

Docker Assignment 4

Step 1: Launched three instances, D4-Jenkins Master, D4-Slave1 and D4-Slave2

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like Dashboard, AWS Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images, AMIs, and AMI Catalog. The main area displays a table titled 'Instances (3/8) Info' with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. Three instances are listed: D4 Linux Jenkins Master (i-01c51ecc3e6b87030), D4 Linux Slave1 (i-030ec11d2b935ddd0), and D4 Linux Slave2 (i-0d5a379180ebc1940). All three instances are shown as 'Running'. Below the table, it says '3 instances selected'. Under the 'Monitoring' section, there are four line charts: CPU utilization (%), Network in (bytes), Network out (bytes), and Network packets in (count), all showing data from 18:00 to 18:30.

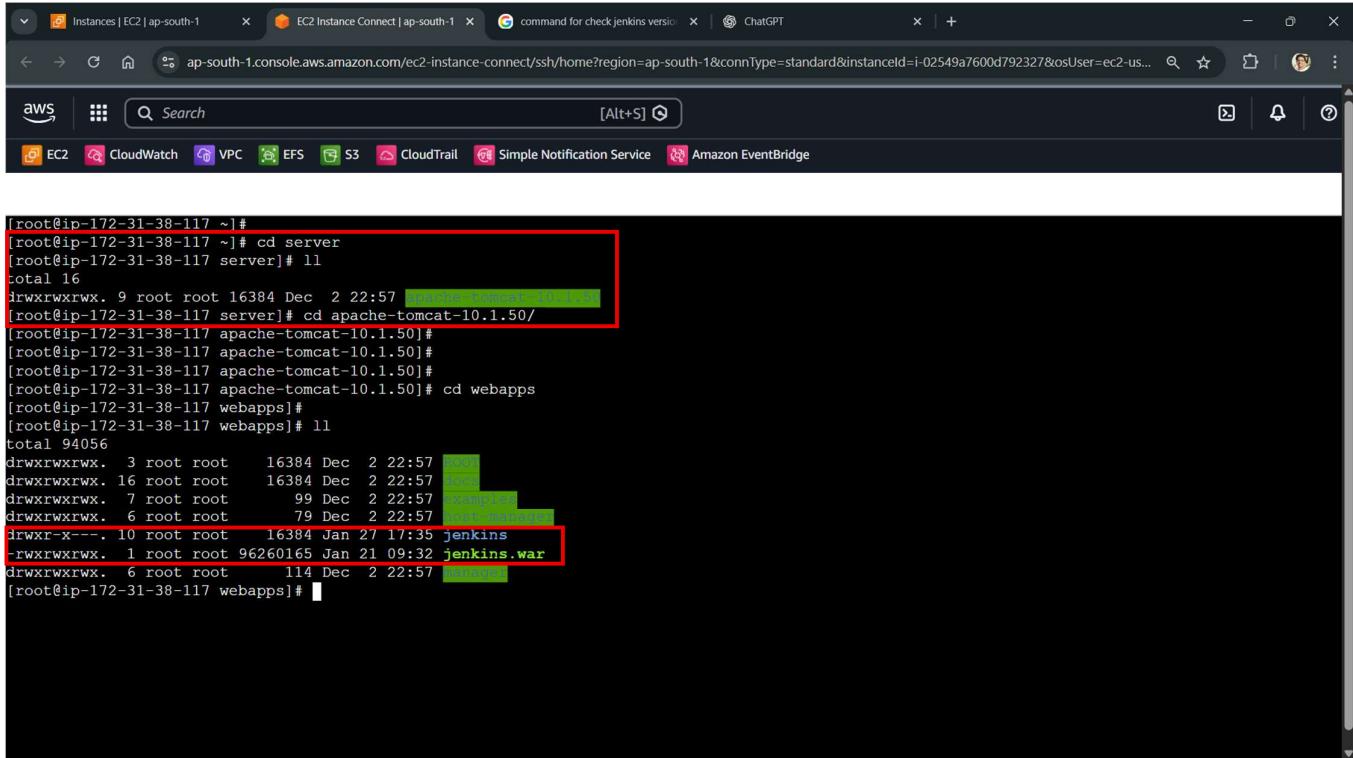
Step 2: Installed Java-17, Git and Docker on all the instance:

The screenshot shows the AWS EC2 Instance Connect terminal for the Jenkins Master instance (i-0bb0216d314c17b07). The terminal window has a red border. The command history shows the following installations:

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

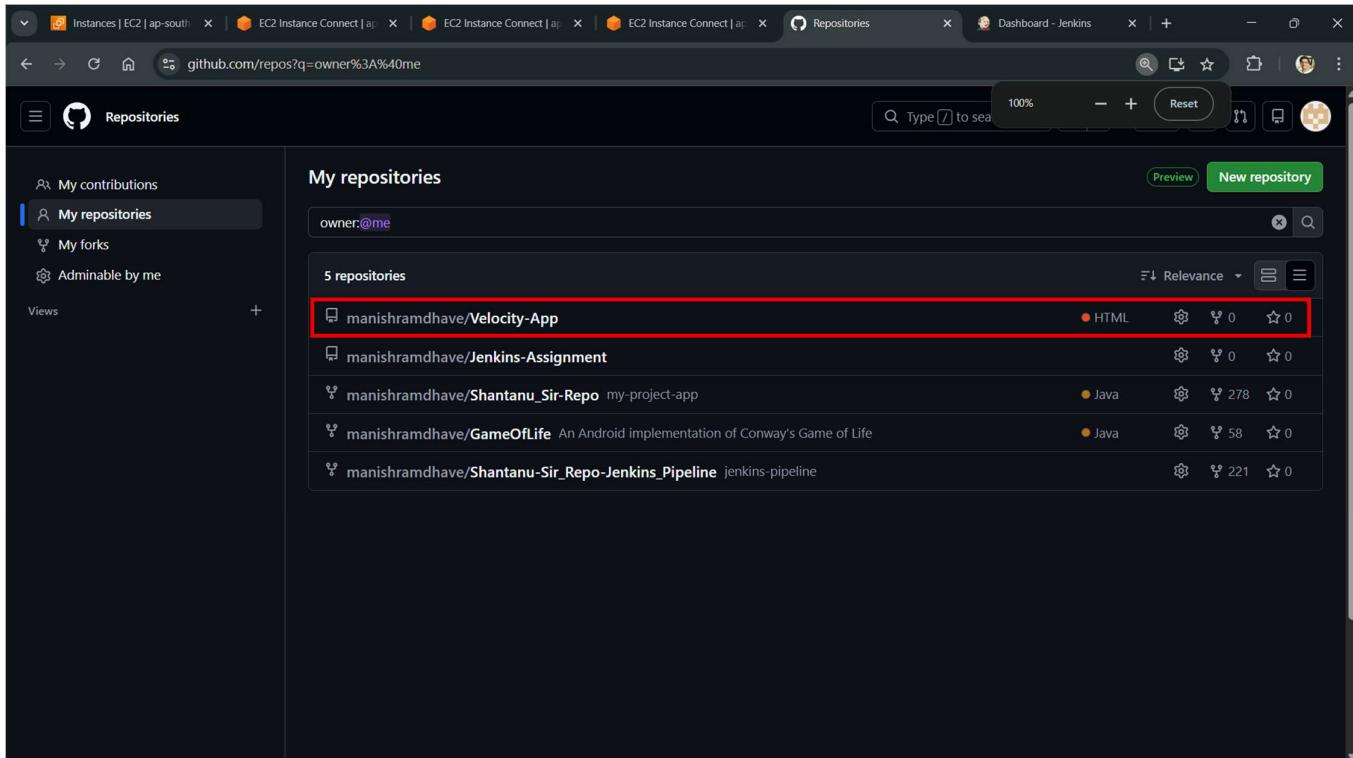
Below the terminal, a message box displays the instance details: i-0bb0216d314c17b07 (Linux Jenkins Master) with PublicIPs: 13.233.132.0 and PrivateIPs: 172.31.40.216.

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]# total 16
drwxrwxrwx. 9 root root 16384 Dec  2 22:57 .
drwxrwxrwx. 1 root root 16384 Dec  2 22:57 ..
[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]# [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 .
drwxrwxrwx. 16 root root 16384 Dec  2 22:57 docs
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 Admirable by me. The main area is titled "My repositories" and shows a search bar with "owner:@me". Below it, there is a table of repositories:

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

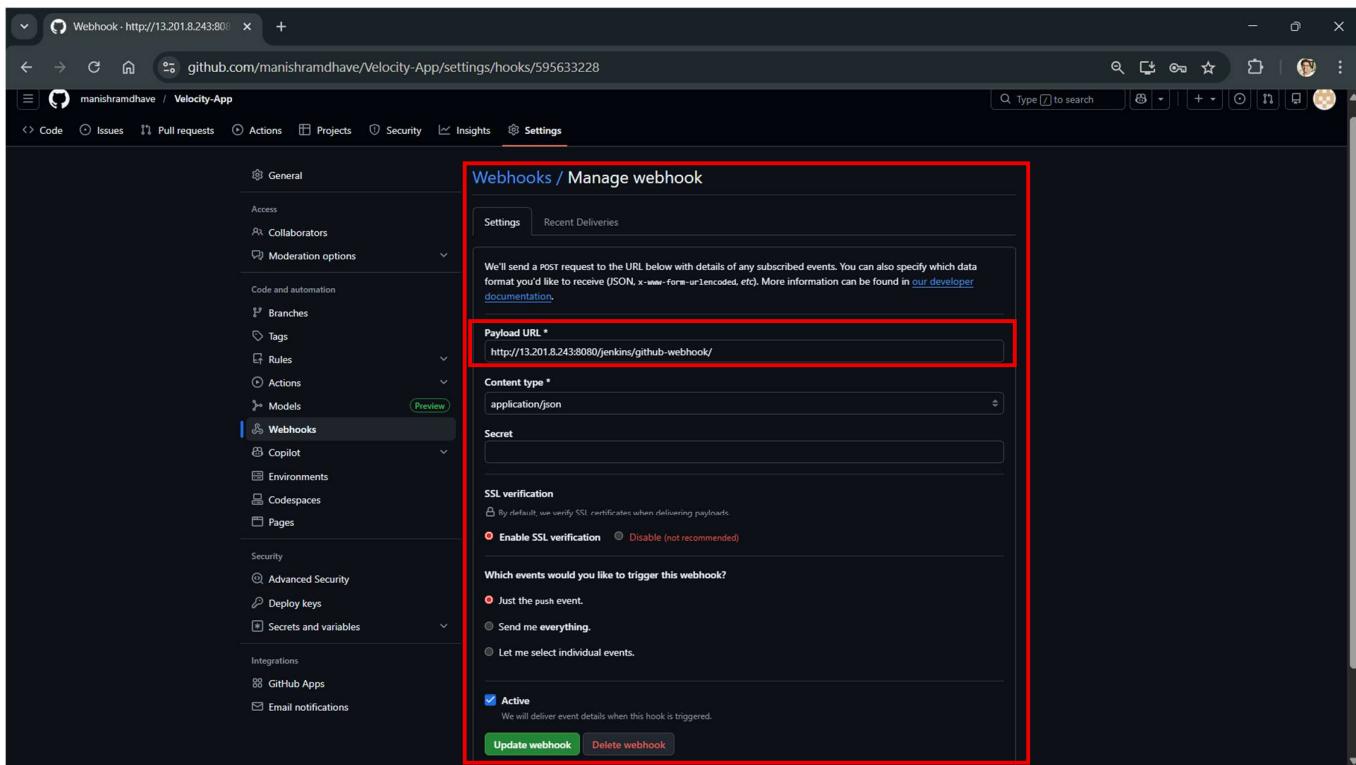
Step 5: Created two branches, **2026Q1** and **2026Q2** in the ‘Velocity-App’ Repository and pushed **two different ‘index.html’ files** and also created **two different ‘Jenkinsfile’** files in respective branches:

The screenshot shows two GitHub repository pages. The top page is for the branch **2026Q1**, which has 3 branches and 0 tags. It contains 56 commits, with the most recent being "Update Jenkinsfile" 36 minutes ago and "Update index.html" 33 minutes ago. The bottom page is for the branch **2026Q2**, which has 3 branches and 0 tags. It contains 15 commits, with the most recent being "Update Jenkinsfile" 54 minutes ago and "Update index.html" 33 minutes ago. Both pages show a red box around the Jenkinsfile and index.html entries in the commit list.

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:

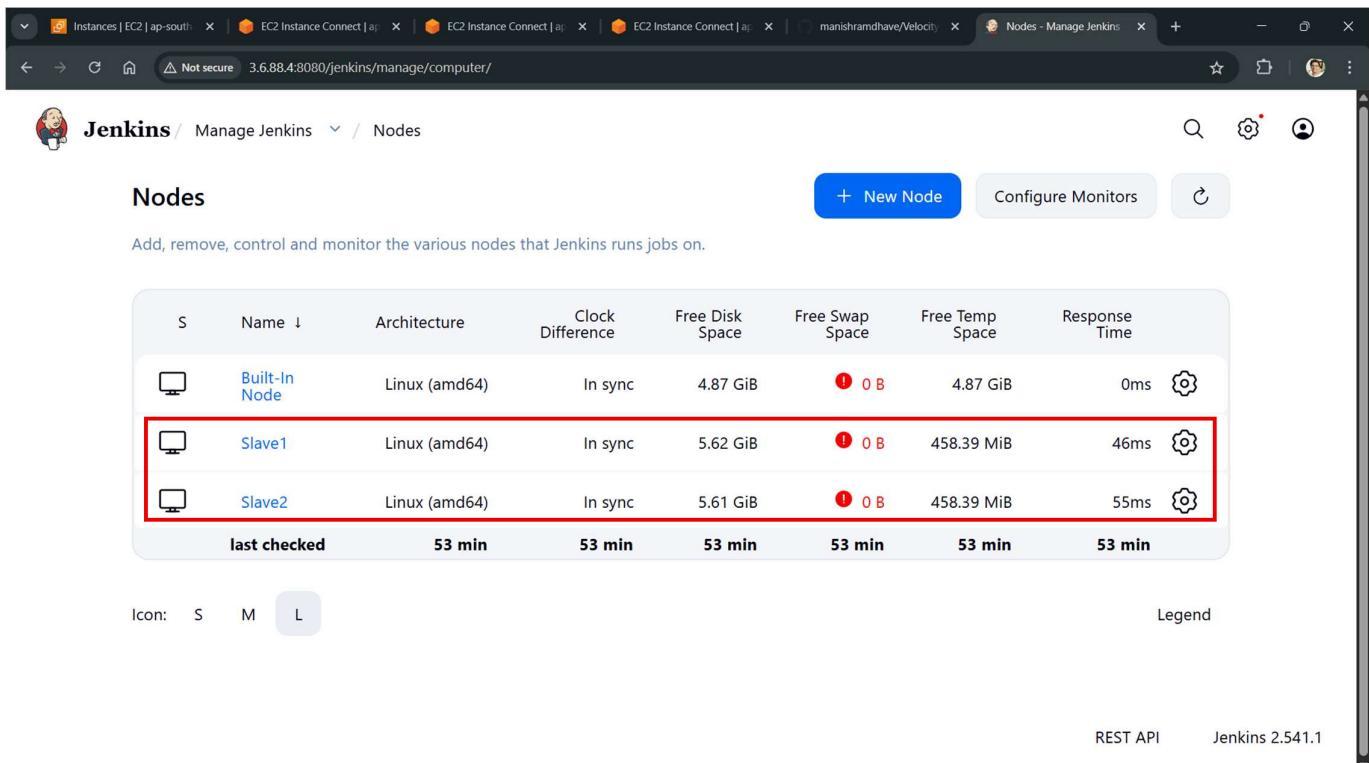
The screenshot shows the Jenkins 'Manage Jenkins' interface with the 'System' configuration selected. In the 'GitHub' section, there is a 'GitHub Servers' configuration. A red box highlights the 'Credentials' dropdown menu, which currently displays 'Git'. Below this, there is a checkbox for 'Manage hooks' and an 'Advanced' dropdown. At the bottom of the screen, there are 'Save' and 'Apply' buttons.

Step 7: Created an API Connection between Jenkins and GitHub Repositories by using ‘Git Webhooks’ (i.e. setting up Jenkins URL in Git) :



The screenshot shows the GitHub 'Webhooks / Manage webhook' settings page. A red box highlights the 'Payload URL' field, which contains the Jenkins URL: `http://13.201.8.243:8080/jenkins/github-webhook/`. The 'Content type' is set to `application/json`. The 'SSL verification' section has 'Enable SSL verification' selected. Under 'Which events would you like to trigger this webhook?', 'Just the push event.' is selected. The 'Active' checkbox is checked. At the bottom are 'Update webhook' and 'Delete webhook' buttons.

Step 8: Created two Nodes for to establish the connection between Jenkins Master and each Slave Instances using the ‘Credential’ and ‘Manually trusted key Verification Strategy’:



The screenshot shows the Jenkins 'Nodes - Manage Jenkins' page under the 'Nodes' tab. A red box highlights the 'Slave1' and 'Slave2' entries in the table. The table columns are: S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. The 'Slave1' and 'Slave2' rows show details: both are Linux (amd64), both are 'In sync', 'last checked' was 53 min ago, and their response times are 46ms and 55ms respectively. The 'Slave1' row also has a red border around it.

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	Built-In Node	Linux (amd64)	In sync	4.87 GiB	0 B	4.87 GiB	0ms
2	Slave1	Linux (amd64)	In sync	5.62 GiB	0 B	458.39 MiB	46ms
3	Slave2	Linux (amd64)	In sync	5.61 GiB	0 B	458.39 MiB	55ms

Step 9: Launched the Jenkins and created two different Pipeline Jobs. **Job1** and **Job2**:

The screenshot shows the Jenkins dashboard with a red box highlighting the list of pipeline jobs. The table has columns: S (Status), W (Work), Name, Last Success, Last Failure, and Last Duration.

S	W	Name	Last Success	Last Failure	Last Duration
Green checkmark	Cloud icon	Doc-Assign4-Pipeline-A	13 min #5	22 min #2	5.4 sec
Green checkmark	Cloud icon	Doc-Assign4-Pipeline-B	12 min #5	23 min #2	5.6 sec

Build Queue: No builds in the queue.

Build Executor Status:

- Built-In Node: 0/2
- Slave1: 0/10
- Slave2: 0/10

REST API Jenkins 2.541.1

Step 10: Integrated Git branches with Jenkins by creating ‘Credentials’ by using the Git Token on both the pipeline jobs:

The screenshot shows the Jenkins Pipeline configuration page for the job 'Doc-Assign2-Pipeline-A'. A red box highlights the 'Pipeline' section where Git repository details and branch specifiers are defined.

Repository URL: https://github.com/manishramdhave/Velocity-App.git

Credentials: ec2-user/******** (Git_token)

Branches to build:

- Branch Specifier (blank for 'any'): */master
- Branch Specifier (blank for 'any'): */2026Q1

Save Apply

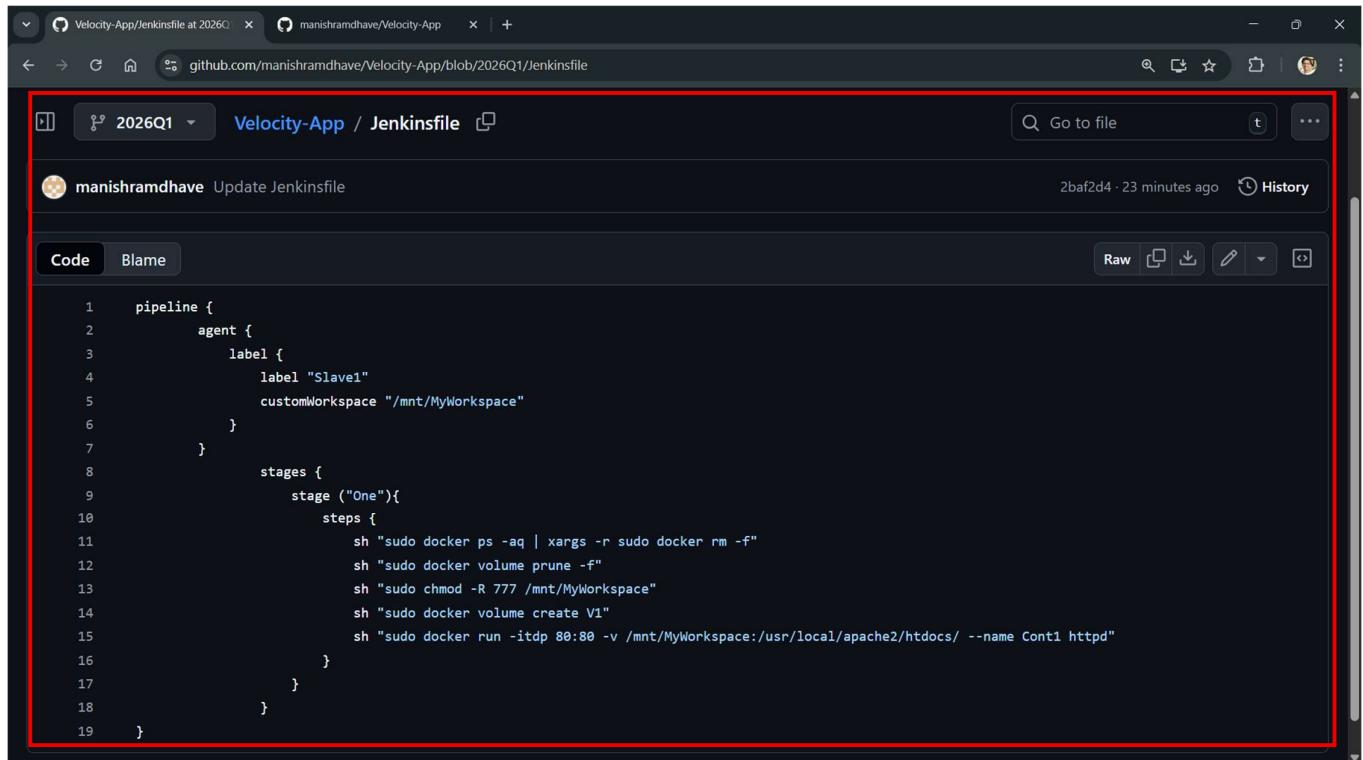
Step 11: Used ‘Pipeline script from SCM’ and selected the SCM as a ‘Git’ which will follow the Jenkins pipeline script by using the ‘Jenkinsfile’ in the repository:

The screenshot shows the Jenkins Pipeline configuration page for a job named 'Doc-Assign2-Pipeline-A'. The 'Pipeline' tab is selected in the sidebar. A red box highlights the 'Repository URL' field containing 'https://github.com/manishramdhav/Velocity-App.git' and the 'Credentials' dropdown set to 'ec2-user/******/(Git_token)'. Below these, there are two 'Branch Specifier' fields: one for 'master' and another for '*/2026Q1'. At the bottom are 'Save' and 'Apply' buttons.

Step 12: Integrated Git branches, 2026Q1 and 2026Q2 with Jenkins by creating ‘Credentials’ by using the Git Token on both the pipeline Job-A and Job-B respectively:

This screenshot is identical to the one above, showing the Jenkins Pipeline configuration page for 'Doc-Assign2-Pipeline-A'. The 'Pipeline' tab is selected. A red box highlights the 'Repository URL' field with 'https://github.com/manishramdhav/Velocity-App.git' and the 'Credentials' dropdown with 'ec2-user/******/(Git_token)'. It also highlights the 'Branch Specifier' fields for 'master' and '*/2026Q1'. The 'Save' and 'Apply' buttons are at the bottom.

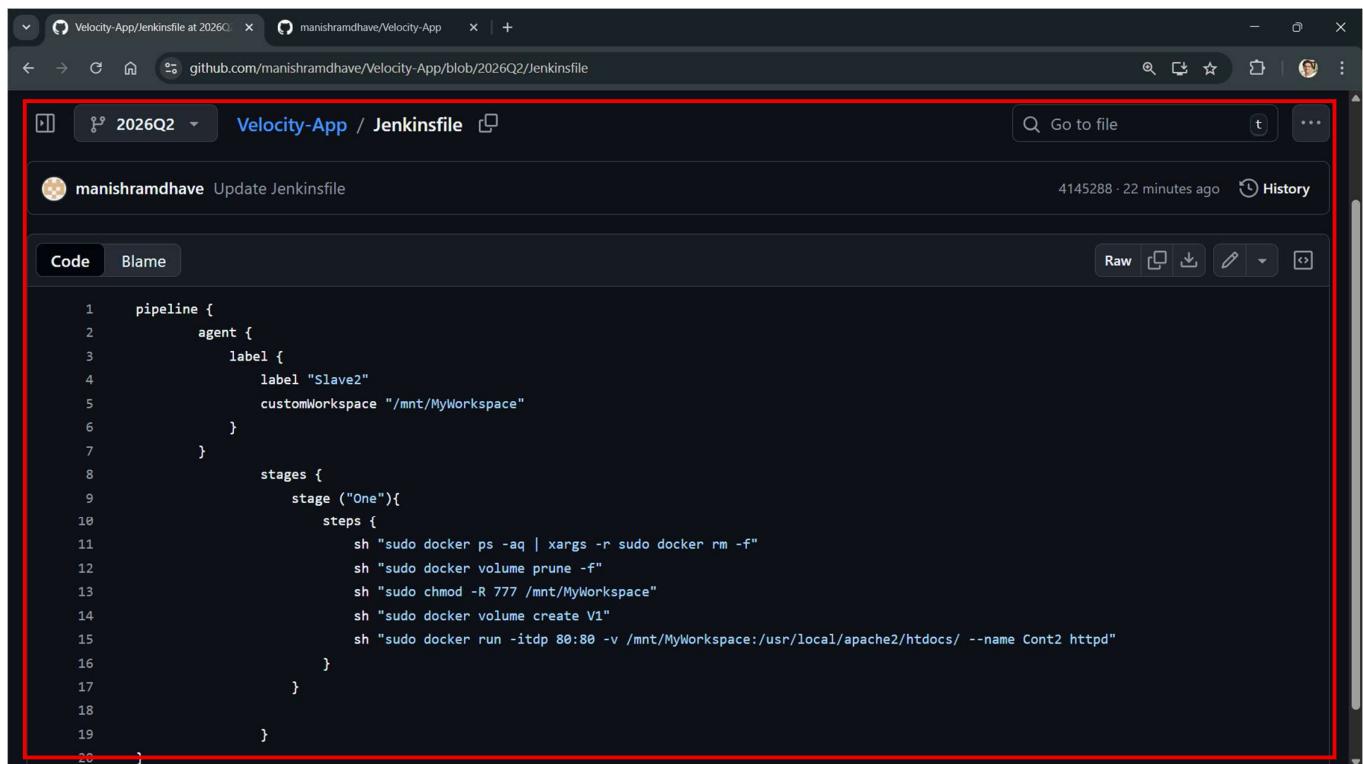
Step 13: Created a ‘Jenkinsfile’ on 2026Q1 branch in which out script is mentioned:



The screenshot shows a GitHub repository page for 'Velocity-App'. A red box highlights the Jenkinsfile content. The file contains a pipeline definition with an agent labeled 'Slave1' and a stage named 'One' containing several shell commands to clean up Docker containers and volumes, and to run an Apache container.

```
1 pipeline {
2     agent {
3         label {
4             label "Slave1"
5             customWorkspace "/mnt/MyWorkspace"
6         }
7     }
8     stages {
9         stage ("One"){
10            steps {
11                sh "sudo docker ps -aq | xargs -r sudo docker rm -f"
12                sh "sudo docker volume prune -f"
13                sh "sudo chmod -R 777 /mnt/MyWorkspace"
14                sh "sudo docker volume create V1"
15                sh "sudo docker run -itdp 80:80 -v /mnt/MyWorkspace:/usr/local/apache2/htdocs/ --name Cont1 httpd"
16            }
17        }
18    }
19 }
```

Step 14: Created a ‘Jenkinsfile’ on 2026Q2 branch in which out script is mentioned:



The screenshot shows a GitHub repository page for 'Velocity-App'. A red box highlights the Jenkinsfile content. The file defines a pipeline with an agent labeled 'Slave2' and a stage named 'One' that performs similar cleanup and containerization steps as the previous Jenkinsfile.

```
1 pipeline {
2     agent {
3         label {
4             label "Slave2"
5             customWorkspace "/mnt/MyWorkspace"
6         }
7     }
8     stages {
9         stage ("One"){
10            steps {
11                sh "sudo docker ps -aq | xargs -r sudo docker rm -f"
12                sh "sudo docker volume prune -f"
13                sh "sudo chmod -R 777 /mnt/MyWorkspace"
14                sh "sudo docker volume create V1"
15                sh "sudo docker run -itdp 80:80 -v /mnt/MyWorkspace:/usr/local/apache2/htdocs/ --name Cont2 httpd"
16            }
17        }
18    }
19 }
```

Result:

- When changes are done in 2026Q1 branch, Build is triggered by ‘Doc-Assign4-Pipeline-A’ and the updated index.html file is hosted from the container ‘Cont1’ of a Slave1 instance by using 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 Manish here Change 12:10AM". The top-right window shows a web browser displaying the contents of 'index.html' from a Jenkins slave instance (IP 13.126.123.119). A red box highlights the page content: "Hosting updated index.html from Container 1 of 2026Q1 Branch Manish here Change 12:10AM". The bottom window shows the Jenkins job 'Doc-Assign4-Pipeline-A #5'. A red box highlights the build status: "#5 (10 Feb 2026, 18:38:48)". Below it, the log shows: "Started by GitHub push by manishramdhav... Started 8.2 sec ago Took 5.4 sec".

- When changes are done in 2026Q2 branch, Build is triggered by ‘Doc-Assign4-Pipeline-B’ and the updated index.html file is hosted from the container ‘Cont2’ of a Slave2 instance by using 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 change 1 Done with Docker Assignment 4". The top-right window shows a web browser displaying the contents of 'index.html' from a Jenkins slave instance (IP 43.205.238.17). A red box highlights the page content: "Hosting updated index.html from Container 2 of 2026Q2 Branch change 1 Done with Docker Assignment 4". The bottom window shows the Jenkins job 'Doc-Assign4-Pipeline-B #5'. A red box highlights the build status: "#5 (10 Feb 2026, 18:39:43)". Below it, the log shows: "Started by GitHub push by manishramdhav... Started 54 sec ago Took 5.6 sec".