

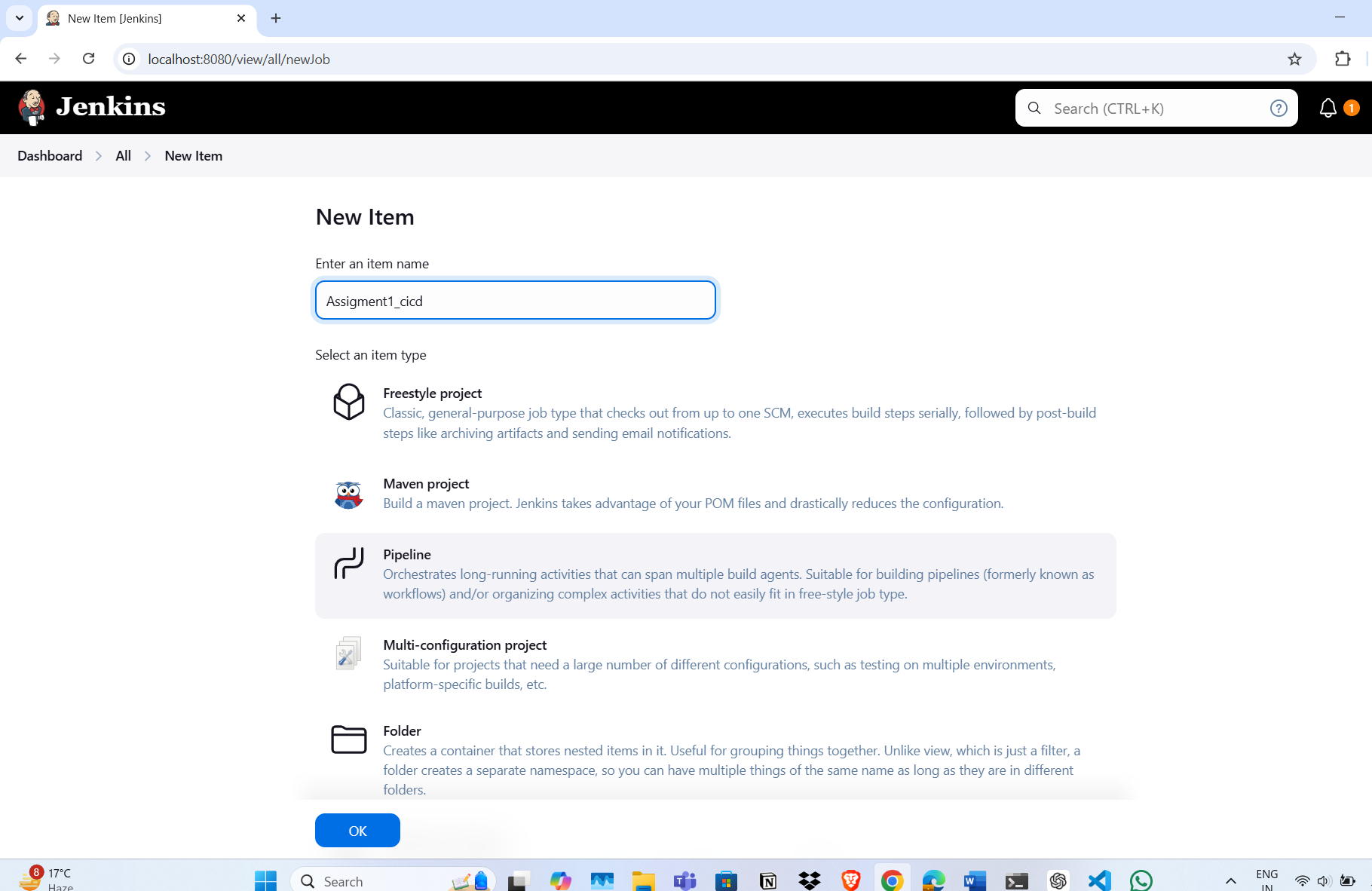
**LAB FILE: CICD PIPELINE SECURITY**

Name – Atrayee Pathak

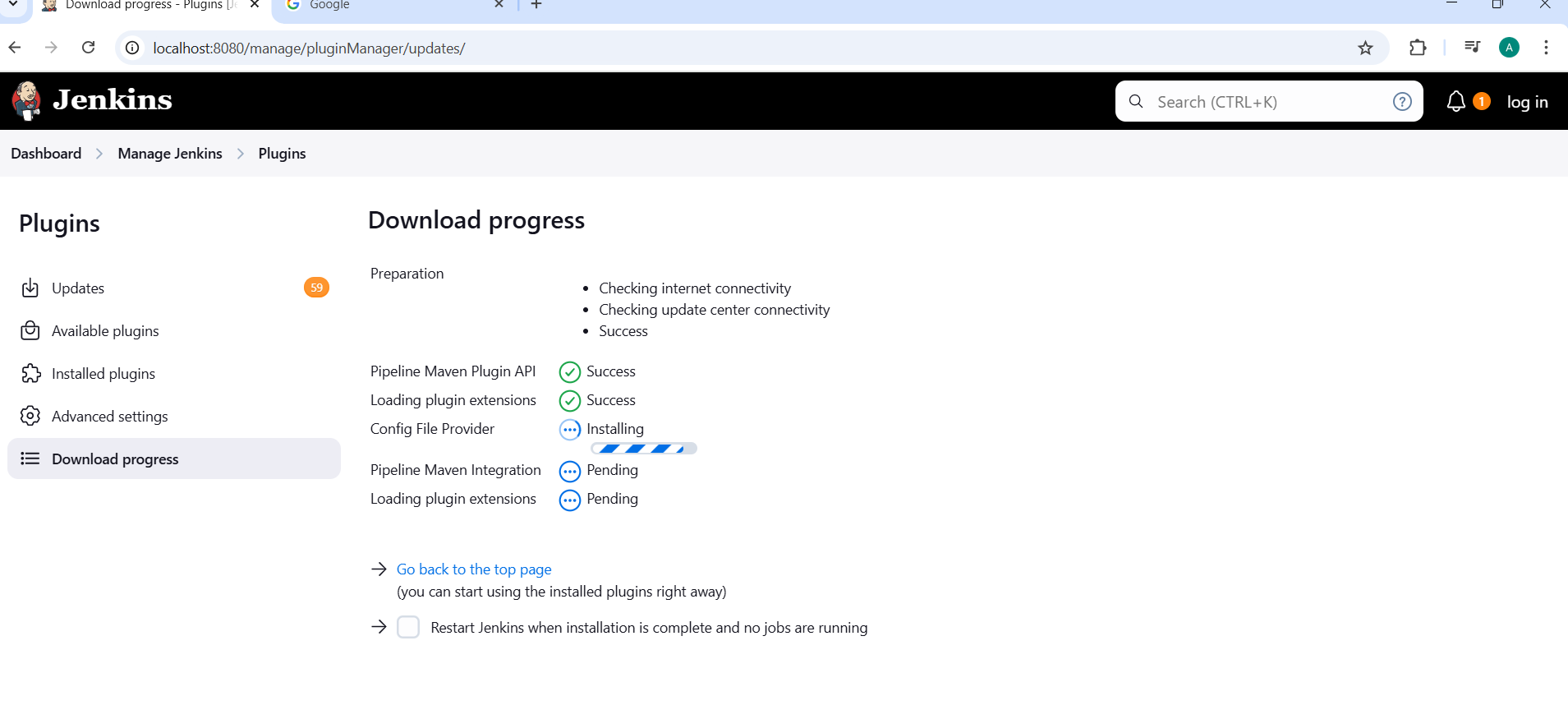
Sap id – 500105632

Batch – B1 (DevOps)

Write the definition of the following steps with the simple implementation in the Jenkins pipeline. Also write the code which you have implemented and attach a screenshot of the output console.



**Installing this plugins**

****

Started by user unknown or anonymous

[Pipeline] Start of Pipeline

[Pipeline] node

Running on Jenkins

in /var/lib/jenkins/workspace/Groovy

[Pipeline] {

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage (Windows Batch Script)

[Pipeline] script

[Pipeline] {

[Pipeline] isUnix

[Pipeline] echo

Skipping Windows Batch command on Unix system

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage (Build Job)

[Pipeline] script

[Pipeline] {

[Pipeline] build

[Pipeline] echo

Warning: Build job 'SomeJobName' not found. Skipping this step.

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage (Checkout from Version Control)

[Pipeline] git

The recommended git tool is: NONE

No credentials specified

> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Groovy/.git # timeout=10

ERROR: Workspace has a .git repository, but it appears to be corrupt.

hudson.plugins.git.GitException: Command "git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Groovy/.git" returned status code 128:

stderr: fatal: not a gitdir '/var/lib/jenkins/workspace/Groovy/.git'

Cloning the remote Git repository

Cloning repository https://github.com/Jahnavi-Anand/jenkins\_pipeline.git

> git init /var/lib/jenkins/workspace/Groovy # timeout=10

> git fetch --tags --force --progress -- https://github.com/Jahnavi-Anand/jenkins\_pipeline.git +refs/heads/\*:refs/remotes/origin/\* # timeout=10

Checking out Revision e1a5f0f2e97a5ce84094efbac2b2adb84ef13a01 (refs/remotes/origin/main)

Commit message: "Update pom.xml"

[Pipeline] stage (Delete Workspace)

[Pipeline] script

[Pipeline] {

[Pipeline] isUnix

[Pipeline] sh

+ find . -type f ! -name pom.xml -delete

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage (Change Directory)

[Pipeline] dir

Running in /var/lib/jenkins/workspace/Groovy/subdirectory

[Pipeline] echo

Now in subdirectory

[Pipeline] stage (Print Message)

[Pipeline] echo

Executing Jenkins pipeline!

[Pipeline] stage (Check File Existence)

[Pipeline] script

[Pipeline] {

[Pipeline] fileExists

[Pipeline] echo

File does not exist!

[Pipeline] }

[Pipeline] // script

[Pipeline] stage (Wait for Input)

[Pipeline] script

[Pipeline] {

[Pipeline] input

Input requested

[Pipeline] echo

User chose: true

[Pipeline] }

[Pipeline] // script

[Pipeline] stage (Detect OS)

[Pipeline] script

[Pipeline] {

[Pipeline] isUnix

[Pipeline] echo

Running on a Unix-like system

[Pipeline] }

[Pipeline] // script

[Pipeline] stage (Print Current Directory)

[Pipeline] script

[Pipeline] {

[Pipeline] pwd

[Pipeline] echo

Current directory: /var/lib/jenkins/workspace/Groovy

[Pipeline] }

[Pipeline] // script

[Pipeline] stage (Read & Write File)

[Pipeline] script

[Pipeline] {

[Pipeline] writeFile

[Pipeline] readFile

[Pipeline] echo

File content: This is a test file

[Pipeline] }

[Pipeline] // script

[Pipeline] stage (Shell Script Execution)

[Pipeline] script

[Pipeline] {

[Pipeline] isUnix

[Pipeline] sh

+ echo Running shell command

Running shell command

[Pipeline] }

[Pipeline] // script

[Pipeline] stage (Pause Execution)

[Pipeline] sleep

Sleeping for 5 sec

[Pipeline] stage (Stash Files)

[Pipeline] stash

Stashed 1 file(s)

[Pipeline] stage (Check Workspace)

[Pipeline] script

[Pipeline] {

[Pipeline] isUnix

[Pipeline] sh

+ ls -l

total 8

-rw-r--r-- 1 jenkins jenkins 541 Feb 26 03:06 pom.xml

-rw-r--r-- 1 jenkins jenkins 19 Feb 26 03:06 somefile.txt

[Pipeline] }

[Pipeline] // script

[Pipeline] stage (Build)

[Pipeline] script

[Pipeline] {

[Pipeline] isUnix

[Pipeline] sh

+ mvn clean package

[INFO] Scanning for projects...

[INFO] Building sample-project 1.0-SNAPSHOT

[INFO] --- clean:3.2.0:clean (default-clean) @ sample-project ---

[INFO] --- compiler:3.13.0:compile (default-compile) @ sample-project ---

[INFO] No sources to compile

[INFO] --- jar:3.4.1:jar (default-jar) @ sample-project ---

[WARNING] JAR will be empty - no content was marked for inclusion!

[INFO] Building jar: /var/lib/jenkins/workspace/Groovy/target/sample-project-1.0-SNAPSHOT.jar

[INFO] BUILD SUCCESS

[Pipeline] }

[Pipeline] // script

[Pipeline] stage (Archive Artifacts)

[Pipeline] archiveArtifacts

Archiving artifacts

Recording fingerprints

[Pipeline] stage (Check Build Output)

[Pipeline] script

[Pipeline] {

[Pipeline] isUnix

[Pipeline] sh

+ ls -l target

total 8

drwxr-xr-x 2 jenkins jenkins 4096 Feb 26 03:06 maven-archiver

-rw-r--r-- 1 jenkins jenkins 1388 Feb 26 03:06 sample-project-1.0-SNAPSHOT.jar

[Pipeline] }

[Pipeline] // script

[Pipeline] stage (Set Custom Environment Variable)

[Pipeline] withEnv

[Pipeline] {

[Pipeline] script

[Pipeline] {

[Pipeline] isUnix

[Pipeline] sh

+ echo New Value

New Value

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] stage (Maven Build)

[Pipeline] script

[Pipeline] {

[Pipeline] withMaven

[Pipeline] // withMaven

[Pipeline] echo

Maven build failed: Could not find specified Maven installation 'Maven\_3.6.3'.

[Pipeline] }

[Pipeline] // script

[Pipeline] stage (Write Output File)

[Pipeline] writeFile

[Pipeline] }

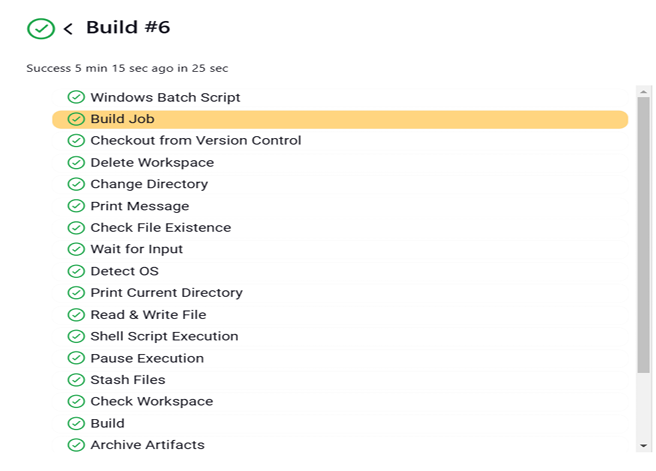
[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS





pipeline {

agent any

environment {

CUSTOM\_VAR = "Hello, Jenkins!"

}

stages {

stage('Windows Batch Script') {

steps {

script {

if (isUnix()) {

echo "Skipping Windows Batch command on Unix system"

} else {

bat 'echo Hello from Windows Batch!'

}

}

}

}

stage('Build Job') {

steps {

script {

try {

build job: 'SomeJobName', wait: true

} catch (Exception e) {

echo "Warning: Build job 'SomeJobName' not found. Skipping this step."

}

}

}

}

stage('Checkout from Version Control') {

steps {

git url: 'https://github.com/Jahnavi-Anand/jenkins\_pipeline.git', branch: 'main'

}

}

stage('Delete Workspace') {

steps {

script {

try {

if (isUnix()) {

sh 'find . -type f ! -name "pom.xml" -delete' // Unix

} else {

bat '''

for /f "delims=" %%i in ('dir /b /a-d ^| findstr /v "pom.xml"') do del "%%i"

'''

}

} catch (Exception e) {

echo "Warning: Failed to delete workspace files. Continuing..."

}

}

}

}

stage('Change Directory') {

steps {

dir('subdirectory') {

echo "Now in subdirectory"

}

}

}

stage('Print Message') {

steps {

echo "Executing Jenkins pipeline!"

}

}

stage('Check File Existence') {

steps {

script {

if (fileExists('somefile.txt')) {

echo "File exists!"

} else {

echo "File does not exist!"

}

}

}

}

stage('Wait for Input') {

steps {

script {

def userInput = input message: 'Continue?', parameters: [booleanParam(defaultValue: true, name: 'Proceed')]

echo "User chose: ${userInput}"

}

}

}

stage('Detect OS') {

steps {

script {

if (isUnix()) {

echo "Running on a Unix-like system"

} else {

echo "Running on Windows"

}

}

}

}

stage('Print Current Directory') {

steps {

script {

echo "Current directory: ${pwd()}"

}

}

}

stage('Read & Write File') {

steps {

script {

writeFile file: 'somefile.txt', text: 'This is a test file'

def content = readFile 'somefile.txt'

echo "File content: ${content}"

}

}

}

stage('Shell Script Execution') {

steps {

script {

if (isUnix()) {

sh 'echo Running shell command'

} else {

echo "Skipping shell command on Windows"

}

}

}

}

stage('Pause Execution') {

steps {

sleep time: 5, unit: 'SECONDS'

}

}

stage('Stash Files') {

steps {

stash includes: '\*\*/\*.txt', name: 'textFiles'

}

}

stage('Check Workspace') {

steps {

script {

if (isUnix()) {

sh 'ls -l'

} else {

bat 'dir'

}

}

}

}

stage('Build') {

steps {

script {

if (isUnix()) {

sh 'mvn clean package'

} else {

bat 'mvn clean package'

}

}

}

}

stage('Archive Artifacts') {

steps {

archiveArtifacts artifacts: 'target/\*.jar', fingerprint: true

}

}

stage('Check Build Output') {

steps {

script {

if (isUnix()) {

sh 'ls -l target'

} else {

bat 'dir target'

}

}

}

}

stage('Set Custom Environment Variable') {

steps {

withEnv(["MY\_VAR=New Value"]) {

script {

if (isUnix()) {

sh 'echo $MY\_VAR'

} else {

bat 'echo %MY\_VAR%'

}

}

}

}

}

stage('Maven Build') {

steps {

script {

try {

withMaven(maven: 'Maven\_3.6.3') {

if (isUnix()) {

sh 'mvn clean package'

} else {

bat 'mvn clean package'

}

}

} catch (Exception e) {

echo "Maven build failed: ${e.message}"

}

}

}

}

stage('Write Output File') {

steps {

writeFile file: 'output.txt', text: 'Jenkins pipeline file output test.'

}

}

}

}