



SIT223-SIT753

Credit Task:

Continuous Integration and  
Deployment with Jenkins and  
GitHub

# Continuous Integration and Deployment with Jenkins and GitHub

This is an individual Credit task.

## Overview

In this assessment task, you will be required to create a Jenkins pipeline that integrates with GitHub. You will also be required to integrate email notifications to inform developers about build status and deployment events. Additionally, for each stage of the pipeline, you will be required to specify or research a tool that can be used for that stage.

## Task Instructions

1. Create a GitHub repository and configure it to be used with Jenkins.
2. Create a Jenkins job that is designated to run the pipeline. The job should be triggered with a short delay whenever a new commit is pushed to the GitHub repository. So once a new commit is detected, the job will eventually run without manual intervention. You do not need to fully implement the webhook for automatic commit events.
3. Define a pipeline with 7 stages, each stage should have a specific task to perform. You will need to **provide a description of the tasks performed in each stage and a tool that could be used.** Note that actual implementation of the steps is not required; only the specified tasks and tools should be printed. The stages should include:

**Stage 1:** Build - Build the code using a build automation tool to compile and package your code. You need to specify at least one build automation tool e.g., Maven.

**Stage 2:** Unit and Integration Tests - run unit tests to ensure the code functions as expected and run integration tests to ensure the different components of the application work together as expected. You need to specify test automation tools for this stage.

**Stage 3:** Code Analysis - integrate a code analysis tool to analyse the code and ensure it meets industry standards. Research and select a tool to analyse your code using Jenkins.

**Stage 4:** Security Scan - perform a security scan on the code using a tool to identify any vulnerabilities. Research and select a tool to scan your code.

**Stage 5:** Deploy to Staging - deploy the application to a staging server (e.g., AWS EC2 instance).

**Stage 6:** Integration Tests on Staging - run integration tests on the staging environment to ensure the application functions as expected in a production-like environment.

**Stage 7:** Deploy to Production - deploy the application to a production server (e.g., AWS EC2 instance).

4. Configure the pipeline to send notification emails to a specified email address at the end of **test** and **security scan** stages. The notification emails should include the status of the stage (success or failure), and logs as attachment.
5. Test the pipeline by making a few commits to the GitHub repository and ensuring that the pipeline runs successfully, and notification emails are sent.

## Task Submission Instructions:

You must submit the following files to OnTrack:

1. A GitHub repository link with a Jenkinsfile containing the pipeline definition. Please make sure that your marking tutor has access to it.
2. Jenkinsfile that defines the pipeline stages and scripts.
3. A demo video that demonstrates the successful execution of the pipeline and the email notification received by you at the end of specified stages. Submit the link to your demo as a PDF document.