



Title of the Assignment	Assignment 2
Course Title	System Deployment and Operation
Lecturer's name	Tom Huynh
Name and Student ID	Nguyen Huynh Quang - s3927039
Submission Date	January 2nd, 2024

TABLE OF CONTENTS

I.	Introduction	3
II.	Objective	3
III.	Main requirement section	
	Step 1: Fork Github Repository	4
	Step 2: Setup the pipeline	5
	Step 3: Make a change on the website	18
	Step 4: Show the CI/CD pipeline	19
	Step 5: Show the website	
IV.	Advanced requirement section	21
V.	Conclusion	32
VI.	References	32

I. Introduction

The objective of this assignment is to build a simple Continuous Integration/ Continuous Deployment (CI/CD) pipeline. CI/CD pipeline is built to enable the faster delivery of software throughout software development life cycle (SDLC) [1]. Continuous Integration (CI) is a process that integrates change from multiple branches in the same repository into the main branch. Meanwhile, Continuous Deployment (CD) takes a further step in automating the deployment process [2].

To complete this assignment, Jenkins and Docker are used to build a simple CI/CD pipeline. Jenkins is an open source continuous integration (CI) server, which manages the build, documentation and packaging of software processes. In the software development process, Jenkins automatically integrates changes in the source code located in repositories such as Github, Gitlab or Bitbucket . [3].

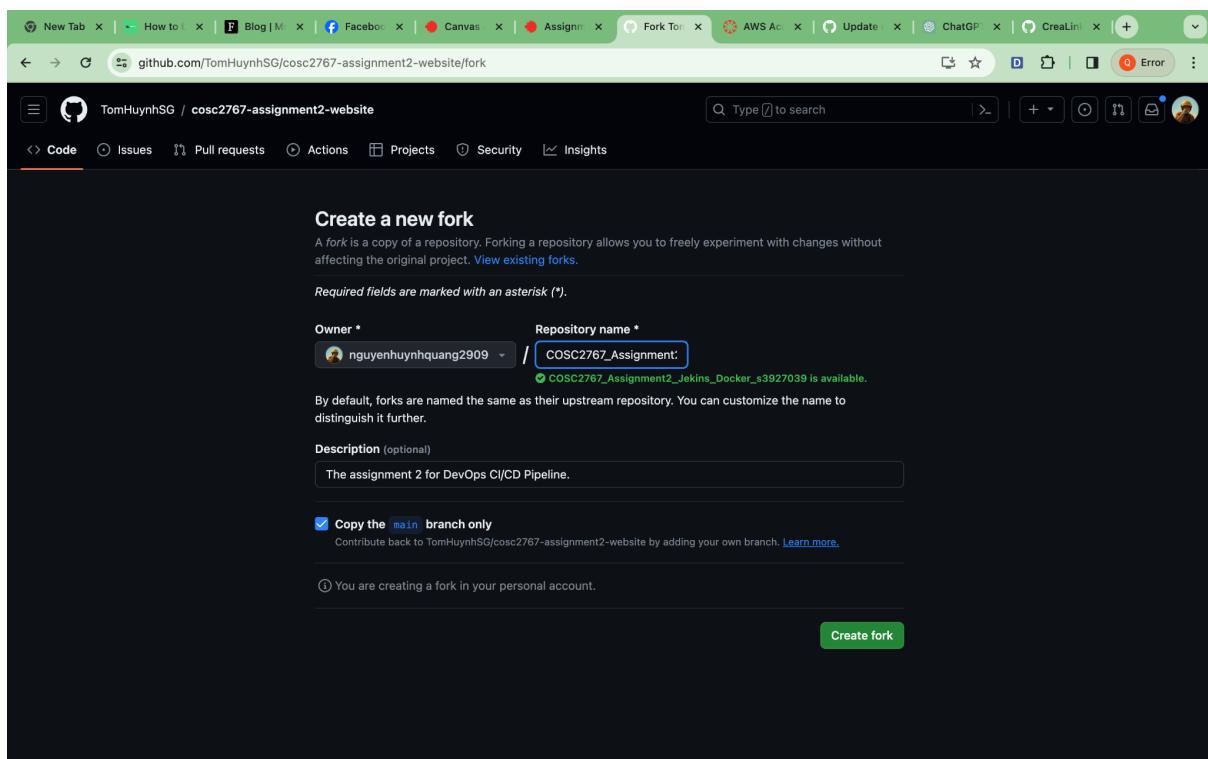
Meanwhile, Docker is an open source platform that enables developers to build and deploy applications into Docker containers. Docker container is a lightweight and standalone executable package that contains source code of the application, as well as operating system (OS) and dependencies that are required for the deployment [4].

II. Objective

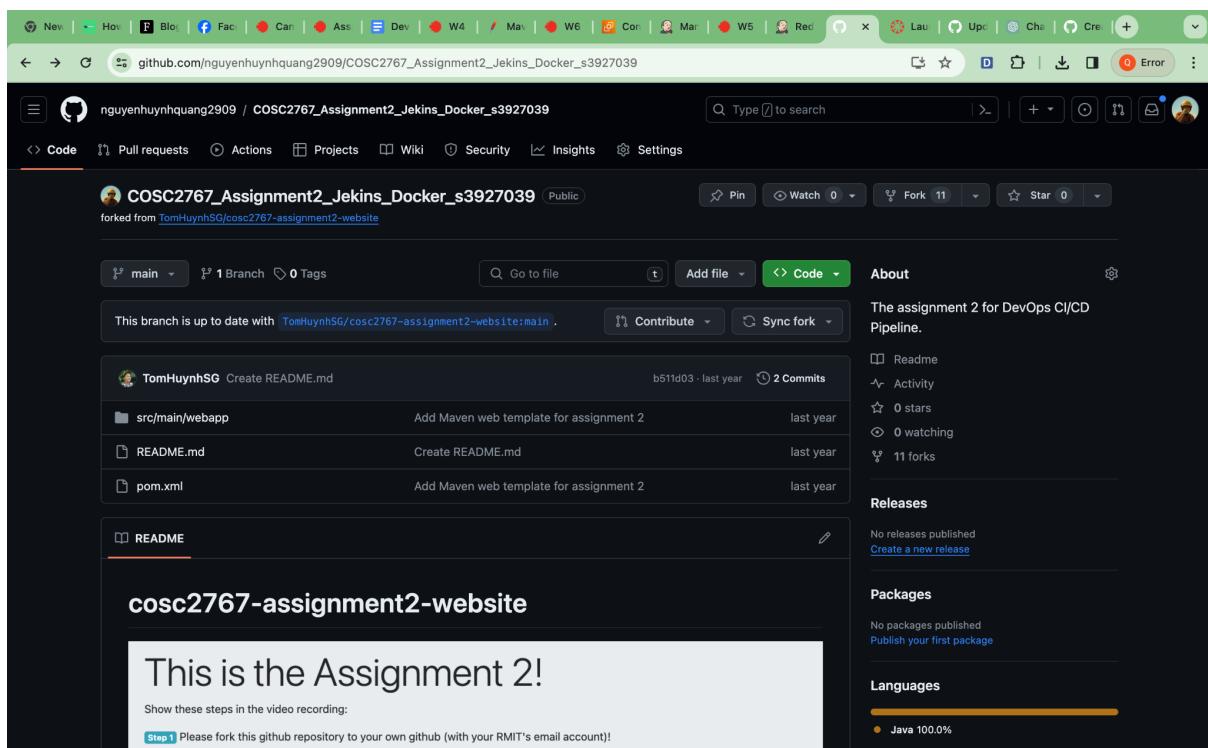
The objective of this assignment is to build a simple Continuous Integration / Continuous Deployment (CI/CD) pipeline using Jenkins and Docker. Jenkins is implemented to integrate change in the Github repos, which pull source code from Github repository and build the project with Maven. After that, Docker will be implemented to handle project deployment with Tomcat.

III. Main Requirement Section

Step 1: Fork Github repository



Fork Github repository to my Github account



Github repository was forked successfully

Step 2: Setup the pipeline

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: s3927039_Jenkins_Server Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, Browse more AMIs Including AMIs from

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.3.2... read more
ami-0759f51a90924c166

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet

Cancel **Launch instance** Review commands

Launched EC2 instance which is used for Jenkins server

Compare security group rules Info

Amazon EC2 evaluates all the rules of the selected security groups to control inbound and outbound traffic. You can select more security groups to view their inbound rules to help you to decide how to secure your instance from incoming traffic.

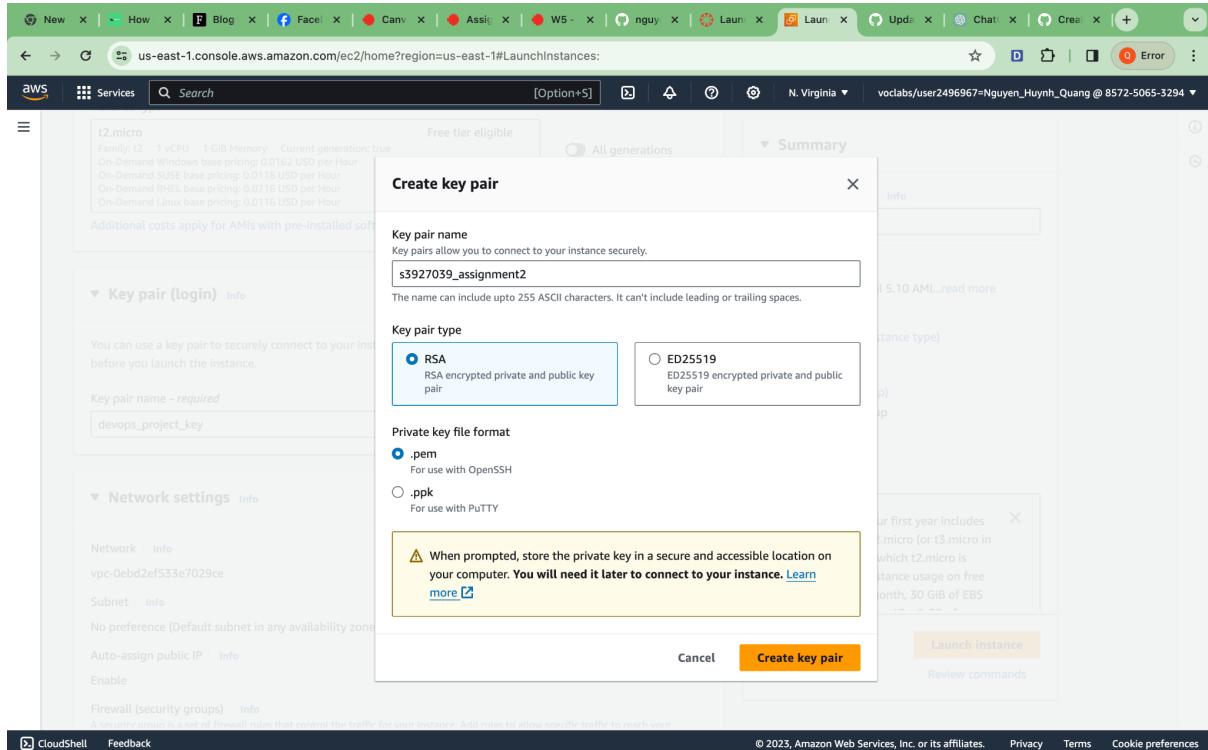
Common security groups: Jenkins_Security_Group sg-006d527ca108fa189 VPC: vpc-0edb2ef53e7029ce

Inbound rules (2)

Security group name	Security group ID	Type	Protocol	Port range	Source	Description
Jenkins_Security_Group	sg-006d527ca108fa189	Custom TCP	tcp	8080	0.0.0.0/0	-
Jenkins_Security_Group	sg-006d527ca108fa189	ssh	tcp	22	0.0.0.0/0	-

Cancel **Select security groups**

Jenkins security group was created with two ports: 22 for SSH and 8080 for Jenkins, which is applied as security group for Jenkins server



A new key pair was generated for this assignment, this key pair will be used to connect into Jenkins and Docker server.

```

Last login: Fri Dec 15 23:27:58 on ttys000
(base) nguyễnhuynhquang2999@NguyễnHuynhQuang2999-MacBook-Air ~ % cd Documents
(base) nguyễnhuynhquang2999@NguyễnHuynhQuang2999-MacBook-Air Documents % cd s3927039-assignment2-key
cd: no such file or directory: s3927039-assignment2-key
(base) nguyễnhuynhquang2999@NguyễnHuynhQuang2999-MacBook-Air Documents % cd "s3927039-assignment2-key"
(base) nguyễnhuynhquang2999@NguyễnHuynhQuang2999-MacBook-Air s3927039-assignment2.key %
s3927039-assignment2.pem
(base) nguyễnhuynhquang2999@NguyễnHuynhQuang2999-MacBook-Air s3927039-assignment2.key % chmod 400 s3927039-assignment2.pem
(base) nguyễnhuynhquang2999@NguyễnHuynhQuang2999-MacBook-Air s3927039-assignment2.key % ssh -l "s3927039-assignment2.pem" ec2-user@ec2-54-174-99-13.compute-1.amazonaws.com
The authenticity of host 'ec2-54-174-99-13.compute-1.amazonaws.com (54.174.99.13)' can't be established.
ED25519 key fingerprint is SHA256:YmN8Y0QxXQL8CdiJ4iEfxXiOrlawm7sz2Bw0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-174-99-13.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

      ##
      ##
      ## AL2 End of Life is 2025-06-30.
      ##
      #> / A newer version of Amazon Linux is available!
      #> / Amazon Linux 2023, GA and supported until 2028-03-15.
      #> / https://aws.amazon.com/linux/amazon-linux-2023/
      #> /m/Amazon Linux 2

-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ip-172-31-44-3 ~]$ sudo su -
[root@ip-172-31-44-3 ~]# sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2023-12-16 08:47:45 -- https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.30.133, 2a04:e42:78::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.30.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85 [=====
Saving to: '/etc/yum.repos.d/jenkins.repo'

100%[=====] 85 --.-K/s in 0s

2023-12-16 08:47:45 (5.69 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]

[root@ip-172-31-44-3 ~]# sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[root@ip-172-31-44-3 ~]# sudo amazon-linux-extras install java-openjdk11
Topic java-openjdk11 has end-of-support date of 2024-09-30
Installing java-11-openjdk
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-java-openjdk11 amzn2extra-kernel-5.10 jenkins
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amazon2extra-docker
amazon2extra-java-openjdk11
amazon2extra-kernel-5.10
jenkins
(1/10): amzn2-core/2/x86_64/group.gz | 3.6 kB 00:00:00
(2/10): amzn2-core/2/x86_64/updateinfo | 2.9 kB 00:00:00
(3/10): amzn2extra-java-openjdk11/2/x86_64/primary_db | 3.0 kB 00:00:00
(4/10): amzn2extra-kernel-5.10/2/x86_64/updateinfo | 3.0 kB 00:00:00
(5/10): amzn2extra-docker/2/x86_64/updateinfo | 2.9 kB 00:00:00
(6/10): amzn2extra-docker/2/x86_64/primary_db | 2.9 kB 00:00:00
(7/10): amzn2extra-java-openjdk11/2/x86_64/updateinfo | 759 kB 00:00:00
(8/10): amzn2extra-kernel-5.10/2/x86_64/primary_db | 172 kB 00:00:00
(9/10): amzn2extra-kernel-5.10/2/x86_64/primary_db | 42 kB 00:00:00
(10/10): jenkins/primary.db | 13 kB 00:00:00
| 105 kB 00:00:00
| 4.9 kB 00:00:00
| 21 kB 00:00:00
| 48 kB 00:00:00

```

After SSH into the Jenkins server, I downloaded the Jenkins package into the EC2 server and installed Java JDK 11, which is compatible with our project.

```

2 httpd_modules      available  [ =1.0  =stable ]
3 memcached1.5     available  \
4   [ =1.5.1  =1.5.16  =1.5.16 ]
5   [ =1.5.16  =1.5.16 ]
6   [ =1.5.16  =1.5.16 ]
7   [ =1.5.16  =1.5.16 ]
8   [ =1.5.16  =1.5.16 ]
9   [ =1.5.16  =1.5.16 ]
10  R3.4             available  \
11   [ =3.4.3  =stable ]
12   [ =3.4.3  =stable ]
13   [ =3.4.3  =stable ]
14   [ =3.4.3  =stable ]
15   [ =3.4.3  =stable ]
16   [ =3.4.3  =stable ]
17   [ =3.4.3  =stable ]
18 libreoffice        available  \
19   [ =5.0.6.2_15  =5.0.6.2_15 ]
20 gimp              available  \
21   [ =2.8.22  =stable ]
22 fdo-kerrel-test  enabled   \
23   [ =17.12.1  =18.03.1  =18.06.1  =18.09.9  =stable ]
24 mate-desktop1.x   available  \
25   [ =1.19.0  =1.20.0  =stable ]
26 GraphicsMagick1.3 available  \
27   [ =1.3.29  =1.3.32  =1.3.34  =stable ]
28 ftomata8.4       available  \
29   [ =8.5.31  =8.5.32  =8.5.38  =8.5.40  =8.5.42  =8.5.50
30   =stable ]
31 epel               available  \
32 lustre2.10        available  \
33   [ =2.10.5  =2.10.8  =stable ]
34 tjava-openjdk11=latest enabled   [ =11  =stable ]
35 lynx              available  [ =stable ]
36 BCC               available  [ =6.x  =stable ]
37 mono              available  [ =5.x  =stable ]
38 nginx1            available  [ =stable ]
39 mod_wsgi           available  [ =stable ]
40 livepatch          available  [ =stable ]
41 kernel             available  [ =stable ]
42 python3.8          available  [ =stable ]
43 haproxy2           available  [ =stable ]
44 collectd           available  [ =stable ]
45 aws-nitro-enclaves cli available  [ =stable ]
46 aws-nitro-enclaves cli available  [ =stable ]
47 R4                available  [ =stable ]
48 kernel-5.4          available  [ =stable ]
49 selinux-ng          available  [ =stable ]
50 tomcat9            available  [ =stable ]
51 unbound1.13         available  [ =stable ]
52 fmaria10.5          available  [ =stable ]
53 kernel-5.10=latest  enabled   [ =stable ]
54 kernel-5.10=latest  enabled   [ =stable ]
55 kernel-5.10=latest  enabled   [ =stable ]
56 redis6             available  [ =stable ]
57 ruby3.0             available  [ =stable ]
58 tpostgresql12        available  [ =stable ]
59 tpostgresql13        available  [ =stable ]
60 mock2              available  [ =stable ]
61 dnsmasq2.85          available  [ =stable ]
62 kernel-5.15          available  [ =stable ]
63 tpostgresql14        available  [ =stable ]
64 firefox             available  [ =stable ]
65 lustre              available  [ =stable ]
66 tphb8.1             available  [ =stable ]
67 awscli              available  [ =stable ]
68 eccl11              available  [ =stable ]
69 dnsmasq              available  [ =stable ]
70 unbound1.17         available  [ =stable ]
71 collected-python3   available  [ =stable ]
72 collected-python3   available  [ =stable ]
73 Note on end-of-support. Use 'info' subcommand.
[root@ip-172-31-44-3 ~]# 
```

Java JDK 11 was installed successfully

```

[ =2.10.5  =2.10.8  =stable ]
33 tjava-openjdk11=latest enabled   [ =11  =stable ]
34 lynx              available  [ =stable ]
35 BCC               available  [ =6.x  =stable ]
36 mono              available  [ =5.x  =stable ]
37 nginx1            available  [ =stable ]
38 mod_wsgi           available  [ =stable ]
39 livepatch          available  [ =stable ]
40 kernel             available  [ =stable ]
41 python3.8          available  [ =stable ]
42 haproxy2           available  [ =stable ]
43 collectd           available  [ =stable ]
44 aws-nitro-enclaves cli available  [ =stable ]
45 R4                available  [ =stable ]
46 kernel-5.4          available  [ =stable ]
47 selinux-ng          available  [ =stable ]
48 tomcat9            available  [ =stable ]
49 unbound1.13         available  [ =stable ]
50 fmaria10.5          available  [ =stable ]
51 kernel-5.10=latest  enabled   [ =stable ]
52 redis6             available  [ =stable ]
53 ruby3.0             available  [ =stable ]
54 tpostgresql12        available  [ =stable ]
55 tpostgresql13        available  [ =stable ]
56 mock2              available  [ =stable ]
57 dnsmasq2.85          available  [ =stable ]
58 kernel-5.15          available  [ =stable ]
59 tpostgresql14        available  [ =stable ]
60 firefox             available  [ =stable ]
61 lustre              available  [ =stable ]
62 tphb8.1             available  [ =stable ]
63 eccl11              available  [ =stable ]
64 tphb8.2             available  [ =stable ]
65 dnsmasq              available  [ =stable ]
66 unbound1.17         available  [ =stable ]
67 collected-python3   available  [ =stable ]
68 collected-python3   available  [ =stable ]
69 Note on end-of-support. Use 'info' subcommand.
[root@ip-172-31-44-3 ~]# service jenkins start
Redirecting to /bin/systemctl status jenkins.service
Unit jenkins.service could not be found.
[root@ip-172-31-44-3 ~]# yum install jenkins
Loaded plugins: extras-suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package jenkins.noarch 0:2.426.2-1.1 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
=====
Package           Arch      Version           Repository      Size
=====
Installing:
jenkins          noarch    2.426.2-1.1      jenkins        85 M
=====
Transaction Summary
=====
Install 1 Package
Total download size: 85 M
Installed size: 85 M
Is this ok [y/d/N]: y
Downloading packages:
[jenkins-2.426.2-1.1.noarch.rpm] 2.8 MB/s | 18 MB 00:00:23 ETA
[root@ip-172-31-44-3 ~]# 
```

In the next step, Jenkins was installed from the Jenkins package which was downloaded from the previous step

```
[root@ip-172-31-44-3 ~]# yum install jenkins
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running Transaction check
--> Package jenkins.noarch 0:2.426.2-1.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version          Repository      Size
=====
Install:
jenkins          noarch   2.426.2-1.1    jenkins        85 M
Transaction Summary
Install 1 Package

Total download size: 85 M
Installed size: 85 M
(1 package)
Downloaded packages:
jenkins-2.426.2-1.1.noarch.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction install
  Installing : Jenkins-2.426.2-1.1.noarch
  Verifying  : Jenkins-2.426.2-1.1.noarch
Installed:
jenkins.noarch 0:2.426.2-1.1

Complete!
[root@ip-172-31-44-3 ~]# service jenkins status
Redirecting to /bin/systemctl status jenkins.service
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; vendor preset: disabled)
    Active: inactive (dead)
[...]
[root@ip-172-31-44-3 ~]# service jenkins start
Redirecting to /bin/systemctl start jenkins.service
[root@ip-172-31-44-3 ~]# service jenkins status
Redirecting to /bin/systemctl status jenkins.service
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; vendor preset: disabled)
    Active: active (running) since Sat 2023-12-16 08:52:00 UTC; 12s ago
      Main PID: 4466 (java)
     Group: /system.slice/jenkins.service
           └─4466 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Dec 16 08:51:32 ip-172-31-44-3.ec2.internal jenkins[4466]: 43647fb0b95e40f7a3c476ff1f187cdf
Dec 16 08:51:32 ip-172-31-44-3.ec2.internal jenkins[4466]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Dec 16 08:51:32 ip-172-31-44-3.ec2.internal jenkins[4466]: ****
Dec 16 08:51:32 ip-172-31-44-3.ec2.internal jenkins[4466]: ****
Dec 16 08:51:32 ip-172-31-44-3.ec2.internal jenkins[4466]: ****
Dec 16 08:51:32 ip-172-31-44-3.ec2.internal jenkins[4466]: 2023-12-16 08:52:00 [id=31] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
Dec 16 08:51:32 ip-172-31-44-3.ec2.internal jenkins[4466]: 2023-12-16 08:52:00 [id=31] INFO hudson.lifecycle.Lifecycle$OnReady: Jenkins is fully up and running
Dec 16 08:52:06 ip-172-31-44-3.ec2.internal systemd[1]: Started Jenkins Continuous Integration Server.
Dec 16 08:52:07 ip-172-31-44-3.ec2.internal jenkins[4466]: 2023-12-16 08:52:07.717+0000 [id=46] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for ...venInstaller
Dec 16 08:52:07 ip-172-31-44-3.ec2.internal jenkins[4466]: 2023-12-16 08:52:07.718+0000 [id=46] INFO hudson.util.Retryer#start: Performed the action check updates server succee...e attempt #1
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-44-3 ~]#
```

After installing, Jenkins server was started successfully, which is ready for the Continuous Integration (CI) job.

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

.....

Continue

<https://www.jenkins.io/redirect/find-jenkins-logs>

Login into Jenkins with initial admin password

Sig X W5 - V Redhat nguyen Launch Update ChatG Creati

Not Secure 54.174.99.13:8080/user/admin/configure

Dashboard > admin > Configure

User View

The view selected by default when navigating to the user's private views

Password

Password:

.....

Confirm Password:

.....

Session Termination

Terminate All Sessions ?

Setting for search

Case-sensitivity

Insensitive search tool

Save Apply

Change the Jenkins login credentials with a new username and password that is easy to remember .

Sig X W5 - V Redhat nguyen Launch Update ChatG Creati

Not Secure 54.174.99.13:8080/login?from=%2Fuser%2Fadmin%2Fconfigure



Sign in to Jenkins

Username

admin

Password

.....

Keep me signed in

Sign in

Login with username and password which is configured in the previous step.

```

Dec 16 08:52:06 ip-172-31-44-3.ec2.internal systemd[1]: Started Jenkins Continuous Integration Server.
Dec 16 08:52:07 ip-172-31-44-3.ec2.internal jenkins[4466]: 2023-12-16 08:52:07.717+0000 [id=46]      INFO      h.m.DownloadService$Downloadable#load: Obtained the updated data file for ...venInstaller
Dec 16 08:52:07 ip-172-31-44-3.ec2.internal jenkins[4466]: 2023-12-16 08:52:07.718+0000 [id=46]      INFO      hudson.util.Retrier#start: Performed the action check updates server success...e attempt #1
Hint: Some lines were ellipsized, use -J to show in full.
[root@ip-172-31-44-3 ~]# cat /var/lib/jenkins/secrets/initialAdminPassword
43647fb095a4af7a3a4476f77187cdff
[root@ip-172-31-44-3 ~]# yum install git
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core/x86_64
Resolving Dependencies
-> Running transaction check
--> Package git.x86_64 0:2.48.1-1.amzn2.0.1 will be installed
--> Processing Dependency: git-core = 2.48.1-1.amzn2.0.1 for package: git-2.48.1-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core-doc = 2.48.1-1.amzn2.0.1 for package: git-2.48.1-1.amzn2.0.1.x86_64
--> Processing Dependency: perl-Git = 2.48.1-1.amzn2.0.1 for package: git-2.48.1-1.amzn2.0.1.x86_64
--> Processing Dependency: perl-Git(tert::ReadKey) for package: git-2.48.1-1.amzn2.0.1.x86_64
--> Running transaction check
--> Package git-core.x86_64 0:2.48.1-1.amzn2.0.1 will be installed
--> Package git-core-doc.noarch 0:2.48.1-1.amzn2.0.1 will be installed
--> Package perl-Git.noarch 0:2.48.1-1.amzn2.0.1 will be installed
--> Package perl-TermReadkey.x86_64 0:2.38-20.amzn2.0.2 will be installed
--> Package perl-Error.noarch 1:0.17028-2.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
=====


| Package                                              | Arch   | Version            | Repository | Size           |
|------------------------------------------------------|--------|--------------------|------------|----------------|
| <hr/>                                                |        |                    |            |                |
| Installing:                                          |        |                    |            |                |
| git                                                  | x86_64 | 2.48.1-1.amzn2.0.1 | amzn2-core | 54 k           |
| Installing for dependencies:                         |        |                    |            |                |
| git-core                                             | x86_64 | 2.48.1-1.amzn2.0.1 | amzn2-core | 10 M           |
| git-core-doc                                         | noarch | 2.48.1-1.amzn2.0.1 | amzn2-core | 3.0 M          |
| perl-Git                                             | noarch | 1:0.17028-2.amzn2  | amzn2-core | 32 k           |
| perl-Git                                             | noarch | 2.48.1-1.amzn2.0.1 | amzn2-core | 41 k           |
| perl-TermReadKey                                     | x86_64 | 2.38-20.amzn2.0.2  | amzn2-core | 51 k           |
| <hr/>                                                |        |                    |            |                |
| Transaction Summary                                  |        |                    |            |                |
| Install 1 Package (+5 Dependent packages)            |        |                    |            |                |
| Total download size: 13 M                            |        |                    |            |                |
| Installed size: 44 M                                 |        |                    |            |                |
| Is this ok [y/N]: y                                  |        |                    |            |                |
| Downloading packages:                                |        |                    |            |                |
| (1/6): git-2.48.1-1.amzn2.0.1.x86_64.rpm             |        |                    |            | 54 kB 00:00:00 |
| (2/6): git-core-doc-2.48.1-1.amzn2.0.1.noarch.rpm    |        |                    |            | 3.0 M 00:00:00 |
| (3/6): perl-Git-1.17028-2.amzn2.noarch.rpm           |        |                    |            | 32 kB 00:00:00 |
| (4/6): perl-Git-2.48.1-1.amzn2.0.1.noarch.rpm        |        |                    |            | 41 kB 00:00:00 |
| (5/6): perl-TermReadKey-2.38-20.amzn2.0.2.x86_64.rpm |        |                    |            | 31 kB 00:00:00 |
| (6/6): git-core-2.48.1-1.amzn2.0.1.x86_64.rpm        |        |                    |            | 10 MB 00:00:00 |
| Total 42 MB/s   13 MB 00:00:00                       |        |                    |            |                |
| Running transaction check                            |        |                    |            |                |
| Running transaction test                             |        |                    |            |                |
| Transaction test succeeded                           |        |                    |            |                |
| Running transaction                                  |        |                    |            |                |
| Installing : git-core-2.48.1-1.amzn2.0.1.x86_64      |        |                    |            | 1/6            |
| Installing : git-core-doc-2.48.1-1.amzn2.0.1.noarch  |        |                    |            | 2/6            |


```

In this step, Git was installed into Jenkins EC2 server for the purpose of pulling code from Github repository.

Install	Name	Released
<input checked="" type="checkbox"/>	GitHub 1.37.3.1	1 mo 28 days ago
<input type="checkbox"/>	External Site/Tool Integrations - github	
<input type="checkbox"/>	GitHub API 1.318-461.v7a_c09c9fa_d63	20 days ago
<input type="checkbox"/>	github - Library plugins (for use by other plugins)	
<input type="checkbox"/>	Caffeine API 3.1.8-133.v17b_1ff2e0599	4 mo 2 days ago
<input type="checkbox"/>	Library plugins (for use by other plugins)	
<input type="checkbox"/>	GitHub Branch Source 1758.v048414714f5d	2 days 13 hr ago
<input type="checkbox"/>	pipeline - github - Source Code Management	
<input type="checkbox"/>	OkHttp 4.11.0-157.v6852a_a_fa_ec11	4 mo 9 days ago
<input type="checkbox"/>	This plugin provides OkHttp for other plugins.	
<input type="checkbox"/>	JavaBeans Activation Framework (JAF) API 1.2.0-6	10 mo ago
<input type="checkbox"/>	Library plugins (for use by other plugins)	
<input type="checkbox"/>	This plugin provides the JavaBeans Activation Framework (JAF) API for other plugins.	

Installed Github plugins in Jenkins GUI, which allows Github integration with Jenkins.

```
[root@ip-172-31-44-3 ~]# cd /opt
[root@ip-172-31-44-3 opt]# pwd
/opt
[root@ip-172-31-44-3 opt]# wget https://dlcdn.apache.org/maven/maven-3/3.9.6/binaries/apache-maven-3.9.6-bin.tar.gz
--2023-12-16 09:07:38-- https://dlcdn.apache.org/maven/maven-3/3.9.6/binaries/apache-maven-3.9.6-bin.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9410588 (9.0M) [application/x-gzip]
Saving to: 'apache-maven-3.9.6-bin.tar.gz'

100%[=====] 9,410,588 --.-K/s in 0.07s

2023-12-16 09:07:38 (126 MB/s) - "apache-maven-3.9.6-bin.tar.gz" saved [9410588/9410588]
```

Downloaded Maven package into Jenkins EC2 instance so that Jenkins is able to build and manage Maven-based projects.

```
apache-maven-3.9.6/lib/jansi-native/
apache-maven-3.9.6/lib/ext/README.txt
apache-maven-3.9.6/lib/ext/hazelcast/README.txt
apache-maven-3.9.6/lib/ext/redisson/README.txt
apache-maven-3.9.6/lib/ext/tomcat/tomcat-juli.jar
apache-maven-3.9.6/lib/maven-embedder-3.9.6.jar
apache-maven-3.9.6/lib/maven-settings-3.9.6.jar
apache-maven-3.9.6/lib/maven-settings-builder-3.9.6.jar
apache-maven-3.9.6/lib/maven-plugin-api-3.9.6.jar
apache-maven-3.9.6/lib/maven-model-builder-3.9.6.jar
apache-maven-3.9.6/lib/maven-builder-support-3.9.6.jar
apache-maven-3.9.6/lib/maven-resolver-api-1.9.18.jar
apache-maven-3.9.6/lib/maven-resolver-util-1.9.18.jar
apache-maven-3.9.6/lib/maven-shared-utils-3.3.4.jar
apache-maven-3.9.6/lib/maven-guava-32.0.1.jar
apache-maven-3.9.6/lib/guava-32.0.1-jre.jar
apache-maven-3.9.6/lib/failureaccess-1.0.1.jar
apache-maven-3.9.6/lib/javax.annotation-api-1.3.2.jar
apache-maven-3.9.6/lib/plexus-utils-3.5.1.jar
apache-maven-3.9.6/lib/plexus-interpolation-1.20.0.jar
apache-maven-3.9.6/lib/plxus-cipher-2.0.jar
apache-maven-3.9.6/lib/plexus-interpolation-1.26.jar
apache-maven-3.9.6/lib/slf4j-api-1.7.36.jar
apache-maven-3.9.6/lib/commons-lang3-3.12.0.jar
apache-maven-3.9.6/lib/maven-client-interactions-1.0.0.jar
apache-maven-3.9.6/lib/maven-repository-metadata-3.9.6.jar
apache-maven-3.9.6/lib/maven-artifact-3.9.6.jar
apache-maven-3.9.6/lib/maven-resolver-provider-3.9.6.jar
apache-maven-3.9.6/lib/maven-resolver-impl-1.9.18.jar
apache-maven-3.9.6/lib/maven-resolver-named-locks-1.9.18.jar
apache-maven-3.9.6/lib/maven-resolver-resolver-1.9.18.jar
apache-maven-3.9.6/lib/eclips-sisu-inject-bean-0.9.0.M2.jar
apache-maven-3.9.6/lib/plexus-component-annotations-3.1.0.jar
apache-maven-3.9.6/lib/maven-compat-3.9.6.jar
apache-maven-3.9.6/lib/wagon-provider-api-3.5.3.jar
apache-maven-3.9.6/lib/org.eclips.sisu.plexus-0.9.0.M2.jar
apache-maven-3.9.6/lib/commons-cl1-1.0.0.jar
apache-maven-3.9.6/lib/commons-lang3-3.5.3.jar
apache-maven-3.9.6/lib/wagon-http-shared-3.5.3.jar
apache-maven-3.9.6/lib/httpclient-4.6.14.jar
apache-maven-3.9.6/lib/wagon-file-3.5.3.jar
apache-maven-3.9.6/lib/jcl-over-slf4j-1.7.36.jar
apache-maven-3.9.6/lib/maven-resolver-connector-basic-1.9.18.jar
apache-maven-3.9.6/lib/maven-resolver-transport-file-1.9.18.jar
apache-maven-3.9.6/lib/maven-resolver-transport-http-1.9.18.jar
apache-maven-3.9.6/lib/httpcore-4.4.16.jar
apache-maven-3.9.6/lib/commons-codec-1.16.0.jar
apache-maven-3.9.6/lib/maven-resolver-transport-wagon-1.9.18.jar
apache-maven-3.9.6/lib/maven-slf4j-provider-3.9.6.jar
apache-maven-3.9.6/lib/jansi-2.4.0.jar
[root@ip-172-31-44-3 opt]# ls -la
total 192
drwxr-xr-x  5 root root   90 Dec 16 09:07 .
dr-xr-xr-x 18 root root  257 Dec 16 08:44 ..
drwxr-xr-x  6 root root   98 Dec 16 09:07 apache-maven-3.9.6
-rw-r--r--  1 root root 9410588 Dec 16 09:07 apache-maven-3.9.6-bin.tar.gz
drwxr-xr-x  4 root root   33 Dec  6 19:44 aws
drwxr-xr-x  2 root root    6 Aug 16 2018 rh
[root@ip-172-31-44-3 opt]# ]
```

Maven package was installed and extracted into /opt directory.

```

apache-maven-3.9.6/lib/maven-artifact-3.9.6.jar
apache-maven-3.9.6/lib/maven-resolver-provider-3.9.6.jar
apache-maven-3.9.6/lib/maven-resolver-provider-remoting-3.9.6.jar
apache-maven-3.9.6/lib/maven-resolver-named-locks-1.9.18.jar
apache-maven-3.9.6/lib/org/eclipse/sisu/inject-0.9.0.M2.jar
apache-maven-3.9.6/lib/maven-compat-3.9.6.jar
apache-maven-3.9.6/lib/maven-wagon-provider-3.5.3.jar
apache-maven-3.9.6/lib/org/eclipse/sisu/pluto-0.9.0.M2.jar
apache-maven-3.9.6/lib/commons-cli-1.5.0.jar
apache-maven-3.9.6/lib/wagon-http-3.5.3.jar
apache-maven-3.9.6/lib/wagon-http-shared-3.5.3.jar
apache-maven-3.9.6/lib/httpclient-4.5.14.jar
apache-maven-3.9.6/lib/maven-wagon-file-3.5.3.jar
apache-maven-3.9.6/lib/org/apache/HttpClient-4.1.7.36.jar
apache-maven-3.9.6/lib/maven-resolver-connector-basic-1.9.18.jar
apache-maven-3.9.6/lib/maven-resolver-transport-file-1.9.18.jar
apache-maven-3.9.6/lib/maven-resolver-transport-http-1.9.18.jar
apache-maven-3.9.6/lib/httpcore-4.4.16.jar
apache-maven-3.9.6/lib/commons-codec-1.16.0.jar
apache-maven-3.9.6/lib/commons-logging-1.2.2.jar
apache-maven-3.9.6/lib/maven-slf4j-provider-3.9.6.jar
apache-maven-3.9.6/lib/jansi-2.4.0.jar
[root@ip-172-31-44-3 opt]# ls -la
total 192
drwxr-xr-x  5 root root   98 Dec 16 09:47 .
drwxr-xr-x 18 root root  257 Dec 16 08:44 ..
drwxr-xr-x  6 root root   99 Dec 16 09:47 apache-maven-3.9.6
-rw-r--r--  1 root root 9418588 Dec 1 11:22 apache-maven-3.9.6-bin.tar.gz
drwxr-xr-x  4 root root  33 Dec 16 09:44 aws
drwxr-xr-x  4 root root  99 Dec 16 09:47 jansi
drwxr-xr-x  2 root root   6 Aug 16 2018 rh
[root@ip-172-31-44-3 opt]# mv apache-maven-3.9.6 maven
[root@ip-172-31-44-3 opt]# ls -la
total 192
drwxr-xr-x  5 root root   77 Dec 16 09:48 .
drwxr-xr-x 18 root root  257 Dec 16 08:44 ..
-rw-r--r--  1 root root 9418588 Dec 1 11:22 apache-maven-3.9.6-bin.tar.gz
drwxr-xr-x  4 root root  33 Dec 16 09:44 aws
drwxr-xr-x  4 root root  99 Dec 16 09:47 jansi
drwxr-xr-x  2 root root   6 Aug 16 2018 rh
[root@ip-172-31-44-3 opt]# ./maven/bin/mvn -
[INFO] Scanning for projects...
[INFO] -----
[INFO] BUILD FAILURE
[INFO] -----
[INFO] Total time:  0.236 s
[INFO] Finished at: 2023-12-16T09:08:46Z
[INFO] -----
[ERROR] The goal you specified requires a project to execute but there is no POM in this directory. Please verify you invoked Maven from the correct directory. --> [Help 1]
[ERROR] 
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR] 
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://wiki.apache.org/confluence/display/MAVEN/MissingProjectException
[root@ip-172-31-44-3 opt]# ./maven/bin/mvn -v
Apache Maven 3.9.6 (0924f5c744d66b6ec92053d08dcc295161ae)
Maven home: /opt/maven
Java version: 11.0.20, vendor: Red Hat, Inc., runtime: /usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.amzn2.0.1.x86_64
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.10.201-191.748.amzn2.x86_64", arch: "amd64", family: "unix"
[root@ip-172-31-44-3 opt]# 
```

Rename the maven package as maven.

```

GNU nano 2.9.8                                .bash_profile                                         Modified
# .bash_profile
#
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs
M2_HOME=/opt/maven
M2=$M2_HOME/bin
JAVA_HOME=/usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.amzn2.0.1.x86_64
PATH=$PATH:$HOME/bin:$JAVA_HOME:$M2
export PATH


```

In this step, Maven and Tomcat were installed in the bash_profile file, which is used for storing environment variables; this will enable the system to recognize and use these plugins.

```

dr-xr-xr-x 18 root root 257 Dec 16 08:44 ..
drwxr-xr-x  6 root root 99 Dec 16 09:07 apache-maven-3.9.6
-rw-r--r--  1 root root 9416868 Dec 16 09:07 apache-maven-3.9.6-bin.tar.gz
drwxr-xr-x  4 root root 33 Dec 16 19:44 aws
drwxr-xr-x  2 root root 6 Aug 16 2018 rh
[root@ip-172-31-44-3 opt]# mv apache-maven-3.9.6 maven
[root@ip-172-31-44-3 opt]# ls -la
total 192
drwxr-xr-x  5 root root 77 Dec 16 09:08 .
drwxr-xr-x  1 root root 257 Dec 16 08:44 ..
-rw-r--r--  1 root root 9416868 Dec 1 11:22 apache-maven-3.9.6-bin.tar.gz
drwxr-xr-x  4 root root 33 Dec 16 19:44 aws
drwxr-xr-x  6 root root 99 Dec 16 09:07 maven
drwxr-xr-x  2 root root 6 Aug 16 2018 rh
[root@ip-172-31-44-3 opt]# ./maven/bin/mvn -
[INFO] Scanning for projects...
[INFO]
[INFO] BUILD FAILURE
[INFO]
[INFO] -----
[INFO] Total time: 0.236 s
[INFO] Finished at: 2023-12-16T09:08:46Z
[INFO] -----
[ERROR] The goal you specified requires a project to execute but there is no POM in this directory. Please verify you invoked Maven from the correct directory. --> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MissingProjectException
[root@ip-172-31-44-3 opt]# ./maven/bin/mvn -v
Apache Maven 3.9.6 (bc24f3c744dd6b6ec920b3cd080dce295161ae)
Maven home: /opt/maven
Java version: 11.0.20.0.8-1.amzn2.0.1.x86_64
Default locale: en_US, platform encoding: UTF-8
OS name: "linux" version: "5.10.261.191.748.amzn2.x86_64", arch: "amd64", family: "unix"
[root@ip-172-31-44-3 opt]# ls -la /usr/lib/jvm
-bash: s: command not found
[root@ip-172-31-44-3 opt]# ls -la /usr/lib/jvm
total 0
drwxr-xr-x  3 root root 379 Dec 16 08:49 .
drwxr-xr-x 34 root root 4096 Dec 16 08:48 ..
drwxr-xr-x  6 root root 68 Dec 16 08:49 java-11-openjdk-11.0.20.0.8-1.amzn2.0.1.x86_64
lrwxrwxrwx  1 root root 21 Dec 16 08:49 jre -> /etc/alternatives/jre
lrwxrwxrwx  1 root root 24 Dec 16 08:49 jre-11 -> /etc/alternatives/jre_11
lrwxrwxrwx  1 root root 52 Dec 16 08:49 jre-11-openjdk -> /etc/alternatives/jre_11_openjdk
lrwxrwxrwx  1 root root 44 Dec 16 08:49 jre-11-openjdk-11.0.20.0.8-1.amzn2.0.1.x86_64 -> java-11-openjdk-11.0.20.0.8-1.amzn2.0.1.x86_64
lrwxrwxrwx  1 root root 29 Dec 16 08:49 jre-openjdk -> /etc/alternatives/jre_openjdk
[root@ip-172-31-44-3 opt]# cd ~
[root@ip-172-31-44-3 ~]# nano .bash_profile
[root@ip-172-31-44-3 ~]# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
[root@ip-172-31-44-3 ~]# source .bash_profile
[root@ip-172-31-44-3 ~]# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/root/bin
[root@ip-172-31-44-3 ~]# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/root/bin:/root/bin
[root@ip-172-31-44-3 ~]# source .bash_profile
[root@ip-172-31-44-3 ~]# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/bin:/root/bin:/root/bin:/usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.amzn2.0.1.x86_64:/opt/maven:/opt/maven/bin
[root@ip-172-31-44-3 ~]#

```

After modifying `.bash_profile`, `.bash_profile` was started to running. As a result, Java and Maven directory paths were included in environment variables.

The screenshot shows the Jenkins interface for managing tools. In the top navigation bar, there are links for New, How, Blog, Facebook, Can, Assi, W4, Mav, W6, Con, Redi, ngu, Laur, Upd, Cha, Crea, and Error. Below the navigation bar, the URL is 54.174.99.13:8080/manage/configureTools/. The main content area shows two sections: 'JDK installations' and 'Git installations'. Under 'JDK installations', a form is displayed for adding a JDK. It has fields for 'Name' (set to 'JDK 11') and 'JAVA_HOME' (set to '/usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.amzn2.0.1.x86_64'). A red error message below the JAVA_HOME field states: ' /usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.amzn2.0.1.x86_64 doesn't look like a JDK directory'. There is also a checkbox for 'Install automatically'. At the bottom of this section is a 'Save' button. Under 'Git installations', there is a single entry for 'Git' with a red 'X' icon. At the bottom of this section are 'Save' and 'Apply' buttons.

Java directory path was added into Jenkins configuration, as the same path that was modified in `.bash_profile`

In the next step, Tomcat server was created to deploy the Maven project and host with Tomcat.

Security group name	Security group ID	Type	Protocol	Port range	Source	Description
Tomcat_Security_Group	sg-0f68d7bb63f07f920	HTTP	tcp	80	0.0.0.0/0	-
Tomcat_Security_Group	sg-0f68d7bb63f07f920	Custom TCP	tcp	8080	0.0.0.0/0	-
Tomcat_Security_Group	sg-0f68d7bb63f07f920	ssh	tcp	22	0.0.0.0/0	-
Tomcat_Security_Group	sg-0f68d7bb63f07f920	HTTPS	tcp	443	0.0.0.0/0	-

The security group allows 4 ports to be accessed: HTTP (80), HTTPS (443), SSH (22) and Tomcat (8080).

```
Last login: Wed Dec 20 13:27:00 UTC 2023 on pts/0
[root@tomcat-server ~]# sudo amazon-linux-extras install java-openjdk11
Topic java-openjdk11 has end-of-support date of 2024-09-30
```

Java JDK 11 was installed in the Tomcat server

```
[root@tomcat-server opt]# wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.84/bin/apache-tomcat-9.0.84.tar.gz
--2023-12-20 13:32:00--  https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.84/bin/apache-tomcat-9.0.84.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11804410 (11M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.84.tar.gz'

100%[=====] 2023-12-20 13:32:00 (152 MB/s) - 'apache-tomcat-9.0.84.tar.gz' saved [11804410/11804410]

[root@tomcat-server opt]# tar -xvzf apache-tomcat-9.0.84.tar.gz
```

Next step, we installed the Tomcat package into the Tomcat server and extracted the package in the /opt directory, which is used for containing optional software packages that are not considered as default installation.

```
[[root@tomcat-server bin]# ln -s /opt/tomcat/bin/startup.sh /usr/local/bin/tomcatup
[[root@tomcat-server bin]# ln -s /opt/tomcat/bin/shutdown.sh /usr/local/bin/tomcatdown
```

Tomcat start and shutdown command shortcuts were created for easy execution.

New credentials

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: admin

Treat username as secret

Password: (redacted)

ID: tomcat_deployer

Description: tomcat_deployer

Create

Tomcat admin credentials were added into Jenkins global credentials, which provides a secure way to manage authentication while orchestrating tasks that involve Tomcat.

The screenshot shows the Jenkins configuration interface for a job named 'BuildAndDeployJob'. The left sidebar lists various configuration sections: General, Source Code Management, Build Triggers, Build Environment, Pre Steps, Build, Post Steps, Build Settings, and Post-build Actions. The 'General' section is currently selected. It contains a 'Description' field with the text 'Build code using Maven and deploy the artifact WAR on Tomcat server'. Below this are several checkboxes for build options: 'Discard old builds', 'GitHub project', 'This project is parameterized', and 'Execute concurrent builds if necessary'. A 'Source Code Management' section is present, showing a 'None' option selected. At the bottom of the screen, there are 'Save' and 'Apply' buttons.

In the next step, Jenkins was created in order to pull code from Github repository, build the project with Maven and host the web application with Tomcat server.

Configure

Source Code Management

Repository URL: `https://github.com/nguyenhuynhquang2909/COSC2767_Assignment2_Jenkins_Docker_s3927039.git`

Credentials: `admin/******** (tomcat_deployer)`

Branches to build: `/main`

Save **Apply**

We add the Github Repository URL into the Jenkins job configuration, which looks for changes in the main branch of the repository.

Configure

Build Triggers

Build whenever a SNAPSHOT dependency is built

Schedule build when some upstream has no successful builds

Trigger builds remotely (e.g., from scripts)

Build after other projects are built

Build periodically

GitHub hook trigger for GITScm polling

Poll SCM

Schedule: `* * * * *`

Build Environment

Use secret text(s) or file(s)

Save **Apply**

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour
Would last have run at Wednesday, December 20, 2023 at 1:52:46 PM Coordinated Universal Time; would next run at Wednesday, December 20, 2023 at 1:52:46 PM Coordinated Universal Time.

In this step, build trigger was set up in order to track for changes in the main repository and execute the job automatically.

The screenshot shows the Jenkins configuration interface for a job named 'BuildAndDeployJob'. The 'Post-build Actions' section is currently active. Inside, there is a configuration for a 'Tomcat 8.x Remote' container. The 'Credentials' dropdown contains 'admin/****** (tomcat_deployer)'. The 'Tomcat URL' field is set to 'http://44.205.57.52:8080/'. A 'Deploy on failure' checkbox is present. At the bottom, there are 'Save' and 'Apply' buttons.

Next step, we configure the post-build actions by looking for WAR file, which is a packaged file used for deploying web applications to Apache Tomcat. After that, we add Tomcat admin credentials and Tomcat server URL to host the web application with Tomcat.

Step 3: Make a change on the website

This is the Assignment 2!

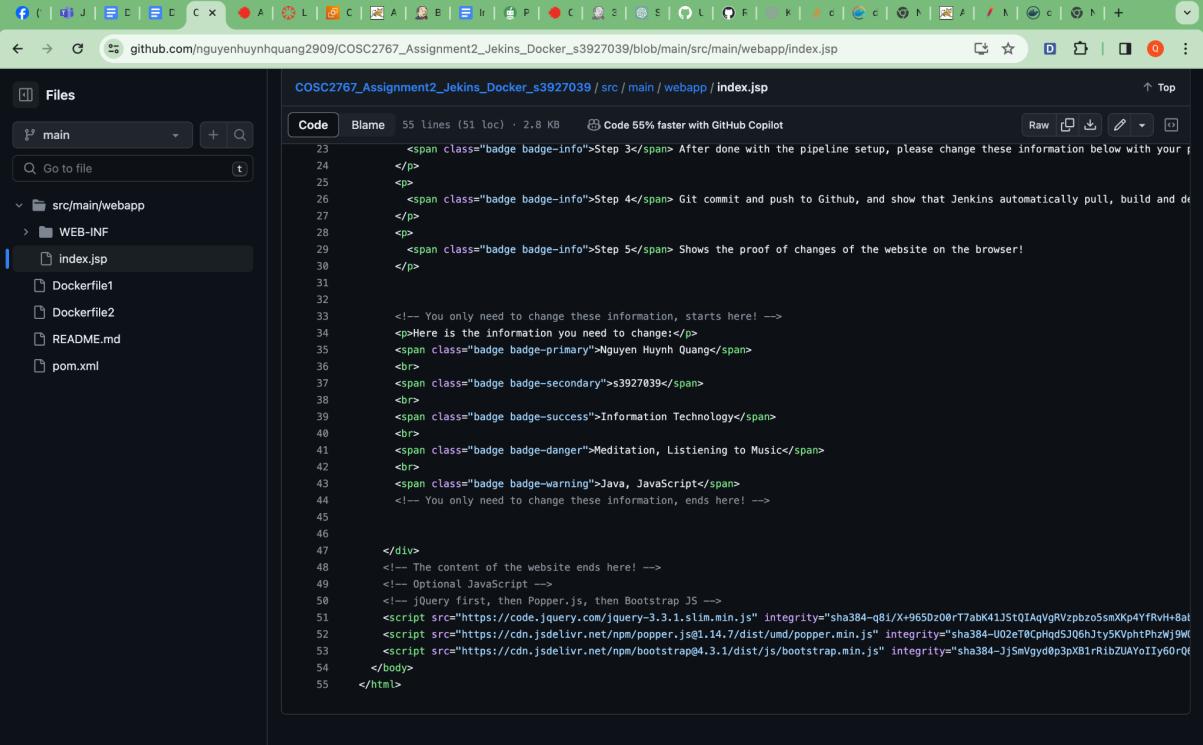
Show these steps in the video recording:

- Step 1** Please fork this github repository to your own github (with your RMIT's email account)!
- Step 2** Setup the pipeline as required in the assignment specification!
- Step 3** After done with the pipeline setup, please change these information below with your personal information on your local laptop!
- Step 4** Git commit and push to Github, and show that Jenkins automatically pull, build and deploy on the Tomcat container!
- Step 5** Shows the proof of changes of the website on the browser!

Here is the information you need to change:

Your name
 Your student ID (sID)
 Your Major (and Minor if possible)
 Your Hobby
 Your Favourite Programming Language

This is the website from the Github repository before making changes to the source code.



The screenshot shows a browser window with the URL github.com/nguyenhuynhquang2909/COSC2767_Assignment2_Jenkins_Docker_s3927039/blob/main/src/main/webapp/index.jsp. The left sidebar shows the project structure with files like Dockerfile1, Dockerfile2, README.md, and pom.xml. The main content area displays the index.jsp file's code. The code includes several placeholder spans for badge information:

```
23     <span class="badge badge-info">Step 3</span> After done with the pipeline setup, please change these information below with your own information!
24   </p>
25   <span class="badge badge-info">Step 4</span> Git commit and push to Github, and show that Jenkins automatically pull, build and deploy the website!
26   </p>
27   <span class="badge badge-info">Step 5</span> Shows the proof of changes of the website on the browser!
28   </p>
29
30
31
32   <!-- You only need to change these information, starts here! -->
33   <p>Here is the information you need to change:</p>
34   <span class="badge badge-primary">Nguyen Huynh Quang</span>
35   <br>
36   <span class="badge badge-secondary">s3927039</span>
37   <br>
38   <span class="badge badge-success">Information Technology</span>
39   <br>
40   <span class="badge badge-danger">Meditation, Listening to Music</span>
41   <br>
42   <span class="badge badge-warning">Java, JavaScript</span>
43
44   <!-- You only need to change these information, ends here! -->
45
46
47   </div>
48   <!-- The content of the website ends here! -->
49   <!-- Optional JavaScript -->
50   <!-- jQuery first, then Popper.js, then Bootstrap JS -->
51   <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8I/X+9650z00r7abK41JStQIAqVgRVzbzo5smXkp4YfRvh+8at9WJ9W" crossorigin="anonymous"></script>
52   <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js" integrity="sha384-U02eT0CpHqdSJQ6hJty5KVphPhZWj9W" crossorigin="anonymous"></script>
53   <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js" integrity="sha384-JjsMvgyd0p3pxB1rRibZUAYoIIy60rQ" crossorigin="anonymous"></script>
54
55   </body>
56 </html>
```

Update the website with my personal information and then push the change into the Github repository.

Step 4: Show the CI/CD pipeline

The screenshot shows a GitHub repository page for 'COSC2767_Assignment2_Jenkins_Docker_s3927039'. The repository has 1 branch and 0 tags. It is 26 commits ahead of 'TomHuynhSG/cosc2767-assignment2-website:main'. The commits are listed as follows:

- nguyenhuynhquang2909 Modify index.jsp (aabe161 - 1 minute ago) 28 Commits
- src/main/webapp Modify index.jsp (1 minute ago)
- Dockerfile1 Update maven_home (2 days ago)
- Dockerfile2 Modify Tomcat image (yesterday)
- README.md Create README.md (2 years ago)
- pom.xml Add Maven web template for assignment 2 (2 years ago)

The README file contains the following content:

```
cosc2767-assignment2-website

This is the Assignment 2!
```

Step 1: Please fork this github repository to your own github (with your RMIT's email account)!!

Push the change into Github repository

The screenshot shows a Jenkins console output page for the 'BuildAndDeployJob' job, run #6. The log output is as follows:

```
Started by an SCM change
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/BuildAndDeployJob
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/BuildAndDeployJob/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/nguyenhuynhquang2909/COSC2767_Assignment2_Jenkins_Docker_s3927039.git #
timeout=10
Fetching upstream changes from https://github.com/nguyenhuynhquang2909/COSC2767_Assignment2_Jenkins_Docker_s3927039.git
> git --version # timeout=10
> git --version # 'git version 2.40.1'
> git fetch --tags --force --progress --
https://github.com/nguyenhuynhquang2909/COSC2767_Assignment2_Jenkins_Docker_s3927039.git +refs/heads/*:refs/remotes/origin/*
timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision aabe1618bc20d9ebe2b610eda108acf58a3bcdef (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f aabe1618bc20d9ebe2b610eda108acf58a3bcdef # timeout=10
Commit message: "Modify index.jsp"
> git rev-list --no-walk 8b1af3d5bd18cd49e56bb38063e271463ca6e08 # timeout=10
Parsing POMs
Established TCP socket on 34047
[BuildAndDeployJob] $ /usr/lib/jvm/java-11-openjdk-11.0.20.0.8-1.amzn2.0.1.x86_64/bin/java -cp /var/lib/jenkins/plugins/maven-plugin/WEB-INF/lib/maven35-agent-1.14.jar:/opt/maven/boot/plexus-classworlds-2.7.0.jar:/opt/maven/conf/logging
jenkins.maven3.agent.Maven35Main /opt/maven /var/lib/jenkins/%jenkins%war/WEB-INF/lib/remoting-3160.vd76b_9ddd10cc.jar
/var/lib/jenkins/plugins/maven-plugin/WEB-INF/lib/maven35-interceptor-1.14.jar /var/lib/jenkins/plugins/maven-plugin/WEB-
```

Jenkins job was triggered automatically to adapt the change in the source code

Step 5: Show the website

This is the Assignment 2!

Show these steps in the video recording:

Step 1 Please fork this github repository to your own github (with your RMIT's email account)!

Step 2 Setup the pipeline as required in the assignment specification!

Step 3 After done with the pipeline setup, please change these information below with your personal information on your local laptop!

Step 4 Git commit and push to Github, and show that Jenkins automatically pull, build and deploy on the Tomcat container!

Step 5 Shows the proof of changes of the website on the browser!

Here is the information you need to change:

Nguyen Huynh Quang
s3927039
Information Technology
Meditation, Listening to Music
Java, JavaScript

The website was deployed successfully.

EC2 > Instances > i-031fe22752b0df0c1 > Connect to instance

Connect to instance [Info](#)

Connect to your instance i-031fe22752b0df0c1 (s3927039_Tomcat_Server) using any of these options

EC2 Instance Connect Session Manager SSH client EC2 serial console

Instance ID: i-031fe22752b0df0c1 (s3927039_Tomcat_Server)

Connection Type:

Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based endpoint, with a private IPv4 address.

Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based endpoint, with a private IPv4 address and a VPC endpoint.

Public IP address copied

44.205.57.52

Username: ec2-user

Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel Connect

The website has the same url with IPv4 of Tomcat server.

IV. Advanced Requirement Section

Create an EC2 Instance for Docker server.

This EC2 instance uses Docker security group, which allows port 22 for SSH and ports 8080-8090 for Docker containers.

```
[[root@docker-server ~]# yum install -y docker
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies

```

Install docker into Docker server after SSH into EC2 instance.

```

EXPLORER          Dockerfile1 X  Dockerfile2  README.md  index.jsp
COSC2767_ASSIGNMENT2_JEKINS_DOCKER_S...
src/main/webapp
WEB-INF
index.jsp
target
Dockerfile1 > ...
You, 8 hours ago | author (You)
1 # Use the OpenJDK 11 image as the base
2 FROM openjdk:11 as build
3
4 # Install Maven using apt-get in the OpenJDK image
5 RUN apt-get update && \
6     apt-get install -y maven
7
8 # At this point, the 'build' stage has Maven and Java JDK 11 installed.
9 # Now, start from the official Jenkins LTS image
10 FROM jenkins/jenkins:lts
11
12 # Switch to root user to install packages
13 USER root
14
15 # Copy Maven and JDK from the build stage into the Jenkins image
16 COPY --from=build /usr/share/maven /usr/share/maven
17 COPY --from=build /usr/lib/jvm/java-11-openjdk-amd64 /usr/lib/jvm/java-11-openjdk-amd64
18
19 # Set JAVA_HOME environment variable
20 ENV JAVA_HOME /usr/lib/jvm/java-11-openjdk-amd64
21
22 # Set M2_HOME environment variable
23 ENV M2_HOME /usr/share/maven
24

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE GITLENS COMMENTS

(base) nguyễnhuyhuquang2909@NguyenH2909sAir COSC2767_Assignment2_Jenkins_Docker_s3927039 % git commit -m "Modify Tomcat image"
[main 8b1af3d] Modify Tomcat image
1 file changed, 1 insertion(+), 1 deletion(-)
(base) nguyễnhuyhuquang2909@NguyenH2909sAir COSC2767_Assignment2_Jenkins_Docker_s3927039 % got push
zsh: command not found: got
(base) nguyễnhuyhuquang2909@NguyenH2909sAir COSC2767_Assignment2_Jenkins_Docker_s3927039 % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 312 bytes | 312.00 KIB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/nguyễnhuyhuquang2909/COSC2767_Assignment2_Jenkins_Docker_s3927039.git
 b958246..8b1af3d main --> main
(base) nguyễnhuyhuquang2909@NguyenH2909sAir COSC2767_Assignment2_Jenkins_Docker_s3927039 %

+ zsh
Java Build ...

OUTLINE
TIMELINE
MYSQL
JAVA PROJECTS
MAVEN

main ⌂ ⌂ 2 ⌂ 1 Connect

The Dockerfile was created in order to create Docker images. This Dockerfile has responsibility to build Jenkins with Maven image. First of all, it pulled and installed Java JDK 11, then installed Maven. After that, it pulled the latest version of Jenkins and copied Maven and JDK from the build stage into the Jenkins image. Finally, it copies JAVA_HOME and M2_HOME, which are directory for Java JDK 11 and Maven respectively into environment variables.

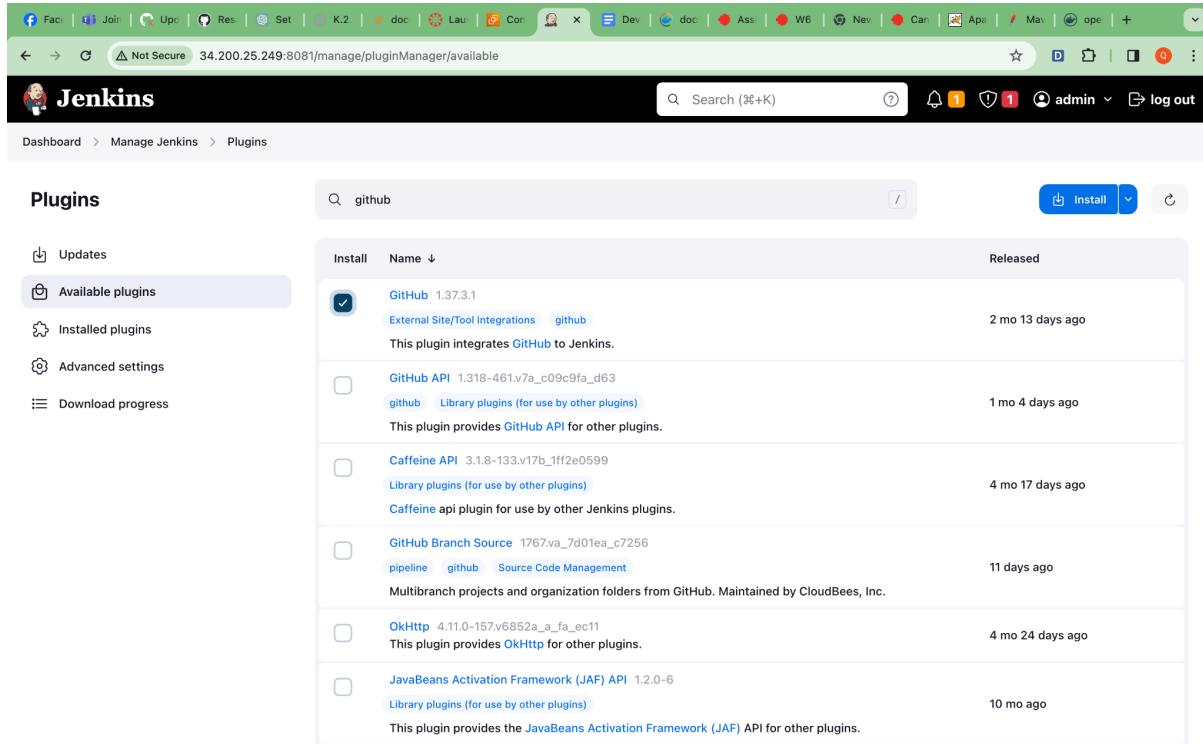
jenkins-maven	latest	3ddb0f122d77	8 hours ago	671MB
----------------------	---------------	---------------------	--------------------	--------------

Jenkins with maven image was build successfully.

```
0584f19e5e96 jenkins-maven "/usr/bin/tini -- /u..." 8 hours ago Up 24 minutes 50000/tcp, 0.0.0.0:8081->8080/tcp, :::8081->8080/tcp jenkins-container
```

We run the jenkins container from the image with the port of 8081:8080.

We print out the log of jenkins-container to get admin initial password in order to access Jenkins.



After logging in and changing admin credentials, in the next step, Github plugins was installed in order to pull code from repository.

The screenshot shows the Jenkins Plugins page. On the left, there's a sidebar with options like 'Updates', 'Available plugins' (which is selected), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main area has a search bar at the top with 'maven' typed in. Below it is a table of available plugins:

Install	Name	Released
<input checked="" type="checkbox"/>	Maven Integration 3.23 Build Tools	4 mo 27 days ago
<input type="checkbox"/>	Config File Provider 959.vcff671a_4518b_...	3 mo 13 days ago
<input type="checkbox"/>	Jira 3.12 External Site/Tool Integrations Maven jira	1 mo 7 days ago
<input type="checkbox"/>	Pipeline Maven Integration 1362.vee39a_d4b_02b_...	1 mo 19 days ago
<input type="checkbox"/>	Cobertura 1.17 Maven Build Reports	2 yr 1 mo ago

The Maven integration plugin was installed for the purpose of building the project with Maven.

The screenshot shows the Jenkins Configure Tools page. Under 'JDK installations', there's a form to add a JDK:

Add JDK

Name:

JAVA_HOME:

● /usr/lib/jvm/java-11-openjdk-amd64 doesn't look like a JDK directory

Install automatically ?

Under 'Git installations', there's a 'Save' button and an 'Apply' button.

Java environment variable was added into Jenkins configuration, which will look for JAVA_HOME directory.

Maven installations

Add Maven

Install automatically ?

Install from Apache

Version: 3.9.6

Add Installer

Save Apply

While the manual Maven installation failed, I switched to installing Maven automatically with the latest version.

```
GNU nano 2.9.8                               Dockerfile
FROM tomcat:latest
RUN cp -R /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps
COPY ./*.war /usr/local/tomcat/webapps
```

GNUnano 2.9.8

Get Help ⌘H Write Out ⌘W Where Is ⌘F Replace ⌘R Cut Text ⌘X Uncut Text ⌘U Justify ⌘J To Spell ⌘T Cur Pos ⌘C Go To Line ⌘G Read 3 lines ⌘R Undo ⌘U Redo ⌘V Mark Text ⌘M Copy Text ⌘C To Bracket ⌘B Back ⌘N Next Word ⌘F Forward ⌘P Left ⌘L Prev Word ⌘A Home ⌘H Right ⌘R Next Word ⌘E End ⌘B

Next step, we build the Dockerfile for the Tomcat image. First of all, it installed the latest version of Tomcat. After that, it copies the content from the webapp.dist directory (which is the default location of Tomcat

distribution) into the webapps directory. Finally, it copy the .war file into the webapps directory of Tomcat container.

The screenshot shows the Apache Tomcat 10.1.17 homepage. At the top, there's a navigation bar with links to Home, Documentation, Configuration, Examples, Wiki, and Mailing Lists. To the right of the navigation is a "Find Help" button. Below the navigation is the Apache logo and the text "APACHE SOFTWARE FOUNDATION". A green banner at the top says "If you're seeing this, you've successfully installed Tomcat. Congratulations!". To the left of the banner is a cartoon cat icon. To the right are three buttons: "Server Status", "Manager App", and "Host Manager". Below the banner, there's a section titled "Recommended Reading" with links to "Security Considerations How-To", "Manager Application How-To", and "Clustering/Session Replication How-To". The main content area is divided into several sections: "Developer Quick Start" (with links to Tomcat Setup, First Web Application, Realms & AAA, JDBC DataSources, Examples, Servlet Specifications, and Tomcat Versions), "Managing Tomcat" (with information about security, users, and configuration files), "Documentation" (with links to Tomcat 10.1 Documentation, Configuration, and Wiki), "Getting Help" (with links to FAQ and Mailing Lists, and a list of available mailing lists like tomcat-announce, tomcat-users, taglibs-user, and tomcat-dev), and "Release Notes", "Changelog", "Migration Guide", and "Security Notices" on the left side.

After building the Tomcat image, I created a Tomcat container with the port of 8082:8080.

```
(base) nguyenvanhquang2909@NguyenH2909sAir-s3927039_assignment2_key % ssh -i "s3927039_assignment2.pem" ec2-user@ec2-34-200-25-249.compute-1.amazonaws.com
Last login: Sat Dec 30 17:14:34 2023 from 27.79.81.107
###_
~\_\####_ Amazon Linux 2
~~ \####\ AL2 End of Life is 2025-06-30.
~~ \#/ \
~~ \#/ \
~~ \V-__>
~~ / A newer version of Amazon Linux is available!
~~ /' Amazon Linux 2023, GA and supported until 2028-03-15.
~/m/' https://aws.amazon.com/linux/amazon-linux-2023/

-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@docker-server ~]$ sudo su
Last login: Sat Dec 30 17:14:38 UTC 2023 on pts/0
[root@docker-server ~]# service docker start
Redirecting to /bin/systemctl start docker.service
[root@docker-server ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
d6ff014a16a8        tomcat-image       "catalina.sh run"   8 hours ago       Exited (255) 32 seconds ago      tomcat-container
0584f19e5e96        jenkins-maven     "/usr/bin/sh -- /u..." 8 hours ago       Exited (255) 32 seconds ago      jenkins-container
1bcd1da9b9c6        1927afe78018      "/bin/sh -c 'apt-get..." 9 hours ago       Exited (100) 9 hours ago      strange_cannon
1927afe78018        1927afe78018      "/bin/sh -c 'apt-get..." 9 hours ago       Exited (100) 9 hours ago      recursing_sinbusi
192d09721217        1927afe78018      "/bin/sh -c 'apt-get..." 9 hours ago       Exited (100) 9 hours ago      infallible_tharp
[root@docker-server ~]# docker start jenkins-maven
Error response from daemon: No such container: jenkins-maven
Error: failed to start containers: jenkins-maven
[root@docker-server ~]# docker start 06
[root@docker-server ~]# docker start d6
[root@docker-server ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              NAMES
d6ff014a16a8        tomcat-image       "catalina.sh run"   8 hours ago       Up 2 seconds      0.0.0.0:8082->8080/tcp, :::8082->8080/tcp      tomcat-container
0584f19e5e96        jenkins-maven     "/usr/bin/sh -- /u..." 8 hours ago       Up 10 seconds    50000/tcp, 0.0.0.0:8081->8080/tcp, :::8081->8080/tcp      jenkins-container
[root@docker-server ~]# passwd dockeradmin
Changing password for user dockeradmin.

New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
passwd: all authentication tokens updated successfully.

[root@docker-server ~]# id dockeradmin
uid=1001(dockeradmin) gid=1001(dockeradmin) groups=1001(dockeradmin)
[root@docker-server ~]# usermod -AG dockeradmin
```

Next step, I created dockeradmin user and create a new password for this user

```
[root@ec2-user@docker-server ~]# sudo su -
Last login: Sat Dec 30 17:14:38 UTC 2023 on pts/0
[root@docker-server ~]# service docker start
Redirecting to /bin/systemctl start docker.service
[root@docker-server ~]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[root@docker-server ~]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d0ff014a16a0 tomcat-image "catalina.sh run" 8 hours ago Exited (255) 32 seconds ago 0.0.0.0:8082->8080/tcp, :::8082->8080/tcp tomcat-container
0584f19e9e6 jenkins-maven "/usr/bin/tini -- /u..." 8 hours ago Exited (255) 32 seconds ago 50000/tcp, 0.0.0.0:8081->8080/tcp, :::8081->8080/tcp jenkins-container
1bcd1aa9b5c 192a7fe7e8b18 "/bin/sh -c 'apt-get..." 9 hours ago Exited (100) 9 hours ago
66889a23819 192a7fe7e8b18 "/bin/sh -c 'apt-get..." 9 hours ago Exited (100) 9 hours ago
70202971227 192a7fe7e8b18 "/bin/sh -c 'apt-get..." 9 hours ago Exited (100) 9 hours ago
[root@docker-server ~]# docker start jenkins-maven
Error response from daemon: No such container: jenkins-maven
Error: failed to start containers: jenkins-maven
[root@docker-server ~]# docker start 85
85
[root@docker-server ~]# docker start d6
d6
[root@docker-server ~]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d0ff014a16a0 tomcat-image "catalina.sh run" 8 hours ago Up 2 seconds 0.0.0.0:8082->8080/tcp, :::8082->8080/tcp tomcat-container
0584f19e9e6 jenkins-maven "/usr/bin/tini -- /u..." 8 hours ago Up 10 seconds 50000/tcp, 0.0.0.0:8081->8080/tcp, :::8081->8080/tcp jenkins-container
[root@docker-server ~]# passwd dockeradmin
Changing password for user dockeradmin.
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
passwd: all authentication tokens updated successfully.
[root@docker-server ~]# id dockeradmin
uid=1001(dockeradmin) gid=1001(dockeradmin) groups=1001(dockeradmin)
[root@docker-server ~]# usermod -aG dockeradmin
Usage: usermod [options] LOGIN

Options:
--comment COMMENT      new value of the GECOS field
--home HOME_DIR        new home directory for the user account
--expiredate EXPIRE_DATE
--inactive INACTIVE
--gid GROUP            force GROUP as new primary group
--groups GROUPS        new list of supplementary GROUPS
--append               append the user to the supplemental GROUPS
                      mentioned by the -G option without removing
                      him/her from other groups
--help                 display this help message and exit
--login NEW_LOGIN       new value for the login name
--lock                 lock the user account
--move-home             move contents of the home directory to the
                      new location (use only with -d)
--non-unique            allow using duplicate (non-unique) UID
--password PASSWORD    use encrypted password for the new password
--rmdir CHROOT_DIR     direct root to rmdir instead of rm -rf
--shell SHELL           new shell to run for the user account
--uid UID                new UID for the user account
--unlock               unlock the user account
--z, --selinux-user SEUSER  new SELinux user mapping for the user account

[root@docker-server ~]# usermod -aG docker dockeradmin
[root@docker-server ~]#
```

Dockeradmin was added into docker group

```
GNU nano 2.9.8                               /etc/ssh/sshd_config                                Modified
#SlogFacility AUTH
#SlogFacility AUTHPRIV
#LogLevel INFO

# Authentication:
#LoginGraceTime 2m
#PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile .ssh/authorized_keys

#AuthorizedPrincipalsFile none

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes
ChallengeResponseAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetTGTFromKerberos no
#KerberosUseKerberos yes

# GSSAPI options
#GSSAPIAuthentication yes
#GSSAPICleanupCredentials no
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no
#GSSAPIEnableGSSAPI no

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# the setting of "PermitRootLogin without-password".
#
```

In this step, password authentication was enabled to allow EC2 login with username and password.

The screenshot shows the Jenkins management interface. In the top navigation bar, the URL is 34.200.25.249:8081/manage/configure. Below it, the breadcrumb navigation shows: Dashboard > Manage Jenkins > System >. A sub-navigation bar includes 'Disable exec' and a question mark icon.

The main content area is titled 'SSH Servers'. It contains a form for adding a new SSH server:

- Name**: docker-server
- Hostname**: 34.200.25.249
- Username**: dockeradmin
- Remote Directory**: (empty field)
- Avoid sending files that have not changed

Below the form are 'Advanced' and 'Test Configuration' dropdowns, and 'Save' and 'Apply' buttons.

In this step, we added docker server IP address and docker admin credentials into Jenkins system

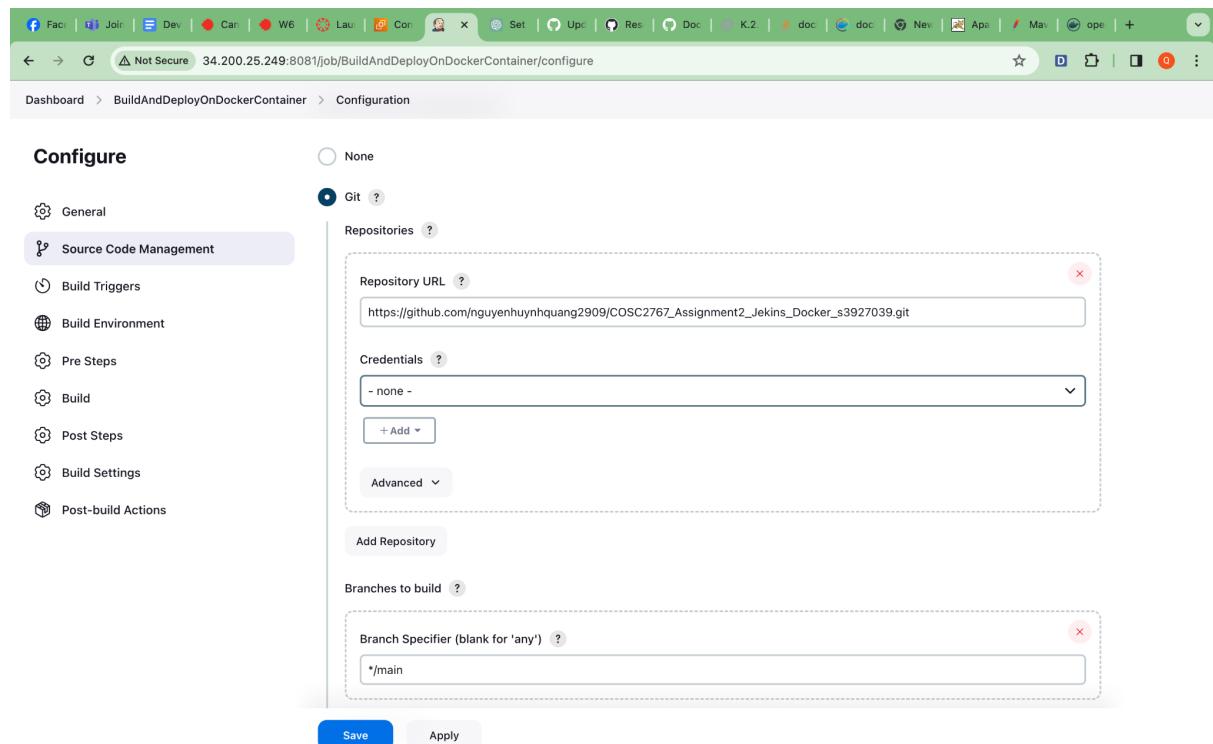
The screenshot shows the Jenkins configuration interface for a job named 'BuildAndDeployOnDockerContainer'. The URL is 34.200.25.249:8081/job/BuildAndDeployOnDockerContainer/configure. The top navigation bar includes a Jenkins logo, a search bar with 'Search (⌘+K)', and user information for 'admin'.

The configuration page has two main sections:

- Configure** sidebar:
 - General** (selected)
 - Source Code Management**
 - Build Triggers**
 - Build Environment**
 - Pre Steps**
 - Build**
 - Post Steps**
 - Build Settings**
 - Post-build Actions**
- General** configuration panel:
 - Description**: Build the project with Maven and deploy to Tomcat docker container
 - Plain text Preview**
 - Discard old builds
 - GitHub project
 - This project is parameterized
 - Execute concurrent builds if necessary
 - Advanced** dropdown
- Source Code Management** section:
 - None

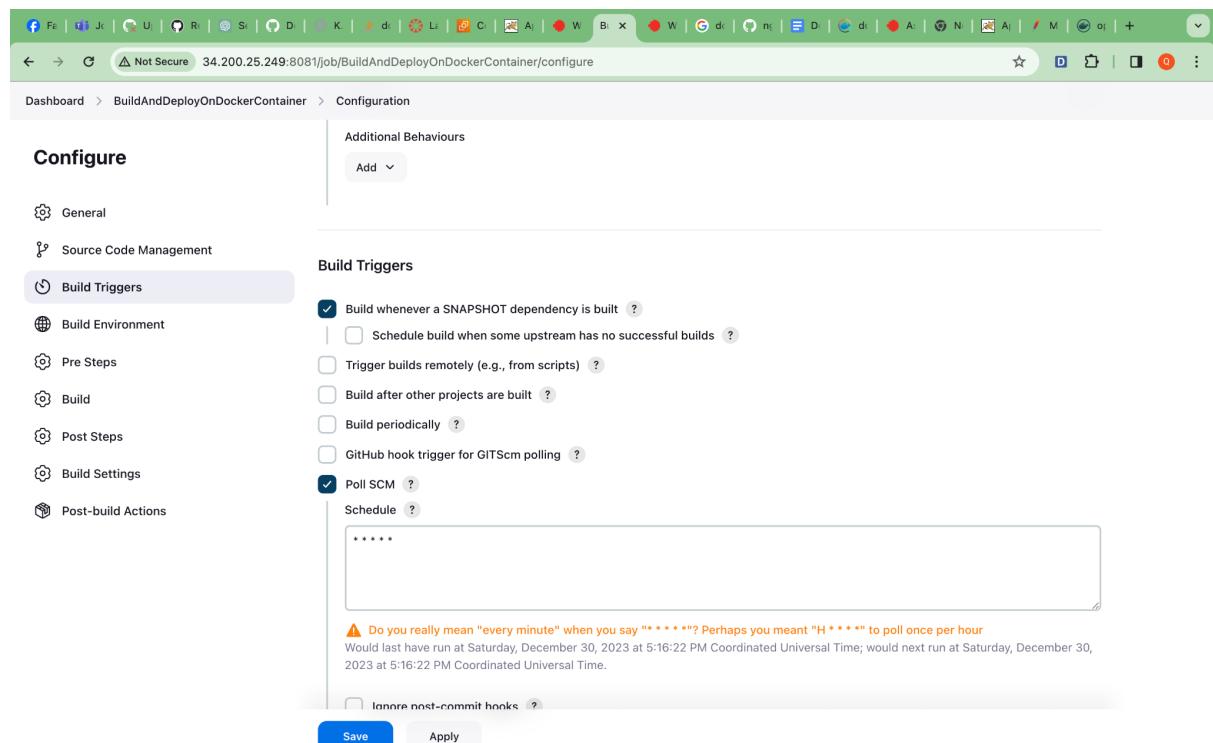
At the bottom are 'Save' and 'Apply' buttons.

Next step, we created a Jenkins job that pull source code from Github repository, builds the project with Maven and deploy the project on Tomcat container.



The screenshot shows the Jenkins job configuration page for 'BuildAndDeployOnDockerContainer'. The 'Source Code Management' section is active, with 'Git' selected. Under 'Repositories', the 'Repository URL' is set to `https://github.com/nguyenhuynhquang2909/COSC2767_Assignment2_Jenkins_Docker_s3927039.git`. The 'Branch Specifier' is set to `+/main`. There are 'Save' and 'Apply' buttons at the bottom.

Github repository URL was added into Jenkins job configuration.



The screenshot shows the Jenkins job configuration page for 'BuildAndDeployOnDockerContainer'. The 'Build Triggers' section is active, with 'Poll SCM' checked. The 'Schedule' dropdown is set to 'H * * * *'. A warning message at the bottom states: '⚠ Do you really mean "every minute" when you say "H * * * *"? Perhaps you meant "H * * * *" to poll once per hour'. There are 'Save' and 'Apply' buttons at the bottom.

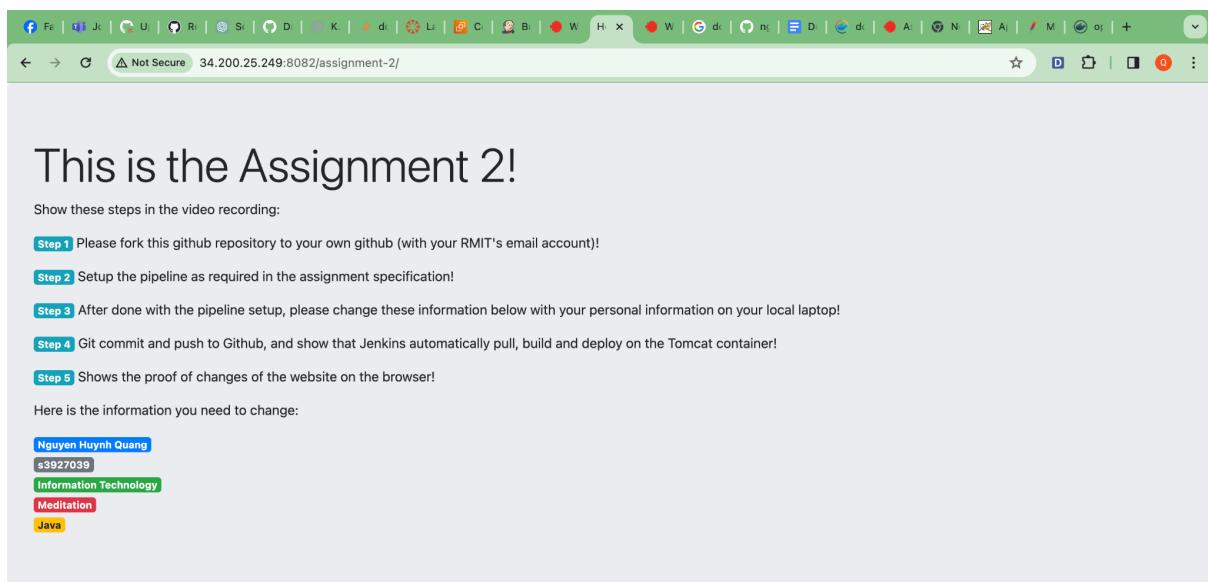
Configuring Jenkins build trigger by looking for changes in source code in every minute and every second.

The screenshot shows the Jenkins configuration interface for a job named "BuildAndDeployOnDockerContainer". The left sidebar lists various configuration sections: General, Source Code Management, Build Triggers, Build Environment, Pre Steps, Build, Post Steps, Build Settings, and Post-build Actions. The "Post-build Actions" section is currently selected and expanded. Inside, there is a "Send build artifacts over SSH" action. Under this action, the "SSH Publishers" section is expanded, showing a "SSH Server" configuration with "Name" set to "docker-server". Below it, the "Transfers" section is expanded, showing a "Transfer Set" configuration with "Source files" set to "target/*.war", "Remove prefix" set to "target", and "Remote directory" left empty. At the bottom of the configuration panel are "Save" and "Apply" buttons.

We choose docker-server as ssh server, which indicates the relative path to the .war file.

The screenshot shows the Jenkins console output for build #2. The left sidebar has "Console Output" selected. The main area displays the build log, which starts with "Started by user admin" and continues with the Maven build process, including fetching dependencies from GitHub, parsing POMs, and starting a TCP socket. The log ends with the command "/var/lib/jenkins_home/plugins/maven-plugin/WEB-INF/lib/maven35-agent-1.14.jar:/var/jenkins_home/tools/hudson.tasks.Maven_MavenInstallation/maven/boot/plexus-classworlds-2.7.0.jar:/var/jenkins_home/tools/hudson.tasks.Maven_MavenInstallation/maven/conf/logging jenkins.maven3.agent.Maven35Main /var/jenkins_home/tools/hudson.tasks.Maven_MavenInstallation/maven /var/jenkins_home/war/WEB-INF/lib/remoting-3.160.vd76b_9dd10cc.jar /var/jenkins_home/plugins/maven-plugin/WEB-INF/lib/maven35-interceptor-1.14.jar /var/jenkins_home/plugins/maven-plugin/WEB-INF/lib/maven35-interceptor-commons-1.14.jar 36405".

After that, we started to build the Jenkins job and this job ran successfully.



The Tomcat container worked as expected.

V. Conclusion

I have learned more about DevOps tools and methodologies from this assignment, particularly when it comes to creating CI/CD pipelines. The purpose of this assignment was to increase the student's understanding of Jenkins and Docker in order to construct CI/CD pipelines that enhance software development and delivery automation.

I was able to gain an understanding of the fundamentals of Docker and Jenkins through this assignment. I also learned how to build a Dockerfile and create Docker images, which was difficult because there were errors in the script that I had to correct. In addition to the knowledge of using Docker and Jenkins that this assignment required, it also called for research skills, which are critical for a DevOps Engineer.

VI. References

[1] "What is a CI/CD pipeline?"

<https://www.redhat.com/en/topics/devops/what-cicd-pipeline>

[2]M. Anastasov, “CI/CD Pipeline: A Gentle Introduction - Semaphore,” *Semaphore*, Jul. 15, 2022. <https://semaphoreci.com/blog/cicd-pipeline>

[3]C. O. Team, “Jenkins,” *Codefresh*, Aug. 30, 2023.
<https://codefresh.io/learn/jenkins/>