



Automating Build Pipelines with Jenkins and GitHub Integration

Description:

To deepen my practical understanding of DevOps tools, I worked on configuring Jenkins to automate the build and testing process of a Java-based application. My objective was to simulate a real-world continuous integration (CI) scenario by connecting Jenkins to a remote GitHub repository and building the code using Maven through a scripted pipeline.

After completing the initial setup, I explored Jenkins' job types and created a pipeline project. I used a publicly available Maven project hosted on GitHub (simple-maven-project-with-tests) and integrated it with Jenkins through the pipeline script. This script defined key stages such as source checkout, build, test, and archive. I ensured Jenkins correctly fetched the code from GitHub, built the project using Maven while skipping tests, and stored the generated .jar file as a build artifact. The project also attempted to run tests and record results, which demonstrated how Jenkins handles testing and reporting within a pipeline.

Through this hands-on work, I gained valuable insights into how Jenkins fits into CI/CD workflows. I learned to configure tools within Jenkins, connect external repositories, manage build dependencies, and define automated processes using pipeline scripts. This experience also gave me a clearer understanding of how teams use Jenkins to increase productivity, detect integration issues early, and standardize software quality checks in real-time.



• Java installation

```
[thisara@localhost ~]$ sudo yum install -y java-1.8.0-openjdk
[sudo] password for thisara:
Updating Subscription Management repositories.
Docker CE Stable - x86_64                11 kB/s | 3.5 kB    00:00
Kubernetes                             1.9 kB/s | 1.7 kB    00:00
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs) 5.2 kB/s | 4.5 kB    00:00
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs) 2.4 MB/s | 74 MB    00:30
Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs) 4.4 kB/s | 4.1 kB    00:00
Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs) 2.5 MB/s | 94 MB    00:38
Package java-1.8.0-openjdk-1:1.8.0.452.b09-2.el8.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[thisara@localhost ~]$
```

• Download and install Jenkins

```
[thisara@localhost ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo \
> https://pkg.jenkins.io/redhat/jenkins.repo
[sudo] password for thisara:
--2025-07-09 02:25:26-- https://pkg.jenkins.io/redhat/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.46.133, 2a04:4e42:56::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|151.101.46.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 71
Saving to: '/etc/yum.repos.d/jenkins.repo'

/etc/yum.repos.d/jenk 100%[=====>]          71  --.-KB/s    in 0s

2025-07-09 02:25:28 (1.99 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [71/71]
```

```
[thisara@localhost ~]$ sudo rpm --import https://pkg.jenkins.io/redhat/jenkins.io-2023.key
[thisara@localhost ~]$
[thisara@localhost ~]$
```

```
[thisara@localhost ~]$ sudo yum upgrade
Updating Subscription Management repositories.
Jenkins                                36 kB/s | 125 kB    00:03
Dependencies resolved.
=====
Package                                Arch    Version                                Repository                                Size
=====
```



```
[thisara@localhost ~]$ sudo yum install jenkins
[sudo] password for thisara:
Updating Subscription Management repositories.
Last metadata expiration check: 0:03:43 ago on Wed 09 Jul 2025 02:33:00 AM PDT.
Dependencies resolved.
=====
Package                        Architecture      Version           Repository        Size
=====
Installing:
jenkins                       noarch            2.518-1.1         jenkins           90 M
=====
```

```
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

c0d75a7e7dff41c88966f5e2f0678d13

This may also be found at: /home/thisara/.jenkins/secrets/initialAdminPassword

*****
*****
*****

2025-07-09 10:00:15.718+0000 [id=33] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2025-07-09 10:00:15.755+0000 [id=25] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
^C2025-07-09 10:00:28.636+0000 [id=27] INFO winstone.Logger#logInternal: JVM is terminating. Shutting down Jetty
2025-07-09 10:00:28.637+0000 [id=27] INFO org.eclipse.jetty.server.Server#doStop: Stopped oejs.Server@2f177a4b
```

```
[thisara@localhost ~]$ sudo systemctl start jenkins
[thisara@localhost ~]$
[thisara@localhost ~]$
[thisara@localhost ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2025-07-09 03:10:07 PDT; 11s ago
     Main PID: 74437 (java)
       Tasks: 58 (limit: 48846)
      Memory: 985.1M
      CGroup: /system.slice/jenkins.service
              └─74437 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jen
```

• Access Jenkins via web browser

```
[thisara@localhost ~]$ curl ifconfig.co
123.231.86.204
[thisara@localhost ~]$
```





- [Show the password](#)

```
[thisara@localhost ~]$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
75e6419fe37b477d8053793191e8feb0
[thisara@localhost ~]$
```

- [Login](#)

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Getting Started

Create First Admin User

Username

admin

Password

••••••••

Confirm password

••••••••



+ New Item

Build History

Build Queue

No builds in the queue.

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a new Job/Item

New Item

Enter an item name

job1

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Execute a shell command

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Execute shell

?

Command

See [the list of available environment variables](#)

echo "hello word"

Advanced

Save

Apply



- **Console output**

✓ **Console Output**

```
Started by user Thisara Kandage
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/job1
[job1] $ /bin/sh -xe /tmp/jenkins16970428616996196854.sh
+ echo 'hello word'
hello word
Finished: SUCCESS
```

✓ **Console Output**

```
Started by user Thisara Kandage
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/job2
[job2] $ /bin/sh -xe /tmp/jenkins15504909147642646551.sh
+ echo test2
test2
+ date
Finished: SUCCESS
```

- **Output shown in /tmp path**

```
drwx----- 2 thisara thisara 6 Jul 9 02:21 Temp-fa20b9cd-3a53-44a1-9600-ce5f6ac4b276
-rw-r--r-- 1 jenkins jenkins 58 Jul 9 04:45 test2
drwx----- 2 root root 6 Jul 9 02:16 vmware-root_1047-4248615092
drwx----- 2 root root 6 Jun 29 09:44 vmware-root_1057-4282302107
-rw-r--r-- 1 jenkins jenkins 4684028 Jul 9 03:09 winstone4567114367568856266.jar
[thisara@localhost tmp]$
[thisara@localhost tmp]$ cat test2
Wed Jul 9 04:45:31 PDT 2025
Wed Jul 9 04:45:36 PDT 2025
[thisara@localhost tmp]$
```



THISARA KANDAGE

UNDERGRADUATE - SLIIT

[E-mail](#) [LinkedIn](#) [GitHub](#) [Website](#)

• Pipeline creation Steps

New Item

Enter an item name

code-pipeline

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

• Pipeline Script

Definition

Pipeline script

Script ?

```
1 node {
2   def mvnHome
3
4   stage('Preparation') {
5     // Get some code from a GitHub repository
6     git 'https://github.com/jglick/simple-maven-project-with-tests.git'
7
8     // Get the Maven tool. This 'M3' must be configured in Jenkins Global Tool Configuration.
9     mvnHome = tool 'M3'
10  }
11
12  stage('Build') {
13    // Run the Maven build
14    withEnv(["MVN_HOME=$mvnHome", "PATH+MAVEN=$mvnHome/bin"]) {
15      if (isUnix()) {
```

try sample Pipeline...

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

Save

Apply



- **Full Script**

```
node {  
  
    def mvnHome = '/usr/bin/mvn' // Change if your Maven path is different  
  
    stage('Preparation') {  
        git 'https://github.com/jglick/simple-maven-project-with-tests.git'  
    }  
  
    stage('Build') {  
        if (isUnix()) {  
            sh "${mvnHome} clean package -DskipTests=true"  
        } else {  
            bat("/${mvnHome}" clean package -DskipTests=true/  
        }  
    }  
  
    stage('Results') {  
        archiveArtifacts 'target/*.jar'  
    }  
}
```




- **Console output After successfully executed**

```
[INFO] Building jar: /var/lib/jenkins/workspace/code-pipeline/target/simple-maven-project-with-
tests-1.0-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 4.328 s
[INFO] Finished at: 2025-07-09T06:01:20-07:00
[INFO] -----
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Results)
[Pipeline] archiveArtifacts
Archiving artifacts
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

//developers.redhat.com

Summary:

- Learned how to create and configure Jenkins jobs using scripted pipelines
- Integrated GitHub with Jenkins for automated source code checkout
- Understood the Jenkins pipeline stages: checkout, build, test, and archive
- Used Maven to compile a Java project and produce a build artifact (.jar)
- Skipped tests in one build to observe Maven flexibility in pipeline
- Gained experience handling build outputs and test result reports
- Observed error messages and learned to debug job configurations
- Developed familiarity with Jenkins interface and environment variables
- Strengthened understanding of Continuous Integration (CI) principles
- Improved readiness for DevOps internship opportunities through hands-on automation practice