

# DEAKIN UNIVERSITY

## PROFESSIONAL PRACTICE IN IT

### ONTRACK SUBMISSION

---

# Task 6.1C Creating a Jenkins Pipeline with GitHub Integration

---

*Submitted By:*

Sathiyamarayanan 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.                  | ◆◆◆◆◇  |

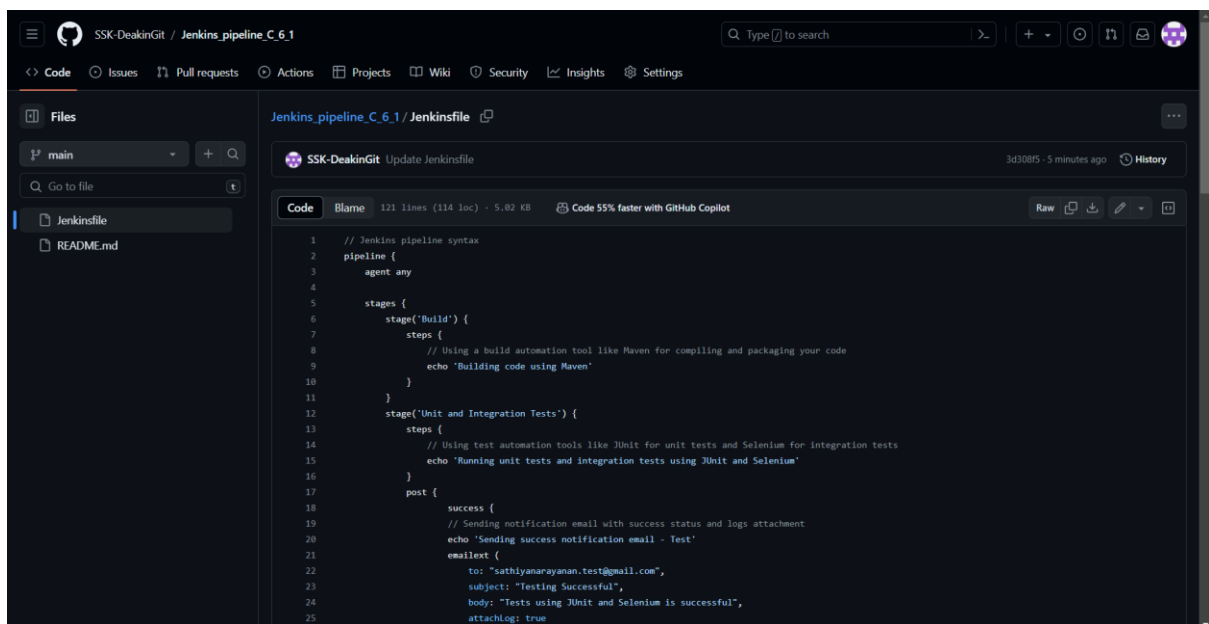
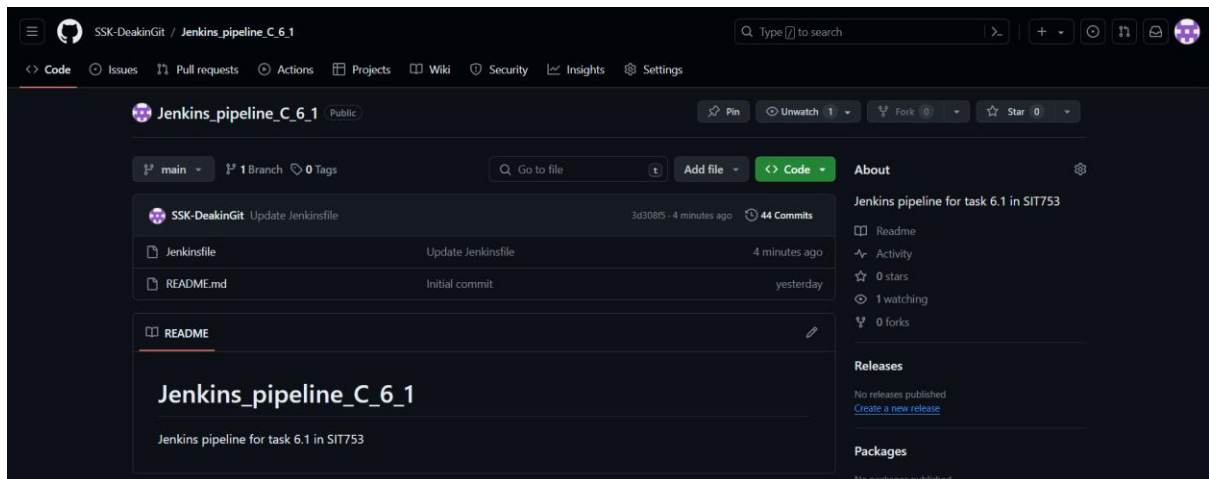
In this task, I got an opportunity to build a project in Jenkins using a Jenkinsfile present in GitHub repository. By completing this task, I got to know how CI is done in real-life projects. So, I believe, this task is strongly related to ULOs 1 and 3

May 3, 2024



# Continuous Integration and Deployment with Jenkins and GitHub

GitHub repository link: [https://github.com/SSK-DeakinGit/Jenkins\\_pipeline\\_C\\_6\\_1](https://github.com/SSK-DeakinGit/Jenkins_pipeline_C_6_1)



# 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'
                emailx (
                    to: "sathiyamarayanan.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'

        emailx (
            to: "sathiyarayanan.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'

    emailx (
        to: "sathiyamarayanantest@gmail.com",
        subject: "Security scan successful",
        body: "Security scan with OWASP ZAP is successful",
        attachLog: true
    )

}

failure {
    // Sending notification email with failure status and logs attachment
    echo 'Sending failure notification email - Security scan'

    emailx (
        to: "sathiyamarayanantest@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'
        emailx (
            to: "sathiyarayanan.test@gmail.com",
            subject: "Integration test successful",
            body: "Integration tests on staging environment using Selenium is successful",
            attachLog: true
        )

    }
    failure {
        // Sending notification email with failure status and logs attachment
        echo 'Sending failure notification email - Integration test'
        emailx (
            to: "sathiyarayanan.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)'

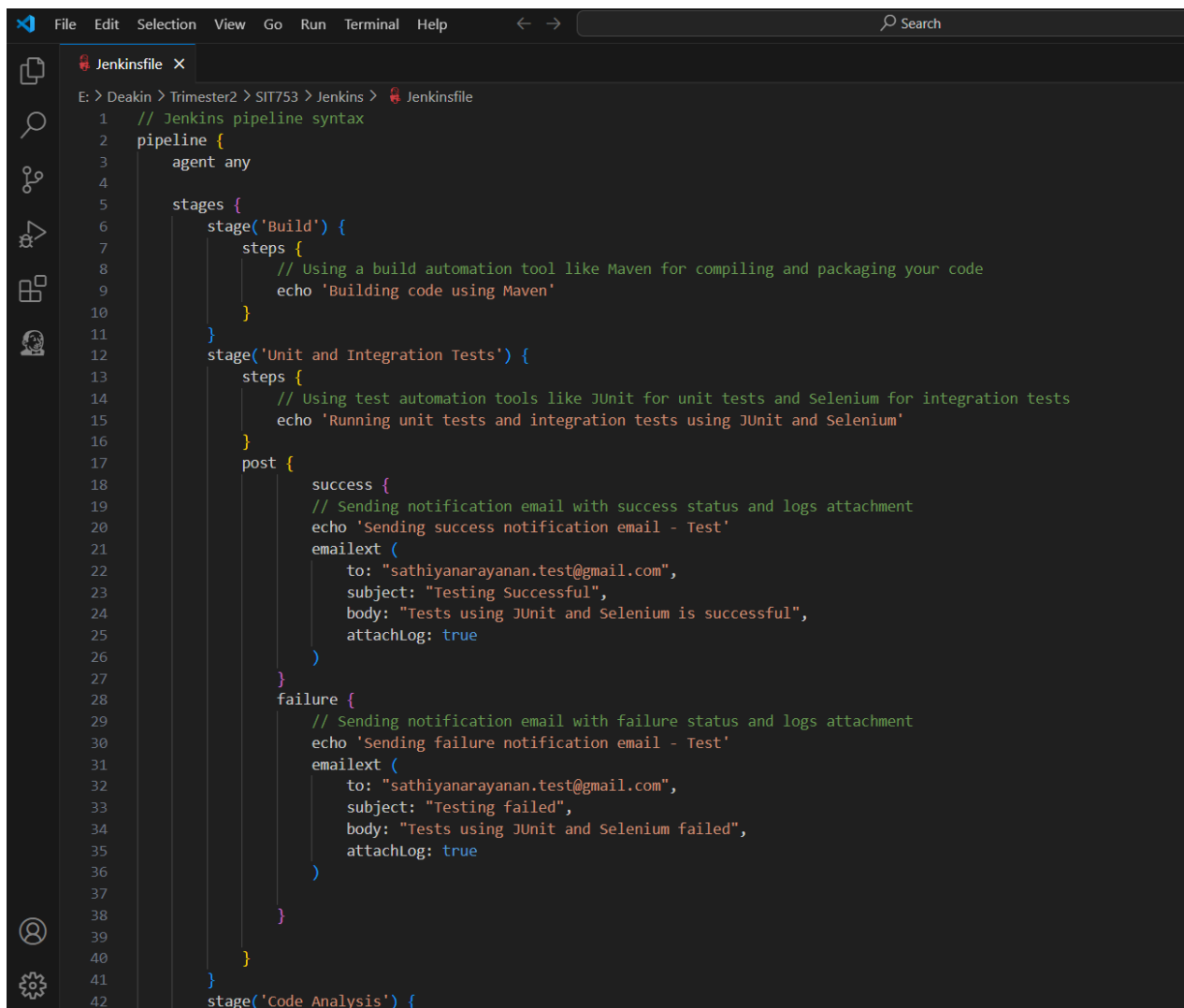
    }

}

}

}

```

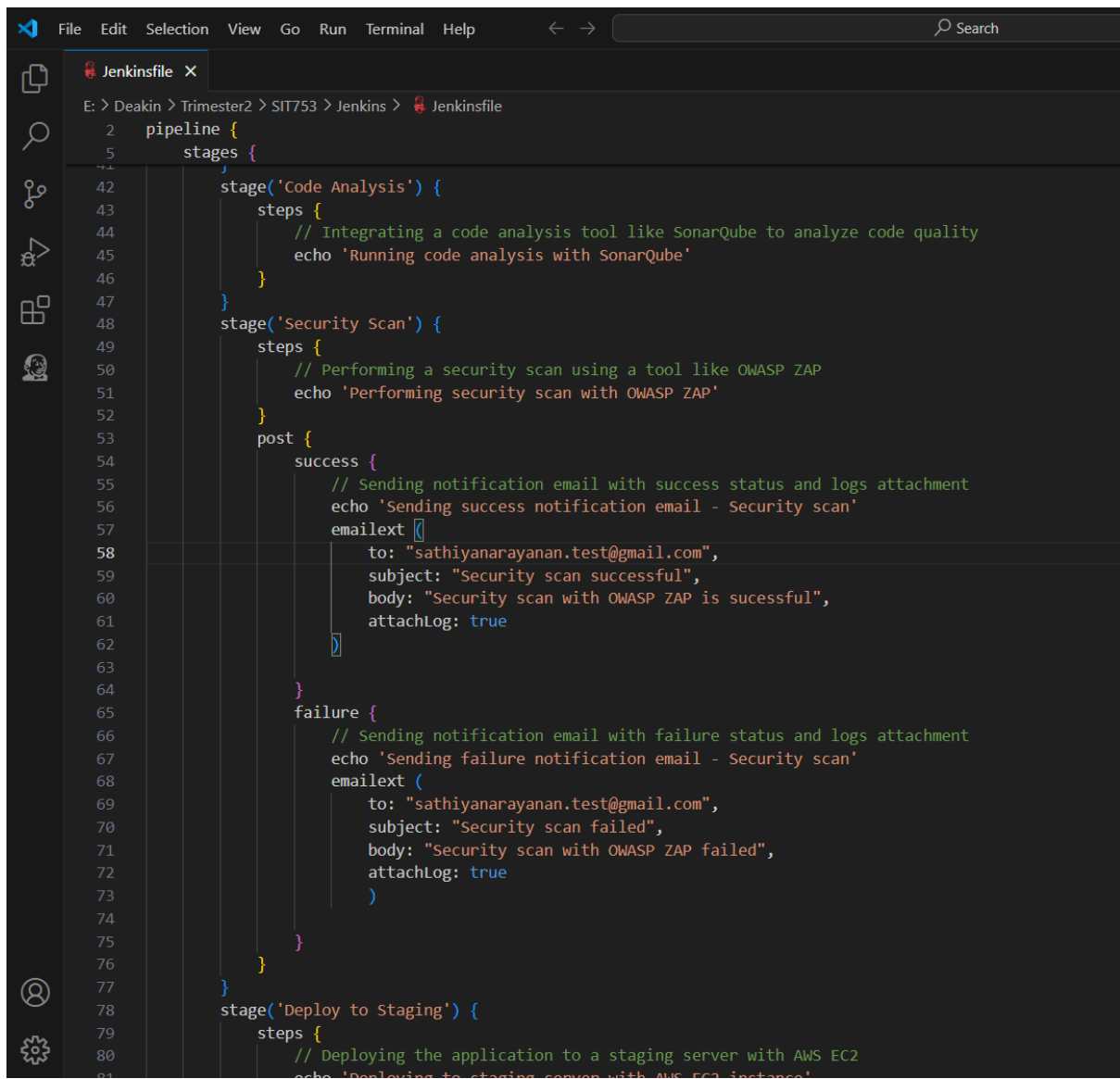


The screenshot shows a code editor with a Jenkinsfile. The file defines a pipeline with three main stages: Build, Unit and Integration Tests, and Code Analysis. The Build stage uses Maven for compilation and packaging. The Unit and Integration Tests stage uses JUnit and Selenium for testing. The Code Analysis stage is partially visible at the bottom. The pipeline also includes a post-build step for sending email notifications on success or failure.

```

E: > Deakin > Trimester2 > SIT753 > Jenkins > Jenkinsfile
1  // Jenkins pipeline syntax
2  pipeline {
3      agent any
4
5      stages {
6          stage('Build') {
7              steps {
8                  // Using a build automation tool like Maven for compiling and packaging your code
9                  echo 'Building code using Maven'
10             }
11         }
12         stage('Unit and Integration Tests') {
13             steps {
14                 // Using test automation tools like JUnit for unit tests and Selenium for integration tests
15                 echo 'Running unit tests and integration tests using JUnit and Selenium'
16             }
17             post {
18                 success {
19                     // Sending notification email with success status and logs attachment
20                     echo 'Sending success notification email - Test'
21                     emailx (
22                         to: "sathiyarayanan.test@gmail.com",
23                         subject: "Testing Successful",
24                         body: "Tests using JUnit and Selenium is successful",
25                         attachLog: true
26                     )
27                 }
28                 failure {
29                     // Sending notification email with failure status and logs attachment
30                     echo 'Sending failure notification email - Test'
31                     emailx (
32                         to: "sathiyarayanan.test@gmail.com",
33                         subject: "Testing failed",
34                         body: "Tests using JUnit and Selenium failed",
35                         attachLog: true
36                     )
37                 }
38             }
39         }
40     }
41 }
42 stage('Code Analysis') {

```



The image shows a code editor window with a dark theme. The title bar at the top reads "Jenkinsfile X". Below the title bar, the file path is displayed as "E: > Deakin > Trimester2 > SIT753 > Jenkins > Jenkinsfile". The editor contains a Jenkinsfile script with the following content:

```
2 pipeline {
3   stages {
4     stage('Code Analysis') {
5       steps {
6         // Integrating a code analysis tool like SonarQube to analyze code quality
7         echo 'Running code analysis with SonarQube'
8       }
9     }
10    stage('Security Scan') {
11      steps {
12        // Performing a security scan using a tool like OWASP ZAP
13        echo 'Performing security scan with OWASP ZAP'
14      }
15      post {
16        success {
17          // Sending notification email with success status and logs attachment
18          echo 'Sending success notification email - Security scan'
19          emailx (
20            to: "sathiyarayanan.test@gmail.com",
21            subject: "Security scan successful",
22            body: "Security scan with OWASP ZAP is successful",
23            attachLog: true
24          )
25        }
26        failure {
27          // Sending notification email with failure status and logs attachment
28          echo 'Sending failure notification email - Security scan'
29          emailx (
30            to: "sathiyarayanan.test@gmail.com",
31            subject: "Security scan failed",
32            body: "Security scan with OWASP ZAP failed",
33            attachLog: true
34          )
35        }
36      }
37    }
38  }
39  stage('Deploy to Staging') {
40    steps {
41      // Deploying the application to a staging server with AWS EC2
42      echo 'Deploying to staging server with AWS EC2 instance'
```



```
File Edit Selection View Go Run Terminal Help ← → Search

Jenkinsfile X
E: > Deakin > Trimester2 > SIT753 > Jenkins > Jenkinsfile
2 pipeline {
5     stages {
78         stage('Deploy to Staging') {
79             steps {
81                 echo 'Deploying to staging server with AWS EC2 instance'
82             }
83         }
84         stage('Integration Tests on Staging') {
85             steps {
86                 // Running integration tests on the staging environment with Selenium
87                 echo 'Running integration tests on staging environment using Selenium'
88             }
89             post {
90                 success {
91                     // Sending notification email with success status and logs attachment
92                     echo 'Sending success notification email - Integration test'
93                     emailx (
94                         to: "sathiyarayanan.test@gmail.com",
95                         subject: "Integration test successful",
96                         body: "Integration tests on staging environment using Selenium is successful",
97                         attachLog: true
98                     )
99                 }
100             }
101             failure {
102                 // Sending notification email with failure status and logs attachment
103                 echo 'Sending failure notification email - Integration test'
104                 emailx (
105                     to: "sathiyarayanan.test@gmail.com",
106                     subject: "Integration test failed",
107                     body: "Integration tests on staging environment using Selenium failed",
108                     attachLog: true
109                 )
110             }
111         }
112     }
113 }
114 stage('Deploy to Production') {
115     steps {
116         // Deploying the application to a production server like AWS EC2
117         echo 'Deploying to production server in AWS EC2 instance'
```

```
File Edit Selection View Go Run Terminal Help ← → Search

Jenkinsfile X
E: > Deakin > Trimester2 > SIT753 > Jenkins > Jenkinsfile
2 pipeline {
5     stages {
113     }
114     stage('Deploy to Production') {
115         steps {
116             // Deploying the application to a production server like AWS EC2
117             echo 'Deploying to production server in AWS EC2 instance'
118         }
119     }
120 }
121 }
```

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