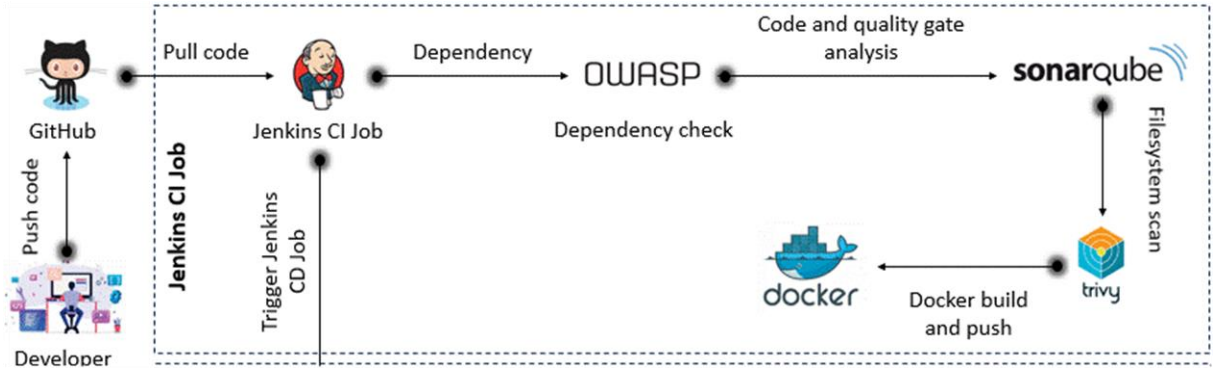


Requirement of this project

- Linux (Ubuntu)
- GitHub (Code)
- Docker (Containerization)
- Jenkins (CI)
- OWASP (Dependency check)
- SonarQube (Quality)
- Trivy (Filesystem Scan)
- Redis (Caching)

This project I complete in AWS UBUNTU EC2 Instance. So all the Requirement I install in EC2 Ubuntu machine.

Docker Install steps:

sudo apt-get update

sudo apt-get install docker.io -y

sudo usermod -aG docker ubuntu && newgrp docker

docker ps

Jenkins Install steps:

sudo apt update -y

sudo apt install fontconfig openjdk-17-jre -y

```
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 -y
sudo apt-get install jenkins -y
sudo systemctl status Jenkins
```

(now Jenkins use port no 8080 so you go tour instance security group and add the port 8080 rules)

Go to web browser search http://ec2 public address:8080

Now you see the Jenkins admin pages where

Username : admin

Password: **********(see the password in your ec2 /var/lib/Jenkins/secrets/initialAdminPassword)

No you install suggest plugins

Go to Jenkins -> manage Jenkins-> search and install the require plugins

- SonarQube Scanner
- Sonar Quality Gates
- OWASP Dependency-Check
- Docker

SonarQube setup

(Go to ubuntu EC2 run sonarqube container)

```
docker run -itd --name SonarQube-Server -p 9000:9000 sonarqube:lts-community
```

(go to instance security group add 9000 port, now go to web browser search http://ec2 public ip:9000 now you login sonarqube username: admin , password : admin)

Trivy Setup

(go to ubuntu ec2)

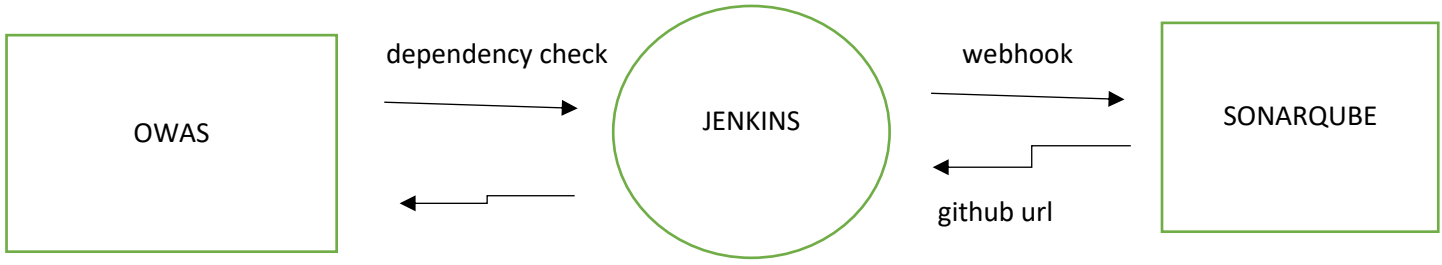
sudo apt-get install wget apt-transport-https gnupg lsb-release -y

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 -y

sudo apt-get install trivy -y



Create WebHook process between sonarqube and jenkins

Go to SonarQube ->Administration->Configuration->Webhooks->create

Name: Jenkins

url: http://ec2public ip:8080/sonarqube-webhook/

create

(token create in sonarqube for Jenkins)

Go to SonarQube ->Administration->security->user->beside administrator click token icon

Name: admin

Generate(copy the token)

Go to Jenkins->manage Jenkins -> Security-> Credentials->global->Add Credentials

Kind: secret text

Scope: Global

Secret: **********(paste the secret)

ID: Sonar

Description: Sonar

Go to Jenkins->manage Jenkins->system->SonarQube servers->add Sonarqube server

Name: Sonar

Server URL: http://ec2public ip:9000

Server Authentication token: Sonar

Install SonarQube Quality Gates tools in Jenkins

Jenkins-> manage Jenkins-> Tools-> SonarQube Scanner installations-> Add SonarQube Scanner

Name: Sonar

Version: (select latest version)

Install OWAS tools in Jenkins

Jenkins-> manage Jenkins-> Tools->Dependency-Check installations->add dependency-check

Name: dc

Install automatically: install from github.com

Create Declarative Pipeline in Jenkins

Go Jenkins-> Manage Jenkins-> New Item

Item name: wanderlust CI-CD

Select pipeline

Description : This is a CI/CD DevSecOps for Wanderlust project

- ✓ GitHub project
  Project url: https://github.com/santanuroul123/wanderlust
- ✓ Throttle builds

Build Triggers

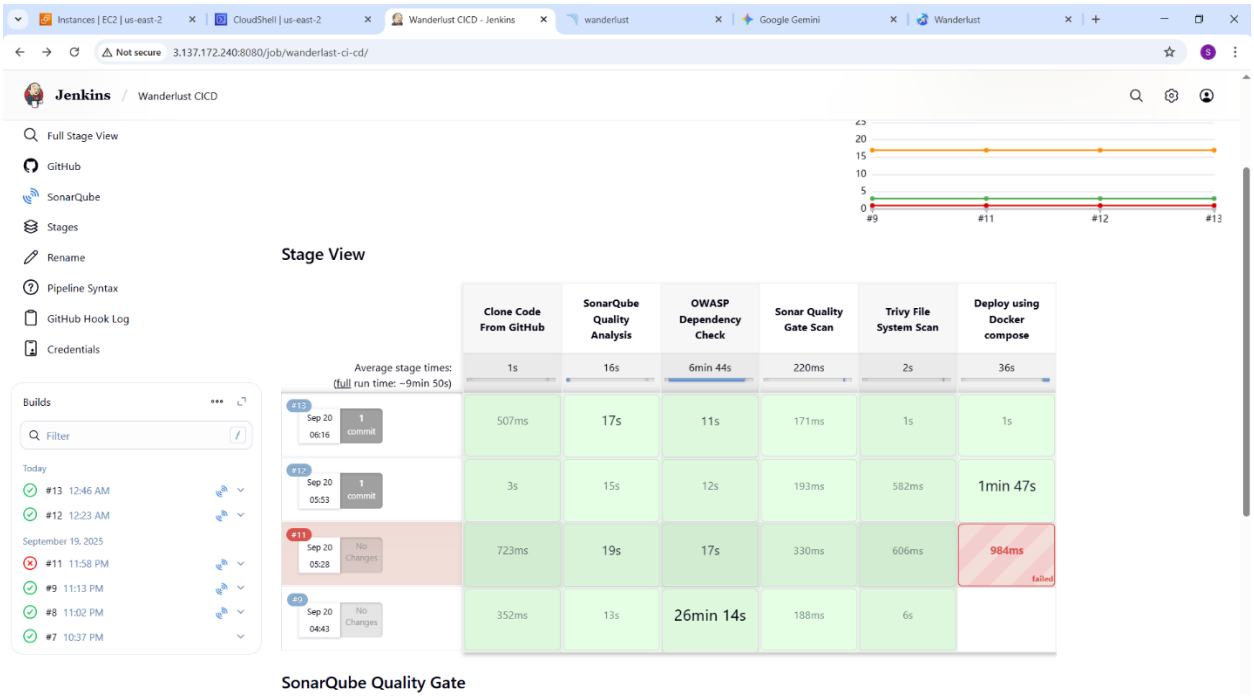
- ✓ GitHub hook trigger for GITScm polling

Advanced Project Options

Display Name: Wanderlust CICD

pipeline scripts

```
pipeline{
    agent any
    environment{
        SONAR_HOME= tool "Sonar"
    }
    stages{
        stage("Clone Code From GitHub"){
            steps{
                git url: "https://github.com/santanuroul123/wanderlust", branch: "main"
            }
        }
        stage("SonarQube Quality Analysis"){
            steps{
                withSonarQubeEnv("Sonar"){
                    sh "$SONAR_HOME/bin/sonar-scanner -Dsonar.projectName=wanderlust -
Dsonar.projectKey=wanderlust"
                }
            }
        }
        stage("OWASP Dependency Check"){
            steps{
                dependencyCheck additionalArguments: '--scan ./', odcInstallation: 'dc'
                dependencyCheckPublisher pattern: '**/dependency-check-report.xml'
            }
        }
        stage("Sonar Quality Gate Scan "){
            steps{
                timeout(time: 2, unit: "MINUTES"){
                    waitForQualityGate abortPipeline: false
                }
            }
        }
        stage("Trivy File System Scan"){
            steps{
                sh "trivy fs --format table -o trivy-fs-report.html ."
            }
        }
        stage("Deploy using Docker compose"){
            steps{
                sh "docker-compose up -d"
            }
        }
    }
}
```



(no go to your instance security group add port 5000 where my application running)

Go to browser serach http://ec2public-ip:5173 and see your application is running