**SEPM**

**Experiment 4**

**Aim:**

To understand Continuous Integration, install and configure Jenkins with Maven / Ant / Gradle to set up 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 painful and 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 in an automated fashion.

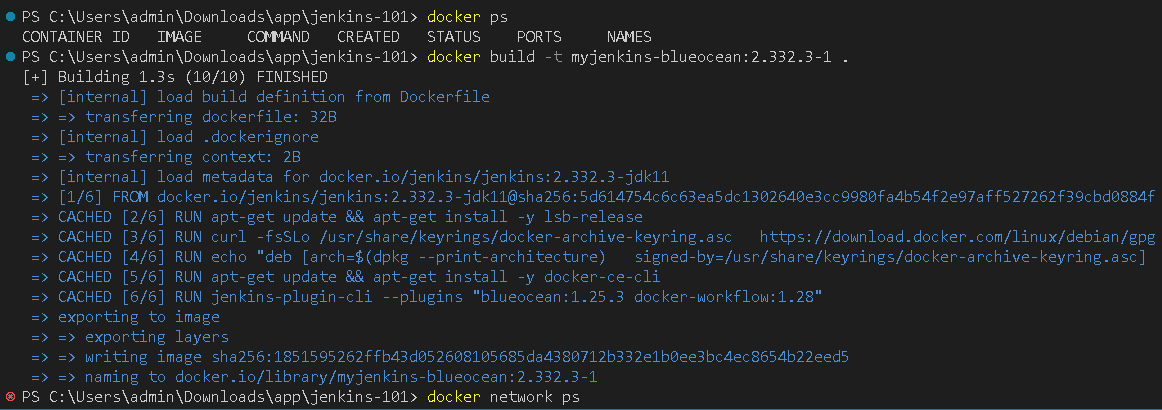
This CI server runs in servlet containers such as Apache Tomcat. Jenkins facilitates continuous integration and continuous delivery in software projects by automating parts related to building, testing, and deployment. This makes it easy for developers to continuously work on the betterment of the product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

Build the Jenkins BlueOcean Docker Image

*docker build -t myjenkins-blueocean:2.332.3-1 .*



Create the network 'jenkins’

*docker network create jenkins*

Run the Container

*docker run --name jenkins-blueocean --restart=on-failure --detach `*

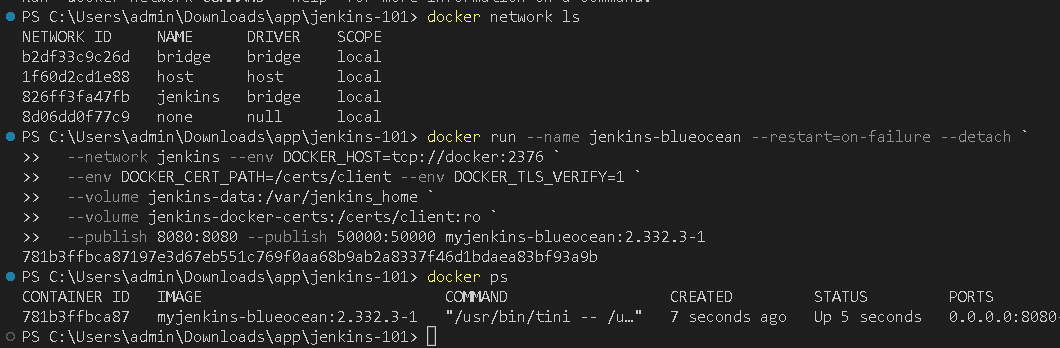
*--network jenkins --env DOCKER\_HOST=tcp://docker:2376 `*

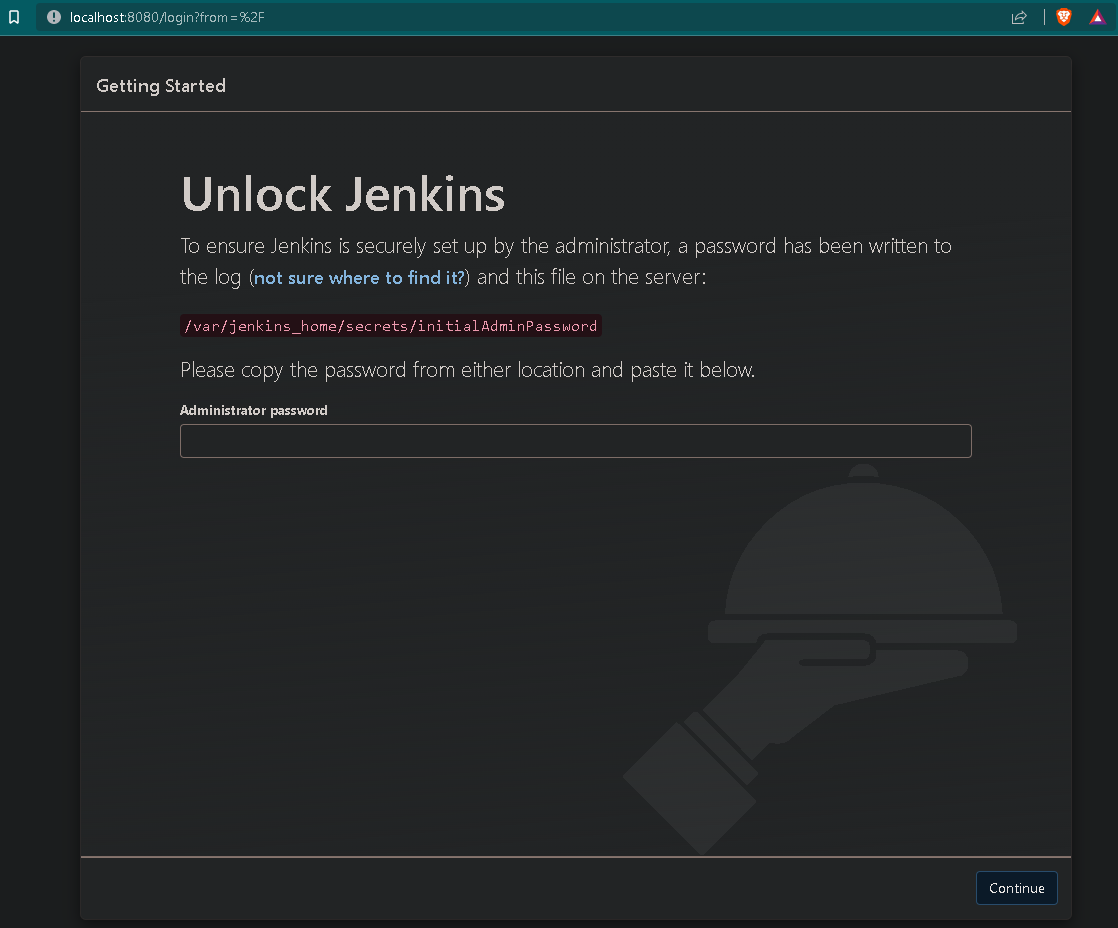
*--env DOCKER\_CERT\_PATH=/certs/client --env DOCKER\_TLS\_VERIFY=1 `*

*--volume jenkins-data:/var/jenkins\_home `*

*--volume jenkins-docker-certs:/certs/client:ro `*

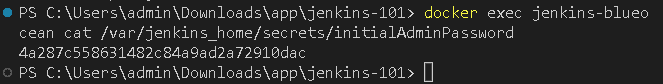
*--publish 8080:8080 --publish 50000:50000 myjenkins-blueocean:2.332.3-1*





Get the Password

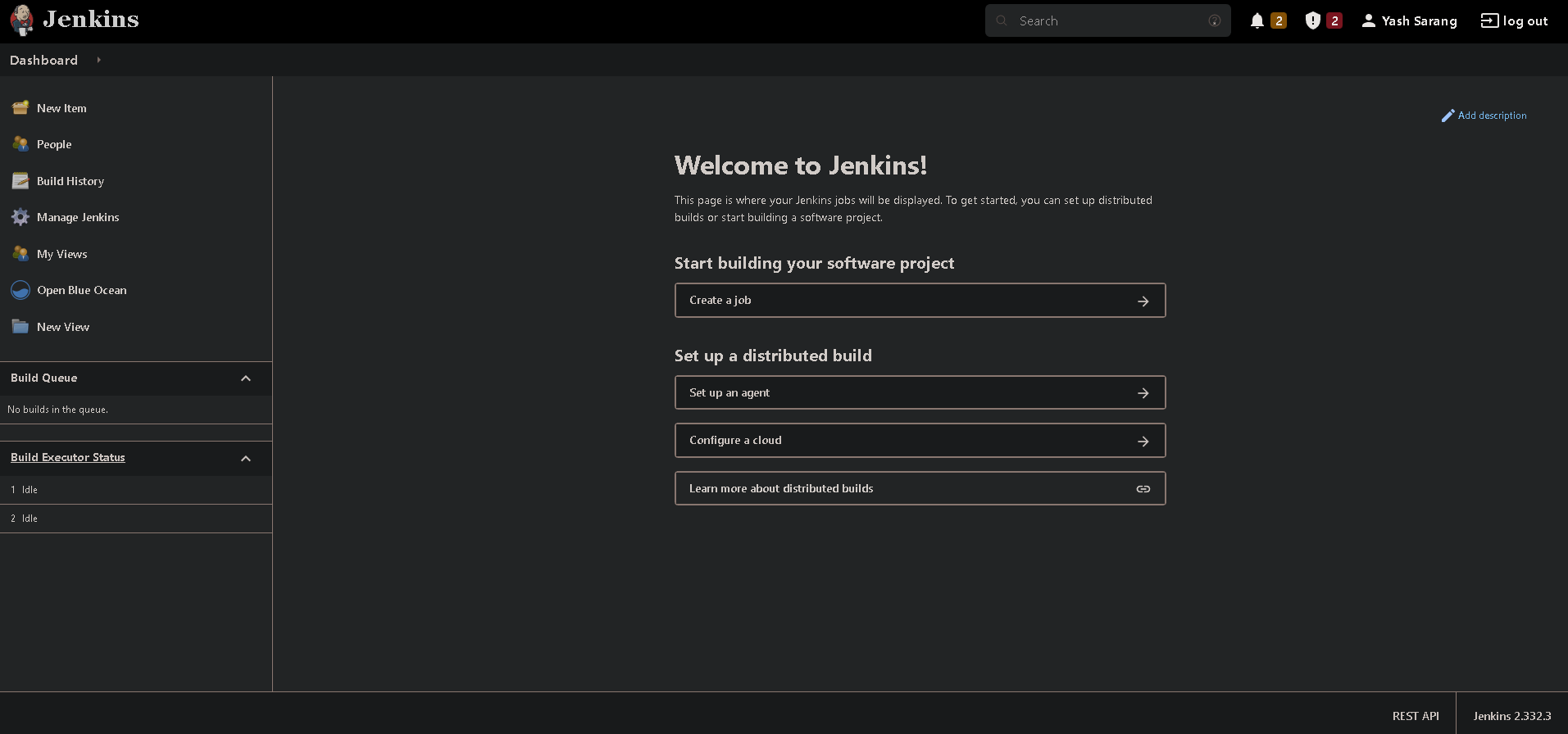
*docker exec jenkins-blueocean cat /var/jenkins\_home/secrets/initialAdminPassword*



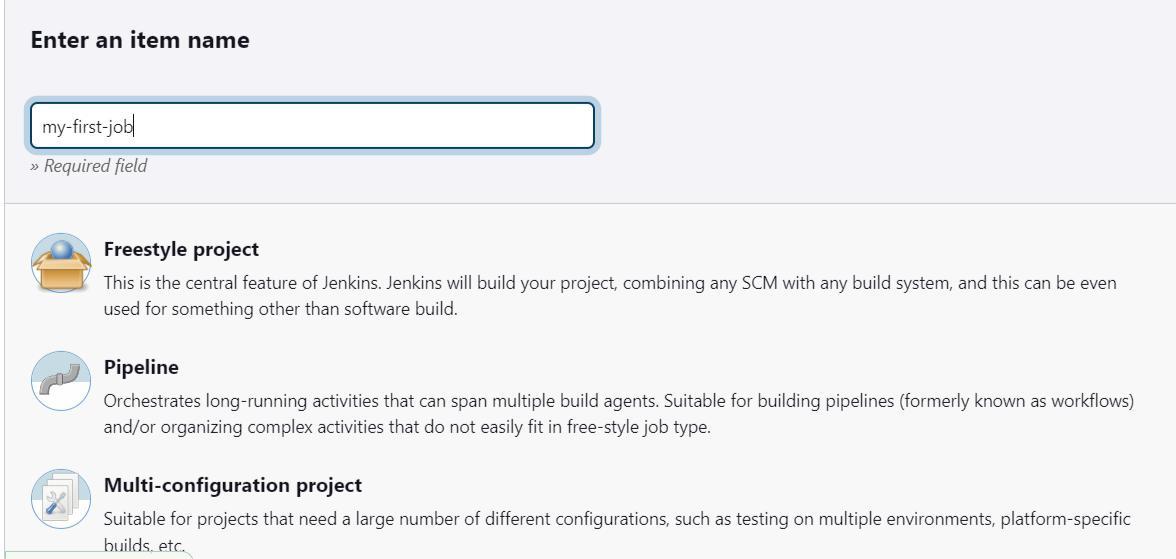
Connect to the Jenkins

*https://localhost:8080/*

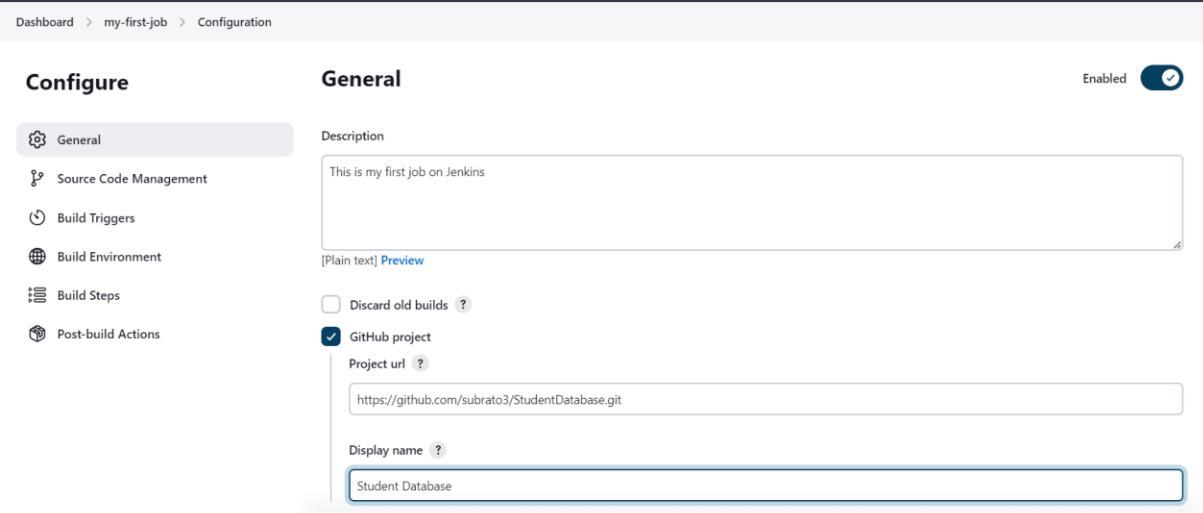


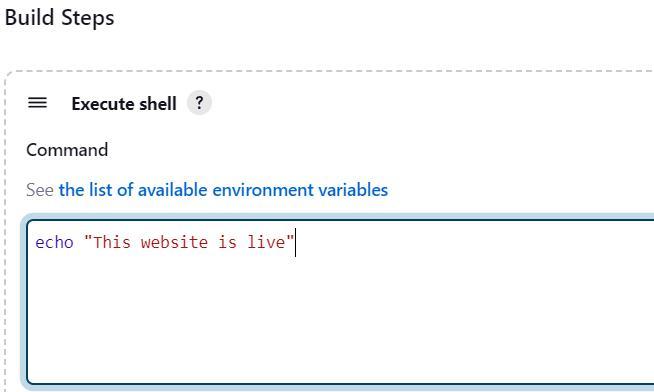


Click on create a job, enter the item name, and select “freestyle project”



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



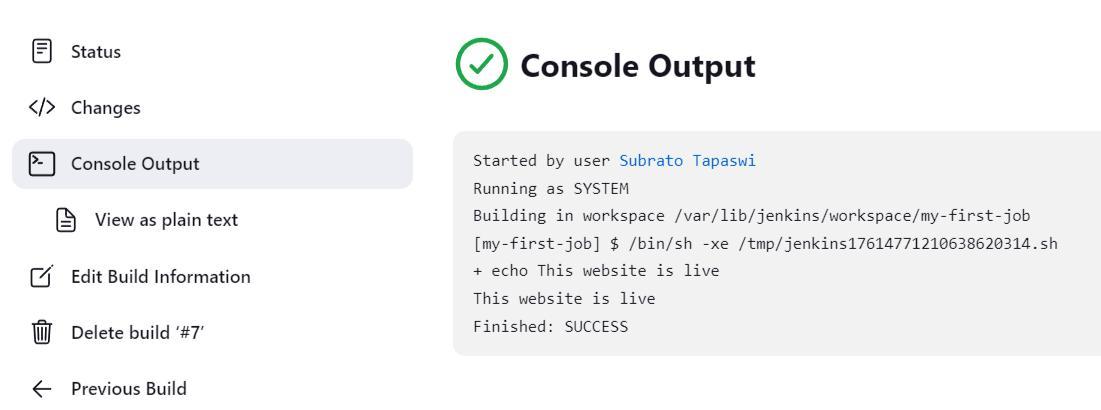


Leave Build Triggers as is and write a shell command to be executed on the build

Your project is ready.



Click on **Build Now** to start building, Click on **Console Output** to see it.



**Conclusion:**

We have studied 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.