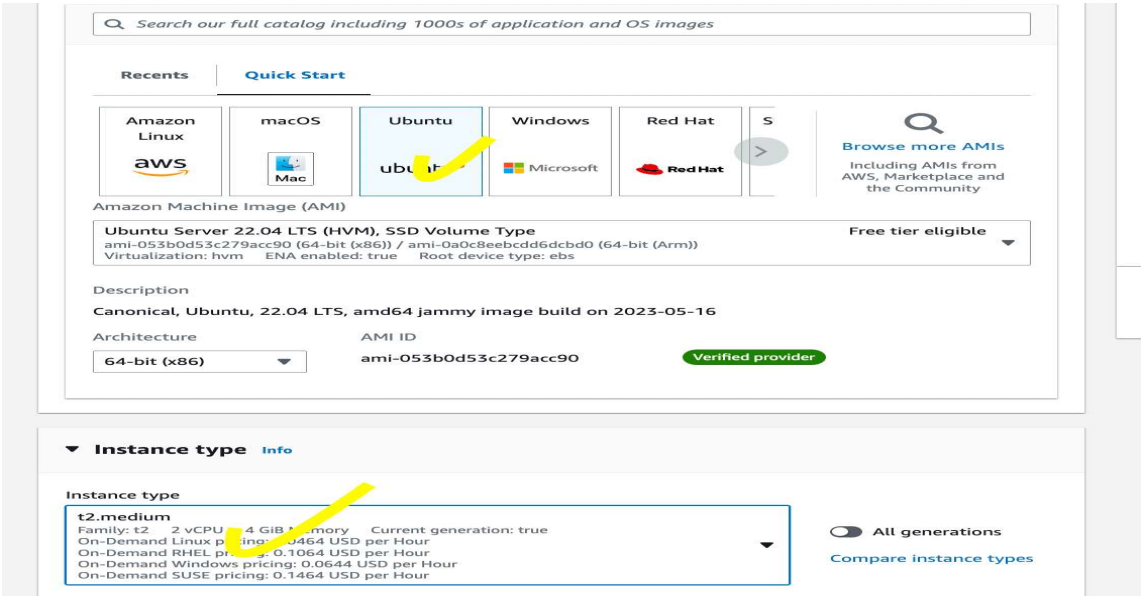
Declarative: Checkout SCM	Git Checkout	Unit Test maven	Integration Test maven	Static code analysis: Sonarqube	Quality Gate Status Check : Sonarqube	Maven Build : maven	Docker Image Build	Docker Image Scan: trivy	Docker Image Push : DockerHub	Docker Image Cleanup : DockerHub
341ms	321ms	12s	10s	30s	449ms	9s	3s	8s	6s	462ms
341ms	321ms	12s	10s	30s	449ms (paused for 8s)	9s	3s	8s	6s	462ms

CI PIPELINE SETUP WITH VARIOUS TOOLS

****You need to complete this in one go ****
100 rupees max it will cost
[8 GB –> 30 GB] and t2.micro -> t2.medium [More memory and more cpu]

Step 1 – Create the Ec2 instance in AWS account with these parameters

EC2 type – Ubuntu t2.medium
EBS volume – 30 GB
Region – US-EAST-1



Step 2 – Connect to EC2 and Install all tools in that system as root user

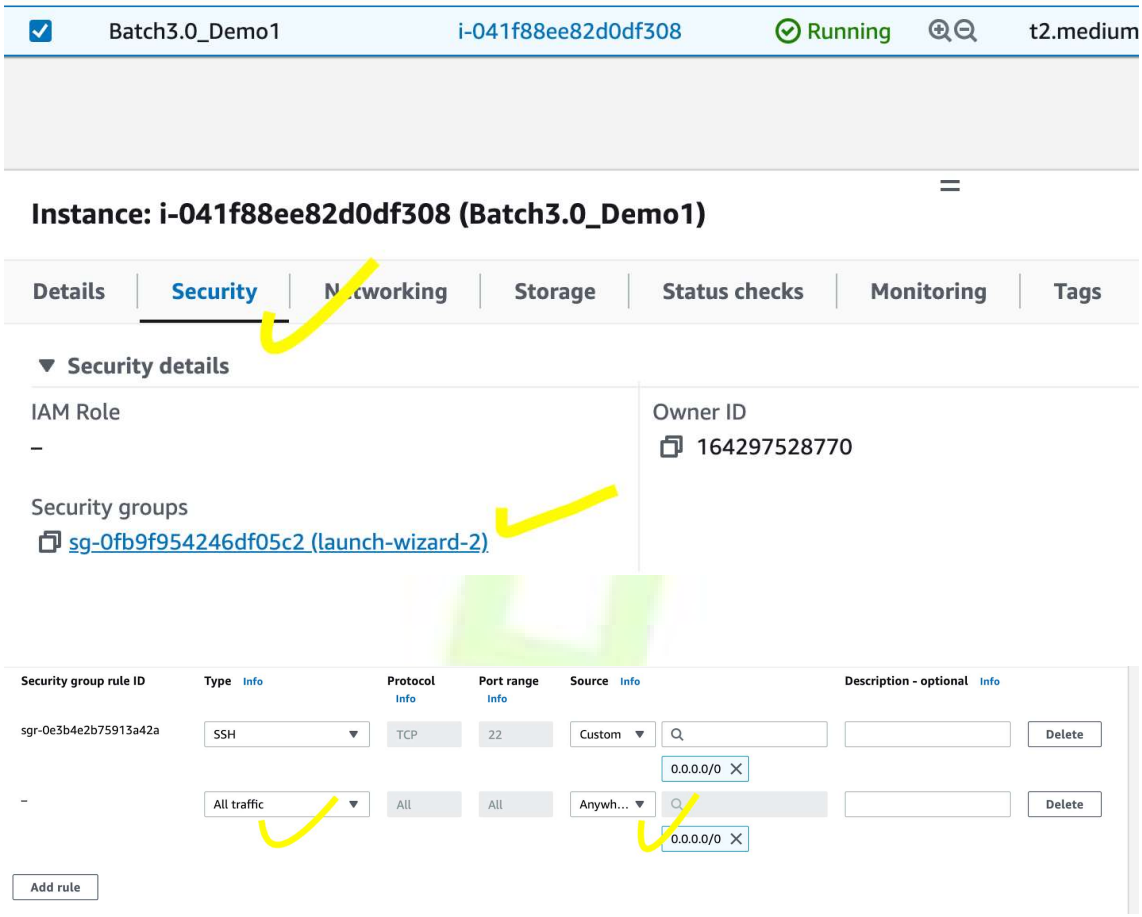
To login as root user - sudo su

Step 3 – Install Jenkins on Ubuntu

Just copy paste the entire commands

https://github.com/praveen1994dec/tools_installation_scripts/blob/main/jenkins.sh

Step 4 – Change the security group of ec2 instance

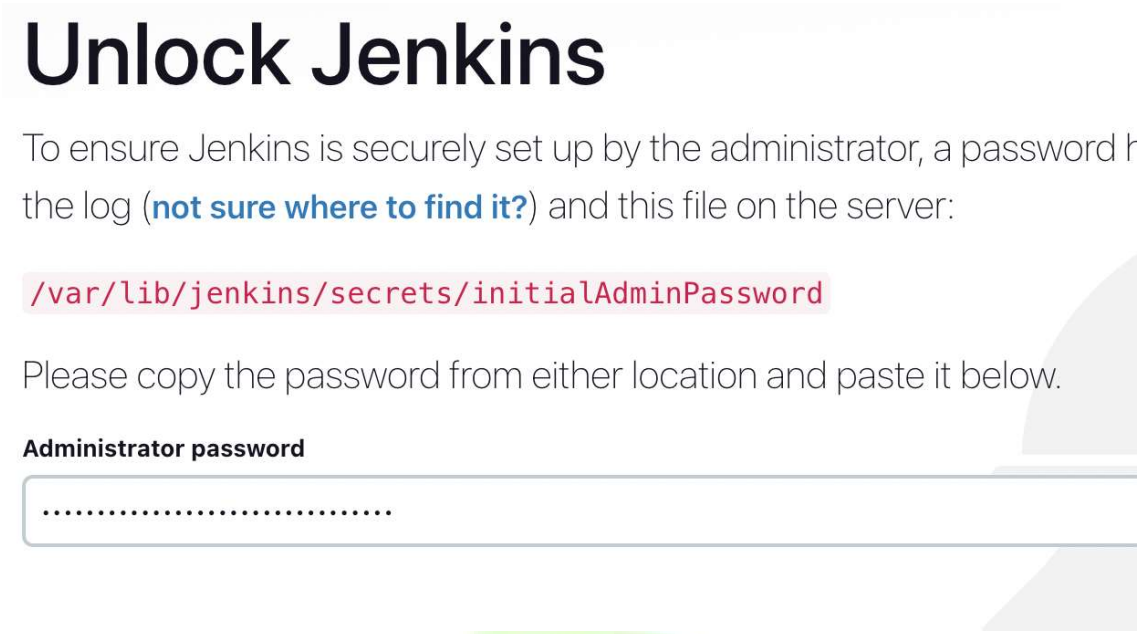


Step 5 – Sign Into Jenkins console

http://<EC2_PUBLIC _IP>:8080/

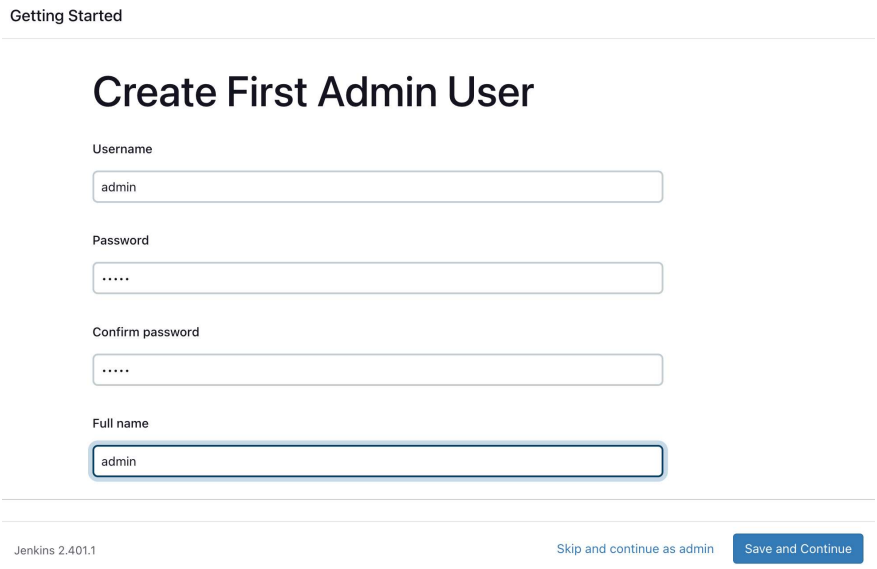
Step 6 – Get the Administrator password by hitting the below command in EC2

```
cat /var/lib/jenkins/secrets/initialAdminPassword
```

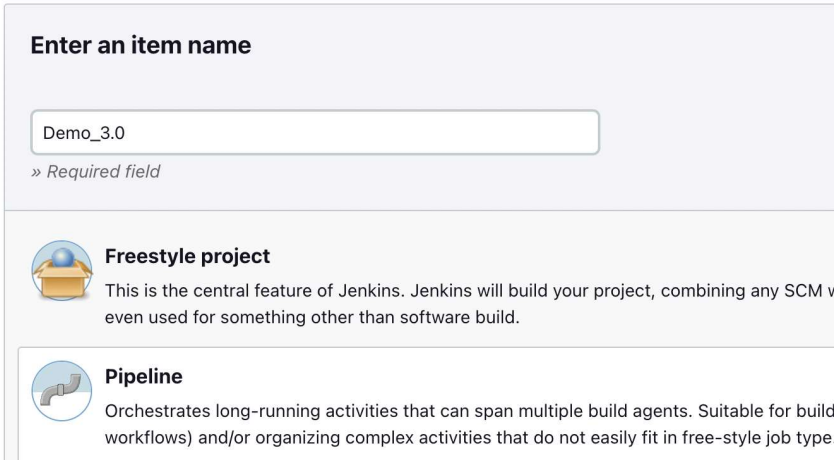


Step 7 – Install all suggested plugins

Step 8 – Create first user

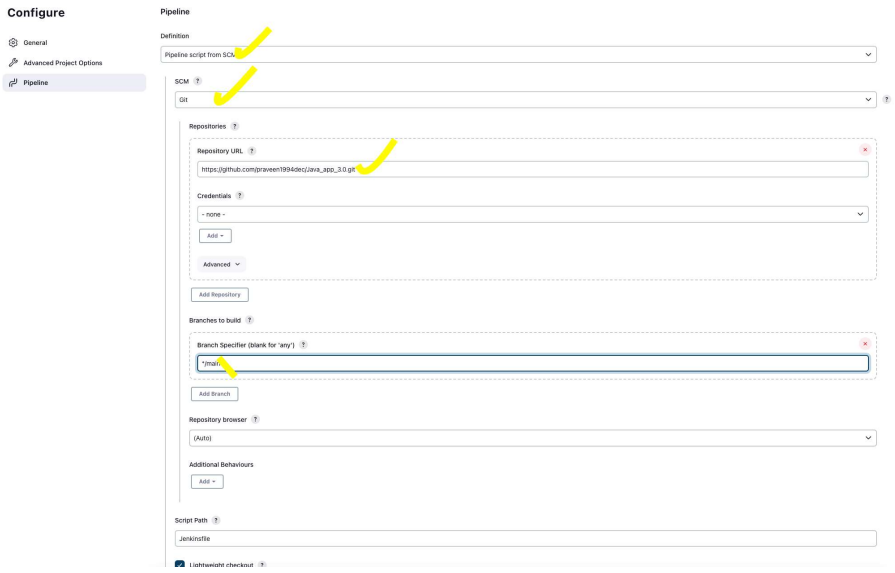


Step 9** – Create a pipeline Job



Step 10 – Add pipeline script as SCM

https://github.com/praveen1994dec/Java_app_3.0.git

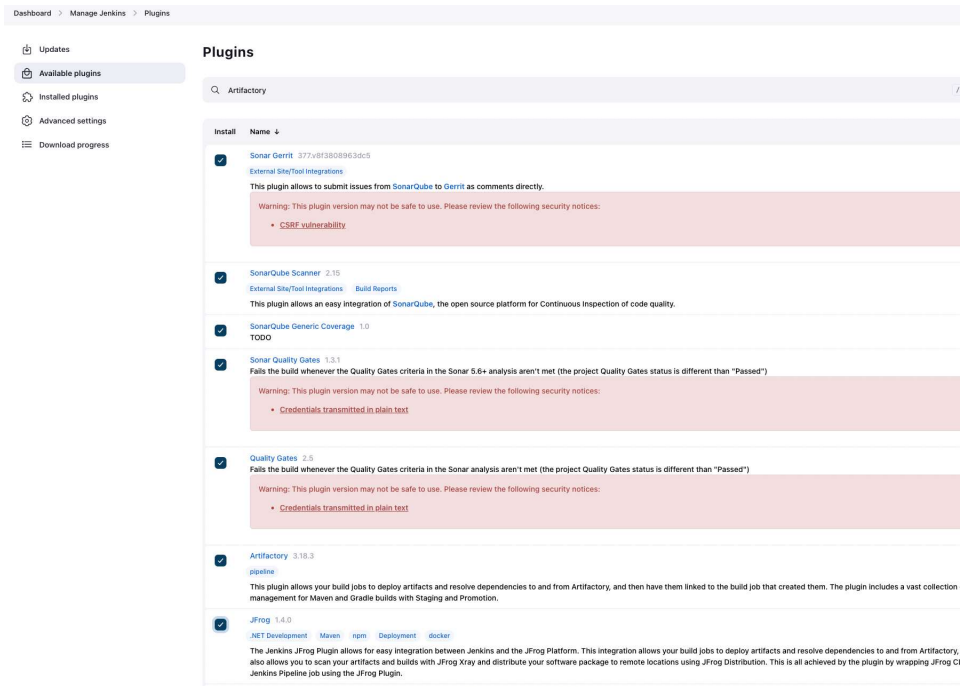


Step 11 – Add the Plugins

Dashboard -> Manage Jenkins -> Plugins -> Available plugins

Plugins for Sonar/Jfrog –

Sonar Gerrit
SonarQube Scanner
SonarQube Generic Coverage
Sonar Quality Gates
Quality Gates
Artifactory
Jfrog



Step 12 – Setup Docker

https://github.com/praveen1994dec/tools_installation_scripts/blob/main/docker.sh

docker -v

Step 13- Install SonarQube

https://github.com/praveen1994dec/tools_installation_scripts/blob/main/sonarqube.sh

Step 13 .1 -> Start docker container if it's not up

docker ps -a [Get the container ID]

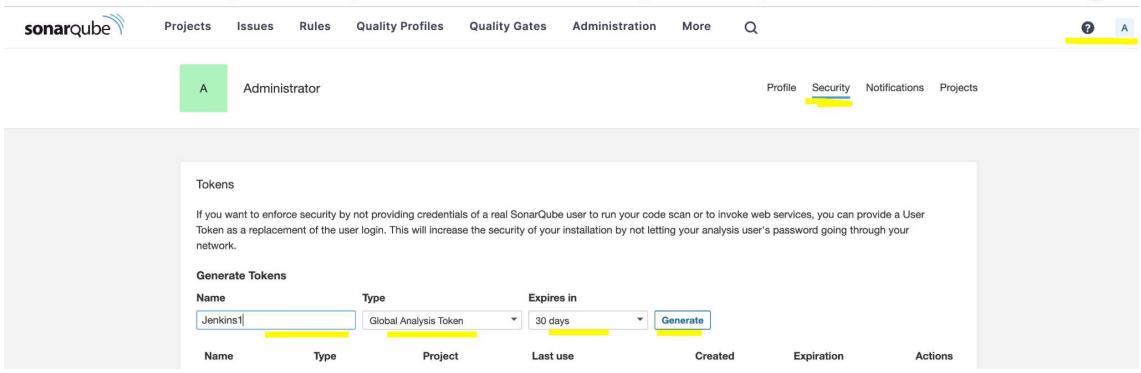
docker start <containerID>

Step 13.2 -> Login into sonar dashboard

Username – admin
Password – admin

Step 13.3 -> Create Sonar token for Jenkins

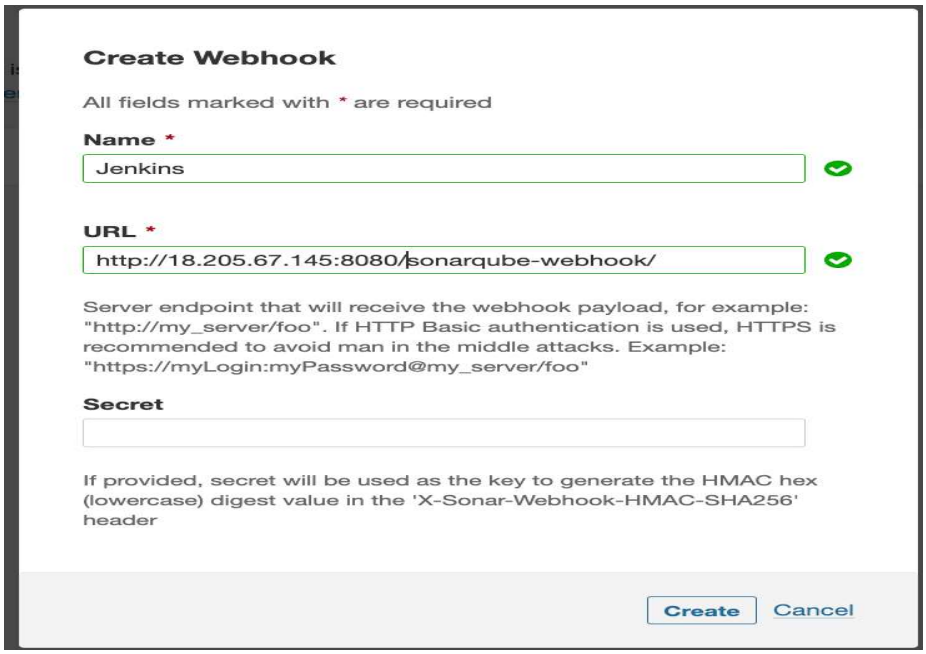
Sonar Dashboard -> Administration -> My Account -> Security -> Create token -> Save the token to some text file



Step 13.4 -> Integrate Sonar to Jenkins

Sonar Dashboard -> Administration -> Configuration -> webhooks -> Add the below name and url and save

http://<EC2_IP>:8080/sonarqube-webhook/



Step 14 – Install Maven

https://github.com/praveen1994dec/tools_installation_scripts/blob/main/Maven.sh

Step 15 – Install TRIVY for docker image scan

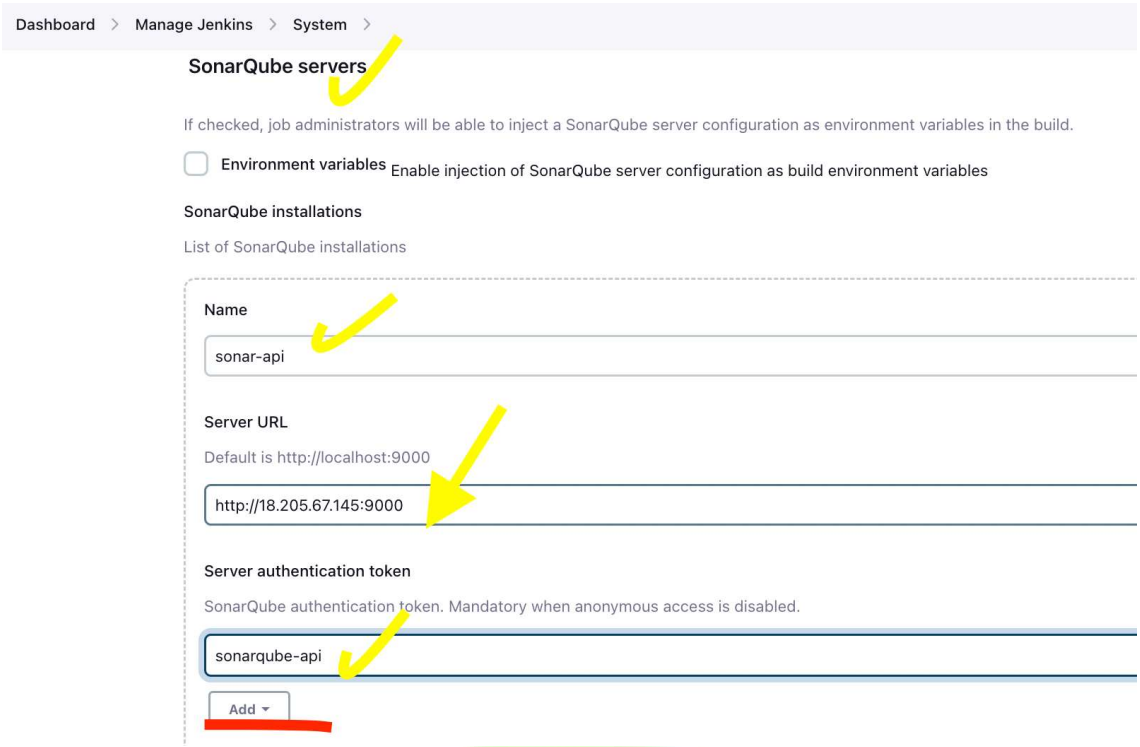
https://github.com/praveen1994dec/tools_installation_scripts/blob/main/trivy.sh

Integrate All tools with Jenkins

Jenkins Dashboard -> Manage Jenkins -> configure system

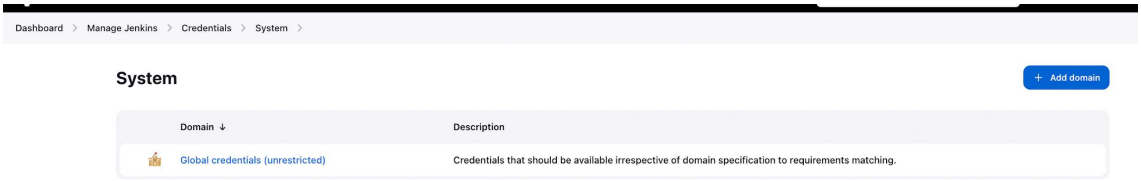
Step 16 – ADD SONARQUBE

Step 16.1 -> Click on sonarqube servers -> add url and name -> Click on add token -> Select Secret text -> Add the sonar token from step13.3 -> Give name of token as sonarqube-api

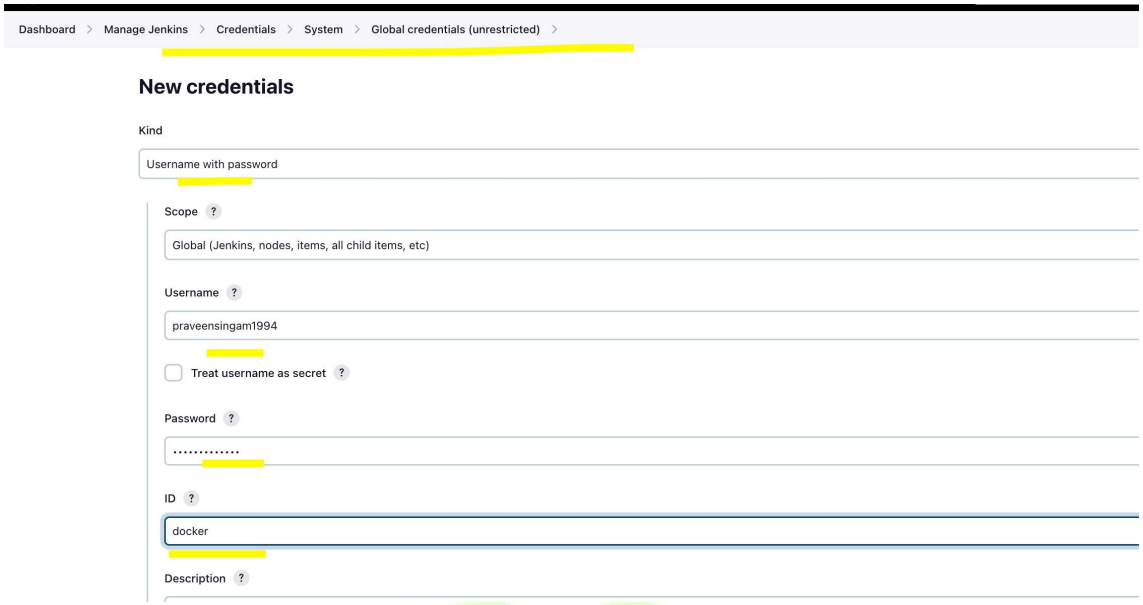


Step 17 - Add the docker HUB credentials ID

Jenkins dashboard -> Manage Jenkins -> Credentials -> System -> click on global credentials

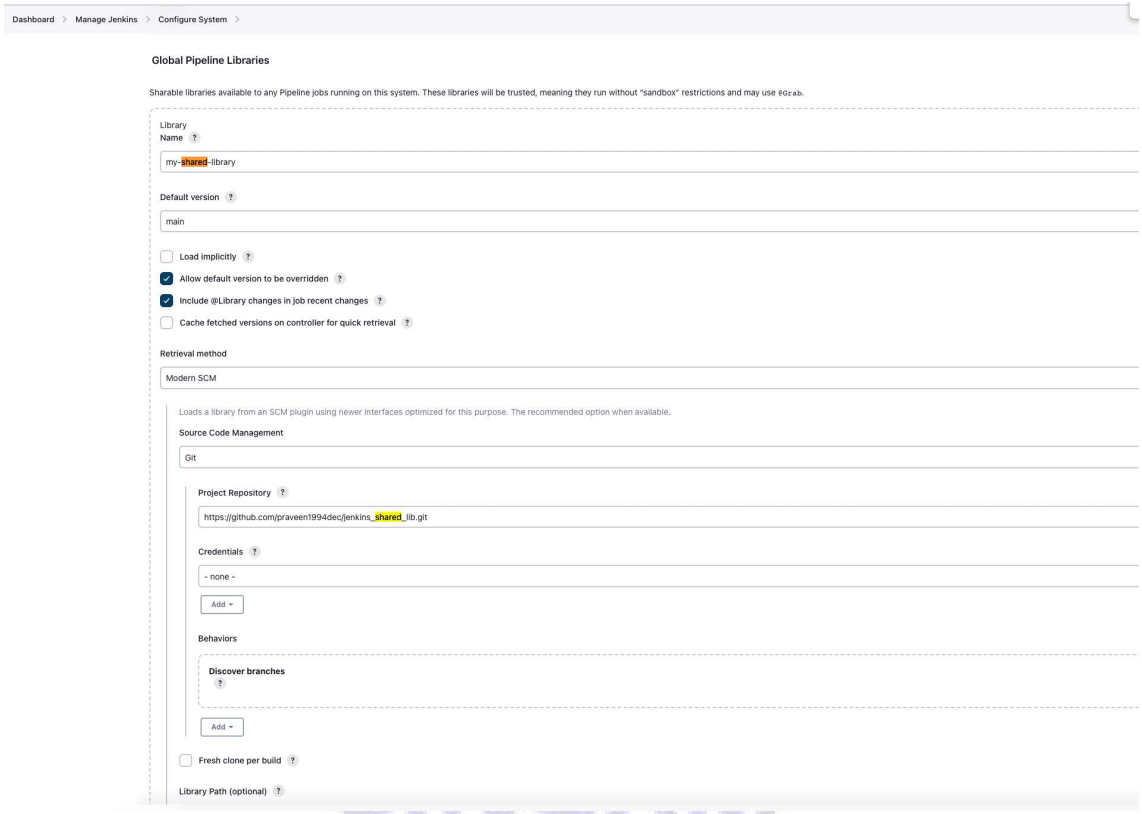


ADD the docker hub credentials with name as docker



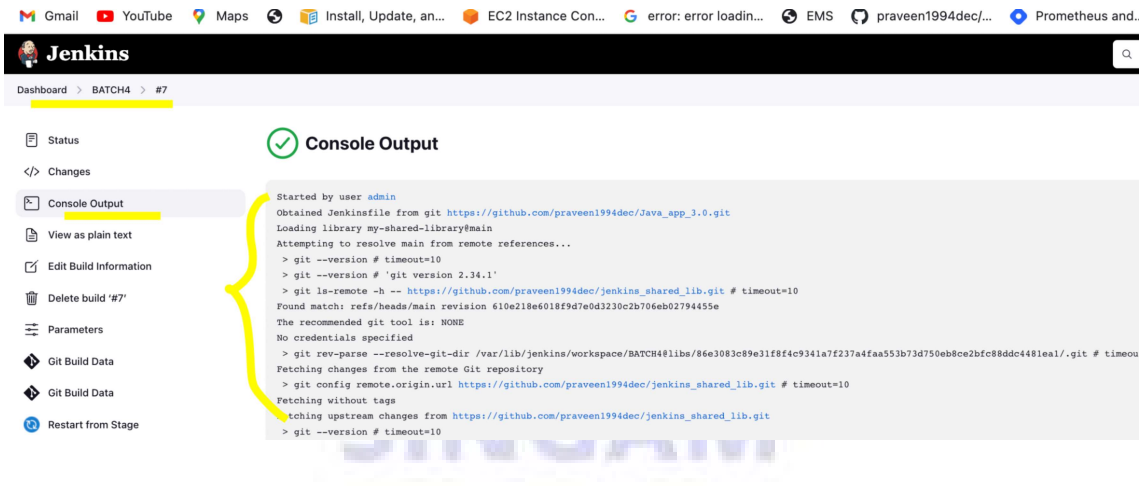
Step 18 – Add the Jenkins Shared library

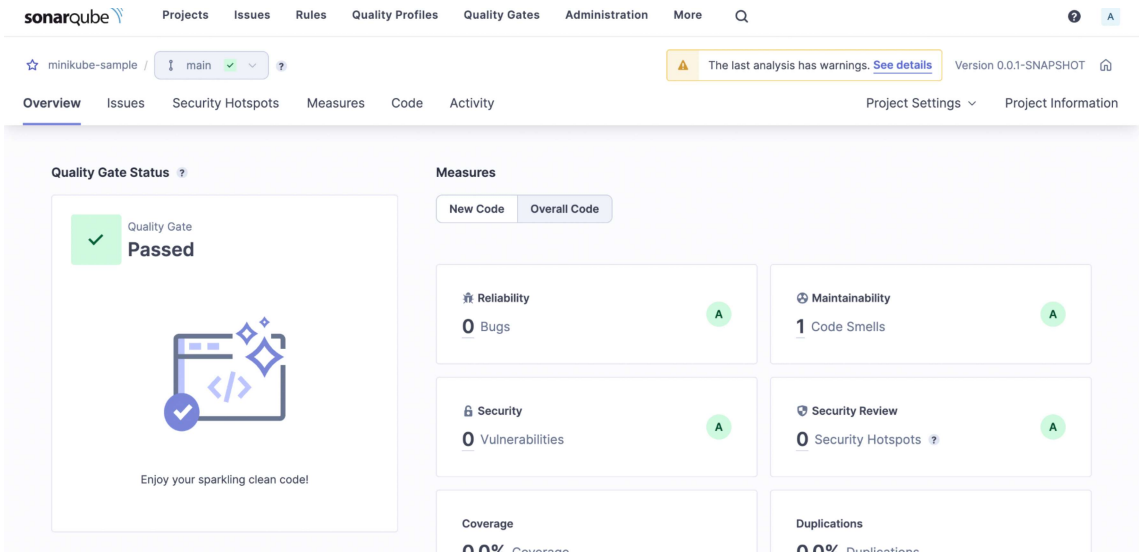
Go to Manage Jenkins -> Configure system -> Global pipeline library -> Add below data
Name - my-shared-library
Default version – main
Git - https://github.com/praveen1994dec/jenkins_shared_lib.git


Dashboard
Manage Jenkins
Configure System
Global Pipeline Libraries
Sharable libraries available to any Pipeline jobs running on this system. These libraries will be trusted, meaning they run without "sandbox" restrictions and may use @Grab.
Library
Name
my-shared-library
Default version
main
Load implicitly
Allow default version to be overridden
Include @Library changes in job recent changes
Cache fetched versions on controller for quick retrieval
Retrieval method
Modern SCM
Loads a library from an SCM plugin using newer interfaces optimized for this purpose. The recommended option when available.
Source Code Management
Git
Project Repository
https://github.com/praveen1994dec/jenkins_shared_lib.git
Credentials
- none -
Add
Behaviors
Discover branches
Add
Fresh clone per build
Library Path (optional)

Step 19 - Once pipeline is Run Check

- The Jenkins logs
- The Trivy scan vulnerabilities
- The sonarqube dashboard for report





SINGAM