**THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY**



**CONTINUOUS DELIVERY AND DEVOPS (LAB) SUBJECT CODE- PCS217P**

Submitted To: Submitted by:

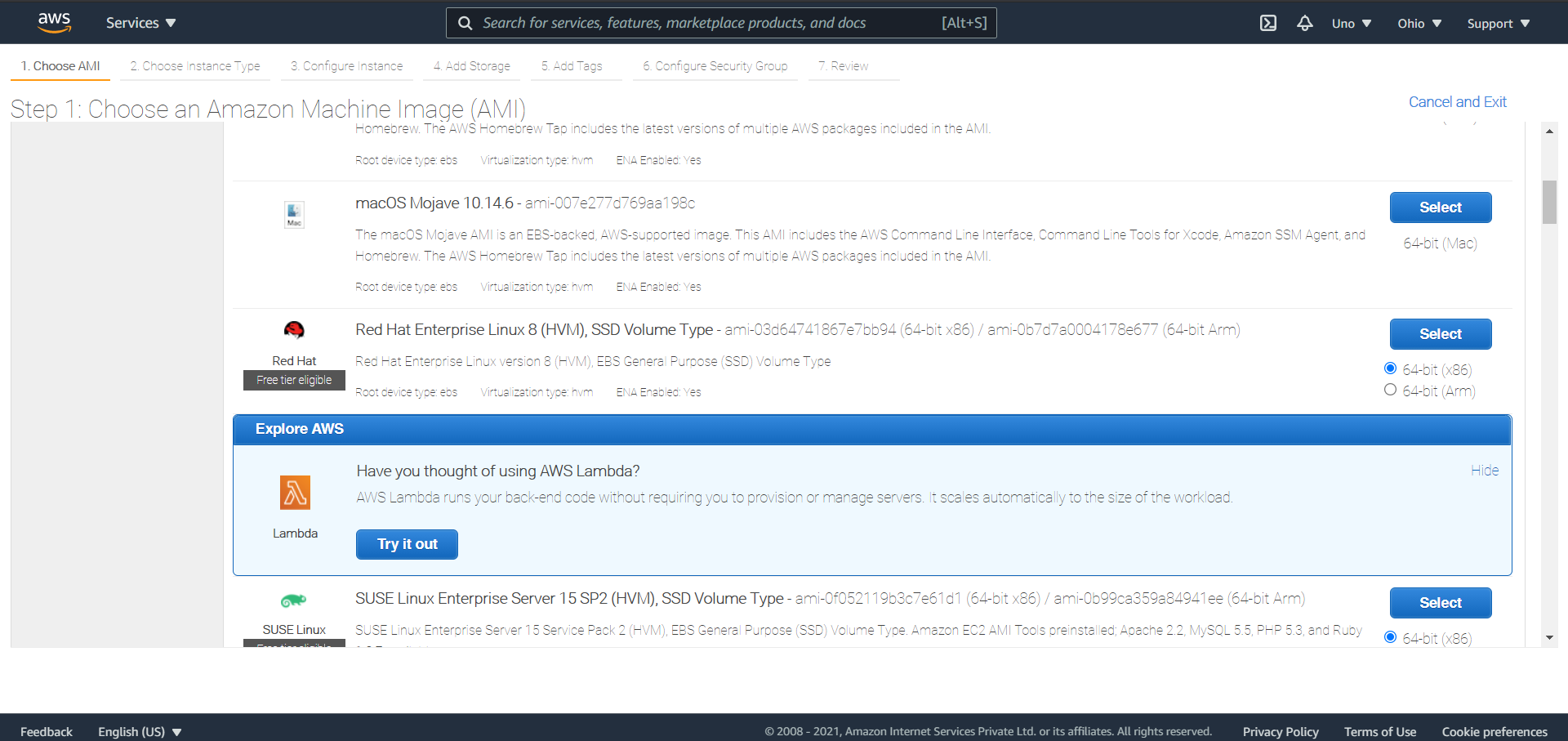
Dr. Gurpal Singh Chhabra Ujjwal Ahuja

(Assistant Professor) (802032055)

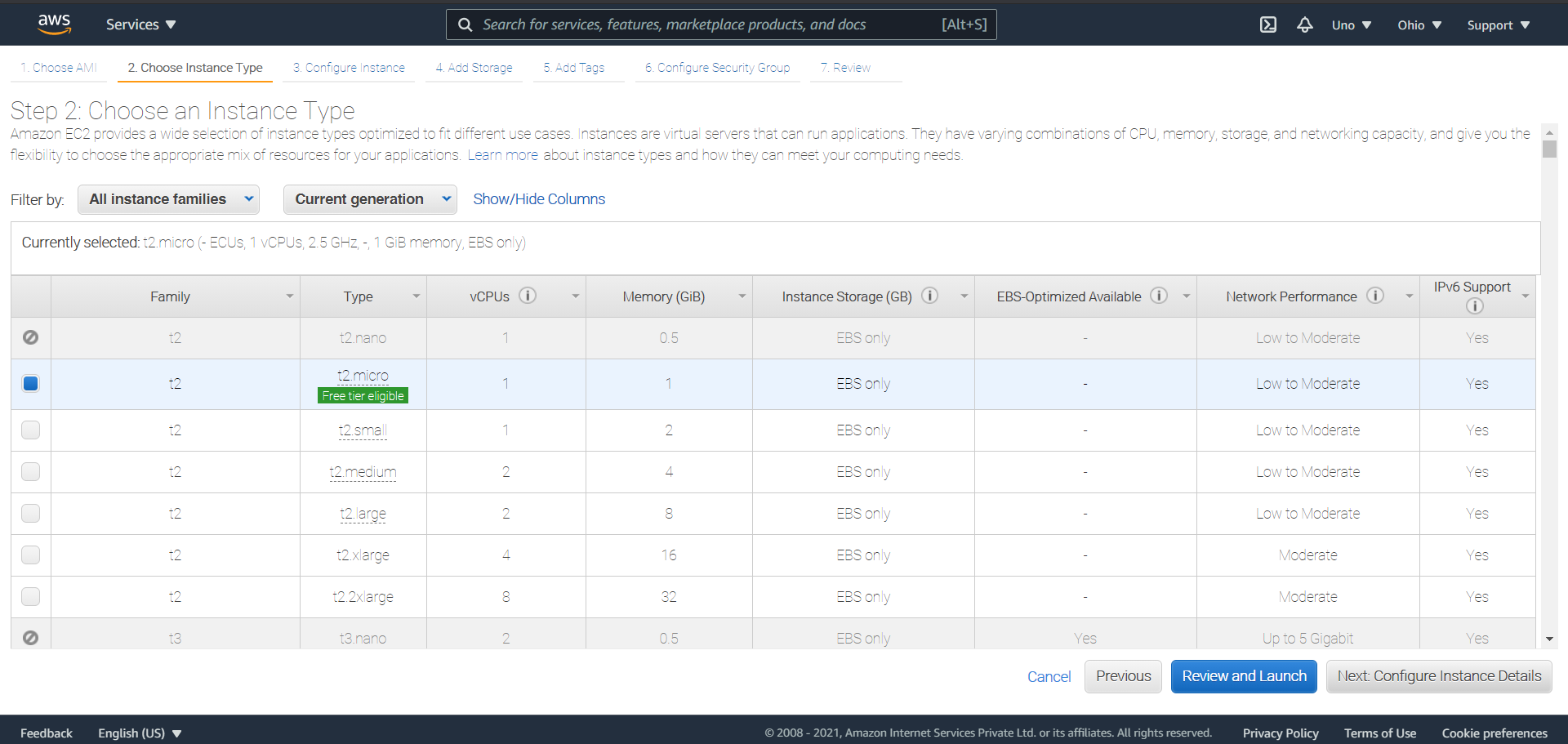
Department of Computer Science and Engineering

**Project on Continuous Integration and Continuous Development through Jenkins**

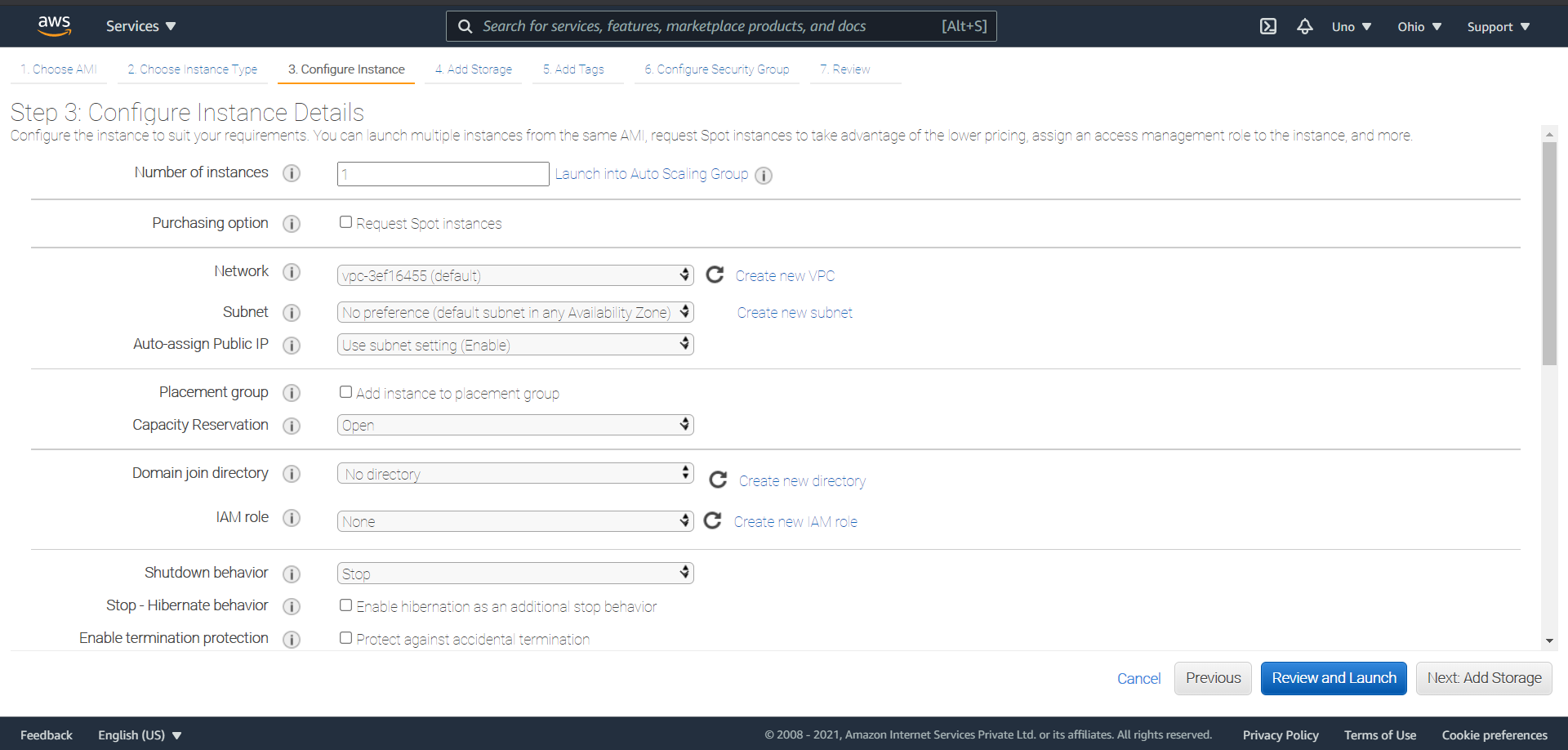
1. Create a free tier Amazon AWS account (Debit/credit card needed for a minimal transaction of Rs.2
2. Create an Amazon ec2 instance

****

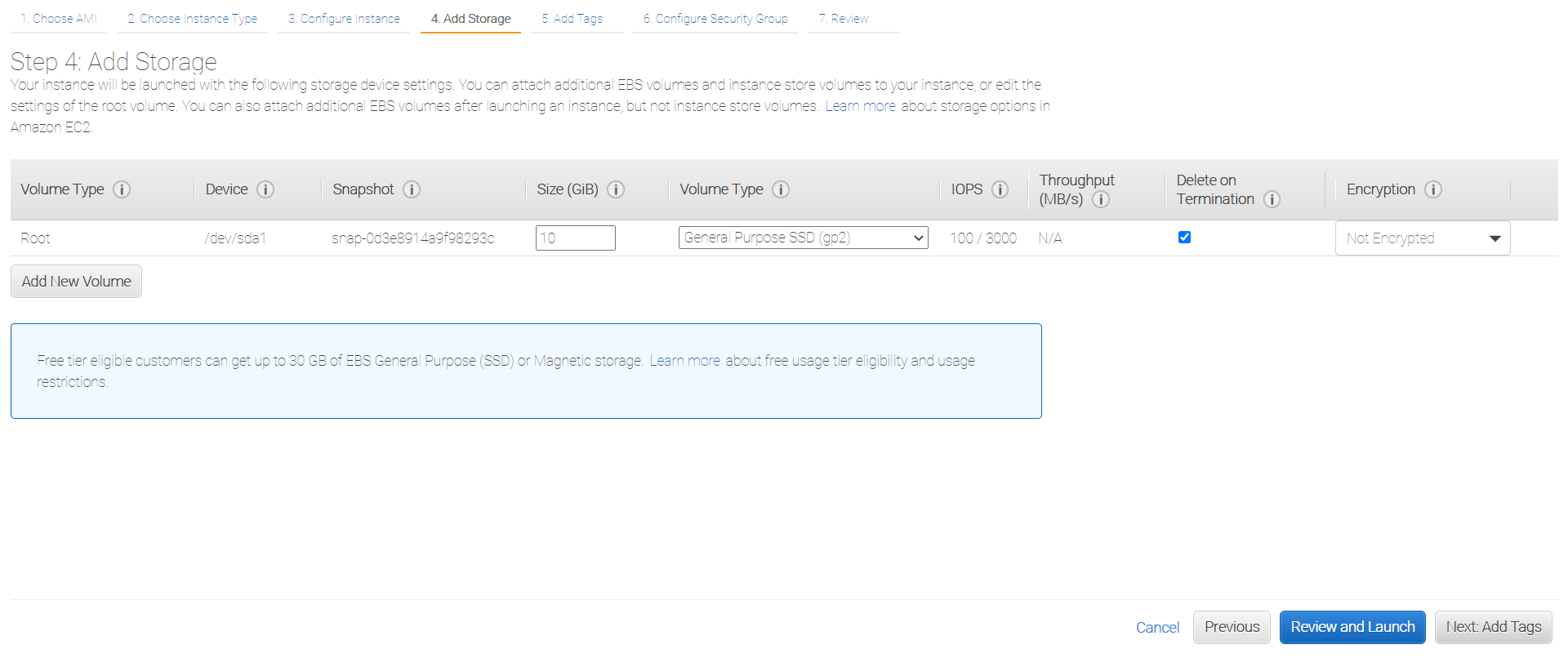
1. Choose your Instance type



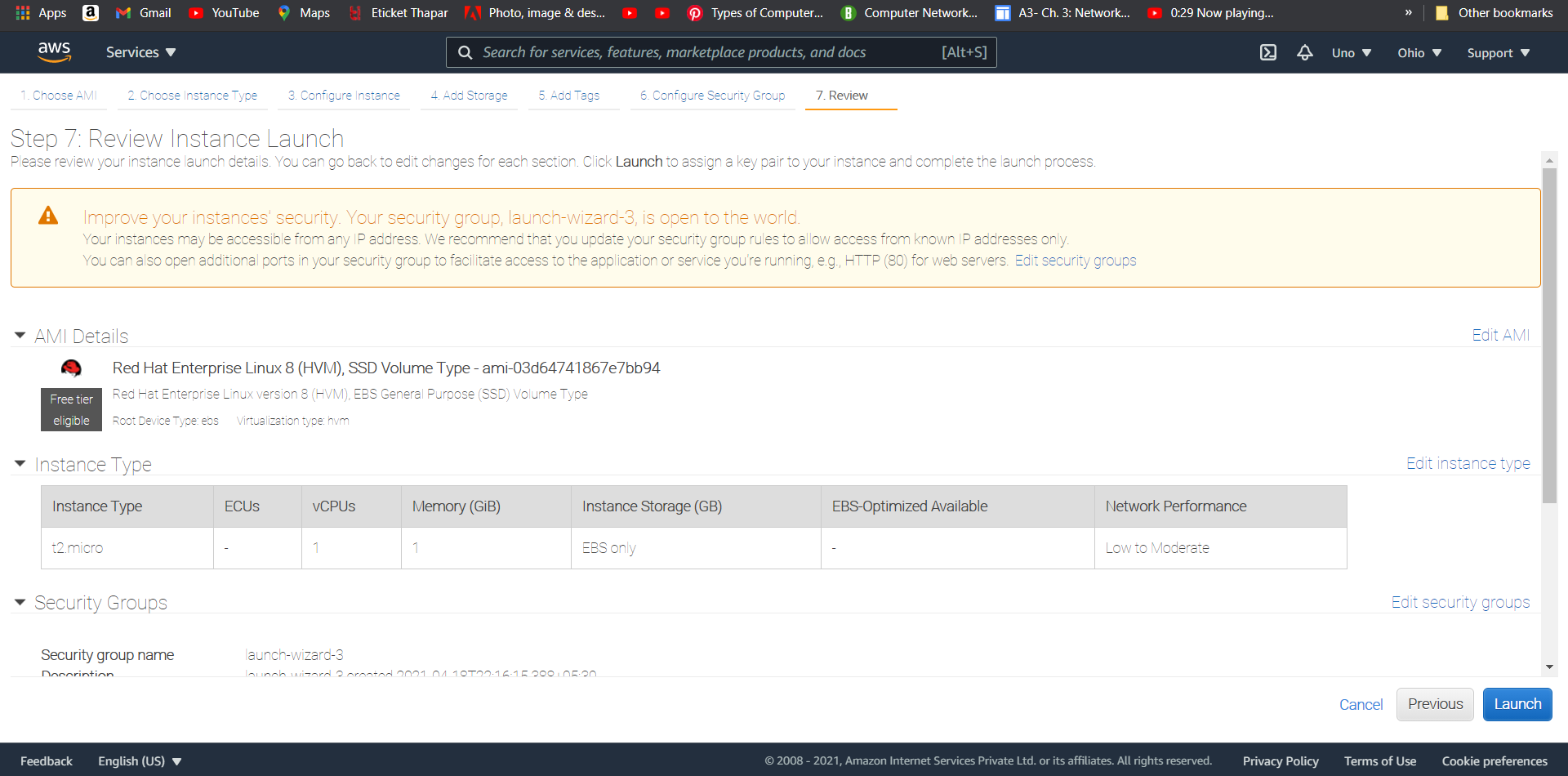
1. Configure Instance Details



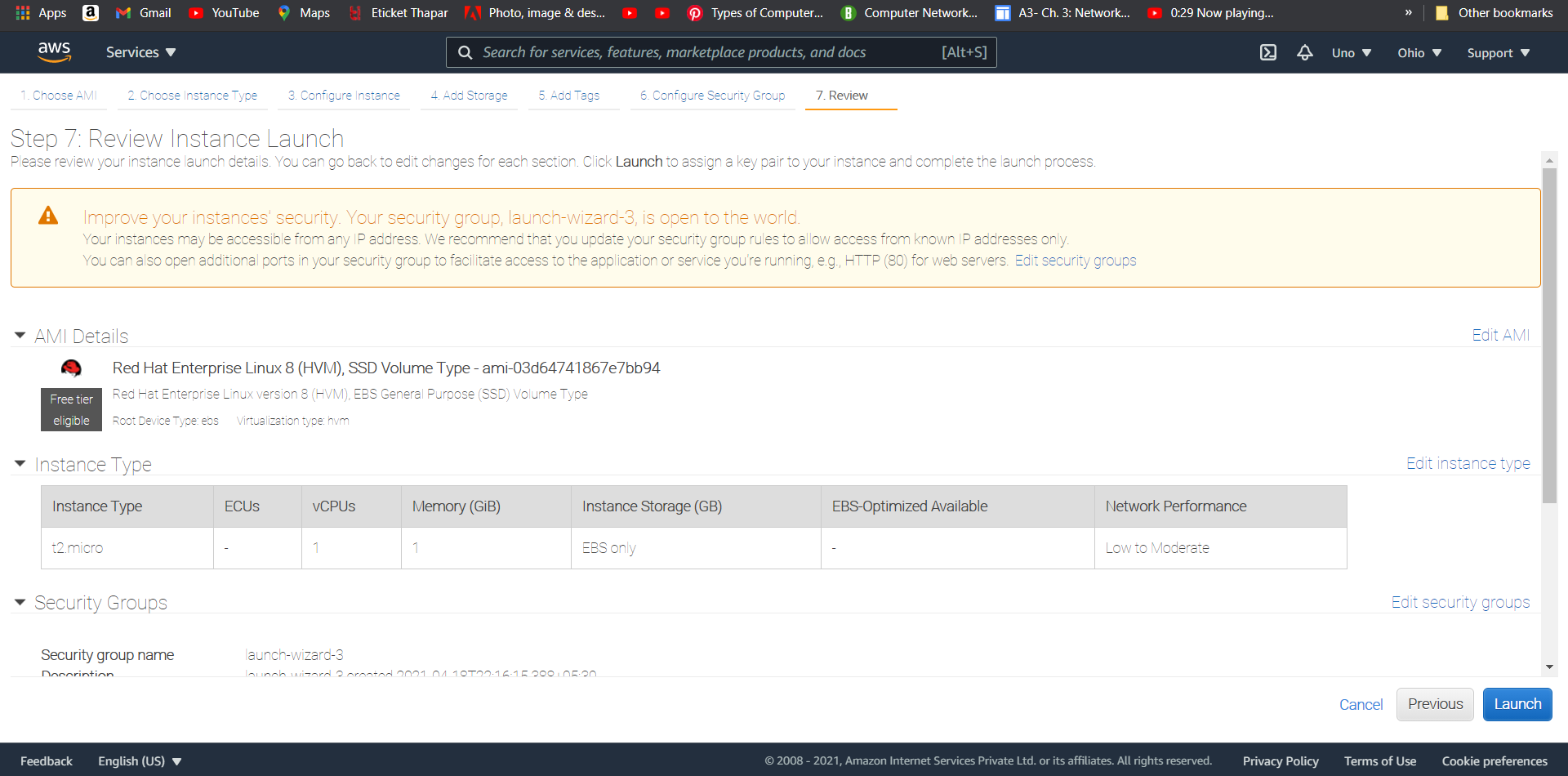
1. Add Storage



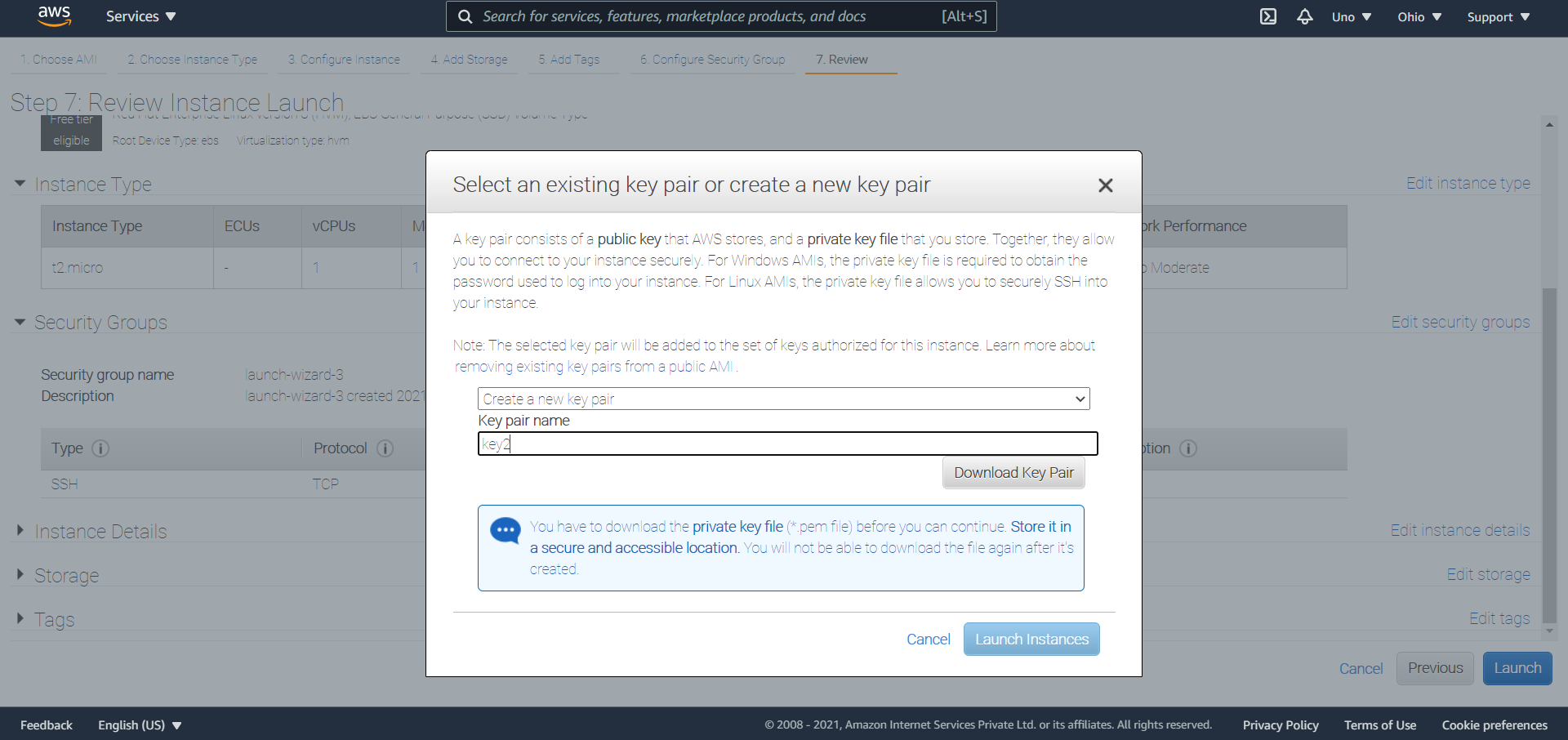
1. Configure Security Group



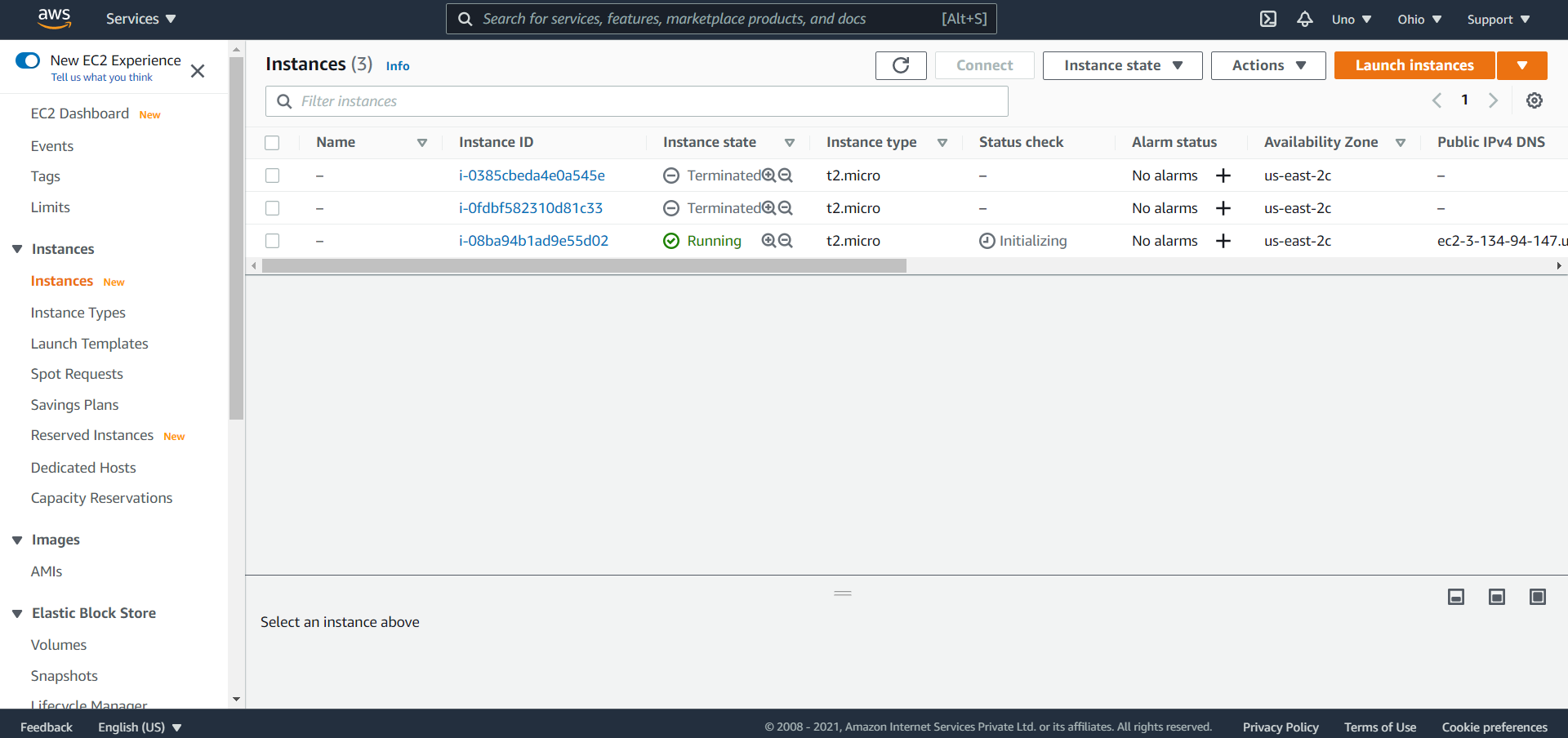
1. Review Instance Launch

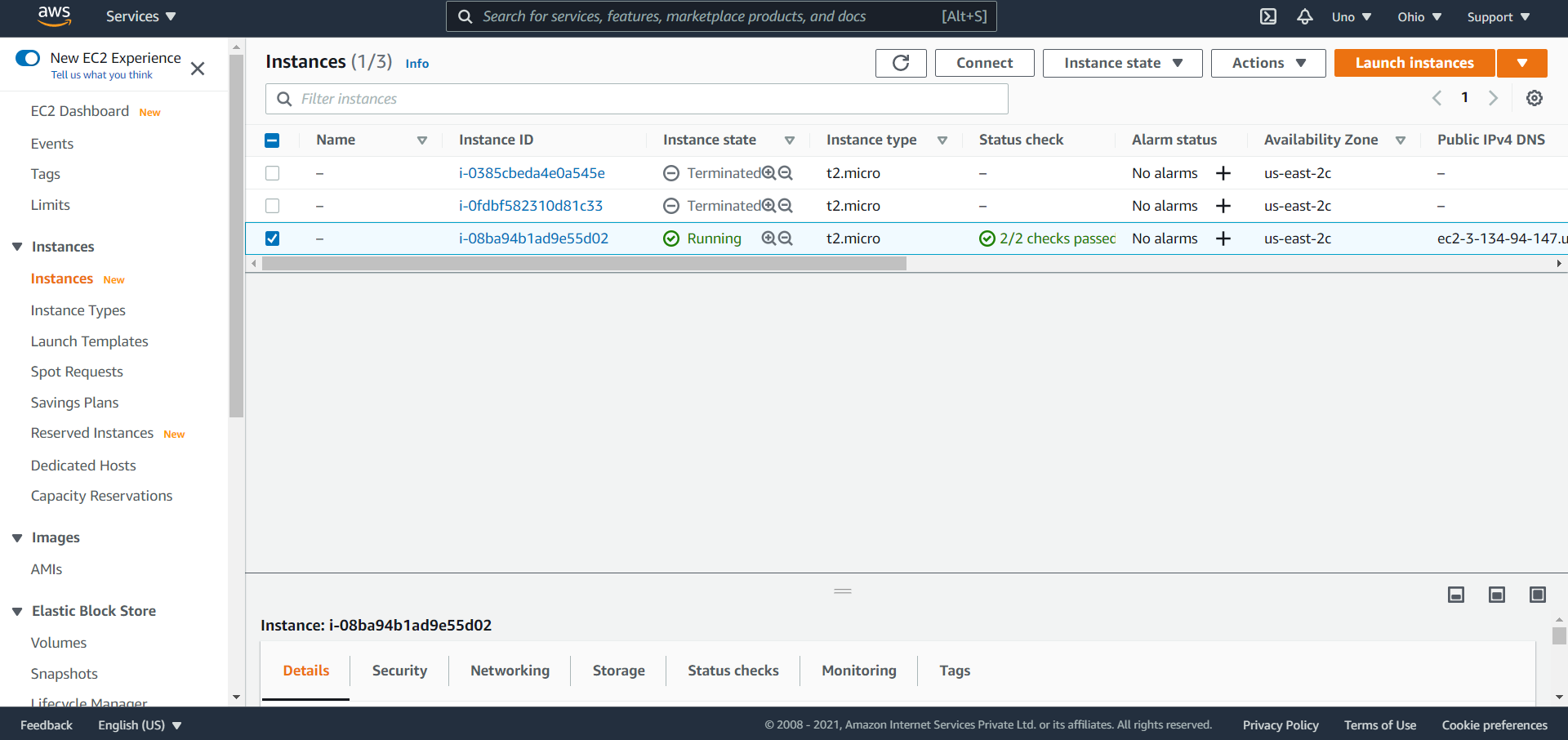


1. After clicking on launch, create a new key pair for your machine

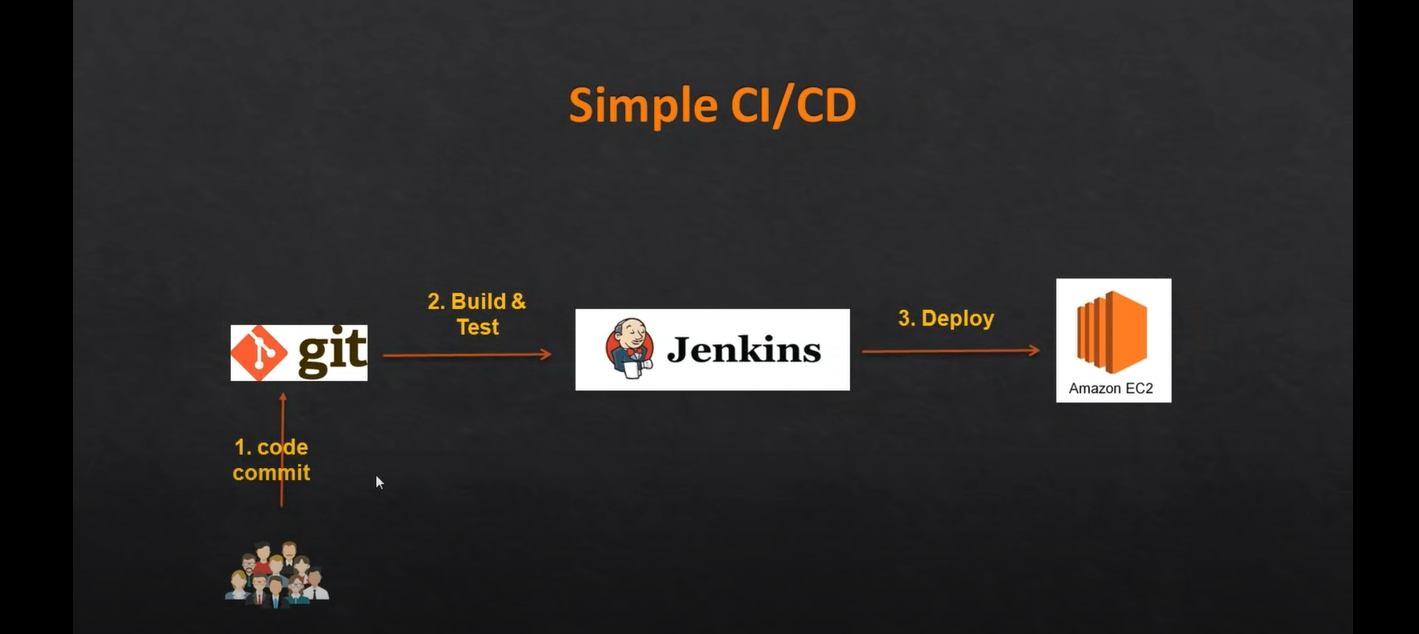


1. Now check for instances in EC2, it may take 1-2 minute for instance to initialize.

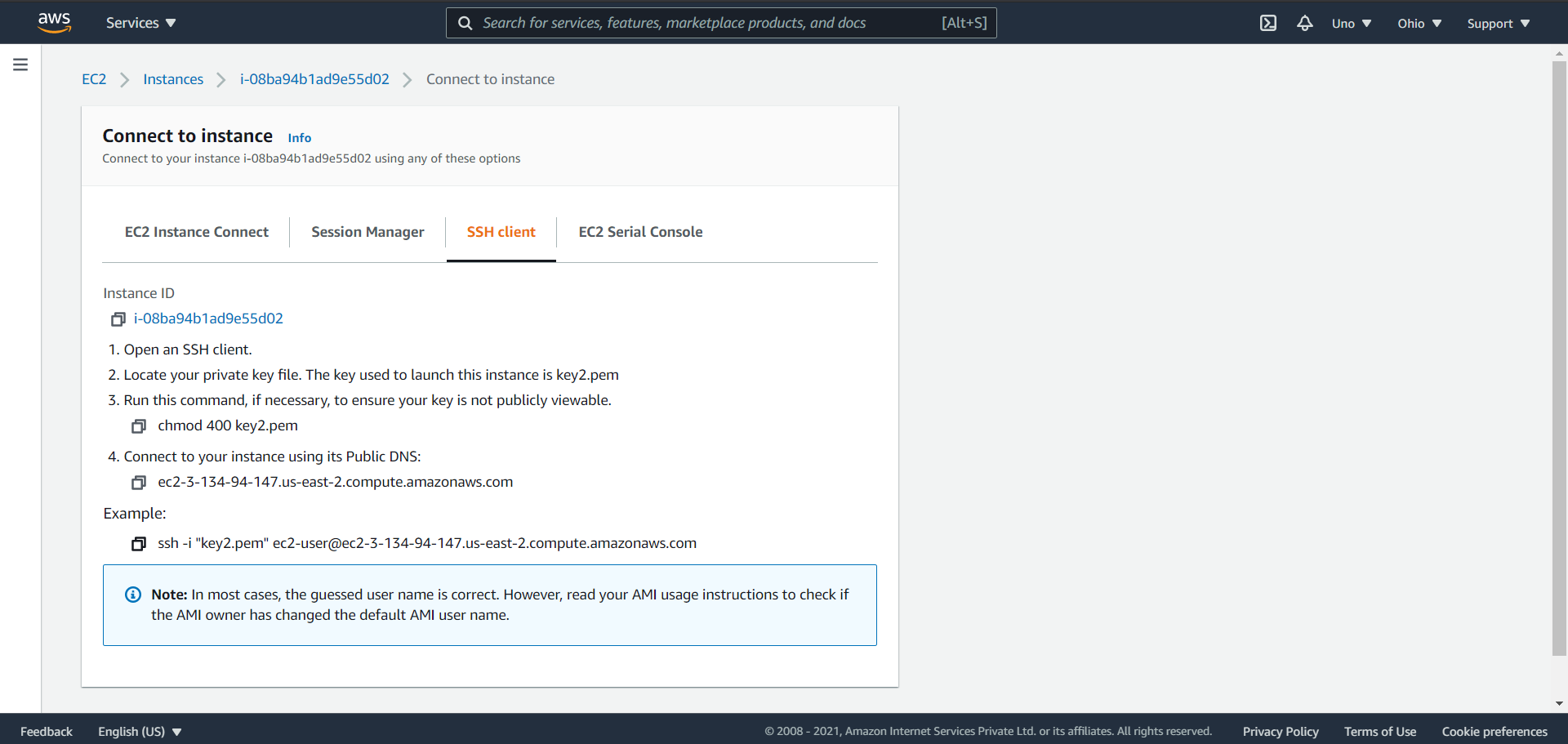


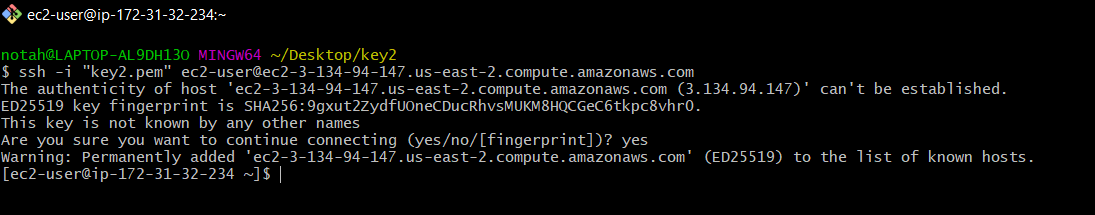


1. Connect to your AWS EC2 instance using git Bash

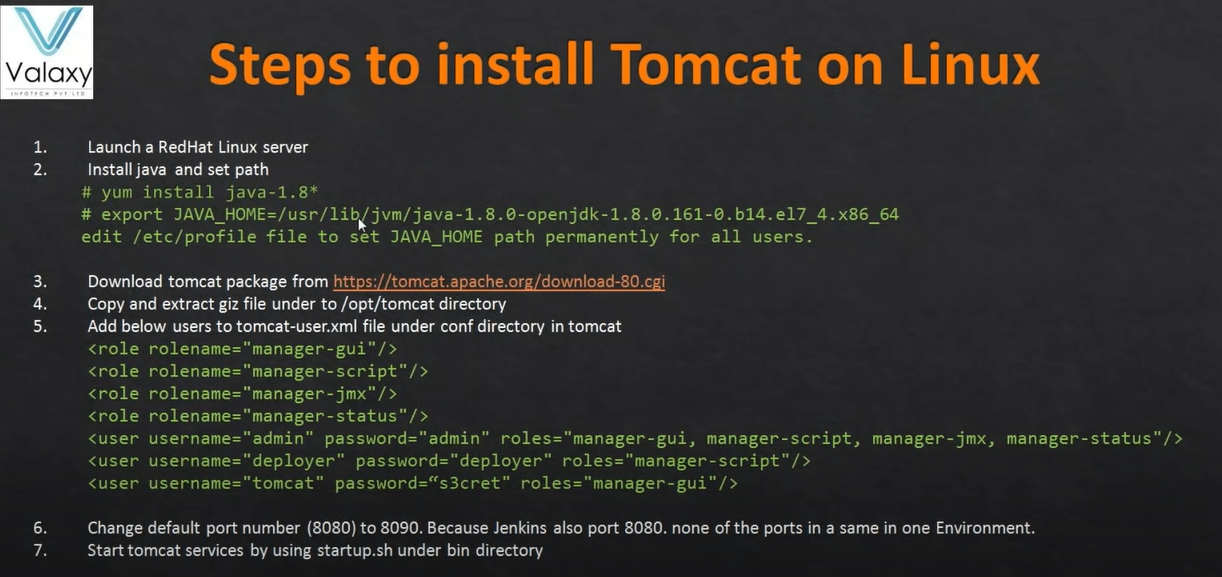


1. Connect to AWS instance



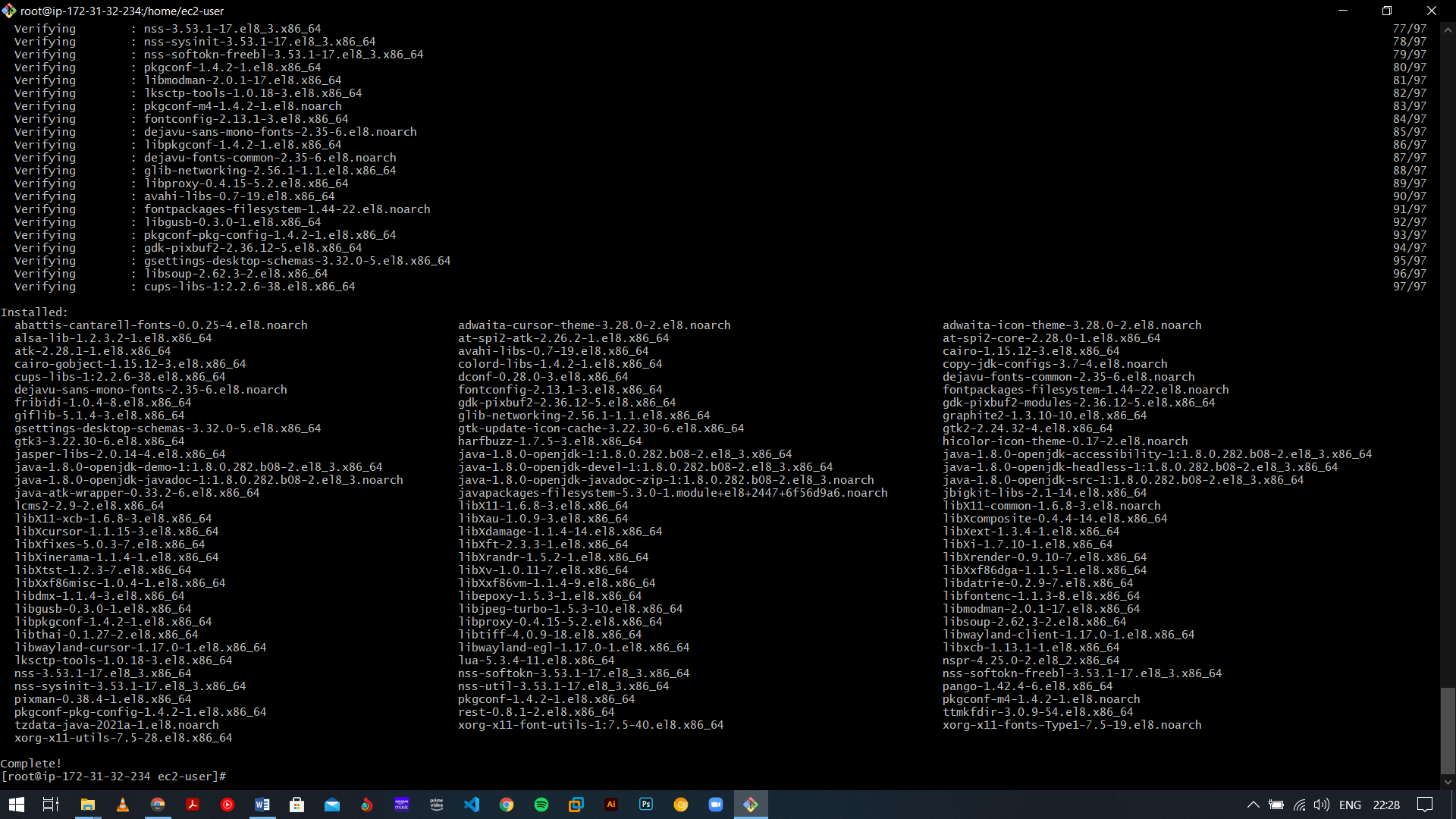


1. We need to deploy war file so we need tomcat server
2. Steps to configure EC2 instance on tomcat server:

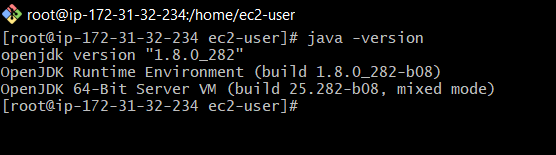


1. First we need to install java

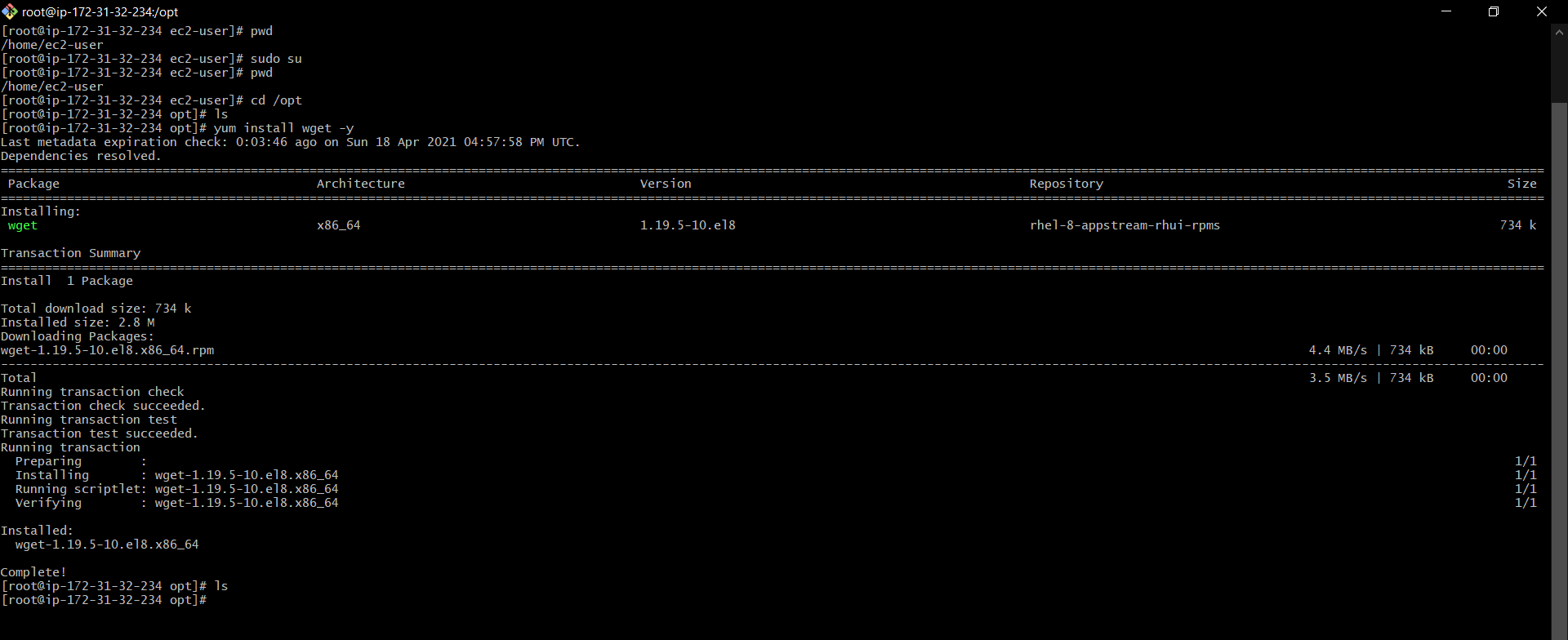




* 1. Check Version



1. wget install command : yum install wget-y

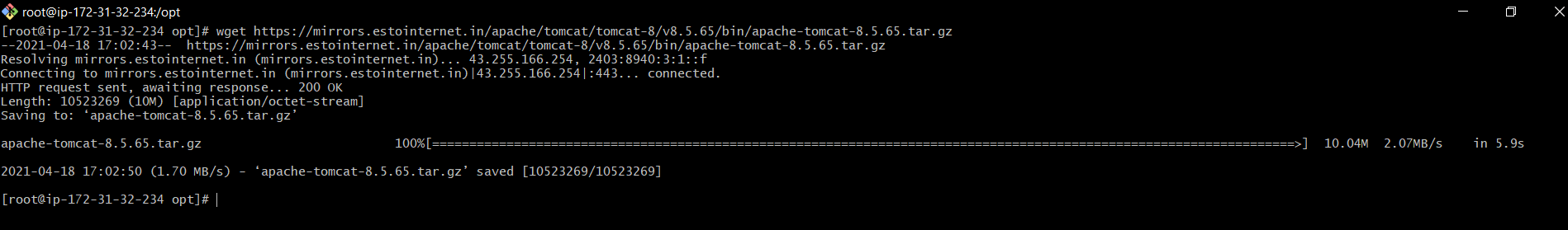


1. Download tomcat

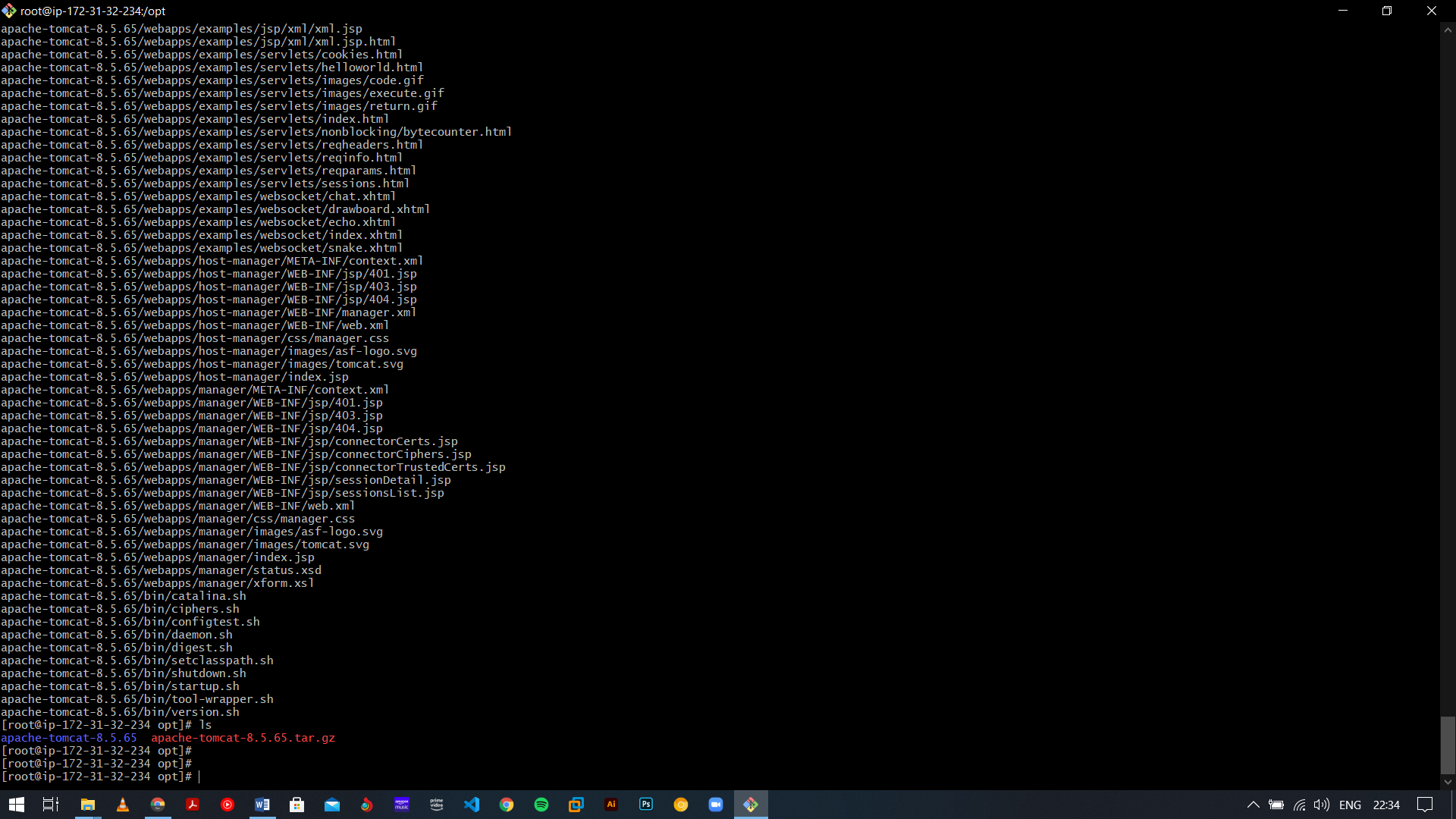
First change the path

See on which path you are using ‘pwd’ command. (It will probably be /root)

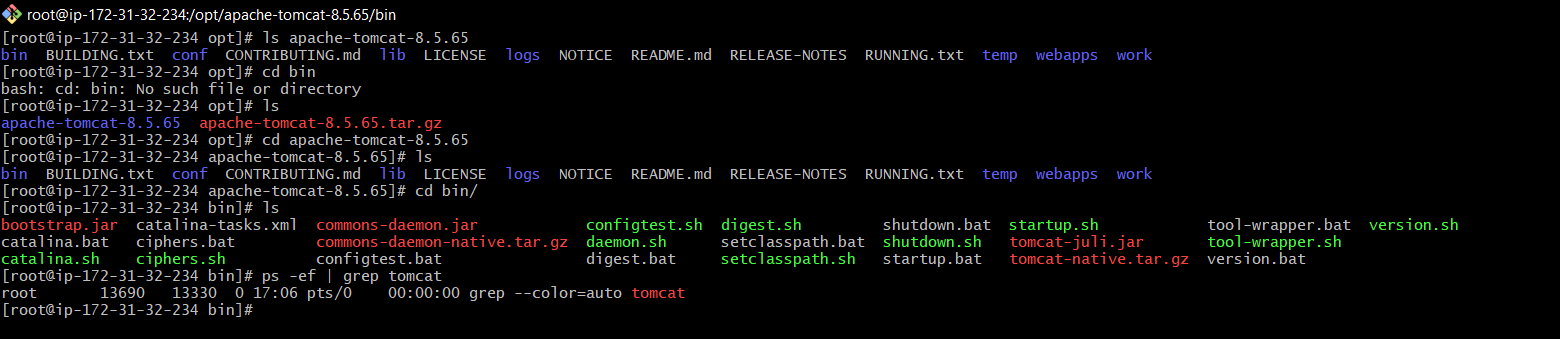
Change the path to ‘cd /opt’



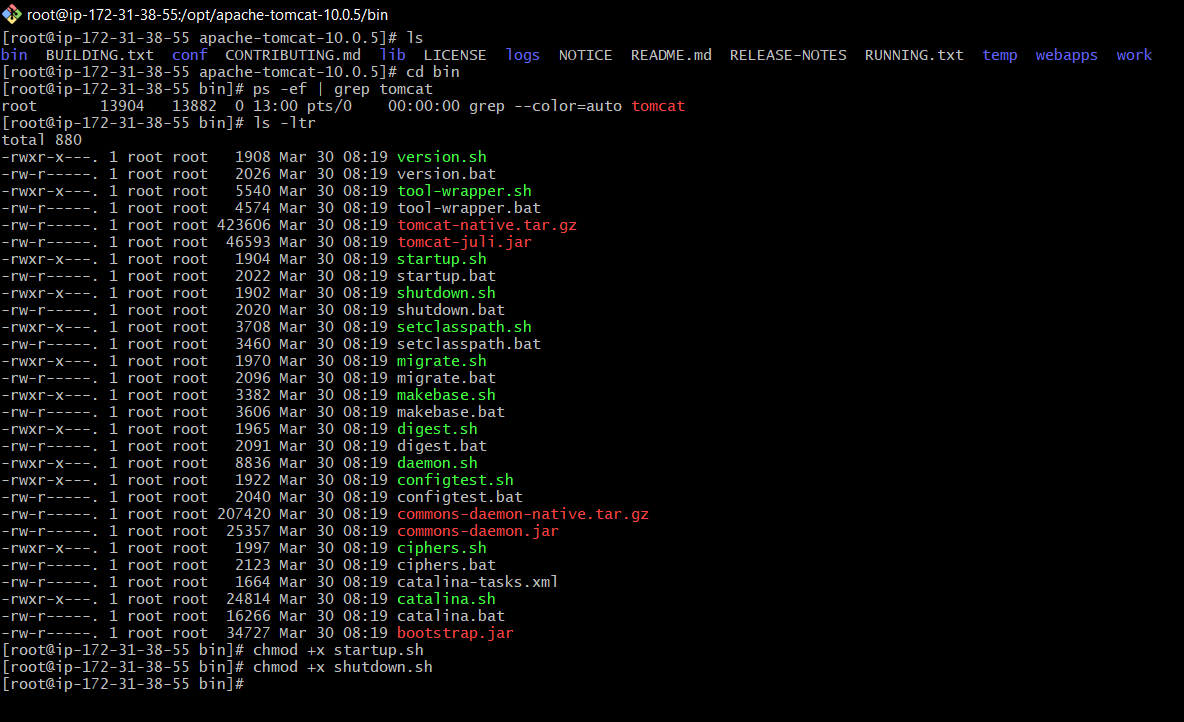
* 1. Then untar and unzip using tar command “tar –zvxf apache-tomcat-8.5.64.tar.gz”



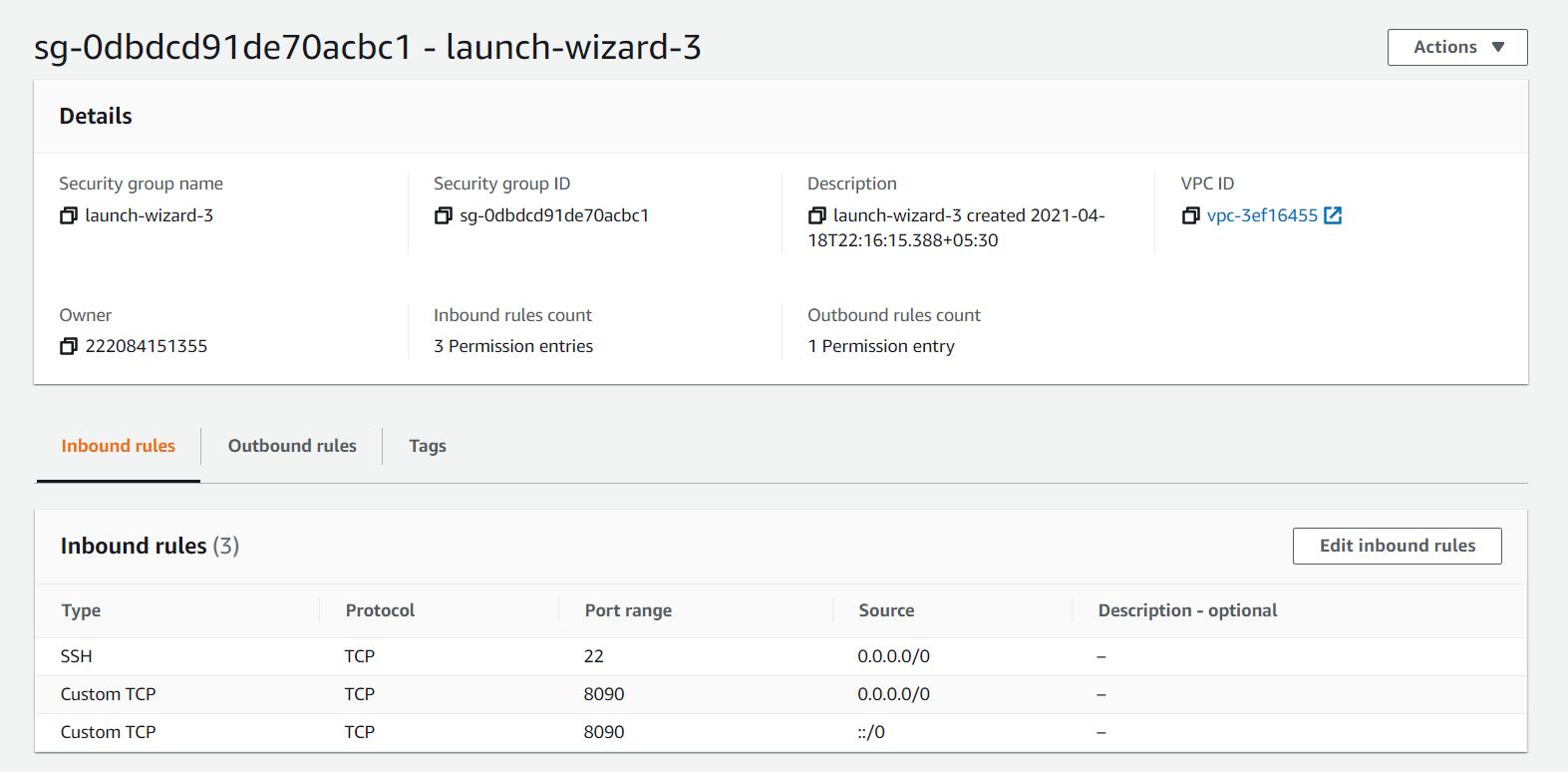
* 1. To check if tomcat is running or not (not)



1. To access default tomcat page first access services under bin directory here is startup.sh and shutdown.sh file:
   1. Give a permission to startup.sh and shutdown.sh:

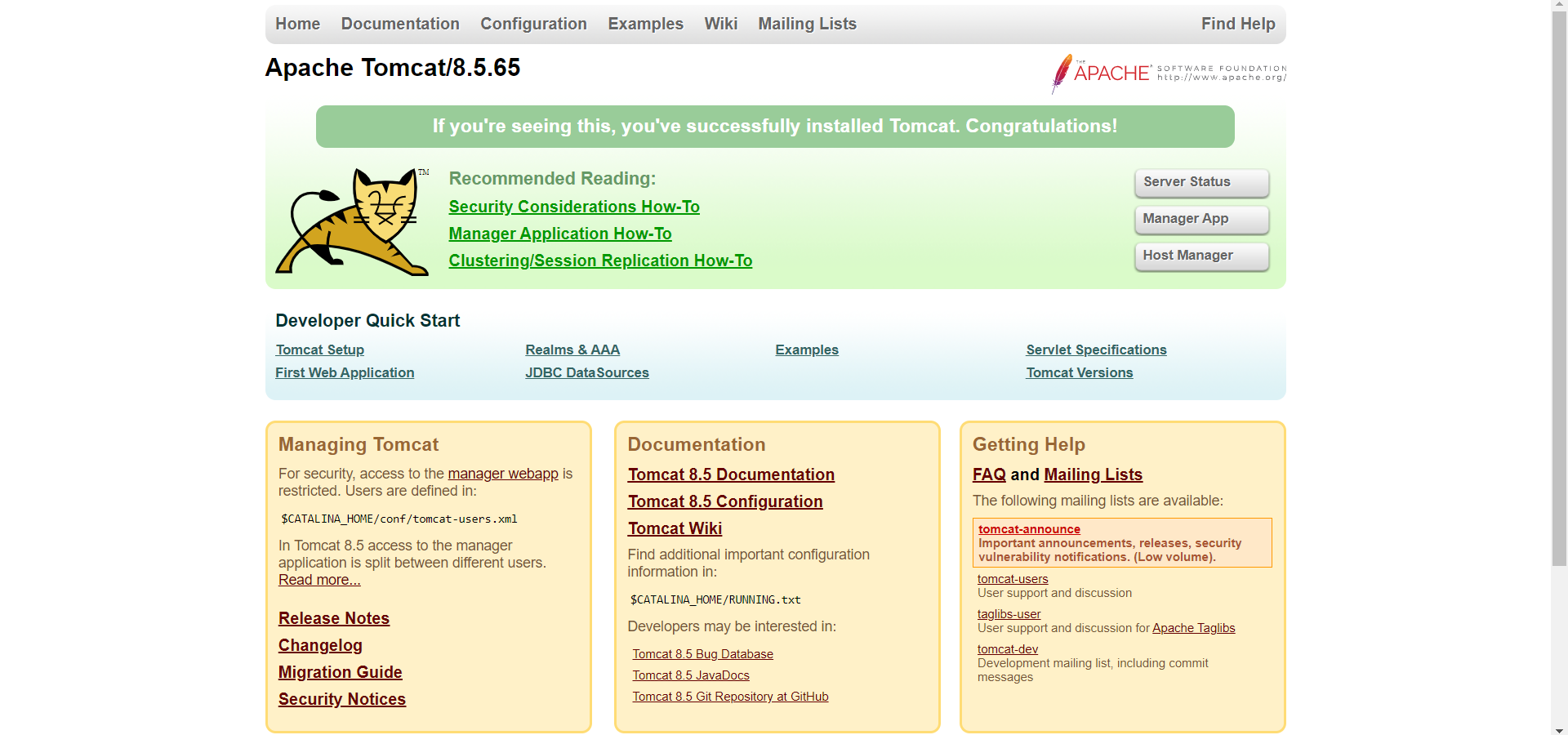


1. By default it will work on port no. 22 so in security group we need to add port 8090:

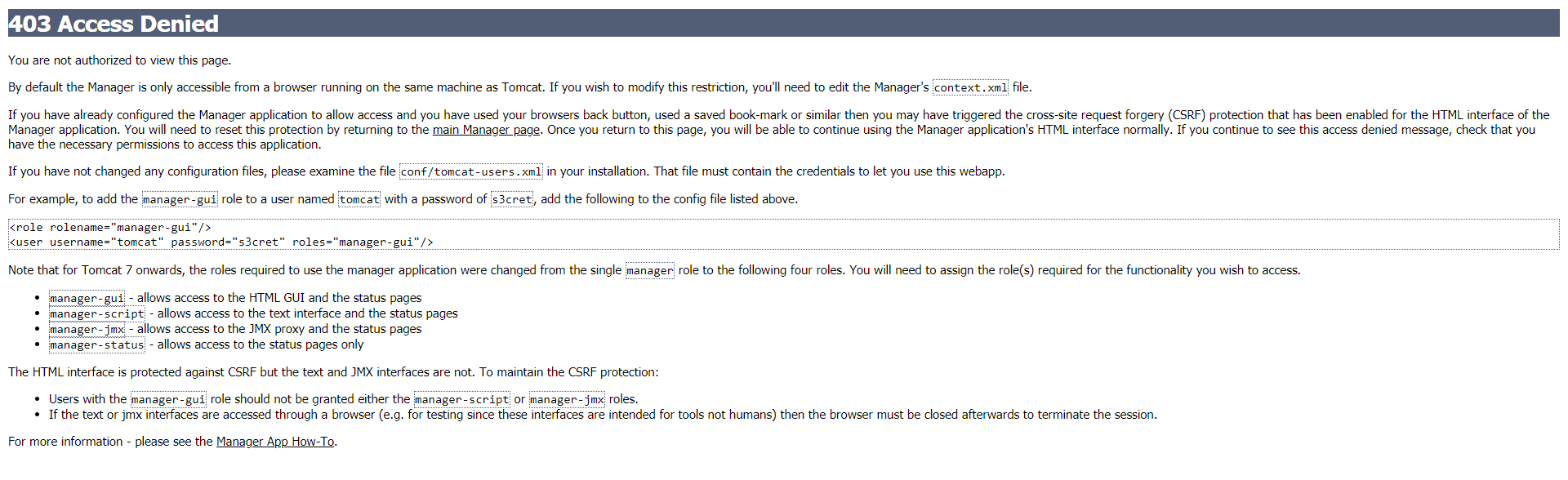


1. To access tomcat in a browser, copy the public ip of your aws instance and add ‘:8090’ at the end.

For example: <http://3.21.162.39:8090/>



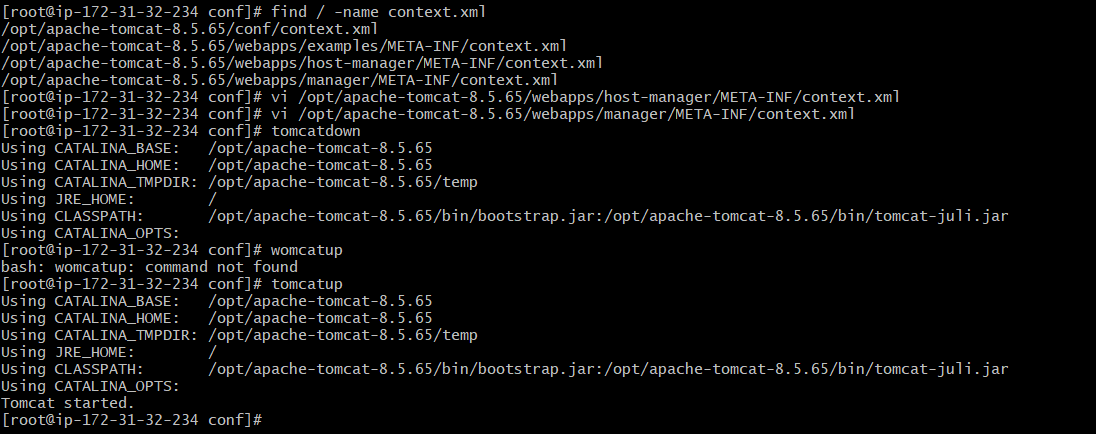
1. If we want to access these manager file then it will access from the same machine as tomcat but we access it from other so we need to edit the context.xml file



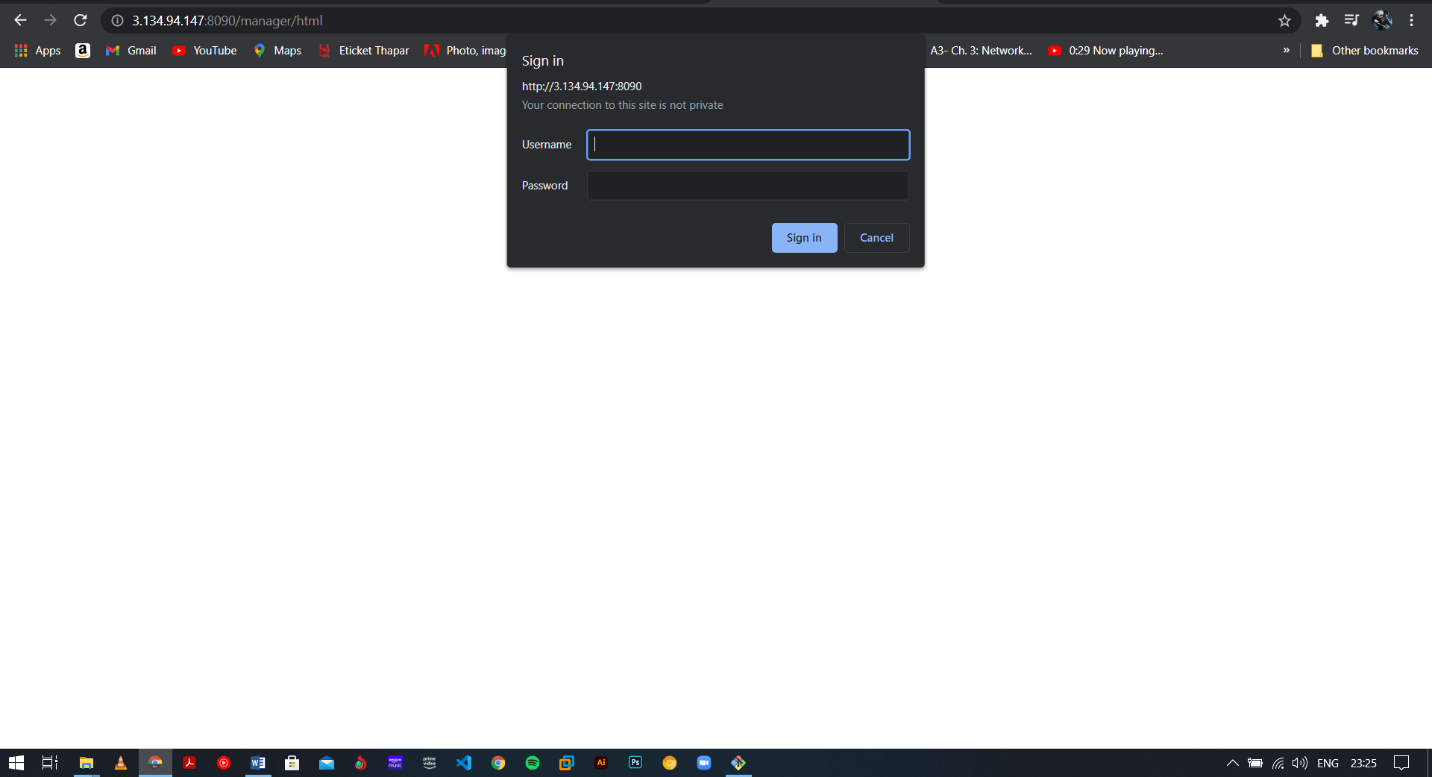
1. Search context.xml file via find command if will give the location

Command: find / -name context.xml

* 1. Edit the context.xml file using vi command



1. Again go back to server and access manager apps then it ask for credential



1. In conf there is a file call tomcat.users.xml , here we need to add user and role
   1. User and role are given below add in between tomcat users:

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

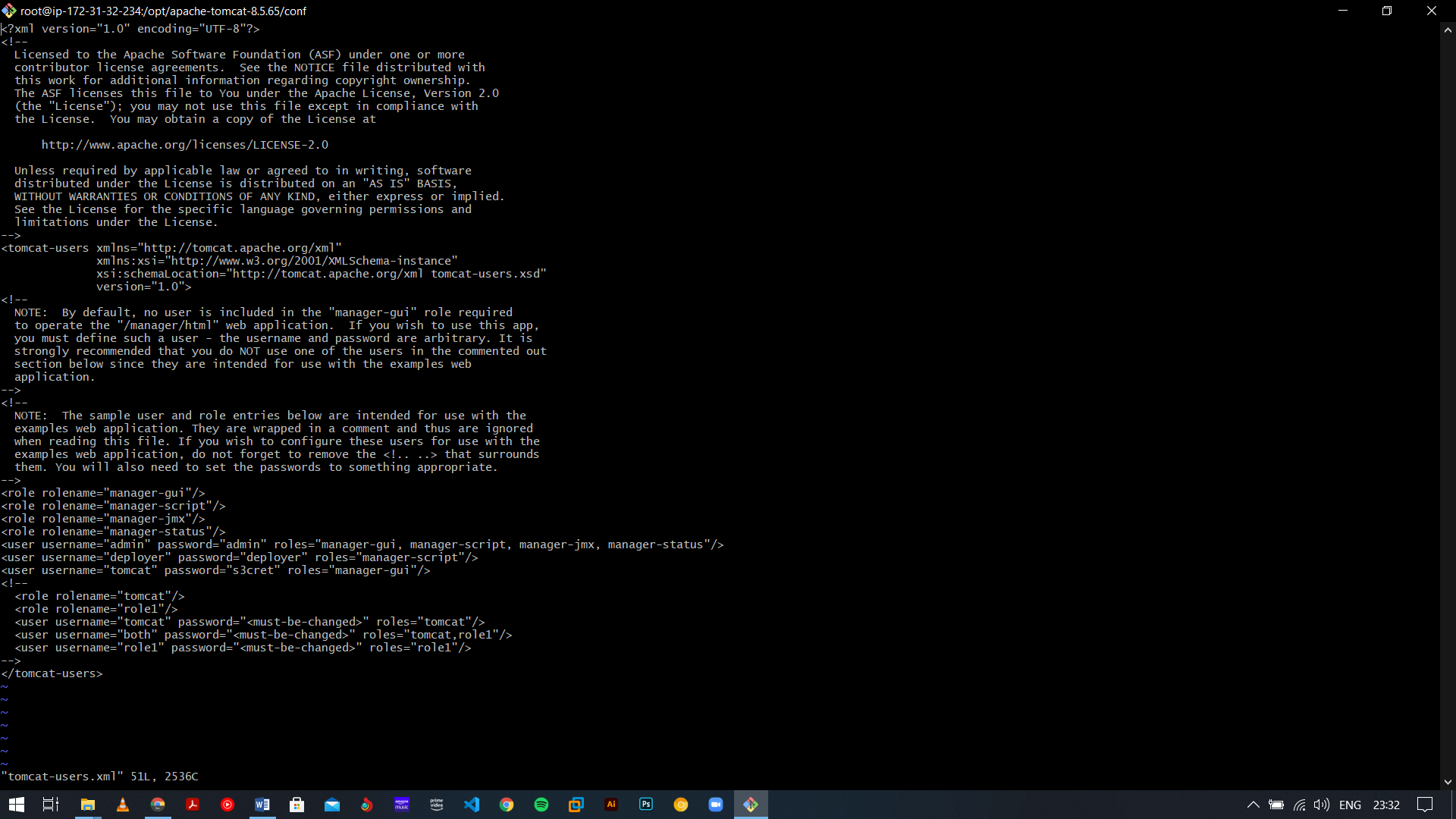
<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

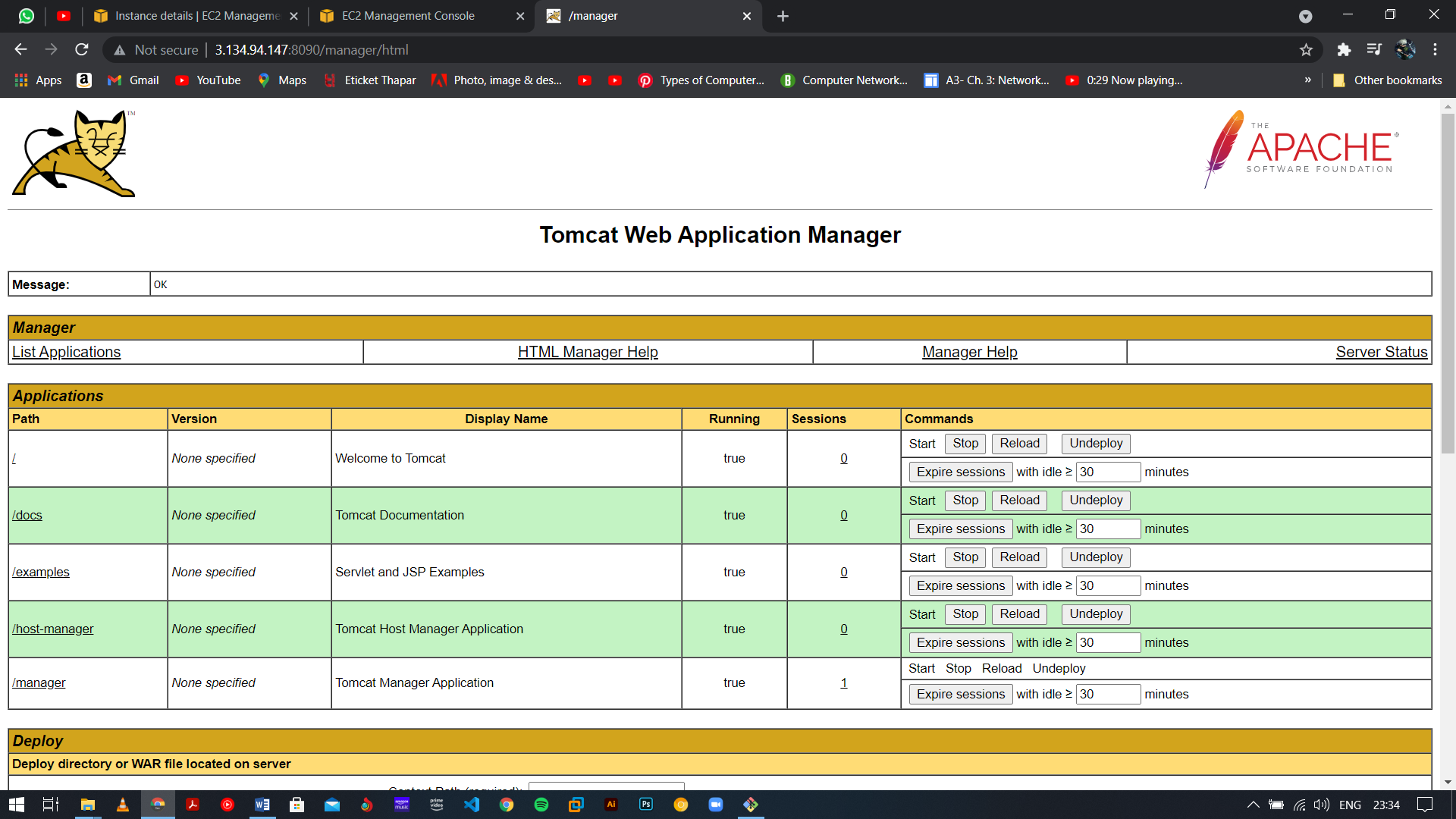
<user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>

<user username="deployer" password="deployer" roles="manager-script"/>

<user username="tomcat" password="s3cret" roles="manager-gui"/>



1. Again refresh tomcat server and access manager app sign in with username and password it look like:



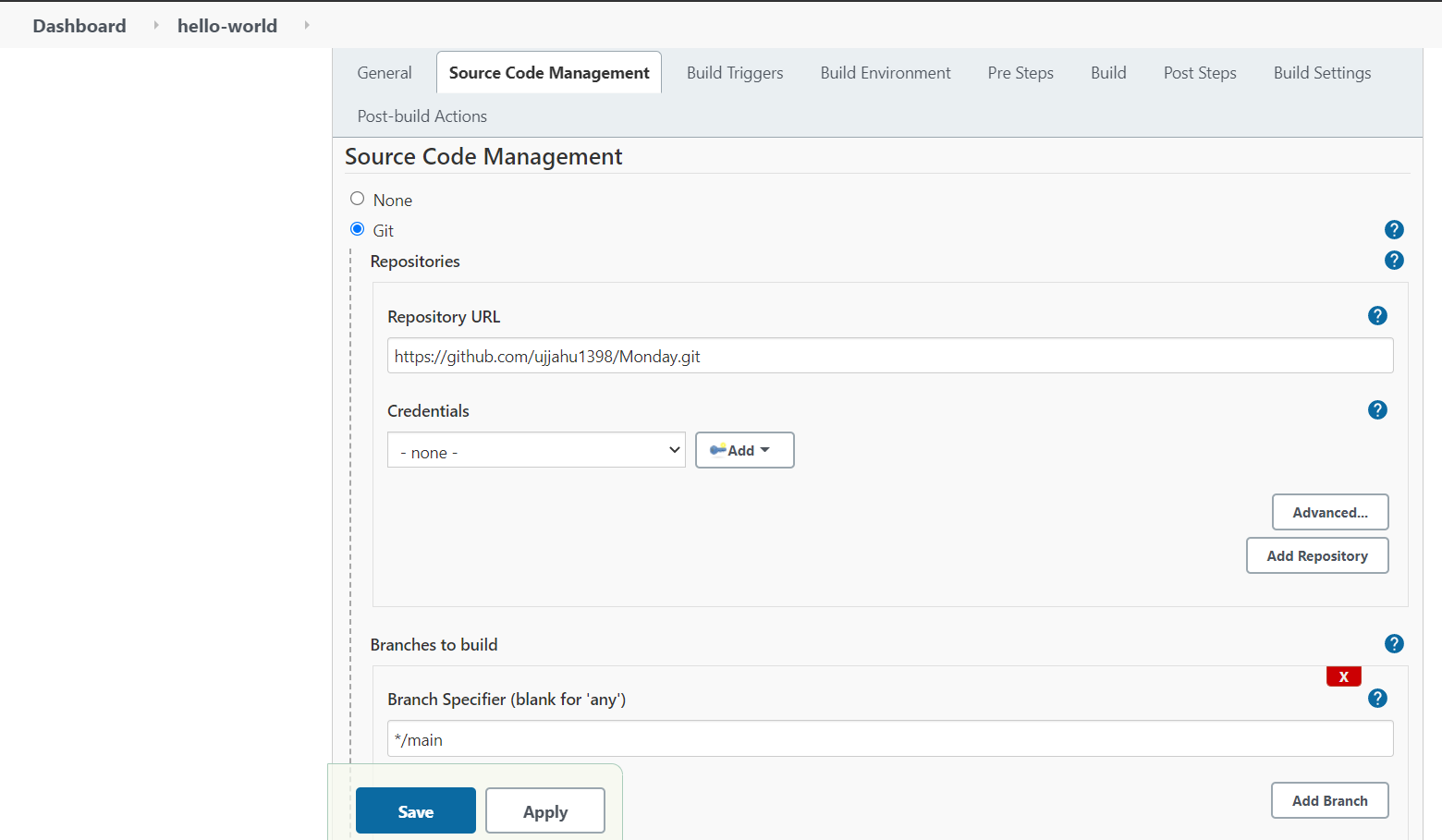
1. Project is cloned where we saved our AWS key.

Link: <https://github.com/ujjahu1398/Monday/blob/main/webapp/src/main/webapp/index.jsp>

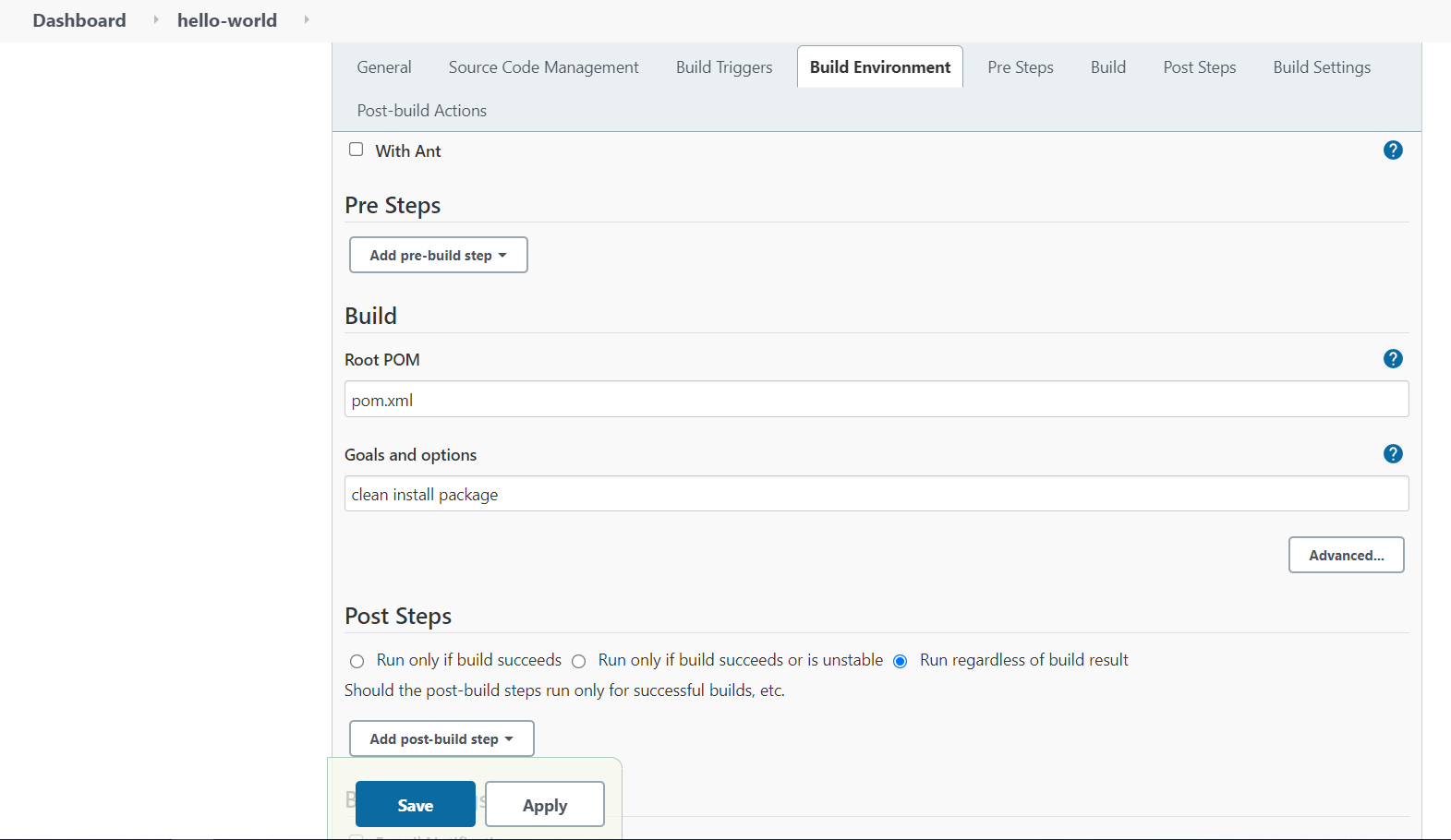
1. Open Jenkins and create a new project



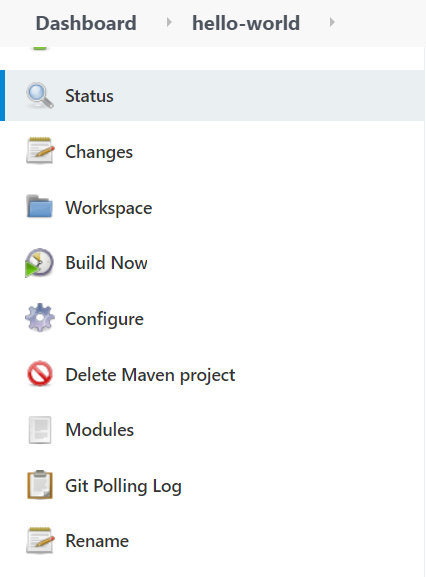
1. Set your git url path



1. Set the goal that means clean old build and just install the maven package



1. Then click on Build Now Manually



1. After build we need to deploy on server. First open your tomcat server, After that we need to give our tomcat credential to our Jenkins therefore we need to install the deploy plugin on Jenkins.

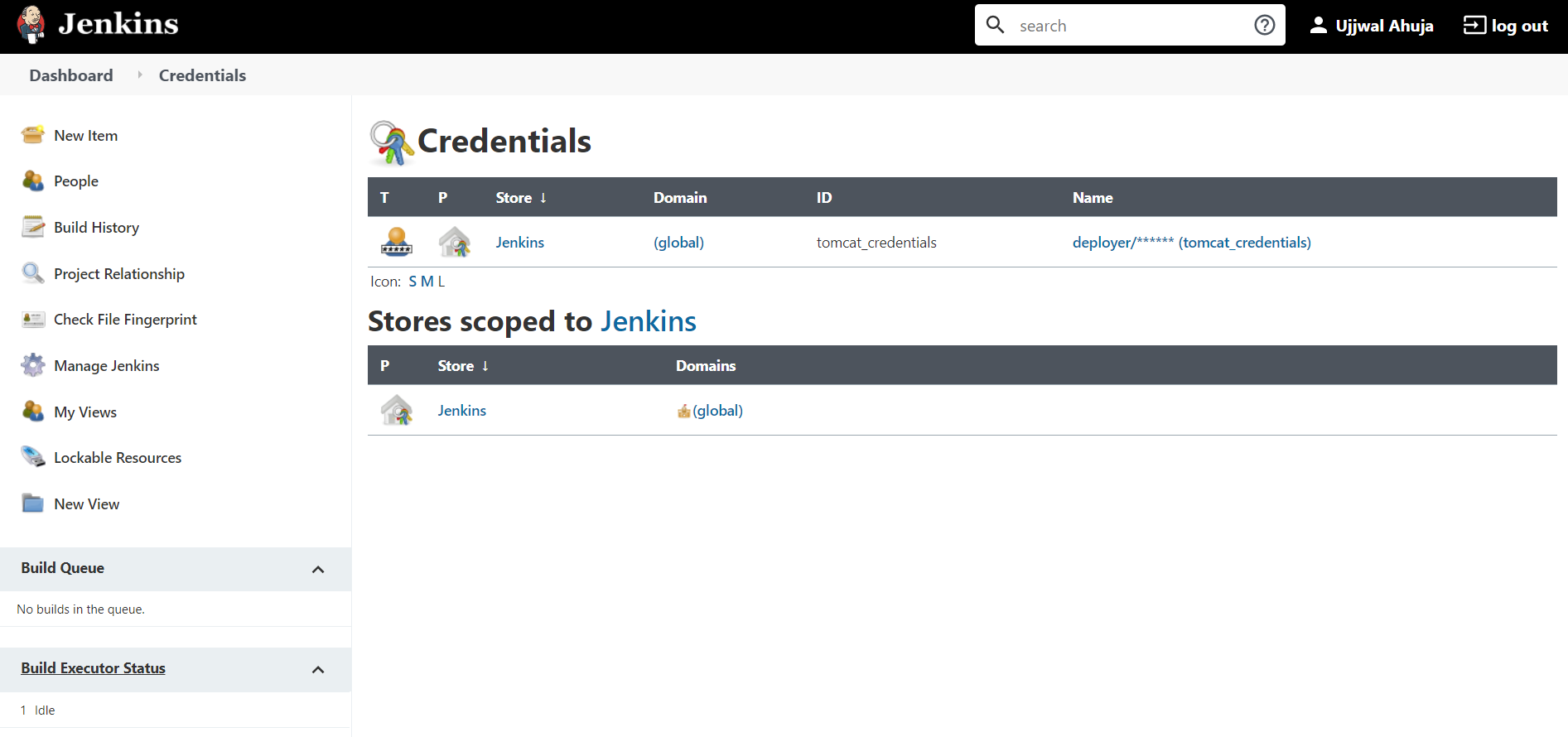
Go with:

Dashboard-> Manage Jenkins-> manage plugin-> Available-> Deploy to container-> Install w/o restart

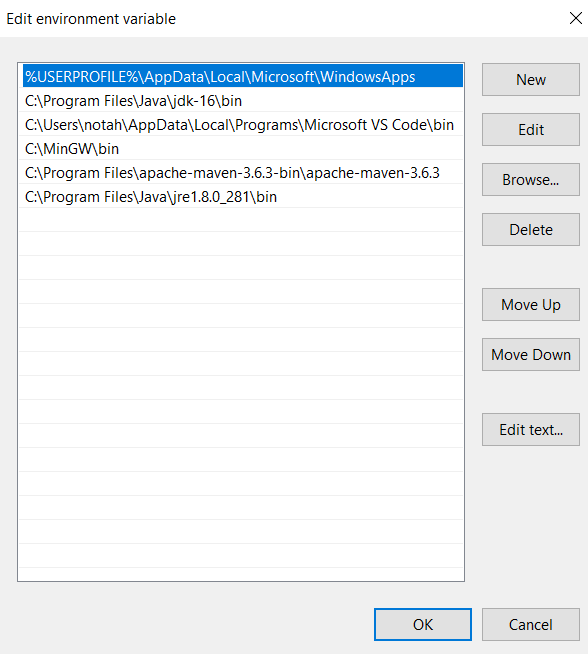
After this we need to set the credentials then,

Go with:

Manage Jenkins-> manage credentials-> global credentials-> add credential

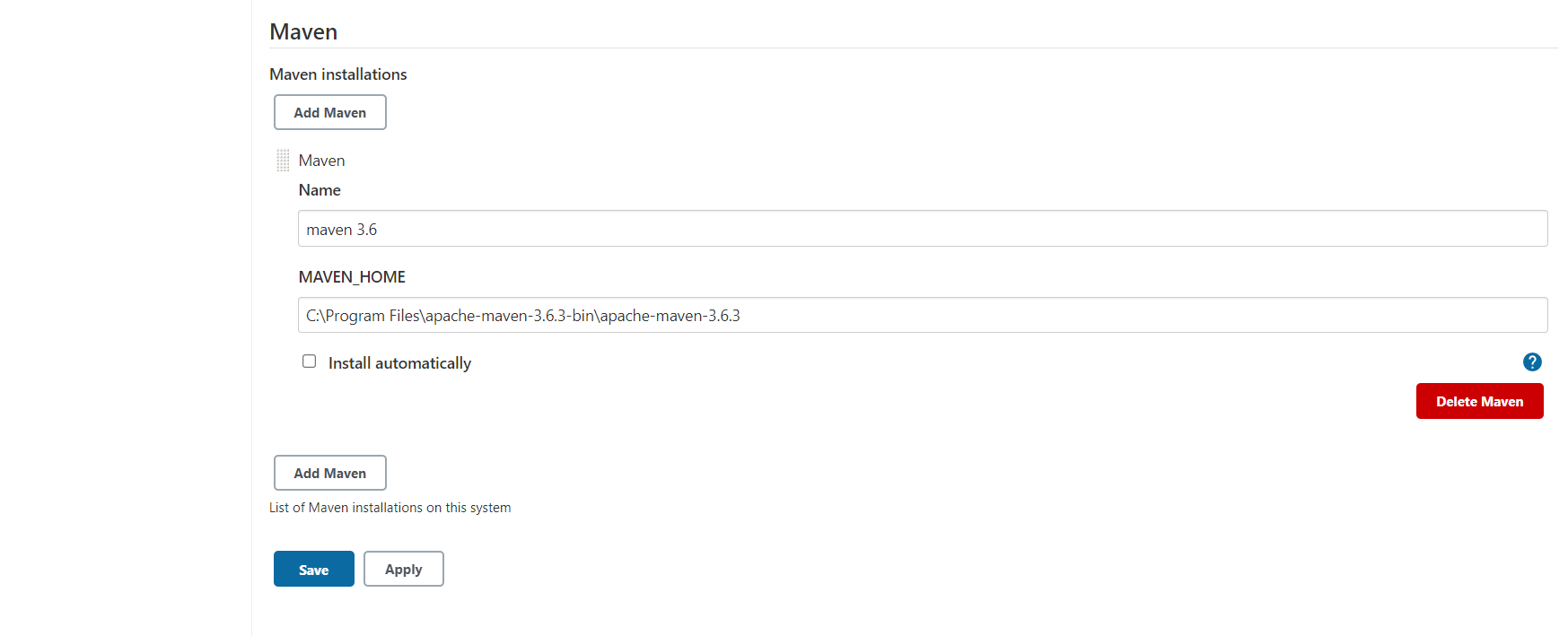


1. Download Apache Mavin and set path in Environment Variables

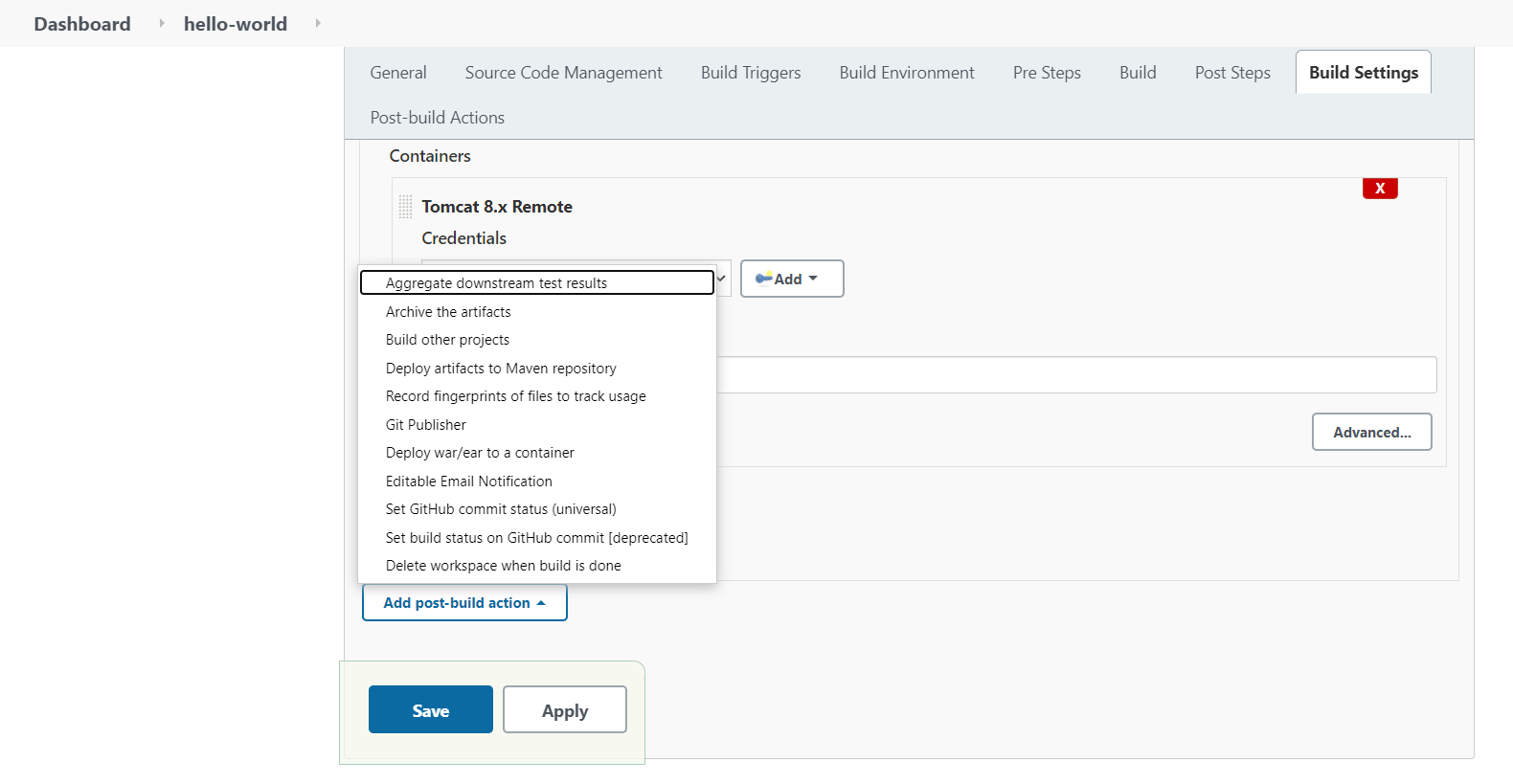


1. For adding maven manually go with:

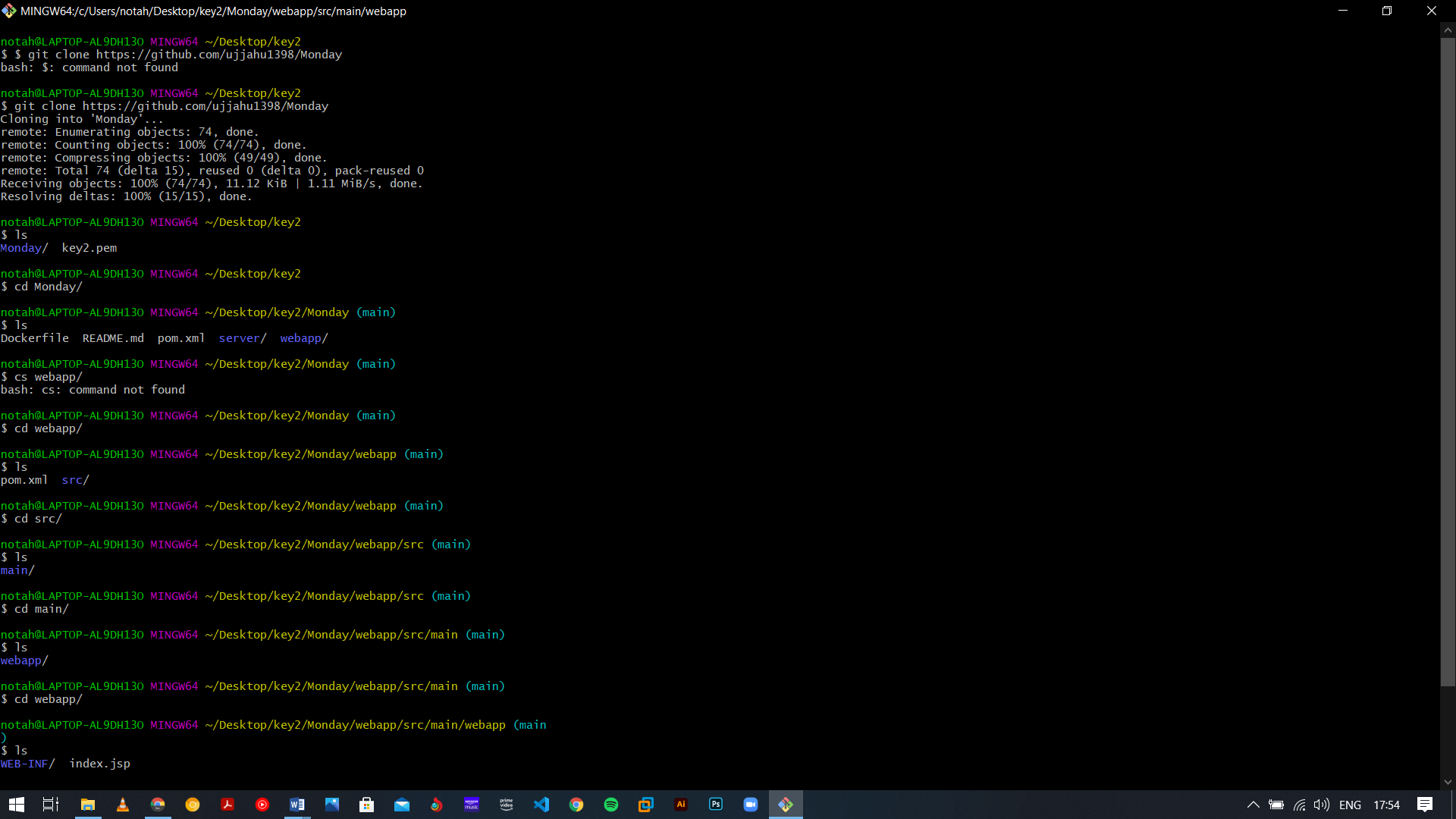
Manage Jenkins-> manage global tool configuration-> add maven

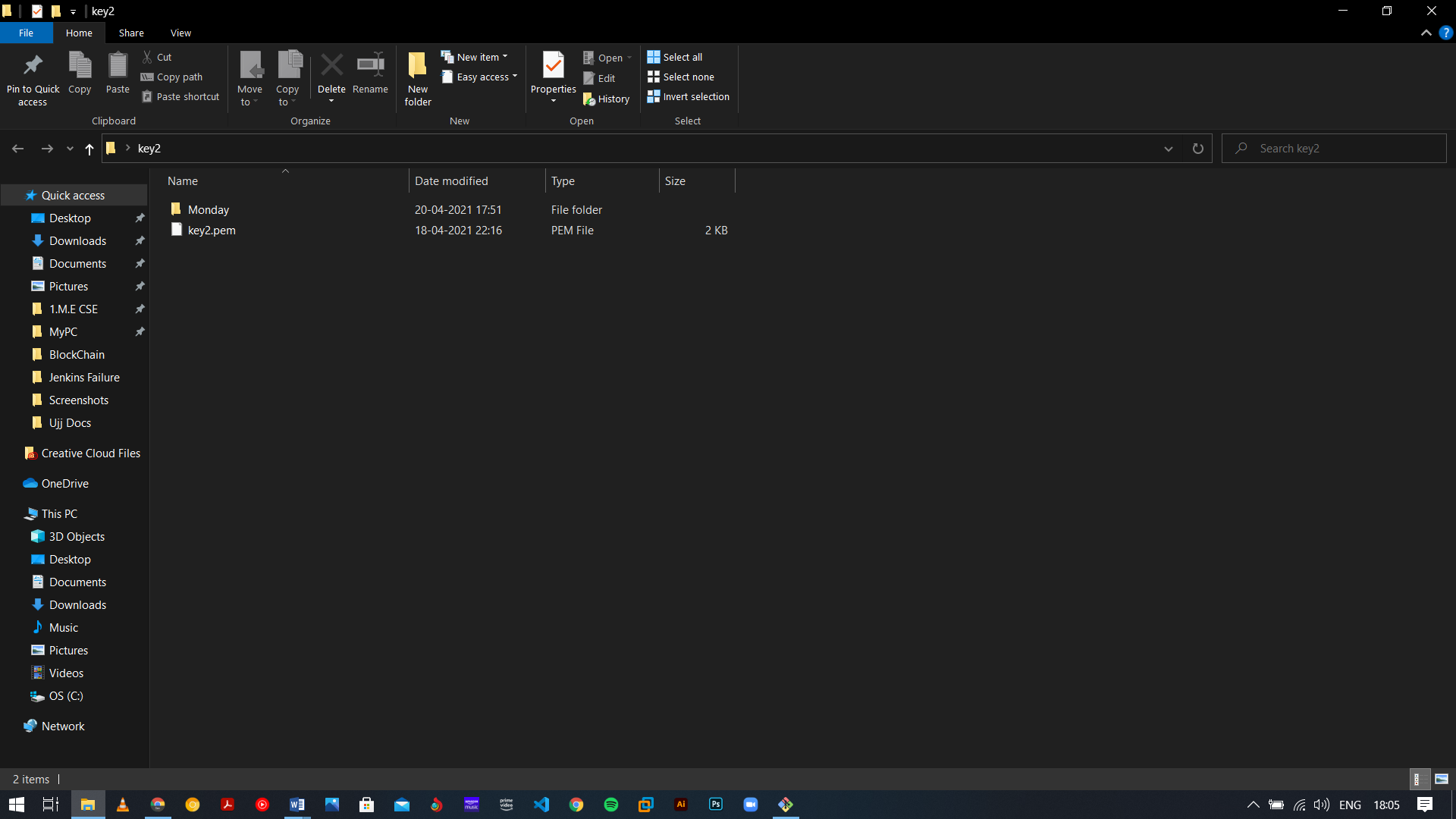


1. Deploy war to container:

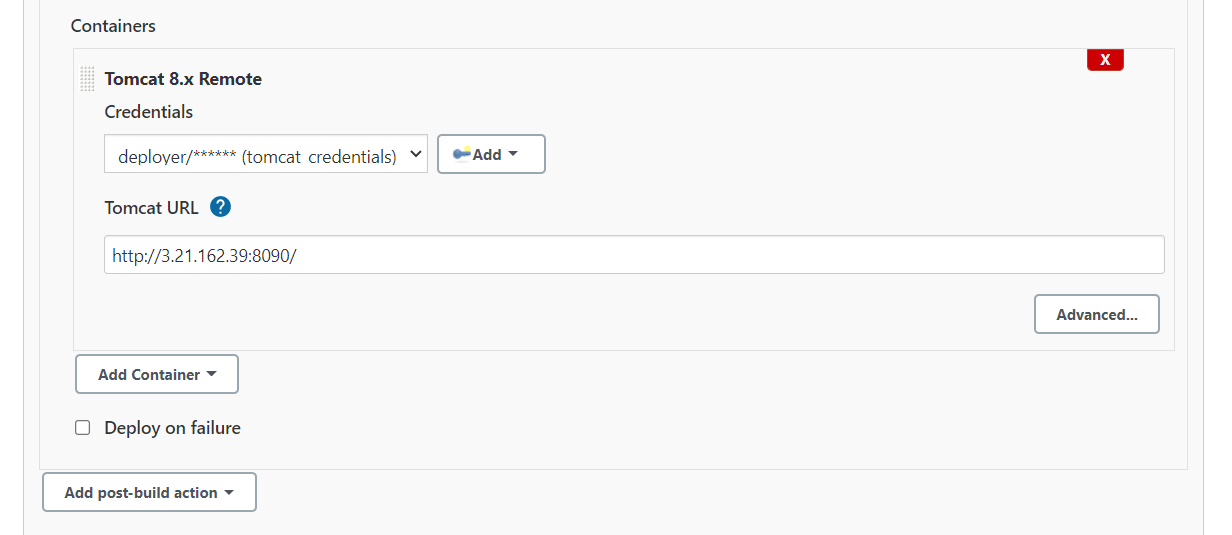


1. Clone the code where your key is saved:

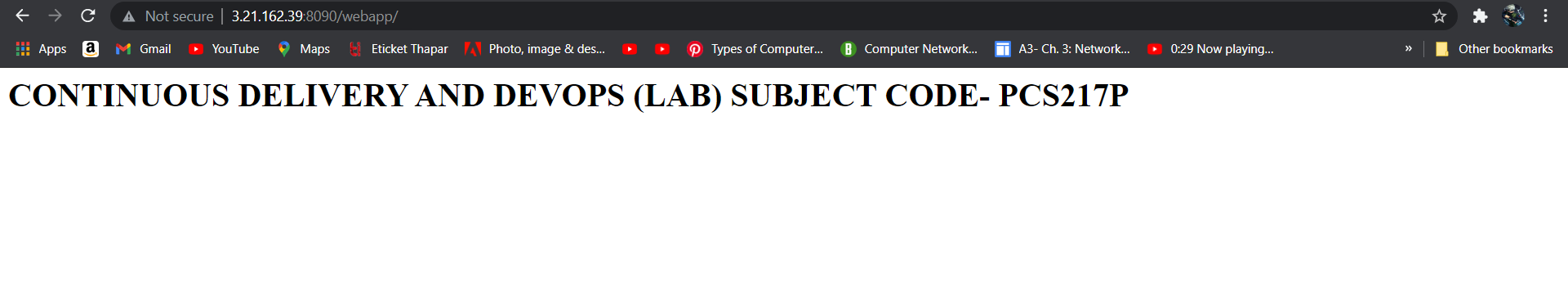




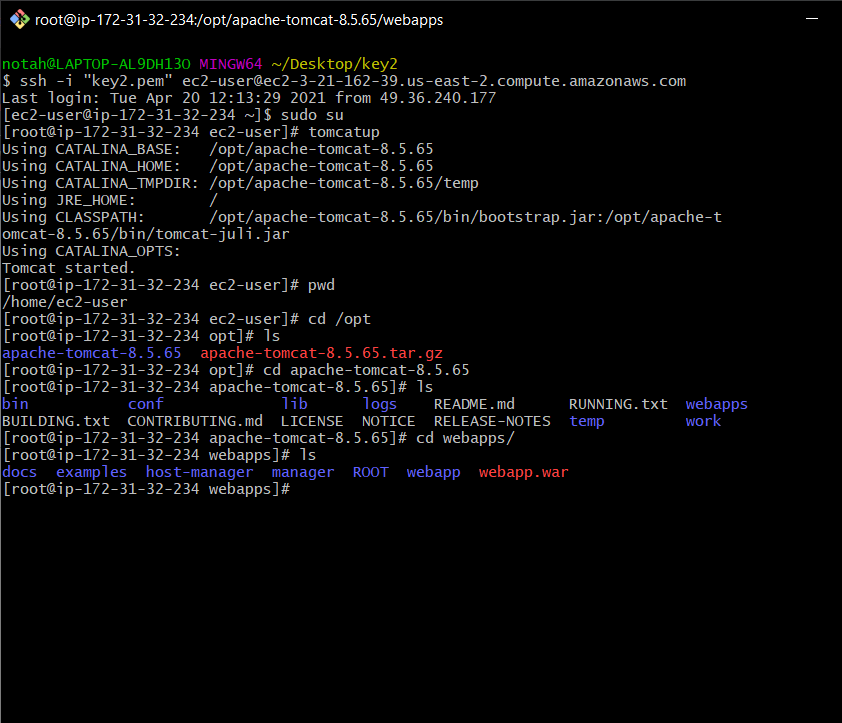
1. When we give the right credential of tomcat then only this war file will deploy on tomcat server



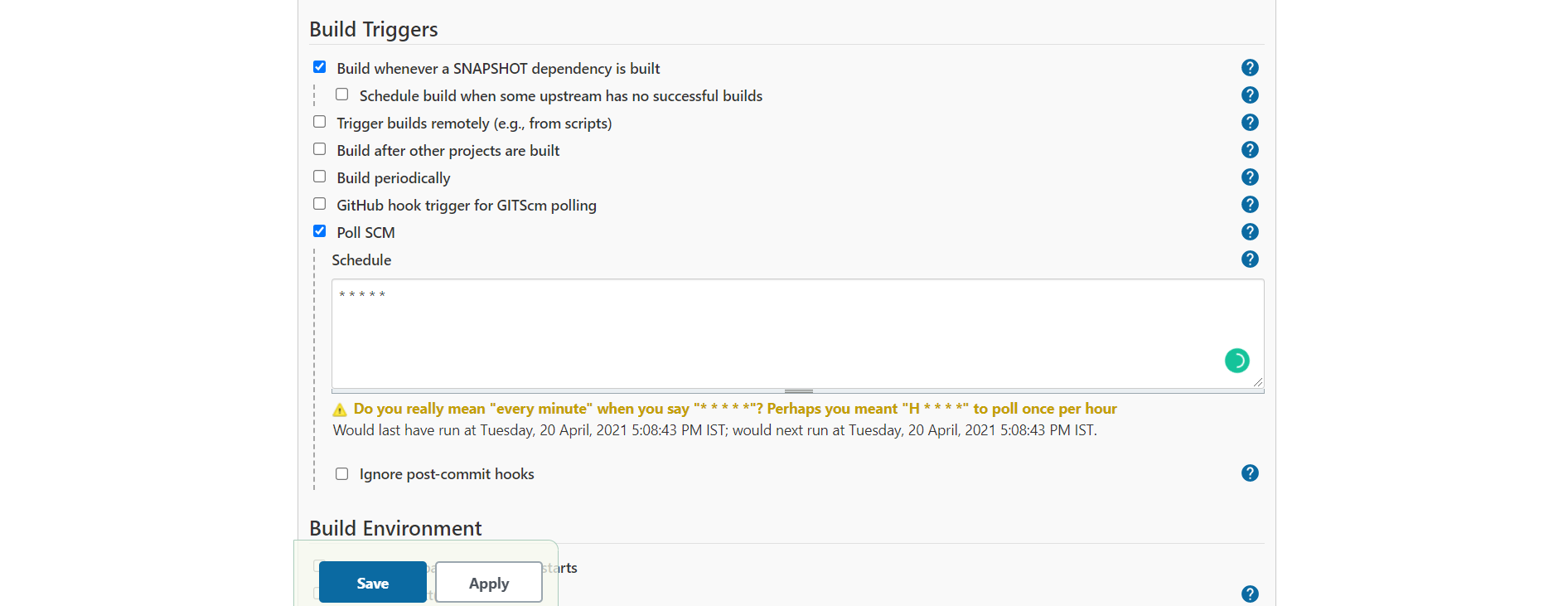
1. Then build now and check the link “ <http://3.21.162.39:8090/webpage/>”



1. Commit changes in your git repository.
2. Then build it if should able to create the war file once war file is create successfully it simply deploy to server



1. Build automatic trigger for 1 minute.



1. check the link “ <http://3.21.162.39:8090/webpage/>” again:

