

Perquisite

Below software / account required for below EC2 webserver creation

- AWS Account
- WinSCP
- Ubuntu in Windows
- PuttyGen /Putty

Create EC2 Instance

Select Amazon Linux AMI 2008

The screenshot shows the AWS Management Console interface for creating an EC2 instance. The browser address bar indicates the URL: <https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard>. The console header shows the user is logged in as 'ChirodipPal' with account ID '6280-3053-0634' and location 'Ohio'. The navigation bar includes 'Services', 'Resource Groups', and a star icon. The main content area is titled 'Step 1: Choose an Amazon Machine Image (AMI)' and includes a search bar with the placeholder text 'Search for an AMI by entering a search term e.g. "Windows"'. Below the search bar, there is a 'Quick Start' sidebar on the left with options: 'My AMIs', 'AWS Marketplace', 'Community AMIs', and a 'Free tier only' filter. The main list of AMIs shows three options: 'Amazon Linux 2 AMI (HVM), SSD Volume Type', 'Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type', and 'Red Hat Enterprise Linux 8 (HVM), SSD Volume Type'. Each AMI entry includes a description, root device type, virtualization type, and a 'Select' button. The 'Amazon Linux 2 AMI' is highlighted with a 'Free tier eligible' badge. The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 11:18 AM on 4/30/2020.

Select t2.micro

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: **All instance types** **Current generation** [Show/Hide Columns](#)

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Instance Details](#)

Select Default VPC

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances [Launch into Auto Scaling Group](#)

Purchasing option ☐ Request Spot instances

Network [Create new VPC](#)

Subnet [Create new subnet](#)

Auto-assign Public IP

Placement group ☐ Add instance to placement group

Capacity Reservation [Create new Capacity Reservation](#)

IAM role [Create new IAM role](#)

Shutdown behavior

Stop - Hibernate behavior ☐ Enable hibernation as an additional stop behavior

Enable termination protection ☐ Protect against accidental termination

Monitoring ☐ Enable CloudWatch detailed monitoring

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

Security group

Open 8080 port

Jenkins - The Complete Tutorial | x Launch instance wizard | EC2 M... x +

← → ↻ 🔒 https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard: 🔍 ☆ 📌 🗑️ 👤 ⋮

aws Services Resource Groups 🔖 🔔 ChirodipPal @ 6280-3053-0634 Ohio Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group
☐ Select an existing security group

Security group name:
Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP	TCP	8080	Custom 0.0.0.0, ::/0	e.g. SSH for Admin Desktop

Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Feedback English (US) © 2006 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Type here to search

Review and Create

Select new or existing Key value Pem file

Jenkins - The Complete Tutorial | x Launch instance wizard | EC2 M... x +

← → ↻ 🔒 https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard: 🔍 ☆ 📌 🗑️ 👤 ⋮

aws Services Resource Groups 🔖 🔔 ChirodipPal @ 6280-3053-0634 Ohio Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

AMI Details [Edit AMI](#)

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-097834fcb3081f51a
Free tier eligible
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups [Edit security groups](#)

Security group name: launch-wizard-4
Description: launch-wizard-4 created 2020-04-30T11:20:55.456+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
Custom TCP Rule	TCP	8080	0.0.0.0/0	

Feedback English (US) © 2006 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Type here to search

EC2 Dashboard

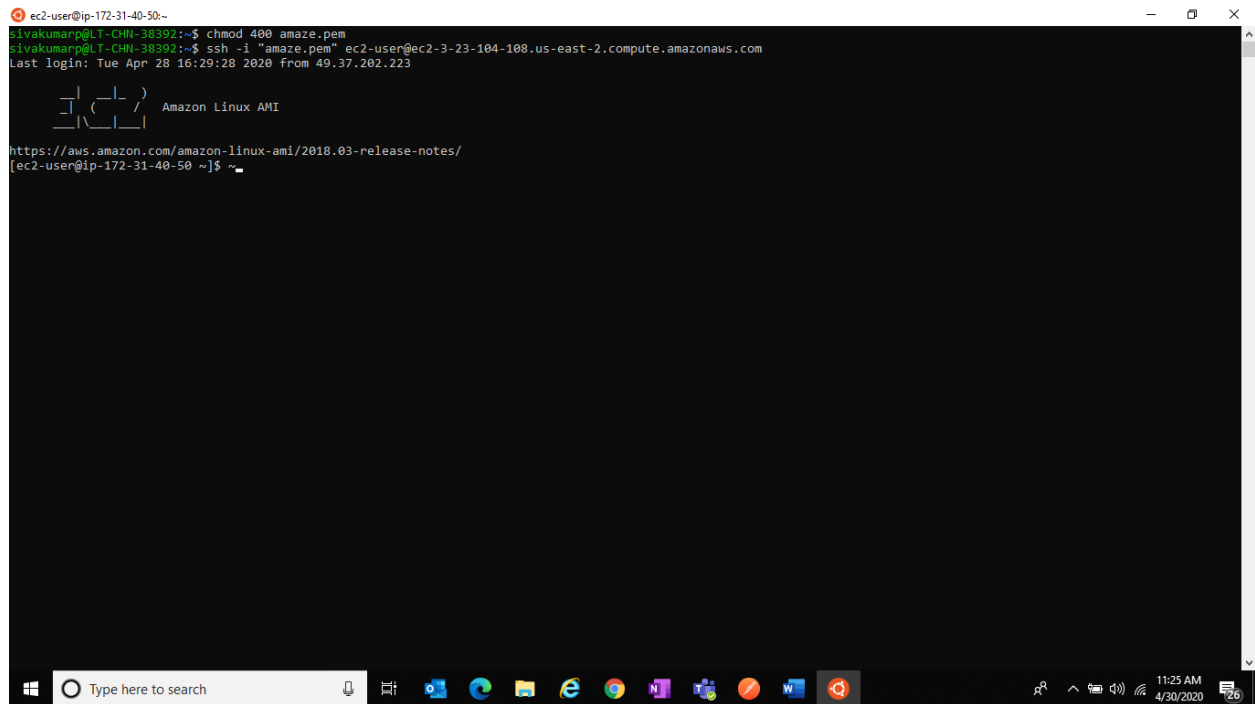
The screenshot shows the AWS Management Console for the EC2 service. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a user profile 'ChirodipPal'. The main content area displays a list of EC2 instances. The first instance, 'homeinsuran...', has ID 'i-0ab5a71551b29d1f3', type 't2.micro', and state 'running'. Below the list, a detailed view for this instance is shown, including its description, status checks, monitoring, and tags. A tooltip is visible over the 'Usage operation' field, stating: 'The operation of the Amazon EC2 Instance and the billing code that is associated with the AMI. Usage operation corresponds to the Instance/Operation column on your AWS Cost and Usage Report (CUR) and in the AWS Price List API. Learn more'. The left sidebar contains navigation links for 'New EC2 Experience', 'Events', 'Tags', 'Reports', 'Limits', 'INSTANCES', 'IMAGES', and 'ELASTIC BLOCK STORE'. The bottom of the screen shows a Windows taskbar with various application icons and the system clock at 11:22 AM on 4/30/2020.

Connect SSH

This screenshot shows the 'Connect to your instance' dialog box in the AWS Management Console. The dialog provides instructions on how to access the instance using SSH. It lists three connection methods: 'A standalone SSH client', 'Session Manager', and 'EC2 Instance Connect (browser-based SSH connection)'. The 'To access your instance:' section includes a four-step guide: 1. Open an SSH client (mentioning PuTTY), 2. Locate the private key file (amaze.pem), 3. Use the chmod command to set permissions (chmod 400 amaze.pem), and 4. Connect to the instance using its Public DNS (ec2-3-23-104-108.us-east-2.compute.amazonaws.com). An example command is provided: `ssh -i "amaze.pem" ec2-user@ec2-3-23-104-108.us-east-2.compute.amazonaws.com`. A note mentions that the username may vary and users should read the AMI usage instructions. A 'Close' button is at the bottom right. The background shows the same EC2 instance list as the previous screenshot, with the system clock at 11:23 AM on 4/30/2020.

Login SSH

Copy your EC2 pem file to your ubuntu folder

A screenshot of a terminal window showing an SSH session. The prompt is 'ec2-user@ip-172-31-40-50:~'. The user enters 'sivakumar@PELT-CMW-38392:~\$ chmod 400 amaze.pem'. Then they enter 'sivakumar@PELT-CMW-38392:~\$ ssh -i "amaze.pem" ec2-user@ec2-3-23-104-108.us-east-2.compute.amazonaws.com'. The terminal shows the login banner for Amazon Linux AMI, including the logo and the URL 'https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/'. The prompt changes to '[ec2-user@ip-172-31-40-50 ~]\$'. The terminal window has a Windows taskbar at the bottom with various application icons and a search bar.

Install Java 8 in Ubutu

Follow the below url

<https://docs.aws.amazon.com/neptune/latest/userguide/iam-auth-connect-prerq.html>

The following are instructions for installing Apache Maven and Java 8 on an Amazon EC2 instance. These are required for the Amazon Neptune Signature Version 4 authentication samples.

To Install Apache Maven and Java 8 on your EC2 instance

1. Connect to your Amazon EC2 instance with an SSH client.
2. Install Apache Maven on your EC2 instance. First, enter the following to add a repository with a Maven package.

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

Enter the following to set the version number for the packages.

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

Then you can use **yum** to install Maven.

```
sudo yum install -y apache-maven
```

3. The Gremlin libraries require Java 8. Enter the following to install Java 8 on your EC2 instance.

```
sudo yum install java-1.8.0-devel
```

4. Enter the following to set Java 8 as the default runtime on your EC2 instance.

```
sudo /usr/sbin/alternatives --config java
```

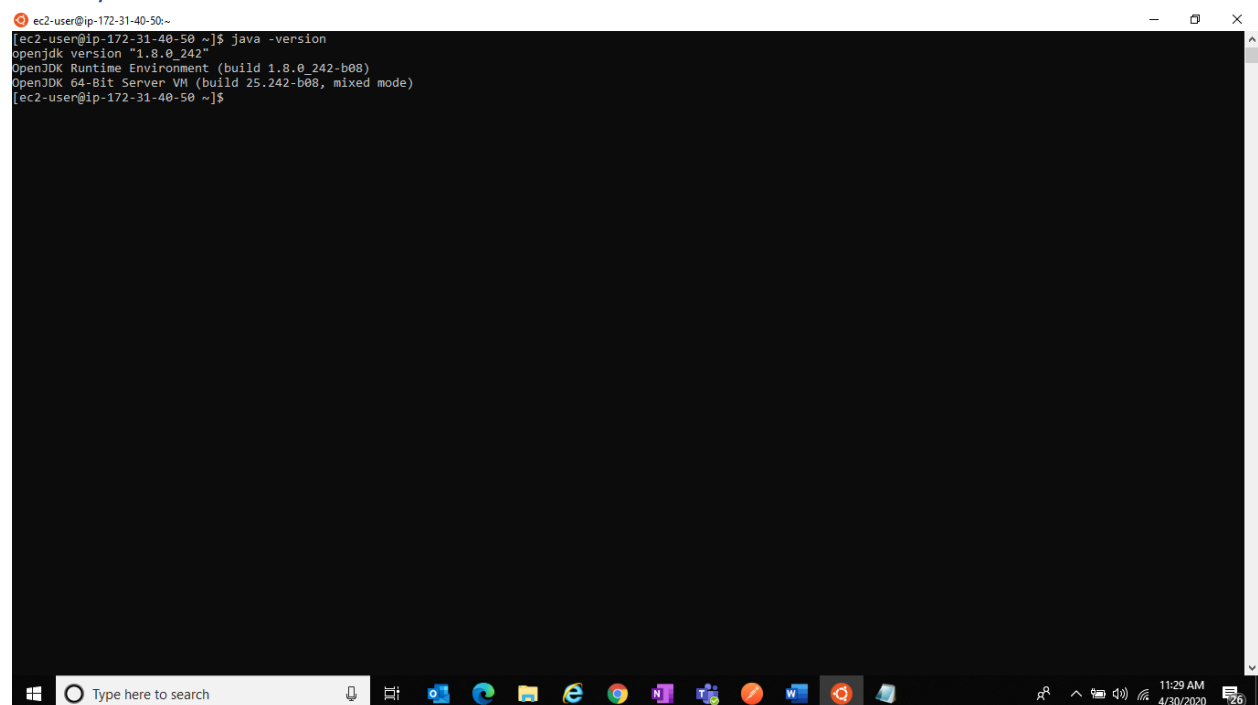
When prompted, enter the number for Java 8.

5. Enter the following to set Java 8 as the default compiler on your EC2 instance.

```
sudo /usr/sbin/alternatives --config javac
```

When prompted, enter the number for Java 8.

Verify the Java Version



```
ec2-user@ip-172-31-40-50:~$ java -version
openjdk version "1.8.0_242"
OpenJDK Runtime Environment (build 1.8.0_242-b08)
OpenJDK 64-Bit Server VM (build 25.242-b08, mixed mode)
ec2-user@ip-172-31-40-50:~$
```

Create Folder in EC2 in SSH

```
sudo chown ec2-user:ec2-user -R /opt
```

```
cd /opt/
```

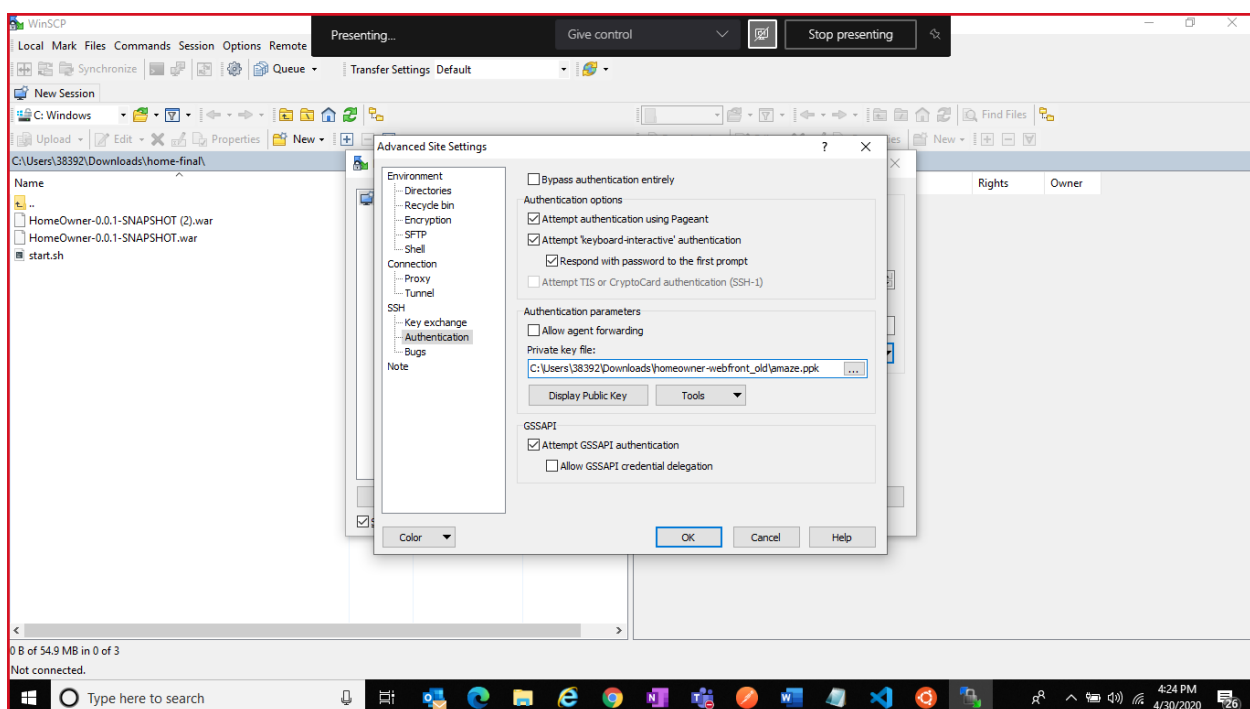
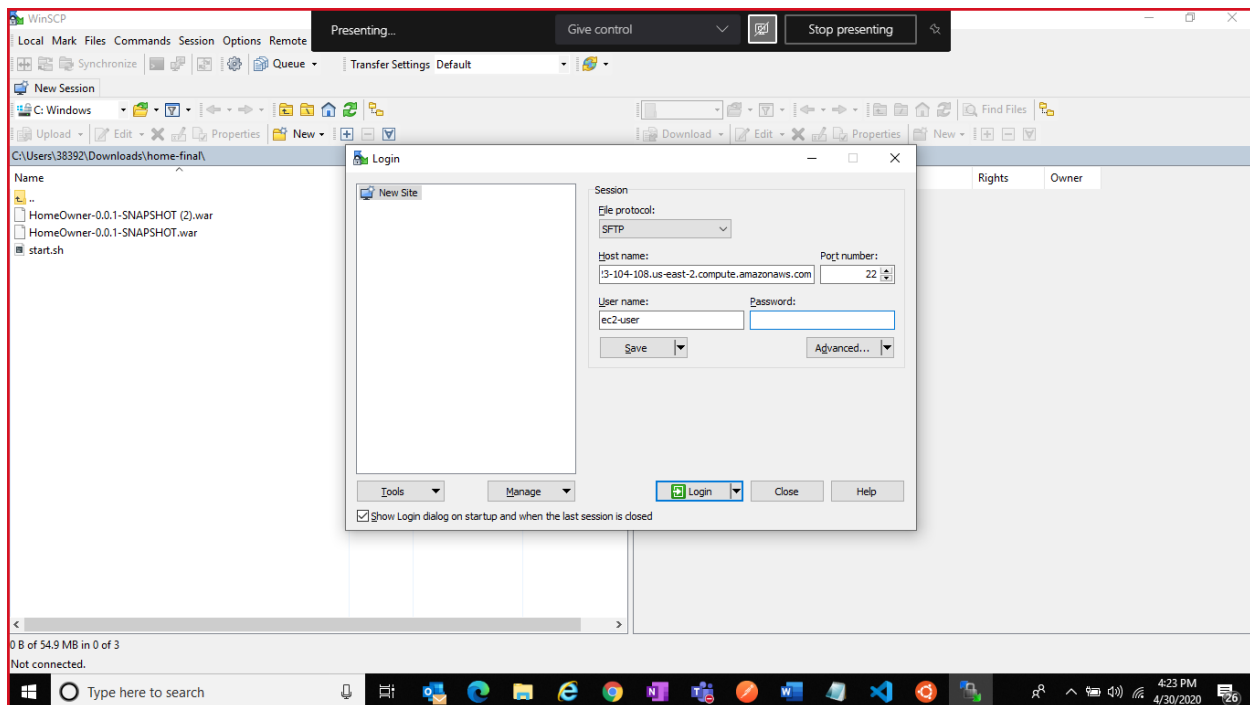
Upload project file

Upload your project file “web Folder” to EC2 “/opt/” Folder and jar folder

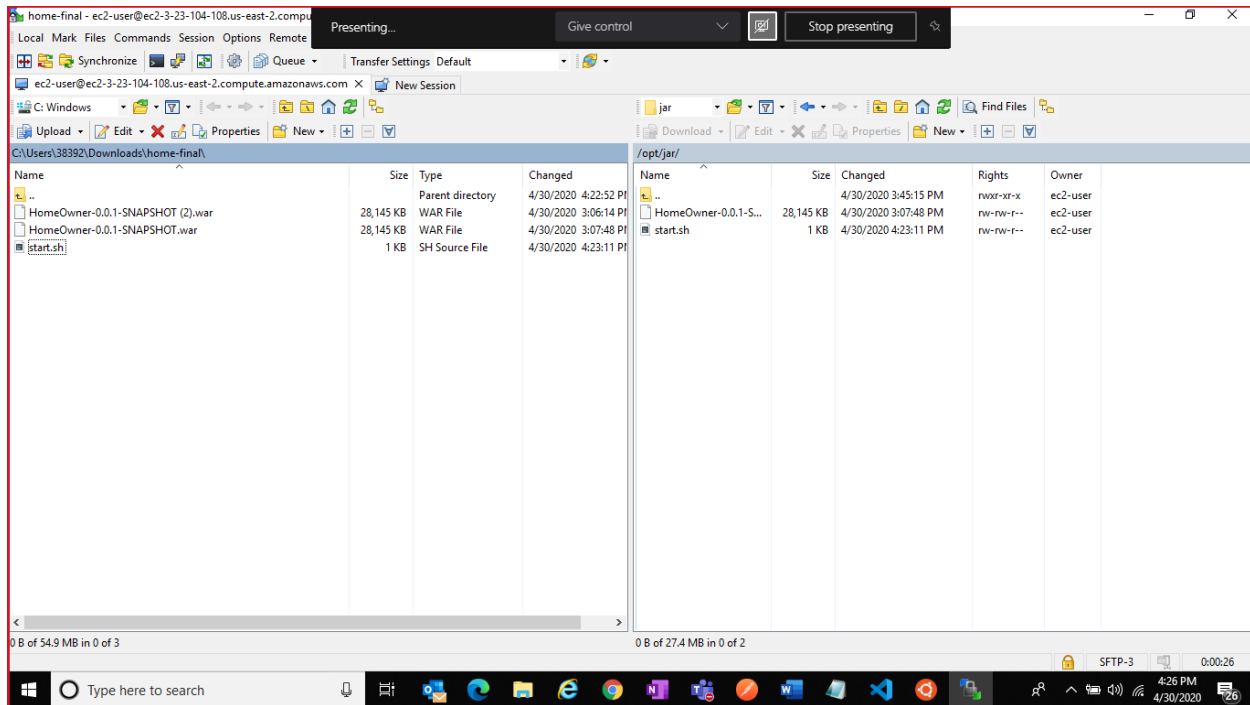
Connect WinSCP.

Using Puttygen. Exe, pegenete the ppk file

Login to SSH



Copy Jar file and start.sh file



Create background job

Deployment option with Maven Spring boot

Execute sh file

```
nohup sh start.sh &
```

List the process

```
ps -ef |grep nohup
```



```
Select ec2-user@ip-172-31-40-50:/opt/jar
[ec2-user@ip-172-31-40-50 opt]$ cd jar
[ec2-user@ip-172-31-40-50 jar]$ java -jar

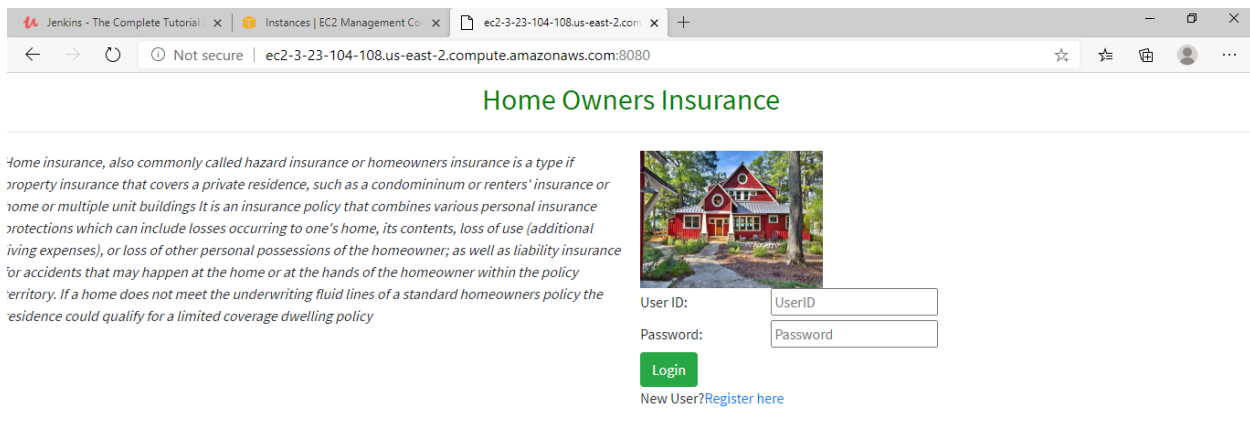
Spring
:: Spring Boot ::
(v2.1.2.RELEASE)

2020-04-30 10:50:43.343 INFO 2920 --- [main] c.perscholas.home_insurance.Application : Starting Application on ip-172-31-40-50 with PID 2920 (/opt/jar/Home
Owner-0.0.1-SNAPSHOT.war started by ec2-user in /opt/jar)
2020-04-30 10:50:43.360 INFO 2920 --- [main] c.perscholas.home_insurance.Application : No active profile set, falling back to default profiles: default
2020-04-30 10:50:47.288 INFO 2920 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2020-04-30 10:50:47.375 INFO 2920 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2020-04-30 10:50:47.376 INFO 2920 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.14]
2020-04-30 10:50:47.406 INFO 2920 --- [main] o.a.catalina.core.AprLifecycleListener : The APR based Apache Tomcat Native library which allows optimal perf
ormance in production environments was not found on the java.library.path: [/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib]
2020-04-30 10:50:49.689 INFO 2920 --- [main] org.apache.jasper.servlet.TldScanner : At least one JAR was scanned for TLDs yet contained no TLDs. Enable
debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup t
ime and JSP compilation time.
2020-04-30 10:50:49.839 INFO 2920 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2020-04-30 10:50:49.839 INFO 2920 --- [main] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 6284 ms
2020-04-30 10:50:50.526 INFO 2920 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2020-04-30 10:50:51.092 INFO 2920 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2020-04-30 10:50:51.106 INFO 2920 --- [main] c.perscholas.home_insurance.Application : Started Application in 9.368 seconds (JVM running for 10.94)

login Page - admin
admin
^C2020-04-30 10:53:02.395 INFO 2920 --- [Thread-3] o.s.s.concurrent.ThreadPoolTaskExecutor : Shutting down ExecutorService 'applicationTaskExecutor'
[ec2-user@ip-172-31-40-50 jar]$ nohup sh start.sh &
[1] 2965
[ec2-user@ip-172-31-40-50 jar]$ nohup: ignoring input and appending output to 'nohup.out'
```

Your frontend

<http://ec2-3-23-104-108.us-east-2.compute.amazonaws.com:8080/>



Copyright © 2016 Homeownersinsurance.com All Rights Reserved

