This project in Spring Boot revolves around managing UserDetails via CRUD operations. Additionally, we'll construct a complete CI/CD pipeline to automate the deployment process.

Steps:-

1. We Need 3 Aws Instance

Master Server :- Jenkins, Sonar-Server, Nexus Server
Kubernetes Master
Kubernetes Slave

2. We Need Some Devops Tools & Plugins

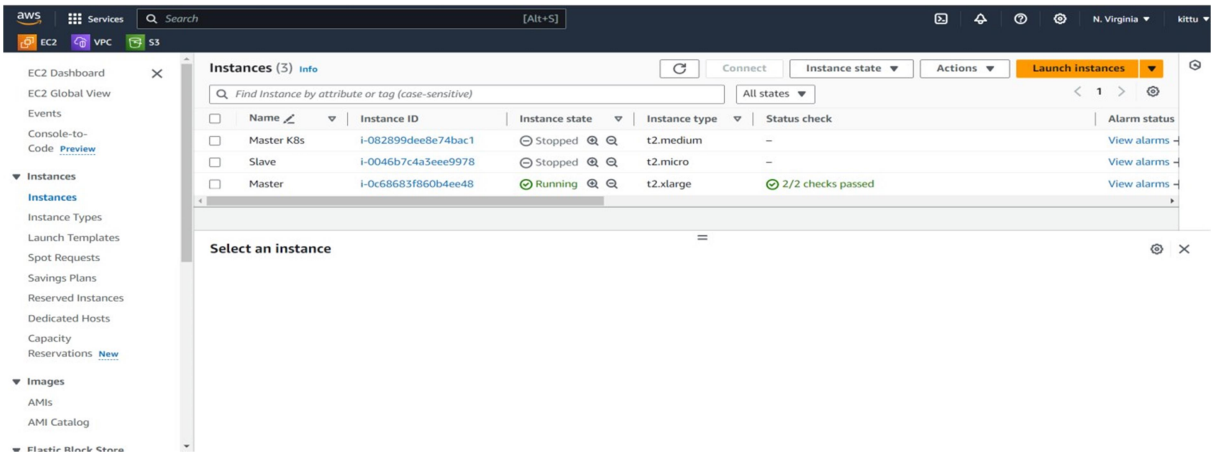
Tools	Plugins
Git	Owsap (for Dependency Check)
GitHub	Sonar-Scanner (for code smell and bugs)
Jenkins	ConfigFileProvider (For configure Nexus)
Maven	Docker & DockerPipeline (for configure docker)
Sonar-Server	Kubernters Credential provider (for congifure Kubernetes)
Nexus server	
Docker	
Kubernetes	

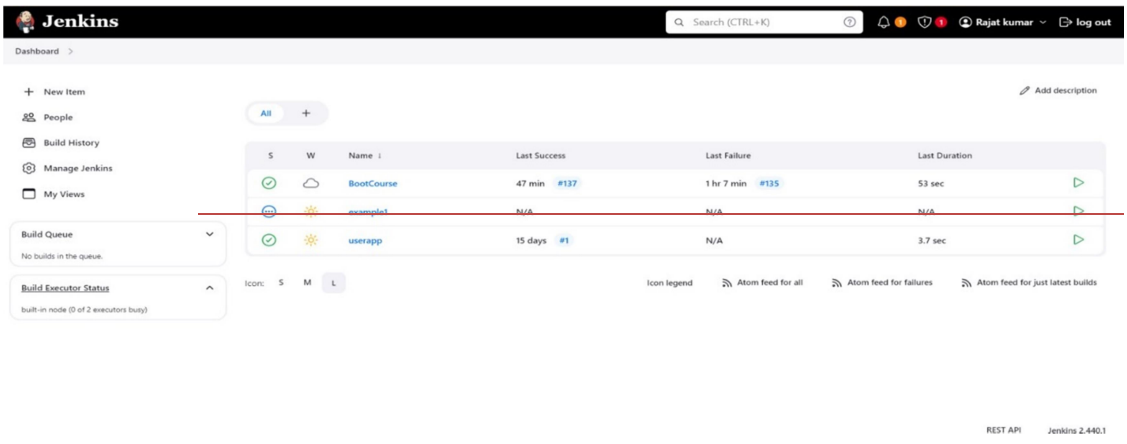
3. FlowChart

Declarative: Tool Install	Git Checkout	Compile	sonar analysis	Package	Dependency check	Maven Deploy	Docker image build	Push Docker image	Docker user network	Docker mysql	Docker deploy

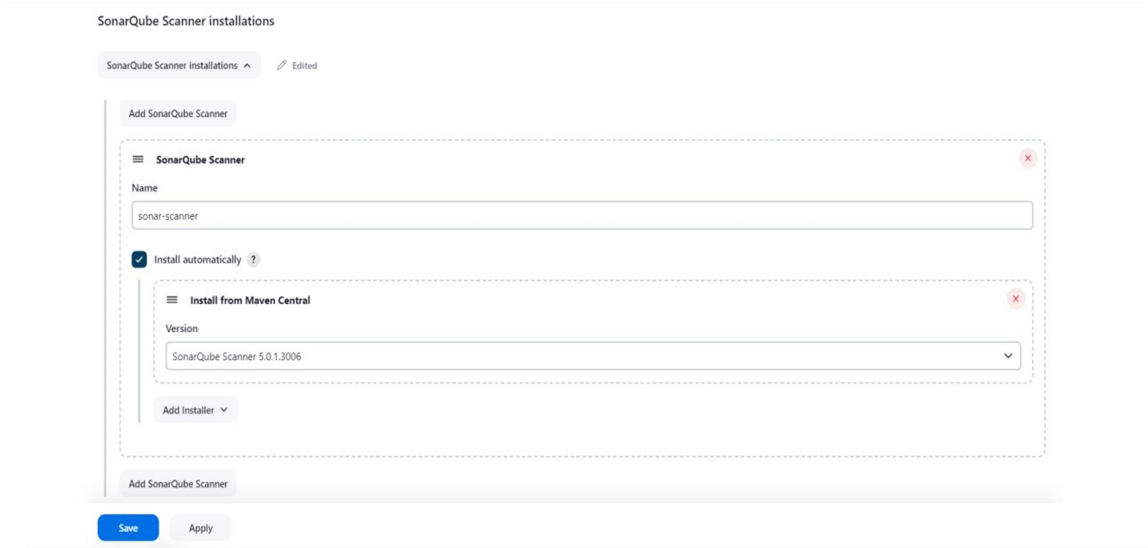
- User write and push the code to GitHub.
- Jenkins fetch the code from Github then compile the code using Maven Tools.
- We will perform Sonar Analysis for Code Smell and checking bugs.
- We will Package the code by using maven Tools.
- We will use Owasp tools for Dependencies check .
- We will Nexus Repo to Keep our Artifact.
- We will Create a Dockerfile and build an from Dockerfile using Docker tool.
- Push Docker image to DockerHub Repo.
- Now, we will a Docker Network and attach to mysql container and myapp contaiener to have communication b/w them.
- After this process we will able to launch our application on internet successfully.

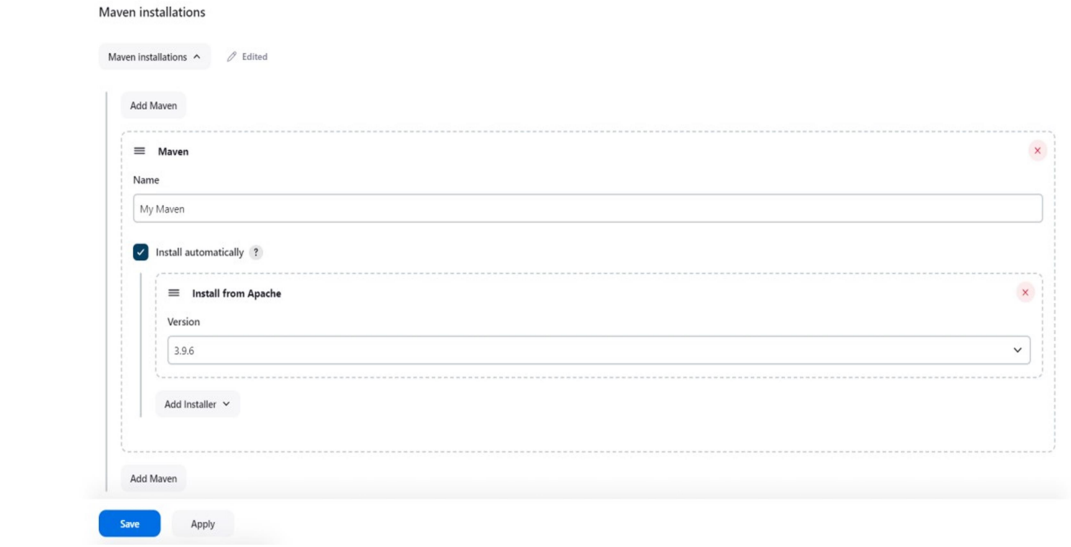
1. Create an Instance
2. Login to Jenkins server and Create a Pipeline Project named as :- BootCourse





3. Install Needed Plugin and Configure tools:- (tools mentioned in flowchart)

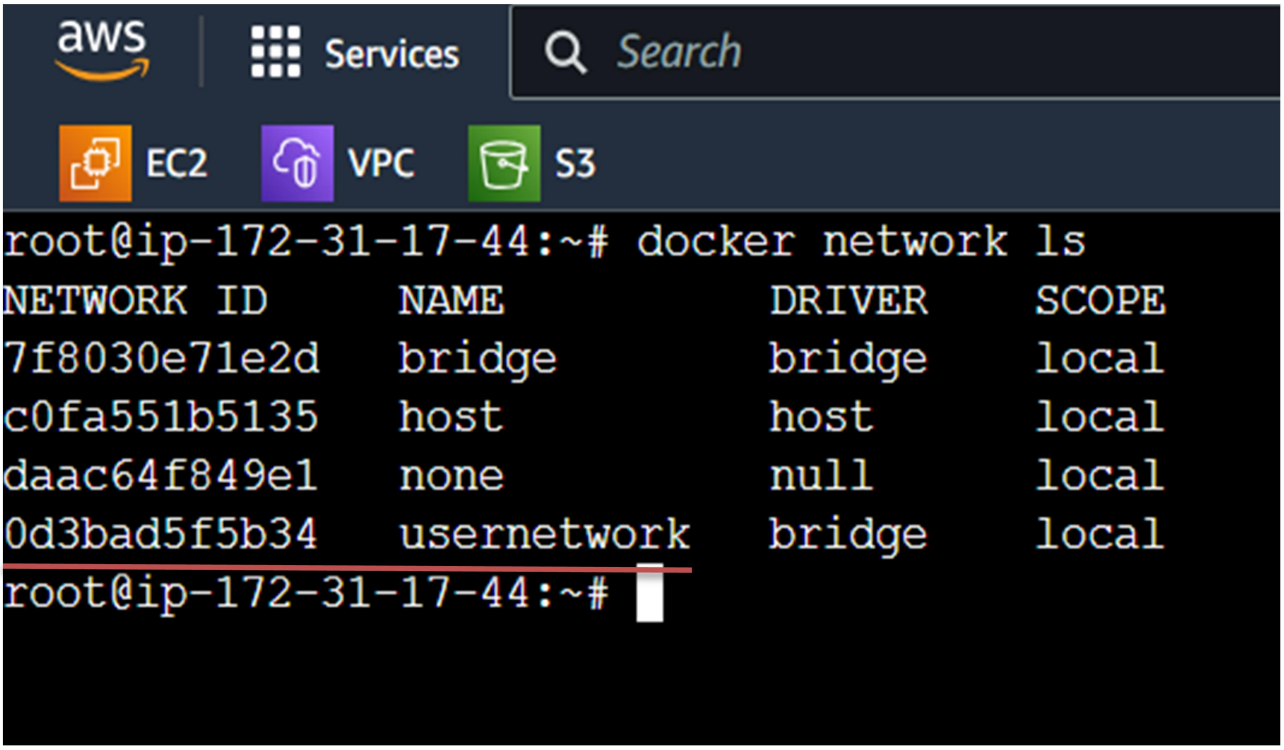




4. Create a Nexus and Sonar server using containerization on Jenkins server

```
034287f2e5d0   sonatype/nexus3          "/opt/sonatype/nexus…"   7 days ago      Up 3 hours      0.0.0.0:8081->8081/tcp, :::8081->8081/tcp
 nexus
2583ee97207e   sonarqube:lts-community   "/opt/sonarqube/dock…"   8 days ago      Up 3 hours      0.0.0.0:9000->9000/tcp, :::9000->9000/tcp
 sonar
root@ip-172-31-17-44:~#
```

5. Create a network

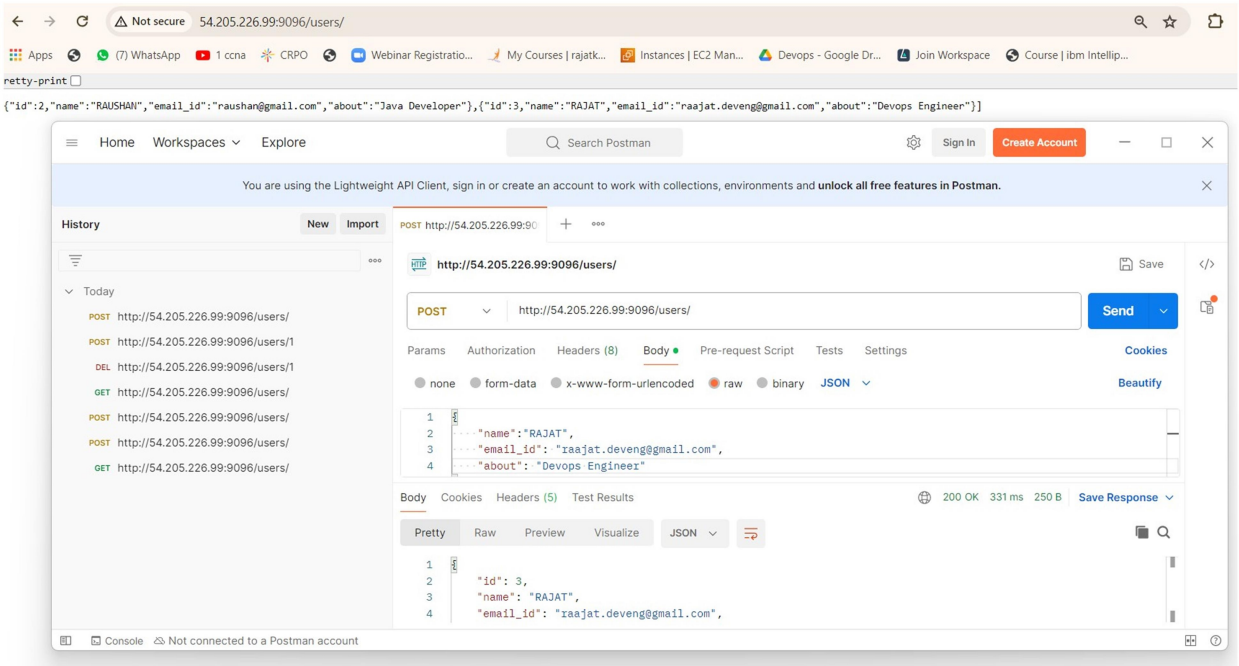


6. Create Mysql container and App container and attached with created network (usernetwork)

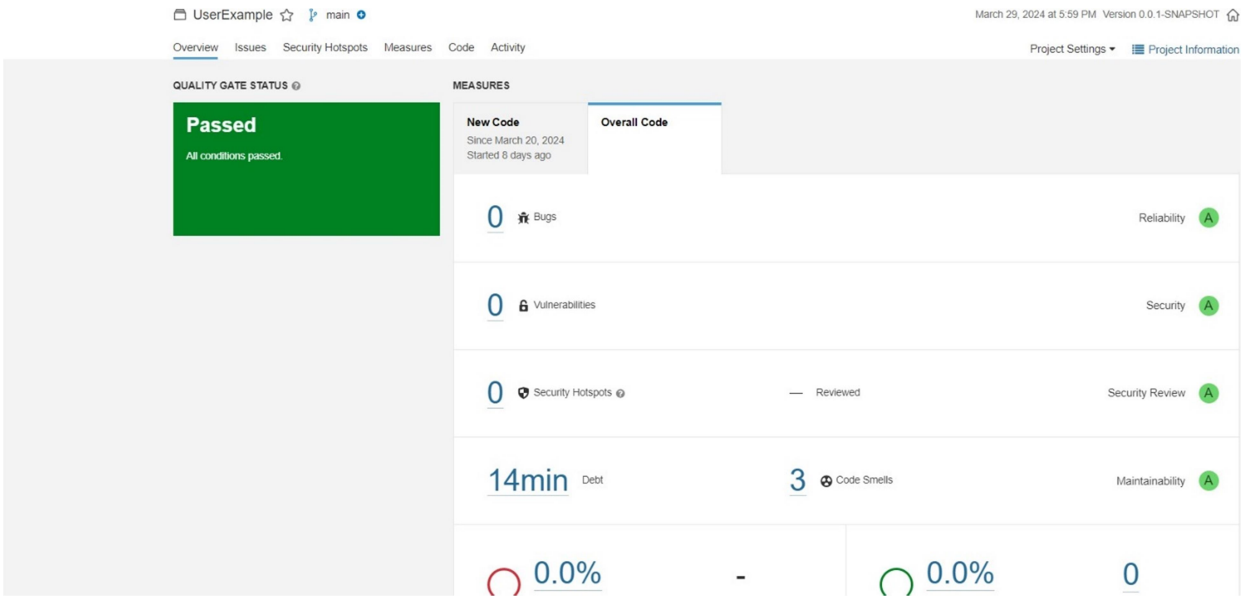
```
root@ip-172-31-17-44:~# docker ps -a
CONTAINER ID   IMAGE                        COMMAND                  CREATED          STATUS          PORTS                                                 NAMES
8b9918a2f5c2   rajatkumar01/users1.0:latest  "java -jar /app/app.…"   About an hour ago  Up 59 minutes   0.0.0.0:9096->9096/tcp, :::9096->9096/tcp             user
9e44b4682cfc   mysql:latest                 "docker-entrypoint.s…"   About an hour ago  Up About an hour  0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp  mysql
034287f2e5d0   sonatype/nexus3              "/opt/sonatype/nexus…"   7 days ago       Up 3 hours      0.0.0.0:8081->8081/tcp, :::8081->8081/tcp             nexus
2583ee97207e   sonarqube:lts-community      "/opt/sonarqube/dock…"   8 days ago       Up 3 hours      0.0.0.0:9000->9000/tcp, :::9000->9000/tcp             sonar
root@ip-172-31-17-44:~#
```

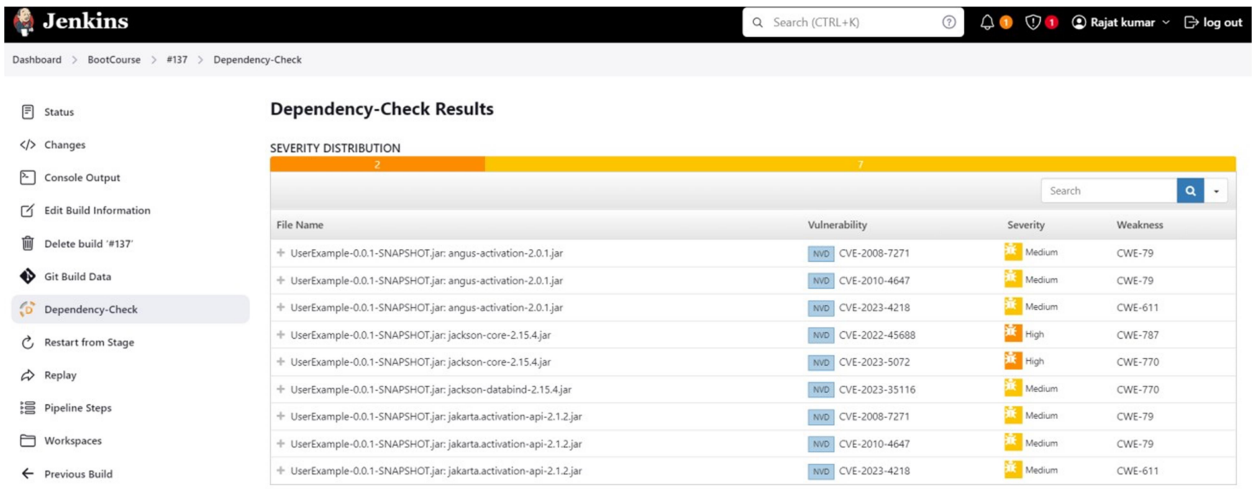
7. Both Container is Up and running we can use Jenkins Public ip and port 9096 to launch app.

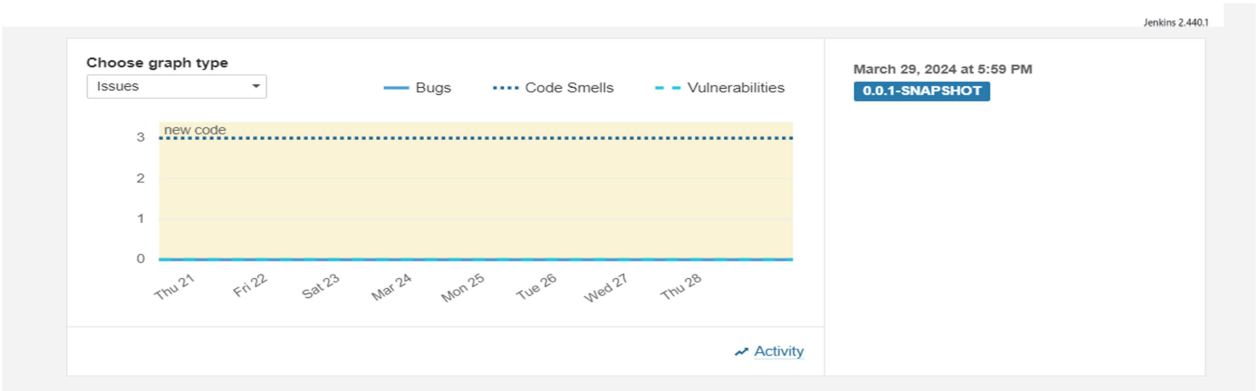
8. As our project is having only backend and database so we a tools Postman to send data to app.



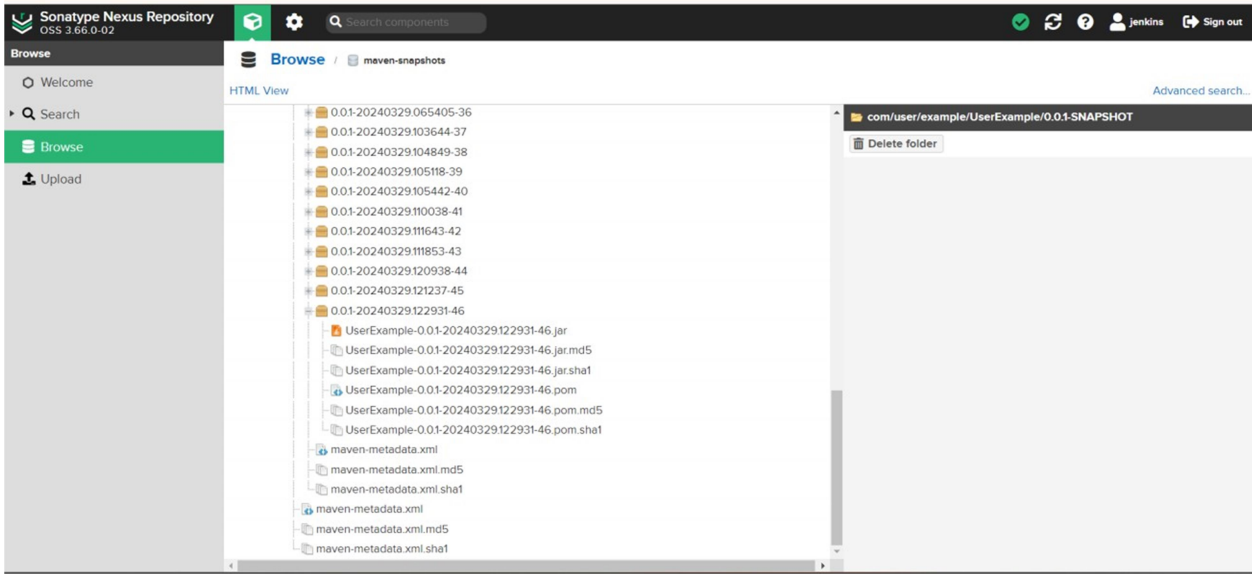
9. To check Dependency check result we have to login sonar server.



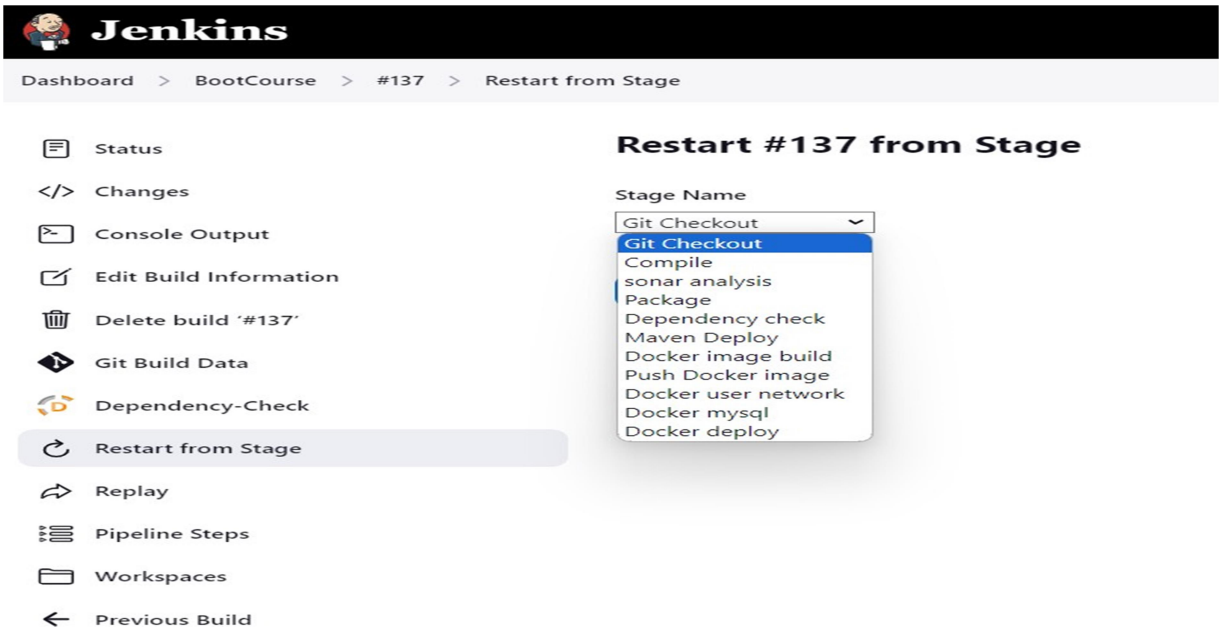




10. To check artifact on Nexus We will have to login on Nexus Server.



Total Stage in this Pipeline :-



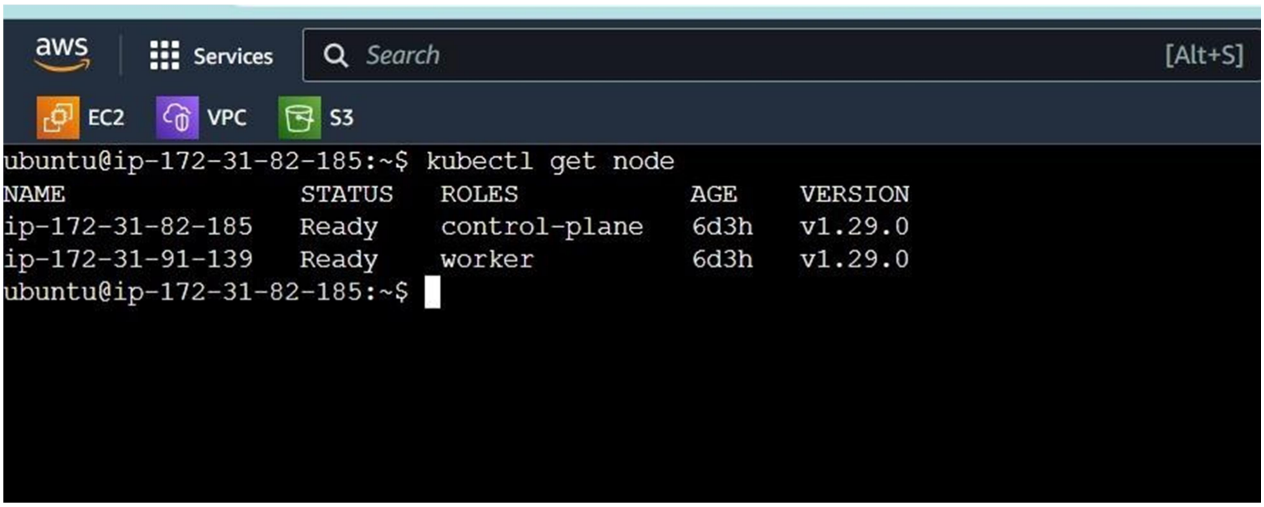
Deployment From Docker is Done.

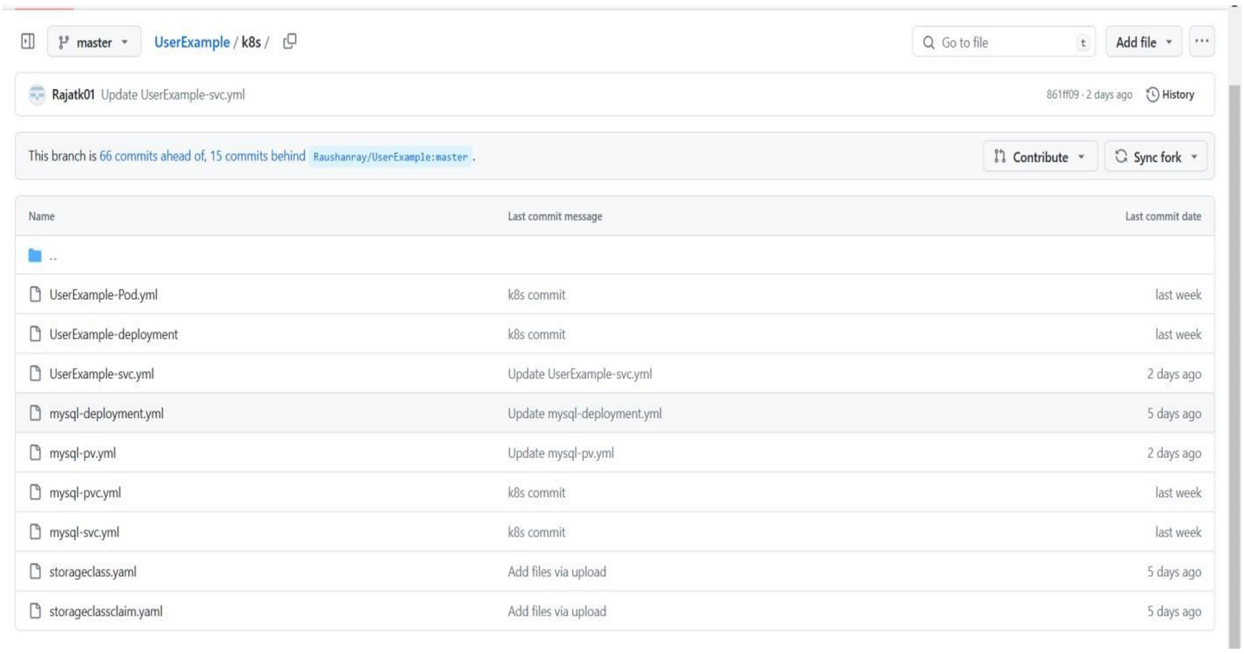
Now we will deploy on k8s using Host Machine

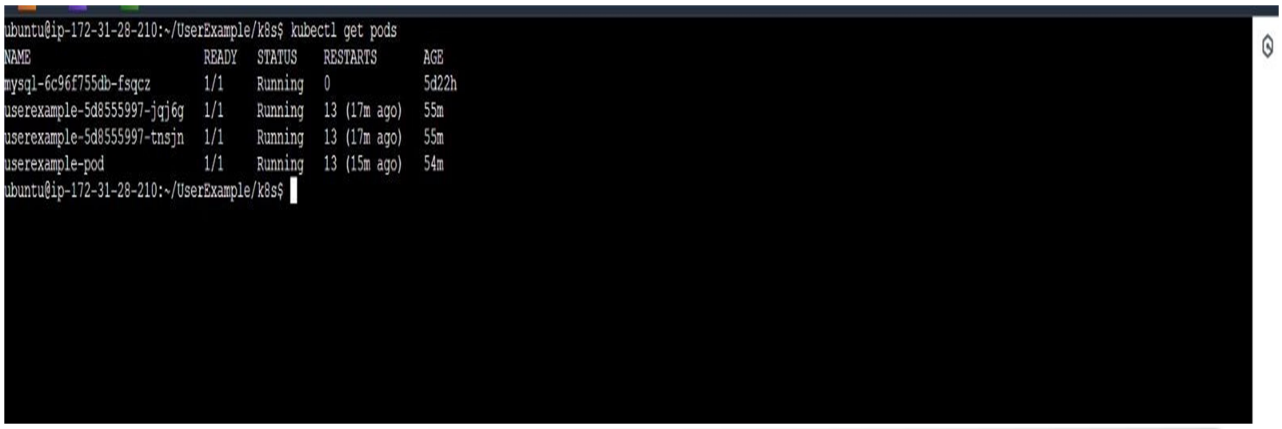
- For k8s deployment we need k8s master and node Connected with kubeadm setup:
- We will use same docker image which is used in docker deployment.
- We will use yaml file for k8s deployment.

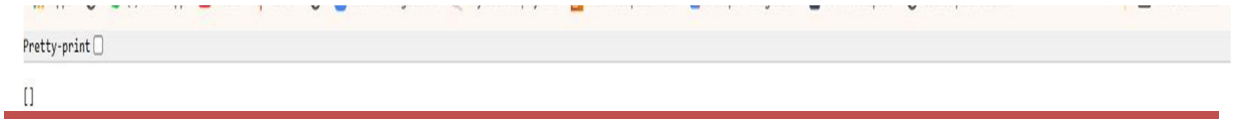
Steps:-

- We will verify Our k8s master and nodes is Ready or not.
- We will apply our yaml file on master.
- We will verify the status of pod weather it is running or not. If our backend and Database pod is running then our deployment is successfully deployed.
- Take Slave Ip and hit the url our project is up and running.









Deployment Done And Working.