

RMIT International University Vietnam

Assignment Cover Page

Subject Code:	COSC2767
Subject Name:	System Deployment and Operations
Location & Campus (SGS or HN) where you study:	SGS
Title of Assignment:	Assignment 2
Student name:	Pham Ngoc Huy
Student Number:	s3926681
Teachers Name:	Tom Huynh
Group Number:	
Assignment due date:	Monday 1st, January 2024
Date of Submission:	01/01/2024
Number of pages including this one:	25

Word Count:	1115
--------------------	------

I declare that in submitting all work for this assessment I have read,
understood and agreed to the content and expectations of the
Assessment Declaration.

I give RMIT University permission to use my work as an
example and for showcase/exhibition display forever.

TABLE OF CONTENTS

1. INTRODUCTION	3
2. OBJECTIVE	3
3. MAIN CONTENT	4
4. ADVANCE REQUIREMENT.....	13
5. CONCLUSION	24
6. REFERENCES.....	25

1. INTRODUCTION

- Continuous Deployment (CD) is a component of the Continuous Delivery Pipeline that automates the movement of new functionality from a staging to production environment^[1]. It is a method of automated software release in which code updates are released to various stages as they pass predetermined tests^[2]. It is a software engineering strategy in which software features are released on a regular and automated basis^[3].
- Jenkins is a self-contained, open source automation server that may be used to automate a wide range of processes related to software development, testing, delivery, and deployment^[1]. It may be installed using native system packages, Docker, or run standalone on any machine that has a Java Runtime Environment (JRE) installed^[4]. Jenkins supports a variety of Dockerfile directives. Docker is a platform that offers reference material for its numerous APIs, command-line interfaces (CLIs), drivers and specifications, and file formats^[5]. Docker can automatically construct images by reading instructions from a Dockerfile. A Dockerfile is a text file that contains all of the commands that a user may use on the command line to build an image.

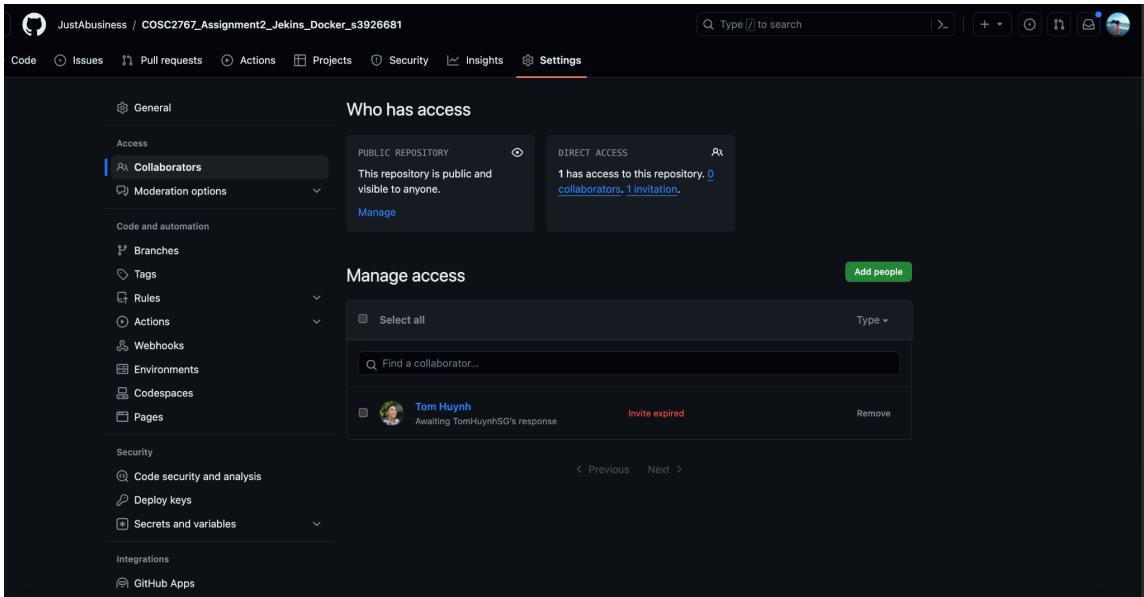
2. OBJECTIVE

- The goal of this project is to use Jenkins and Docker to create a basic Continuous Integration / Continuous Deployment (CI/CD) pipeline. Jenkins is used to integrate changes in Github repositories, which extract source code from Github and build the project with Maven. Following that, Docker will be used to manage project deployment using Tomcat.

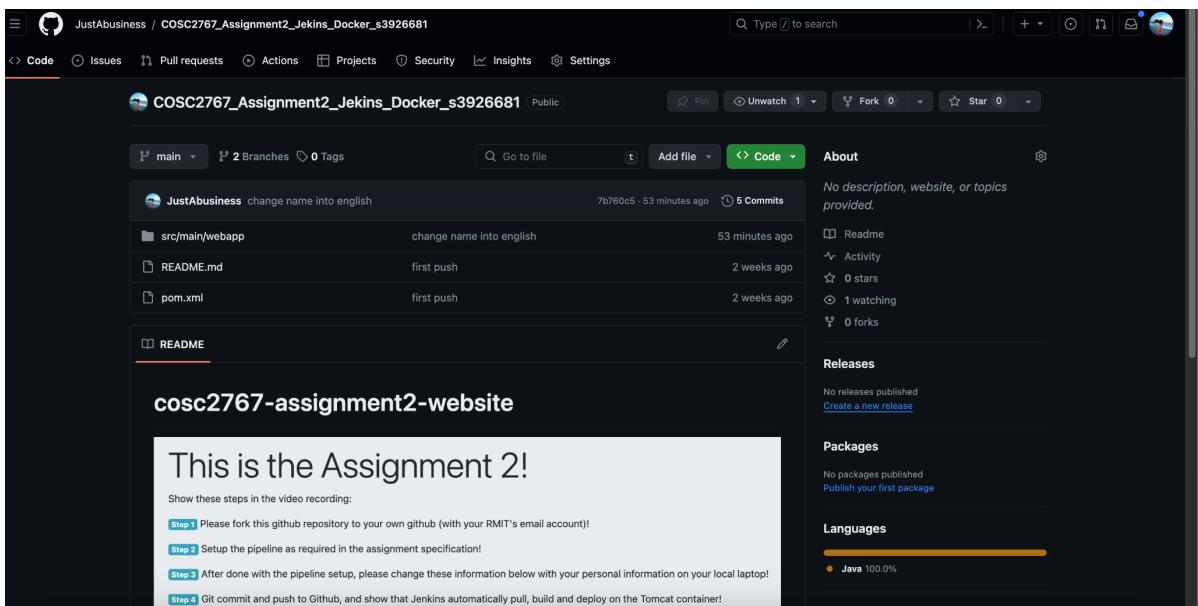
3. MAIN SECTION

Step 1: Fork Github repository

- Invite Collaborator

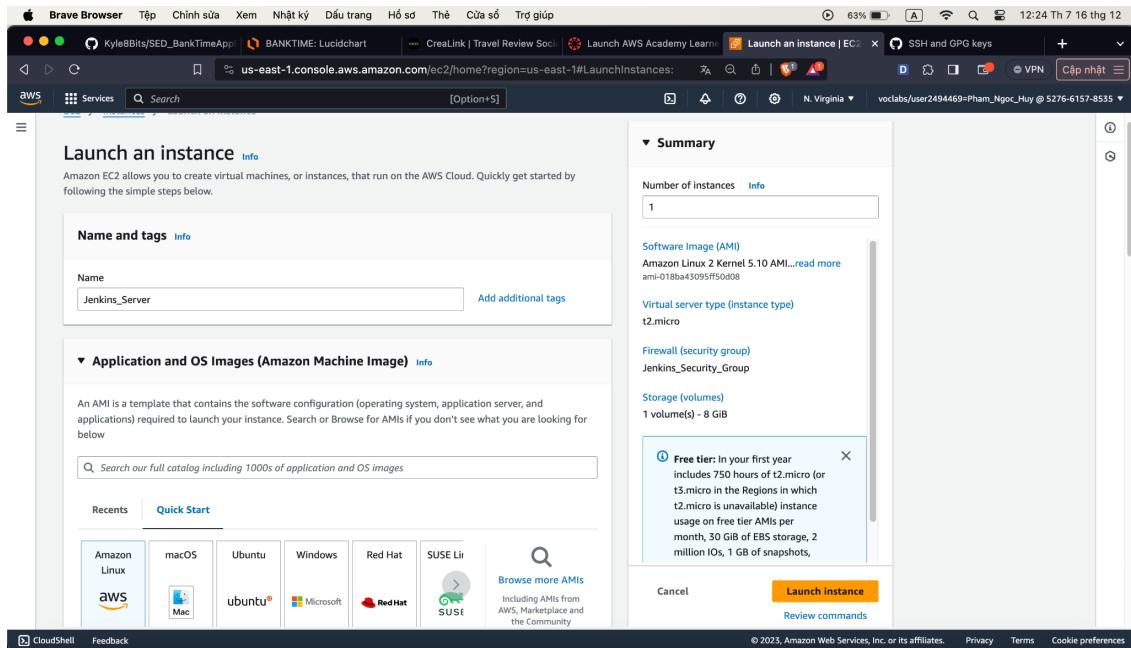


- Github repository fork successfully



Step 2: Setup the pipeline :

- Launched an EC2 instance for the Jenkins server



- Display jenkins instance on dashboard :

Instance	ID	State	Type	Check Status	Alarms	Region	AMI ID
git-devops	i-047fe18302e4a13e0	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-89-21
Jenkins_Server	i-0e043eeb3498c9972	Running	t2.micro	Initializing	No alarms	us-east-1b	ec2-52-23-11

Instance: i-0e043eeb3498c9972 (Jenkins_Server)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary

Instance ID	i-0e043eeb3498c9972 (Jenkins_Server)	Public IPv4 address	52.23.166.32 [open address]	Private IPv4 addresses	172.31.21.146
IPv6 address	-	Instance state	Running	Public IPv6 DNS	ec2-52-23-166-32.compute-1.amazonaws.com [open address]
Hostname type	IP name: ip-172-31-21-146.ec2.internal	Private IP DNS name (IPv4 only)	ip-172-31-21-146.ec2.internal		

- Connect to jenkins instance through SSH connect:

```

Tải về — ec2-user@ip-172-31-21-146:~ — ssh -i aws_academy.pem ec2-user@ec2-52-23-166-32.compute-1.amazonaws.com — 170x54
Last login: Sat Dec 16 12:11:40 on ttys098
phamngochuy@Air-cua-PHAM-3 ~ % ls
Desktop      Downloads      Movies       Pictures      Public       crealink
Documents    Library       Music        Postman     code
phamngochuy@Air-cua-PHAM-3 ~ % cd Downloads
phamngochuy@Air-cua-PHAM-3 Downloads % ls
Fira_Code_v6.2_2           Postman for macOS (x64).zip      Visual Studio Code.app      node-v18.18.2.pkg
Group Activ. 3, Group 7.docx Rectangle_Pro_v3.0.6.dmg      aws_academy.pem      s3926681.a2.docx
Group14_Sources             Sourcetree.app      crealink      unidecode-0.14.6-py3.10.egg
phamngochuy@Air-cua-PHAM-3 Downloads % ssh -i "aws_academy.pem" ec2-user@ec2-52-23-166-32.compute-1.amazonaws.com
The authenticity of host 'ec2-52-23-166-32.compute-1.amazonaws.com (52.23.166.32)' can't be established.
ED25519 key fingerprint is SHA256:aj301wgXPnnBOnyMPoFrh64zW/sNjW1zNpQ62Ycq4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'ec2-52-23-166-32.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

      #
      ####_      Amazon Linux 2
~~ \####\      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.
~/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@ip-172-31-21-146 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-21-146 ~]$ ]

```

- After that, I installed Java JDK 11, which is suitable with our project, after downloading the Jenkins package to the EC2 server.

```

[root@ip-172-31-21-146 ~]# java -version
openjdk version "11.0.20" 2023-07-18 LTS
OpenJDK Runtime Environment (Red Hat 11.0.20_0.8-1.amzn2.0.1) (build 11.0.20+8-LTS)
OpenJDK 64-Bit Server VM (Red Hat 11.0.20_0.8-1.amzn2.0.1) (build 11.0.20+8-LTS, mixed mode, sharing)
[root@ip-172-31-21-146 ~]#

```

```

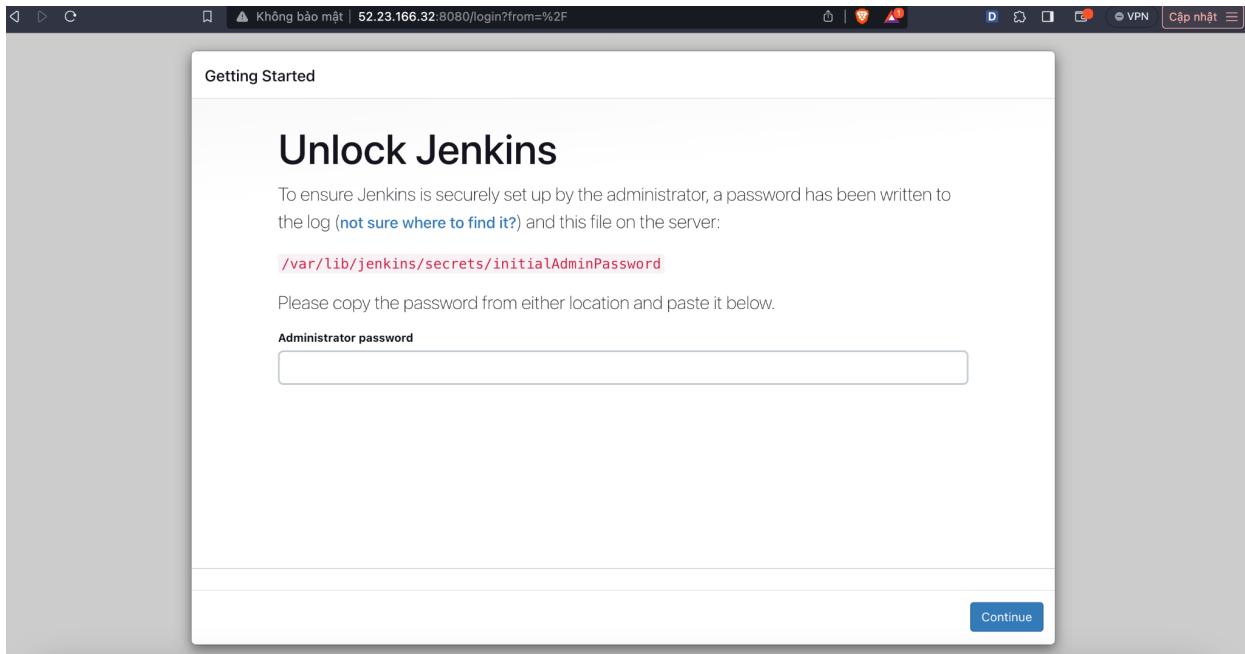
[root@ip-172-31-21-146 ~]# yum install jenkins
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package jenkins.noarch 0:2.426.2-1.1 will be installed
--> Finished Dependency Resolution
Dependencies Resolved

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

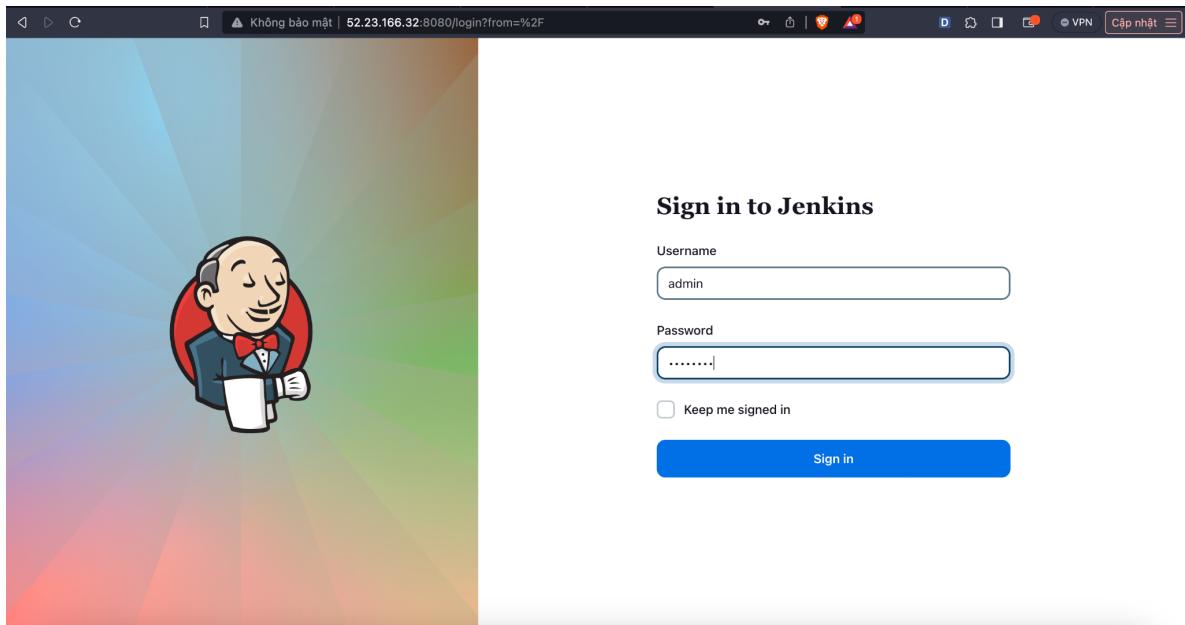
Total download size: 85 M
Installed size: 85 M
Is this ok [y/d/N]: y
Downloading packages:
jenkins-2.426.2-1.1.noarch.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : jenkins-2.426.2-1.1.noarch
  Verifying  : jenkins-2.426.2-1.1.noarch
1/1
Installed:
  jenkins.noarch 0:2.426.2-1.1
Complete!

```

- Jenkins server was successfully launched after installation, and it is now ready for the Continuous Integration (CI) task.



- Change new password, and login again with Jenkins



- In this stage, Git was installed on Jenkins' EC2 server in order to pull code from the Github repository.

```
[root@ip-172-31-21-146 ~]# yum install git
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:2.40.1-1.amzn2.0.1 will be installed
--> Processing Dependency: git-core = 2.40.1-1.amzn2.0.1 for package: git-2.40.1-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core-doc = 2.40.1-1.amzn2.0.1 for package: git-2.40.1-1.amzn2.0.1.x86_64
--> Processing Dependency: perl-Git = 2.40.1-1.amzn2.0.1 for package: git-2.40.1-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Git) for package: git-2.40.1-1.amzn2.0.1.x86_64
--> Running transaction check
--> Package git-core.x86_64 0:2.40.1-1.amzn2.0.1 will be installed
--> Package git-core-doc.noarch 0:2.40.1-1.amzn2.0.1 will be installed
--> Package perl-Git.noarch 0:2.40.1-1.amzn2.0.1 will be installed
--> Processing Dependency: perl(Error) for package: perl-Git-2.40.1-1.amzn2.0.1.noarch
--> Package perl-TermReadkey.x86_64 0:2.30-20.amzn2.0.2 will be installed
--> Running transaction check
--> Package perl-Error.noarch 1:0.17020-2.amzn2 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
```

- Installed Github plugins in Jenkins GUI, allowing Jenkins to integrate with Github.

The screenshot shows the Jenkins Plugins management interface. On the left, there's a sidebar with options: Updates (1), Available plugins, **Installed plugins**, Advanced settings, and Download progress. The main area has a search bar at the top with 'Github' typed in. Below it, a warning message says: 'Warning: This Jenkins instance requires a restart. Changing the state of plugins at this time is strongly discouraged. Restart Jenkins before proceeding.' A table lists two GitHub-related plugins:

Name	Enabled
GitHub API Plugin 1.318-461.v7a_c09c9fa_d63	<input checked="" type="checkbox"/> <input type="checkbox"/>
GitHub plugin 1.37.3.1	<input checked="" type="checkbox"/> <input type="checkbox"/>

At the bottom, a message says: '⚠ Changes will take effect when you restart Jenkins' with a 'Restart Once No Jobs Are Running' button.

- Installed the Maven package on Jenkins' EC2 machine so that Jenkins can create and manage Maven-based projects.

```
[root@ip-172-31-21-146 ~]# ls -la
total 24
drwxr-x--- 4 root root 175 Dec 29 14:39 .
drwxr-xr-x 18 root root 257 Dec 16 05:27 ..
-rw-r--r-- 1 root root 947 Dec 30 04:38 .bash_history
-rw-r--r-- 1 root root 18 Oct 18 2017 .bash_logout
-rw-r--r-- 1 root root 310 Dec 29 15:05 .bash_profile
-rw-r--r-- 1 root root 176 Oct 18 2017 .bashrc
drwxr-xr-x 4 root root 61 Dec 29 14:39 COSC2767_Assignment2_Jenkins_Docker_s3926681
-rw-r--r-- 1 root root 100 Oct 18 2017 .cshrc
drwxr----- 2 root root 80 Dec 29 14:39 .ssh
-rw-r--r-- 1 root root 129 Oct 18 2017 .tcshrc
[root@ip-172-31-21-146 ~]#
```

- The path to the Java directory was added to Jenkins settings as the same path that was adjusted in bash_profile.

- The Tomcat Server was then constructed in order to deploy the Maven project and host it with Tomcat.

- Check if the Tomcat server was configured with Java JDK 11.

```

[ec2-user@ip-172-31-17-56 ~]$ ssh -i "aws_ec2key.pem" ec2-user@ec2-54-152-242-167.compute-1.amazonaws.com
The authenticity of host 'ec2-54-152-242-167.compute-1.amazonaws.com (54.152.242.167)' can't be established.
ED25519 key fingerprint is SHA256:fsXMBddD45F0m9UjnhBjzzSxjRIM1/gab1Y+cw4Ls.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-152-242-167.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

'          #
`~\_ ##_.      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@ip-172-31-17-56 ~]$ ]

```

```

[ec2-user@ip-172-31-17-56 ~]$ java -version
openjdk version "11.0.20" 2023-07-18 LTS
OpenJDK Runtime Environment (Red Hat-11.0.20.0.8-1.amzn2.0.1) (build 11.0.20+8-LTS)
OpenJDK 64-Bit Server VM (Red Hat-11.0.20.0.8-1.amzn2.0.1) (build 11.0.20+8-LTS, mixed mode, sharing)
[ec2-user@ip-172-31-17-56 ~]$ 

```

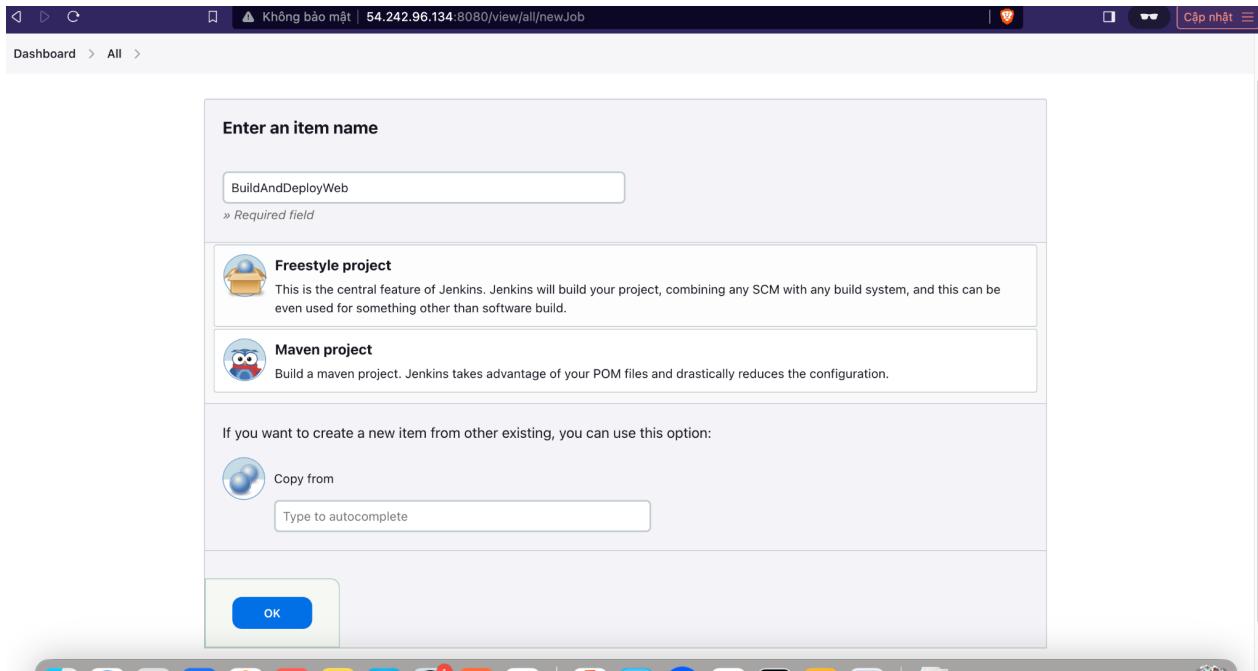
- Tomcat admin credentials have been integrated to Jenkins global credentials, providing a safe mechanism to handle authentication while orchestrating Tomcat-related jobs.

The screenshot shows the Jenkins Global credentials management interface. The URL is <http://62.23.166.32:8080/manage/credentials/store/system/domain/>. The page title is "Global credentials (unrestricted)". It displays a table with one row:

ID	Name	Kind	Description
	tomcat_deploy_website	admin/******** (tomcat_deploy_website)	Username with password

At the bottom right of the table is a blue "Edit" icon. Below the table, there are icons for "S", "M", and "L". At the very bottom of the page, it says "REST API Jenkins 2.426.2".

- Jenkins was then established to fetch code from the Github source, build the project with Maven, and host the web application with the Tomcat server.



- We add the Github Repository URL to the Jenkins task setup, which searches for changes in the repository's main branch.

- In this phase, a build trigger was configured to monitor for changes in the main repository and automatically perform the task.

Configure

- General
- Source Code Management
- Build Triggers**
- Build Environment
- Pre Steps
- Build
- Post Steps
- Build Settings
- Post-build Actions

Poll SCM

Schedule: *****

Build Environment

Save **Apply**

- The following step is to set the post-build steps by searching for a WAR file, which is a packaged file used for delivering web applications to Apache Tomcat. Then, to host the web application using Tomcat, we add Tomcat admin credentials and the Tomcat server URL.

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Pre Steps
- Build
- Post Steps
- Build Settings
- Post-build Actions**

Deploy war/ear to a container

WAR/EAR files: **/*.war

Context path:

Containers

Tomcat 9.x Remote

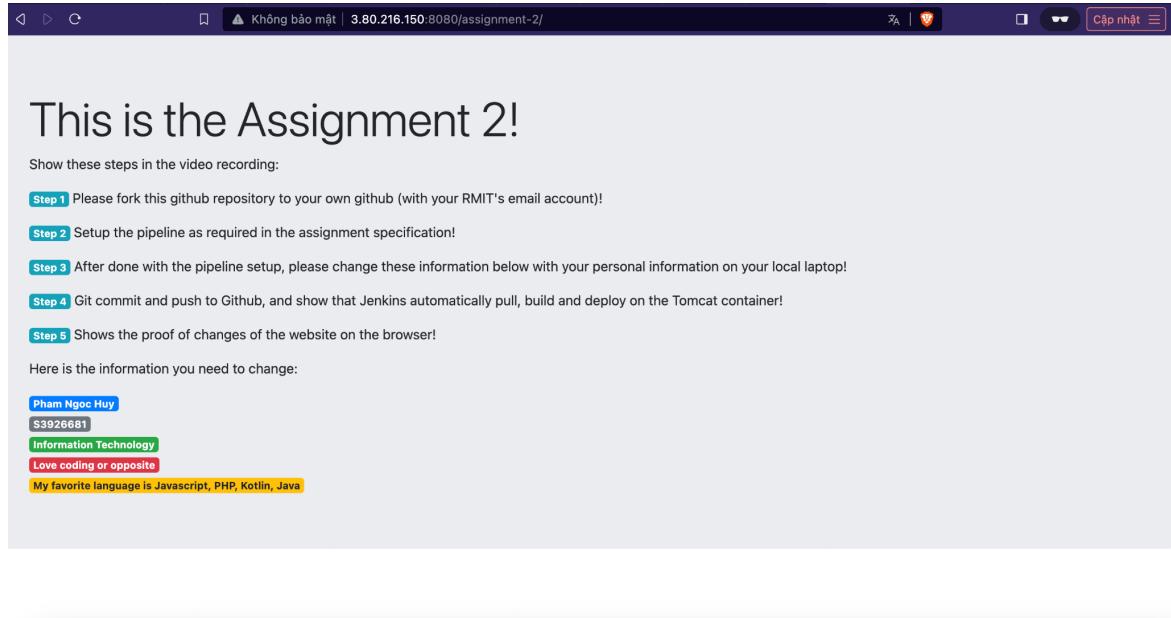
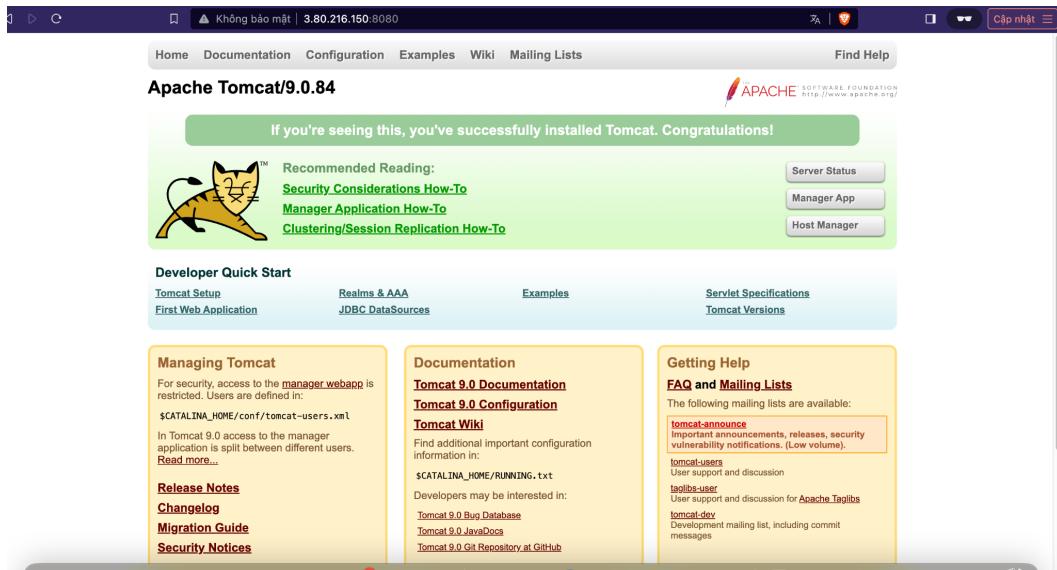
Credentials: admin/******** (tomcat_deploy_website)

Tomcat URL: http://3.80.216.150:8080/

Advanced

Save **Apply**

Step 3 : Display website :



4. ADVANCED REQUIREMENT:

- Make an EC2 instance for Docker server.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, and Network & Security. The main area displays a table of instances. One instance, 'Docker_Server_Assignment' (Instance ID: i-096633e4affc7814), is selected and highlighted in blue. The details pane for this instance shows the following information:

- Details:** Public IPv4 address: 54.196.196.105, Private IP: 172.31.19.195.
- Security:** Instance state: Running.
- Networking:** IP address: -.
- Storage:** Hostname type: ip-172-31-19-195.ec2.internal, IP name: ip-172-31-19-195.ec2.internal.
- Status checks:** Answer private resource DNS name: IPv4 (A).
- Monitoring:** Auto-assigned IP address: 54.196.196.105 (Public IP).
- VPC:** VPC ID: vpc-055be57c1acece575.

- Install docker package

```

Tài và — root@ip-172-31-19-195:~— ssh -i aws_academy.pem ec2-user@ec2-54-196-196-105.compute-1.amazonaws.com — 162x52
Group Activ. 3, Group 7.pages Sourcetree.app node-v18.18.2.pkg
Group14_Sources Visual Studio Code.app s3926681_a2.docx
phamngoc1995@Book-Air-cua-PHAM-NGOC:~% ssh -i "aws_academy.pem" ec2-user@ec2-54-196-196-105.compute-1.amazonaws.com
The authenticity of host 'ec2-54-196-196-105.compute-1.amazonaws.com (54.196.196.105)' can't be established.
ED25519 key fingerprint is SHA256:V4x4D7520wesnSaPsGeecdrt3zXWzchEJQq8uzpWh8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-196-196-105.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

# 
# Amazon Linux 2
# 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/
# 

[bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ip-172-31-19-195 ~]$ sudo su -
[root@ip-172-31-19-195 ~]# yum install docker -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 0:20.10.25-1.amzn2.0.3 will be installed
--> Processing Dependency: containerd.x86_64 >= 1.3.2 for package: docker-20.10.25-1.amzn2.0.3.x86_64
--> Processing Dependency: libgroup >= 0.40.rc1-5.15 for package: docker-20.10.25-1.amzn2.0.3.x86_64
--> Processing Dependency: runc >= 1.0.0 for package: docker-20.10.25-1.amzn2.0.3.x86_64
--> Processing Dependency: pigz for package: docker-20.10.25-1.amzn2.0.3.x86_64
--> Running transaction check
--> Package containerd.x86_64 0:1.7.2-1.amzn2.0.1 will be installed
--> Processing Dependency: libselinux >= 2.11.amzn2 will be installed
--> Package libselinux.x86_64 0:2.11.3-4.1.amzn2.0.1 will be installed
--> Package pigz.x86_64 0:1.1.7-4.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version            Repository          Size
=====
Installing:
docker            x86_64   20.10.25-1.amzn2.0.3    amzn2extra-docker  43 M
Installing for dependencies:
containerd        x86_64   1.7.2-1.amzn2.0.1    amzn2extra-docker  30 M
libgroup          x86_64   0.41-21.amzn2          amzn2-core          66 k
runc              x86_64   2.3.4-1.amzn2.0.1    amzn2-core          81 k
pigz              x86_64   1.1.7-4.amzn2          amzn2extra-docker  3.0 M

Transaction Summary
=====
```

- Pull the tomcat image from Docker Hub:

```

[root@ip-172-31-19-195 ~]# docker pull tomcat
Using default tag: latest
latest: Pulling from library/tomcat
3d181f9be59: Pull complete
bf838805bd9f: Pull complete
b39426fdaf82: Pull complete
b46f6a954797: Pull complete
fad7501a603a: Pull complete
9a85227543ee: Pull complete
9f254810c153: Pull complete
2e60b97b5a62: Pull complete
Digest: sha256:1107d758acf1bef7748969e0f84c1d071490c1da46a97ab2c04e5cd12f4e4a2b
Status: Downloaded newer image for tomcat:latest
docker.io/library/tomcat:latest
[root@ip-172-31-19-195 ~]# 
```

- Start a container based on this Tomcat image:

```
[root@ip-172-31-19-195 ~]# docker run -d --name tomcat-container -p 8081:8080 tomcat
981a2b39a64e1b6fd386c2e5efd11b52050a61e547afb02ed0c172d24aacaea
[root@ip-172-31-19-195 ~]# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
981a2b39a64e tomcat "catalina.sh run" 4 seconds ago Up 3 seconds 8080/tcp, 0.0.0.0:8081->8080/tcp, :::8081->8080/tcp tomcat-container
[root@ip-172-31-19-195 ~]#
```

- Create a dockerfile with instructions for building a Tomcat server image from a basic Centos image from Dockerhub.

```
GNU nano 2.9.8 Dockerfile Modified
^G Get Help ^A Write Out ^W Where Is ^X Cut Text ^K Uncut Text ^J Justify ^C Cur Pos ^U Undo
^X Exit ^R Read File ^N Replace ^U Go To Line ^M-^D Redo ^M-A Mark Text ^M-] To Bracket ^M-^A Previous
^C Copy Text ^M-W Whereis Next ^M-^V Next
```

- Build this Docker image from this Dockerfile :

```
[root@ip-172-31-19-195 ~]# docker build -t my-tomcat-server .
Sending build context to Docker daemon 9.216kB
Step 1/2 : FROM tomcat:latest
--> f42b2599b503
Step 2/2 : RUN cp -R /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps
--> Using cache
--> 5bc2e4613a1b
Successfully built 5bc2e4613a1b
Successfully tagged my-tomcat-server:latest
[root@ip-172-31-19-195 ~]#
```

- Using the new image, create a container.

```
[root@ip-172-31-19-195 ~]# docker run -d --name tomcat-container-2 -p 8082:8080 my-tomcat-server
81819bf1fc15a7b06109dc13321a1b29b08f1bb4e3d30b4c9c1c7faf38f2c
[root@ip-172-31-19-195 ~]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
81819bf1fc15 my-tomcat-server "catalina.sh run" 3 seconds ago Up 2 seconds 0.0.0.0:8082->8080/tcp, :::8082->8080/tcp tomcat-container-2
81a2b39a64e4 tomcat "catalina.sh run" 20 minutes ago Up 20 minutes 8080/tcp, 0.0.0.0:8081->8080/tcp, :::8081->8080/tcp tomcat-container
[root@ip-172-31-19-195 ~]#
```

- Create a new docker admin user and add that user to the docker group and may do docker-related tasks on this EC2 server.

```
[root@ip-172-31-19-195 ~]# useradd dockeradmin
[root@ip-172-31-19-195 ~]# passwd dockeradmin
Changing password for user dockeradmin.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

- Check which group the user “docker admin” is a member:

```
[root@ip-172-31-19-195 ~]# id dockeradmin
id=1001(dockeradmin) gid=1001(dockeradmin) groups=1001(dockeradmin)
```

- Add the “docker admin” users to the docker group :

```
[root@ip-172-31-19-195 ~]# usermod -aG docker dockeradmin
[root@ip-172-31-19-195 ~]#
```

- Open a new terminal window and ssh into Docker-Server as the “docker admin” user.

```
[ec2-user@ip-172-31-19-195 ~]$ ssh dockeradmin@ec2-54-196-196-105.compute-1.amazonaws.com
The authenticity of host 'ec2-54-196-196-105.compute-1.amazonaws.com (172.31.19.195)' can't be established.
ECDSA key fingerprint is SHA256:VuI/YfnKrhHK56pvpeeh8rcIuhn+R7/rryu3Neg6U.
ECDSA key fingerprint is MD5:d4:f8:82:14:48:64:03:0b:f7:48:b1:f1:f7:55:7f:30.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-54-196-196-105.compute-1.amazonaws.com,172.31.19.195' (ECDSA) to the list of known hosts.
dockeradmin@ec2-54-196-196-105.compute-1.amazonaws.com's password:
               #
               #_
               #####      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.
               _/`_ /      https://aws.amazon.com/linux/amazon-linux-2023/
[master@ip-172-31-19-195 ~]$
```

- Back to Jenkins Plugins page , look for “publish over SSH” plugins.

Name	Enabled
Publish Over SSH 1.25	<input checked="" type="checkbox"/>

Dashboard > Manage Jenkins > Plugins

Plugins

Updates 1

Available plugins

Installed plugins

Advanced settings

Download progress

- Setup Elastic Ip address for Docker server to ensure the IPv4 address cannot be changed.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations (New), Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, and Network & Security. The main area displays a table titled "Instances (1/10) Info" with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Available. Four instances are listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Available
s3926681-assignment1	i-01b6e2cd0492078ba	Running	t2.micro	2/2 checks passed	No alarms	us-east-1
Docker_Server	i-0f30d01e9c67ca834	Terminated	t2.micro	-	No alarms	us-east-1
Ansible_Server	i-004f3aa14b35da8bb	Running	t2.micro	2/2 checks passed	No alarms	us-east-1
Tomcat_Server	i-0a0609cdbfb9644	Running	t2.micro	2/2 checks passed	No alarms	us-east-1

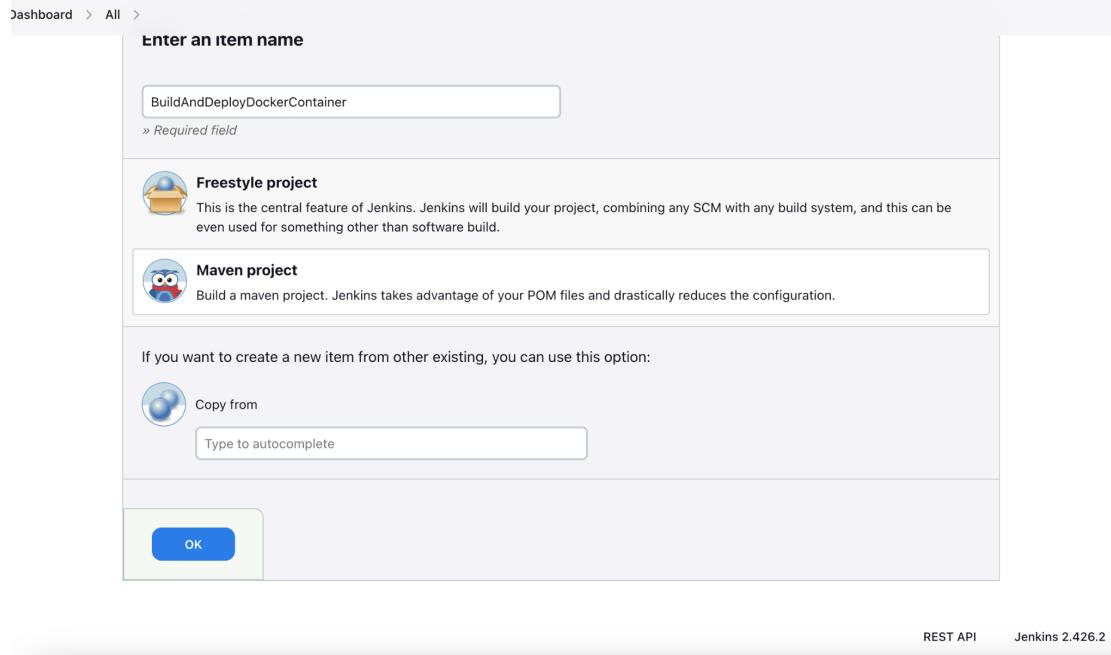
Below the table, a detailed view for the instance "i-096633e4affcf7814 (Docker_Server_Assignment)" is shown. It includes sections for Instance summary, Public IPv4 address (3.210.182.221), Private IPv4 addresses (172.31.19.195), Public IPv4 DNS (ec2-3-210-182-221.compute-1.amazonaws.com), Hostname type (IP name: ip-172-31-19-195.ec2.internal), Answer private resource DNS name (IPv4 (A)), Instance type (t2.micro), VPC ID (vpc-055be57c1acece575), and AWS Compute Optimizer finding (Opt-in to AWS Compute Optimizer for recommendations).

- Authentication of Docker using the “docker admin” user:

The screenshot shows the Jenkins Manage Jenkins > System configuration page. Under the "SSH Server" section, there are fields for Name (docker-server), Hostname (3.210.182.221), Username (dockeradmin), and Remote Directory (empty). There's also a checkbox for "Avoid sending files that have not changed". At the bottom, there are "Save" and "Apply" buttons.

Use password authentication, or use a different key
Passphrase / Password:

- Create item for build and deploy docker container



REST API Jenkins 2.426.2

- Jenkins job settings and changes:

The screenshot shows the Jenkins 'Configuration' page for the 'BuildAndDeployDockerContainer' job. The left sidebar lists 'Configure' sections: General, Source Code Management, Build Triggers, Build Environment, Pre Steps, Build, Post Steps, Build Settings, and Post-build Actions (which is selected). The main panel is titled 'Transfers' and contains a 'Transfer Set' section. It includes fields for 'Source files' (target/*.war), 'Remove prefix' (target), 'Remote directory' (empty), and an 'Exec command' box containing the following Docker build and deployment script:

```
cd /home/dockeradmin  
docker build -t tomcat:v2  
docker stop tomcat-container-final  
docker rm tomcat-container-final  
docker run -d --name tomcat-container-final -p 8087:8080 tomcat:v2
```

At the bottom, a note says: 'All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)'. There are 'Save' and 'Apply' buttons at the bottom.

- Test the job

Dashboard > BuildAndDeployDockerContainer > #4 > Console Output

```

[Tiếng Anh] [Tiếng Việt] [X]
[INFO] Assembling webapp [assignment-2] in
[/var/lib/jenkins/workspace/BuildAndDeployDockerContainer/target/assignment-2]
[INFO] Processing war project
[INFO] Copying webapp resources
[/var/lib/jenkins/workspace/BuildAndDeployDockerContainer/src/main/webapp]
[INFO] Building war: /var/lib/jenkins/workspace/BuildAndDeployDockerContainer/target/assignment-2.war
[INFO]
[INFO] --- install:3.1.1:install (default-install) @ assignment-2 ---
[INFO] Installing /var/lib/jenkins/workspace/BuildAndDeployDockerContainer/pom.xml to
/var/lib/jenkins/.m2/repository/vn/edu/rmit/assignment-2/1.0-SNAPSHOT/assignment-2-1.0-SNAPSHOT.pom
[INFO] Installing /var/lib/jenkins/workspace/BuildAndDeployDockerContainer/target/assignment-2.war to
/var/lib/jenkins/.m2/repository/vn/edu/rmit/assignment-2/1.0-SNAPSHOT/assignment-2-1.0-SNAPSHOT.war
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time:  5.698 s
[INFO] Finished at: 2023-12-30T04:28:19Z
[INFO]
[INFO] -----
[JENKINS] Archiving /var/lib/jenkins/workspace/BuildAndDeployDockerContainer/pom.xml to
vn.edu.rmit/assignment-2/1.0-SNAPSHOT/assignment-2-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/BuildAndDeployDockerContainer/target/assignment-2.war to
vn.edu.rmit/assignment-2/1.0-SNAPSHOT/assignment-2-1.0-SNAPSHOT.war
channel stopped
SSH: Connecting from host [ip-172-31-21-146.ec2.internal]
SSH: Connecting with configuration [docker-server] ...
SSH: EXEC: completed after 1,601 ms
SSH: Disconnecting configuration [docker-server] ...
SSH: Transferred 1 file(s)
Finished: SUCCESS

```

- Check the Tomcat server in port 8087:

Không bảo mật | 3.210.182.221:8087

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/10.1.17

If you're seeing this, you've successfully installed Tomcat. Congratulations!

 Recommended Reading:
[Security Considerations How-To](#)
[Manager Application How-To](#)
[Clustering/Session Replication How-To](#)

Developer Quick Start
[Tomcat Setup](#) [First Web Application](#) [Realms & AAA](#) [JDBC DataSources](#) Examples Servlet Specifications
[Tomcat Versions](#)

Managing Tomcat
For security, access to the [manager webapp](#) is restricted. Users are defined in: [\\$CATALINA_HOME/conf/tomcat-users.xml](#)
In Tomcat 10.1 access to the manager application is split between different users.
[Read more...](#)

Documentation
[Tomcat 10.1 Documentation](#) [Tomcat 10.1 Configuration](#) [Tomcat Wiki](#)
Find additional important configuration information in:
[\\$CATALINA_HOME/RUNNING.txt](#)
Developers may be interested in:
[Tomcat 10.1 Bug Database](#) [Tomcat 10.1 JavaDocs](#) [Tomcat 10.1 Git Repository at GitHub](#)

Getting Help
FAQ and Mailing Lists
The following mailing lists are available:

tomcat-announce Important announcements, releases, security vulnerability notifications. (Low volume).

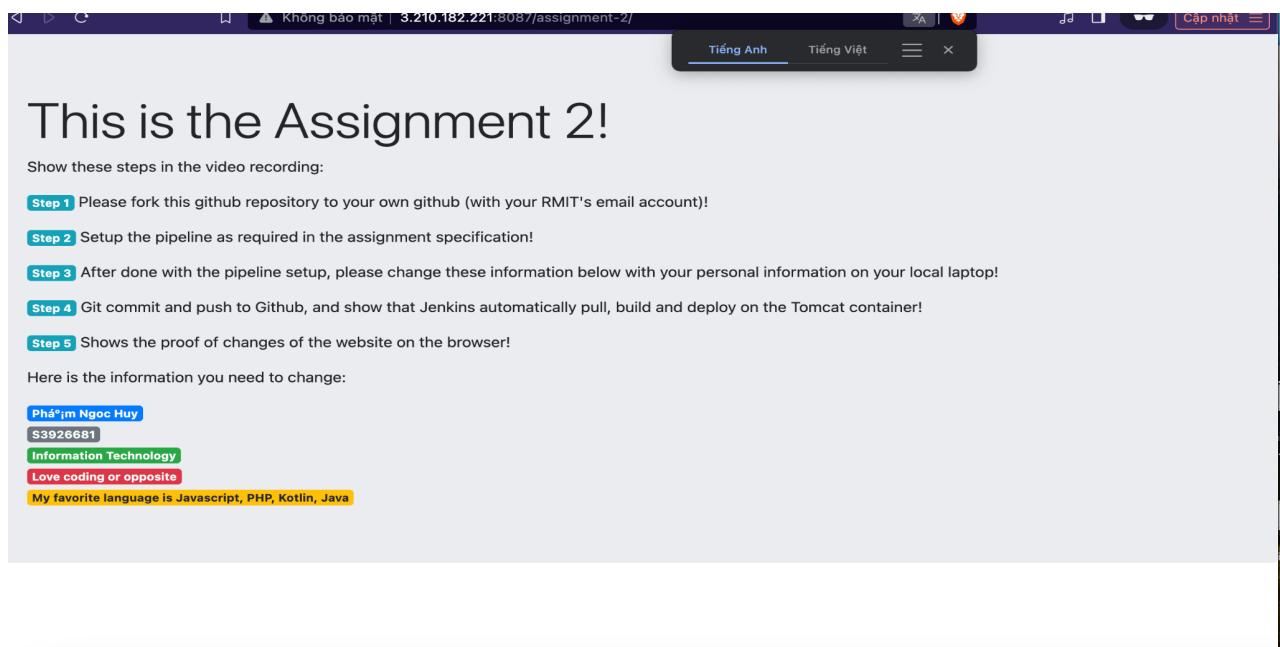
tomcat-user User support and discussion
--

taglibs-user User support and discussion for Apache Taglibs
--

tomcat-dev Development mailing list, including commit messages

<https://cwiki.apache.org/confluence/display/TOMCAT/Specifications>

- Visit the webpage :



5. CONCLUSION:

- This project gave students hands-on knowledge of how to design a CI/CD pipeline with Jenkins and Docker. It allowed me to put the theoretical information I had acquired up until Week 6 to use in a real-world context. Setting up AWS EC2 instances for Jenkins and Tomcat servers, as well as automating the deployment process, was a huge learning experience. The assignment also stressed the significance of version control services such as Github and how they interface with CI/CD workflows. I discovered how to fork a repository, clone it to my local machine, then push changes back to it.
- Reflecting on the experience, I recognize the potential of automation in software development. The ability to commit a change and have it immediately tested and deployed without any interaction is tremendously efficient. Overall, this assignment has helped me improve my DevOps skills and understand the practical aspects of CI/CD pipelines. It has prepared me for future initiatives in which I will be able to apply these abilities to increase efficiency and production.

6. REFERENCES .

+ [1] “Continuous Deployment”, *Scaled Agile*, 9 January 2023.
<https://scaledagileframework.com/continuous-deployment/>

+ [2] “The fundamentals of continuous deployment in DevOps”, *Resource Github*, May 23 2022.
<https://resources.github.com/devops/fundamentals/ci-cd/deployment/>

+ [3] “Continuous Deployment” , *Wikipedia*, 9 August 2023
https://en.wikipedia.org/wiki/Continuous_deployment

- + [4] “What is Jenkins ? “, *Jenkins*
<https://www.jenkins.io/doc/>
- + [5] “Format of the Dockerfile” , *Docker docs*
<https://docs.docker.com/engine/reference/builder/>

