pipeline {

agent any

stages {

stage('Build') {

steps {

echo 'Build the code using a build automation tool.'

echo 'Tool: Maven'

}

}

stage('Unit and Integration Tests') {

steps {

echo 'Run unit tests and integration tests.'

echo 'Tool: JUnit for unit tests.'

echo 'Tool: Selenium foor integration tests.'

}

}

stage('Code Analysis') {

steps {

echo 'Analyze the code to ensure it meets industry standards.'

echo 'Tool: SonarQube'

}

}

stage('Security Scan') {

steps {

echo 'Perform a security scan to identify vulnerabilities.'

echo 'Tool: OWASP ZAP'

}

}

stage('Deploy to Staging') {

steps {

echo 'Deploy the application to a staging server.'

echo 'Staging Server: AWS EC2 instance'

}

}

stage('Integration Tests on Staging') {

steps {

echo 'Run integration tests on the staging environment.'

echo 'Tool: Selenium'

}

}

stage('Deploy to Production') {

steps {

echo 'Deploy the application to a production server.'

echo 'Production Server: AWS EC2 instance'

}

}

}

post {

always {

echo 'Pipeline has finished.'

}

success {

script {

def logFile = "${env.BUILD\_ID}.log"

writeFile file: logFile, text: currentBuild.rawBuild.getLog().join("\n")

emailext(

to: 'developer@example.com',

subject: "Build Successful: ${env.JOB\_NAME} Build #${env.BUILD\_NUMBER}",

body: """Good news! The build was successful. Check console output at ${env.BUILD\_URL} to view the results.""",

attachLog: true,

attachmentsPattern: logFile

)

}

}

failure {

script {

def logFile = "${env.BUILD\_ID}.log"

writeFile file: logFile, text: currentBuild.rawBuild.getLog().join("\n")

emailext(

to: 'developer@example.com',

subject: "Build Failed: ${env.JOB\_NAME} Build #${env.BUILD\_NUMBER}",

body: """Unfortunately, the build failed. Check console output at ${env.BUILD\_URL} to view the results.""",

attachLog: true,

attachmentsPattern: logFile

)

}

}

}

}