MASS Java AWS Setup Guide

August 31, 2016

1. Intro

This document is to help understand how to create and run AWS EC2 (Elastic Compute Cloud) instances for the purposes of running MASS Java and its applications on multiple computing nodes without having access to a distributed computing group.

2. Table of Contents

1. Intro 1
2. Table of Contents 1
3. Launching EC2 Environments 1
3.1. Selecting Image 1
3.2. Security Groups 1
3.3. Creating Key Pair 2
4. Instance Setup 2
4.1. Connecting to the Instance 2
4.2. Building MASS 2
4.3. Configuring Key 2
4.4. Configuring Your Application 3
5. Creating Slave Nodes 3
5.1. Saving Image 3
5.2. Launching Slave Nodes 3
5.3. Using Slave Nodes 3

3. Launching EC2 Environments

After logging into the AWS console select EC2 from the pull down menu at the top. The EC2 service allows for the creation of virtual computing environments in the cloud called **instances**. From the menu on the left side, select *Instances* to view your EC2 instances.

3.1. Selecting Image

After choosing to launch an instance with the *Launch Instance* button, the console will prompt you to select an image to launch. A previously saved MASS Java image can be selected from the *My AMIs* menu on the left. If you had not previously saved an image, you will need to create your own.

If you do not have a previously saved image, you can create your own image to install MASS on your instance and run your application. After choosing to launch an instance, choose an Ubuntu AMI and then choose what type of instance to launch. The *t2.micro* instance type is free and it is recommended to choose this one, but the paid ones work just as well.

3.2. Security Groups

Before you launch your instance, you will be prompted to select a Security Group. It is recommended to set the option pulldown tab as *Anywhere* so you can SSH into your instance from any IP address. Since this instance will not be open to the public, it should not cause any security concerns. For the *Type* pulldown option select *Custom TCP Rule* so you can input custom allowed ports. Port 22 is necessary for SSH connections, and MASS uses port 3400 and port 3401. If you want to use the debugger, you will also have to allow another port of your choosing. Other rules can be added to your security group later if necessary, detailed here.

3.3. Creating Key Pair

To access your *instance*, you must have a private key. Upon pressing the *Launch* button, you will be prompted to create a new key or select an existing key. For your first instance launch, you will need to create a new key pair. If you create a new key pair, you will need to download the *.pem* file and store it in a secure and easily accessible place. Without it, you will not be able to log on to any of the *instances* that you create using it.

4. Instance Setup

4.1. Connecting to the Instance

After launching, you can connect to your instance using SSH or putty. Pressing the *Connect* button near the top left of the window to find instructions on connecting. The default username for a Ubuntu instance is *ubuntu*.

4.2. Building MASS

You can transfer files to your instance using FTP or WinSCP. You can also install Git and Maven to build the most current version of MASS from the command line by using the commands below.

```
$sudo apt-get update
$sudo apt-get install default-jre
$sudo apt-get install default-jdk
$sudo apt-get install maven
$sudo apt-get install git
$git clone -b develop https://bitbucket.org/mass_library_developers/mass_java_core.git
$cd mass_java_core
$mvn package
```

This will create a MASS JAR executable in the */target* directory.

4.3. Configuring Key

After logging into your instance, you must configure the key to allow use across multiple nodes. To do this transfer a copy the *.pem* private key for this instance from your secure place to your instance. Then use the following commands to configure your key.

```
$chmod 400 YourKeyFileName.pem
$mv YourKeyFileName.pem ~/.ssh/id_rsa
$ssh-keygen -y -f ~/.ssh/id_rsa > ~/.ssh/id_rsa.pub
```

To test this, create a slave *instance* using the steps listed in section five and SSH into the new *instance* by using the following command and typing yes when prompted. The *IP_ADDRESS* field should be either the public IP address for this *instance* or the public DNS.

```
$ssh ubuntu@IP_ADDRESS
```

If you are able to successfully SSH into the remote *instance*, then it works. This key will be used by MASS to SSH into the remote nodes as well, so it is critical that it works.

4.4. Configuring Your Application

Upload your application and compile your application using *javac*. You can test that it works by running it on the single node.

5. Creating Slave Nodes

At this point, you should be able to run any MASS application, but since MASS makes use of multiple computing nodes you will need to create multiple EC2 instances of the same type as before. This can be easily done by saving your newly created instance and launching many of that saved instance.

5.1. Saving Image

Once you have configured your instance, sign back into the AWS console and go to the EC2 instances tab. Right click on your instance, and under the *Image* tab select the option to create an image. Give your image a name and save the AMI number assigned to it. This number will be in the format *ami-[0-f]**, with the [0-f]* being any number of hexadecimal numbers.

5.2. Launching Slave Nodes

To launch the slave nodes, simply follow the process for launching the original instance, only instead select the image you just saved. For the key, use the existing key from your original instance, and use the same security group.

5.3. Using Slave Nodes

Make sure you are using the version of MASS that allows for the use of the *<privatekey>* element in *nodes.xml*. To use the slave nodes, add their IP or private DNS to your *nodes.xml*. For the *<username>* element, use *ubuntu*, and use *.ssh/id_rsa* for the *<privatekey>* element.

It is suggested to not leave your *instances* running when you are finished, so right click on the *instance*, and under the *Instance State* tab select *Stop*. Be careful when starting up the instance up again, the IP address and public DNS will change, so make sure you reflect that when logging in to the *instance* and in the *nodes.xml* file. You can allocate static IP addresses by selecting the *Elastic IPs* tab on the left side and associating it with your instance, though this is not recommended because your account will be charged every hour the instance is not running.