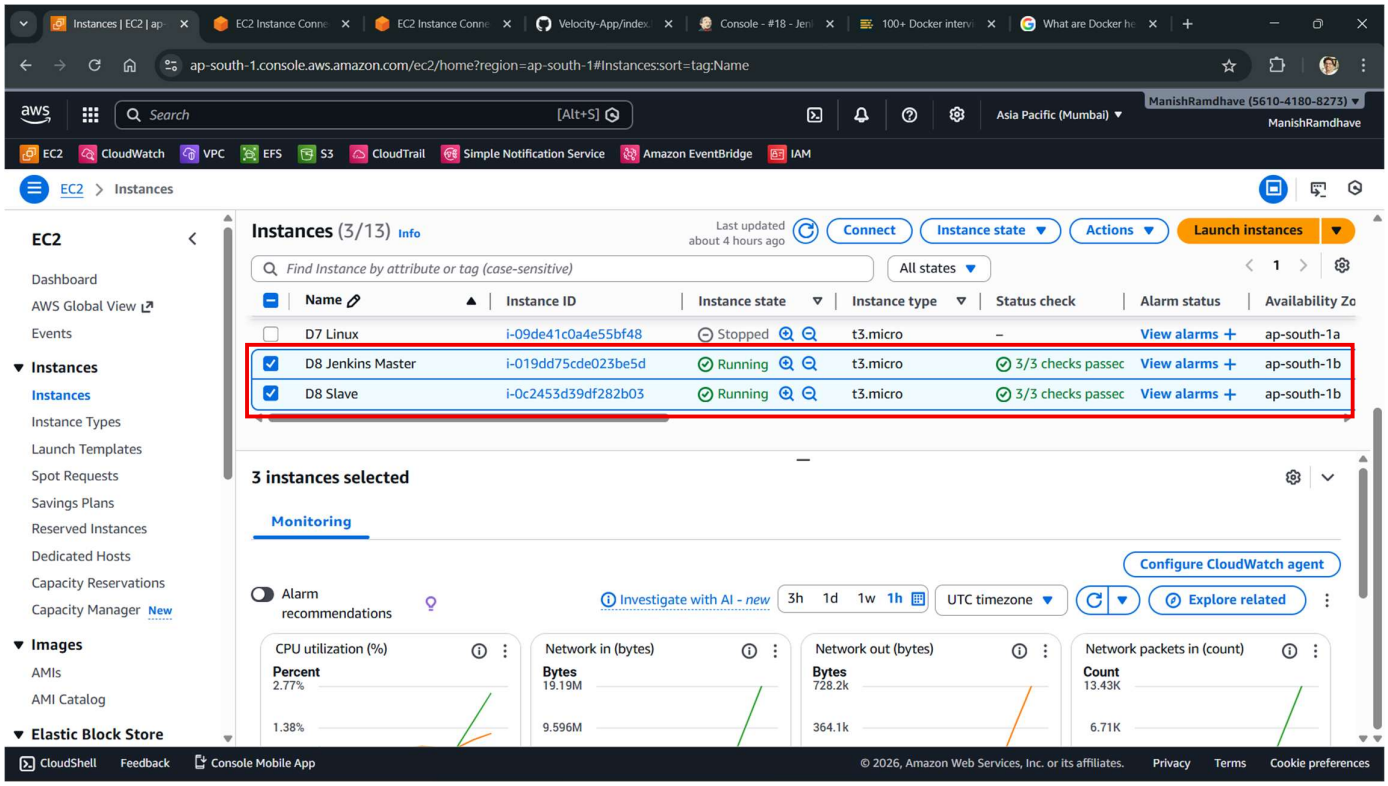
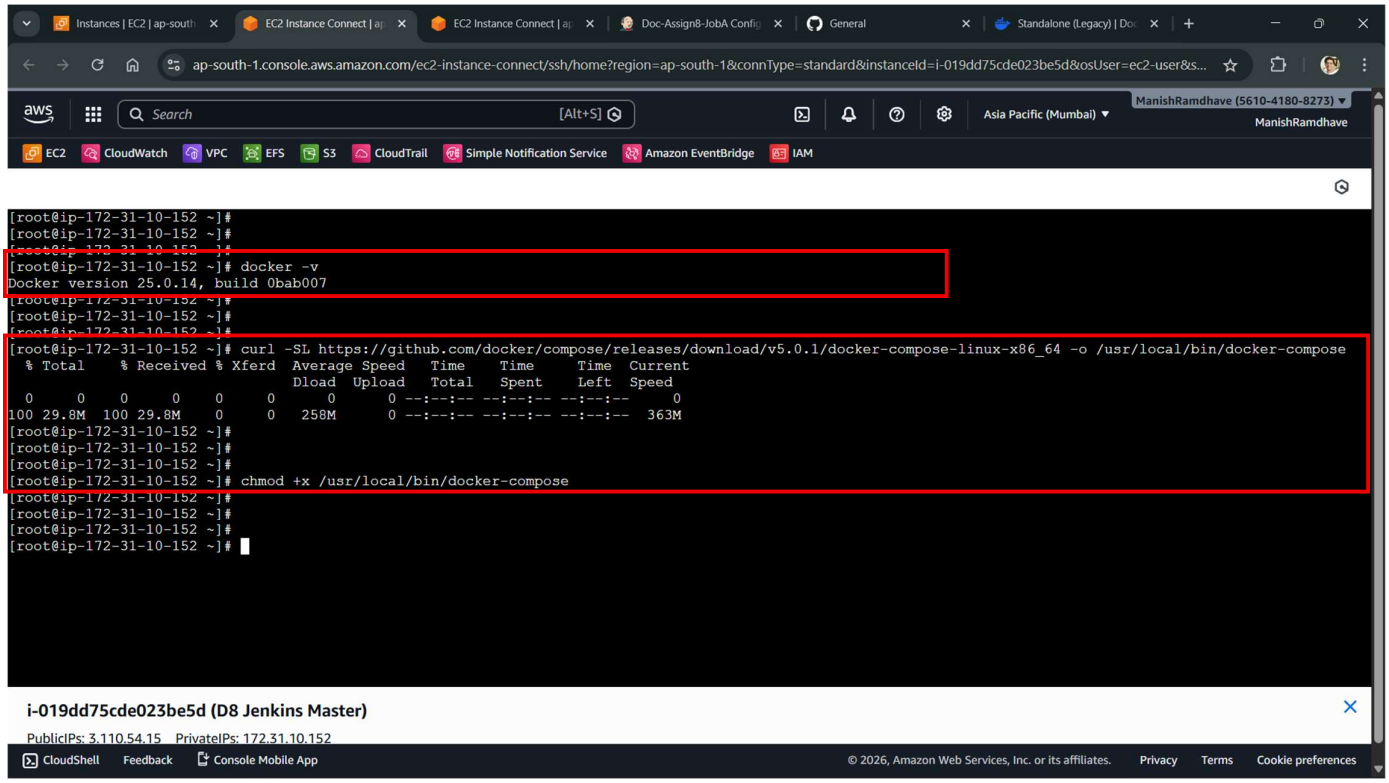
Docker Assignment 8 (Task 2)

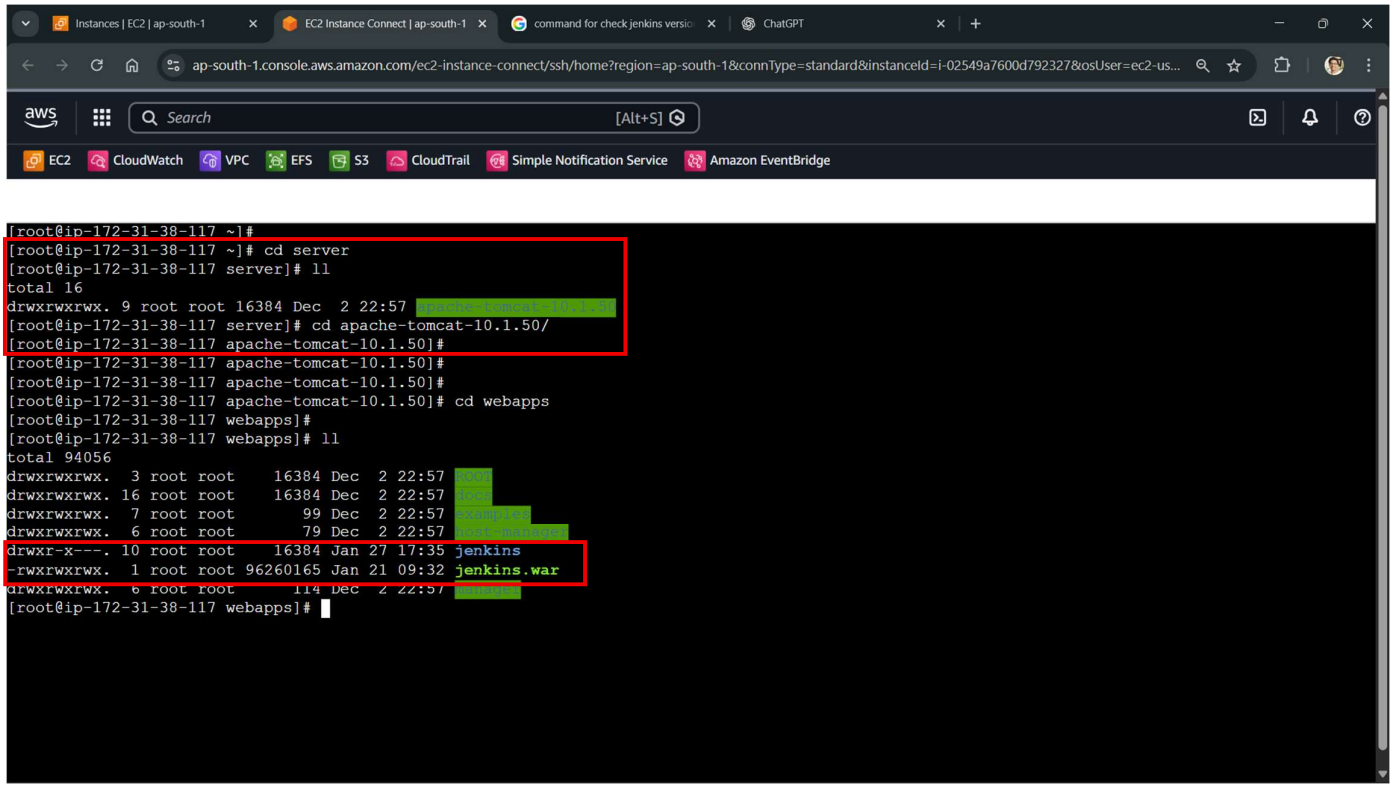
Step 1: Launched instances for our **Jenkins Master and Slave:**



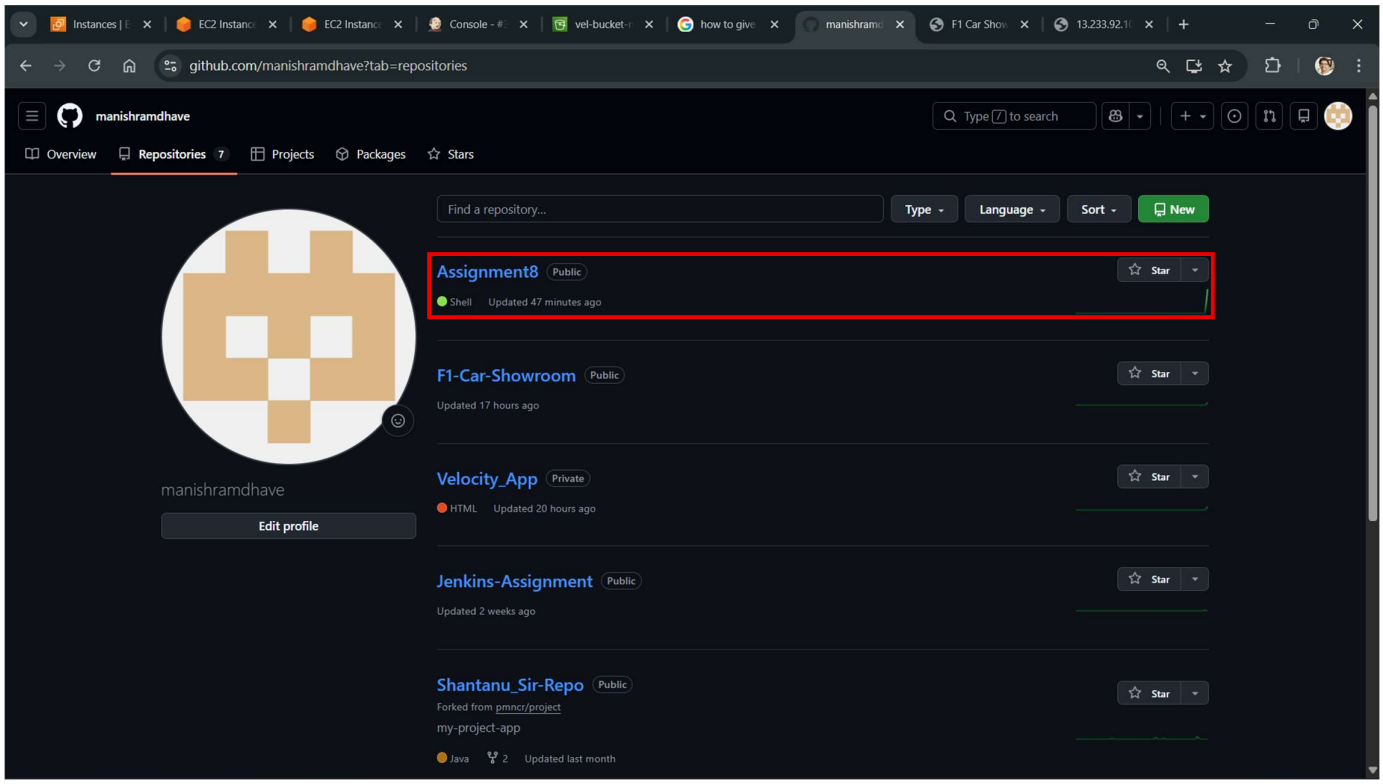
Step 2: Installed **Java-17** and **Docker** and **Docker Compose on the Jenkins Master Instance** and **Slave** also:



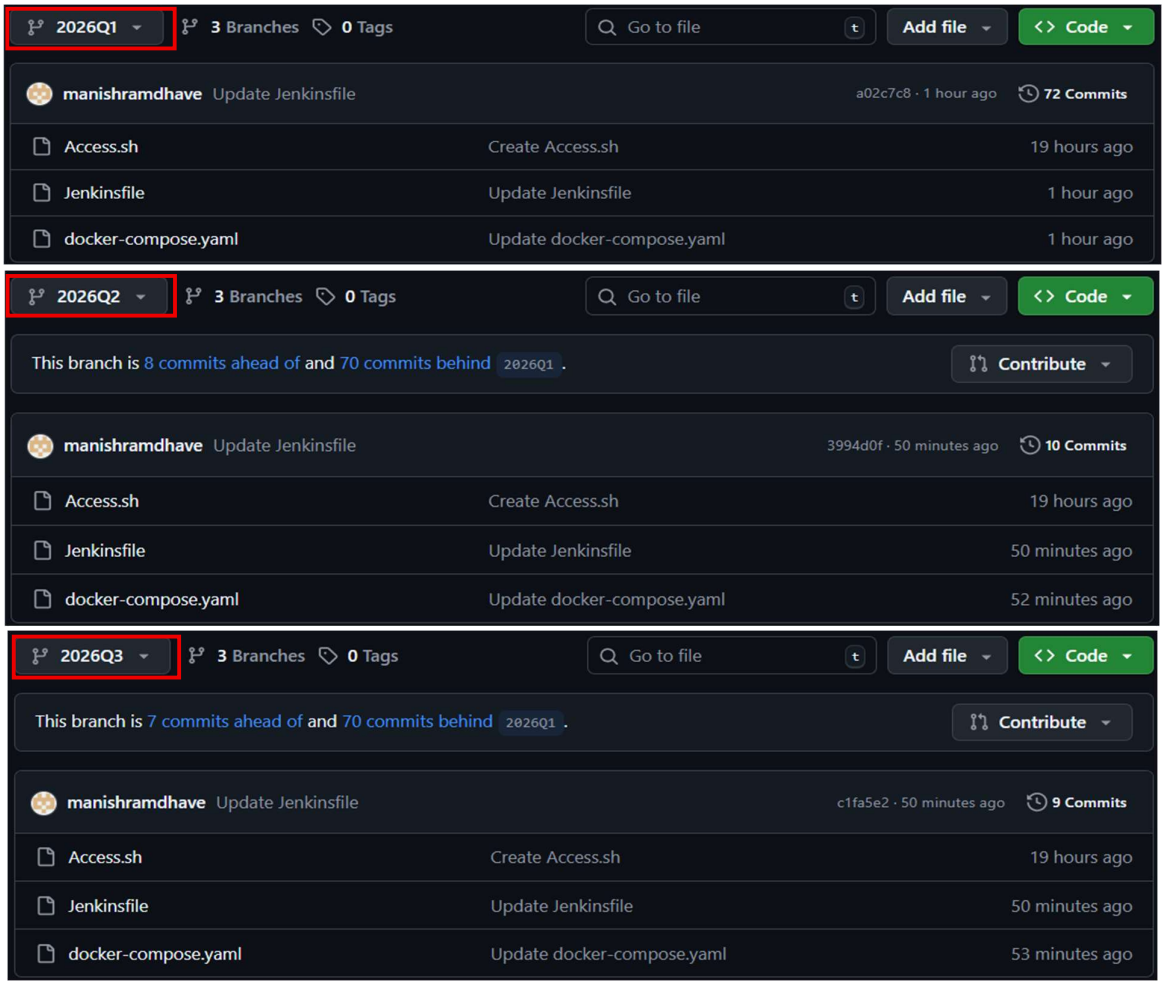
Step 3: Installed **Apache-Tomcat-10 and Jenkins** on the **Jenkins Master Instance**:



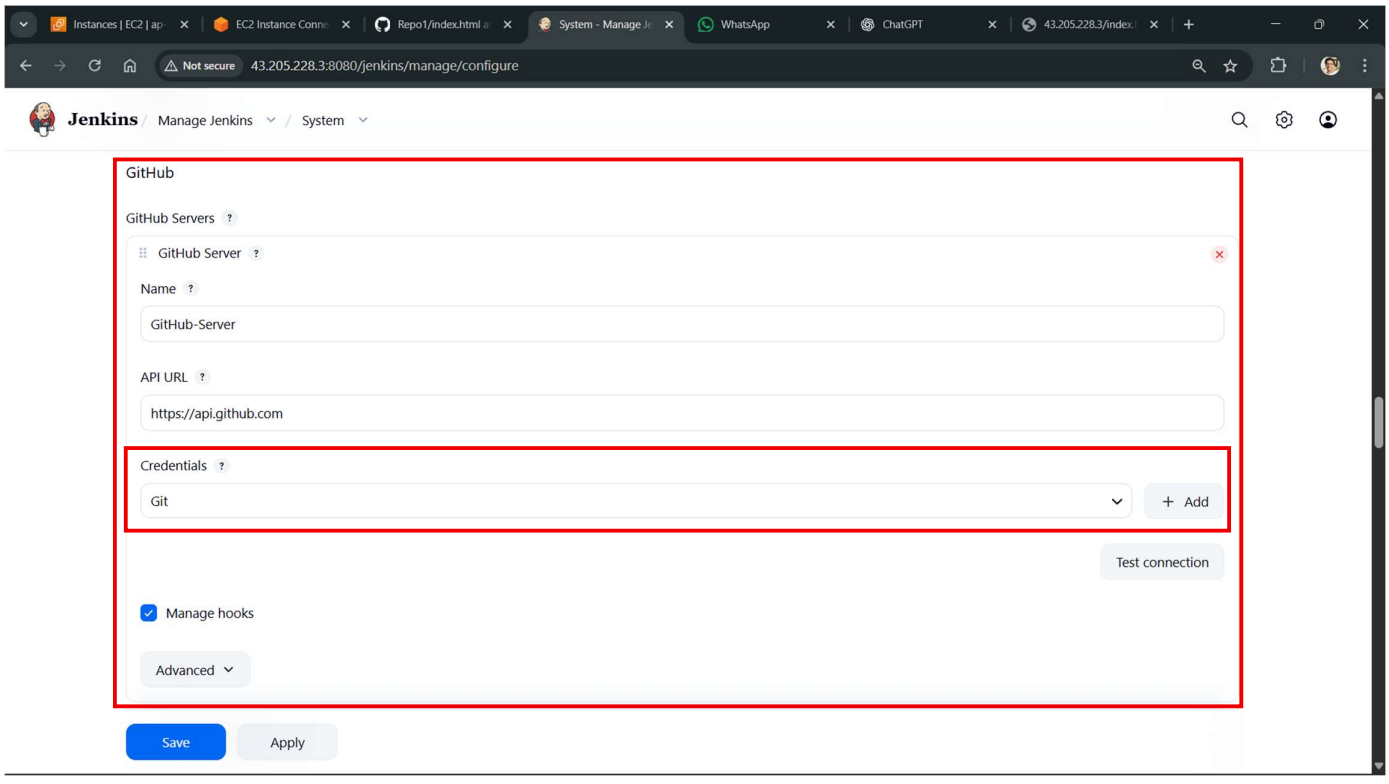
Step 4: Made a Private Repository named **'Assignment8'** in GitHub account:



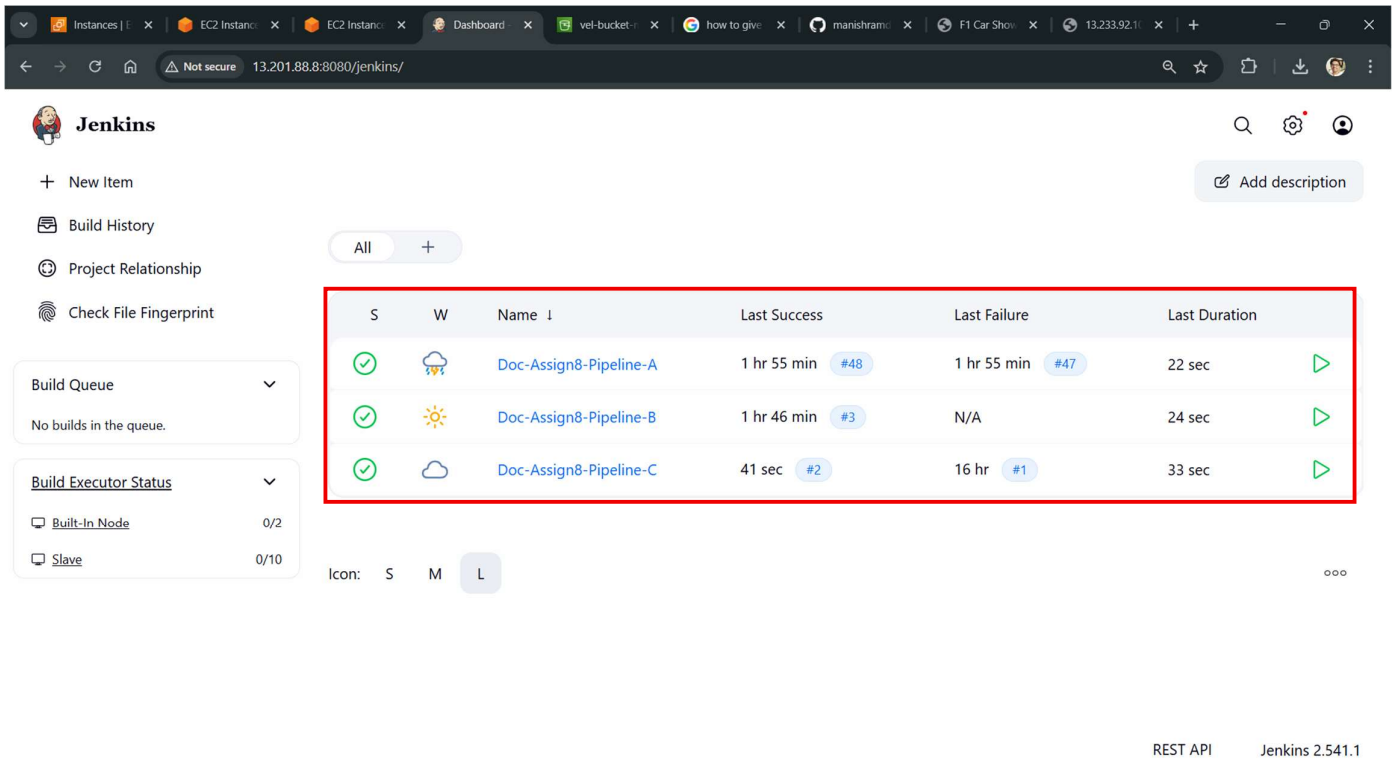
Step 5: Created three branches, **2026Q1, 2026Q2 and 2026Q3** in the '**Velocity-App**' Repository and pushed **three different 'Access.sh', 'Jenkinsfile'and 'docker-compose.yaml'** files in all the respective branches:



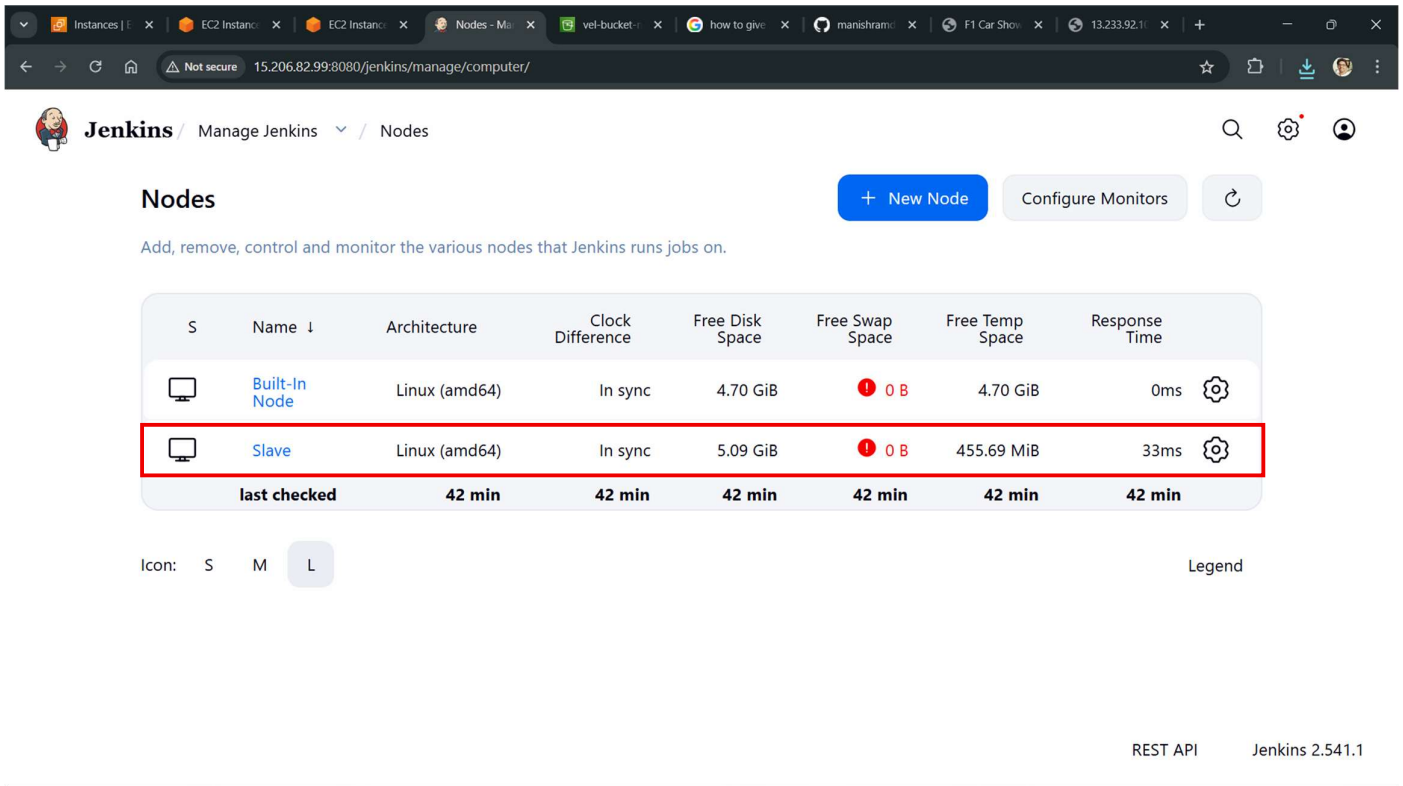
Step 6: Created an **API Connection between Jenkins to GitHub** Repositories in 'Manage Jenkins' by creating **a Secret Text (Credential)** using a GitHub Token in Jenkins:



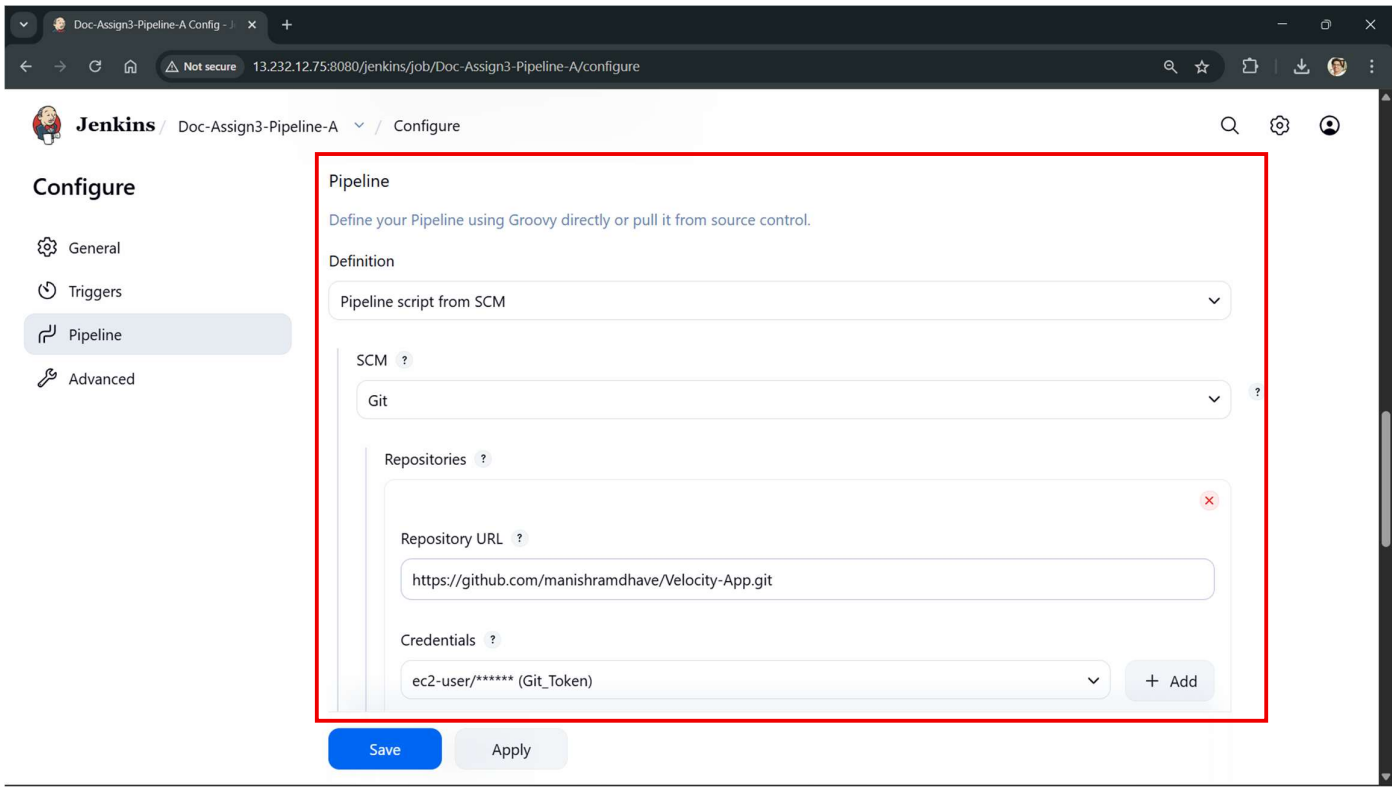
Step 7: Launched the **Jenkins** and created three different **Pipeline Jobs** in it:



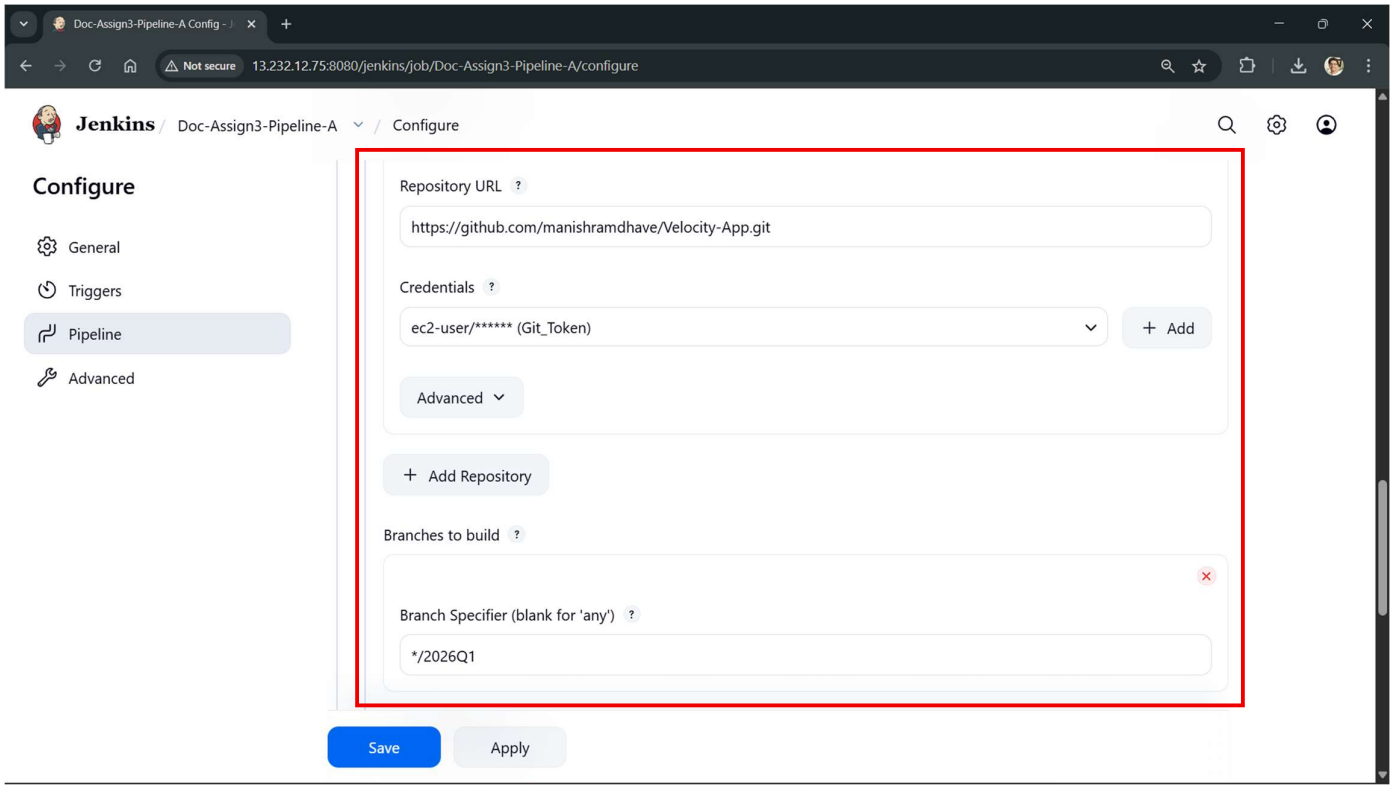
Step 8: Created an '**Node' Connection between Jenkins Master and Slave instances** 'Manage Jenkins' by creating **a SSH Username and Key (Credential)** using a Key-Pair and **Manually trusted key-verification Strategy**:



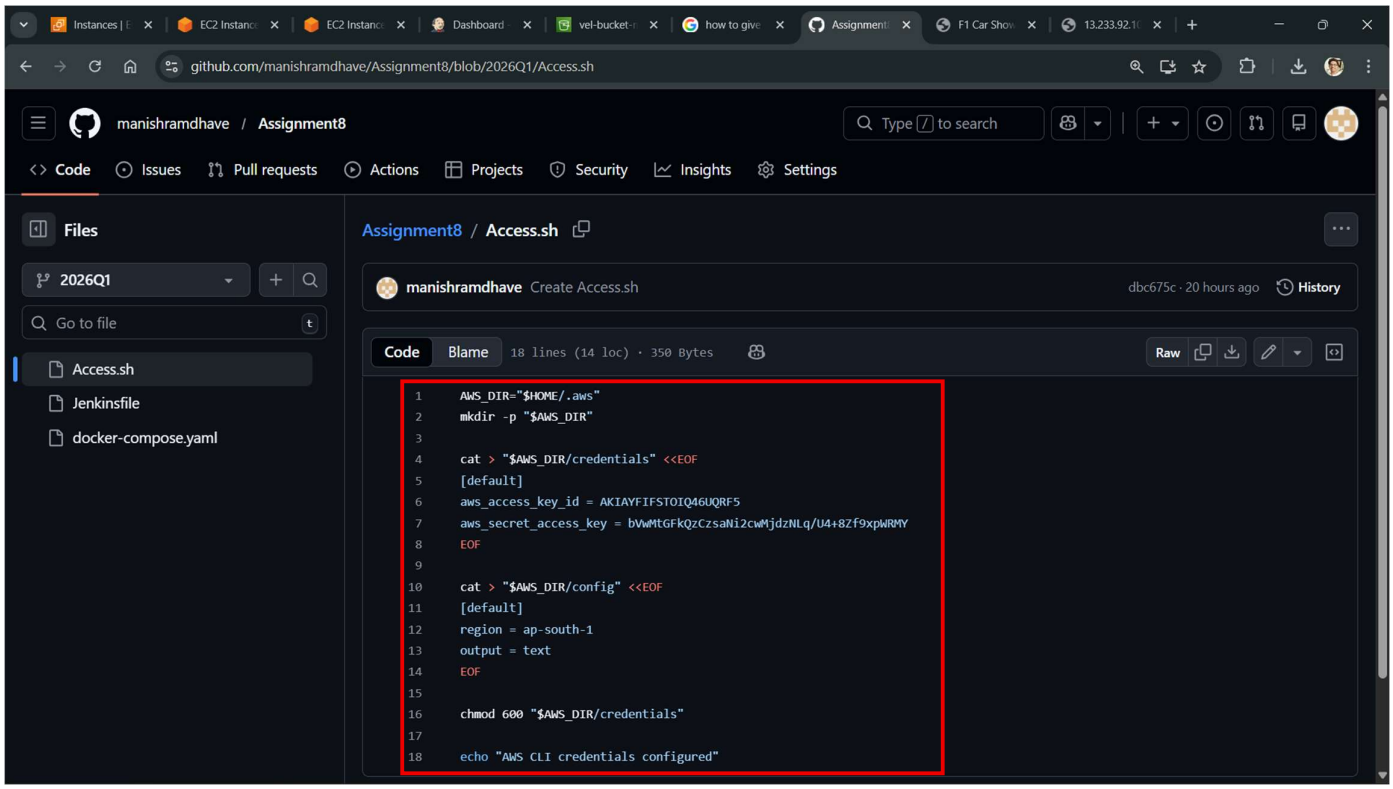
Step 9: Used **'Pipeline script from SCM'** and selected the SCM as a **'Git'** which will follow the Jenkins pipeline script by using the **'Jenkinsfile'** in the repository **in each pipeline job**:



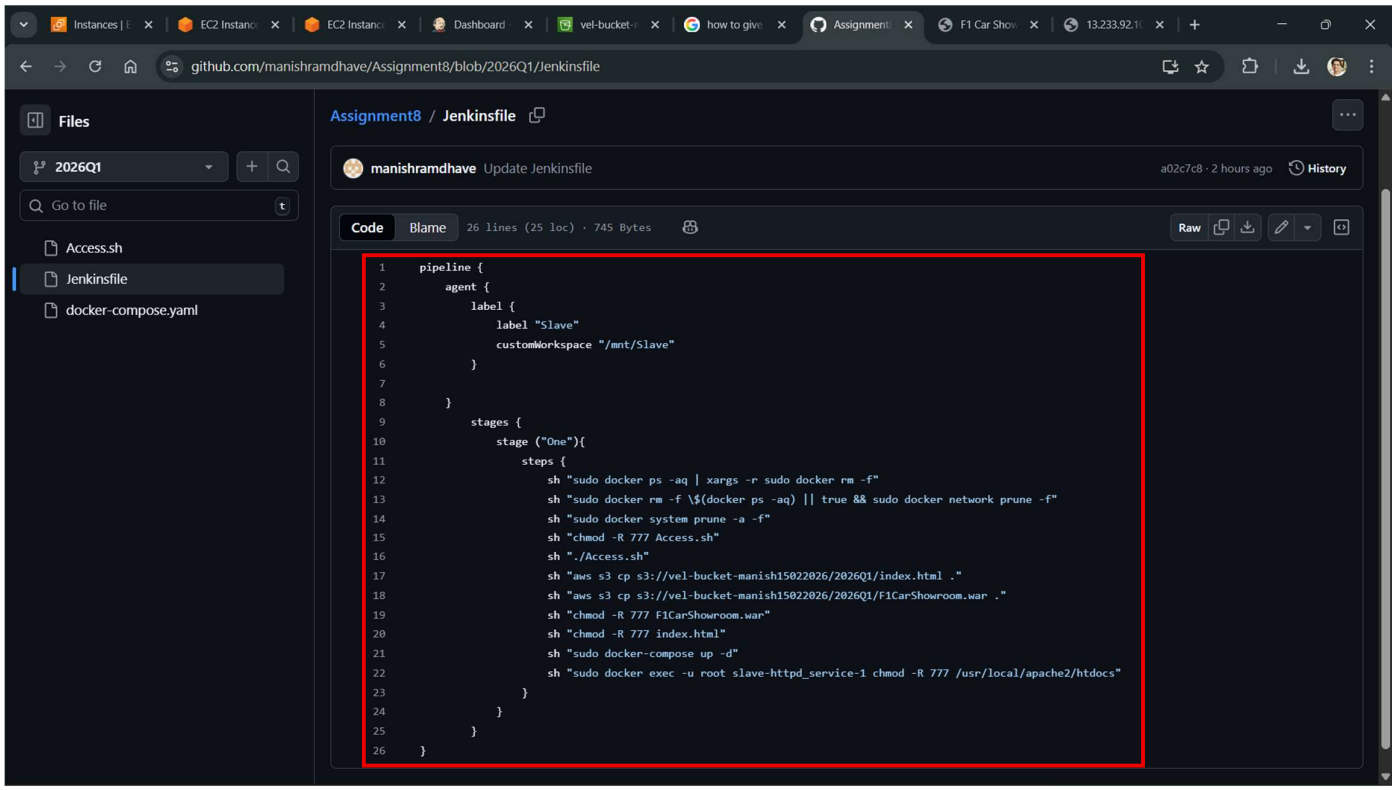
Step 10: Integrated all Git branches, **2026Q1, 2026Q2 and 2026Q3** with Jenkins by creating 'Credentials' by using the Git Token **on all the pipeline** respectively:



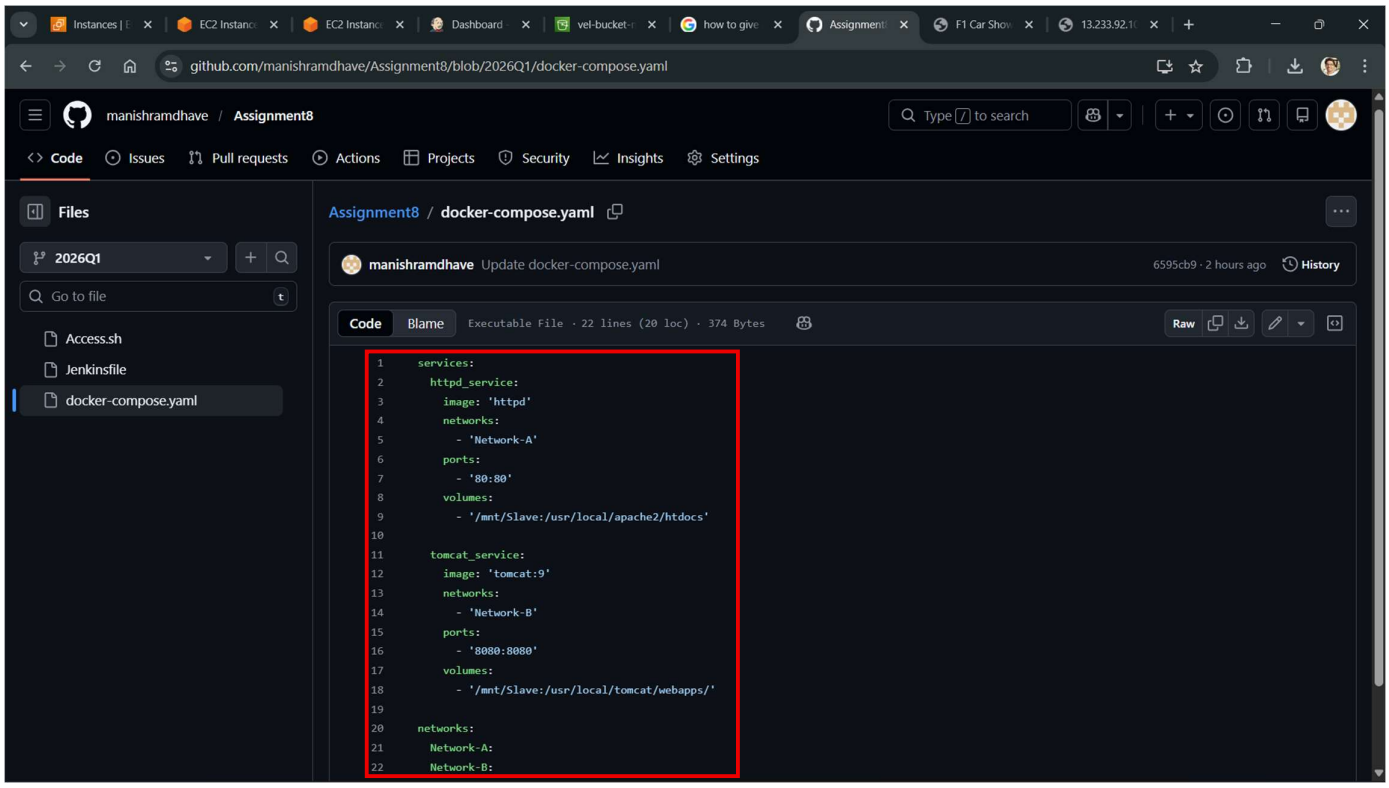
Step 11: Added AWS Configurations (i.e. **Access Key, Secret Key** etc.) in the **'Access.sh'** file which run by Jenkinsfile for **AWS CLI command processes in each Branch**:



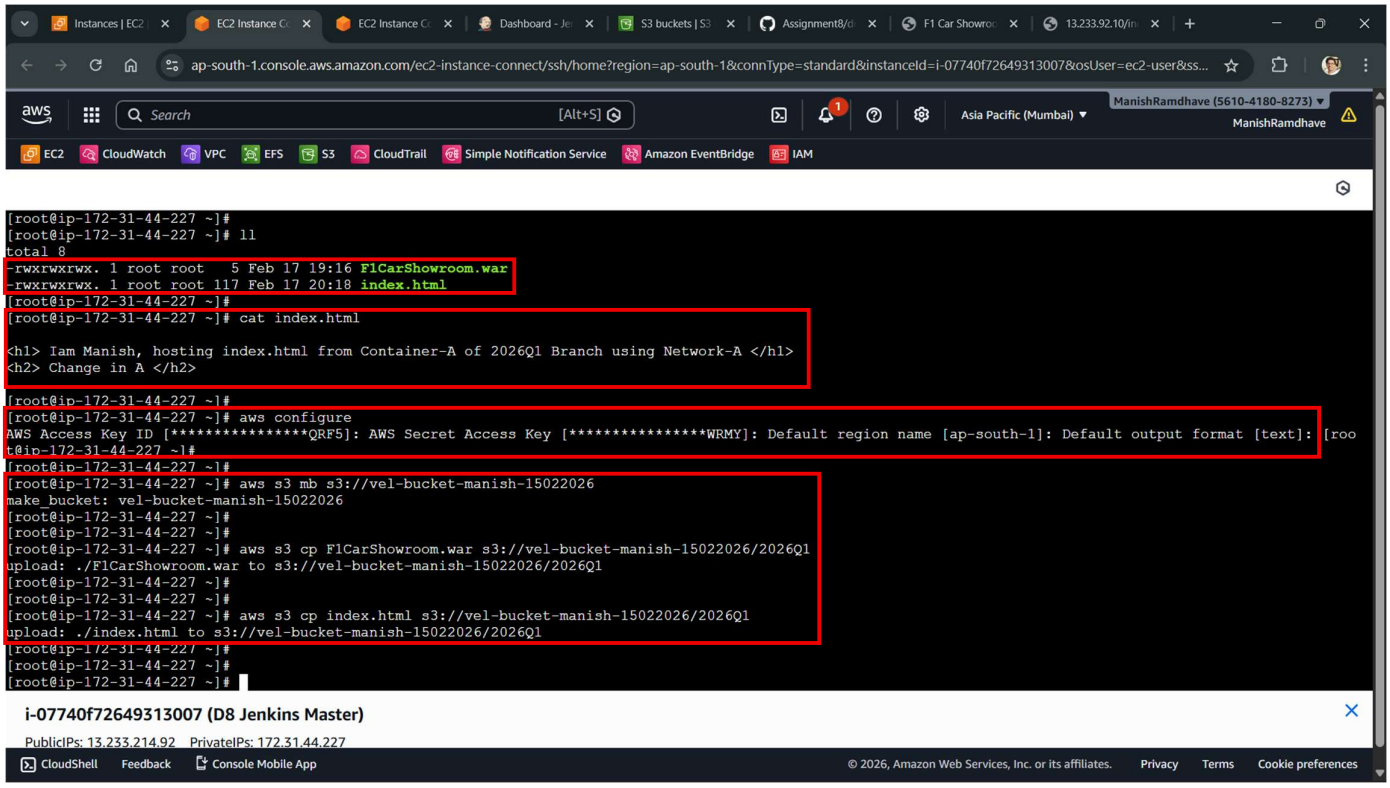
Step 12: Added AWS Configurations (i.e. **Access Key, Secret Key** etc.) in the **'Access.sh'** file which run by Jenkinsfile for **AWS CLI command processes**:



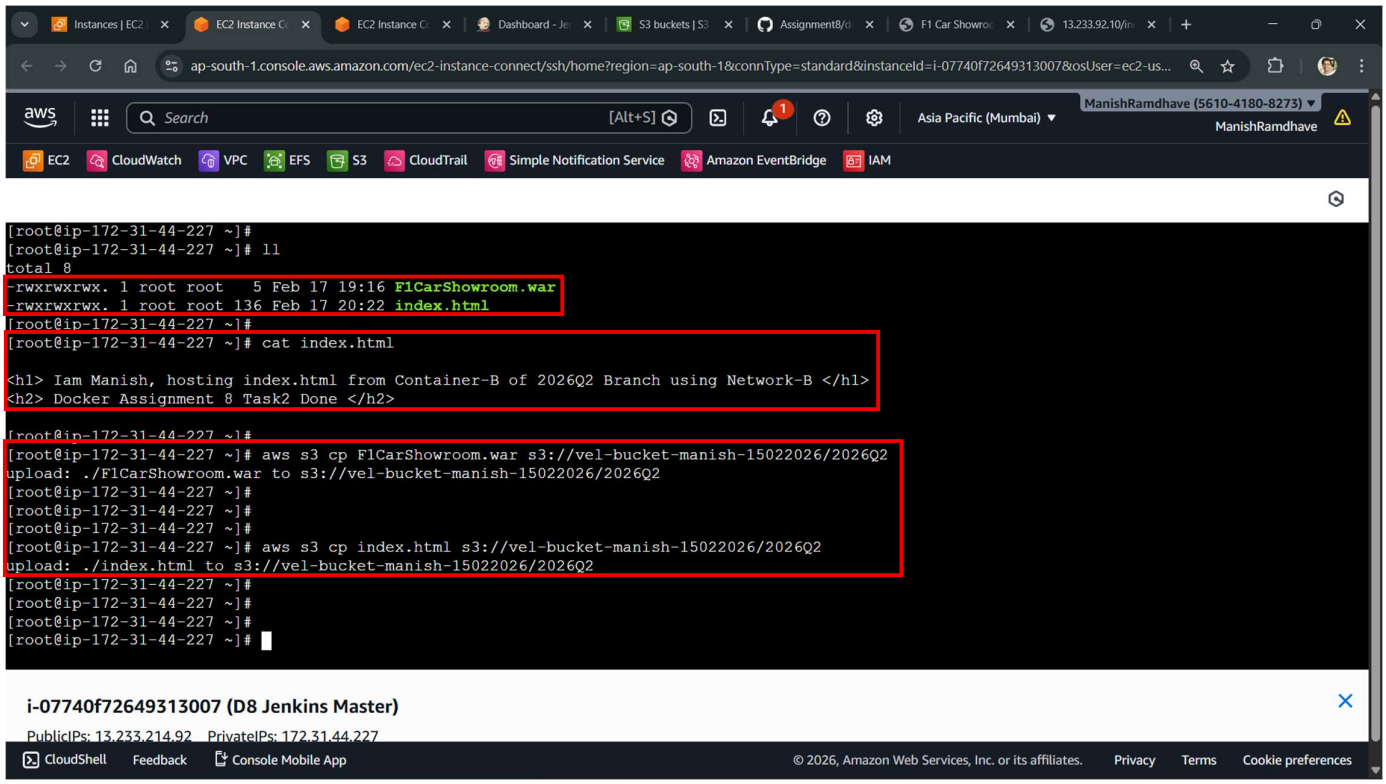
Step 13: Added **'docker-compose.yaml'** file which will run by Jenkinsfile for Container creation in each branch:



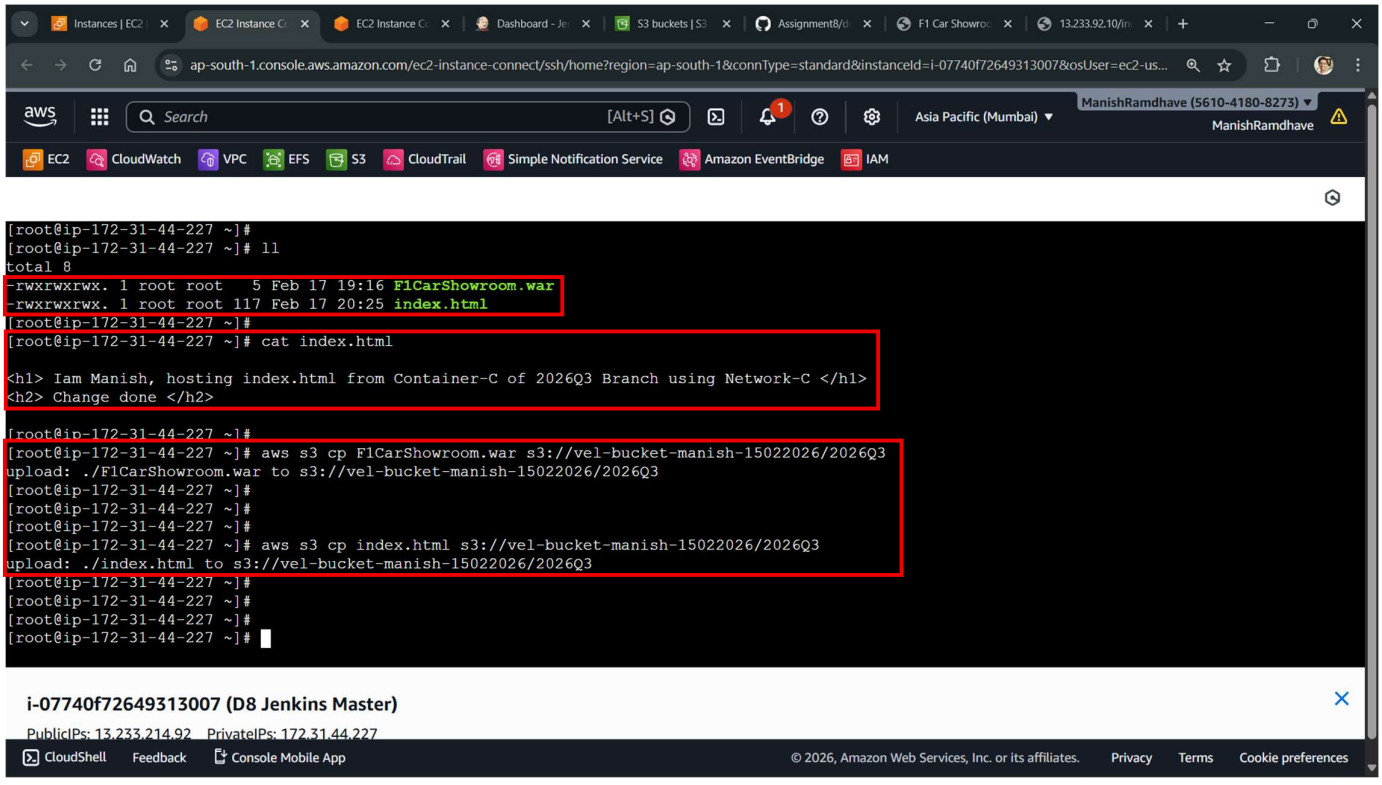
Step 14: By using the AWS CLI commands, we have uploaded the **'F1CarShowroom.war' and 'index.html' files** in the **'vel-bucket-manish15022026/2026Q1' S3 Bucket folder:**



Step 15: By using the AWS CLI commands, we have uploaded the **'F1CarShowroom.war' and 'index.html' files** in the **'vel-bucket-manish15022026/2026Q1' S3 Bucket folder:**

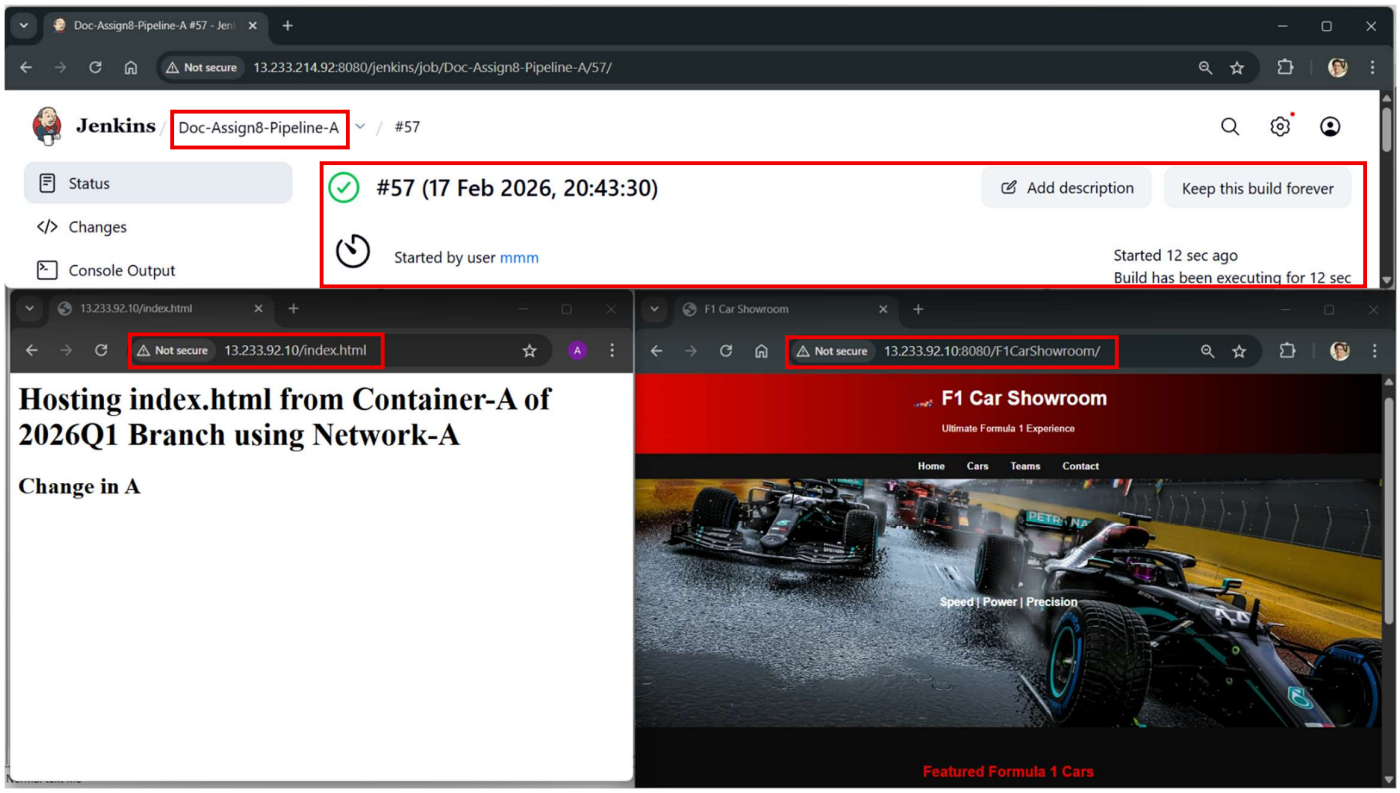


Step 16: By using the AWS CLI commands, we have uploaded the **'F1CarShowroom.war' and 'index.html' files** in the **'vel-bucket-manish15022026/2026Q1' S3 Bucket folder:**

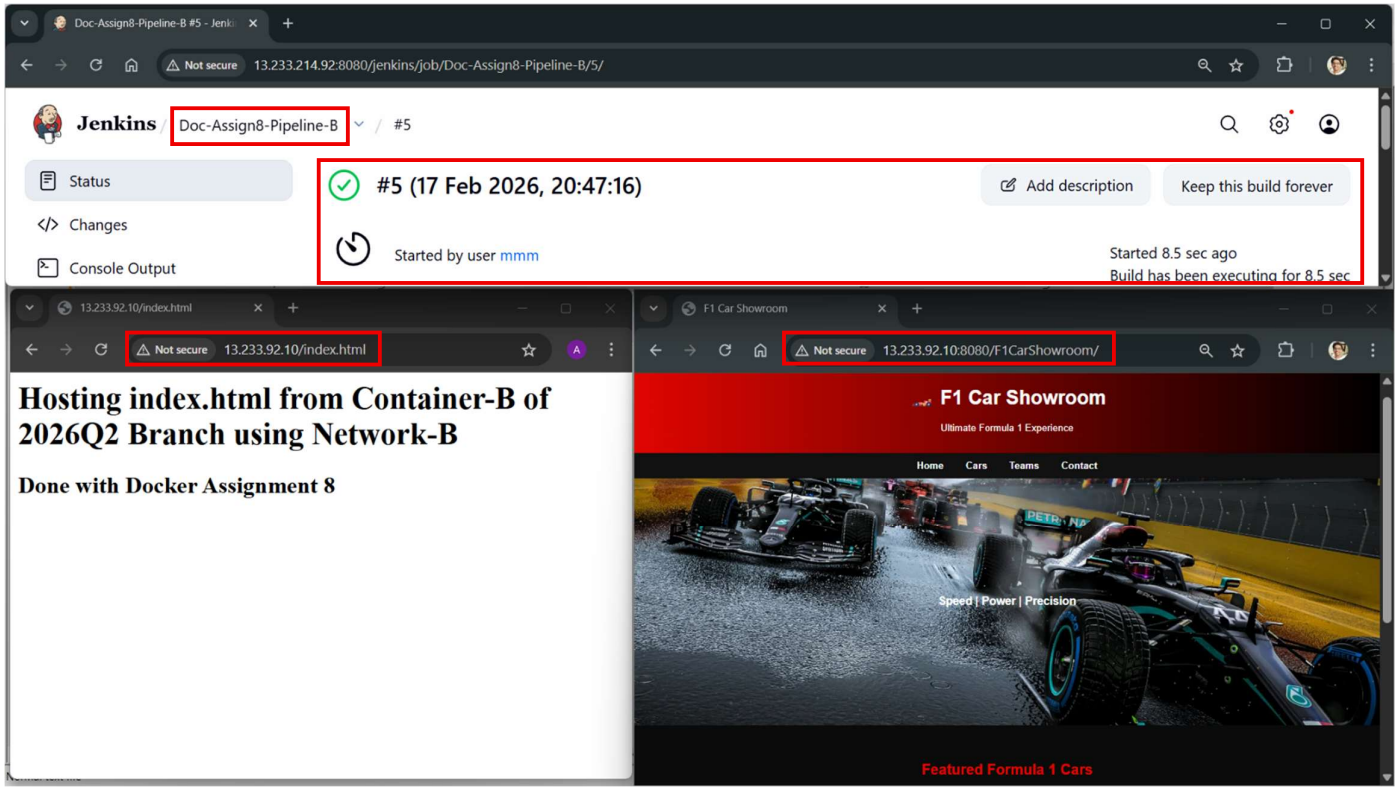


Results:

1. When **Build** is done in by **'Doc-Assign8-Pipeline-A',** the **'F1CarShowroom' application and 'index.html'** file is hosted from the **container** of **'Network-A'** by following the **'docker-compose.yaml'** script in the **GitHub Repository** of the **2026Q1 Branch** by using **Port No.80**:



2. When **Build** is done in by **'Doc-Assign8-Pipeline-B',** the **'F1CarShowroom' application and 'index.html'** file is hosted from the **container** of **'Network-B'** by following the **'docker-compose.yaml'** script in the **GitHub Repository** of the **2026Q2 Branch** by using **Port No.80**:



3. When **Build** is done in by **'Doc-Assign8-Pipeline-C',** the **'F1CarShowroom' application and 'index.html'** file is hosted from the **container** of **'Network-A'** by following the **'docker-compose.yaml'** script in the **GitHub Repository** of the **2026Q3 Branch** by using **Port No.80**: