**CI/CD Deployment for Springboot Application:**

Technologies involved

* Spring Boot Application in Java (RestFul Service in Spring Container)
* Maven (Build)
* Github (Code Repository)
* AWS S3 (To store artifacts eg. .jar files)
* AWS EC2Instance
* **Representing visually what we are doing here.**

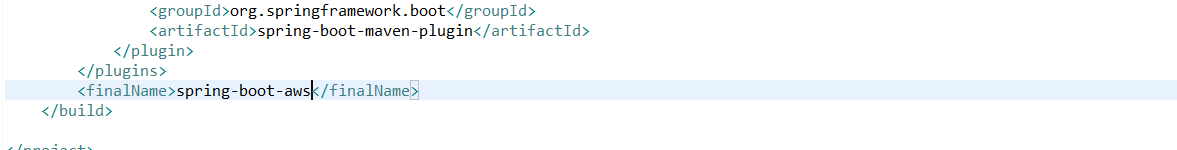


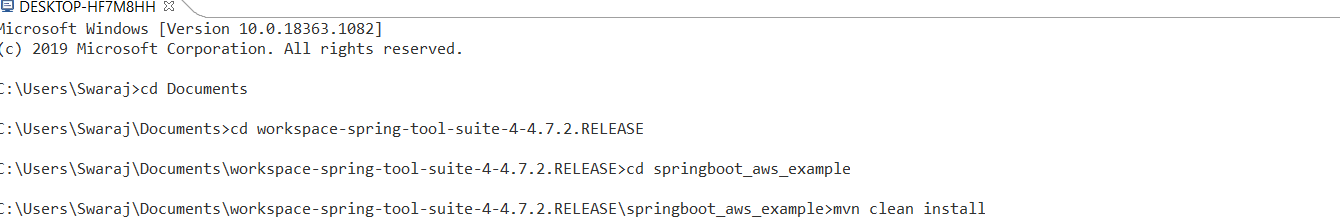
**Step 1:**

Create a simple spring boot application in IDE, eclipse or you can check out the code committed here.

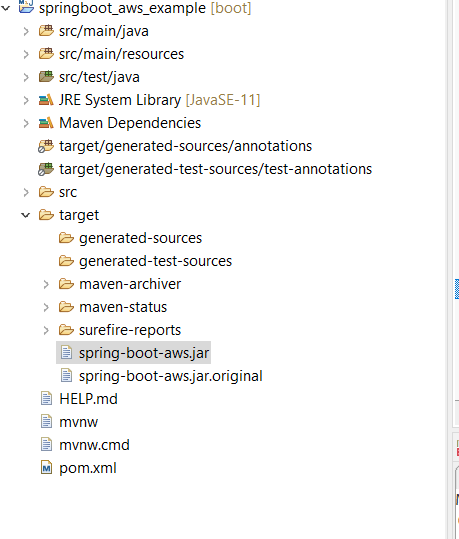
After Step 1:

* Perform maven clean build
* Run the spring boot application
* Create JAR File Using Final Name and command mvn clean install

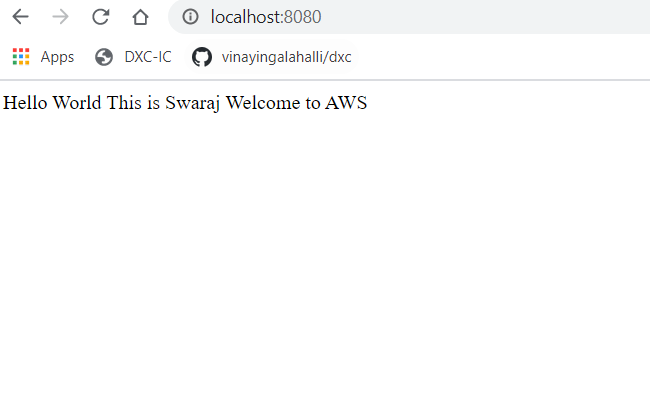




* JAR File Created

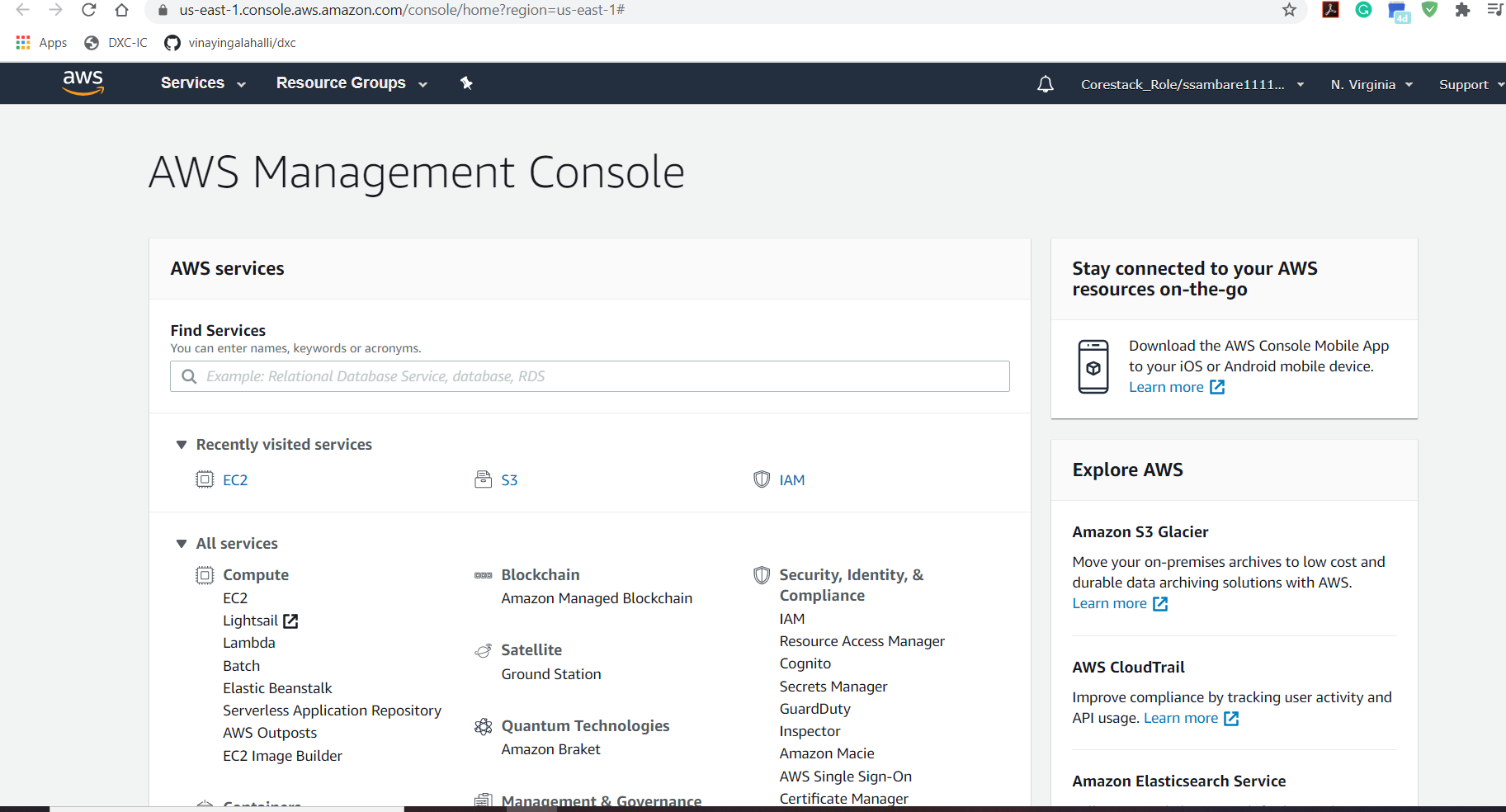


* output should be below

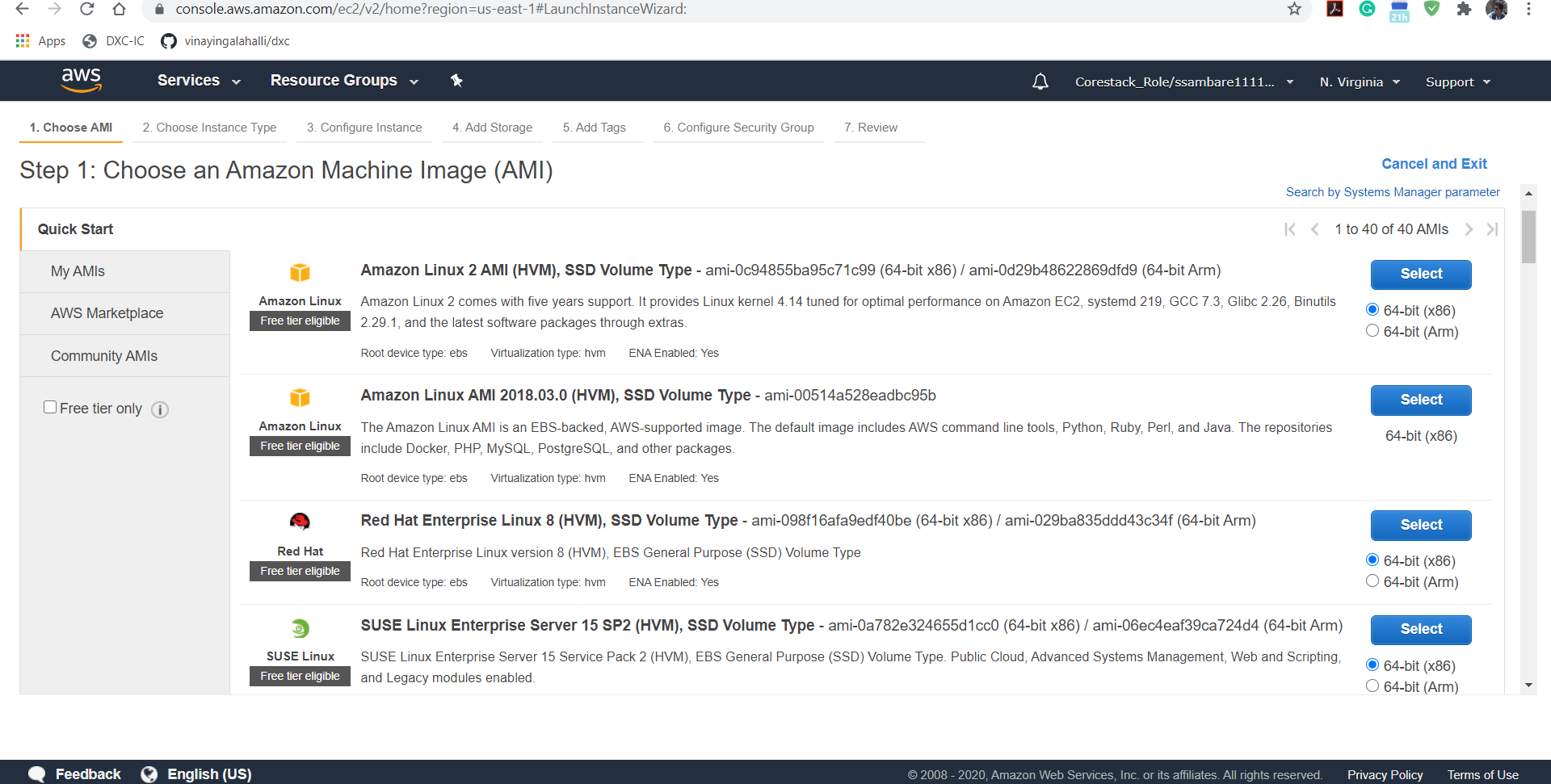


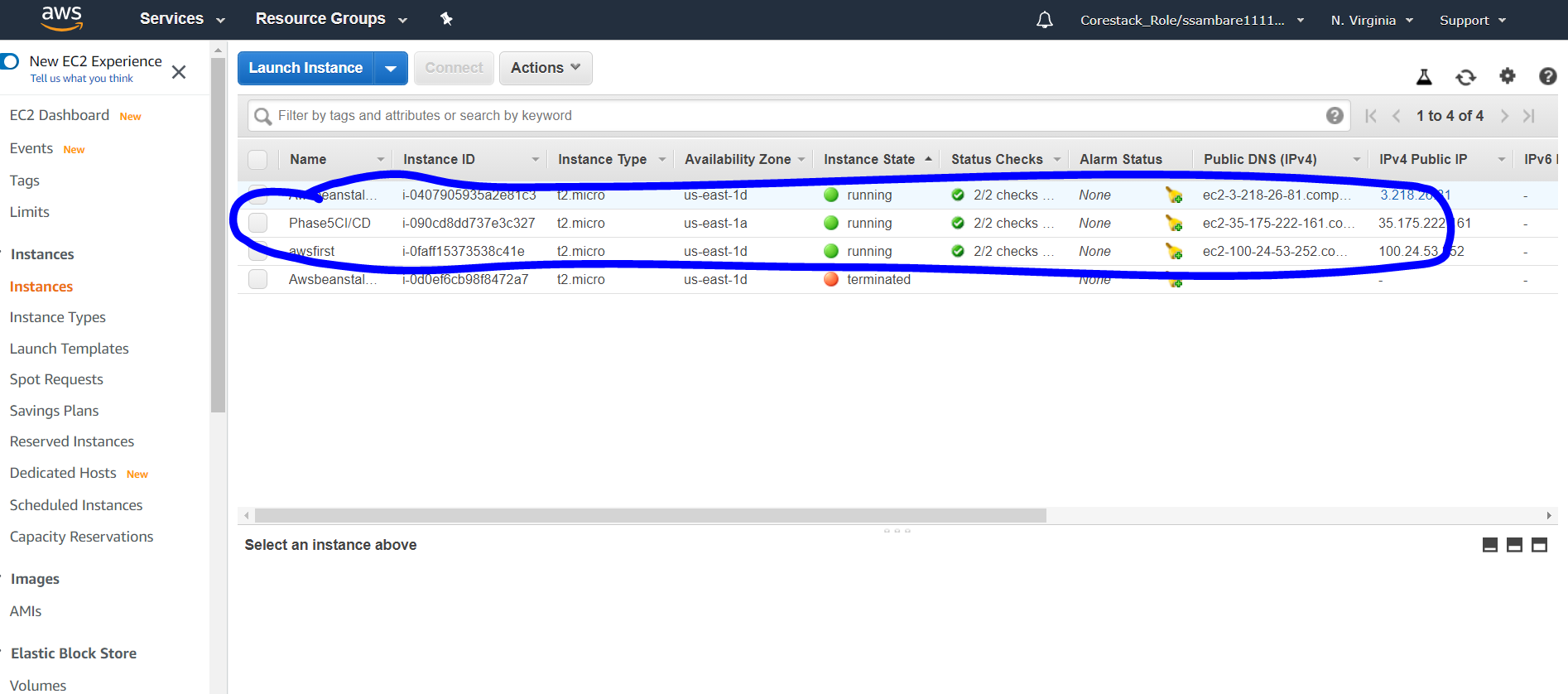
**Step 2:**

* Login to your AWS account
* Create an instance

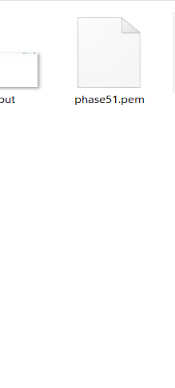


Select Amazon Linux AMI

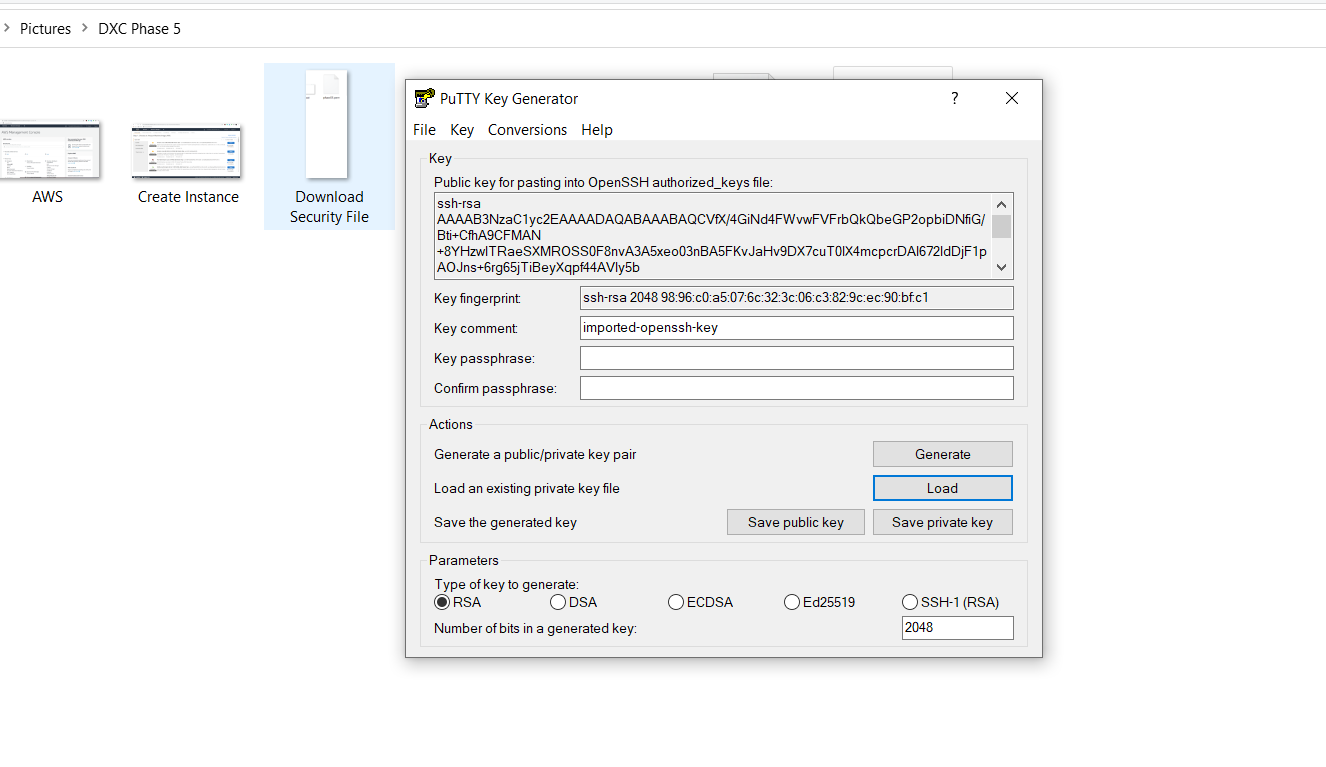
Instance Created



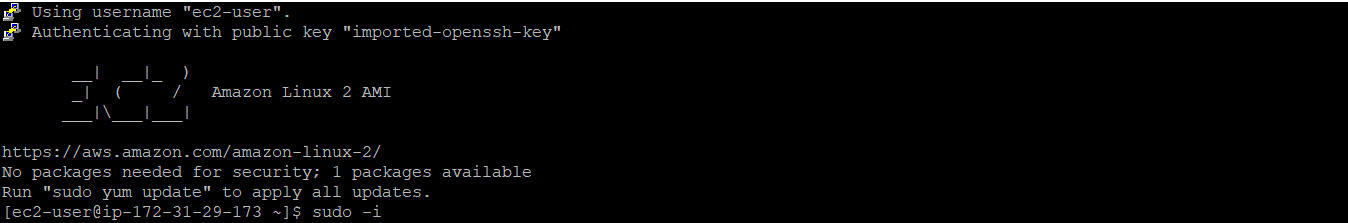
Download Security Key:



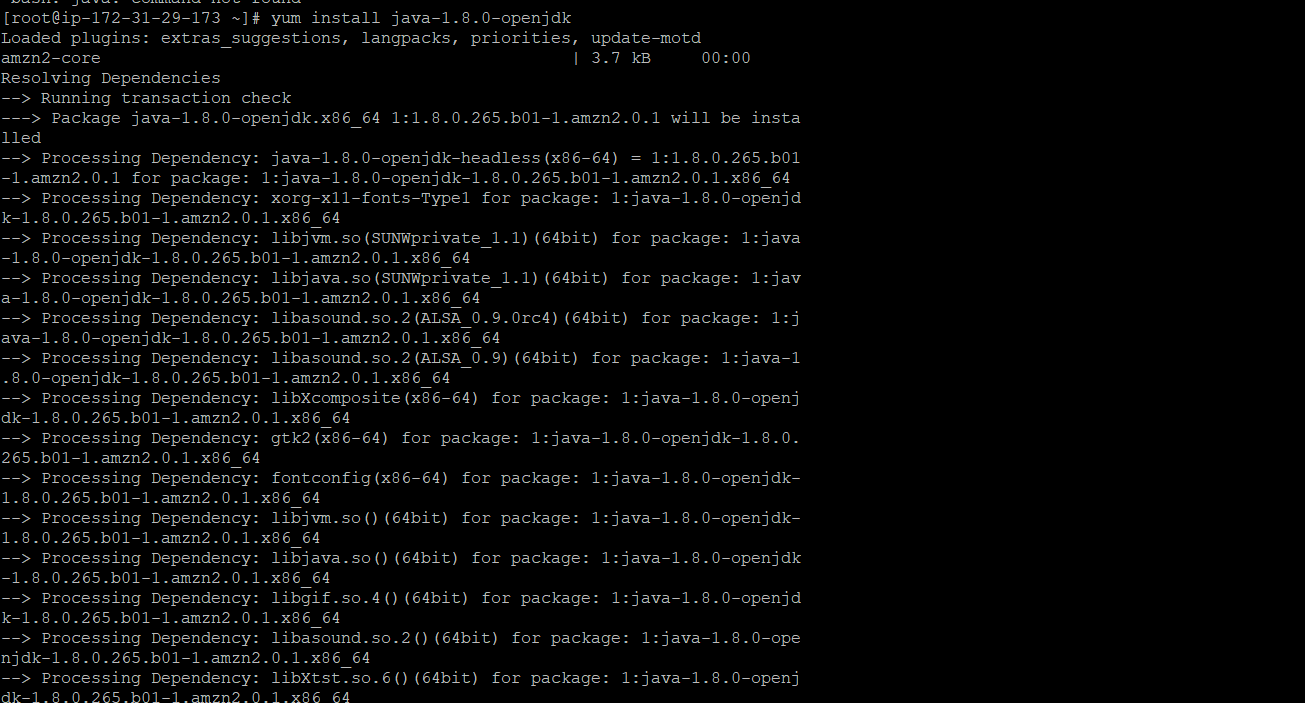
Use PuttyGen to Decode the security file:



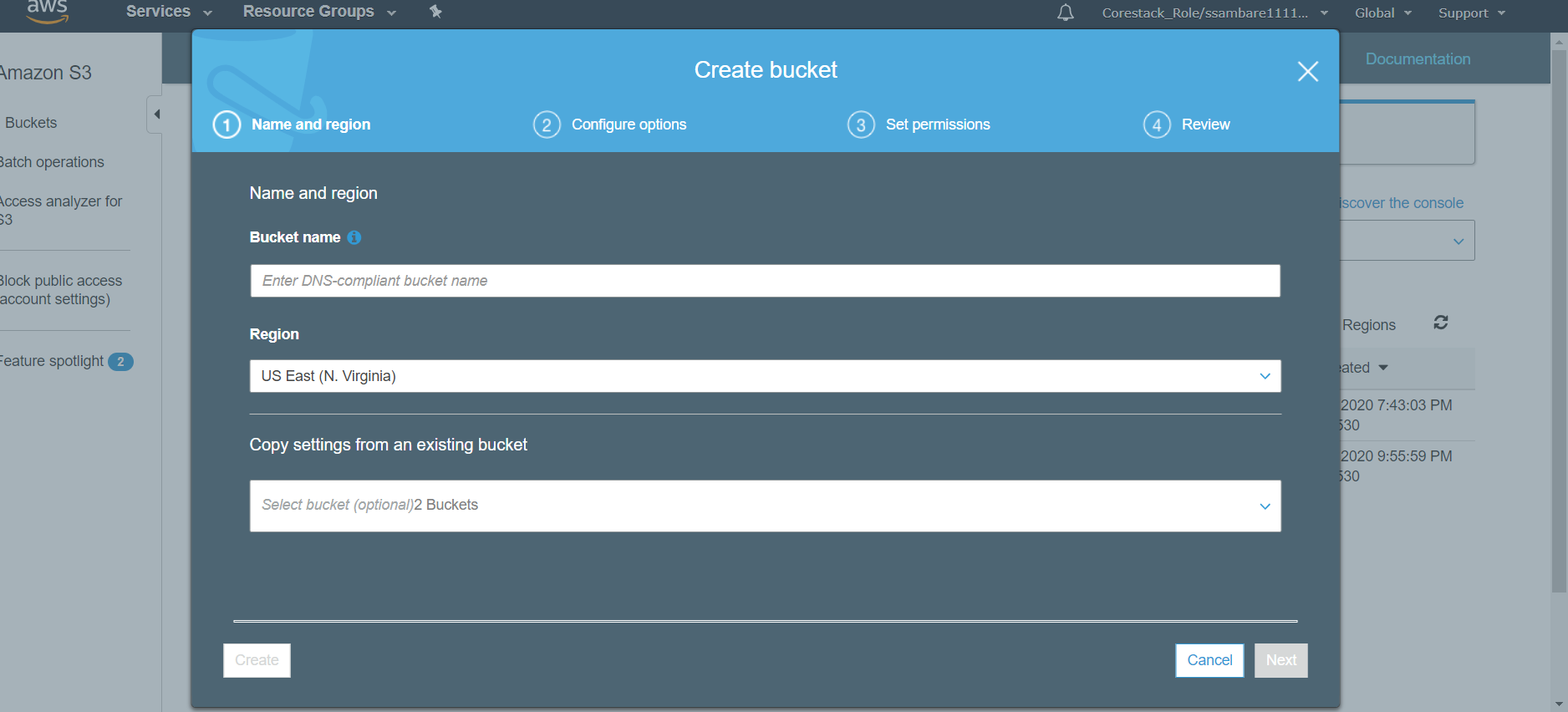
After this step Using Putty Launch the Amazon Linux:

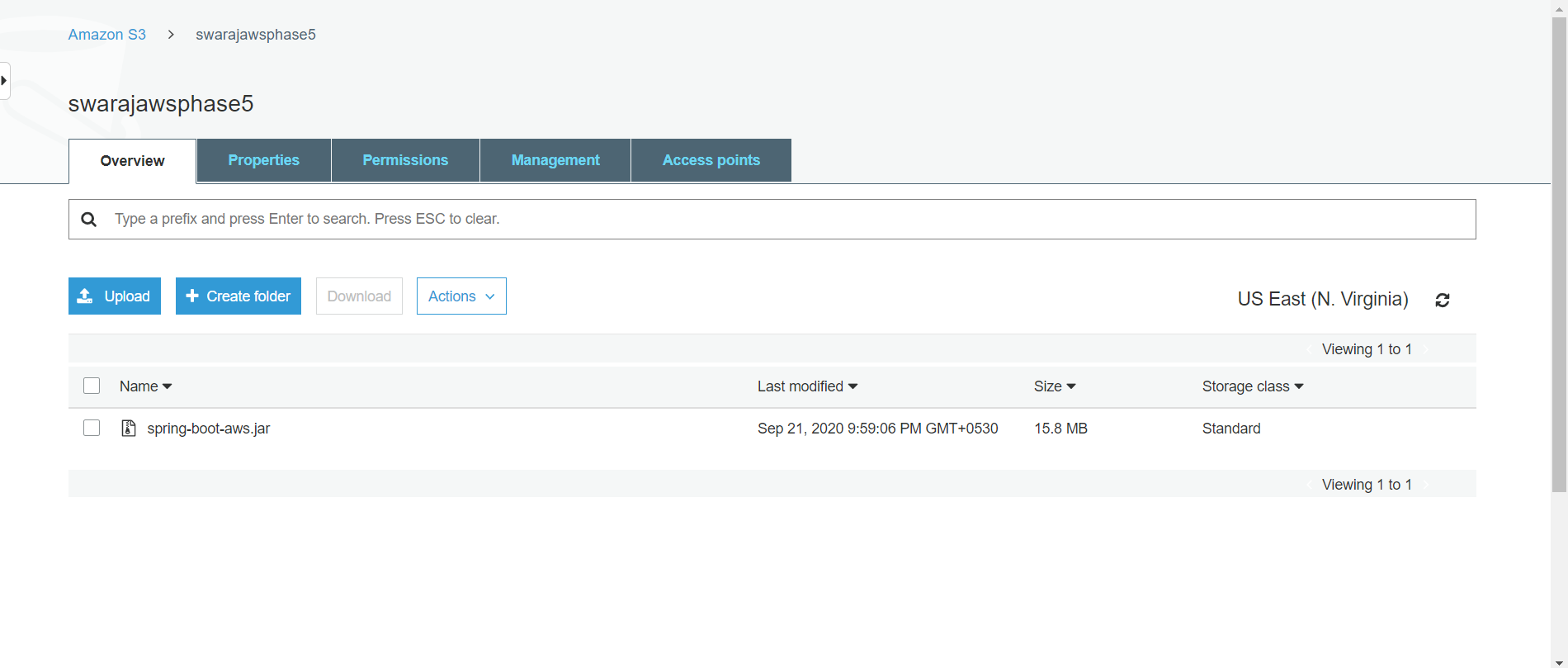


Install JDK 1.8 in the Amazon Linux 2 Instance:

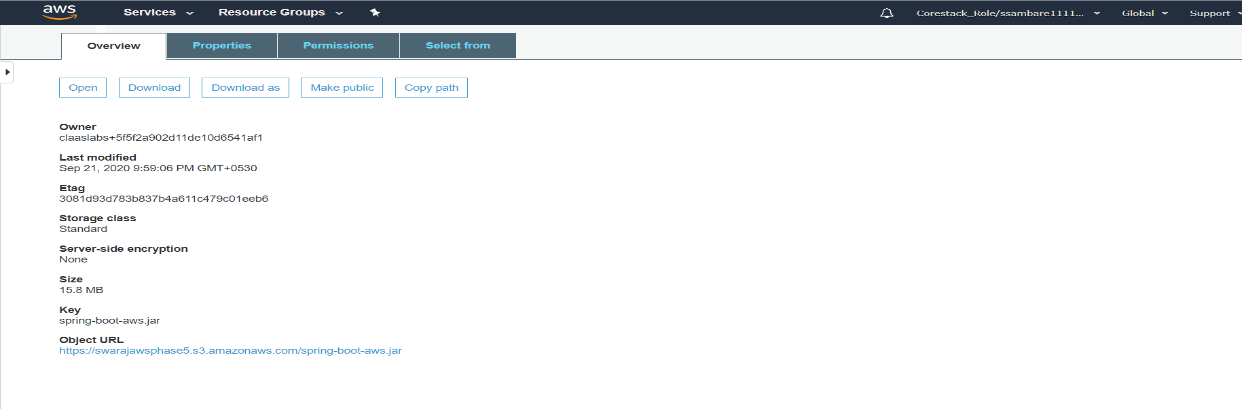


Create an S3 Bucket in AWS to add the JAR file and make that file public:

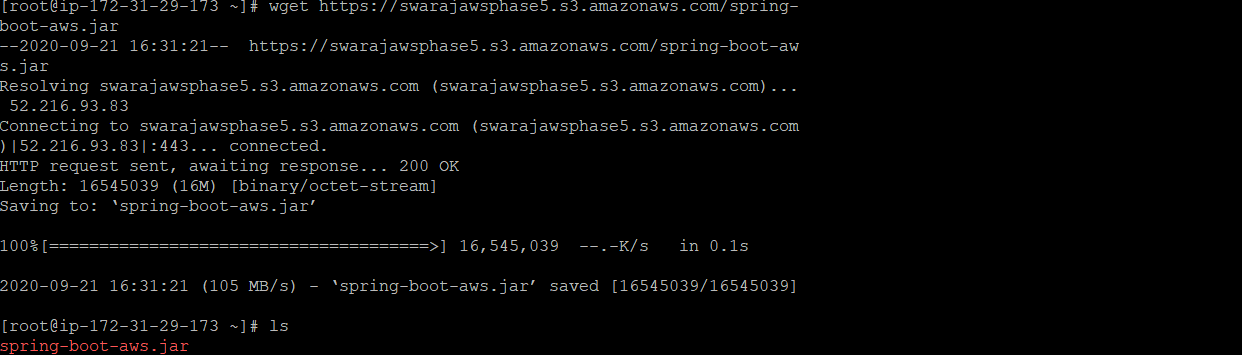




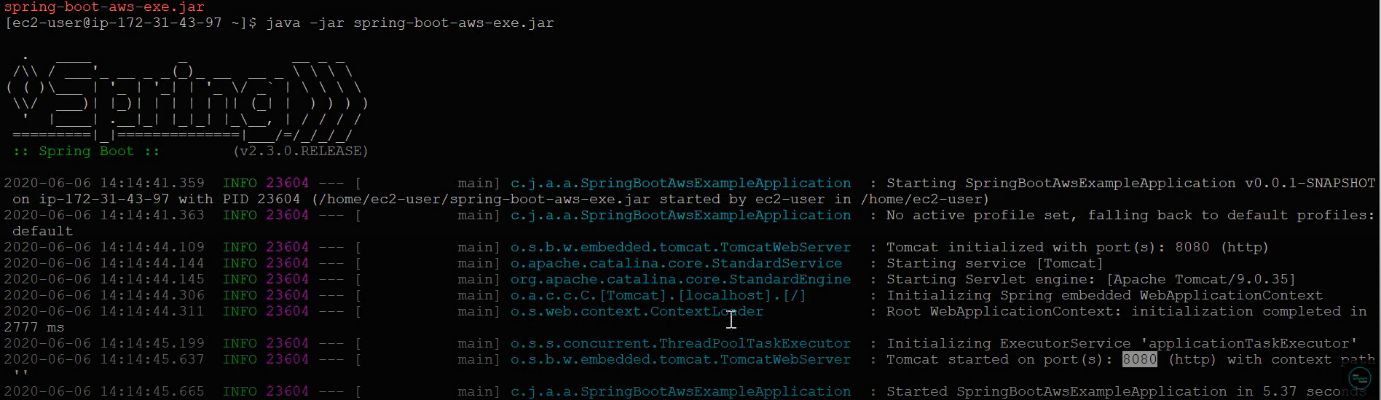
Copy URL and make it Public:



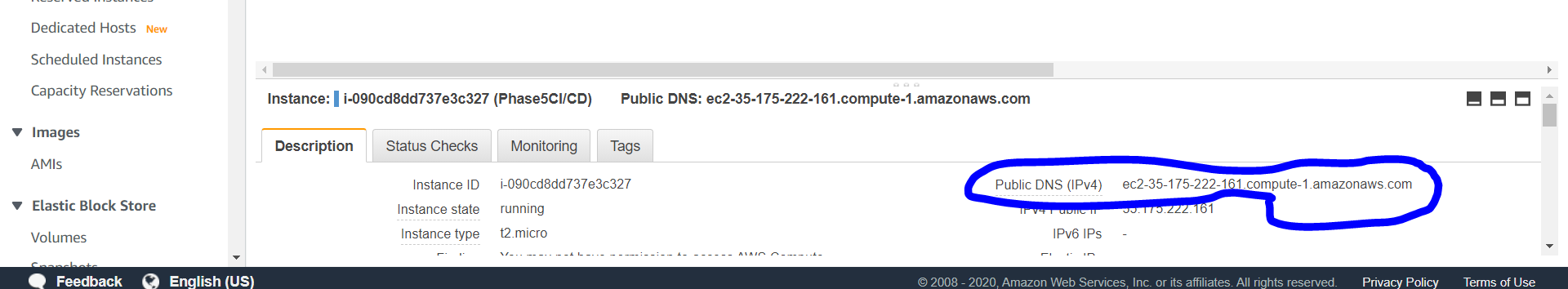
Add the JAR file in the created amazon linux instance using the link copied from S3 Bucket:



Run the JAR file using the command java -jar JARname.jar:



Copy the Public DNS from the instance created:



Whenever you run this <http://ec2-35-175-222-161.compute-1.amazonaws.com:8080/> you will get this output:

