Cloud session 19/03/2022

Let's start

→ Created a new Maven project

→ Ibrahim's teacher shared the pom .xml file with us (below are the dependencies just as a reference)

```
<dependencies>

    <dependency>
      <groupId>io.rest-assured</groupId>
      <artifactId>rest-assured</artifactId>
      <version>4.3.0</version>
    </dependency>
    <dependency>
      <groupId>org.json</groupId>
      <artifactId>json</artifactId>
      <version>20190722</version>
    </dependency>
    <dependency>
      <groupId>org.testng</groupId>
      <artifactId>testng</artifactId>
      <version>7.1.0</version>
    </dependency>
    <dependency>
      <groupId>com.google.code.gson</groupId>
      <artifactId>gson</artifactId>
      <version>2.8.6</version>
    </dependency>
    <dependency>
      <groupId>org.codehaus.jackson</groupId>
      <artifactId>jackson-mapper-asl</artifactId>
      <version>1.9.13</version>
    </dependency>
    <dependency>
      <groupId>org.seleniumhq.selenium</groupId>
      <artifactId>selenium-java</artifactId>
      <version>3.141.59</version>
    </dependency>
    <dependency>
```

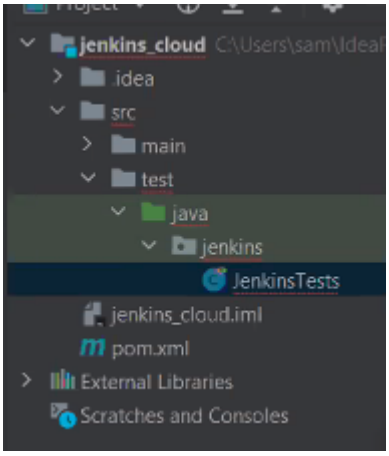
```
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.13</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager -->
    <dependency>
        <groupId>io.github.bonigarcia</groupId>
        <artifactId>webdrivermanager</artifactId>
        <version>4.2.2</version>
    </dependency>

  </dependencies>

  <build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-surefire-plugin</artifactId>
            <version>3.0.0-M5</version>
        </plugin>
    </plugins>
  </build>
```

→ we created a “jenkins” package under the java folder under test
→ we created a “JenkinsTests” class in this package



→ We started typing in the “JenkinsTests” class...

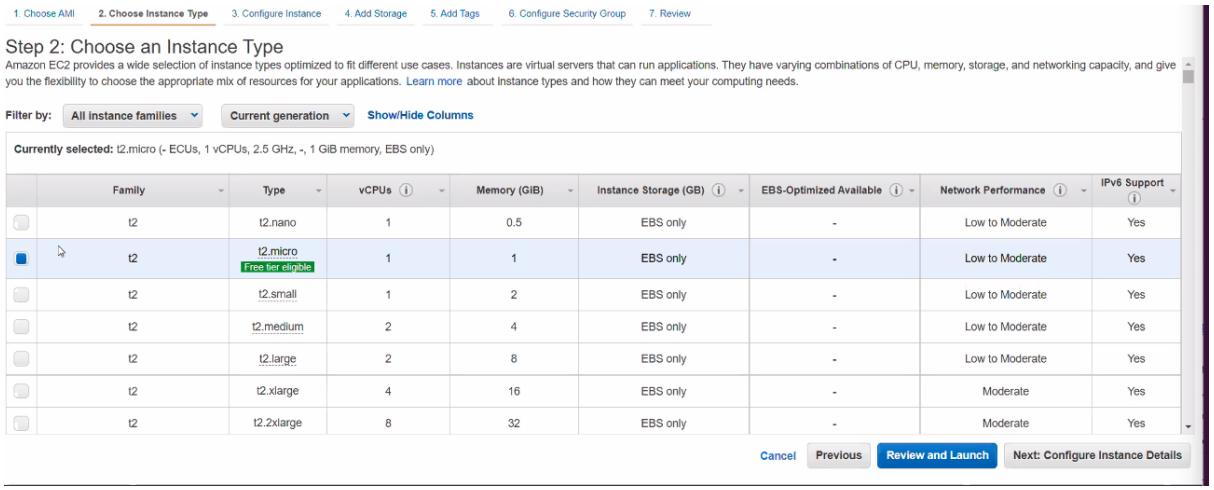
Once ready there, we opened our AWS - as a root user. Then →

Q: Hocam in a team Do all members of the Team use the Jenkins or Aws or just Lead?

A: Everybody uses it.

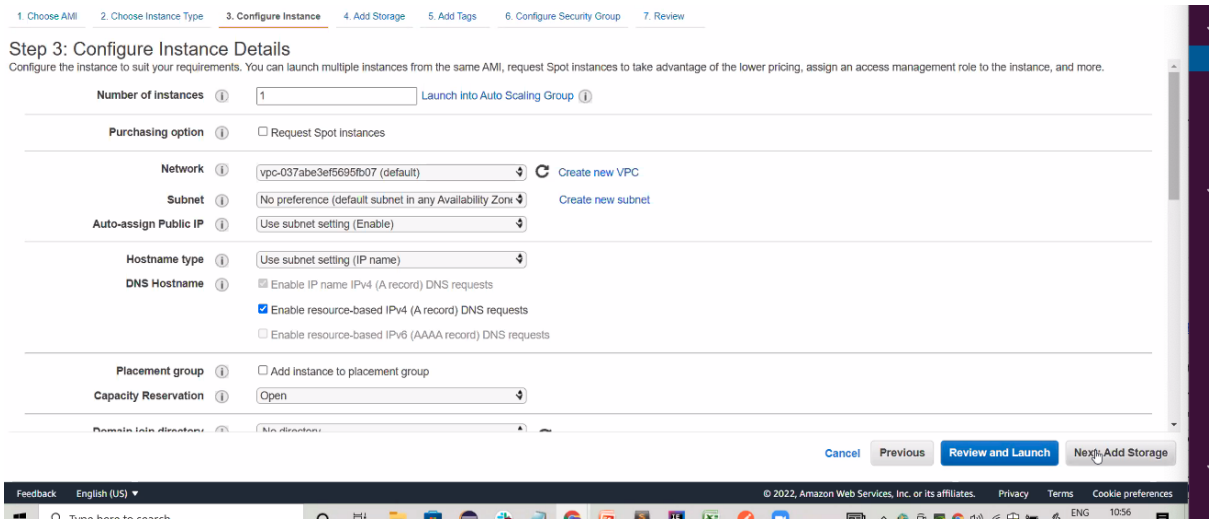
We created a new Instance → Instances → Launch New Instance → Ubuntu →

Then:



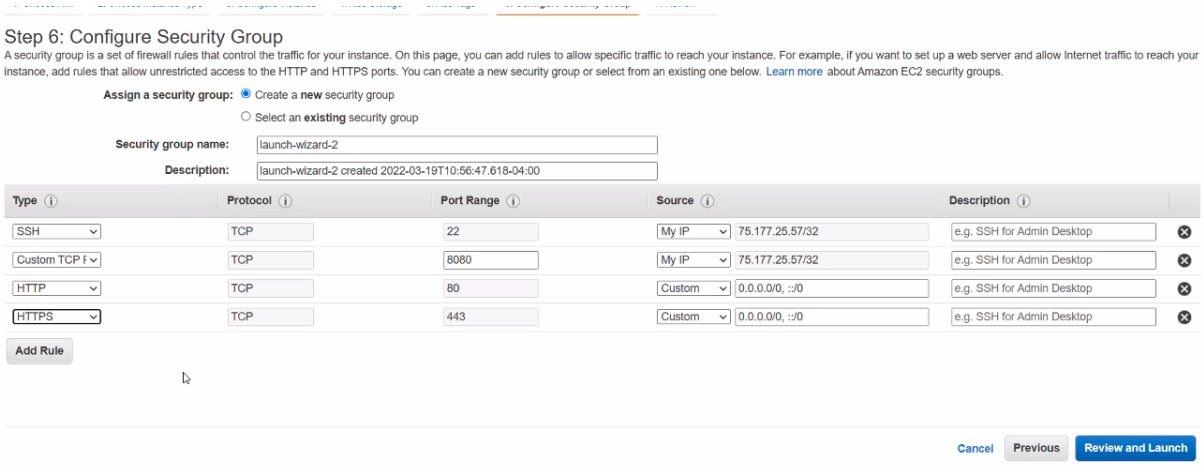
→ Select the free tier

→



→ Then we just make the size 16

→ Give a name → Then click on the grey Configure button



→ Review and Launch

→ Choose an existing key pair or create a new one

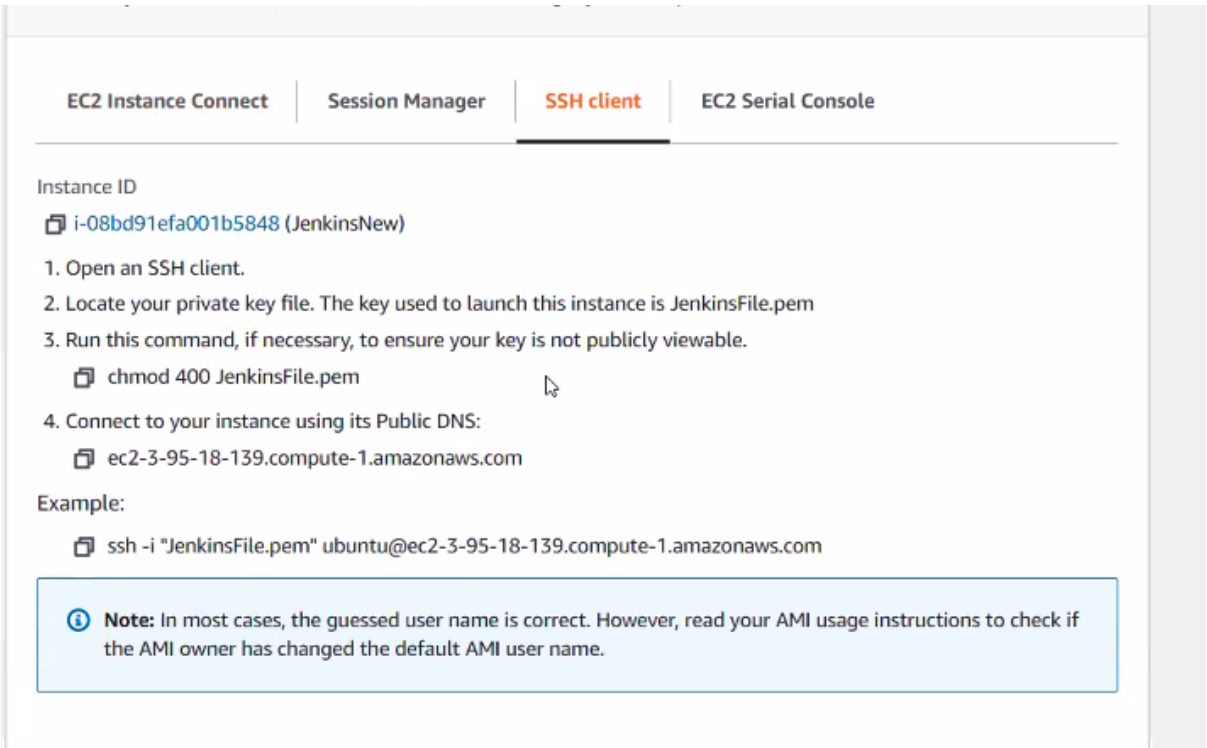
→ Then we are done - it can say pending, refresh it and it will be done

For Mac users →

- Open the terminal
- Find the location of your Jenkins file - We type cd and space and drag and drop it to the terminal and delete the file extension, so we're in the right location, hit enter
- Then we type "chmod 400 filename.pem"

→ Now back to AWS → on our instance we click on connect:

We see this page when we click on SSH client:



→ We copy the ssh bit in Example: and then put it in our terminal and hit enter, it should show something like:

```
1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-31-217:~$
```

→ Then we need to update our tools, type these comment on the terminal and hit the enter

1. Update and get ready all tools and packages on Ubuntu Server

- sudo apt-get update
- sudo apt-get upgrade

First - “sudo apt-get update”, then hit enter
Second - “sudo apt-get upgrade”, then hit enter

→ Give it some time to work
→ Next - we need to install the libraries, continue on the terminal:

First command: “sudo apt-get install -y libappindicator1 fonts-liberation” and hit enter
Second command: “wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb” and hit enter
Third command: “sudo dpkg -i google-chrome*.deb” and hit enter
Forth command: “sudo apt-get install -f” and hit enter
Fifth command: we rerun this one “sudo dpkg -i google-chrome*.deb” and hit enter

1. Install latest Chrome Binary on Ubuntu Server

- sudo apt-get install -y libappindicator1 fonts-liberation
- wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
- sudo dpkg -i google-chrome*.deb
- sudo apt-get install -f

→ we go to google → search for “chromedriver download” and needs to be the same as ours, we need this:
https://chromedriver.storage.googleapis.com/index.html?path=99.0.4844.51/

2. Install latest Chrome Driver on Ubuntu Server

- sudo apt install unzip
- wget https://chromedriver.storage.googleapis.com/99.0.4844.35/chromedriver_linux64.zip
- unzip chromedriver_linux64.zip
- sudo mv chromedriver /usr/bin/chromedriver

→Go back to terminal and put the command “sudo apt install unzip” and hit the enter
→ Then this command: “wget https://chromedriver.storage.googleapis.com/**99.0.4844.35**/chromedriver_linux64.zip” and hit enter
→ Then put this command “unzip chromedriver_linux64.zip” and hit enter
→ Then put this command “sudo mv chromedriver /usr/bin/chromedriver” and hit enter

Nex stepst:

1) We’re installing Java now → put and hit enter “sudo apt-get install default-jdk”
2) We’re installing Maven → put and hit enter “sudo apt install maven”
3) We can install git (we have it already but just in case) → “sudo apt install git”

Main tool installations

- sudo apt-get install default-jdk
- sudo apt install maven
- sudo apt install git

Q: Do we need to install it every time in our instances?
A: We install them once, and use them again and again.

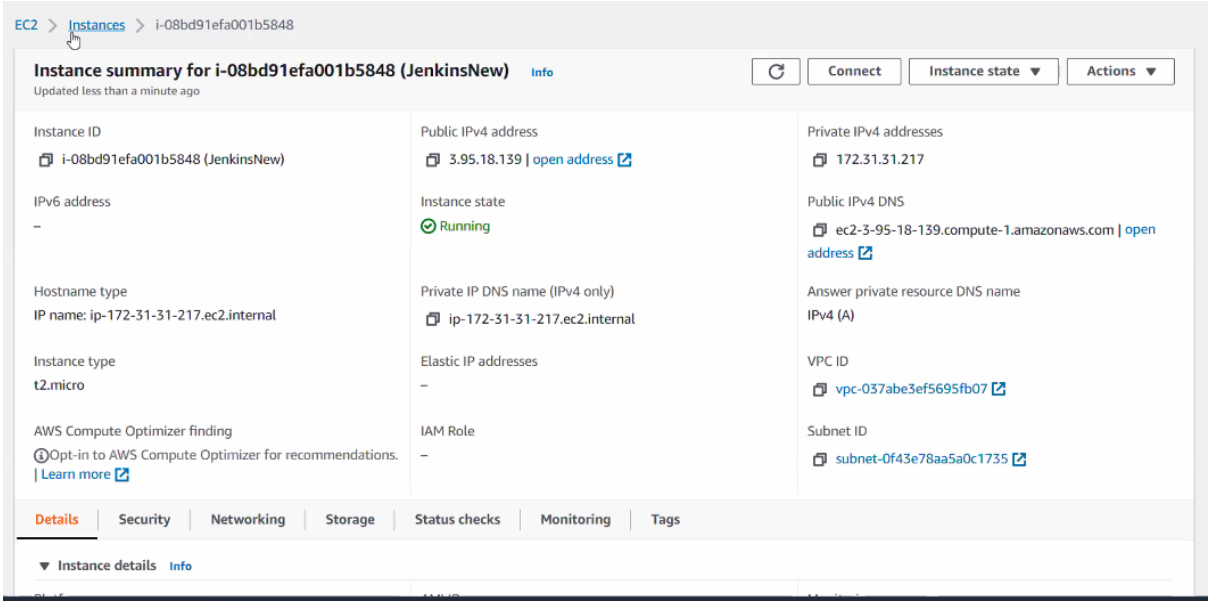
Now Jenkins installation:

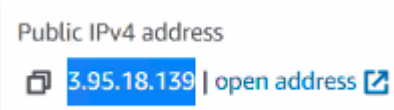
1) Type this and hit enter —> wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -" - we need to see OK message
2) Next: copy exactly as it is, even the single code —>

sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > \
/etc/apt/sources.list.d/jenkins.list'

3) Updating our tools with this one → sudo apt-get update
4) Final comment → sudo apt-get install jenkins

→ Next - we go to our Instance:

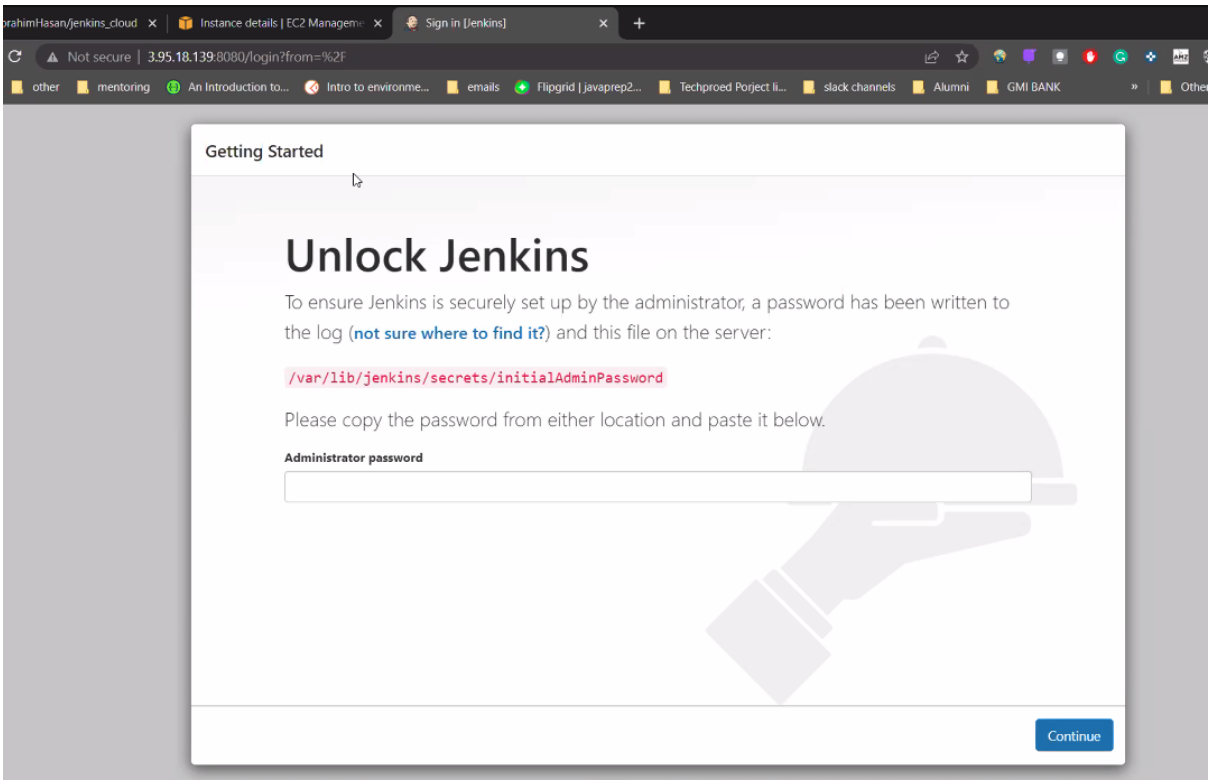




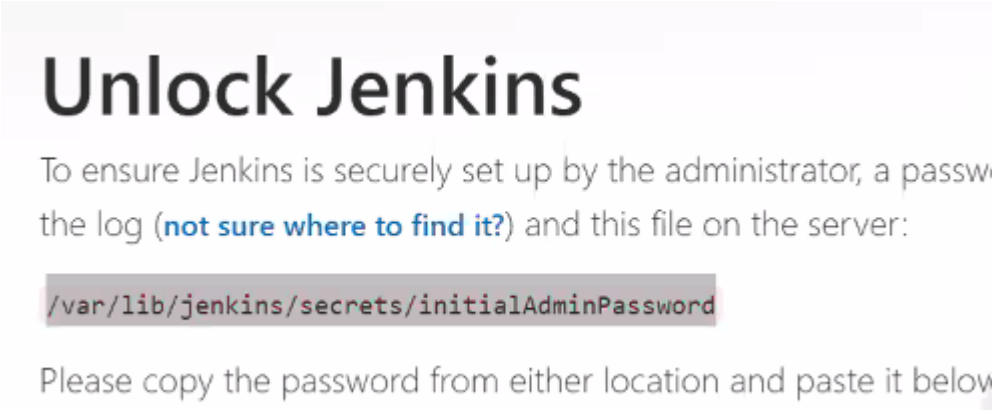
→ Public IP address → copy it

and paste it to your browser and add the :8080 to it, then hit enter

→ We should see this page:

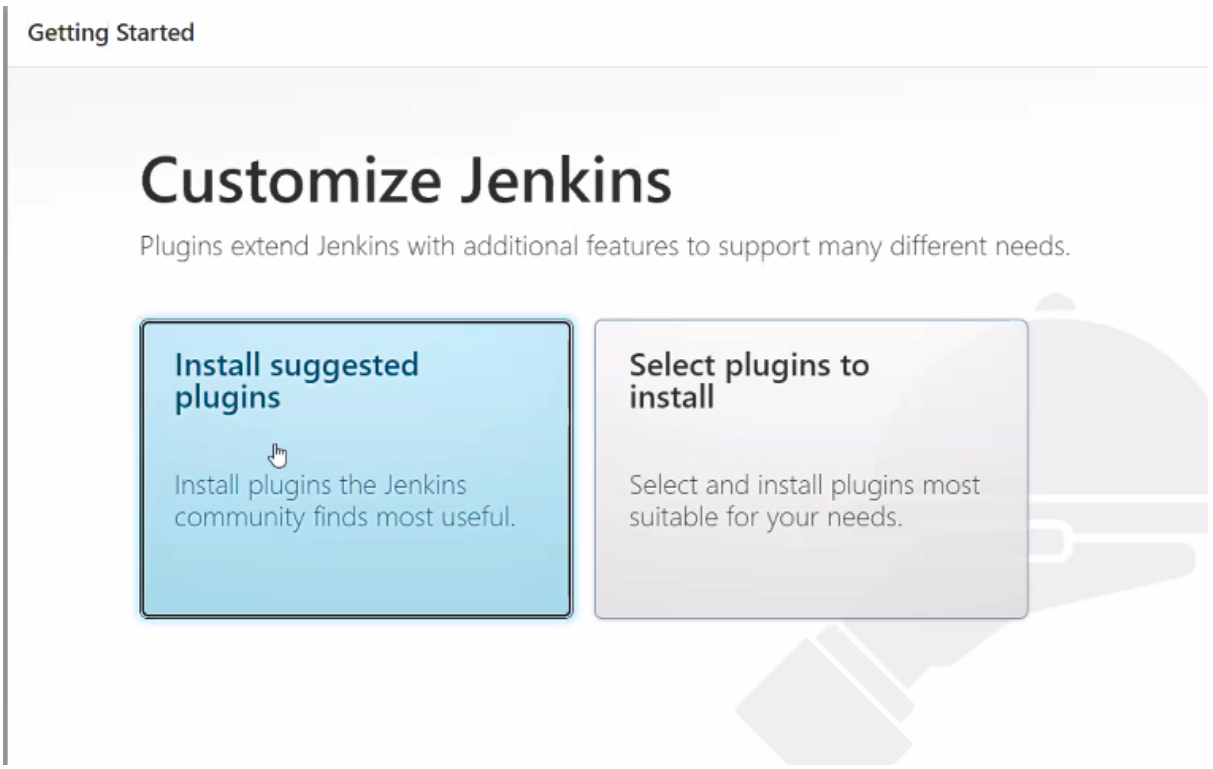


→ Then we copy this part:

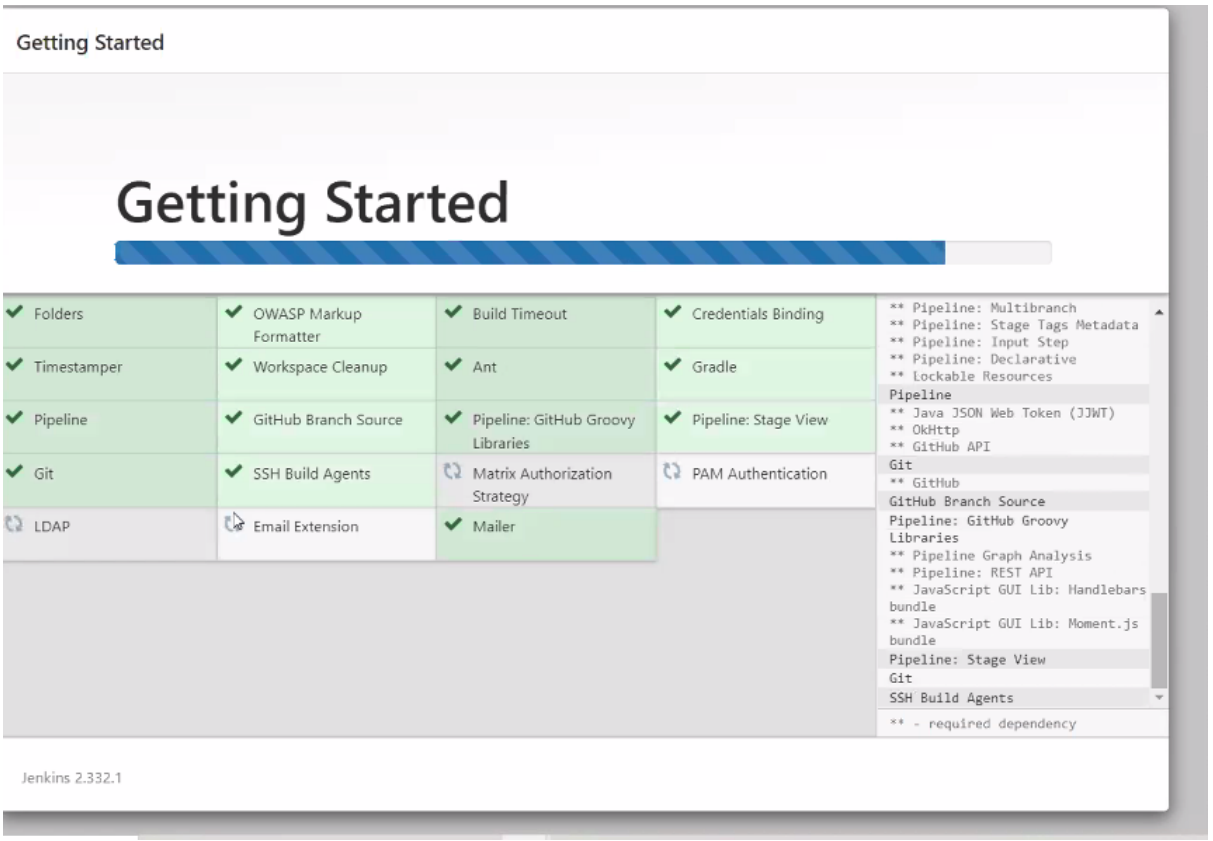


→ then add this bit and copy this to the terminal → sudo cat /var/lib/jenkins/secrets/initialAdminPassword

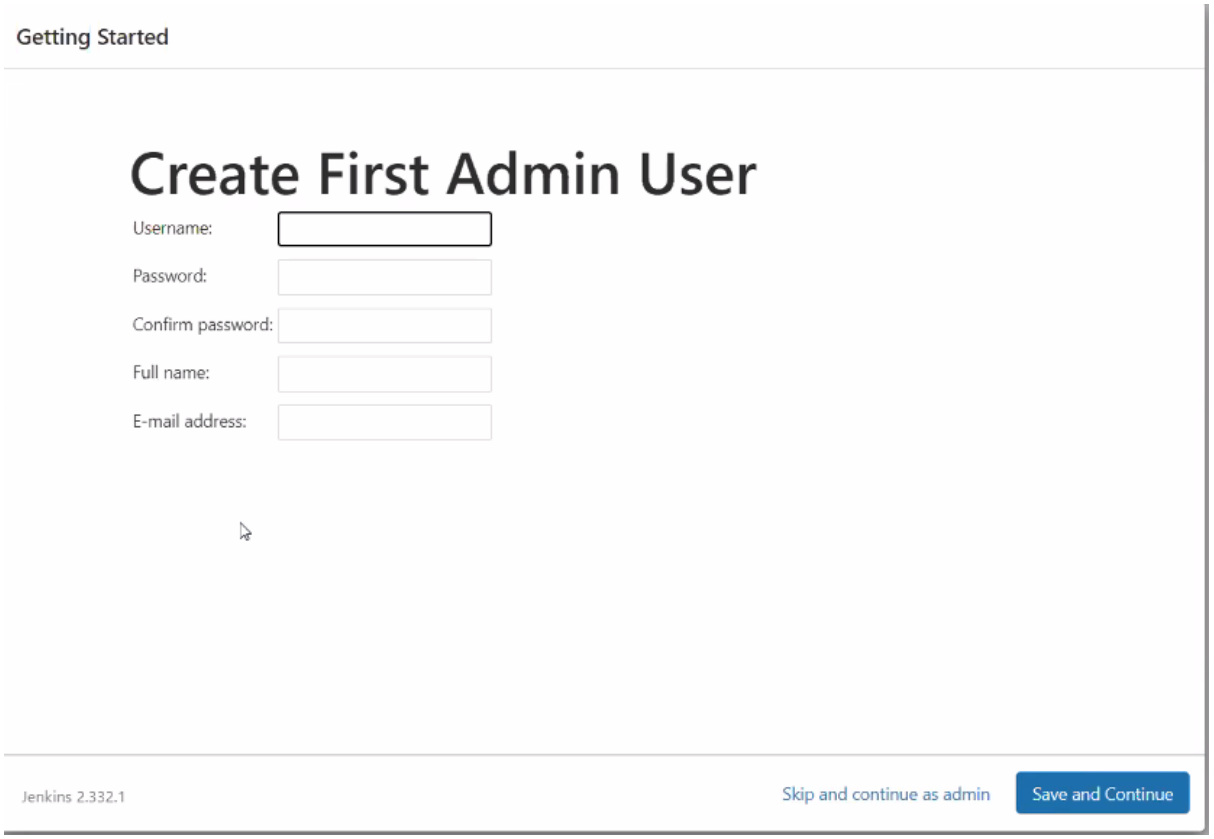
→ You'll get your pass on the terminal - then copy and paste it to the pass space on the Unlock Jenkins then choose the left box:



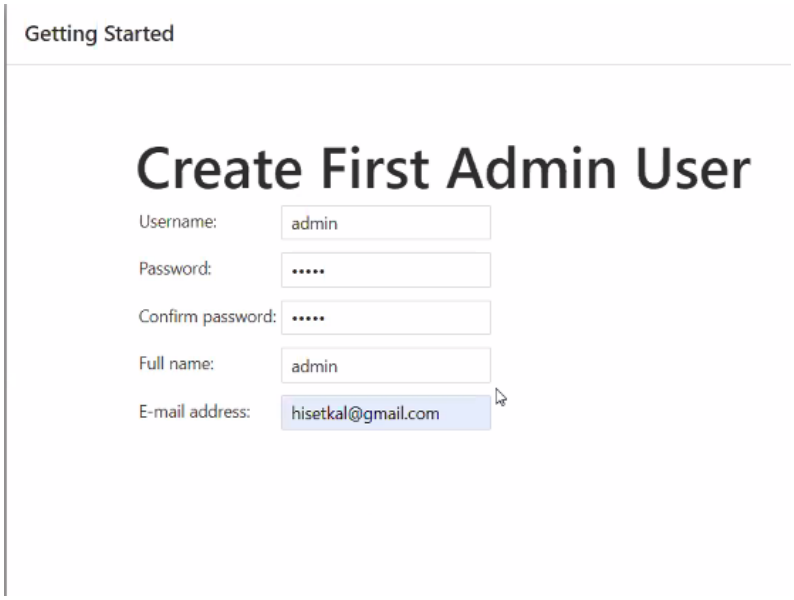
We'll see something like this:



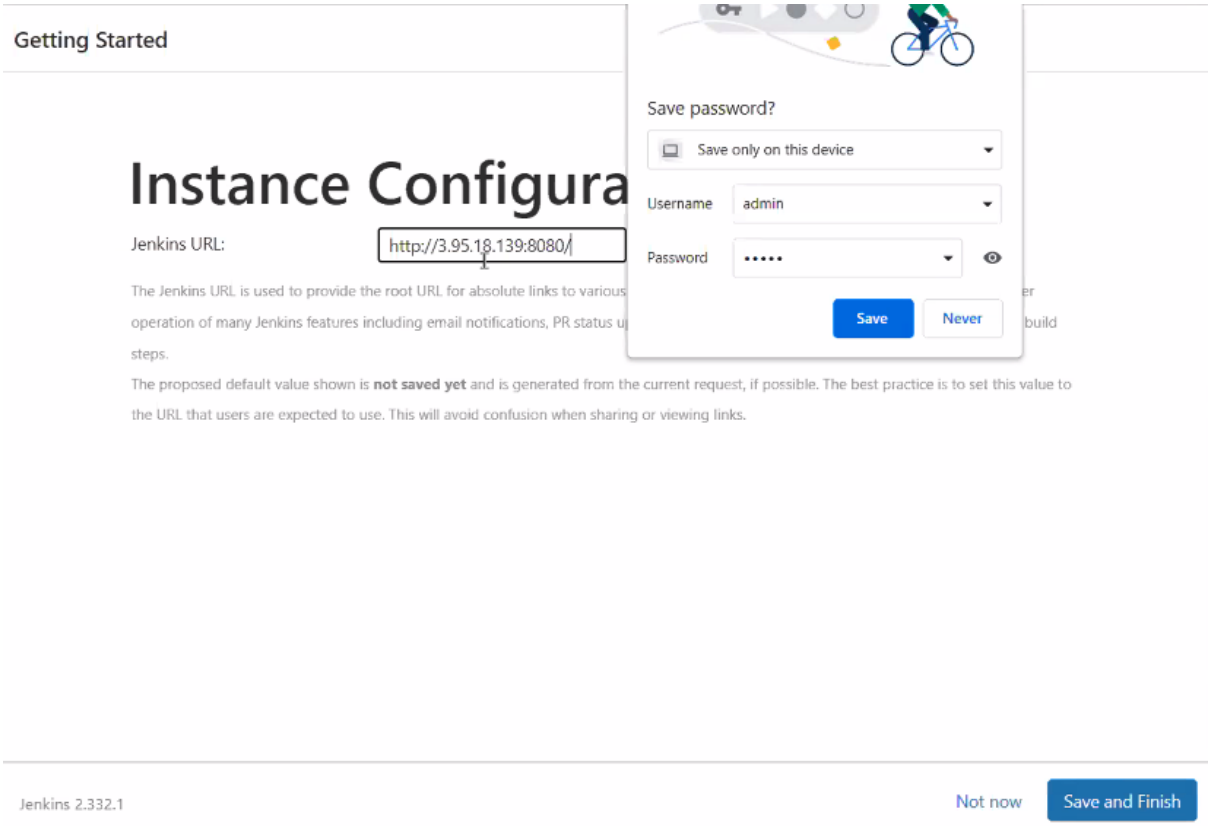
→ Once done we see this page:



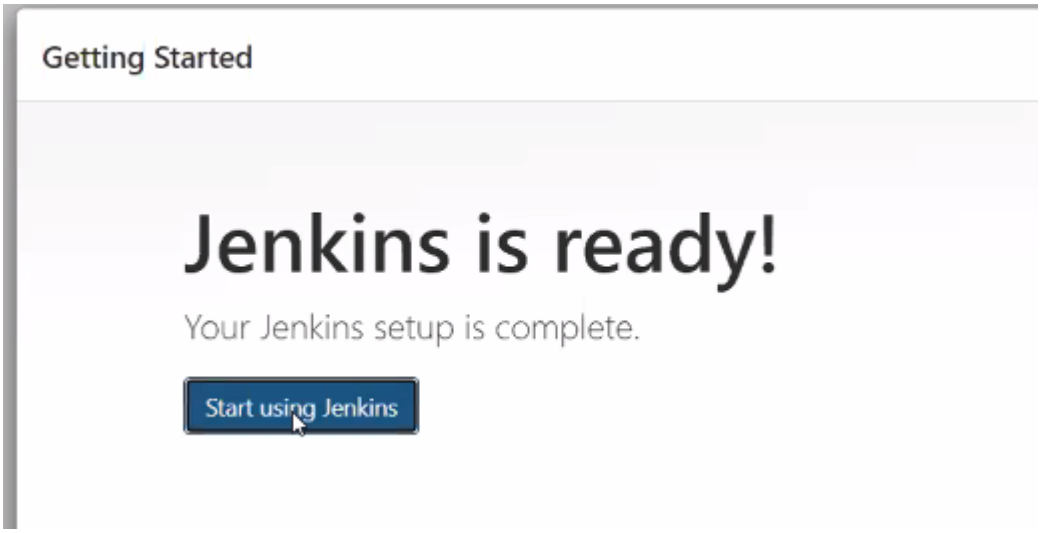
→ Put admin everywhere - so we'll never forget it :) Just the e-mail should be an actual one.



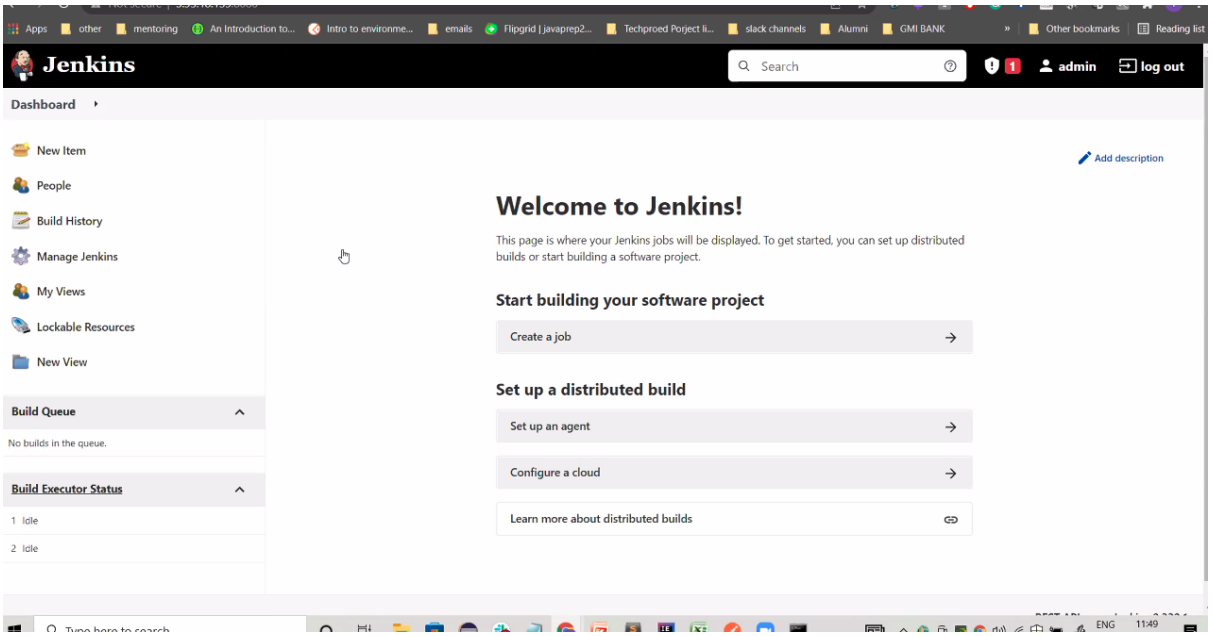
→ Then we hit the next button, we see this:



→ Then we see this page:



→ And we are ready with our Jenkins setup FINALLY 🚀



→ Next - we click on Manage Jenkins →
→ Click on “Global Tool Configuration”
→ we click on “Add JDK” - deselect the default install
→ as a Name, put : “JAVA_HOME”
→ Under it put this location - same for everyone → /usr/lib/jvm/java-11-openjdk-amd64
→ Git → name can be left as Default
→ path → /usr/lib/git-core/git
→ Hit Apply
→ Click on Add Maven
→ Name: MAVEN_HOME
→ Under it give a location→ /usr/share/maven
→ Click on Apply and then on Save
→ Click on Dashboard → Click on New Item → Enter a name: JenkinsProject
→ Choose the Freestyle project → hit OK
The we see this page:

Scroll down → Source Code Management → tick Git → URL will the URL from GitHub → https://github.com/HalilIbrahimHasan/jenkins_cloud.git (like your own project URL from Git - the one we use for cloning the project)
→ make sure the branch is correct

→ Build → Add Build Step → choose Invoke top-level Maven targets → then give the name MAVEN_HOME (it will appear on the drop-down) to the Maven Version field
→ Under it for Goals - write there → clean install
→ Then hit Apply and then Save

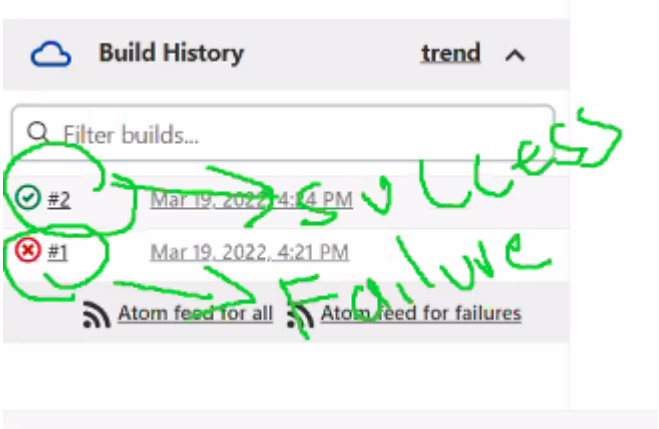
<< Now our project is ready >>

→ We go back to the IJ and run the class again - names changed again :)

→ Let's see the error message on the cloud →
We're on the project page → click on "Build Now" → starts running our scripts → click on Console Out and you'll see the tests → it fails due to the data change.

→ we rerun the IJ and updated the data with the latest one → then rerun it again → saw the pass message on the IJ →
→ we push our updates → git add. → git commit -m "jenkins updates" → git push

→ then went back to the cloud and rerun it there so we could see the success (Build Now - the second built → Console Out → and we can see the SUCCESS at the bottom of the page 🚀



→ if we click on the "Build Now" this will start a new run, this will be the third one for us.