbox"/>	sg-d08d2fe3	launch-wizard-7		launch-wizard-7 created 2014-12-04T14:39:31.044-05:00

When you click on “Create Security Group”, you get a pop-up that looks like the following screenshot. The green arrows point to text boxes that need to be changed. For the “Security group name”, enter any valid name (meaning it contains only letters, numbers, or underscores). For “Description” enter anything that is descriptive. For “VPC”, select one from the dropdown menu. If the only selection is “No VPC” or those on the dropdown menu are different than one created using Appendix C’s instructions then use Appendix C’s instructions to create one.

**Figure 17. Create Security Group Screenshot 1**



The VPC selected above should be the same as the one selected for the Network of Figure 4 which is in Step 3 of Task1, above.

Next you will add “Inbound” rules that define the ports needed by the HPCC System for communicating with us (e.g. through ECL Watch) and its various components. When you get done, your “Inbound” rules should look like the following screenshot, which has been labeled as Table 3.

**Table 3. HPCC System's Inbound Rules**

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
Custom TCP Rule	TCP	8002	0.0.0.0/0
Custom TCP Rule	TCP	8015	0.0.0.0/0
Custom TCP Rule	TCP	8010	0.0.0.0/0
Custom TCP Rule	TCP	8145	0.0.0.0/0
SSH	TCP	22	0.0.0.0/0
All TCP	TCP	0 - 65535	0.0.0.0/0
Custom TCP Rule	TCP	8050	0.0.0.0/0
All UDP	UDP	0 - 65535	0.0.0.0/0
Custom TCP Rule	TCP	8008	0.0.0.0/0
Custom TCP Rule	TCP	9876	0.0.0.0/0
All ICMP	All	N/A	0.0.0.0/0

To get your “Inbound” rules to look like the above, on the “Create Security Group” pop-up (see the following screenshot) do the following.

Start by clicking on “Add Rule” (where green arrow points).

**Figure 18. Create Security Group Screenshot 2**

**Create Security Group** [X]

Security group name ⓘ

Description ⓘ

VPC ⓘ

Security group rules:

**Inbound** Outbound

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
This security group has no rules			

This causes the pop-up to change so it looks like the following screenshot. A new “Inbound” rule has been added (where green arrow points in the following screenshot).

Then, you modify this new “Inbound” rule to look exactly like one of the rules in Table 3, above. And since there are 11 “Inbound” rules in Table 3, you add 11 rules and modify them.

For example, to change the added rule shown in the screenshot below to the 5<sup>th</sup> rule of Table 3, the one whose type is SSH, you would select SSH from the “Type” dropdown menu, which automatically changes “port range” column of the newly added rule to 22, which matches that of Table 3. And, if you select “Anywhere” from the “Source” dropdown menu then the value of the “source” column matches that of Table 3. So, you have finished created an “Inbound” rule that matches the SSH rule of Table 3.

The only rule that will look incorrect is the rule whose type is “All ICMP”. Table 3 says that its Protocol value should be “All” and its Port Range value should be “N/A”. But, when you are setting up this rule, the Protocol value will be “ICMP” and the Port Range value will be “0-65535”. But, once you have clicked on SAVE, these values will change to match what is in Table 3.

**Figure 19. Create Security Group Screenshot 3**

**Create Security Group** [X]

Security group name ⓘ my-hpcc-security-group

Description ⓘ this security group contains the ports needed for the hpcc system

VPC ⓘ vpc-dbe731be (10.0.0.0/16) | vpc-exercise-vpc

Security group rules:

Inbound Outbound

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
Custom TCP Rule	TCP	0	Custom IP

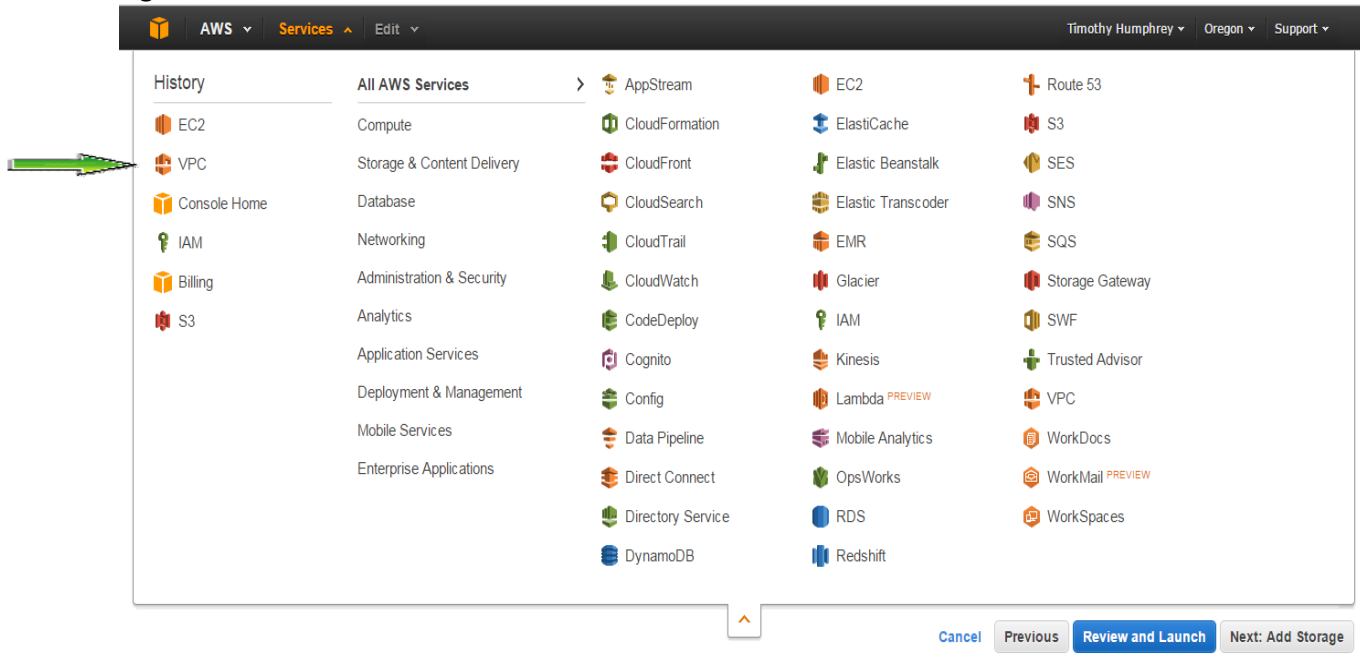
Add Rule

## Appendix C. Setting up a Virtual Private Cloud (VPC)

There are two places in the instance launching process, Task 1, that you need the same VPC – Step 3 and Step 6. In Figure 4 of Step 3 you need to select a VPC for Network. And, for Step 6 you need to select or create a security group which has the same VPC as the Network selected in Figure 4 of Step 3.

Use the aws console to setup a VPC. So, start by typing in the aws console's address into your browser, <http://console.aws.amazon.com>. Then login and click on “Services” at the top left corner of the web page. This will bring-up a web page that looks like the following screenshot. Click on VPC (pointed to by the green arrow in the following screenshot).

**Figure 20. AWS Services**



The next web page you see should look like the following screenshot. Click on “Start VPC Wizard” (pointed to by the green arrow in the next screenshot).

**Figure 21. VPC Dashboard**

**VPC Dashboard**

Filter by VPC: None

**Resources**

Note: Your Instances will launch in the US West (Oregon) region.

You are using the following Amazon VPC resources in the US West (Oregon) region:

2 VPCs	1 Internet Gateway
1 Subnet	3 Route Tables
2 Network ACLs	0 Elastic IPs
31 Security Groups	0 Running Instances
0 VPC Peering Connections	0 Customer Gateways
0 VPN Connections	0 Virtual Private Gateways

**VPN Connections**

Amazon VPC enables you to use your own isolated resources within the AWS cloud, and then connect those resources directly to your own datacenter using industry-standard encrypted IPsec VPN connections.

VPN Connections	Customer Gateways	VPC ID	Status
You do not have any VPNs.			

**Service Health**

Current Status	Details
Amazon VPC - US West (Oregon)	Service is operating normally
Amazon EC2 - US West (Oregon)	[RESOLVED] Network connectivity

[View complete service health details](#)

**Additional Information**

[VPC Documentation](#)  
[All VPC Resources](#)  
[Forums](#)  
[Report an Issue](#)

© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#) [Feedback](#)

The next web page you see should look like the following screenshot. This web page has three vertical tabs on the left and the top one, “VPC with Single Public Subnet” is selected. Click on “Select” (pointed to by the green arrow in the next screenshot).

**Figure 22. Step 1: Select a VPC Configuration**

**Step 1: Select a VPC Configuration**

**VPC with a Single Public Subnet**

Your instances run in a private, isolated section of the AWS cloud with direct access to the Internet. Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances.

**Creates:**

A /16 network with a /24 subnet. Public subnet instances use Elastic IPs or Public IPs to access the Internet.

[Select](#)

**Public Subnet**

Amazon Virtual Private Cloud

[Cancel and Exit](#)



The next web page you see should look like the following screenshot. There is only one text box you need to change, “VPC name:” (pointed to by the green arrow in the next screenshot). So, enter a name for the VPC.

**Figure 23. Step 2: VPC with a Single Public Subnet**

Step 2: VPC with a Single Public Subnet

IP CIDR block:\*  (65531 IP addresses available)

VPC name:

Public subnet:\*  (251 IP addresses available)

Availability Zone:\*

Subnet name:

You can add more subnets after AWS creates the VPC.

Enable DNS hostnames:\* ☒ Yes ☐ No

Hardware tenancy:\*

Enable ClassicLink:\* ☐ Yes ☒ No

[Cancel and Exit](#) [Back](#) [Create VPC](#)

The name I gave my VPC is “vpc-my-hpcc”. The next screenshot shows the previous screenshot with my VPC’s name in the “VPC name” text box (pointed to by the left green arrow in the next screenshot).

Once the “VPC name” text box is filled-in, click on “Create VPC” (pointed to by the right green arrow in the next screenshot).

**Figure 24. Step 2: VPC with Single Public Subnet (filled-in)**

Step 2: VPC with a Single Public Subnet

IP CIDR block:\*  (65531 IP addresses available)

VPC name:

Public subnet:\*  (251 IP addresses available)

Availability Zone:\*

Subnet name:

You can add more subnets after AWS creates the VPC.

Enable DNS hostnames:\* ☒ Yes ☐ No

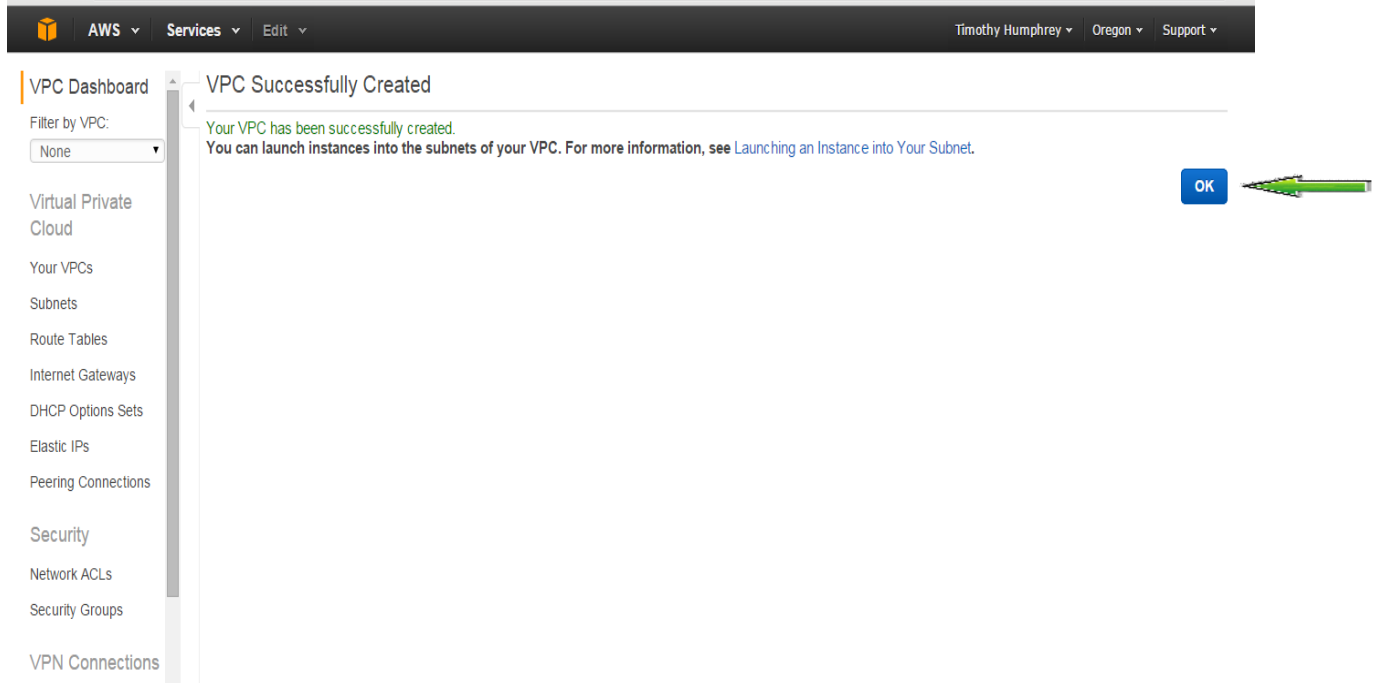
Hardware tenancy:\*

Enable ClassicLink:\* ☐ Yes ☒ No

[Cancel and Exit](#) [Back](#) [Create VPC](#)

The next web page you see should look like the following screenshot. Click on “OK” (pointed to by the green arrow in the next screenshot).

**Figure 25. VPC Successfully Created**



The next web page you see should look like the following screenshot. This is your list of VPCs. The following screenshot shows my list of VPCs. You will notice that I have two VPCs in my list. The one I just created, “vpc-my-hpcc” is the one pointed to by the green arrow in the following screenshot.

We have completed the creation of a VPC.

**Figure 26. VPC Dashboard - Your VPCs**

The screenshot displays the AWS VPC Dashboard. At the top, there is a navigation bar with the AWS logo, 'AWS' dropdown, 'Services' dropdown, 'Edit' dropdown, and user information: 'Timothy Humphrey', 'Oregon', and 'Support' dropdown. Below the navigation bar, the left sidebar shows the 'VPC Dashboard' section with a 'Filter by VPC' dropdown set to 'None'. The sidebar also lists other AWS services: 'Virtual Private Cloud', 'Subnets', 'Route Tables', 'Internet Gateways', 'DHCP Options Sets', 'Elastic IPs', and 'Peering Connections'. A green arrow points to the 'Virtual Private Cloud' link. The main content area has a 'Create VPC' button and an 'Actions' dropdown. Below this is a search bar with the text 'Search VPCs and their properties'. A pagination bar shows '<< 1 to 2 of 2 VPCs >>'. A table lists the VPCs:

<input type="checkbox"/>	Name	VPC ID	State	VPC CIDR	DHCP options set	Route table	Network ACL	Tenancy	Def
<input type="checkbox"/>	vpc-my-hpcc	vpc-19a81a7c	available	10.0.0.0/16	dopt-8ab7afe8	rtb-ed1da688	acl-6e14a80b	Default	No
<input type="checkbox"/>	vpc-exercise-vpc	vpc-dbe731be	available	10.0.0.0/16	dopt-8ab7afe8	rtb-5ea0703b	acl-e9fa2d8c	Default	No

Below the table is a horizontal scrollbar and a text prompt 'Select a VPC above'.