

# Docker Assignment 1

## Step 1: Launched an instances for our Jenkins Master:

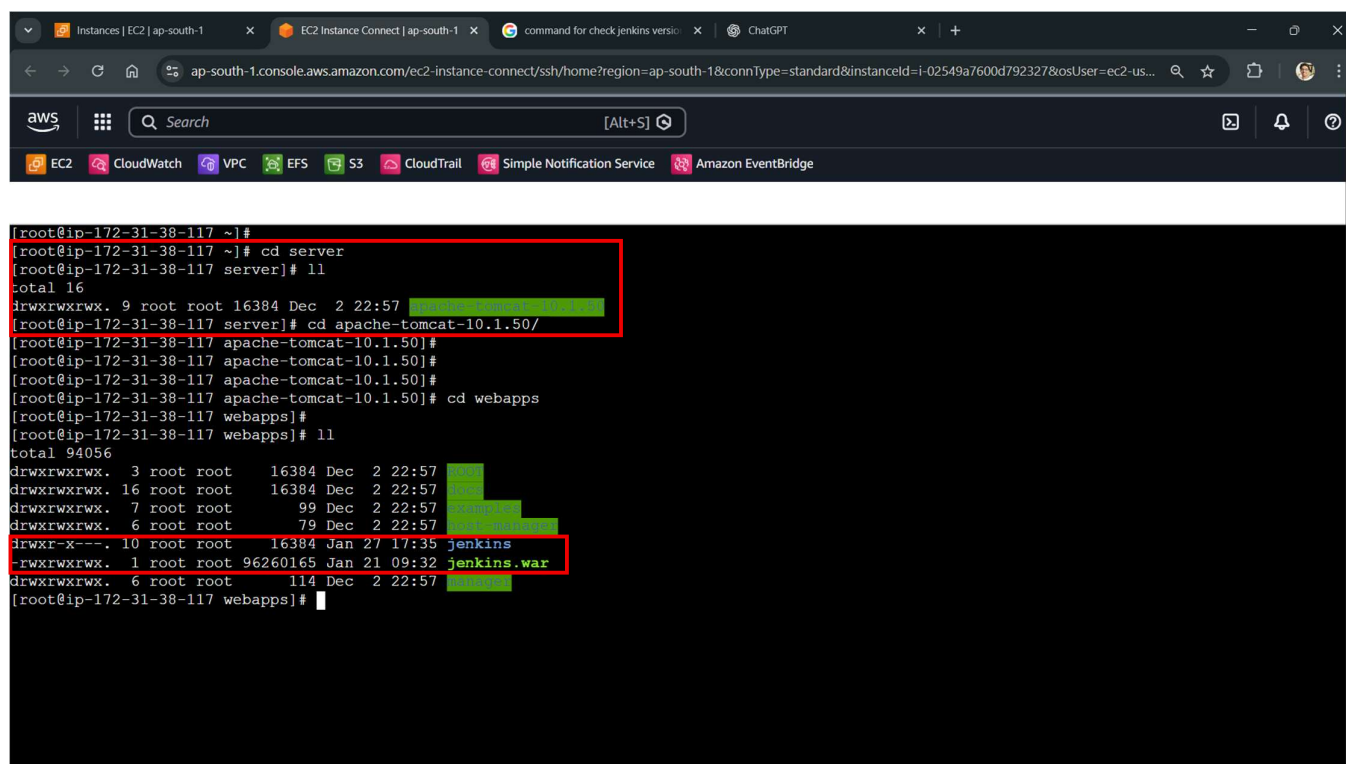
The screenshot shows the AWS Management Console for the 'ap-south-1' region. The 'Instances' page is active, displaying a table of EC2 instances. One instance, 'Linux Jenkins Master' (ID: i-0bb0216d314c17b07), is highlighted with a red box. It is in a 'Running' state and uses the 't3.micro' instance type. Below the table, the details for this instance are shown, including its public IP address (13.233.132.0) and its running state.

## Step 2: Installed Java-17, Git and Docker on the Jenkins Master Instance:

The screenshot shows the AWS Management Console for the 'ap-south-1' region. The 'EC2 Instance Connect' page is active, displaying the terminal for the instance 'Linux Jenkins Master' (ID: i-0bb0216d314c17b07). The terminal output shows the installation of Java 17, Git, and Docker, with the versions of each tool displayed.

```
[root@ip-172-31-40-216 ~]# java --version
openjdk 17.0.18 2026-01-20 LTS
OpenJDK Runtime Environment Corretto-17.0.18.9.1 (build 17.0.18+9-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.18.9.1 (build 17.0.18+9-LTS, mixed mode, sharing)
^[[B^[[A^[[B[root@ip-172-31-40-216 ~]#
[root@ip-172-31-40-216 ~]# git -v
git version 2.50.1
[root@ip-172-31-40-216 ~]#
[root@ip-172-31-40-216 ~]# docker -v
Docker version 25.0.14, build 0bab007
[root@ip-172-31-40-216 ~]#
[root@ip-172-31-40-216 ~]#
```

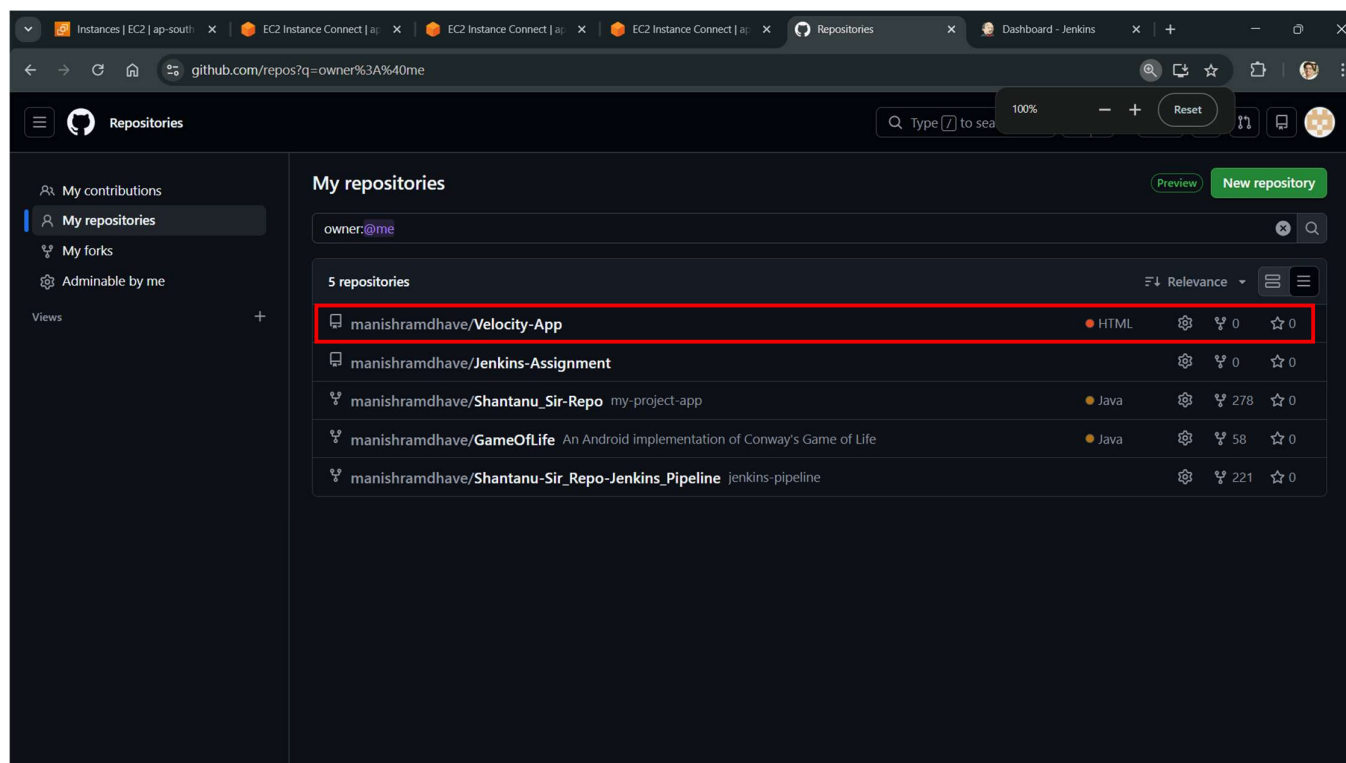
### Step 3: Installed Apache-Tomcat-10 and Jenkins on the Jenkins Master Instance:



The screenshot shows the AWS Management Console with a terminal window open for an EC2 instance. The terminal output shows the following commands and results:

```
[root@ip-172-31-38-117 ~]#  
[root@ip-172-31-38-117 ~]# cd server  
[root@ip-172-31-38-117 server]# ll  
total 16  
drwxrwxrwx. 9 root root 16384 Dec 2 22:57 apache-tomcat-10.1.50/  
[root@ip-172-31-38-117 server]# cd apache-tomcat-10.1.50/  
[root@ip-172-31-38-117 apache-tomcat-10.1.50]#  
[root@ip-172-31-38-117 apache-tomcat-10.1.50]#  
[root@ip-172-31-38-117 apache-tomcat-10.1.50]# cd webapps  
[root@ip-172-31-38-117 webapps]#  
[root@ip-172-31-38-117 webapps]# ll  
total 94056  
drwxrwxrwx. 3 root root 16384 Dec 2 22:57 jenkins  
drwxrwxrwx. 16 root root 16384 Dec 2 22:57 jenkins.war  
drwxrwxrwx. 7 root root 99 Dec 2 22:57 jenkins.war  
drwxrwxrwx. 6 root root 79 Dec 2 22:57 jenkins.war  
drwxr-xr-x. 10 root root 16384 Jan 27 17:35 jenkins  
-rwxrwxrwx. 1 root root 96260165 Jan 21 09:32 jenkins.war  
drwxrwxrwx. 6 root root 114 Dec 2 22:57 jenkins.war  
[root@ip-172-31-38-117 webapps]#
```

### Step 4: Made a Private Repository named 'Velocity-App' in GitHub account:



Step 5: Created three branches, **2026Q1**, **2026Q2** and **2026Q3** in the ‘**Velocity-App**’ Repository and pushed **three different ‘index.html’** files and also created **three different ‘Jenkinsfile’** files in respective branches:

The image displays three screenshots of a GitHub repository named 'Velocity-App'. Each screenshot shows a different branch: 2026Q1, 2026Q2, and 2026Q3. In each branch, the 'Jenkinsfile' and 'index.html' files are highlighted with a red box, indicating updates. The 2026Q1 branch shows updates to both files 33 minutes ago. The 2026Q2 branch shows updates to both files 33 minutes ago. The 2026Q3 branch shows updates to both files 2 days ago.

Branch	File	Update	Time
2026Q1	Jenkinsfile	Update Jenkinsfile	36 minutes ago
	index.html	Update index.html	33 minutes ago
2026Q2	Jenkinsfile	Update Jenkinsfile	54 minutes ago
	index.html	Update index.html	33 minutes ago
2026Q3	Jenkinsfile	Update Jenkinsfile	1 hour ago
	index.html	2026Q3	2 days ago

Step 7: Launched the Jenkins and created three different Pipeline Jobs:

The image shows the Jenkins dashboard with three pipeline jobs listed. The jobs are highlighted with a red box. The jobs are: Doc-Assign1-Pipeline-A, Doc-Assign1-Pipeline-B, and Doc-Assign1-Pipeline-C. The dashboard also shows the Jenkins logo, a search bar, and a 'New Item' button.

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☁	Doc-Assign1-Pipeline-A	35 min #9	41 min #7	8.3 sec
✓	☁	Doc-Assign1-Pipeline-B	44 sec #18	14 min #16	6.8 sec
✗	☁	Doc-Assign1-Pipeline-C	N/A	49 min #5	1 min 0 sec

Icon: S M L

REST API Jenkins 2.541.1

Step 6: Created an **API Connection between Jenkins to GitHub** Repositories in 'Manage Jenkins' by creating a **Secret Text (Credential)** using a GitHub Token in Jenkins:

The screenshot shows the Jenkins 'Manage Jenkins' configuration page for GitHub. The 'GitHub Servers' section is highlighted with a red box. It contains a 'GitHub Server' configuration with the following fields:

- Name:** GitHub-Server
- API URL:** https://api.github.com
- Credentials:** Git (selected from a dropdown menu)
- Manage hooks:** ☒

At the bottom of the configuration section are 'Save' and 'Apply' buttons.

Step 8: Integrated **all three Git branches with Jenkins in each pipeline job respectively** by creating 'Credentials' by using the Git Token on both the pipeline jobs:

The screenshot shows the Jenkins 'Configure' page for a pipeline job. The 'Repository URL' and 'Credentials' fields are highlighted with a red box. The 'Repository URL' is 'https://github.com/manishramdhav/VelocityEngine'. The 'Credentials' dropdown is set to 'ec2-user/\*\*\*\*\* (Git\_token)'. Below, the 'Branches to build' section shows two branch specifiers: '\*/\*master' and '\*/\*2026Q1'. 'Save' and 'Apply' buttons are at the bottom.

Step 9: Used **‘Pipeline script from SCM’** and selected the SCM as a **‘Git’** which will follow the Jenkins pipeline script by using the **‘Jenkinsfile’** in the repository **in each pipeline job**:

The screenshot shows the Jenkins configuration page for a pipeline named 'Doc-Assign8-Pipeline-C'. The left sidebar has a 'Configure' section with options: General, Triggers, Pipeline (selected), and Advanced. The main area is titled 'Pipeline' and contains the following fields:

- Definition:** Pipeline script from SCM (dropdown)
- SCM:** Git (dropdown)
- Repositories:** A list containing one repository with the URL `https://github.com/manishramdhav/Velocity_App.git` and credentials `Manish/***** (Git)`.

Buttons for 'Save' and 'Apply' are at the bottom.

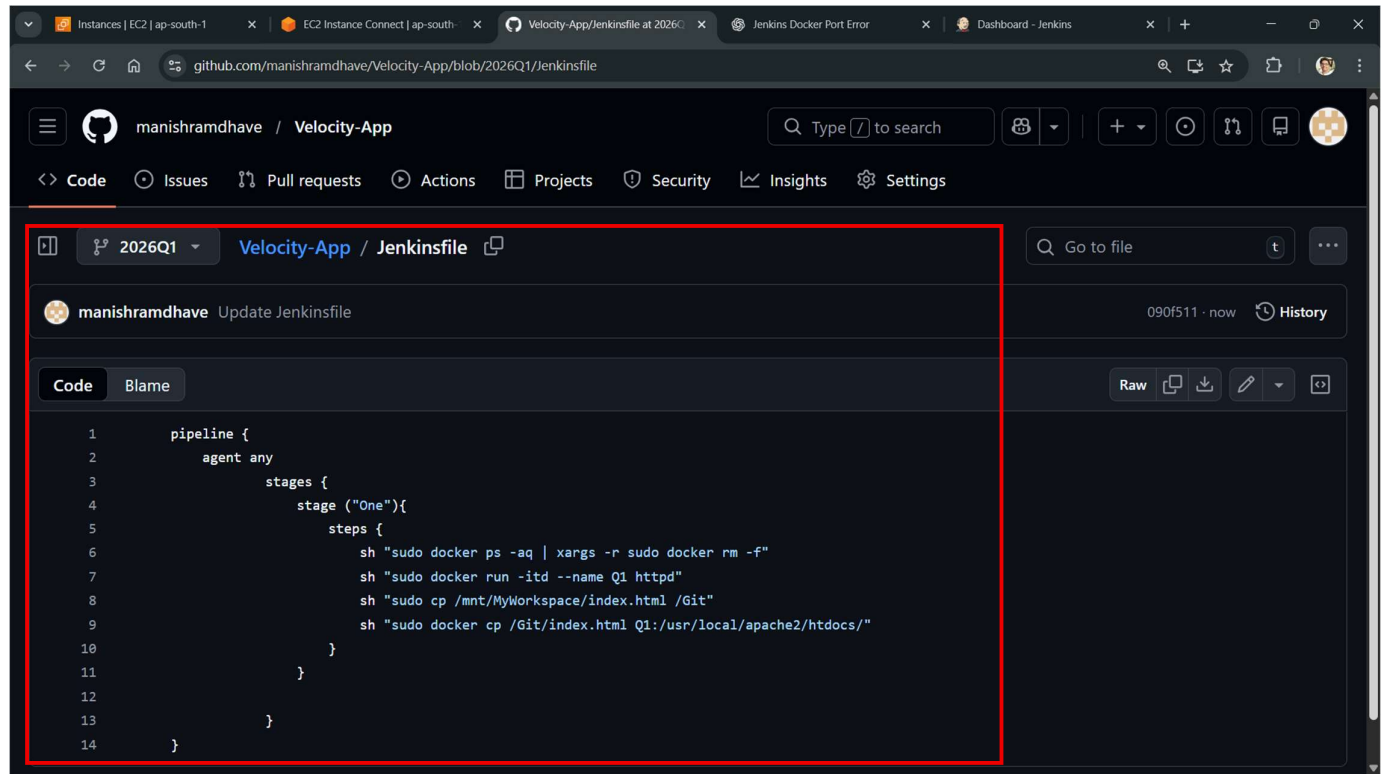
Step 10: Integrated all Git branches, **2026Q1**, **2026Q2** and **2026Q3** with Jenkins by creating **‘Credentials’** by using the Git Token on **both the pipeline Job-A and Job-B** respectively:

The screenshot shows the Jenkins configuration page for a pipeline named 'Doc-Assign2-Pipeline-A'. The left sidebar has a 'Configure' section with options: General, Triggers, Pipeline (selected), and Advanced. The main area is titled 'Repository' and contains the following fields:

- Repository URL:** `https://github.com/manishramdhav/Velocity-App.git`
- Credentials:** `ec2-user/***** (Git_token)`
- Advanced:** A dropdown menu.
- + Add Repository:** A button.
- Branches to build:** A list containing two branches: `*/master` and `*/2026Q1`.

Buttons for 'Save' and 'Apply' are at the bottom.

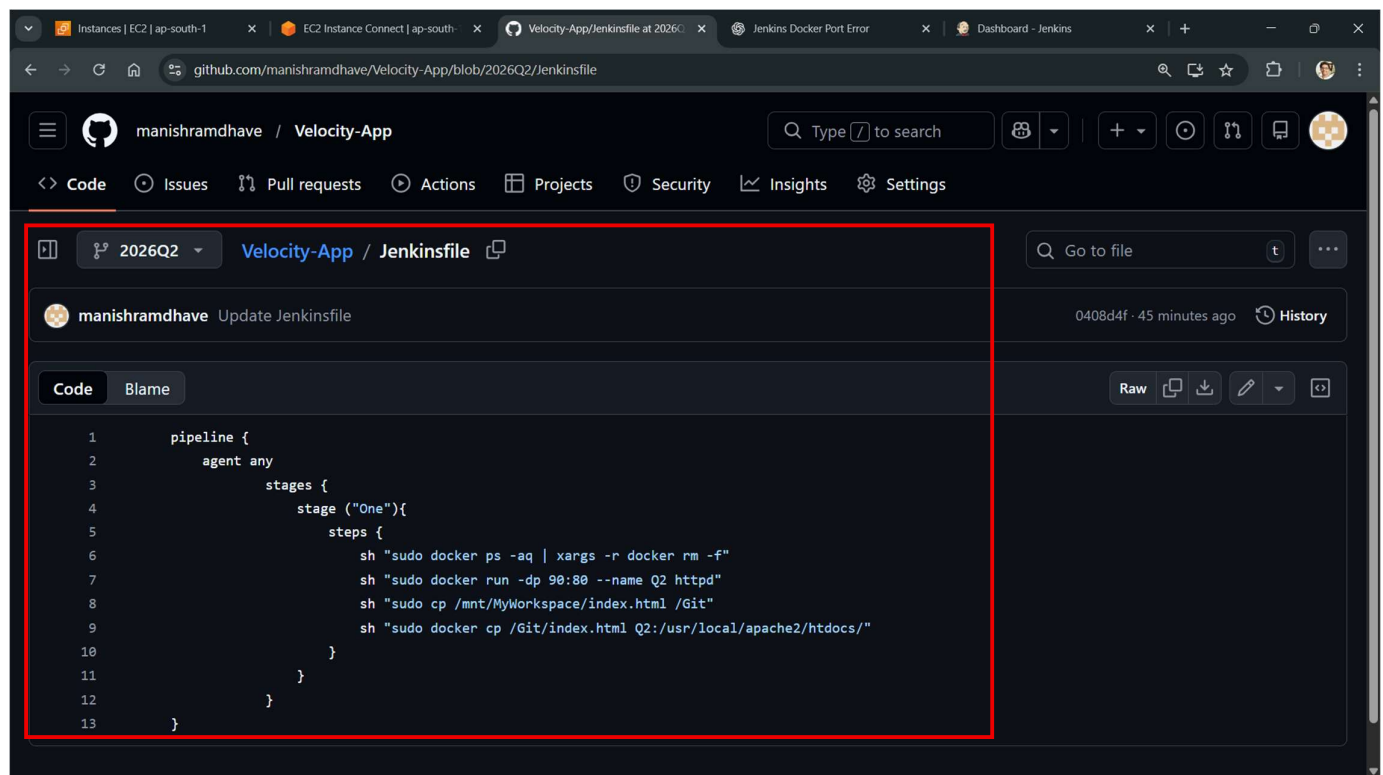
Step 11: Created a 'Jenkinsfile' on 2026Q1 branch in which the script is mentioned:



The screenshot shows the GitHub web interface for the repository 'manishramdhare / Velocity-App'. The '2026Q1' branch is selected, and the 'Jenkinsfile' is open in the 'Code' tab. The file content is as follows:

```
1 pipeline {
2   agent any
3   stages {
4     stage ("One"){
5       steps {
6         sh "sudo docker ps -aq | xargs -r sudo docker rm -f"
7         sh "sudo docker run -itd --name Q1 httpd"
8         sh "sudo cp /mnt/MyWorkspace/index.html /Git"
9         sh "sudo docker cp /Git/index.html Q1:/usr/local/apache2/htdocs/"
10      }
11    }
12  }
13 }
14 }
```

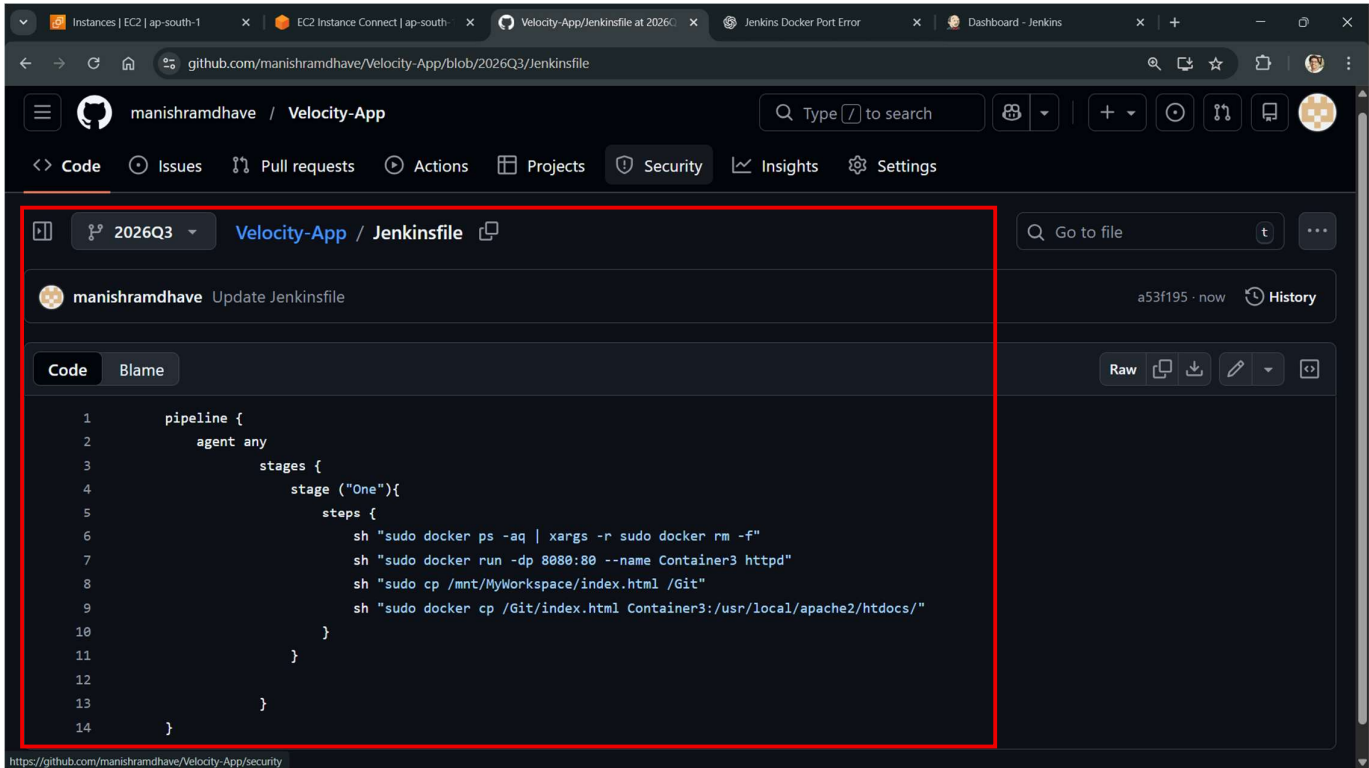
Step 12: Created a 'Jenkinsfile' on 2026Q2 branch in which the script is mentioned:



The screenshot shows the GitHub web interface for the repository 'manishramdhare / Velocity-App'. The '2026Q2' branch is selected, and the 'Jenkinsfile' is open in the 'Code' tab. The file content is as follows:

```
1 pipeline {
2   agent any
3   stages {
4     stage ("One"){
5       steps {
6         sh "sudo docker ps -aq | xargs -r docker rm -f"
7         sh "sudo docker run -dp 90:80 --name Q2 httpd"
8         sh "sudo cp /mnt/MyWorkspace/index.html /Git"
9         sh "sudo docker cp /Git/index.html Q2:/usr/local/apache2/htdocs/"
10      }
11    }
12  }
13 }
```

Step 13: Created a 'Jenkinsfile' on 2026Q3 branch in which the script is mentioned:



The screenshot shows a GitHub web interface for the repository 'manishramdhare / Velocity-App'. The '2026Q3' branch is selected, and the file 'Jenkinsfile' is open. The file content is a Jenkins pipeline script. The script defines a pipeline with an 'any' agent and a single stage named 'One'. This stage contains three steps: running 'docker ps' to list containers, running 'docker run' to start a container named 'Container3' with 'httpd' on port 8080, and copying 'index.html' from the local workspace to the container's file system.

```
1 pipeline {
2   agent any
3   stages {
4     stage ("One"){
5       steps {
6         sh "sudo docker ps -aq | xargs -r sudo docker rm -f"
7         sh "sudo docker run -dp 8080:80 --name Container3 httpd"
8         sh "sudo cp /mnt/MyWorkspace/index.html /Git"
9         sh "sudo docker cp /Git/index.html Container3:/usr/local/apache2/htdocs/"
10      }
11    }
12  }
13 }
14 }
```



## Results:

1. When changes are done in 2026Q1 branch, Build is triggered by **'Doc-Assign1-Pipeline-A'** and the updated index.html file is hosted from the **container Q1** by following the Pipeline script in the **'Jenkinsfile'** file of the same branch:

The screenshot displays two browser windows. The left window shows the GitHub repository for 'Velocity-App' with the '2026Q1' branch selected. The 'index.html' file is highlighted, showing its content: 'Hosting updated index.html from Container 1 of 2026Q1 Branch change 1 Change 2'. The right window shows the Jenkins build interface for 'Doc-Assign1-Pipeline-A' #7, which was triggered by a GitHub push. The build status is 'Completed' (green checkmark) and it took 6.8 seconds.

2. When changes are done in 2026Q2 branch, Build is triggered by **'Doc-Assign1-Pipeline-B'** and the updated index.html file is hosted from the **container Q2** by following the Pipeline script in the **'Jenkinsfile'** file of the same branch:

The screenshot displays two browser windows. The left window shows the GitHub repository for 'Velocity-App' with the '2026Q2' branch selected. The 'index.html' file is highlighted, showing its content: 'Hosting updated index.html from Container 2 of 2026Q2 Branch change 1 Manish here Done with Docker Assignment 1'. The right window shows the Jenkins build interface for 'Doc-Assign1-Pipeline-B' #8, which was triggered by a GitHub push. The build status is 'Completed' (green checkmark) and it took 6.7 seconds.



3. When changes are done in 2026Q3 branch, we have modified our **index.html** file to run it on **'Port no.8080'**. Its Build should be triggered by **'Doc-Assign1-Pipeline-C'** and the updated **index.html** file should be hosted from the **container Q3** by following the Pipeline script in the **'Jenkinsfile'** file of the same branch.

But as our **Jenkins Master** is running on **Tomcat Server** whose default port no. is 8080, it is only reserved by the Jenkins and it gives the error that **'Listen tcp4 0.0.0.0:8080:bind: address already in use.'** Hence the **index.html** file from **2026Q3** branch will not host from the **container Q3**.

The image displays a Jenkins build console and a browser window. The Jenkins console shows a build failure for 'Doc-Assign1-Pipeline-C' with the error message: 'docker: Error response from daemon: driver failed programming external connectivity on endpoint Container3 (6aca3b046cd47ebac9265b54864c4b93730509a5d4a65e99dd75e86685adce7a): Error starting userland proxy: listen tcp4 0.0.0.0:8080: bind: address already in use.' The browser window shows an 'HTTP Status 404 – Not Found' error for the requested resource [/index.html].

**Jenkins Console Output:**

```
docker: Error response from daemon: driver failed programming
external connectivity on endpoint Container3
(6aca3b046cd47ebac9265b54864c4b93730509a5d4a65e99dd75e86685adce
7a): Error starting userland proxy: listen tcp4 0.0.0.0:8080:
bind: address already in use.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
```

**Browser Error Message:**

**HTTP Status 404 – Not Found**

**Type:** Status Report

**Message:** The requested resource [/index.html] is not available

**Description:** The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.

**Apache Tomcat/10.1.52**