Deakin University

Professional Practice in IT

OnTrack Submission

Task 6.1C Creating a Jenkins Pipeline with GitHub Integration

Submitted By:
Sathiyanarayanan Senthil Kumar
s223789819
2024/05/03 23:53

Tutor:
Sam Kharazmi

Outcome	Weight
Engage with processes, tools, and practices associated with agile project management across all phases of the dev-ops lifecycle, and use software tools to contribute to real-world projects at each stage in an effective manner	♦♦♦♦♦
Evaluate, review, and synthesise real-world scenarios to inform discussion and practice of IT, and relate to professional practice, codes of ethics, and principles of intellectual property and its protection.	♦♦♦◊◊

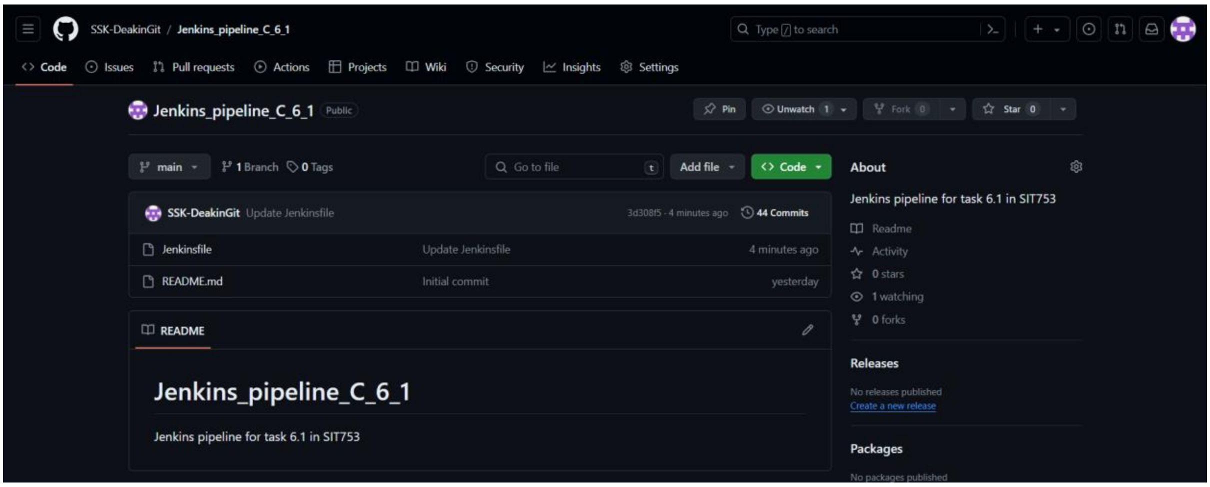
In this task, I got an opportunity to build a project in Jenkins using a Jenkinsfile present in GitHub repository. By completing this task, I got to know how CI is done in real-life projects. So, I believe, this task is strongly related to ULOs 1 and 3

May 3, 2024

Produced by Doubtfire

Continuous Integration and Deployment with Jenkins and GitHub

GitHub repository link: https://github.com/SSK-DeakinGit/Jenkins_pipeline_C_6_1



Jenkins_pipeline_C_6_1 Public

SSK-DeakinGit Update Jenkinsfile		3d308f5 · 4 minutes ago 44 Commits
Jenkinsfile	Update Jenkinsfile	4 minutes ago
README.md	Initial commit	yesterday

README

Jenkins_pipeline_C_6_1

Jenkins pipeline for task 6.1 in SIT753

About

Jenkins pipeline for task 6.1 in SIT753

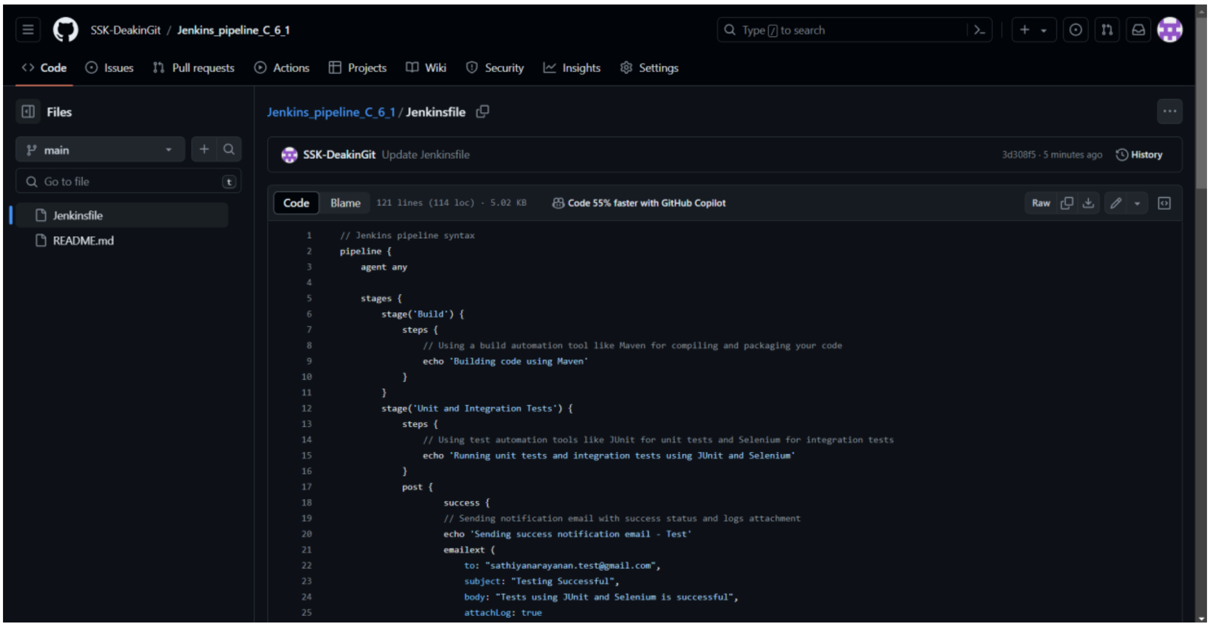
Readme
Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published



Jenkins_pipeline_C_6_1 / Jenkinsfile

SSK-DeakinGit Update Jenkinsfile 3d308f5 · 5 minutes ago

121 lines (114 loc) · 5.02 KB

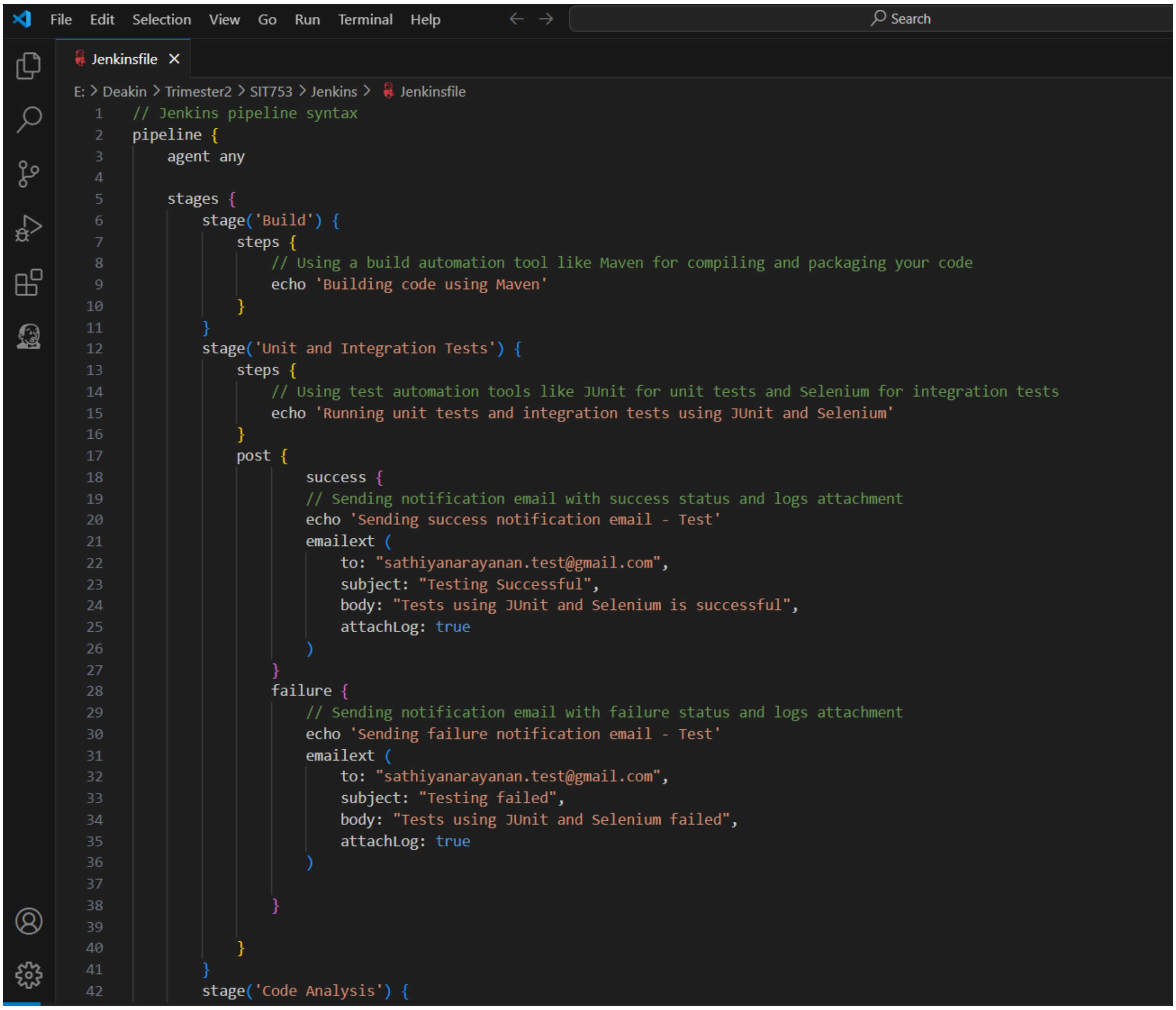
```
// Jenkins pipeline syntax
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                // Using a build automation tool like Maven for compiling and packaging your code
                echo 'Building code using Maven'
            }
        }
        stage('Unit and Integration Tests') {
            steps {
                // Using test automation tools like JUnit for unit tests and Selenium for integration tests
                echo 'Running unit tests and integration tests using JUnit and Selenium'
            }
            post {
                    success {
                    // Sending notification email with success status and logs attachment
                    echo 'Sending success notification email - Test'
                    emailext (
                        to: "sathiyanarayanan.test@gmail.com",
                        subject: "Testing Successful",
                        body: "Tests using JUnit and Selenium is successful",
                        attachLog: true
```

Jenkinsfile

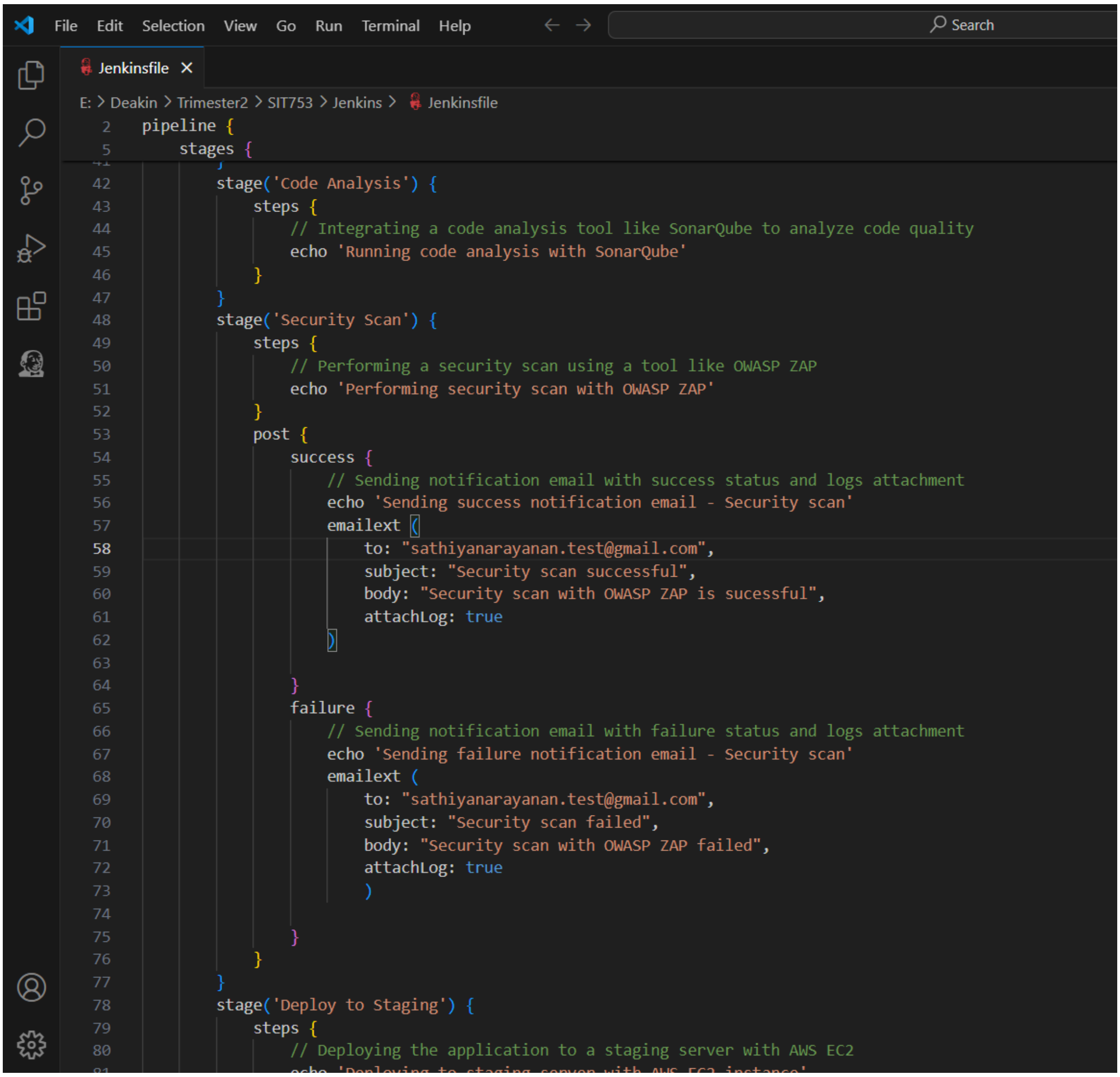
```
// Jenkins pipeline syntax
pipeline {
  agent any

  stages {
    stage('Build') {
      steps {
        // Using a build automation tool like Maven for compiling and packaging your code
        echo 'Building code using Maven'
      }
    }
    stage('Unit and Integration Tests') {
      steps {
        // Using test automation tools like JUnit for unit tests and Selenium for integration tests
        echo 'Running unit tests and integration tests using JUnit and Selenium'
      }
      post {
          success {
          // Sending notification email with success status and logs attachment
          echo 'Sending success notification email - Test'
          emailext (
            to: "sathiyanarayanan.test@gmail.com",
            subject: "Testing Successful",
            body: "Tests using JUnit and Selenium is successful",
            attachLog: true
          )
        }
        failure {
            // Sending notification email with failure status and logs attachment
            echo 'Sending failure notification email - Test'
            emailext (
                to: "sathiyanarayanan.test@gmail.com",
                subject: "Testing failed",
                body: "Tests using JUnit and Selenium failed",
                attachLog: true
            )

        }

    }
}
stage('Code Analysis') {
    steps {
        // Integrating a code analysis tool like SonarQube to analyze code quality
        echo 'Running code analysis with SonarQube'
    }
}
stage('Security Scan') {
    steps {
        // Performing a security scan using a tool like OWASP ZAP
        echo 'Performing security scan with OWASP ZAP'
    }
    post {
        success {
            // Sending notification email with success status and logs attachment
```

```
      echo 'Sending success notification email - Security scan'
      emailext (
        to: "sathiyanarayanan.test@gmail.com",
        subject: "Security scan successful",
        body: "Security scan with OWASP ZAP is sucessful",
        attachLog: true
      )

    }
    failure {
      // Sending notification email with failure status and logs attachment
      echo 'Sending failure notification email - Security scan'
      emailext (
        to: "sathiyanarayanan.test@gmail.com",
        subject: "Security scan failed",
        body: "Security scan with OWASP ZAP failed",
        attachLog: true
        )

    }
  }
}
stage('Deploy to Staging') {
  steps {
    // Deploying the application to a staging server with AWS EC2
    echo 'Deploying to staging server with AWS EC2 instance'
  }
}
stage('Integration Tests on Staging') {
  steps {
    // Running integration tests on the staging environment with Selenium
    echo 'Running integration tests on staging environment using Selenium'
  }
  post {
    success {
      // Sending notification email with success status and logs attachment
      echo 'Sending success notification email - Integration test'
      emailext (
        to: "sathiyanarayanan.test@gmail.com",
        subject: "Integration test successful",
        body: "Integration tests on staging environment using Selenium is sucessful",
        attachLog: true
      )

    }
    failure {
      // Sending notification email with failure status and logs attachment
      echo 'Sending failure notification email - Integration test'
      emailext (
        to: "sathiyanarayanan.test@gmail.com",
        subject: "Integration test failed",
        body: "Integration tests on staging environment using Selenium failed",
        attachLog: true
        )

    }
  }
}
```

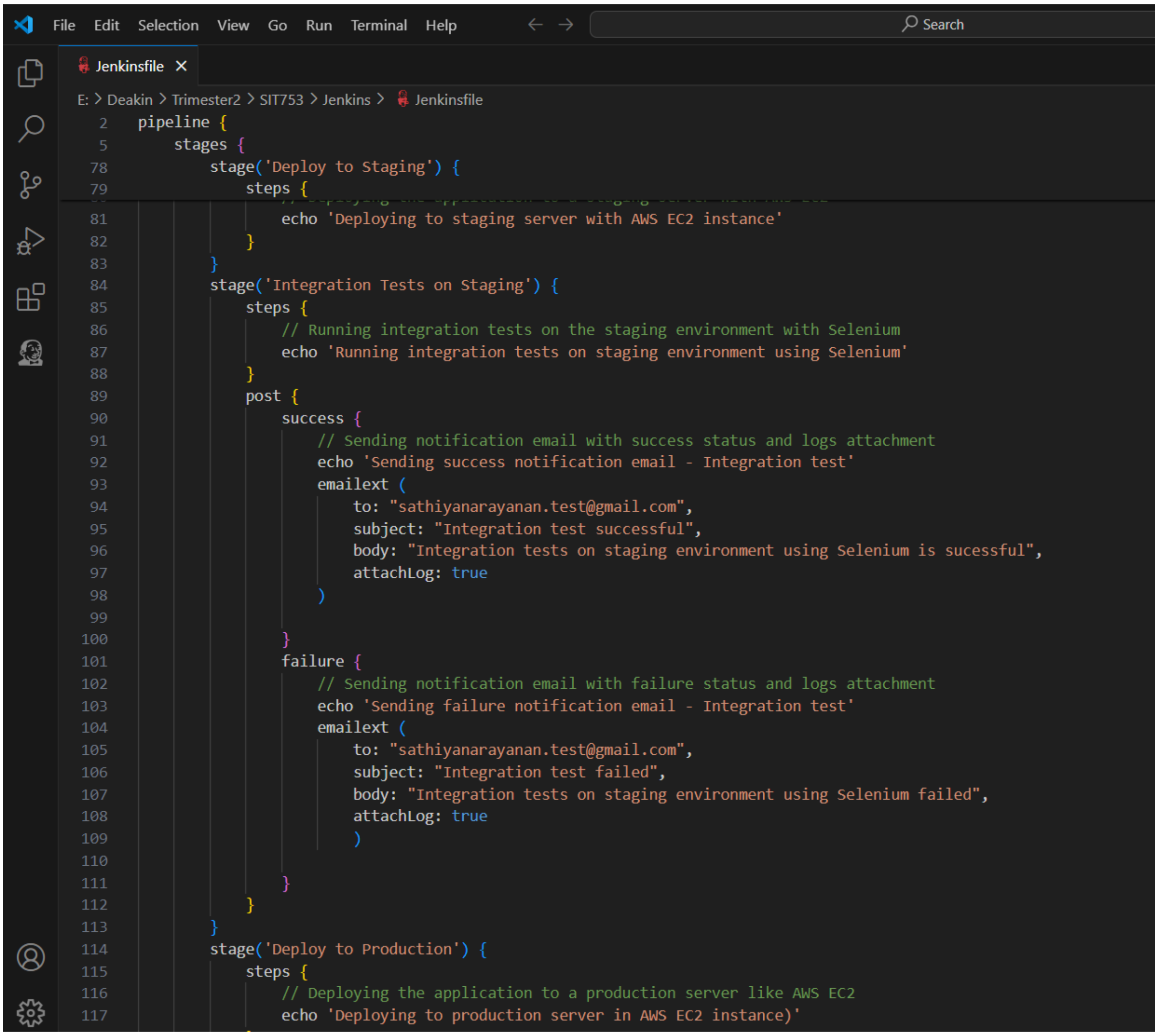
```
        stage('Deploy to Production') {
            steps {
                // Deploying the application to a production server like AWS EC2
                echo 'Deploying to production server in AWS EC2 instance)'
            }
        }
    }
}
```



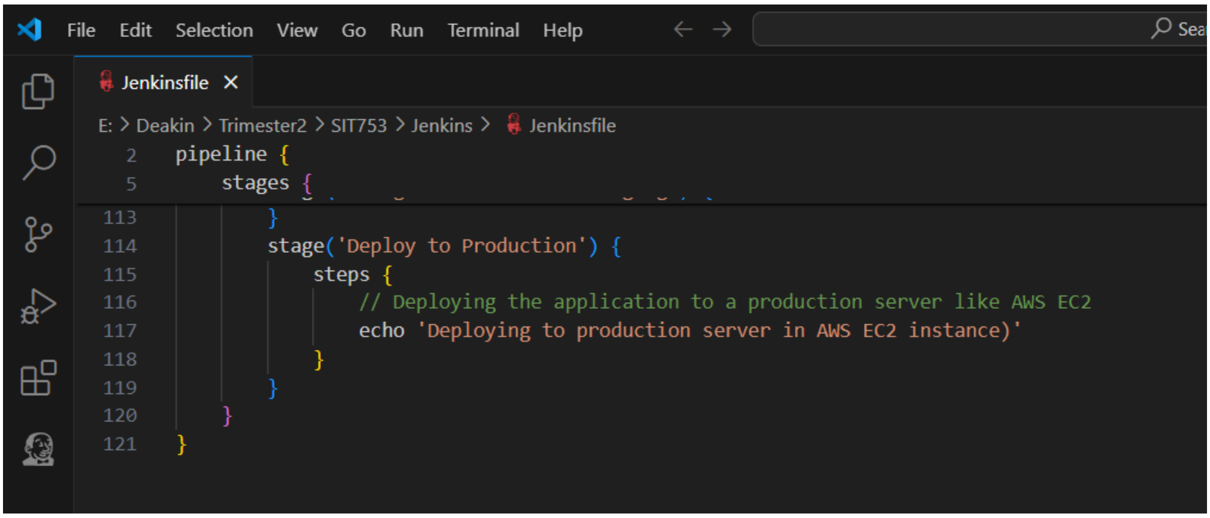
```
// Jenkins pipeline syntax
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                // Using a build automation tool like Maven for compiling and packaging your code
                echo 'Building code using Maven'
            }
        }
        stage('Unit and Integration Tests') {
            steps {
                // Using test automation tools like JUnit for unit tests and Selenium for integration tests
                echo 'Running unit tests and integration tests using JUnit and Selenium'
            }
            post {
                    success {
                    // Sending notification email with success status and logs attachment
                    echo 'Sending success notification email - Test'
                    emailext (
                        to: "sathiyanarayanan.test@gmail.com",
                        subject: "Testing Successful",
                        body: "Tests using JUnit and Selenium is successful",
                        attachLog: true
                    )
                }
                failure {
                    // Sending notification email with failure status and logs attachment
                    echo 'Sending failure notification email - Test'
                    emailext (
                        to: "sathiyanarayanan.test@gmail.com",
                        subject: "Testing failed",
                        body: "Tests using JUnit and Selenium failed",
                        attachLog: true
                    )

                }

            }
        }
        stage('Code Analysis') {
```



```
pipeline {
    stages {
        stage('Code Analysis') {
            steps {
                // Integrating a code analysis tool like SonarQube to analyze code quality
                echo 'Running code analysis with SonarQube'
            }
        }
        stage('Security Scan') {
            steps {
                // Performing a security scan using a tool like OWASP ZAP
                echo 'Performing security scan with OWASP ZAP'
            }
            post {
                success {
                    // Sending notification email with success status and logs attachment
                    echo 'Sending success notification email - Security scan'
                    emailext (
                        to: "sathiyanarayanan.test@gmail.com",
                        subject: "Security scan successful",
                        body: "Security scan with OWASP ZAP is sucessful",
                        attachLog: true
                    )

                }
                failure {
                    // Sending notification email with failure status and logs attachment
                    echo 'Sending failure notification email - Security scan'
                    emailext (
                        to: "sathiyanarayanan.test@gmail.com",
                        subject: "Security scan failed",
                        body: "Security scan with OWASP ZAP failed",
                        attachLog: true
                        )

                }
            }
        }
        stage('Deploy to Staging') {
            steps {
                // Deploying the application to a staging server with AWS EC2
                echo 'Deploying to staging server with AWS EC2 instance'
```



```
pipeline {
    stages {
        stage('Deploy to Staging') {
            steps {
                echo 'Deploying to staging server with AWS EC2 instance'
            }
        }
        stage('Integration Tests on Staging') {
            steps {
                // Running integration tests on the staging environment with Selenium
                echo 'Running integration tests on staging environment using Selenium'
            }
            post {
                success {
                    // Sending notification email with success status and logs attachment
                    echo 'Sending success notification email - Integration test'
                    emailext (
                        to: "sathiyanarayanan.test@gmail.com",
                        subject: "Integration test successful",
                        body: "Integration tests on staging environment using Selenium is sucessful",
                        attachLog: true
                    )

                }
                failure {
                    // Sending notification email with failure status and logs attachment
                    echo 'Sending failure notification email - Integration test'
                    emailext (
                        to: "sathiyanarayanan.test@gmail.com",
                        subject: "Integration test failed",
                        body: "Integration tests on staging environment using Selenium failed",
                        attachLog: true
                        )

                }
            }
        }
        stage('Deploy to Production') {
            steps {
                // Deploying the application to a production server like AWS EC2
                echo 'Deploying to production server in AWS EC2 instance)'
```



```
pipeline {
    stages {
        }
        stage('Deploy to Production') {
            steps {
                // Deploying the application to a production server like AWS EC2
                echo 'Deploying to production server in AWS EC2 instance)'
            }
        }
    }
}
```

Presentation link: https://youtu.be/z2D15CTLVmk