

Jenkins Assignment 4

Step 1: Launched four instances, Jenkins Master, Slave1, Slave2 and Slave3

The screenshot shows the AWS Management Console for the 'Instances' page. The left sidebar contains navigation links for EC2, Dashboard, AWS Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, and Lifecycle Manager. The main content area displays a table of 5 instances:

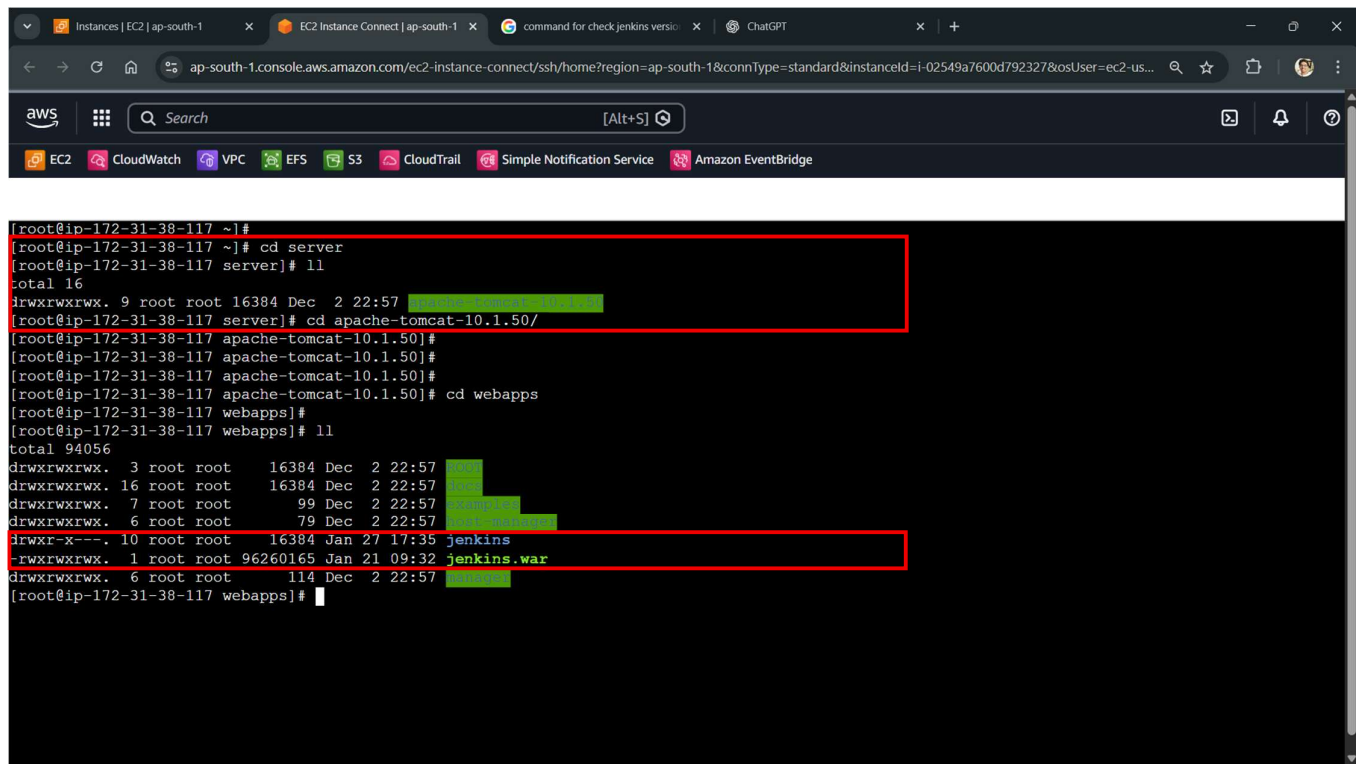
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Jenkins Master	i-03bfc0e5ce6262048	Running	t3.micro	3/3 checks passed	View alarms +	ap-south-1a	ec2-43-205-210-
Slave1	i-07c71e056e325569f	Running	t3.micro	3/3 checks passed	View alarms +	ap-south-1a	ec2-52-66-184-1
Slave2	i-0bc1750315c74562f	Running	t3.micro	3/3 checks passed	View alarms +	ap-south-1a	ec2-65-2-183-9.a
Slave3	i-06eeb09f151a1ed2f	Running	t3.micro	3/3 checks passed	View alarms +	ap-south-1a	ec2-13-201-56-2

Below the table, there are monitoring graphs for CPU utilization, network in/out, and network packets. The CPU utilization graph shows a peak around 21:00. The network in/out graphs show data for the same time period.

Step 2: Installed Java -17 and Apache-HTTPD on all the instance:

```
[root@ip-172-31-36-242 ~]#  
[root@ip-172-31-36-242 ~]#  
[root@ip-172-31-36-242 ~]#  
[root@ip-172-31-36-242 ~]# java -version  
openjdk version "17.0.17" 2025-10-21 LTS  
OpenJDK Runtime Environment Corretto-17.0.17.10.1 (build 17.0.17+10-LTS)  
OpenJDK 64-Bit Server VM Corretto-17.0.17.10.1 (build 17.0.17+10-LTS, mixed mode, sharing)  
[root@ip-172-31-36-242 ~]#  
[root@ip-172-31-36-242 ~]#  
[root@ip-172-31-36-242 ~]#  
[root@ip-172-31-36-242 ~]#  
[root@ip-172-31-36-242 ~]#  
[root@ip-172-31-36-242 ~]# service httpd status  
Redirecting to /bin/systemctl status httpd.service  
o httpd.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)  
   Active: inactive (dead)  
   Docs: man:httpd.service(8)  
[root@ip-172-31-36-242 ~]#
```

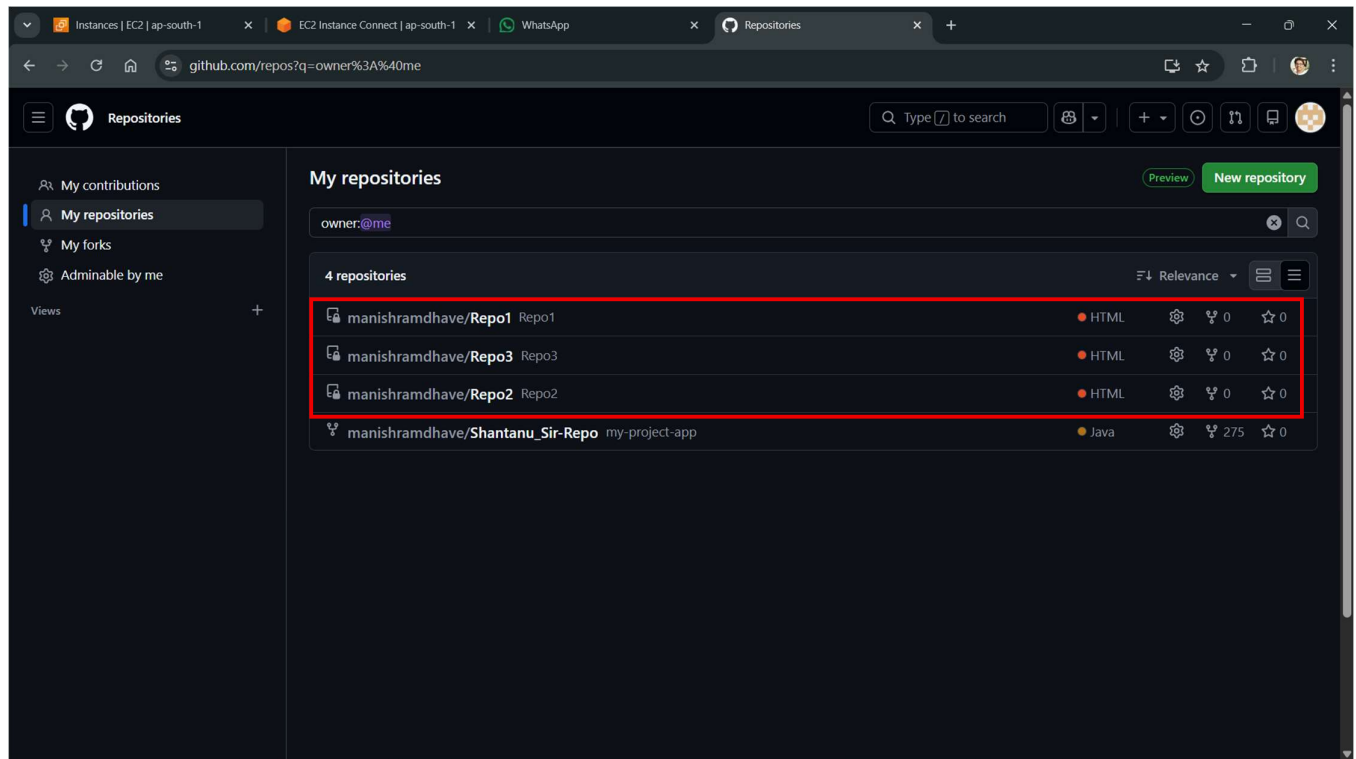
Step 3: Installed Apache-Tomcat-10 and Jenkins on the Jenkins Master Instance:



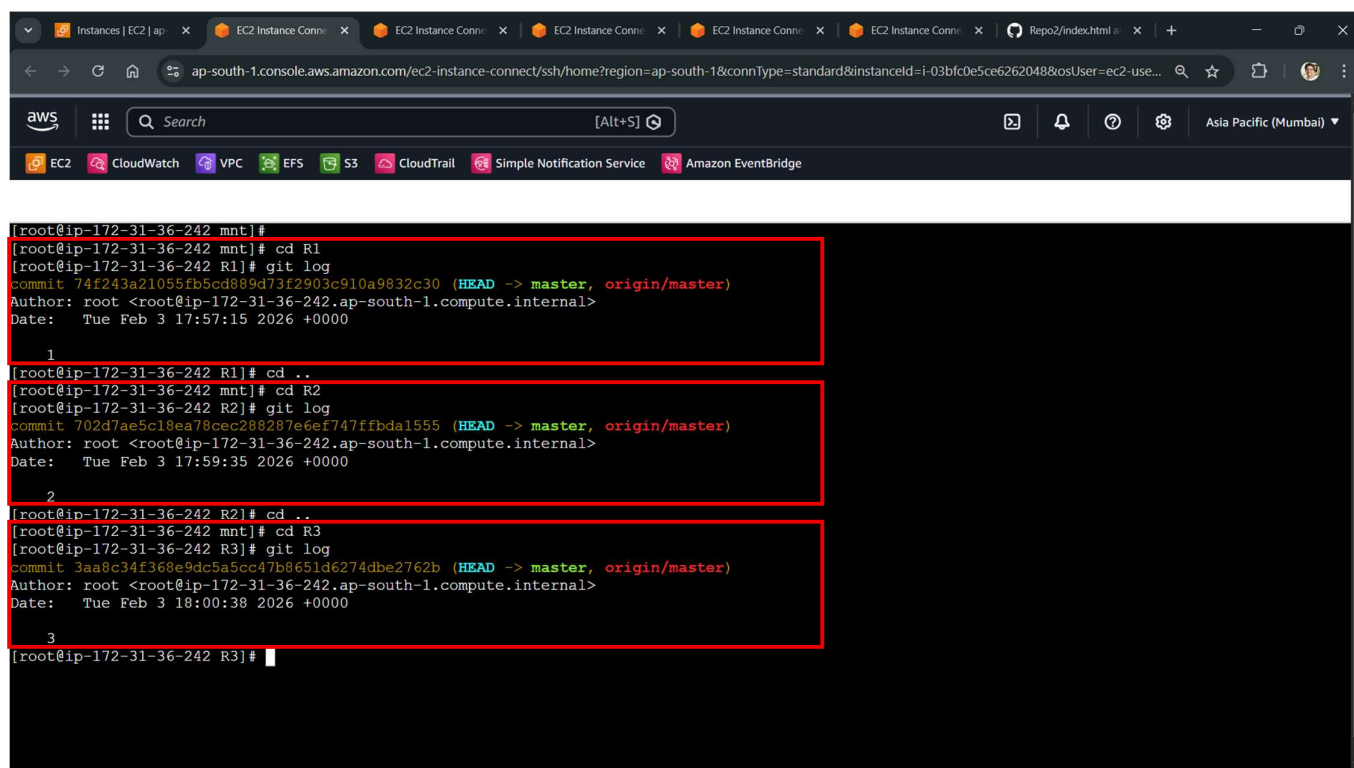
The screenshot shows a terminal window within the AWS console, connected to an EC2 instance. The user is in the root directory of the instance. They first navigate to the 'server' directory and list its contents, showing a file named 'apache-tomcat-10.1.50'. Then, they navigate to the 'apache-tomcat-10.1.50' directory and list its contents, showing a file named 'webapps'. Finally, they navigate to the 'webapps' directory and list its contents, showing a file named 'jenkins.war'. The terminal output is as follows:

```
[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 .  
drwxrwxrwx. 16 root root 16384 Dec 2 22:57 ..  
drwxrwxrwx. 7 root root 99 Dec 2 22:57 .svn  
drwxrwxrwx. 6 root root 79 Dec 2 22:57 .svn  
drwxr-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 .svn  
[root@ip-172-31-38-117 webapps]#
```

Step 4: Made three Private Repositories Repo1, Repo2 and Repo3 in GitHub account:

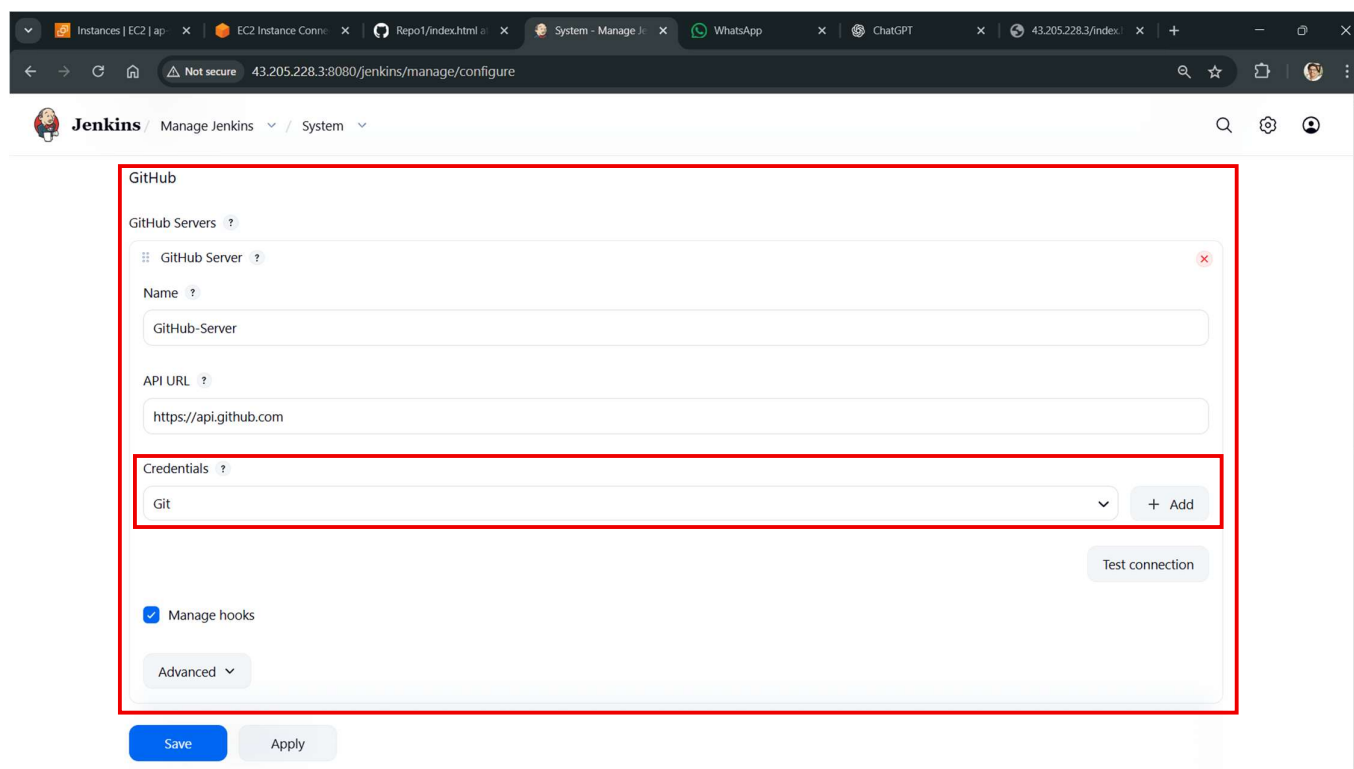


Step 5: Pushed three different index.html files on repositories Repo1, Repo2 and Repo3 respectively:



```
[root@ip-172-31-36-242 mnt]#  
[root@ip-172-31-36-242 mnt]# cd R1  
[root@ip-172-31-36-242 R1]# git log  
commit 74f243a21055fb5cd889d73f2903c910a9832c30 (HEAD -> master, origin/master)  
Author: root <root@ip-172-31-36-242.ap-south-1.compute.internal>  
Date: Tue Feb 3 17:57:15 2026 +0000  
1  
[root@ip-172-31-36-242 R1]# cd ..  
[root@ip-172-31-36-242 mnt]# cd R2  
[root@ip-172-31-36-242 R2]# git log  
commit 702d7ae5c18ea78cec288287e6ef747ffbdal555 (HEAD -> master, origin/master)  
Author: root <root@ip-172-31-36-242.ap-south-1.compute.internal>  
Date: Tue Feb 3 17:59:35 2026 +0000  
2  
[root@ip-172-31-36-242 R2]# cd ..  
[root@ip-172-31-36-242 mnt]# cd R3  
[root@ip-172-31-36-242 R3]# git log  
commit 3aa8c34f368e9dc5a5cc47b8651d6274dbe2762b (HEAD -> master, origin/master)  
Author: root <root@ip-172-31-36-242.ap-south-1.compute.internal>  
Date: Tue Feb 3 18:00:38 2026 +0000  
3  
[root@ip-172-31-36-242 R3]#
```

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:



Jenkins / Manage Jenkins / System

GitHub

GitHub Servers ?

GitHub Server ?

Name ?

GitHub-Server

API URL ?

https://api.github.com

Credentials ?

Git

+ Add

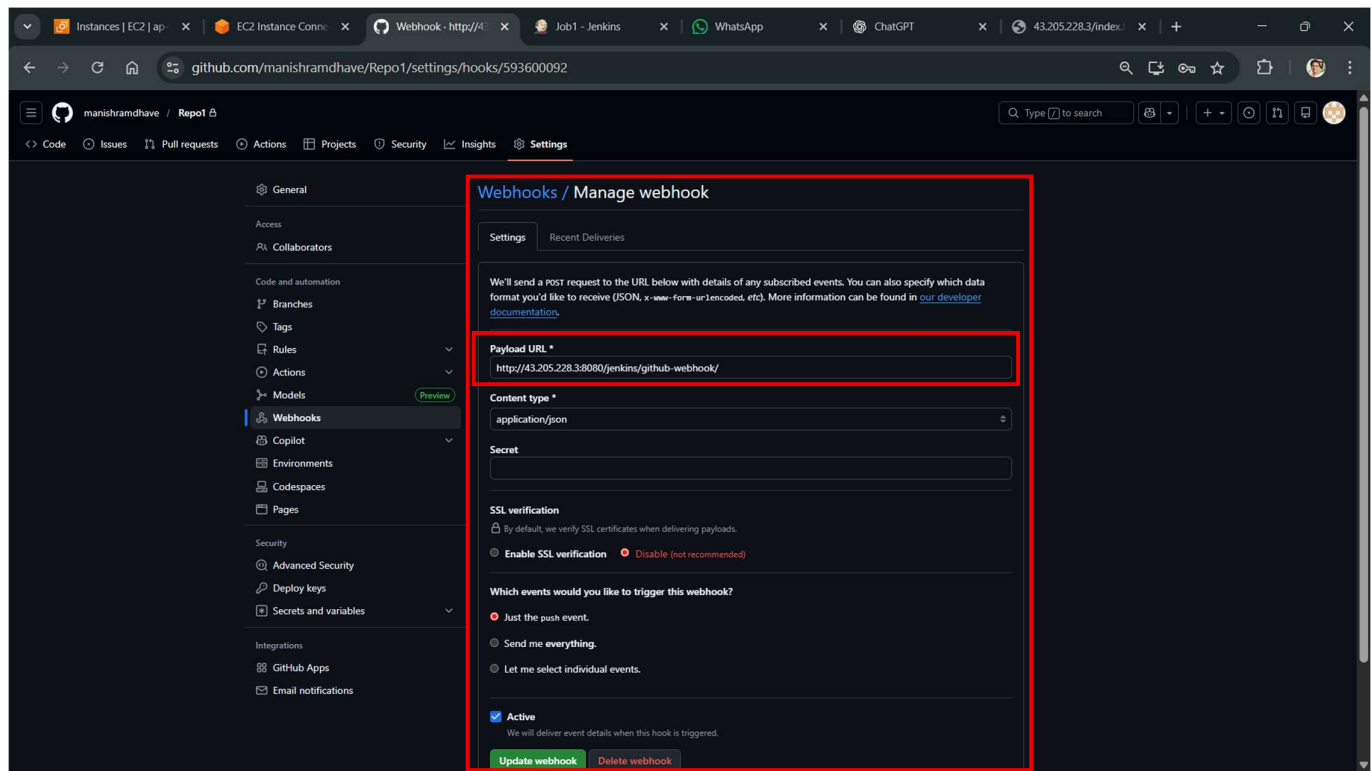
Test connection

☒ Manage hooks

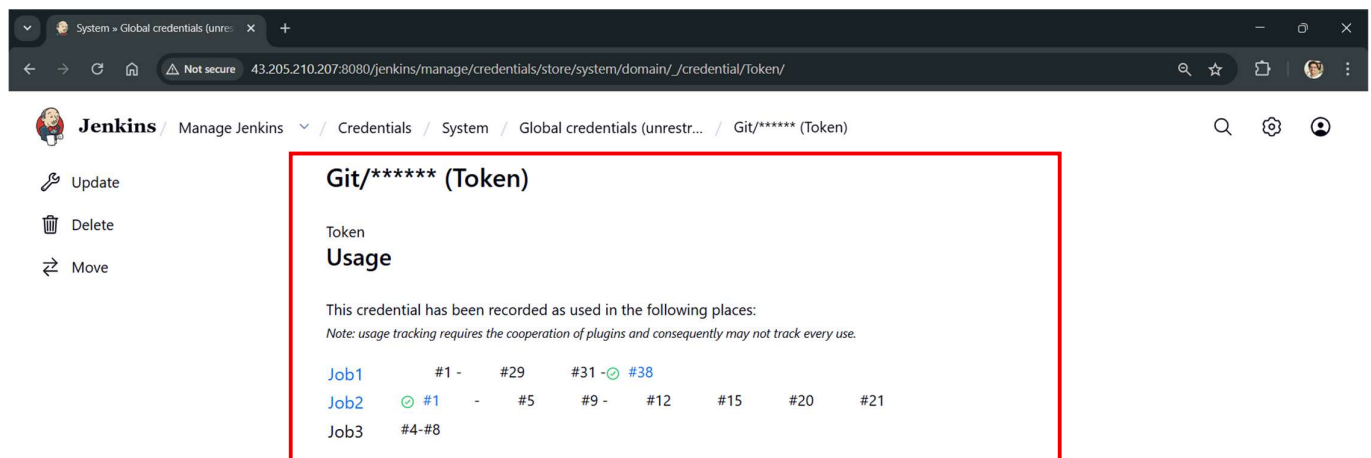
Advanced

Save Apply

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



Step 8: Created a ‘Username and Password’ Credential for accessing the Private Git Repositories using the GitHub Token:



Step 9: Created a 'SSH Username with private key' Credential for Jenkins Master and Slave Connections (One Credential is enough for all the Slaves):

Update credentials

Scope ?
Global (Jenkins, nodes, items, all child items, etc)

ID ?
Slave

Description ?
Slave

Username
ec2-user

☐ Treat username as secret ?

Private Key
☒ Enter directly
Key
Concealed for Confidentiality Replace

Save

Step 10: Created three Nodes for to establish the connection between Jenkins Master and Slave Instances using the 'Credential' and 'Manually trusted key Verification Strategy':

Nodes

+ New Node Configure Monitors Refresh

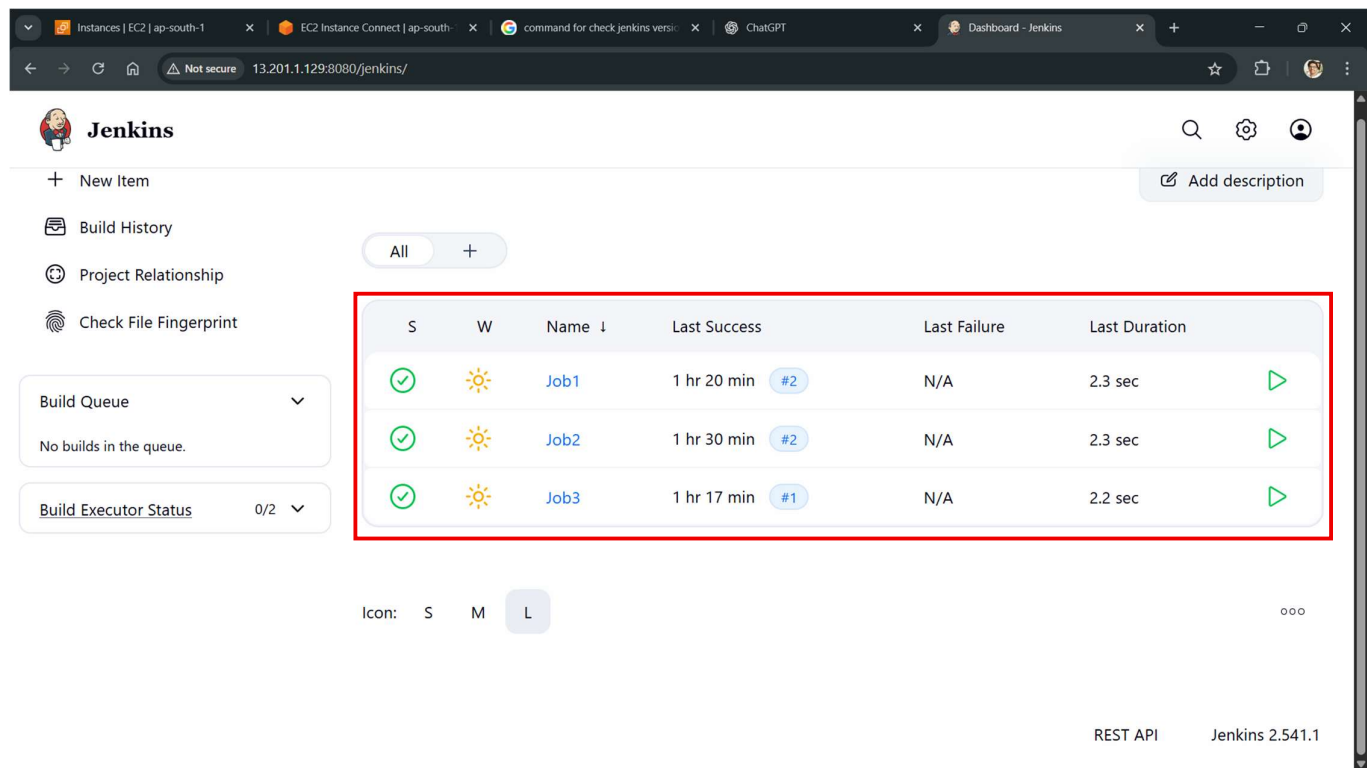
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	5.64 GiB	0 B	5.64 GiB	0ms
	Slave1	Linux (amd64)	In sync	6.02 GiB	0 B	458.39 MiB	90ms
	Slave2	Linux (amd64)	In sync	6.02 GiB	0 B	458.38 MiB	71ms
	Slave3	Linux (amd64)	In sync	6.03 GiB	0 B	458.26 MiB	51ms
last checked		44 min	44 min	44 min	44 min	44 min	44 min

Icon: S M L Legend

REST API Jenkins 2.541.1

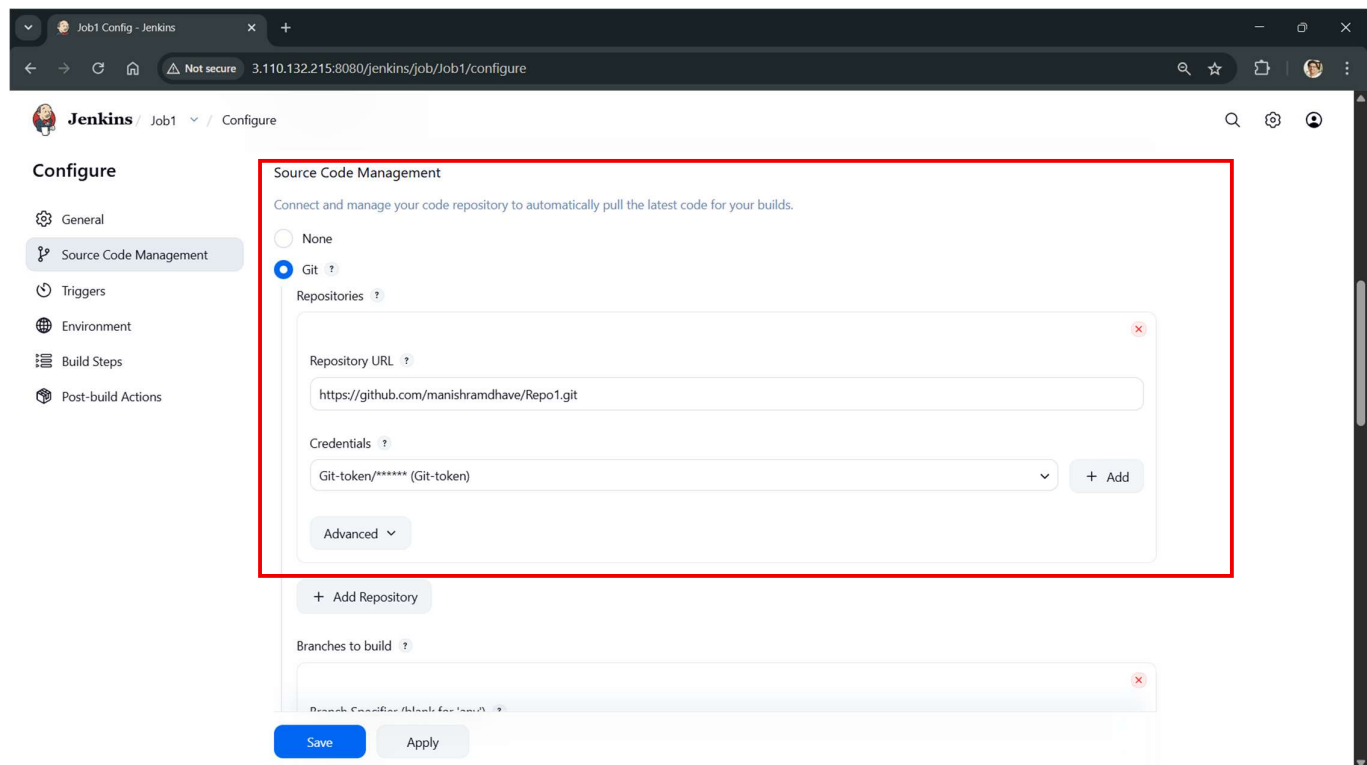
Step 11: Launched the Jenkins and created three different Jobs Job1, Job2 and Job3:



The screenshot shows the Jenkins dashboard with the following table of jobs:

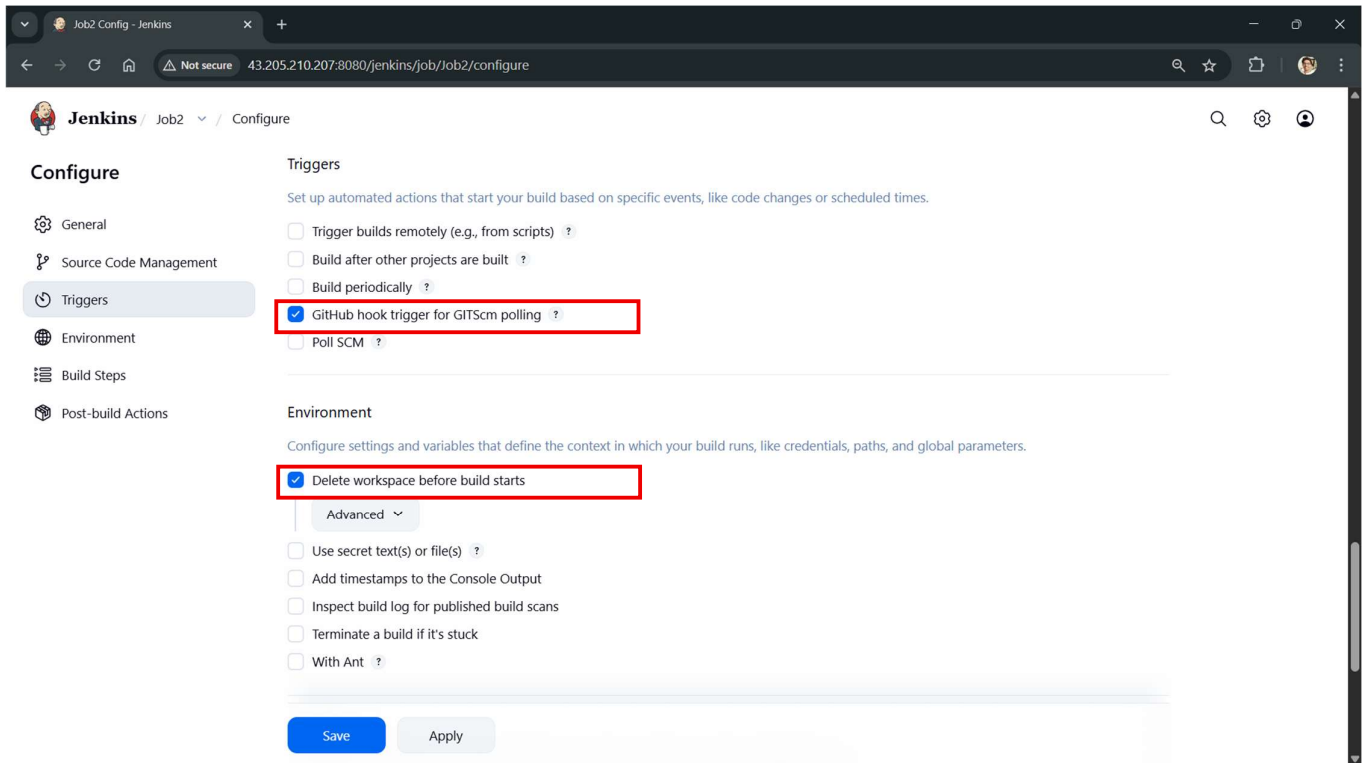
S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	Job1	1 hr 20 min #2	N/A	2.3 sec
✓	☀	Job2	1 hr 30 min #2	N/A	2.3 sec
✓	☀	Job3	1 hr 17 min #1	N/A	2.2 sec

Step 12: Integrated Git with Jenkins by creating 'Credentials' by using the Git Token:

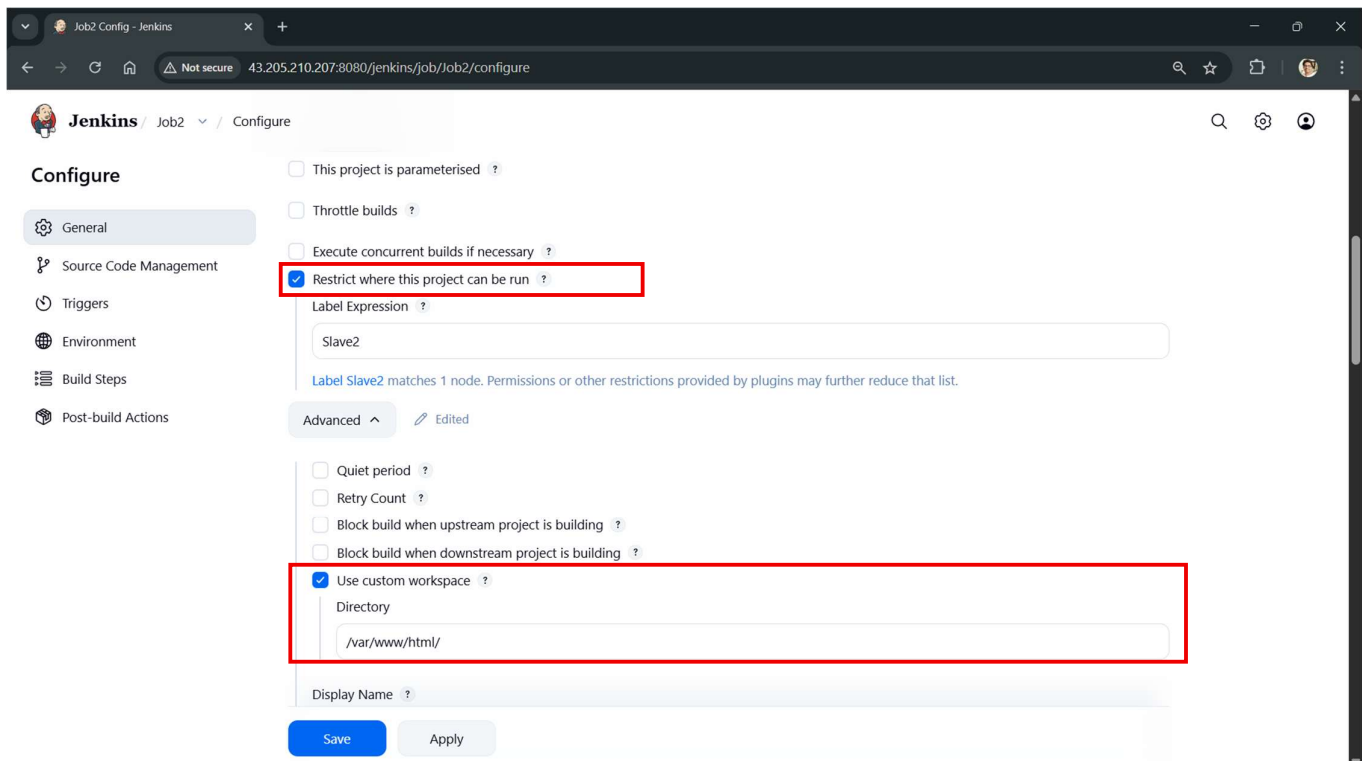


The screenshot shows the Jenkins 'Configure' page for Job1, with the 'Source Code Management' section highlighted. The 'Git' option is selected, and the 'Repository URL' is set to 'https://github.com/manishramdhave/Repo1.git'. The 'Credentials' dropdown is set to 'Git-token/***** (Git-token)'. The 'Advanced' section is expanded, and the 'Add Repository' button is visible. The 'Branches to build' section is also visible.

Step 13: Enabled **‘GitHub hook Trigger for GITScm polling’** while configuring all three Jenkins Jobs and also enabled **‘Delete workspace before build starts’**:

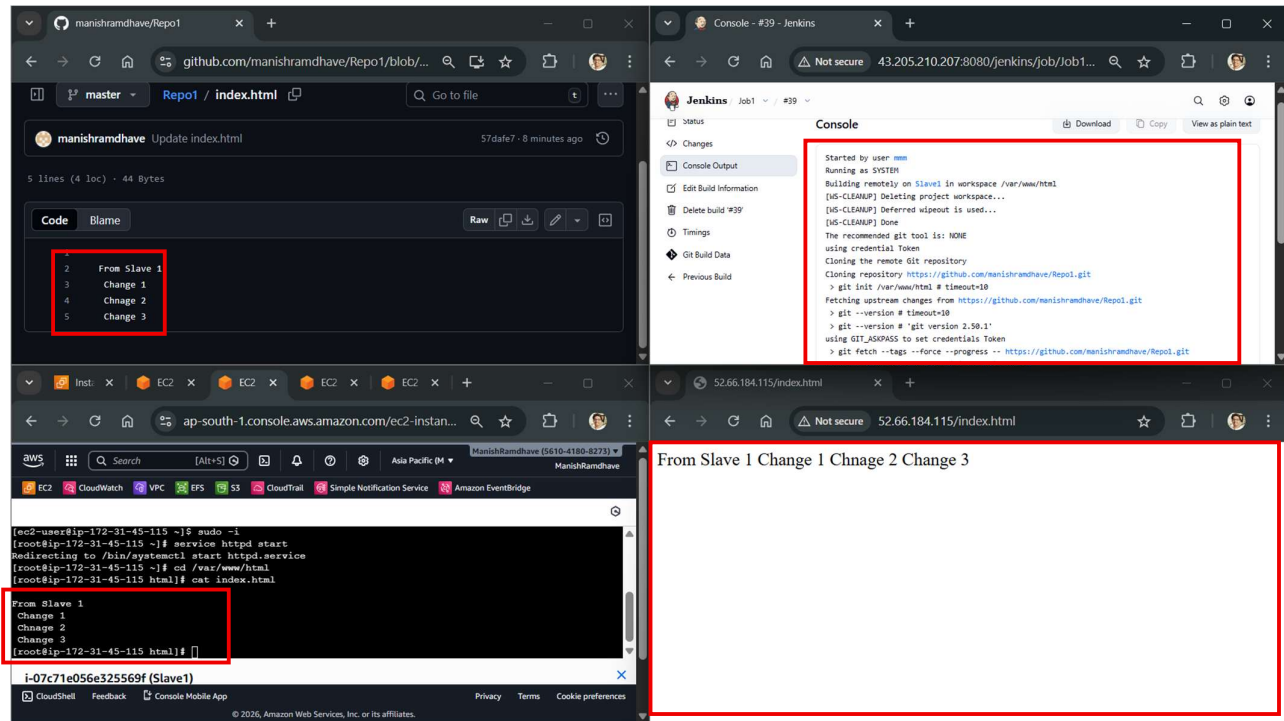


Step 14: Enabled **‘Restrict where this project can be run’** and used a custom workspace to deploy the index.html file from Git Repositories to **‘/var/www/html/’**:

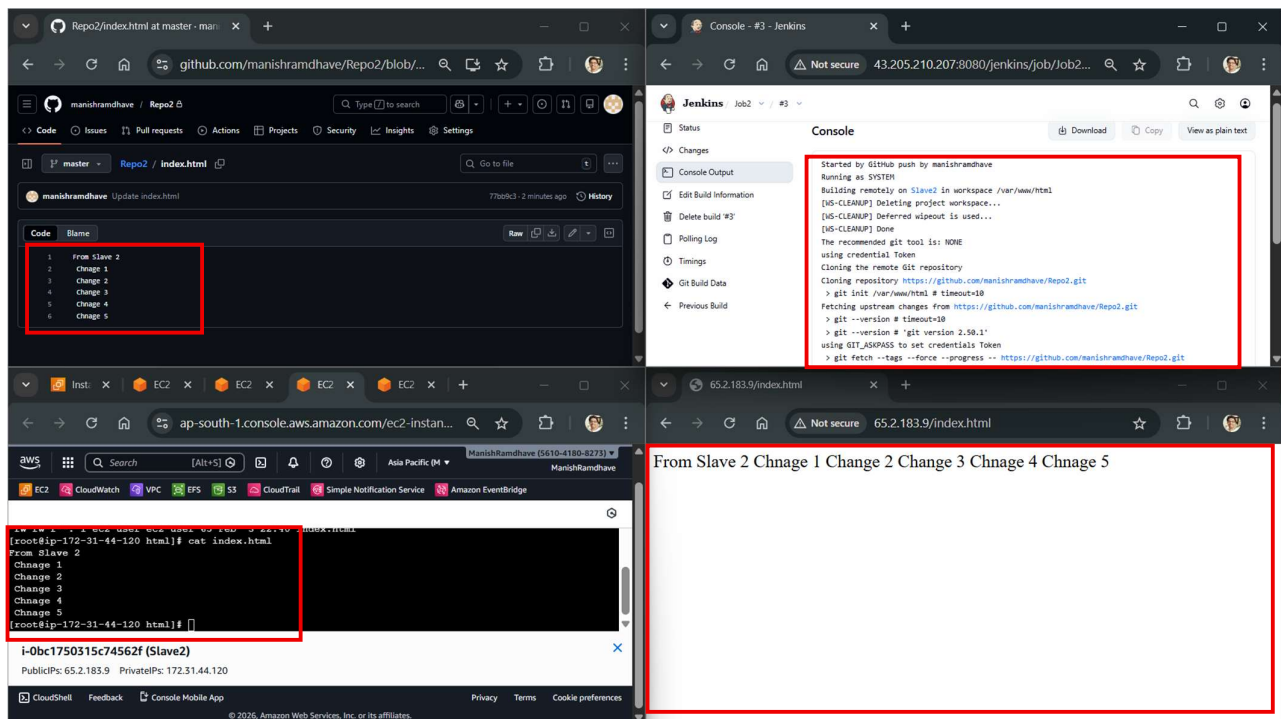


Result:

1. When changes are done in Git Repo1, Build is triggered by Job1 and the index.html file from Repo1 is hosted in the deployment folder of a **Slave1** instance:



2. When changes are done in Git Repo2, Build is triggered by Job2 and the index.html file from Repo2 is hosted in the deployment folder of a **Slave2** instance::



3. When changes are done in Git Repo3, Build is triggered by Job3 and the index.html file from Repo3 is hosted in the deployment folder of a **Slave3** instance::

The image is a composite of four screenshots illustrating a CI/CD pipeline:

- Top Left (GitHub):** Shows the 'Repo3' repository with a commit 'Update index.html' by 'manishramdhav'. The diff shows changes from 'Slave 3'.
- Top Right (Jenkins Console):** Shows the Jenkins console output for 'Job3'. The build process includes cloning the repository, fetching upstream changes, and running 'git fetch --tags --force --progress -- https://github.com/manishramdhav/Repo3.git'.
- Bottom Left (AWS CloudShell):** Shows the terminal output of an EC2 instance. The command 'cat index.html' is executed, displaying the commit message: 'From Slave 3 Chnage 1 Chnage 2 Hi there, I am Manish Ramdhav'.
- Bottom Right (Web Browser):** Shows the hosted 'index.html' file in a web browser, displaying the commit message content.