

Docker Assignment 1

Step 1: Launched an instances for our Jenkins Master:

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with 'EC2' selected, followed by 'Dashboard', 'AWS Global View', and 'Events'. Under 'Instances', there are sections for 'Instances Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', and 'Capacity Manager'. Below these are 'Images' (AMIs, AMI Catalog) and 'Elastic Block Store'. The main content area is titled 'Instances (1/3) Info'. It has a search bar and filters for 'Name', 'Instance ID', 'Instance state', 'Instance type', 'Status check', 'Alarm status', and 'Availability zone'. A red box highlights the table row for 'Linux Jenkins Master' (Instance ID: i-0bb0216d314c17b07, State: Running, Type: t3.micro). Below the table, a detailed view for 'i-0bb0216d314c17b07 (Linux Jenkins Master)' is shown with tabs for 'Details', 'Status and alarms', 'Monitoring', 'Security', 'Networking', 'Storage', and 'Tags'. The 'Details' tab is active, showing 'Instance ID: i-0bb0216d314c17b07', 'Public IPv4 address: 13.233.132.0', 'Private IPv4 addresses: 172.31.40.216', 'Public DNS: ec2-13-233-132-0.ap-south-1.compute.amazonaws.com', and 'Instance state: Running'. The bottom of the page includes links for 'CloudShell', 'Feedback', and 'Console Mobile App', along with copyright information for 2026, Amazon Web Services, Inc. or its affiliates.

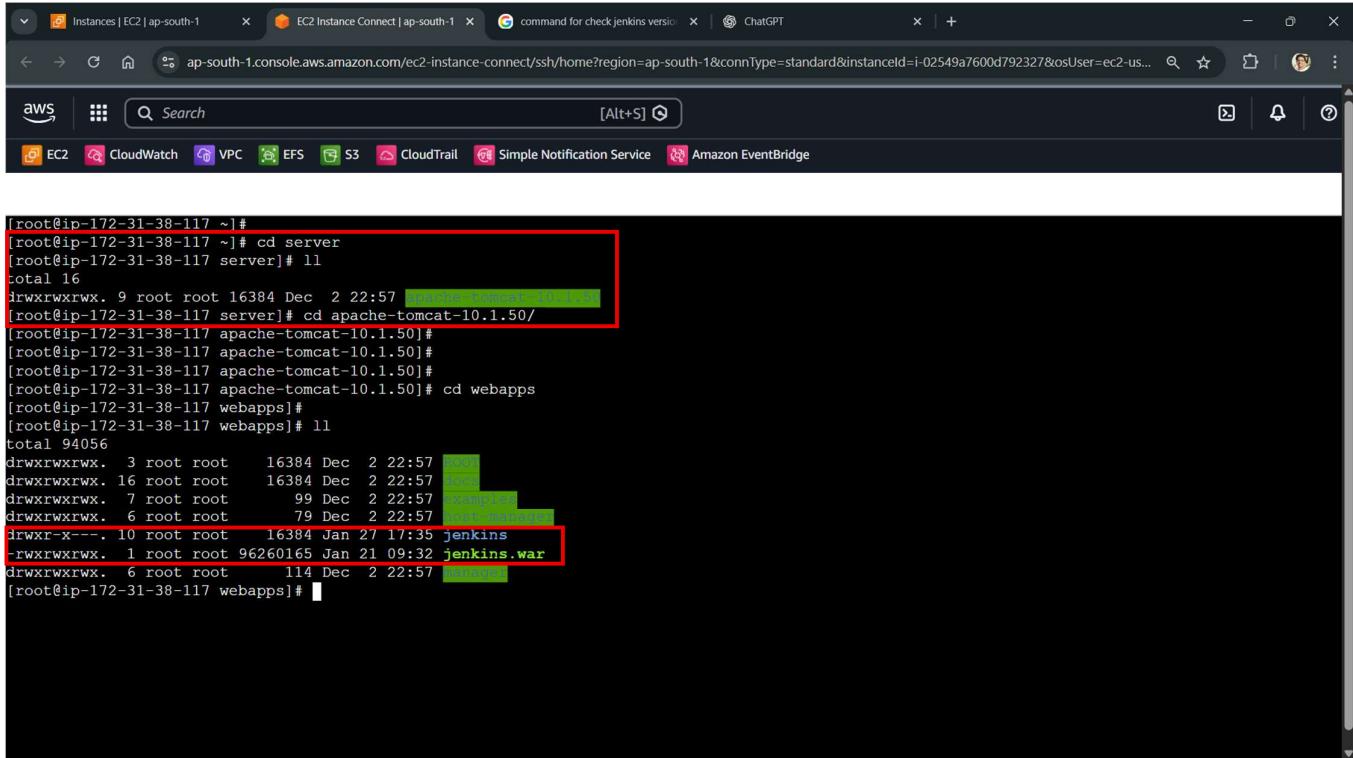
Step 2: Installed Java-17, Git and Docker on the Jenkins Master Instance:

The screenshot shows the AWS EC2 Instance Connect terminal. The top navigation bar is highlighted with a red box, showing 'Search', 'ASK Amazon', and other service icons. The main terminal window displays a root shell session on the Jenkins Master instance. The user has installed Java 17, checked the git version, and installed Docker. The terminal output is as follows:

```
[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 ~]#
[root@ip-172-31-40-216 ~]# ]
```

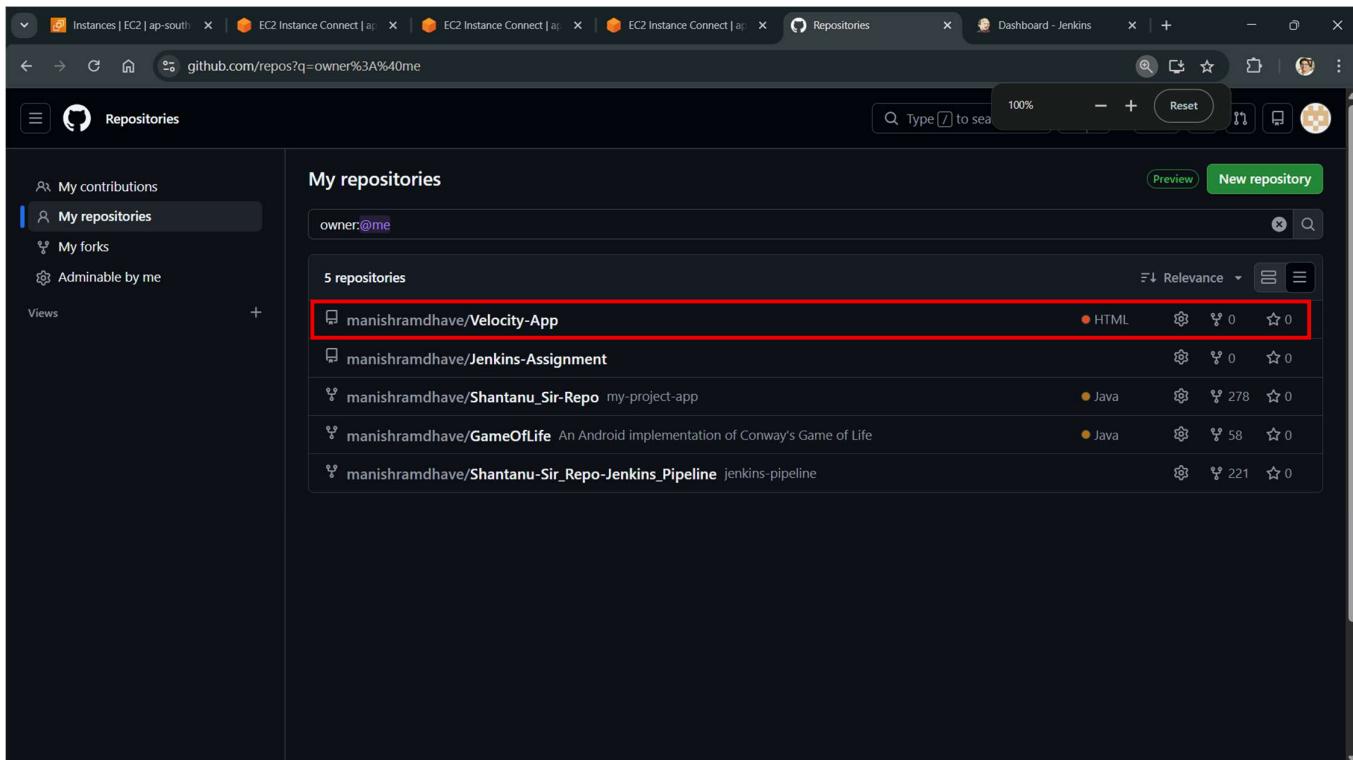
At the bottom of the terminal, a modal window for the instance details is open, showing the instance ID 'i-0bb0216d314c17b07 (Linux Jenkins Master)', Public IP '13.233.132.0', Private IP '172.31.40.216', and a close button. The bottom of the screen also includes links for 'CloudShell', 'Feedback', and 'Console Mobile App', along with copyright information for 2026, Amazon Web Services, Inc. or its affiliates.

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



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

Step 4: Made a Private Repository named ‘Velocity-App’ in GitHub account:



The screenshot shows a GitHub repository list page. The sidebar on the left has options: My contributions, My repositories (which is selected and highlighted in blue), My forks, and Adminalbe by me. The main area is titled "My repositories" and shows a search bar with "owner:@me". Below it, there is a table with the following data:

Repository	Language	Issues	Stars
manishramdhave/Velocity-App	HTML	0	0
manishramdhave/Jenkins-Assignment		0	0
manishramdhave/Shantanu_Sir-Repo	my-project-app	278	0
manishramdhave/GameOfLife	An Android implementation of Conway's Game of Life	58	0
manishramdhave/Shantanu-Sir_Repo-Jenkins_Pipeline	jenkins-pipeline	221	0

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 contains three separate screenshots of a GitHub repository interface, each showing a different branch: 2026Q1, 2026Q2, and 2026Q3. Each screenshot highlights the commit history for those branches, specifically focusing on the 'Jenkinsfile' and 'index.html' files.

2026Q1 Branch Details:

- Commits: 56 Commits
- Latest Commit: 5b64415 · 33 minutes ago
- Files Updated:
 - Jenkinsfile (Update Jenkinsfile, 36 minutes ago)
 - index.html (Update index.html, 33 minutes ago)

2026Q2 Branch Details:

- This branch is 14 commits ahead of and 55 commits behind 2026Q1.
- Latest Commit: Beab65d · 33 minutes ago
- Files Updated:
 - Jenkinsfile (Update Jenkinsfile, 54 minutes ago)
 - index.html (Update index.html, 33 minutes ago)

2026Q3 Branch Details:

- This branch is 7 commits ahead of and 76 commits behind 2026Q1.
- Latest Commit: 16feb28 · 1 hour ago
- Files Updated:
 - 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:

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

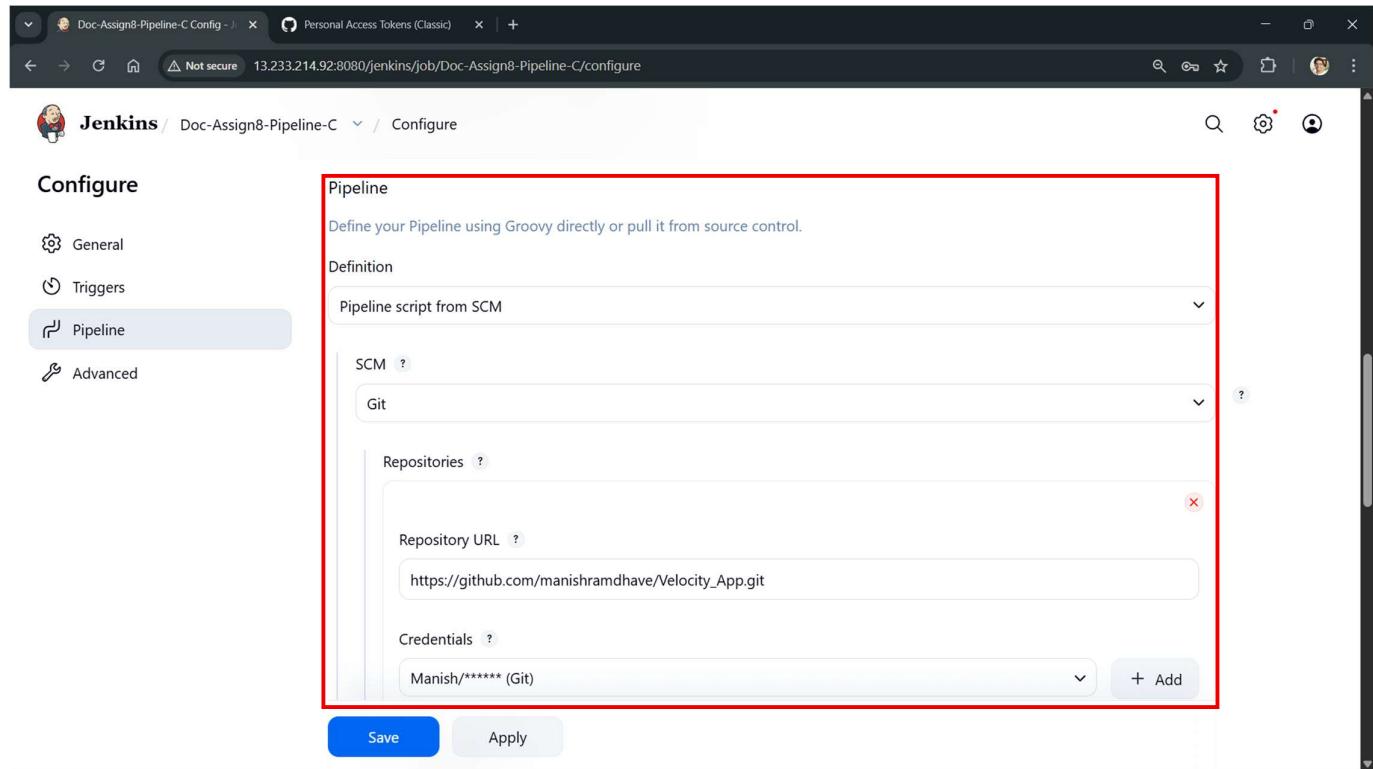
Step 6: Created an API Connection between Jenkins to GitHub Repositories in ‘Manage Jenkins’ by creating a Secret Text (Credential) using a GitHub Token in Jenkins:

The screenshot shows the Jenkins 'Manage Jenkins' interface with a red box highlighting the 'GitHub' configuration section. Inside this box, the 'GitHub Servers' section is visible, showing a single server named 'GitHub-Server' with the API URL set to 'https://api.github.com'. Below this, the 'Credentials' section is also highlighted with a red box, showing a single entry named 'Git'. At the bottom of the page 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:

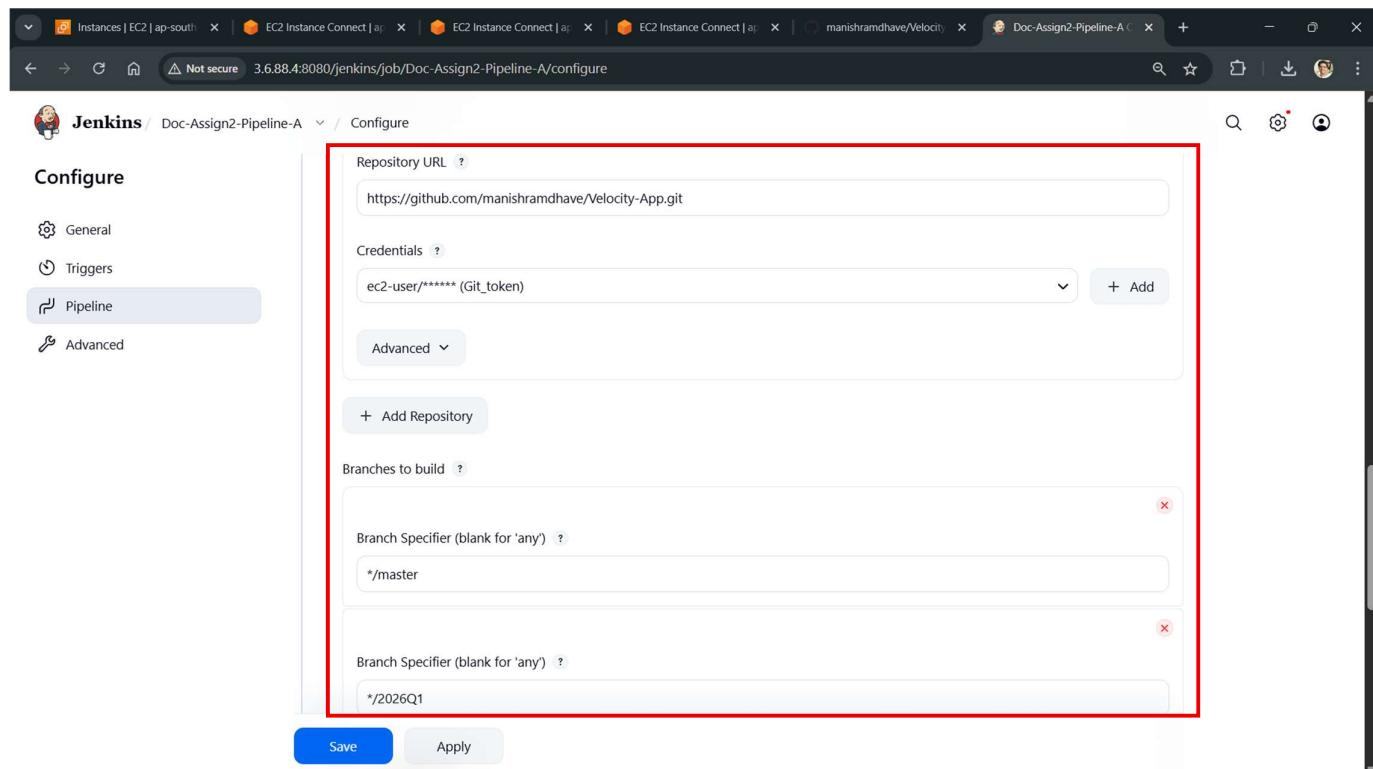
The screenshot shows the Jenkins pipeline job configuration page for 'Doc-Assign2-Pipeline-A'. A red box highlights the 'Repository URL' field containing 'https://github.com/manishramdhave/Velocity-App.git' and the 'Credentials' dropdown menu which is expanded to show 'ec2-user/******** (Git_token)'. Below this, the 'Branches to build' section is shown with two branch specifiers: '*/*master' and '*/*2026Q1'. At the bottom of the page are 'Save' and 'Apply' buttons.

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:



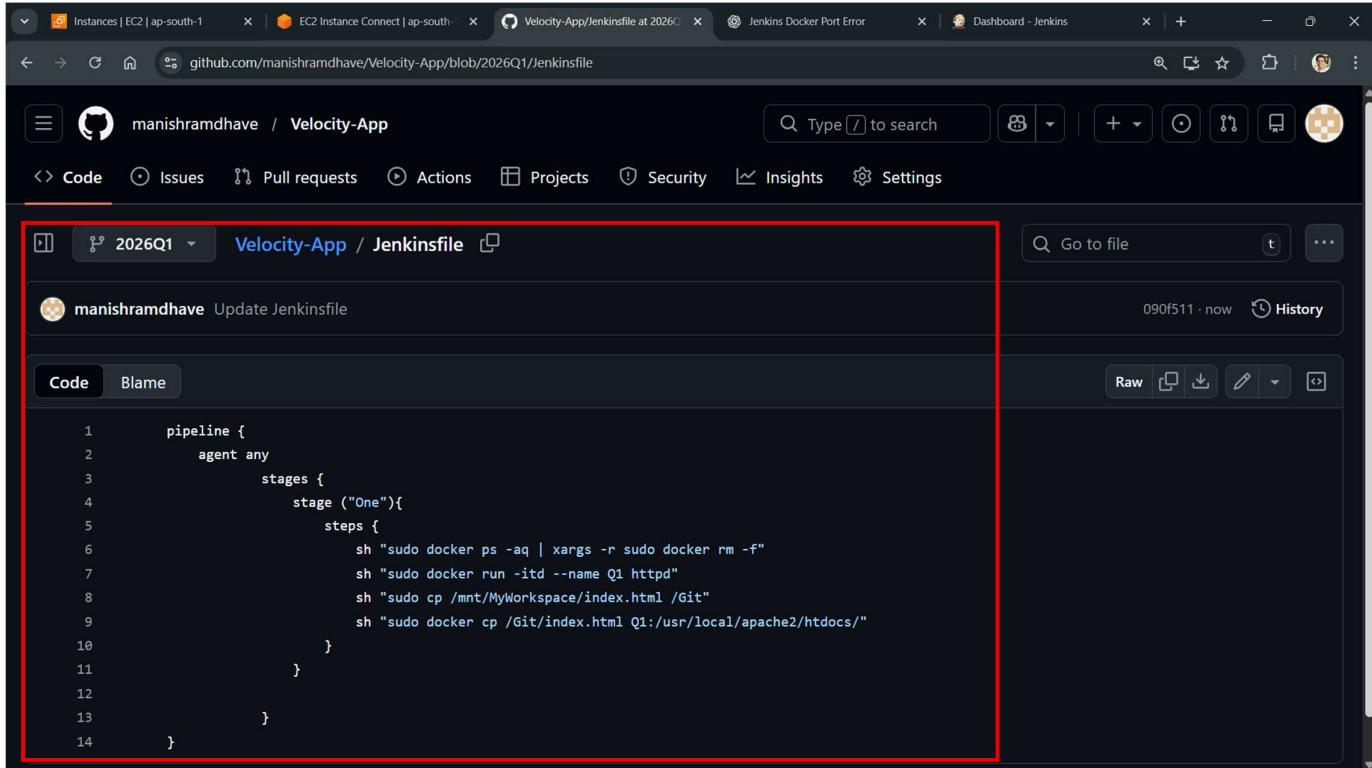
The screenshot shows the Jenkins Pipeline configuration page for a job named 'Doc-Assign8-Pipeline-C'. The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area is titled 'Pipeline' and contains the sub-section 'Definition'. Under 'Definition', 'Pipeline script from SCM' is selected. Below it, 'SCM' is set to 'Git'. A 'Repositories' section shows a single repository with 'Repository URL' as 'https://github.com/manishramdhave/Velocity_App.git' and 'Credentials' as 'Manish/******** (Git)'. A red box highlights this entire configuration section.

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 Pipeline configuration page for a job named 'Doc-Assign2-Pipeline-A'. The left sidebar has tabs for General, Triggers, Pipeline (selected), and Advanced. The main area has a 'Repository URL' of 'https://github.com/manishramdhave/Velocity_App.git' and a 'Credentials' dropdown set to 'ec2-user/******** (Git_token)'. Below these, an 'Advanced' button is shown. A red box highlights the 'Branches to build' section, which contains two 'Branch Specifier' fields: one with '/master' and another with '*/2026Q1'. At the bottom are 'Save' and 'Apply' buttons.

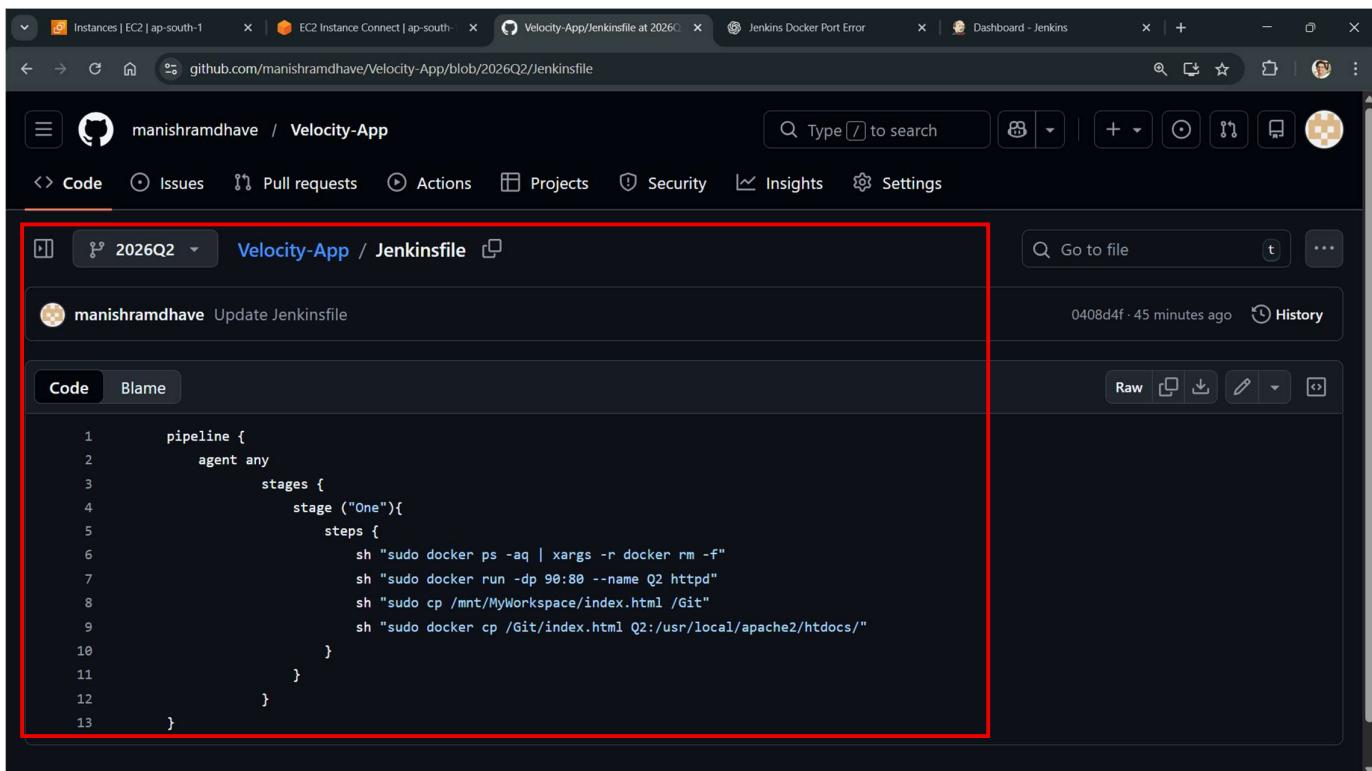
Step 11: Created a ‘Jenkinsfile’ on 2026Q1 branch in which the script is mentioned:



The screenshot shows a GitHub repository page for 'Velocity-App'. The 'Code' tab is selected, and the file 'Velocity-App/Jenkinsfile' is displayed. A red box highlights the Jenkinsfile content. The code defines a pipeline with one stage named 'One' containing steps to stop existing Docker containers, run a new container named Q1 with port 80 mapped to 80, copy index.html from the workspace to the container's /Git directory, and copy it to the Apache2 htdocs directory.

```
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 -tid --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    }
```

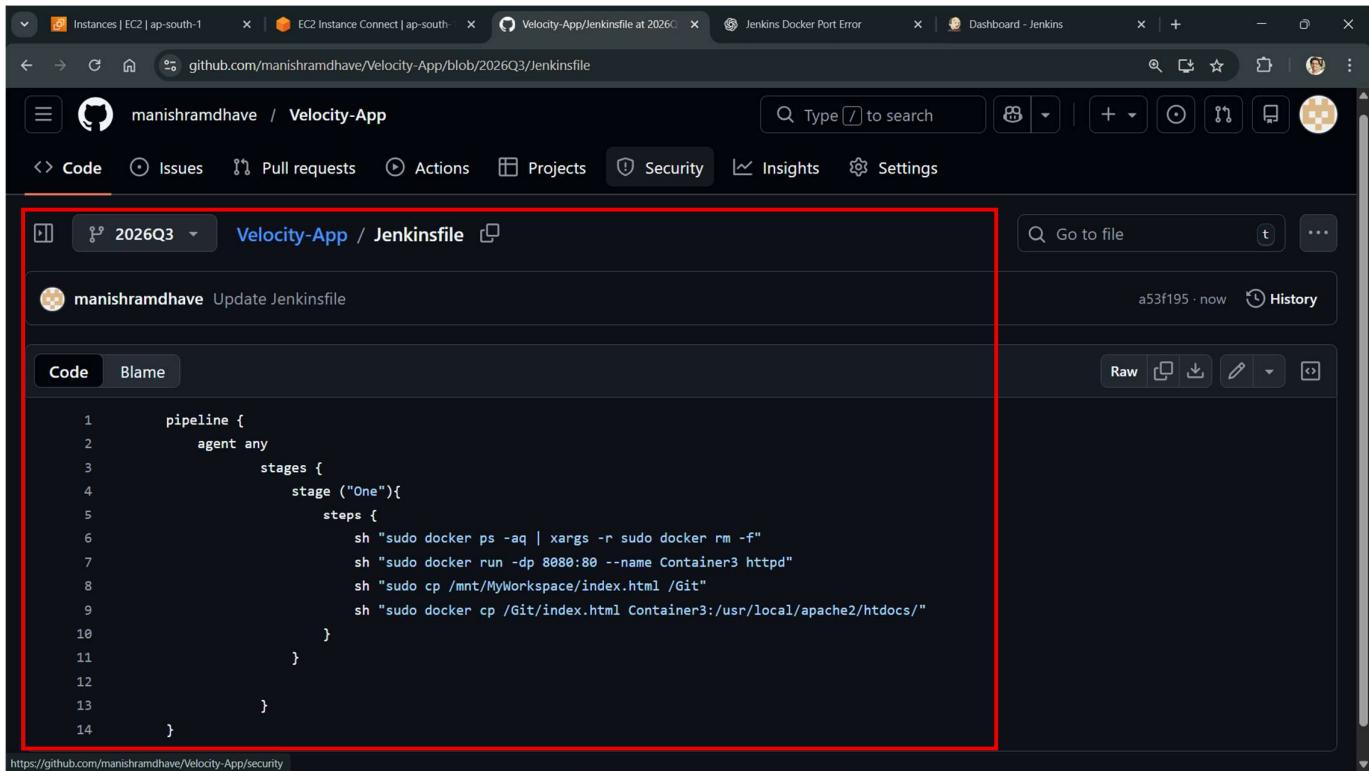
Step 12: Created a ‘Jenkinsfile’ on 2026Q2 branch in which the script is mentioned:



The screenshot shows a GitHub repository page for 'Velocity-App'. The 'Code' tab is selected, and the file 'Velocity-App/Jenkinsfile' is displayed. A red box highlights the Jenkinsfile content. The code is identical to the one in Step 11, defining a pipeline with a single stage named 'One' that performs the same set of Docker commands to update the application.

```
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        }
14    }
```

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



The screenshot shows a GitHub repository interface for the 'Velocity-App' project. A red box highlights the code editor area where a Jenkinsfile is displayed. 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 -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 }
```

The Jenkinsfile defines a pipeline with one stage named 'One'. Within this stage, several shell commands are executed to manage Docker containers and copy files.

Results:

- 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 top-left window shows a GitHub commit history for the 'Velocity-App' repository in the '2026Q1' branch. A red box highlights the commit message: "Hosting updated index.html from Container 1 of 2026Q1 Branch". The top-right window shows a browser tab at '13.233.128.97/index.html' with the message "Hosting updated index.html from Container 1 of 2026Q1 Branch change 1 Change 2". The bottom window shows a Jenkins job log for 'Doc-Assign1-Pipeline-A #7'. A red box highlights the build status: "#7 (10 Feb 2026, 04:37:04)". The log details the build was started by GitHub push and took 6.8 seconds.

- 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 top-left window shows a GitHub commit history for the 'Velocity-App' repository in the '2026Q2' branch. A red box highlights the commit message: "Hosting updated index.html from Container 2 of 2026Q2 Branch". The top-right window shows a browser tab at '13.233.128.97:90/index.html' with the message "Hosting updated index.html from Container 2 of 2026Q2 Branch change 1 Manish here Done with Docker Assignment 1". The bottom window shows a Jenkins job log for 'Doc-Assign1-Pipeline-B #8'. A red box highlights the build status: "#8 (10 Feb 2026, 04:39:19)". The log details the build was started by GitHub push and 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 contains four screenshots illustrating the Jenkins pipeline process:

- Top-left Screenshot:** A GitHub pull request page for the Velocity-App repository. It shows a commit from manishramdhave titled "Update index.html" that has been merged into the 2026Q3 branch. The commit message includes the note "Hosting updated index.html from Container3 of 2026Q3 Branch".
- Top-right Screenshot:** A Jenkins console window for build #8 of the Doc-Assign1-Pipeline-C job. The log output shows a Docker error message:

```
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.
```
- Bottom-left Screenshot:** The Jenkins job details for Doc-Assign1-Pipeline-C #8. It shows the build was triggered by a GitHub push from manishramdhave and completed successfully, taking 4.9 seconds. The status bar indicates "Started 9.3 sec ago".
- Bottom-right Screenshot:** A browser window showing a 404 Not Found error for the URL 13.233.128.97:8080/index.html. The error page is titled "HTTP Status 404 – Not Found" and provides a status report and Apache Tomcat version information.