**Individual Users on EC2 with dedicated PPKs.**

Say you are creating user "user1", Create a key on your own machine by entering the following:

# ssh-keygen -b 1024 -f user1 -t dsa

Don't use a paraphrase -- just hit enter. You should now have two files compiled: user1 and user1.pub

# chmod 600 user1.pub

Now transfer the public key file (user1.pub) from your computer to the server. In this example, I use the /tmp/ directory of destination Server

Now SSH into your destination server using an account with root access, you will now need to create the user and also create the necessary files and ownership for you to use the key you just created:

# sudo su (if needed)

# useradd -c "firstname lastname" user1

# cd /home/user1

# mkdir .ssh

# chmod 700 .ssh

# chown user1:user1 .ssh

# cat /tmp/user1.pub >> .ssh/authorized\_keys

# chmod 600 .ssh/authorized\_keys

# chown user1:user1 .ssh/authorized\_keys

Once you've done this, exit out back to your own machine, then try to SSH using the new credential and user account you've created:

Rename the user1 file to user1.pem

# mv user1 user1.pem

Change the permissions to pem file

#chmod 400 user1.pem

Now login to Destination Server

# ssh -i user1.pem [user1@xx.XX.XX.XX](mailto:user1@xx.XX.XX.XX)

You should be in! ☺ ☺ ☺

**Installing Tomcat on AWS EC2 Linux Instance**.

1. Launch Ec2 Linux Instance.

2. Create a new keypair or use existing keypair

3. Open the Port 80 & 8080 on your security groups where EC2 is running.

4. Login to Linux Instance: For Example ApacheServer01

#ssh -i keypair.pem ec2-user@XX.XX.XX.XX

Become sudo if you need

#sudo su

You need to download the java before tomcat installation. Created a directory, in our scenario, we have attached & mounted encrypted EBS volume

#cd /Data

Download the java (jdk-8u111-linux-x64.tar.gz)

#wget http://download.oracle.com/otn-pub/java/jdk/7u40-b43/jdk-8u111-linux-x64.tar.gz

Once downloaded we need to unpack it using tar: tar zxpvf jsdk-8u111-linux-x64.tar.gz. This creates the directory jdk1.8.0\_111 in which Java is installed.

Now we set the Java Home and will put Java into the path of our users using the following commands:

For Users : add below to .bash\_profile

For All Users : Add below to /etc/profile

#vi /etc/profile

...

...

***JAVA\_HOME=/Data/jdk1.8.0\_111/***

***export JAVA\_HOME***

***PATH=$JAVA\_HOME/bin:$PATH***

***export PATH***

Save & quit the file.

#

Restart the /etc/profile

#***source /etc/profile***

If no errors then you are good to go.

You can verify that Java is installed by doing: ***java -version*** this should print out java version “***1.8.0\_111***”.

Download, install and configure Tomcat 7

We are going to download and install Tomcat 7 in the /Data directory by running the following two commands:

*#wget http://ftp.cixug.es/apache/tomcat/tomcat-7/7.0.72/bin/apache-tomcat-7.0.72.tar.gz*

*#tar zxpvf apache-tomcat-7.0.72.tar.gz*

Tomcat will be unpacked and installed in the apache-tomcat-7.0.72 directory. We will now finish by configuring the Tomcat users and setting tomcat to launch automatically whenever the server restarted.

To configure Tomcat to launch automatically create a file called "tomcat" in the directory /etc/rc.d/init.d/ with the following contents:

#vi /etc/rc.d/init.d/tomcat

***!/bin/sh***

***# Tomcat init script for Linux.***

***#***

***# chkconfig: 2345 96 14***

***# description: The Apache Tomcat servlet/JSP container.***

***JAVA\_HOME=/Data/jdk1.8.0\_111/***

***CATALINA\_HOME=/Data/apache-tomcat-7.0.72***

***export JAVA\_HOME CATALINA\_HOME***

***exec $CATALINA\_HOME/bin/catalina.sh $\****

#

Next, execute the following commands to set the proper permissions for your init script and enable Tomcat for auto-launch:

#chmod 755 /etc/rc.d/init.d/tomcat

#chkconfig --level 2345 tomcat on

#/etc/init.d/tomcat start

Tomcat should now be automatically launched whenever your server restarts.

To deploy war files - copy your war files to ***/Data/apache-tomcat-7.0.72/webapps***

Once the instance has rebooted, go to your browser and enter the public DNS of your instance followed by the port 8080. It should look something like this: ***10.1.6.165:8080***. You will see the Tomcat Server Home Page.

**Setting up Subversion on Amazon EC2**

Create an instance (Linux EC2)

Add Encrypted EBS VOlume

Open 80, 3690(TCP&UDP) ports on security groups where SVN is running

Once the instance is ready , login to instance with your keypair

Software Installation

*#yum update -y*

Visit the public/private ip in your browser: http://YOUR\_INSTANCE\_IP, should see Amazon Linux AMI Test Page if Apache is installed and running

If apache not installed, install apache also.

Install php modules & httpd

***#yum install httpd-\* php-mysql\****

start the httpd service

***#service httpd restart***

Install subversion and mod\_dav\_svn (should see a long list of all changes):

***# sudo yum –y install mod\_dav\_svn***

***# sudo yum –y install subversion***

Edit the Apache configuration file for subversion:

***# sudo vi /etc/httpd/conf.d/subversion.conf***

Replace/add subversion.conf content with:

***LoadModule dav\_svn\_module modules/mod\_dav\_svn.so***

***LoadModule authz\_svn\_module modules/mod\_authz\_svn.so***

***<Location /repos>***

***DAV svn***

***SVNParentPath /ProtoRepos/svn***

***# Limit write permission to list of valid users.***

***AuthType Basic***

***AuthName "Authorization Realm"***

***AuthUserFile /ProtoRepos/svn-auth/passwd***

***AuthzSVNAccessFile //ProtoRepos/svn-auth/access***

***Require valid-user***

***</Location>***

#

Create the directory which will contain the subversion repository:

***#mkdir /ProtoRepos/svn***

Create the directory which will contain the permissions files.

***#mkdir /ProtoRepos/svn-auth***

Create the permission file:

**#vi /ProtoRepos/svn-auth/access**

*[/]*

*jpersson = rw*

*madevang = rw*

*rajasekarj = rw*

*jonese = rw*

*...*

#

Create and add to the password file (use -c the first time to create)

***#htpasswd -cb /ProtoRepos/svn-auth/passwd jpersson jpersson***

Repeat the same for other users

Create a repository (REPONAME is the name of your repository eg. pspproto)

***#cd /ProtoRepos/svn***

***#svnadmin create pspproto***

Change files authorization (again after creating new repos too):

***#chown -R apache.apache /ProtoRepos/svn /ProtoRepos/svn-auth***

***#chmod 600 /ProtoRepos/svn-auth/access /ProtoRepos/svn-auth/passwd***

Start apache web server:

***# service httpd restart***

To make sure apache always starts on boot:

***# chkconfig httpd on***

Verify the subversion repo by opening in a browser:

***http://10.1.6.162/repos/pspproto***

You’re done! Connect via Tortoise or your own SVN client using the URL above.

**THANK YOU**