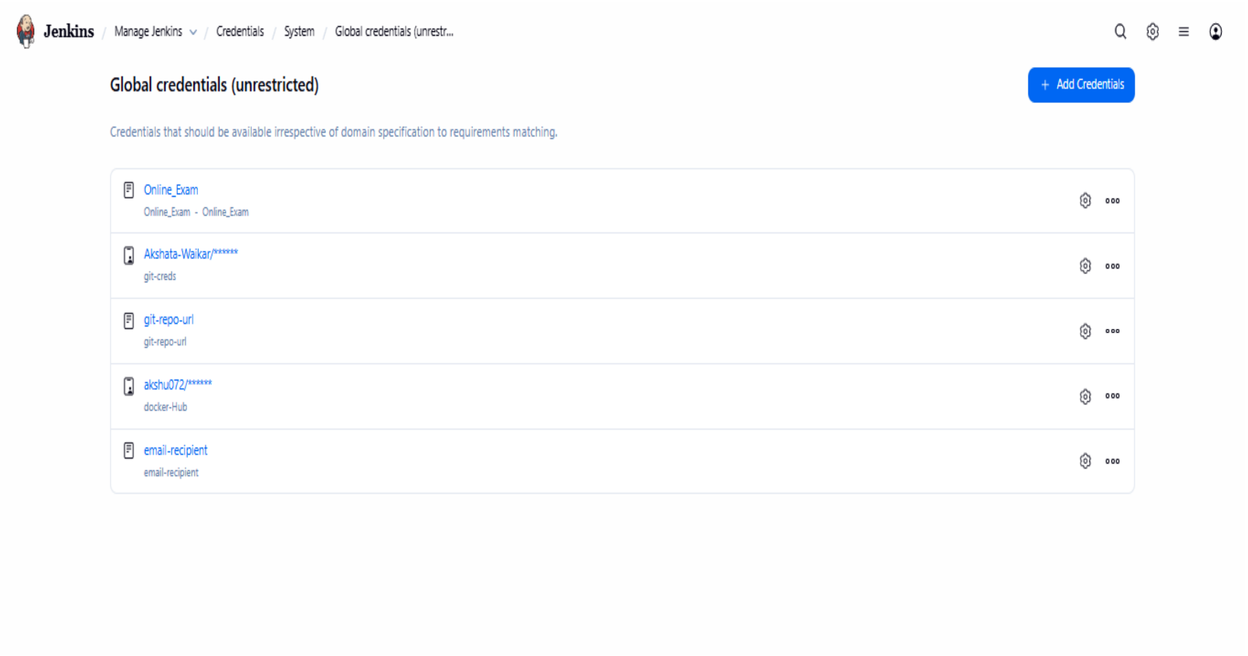
Jenkins pipleline Project

Credentials save in Jenkins:

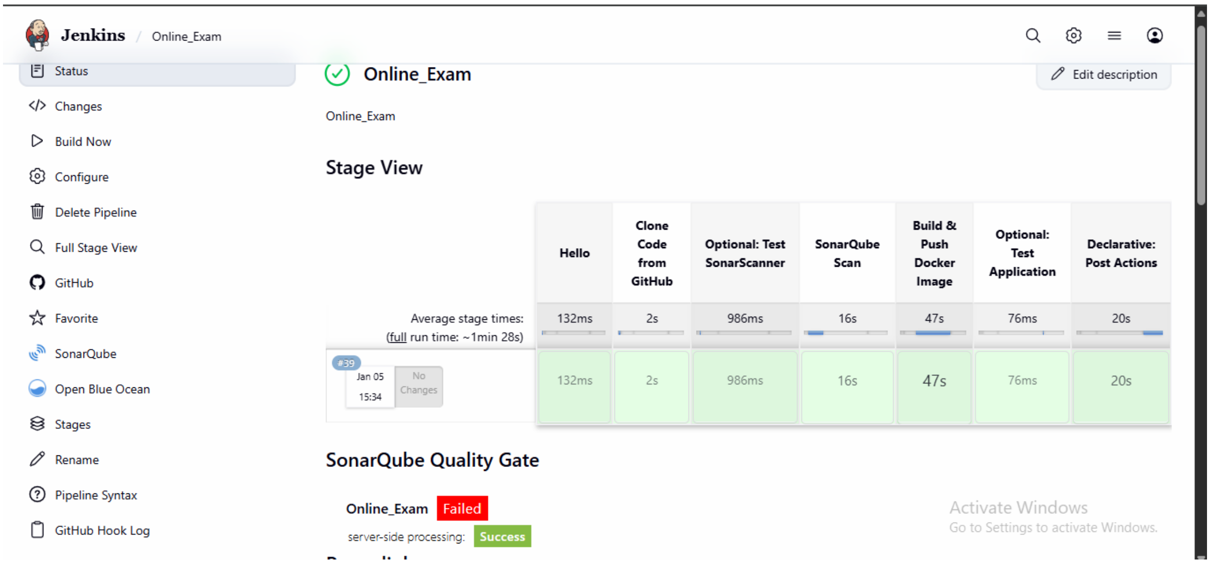


Global credentials (unrestricted)

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

- Online_Exam
Online_Exam - Online_Exam
- Akshata-Waikar/******
git-creds
- git-repo-url
git-repo-url
- akshu072/******
docker-Hub
- email-recipient
email-recipient

Pipeline run successfully :



Online_Exam

Online_Exam

Stage View

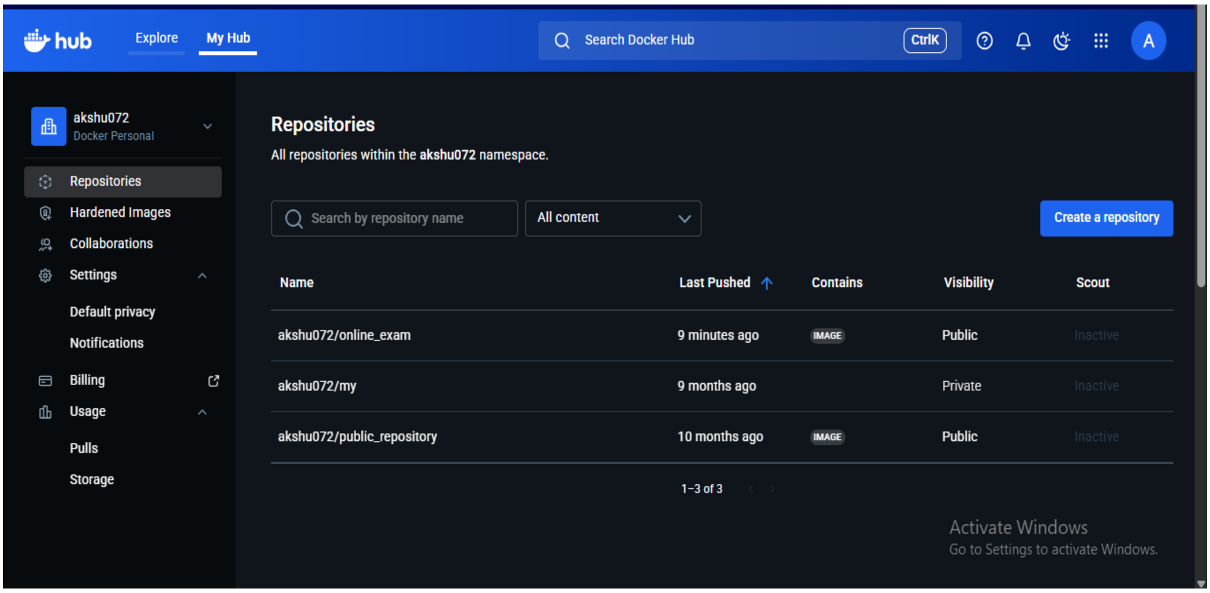
	Hello	Clone Code from GitHub	Optional: Test SonarScanner	SonarQube Scan	Build & Push Docker Image	Optional: Test Application	Declarative: Post Actions
Average stage times: (full run time: ~1min 28s)	132ms	2s	986ms	16s	47s	76ms	20s
#39 Jan 05 15:34 No Changes	132ms	2s	986ms	16s	47s	76ms	20s

SonarQube Quality Gate

Online_Exam Failed

server-side processing: Success

Push image to Docker-Hub :



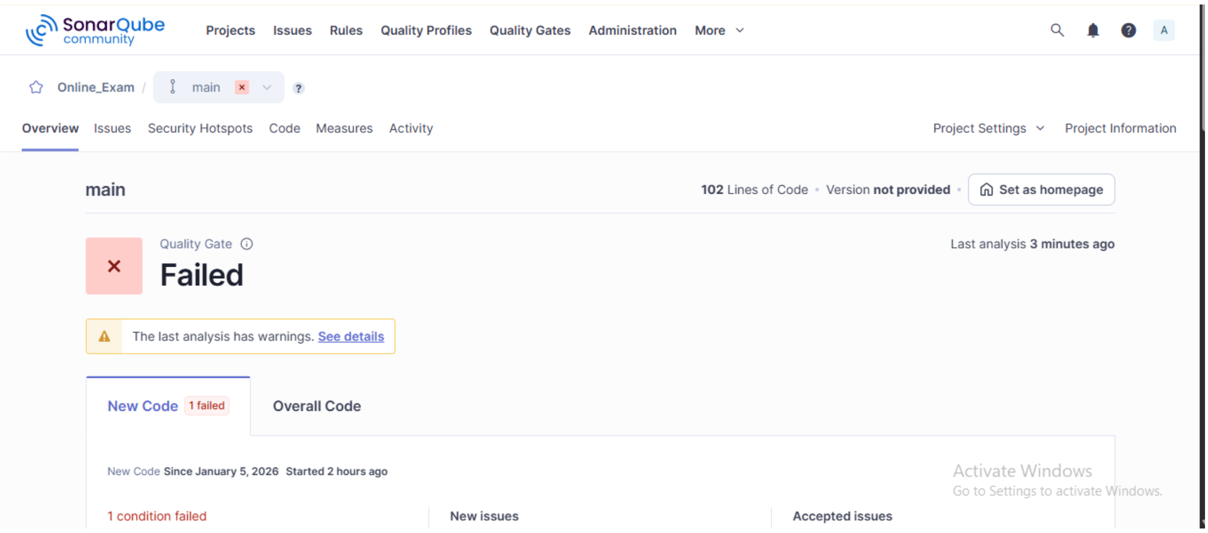
Repositories

All repositories within the **akshu072** namespace.

Name	Last Pushed	Contains	Visibility	Scout
akshu072/online_exam	9 minutes ago	IMAGE	Public	Inactive
akshu072/my	9 months ago		Private	Inactive
akshu072/public_repository	10 months ago	IMAGE	Public	Inactive

1–3 of 3

SonarQube :



main

102 Lines of Code · Version **not provided**

Quality Gate

Failed

Last analysis **3 minutes ago**

The last analysis has warnings. See details

New Code **Since January 5, 2026** **Started 2 hours ago**

1 condition failed

New issues

Accepted issues

GitHub repository link : https://github.com/Akshata-Waikar/Online_Exam.git

Jenkins pipeline code :

```
pipeline {
    agent any

    stages {

        stage('Hello') {
            steps {
                echo 'Hello World'
            }
        }

        stage('Clone Code from GitHub') {
            steps {
                script {
                    echo 'Cloning repository...'
                    withCredentials([usernamePassword(
                        credentialsId: 'git-creds',
                        usernameVariable: 'GIT_USER',
                        passwordVariable: 'GIT_TOKEN'
                    )]) {
                        bat """
                        if exist project rmdir /s /q project
                        git clone https://%GIT_USER%:%GIT_TOKEN%@github.com/Akshata-Waikar/Online_Exam.git project
                        """
                    }
                }
            }
        }

        stage('Optional: Test SonarScanner') {
            steps {
                echo 'Checking if SonarScanner CLI is available...'
                bat 'C:\\sonar-scanner-7.3.0.5189-windows-x64\\bin\\sonar-scanner -v || echo "SonarScanner not found"'
            }
        }

        stage('SonarQube Scan') {
            steps {
                script {
                    withSonarQubeEnv('sonar-local') {
                        withCredentials([string(credentialsId: 'Online_Exam', variable: 'SONAR_TOKEN')]) {
                            bat """
                            cd project
                            C:\\sonar-scanner-7.3.0.5189-windows-x64\\bin\\sonar-scanner ^
                              -Dsonar.projectKey=Online_Exam ^
                              -Dsonar.sources=. ^
                              -Dsonar.host.url=%SONAR_HOST_URL% ^
                              -Dsonar.login=%SONAR_TOKEN%
                            """
                        }
                    }
                }
            }
        }

        stage('Quality Gate') {
            steps {
                script {
                    timeout(time: 5, unit: 'MINUTES') { // wait max 5 min for Quality Gate
                        def qg = waitForQualityGate()
                        echo "Quality Gate status: ${qg.status}"
                        if (qg.status == 'OK') {
                            echo "Quality Gate passed successfully."
                        } else {
                            error "Pipeline aborted due to Quality Gate failure: ${qg.status}"
                        }
                    }
                }
            }
        }

        stage('Build & Push Docker Image') {
            steps {
                script {
                    withCredentials([usernamePassword(
                        credentialsId: 'docker-Hub',
                        usernameVariable: 'DOCKER_USER',
                        passwordVariable: 'DOCKER_PASSWORD'
                    )]) {
                        bat """
                        echo %DOCKER_PASSWORD% | docker login -u %DOCKER_USER% --password-stdin
                        docker build -t akshu072/online_exam:latest project
                        docker push akshu072/online_exam:latest
                        """
                    }
                }
            }
        }

        stage('Optional: Test Application') {
            steps {
                echo 'You can add any application tests here if needed'
            }
        }
    }

    post {
        always {
            echo 'Cleaning up...'
            bat 'docker logout'
        }

        success {
            echo 'Build Succeeded! Sending email...'
            withCredentials([string(credentialsId: 'email-recipient', variable: 'EMAIL_TO')]) {
                emailext(
                    subject: "SUCCESS: Jenkins Build ${currentBuild.fullDisplayName}",
                    body: """<p>Good news! Your Jenkins build succeeded.</p>
                             <p>Build URL: ${env.BUILD_URL}</p>""",
                    to: "${EMAIL_TO}",
                    mimeType: 'text/html'
                )
            }
        }

        failure {
            echo 'Build Failed! Sending email...'
            withCredentials([string(credentialsId: 'email-recipient', variable: 'EMAIL_TO')]) {
                emailext(
                    subject: "FAILURE: Jenkins Build ${currentBuild.fullDisplayName}",
                    body: """<p>Oops! Your Jenkins build failed.</p>
                             <p>Check console output here: ${env.BUILD_URL}</p>""",
                    to: "${EMAIL_TO}",
                    mimeType: 'text/html'
                )
            }
        }
    }
}
```

Name : Akshata Waikar