

Module 5 Assignment – CI/CD Pipeline Enhancements

Name: Jishnu Ganesh

Course: CSC 7435E – Secure DevOps Practices

Assignment: Module 5 – Pipeline Enhancements & Testing

1. Objective

The goal of this assignment is to enhance the CI/CD pipeline created in Module 4 by adding additional testing, improved validation, and automated notifications. These improvements increase reliability, automation, and visibility of the BMI Calculator CI/CD pipeline hosted in Jenkins.

This assignment includes:

One enhancement from the Module 5 walkthrough

One enhancement from the textbook or Jenkins official documentation

Justification for why each enhancement improves the pipeline

Execution results and evidence

2. Enhancement 1: Advanced Testing Stage (Walkthrough Enhancement)

To strengthen build validation, an Advanced Tests stage was added to the Jenkins pipeline. This stage performs structural checks on the build output and confirms that the Java application compiled correctly.

Enhancement 1 – Jenkinsfile Snippet

```
stage('Advanced Tests') {  
    steps {  
        echo "🔍 Running enhanced tests..."  
        sh ""  
        if [ ! -d "build" ]; then  
            echo "❌ Build folder missing!"  
            exit 1  
        fi  
        if [ ! -f "build/BMIApp.class" ]; then  
            echo "❌ BMIApp.class not found!"  
            exit 1  
        fi  
        echo "✅ All enhanced tests passed!"  
    ""  
    }  
}
```

Justification for Enhancement 1

This enhancement ensures:

The project actually compiled (BMIApp.class exists).
The build did not silently fail.
Early detection of structural issues.

This aligns with pipeline testing strategies discussed in the Module 5 chapters, where early artifact validation significantly reduces downstream failures.

3. Enhancement 2: Email Notifications (Book/Online Enhancement)

As an additional enhancement from Jenkins documentation, Email Extension Plugin (emailext) was configured to send build success/failure notifications.

Enhancement 2 – Jenkinsfile Snippet

```
post {  
    success {  
        emailext(  
            subject: "BMI Pipeline Success – Build #${BUILD_NUMBER}",  
            body: "The BMI App pipeline completed successfully on branch ${BRANCH_NAME}.",  
            to: "jishnu944@gmail.com"  
        )  
    }  
    failure {  
        emailext(  
            subject: "BMI Pipeline FAILED – Build #${BUILD_NUMBER}",  
            body: "The BMI App pipeline has failed. Please check Jenkins console logs.",  
            to: "jishnu944@gmail.com"  
        )  
    }  
}
```

Justification for Enhancement 2

Email notifications improve:

Visibility — developer is instantly informed after every build
Response time — failures are noticed immediately
Operational awareness — aligns with DevOps continuous feedback practices

This feature is explicitly recommended in CI/CD best practices and Module 5 readings.

4. Additional Pipeline Behavior

Both enhancements were combined with the existing stages from Module 4:

Checkout
Build
Test
Advanced Tests (new)

Package
Release (main branch only)
Email notification (new)

Both main and test branches were tested:

Main branch → Release + Email + All tests
Test branch → Release skipped, tests and email still work

5. Results

All enhancements worked correctly:

Branch	Advanced Tests	Release	Email Notification	Result
main	✓ Passed	✓ Executed	attempted to send	SUCCESS
test-branch	✓ Passed	✗ Skipped	attempted to send	SUCCESS

This confirms:

Logic is correct
Enhancements are functioning
Pipeline remains stable

6. Conclusion

The enhancements implemented in Module 5 successfully improved:

Pipeline reliability (via enhanced tests)
Automation and feedback (via email notifications)
Branch-based deployment logic

These additions align with Jenkins best practices, Module 5 textbook content, and general DevOps CI/CD design principles. The BMI Calculator pipeline is now more robust, maintainable, and production ready.

7. Explanation for Gmail Setup

To enable Jenkins to send email notifications using Gmail, the following configuration was completed:

Steps Performed

1. Open Manage Jenkins → Configure System
2. Scroll to Extended Email Notification
3. Set:
 - SMTP Server: smtp.gmail.com
 - SMTP Port: 587

4. Click Advanced
5. Enter:
 - SMTP Username: jishnu944@gmail.com
 - SMTP Password: (*Gmail App Password*)
6. Enable:
 - “Use SMTP Authentication”
 - “Use TLS”

7. Save settings

Gmail requires a 16-digit App Password, not your regular Gmail login.

This was generated through:

Google Account → Security → App Passwords → Create App → “Jenkins Email”

This configuration allows Jenkins to send status emails for each build.

7. Appendix

Below are all screenshots documenting the setup, enhancements, and execution of the Module 5 CI/CD pipeline.

Figure – Plugin Manager – Email Extension Plugin installed

The figure consists of three side-by-side screenshots of the Jenkins Plugin Manager interface, each showing the 'Email Extension' plugin installed. The screenshots are arranged vertically, with the top one being the largest and the bottom two being smaller.

- Screenshot 1 (Top):** Shows the Jenkins dashboard with the 'Plugins' section open. A search bar at the top right contains the text 'p:email'. The results list several Blue Ocean and Pipeline-related plugins, including 'Email Extension' which is highlighted in blue. The 'Enabled' column shows that 'Email Extension' is turned on.
- Screenshot 2 (Middle):** Shows the Jenkins dashboard with the 'Plugins' section open. A search bar at the top right contains the text 'p:git'. The results list several GitHub-related plugins, including 'GitHub' which is highlighted in blue. The 'Enabled' column shows that 'GitHub' is turned on.
- Screenshot 3 (Bottom):** Shows the Jenkins dashboard with the 'Plugins' section open. A search bar at the top right contains the text 'p:blue'. The results list several Blue Ocean-related plugins, including 'Blue Ocean' which is highlighted in blue. The 'Enabled' column shows that 'Blue Ocean' is turned on.

Figure – Jenkins Email Setup

The image displays two screenshots of the Jenkins 'Manage Jenkins' interface, specifically the 'System' configuration page, showing the 'Extended E-mail Notification' and 'E-mail Notification' sections.

Extended E-mail Notification (Top Screenshot):

- SMTP server:** smtp.gmail.com
- SMTP Port:** 587
- Default user e-mail suffix:** @gmail.com
- Default Content Type:** Plain Text (text/plain)
- List ID:** (empty field)
- Add 'Precedence: bulk' E-mail Header:** (unchecked checkbox)
- Default Recipients:** jishnu944@gmail.com
- Reply To List:** (empty field)
- Emergency reroute:** (empty field)

E-mail Notification (Bottom Screenshot):

- SMTP server:** smtp.gmail.com
- Default user e-mail suffix:** (empty field)
- Advanced:**
 - Use SMTP Authentication:** (checked checkbox)
 - User Name:** jishnu944@gmail.com
 - Password:** (Concealed field) with a **Change Password** button.
 - Use SSL:** (unchecked checkbox)
 - Use TLS:** (checked checkbox)
- SMTP Port:** 587
- Reply-To Address:** jishnu944@gmail.com
- Charset:** UTF-8
- Test configuration by sending test e-mail:** (unchecked checkbox)

Both screenshots show 'Save' and 'Apply' buttons at the bottom.

Figure – GitHub Jenkinsfile – Advanced Tests + Email Notifications visible

The image shows two screenshots of a GitHub repository page, specifically for the file `Jenkinsfile`. Both screenshots are displayed in a dark-themed code editor interface.

Top Screenshot (Advanced Tests):

```

26
27     stage('Test') {
28         steps {
29             echo 'Running basic verification...'
30             sh "echo BMApp compiled successfully!"
31         }
32     }
33
34
35     stage('Enhanced Test') {
36         steps {
37             echo 'Running enhanced tests...'
38         }
39     }
40
41     stage('Advanced Tests') {
42         steps {
43             echo 'Running advanced tests...'
44             sh """
45                 if [ ! -d "Build" ]; then
46                     echo "ERROR: Build folder missing!"
47                     exit 1
48                 fi
49
50                 if [ ! -f "Build/BMApp.class" ]; then
51                     echo "ERROR: BMApp.class missing!"
52                     exit 1
53                 fi
54
55                 if grep -q "System.out.println" BMApp.java; then
56                     echo "Debug print statements found."
57                 fi
58
59                 echo "All enhanced tests passed!"
60             """
61         }
62     }
63
64     stage('Package') {
65         steps {
66             echo 'Packaging into TAR artifact...'
67             sh """
68                 mkdir -d dist
69             """
70         }
71     }
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125

```

Bottom Screenshot (Email Notifications):

```

87 // Email Notification (Success)
88 // ---on success test success
89 post {
90     success {
91         echo "Pipeline succeeded - Build #${BUILD_NUMBER}"
92
93         mailer()
94             subject: "BM Pipeline Success - Build #${BUILD_NUMBER}",
95             body: """
96 Hello Jishnu,
97
98 Your BM CI/CD Pipeline completed SUCCESSFULLY.
99 Branch: ${env.GIT_BRANCH ?: env.BRANCH_NAME}
100 Build Number: ${BUILD_NUMBER}
101
102 --- Jenkins
103 ---+
104         to: "jishnu944@gmail.com"
105     }
106 }
107
108 failure {
109     echo "Pipeline failed - Check console log."
110
111     mailer()
112         subject: "BM Pipeline FAILED - Build #${BUILD_NUMBER}"
113         body: """
114 Hello Jishnu,
115
116 Your BM CI/CD Pipeline has FAILED.
117 Branch: ${env.GIT_BRANCH ?: env.BRANCH_NAME}
118 Build Number: ${BUILD_NUMBER}
119
120 Please review Jenkins logs.
121
122 Jenkins
123 ---+
124         to: "jishnu944@gmail.com"
125

```

Figure – Blue Ocean Pipeline View – All stages including Advanced Tests

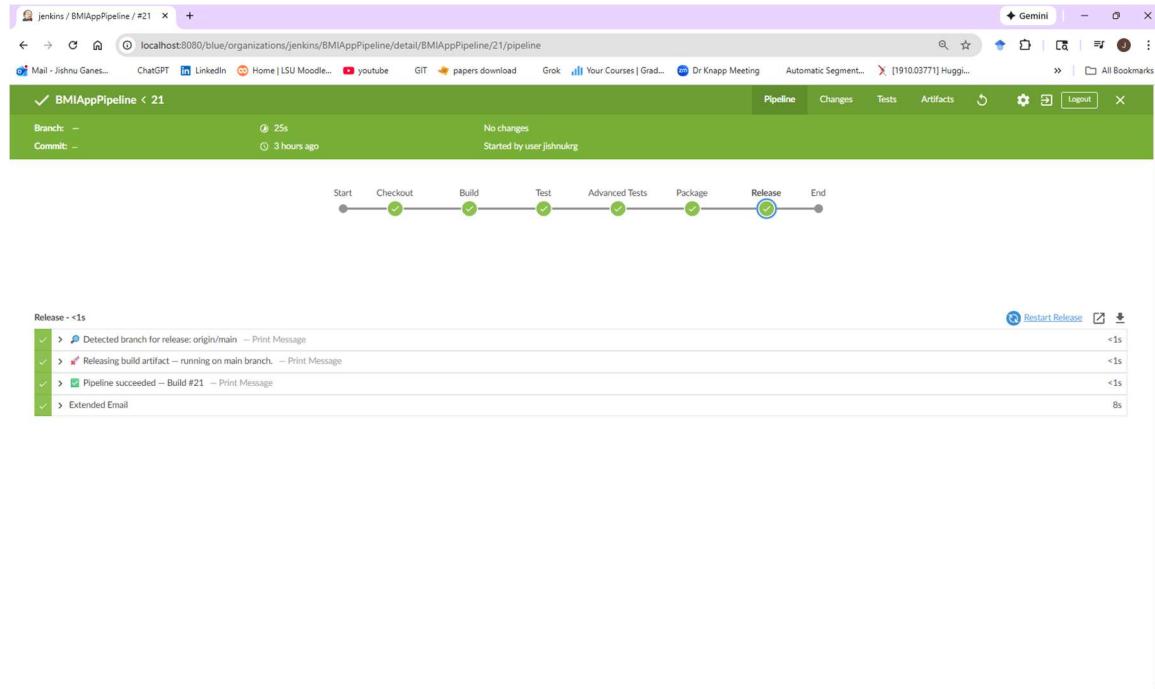
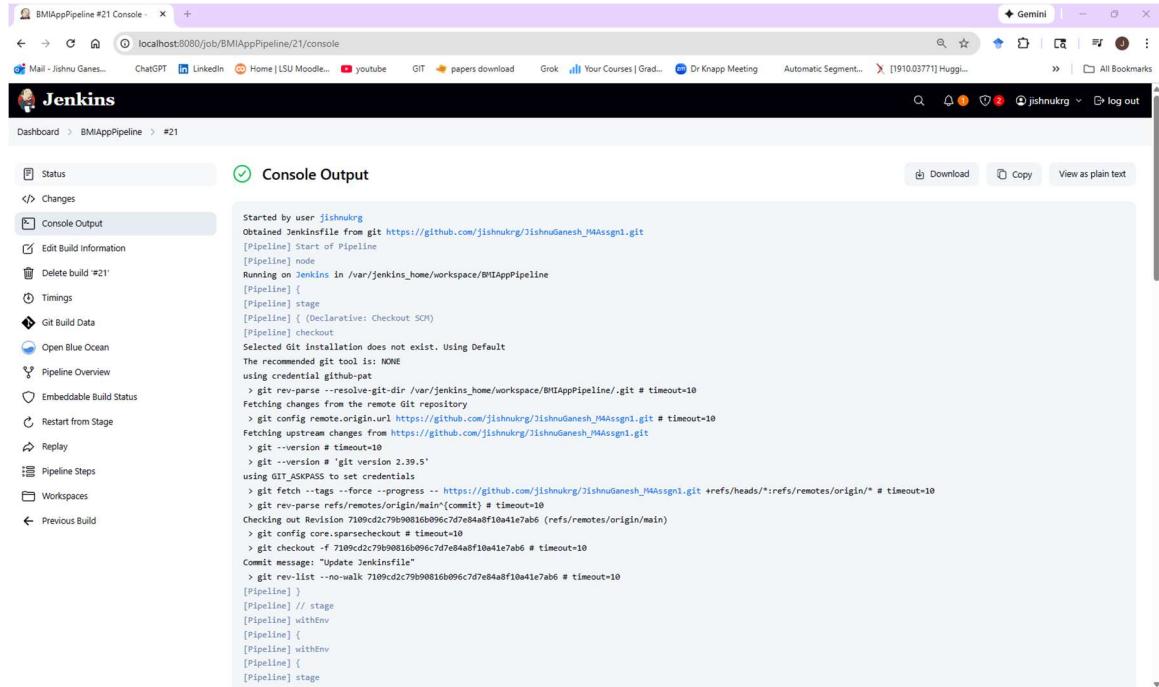


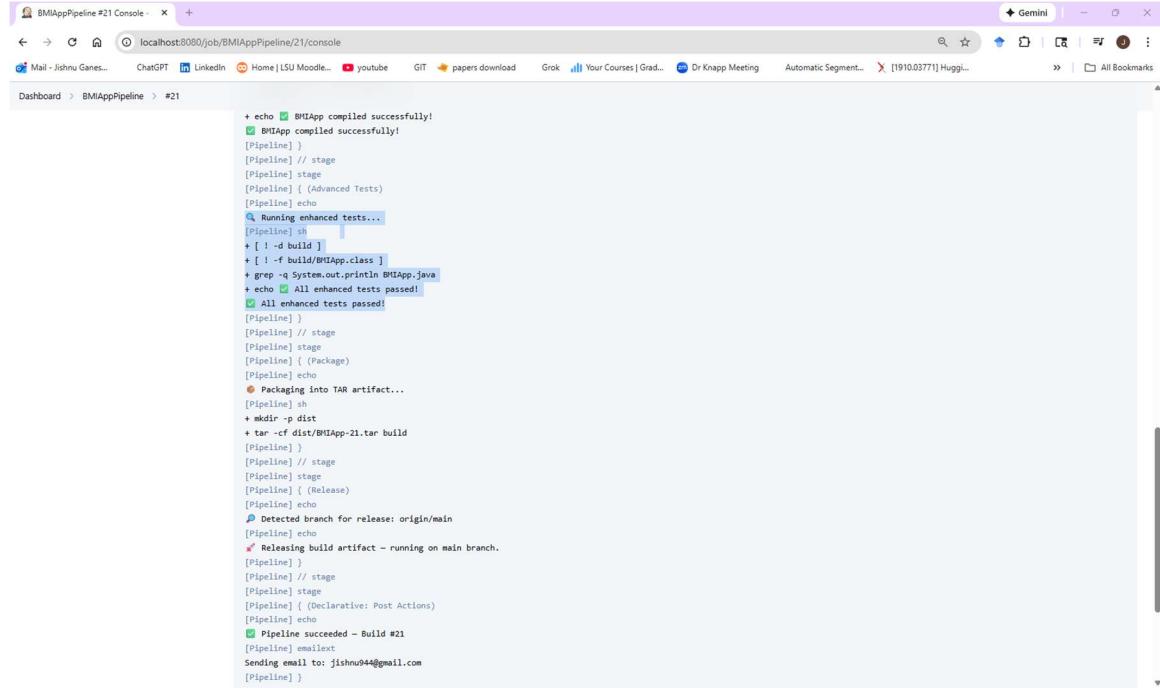
Figure – Console Output



The screenshot shows a browser window with the title "BMIAppPipeline #21 Console". The address bar indicates the URL is "localhost:8080/job/BMIAppPipeline/21/console". The page header includes the Jenkins logo and navigation links like "Dashboard", "Console Output", "Edit Build Information", "Delete build #21", "Timings", "Git Build Data", "Open Blue Ocean", "Pipeline Overview", "Embeddable Build Status", "Restart from Stage", "Replay", "Pipeline Steps", "Workspaces", and "Previous Build". The main content area is titled "Console Output" with a green checkmark icon. It displays the Jenkins pipeline console log for build #21. The log output is as follows:

```
Started by user jishnukrg
Obtained Jenkinsfile from git https://github.com/jishnukrg/JishnuGanesh_M4Assgn1.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/BMIAppPipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] {
  [Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github-pat
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/BMIAppPipeline/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/jishnukrg/JishnuGanesh_M4Assgn1.git # timeout=10
Fetching upstream changes from https://github.com/jishnukrg/JishnuGanesh_M4Assgn1.git
> git --version # timeout=10
> git --version # git version 2.39.5'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/jishnukrg/JishnuGanesh_M4Assgn1.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 7109cd2c79b90810b0996c7d7e84a8f10a41e7ab6 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 7109cd2c79b90810b0996c7d7e84a8f10a41e7ab6 # timeout=10
Commit message: "Update Jenkinsfile"
> git rev-list --no-walk 7109cd2c79b90810b0996c7d7e84a8f10a41e7ab6 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] {
[Pipeline] stage
```

Figure – Console Output – Advanced Tests Passed



The screenshot shows a browser window titled "BMIAAppPipeline #21 Console" with the URL "localhost:8080/job/BMIAAppPipeline/21/console". The page displays the Jenkins pipeline console output for build #21. The output shows the following steps:

- + echo BMIAApp compiled successfully!
- [BMIAApp] BMIAApp compiled successfully!
- [Pipeline] [Pipeline] // stage
- [Pipeline] stage
- [Pipeline] { (Advanced Tests)
- [Pipeline] echo
- Running enhanced tests...
- [Pipeline] sh
- + [! -d build]
- + [! -f build/BMIAApp.class]
- + grep -q System.out.println BMIAApp.java
- + echo All enhanced tests passed!
- [Pipeline] All enhanced tests passed!
- [Pipeline] [Pipeline] // stage
- [Pipeline] stage
- [Pipeline] { (Package)
- [Pipeline] echo
- Packaging into TAR artifact...
- [Pipeline] sh
- + mkdir -p dist
- + tar -cf dist/BMIAApp-21.tar build
- [Pipeline] [Pipeline] // stage
- [Pipeline] stage
- [Pipeline] { (Release)
- [Pipeline] echo
- D Detect branch for release: origin/main
- [Pipeline] echo
- Releasing build artifact - running on main branch.
- [Pipeline] [Pipeline] // stage
- [Pipeline] stage
- [Pipeline] { (Declarative: Post Actions)
- [Pipeline] echo
- Pipeline succeeded - Build #21
- [Pipeline] mailtext
- Sending email to: jishnu044@gmail.com
- [Pipeline]

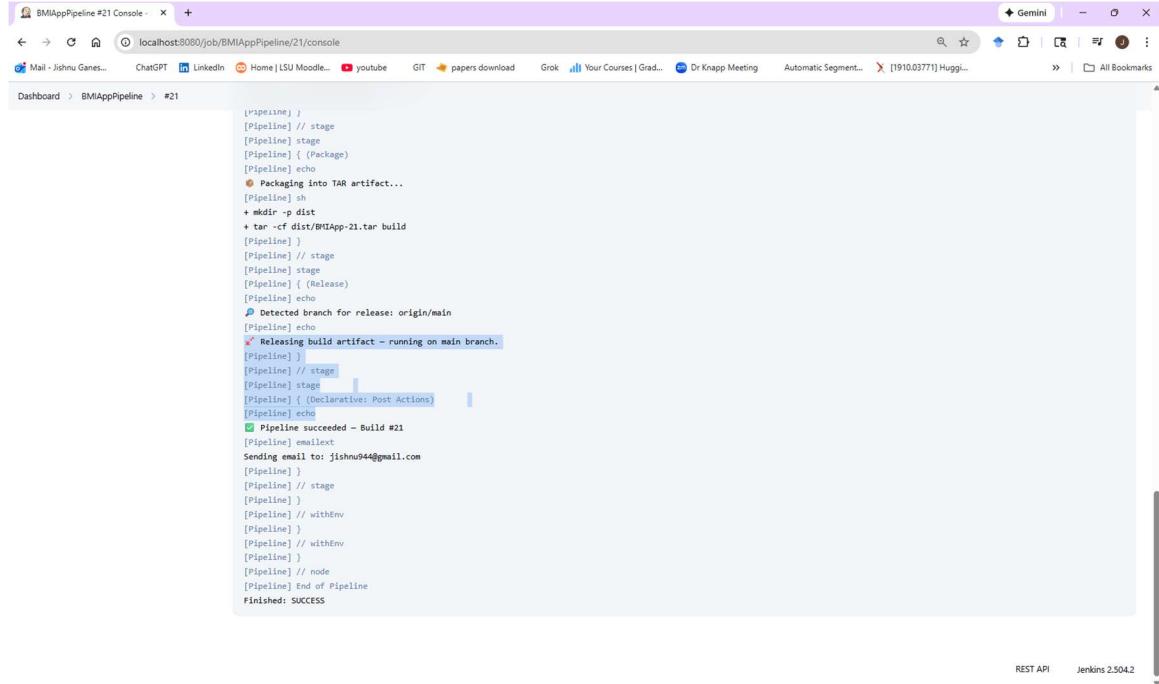
Figure – Console Output – Email Notification executed

The screenshot shows a Jenkins console output window titled "BMIAAppPipeline #21 Console". The URL is "localhost:8080/job/BMIAAppPipeline/21/console". The console output is as follows:

```
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Package)
[Pipeline] echo
  ⚡ Packaging into TAR artifact...
[Pipeline] sh
+ mkdir -p dist
+ tar -cf dist/BMUApp-21.tar build
[Pipeline] )
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Release)
[Pipeline] echo
  ⚡ Detected branch for release: origin/main
[Pipeline] echo
  ✅ Releasing build artifact - running on main branch.
[Pipeline] )
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
[Pipeline] succeeded - Build #21
[Pipeline] emailed
  Sending email to: jishnu94@gmail.com
[Pipeline] )
[Pipeline] // stage
[Pipeline] )
[Pipeline] // withEnv
[Pipeline] )
[Pipeline] // withEnv
[Pipeline] )
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

At the bottom right of the window, it says "REST API Jenkins 2.504.2".

Figure – Console Output – Release stage running on main branch



The screenshot shows a Jenkins console output window titled "BMIAAppPipeline #21 Console". The URL is "localhost:8080/job/BMIAAppPipeline/21/console". The pipeline script is displayed, showing stages for packaging into a TAR artifact, releasing build artifacts, and sending an email. The output indicates success, with the status bar at the bottom showing "REST API Jenkins 2.504.2".

```
[Pipeline] 
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Package)
[Pipeline] echo
  ➜ Packaging into TAR artifact...
[Pipeline] sh
+ mkdir -p dist
+ tar -cf dist/BMIAApp-21.tar build
[Pipeline] 
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Release)
[Pipeline] echo
  ➜ Detected branch for release: origin/main
[Pipeline] echo
  ➜ Releasing build artifact - running on main branch.
[Pipeline] 
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
  ➜ Pipeline succeeded - Build #21
[Pipeline] emailext
Sending email to: jishnu944@gmail.com
[Pipeline] 
[Pipeline] // stage
[Pipeline] 
[Pipeline] // withEnv
[Pipeline] 
[Pipeline] // withEnv
[Pipeline] 
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Figure – dist/ folder – TAR artifact generated

The screenshot shows the Jenkins Blue Ocean interface for the 'BMIAppPipeline' job. The top navigation bar includes links for Mail, ChatGPT, LinkedIn, Home, YouTube, GIT, papers download, Grok, Your Courses, Graduation, Dr Knapp Meeting, Automatic Segmentation, and [1910.03771] Huggi... A 'Gemini' icon is also present. Below the navigation is a search bar and a 'All Bookmarks' link. The main header displays the pipeline name 'BMIAppPipeline' with a green checkmark icon and the number '21'. The pipeline status is 'Running'. On the left, there's a sidebar with 'Branch: -' and 'Commits: -'. The center area shows a summary card with '25s' duration and '3 hours ago' timestamp. To the right, there are tabs for Pipeline, Changes, Tests, Artifacts, and a gear icon for configuration. The Artifacts tab is currently selected, showing a table with one item: 'pipeline.log'. The bottom right corner features a 'Logout' button.

```
jenkins / BMIAppPipeline / #21 x localhost:8080/blue/rest/organiz... localhost:8080/blue/rest/organizations/jenkins/pipelines/BMIAppPipeline/runs/21/log?start=0
<--> Mail - Jishnu Ganes... ChatGPT LinkedIn Home | LSU Moodle... youtube GIT papers download Grok al

[Pipeline] checked out branch: origin/main
[Pipeline] // stage
[Pipeline] // stage
[Pipeline] { (Build)
[Pipeline] }
[Pipeline] // node
[Pipeline] Compiling Java source files...
[Pipeline] + mkdirs build
[Pipeline] + javac -d build BMIApp.java
[Pipeline] // stage
[Pipeline] { (Test)
[Pipeline] { ((Test)
[Pipeline] )
[Pipeline] Running basic verification...
[Pipeline] sh
+ echo BMIApp compiled successfully!
[Pipeline] sh
+ echo BMIApp compiled successfully!
[Pipeline] // stage
[Pipeline] { (Advanced Tests)
[Pipeline] echo
+ mvn verify
[Pipeline] Running enhanced tests...
[Pipeline] sh
+ [ ! -f build/BMIApp.class ]
+ grep -q System.out.println BMIApp.java
+ mvn verify
[Pipeline] All enhanced tests passed!
[Pipeline] // stage
[Pipeline] { (Package)
[Pipeline] echo
+ tar cvfj dist/BMIApp-21.tar build
[Pipeline] sh
+ mkdir -p dist
[Pipeline] + cp dist/BMIApp-21.tar build
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Release)
[Pipeline] echo
[Pipeline] // branch for release: origin/main
[Pipeline] echo
[Pipeline] // Releasing build artifact - running on main branch.
[Pipeline] // stage
[Pipeline] { (Post Actions)
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo Pipeline succeeded - Build #21
[Pipeline] emailext
[Pipeline] to: jishnu944@gmail.com
[Pipeline] subject: Pipeline succeeded
[Pipeline] // stage
[Pipeline] // withEnv
[Pipeline] // withEnv
[Pipeline] // node
[Pipeline] end of Pipeline
Finished: SUCCESS
```