

Module 6 Assignment – Monitoring & CI Enhancements

Name: Jishnu Ganesh

Course: CSC 7435E – Secure DevOps Practices

Assignment: Module 6 – Monitoring & Secure Enhancements

1. Objective

The goal of this assignment is to enhance my existing Jenkins CI/CD pipeline (created in Module 4 & 5) by adding monitoring, logging improvements, and secure DevOps checks described in the Module 6 walkthrough.

These enhancements improve:

- Observability
- Troubleshooting
- Build stability
- Security awareness

This document explains what enhancements I added, why I added them, and includes screenshots as reference.

2. Enhancement 1: Monitoring Using Jenkins Timestamper (Walkthrough Enhancement)

To improve build observability, I enabled Timestamper so that every console log line shows a timestamp.

What this enhancement adds

- Allows tracking how long each step takes
- Helps investigate pipeline slowness
- Supports auditing and incident response

Where it is enabled

Manage Jenkins → Configure System → Timestamper
“Enable timestamps for all pipeline builds”

3. Enhancement 2: Monitoring & Security Stage in Jenkinsfile (Walkthrough Enhancement)

I added a new stage called Monitor & Security Checks, which performs lightweight security scanning and logging.

```
stage('Monitor & Security Checks') {
    steps {
        echo "🔍 Running monitoring and security checks..."
        sh ""
        echo "Checking for risky patterns..."
        if grep -q "Runtime.exec" BMIApp.java; then
            echo "⚠️ Security Risk: Usage of Runtime.exec detected!"
        fi
    }
}
```

```

        echo "Logging build status and file structure..."
        ls -R .
    ""
}
}

```

Why this enhancement is important

- Helps detect dangerous Java patterns like Runtime.exec
- Provides a transparent log of the workspace during the build
- Supports secure DevOps practices discussed in Module 6

4. Enhancement 3 – Build Failure Diagnostics (Additional CI Enhancement)

To support monitoring and debugging production pipelines, I added a failure diagnostics block to capture failure details.

```

post {
    failure {
        echo "✖ Build failed – collecting diagnostics..."
        sh ""
            echo "---- WORKSPACE CONTENTS ----"
            ls -R .
            echo "---- JAVA VERSION ----"
            java -version
        ""
    }
}

```

Why this enhancement matters

- Provides immediate context when a build fails
- Helps debug issues faster
- Reduces MTTR (Mean Time To Recovery)

5. Jenkinsfile (Final Version for Module 6)

Below is the combined Jenkinsfile including all Module 6 enhancements:

```

pipeline {
    agent any

    stages {
        stage('Checkout') {
            steps {
                checkout scm

```

```

        echo "Checked out code successfully."
    }
}

stage('Build') {
    steps {
        sh ""
        mkdir -p build
        javac -d build BMIApp.java
        ""
    }
}

stage('Test') {
    steps {
        sh 'echo Running basic test: BMIApp compiled!'
    }
}

stage('Monitor & Security Checks') {
    steps {
        echo "🔑 Running monitoring and security checks..."
        sh ""
        echo "Checking for risky patterns..."
        if grep -q "Runtime.exec" BMIApp.java; then
            echo "⚠️ Security Risk: Runtime.exec detected!"
        fi

        echo "Listing workspace for monitoring:"
        ls -R .
        ""
    }
}

stage('Package') {
    steps {
        sh ""
        mkdir -p dist
        tar -cf dist/BMIApp-$BUILD_NUMBER.tar build
        ""
    }
}

post {
    success {

```

```

        echo "✅ Build completed successfully."
    }
failure {
    echo "❌ Build failed – collecting diagnostics..."
    sh ""
    echo "---- WORKSPACE CONTENTS ----"
    ls -R .
    echo "---- JAVA VERSION ----"
    java -version
    ""
}
}
}

```

6. Justification of Enhancements

Timestamper Plugin

- Gives precise timing data
- Helps troubleshoot slow stages
- Essential for monitoring pipeline performance

Monitoring & Security Stage

- Detects risky code patterns (Security best practice)
- Increases pipeline transparency
- Helps understand pipeline health over time

Failure Diagnostics

- Reduces debugging time
- Supports better incident response
- Captures environment details on failure

These enhancements directly align with the secure DevOps practices.

7. Appendix

Below are all screenshots documenting the setup, enhancements, and execution of the Module 6.

Figure – Plugin Manager – along with Timestamper Plugin Enabled.

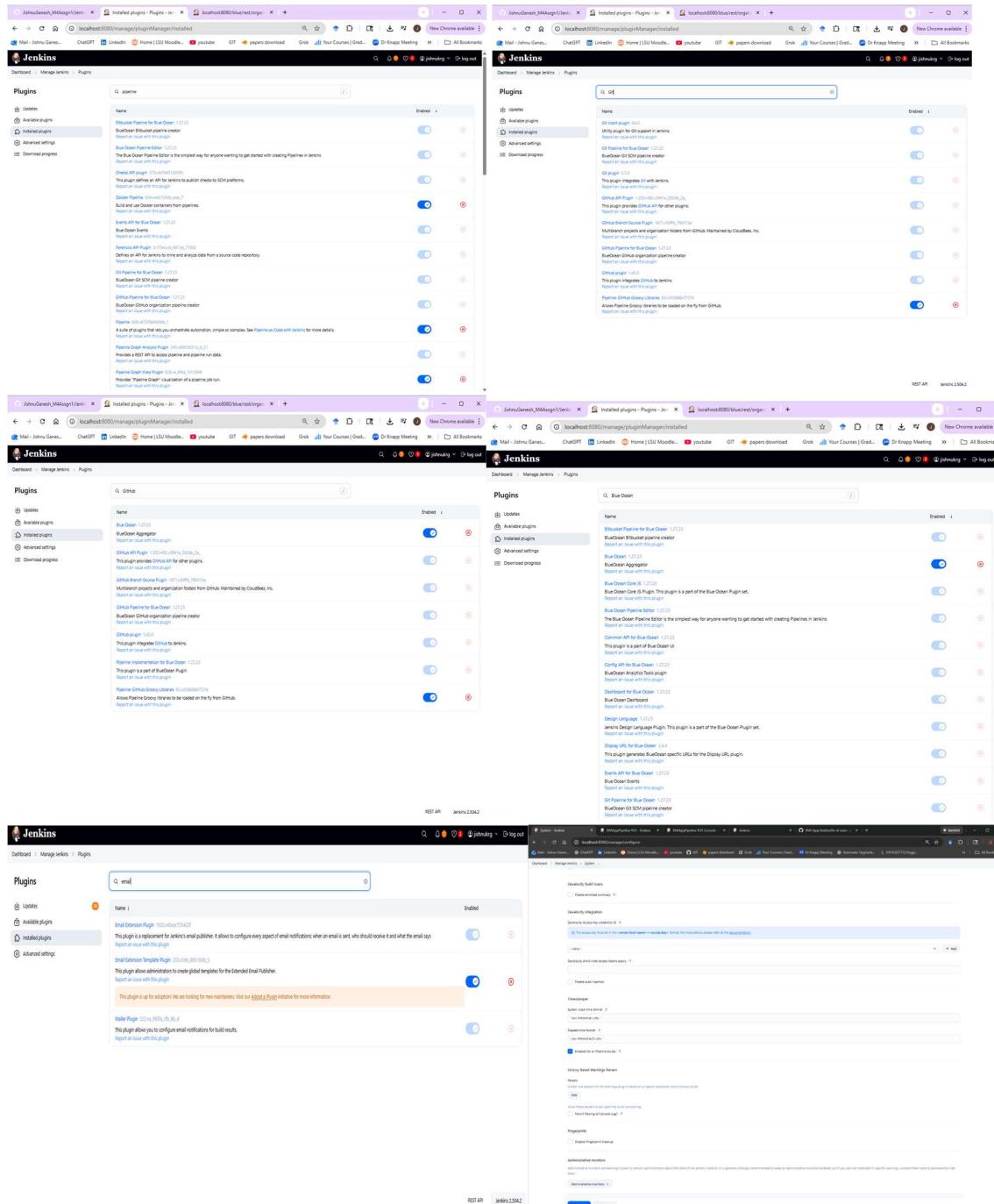


Figure – Jenkinsfile on GitHub

The screenshot shows a GitHub repository named "BMI-App" with a "Jenkinsfile" in the main directory. The Jenkinsfile defines a pipeline with stages for packaging, health monitoring, and release. It includes steps for creating a tar artifact, collecting basic information, and sending email notifications for success or failure.

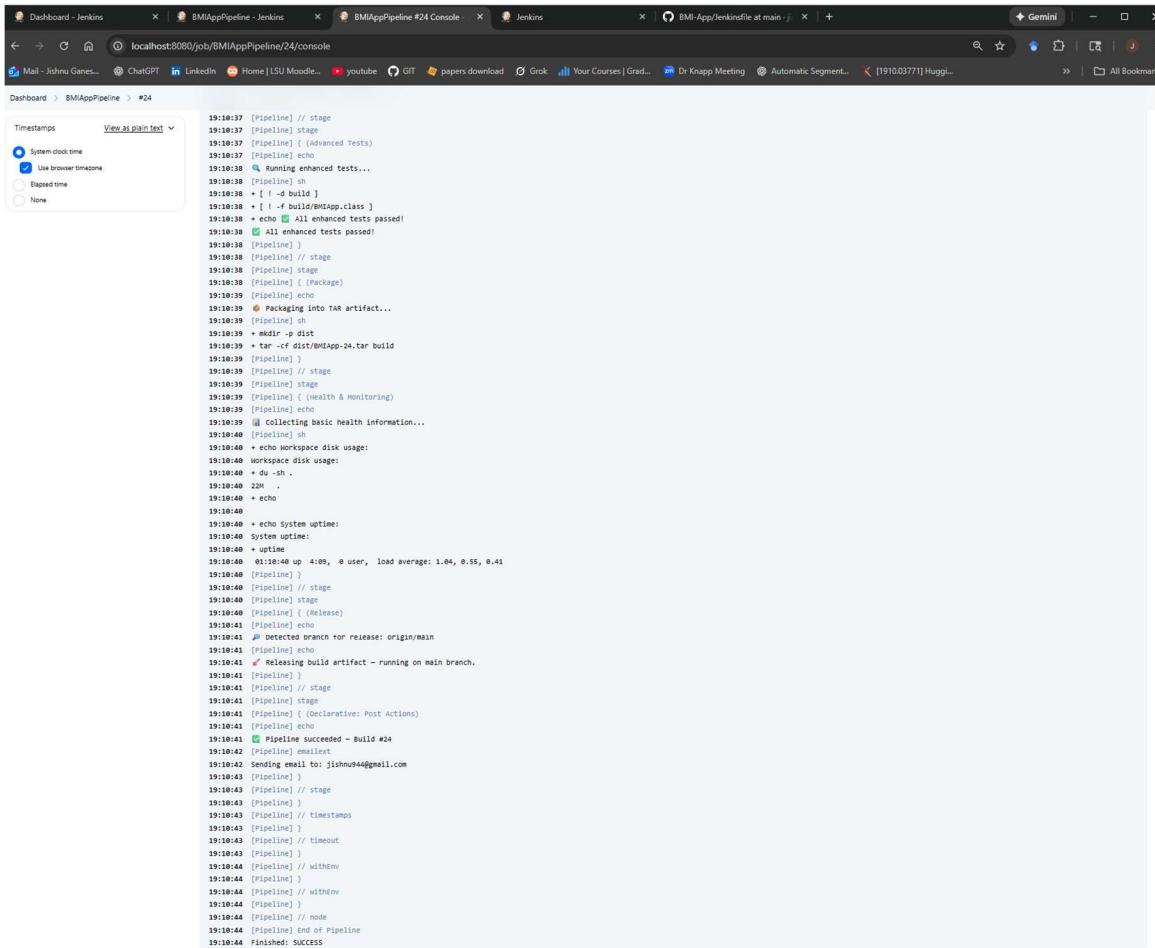
```
steps {
    echo 'Packaging into TAR artifact...'
    sh ''
    mkdir -p dist
    tar -cf dist/BMIApp-$BUILD_NUMBER.tar build
}
}

// New for Module N - Health & Monitoring stage
stage('Health & Monitoring') {
    steps {
        echo 'Collecting basic health information...'
        sh ''
        echo "Workspace disk usage"
        du -sh . || echo "du command not available"
        echo "System uptime"
        echo "System uptime"
        uptime || echo "uptime command not available"
    }
}

stage('Release') {
    when {
        expression {
            def branch = env.GIT_BRANCH
            env.BRANCH_NAME
            echo "Detected branch for release: ${branch}"
            return branch.contains('main')
        }
    }
    steps {
        echo "Releasing build artifact - running on main branch."
    }
}

post {
    success {
        echo "Pipeline succeeded - Build #${BUILD_NUMBER}"
        emalltext(
            subject: "BMI Pipeline SUCCESS - Build #${BUILD_NUMBER}",
            to: "jishnu944@gmail.com",
            body: "The BMIApp pipeline finished successfully on branch ${BRANCH_NAME}.\\nBuild URL: ${BUILD_URL}"
        )
    }
    failure {
        echo "Pipeline failed - Build #${BUILD_NUMBER}"
        emalltext(
            subject: "BMI Pipeline FAILED - Build #${BUILD_NUMBER}",
            to: "jishnu944@gmail.com",
            body: "The BMIApp pipeline FAILED on branch ${BRANCH_NAME}.\\nPlease check logs: ${BUILD_URL}"
        )
    }
}
```

Figure – Jenkins Console Output With Timestamps



The screenshot shows a browser window with multiple tabs open. The active tab displays Jenkins console output for a build named 'BMIAppPipeline #24'. The output is timestamped and shows various stages of the pipeline, including enhanced tests, packaging into a TAR artifact, and sending an email. A sidebar on the left allows selecting different timestamp options: System clock time, Use browser timezone, Elapsed time, or None.

```
19:18:37 [Pipeline] // stage
19:18:37 [Pipeline] stage
19:18:37 [Pipeline] (advanced Tests)
19:18:37 [Pipeline] echo
19:18:38 ☺ Running enhanced tests...
19:18:38 [Pipeline] sh
19:18:38 + [ ! -d build ]
19:18:38 + [ ! -f build/BMIApp.class ]
19:18:38 + echo ☻ All enhanced tests passed!
19:18:38 ☻ All enhanced tests passed!
19:18:38 [Pipeline] }
19:18:38 [Pipeline] // stage
19:18:38 [Pipeline] stage
19:18:38 [Pipeline] (Package)
19:18:39 [Pipeline] echo
19:18:39 ☺ Packaging into TAR artifact...
19:18:39 [Pipeline] sh
19:18:39 + mkdir -p dist
19:18:39 + tar -cf dist/BMIApp-24.tar build
19:18:39 [Pipeline] }
19:18:39 [Pipeline] // stage
19:18:39 [Pipeline] stage
19:18:39 [Pipeline] (Health & Monitoring)
19:18:39 [Pipeline] echo
19:18:39 ☺ Collecting basic health information...
19:18:40 [Pipeline] sh
19:18:40 + echo workspace disk usage:
19:18:40 Workspace disk usage:
19:18:40 + du -sh .
19:18:40 22M .
19:18:40 + echo
19:18:40
19:18:40 + echo system uptime:
19:18:40 System uptime:
19:18:40 + uptime
19:18:40 01:18:40 up 4:09, 0 user, load average: 1.04, 0.55, 0.41
19:18:40 [Pipeline] }
19:18:40 [Pipeline] // stage
19:18:40 [Pipeline] stage
19:18:40 [Pipeline] (Release)
19:18:41 [Pipeline] echo
19:18:41 [Pipeline] search branch for release: origin/main
19:18:41 [Pipeline] echo
19:18:41 ✅ Releasing build artifact - running on main branch.
19:18:41 [Pipeline] }
19:18:41 [Pipeline] // stage
19:18:41 [Pipeline] stage
19:18:41 [Pipeline] (Declarative: Post Actions)
19:18:41 [Pipeline] echo
19:18:41 ☻ Pipeline succeeded - Build #24
19:18:42 [Pipeline] emailext
19:18:42 Sending email to: jishnu94@gmail.com
19:18:43 [Pipeline] }
19:18:43 [Pipeline] // stage
19:18:43 [Pipeline] }
19:18:43 [Pipeline] // timestamps
19:18:43 [Pipeline] }
19:18:43 [Pipeline] // timeout
19:18:43 [Pipeline] }
19:18:44 [Pipeline] // withEnv
19:18:44 [Pipeline] }
19:18:44 [Pipeline] // withEnv
19:18:44 [Pipeline] }
19:18:44 [Pipeline] // node
19:18:44 [Pipeline] End of Pipeline
19:18:44 Finished: SUCCESS
```

Figure – Blue Ocean Pipeline With New Stage

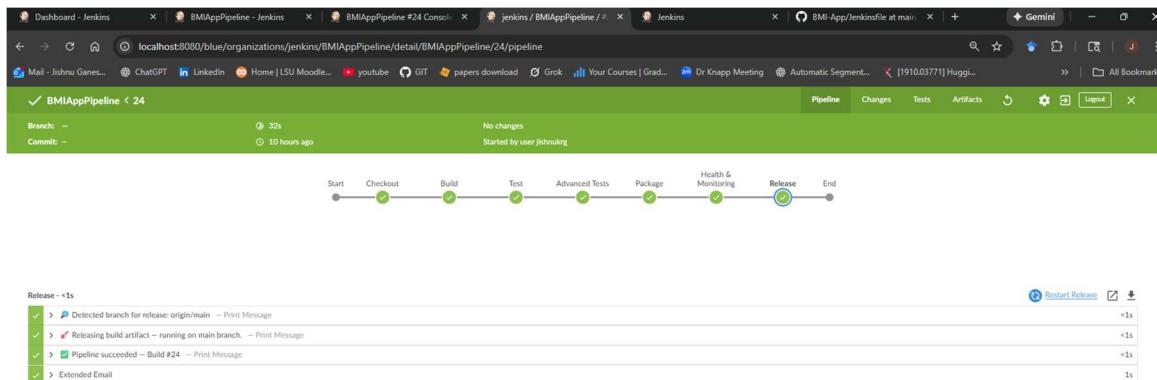


Figure – Pipeline Steps

Jenkins

Dashboard > BMIAppPipeline > #24 > Pipeline Steps

Step	Arguments	Status
Start of Pipeline - (25 sec in block)		
node - (21 sec in block)		
node block - (20 sec in block)		
stage - (2.1 sec in block)	Declarative: Checkout SCM	
stage block (Declarative: Checkout SCM) - (1.7 sec in block)		
checkout - (1.4 sec in self)		
withEnv - (1.7 sec in block)	GIT_BRANCH, GIT_COMMIT, GIT_PREVIOUS_COMMIT, GIT_PREVIOUS_SUCCESSFUL_COMMIT, GIT_URL	
withinEnv block - (1.7 sec in block)		
withEnv - (1.7 sec in block)	BRANCH_NAME	
withEnv block - (1 sec in block)		
timeout - (16 sec in block)		
timeout block - (16 sec in block)		
timestamps - (16 sec in block)		
timestamps block - (16 sec in block)		
stage - (1.6 sec in block)	Checkout	
stage block (Checkout) - (1.4 sec in block)		
checkout - (1 sec in self)		
echo - (59 ms in self)	Checked out branch: origin/main	
stage - (7.4 sec in block)	Build	

localhost:8080/job/BMIAppPipeline/24/flowGraphTable/

Dashboard > BMIAppPipeline > #24 > Pipeline Steps

stage block (Advanced tests) - (0.8 sec in block)		
echo - (0.11 sec in self)	Running enhanced tests...	
sh - (0.36 sec in self)	if [! -d "build"]; then echo "X Build folder missing!" exit 1 fi if [! -f "build/BMIApp.class"]; then echo "X BMIApp.class not found!" exit 1 fi echo "G All enhanced tests passed!"	
stage - (0.94 sec in block)	Package	
stage block (Package) - (0.64 sec in block)		
echo - (87 ms in self)	Packaging into TAR artifact...	
sh - (0.36 sec in self)	mkdir -p dist tar -cf dist/BMIApp-\$BUILD_NUMBER.tar build	
stage - (1 sec in block)	Health & Monitoring	
stage block (Health & Monitoring) - (0.63 sec in block)		
echo - (84 ms in self)	Collecting basic health information...	
sh - (0.38 sec in self)	echo "Workspace disk usage:" du -sh . echo "du command not available" echo "System uptime:" uptime echo "uptime command not available"	
stage - (0.83 sec in block)	Release	
stage block (Release) - (0.51 sec in block)		
echo - (0.15 sec in self)	Detected branch for release: origin/main	
echo - (65 ms in self)	Releasing build artifact — running on main branch.	
stage - (1.8 sec in block)	Declarative: Post Actions	
stage block (Declarative: Post Actions) - (1.7 sec in block)		
echo - (0.13 sec in self)	Pipeline succeeded — Build #24	
emailExt - (1.3 sec in self)		

Jenkins 2.504.2