CICD DEVSECOPS PROJECT

OVERVIEW OF THE PROJECT

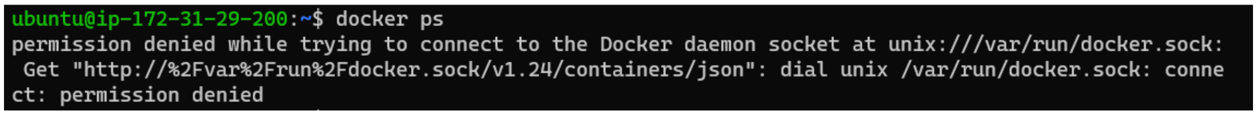
The code hosted on GitHub will undergo a quality check with SonarQube and dependency verification using OWASP's tools. For security scanning of Docker images, Trivy will be utilized. The deployment of the Docker container and the automation of these processes will be managed through Jenkins.

Code should be sent from Jenkins to SonarQube for analysis and the analyzed code should be received by Jenkins . This can be done through webhook token for authentication (Integrating Jenkins and SonarQube).
OWASP has libraries/packages and Jenkins requires a dependency check which can be fully filled by OWASP from Jenkins Tools Section.

AWS config setup:
Ubuntu OS, Choosed t2.large for instance type as t2.micro becomes an issue for memory for deploying and configuring the tools, key-pair, allowing SSH and configure the inbound rules accordingly

INSTALLATION OF Docker and Docker Compose

sudo apt-get update
sudo apt-get install docker.io -y
sudo apt-get install docker-compose -y

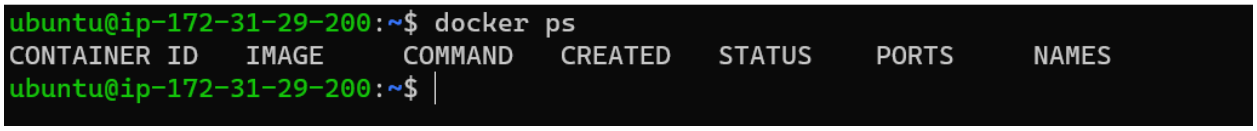


```
ubuntu@ip-172-31-29-200:~$ docker ps
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock:
 Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/json": dial unix /var/run/docker.sock: conne
ct: permission denied
```

We can resolve this error by :

sudo usermod -aG docker $USER   ## it means that we modified user settings using usermod, and append the user to a specified group docker
sudo reboot

Now the command is displaying the required O/p



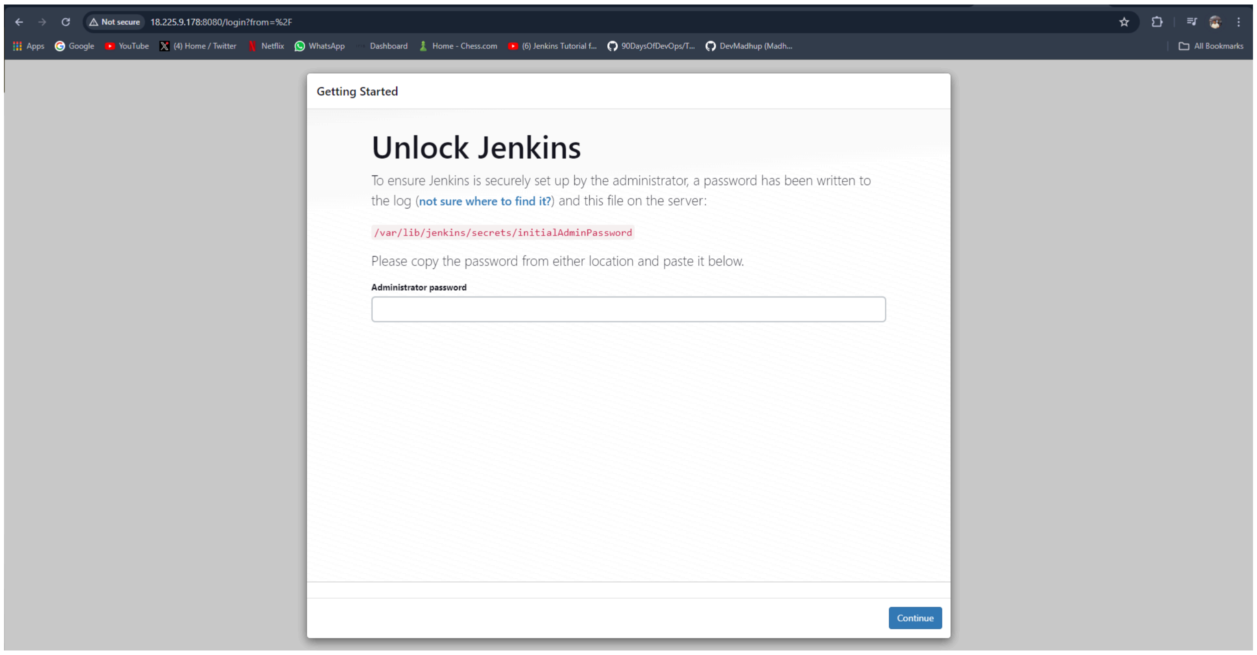
```
ubuntu@ip-172-31-29-200:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
ubuntu@ip-172-31-29-200:~$ 
```

INSTALLATION OF JENKINS

Installing Java : sudo apt install fontconfig openjdk-17-jre

Installing Jenkins (LTS) : https://www.jenkins.io/doc/book/installing/linux/ and jenkins run on port 8080, configure the inbound rules for the same in AWS→EC2→SG

JENKINS SETUP


Not secure 18.225.9.178:8080/login?from=%2F

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

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

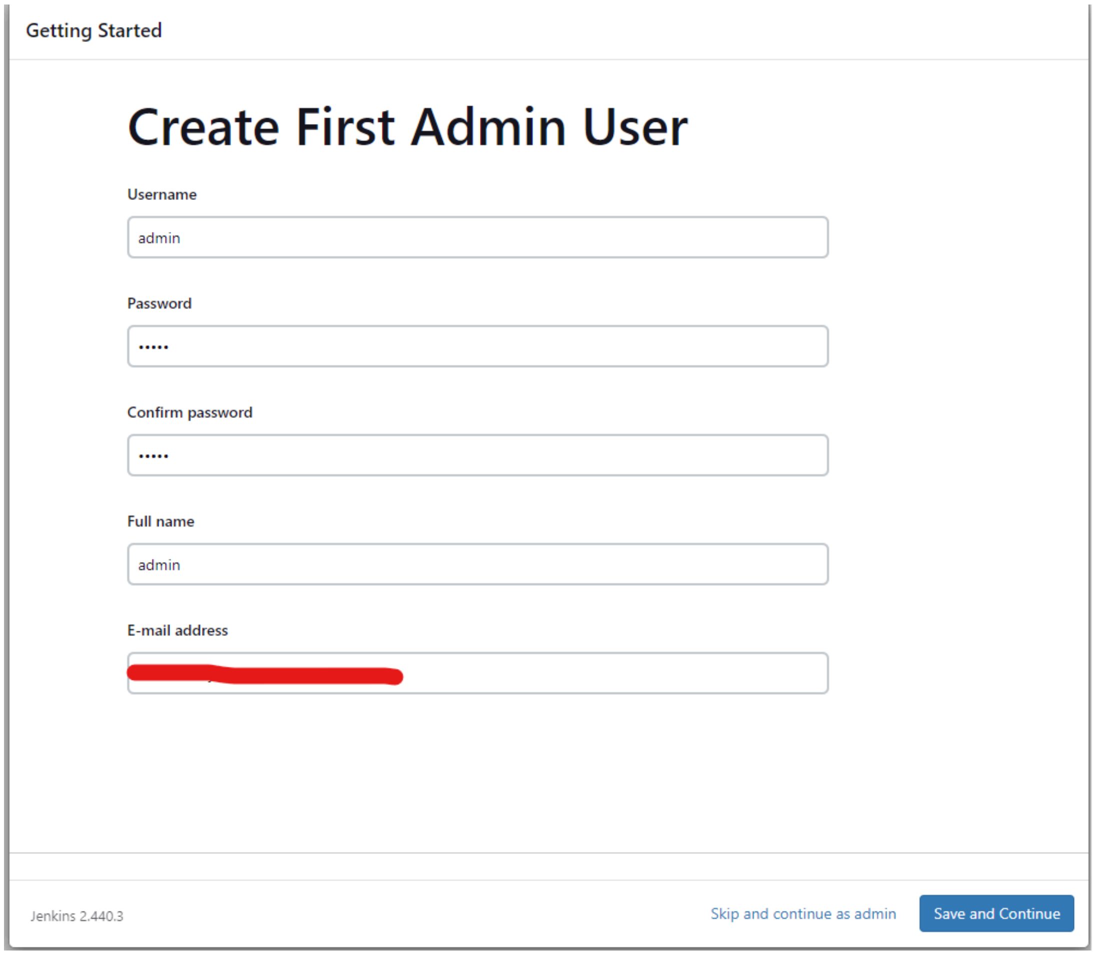
Please copy the password from either location and paste it below.

Administrator password

Retrieve the admin password from /var/lib/jenkins/secrets/initialAdminPassword
Command : sudo cat /var/lib/jenkins/secrets/initialAdminPassword
Password : 46fde6d6886248f9ba9e5f6caf41b188
Password(latest) : 7b002cbc5e224cafb52a49c4797584c8

Install the required Plugins and create a Admin user (First)

Jenkins URL : http://3.14.153.204:8080/
Jenkins URL(latest) : http://18.217.74.67:8080/


Getting Started

Create First Admin User

Username: admin

Password: •••••

Confirm password: •••••

Full name: admin

E-mail address: [redacted]

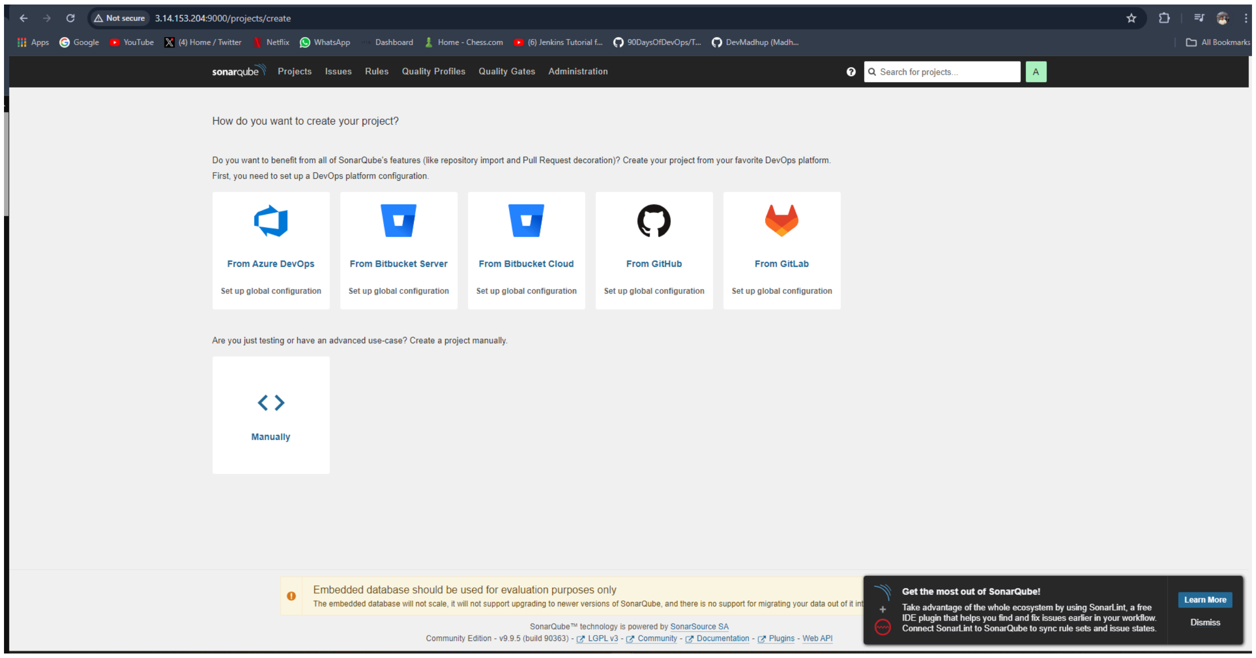
Jenkins 2.440.3

Installation of SonarQube

Install SonarQube from docker image
Command :docker run -itd --name sonarqube-server -p 9000:9000 sonarqube:lts-community

After that, open the URL which is public_ip:9000 in a webpage and we can see that SonarQube has been deployed and running on port 9000.
Initial username and password is admin for SonarQube
New-password : test@123

Edit inbound rule for 9000 for running SonarQube


Not secure 3.14.153.204:9000/projects/create
Apps Google YouTube (4) Home / Twitter Netflix WhatsApp Dashboard Home - Chess.com (6) Jenkins Tutorial f... 90DaysOfDevOps/T... DevMadhup (Madh... All Bookmarks
sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration
Search for projects...
A

How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform.
First, you need to set up a DevOps platform configuration.

From Azure DevOps	From Bitbucket Server	From Bitbucket Cloud	From GitHub	From GitLab
Set up global configuration	Set up global configuration	Set up global configuration	Set up global configuration	Set up global configuration

Are you just testing or have an advanced use-case? Create a project manually.

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 int

SonarQube™ technology is powered by SonarSource SA
Community Edition - v9.9.5 (build 90363) - LGPL v3 - Community - Documentation - Plugins - Web API

Get the most out of SonarQube!
Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow. Connect SonarLint to SonarQube to sync rule sets and issue states.

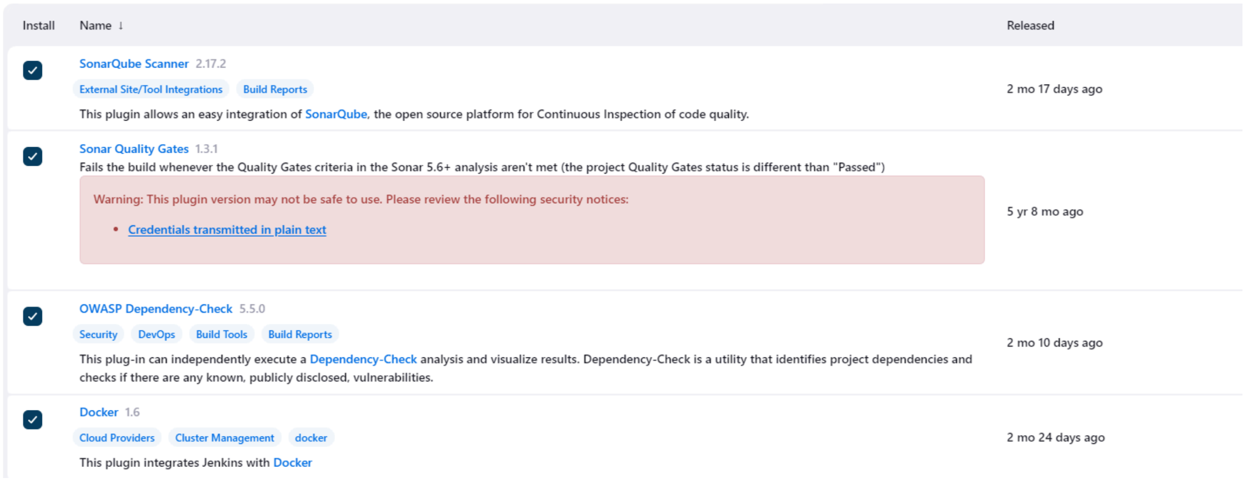
INSTALLATION OF TRIVY

Installation of Trivy: https://aquasecurity.github.io/trivy/v0.18.3/installation/ (ubuntu)

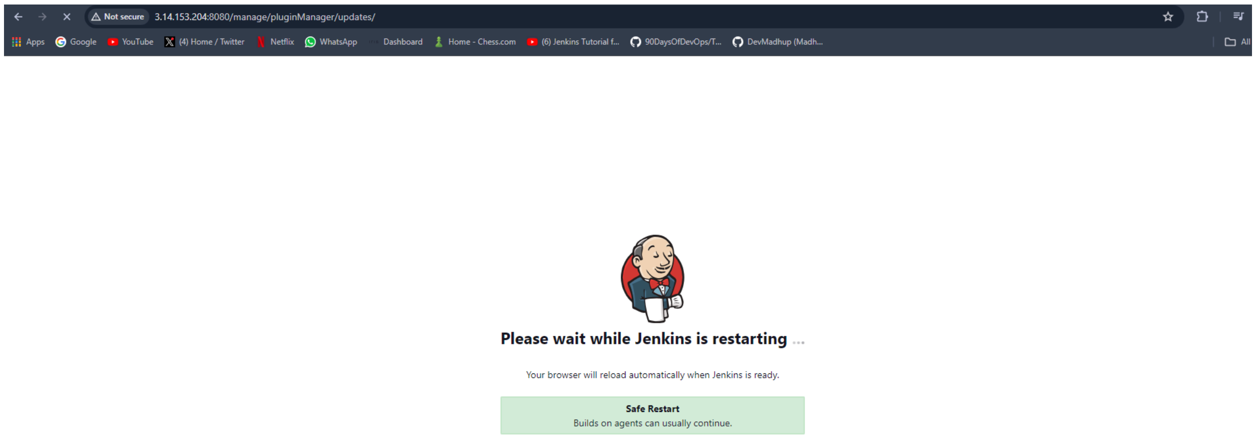
Jenkins Plugins Installations

Dashboard → Manage Jenkins → Plugins → Available Plugins

Opt for Restart Jenkins when installation is complete and no jobs are running and jenkins will automatically restart once the downloading of the plugins are completed and login with username and password



Install	Name ↓	Released
✓	SonarQube Scanner 2.17.2 External Site/Tool Integrations · Build Reports This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.	2 mo 17 days ago
✓	Sonar Quality Gates 1.3.1 Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed") Warning: This plugin version may not be safe to use. Please review the following security notices: • Credentials transmitted in plain text	5 yr 8 mo ago
✓	OWASP Dependency-Check 5.5.0 Security · DevOps · Build Tools · Build Reports 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.	2 mo 10 days ago
✓	Docker 1.6 Cloud Providers · Cluster Management · docker This plugin integrates Jenkins with Docker	2 mo 24 days ago



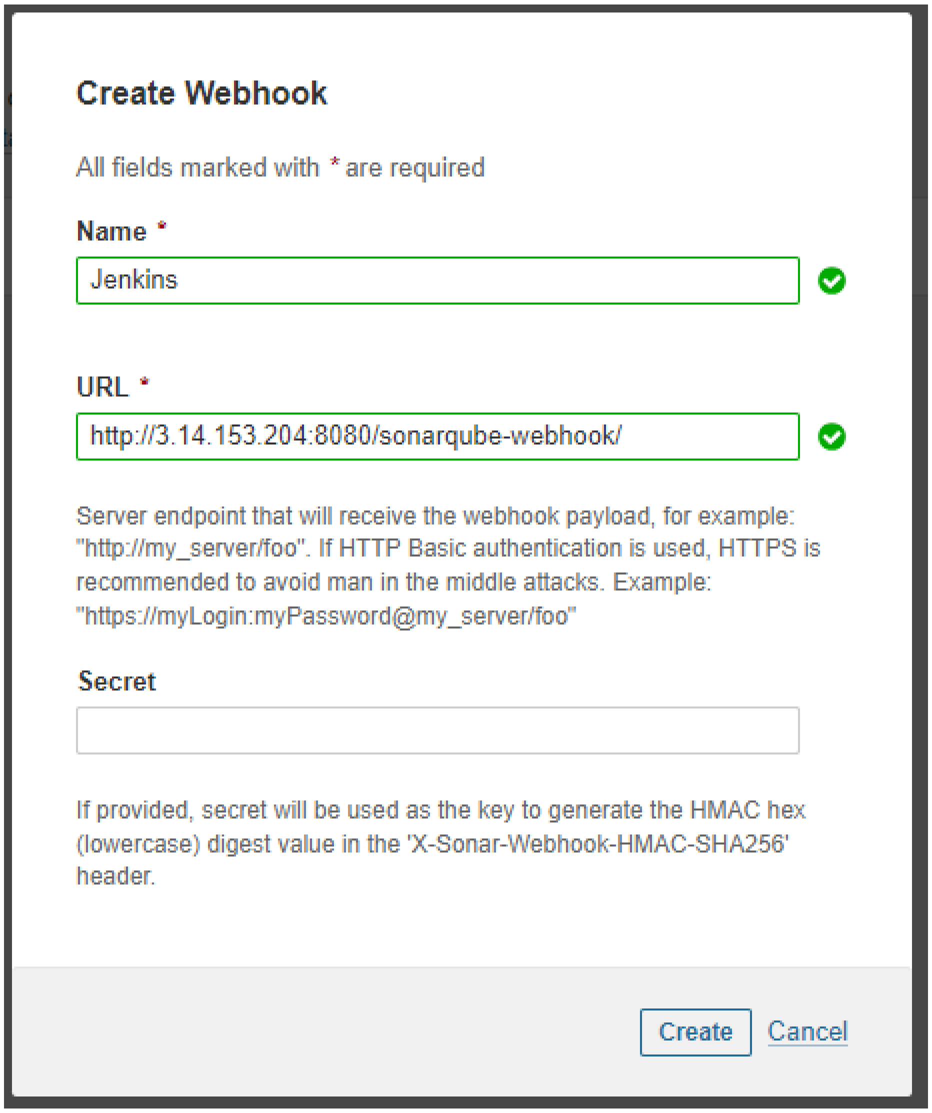
Please wait while Jenkins is restarting ...

Your browser will reload automatically when Jenkins is ready.

Safe Restart
Builds on agents can usually continue.

Now we need to integrated JENKINS and SonarQube

First we create a webhook from SonarQube (Administration → Configuration → Webhooks)



Create Webhook

All fields marked with * are required

Name *

Jenkins

URL *

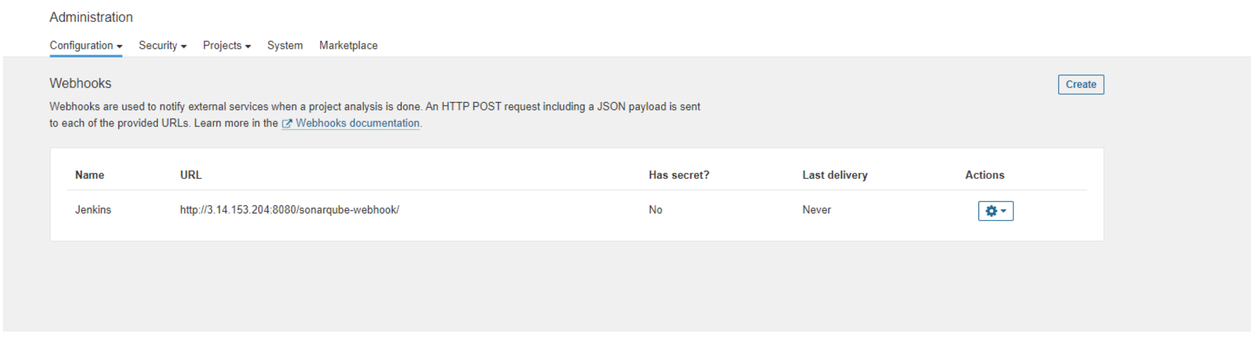
http://3.14.153.204:8080/sonarqube-webhook/

Server endpoint that will receive the webhook payload, for example: "http://my_server/foo". If HTTP Basic authentication is used, HTTPS is recommended to avoid man in the middle attacks. Example: "https://myLogin:myPassword@my_server/foo"

Secret

If provided, secret will be used as the key to generate the HMAC hex (lowercase) digest value in the 'X-Sonar-Webhook-HMAC-SHA256' header.

The URL field path should be Jenkins-URL/sonarqube-webhook/
(For Authentication and integration purpose)

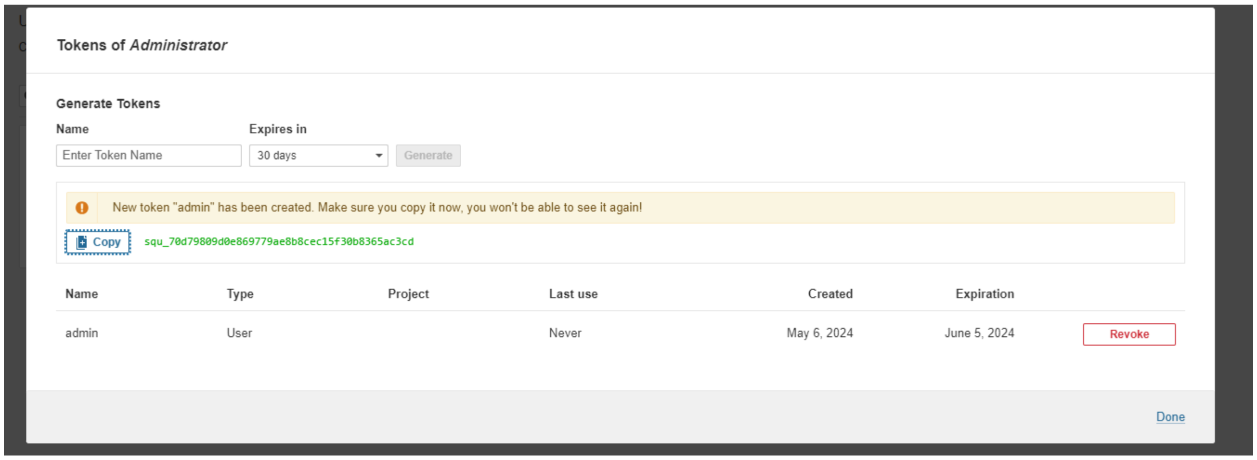


Webhooks

Webhooks are used to notify external services when a project analysis is done. An HTTP POST request including a JSON payload is sent to each of the provided URLs. Learn more in the Webhooks documentation.

Name	URL	Has secret?	Last delivery	Actions
Jenkins	http://3.14.153.204:8080/sonarqube-webhook/	No	Never	⚙

Now we create a token (Administration → Security → Users)
“Copy the token value and store it”


Tokens of *Administrator*

Generate Tokens

Name	Expires in	
Enter Token Name	30 days	Generate

New token "admin" has been created. Make sure you copy it now, you won't be able to see it again!

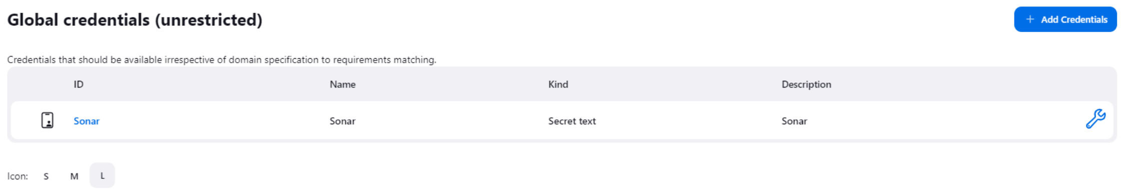
Copy `squ_70d79809d0e869779ae8b8cec15f30b8365ac3cd`

Name	Type	Project	Last use	Created	Expiration	
admin	User		Never	May 6, 2024	June 5, 2024	Revoke

Done

In Jenkins ,
Manage Jenkins → Security → Credentials

Add Credentials of “Kind” = Secret text , Secret = token (squ_70d79809d0e869779ae8b8cec15f30b8365ac3cd),   ID and Description is Sonar
Latest token = squ_c4e959626c0b2aed5c748e1a7df7fe6092f9f254


Global credentials (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

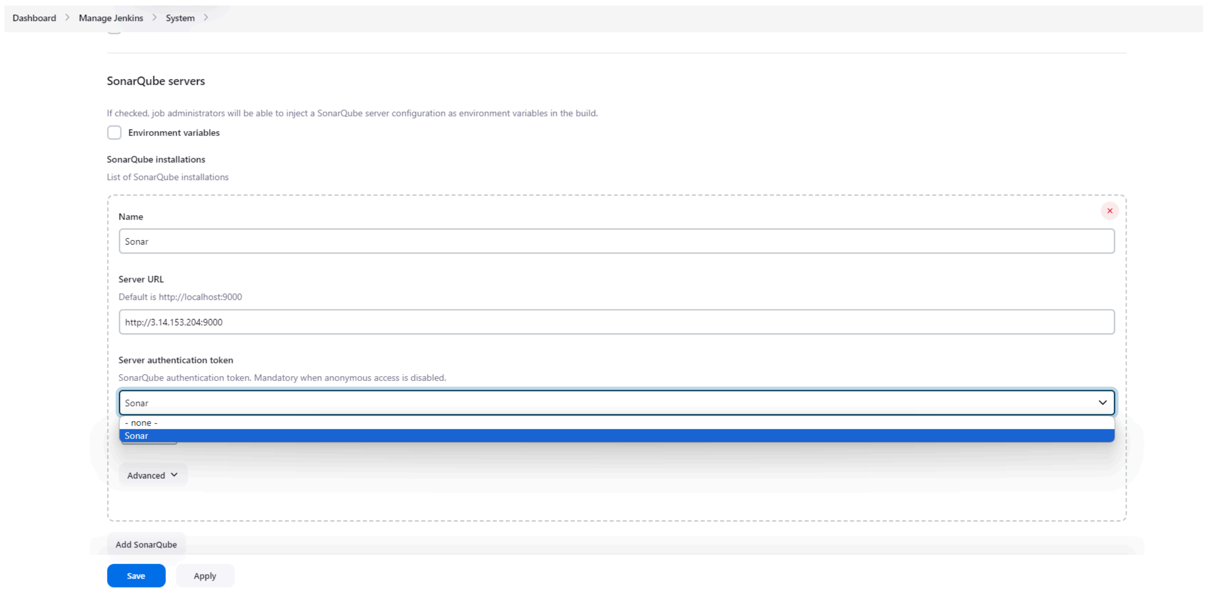
ID	Name	Kind	Description
Sonar	Sonar	Secret text	Sonar

Icon: S M L

.

Now the Server Authentication token for SOnarQube is enabled/imported and a drop down option of “Sonar” can be selected. (For Authentication and integration purpose)

In JenKins
Manage Jenkins → System


Dashboard > Manage Jenkins > System >

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

Environment variables

SonarQube installations

List of SonarQube installations

Name: Sonar

Server URL

Default is http://localhost:9000

http://3.14.153.204:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

Sonar

- none -
Sonar

Advanced

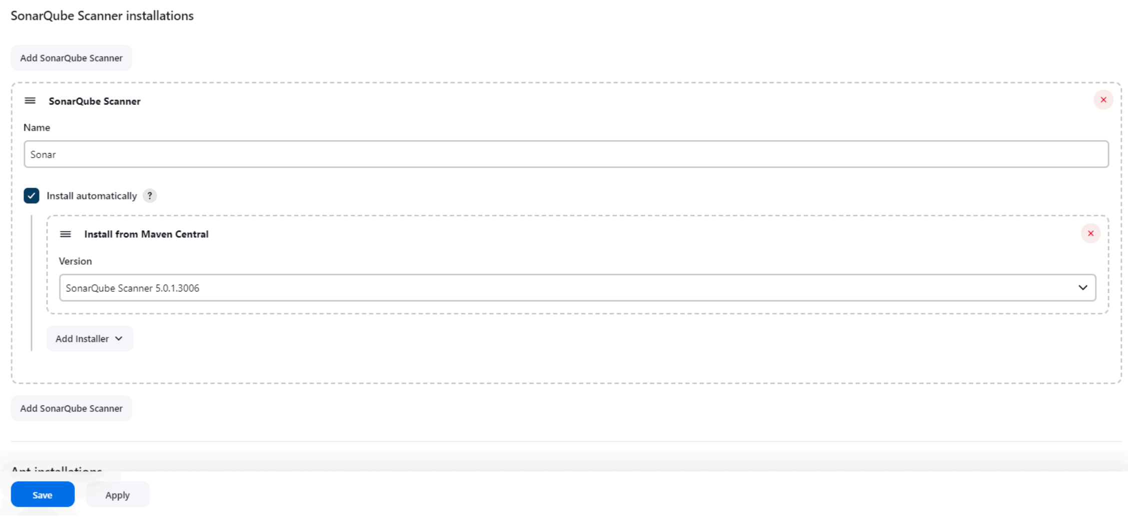
Add SonarQube

Save Apply

Now both Jenkins and SonarQube is been integrated

Setup SonarQube Scanner :

Manage Jenkins → Tools


SonarQube Scanner installations

Add SonarQube Scanner

SonarQube Scanner

Name: Sonar

☑ Install automatically

Install from Maven Central

Version: SonarQube Scanner 5.0.1.3006

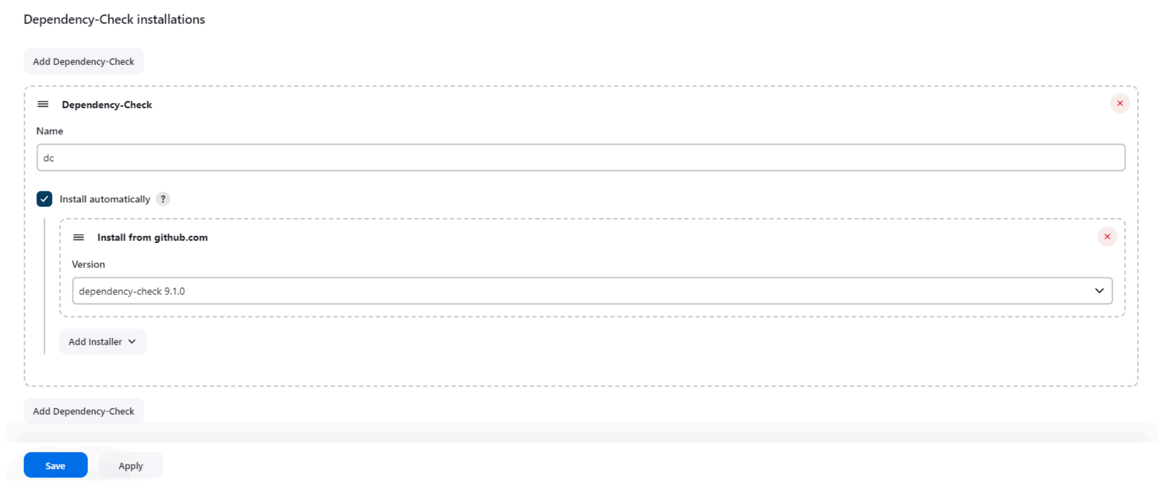
Add Installer

Add SonarQube Scanner

Save Apply

Setup Dependency Check

Manage Jenkins → Tools



Dependency-Check installations

Add Dependency-Check

Dependency-Check

Name: dc

☑ Install automatically

Install from github.com

Version: dependency-check 9.1.0

Add Installer

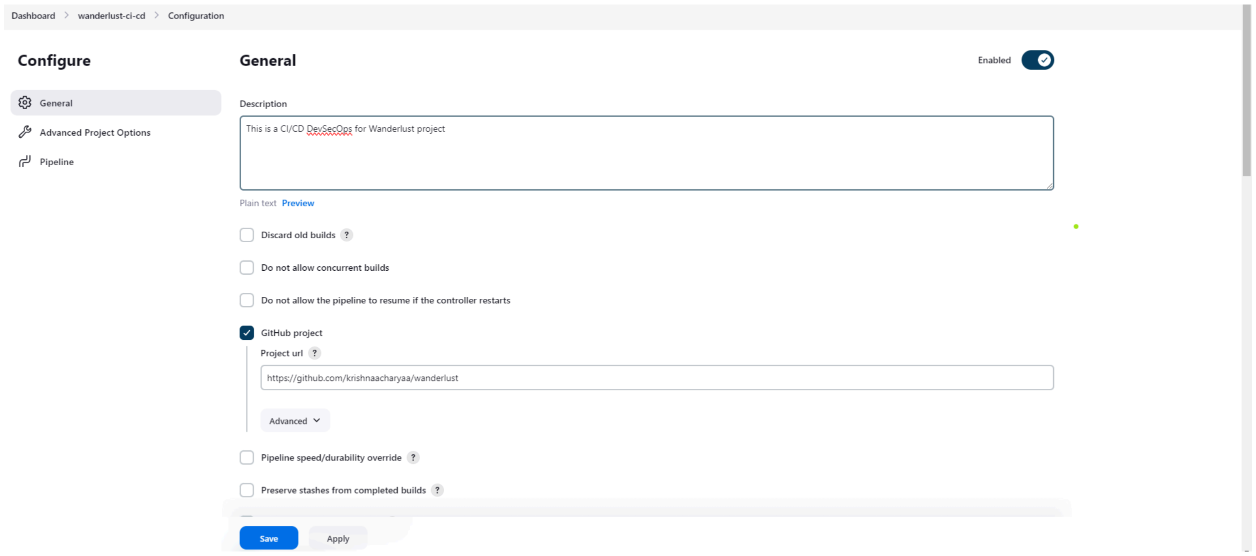
Add Dependency-Check

Save Apply

In Add Installer choose github

CREATING A PIPELINE :

+new item → wanderlust-ci-cd(give a name and choose pipeline) → configuration



Dashboard > wanderlust-ci-cd > Configuration

Configure

General

Advanced Project Options

Pipeline

General — Enabled

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

Plain text Preview

☐ Discard old builds

☐ Do not allow concurrent builds

☐ Do not allow the pipeline to resume if the controller restarts

☑ GitHub project

Project url: https://github.com/krishnaacharyaa/wanderlust

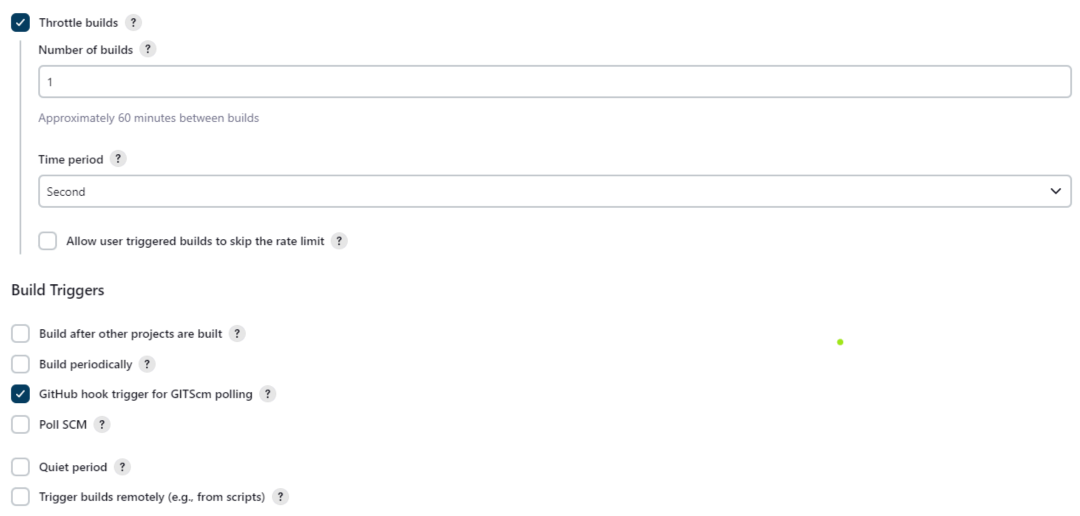
Advanced

☐ Pipeline speed/durability override

☐ Preserve stashes from completed builds

Save Apply

.



[✓] Throttle builds ?

Number of builds ?

1

Approximately 60 minutes between builds

Time period ?

Second

[ ] Allow user triggered builds to skip the rate limit ?

Build Triggers

[ ] Build after other projects are built ?

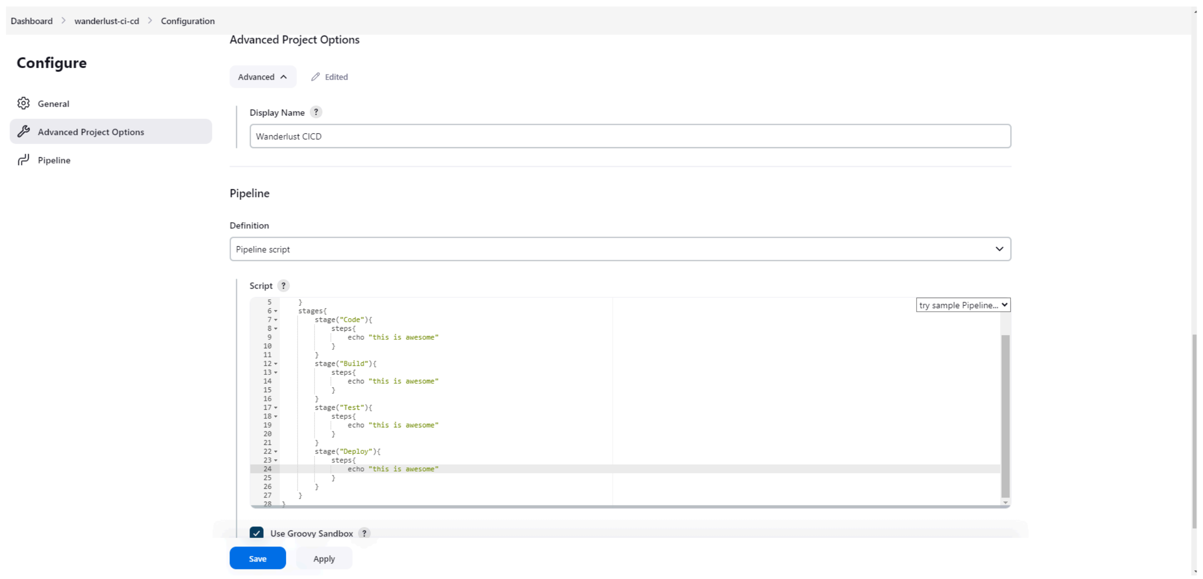
[ ] Build periodically ?

[✓] GitHub hook trigger for GITScm polling ?

[ ] Poll SCM ?

[ ] Quiet period ?

[ ] Trigger builds remotely (e.g., from scripts) ?



Advanced Project Options

Advanced ^ Edited

Display Name ?

Wanderlust CICD

Pipeline

Definition

Pipeline script

Script ?

```
    }
    stages{
        stage("Code"){
            steps{
                echo "this is awesome"
            }
        }
        stage("Build"){
            steps{
                echo "this is awesome"
            }
        }
        stage("Test"){
            steps{
                echo "this is awesome"
            }
        }
        stage("Deploy"){
            steps{
                echo "this is awesome"
            }
        }
    }
}
```

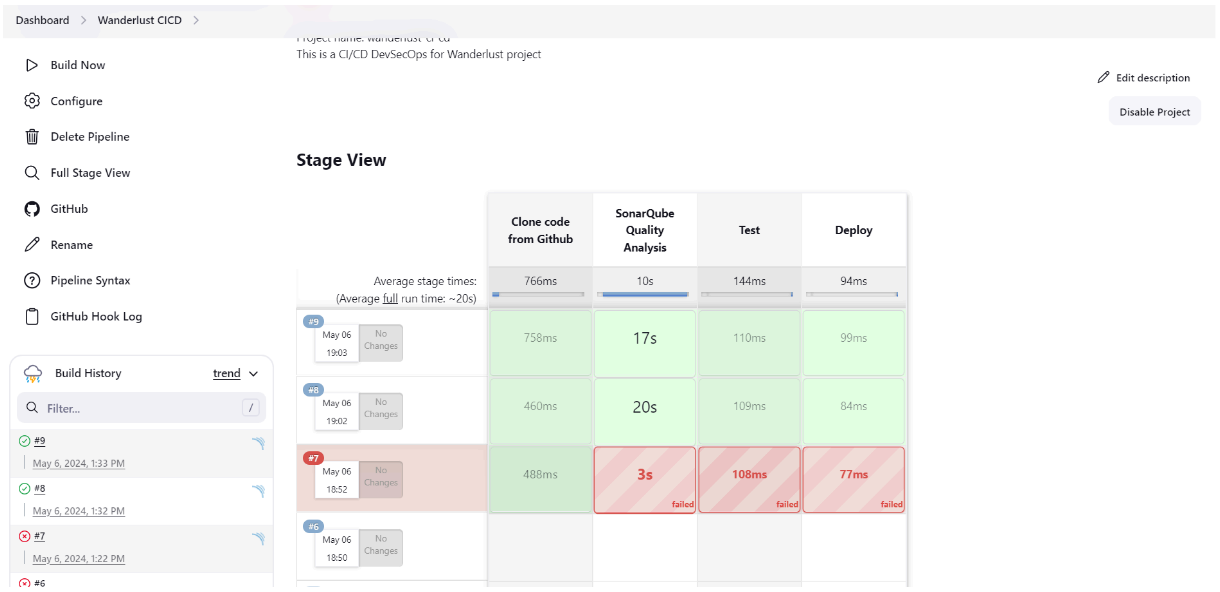
[✓] Use Groovy Sandbox ?

Pipeline script :

```
pipeline{
    agent any
    environment{
        SONAR_HOME= tool "Sonar"
    }
    stages{
        stage("Code"){
            steps{
                echo "this is awesome"
            }
        }
        stage("Build"){
            steps{
                echo "this is awesome"
            }
        }
        stage("Test"){
            steps{
                echo "this is awesome"
            }
        }
        stage("Deploy"){
            steps{
                echo "this is awesome"
            }
        }
    }
}
```

Testing the pipeline script for Code and Test(sonarqube)

```
pipeline{
    agent any
    environment{
        SONAR_HOME= tool "Sonar"
    }
    stages{
        stage("Clone code from Github"){
            steps{
                git url: "https://github.com/krishnaacharyaa/wanderlust.git/", branch: "devops"
            }
        }
        stage("SonarQube Quality Analysis"){
            steps{
                withSonarQubeEnv("Sonar"){
                    sh "$SONAR_HOME/bin/sonar-scanner -Dsonar.projectName=wanderlust
-Dsonar.projectKey=wanderlust"
                }
            }
        }
        stage("Test"){
            steps{
                echo "this is awesome"
            }
        }
        stage("Deploy"){
            steps{
                echo "this is awesome"
            }
        }
    }
}
```



Project name: wanderlust-ci-cd
This is a CI/CD DevSecOps for Wanderlust project

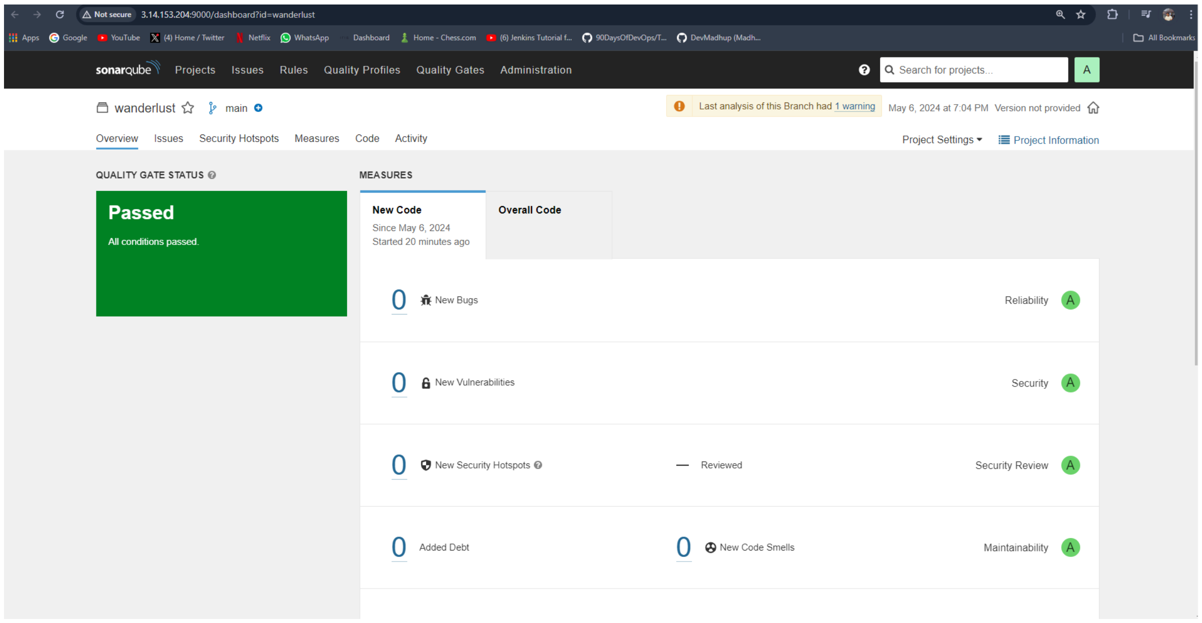
	Clone code from Github	SonarQube Quality Analysis	Test	Deploy
Average stage times: (Average full run time: ~20s)	766ms	10s	144ms	94ms
#9 May 06 19:03 No Changes	758ms	17s	110ms	99ms
#8 May 06 19:02 No Changes	460ms	20s	109ms	84ms
#7 May 06 18:52 No Changes	488ms	3s failed	108ms failed	77ms failed
#6 May 06 18:50 No Changes				

#9 May 6, 2024, 1:33 PM
#8 May 6, 2024, 1:32 PM
#7 May 6, 2024, 1:22 PM
#6

Few error/s were received and it's was fixed by adding the github url path properly (good to give the github url including .git/ path and do mention the branch) and add the admin email field anywhere mentioned while setting up Jenkins
Keep the token generated from SonarQube handy, resolved an error again re-authenticating the token to jenkins in SonarQube fields and now the above image is good to go as all are running fine.

Error logs for SonarQube Quality Analysis block
ERROR: Error during SonarScanner execution ERROR: Not authorized. Please check the properties sonar.login and sonar.password. ERROR:

O/P of SonarQube: http://3.14.153.204:9000/dashboard?id=wanderlust



Last analysis of this Branch had 1 warning May 6, 2024 at 7:04 PM Version not provided

QUALITY GATE STATUS

Passed

All conditions passed.

MEASURES

New Code Since May 6, 2024 Started 20 minutes ago

Overall Code

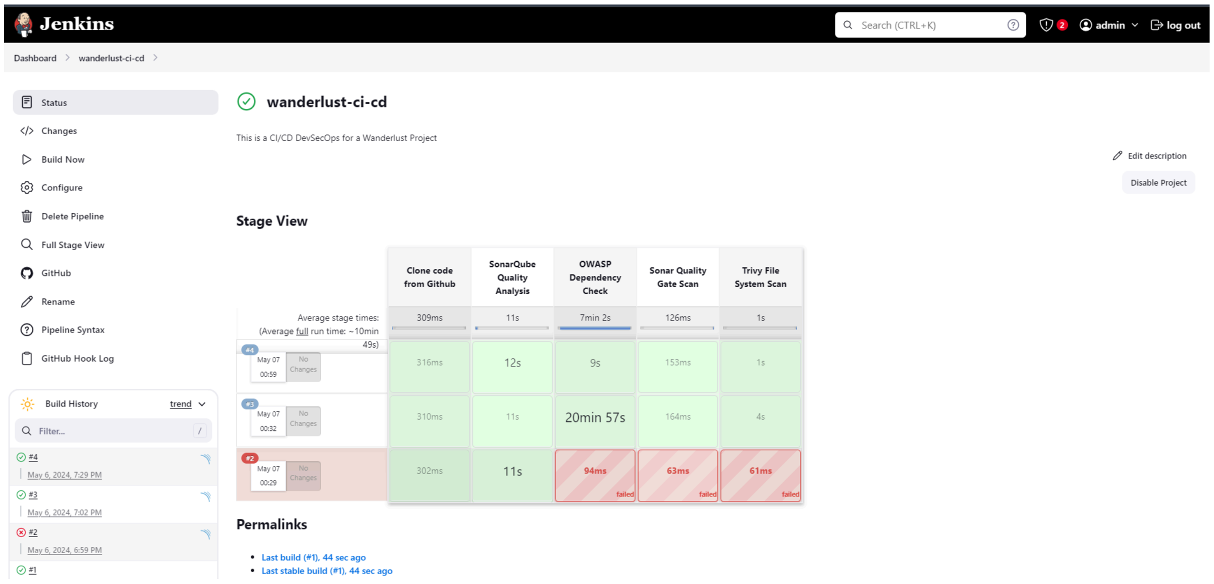
0 New Bugs — Reliability A

0 New Vulnerabilities — Security A

0 New Security Hotspots — Reviewed — Security Review A

0 Added Debt — 0 New Code Smells — Maintainability A

```
pipeline{
  agent any
  environment{
     SONAR_HOME= tool "Sonar"
  }
  stages{
     stage("Clone code from Github"){
        steps{
           git url: "https://github.com/krishnaacharyaa/wanderlust.git/", branch: "devops"
        }
     }
     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 ."
        }
     }
  }
}
```



wanderlust-ci-cd

This is a CI/CD DevSecOps for a Wanderlust Project

Stage View

	Clone code from Github	SonarQube Quality Analysis	OWASP Dependency Check	Sonar Quality Gate Scan	Trivy File System Scan
Average stage times: (Average full run time: ~10min 49s)	309ms	11s	7min 2s	126ms	1s
#4 May 07 00:59 No Changes	316ms	12s	9s	153ms	1s
#3 May 07 00:32 No Changes	310ms	11s	20min 57s	164ms	4s
#2 May 07 00:29 No Changes	302ms	11s	94ms failed	63ms failed	61ms failed

Build History

- #4 May 6, 2024, 7:29 PM
- #3 May 6, 2024, 7:02 PM
- #2 May 6, 2024, 6:59 PM
- #1

Permalinks

- Last build (#1), 44 sec ago
- Last stable build (#1), 44 sec ago

Now we need to run the app

```
 pipeline{
    agent any
    environment{
        SONAR_HOME= tool "Sonar"
    }
    stages{
        stage("Clone code from Github"){
            steps{
                git url: "https://github.com/krishnaacharyaa/wanderlust.git/", branch: "devops"
            }
        }
        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"
            }
        }
    }
}
```

And add the following commands

```
sudo usermod -aG docker jenkins      # This commands add jenkins user to the docker group
```

sudo systemctl restart docker	# Restarts docker service as config changes where made
sudo systemctl restart jenkins	# Restarts Jenkins service as config changes where made

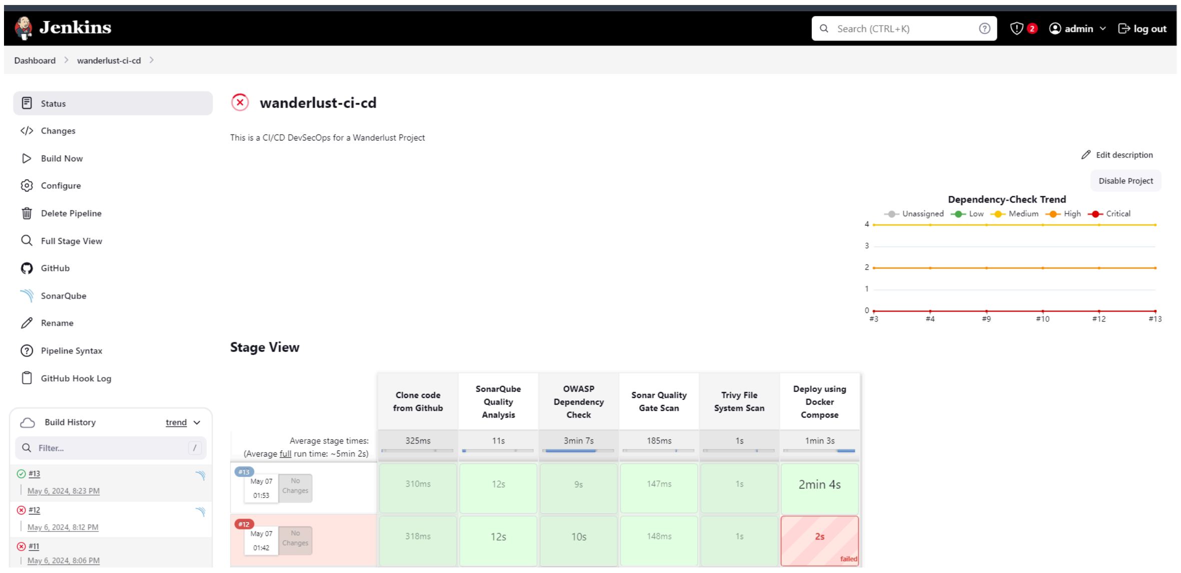
And encounter an error
File "/usr/lib/python3/dist-packages/docker/api/client.py", line 214, in _retrieve_server_version
  return self.version(api_version=False)["ApiVersion"]

Sol: we have to install latest docker compose version

So first we have to remove the docker compose installed by
sudo apt-get remove docker-compose

Now install latest docker compose version(v2) by
curl -SL https://github.com/docker/compose/releases/download/v2.27.0/docker-compose-linux-x86_64 -o /usr/local/bin/docker-compose

The above command downloads the docker compose in path /usr/local/bin directory
Execute permission : sudo chmod +x /usr/local/bin/docker-compose



wanderlust-ci-cd

This is a CI/CD DevSecOps for a Wanderlust Project

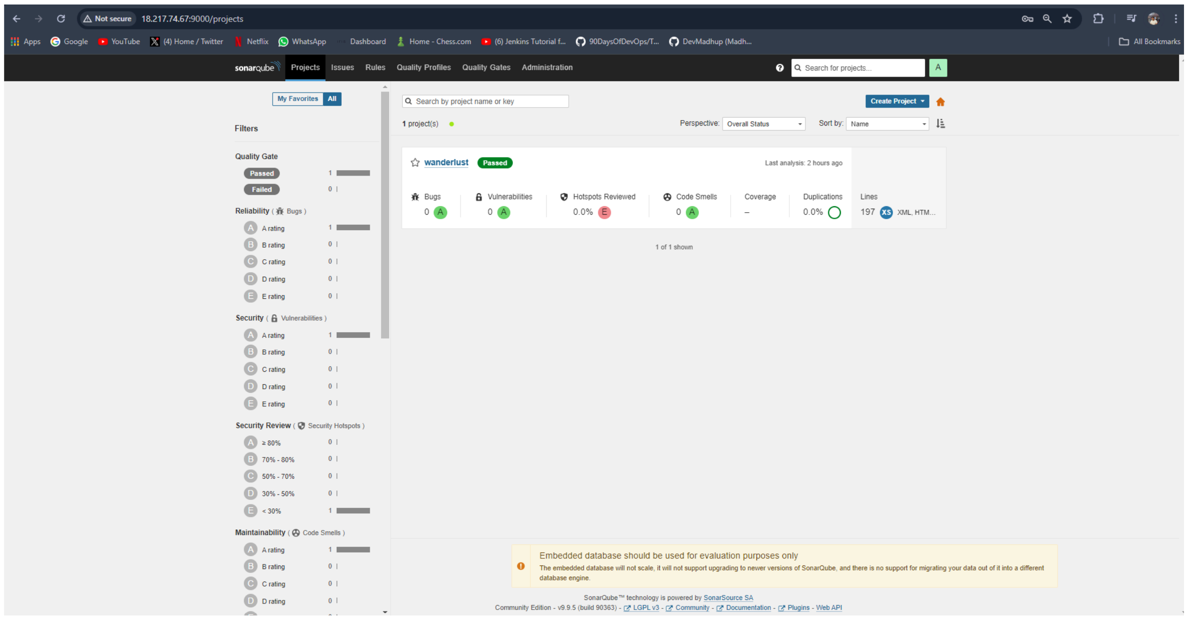
#13 May 6, 2024, 8:23 PM
#12 May 6, 2024, 8:12 PM
#11 May 6, 2024, 8:06 PM

Stage View

	Clone code from Github	SonarQube Quality Analysis	OWASP Dependency Check	Sonar Quality Gate Scan	Trivy File System Scan	Deploy using Docker Compose
Average stage times: (Average full run time: ~5min 2s)	325ms	11s	3min 7s	185ms	1s	1min 3s
#13 May 07 01:53 No Changes	310ms	12s	9s	147ms	1s	2min 4s
#12 May 07 01:42 No Changes	318ms	12s	10s	148ms	1s	2s failed

Finally all the blocks/stages of the pipeline are running without any error.

SonarQube code checks


18.217.74.67:9000/projects

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
0 A	0 A	0.0% E	0 A	–	0.0%	197 XS XML, HTM...

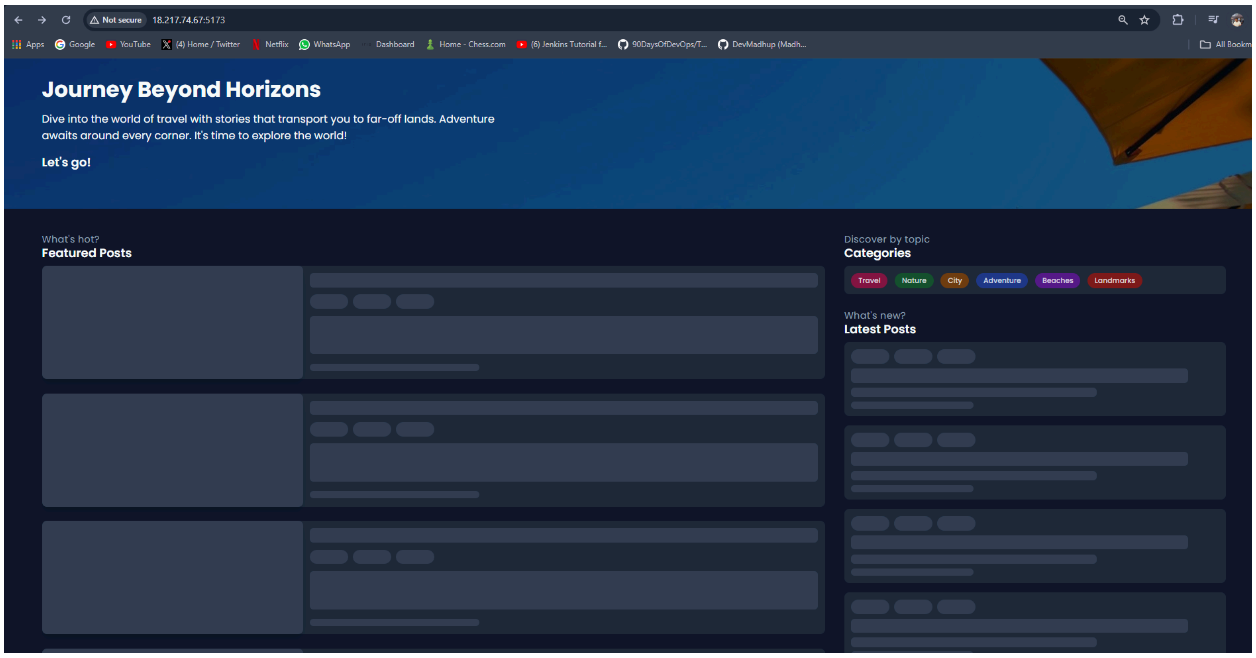
wanderlust Passed — Last analysis: 2 hours ago

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 - v9.9.5 (build 90363) - LGPL v3 - Community - Documentation - Plugins - Web API

The backend runs on the port 5173 and frontend runs on 5000, configured the inbound rules for the same.

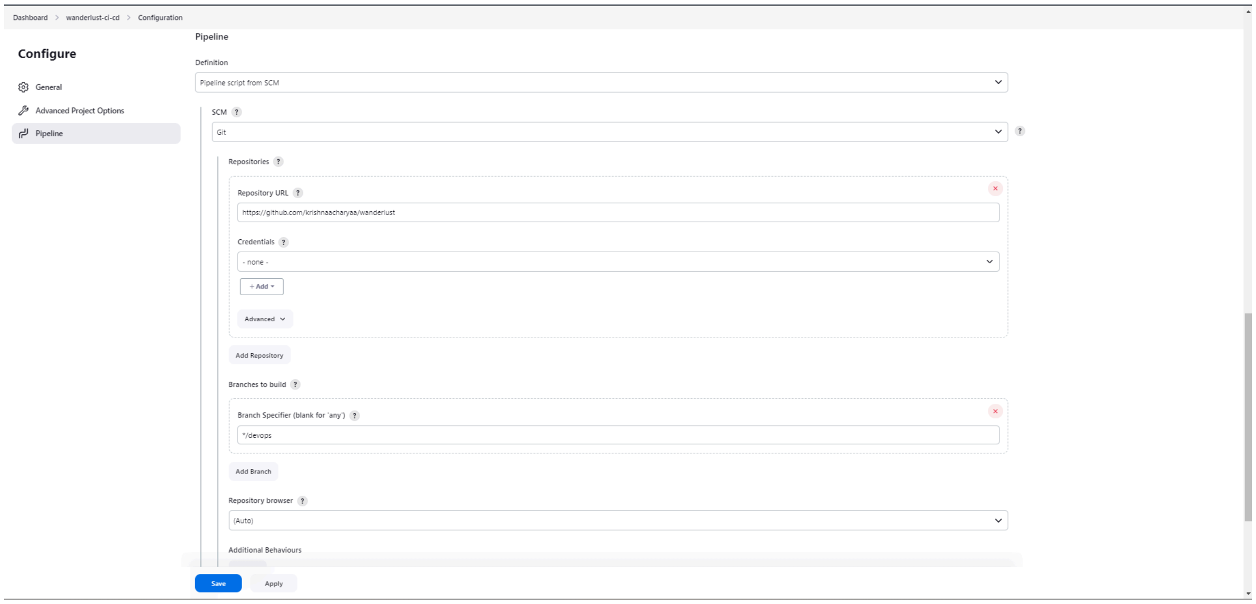

18.217.74.67:5173

Journey Beyond Horizons

Dive into the world of travel with stories that transport you to far-off lands. Adventure awaits around every corner. It's time to explore the world!

Let's go!

Pick the source code from SCM by the following for declarative pipeline



Dashboard > wanderlust-ci-cd > Configuration

Configure

- General
- Advanced Project Options
- Pipeline

Pipeline

Definition: Pipeline script from SCM

SCM: Git

Repositories

Repository URL: https://github.com/krishnaacharyaa/wanderlust

Credentials: - none -

+ Add

Advanced

Add Repository

Branches to build

Branch Specifier (blank for 'any'): */devops

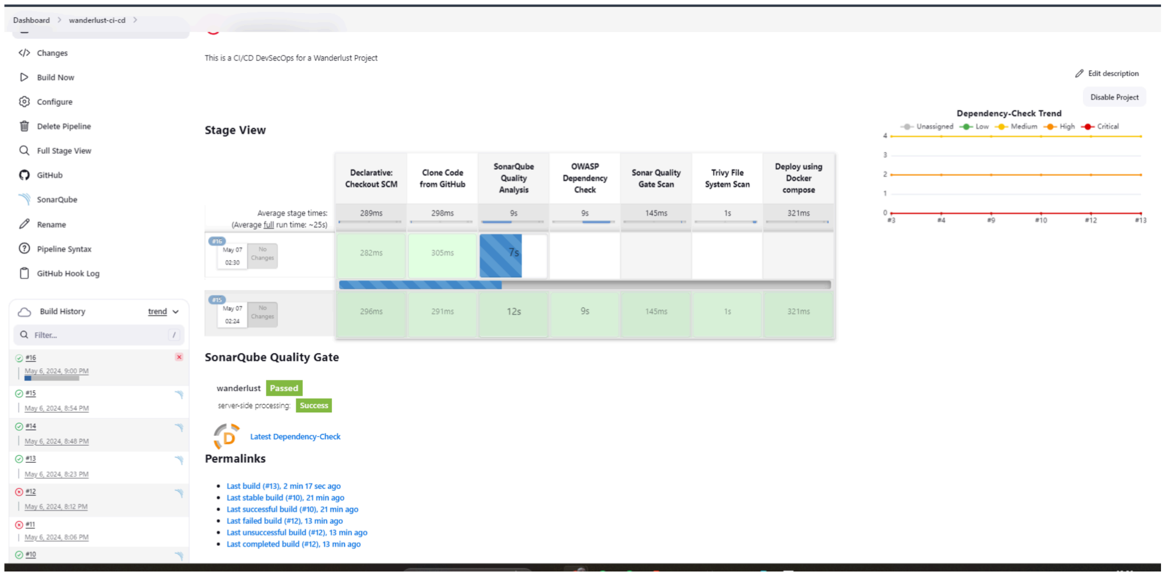
Add Branch

Repository browser: (Auto)

Additional Behaviours

Save Apply

After checking the pipeline, O/P below



Dashboard > wanderlust-ci-cd >

- Changes
- Build Now
- Configure
- Delete Pipeline
- Full Stage View
- GitHub
- SonarQube
- Rename
- Pipeline Syntax
- GitHub Hook Log

This is a CI/CD DevSecOps for a Wanderlust Project

Edit description

Disable Project

Stage View

	Declarative: Checkout SCM	Clone Code from GitHub	SonarQube Quality Analysis	OWASP Dependency Check	Sonar Quality Gate Scan	Trivy File System Scan	Deploy using Docker compose
Average stage times: (Average full run time: ~25s)	289ms	298ms	9s	9s	145ms	1s	321ms
#16 May 07 02:30 No Changes	282ms	305ms	7s				
#15 May 07 02:24 No Changes	296ms	291ms	12s	9s	145ms	1s	321ms

Build History trend

Filter...

- #16 May 6, 2024, 9:00 PM
- #15 May 6, 2024, 8:54 PM
- #14 May 6, 2024, 8:48 PM
- #13 May 6, 2024, 8:23 PM
- #12 May 6, 2024, 8:12 PM
- #11 May 6, 2024, 8:06 PM
- #10

SonarQube Quality Gate

wanderlust Passed

server-side processing: Success

Latest Dependency-Check

Permalinks

- Last build (#13), 2 min 17 sec ago
- Last stable build (#10), 21 min ago
- Last successful build (#10), 21 min ago
- Last failed build (#12), 13 min ago
- Last unsuccessful build (#12), 13 min ago
- Last completed build (#12), 13 min ago

Now the pipeline is controlled by the filename Jenkinsfile (github) which is under the branch devops