**CI – Continuous Integration**

**CD - Continuous Delivery**

**CD - Continuous Deployment**

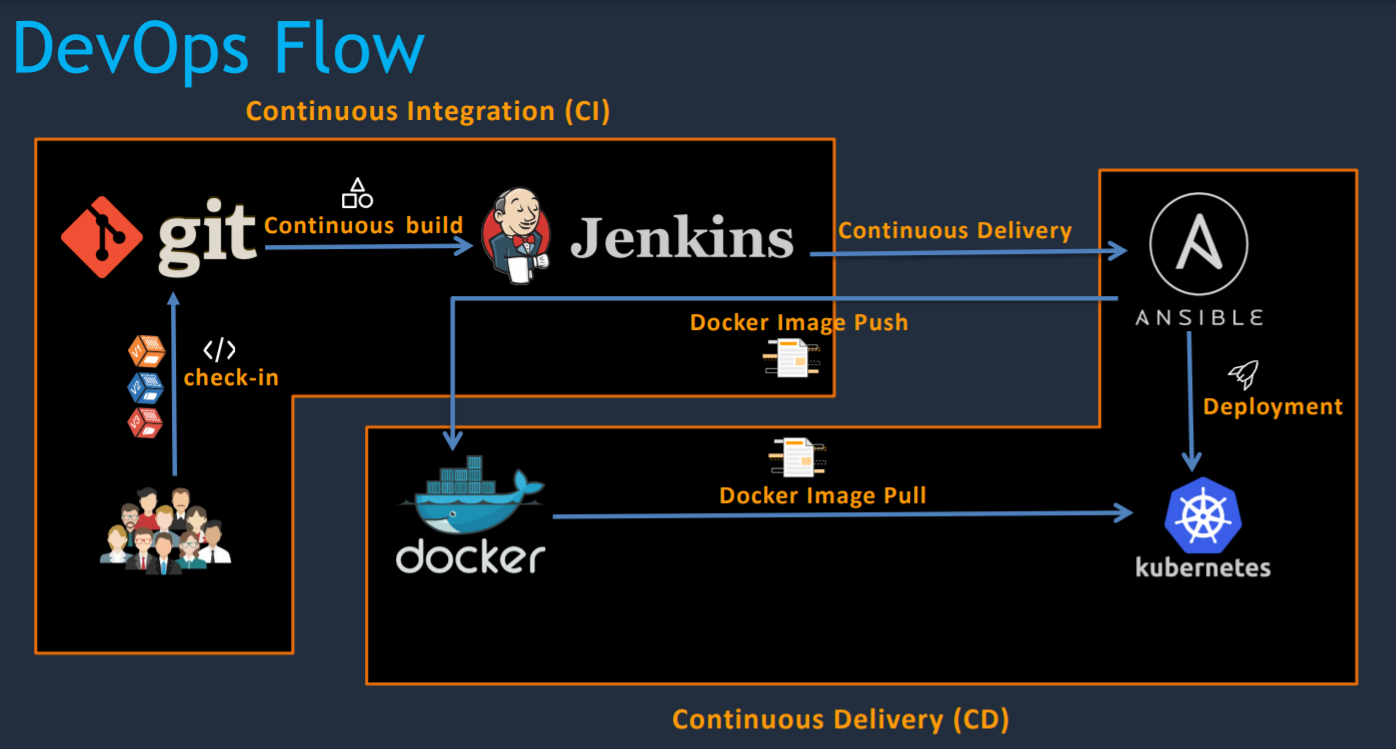
**Introduction**

Developer pushes their code onto source code management system (In our case GIT is source code management system). Once code is available to source code management system CI tool could able to identify and pull code automatically, run, build and unit test. Here our CI tool is Jenkin. At the end of successful continuous integration process, we could able to get an artifact (WAR). Deploying this artifact on your targeted environment (We called staging to this deploying artifact process and environment we can say for tomcat) we called continuous delivery. On staging environment, we can do our functional (Done by testing team) and performance (Done by NFT team) testing. Once testing looks fine we can deploy this to production environment. Doing deployment on production environment without manual intervention is called continuous deployment.

Continuous Integration – Process up to generating artifact (WAR).

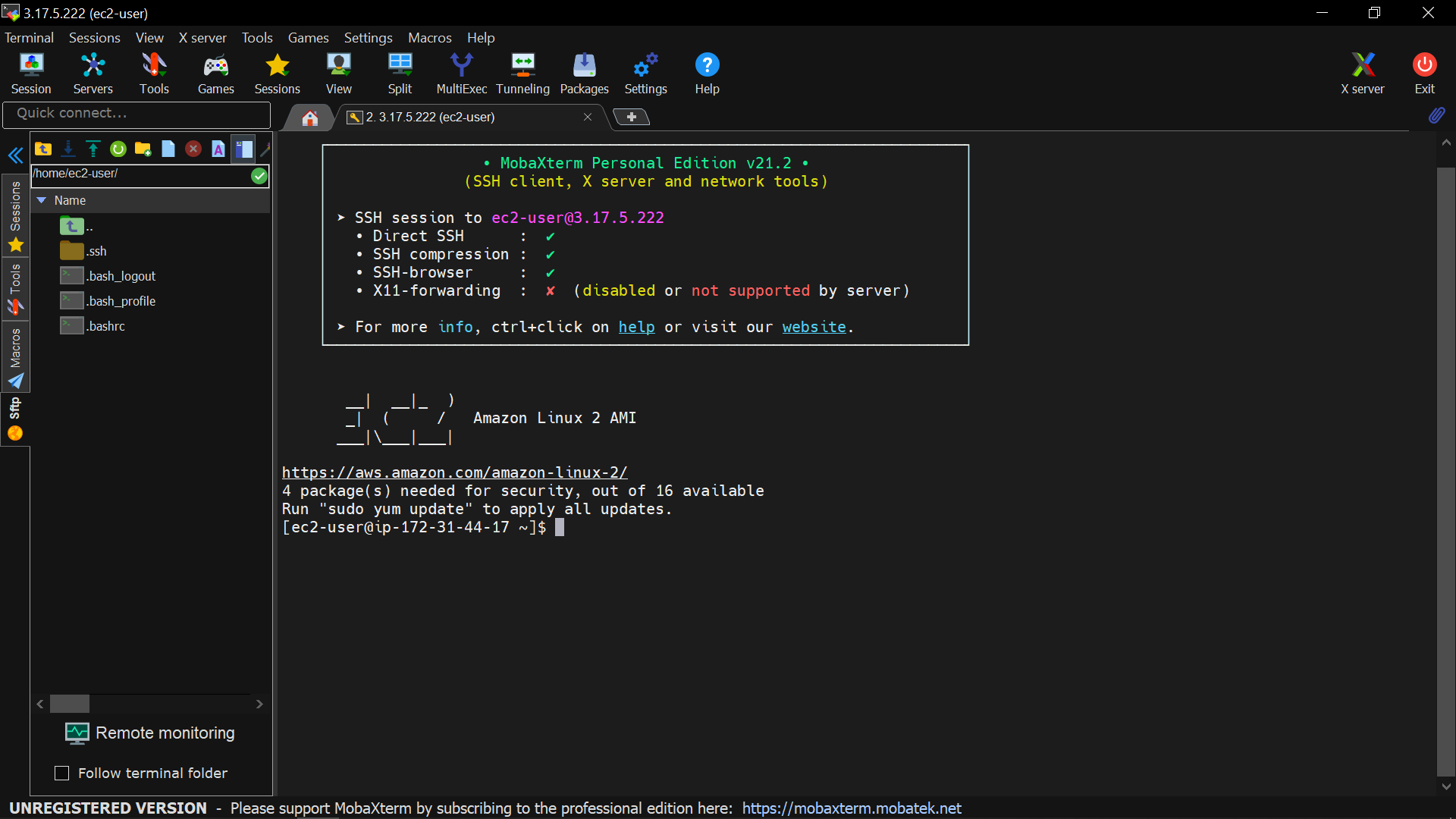
Continuous Delivery – Artifact (WAR) get deployed on targeted environment.

Continuous Deployment - Continuous Integration + Continuous Delivery.

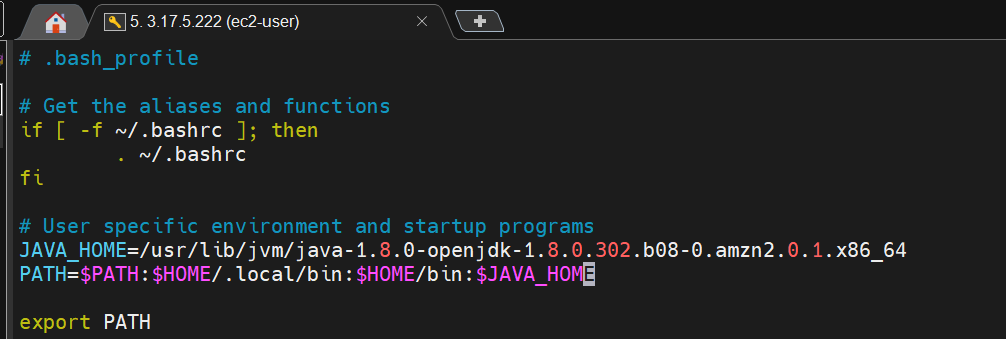


**Install Jenkins on AWS EC2**

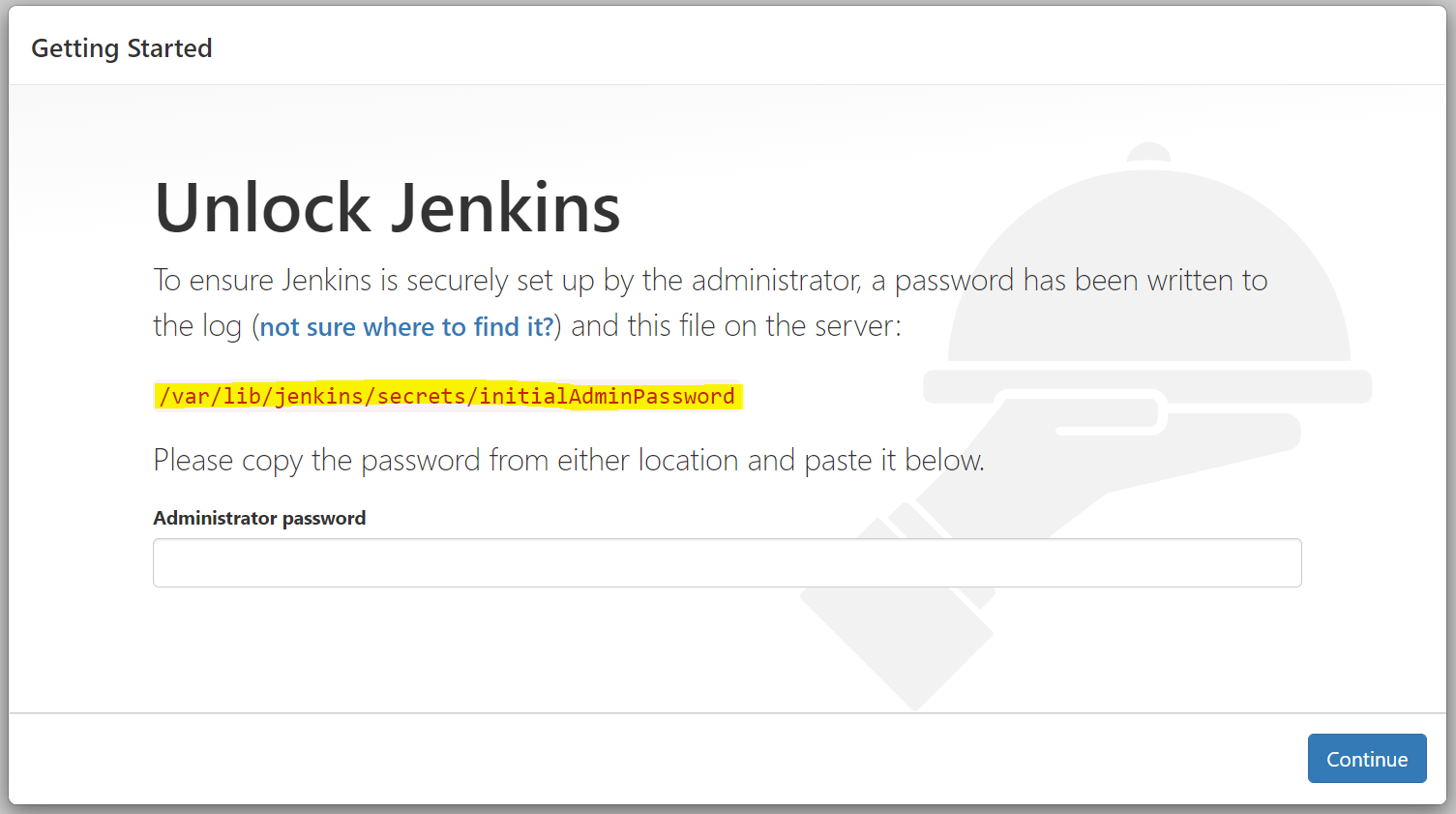
* Log In to your AWS account.
* Click on Services 🡪 Click on EC2 🡪 Click on Launch Instance 🡪 Select Amazon Linux 2 AMI (HVM), SSD Volume Type 🡪 Click on Next: Configure Instance Details 🡪 Click on Next: Add storage 🡪 Click on Next: Add Tags 🡪 Click on Add Tag 🡪 In Key write “Name” 🡪 In Value Write “Jenkins\_Server” 🡪 Click on Next: Configure Security Group 🡪 In Security group name write “DevOps\_Project\_SG” 🡪 In Description write “DevOps\_Project\_SG” 🡪 Click on Add Rule 🡪 In Port Range write “8080” 🡪 Click on Review and Launch 🡪 Click on Launch 🡪 Click on “Create a new key pair” 🡪 Give key pair name “DevOps\_Project” 🡪 Click on Download key pair 🡪 Click on Launch Instances 🡪 Click on View Instances 🡪 Click on created Instance 🡪 Copy public IP address.
* Download and install MobaXterm (We can use putty also but in MobaXterm we no need to convert pem file to ppk)
* Open MobaXterm 🡪 Click on session 🡪 Click on SSH 🡪 Add copied public IP address from AWS into Remote host 🡪 Click on Advhance SSH settings 🡪 Click on use private key 🡪 Browse downloaded pem file 🡪 Click on specify Username 🡪 write “ec2-user” 🡪 Click on OK 🡪 we able to see below screen



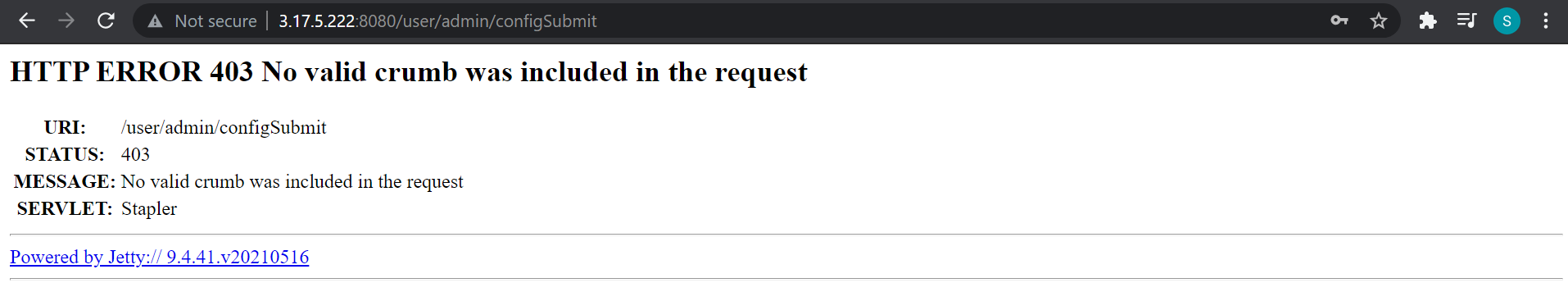
* Be a root user because we are installing many setups.
* To became root user type command: sudo su –
* CMD: clear 🡪 To clear screen data
* CMD: java -version 🡪 To check java version
* CMD: yum remove java-1.7.0.\* 🡪 To remove java 1.7 version
* CMD: yum install java-1.8\* 🡪 To install java 1.8
* To set java into environment variable:
* Find java path 🡪 CMD: find /usr/lib/jvm/java-1.8\* | head -n 3 🡪 Copy last entry 🡪 vi .bash\_profile 🡪 add Java home and that java home in PATH as shown in below screen shot



* To affect this environment variable, restart the session.
* CMD: echo $JAVA\_HOME 🡪 we can check java home path
* We cannot install Jenkins directly. First we need to add Jenkins code from repository.
* On google search for download Jenkins 🡪 click on first link which ends with .io 🡪 under LTS (Long Term service) use Red HAT 🡪 Follow commands one by one (This may change in future, below command is not final visit Jenkins’s website for latest command)
* sudo wget -O /etc/yum.repos.d/jenkins.repo <https://pkg.jenkins.io/redhat-stable/jenkins.repo>
* sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io.key>
* yum install Jenkins
* CMD: service jenkins status 🡪 To check Jenkins is start or stop
* If Jenkins disabled or inactive or stopped then start it
* CMD: service jenkins start 🡪 to start Jenkins
* To check Jenkins on browser take public IP of our aws and port number of that and execute on brower e.g. 3.17.5.222:8080
* After executing formed url you able to see below



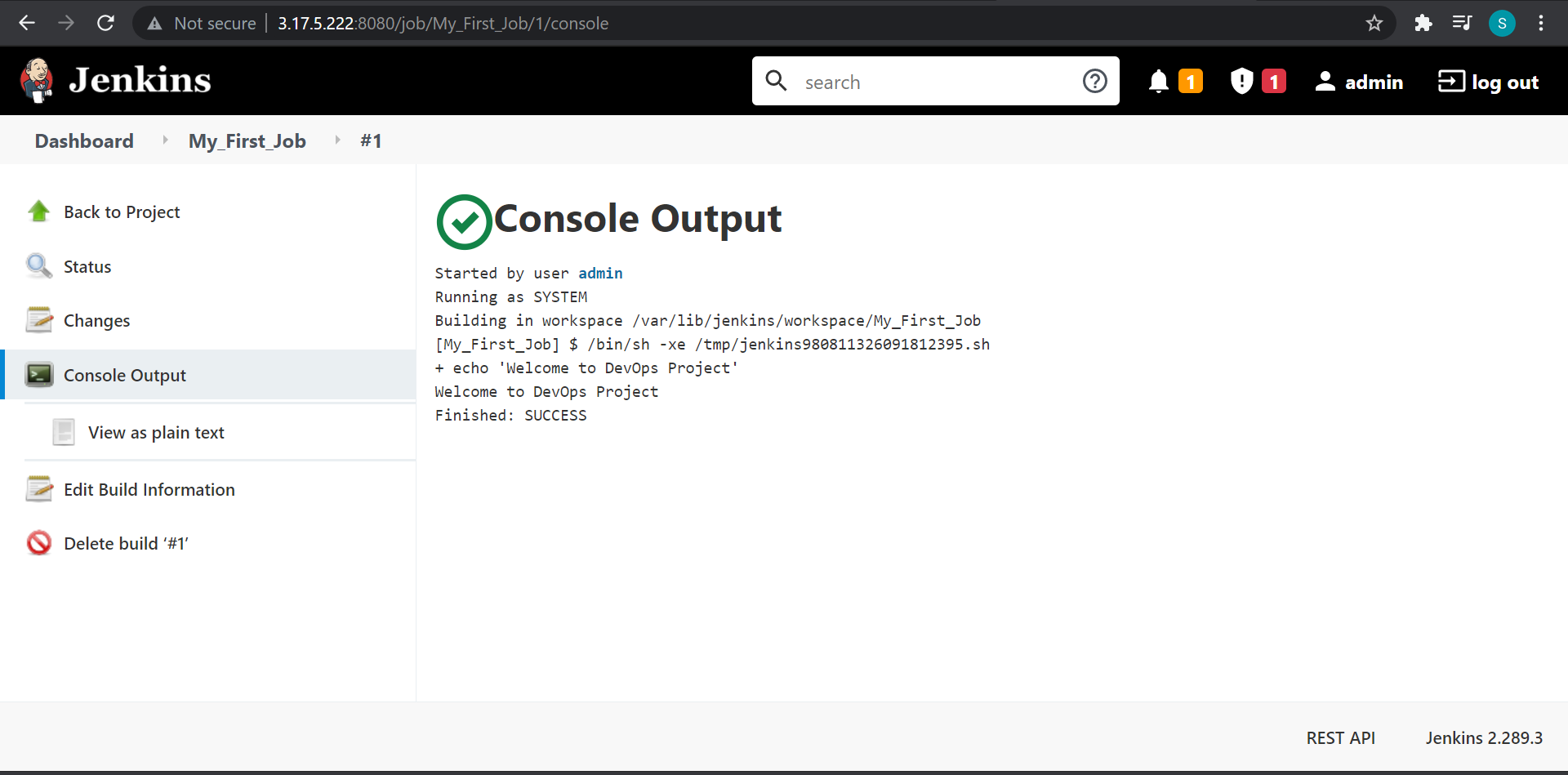
* We can find default autogenerated password on highlighted location from image.
* CMD: cat /var/lib/jenkins/secrets/initialAdminPassword 🡪 to open file 🡪 copy password and use it on browser 🡪 Click on continue 🡪 Close Customize Jenkins window 🡪 Click on start using Jenkins .
* We have to change password 🡪 Click on admin 🡪 Click on configure 🡪 Change password 🡪 Click on apply 🡪 Click on Save 🡪 After successful save will get logged out and able to see below message



* Reload website i.e., <http://3.17.5.222:8080/>
* Enter new username: admin and password: \*\*\*\*\*\*
* We need to set java home path in Jenkins
* Click on manage Jenkins 🡪 Click on Global Tool Configuration 🡪 Click on Add JDK 🡪 In name write “JAVA\_HOME” 🡪 in JAVA\_HOME write java home from session 🡪 Click on Apply

**Run First Jenkins Job**

* Go to UI 🡪 Click on dashboard 🡪 Click on New Item 🡪 Give name “My\_First\_Job” 🡪 Click on free style project 🡪 Click on OK.
* Write description under description 🡪 under build select batch and shell as per your requirement.
* In my case I am using Execute shell.
* Under commands add shell script command e.g., echo "Welcome to DevOps Project"
* Click on Apply 🡪 Click on Save 🡪 once complete with job part Click on Build Now 🡪 After clicking on build now we can see in build history 🡪 Click on build from build history 🡪 Click on Console output to see console output 🡪 below window we can see



**Install and setup git**

* Go to AWS session console
* CMD: yum install git -y 🡪 Install git
* Go to Jenkins UI 🡪 Click on dashboard 🡪 Click on Manage Jenkins 🡪 Click on manage plugins 🡪 Click on Available 🡪 Search GitHub 🡪 Click on GitHub checkbox 🡪 Click on install without restart 🡪 After installation click on Jenkins’s logo to go on homepage
* After installing git plugin git option will get enable in Global Tool Configuration.
* Click on Manage Jenkins 🡪 Global Tool Configuration 🡪 Under Git section 🡪 give “github” name under Name 🡪 gave path “/usr/bin/git” under Path to Git executable 🡪 Click on apply 🡪 Click on save

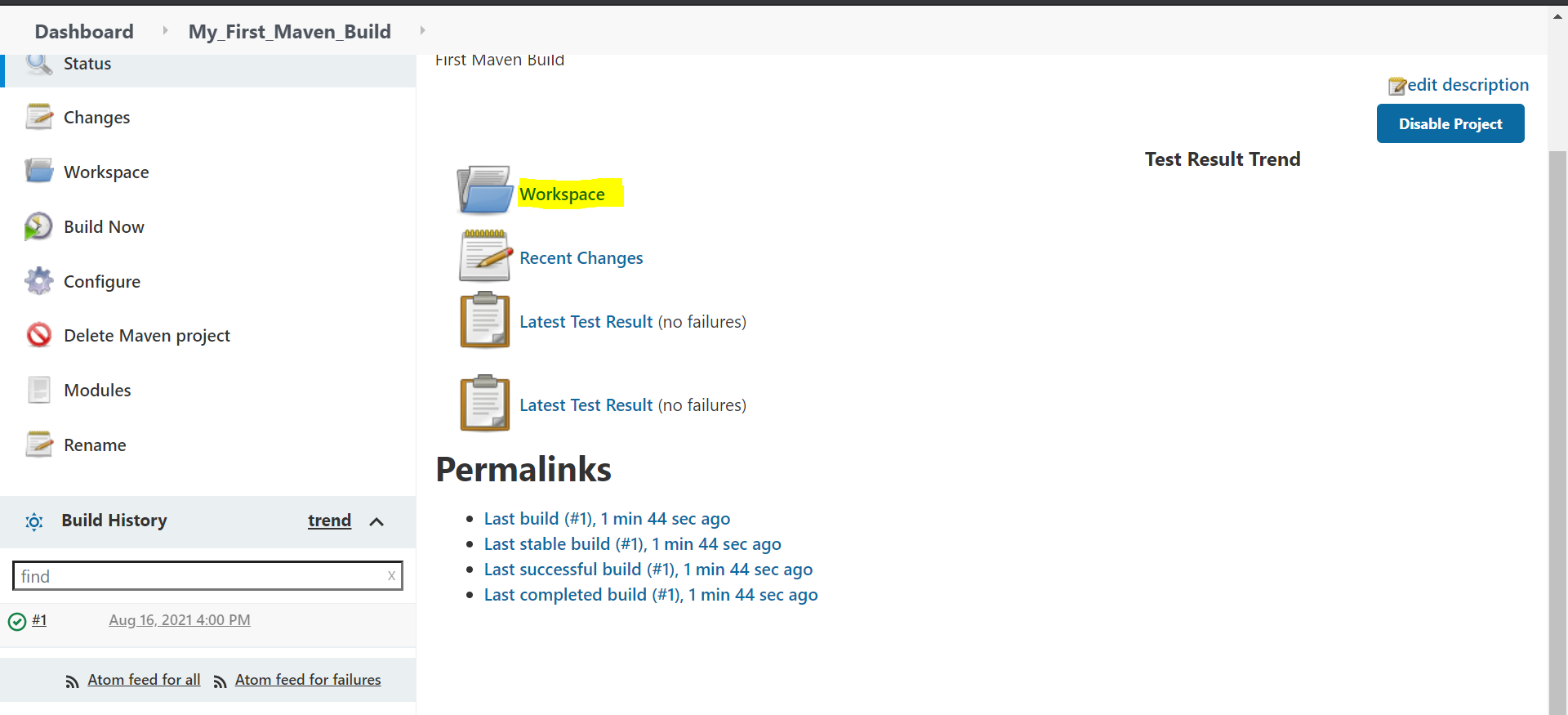
**Maven Setup**

* Search download maven on google
* Copy Binary tar.gz archive link
* Go to Jenkins server console
* CMD: cd /opt 🡪 to go opt directory location
* CMD: wget <https://dlcdn.apache.org/maven/maven-3/3.8.2/binaries/apache-maven-3.8.2-bin.tar.gz> 🡪 To download maven
* CMD: tar -xvzf apache-maven-3.8.2-bin.tar.gz 🡪 To unzip .tar.gz file
* CMD: mv apache-maven-3.8.2 maven 🡪 Rename apache-maven-3.8.2 to maven
* CMD: vi ~/.bash\_profile 🡪 To edit .bash\_profile(~ is for go to root directory)
* Add M2=/opt/maven/bin
* Add M2\_HOME=/opt/maven
* Add to path $M2 and $M2\_HOME 🡪 press esc 🡪 type :wq 🡪Enter
* Exit 🡪 Relogin to jenkin server console

Go to Jenkins UI 🡪 Dashboard 🡪 Manage Jenkins 🡪 Manage plugins 🡪 Available 🡪 Check Maven integration and Maven invoker 🡪 install without restart🡪 After complete go to dashboard 🡪 Manage Jenkins 🡪 Global Tool Configuration 🡪 Add Maven 🡪 write M2\_HOME 🡪 /opt/maven 🡪 Apply 🡪 Save

**Create First Maven Job**

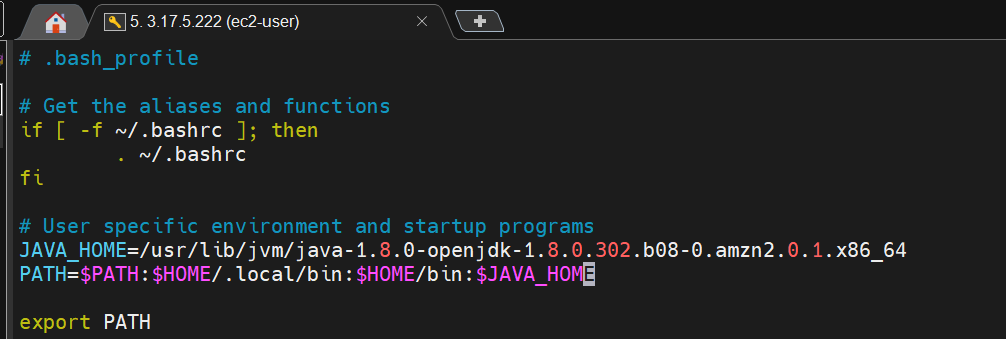
* Go to Jenkins UI 🡪 Dashboard 🡪 New Item 🡪 Give name My\_First\_Maven\_Build 🡪 select maven project 🡪 OK🡪 give description 🡪 Under source code management select git 🡪 paste your git repo url 🡪 it is public repo no need to add credential 🡪 give branch code in my case its Master 🡪 Under build section write pom.xml 🡪 In goals and option write “clean install package” 🡪 Apply 🡪 Save 🡪 Click on build now
* When you run any job Jenkins copy your code in workspace directory even outcome copy in workspace directory



* To check on AWS Jenkins console 🡪 cd /var/lib/jenkins/workspace/

**Setup tomcat server**

* We have to create new AWS instance for amazon server.
* Go to Aws 🡪 Services 🡪 EC2 🡪 Launch Instance 🡪 Select Amazone linux 2 64bit 🡪 Select 🡪 Next: configure Instance Details 🡪 Next: Add Storage 🡪 Next: Add Tags 🡪 Add Tag 🡪 In Key “Name” 🡪 In Value “Tomcat\_Server” 🡪 Next: Configure Security Group 🡪 Review and Launch 🡪 Click on Select existing security group 🡪Select DevOps\_Project\_SG 🡪 Review and Launch 🡪 Launch 🡪 Choose an existing Key pair 🡪 Launch Instance 🡪 View Instance
* Go to MobaXtream 🡪 Session 🡪 New Session 🡪 in Remote hoste add public IP of tomcat server 🡪 Advance SSH settings 🡪 click on use private key🡪 Browse .pem file 🡪 click on specify username 🡪 give ec2-user 🡪 Click Ok
* Install java on tomcat server session console 🡪 yum install java-1.8\*
* To set java into environment variable:
* Find java path 🡪 CMD: find /usr/lib/jvm/java-1.8\* | head -n 3 🡪 Copy last entry i.e /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.302.b08-0.amzn2.0.1.x86\_64 🡪 vi .bash\_profile 🡪 add Java home and that java home in PATH as shown in below screen shot



* Search on google download tomcat 🡪 select tomcat 8 server 🡪 copy gz.tar file link 🡪 Go to Tomcat server session console 🡪 CMD: sudo su - 🡪 CMD: wget <https://downloads.apache.org/tomcat/tomcat-8/v8.5.69/bin/apache-tomcat-8.5.69.tar.gz> 🡪 tar -xvzf apache-tomcat-8.5.69.tar.gz 🡪 mv apache-tomcat-8.5.69 tomcat 🡪cd /tomcat/bin/ 🡪 sh ./startup.sh 🡪 After tomcat start go to web browser 🡪 copy public IP of tomcat server and form url like “Public IP:8080” 🡪 You able to see tomcat web page.
* If you click on Manager App on browser it shows 403 forbidden issue, To solve this issue 🡪 go to tomcat server session 🡪 find context file with CMD: find / -name context.xml 🡪 we not using context file from conf we using context file from other place🡪 CMD: vi /opt/tomcat/webapps/manager/META-INF/context.xml 🡪 press i 🡪 comment <Valve className="org.apache.catalina.valves.RemoteAddrValve"

allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> 🡪 do this for other context file 🡪 go to tomcat location 🡪 sh ./shutdown.sh 🡪 sh ./startup.sh🡪 go to browser 🡪 refresh tomcat webpage 🡪 click on Manager App🡪 it will ask username and password 🡪 To add username and password go to tomcat server session console 🡪 CMD: cd /opt/tomcat/conf/ 🡪 CMD: vi tomcat-users.xml 🡪 press i 🡪 add below user 🡪

<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"/>

* Save 🡪 Stop tomcat 🡪 start tomcat

**Deploy a war file on tomcat server using jenkin sever**

* Go to Jenkins UI 🡪 Go to dashboard 🡪 Manage Jenkins 🡪 Manage Plugins 🡪 Available 🡪 search for Deploy to container 🡪 Install without container 🡪 Go to dashboard 🡪 New Item 🡪 Name it “Deploy\_on\_Tomcat\_Server” 🡪 Maven Project 🡪 Ok 🡪 Give description 🡪 select Git 🡪 Add repository <https://github.com/ShubhamLekule/hello-world.git> 🡪 Under Goals and Option 🡪 clean install package 🡪 Under post build action select deploy war/ear to a container 🡪 WAR/EAR files give “\*\*/\*.war” 🡪 select tomcat 8 container 🡪 Click on add 🡪 select Jenkins 🡪 username “deployer” 🡪 password “deployer” 🡪 Id “deployer\_user” 🡪 Description “user to deploy on tomcat VM” 🡪 Add 🡪 select deployer user 🡪 under tomcat URL add tomcat server url eg. <http://3.139.235.156:8080/> 🡪Apply 🡪 Save 🡪 Click on build Now 🡪 After build success use url <http://3.139.235.156:8080/webapp/> to check.

**Deploy on VM through Poll SCM**

* Go to Jenkins dashboard 🡪 open existing Jenkins’s job i.e., Deploy\_on\_Tomcat\_Server 🡪 Configure 🡪select Poll SCM under build trigger option 🡪 under schedule add cron job 🡪 for every minute use “\* \* \* \* \*” 🡪 Apply 🡪 Save
* After changing in project and committing and pushing to git 🡪 in next minute automatically Deploy\_on\_Tomcat\_Server job initiated.