

# Docker Assignment 2

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

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like Dashboard, AWS Global View, Events, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images, AMIs, AMI Catalog, and Elastic Block Store Volumes, Snapshots, Lifecycle Manager. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
Jenkins Master (Docker)	i-089629caf8d7bfc2e	Running	t3.micro	Initializing	<a href="#">View alarms +</a>	ap-south-1a	ec2-3-6-88-42
Linux Slave 1	i-05ead13f1fe870d94	Running	t3.micro	3/3 checks passed	<a href="#">View alarms +</a>	ap-south-1a	ec2-43-205-21-102
Linux Slave 2	i-0b533c87befd61e57	Running	t3.micro	3/3 checks passed	<a href="#">View alarms +</a>	ap-south-1a	ec2-43-205-21-102

Below the table, it says "3 instances selected". Under the "Monitoring" tab, there are four line charts: CPU utilization (%), Network in (bytes), Network out (bytes), and Network packets in (count). The CPU utilization chart shows a peak around 51.6% at approximately 10 minutes ago. The Network in and out charts show Bytes over time, with Network in peaking at 12.52M and Network out at 3.144M. The Network packets in chart shows Count peaking at 8.72K.

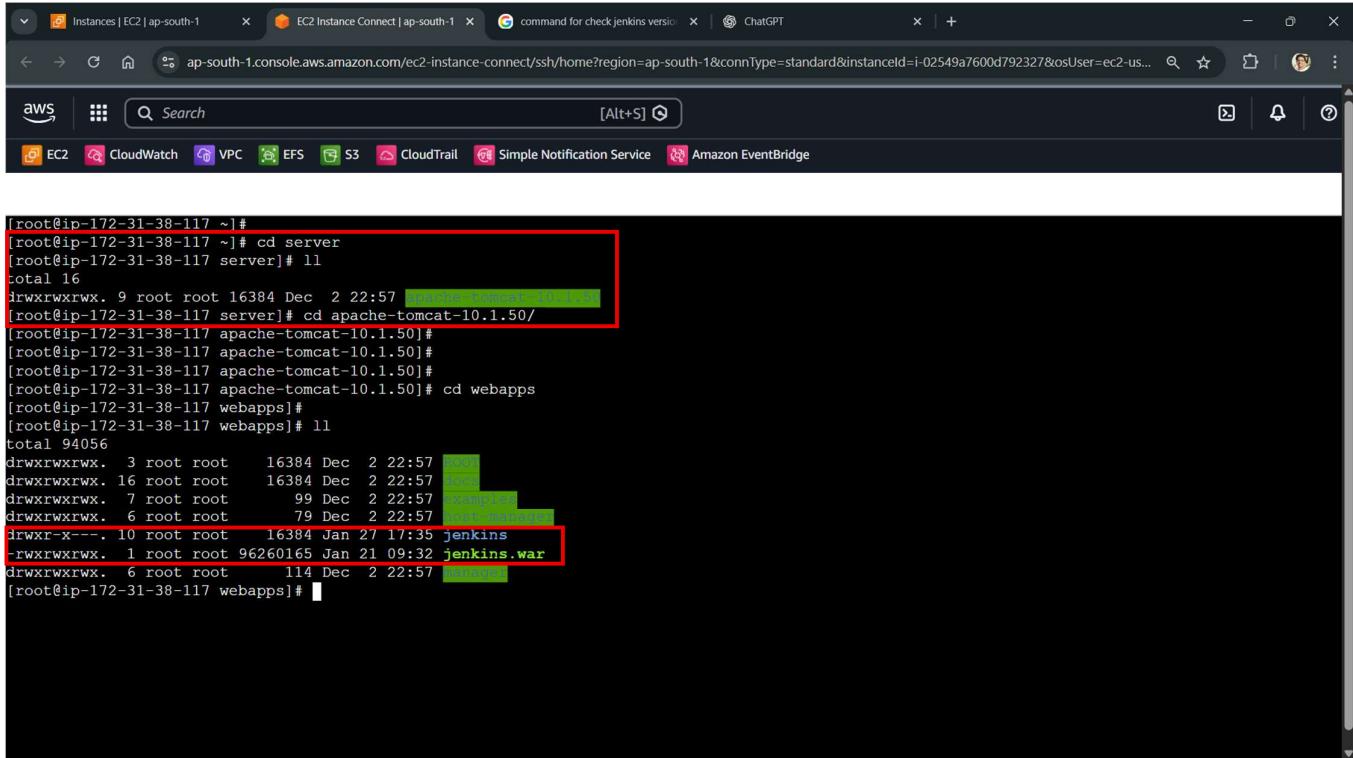
Step 2: Installed Java-17, Git and Docker on all the instance:

The screenshot shows a CloudShell session on an EC2 instance. 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 ~]# ]
```

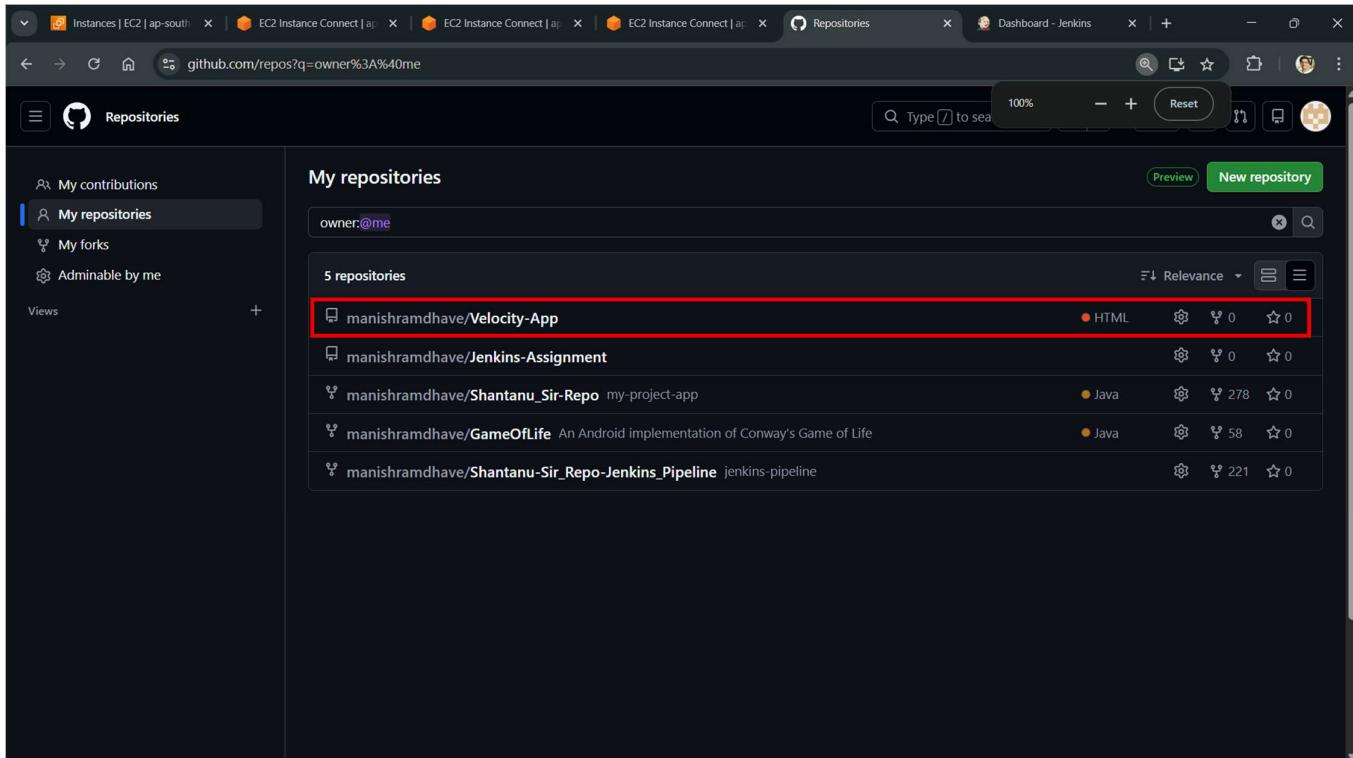
At the bottom, it shows the instance details: i-0bb0216d314c17b07 (Linux Jenkins Master) with PublicIPs: 13.233.132.0 and PrivateIPs: 172.31.40.216.

### 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
<a href="#">manishramdhave/Velocity-App</a>	HTML	0	0
<a href="#">manishramdhave/Jenkins-Assignment</a>		0	0
<a href="#">manishramdhave/Shantanu_Sir-Repo</a>	my-project-app	278	0
<a href="#">manishramdhave/GameOfLife</a>	An Android implementation of Conway's Game of Life	58	0
<a href="#">manishramdhave/Shantanu-Sir_Repo-Jenkins_Pipeline</a>	jenkins-pipeline	221	0

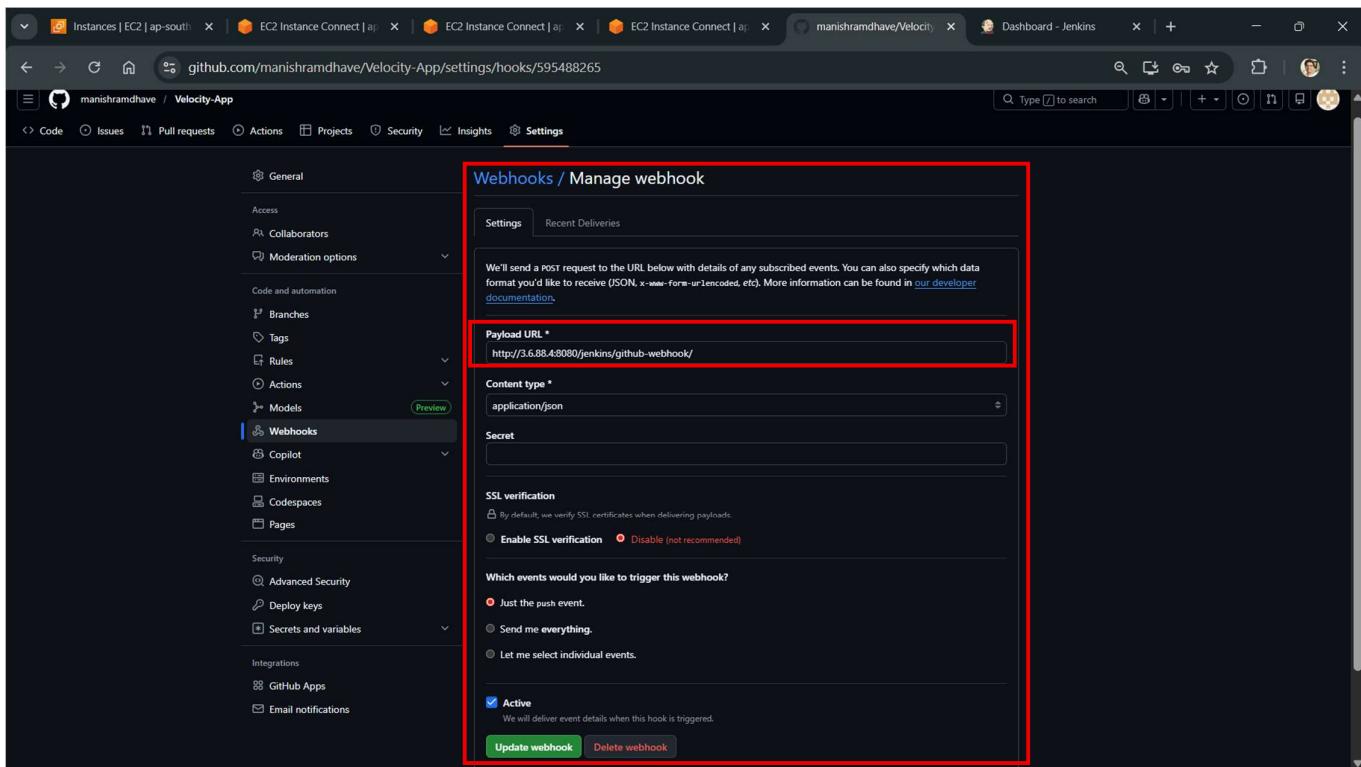
Step 5: Created two branches, **2026Q1** and **2026Q2** in the ‘Velocity-App’ Repository and pushed **two different ‘index.html’ files** and also created **two different ‘Jenkinsfile’** files in respective branches:

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:

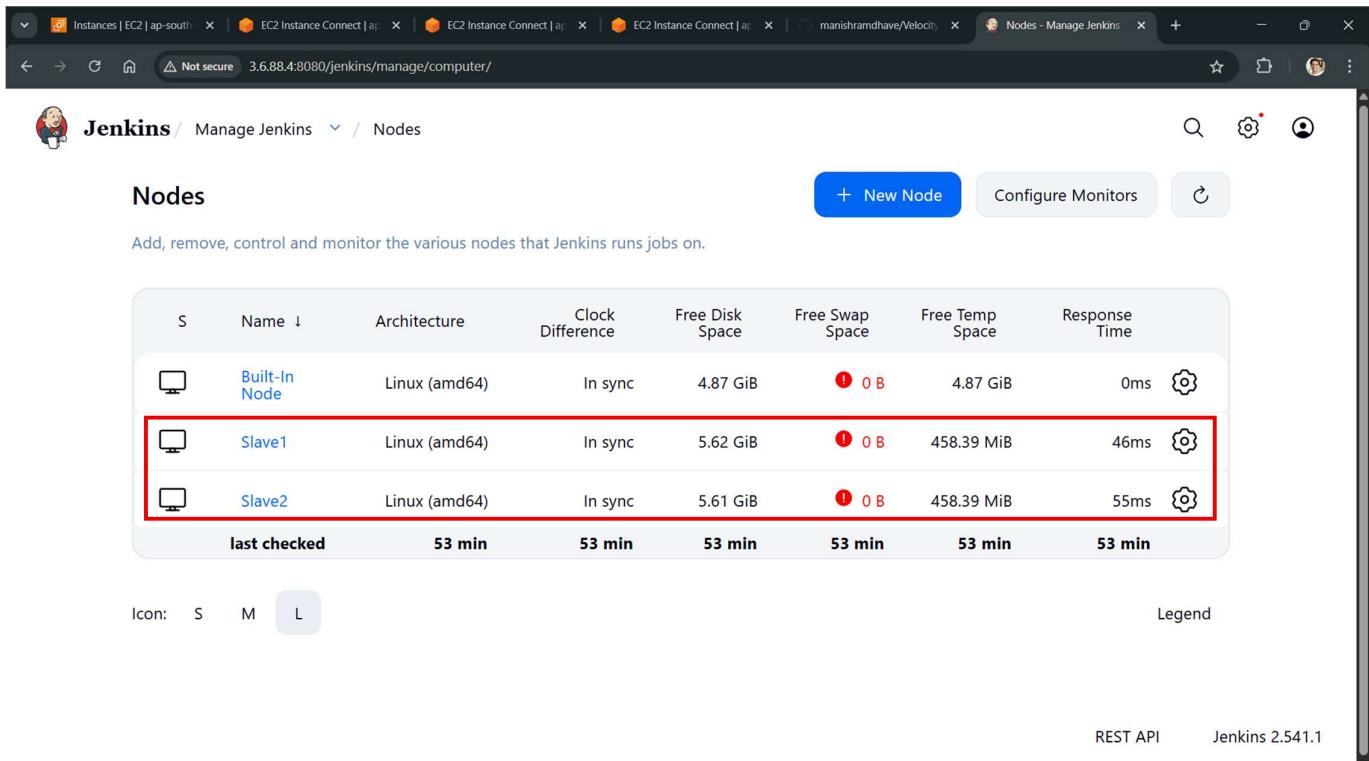
The screenshot shows the Jenkins 'Manage Jenkins' interface under the 'System' section. It displays the GitHub configuration. A red box highlights the 'Credentials' dropdown menu, which currently shows 'Git'. Other fields visible include 'Name' (GitHub-Server) and 'API URL' (https://api.github.com). There is also a 'Test connection' button and a 'Manage hooks' checkbox. At the bottom, 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 main configuration area. The 'Payload URL \*' field contains `http://3.6.88.4:8080/jenkins/github-webhook/`. The 'Content type \*' dropdown 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' management page. A red box highlights the list of nodes. The nodes are: 'Built-In Node' (Architecture: Linux (amd64), Clock Difference: In sync, Free Disk Space: 4.87 GiB, Free Swap Space: 0 B, Free Temp Space: 4.87 GiB, Response Time: 0ms), 'Slave1' (Architecture: Linux (amd64), Clock Difference: In sync, Free Disk Space: 5.62 GiB, Free Swap Space: 0 B, Free Temp Space: 458.39 MiB, Response Time: 46ms), and 'Slave2' (Architecture: Linux (amd64), Clock Difference: In sync, Free Disk Space: 5.61 GiB, Free Swap Space: 0 B, Free Temp Space: 458.39 MiB, Response Time: 55ms). Below the table, there's a legend for icons: S, M, L. The bottom right corner shows 'REST API' and 'Jenkins 2.541.1'.

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 pipeline jobs section. The table displays the following information:

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	rain	Doc-Assign2-Pipeline-A	2 min 54 sec #22	6 min 4 sec #20	7.1 sec
✓	cloud	Doc-Assign2-Pipeline-B	59 sec #6	6 min 4 sec #5	6.7 sec

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 job configuration page for 'Doc-Assign8-Pipeline-C'. The 'Pipeline' section is highlighted with a red box. The configuration details are as follows:

- Definition:** Pipeline script from SCM
- SCM:** Git
- Repositories:**
  - Repository URL:** https://github.com/manishramdhave/Velocity\_App.git
  - Credentials:** Manish/\*\*\*\*\*\*\*\* (Git)

**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:**

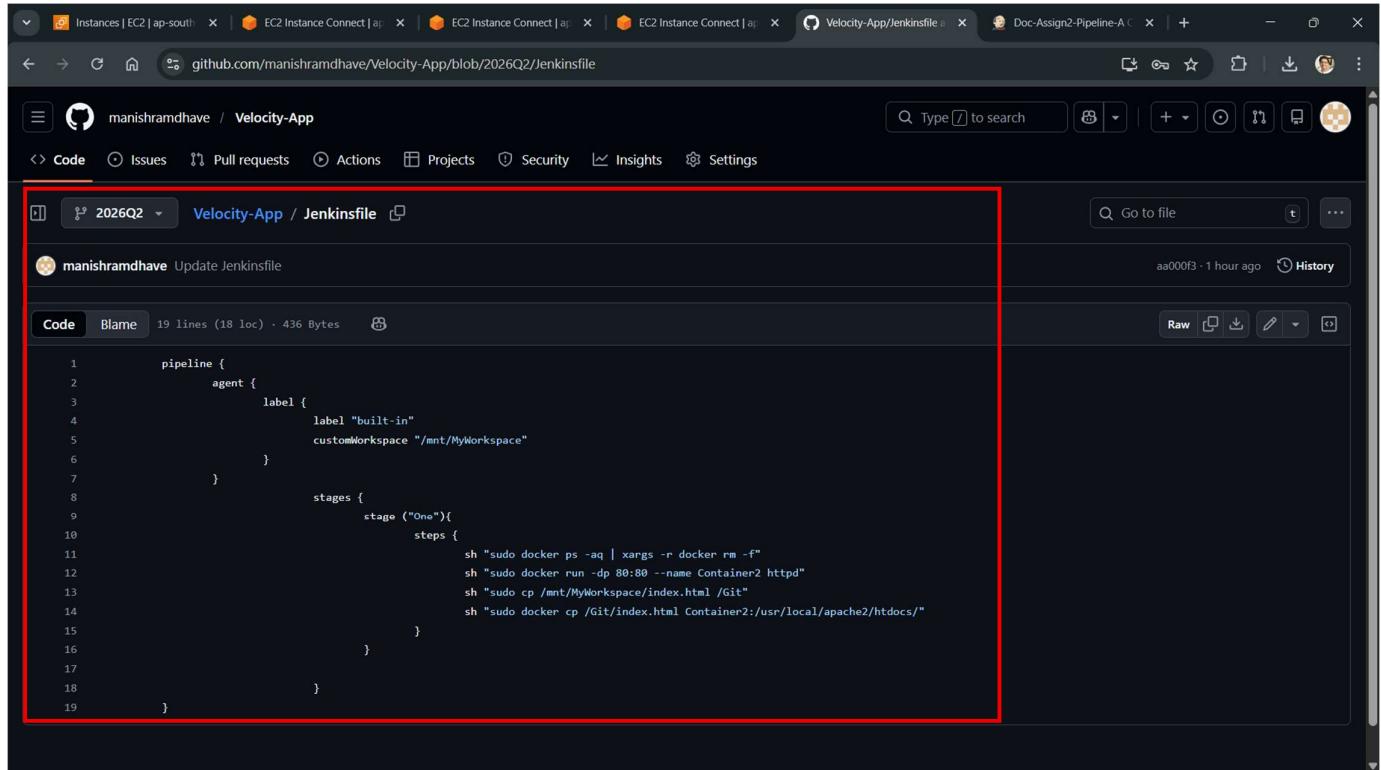
The screenshot shows the Jenkins Pipeline configuration page for the 'Doc-Assign2-Pipeline-A' job. A red box highlights the central configuration area. It includes the 'Repository URL' field with 'https://github.com/manishramdhave/Velocity-App.git', the 'Credentials' dropdown set to 'ec2-user/\*\*\*\*\*\*/(Git\_token)', and the 'Branches to build' section. The 'Branch Specifier (blank for 'any')' dropdown contains two entries: '\*/master' and '\*/2026Q1'. At the bottom are 'Save' and 'Apply' buttons.

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

The screenshot shows the GitHub repository 'manishramdhave/Velocity-App'. A red box highlights the code editor for the 'Jenkinsfile' on the '2026Q1' branch. The file content is as follows:

```
1     pipeline {
2         agent {
3             label {
4                 label "built-in"
5                 customWorkspace "/mnt/MyWorkspace"
6             }
7         }
8         stages {
9             stage ("One"){
10                steps {
11                    sh "docker ps -aq | xargs -r docker rm -f"
12                    sh "sudo docker run -dp 80:80 --name Container1 httpd"
13                    sh "sudo cp /mnt/MyWorkspace/index.html /Git"
14                    sh "sudo docker cp /Git/index.html Container1:/usr/local/apache2/htdocs/"
15                }
16            }
17        }
18    }
```

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



The screenshot shows a GitHub repository page for 'Velocity-App'. The 'Code' tab is selected, displaying the Jenkinsfile. A red box highlights the Jenkinsfile code. The code defines a pipeline with an agent labeled 'built-in' using a custom workspace. It contains a single stage named 'One' with a series of shell steps to run Docker commands, copy files, and update Apache's htdocs directory.

```
1     pipeline {
2         agent {
3             label {
4                 label "built-in"
5                 customWorkspace "/mnt/MyWorkspace"
6             }
7         }
8         stages {
9             stage ("One"){
10                steps {
11                    sh "sudo docker ps -aq | xargs -r docker rm -f"
12                    sh "sudo docker run -dp 80:80 --name Container2 httpd"
13                    sh "sudo cp /mnt/MyWorkspace/index.html /Git"
14                    sh "sudo docker cp /Git/index.html Container2:/usr/local/apache2/htdocs/"
15                }
16            }
17        }
18    }
19 }
```

# Result:

- When changes are done in 2026Q1 branch, Build is triggered by ‘Doc-Assign2-Pipeline-A’ and the updated index.html file is hosted from the container of a **Slave1 instance** by using the Pipeline script in the ‘Jenkinsfile’ file of the same branch:

The screenshot displays four browser windows illustrating the Jenkins pipeline process for the 2026Q1 branch.

- Top Left:** GitHub repository page for Velocity-App showing a pull request for the 2026Q1 branch. A red box highlights the commit message: "Hosting updated index.html from Container 1 of 2026Q1 Branch Change 1 Change 2 Done with Docker Assignment 2".
- Top Right:** GitHub Jenkinsfile page for the 2026Q1 branch. A red box highlights the Jenkinsfile code:

```
pipeline {
    agent {
        label "slave1"
        customWorkspace "/mnt/myworkspace"
    }
    stages {
        stage ("One"){
            steps {
                sh "sudo docker ps -aq | xargs -r sudo docker rm -f"
                sh "sudo docker run -dp 80:80 --name Container1 httpd"
                sh "sudo cp /mnt/myworkspace/index.html /git"
                sh "sudo docker cp /git/index.html Container1:/usr/local/apache2/htdocs/"
            }
        }
    }
}
```
- Bottom Left:** Jenkins job details for Doc-Assign2-Pipeline-A #24. A red box highlights the build status: "24 (10 Feb 2026, 00:53:26)".
- Bottom Right:** Browser window showing the deployed content at 52.66.208.88/index.html. A red box highlights the message: "Hosting updated index.html from Container 1 of 2026Q1 Branch Change 1 Chnage 2 Done with Docker Assignment 2".

- When changes are done in 2026Q2 branch, Build is triggered by ‘Doc-Assign2-Pipeline-B’ and the updated index.html file is hosted from the container of a **Slave2 instance** by using the Pipeline script in the ‘Jenkinsfile’ file of the same branch:

The screenshot displays four browser windows illustrating the Jenkins pipeline process for the 2026Q2 branch.

- Top Left:** GitHub repository page for Velocity-App showing a pull request for the 2026Q2 branch. A red box highlights the commit message: "Hosting updated Index.html from Container 2 of 2026Q2 Branch Change 1 Change 2 Manish here".
- Top Right:** GitHub Jenkinsfile page for the 2026Q2 branch. A red box highlights the Jenkinsfile code:

```
pipeline {
    agent {
        label "slave2"
        customWorkspace "/mnt/myworkspace"
    }
    stages {
        stage ("One"){
            steps {
                sh "sudo docker ps -aq | xargs -r sudo docker rm -f"
                sh "sudo docker run -dp 80:80 --name Container2 httpd"
                sh "sudo cp /mnt/myworkspace/index.html /git"
                sh "sudo docker cp /git/index.html Container2:/usr/local/apache2/htdocs/"
            }
        }
    }
}
```
- Bottom Left:** Jenkins job details for Doc-Assign2-Pipeline-B #16. A red box highlights the build status: "#16 (10 Feb 2026, 00:49:31)".
- Bottom Right:** Browser window showing the deployed content at 13.126.38.14/index.html. A red box highlights the message: "Hosting updated index.html from Container 2 of 2026Q2 Branch Change 1 Change 2 Manish here".