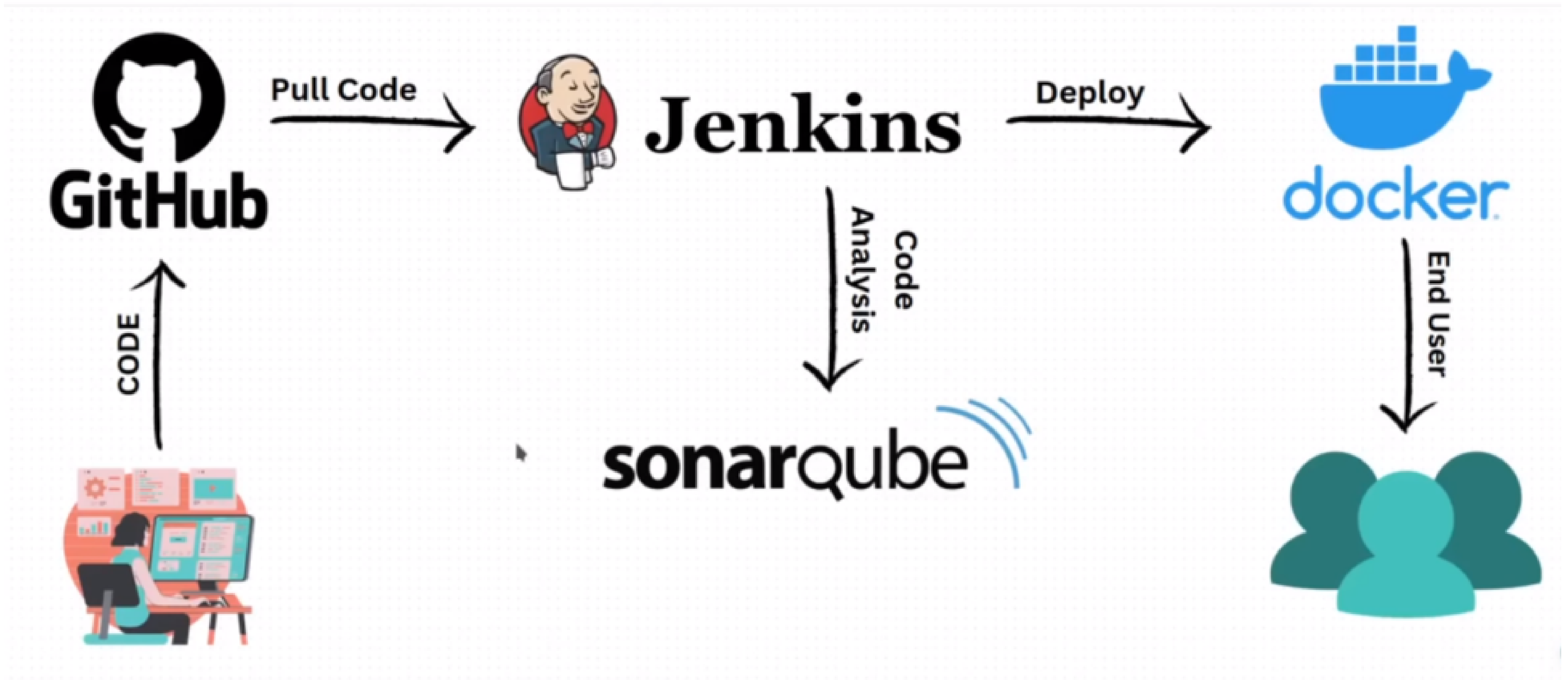
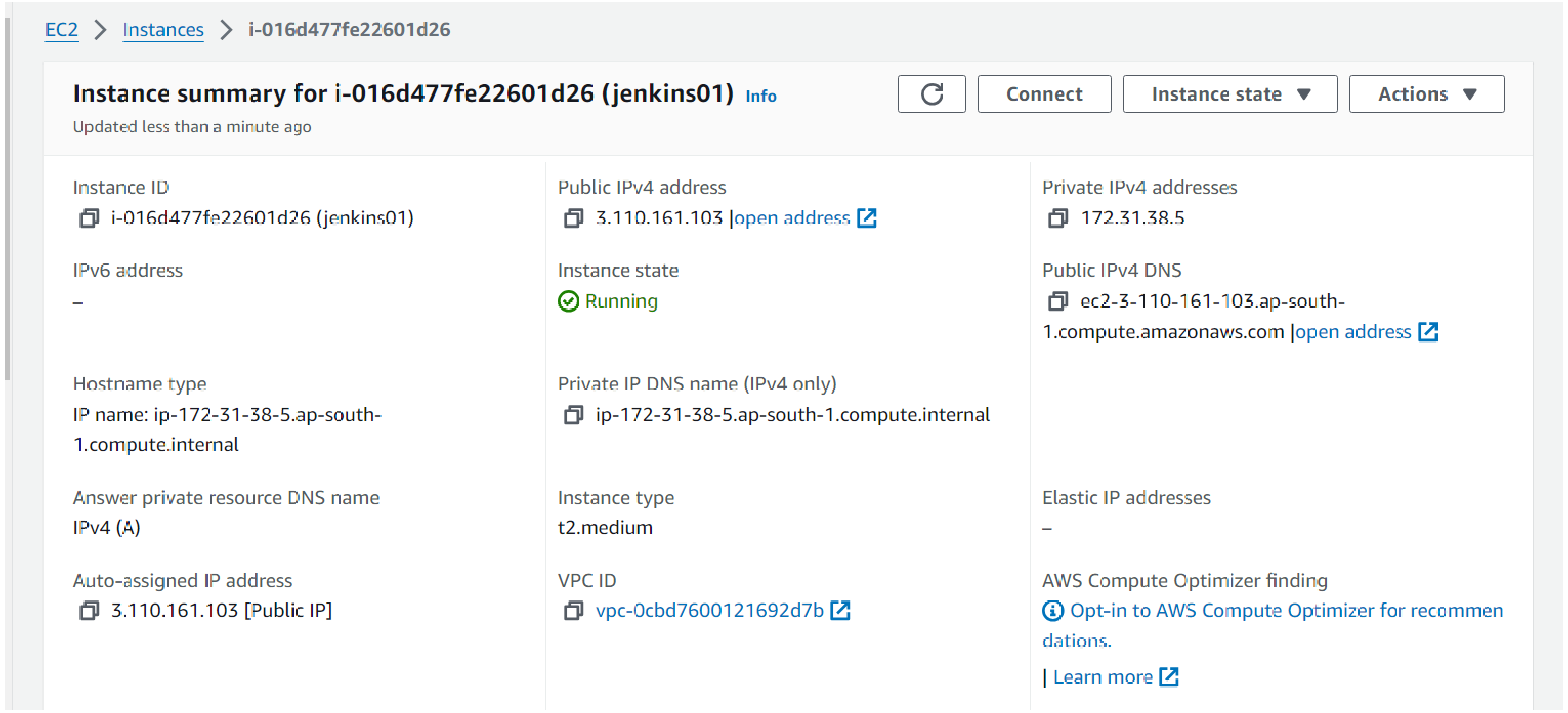
Deploying three tier architecture application [frontend, backend and database]



CI/CD Project with Github-Jenkins-Sonarqube-Docker

suyash bobade

step1: launch an Ec2 instance with t2.medium size and configured jenkins with the help of below commands:


EC2 > Instances > i-016d477fe22601d26

Instance summary for i-016d477fe22601d26 (jenkins01) Info

Updated less than a minute ago

Instance ID i-016d477fe22601d26 (jenkins01)	Public IPv4 address 3.110.161.103 |open address	Private IPv4 addresses 172.31.38.5
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-3-110-161-103.ap-south-1.compute.amazonaws.com |open address
Hostname type IP name: ip-172-31-38-5.ap-south-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-38-5.ap-south-1.compute.internal	
Answer private resource DNS name IPv4 (A)	Instance type t2.medium	Elastic IP addresses –
Auto-assigned IP address 3.110.161.103 [Public IP]	VPC ID vpc-0cbd7600121692d7b	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more

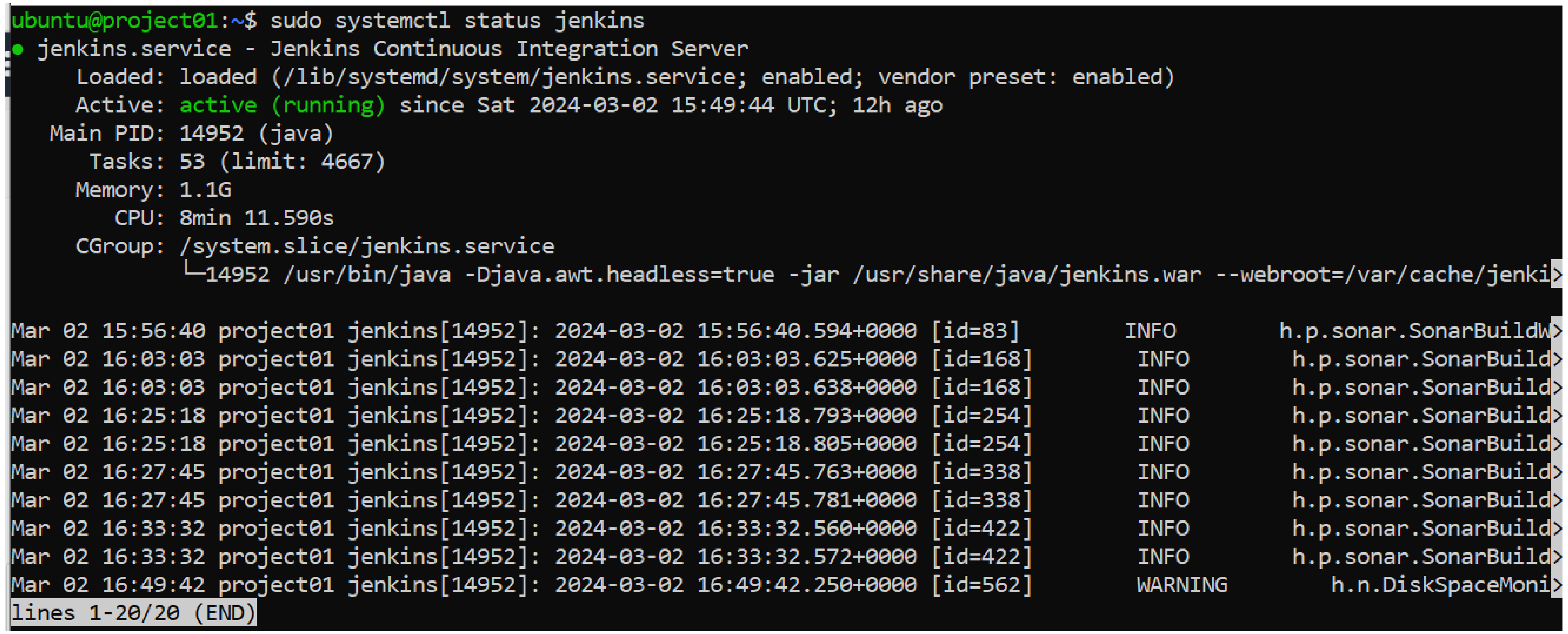
step:2 installation of java

```
sudo apt update

sudo apt install fontconfig openjdk-17-jre

java -version
```

#jenkins installation

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

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

sudo apt-get install jenkins  -y
```

step3: jenkins has been installed successfully let's configured jenkins server with necessary dependencies which are required for our project.



```
ubuntu@project01:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-03-02 15:49:44 UTC; 12h ago
   Main PID: 14952 (java)
      Tasks: 53 (limit: 4667)
     Memory: 1.1G
        CPU: 8min 11.590s
     CGroup: /system.slice/jenkins.service
             └─14952 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenki>

Mar 02 15:56:40 project01 jenkins[14952]: 2024-03-02 15:56:40.594+0000 [id=83]        INFO        h.p.sonar.SonarBuildW>
Mar 02 16:03:03 project01 jenkins[14952]: 2024-03-02 16:03:03.625+0000 [id=168]        INFO        h.p.sonar.SonarBuild>
Mar 02 16:03:03 project01 jenkins[14952]: 2024-03-02 16:03:03.638+0000 [id=168]        INFO        h.p.sonar.SonarBuild>
Mar 02 16:25:18 project01 jenkins[14952]: 2024-03-02 16:25:18.793+0000 [id=254]        INFO        h.p.sonar.SonarBuild>
Mar 02 16:25:18 project01 jenkins[14952]: 2024-03-02 16:25:18.805+0000 [id=254]        INFO        h.p.sonar.SonarBuild>
Mar 02 16:27:45 project01 jenkins[14952]: 2024-03-02 16:27:45.763+0000 [id=338]        INFO        h.p.sonar.SonarBuild>
Mar 02 16:27:45 project01 jenkins[14952]: 2024-03-02 16:27:45.781+0000 [id=338]        INFO        h.p.sonar.SonarBuild>
Mar 02 16:33:32 project01 jenkins[14952]: 2024-03-02 16:33:32.560+0000 [id=422]        INFO        h.p.sonar.SonarBuild>
Mar 02 16:33:32 project01 jenkins[14952]: 2024-03-02 16:33:32.572+0000 [id=422]        INFO        h.p.sonar.SonarBuild>
Mar 02 16:49:42 project01 jenkins[14952]: 2024-03-02 16:49:42.250+0000 [id=562]        WARNING        h.n.DiskSpaceMoni>
lines 1-20/20 (END)
```

***tip: make sure you have installed docker & docker compose and it's version more than 1.26 (above)**

sudo apt install docker.io -y

#docker-compose installation below

sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

```
sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
docker-compose --version
```

step4: trivy installtion steps:

(Trivy is an open-source vulnerability scanner for containers and containerized applications. It is designed to scan container images for known vulnerabilities in their dependencies.)

First, you need to add the Trivy repository and GPG key:

```
sudo apt-get install -y wget apt-transport-https gnupg lsb-release
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | sudo apt-key add -
echo deb https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main | sudo tee -a
/etc/apt/sources.list.d/trivy.list
sudo apt-get update
sudo apt-get install trivy -y
trivy -v
```

step5: install dependancy for nodejs (nodejs16)

sudo apt update

#Install the `curl` package if you don't have it:

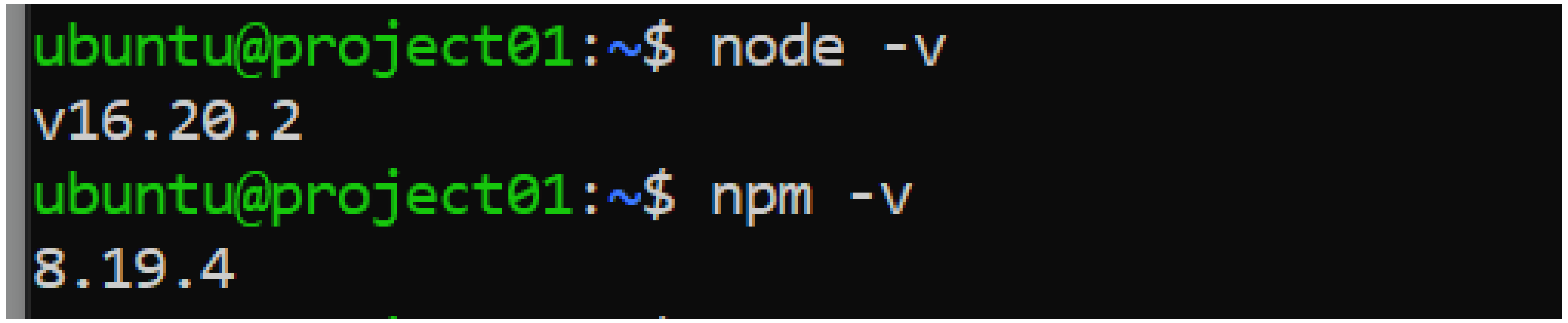
sudo apt install curl

# Download and install Node.js 16 using the following commands:

curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash -
sudo apt-get install -y nodejs

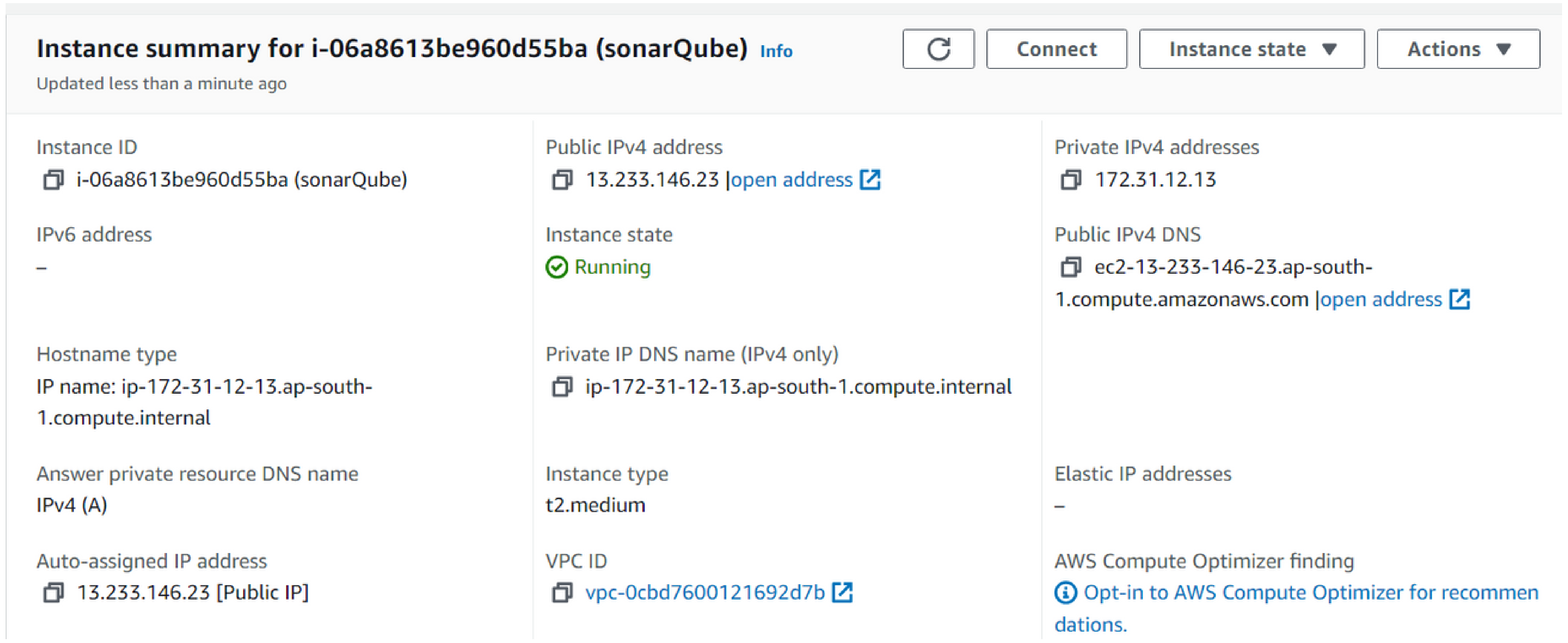
#To check if Node.js and npm (Node Package Manager) were installed successfully, you can run:

node -v
npm -v



```
ubuntu@project01:~$ node -v
v16.20.2
ubuntu@project01:~$ npm -v
8.19.4
```

step6: sonarQube installation on the different server. launch an Ec2 instance with t2.medium size and configured as per the below instruction.


Instance summary for i-06a8613be960d55ba (sonarQube)
Updated less than a minute ago

Instance ID i-06a8613be960d55ba (sonarQube)	Public IPv4 address 13.233.146.23 |open address	Private IPv4 addresses 172.31.12.13
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-13-233-146-23.ap-south-1.compute.amazonaws.com |open address
Hostname type IP name: ip-172-31-12-13.ap-south-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-12-13.ap-south-1.compute.internal	
Answer private resource DNS name IPv4 (A)	Instance type t2.medium	Elastic IP addresses –
Auto-assigned IP address 13.233.146.23 [Public IP]	VPC ID vpc-0cbd7600121692d7b	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations.

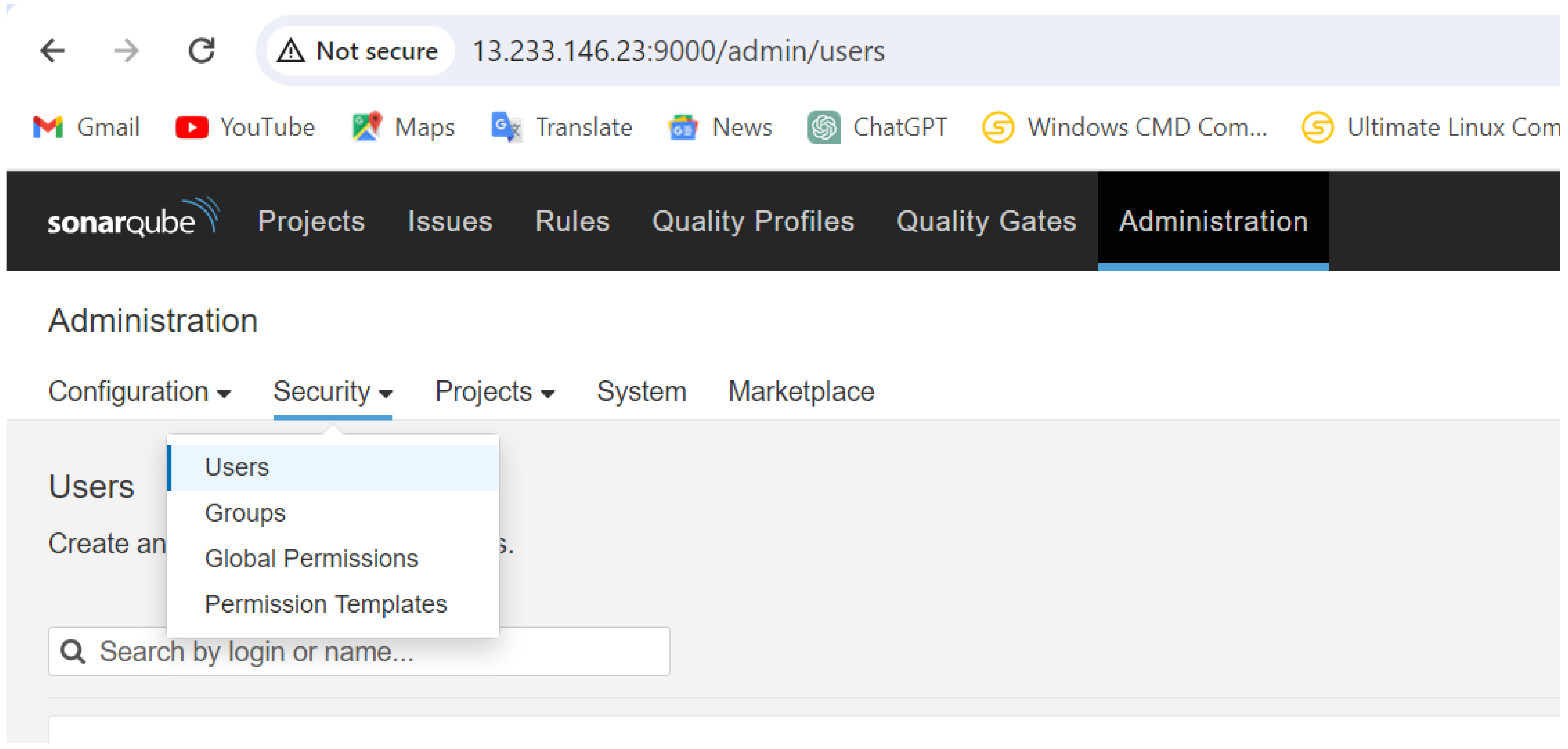
```
sudo apt update

sudo apt install docker.io -y

sudo docker run -d -p 9000:9000 --name sonar sonarqube:lts-community

#make sure to add 9000 port in your instance security group.
```

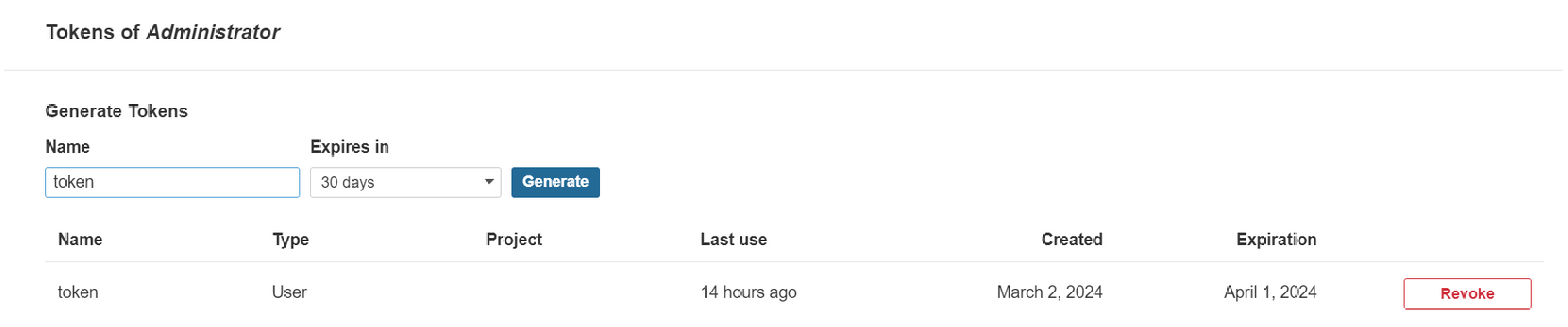
#Now access sonarqube server using instance public Ip
http://SonarQubePublicIP:9000

Default User is admin and Password is admin of sonarqube. After login change the password

step7: now generate a token and configured it with jenkins server as below

go to the 'Administration' > security > users as below





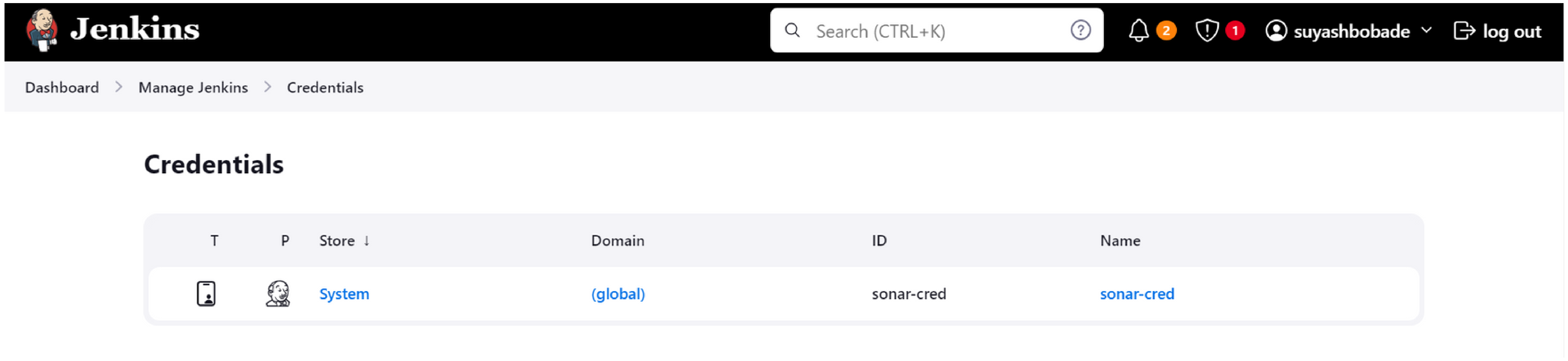
Tokens of *Administrator*

Generate Tokens

Name	Expires in
token	30 days

Name	Type	Project	Last use	Created	Expiration	
token	User		14 hours ago	March 2, 2024	April 1, 2024	Revoke

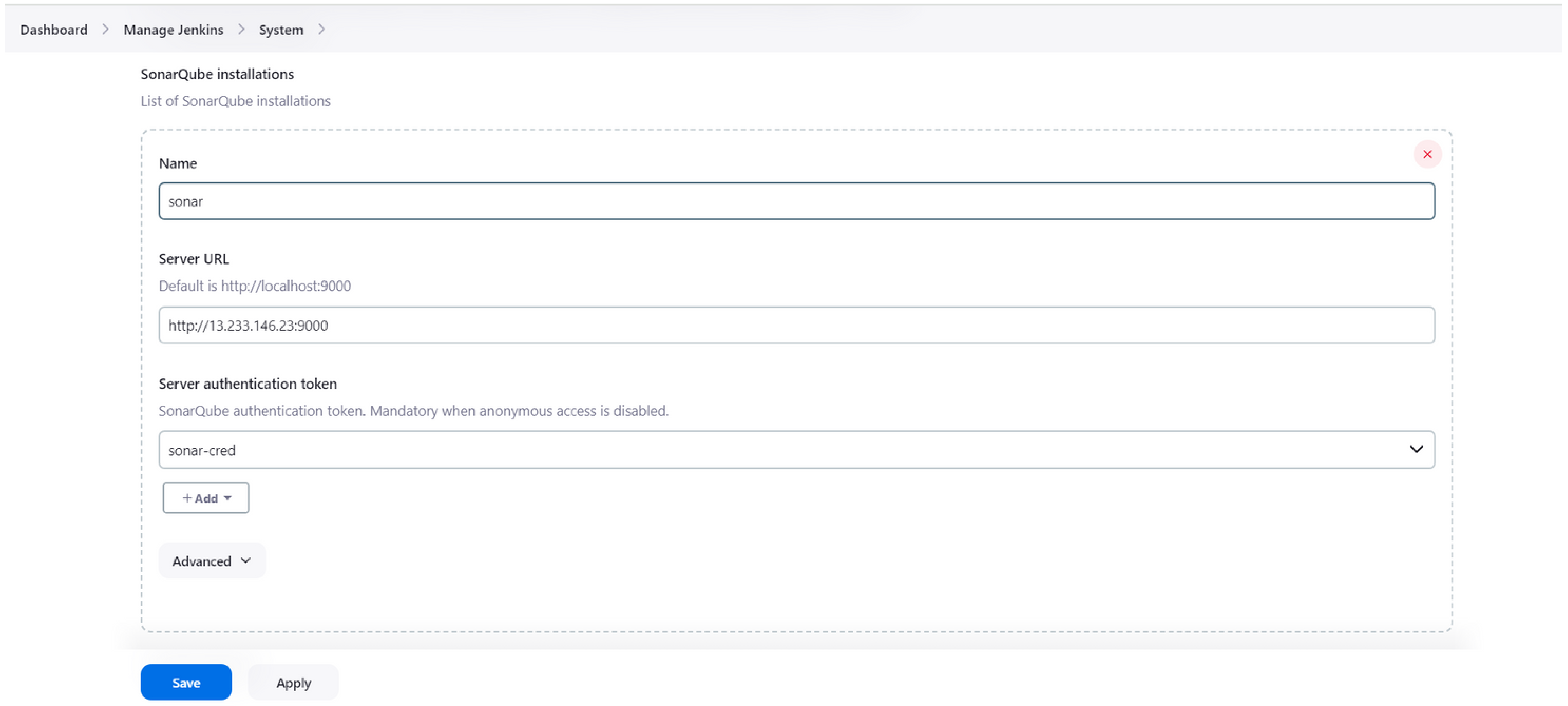
#after generating token go to the jenkins server, go to Dashboard> manage jenkins > give it to the 'credentials' click on 'global credentials' choose 'secret test' give name and then configured it with 'system as following'



Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	sonar-cred	sonar-cred

go to the 'dashboard'> 'manage jenkins' > 'system' choose 'sonarquebe installation' and configure.


Dashboard > Manage Jenkins > System >

SonarQube installations

List of SonarQube installations

Name: sonar

Server URL

Default is http://localhost:9000

http://13.233.146.23:9000

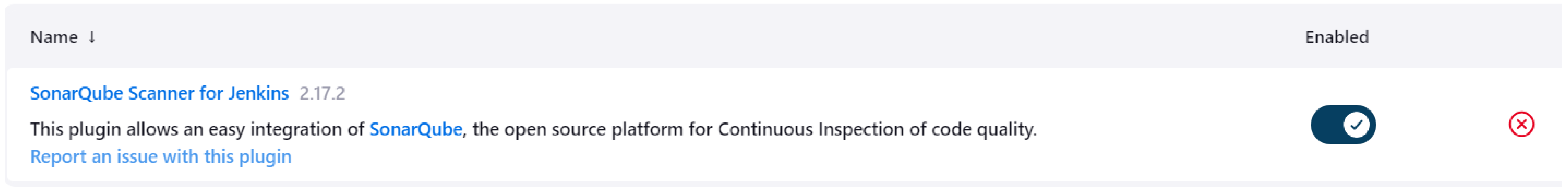
Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

sonar-cred

#now the sonarQube server has been configured succesfully with jenkins server.

step8: now let's install plugins which are required for the project.

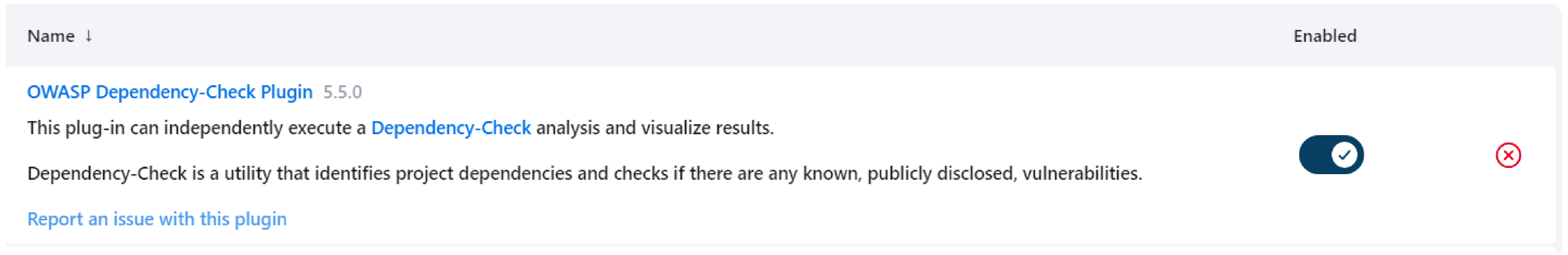


Name ↓ | Enabled

SonarQube Scanner for Jenkins 2.17.2

This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.

Report an issue with this plugin



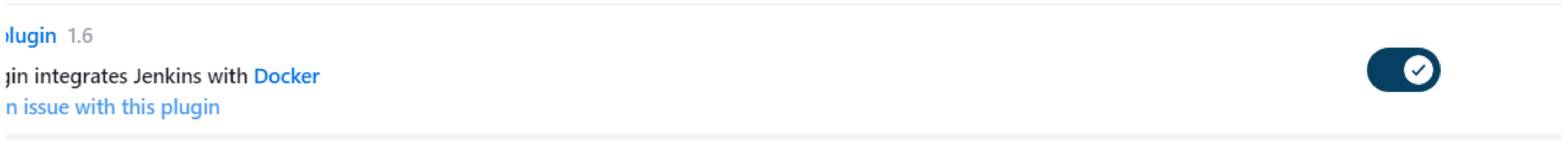
Name ↓ | Enabled

OWASP Dependency-Check Plugin 5.5.0

This plug-in can independently execute a Dependency-Check analysis and visualize results.

Dependency-Check is a utility that identifies project dependencies and checks if there are any known, publicly disclosed, vulnerabilities.

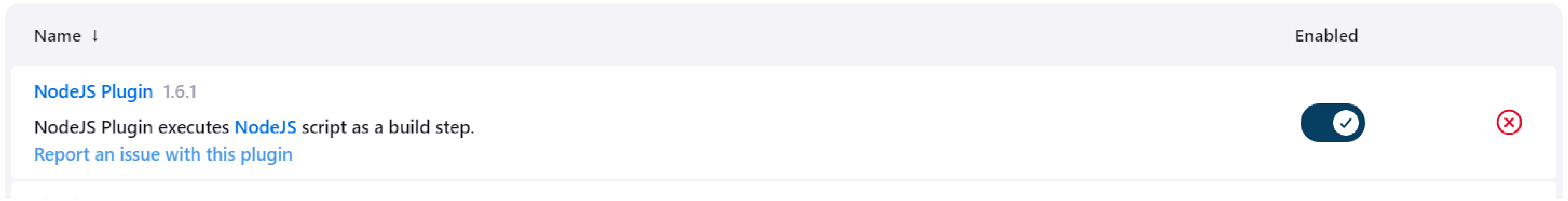
Report an issue with this plugin



ılugin 1.6

ȷin integrates Jenkins with Docker

n issue with this plugin

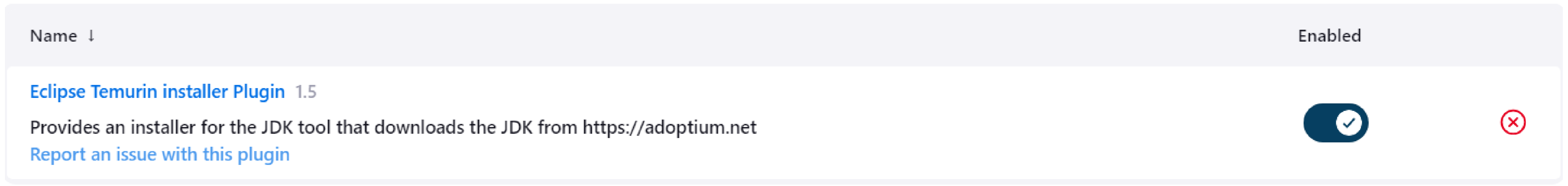


Name ↓ | Enabled

NodeJS Plugin 1.6.1

NodeJS Plugin executes NodeJS script as a build step.

Report an issue with this plugin



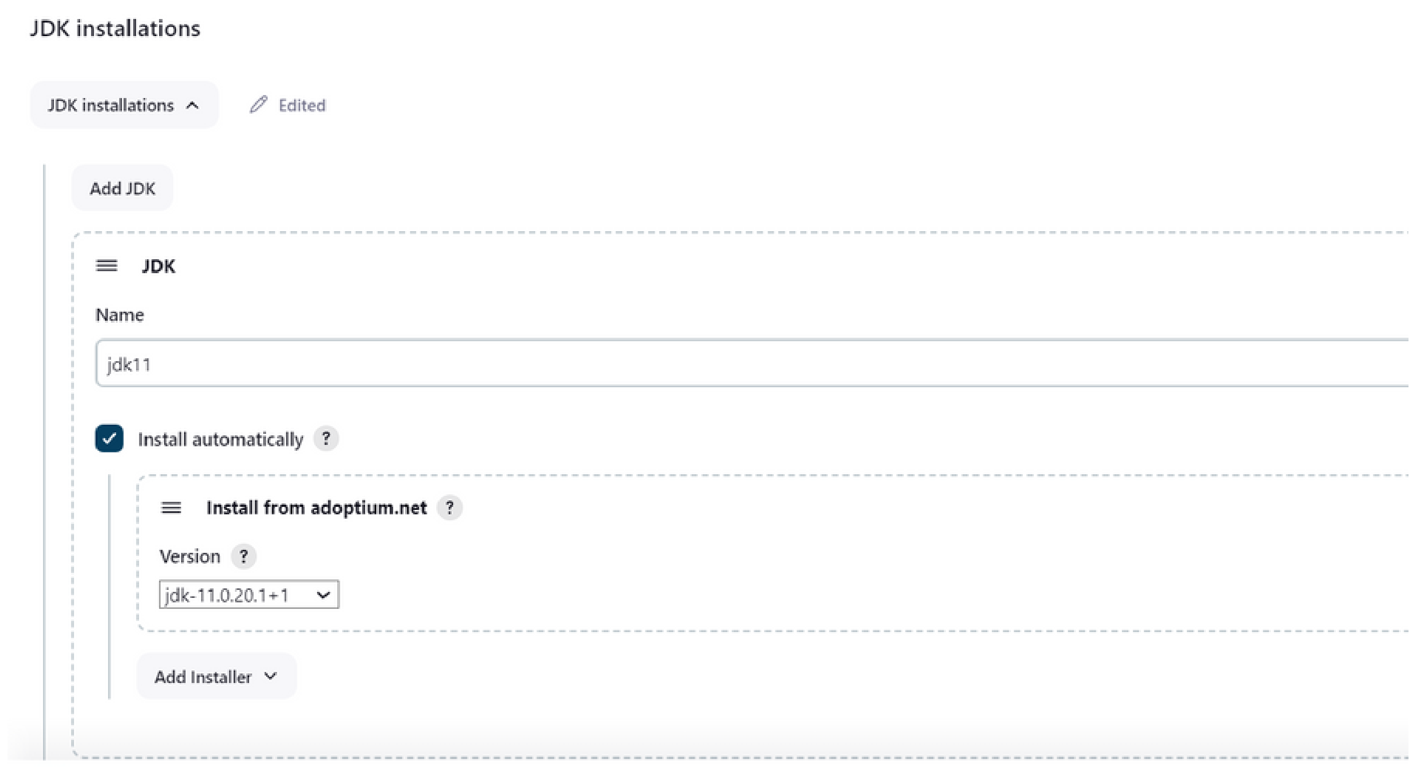
Name ↓ | Enabled

Eclipse Temurin installer Plugin 1.5

Provides an installer for the JDK tool that downloads the JDK from https://adoptium.net

Report an issue with this plugin

step9: now configure tools as below. go to the 'dashboard' > 'manage jenkins' > 'tool' as following.


JDK installations

JDK installations ^   Edited

Add JDK

JDK

Name

jdk11

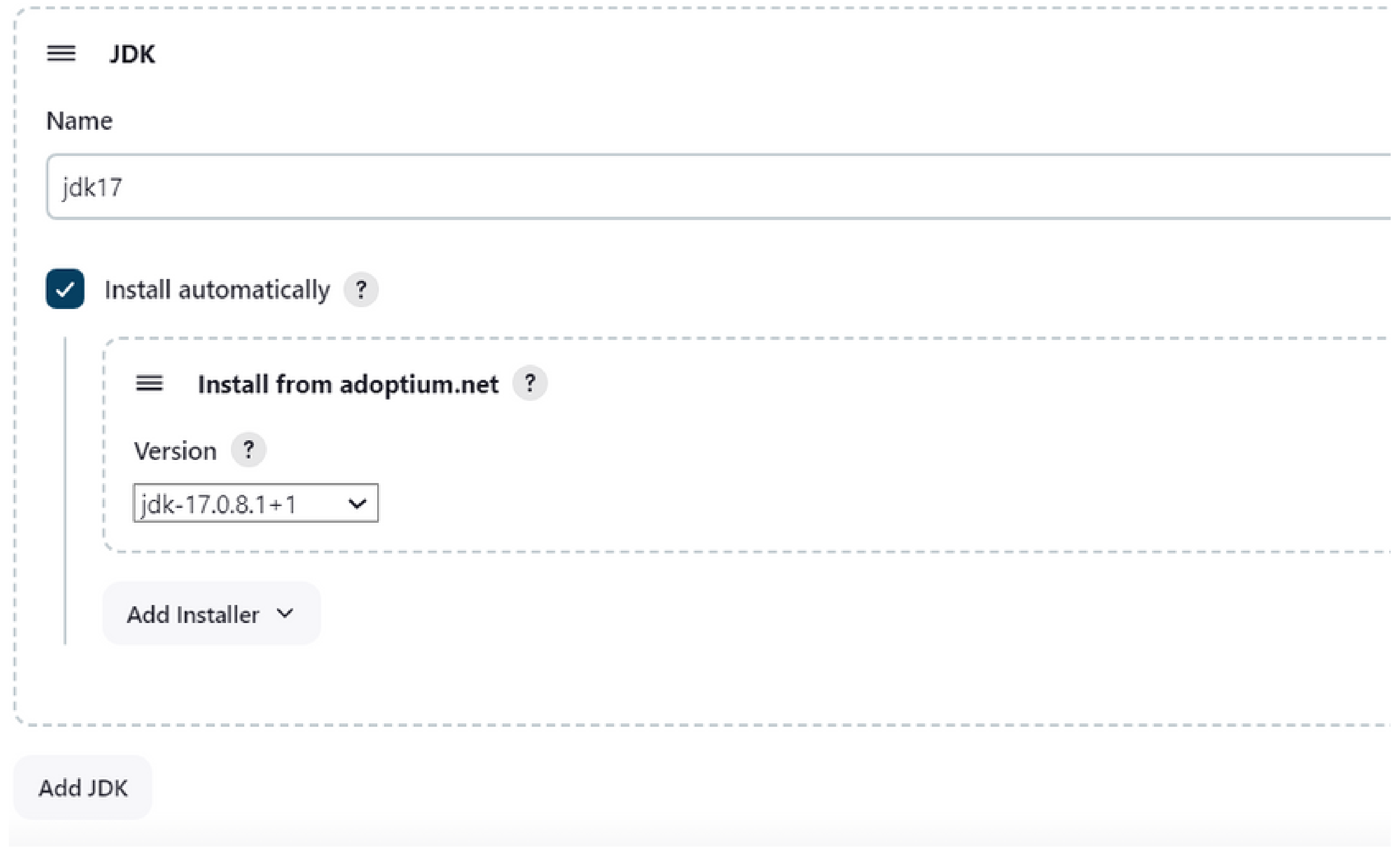
✓ Install automatically ?

Install from adoptium.net ?

Version ?

jdk-11.0.20.1+1

Add Installer


JDK

Name

jdk17

✓ Install automatically ?

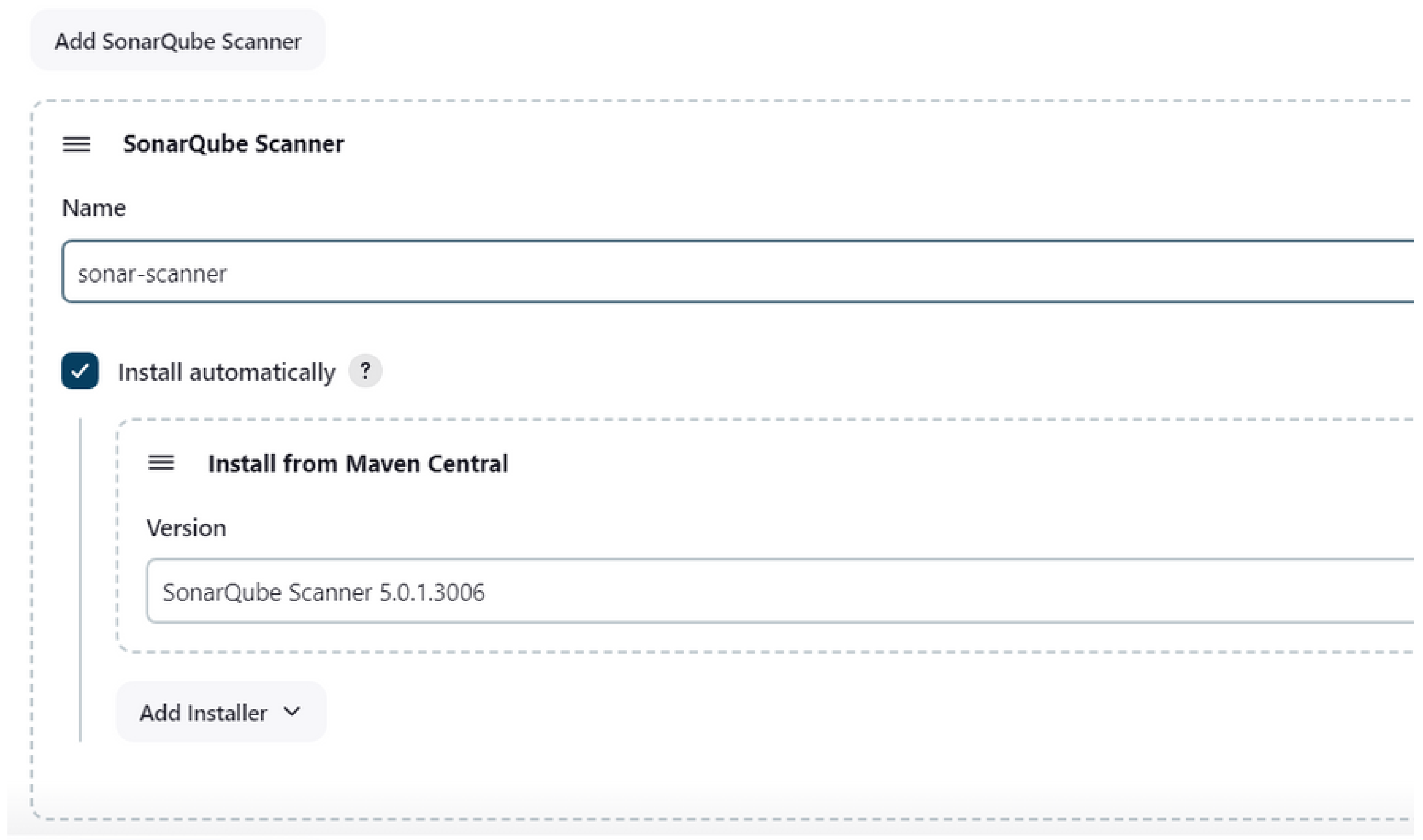
Install from adoptium.net ?

Version ?

jdk-17.0.8.1+1

Add Installer

Add JDK


Add SonarQube Scanner

SonarQube Scanner

Name

sonar-scanner

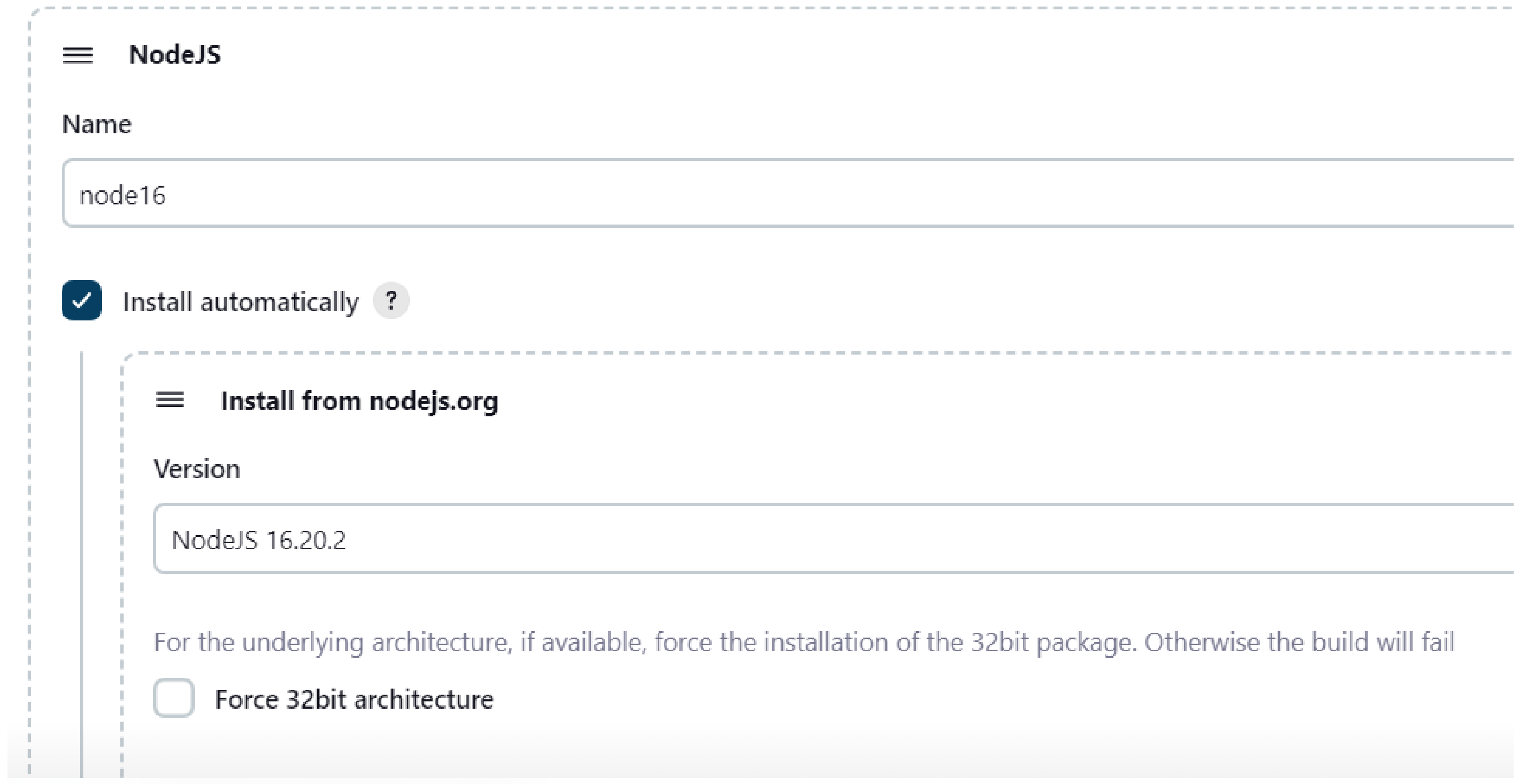
✓ Install automatically ?

Install from Maven Central

Version

SonarQube Scanner 5.0.1.3006

Add Installer


NodeJS

Name

node16

✓ Install automatically ?

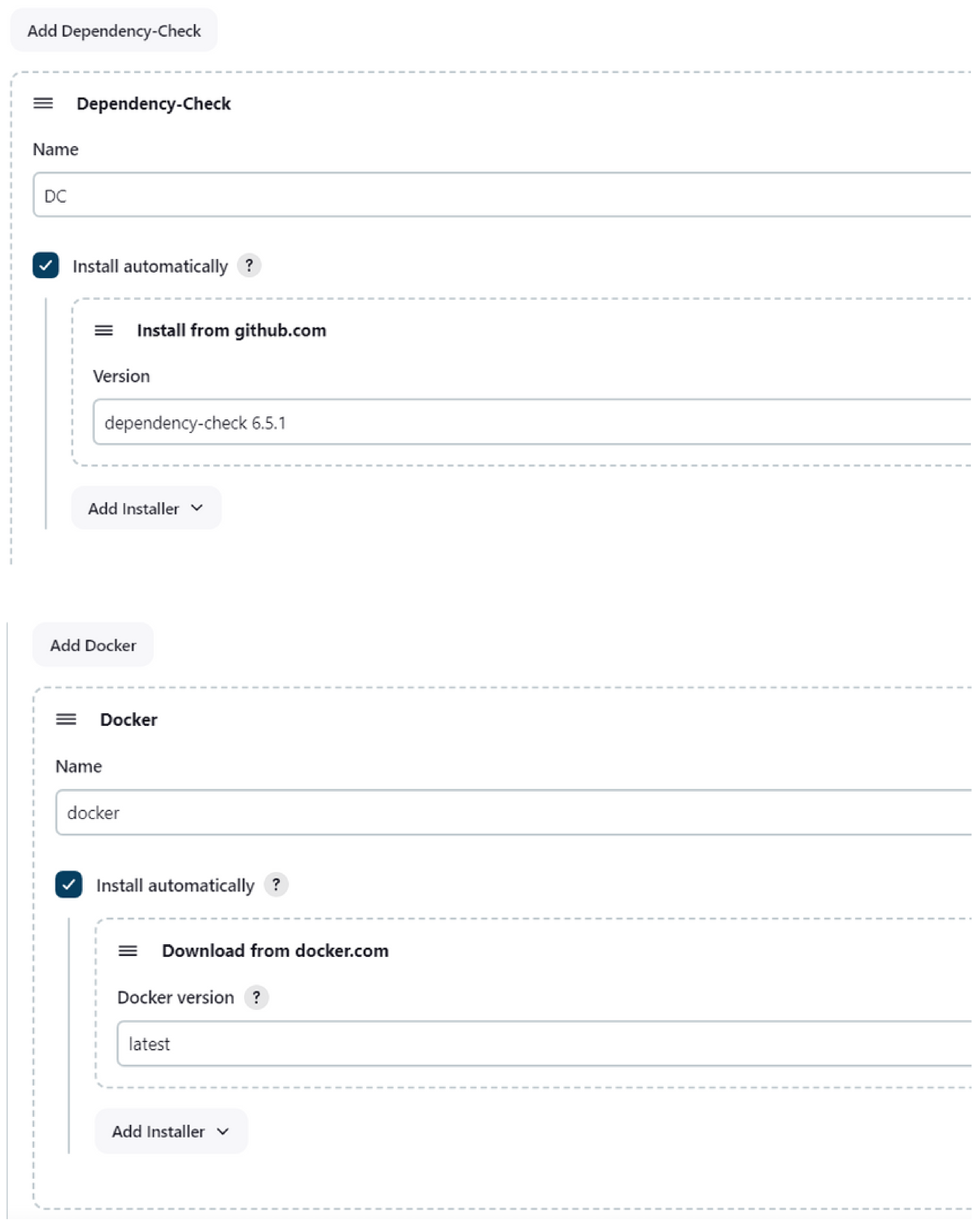
Install from nodejs.org

Version

NodeJS 16.20.2

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail

☐ Force 32bit architecture


Add Dependency-Check

Dependency-Check

Name

DC

✓ Install automatically ?

Install from github.com

Version

dependency-check 6.5.1

Add Installer

Add Docker

Docker

Name

docker

✓ Install automatically ?

Download from docker.com

Docker version ?

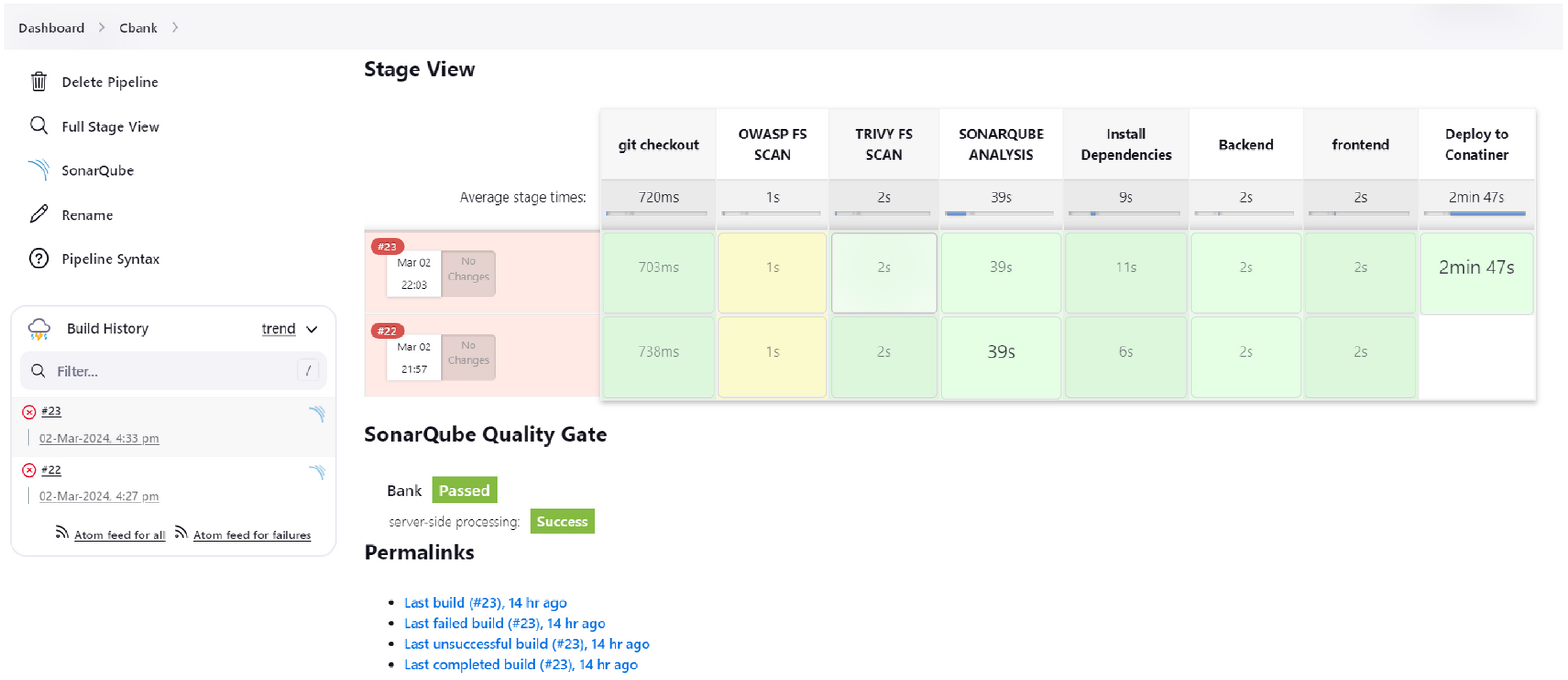
latest

Add Installer

step10: now go to 'dashboard' create 'New item' give a suitable name and choose pipeline project and write pipelines as follow. make sure to go step by step.

#repository link: https://github.com/suyash3903/fullstack-bank.git

#pipeline syntax (i will suggest to go with "hello word pipeline syntax", try to resolve the errors,bugs during deployment.)



Stage View

	git checkout	OWASP FS SCAN	TRIVY FS SCAN	SONARQUBE ANALYSIS	Install Dependencies	Backend	frontend	Deploy to Conatiner
Average stage times:	720ms	1s	2s	39s	9s	2s	2s	2min 47s
#23 Mar 02 22:03 No Changes	703ms	1s	2s	39s	11s	2s	2s	2min 47s
#22 Mar 02 21:57 No Changes	738ms	1s	2s	39s	6s	2s	2s	

#23 02-Mar-2024, 4:33 pm
#22 02-Mar-2024, 4:27 pm

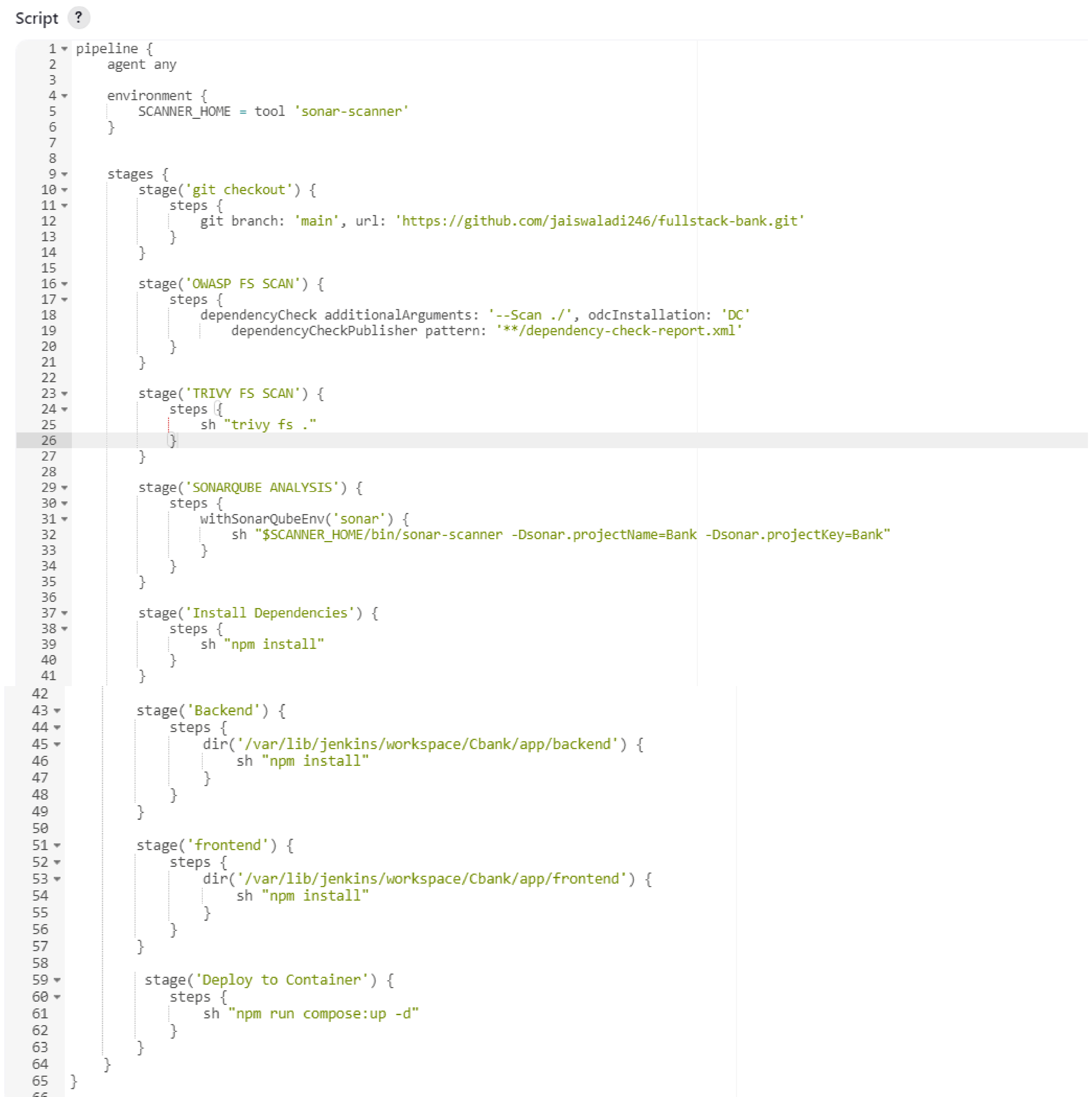
SonarQube Quality Gate

Bank Passed
server-side processing: Success

Permalinks

- Last build (#23), 14 hr ago
- Last failed build (#23), 14 hr ago
- Last unsuccessful build (#23), 14 hr ago
- Last completed build (#23), 14 hr ago

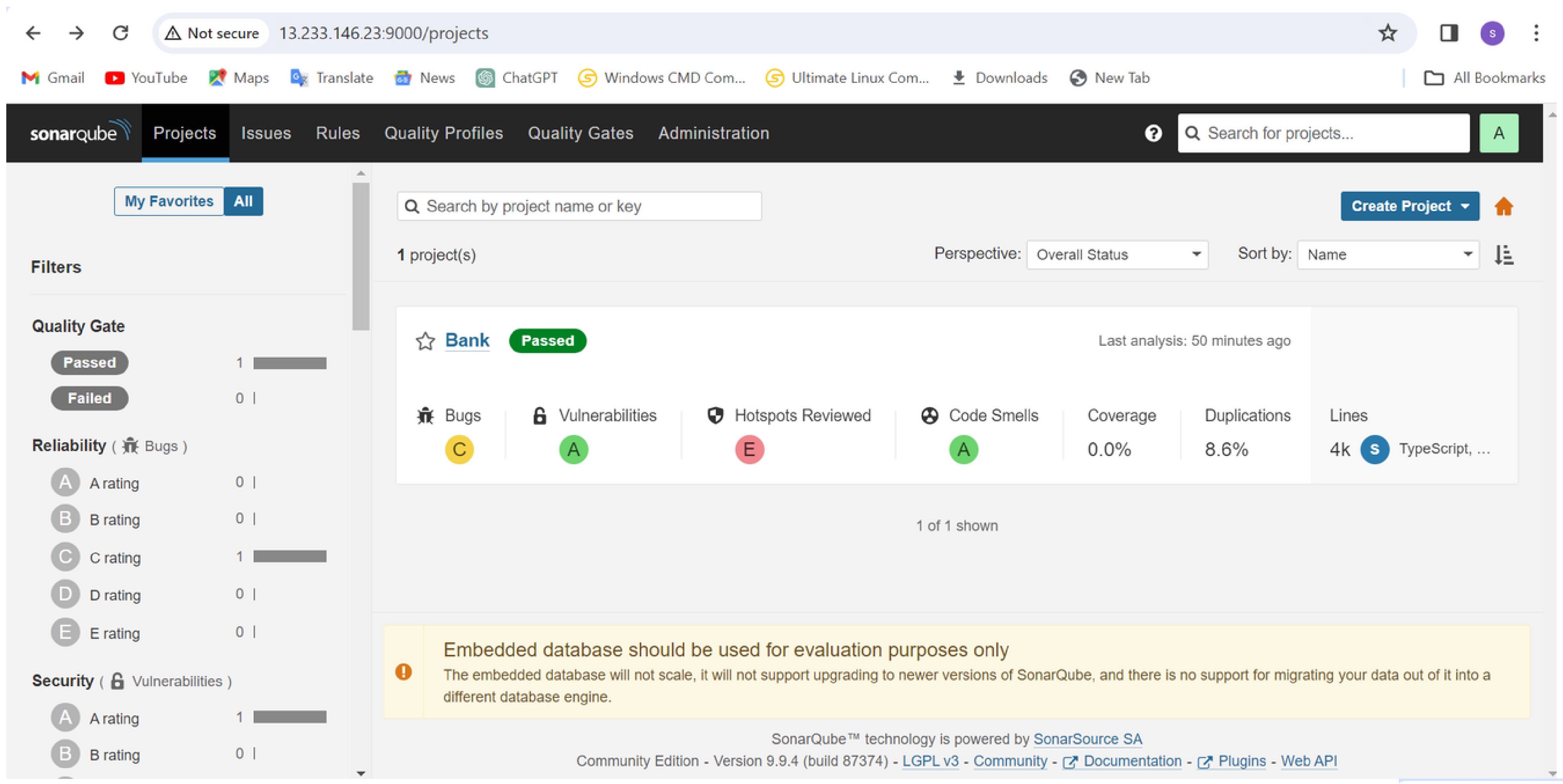
#pipeline



Script

```
pipeline {
    agent any

    environment {
        SCANNER_HOME = tool 'sonar-scanner'
    }


    stages {
        stage('git checkout') {
            steps {
                git branch: 'main', url: 'https://github.com/jaiswaladi246/fullstack-bank.git'
            }
        }

        stage('OWASP FS SCAN') {
            steps {
                dependencyCheck additionalArguments: '--Scan ./', odcInstallation: 'DC'
                    dependencyCheckPublisher pattern: '**/dependency-check-report.xml'
            }
        }

        stage('TRIVY FS SCAN') {
            steps {
                sh "trivy fs ."
            }
        }

        stage('SONARQUBE ANALYSIS') {
            steps {
                withSonarQubeEnv('sonar') {
                    sh "$SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Bank -Dsonar.projectKey=Bank"
                }
            }
        }

        stage('Install Dependencies') {
            steps {
                sh "npm install"
            }
        }

        stage('Backend') {
            steps {
                dir('/var/lib/jenkins/workspace/Cbank/app/backend') {
                    sh "npm install"
                }
            }
        }

        stage('frontend') {
            steps {
                dir('/var/lib/jenkins/workspace/Cbank/app/frontend') {
                    sh "npm install"
                }
            }
        }

         stage('Deploy to Container') {
            steps {
                sh "npm run compose:up -d"
            }
        }
    }
}
```

#sonarQube analysis code quality check report



Not secure 13.233.146.23:9000/projects

Gmail YouTube Maps Translate News ChatGPT Windows CMD Com... Ultimate Linux Com... Downloads New Tab All Bookmarks

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration Search for projects...

My Favorites All

Search by project name or key

Create Project

1 project(s)

Perspective: Overall Status Sort by: Name

Filters

Quality Gate

Passed	1	
Failed	0	

Reliability (Bugs)

A rating	0
B rating	0
C rating	1
D rating	0
E rating	0

Security (Vulnerabilities)

A rating	1
B rating	0

Bank Passed Last analysis: 50 minutes ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
C	A	E	A	0.0%	8.6%	4k S TypeScript, ...

1 of 1 shown

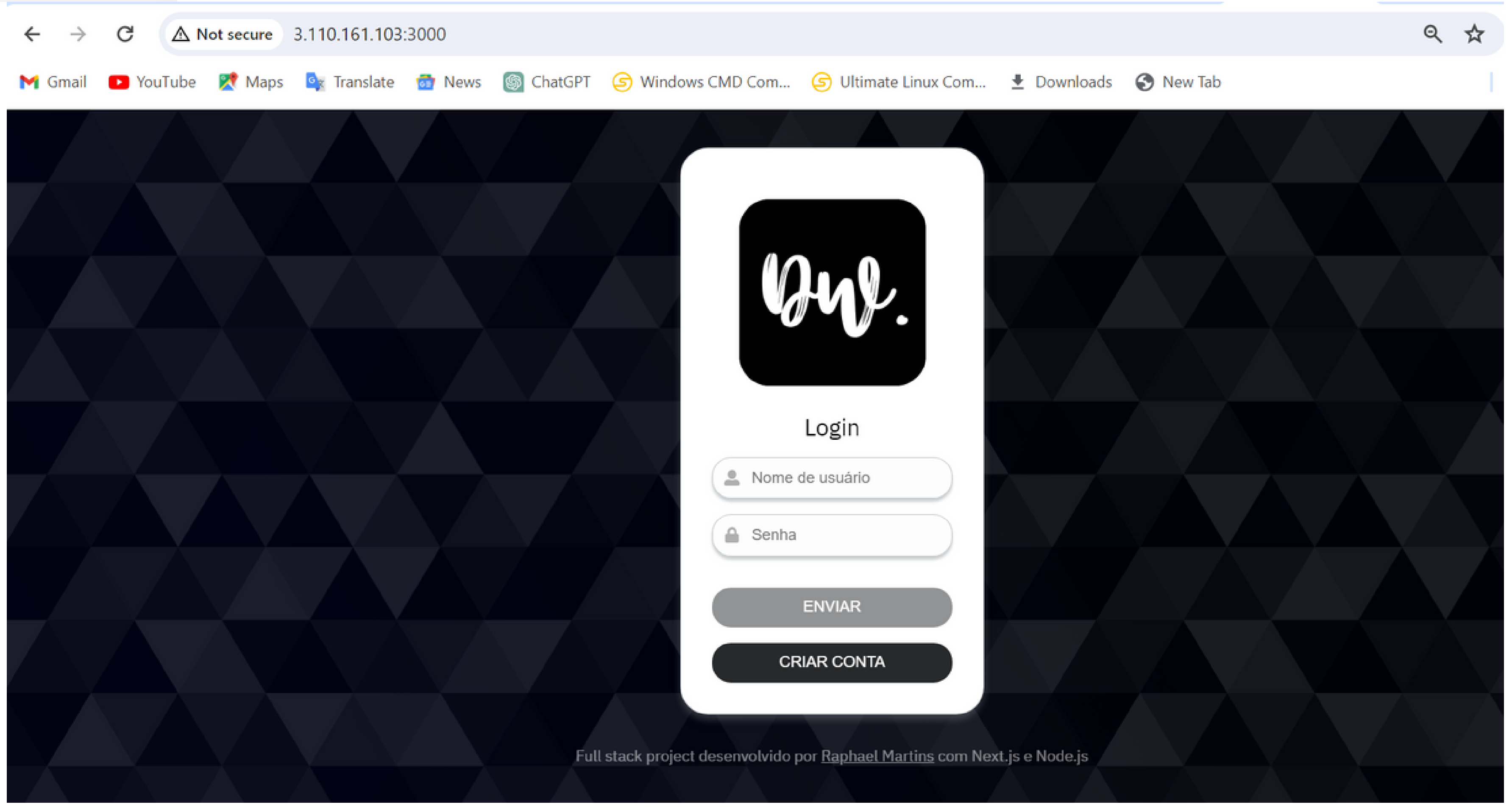
Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA

Community Edition - Version 9.9.4 (build 87374) - LGPL v3 - Community - Documentation - Plugins - Web API

#congratulations application has been deployed



Not secure 3.110.161.103:3000

Gmail YouTube Maps Translate News ChatGPT Windows CMD Com... Ultimate Linux Com... Downloads New Tab

Login

Nome de usuário

Senha

ENVIAR

CRIAR CONTA

Full stack project desenvolvido por Raphael Martins com Next.js e Node.js