SEPM Lab- Experiment 4

Aim: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job

Theory:

Continuous Integration: Continuous integration (CI) is a development practice where development teams make small, frequent changes to code. An automated build verifies the code each time developers check their changes into the version control repository.

As a result, development teams can detect problems early. Continuous integration is the first part of CI/CD, a practice that enables application development teams to release incremental code changes to production quickly and regularly.

Advantages:

- **Reduced integration risk:** More often than not, working on projects means multiple people are working on separate tasks or parts of the code. The more people, the riskier the integration. Depending on how bad the problem really is, debugging and solving the issue can be really painful and can potentially mean a lot of changes to the code. Integrating on a daily basis or even more frequently can help reduce these kinds of problems to a minimum.
- **Higher code quality:** Not having to worry about the problems, and focusing more on the functionality of the code results in a higher quality product.
- **The code in version control works** If you commit something that breaks the build, you and your team get the notice immediately and the problem is fixed before anyone else pulls the “broken” code.
- **Reduced friction between team members**: Having an impartial system in place reduces the frequency of quarrels between team members.
- **The quality-of-life improvement for testers:** Having different versions and builds of the code can help isolate and trace bugs efficiently, and it makes life easier for the QA team.
- **Less time deploying Deploying:** projects can be very tedious and time-consuming, and automating that process makes perfect sense.

JENKINS:

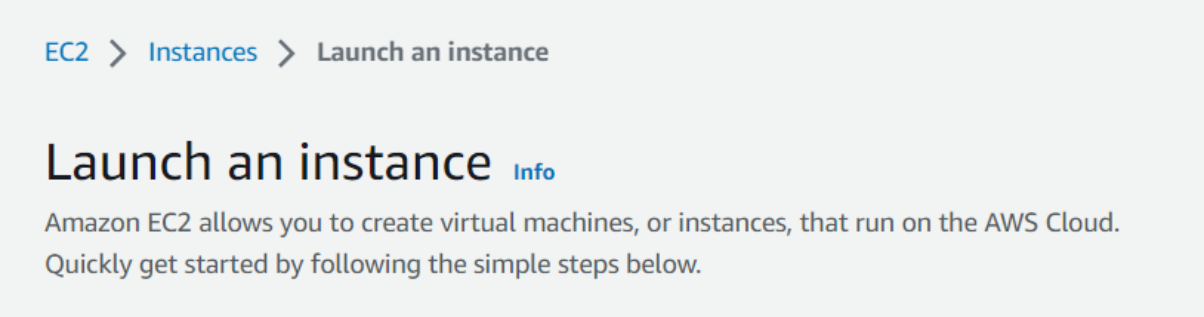
Jenkins is an open-source server that is written entirely in Java. It lets you execute a series of actions to achieve the continuous integration process, that too in an automated fashion.

This CI server runs in servlet containers such as Apache Tomcat. Jenkins facilitates continuous integration and continuous delivery in software projects by automating parts related to build, test, and deployment. This makes it easy for developers to continuously work on the betterment of the product by integrating changes to the project.

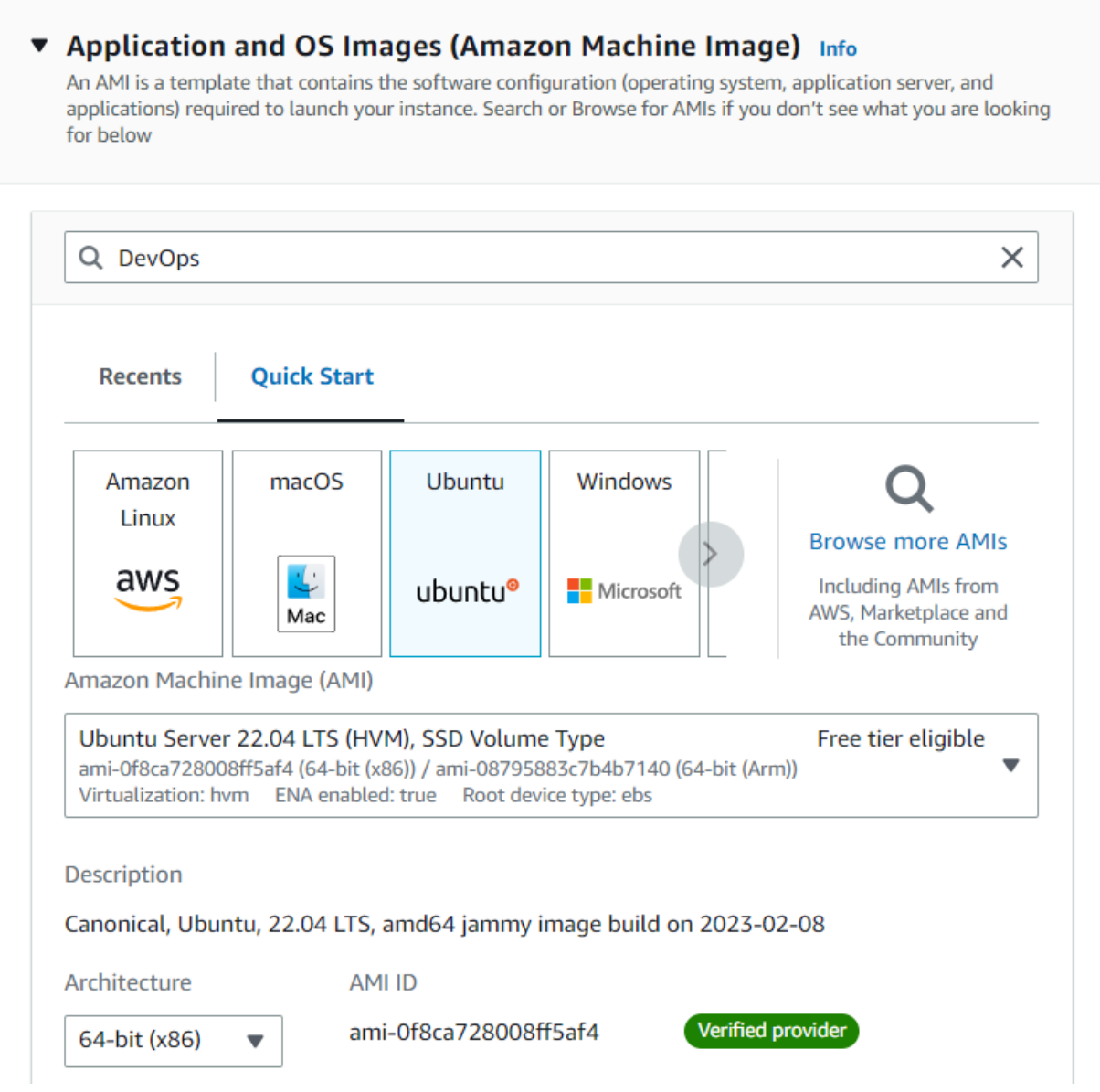
Jenkins automates the software builds in a continuous manner and lets the developers know about the errors at an early stage. A strong Jenkins community is one of the prime reasons for its popularity. Jenkins is not only extensible but also has a thriving plugin ecosystem.

INSTALLATION:

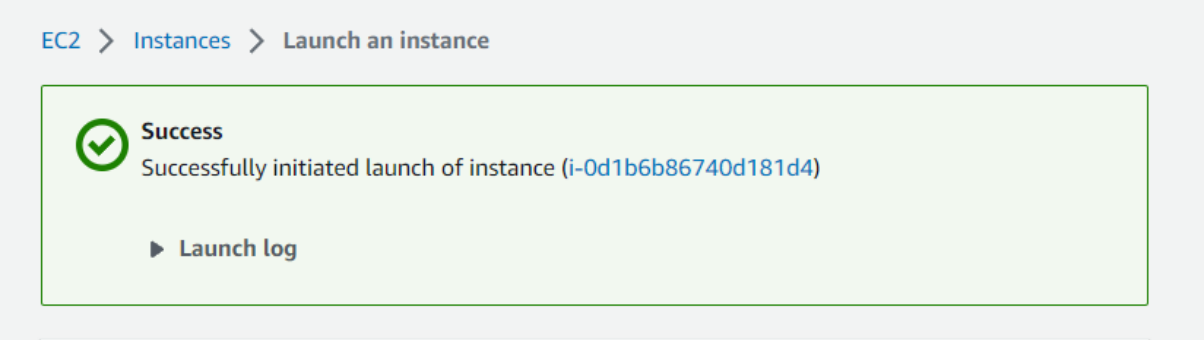
1. Login to the **AWS Console**, and launch an instance of **EC2**



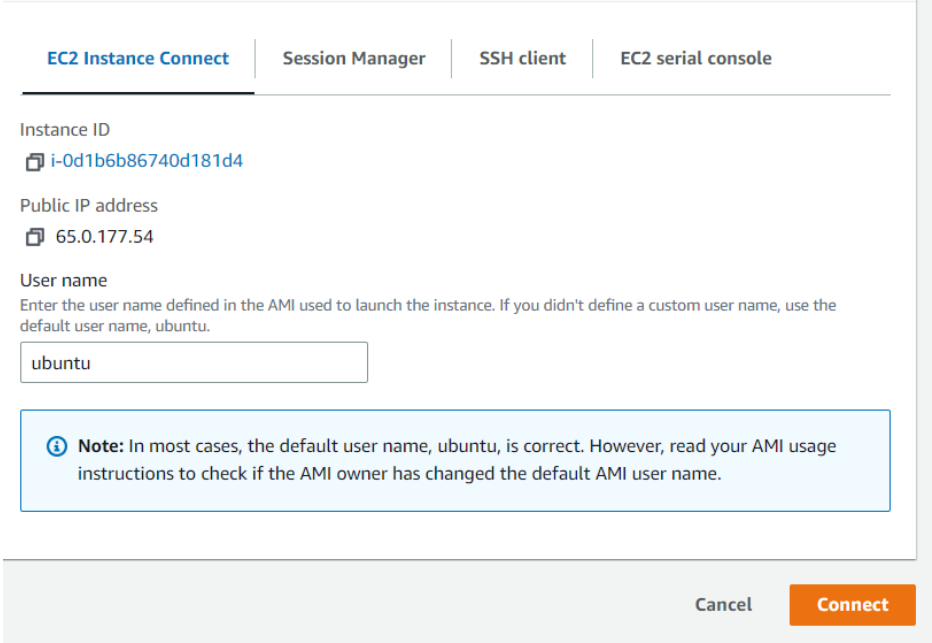
2. Select **Ubuntu** from the options, make sure you select the **free tier option**.
    a. You can also **create a key** and **configure storage** accordingly



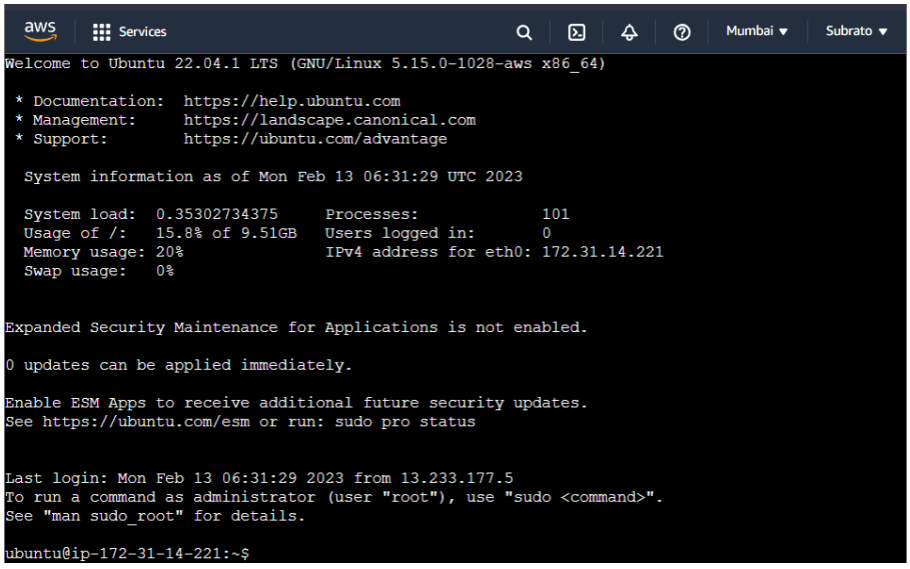
3. After all configurations, click on **Launch Instance** to start it.



4. Click on **Connect** to start the virtual cloud server.



5. The console is now launched, clear the console using “clear” command



6. Run these commands on the console one by one

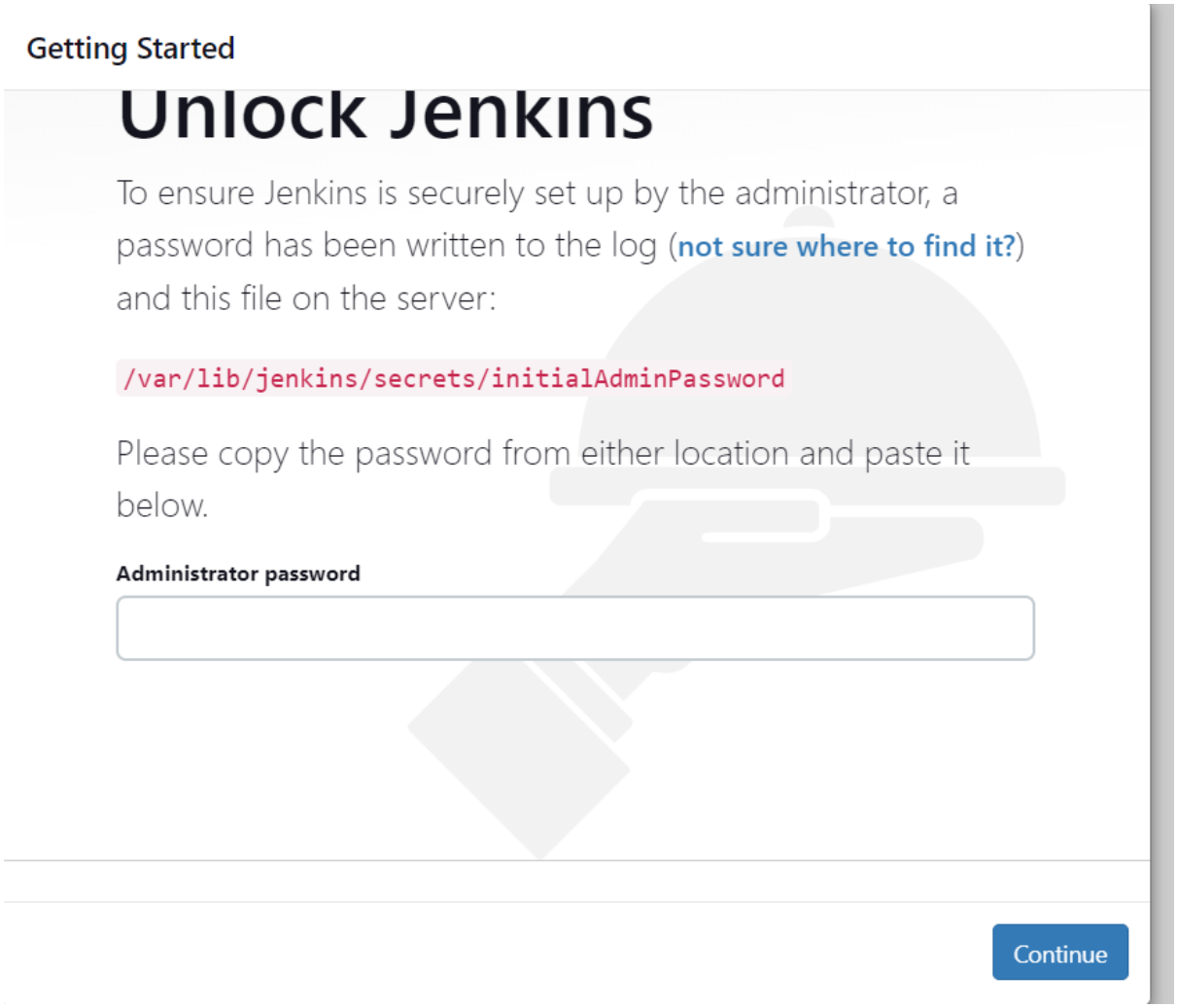
```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
```

7. Install Java Runtime Environment using the command

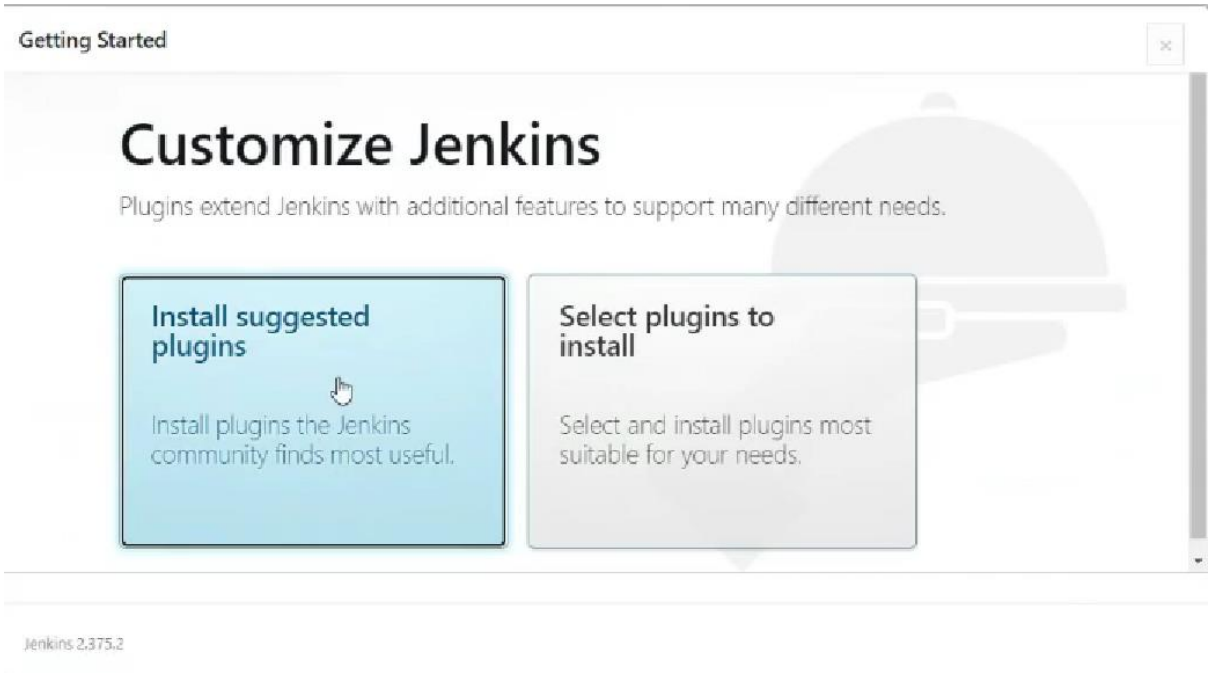
   sudo apt install openjdk-11-jre

8. Type **systemctl status jenkins** to start the Jenkins on the http address that your Instance has, if there's an error, try opening the port and set it to 8080

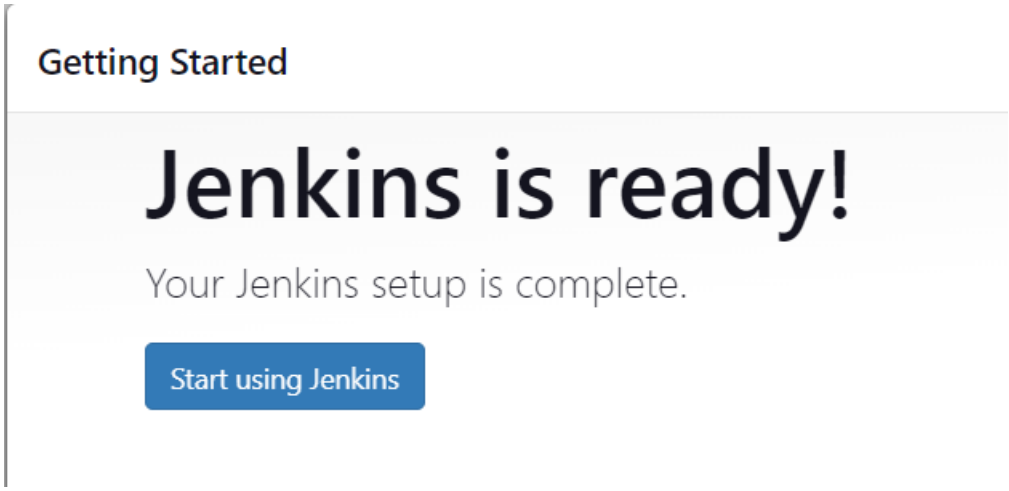
9. Type sudo cat (location given below) to know the password and paste it here and click on continue



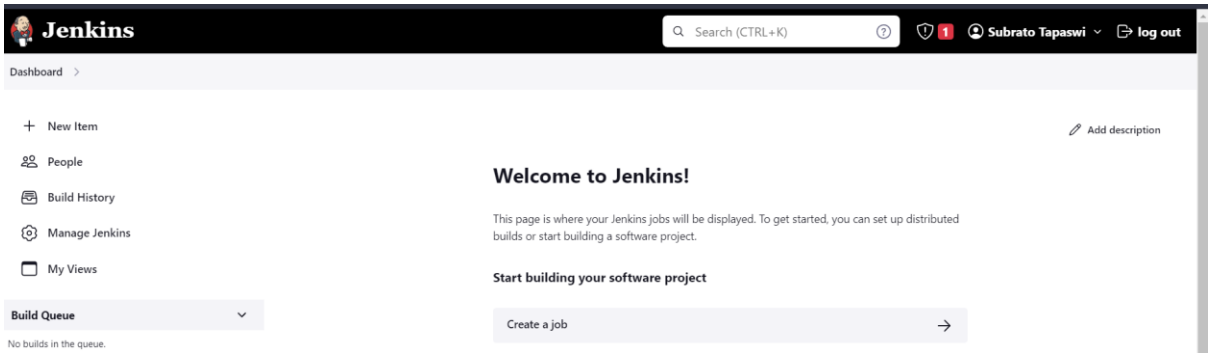
10. You will be redirected to the next page, click on Install suggested plugins and wait till all the installations finish



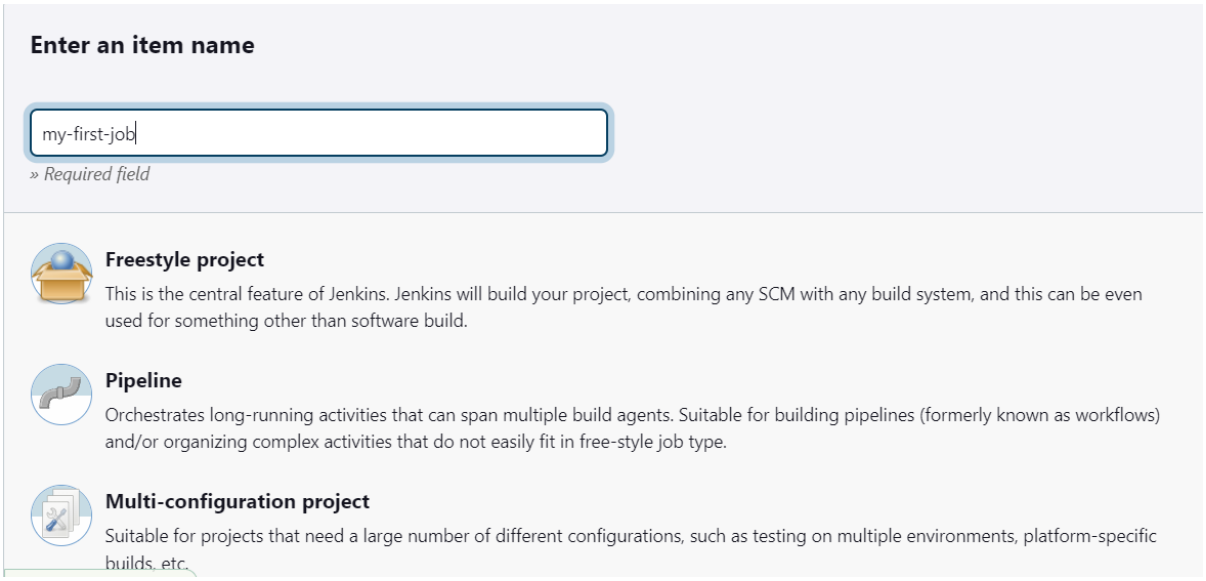
11. Jenkins is then ready to be used.



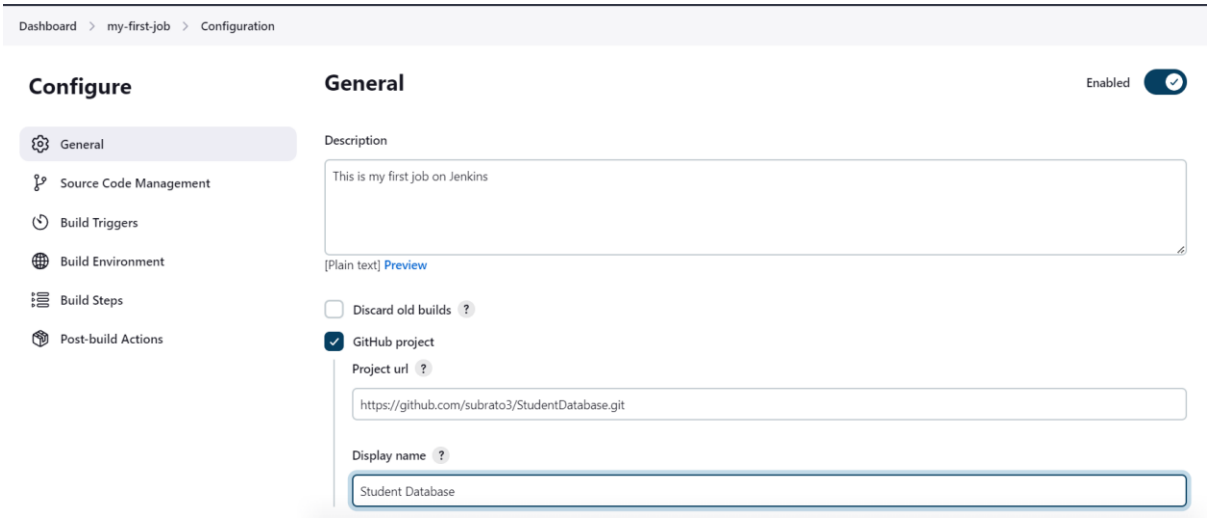
12. You will be redirected to the Dashboard upon clicking “Start using Jenkins”



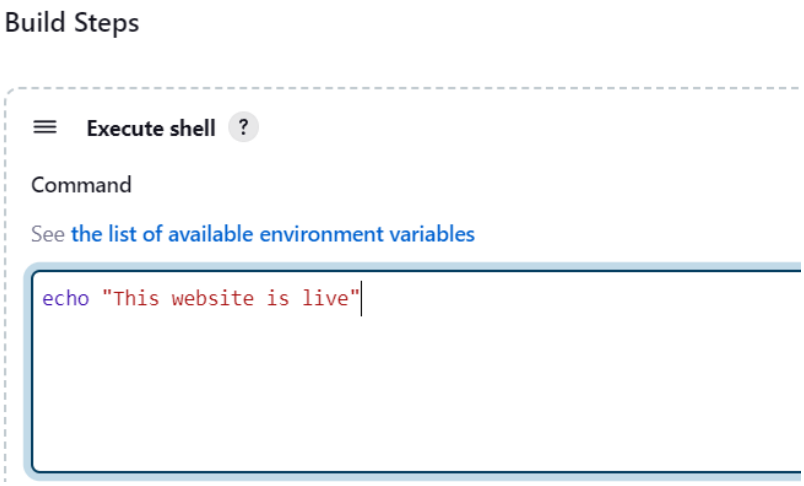
13. Click on create a job, enter the item name and select “freestyle project”



14. Write a description under the general section. Optionally, add a Git Repository if needed

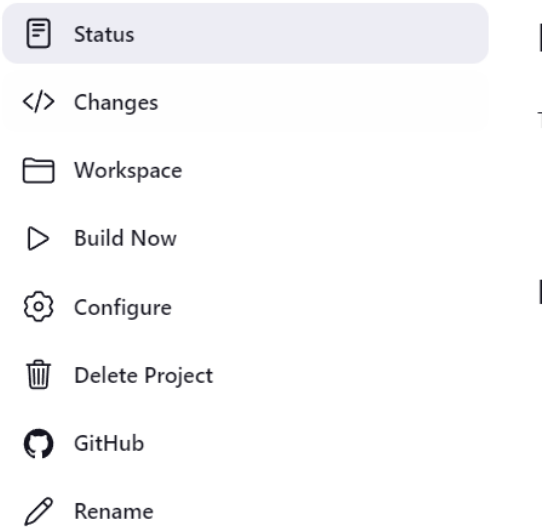


15. Leave Build Triggers as is and write a shell command to be executed on the build.

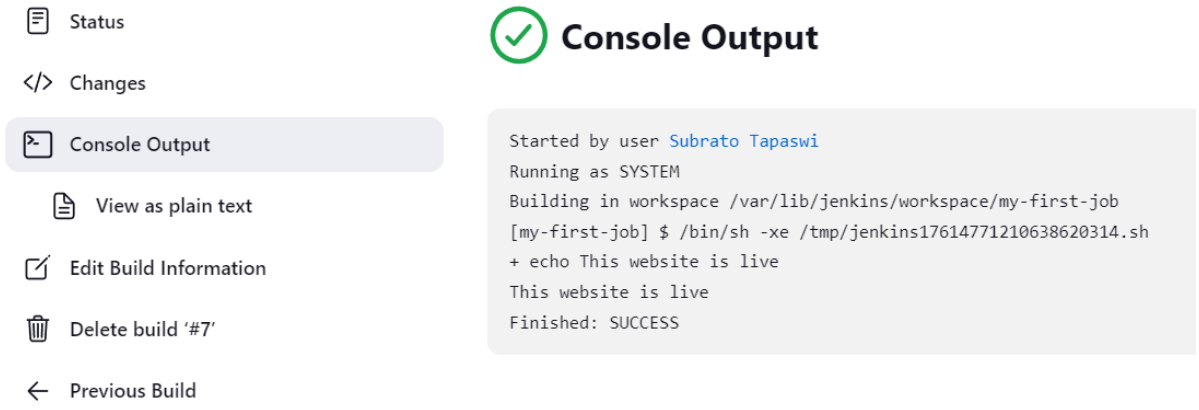


16. Your project is ready.

17. Click on **Build Now** to start building, thereafter click on any one of your builds to see the output.



18. Click on **Console Output** to see it.



Conclusion: We have studied about Continuous Integration and its benefits. We also studied and installed Jenkins, along with its prerequisites on Ubuntu of Amazon AWS EC2. Lastly, we created our first freestyle build job on Jenkins.