This procedure is used to setup an Amazon Web Service (AWS) Elastic Compute Cloud (EC2) Linux instance that is configured to run the META application (at least as much of it that can run on Linux). It is assumed that the reader has an account on <https://meta-arrow.signin.aws.amazon.com/console>.

**Setup security group: Open up Instance TCP ports for connection**

In the EC2 tab of the AWS site, select the Security Groups link located in the left pane.

You can create a new security group by pressing the create security button and complete the procedure by filling in the name, description. Keep the default for the VPC, which is “no VPC”.

Select the Default security group or the new security group that was created. The details should display below

Select the Inbound tab

Enter a Custom TCP Rule for port 22 (SSH), 80 (HTTP), 5900, 5901 (VNC). For each port, enter the port number in the Port Range field, leave the Source as 0.0.0.0/0 for each and select Add Rule.

After entering the TCP rules, select Apply Rule Changes.

**Create an EC2 Instance**

In the EC2 tab of the AWS site, select the Instances link located in the left pane

Select the Launch Instance button

In the Quick Start tab, select the RHEL 6.1 64 bit, ami-5e837b37 image

In the Instance Type drop down, select Large (m1.large), then select Continue

In the Availability Zone, select Zone: us-east-1c and then select Continue

In the next window (Advanced Instance Options), select Continue

In the next window (Add tags to your instance), select Continue

The next window is Key Pairs window. If you haven’t created a Key Pair in the past, you either need to create one or choose an existing one from the drop down list and get the actual key file from someone else.

To create a Key Pair, select the associated radio button

Enter a key name (e.g., your username)

Select Create and Download your key

Store the resulting key file (e.g., username.pem) to known location (e.g., c:\keys\meta)

In the next window (Security Groups), select default groups or the new security group, and select create.

In the next window (Review Settings), select Launch

**Create and attach an EBS volume (i.e. hard drive space) to Instance**

In the EC2 tab of the AWS site, select the Volumes link located in the left pane

Select the Create Volume button in the EBS Volumes pane

In the Create Volume window, enter a Size: 80 GB and Availability Zone: us-east-1c

Select Yes, Create. The new volume should show in the list of volumes

Right Click on the new volume and select Attach Volume

In the Attach Volume window select the Instance ID of the instance that you want to connect to

Enter /dev/sdh for the device, then select Yes, Attach

If an error displays indicating an Invalid value for unix Device, increment the last letter of the device name (e.g., /dev/sdi) and try again. Repeat incrementing the letter until it accepts it.

**Configure SSH connections on Client computer**

In a Cygwin window, cd to /home/<username>/.ssh

Create a config file under .ssh

Add the following lines to the config file. The Public DNS is provided in the EC2 Instance details (e.g., ec2-50-17-74-76.compute-1.amazonaws.com). The LocalForward line allows for VNC connection.

Host ROOT

User root

HostName <Instance’s Public DNS>

LocalForward 5901 localhost:5901

Navigate to key folder (e.g., cd /cygdrive/c/keys/meta)

Verify ability to connect via ssh using the config file (e.g., ssh -i <key name>.pem ROOT)

**Configure Putty connection on Client computer (Optional)**

On client computer, start application PuTTYgen.

Select the Load button

In the Load Private Key window, set Files of type to All Files

Navigate to the location of the key file, select it, then select the Open button

Select OK in the pop-up message

In the PuTTY Key Generator window, select the Save private key button

Select Yes in the pop-up message

Type in the name of the key file. The .ppk extension will be added automatically

Launch PuTTY

Set the Session Hostname to root@<public DNS> (e.g., root@ec2-50-16-100-24.compute-1.amazonaws.com)

Set the Port to 22

Under Connection->SSH->Authority, use the Browse button to set the Private key file path

Save the configuration to something recognizable like Meta Cloud Computer

Select Open to establish the connection

**Install necessary OS packages**

ssh into the Instance as root

Run the “yum list” command.

If you notice notice [Errno 14] Peer cert cannot be verified or peer cert invalid errors, or errors that stop the yum list from listing servers, then you may apply the workaround that AWS has provided (septempter 21, 2011). Use the update that applies to redhat 6.x. The list of relevant updates are provided below:

For customers in US-East:  
RHEL5: rpm -Uhv <http://redhat-clientconfig-us-east-1.s3.amazonaws.com/rh-amazon-rhui-client-2.2.16-1.el5.noarch.rpm>  
RHEL6: rpm -Uhv <http://redhat-clientconfig-us-east-1.s3.amazonaws.com/rh-amazon-rhui-client-2.2.16-1.el6.noarch.rpm>  
  
For customers in US-West:  
RHEL5: rpm -Uhv <http://redhat-clientconfig-us-west-1.s3.amazonaws.com/rh-amazon-rhui-client-2.2.16-1.el5.noarch.rpm>  
RHEL6: rpm -Uhv <http://redhat-clientconfig-us-west-1.s3.amazonaws.com/rh-amazon-rhui-client-2.2.16-1.el6.noarch.rpm>  
  
For customers in EU-West:  
RHEL5: rpm -Uhv <http://redhat-clientconfig-eu-west-1.s3.amazonaws.com/rh-amazon-rhui-client-2.2.16-1.el5.noarch.rpm>  
RHEL6: rpm -Uhv <http://redhat-clientconfig-eu-west-1.s3.amazonaws.com/rh-amazon-rhui-client-2.2.16-1.el6.noarch.rpm>  
  
For customers in AP-Northeast:  
RHEL5: rpm -Uhv <http://redhat-clientconfig-ap-northeast-1.s3.amazonaws.com/rh-amazon-rhui-client-2.2.16-1.el5.noarch.rpm>  
RHEL6: rpm -Uhv <http://redhat-clientconfig-ap-northeast-1.s3.amazonaws.com/rh-amazon-rhui-client-2.2.16-1.el6.noarch.rpm>  
  
For customers in AP-Southeast:  
RHEL5: rpm -Uhv <http://redhat-clientconfig-ap-southeast-1.s3.amazonaws.com/rh-amazon-rhui-client-2.2.16-1.el5.noarch.rpm>  
RHEL6: rpm -Uhv <http://redhat-clientconfig-ap-southeast-1.s3.amazonaws.com/rh-amazon-rhui-client-2.2.16-1.el6.noarch.rpm>

Run yum clean all.

Install the following packages using yum (e.g., yum install tigervnc-server). Install the latest available versions. The “yum list” command lists all available packages and versions. The versions of packages loaded on a prior instance are provided for reference.

tigervnc-server x86\_64 1.0.90-0.15.20110314svn4359.el6\_1.1

xorg-x11-xinit x86\_64 1.0.9-13.el6

gnome-desktop x86\_64 2.28.2-8.el6

nautilus x86\_64 2.28.4-15.el6

gnome-session x86\_64 2.28.0-15.el6

gnome-panel x86\_64 2.30.2-14.el6

firefox x86\_64 3.6.18-1.el6\_1

gcc-c++ x86\_64 4.4.5-6.e16

libX11-devel x86\_64 1.3-2.e16

openssl-devel x86\_64 1.0.0-10.e16\_1.4

system-config-firewall x86\_64 1.2.27-3.el6\_1.3

these packages can be installed in the following order

1. yum install Xorg
2. yum install gcc-c++
3. yum install libX11-devel
4. yum install openssl-devel
5. yum install gnome-panel
6. yum install gnome-desktop
7. yum install nautilus
8. yum install firefox
9. yum install tigervnc-server

reboot the server

**Create a user account**

ssh into the Instance as root

Enter adduser <username>

Enter passwd <username>. Enter password as directed. Ignore message complaining about simplicity of password

Copy the .ssh directory under /root to /home/<username>

Change ownership of the copied directory to the new user (e.g., chown –R <username> /home/<username>/.ssh

Change the permission of the folder with, chmod 700 .ssh

In a Cygwin window on the client computer, cd to /home/<username>/.ssh

Edit the config file under .ssh

Copy and paste the existing lines to the end of the file. Then change Host to USER and User to <username>

Add the line LocalForward 8080 localhost:8080

Add the line RemoteForward 10000 boundry.alphatech.com:11011

The config file should now look something like the following

Host ROOT

User root

HostName <Instance’s Public DNS>

LocalForward 5901 localhost:5901

Host USER

User <username>

HostName <Instance’s Public DNS>

LocalForward 5901 localhost:5901

LocalForward 8080 localhost:8080

RemoteForward 10000 boundry.alphatech.com:11011

Navigate to key folder (e.g., cd /cygdrive/c/keys/meta)

Verify ability to connect via ssh (e.g., ssh -i <key name>.pem USER)

**Configure Virtual Network Connection (VNC)**

*Note, this section assumes that VNC server is not already running on the Instance. You can check using the command ps –elf | grep vnc*

ssh into the Instance as <username>

Enter vncserver –geometry 1400x1000

Note: Instances of the vncserver that are not needed, can be removed with the following command if desired

Vncserver –kill :x, where x is the display number

On client computer, startup VNC Viewer

In the Connection Details window, enter localhost:1 for Server , then select OK

In the Authentication window, enter the password for <username>

The VNC desktop window should now be displayed

If the vnc client is unable to authenticate the user, reset the vncpassword on the server (logged in as the <username>) with the following command

vncpasswd

[kwf@ip-10-2-30-142 ~]$ vncpasswd

Password:

Verify:

On the VNC desktop, right-click the background and select Create Launcher

In the Create Launcher window, enter xterm for Name and xterm for Command, then select OK. An xterm icon should now be displayed on the desktop

Double left-click the xterm icon, verify that an XTerm displays

**Mount the EBS volume to /mnt/data-store**

Log into the Instance as root

Navigate to /dev (cd /dev) and list the directory contents (ls)

The items beginning with the letters xvd (e.g., xvde1, xvdj, xvdk,xvdl) are the block devices. The recently attached EBS volume is the last one in the series (e.g., xvdl). *Note, this is not well documented in the EC2 UG or the Forum. This was discovered by un-attaching the volume and noticing with block device disappeared from /dev.*

Make a file system on the volume (e.g., mkfs –T –ext3 /dev/xvdl)

Make the mount directory (e.g., mkdir /mnt/data-store)

Mount the device (e.g., mount /dev/xvdl /mnt/data-store)

You can verify this mount by inspecting the /etc/mtab file. There will be an entry for the /dev/xvdl block device.

Note, after stopping and restarting the Instance, you will need to remount the device

/dev/fstab may optionally be used to statically map the filesystem for automount.

**Install development tools**

The tools are available for download from the Meta Deployment shared folder on Karl Fischer’s laptop gldlkwf1. You can map a local drive to gldlkwf1\Meta Deployment for easy access. All tools but Magic Draw (MD) can alternatively be downloaded from the Web. To copy a file from a client computer to the Instance (or back), you can use Secure Copy (SCP) or Putty Secure Copy (PSCP). To do so, it is convenient to navigate to the keys directory (e.g., cd /cygdrive/c/keys/meta or c:\keys\meta). The syntax for each command will be as follows:

scp –i <key file>.pem <source file path> <username>@<Instance Public DNS>:<destination file path> (e.g., scp –I meta.pem hello.txt root@ec2-50-17-171-124.comput-1.amazonaws.com:/home/kwf)

pscp –i <key file>.ppk <source file path> <username>@<Instance Public DNS>:<destination file path> (e.g., pscp –I meta.ppk hello.txt root@ec2-50-17-171-124.comput-1.amazonaws.com:/home/kwf)

Copy the development tool installation packages to the Instance as described above

Log into the Instance as root

Install the following development tools using the actions specified for each tool.

jdk-6u22-linux-x64.bin - move the installation file to /opt. make the file executable via “chmod 700”, then execute it via “./jdk-6u22-linux-x64.bin.

clisp-2.4.4.tar.gz – uncompress to /opt (e.g., tar –xzvf clisp-2.4.4.tar.gz)

apache-maven-3.0.3.tar.gz – uncompress to /opt

apache-ant-1.8.2.tar.gz - uncompress to /opt

Special instruction to install Magic Draw

The installation must either be done by vnc’ing to the server and then run the install procedure on linux,

or

You can start Xwindow server from your cygwin on the client laptop with the following command: startxwin

run the following ssh command: ssh -i <key name>.pem ROOT

and then run the install procedure on linux.

MD\_UML\_170\_sp1\_unix.sh - make installation script executable (chmod 700) and execute it. This is an Install Anywhere script which includes the archive. When prompted, change the installation directory from /root/MagicDraw\_UML to /opt/MagicDraw. Open up privileges on /opt/MagicDraw directory so that plug-ins can be installed by the application when run by a user.

Special instruction to install Springsource tools

The installation must either be done by vnc’ing to the server and then run the install procedure for linux ,

or

You can start Xwindow server from your cygwin on the client laptop with the following command: startxwin

run the following ssh command:

ssh -i <key name>.pem ROOT

set the X11 DISPLAY variable: export DISPLAY=:0.0

and then run the install procedure on linux.

springsource-tool-suite-2.6.1.RELEASE-e3.6.2-linux-gtk-x86\_64-installer.sh - make installation script executable (chmod 700) and execute it. Install it to /opt/springsource, when it asks for the installation path. Select default packages.

When prompted for the JDK path, select /opt/jdk1.6.0\_22.

Open up privileges on /opt/springsource directory so that subdirectories can be created by the application when run by a user account.

springsource-tool-suite-2.6.1.RELEASE-e3.6.2-linux-gtk-x86\_64-installer.sh - make installation script executable (chmod 700) and execute it. Install it to /opt/springsource. When prompted for the JDK path, select /opt/jdk1.6.0\_22. Open up privileges on /opt/springsource directory so that subdirectories can be created by the application when run by a user account.

Note that we need the following tool for the initial setup of windows instance.

rdesktop-1.7.0.tar.gz – uncompress to /opt. Navigate to /opt/rdesktop-1.7.0. Build the source code and install the executable using the commands “./configure”, “make”, “make install”.

Add the following soft links to /usr/local/bin using command ln –s <target> <link name>

java -> /opt/jdk1.6.0\_22/bin/java

ant -> /opt/apache-ant-1.8.3/bin/ant

mvn -> /opt/apache-maven-3.0.3/bin/mvn

sts -> /opt/springsource/sts-2.6.1.RELEASE/STS

*Note, a MagicDraw link was not added since it needs to be executed in directory /opt/MagicDraw using the command ./bin/mduml. Make sure mduml is executable (chmod 744)*

**Configure the Firewall to accept connections on ports 80 and 8080**

Log into the Instance as root

Enter system-config-firewall

In the Trusted Services window, check the WWW (HTTP) box

In the Other Ports window, select the Add button

In the Port and Protocol pop-up window, scroll down and select Port 8080 TCP webcache, and then select OK

Select File->Apply, and then exit out of the application

**Configure and Run ARRoW and Magic Draw**

Establish a VNC connection to the instance and login as <username>

Set environmental variable JAVA\_HOME to /opt/jdk1.6.0\_22 in .bashrc (e.g., export JAVA\_HOME=/opt/jdk1.6.0\_22)

Set environmental variable MDUML\_HOME to /opt/MagicDraw in .bashrc

Check that /usr/local/bin is first in your user account PATH by using the command “env | grep PATH”. If it’s not, add “export PATH=/usr/local/bin:$PATH” to your .bashrc file.

Navigate to /mnt/data-store

perform the following commands as su

Checkout the META trunk to a /mnt/data-store (e.g., svn checkout svn+ssh://<username>@cvsext.ait.na.baesystems.com/proj/meta/svn/trunk)

Apply the <username> ownerships to the trunk directory with the following command:

chown –R <username> trunk

chgrp –R <username> trunk

Navigate to trunk/arrow-mvn-init

Enter mvn install

Navigate to trunk/ArrowManualArtifacts

Enter mvn install

Navigate to trunk/arrow-mvn-all

Enter mvn install

Navigate to trunk/mdplugin

Enter mvn install

Navigate to trunk/ArrowWebServices

Enter mvn tomcat:run to run on port 8080. If you want to run on port 80, add the parameter “-Dmaven.tomcat.port=80”

Start the FireFox web browser by clicking on the Web Browser icon on the menu bar of the VNC desktop

Enter the address <http://localhost:8080/ArrowWebServices/index.html>

The ARRoW menu window should display

Navigate to /opt/MagicDraw

Enter ./bin/mduml

Magic Draw should startup

Follow Magic Draw directions to connect to the server and download any missing plug-ins

Re-start Magic Draw after installing plug-ins

**Configure and Run the Component Model Library (CML)**

Establish a VNC connection to the instance and login as <username>

Navigate to /trunk/Component ModelLibrary/CMLServer

Unzip artifactory-2.3.3.2

Copy artifactory.config.orig.xml to ./artifactory-2.3.3.1/etc/artifactory.config.xml

Copy jetty.xml to ./artifactory-2.3.3.1/etc/jetty.xml

Create directory (i.e., mkdir) ./artifactory-2.3.3.1/logs

Login as root (e.g., su root) and start Artifactory using the command ./start.sh

Log back in as <username> and navigate to /trunk/models/Armor

Enter mvn –gs cml\_settings.xml deploy

Start the FireFox web browser by clicking on the Web Browser icon on the menu bar of the VNC desktop

Enter the address <http://localhost:8888/artifactory/webapp/browserepo.html>

The Artifactory browser should display and the user should be able to locate the Armor artifact

**Create an Elastic IP address and associate it with an instance**

In the EC2 tab of the AWS site, select the Elastic IPs link located in the left pane

Select the Allocate New Address button. The Allocate New Address pop-up window should display with EC2 selected in the drop-down menu

Select Yes, Allocate. A new IP address line item will display in the main window

*Note, the following steps require that an EC2 instance is running*

Right-click the new line item and select Associate. The Associate Address pop-up window should display

Select an instance from the Select an instance drop-down menu, and then select Yes, Associate. The newly associated instance ID should display in the IP address line item

**Create an Amazon Machine Image (AMI)**

*Note, the following steps require that an EC2 instance exists. I’m not sure if it needs to be running*

In the EC2 tab of the AWS site, select the Instances link located in the left pane.

Right-click the desired instance and select Create Image (EBS AMI). The Create Image window should display

Give the image a meaningful name (e.g., linux-karl) and optional description, and then select Create This Image

The image creation process takes several minutes. The instance is stopped,

snapshots of the disk volumes are created, and then the instance is restarted

In the EC2 tab of the AWS site, select the AMIs link located in the left panel. A line item for the new image should be displayed

To start an instance from the new AMI, right-click the AMI and select Launch Instance

Note: you may need to reset the vncpasswd on a successful ssh login as user.

*Note, it’s very important to keep track of the items associated with an AMI and instance so that they are not accidentally deleted. For sanity’s sake, it may be worth keeping a record of the the items associated with a given instance and AMI including: AMI ID, Security Group ID, Snapshot IDs, and Volume IDs. An example follows:*

*AMI ID: ami-8d9457e4, Security group: sg-a94073c0,*

*Snapshots: snap-32851a52 35 GiB, snap-36851a56 80 GiB*

*Volumes: vol-a29c19c8 35 GiB, vol a09c19ca 80 GiB*