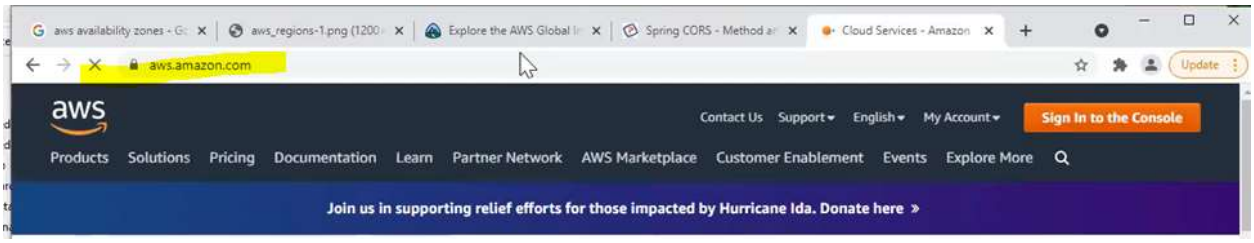
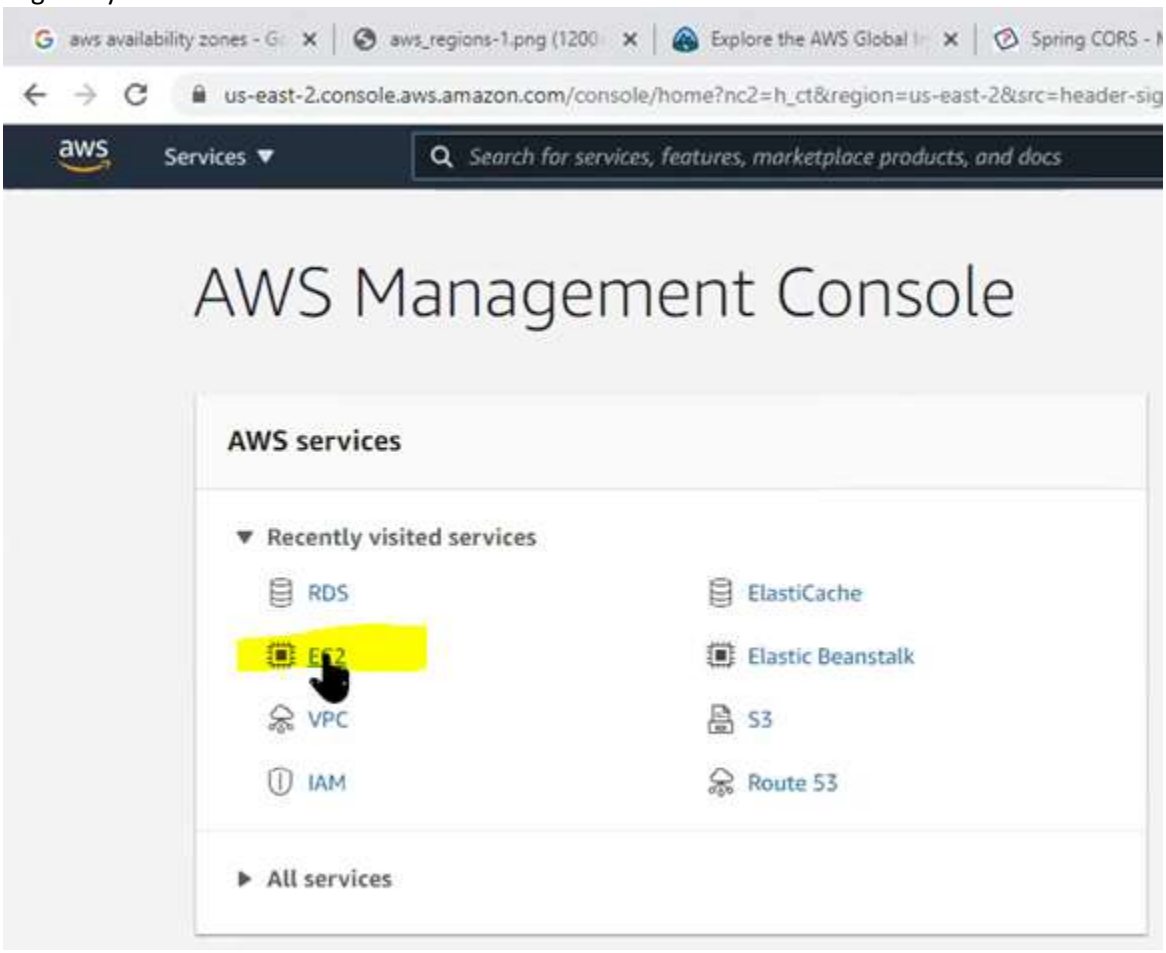
This document walks through the steps involved to build a working EC2 instance on AWS

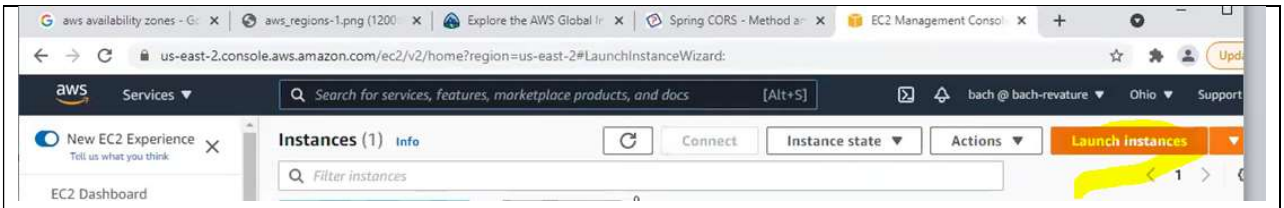
- Build an EC2 instance
- Install Git, Java, Maven, and Tomcat
- Update the java and javac version
- Deploy war file on AWS



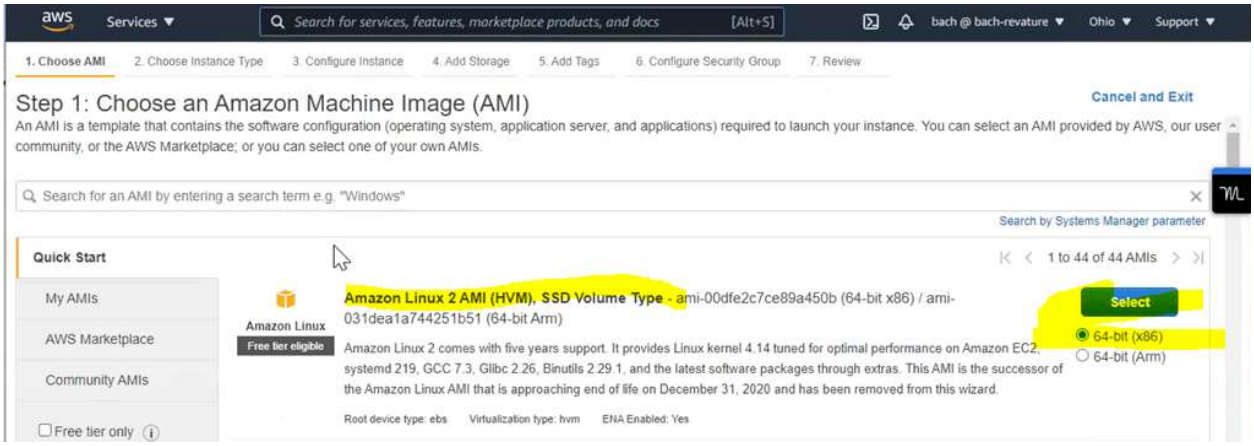
Log into you AWS account the find and select EC2

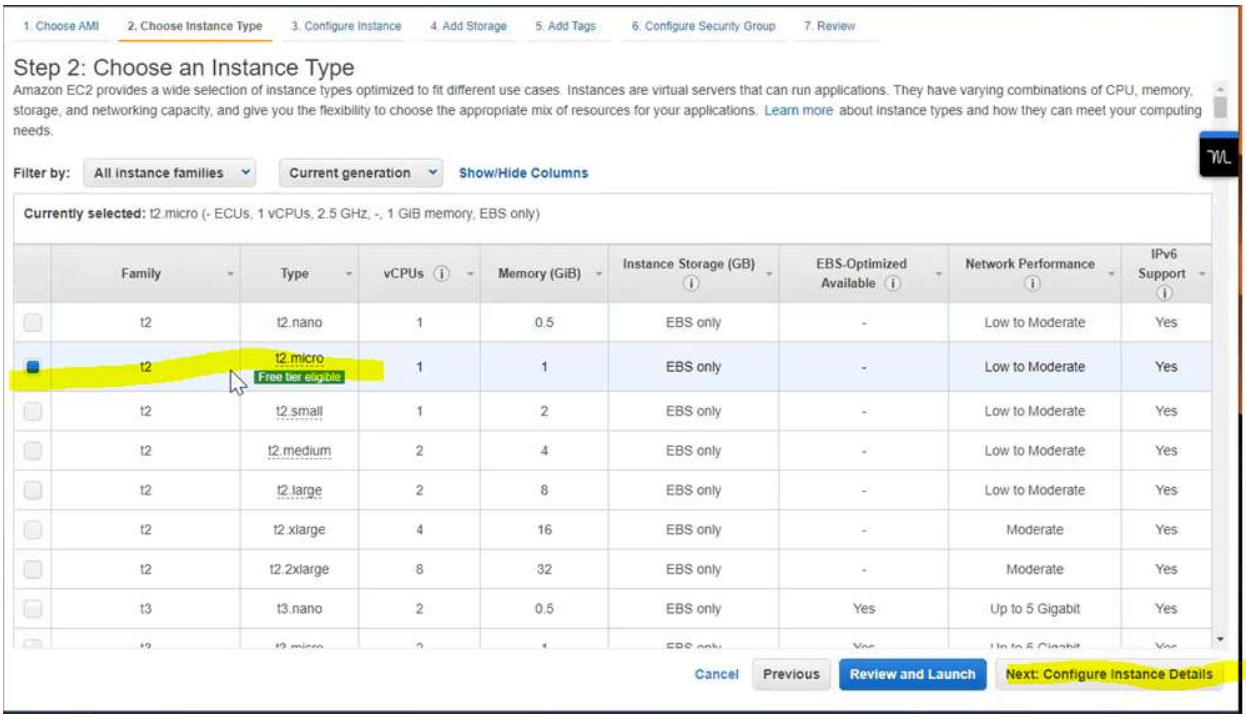


Launch instance

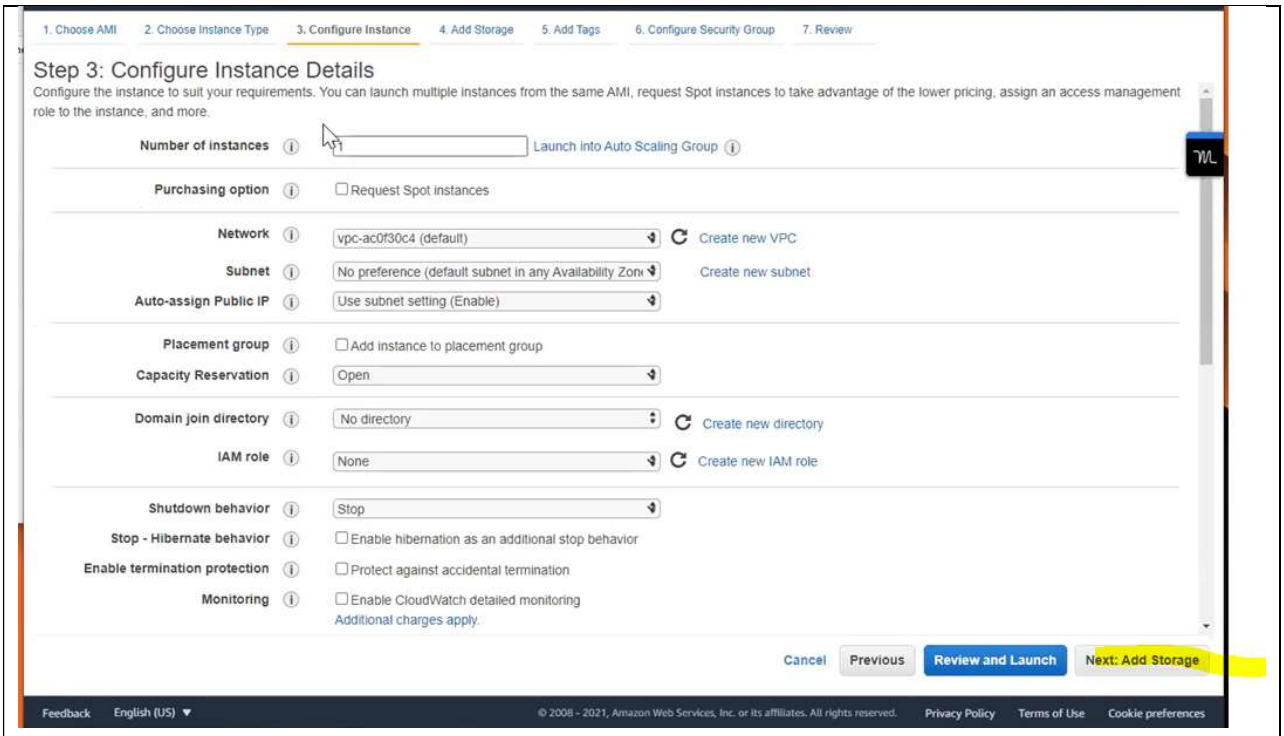


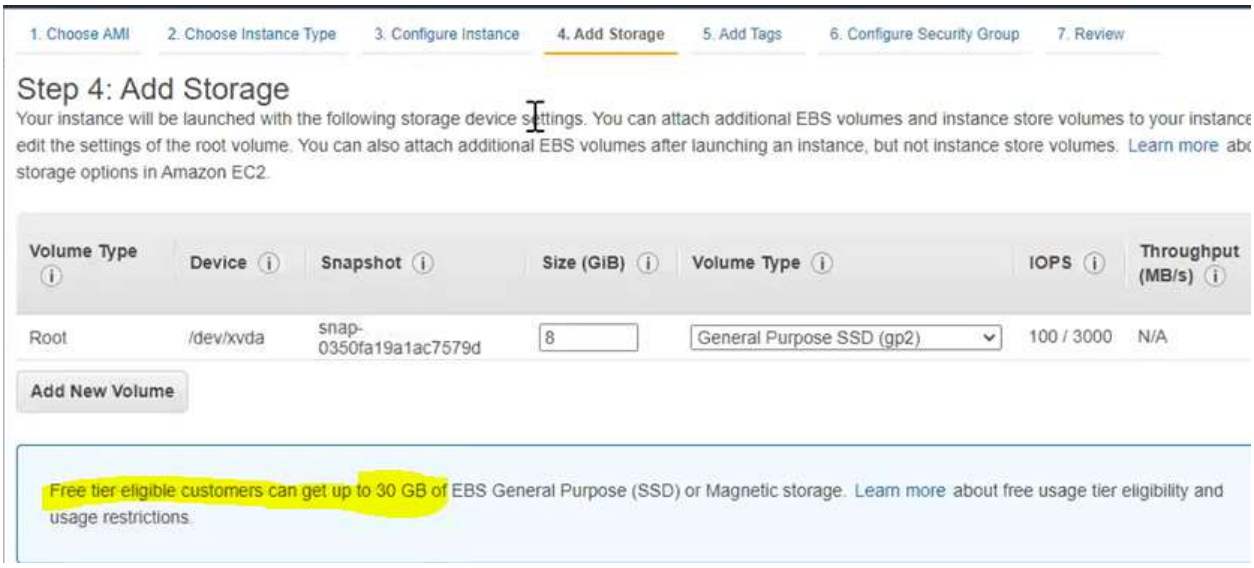
This button may be in a different location depending if you already have created an instance.



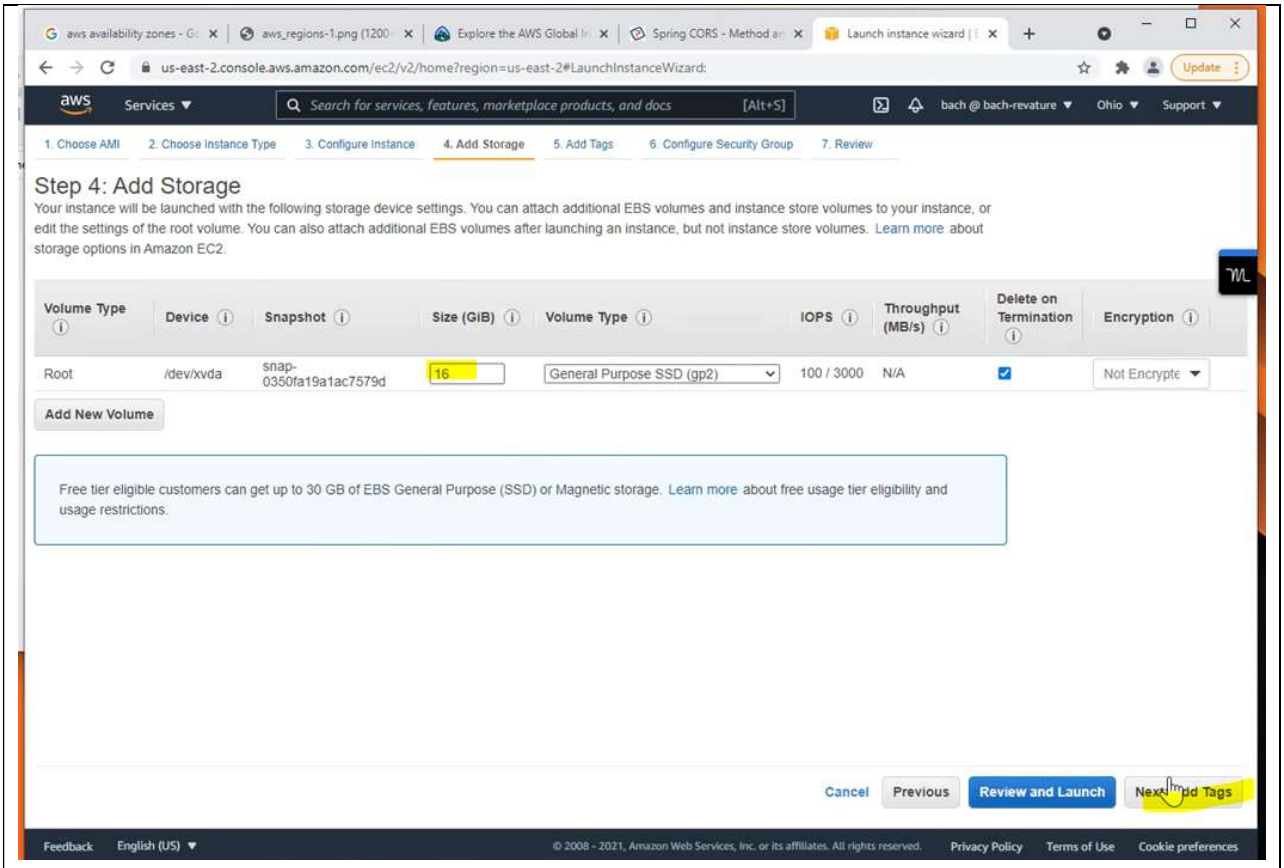


No changes accept defaults

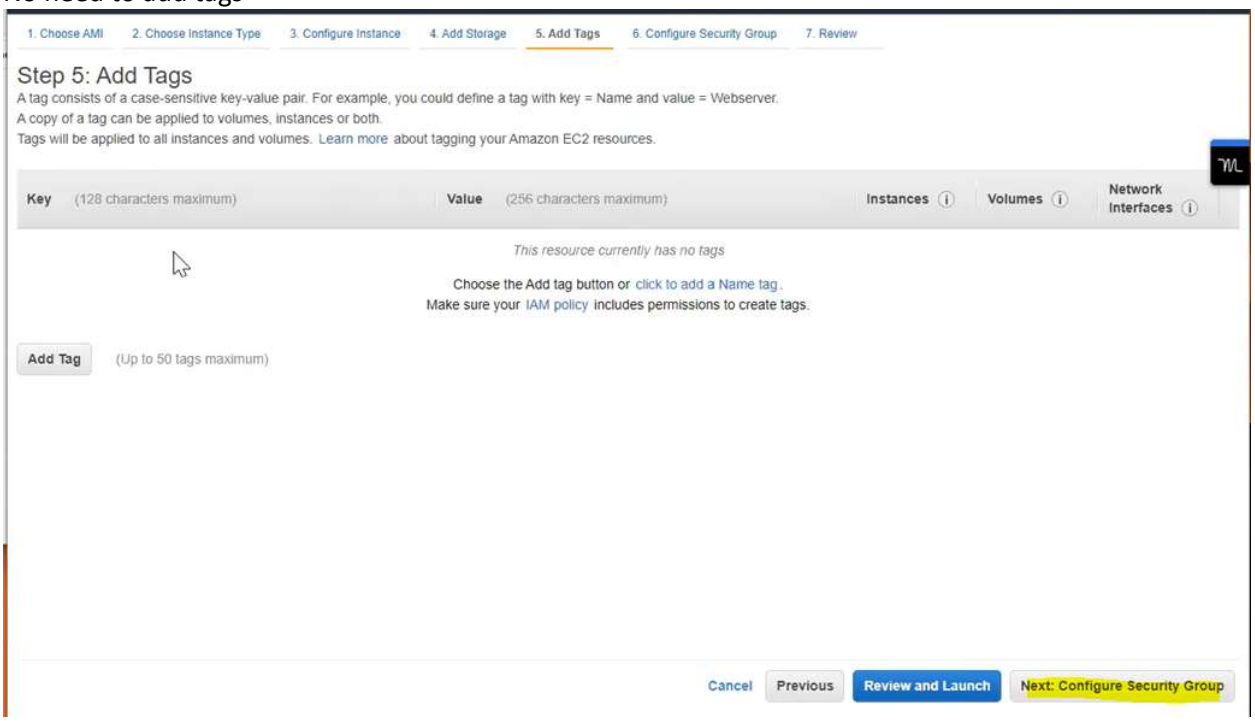




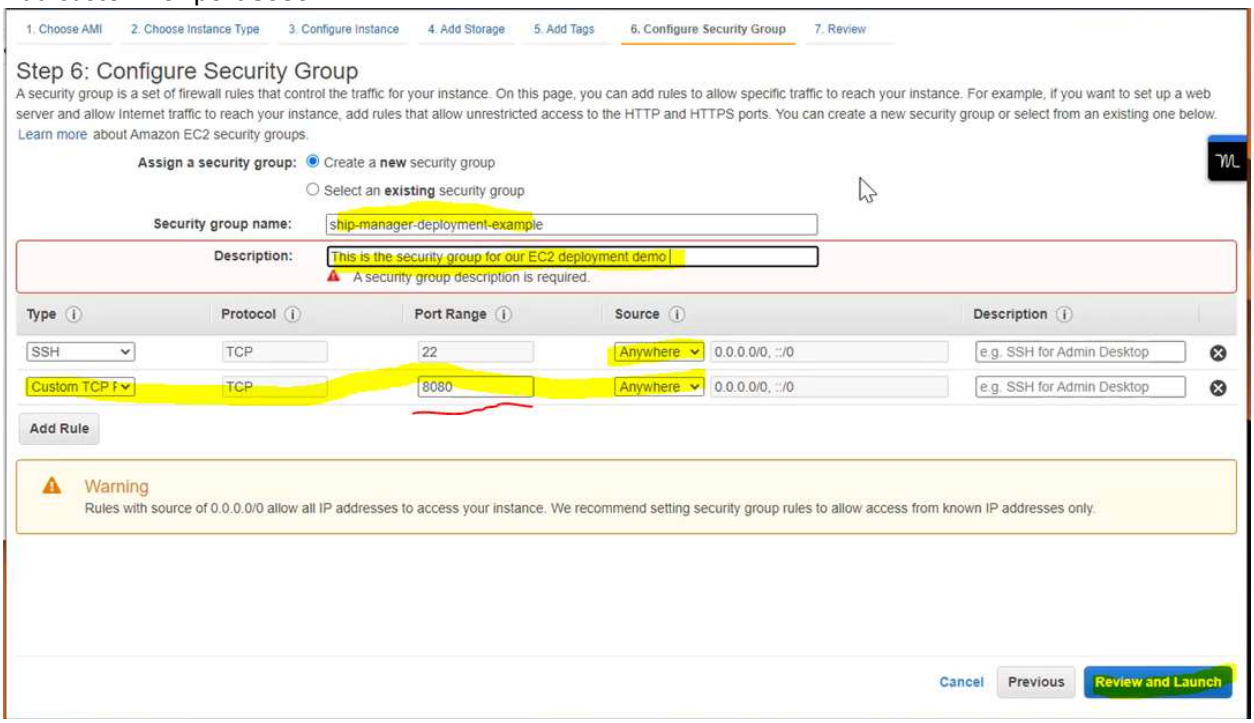
Change storage if you want



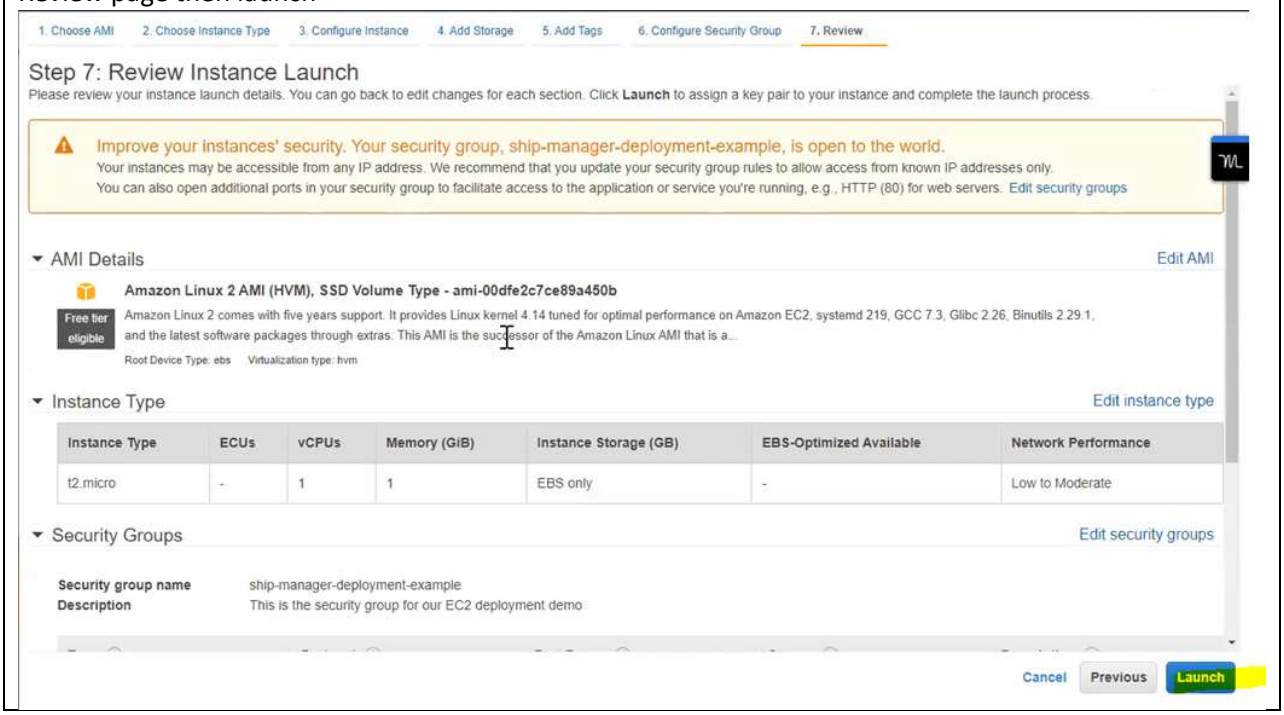
No need to add tags



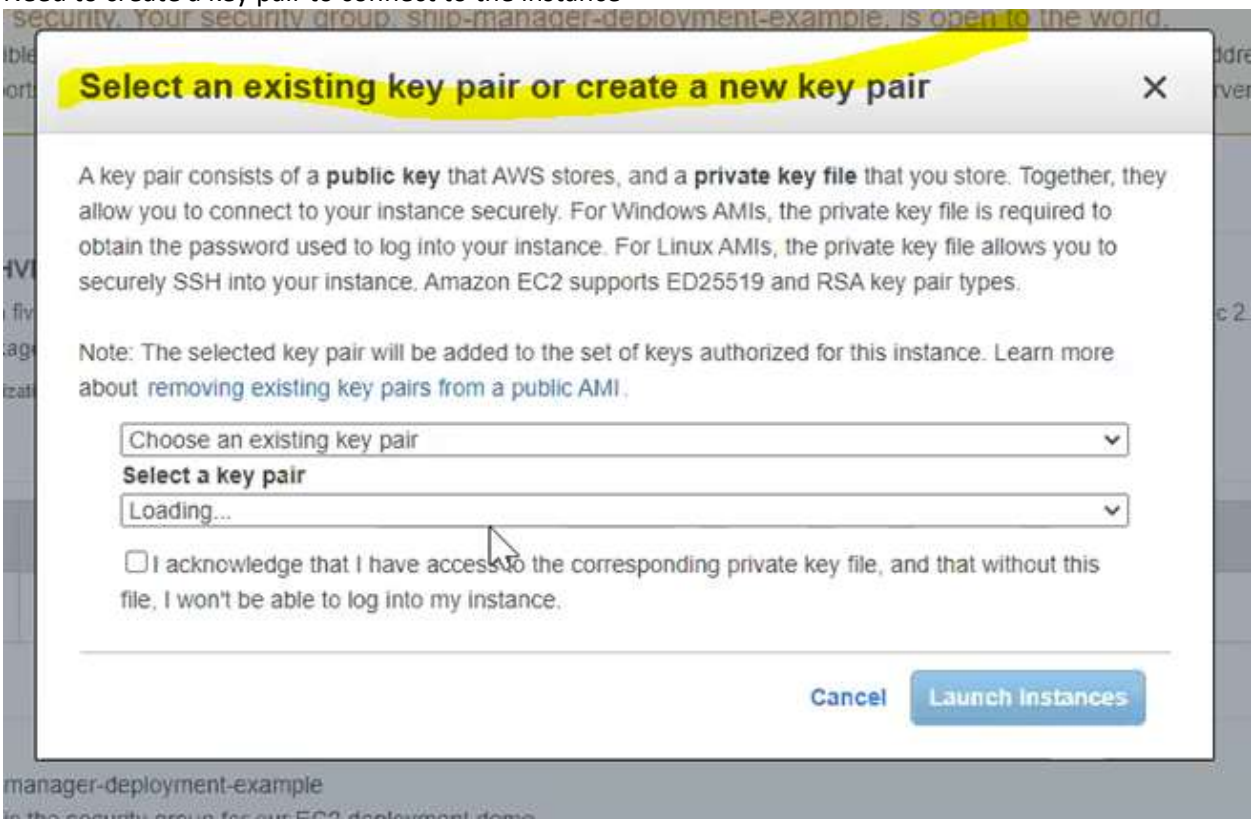
Configure security group making changes highlighted below.
Give some name and description.
Change port 22 to anywhere
Add custom for port 8080



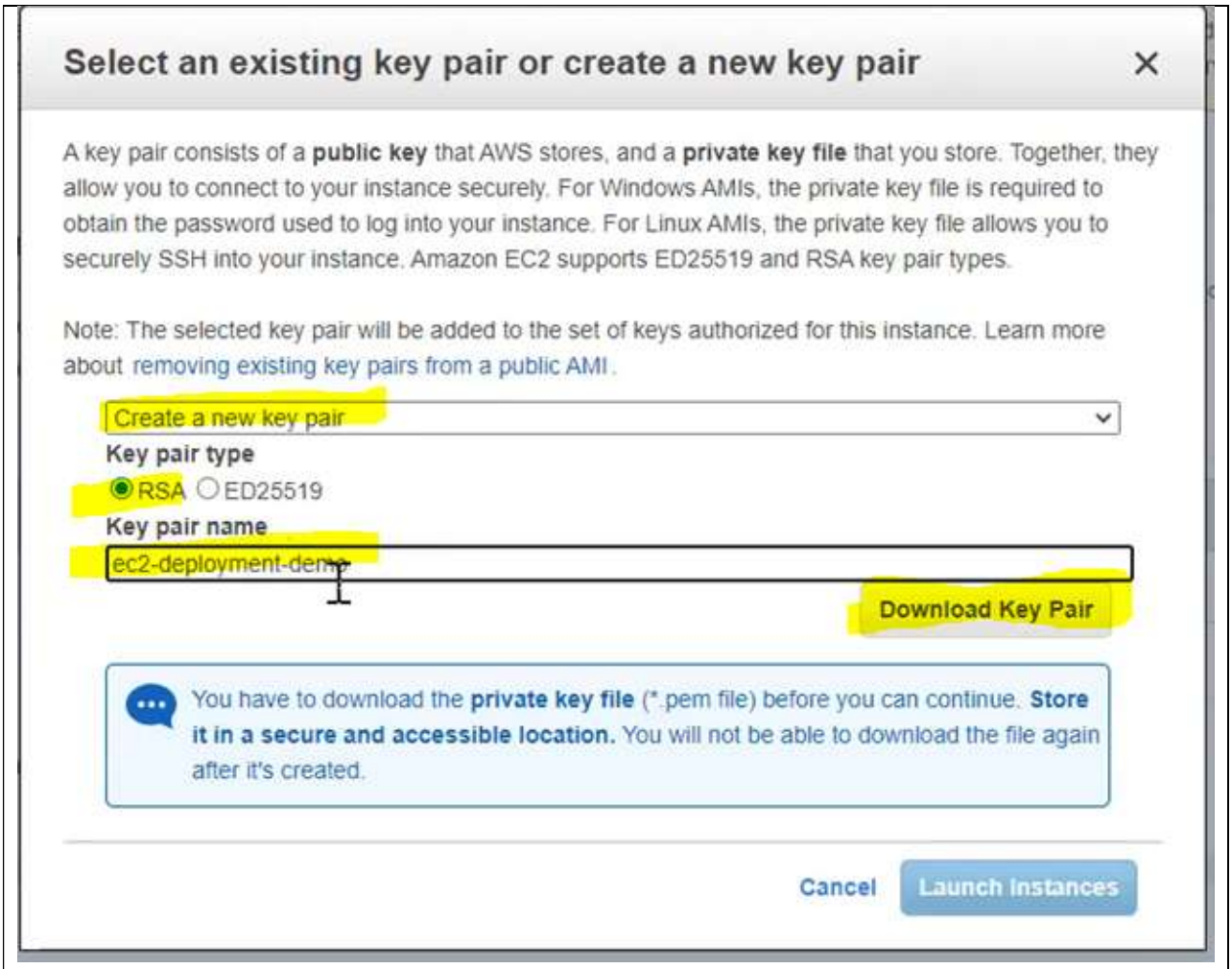
Review page then launch



Need to create a key pair to connect to the instance

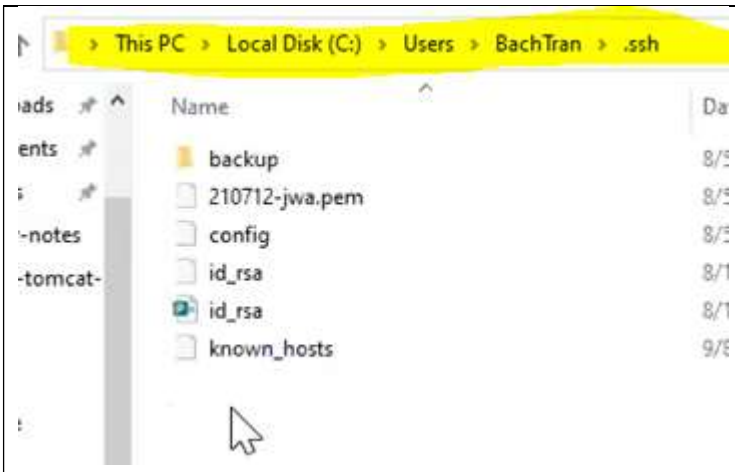


Select Create a new key pair
Keep RSA
Enter a Key pair name



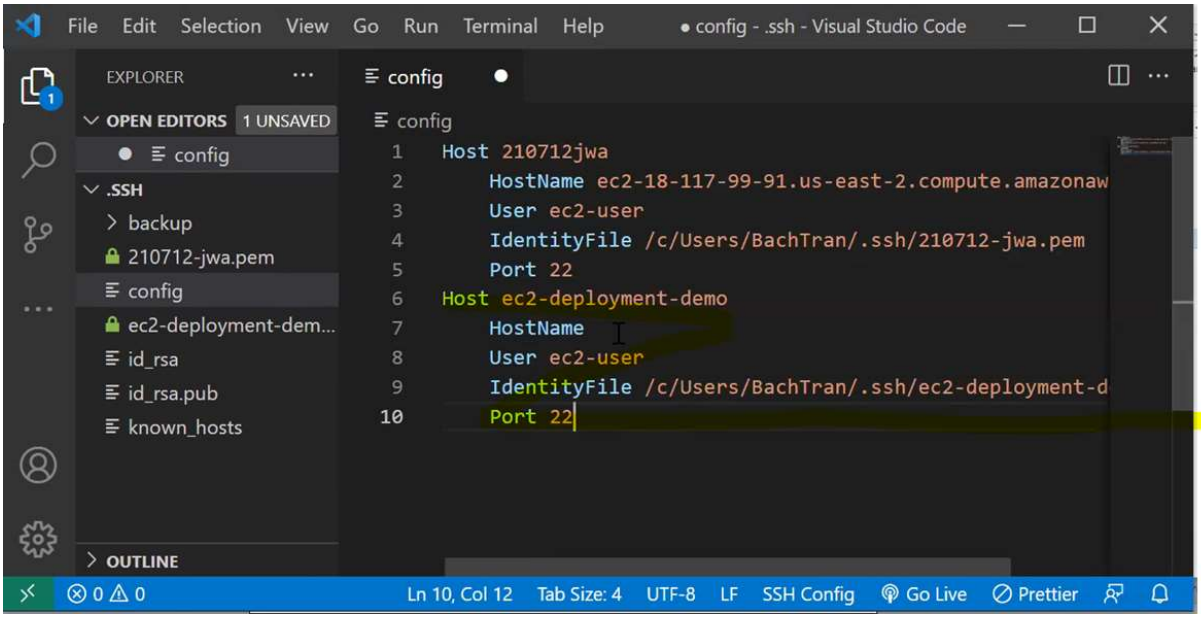
Download key pair
Keep file in secure location
Do not lose or you will not be able to access your EC2 instance
SHOULD NOT push file to git hub
ONLY share with people who will do the deployment

The file needs to be placed on your c drive under your user folder in the .ssh folder.

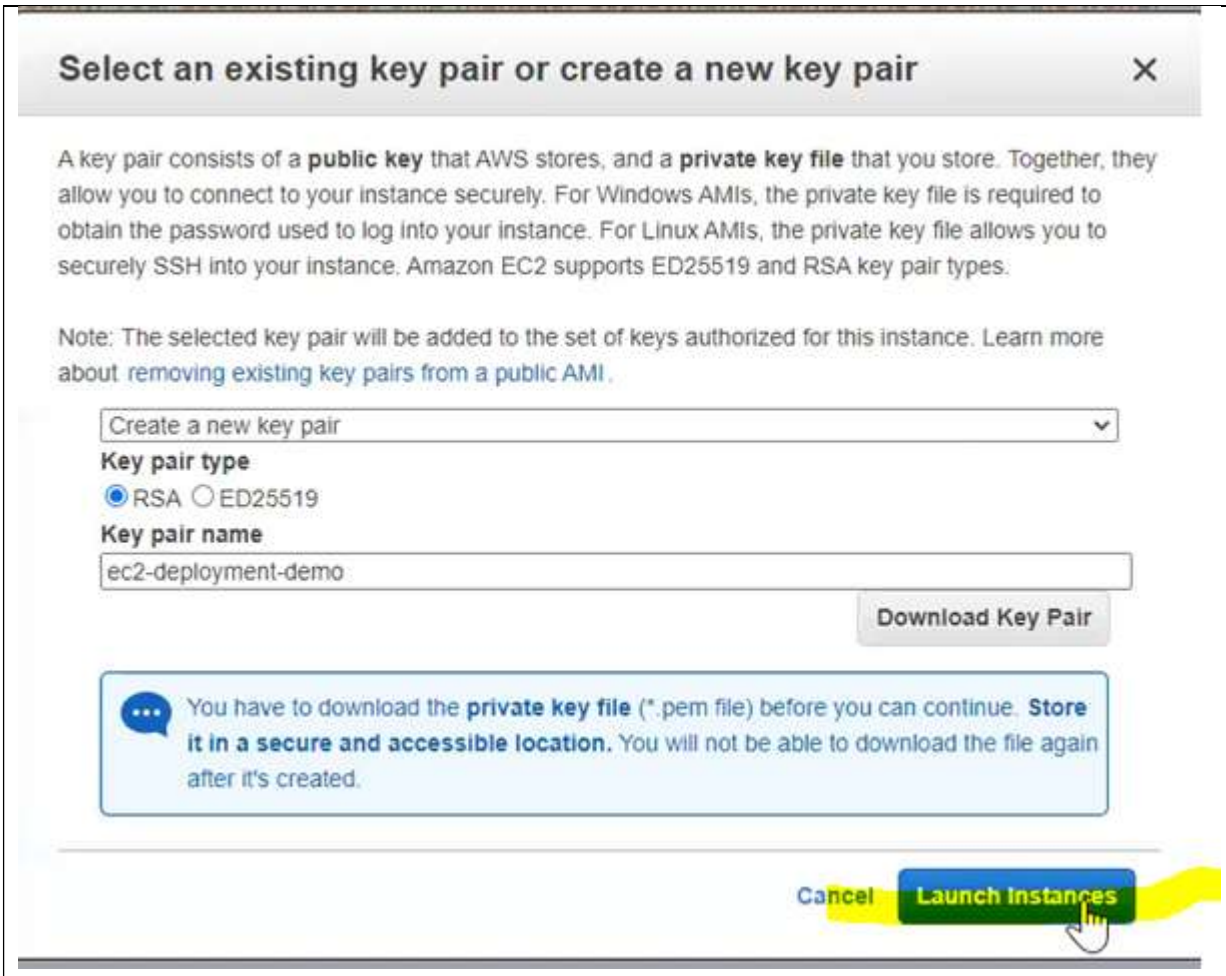


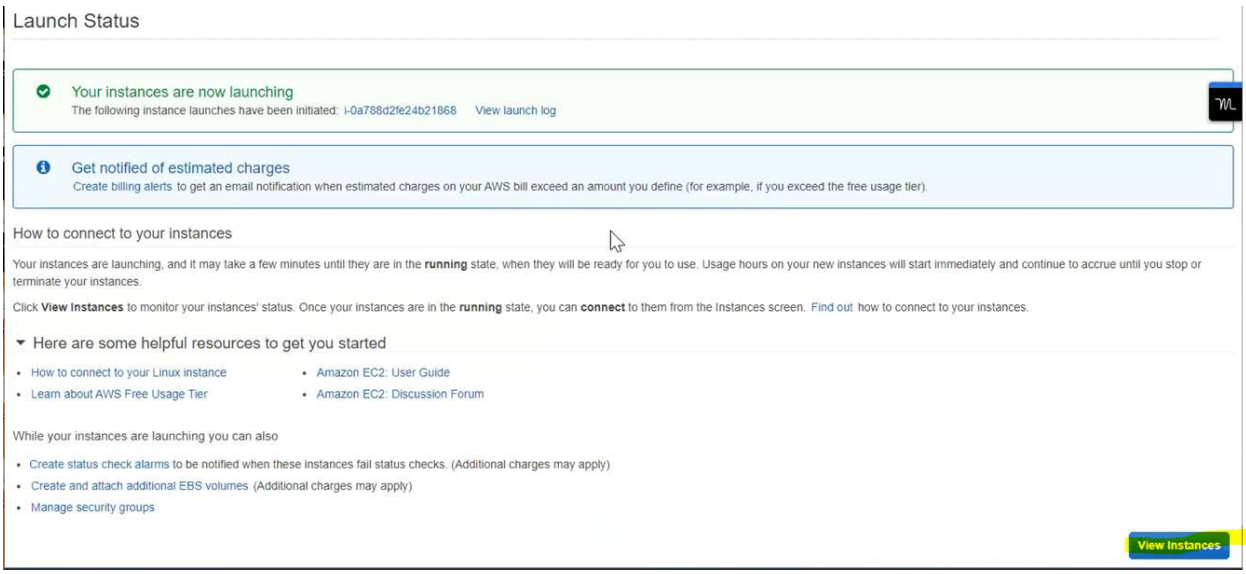
I did not have this folder so I created it.

Another file needed is a config file without an extension.
If you don't have one then create one.
The first host was already in Bach's file and don't think it is needed except for the HostName being your aws ec2 host. This should be in the ec2-deployment-demo host or whatever you call it.

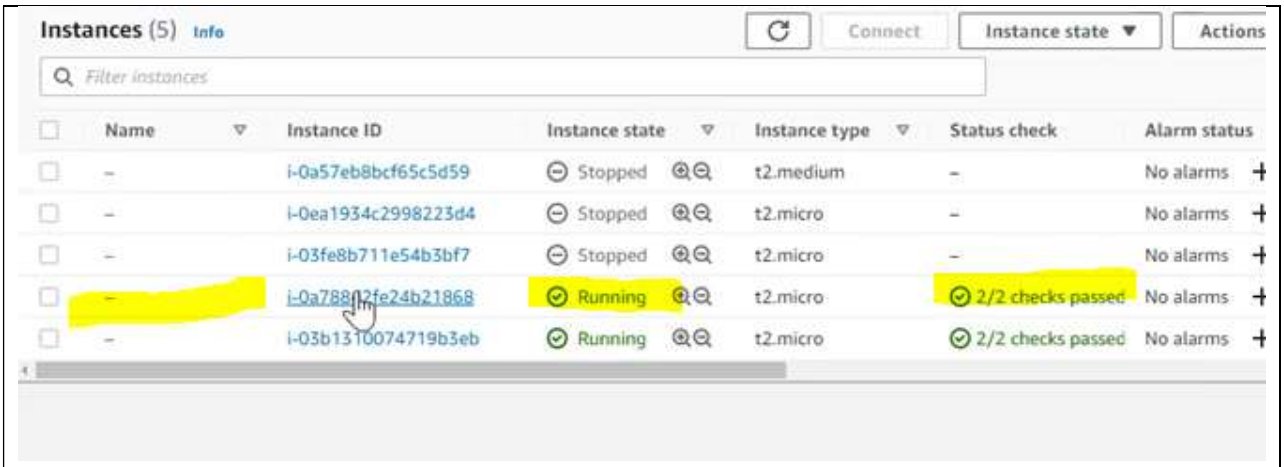


You will need to wait until your instance is up and running to find the HostName for the config file.



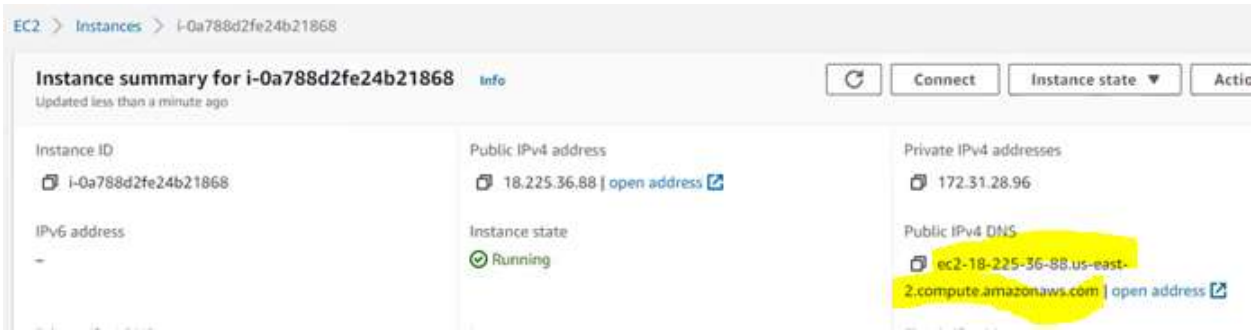


Should look something like this after starting and passing the status check.

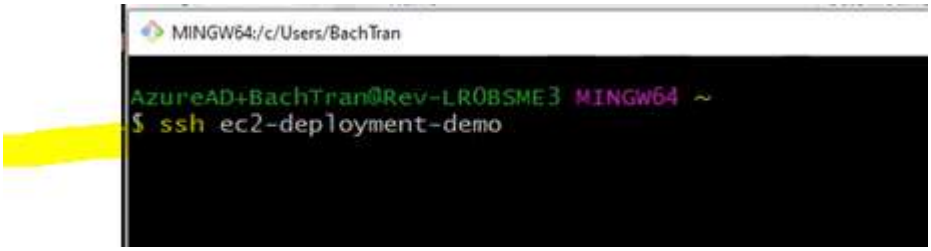


Now click on the instance link to go get the HostName for the config file.

Copy the Public IPv4 DNS to the HostName in your config file.

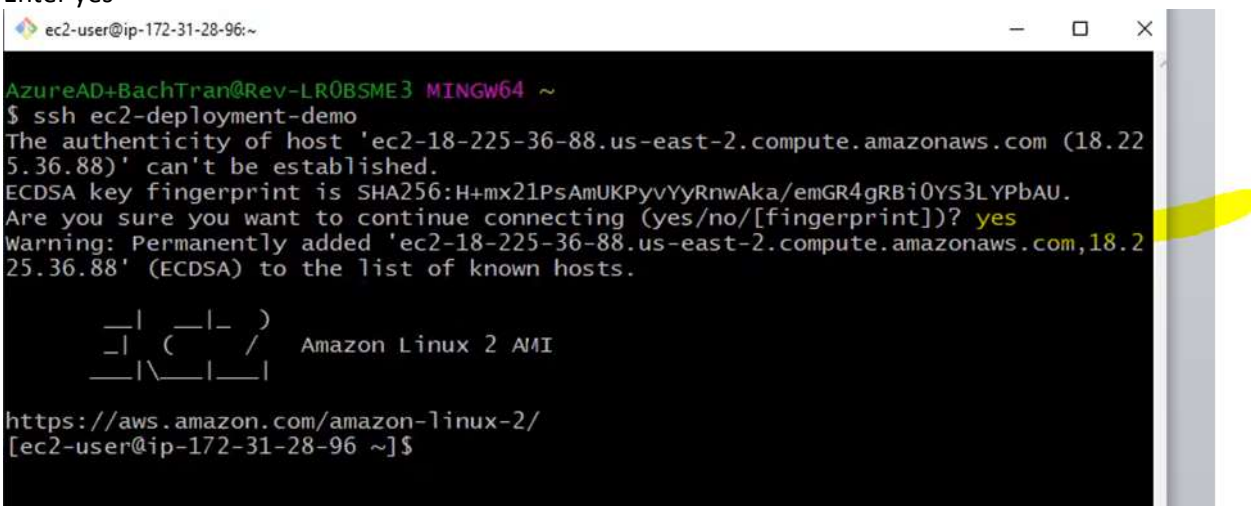


Now we will connect to the instance using the config file through git bash



ssh your file
`ssh amt-mvc-deploy`

Enter yes

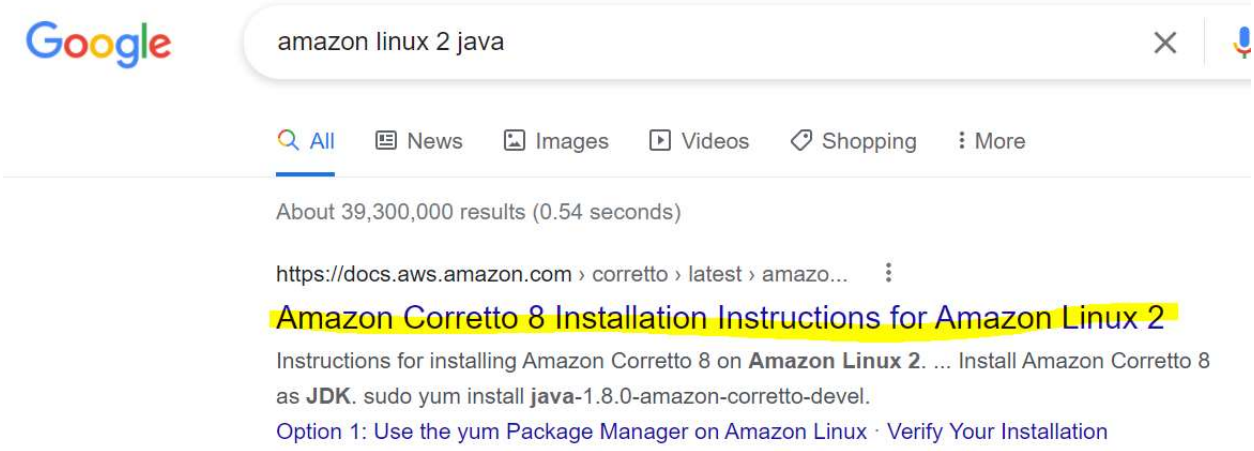


The above did not work for me at first. Instead I found the following.
If it does not work for you then something is probably wrong in your config file.
I did not have the correct User name of **ec2-user**.
Your file should look something like this:

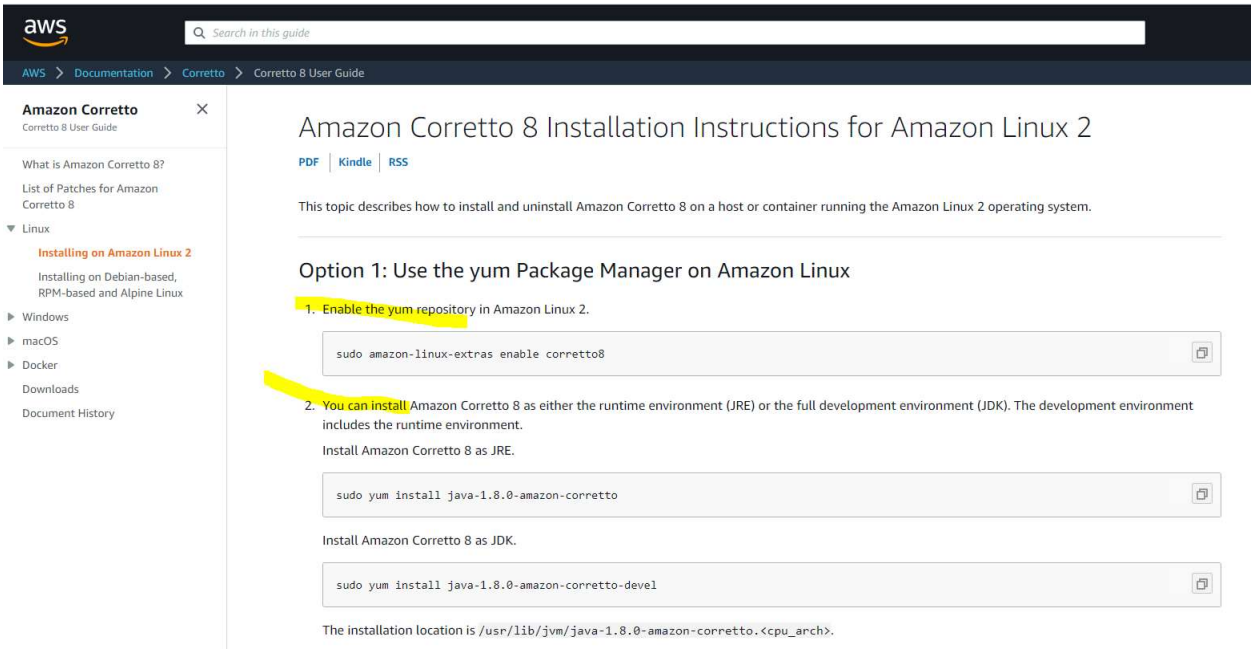
```
Host amt-mvc-deploy
 HostName ec2-18-117-158-53.us-east-2.compute.amazonaws.com
 User ec2-user
 IdentityFile /c/Users/tlw8748253/.ssh/amt-mvc-keys.pem
 Port 22
```

There is nothing in the server directory so we need to install Git, Java and Tomcat

Do a google search for “amazon linux 2 java”



https://docs.aws.amazon.com/corretto/latest/corretto-8-ug/amazon-linux-install.html



Follow instructions on the page

Option 1: Use the yum Package Manager on Amazon Linux

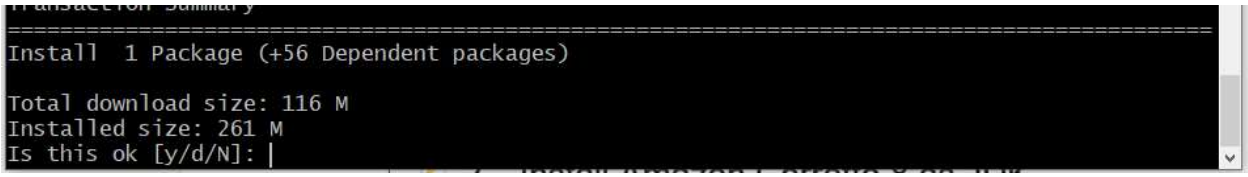
1. Enable the yum repository in Amazon Linux 2.

sudo amazon-linux-extras enable corretto8

Don't install the JRE
~~sudo yum install java-1.8.0-amazon-corretto~~

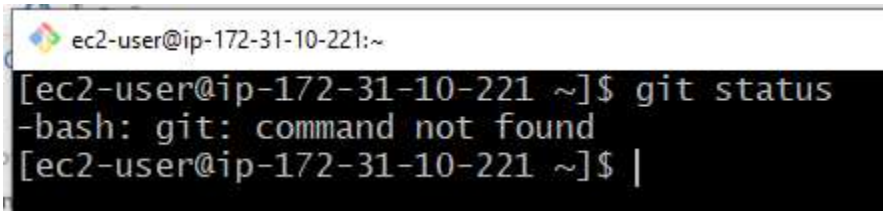
2. Install Amazon Corretto 8 as JDK.

DO install the JDK
sudo yum install java-1.8.0-amazon-corretto-devel

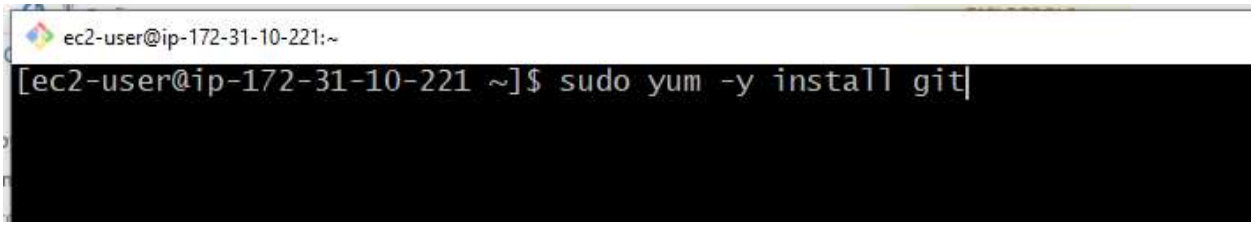


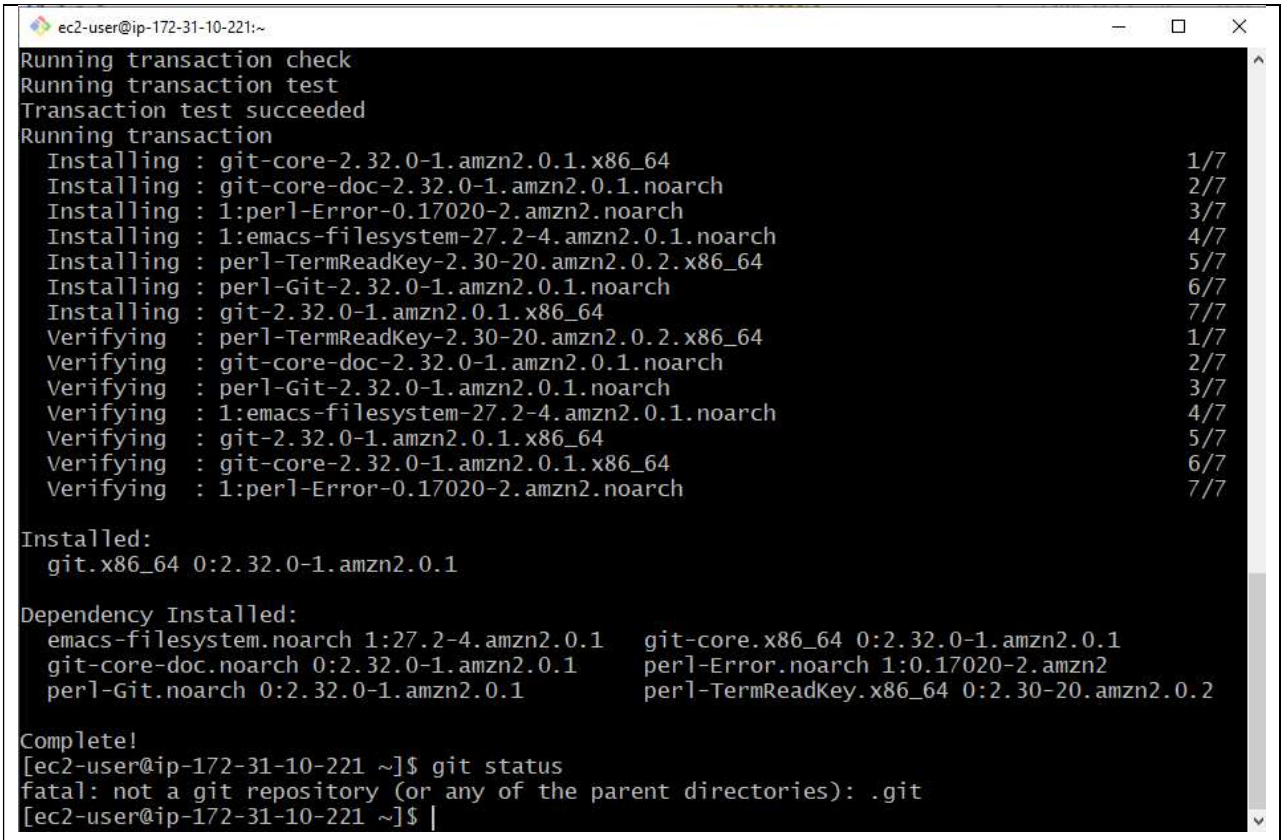
Answer y at the prompt

Git needs to be installed as well.



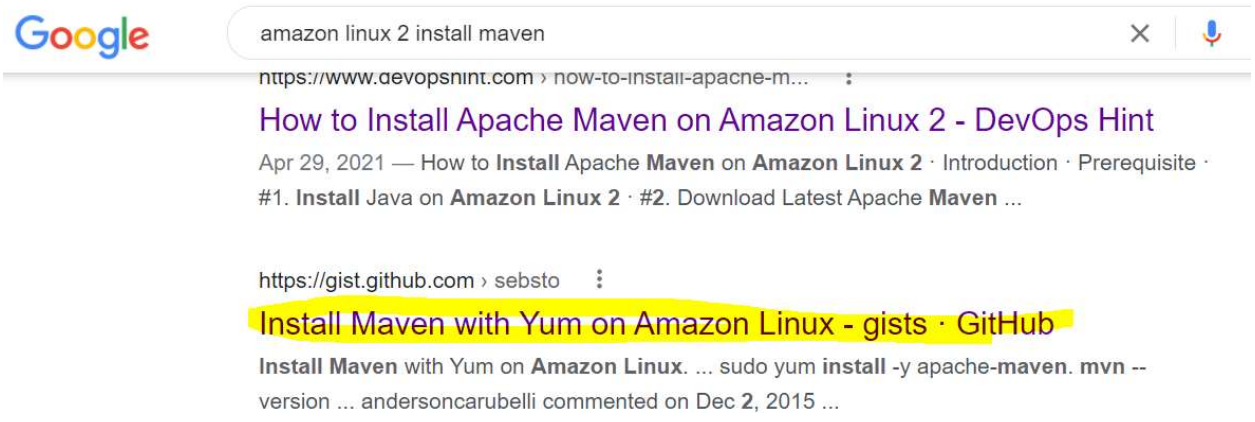
sudo yum -y install git



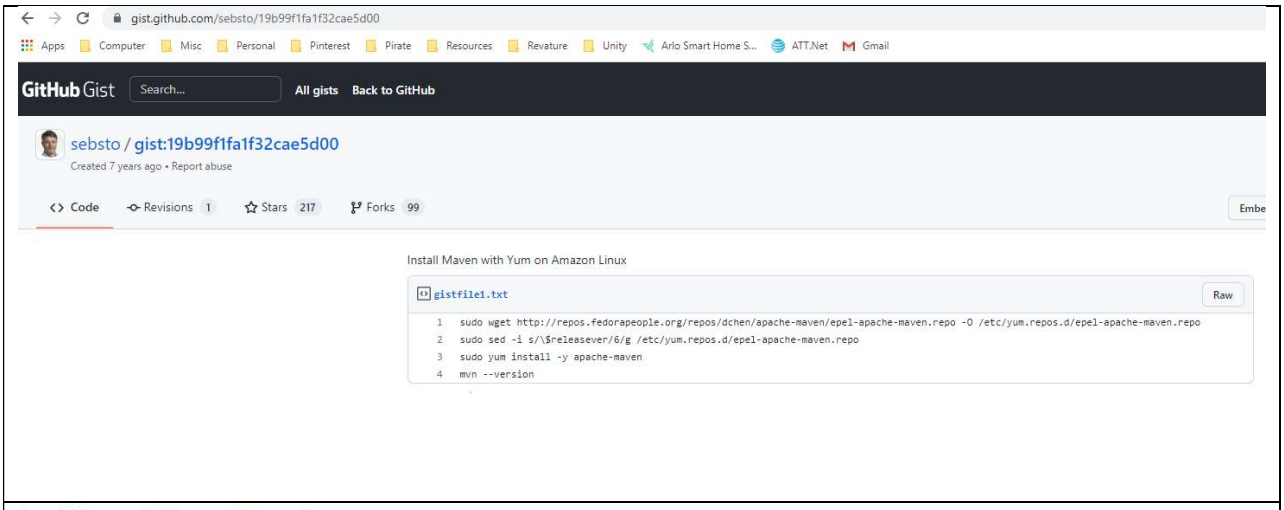


Will need git in future to clone repositories like Jenkins

Next install Maven on Linux (Google search)



https://gist.github.com/sebsto/19b99f1fa1f32cae5d00



Install Maven with Yum on Amazon Linux

gistfile1.txt — Raw

```
1  sudo wget http://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo
2  sudo sed -i s/\$releasever/6/g /etc/yum.repos.d/epel-apache-maven.repo
3  sudo yum install -y apache-maven
4  mvn --version
```

Copy and paste one at a time or all at once

```
sudo wget http://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo

sudo sed -i s/\$releasever/6/g /etc/yum.repos.d/epel-apache-maven.repo

sudo yum install -y apache-maven

mvn --version
```

```
Complete!
[ec2-user@ip-172-31-10-221 ~]$ mvn --version
mvn --version
Apache Maven 3.5.2 (138edd61fd100ec658bfa2d307c43b76940a5d7d; 2017-10-18T07:58:13Z)
Maven home: /usr/share/apache-maven
Java version: 1.7.0_261, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.261-2.6.22.2.amzn2.0.1.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.14.243-185.433.amzn2.x86_64", arch: "amd64", family: "unix"
[ec2-user@ip-172-31-10-221 ~]$
```

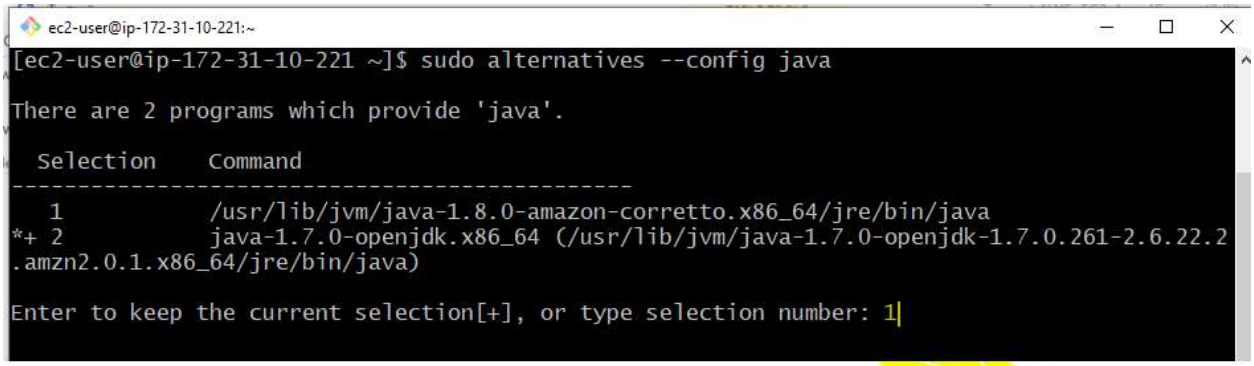
You can also check the Java version

java -version

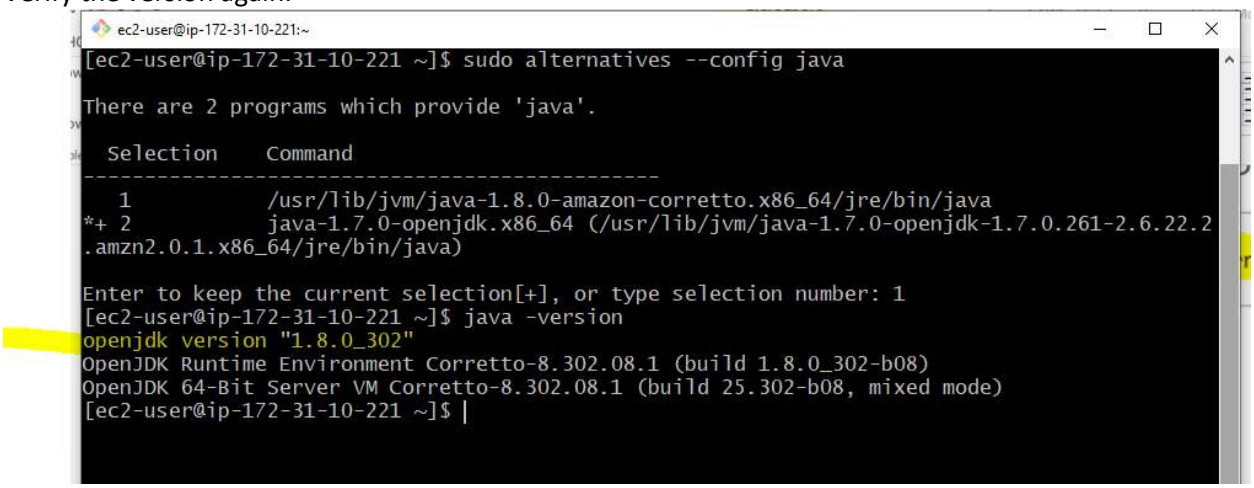
```
ec2-user@ip-172-31-10-221:~
[ec2-user@ip-172-31-10-221 ~]$ java -version
java version "1.7.0_261"
OpenJDK Runtime Environment (rhel-2.6.22.2.amzn2.0.1-x86_64 u261-b02)
OpenJDK 64-Bit Server VM (build 24.261-b02, mixed mode)
[ec2-user@ip-172-31-10-221 ~]$
```

It appears that Java 1.8 was not installed and not installed for Bach as well so:
Further down the AWS page are other sudo commands.


```
sudo alternatives --config java
```

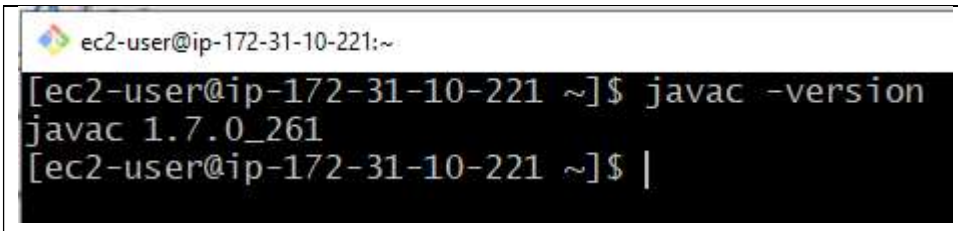


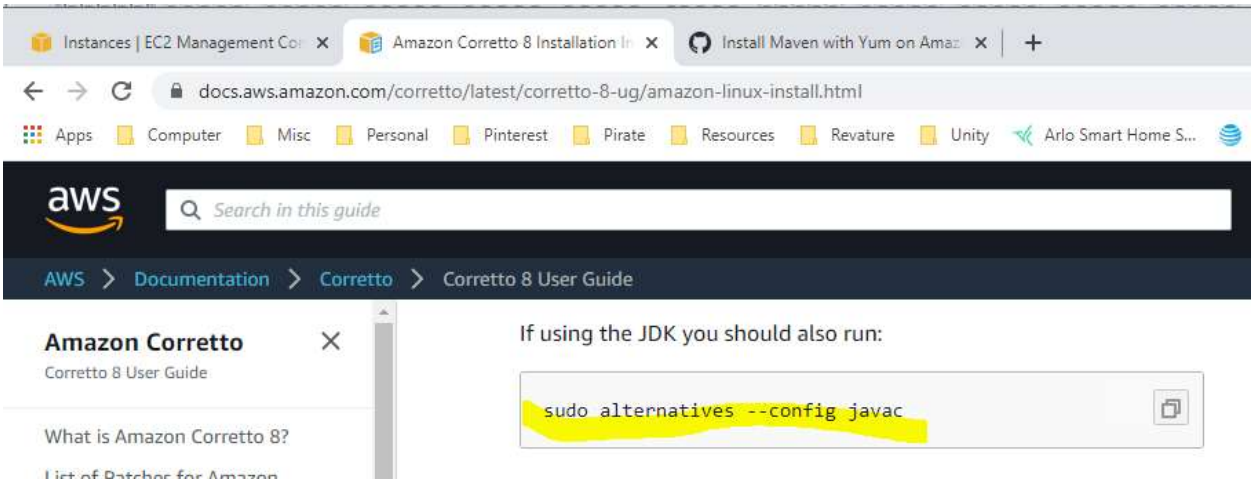
Enter 1 for java 1.8
Verify the version again:



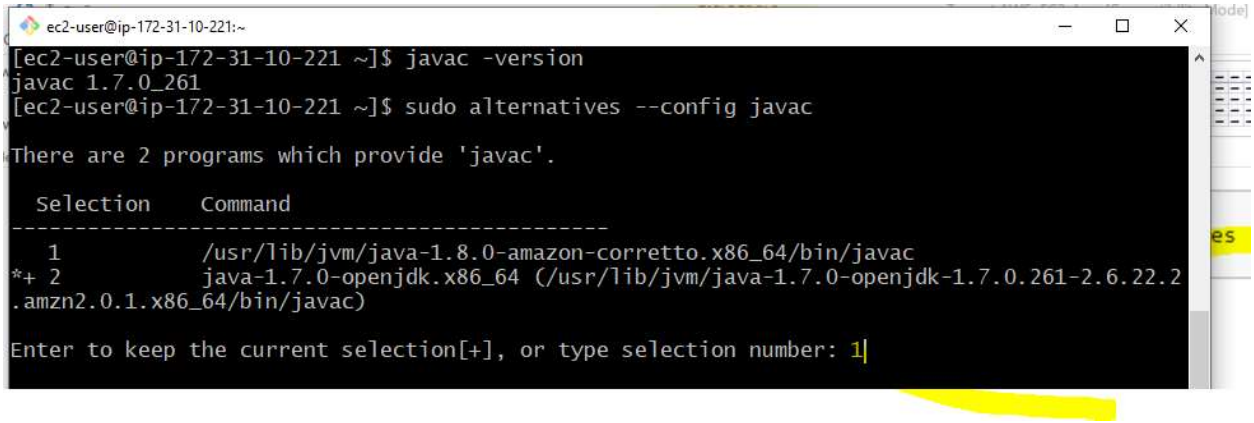
Now it is java 1.8

However javac version is not correct:

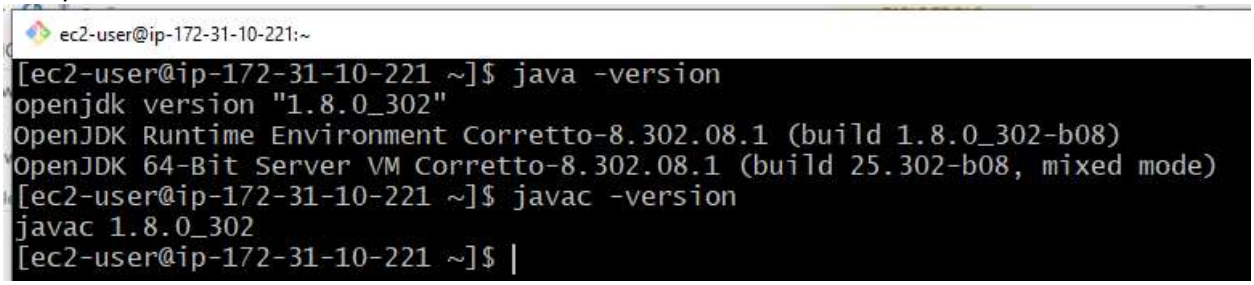





```
sudo alternatives --config javac
```



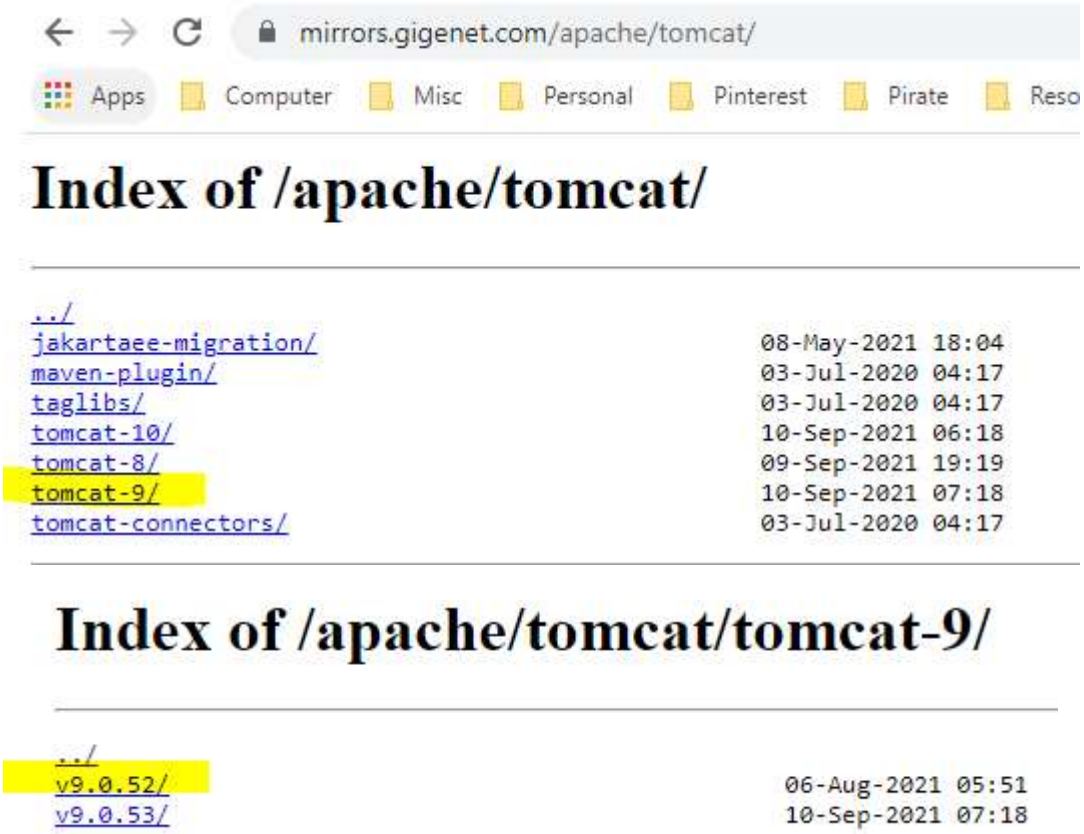
Enter 1 again
Verify both:



Maven is also at 1.8

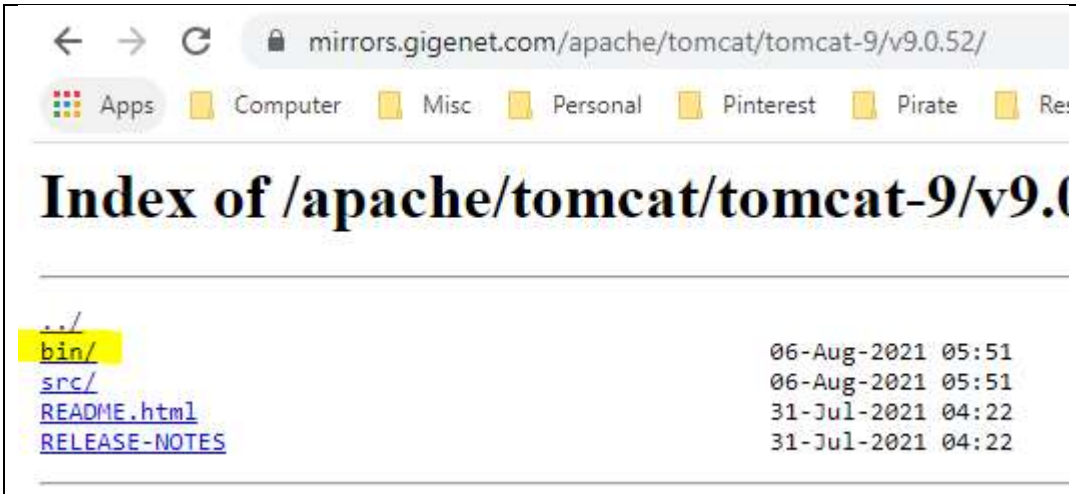
```
ec2-user@ip-172-31-10-221:~
[ec2-user@ip-172-31-10-221 ~]$ mvn -version
Apache Maven 3.5.2 (138edd61fd100ec658bfa2d307c43b76940a5d7d; 2017-10-18T07:58:13Z)
Maven home: /usr/share/apache-maven
Java version: 1.8.0_302, vendor: Amazon.com Inc.
Java home: /usr/lib/jvm/java-1.8.0-amazon-corretto.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.14.243-185.433.amzn2.x86_64", arch: "amd64", family: "unix"
[ec2-user@ip-172-31-10-221 ~]$
```

Next download Tomcat
https://mirrors.gigenet.com/apache/tomcat/

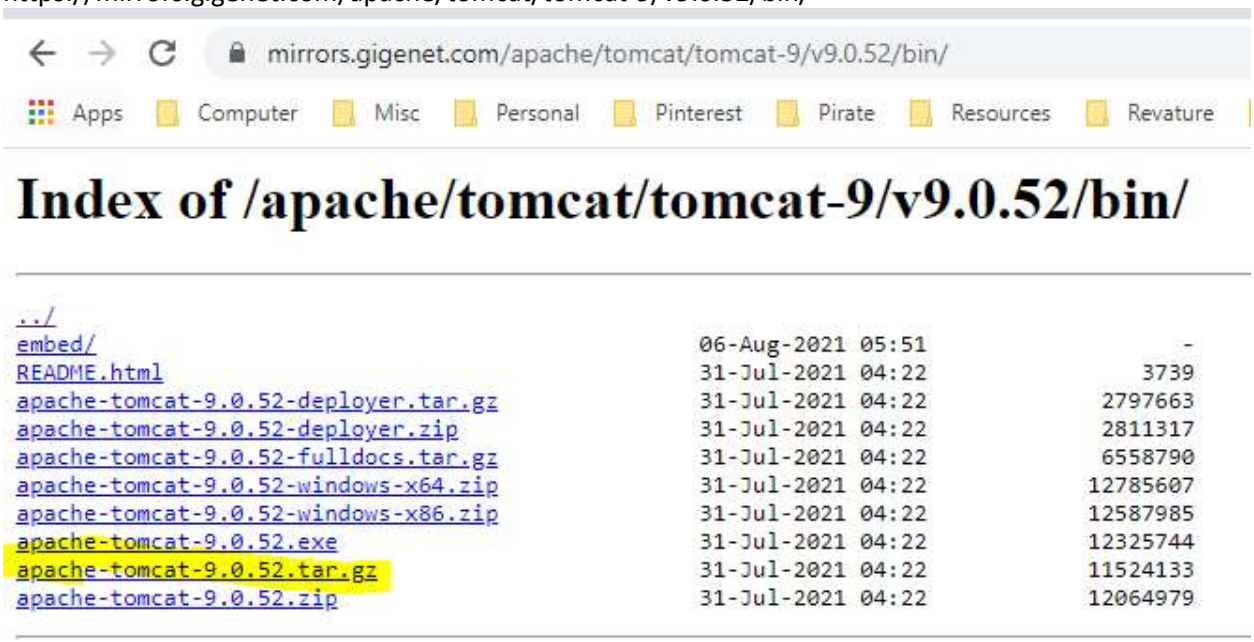


Bach installed v9.0.52 on Sept 8th. A newer version was added on Sept 10th. I installed v9.0.52 to be consistent with the video and his instructions.

https://mirrors.gigenet.com/apache/tomcat/tomcat-9/v9.0.52/



Click bin link
https://mirrors.gigenet.com/apache/tomcat/tomcat-9/v9.0.52/bin/



Need: apache-tomcat-9.0.52.tar.gz
https://mirrors.gigenet.com/apache/tomcat/tomcat-9/v9.0.52/bin/apache-tomcat-9.0.52.tar.gz

run wget command
wget https://mirrors.gigenet.com/apache/tomcat/tomcat-9/v9.0.52/bin/apache-tomcat-9.0.52.tar.gz

```
ec2-user@ip-172-31-10-221:~
[ec2-user@ip-172-31-10-221 ~]$ mvn -version
Apache Maven 3.5.2 (138edd61fd100ec658bfa2d307c43b76940a5d7d; 2017-10-18T07:58:13Z)
Maven home: /usr/share/apache-maven
Java version: 1.8.0_302, vendor: Amazon.com Inc.
Java home: /usr/lib/jvm/java-1.8.0-amazon-corretto.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.14.243-185.433.amzn2.x86_64", arch: "amd64", family: "unix"
[ec2-user@ip-172-31-10-221 ~]$ wget https://mirrors.gigenet.com/apache/tomcat/tomcat-9/v9.0.52/bin/apache-tomcat-9.0.52.tar.gz
--2021-09-11 00:35:10--  https://mirrors.gigenet.com/apache/tomcat/tomcat-9/v9.0.52/bin/apache-tomcat-9.0.52.tar.gz
Resolving mirrors.gigenet.com (mirrors.gigenet.com)... 69.65.16.171, 2001:1850:f000:f000:f000:f000::
Connecting to mirrors.gigenet.com (mirrors.gigenet.com)|69.65.16.171|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11524133 (11M) [application/octet-stream]
Saving to: 'apache-tomcat-9.0.52.tar.gz'

100%[=======================================================================================>] 11,524,133  29.1MB/s   in 0.4s

2021-09-11 00:35:11 (29.1 MB/s) - 'apache-tomcat-9.0.52.tar.gz' saved [11524133/11524133]

[ec2-user@ip-172-31-10-221 ~]$
```

```
ec2-user@ip-172-31-10-221:~
[ec2-user@ip-172-31-10-221 ~]$ ls
apache-tomcat-9.0.52.tar.gz
[ec2-user@ip-172-31-10-221 ~]$
```

Now unzip the tar file:
tar -zxvf apache-tomcat-9.0.52.tar.gz

```
ec2-user@ip-172-31-10-221:~
[ec2-user@ip-172-31-10-221 ~]$ ls
apache-tomcat-9.0.52.tar.gz
[ec2-user@ip-172-31-10-221 ~]$ tar -zxvf apache-tomcat-9.0.52.tar.gz
```

```
ec2-user@ip-172-31-10-221:~
[ec2-user@ip-172-31-10-221 ~]$ ls
apache-tomcat-9.0.52  apache-tomcat-9.0.52.tar.gz
[ec2-user@ip-172-31-10-221 ~]$
```

Creates directory: apache-tomcat-9.0.52

cd apache-tomcat-9.0.52
ls

```
ec2-user@ip-172-31-10-221:~/apache-tomcat-9.0.52
[ec2-user@ip-172-31-10-221 ~]$ ls
apache-tomcat-9.0.52  apache-tomcat-9.0.52.tar.gz
[ec2-user@ip-172-31-10-221 ~]$ cd apache-tomcat-9.0.52
[ec2-user@ip-172-31-10-221 apache-tomcat-9.0.52]$ ls
bin  BUILDING.txt  conf  CONTRIBUTING.md  lib  LICENSE  logs  NOTICE  README.md  RELEASE-NOTES  RUNNING.txt  temp  webapps  work
[ec2-user@ip-172-31-10-221 apache-tomcat-9.0.52]$
```

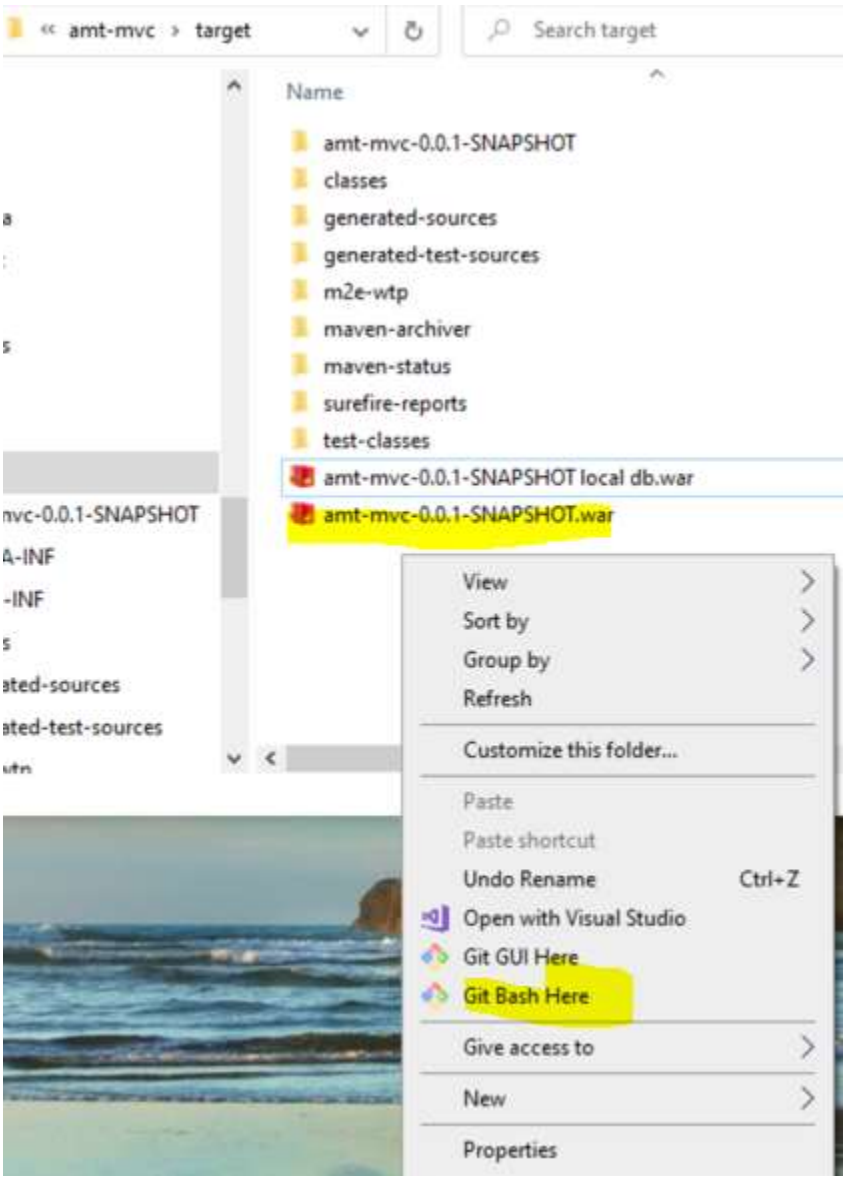
Has same directory structure as on our local machine

Need to copy our war file to Tomcat on AWS.
Will be using scp commands for more info:

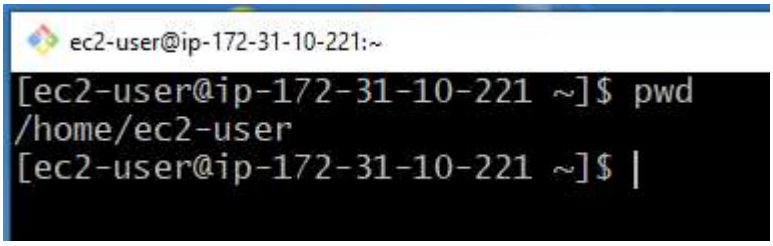
https://linuxize.com/post/how-to-use-scp-command-to-securely-transfer-files/

```
scp [OPTION] [user@]SRC_HOST:]file1 [user@]DEST_HOST:]file2
```

First using Windows explorer navigate to the directory with the war file to upload
Right click and open git bash



In the other git bash window you want to be in /home/ec2-user



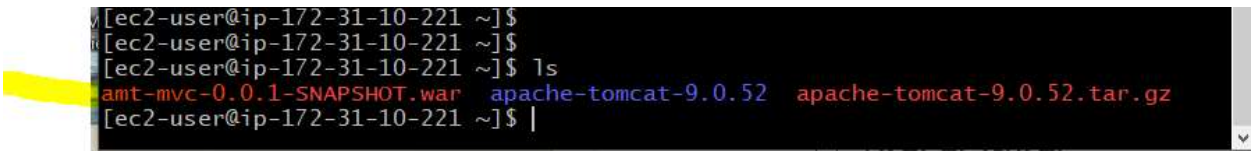
In war file git bash do ls

Where: **amt-mvc-0.0.1-SNAPSHOT.war** is the name of your war file.

```
scp amt-mvc-0.0.1-SNAPSHOT.war amt-mvc-deploy:/home/ec2-user
scp amt-mvc.war amt-mvc-deploy:/home/ec2-user
```

with this config file in the .ssh directory

```
Host amt-mvc-deploy
 HostName ec2-18-216-251-223.us-east-2.compute.amazonaws.com
 User ec2-user
 IdentityFile /c/Users/tlw8748253/.ssh/amt-mvc-keys.pem
 Port 22
```



Verify file is on server with ls command

Rename file with mv command:
mv amt-mvc-0.0.1-SNAPSHOT.war amt-mvc.war

```
ec2-user@ip-172-31-10-221:~
[ec2-user@ip-172-31-10-221 ~]$ ls
amt-mvc.war  apache-tomcat-9.0.52  apache-tomcat-9.0.52.tar.gz
[ec2-user@ip-172-31-10-221 ~]$
```

Now move the file to the apache directory
mv amt-mvc.war apache-tomcat-9.0.52/webapps/

```
[ec2-user@ip-172-31-10-221 ~]$ ls
apache-tomcat-9.0.52  apache-tomcat-9.0.52.tar.gz
[ec2-user@ip-172-31-10-221 ~]$ ls apache-tomcat-9.0.52/webapps/
amt-mvc.war  docs  examples  host-manager  manager  ROOT
[ec2-user@ip-172-31-10-221 ~]$
```

Go to apache bin directory
cd apache-tomcat-9.0.52/bin

```
[ec2-user@ip-172-31-10-221 bin]$ ls
bootstrap.jar                configtest.sh     shutdown.sh
catalina.bat                 daemon.sh         startup.bat
catalina.sh                  digest.bat        startup.sh
catalina-tasks.xml           digest.sh         tomcat-juli.jar
ciphers.bat                  makebase.bat      tomcat-native.tar.gz
ciphers.sh                   makebase.sh       tool-wrapper.bat
commons-daemon.jar           setclasspath.bat  tool-wrapper.sh
commons-daemon-native.tar.gz setclasspath.sh   version.bat
configtest.bat               shutdown.bat      version.sh
[ec2-user@ip-172-31-10-221 bin]$ ^C
[ec2-user@ip-172-31-10-221 bin]$
```

Run the startup script
./startup.sh

```
[ec2-user@ip-172-31-10-221 bin]$ ./startup.sh
Using CATALINA_BASE:   /home/ec2-user/apache-tomcat-9.0.52
Using CATALINA_HOME:   /home/ec2-user/apache-tomcat-9.0.52
Using CATALINA_TMPDIR: /home/ec2-user/apache-tomcat-9.0.52/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ec2-user/apache-tomcat-9.0.52/bin/bootstrap.jar:/hom
e/ec2-user/apache-tomcat-9.0.52/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[ec2-user@ip-172-31-10-221 bin]$
```

To stop the Tomcat use:
./shutdown.sh
In the bin directory
Do not stop it at this time

Now try and send in a request.
You will need your instance's public IPv4

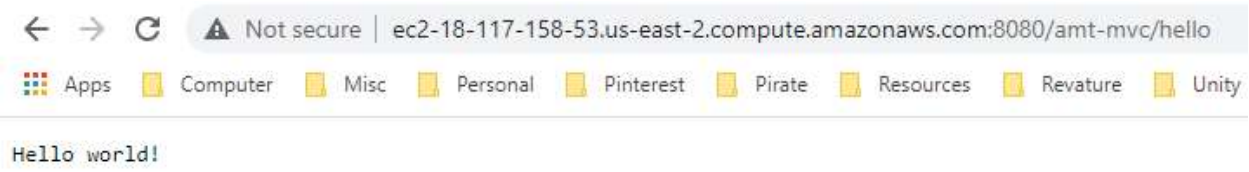
ec2-18-216-251-223.us-east-2.compute.amazonaws.com

replace localhost in Postman urls: http://localhost:8080/amt-mvc/hello from

from localhost to your instance's public IPv4 like mine:

ec2-18-117-158-53.us-east-2.compute.amazonaws.com

Which results in:
http://ec2-18-117-158-53.us-east-2.compute.amazonaws.com:8080/amt-mvc/hello



At the time of writing this document the above url is working and EC2 instance is up and running if you want to try from a browser.

NOTE: the AWS endpoint will change any time you restart the EC2 instance.

This should conclude the building and deployment of the EC2 instance.