

Module 4 Assignment 2

Name: Jishnu Ganesh

Course: CSC 7435E – Secure DevOps Practices

Assignment: Module 4 Assignment 2 – Jenkins CI/CD Pipeline

GitHub Repository URL: https://github.com/jishnukrg/JishnuGanesh_M4Assgn1

Jenkins Job Name: BMIAppPipeline

1. Objective

The purpose of this assignment is to implement a CI/CD pipeline using Jenkins and GitHub, replicating the course walkthrough and customizing it for my BMI Calculator app. The goal is to automate code integration, building, testing, packaging, and conditional release processes, ensuring secure and efficient continuous delivery.

2. Jenkins Setup

Steps completed to configure Jenkins:

1. Installed required plugins: Git, Pipeline, and Blue Ocean.
2. Added GitHub Personal Access Token (PAT) under credentials named 'github-pat'.
3. Created a pipeline job named 'BMIAppPipeline' and linked it to my GitHub repository.
4. Configured the branch specifier to '*/main' and ensured the Jenkinsfile was at the repository root.

3. Pipeline Configuration

The Jenkinsfile automates the build, test, and packaging process for the Java BMI application. A conditional release stage was added to ensure releases occur only from the 'main' branch.

4. Jenkinsfile Overview

Stages in the Jenkinsfile include:

- Checkout – Pulls the code from GitHub.
- Build – Compiles Java source files.
- Test – Runs basic verification checks.
- Package – Packages compiled files into a '.tar' artifact.
- Release – Executes only when the branch is 'main'.

5. Modification Description (Required Customization)

The modification involved adding a conditional 'Release' stage that only executes when the build runs on the 'main' branch. This ensures controlled, production-only deployments. The condition uses Jenkins environment variables as shown below:

- when {
 expression { env.GIT_BRANCH?.contains('main') }
}

This customization was verified by testing both 'main' and 'test-branch'. The main branch successfully triggered a release, while the test branch correctly skipped it.

6. Pipeline Execution Evidence

Pipeline runs were tested under two scenarios:

- Main Branch: Release stage executed successfully.
- Test Branch: Release stage skipped as expected.

The results confirmed that the pipeline logic and automation were implemented correctly.

7. Results and Observations

Branch	Release Stage Behavior	Result
main	<input checked="" type="checkbox"/> Executed successfully	Artifact released and pipeline succeeded
test-branch	<input type="radio"/> Skipped as expected	No release, build successful

8. Conclusion

The CI/CD pipeline was implemented and customized successfully using Jenkins and GitHub integration. It automates building, testing, packaging, and conditional deployment for the BMI Calculator app. The modification ensures that only the 'main' branch can trigger releases, promoting stable and controlled deployments. Testing across multiple branches validated the functionality, meeting all assignment objectives and ensuring readiness for future CI/CD enhancements.

10. Appendix – Screenshots

Below are all screenshots documenting the setup and execution of the Jenkins pipeline. Each figure includes a short caption explaining its purpose.

Figures: Jenkins Plugin Manager – Installed Git, Pipeline, and Blue Ocean.

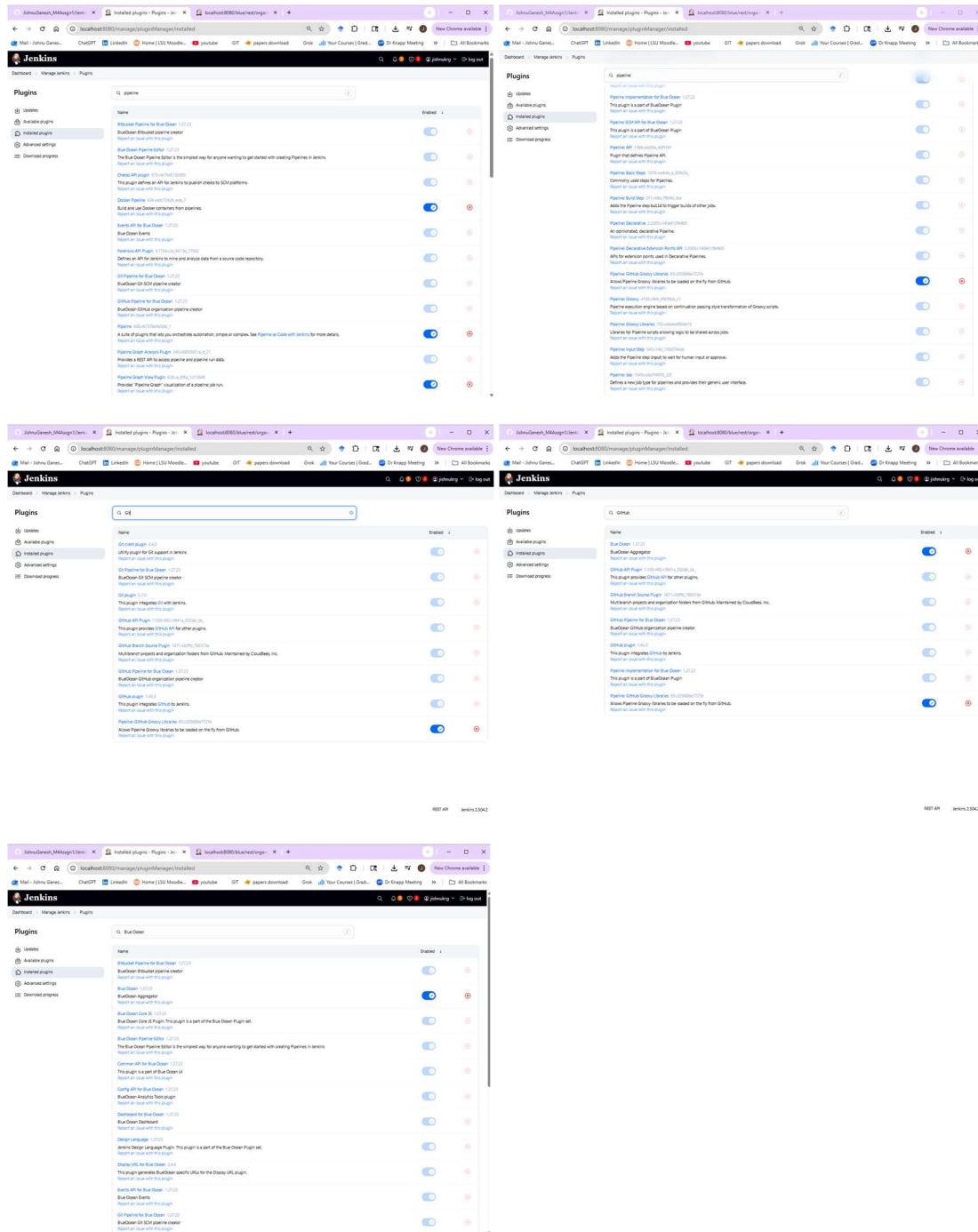


Figure: GitHub Credentials Setup – Personal Access Token added as 'github-pat'.

The screenshot shows a browser window with three tabs open: 'JishnuGanesh_M4Assgn1/Jenkins', 'System > Global credentials (unrestricted)', and 'localhost:8080/blue/rest/organ...'. The main content area is the Jenkins 'Global credentials (unrestricted)' page. It displays a table with one row of data:

ID	Name	Kind	Description
github-pat	jishnukrg/*****	Username with password	

Below the table, there are icons for 'S' (Small), 'M' (Medium), and 'L' (Large) to change the table size. The top navigation bar includes links for Mail, ChatGPT, LinkedIn, Home | LSU Moodle..., youtube, GIT, papers download, Grok, Your Courses | Grad..., Dr Knapp Meeting, and All Bookmarks. The user 'jishnukrg' is logged in.

Figure: Jenkins Job Configuration – Repository URL, branch, and Jenkinsfile path set.

The figure consists of two screenshots of the Jenkins job configuration interface, specifically focusing on the 'Pipeline' section.

Screenshot 1 (Top): Pipeline Configuration - Triggers Tab

- Triggers:**
 - Build after other projects are built
 - Build periodically
 - GitHub hook trigger for GitSCM polling
 - Poll SCM
 - Schedule
- Pipeline:**
 - Definition: Pipeline script from SCM
 - Source: SCM (Git)
 - Repositories:
 - Repository URL: `https://github.com/jishnukr/jishnuGanesh_M4Assgn1.git`
 - Credentials: `jishnukr*****`

Screenshot 2 (Bottom): Pipeline Configuration - Pipeline Syntax Tab

- Repositories:**
 - Repository URL: `https://github.com/jishnukr/jishnuGanesh_M4Assgn1.git`
 - Credentials: `jishnukr*****`
- Branches to build:**
 - Branch Specifier (blank for 'any'): `*/*main`
- Script Path:** `Jenkinsfile`
- Advanced:**
 - Lightweight checkout

Figure: Blue Ocean – Test Branch showing Release skipped (gray).

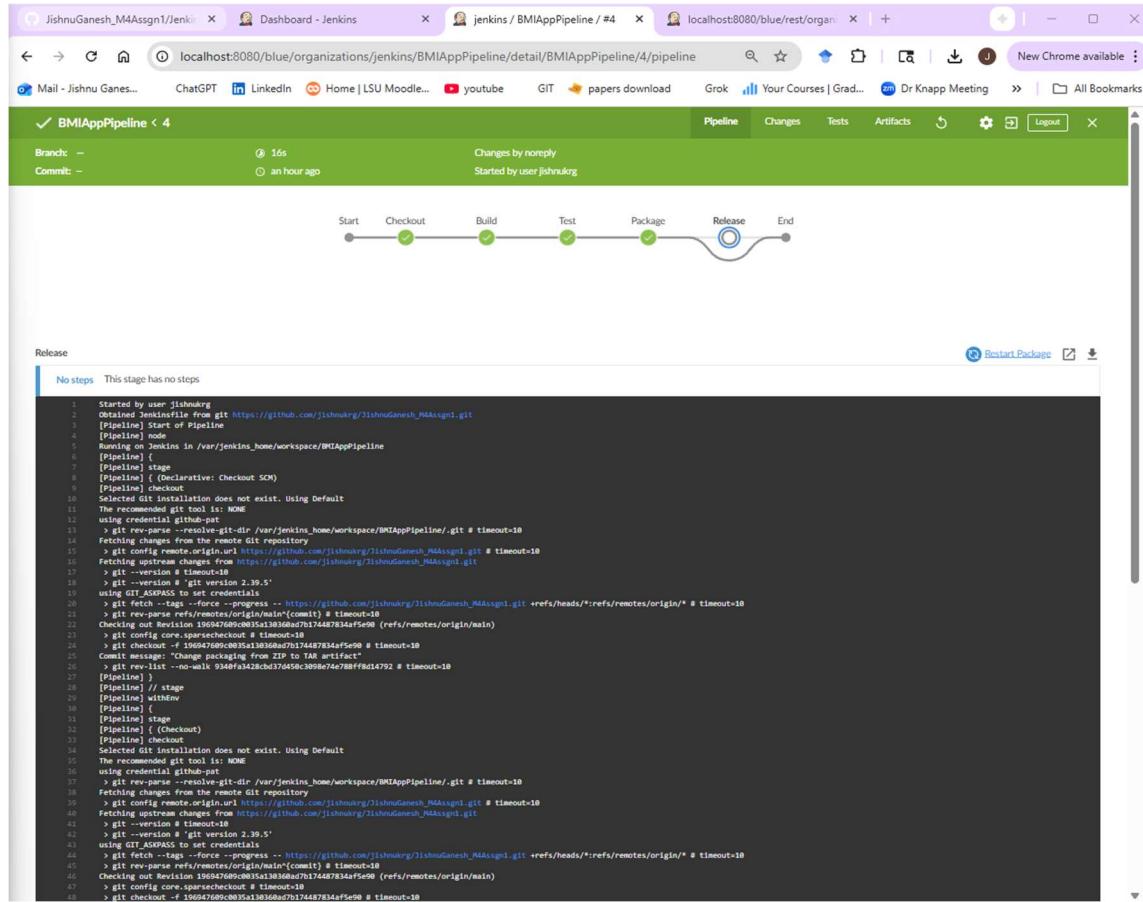


Figure: Blue Ocean – Main Branch showing all stages green including Release.

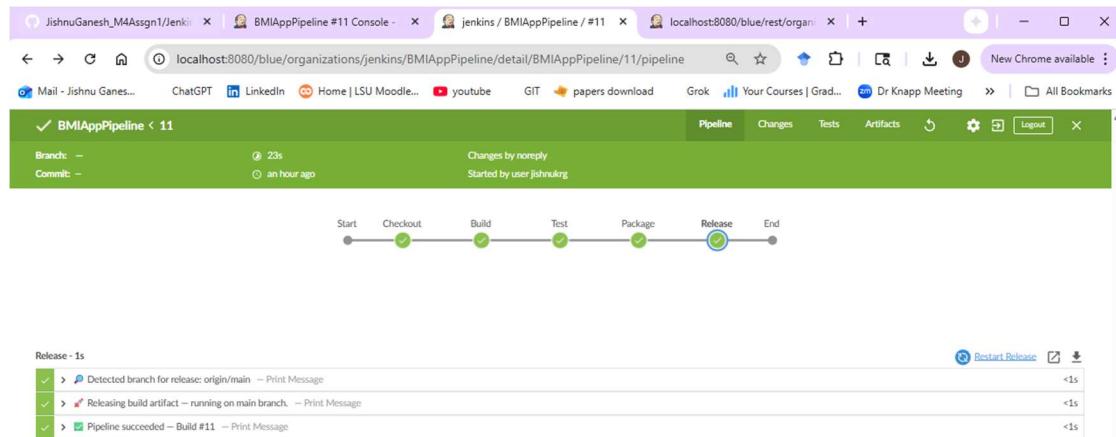
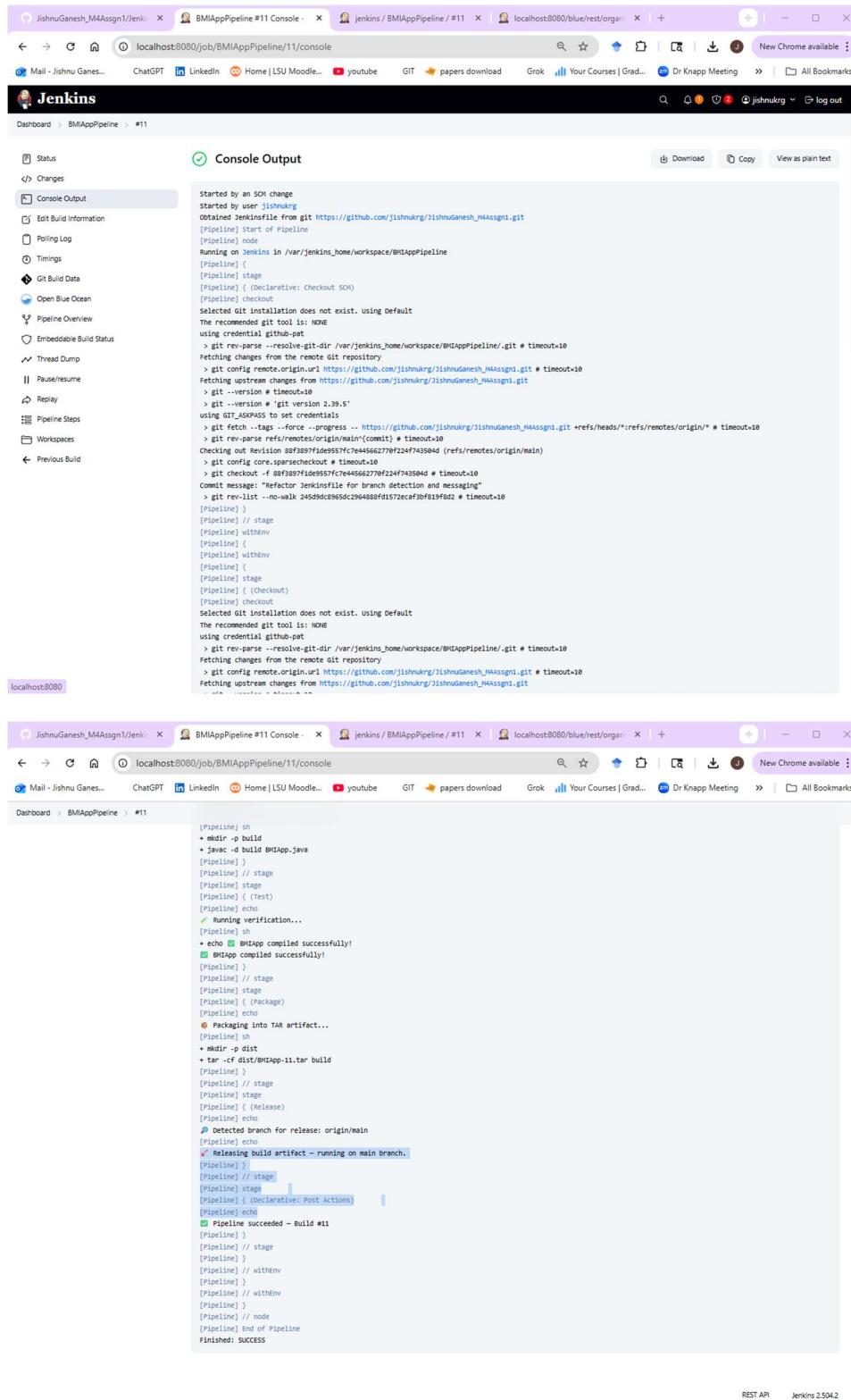


Figure: Console Output – Main Branch successful run with '🚀 Releasing build artifact'.



```

JishnuGanesh_M4Assgn1/jenkins | BMIAppPipeline #11 Console | jenkins / BMIAppPipeline / #11 | localhost:8080/blue/rest/organizations/jenkins | + New Chrome available | jishnukrgrg | log out
Dashboard > BMIAppPipeline > #11

Status | Changes | Console Output | Edit Build Information | Polling Log | Timings | Git Build Data | Open Blue Ocean | Pipeline Overview | Embeddable Build Status | Thread Dump | Pause/resume | Replay | Pipeline Steps | Workspaces | Previous Build

Console Output
Started by an SCM change
Started by user jishnukrgrg
Obtained Jenkinsfile from git https://github.com/jishnukrgrg/jishnuganesh_M4Assgn1.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/BMIAppPipeline
[Pipeline] [
[Pipeline] stage
[Pipeline] {
  (Declarative: Checkout SCM)
[Pipeline] }
[Pipeline] }
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/jishnukrgrg/jishnuganesh_M4Assgn1.git # timeout=10
Fetching upstream changes from https://github.com/jishnukrgrg/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/jishnukrgrg/jishnuganesh_M4Assgn1.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main{commit} # timeout=10
Checking out Revision 7f0e4456270f224f74598d (refs/remotes/origin/main)
> git clone -q --no-checkout --branch main --single-branch & git checkout -f 8ff7a0397f1de9557f7c7a456277ff224f74598d # timeout=10
Commit message: "Refactor Jenkinsfile for branch detection and messaging"
> git rev-list --no-walk 2456dc8956cc294408fd1572ccaf3fb619fb02 # timeout=10
[Pipeline] [
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] [
[Pipeline] withEnv
[Pipeline] [
[Pipeline] stage
[Pipeline] {
  (Checkout)
[Pipeline] checkout
[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/jishnukrgrg/jishnuganesh_M4Assgn1.git # timeout=10
Fetching upstream changes from https://github.com/jishnukrgrg/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/jishnukrgrg/jishnuganesh_M4Assgn1.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main{commit} # timeout=10
Checking out Revision 7f0e4456270f224f74598d (refs/remotes/origin/main)
> git clone -q --no-checkout --branch main --single-branch & git checkout -f 8ff7a0397f1de9557f7c7a456277ff224f74598d # timeout=10
Commit message: "Refactor Jenkinsfile for branch detection and messaging"
> git rev-list --no-walk 2456dc8956cc294408fd1572ccaf3fb619fb02 # timeout=10
[Pipeline] [
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] [
[Pipeline] withEnv
[Pipeline] [
[Pipeline] stage
[Pipeline] {
  (Test)
[Pipeline] echo
[Pipeline] echo
Running verification...
[Pipeline] sh
+ echo BMIAp compiled successfully!
BMIAp compiled successfully!
[Pipeline] [
[Pipeline] // stage
[Pipeline] stage
[Pipeline] {
  (Package)
[Pipeline] echo
Packaging into TAR artifact...
[Pipeline] sh
+ mkdir -p dist
+ tar -cf dist/BMIAp-11.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 #11
[Pipeline] [
[Pipeline] // stage
[Pipeline] [
[Pipeline] // withEnv
[Pipeline] [
[Pipeline] // withEnv
[Pipeline] [
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

Figure: Console Output – Test Branch skipped Release stage confirmation.

JishnuGanesh_M4Assgn1/jenkins Jenkins Pipeline #4 Console -> localhost:8080/blue/rest/organizations/jenkins

Mail - Jishnu Ganesh... ChatGPT LinkedIn Home | LSU Moodle... youtube GIT papers download Grok Your Courses | Grad... Dr Knapp Meeting All Bookmarks

Console Output

Started by user jishnukrug
Obtained Jenkinsfile from git https://github.com/jishnukrug/jishnuGanesh_M4Assgn1.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/BMIAppPipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] { (Declarative: Post Actions)
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/jishnukrug/jishnuGanesh_M4Assgn1.git # timeout=10
Fetching upstream changes from https://github.com/jishnukrug/jishnuGanesh_M4Assgn1.git
git -version # timeout=10
git -version # 'git' version 2.39.5
using GIT_ASKPASS to set credentials
git -get-url https://github.com/jishnukrug/jishnuGanesh_M4Assgn1.git +refs/heads/*:refs/remotes/origin/* # timeout=10
git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 19e047d09c0815a130360ba07b17447873aafe5e#(refs/remotes/origin/main)
git config core.sparsecheckout # timeout=10
git config core.sparsecheck
Commit message: "Change packaging from ZIP to TAR artifact"
git rev-list --no-walk 934fffa3428cb3d37d458c3098e74e78ff8d14792 # timeout=10
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Checkout)
[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/jishnukrug/jishnuGanesh_M4Assgn1.git # timeout=10
Fetching upstream changes from https://github.com/jishnukrug/jishnuGanesh_M4Assgn1.git
git -version # timeout=10
git -version # 'git' version 2.39.5
using GIT_ASKPASS to set credentials

JishnuGanesh_M4Assgn1/jenkins Jenkins Pipeline #4 Console -> localhost:8080/blue/rest/organizations/jenkins

Mail - Jishnu Ganesh... ChatGPT LinkedIn Home | LSU Moodle... youtube GIT papers download Grok Your Courses | Grad... Dr Knapp Meeting All Bookmarks

Dashboard > BMIAppPipeline > #4

```
[Pipeline] stage  
[Pipeline] { (Build)  
[Pipeline] echo  
Compiling Java source files...  
[Pipeline] sh  
+ mkdir -p build  
+ javac -d build BMIApp.java  
[Pipeline] // stage  
[Pipeline] stage  
[Pipeline] { (Test)  
[Pipeline] echo  
Running simple verification...  
[Pipeline] sh  
+ echo BMIApp compiled successfully!  
+ echo BMIApp compiled successfully!  
[Pipeline] // stage  
[Pipeline] stage  
[Pipeline] { (Package)  
[Pipeline] echo  
Packaging into TAR artifact...  
[Pipeline] sh  
+ tar -cf dist/BMIApp-4.tar build  
[Pipeline] // stage  
[Pipeline] stage  
[Pipeline] { (Release)  
Stage "Release" skipped due to when conditional  
[Pipeline] getContext  
[Pipeline] // stage  
[Pipeline] // stage  
[Pipeline] stage  
[Pipeline] { (Declarative: Post Actions)  
[Pipeline] echo  
Pipeline succeeded - build #4  
[Pipeline] // stage  
[Pipeline] // stage  
[Pipeline] // withEnv  
[Pipeline] // node  
[Pipeline] End of Pipeline
```

Finished: SUCCESS

Figure: Dist Folder – Generated TAR artifact after successful pipeline run.

