Table of Contents

Build an AWS EC2 instance with a Tomcat Server 3

Log into you AWS account 3

Launch instance 3

Step 1: Choose an Amazon Machine Image 4

Step 2: Choose an Instance Type 4

Step 3: Configure Instance Details 4

Step 4: Add Storage 5

Step 5: Add Tags 5

Step 6: Configure Security Group 6

Step 7: Review Instance Launch 6

Create a key pair to connect to the instance 7

Select Create a new key pair 7

Launch the Instance 8

View Instances 8

Instance Summary 8

Local machine .ssh folder 9

Configuration file related to the Key Pair 9

Connect to the AWS EC2 Instance 10

Open a Git Bash window 10

Enter the ssh command 10

Program Installation on New EC2 Instance 11

Install Java on EC2 Instance 11

Option 1: Use the yum Package Manager on Amazon Linux 11

Install Git on EC2 Instance 12

Install Maven on EC2 Instance 13

Run the Maven install commands 13

Update installed software versions 14

Check the Java JDK/JRE version 14

If Java JDK/JRE 1.8 was not installed 14

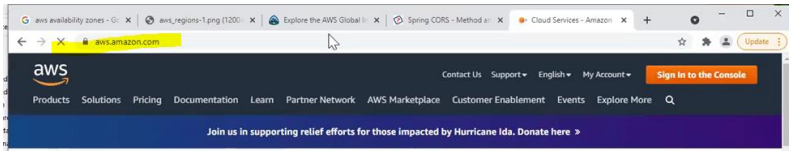
Check the Java Compiler version 15

If Java Compiler 1.8 was not installed 15
Check the Maven version 16
Install Apache Tomcat on EC2 Instance 17
Download Tomcat 17
Run the wget command 18
Extract the tar file 18
The installation of required software on the EC2 instance is completed. 19
Installing a war file on the EC2 instance 20
Copy a project war file from a local machine 20
On the local machine 20
In the EC2 window 21
Verify file is on server with ls command 21
Rename file with mv command: 21
Move the file to the apache directory 21
Start the EC2 Tomcat Server 22
Go to apache bin directory 22
Run the startup script 22
Shutdown Script 22
Test the Project Deployed on the EC2 Instance 23
NOTE: the AWS endpoint will change any time you restart the EC2 instance. 23
This concludes the building of an EC2 instance and the deployment of a war file. 23

Build an AWS EC2 instance with a Tomcat Server

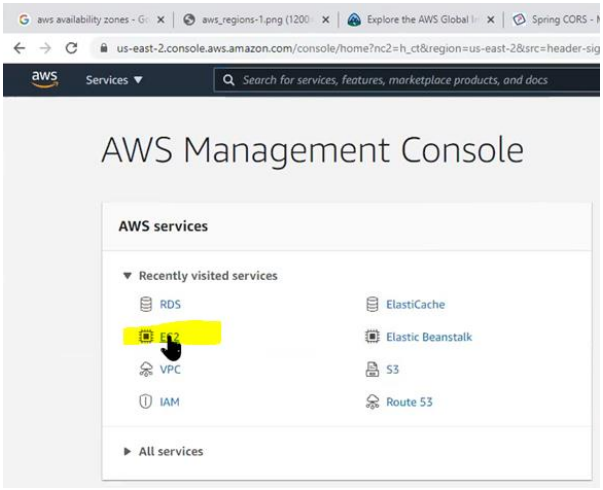
This document walks through the steps involved to build a working EC2 instance on AWS

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



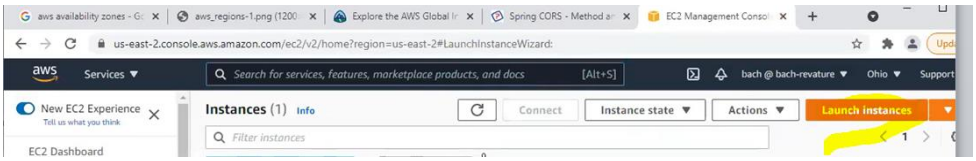
Log into you AWS account

Find and select the EC2 link



Launch instance

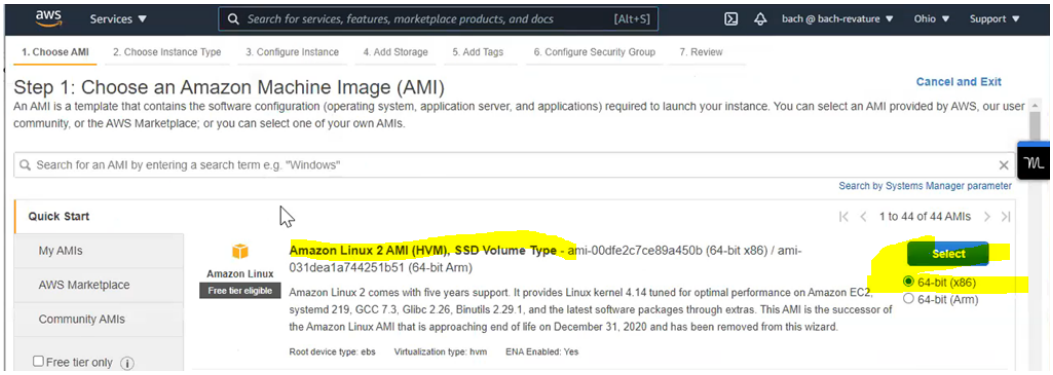
This button may be in a different location depending if you already have created an instance.



Step 1: Choose an Amazon Machine Image

Select a free tier 64-bit (x86)

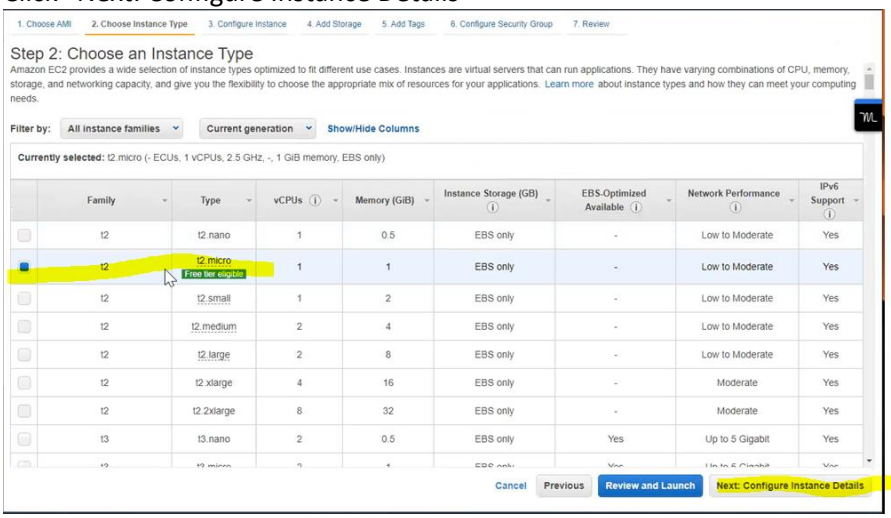
Click “Select”



Step 2: Choose an Instance Type

Select “t2 micro” free tier

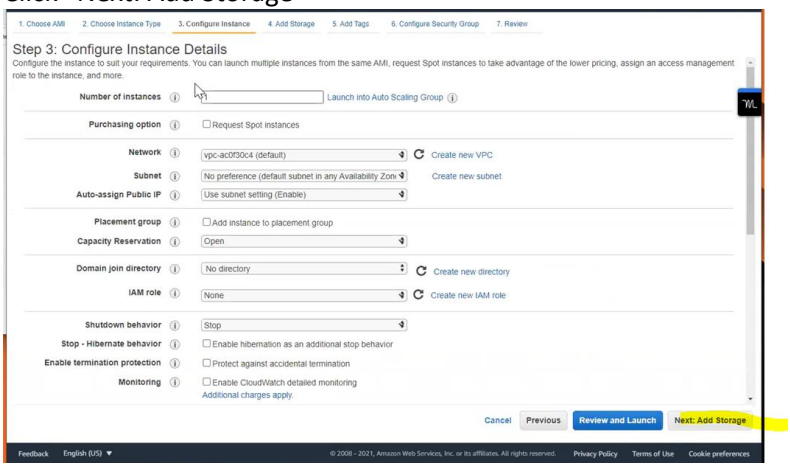
Click “Next: Configure Instance Details”



Step 3: Configure Instance Details

Accept the default

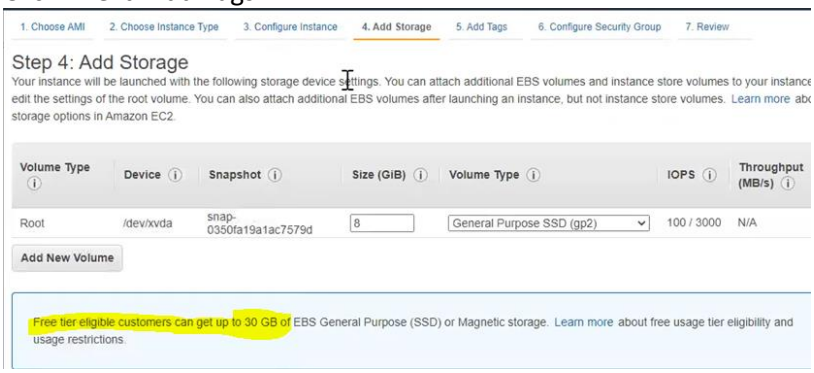
Click “Next: Add Storage”

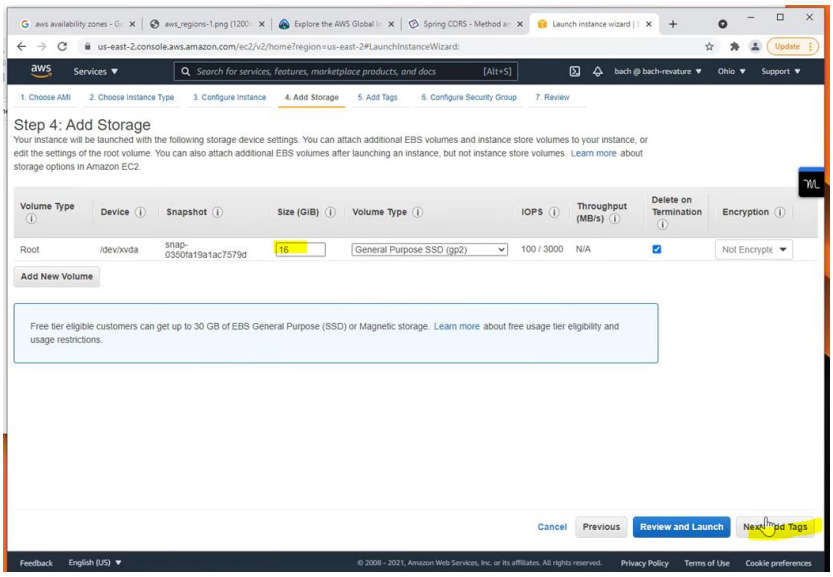


Step 4: Add Storage

Change the storage amount if desired

Click “Next: Add Tags”

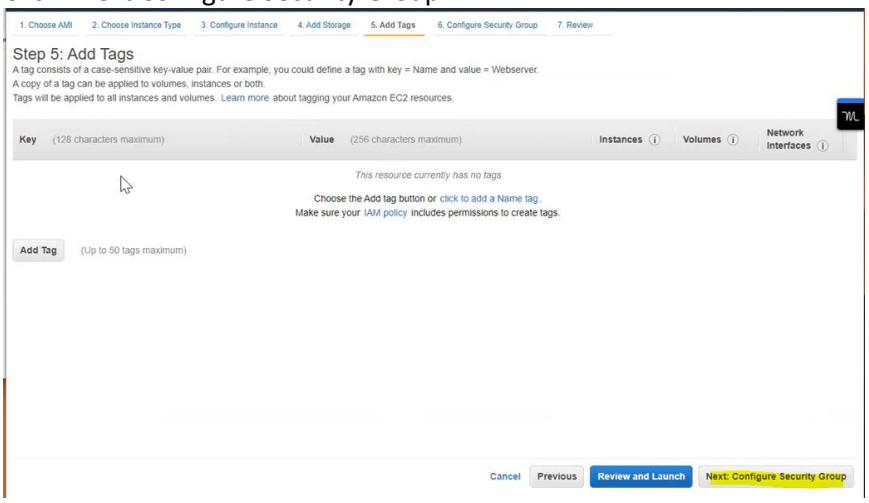




Step 5: Add Tags

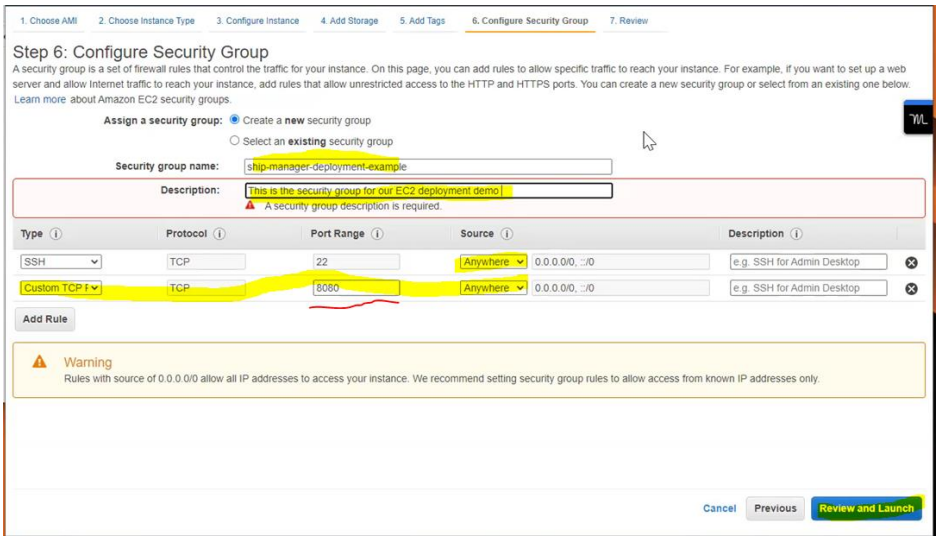
Accept default

Click “Next Configure Security Group”



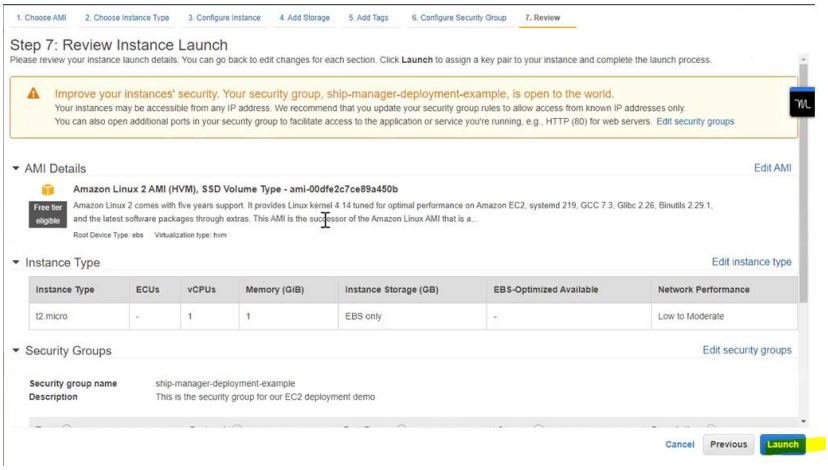
Step 6: Configure Security Group

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



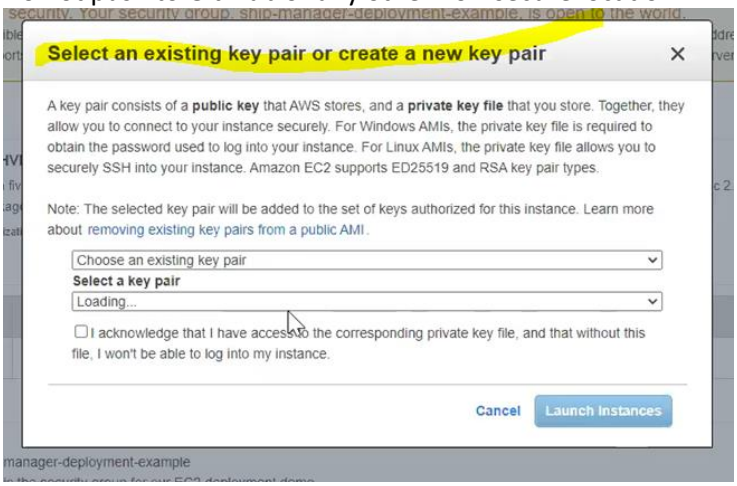
Step 7: Review Instance Launch

Review the instance information
Click “Launch”



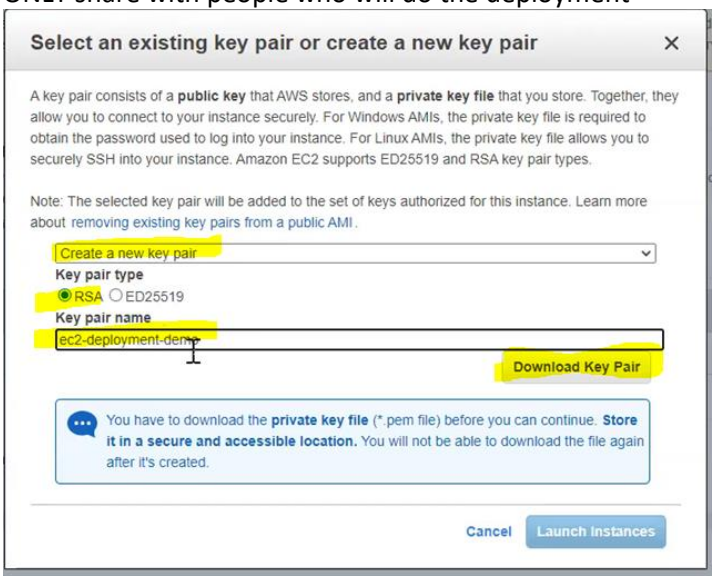
Create a key pair to connect to the instance

You will need this file on your local machine to access the EC2 instance to install files and programs
This file is sensitive and should be kept in a secure location.
Do not push to Git Hub or any other non-secure location.



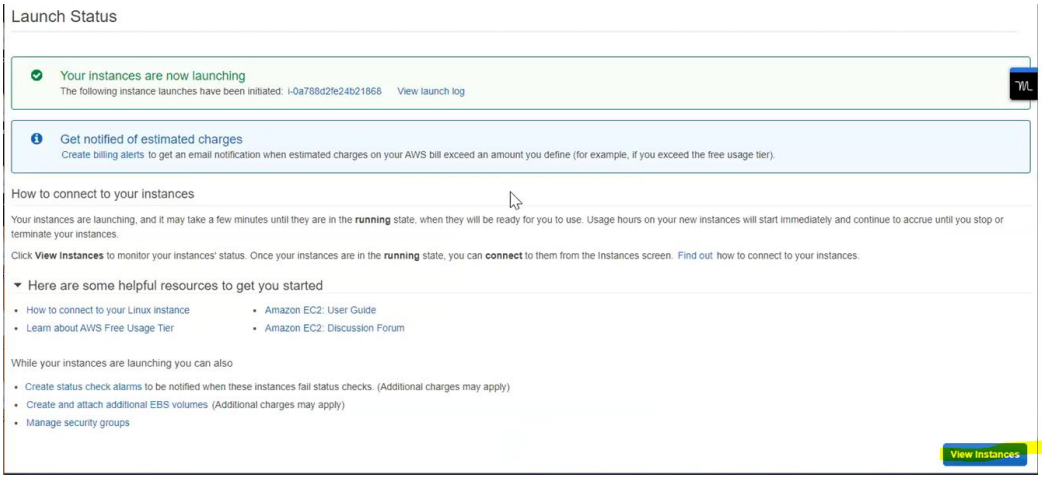
Select Create a new key pair

Keep RSA
Enter a Key pair name
Click “Download Key Pair”
Keep file in secure location
ONLY share with people who will do the deployment



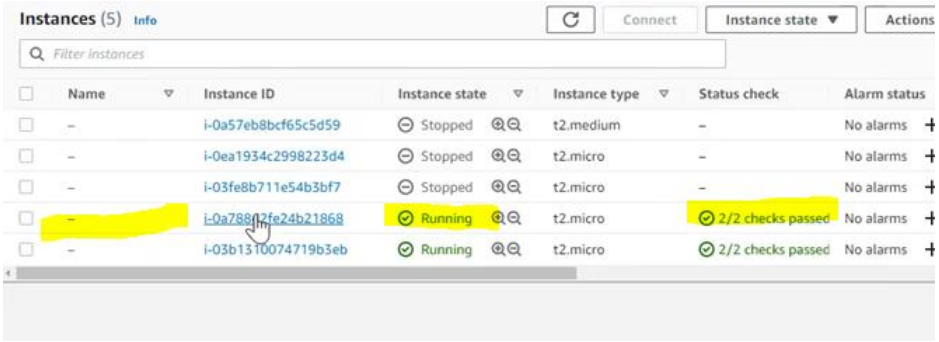
Launch the Instance

After downloading the Key Pair
Click “Launch Instances”
Launch Status page
Click “View Instances”



View Instances

Should look something like this after starting and passing the status check.
Click on the instance link
This will provide access for the HostName to add to a configuration file



Instance Summary

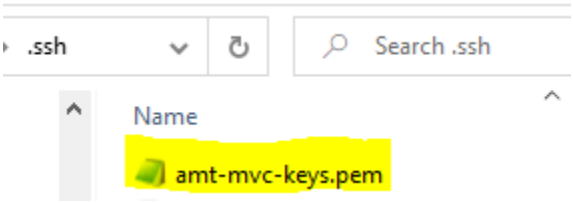
Copy the Public IPv4 DNS to the HostName to place in a configuration file.

Local machine .ssh folder

Check if there is a .ssh folder on your local machine.
If the folder does not exist, create it.

C:\Users\<your username>\.ssh

Move the “key-Pair” file created in the “Download Key Pair” step
amt-mvc-keys.pem is the file I created for my EC2 instance.



Configuration file related to the Key Pair

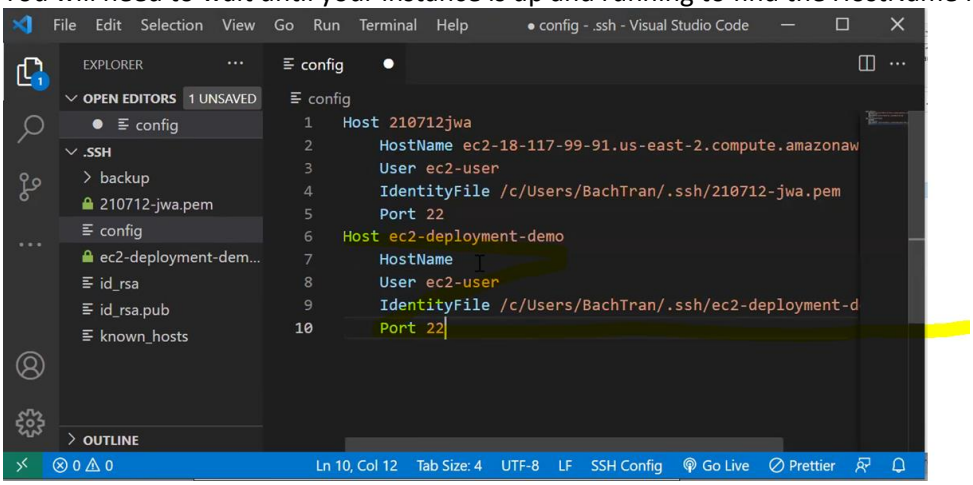
Edit or Create a file named “config” with no extension
Use the Public IPv4 DNS host name copied in the Instance Summary step
Use VS Code or notepad create a file with:

Host <*supply-a-hostname*> HostName <*your-ec2… compute.amazonaws.com*> User **ec2-user ← use this name** IdentityFile /c/Users/<*your-username*>/.ssh/*your-keys.pem* **Port 22 ← use port 22**

The file I created:

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

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



Connect to the AWS EC2 Instance

Open a Git Bash window

This can be open anywhere

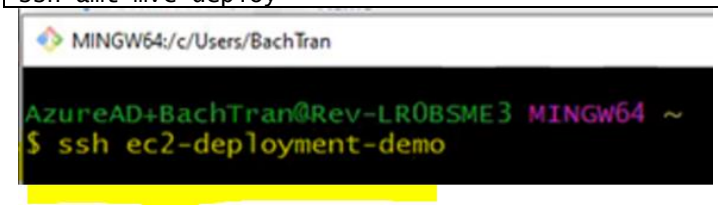
Enter the ssh command

Connect to the EC2 instance using Host <*supply-a-hostname*> defined in the config file

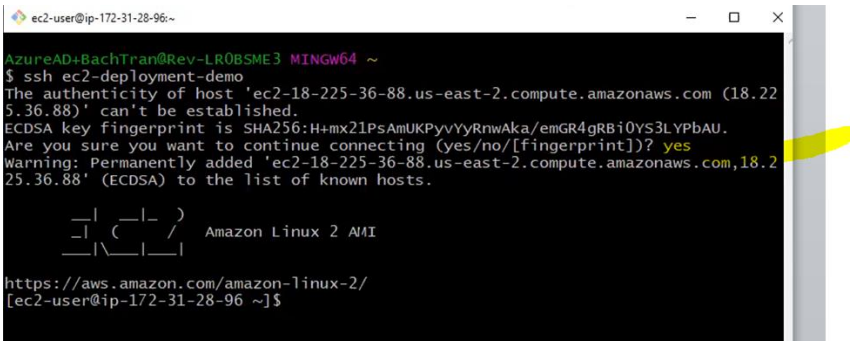
```
ssh <supply-a-hostname>
```

Example:

```
ssh amt-mvc-deploy
```



If you should see a prompt with (yes/no/[fingerprint])?
Type yes and press the Enter Key



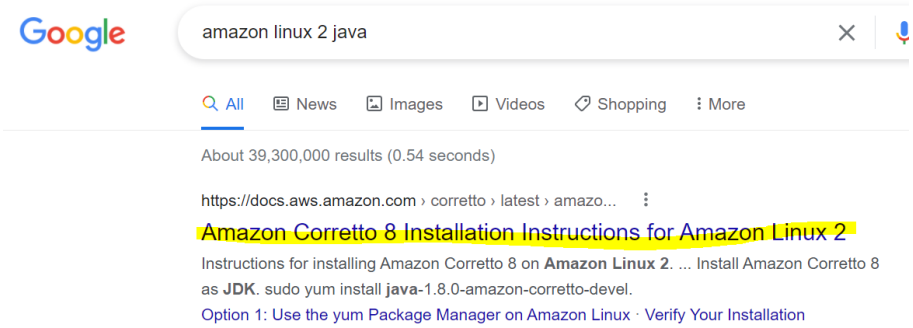
Program Installation on New EC2 Instance

There is nothing in the server directory so we need to install

- Java
- Git
- Maven
- Tomcat

Install Java on EC2 Instance

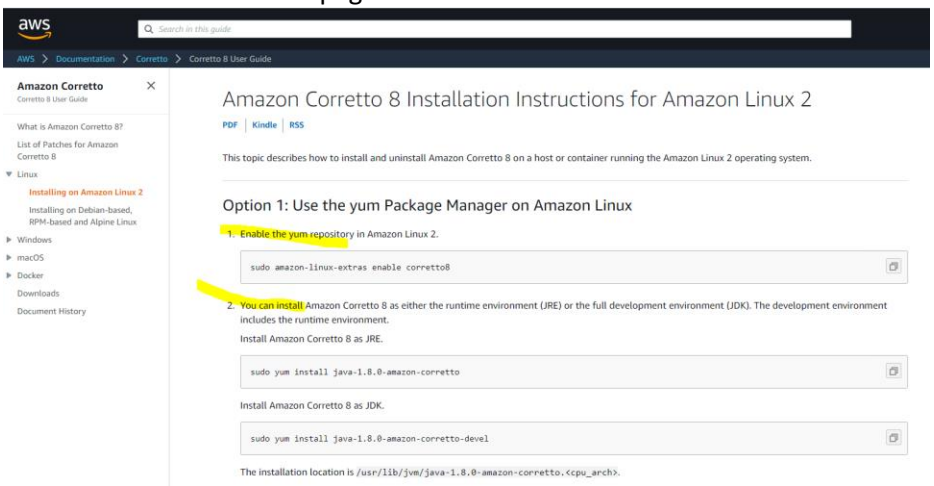
Perform a Google search for “amazon linux 2 java”



Navigate to page:

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

Type y and press enter

```
Transaction Summary
================================================================================
Install  1 Package (+56 Dependent packages)

Total download size: 116 M
Installed size: 261 M
Is this ok [y/d/N]: |
```

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

Install Git on EC2 Instance

Git needs to be installed as well.

Will need git in future to clone repositories like Jenkins

```
ec2-user@ip-172-31-10-221:~
[ec2-user@ip-172-31-10-221 ~]$ git status
-bash: git: command not found
[ec2-user@ip-172-31-10-221 ~]$ |
```

Type the command and press enter

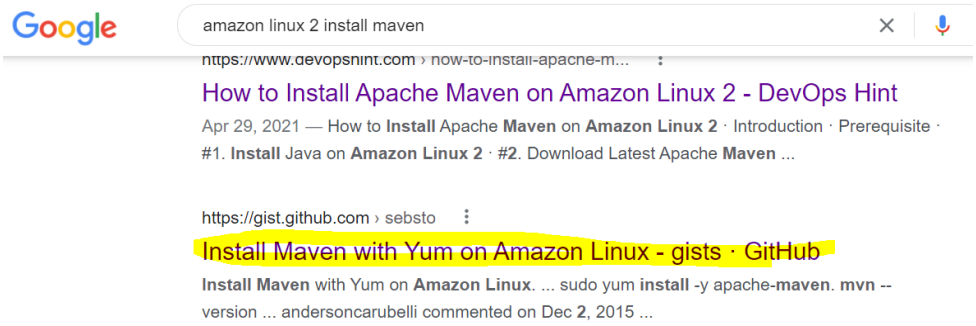
sudo yum -y install git

```
ec2-user@ip-172-31-10-221:~
[ec2-user@ip-172-31-10-221 ~]$ sudo yum -y install git
```

```
ec2-user@ip-172-31-10-221:~
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : git-core-2.32.0-1.amzn2.0.1.x86_64                           1/7
  Installing : git-core-doc-2.32.0-1.amzn2.0.1.noarch                       2/7
  Installing : 1:perl-Error-0.17020-2.amzn2.noarch                          3/7
  Installing : 1:emacs-filesystem-27.2-4.amzn2.0.1.noarch                   4/7
  Installing : perl-TermReadKey-2.30-20.amzn2.0.2.x86_64                    5/7
  Installing : perl-Git-2.32.0-1.amzn2.0.1.noarch                           6/7
  Installing : git-2.32.0-1.amzn2.0.1.x86_64                                7/7
  Verifying  : perl-TermReadKey-2.30-20.amzn2.0.2.x86_64                    1/7
  Verifying  : git-core-doc-2.32.0-1.amzn2.0.1.noarch                       2/7
  Verifying  : perl-Git-2.32.0-1.amzn2.0.1.noarch                           3/7
  Verifying  : 1:emacs-filesystem-27.2-4.amzn2.0.1.noarch                   4/7
  Verifying  : git-2.32.0-1.amzn2.0.1.x86_64                                5/7
  Verifying  : git-core-2.32.0-1.amzn2.0.1.x86_64                           6/7
  Verifying  : 1:perl-Error-0.17020-2.amzn2.noarch                          7/7

Installed:
  git.x86_64 0:2.32.0-1.amzn2.0.1

Dependency Installed:
  emacs-filesystem.noarch 1:27.2-4.amzn2.0.1   git-core.x86_64 0:2.32.0-1.amzn2.0.1
  git-core-doc.noarch 0:2.32.0-1.amzn2.0.1     perl-Error.noarch 1:0.17020-2.amzn2
  perl-Git.noarch 0:2.32.0-1.amzn2.0.1         perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2

Complete!
[ec2-user@ip-172-31-10-221 ~]$ git status
fatal: not a git repository (or any of the parent directories): .git
[ec2-user@ip-172-31-10-221 ~]$ |
```

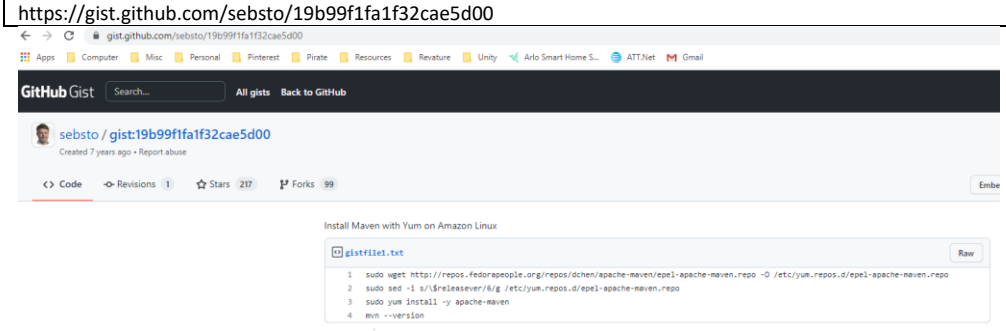
Install Maven on EC2 Instance

Perform a Google search for “amazon linux install maven”



Navigate to page:

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
```

Run the Maven install commands

Copy and paste the commands one at a time or all at once

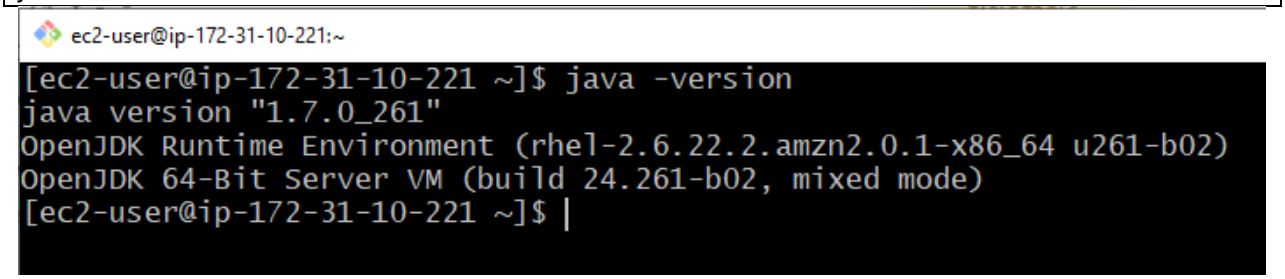
```
sudo wget http://repos.fedorapeople.org/repos/dchen/apache-maven/epel-apache-maven.repo -O /etc/yum.repos.d/epel-apache-maven.repo
sudo sed -i s/\$releasever/6/g /etc/yum.repos.d/epel-apache-maven.repo
sudo yum install -y apache-maven
mvn --version
```

```
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 ~]$
```

Update installed software versions

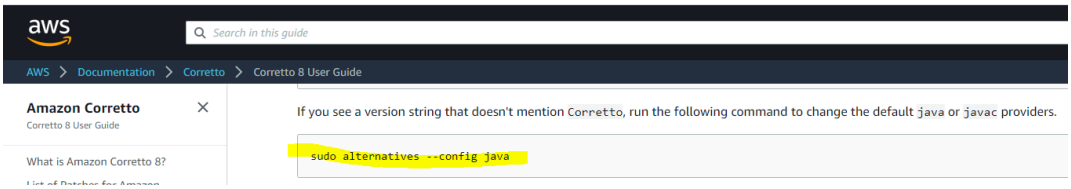
Check the Java JDK/JRE version

```
java -version
```



If Java JDK/JRE 1.8 was not installed

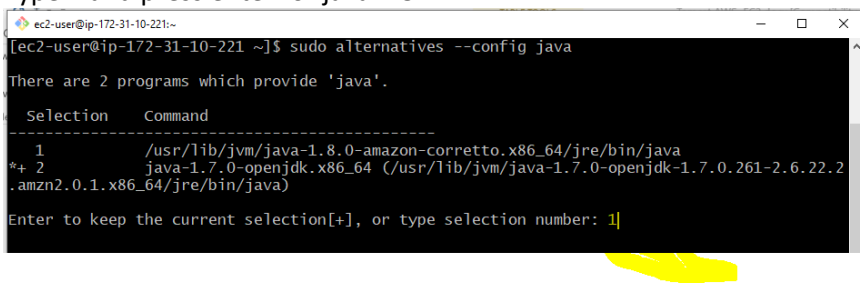
Scroll further down the AWS page are other sudo commands.



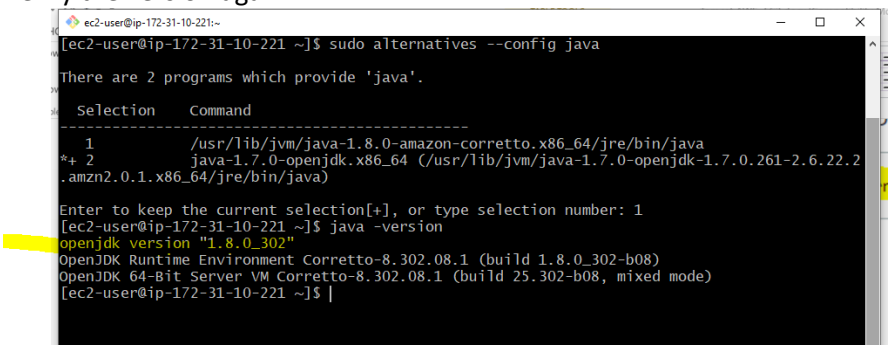
Type the sudo command and press enter

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

Type 1 and press enter for java 1.8



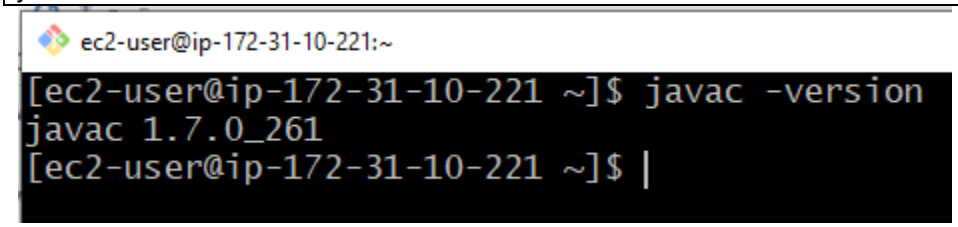
Verify the version again:



Now it is java 1.8

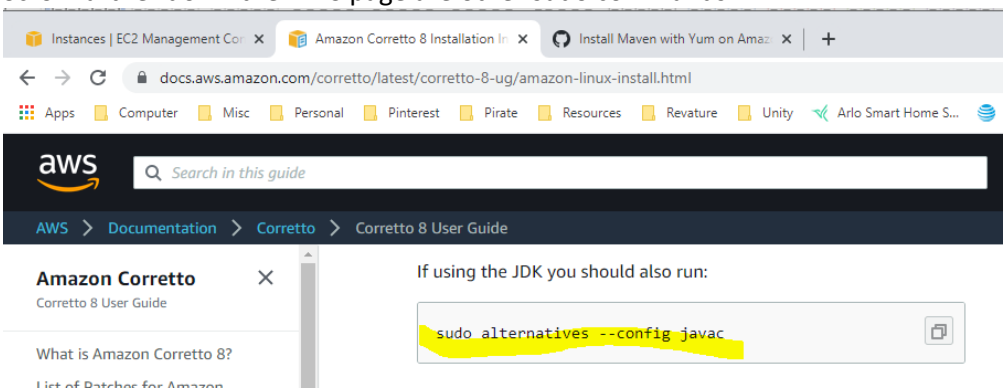
Check the Java Compiler version

```
javac -version
```



If Java Compiler 1.8 was not installed

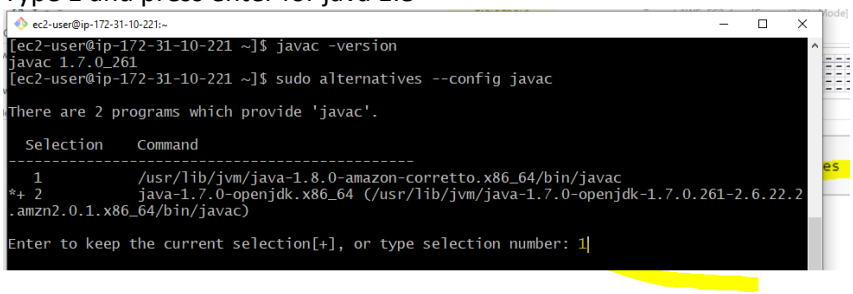
Scroll further down the AWS page are other sudo commands.



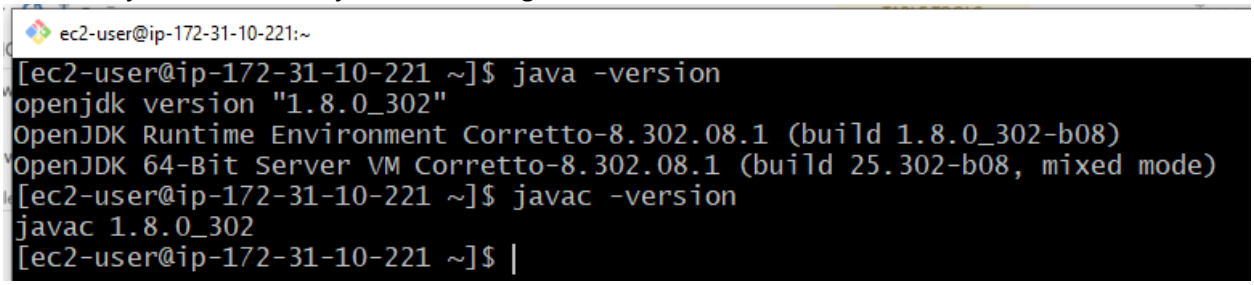
Use the sudo command

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

Type 1 and press enter for java 1.8



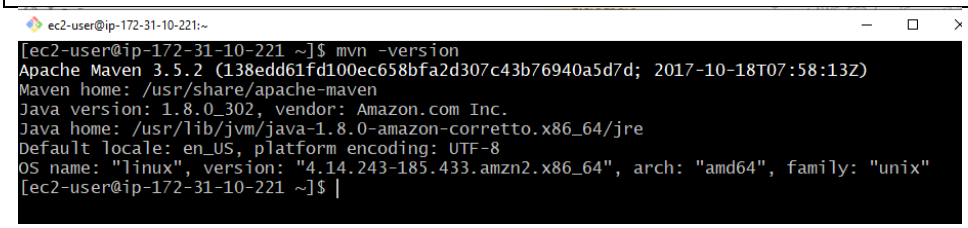
Check the java -version and javac -version again



Check the Maven version

Type the command and press enter

```
mvn -version
```



The version should be correct. There was no alternative in the lecture.

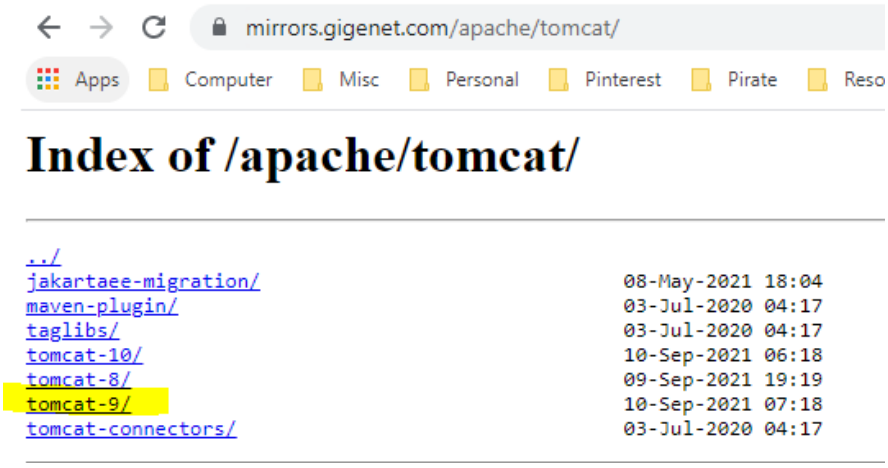
Install Apache Tomcat on EC2 Instance

Download Tomcat

Enter the URL in a browser

https://mirrors.gigenet.com/apache/tomcat/

Select tomcat-9 link

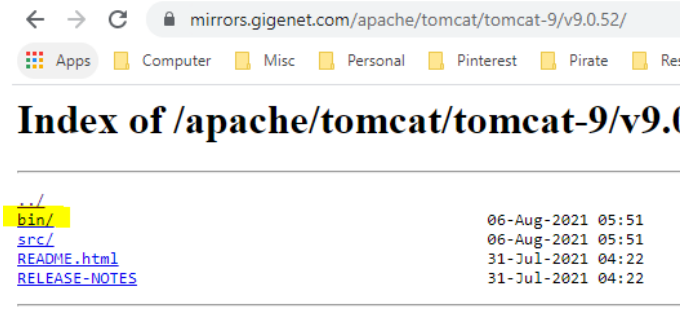


Select v9.0.52 link

The instructor in the video installed v9.0.52 on Sept 8th 2021. A newer version was added on Sept 10th. I installed v9.0.52 to be consistent with the instructions in the video.

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

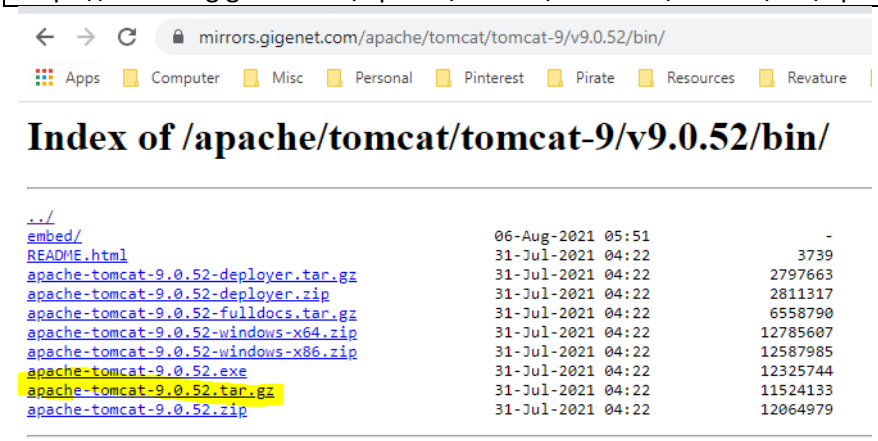
Select bin link



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

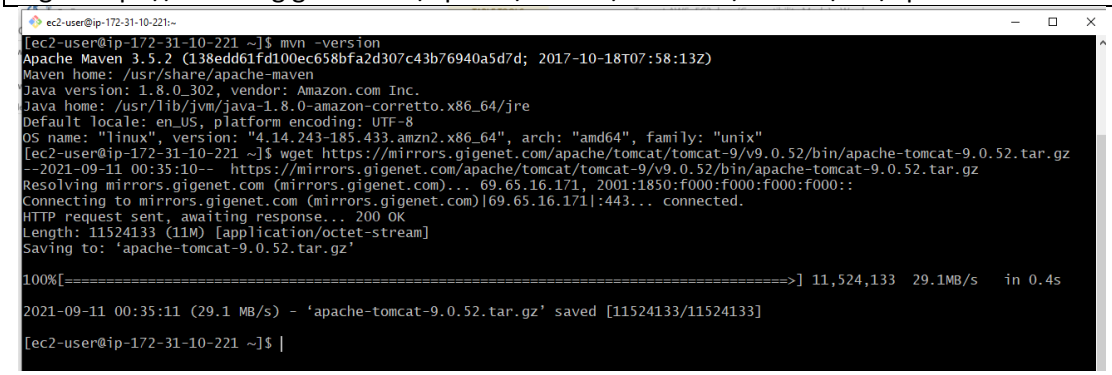
Need: the apache-tomcat-9.0.52.tar.gz file

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

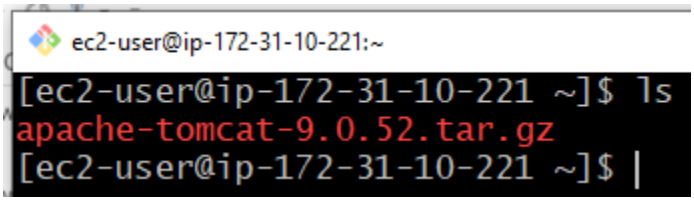


Run the wget command

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



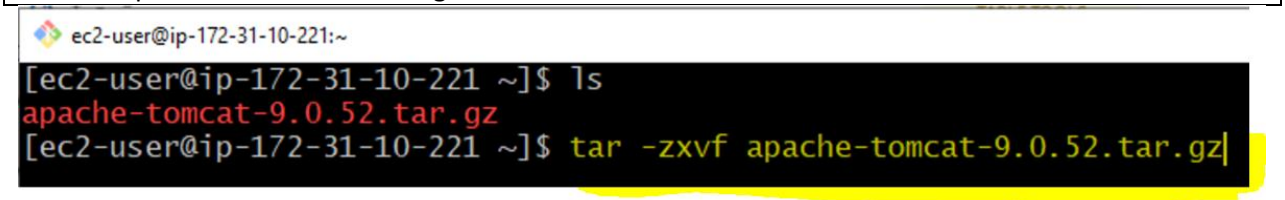
Verify file was copied, do an ls command



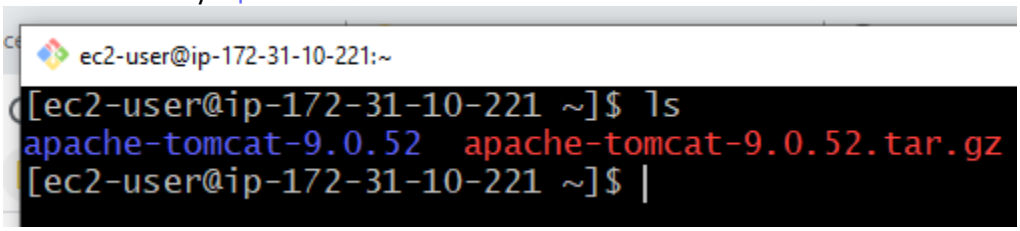
Extract the tar file

Type the command and press enter

tar -zxvf apache-tomcat-9.0.52.tar.gz



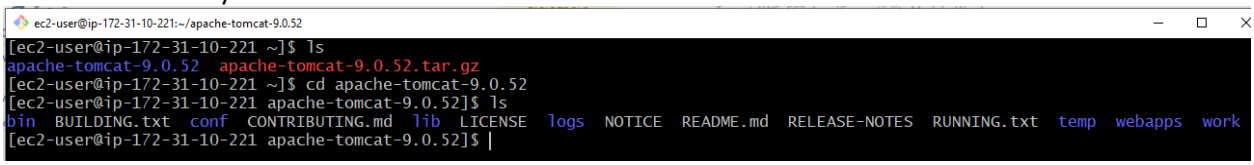
Creates directory: apache-tomcat-9.0.52



Look at the apache-tomcat-9.0.52 directory

```
cd apache-tomcat-9.0.52
ls
```

Has same directory structure as an installation on a local machine



The installation of required software on the EC2 instance is completed.

Installing a war file on the EC2 instance

Copy a project war file from a local machine

Will be using scp commands for more info on scp visit:

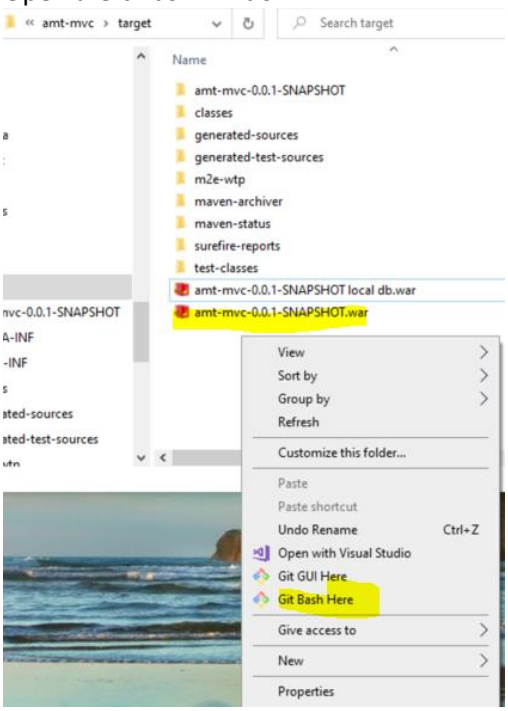
https://linuxize.com/post/how-to-use-scp-command-to-securely-transfer-files/

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

On the local machine

Use Windows explorer to navigate to the directory where the war file is located

Open a Git Bash window



Do an ls command to locate the war file

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

You might want to rename the file with Window explorer prior to pushing the file

Push the war file to the EC2 server

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

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

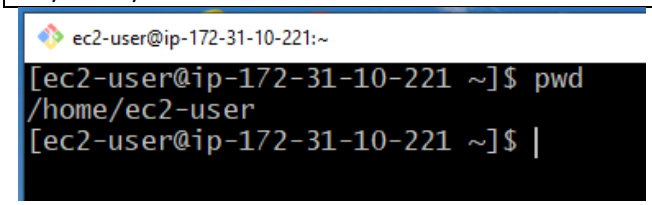
The scp command will use the configuration information in the config file in the .ssh folder on the local machine.

```
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
```

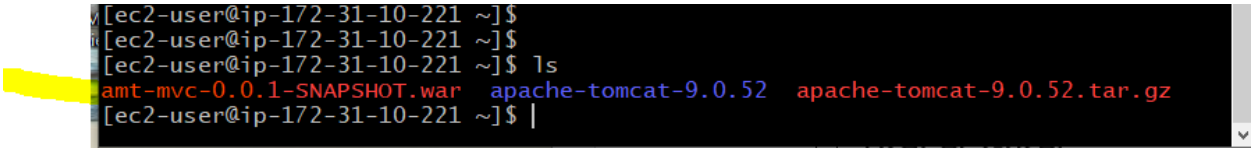
In the EC2 window

Move to the /home/ec2-user directory

```
cd /home/ec2-user
```

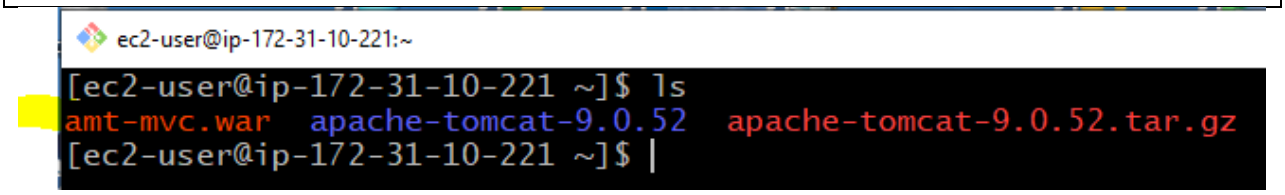


Verify file is on server with ls command



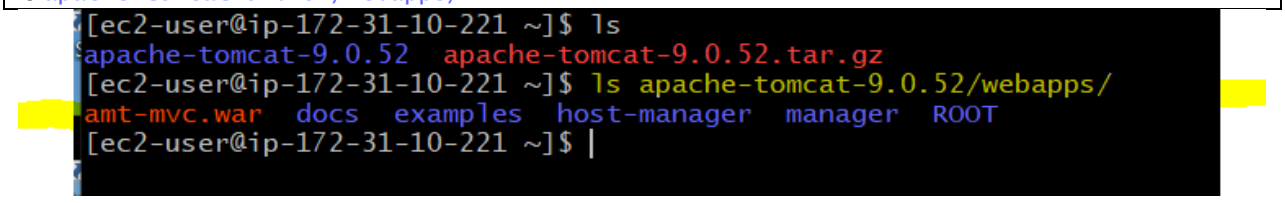
Rename file with mv command:

```
mv amt-mvc-0.0.1-SNAPSHOT.war amt-mvc.war
```



Move the file to the apache directory

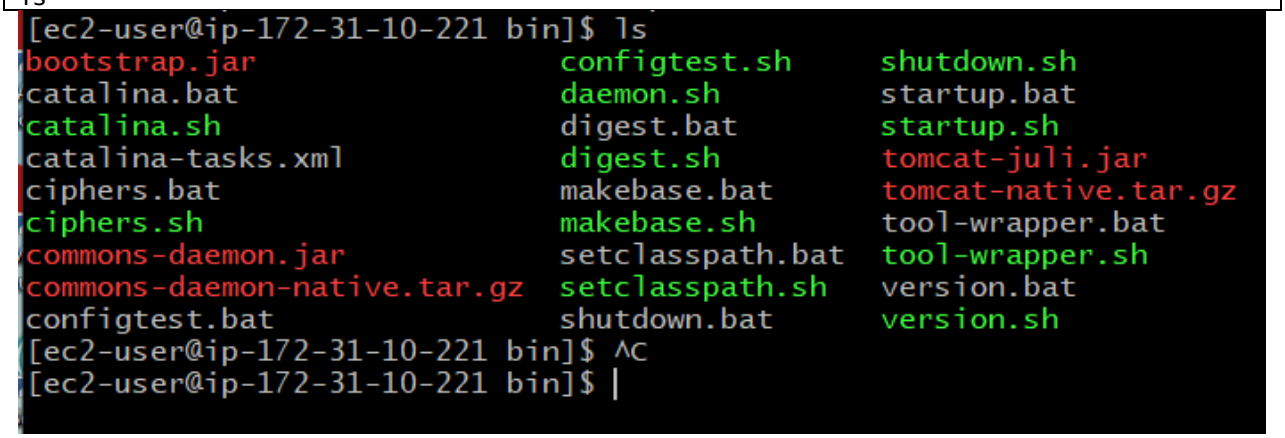
```
mv amt-mvc.war apache-tomcat-9.0.52/webapps/
ls apache-tomcat-9.0.52/webapps/
```



Start the EC2 Tomcat Server

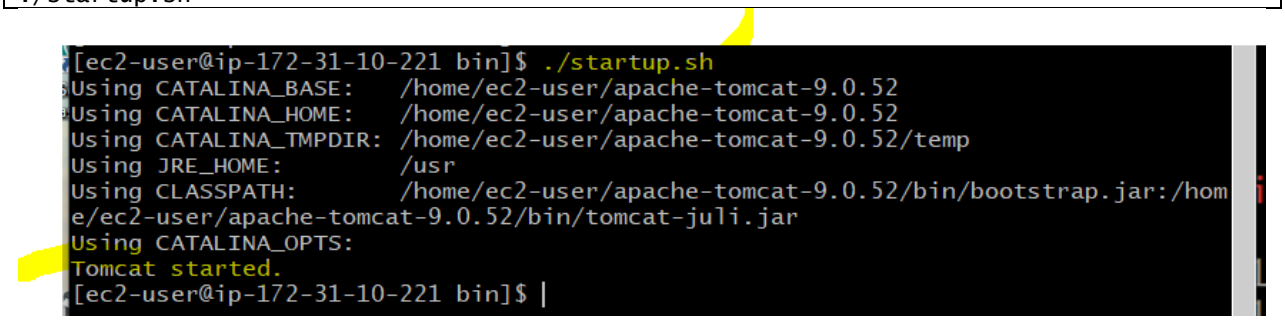
Go to apache bin directory

```
cd apache-tomcat-9.0.52/bin
ls
```



Run the startup script

```
./startup.sh
```



Shutdown Script

Do not stop it at this time

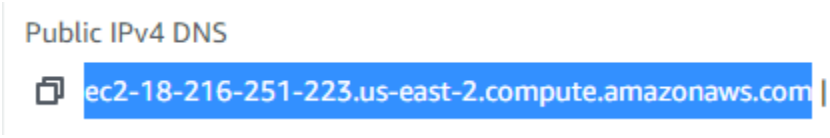
In the bin directory

To stop the Tomcat use:

```
./shutdown.sh
```

Test the Project Deployed on the EC2 Instance

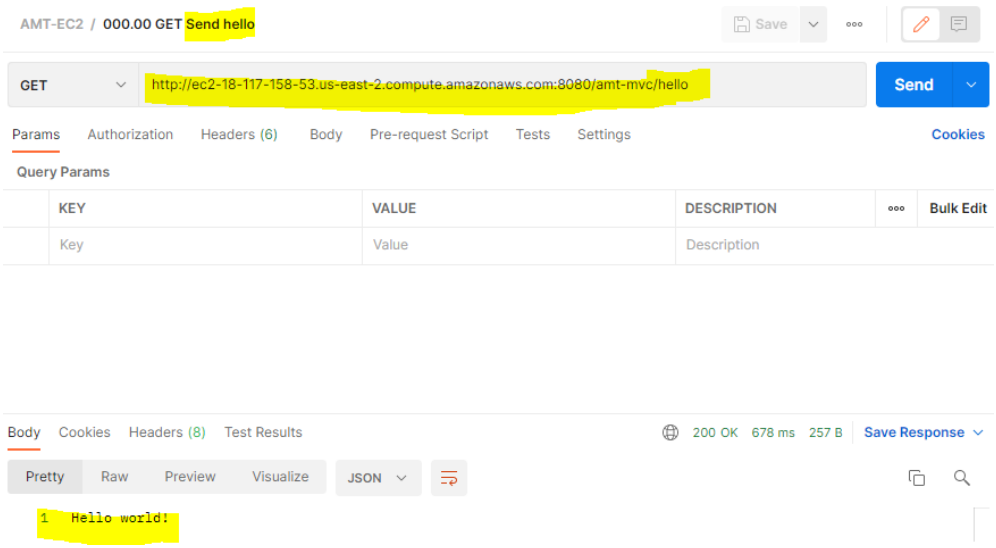
Now try and send in a request.
You will need your instance's public IPv4
This should have been copied prior to setting up the config file in the .ssh folder



This is used in place of localhost in the connection URL for your program
For my program deployed on the EC2 instance the configuration was:

Went from:
http://localhost:8080/amt-mvc/hello
To:
http://ec2-18-117-158-53.us-east-2.compute.amazonaws.com:8080/amt-mvc/hello

My program has a very simple end point to perform initial test of the MVC application



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

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

This concludes the building of an EC2 instance and the deployment of a war file.