

Pipeline Project: Code Analysis, Quality Gate, and AWS

Pipeline Project: Jenkins Pipeline, SonarCloud, and AWS

Problem Statement:

-A client is working on a project that will assist medical doctors and patients. When the doctors log in, they would see patients' information and when the patient logs in, they would see their record and appointments. The developers used JAVA language to write the code and used maven as the build tool to compile the source code. The client wants the project to be built automation using a pipeline method. He also wants a way to be able to check the quality of the code before sending the artifact to a cloud repository. Lastly, integrate a notification service to let someone know if a build fails or is complete.

Solution Statement:

- I am going to create a Jenkins server and implement the tools that'll use to automate the build and deployment of the project artifact to a cloud repository.

Key components:

- Linux Centos7 server: For the Jenkins server
- GitHub: To store the source code
- Git: To access and manage the Github and the Jenkins server
- AWS: For providing cloud services
- Amazon ECR: To store the final artifact
- AWS IAM user: To provide access between the Jenkins server and the ECR repository
- Docker: To build the dockerfile
- Maven Pipeline: To automate the build of the project
- Jenkinsfile: For the maven pipeline
- SonarCloud: For the code analysis and quality gate
- Slack channel: For the notification service
- more

Pushing the source to the working repository in GitHub

```
MINGW64:c/Users/andre/Desktop/Pro_am/Qualitygate_AWS
create mode 100644 trash/userauth.sql


andre@DESKTOP-FTEJMF0 MINGW64 ~/Desktop/Pro_am/Qualitygate_AWS (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

andre@DESKTOP-FTEJMF0 MINGW64 ~/Desktop/Pro_am/Qualitygate_AWS (main)
$ git push origin main
Enumerating objects: 2222, done.
Counting objects: 100% (2222/2222), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2195/2195), done.
Writing objects: 100% (2220/2220), 16.02 MiB | 412.00 KiB/s, done.
Total 2220 (delta 279), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (279/279), done.
To https://github.com/Niandre95/Qualitygate_AWS.git
   b38dbb0..885d487  main -> main


andre@DESKTOP-FTEJMF0 MINGW64 ~/Desktop/Pro_am/Qualitygate_AWS (main)
$
```









Checking to see if I successfully push the source code into the repository

 **Niandre95** / **Qualitygate_AWS** Public

<> Code Issues Pull requests Actions Projects Bookmarks

main 1 branch 0 tags

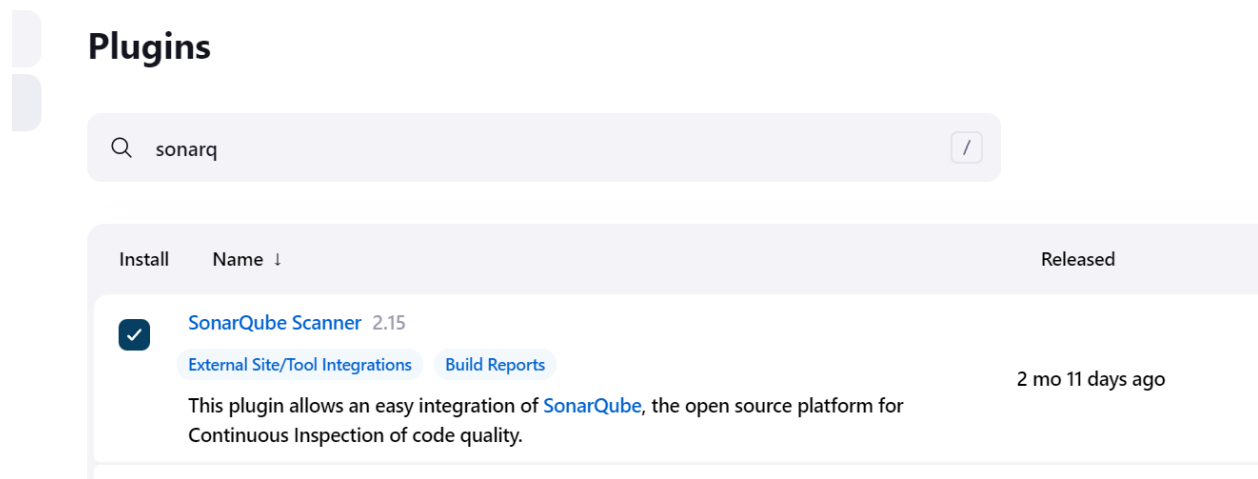
 **Niandre95** minor change

	src	code commit to github
	trash	code commit to github
	Dockerfile	code commit to github
	Jenkinsfile	minor change
	README.md	code commit to github
	mvnw	code commit to github
	mvnw.cmd	code commit to github
	pom.xml	code commit to github

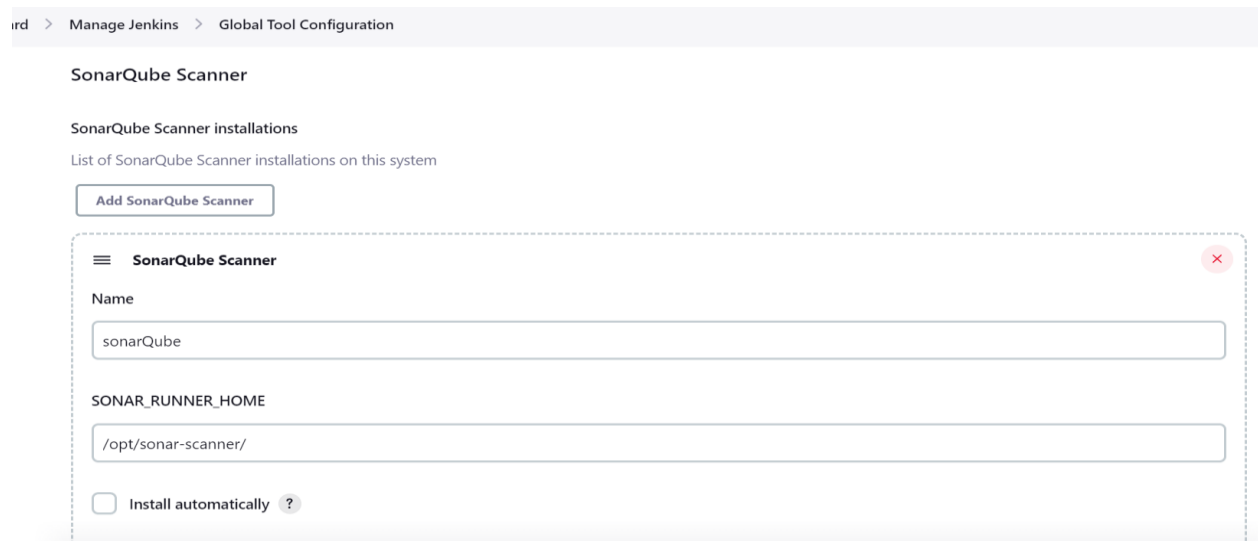
Checking to see if the sonar scanner is configured correctly on the Jenkins server

```
root@jenkinshost-server:~  
[vagrant@jenkinshost-server opt]$ sudo vim ~/.bash_profile  
[vagrant@jenkinshost-server opt]$ sudo -i  
[root@jenkinshost-server ~]# vim ~/.bash_profile  
[root@jenkinshost-server ~]# source ~/.bash_profile  
[root@jenkinshost-server ~]# echo $SONAR_RUNNER_HOME  
/opt/sonar-scanner  
[root@jenkinshost-server ~]#
```

Install SonarQube Scanner in the Jenkins server

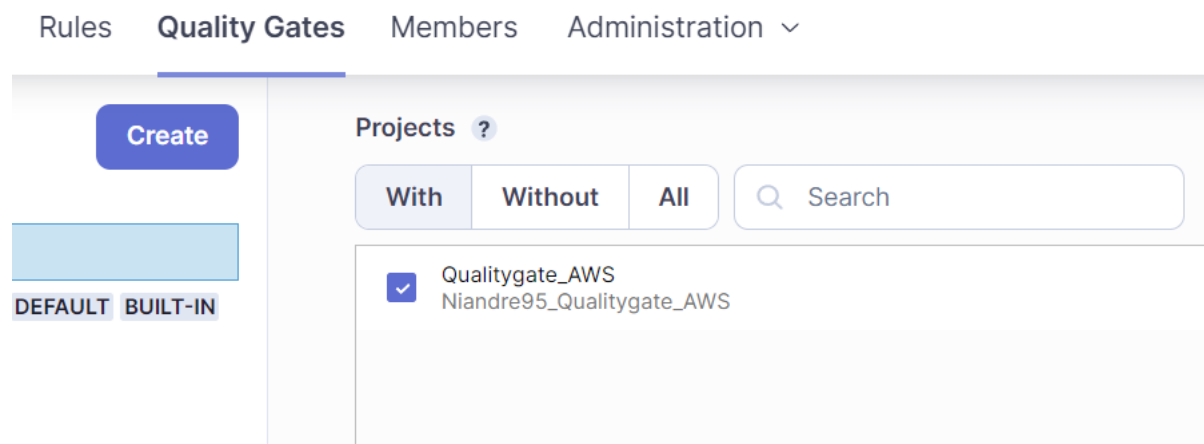


Setting up SonarQube in the global tool configuration



Installing Docker for the Pipeline

Setting up the quality gate for the project



Creating an IAM used to manage the connection between our Jenkins Server and the ECR repository that's going to store the Docker Image

Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details		
User name jenkins-ecr	Console password type None	Require password reset No

Permissions summary

< 1 >

Creating an ECR repository to store the Docker Image

Create repository

General settings

Visibility settings [Info](#)

Choose the visibility setting for the repository.

☒ **Private**

Access is managed by IAM and repository policy permissions.

☐ **Public**

Publicly visible and accessible for image pulls.

Repository name

Provide a concise name. A developer should be able to identify the repository contents by the name.

868016059835.dkr.ecr.us-east-1.amazonaws.com/

devop_repository


16 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, periods and forward slashes.

Installing a set of plugins in the Jenkins web interface to manage credentials and connect to ECR.

ECR		
Install	Name ↓	Released
<input checked="" type="checkbox"/>	CloudBees AWS Credentials 191.vcb_f183ce58b_9 aws Allows storing Amazon IAM credentials within the Jenkins Credentials API. Store Amazon IAM access keys (AWSAccessKeyId and AWSSecretKey) within the Jenkins Credentials API. Also support IAM Roles and IAM MFA Token. 11 mo ago	
<input type="checkbox"/>	Amazon Web Services SDK :: ECR 1.12.397-362.v050e9394cf8e Library plugins (for use by other plugins) aws ECR module for the AWS SDK for Java . 1 day 16 hr ago	
<input checked="" type="checkbox"/>	Amazon ECR 1.114.vfd22430621f5 aws This plugin generates Docker authentication token from Amazon Credentials to access 5 days 10 hr ago	


Creating the Pipeline job in the Jenkins web interface

» *Required field*




Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with something other than software build.




Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building organizing complex activities that do not easily fit in free-style job type.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on r etc

Adding the sonar cloud project properties in the pom.xml file

```

15         <description>Demo project for Spring Boot</description>
16         <packaging>jar</packaging>
17         <properties>
18             <java.version>1.8</java.version>
19             <sonar.organization>niandre95</sonar.organization>
20             <sonar.host.url>https://sonarcloud.io</sonar.host.url>
21         </properties>
22     
```

Creating a webhook for the project

Webhooks

[Create](#)

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	Secret	Last delivery
analysis-ci	http://jenkins:8080/sonarqube-webhook/	No	Never

Adding Jenkins to the slack channel



[Browse apps](#) > [Jenkins CI](#) > New configuration



Jenkins CI

An open-source continuous integration server.

Jenkins CI is a customisable continuous integration server with over 600 plugin

This integration will post build notifications to a channel in Slack.

Post to channel

Start by choosing a channel where Jenkins notifications will be posted.

[Add Jenkins CI integ](#)

Setting up the configuration system variable for Slack in Jenkins

Slack

Workspace ?

jenkinsjobsgroup

Credential ?

slack-token_

+ Add

Default channel / member id ?

#jenkinscid

Pipeline output

Progress (1): 49/56 kB

Progress (1): 56 kB

Downloaded from central: <https://repo.maven.apache.org/maven2/org/apache/commons/commons-parent/34/commons-parent-34.pom> (56 kB at 427 kB/s)

Downloading from central: <https://repo.maven.apache.org/maven2/commons-codec/commons-codec/1.11/commons-codec-1.11.pom>

Progress (1): 14 kB

Downloaded from central: <https://repo.maven.apache.org/maven2/commons-codec/commons-codec/1.11/commons-codec-1.11.pom> (14 kB at 62 kB/s)

Downloading from central:

<https://repo.maven.apache.org/maven2/org/apache/httpcomponents/httpcore/4.4.14/httpcore-4.4.14.pom>

Progress (1): 5.0 kB

Downloaded from central: <https://repo.maven.apache.org/maven2/org/apache/httpcomponents/httpcore/4.4.14/httpcore-4.4.14.pom> (5.0 kB at 171 kB/s)

Downloading from central: <https://repo.maven.apache.org/maven2/org/apache/httpcomponents/httpcomponents-core/4.4.14/httpcomponents-core-4.4.14.pom>

Progress (1): 13 kB

Downloaded from central: <https://repo.maven.apache.org/maven2/org/apache/httpcomponents/httpcomponents-core/4.4.14/httpcomponents-core-4.4.14.pom> (13 kB at 124 kB/s)

Downloading from central: <https://repo.maven.apache.org/maven2/org/apache/maven/doxia/doxia-integration-tools/1.11.1/doxia-integration-tools-1.11.1.pom>

```
Slack Notifications.  
[Pipeline] slackSend  
Slack Send Pipeline step running, values are - baseUrl: <empty>, teamDomain: jenkinsjobsgroup, channel:  
#jenkinsci, color: good, botUser: false, tokenCredentialId: slack-token, notifyCommitters: false, iconEmoji:  
<empty>, username: <empty>, timestamp: <empty>  
[Pipeline] }  
[Pipeline] // stage  
[Pipeline] }  
[Pipeline] // withEnv  
[Pipeline] }  
[Pipeline] // withEnv  
[Pipeline] }  
[Pipeline] // withEnv  
[Pipeline] }  
[Pipeline] // node  
[Pipeline] End of Pipeline  
Finished: SUCCESS
```



jenkins APP 18:21

SUCCESS: Job code_analysis-ci build 1

More info at: http://192.168.56.179:8080/job/code_analysis-ci/1/

B I ↺ 🔗 ⌵ ⌶ ⌷ ⌸

Stage output

🔍 Search description
Disable Project

Stage View

Average stage times:
(Average full run time: ~7min 34s)

	Declarative: Checkout SCM	Declarative: Tool Install	Checkout	Code Build	Test	Checkstyle Analysis	SonarQube Analysis	Quality Gate	Build Image	Deploy image	Declarative: Post Actions
#1 Feb 08 18:13 No Changes	16s	283ms	1s	1min 41s	20s	37s	1min 18s	4s	2min 0s	56s	554ms

Permalinks

ECR repository

Amazon ECR > Repositories > devop_repository

devop_repository

View push commandsEdit

Images (1)

RefreshDeleteDetailsScan

Search artifacts

< 1 > ⚙

<input type="checkbox"/>	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Scan status	Vulnerabilities
<input type="checkbox"/>	1	Image	February 08, 2023, 18:21:04 (UTC-06)	382.11	Copy URI	sha256:4c0e0210fdedf0e...	-	-