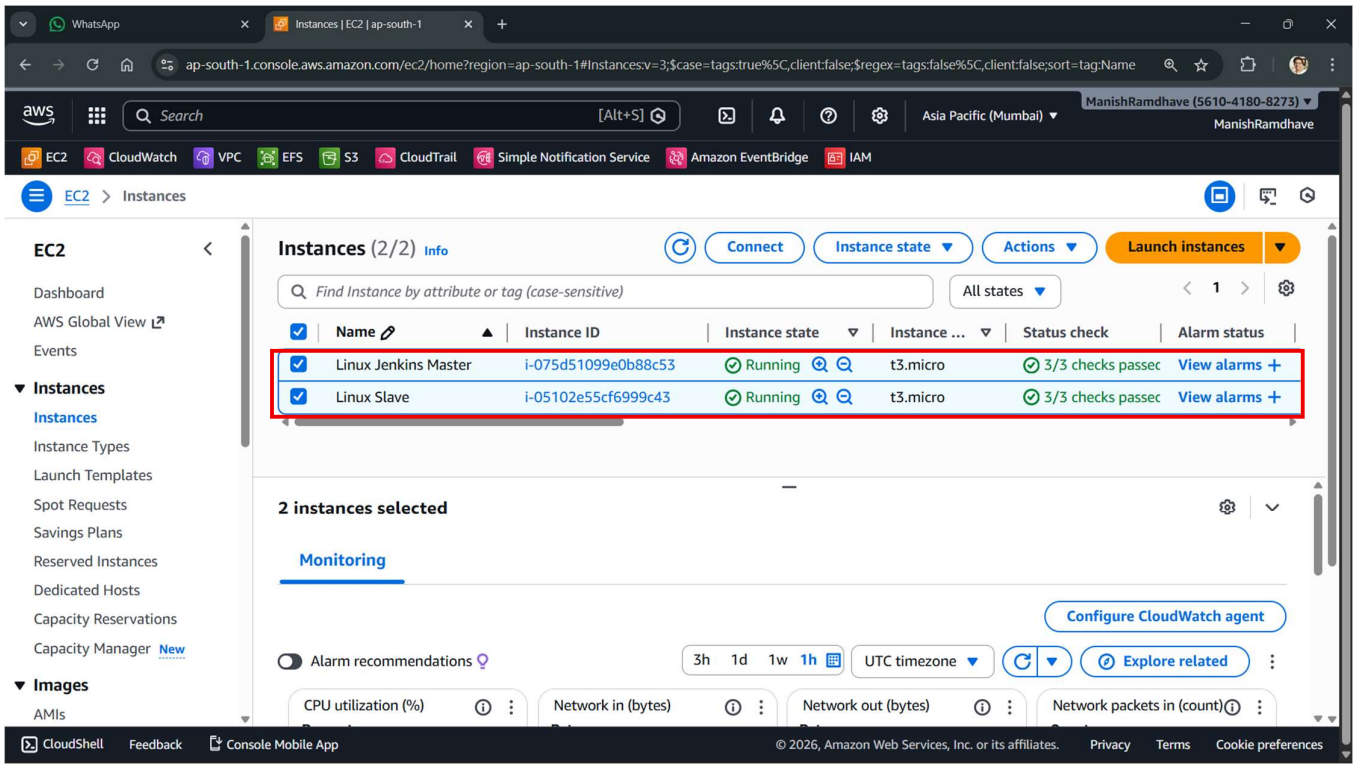
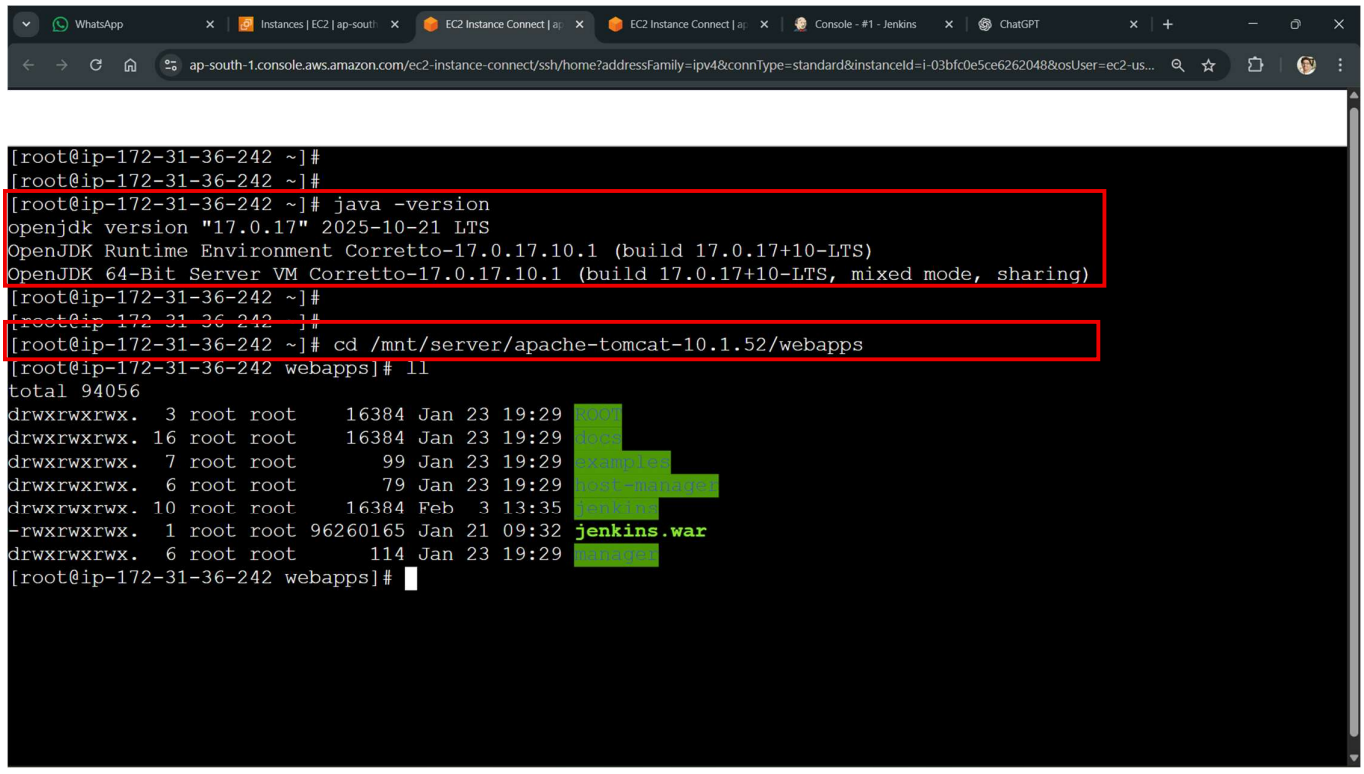
Jenkins Assignment 5

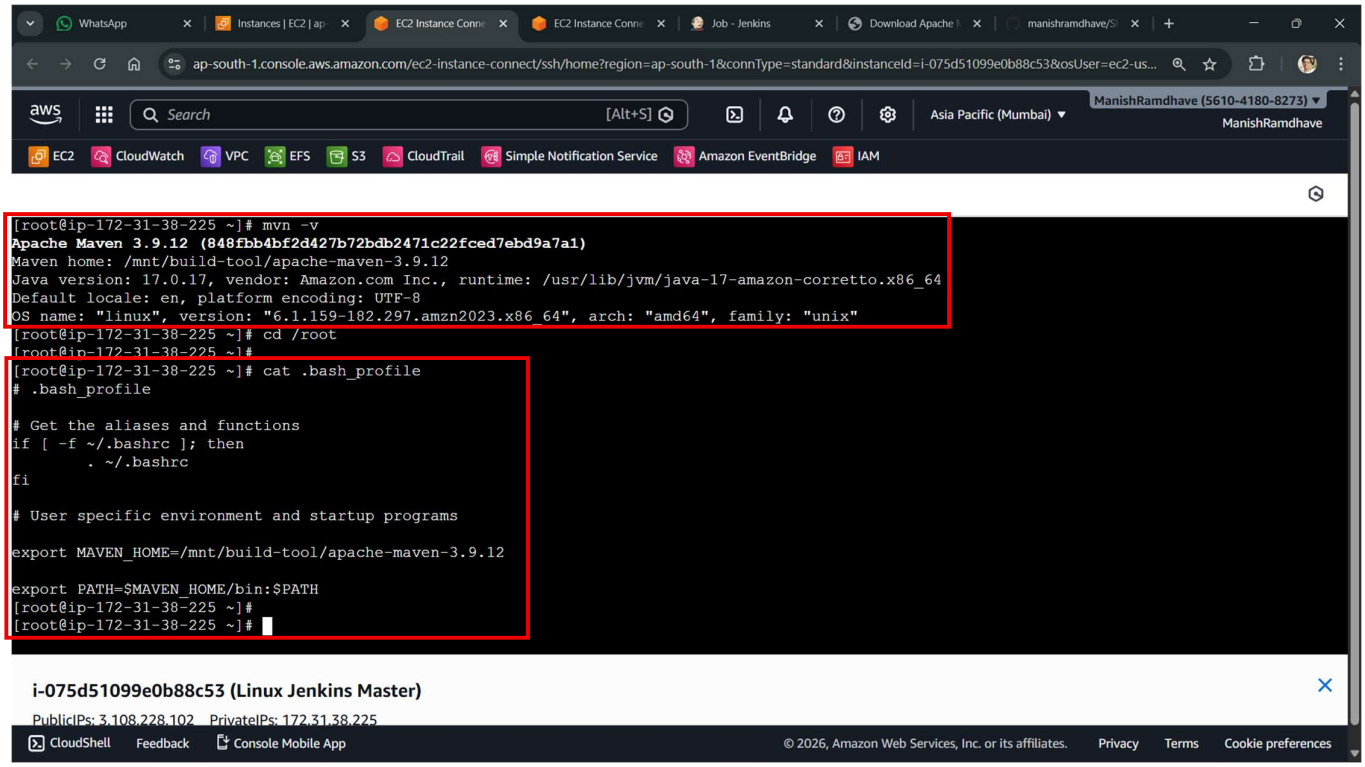
Step 1: Launched two instances, Jenkins Master and Slave1



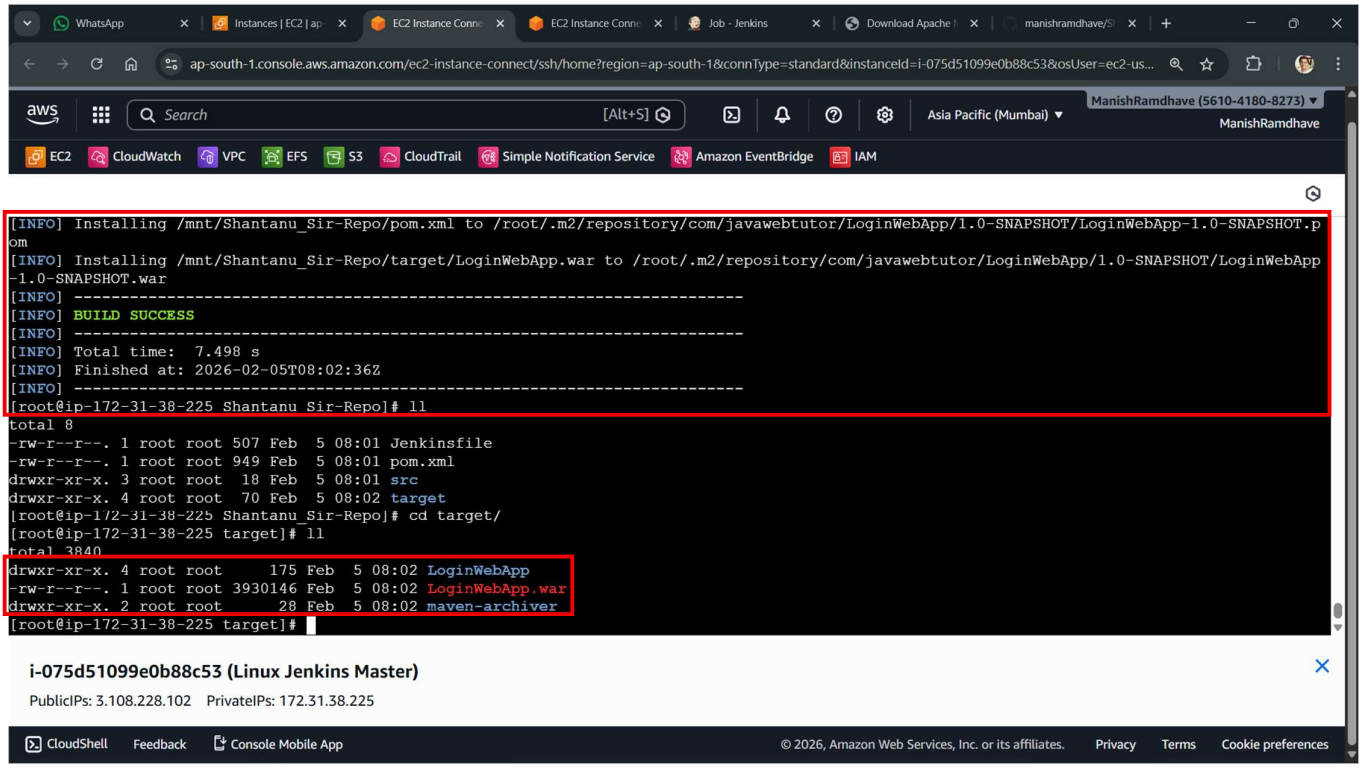
Step 2: Installed Java -17 and Apache-Tomcat 10 **on both the instance**:



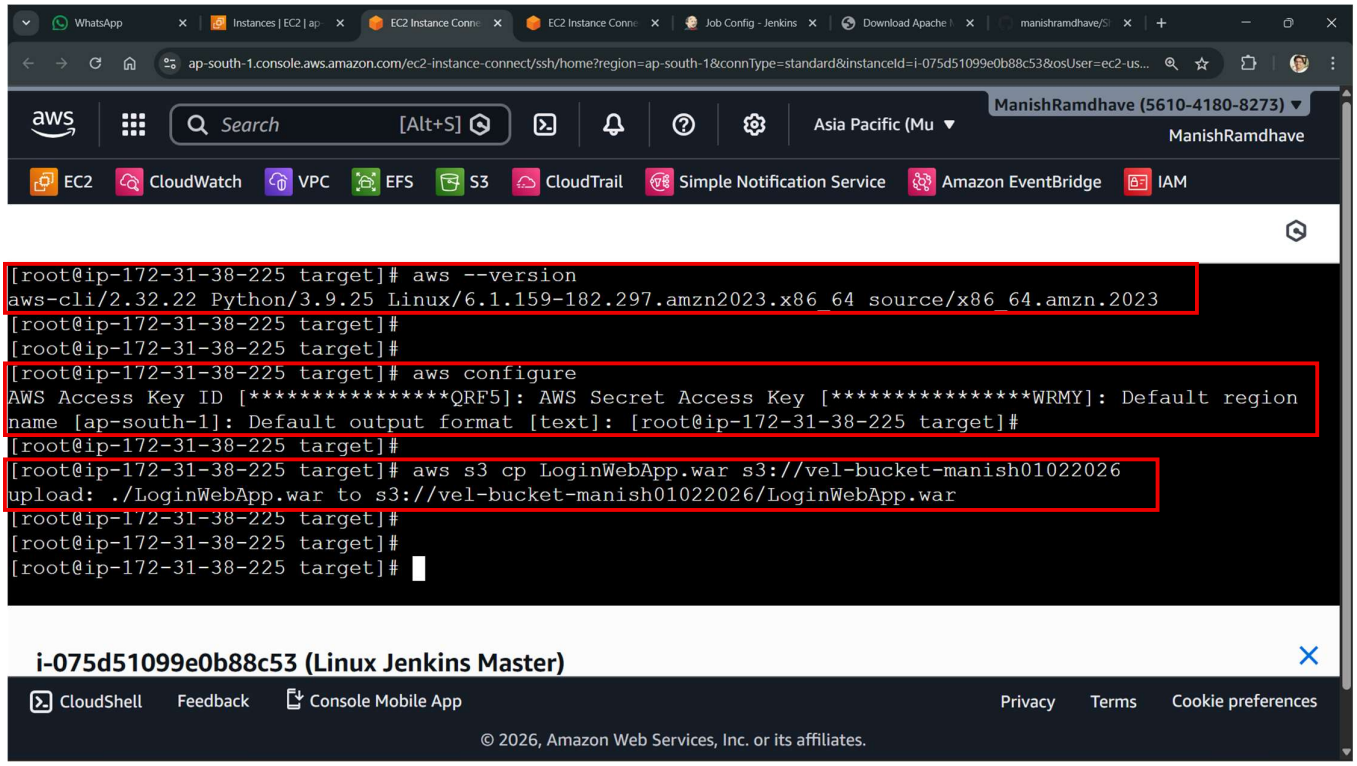
Step 3: Installed Step 2: Installed Java -17 and set an **environmental variable and its path too**:



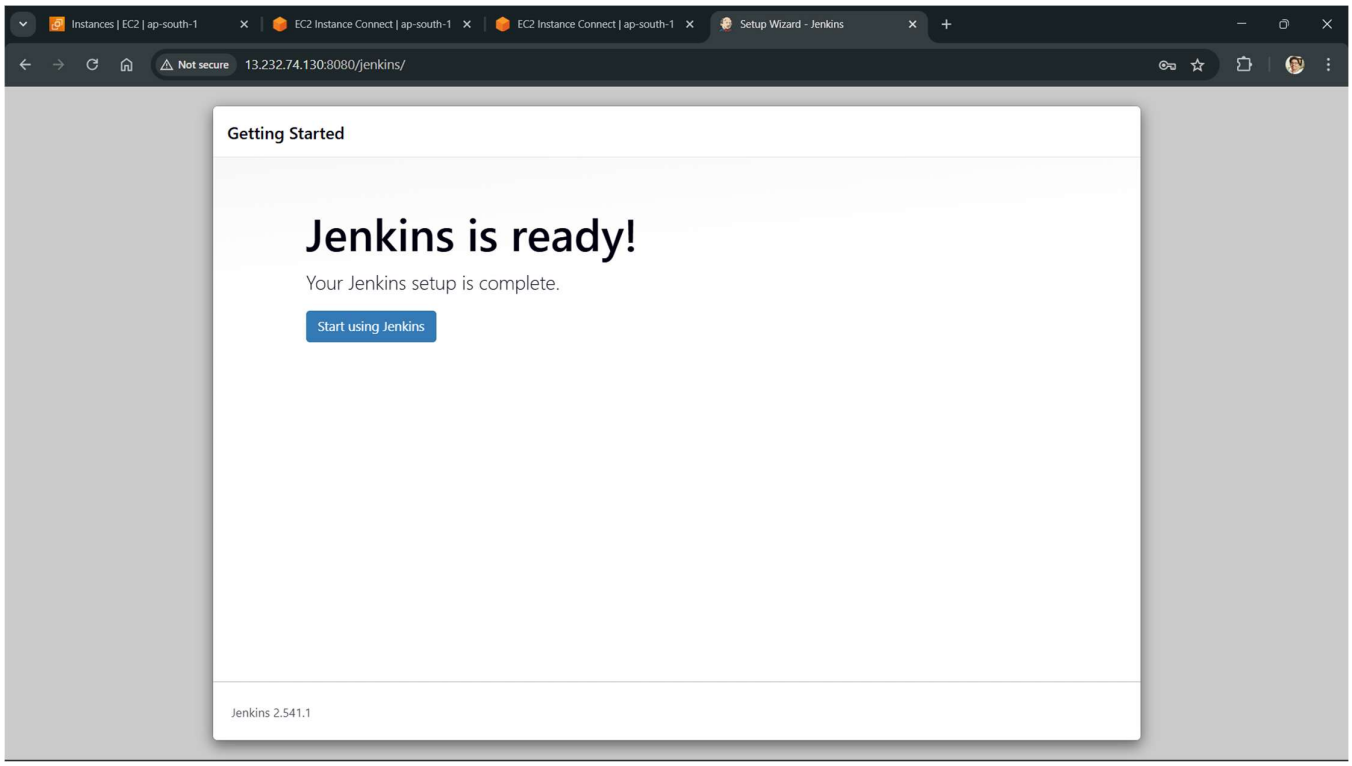
Step 4: Cloned the forked repository and installed all dependancies to download the **'LogiWebApp.war'** application file:



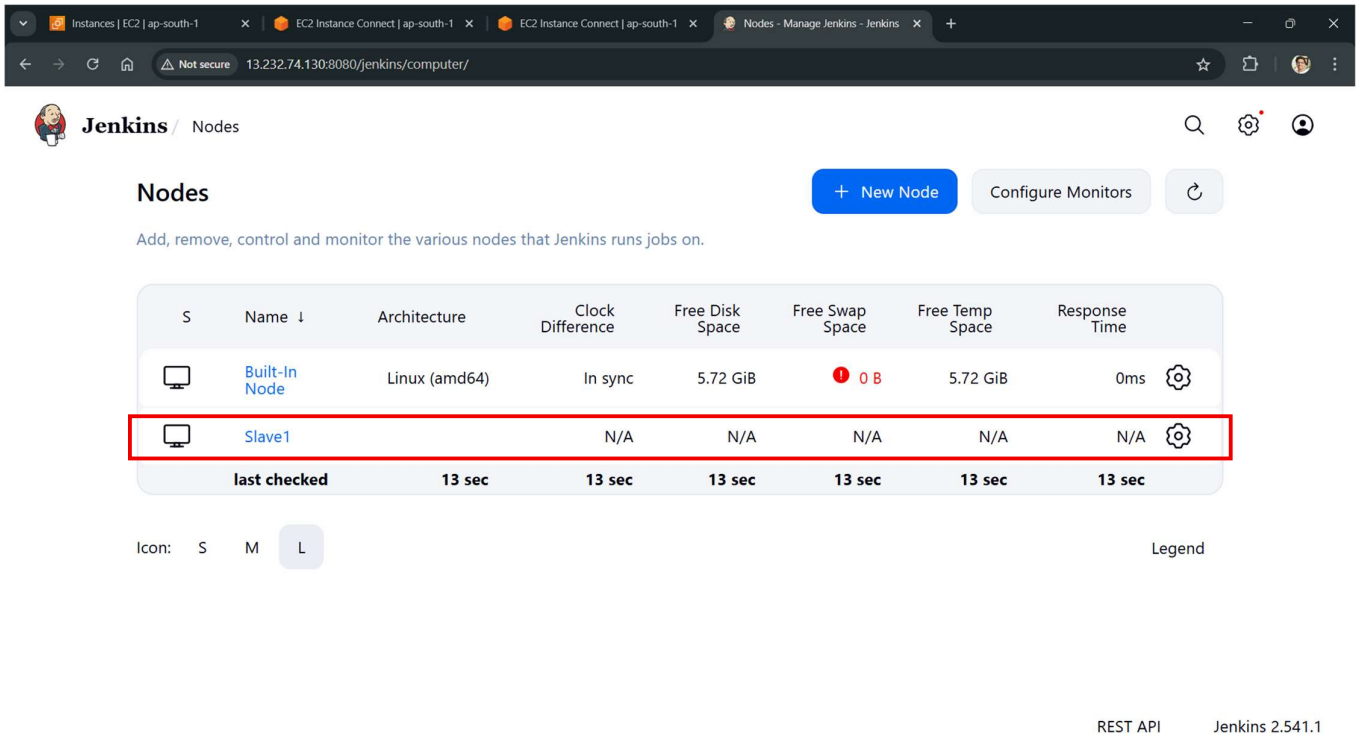
Step 5: On **'Jenkins Master',** the **'LoginWebApp.war'** file is uploaded by using the AWS CLI commands. First, we have configured the AWS with Access and Secret Keys. Then we have created a bucket named **'vel-bucket-manish01022026'** and uploaded the application .war file into the bucket:



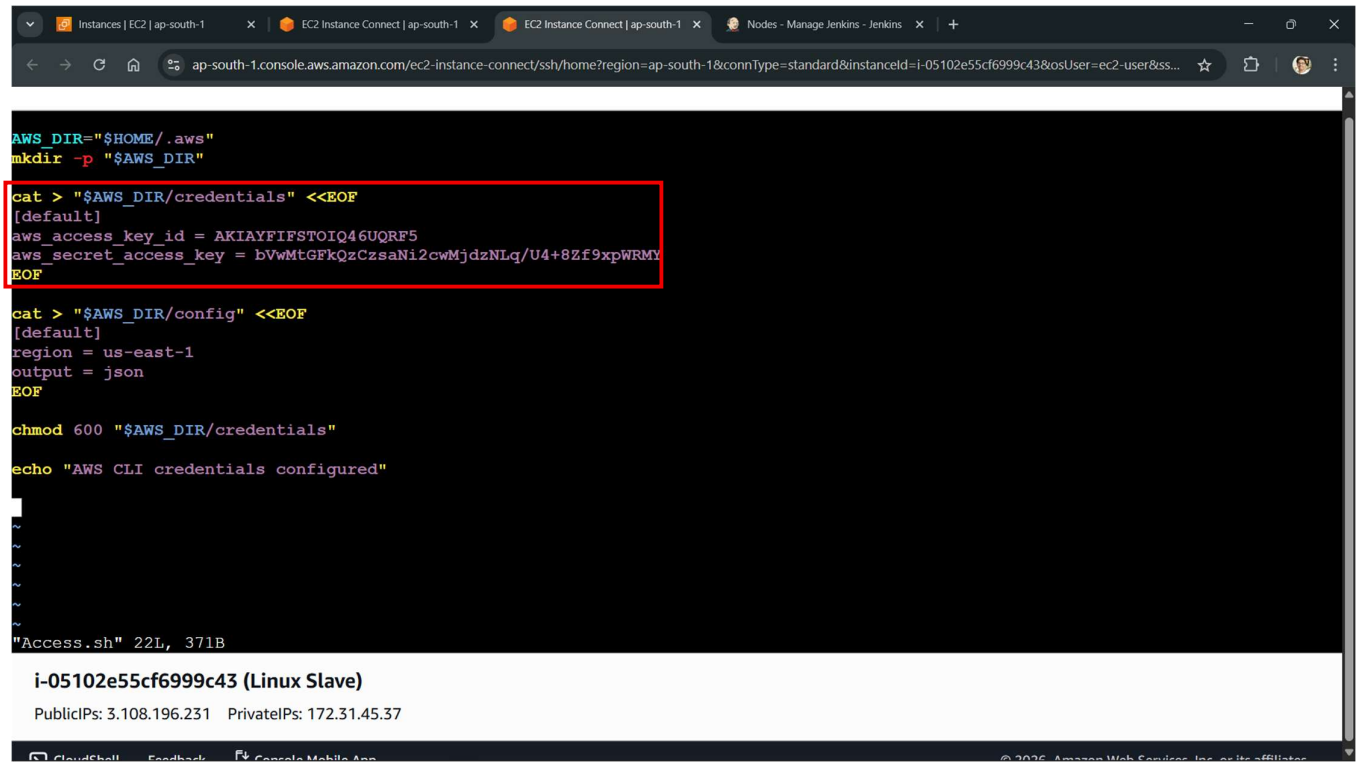
Step 6: Installed Jenkins on the **Jenkins Master Instance**:



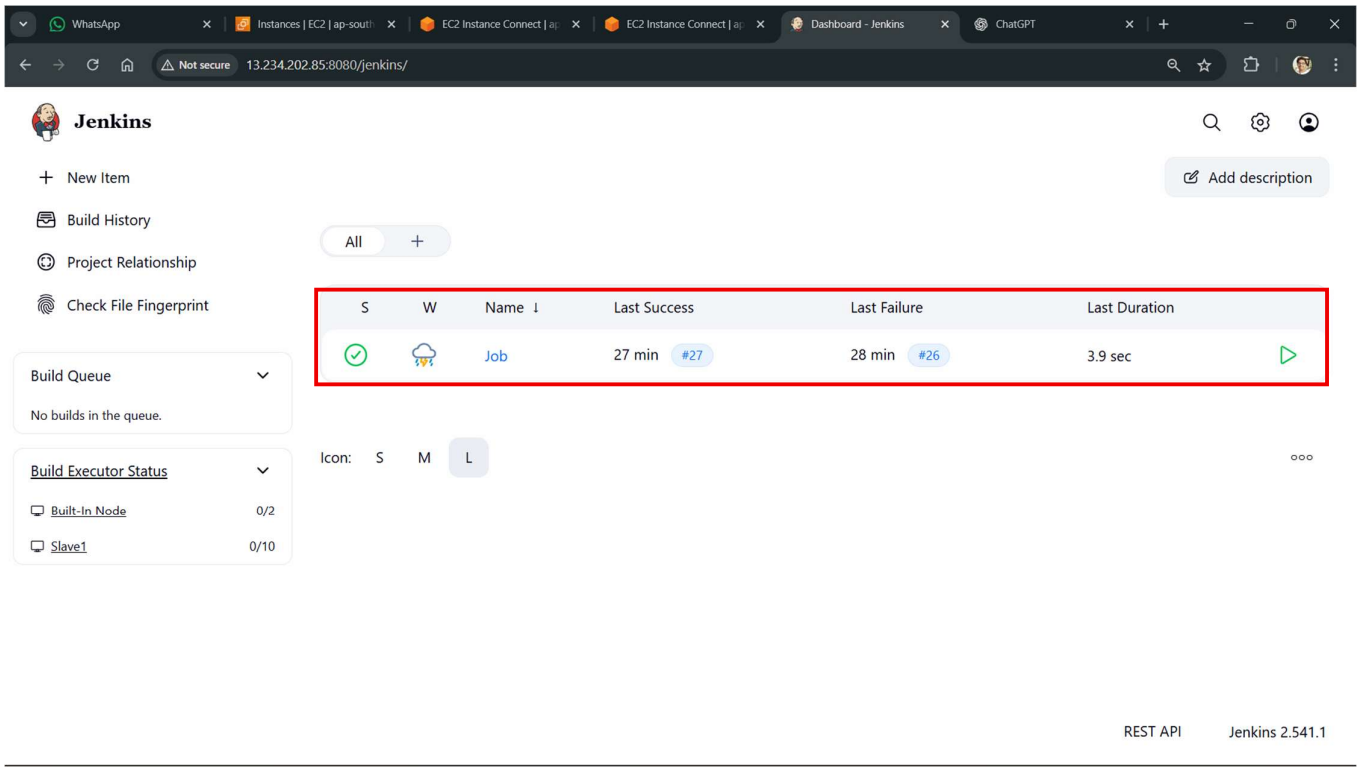
Step 7: Created a Node for to establish the connection between Jenkins Master and Slave Instance using the **'Credential'** and **'Manually trusted key Verification Strategy'**:



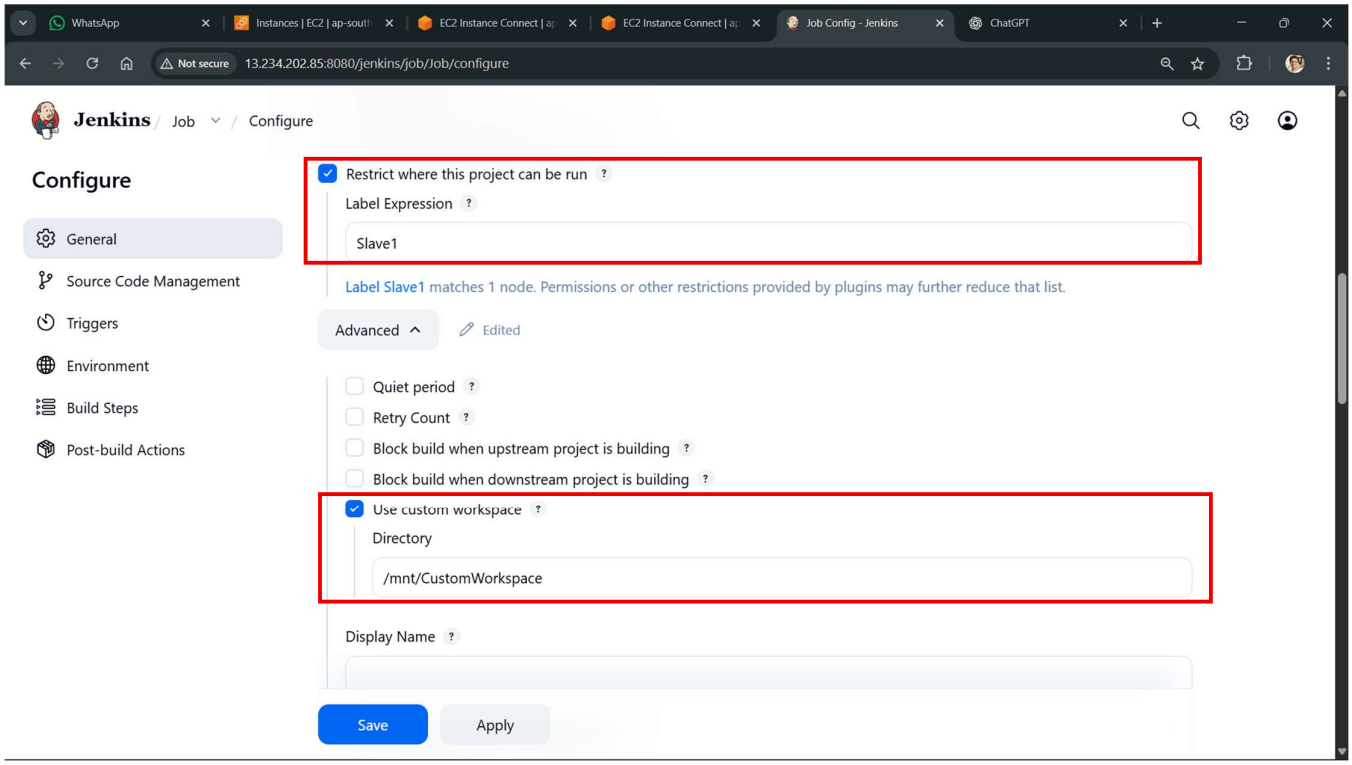
Step 8: Created an **'Access.sh'** shell script file for AWS Configuration **on Slave1 instance** for Authentication purpose. This script will be directly run by Jenkins 'Job':



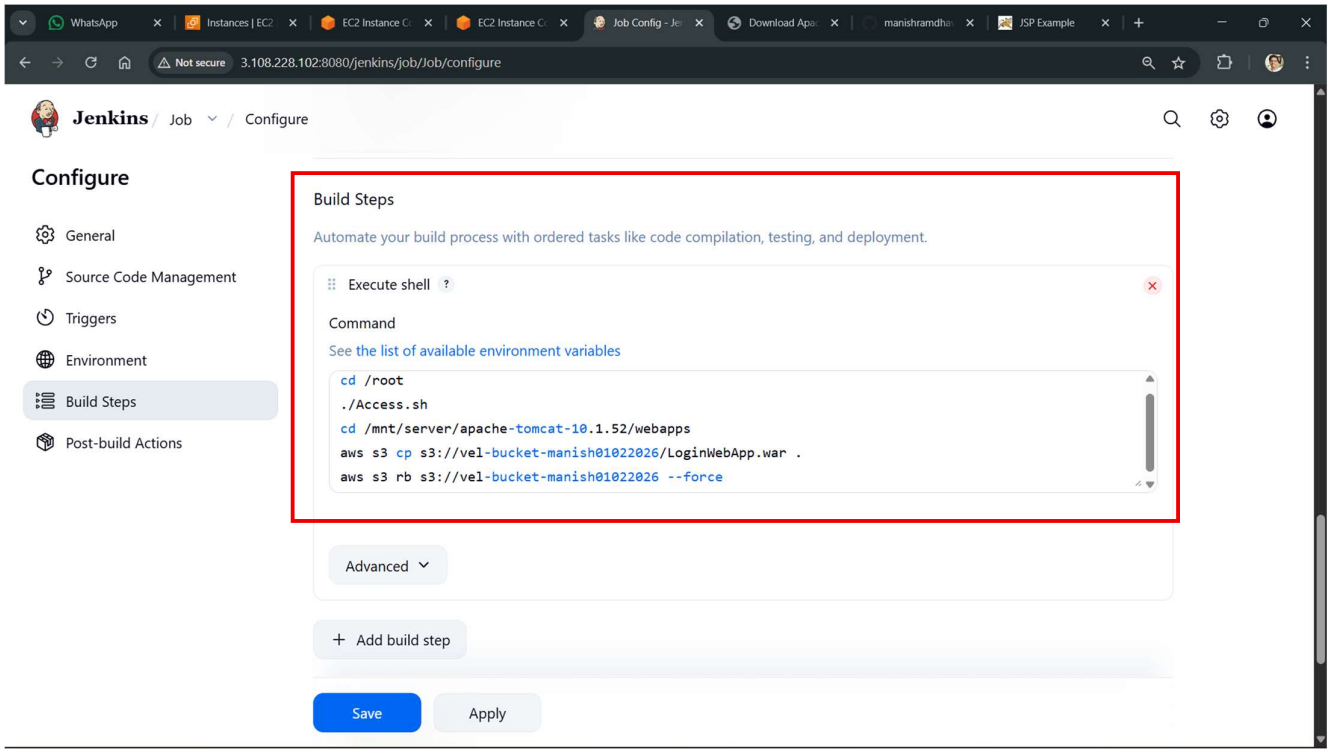
Step 9: Launched the Jenkins and created a Freestyle Job:



Step 10: Enabled **'Restrict where this project can be run'** on only Slave1 Instance and **used a custom workspace** to **'/mnt/CustomWorkspace/'**:

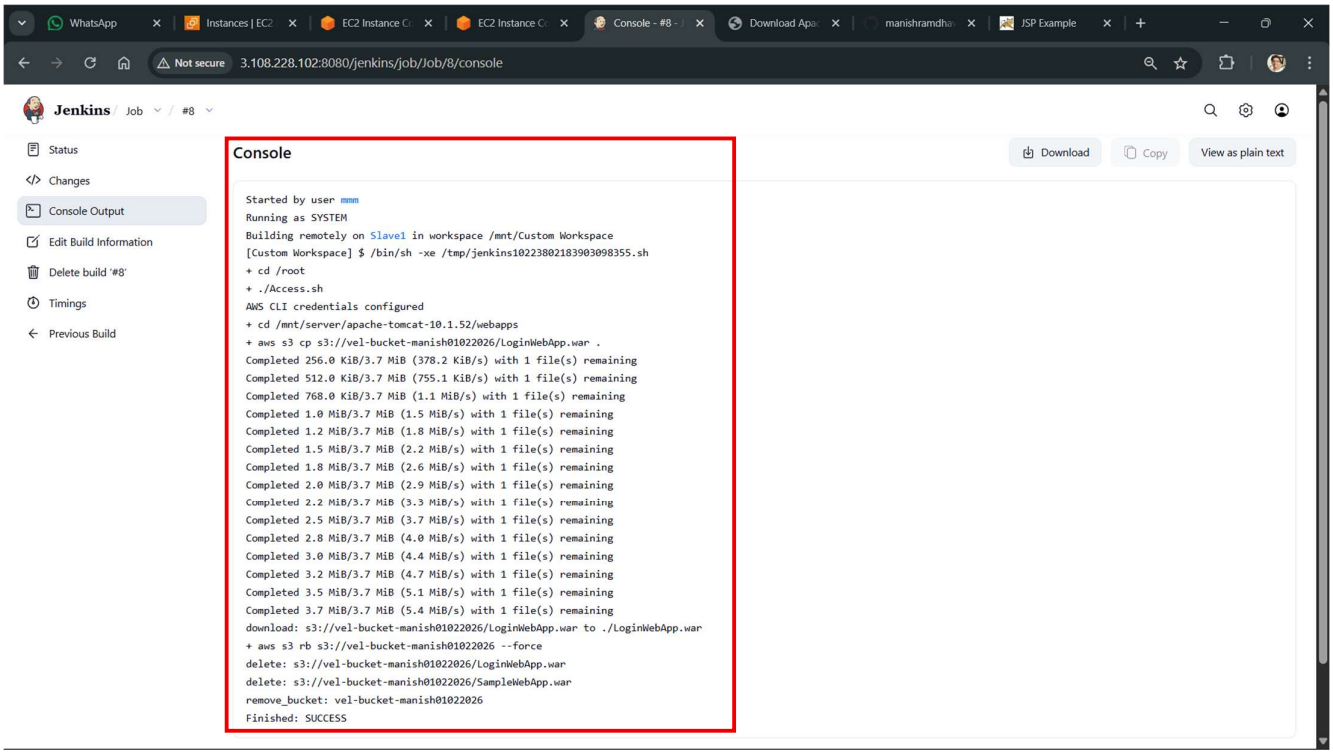


Step 11: Executed a shell script to run AWS CLI commands to run only on the Slave instance. It will first run the 'Access.sh' shell script for AWS configuration and Authentication and then, it will download the 'LoginWebApp.war to the Tomcat-10-Server and after that, it will delete the same bucket forcefully:



Result:

1. When Build is done by 'Job', our 'LoginWebApp.war' file is first downloaded from the S3 Bucket (vel-bucket-manish01022026) and deployed on Tomcat Server of a **Slave1 instance:**



2. Our **LoginWebApp** Application running successfully on **Tomcat Server**: