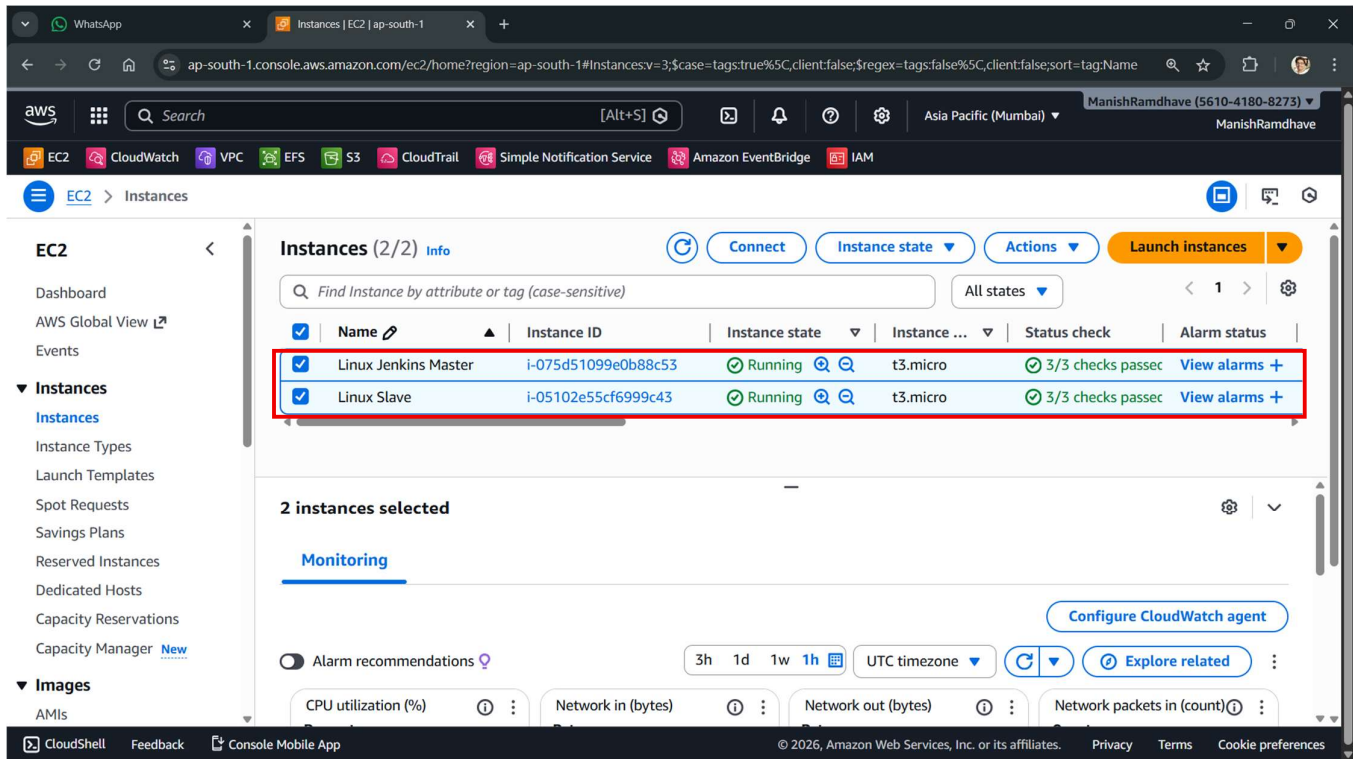
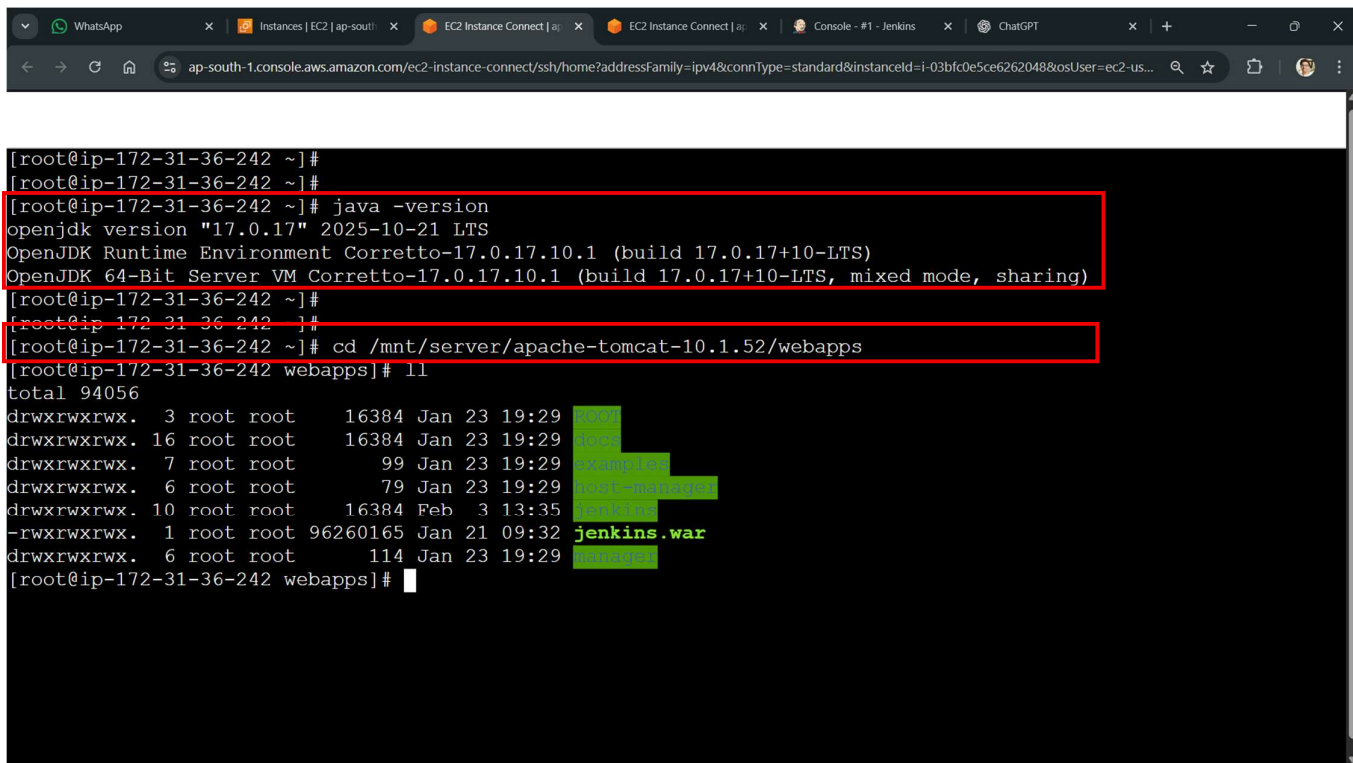


Jenkins Assignment 5

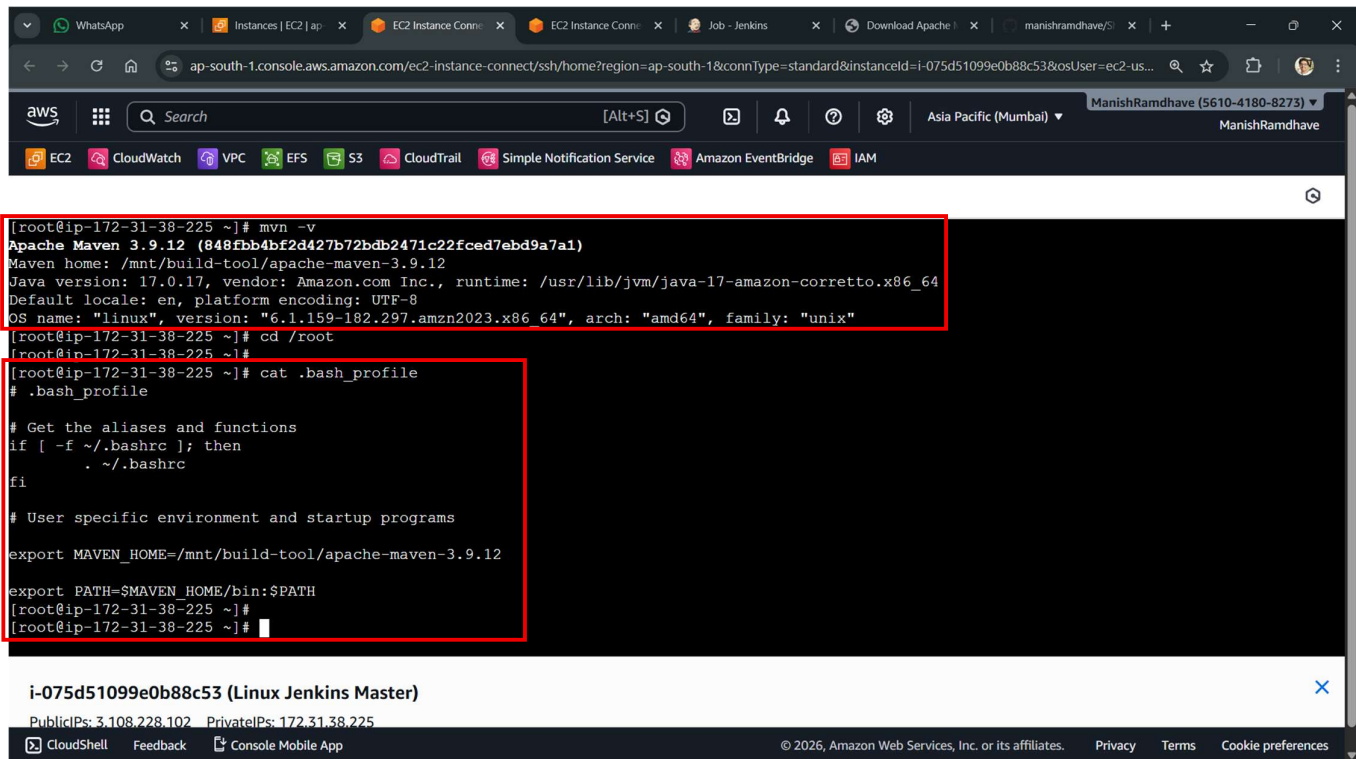
Step 1: Launched two instances, Jenkins Master and Slave1



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:



```
[root@ip-172-31-38-225 ~]# mvn -v
Apache Maven 3.9.12 (848fbb4bf2d427b72bdb2471c22fcd7ebd9a7a1)
Maven home: /mnt/build-tool/apache-maven-3.9.12
Java version: 17.0.17, vendor: Amazon.com Inc., runtime: /usr/lib/jvm/java-17-amazon-corretto.x86_64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.1.159-182.297.amzn2023.x86_64", arch: "amd64", family: "unix"
[root@ip-172-31-38-225 ~]# cd /root
[root@ip-172-31-38-225 ~]#
[root@ip-172-31-38-225 ~]# cat .bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

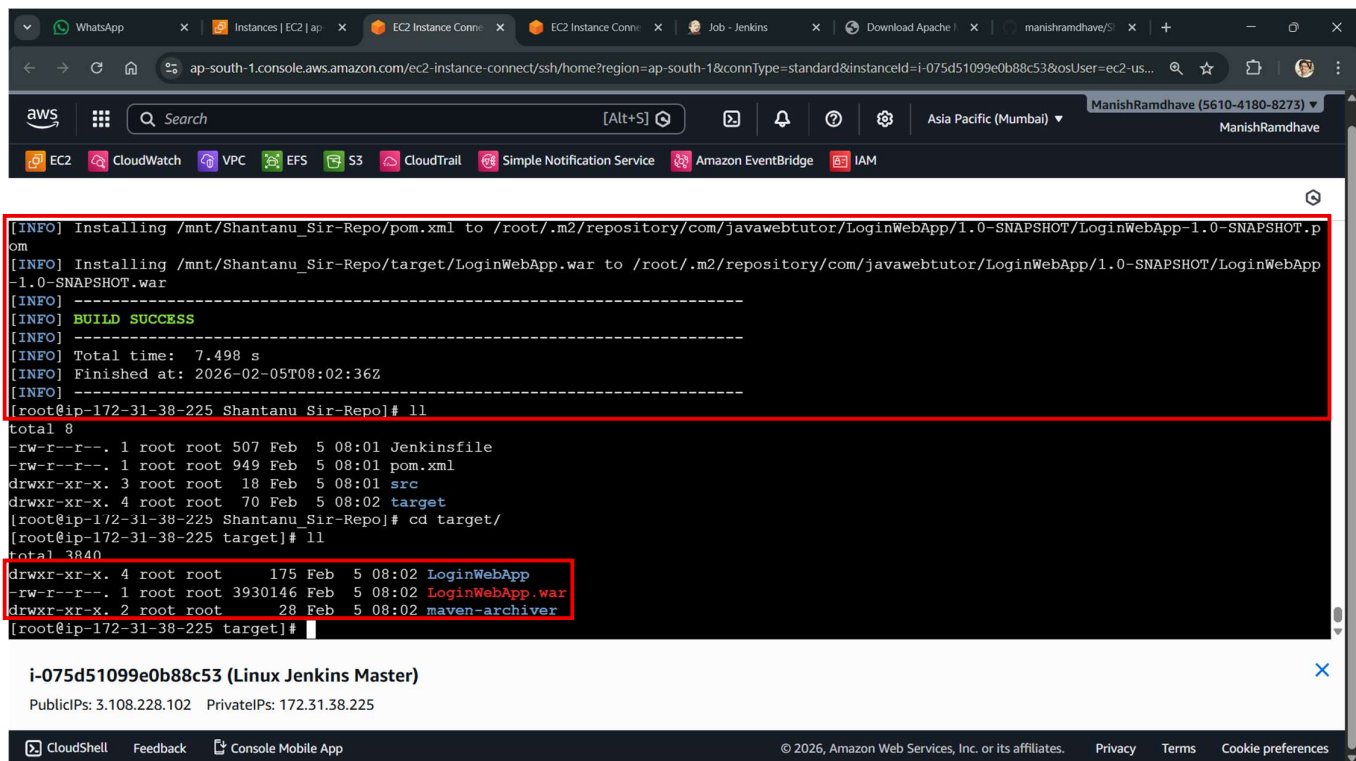
export MAVEN_HOME=/mnt/build-tool/apache-maven-3.9.12

export PATH=$MAVEN_HOME/bin:$PATH
[root@ip-172-31-38-225 ~]#
[root@ip-172-31-38-225 ~]#
```

i-075d51099e0b88c53 (Linux Jenkins Master)
PublicIPs: 3.108.228.102 PrivateIPs: 172.31.38.225

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 4: Cloned the forked repository and installed all dependencies to download the **‘LogiWebApp.war’** application file:

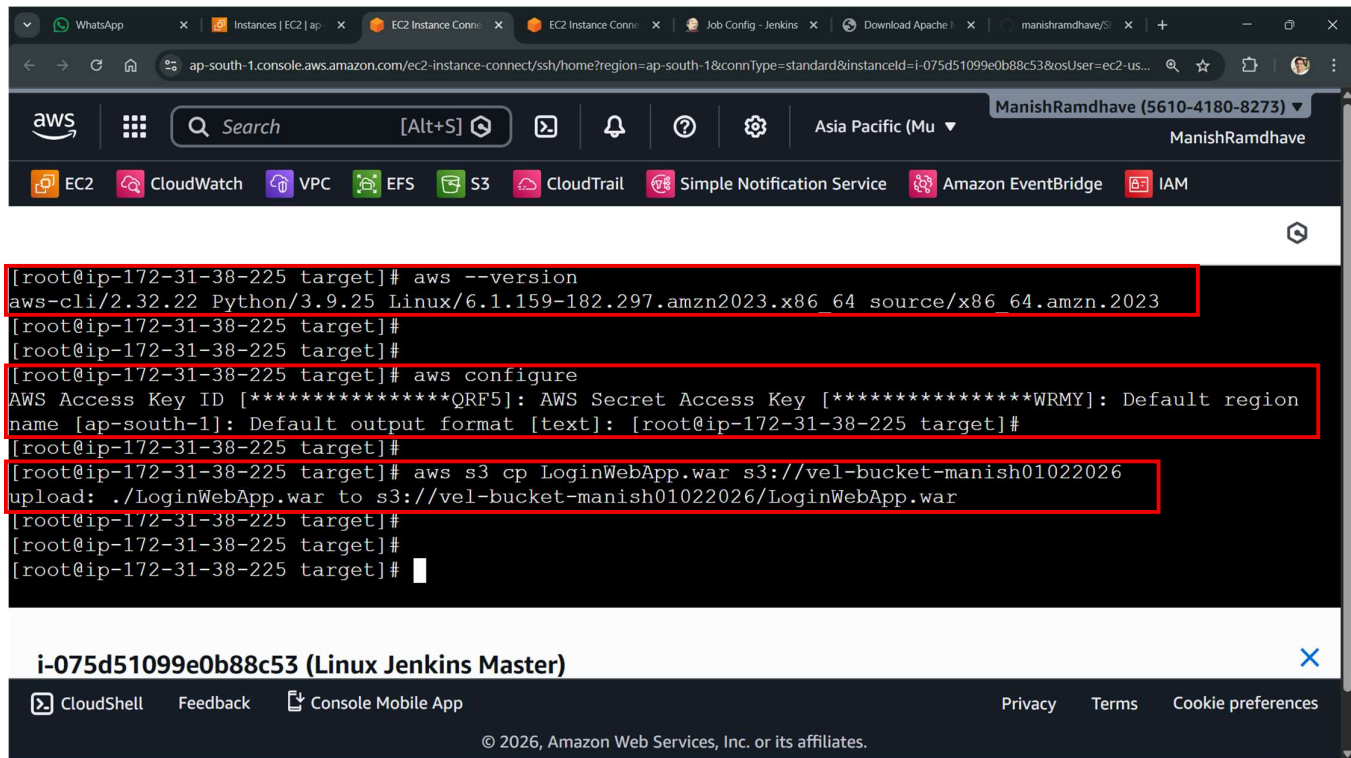


```
[INFO] Installing /mnt/Shantanu_Sir-Repo/pom.xml to /root/.m2/repository/com/javawebtutor/LoginWebApp/1.0-SNAPSHOT/LoginWebApp-1.0-SNAPSHOT.pom
[INFO] Installing /mnt/Shantanu_Sir-Repo/target/LoginWebApp.war to /root/.m2/repository/com/javawebtutor/LoginWebApp/1.0-SNAPSHOT/LoginWebApp-1.0-SNAPSHOT/LoginWebApp-1.0-SNAPSHOT.war
[INFO] BUILD SUCCESS
[INFO] Total time: 7.498 s
[INFO] Finished at: 2026-02-05T08:02:36Z
[INFO]
[root@ip-172-31-38-225 Shantanu_Sir-Repo]# ll
total 8
-rw-r--r--. 1 root root 507 Feb  5 08:01 Jenkinsfile
-rw-r--r--. 1 root root 949 Feb  5 08:01 pom.xml
drwxr-xr-x. 3 root root 18 Feb  5 08:01 src
drwxr-xr-x. 4 root root 70 Feb  5 08:02 target
[root@ip-172-31-38-225 Shantanu_Sir-Repo]# cd target/
[root@ip-172-31-38-225 target]# ll
total 3840
drwxr-xr-x. 4 root root 175 Feb  5 08:02 LoginWebApp
-rw-r--r--. 1 root root 3930146 Feb  5 08:02 LoginWebApp.war
drwxr-xr-x. 2 root root 28 Feb  5 08:02 maven-archiver
[root@ip-172-31-38-225 target]#
```

i-075d51099e0b88c53 (Linux Jenkins Master)
PublicIPs: 3.108.228.102 PrivateIPs: 172.31.38.225

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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:

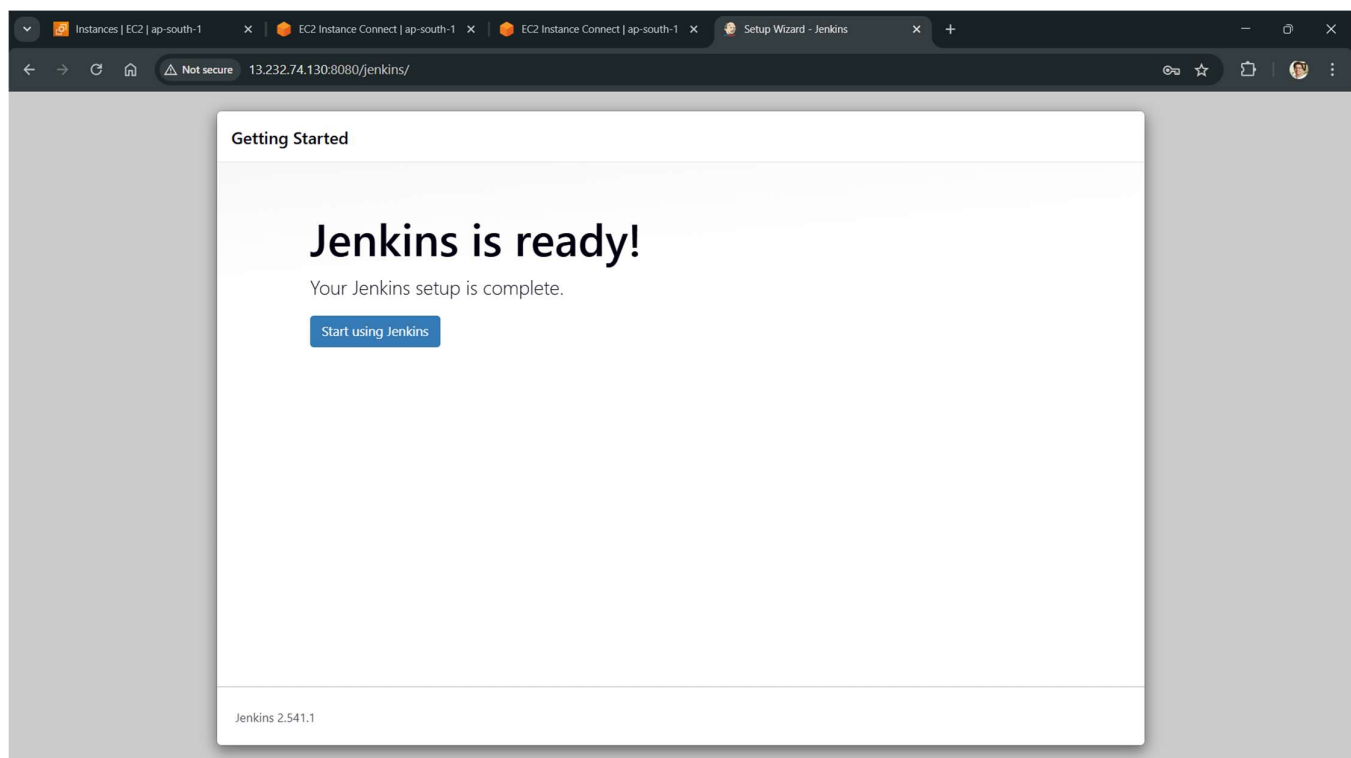


The screenshot shows a terminal window with the following commands and output:

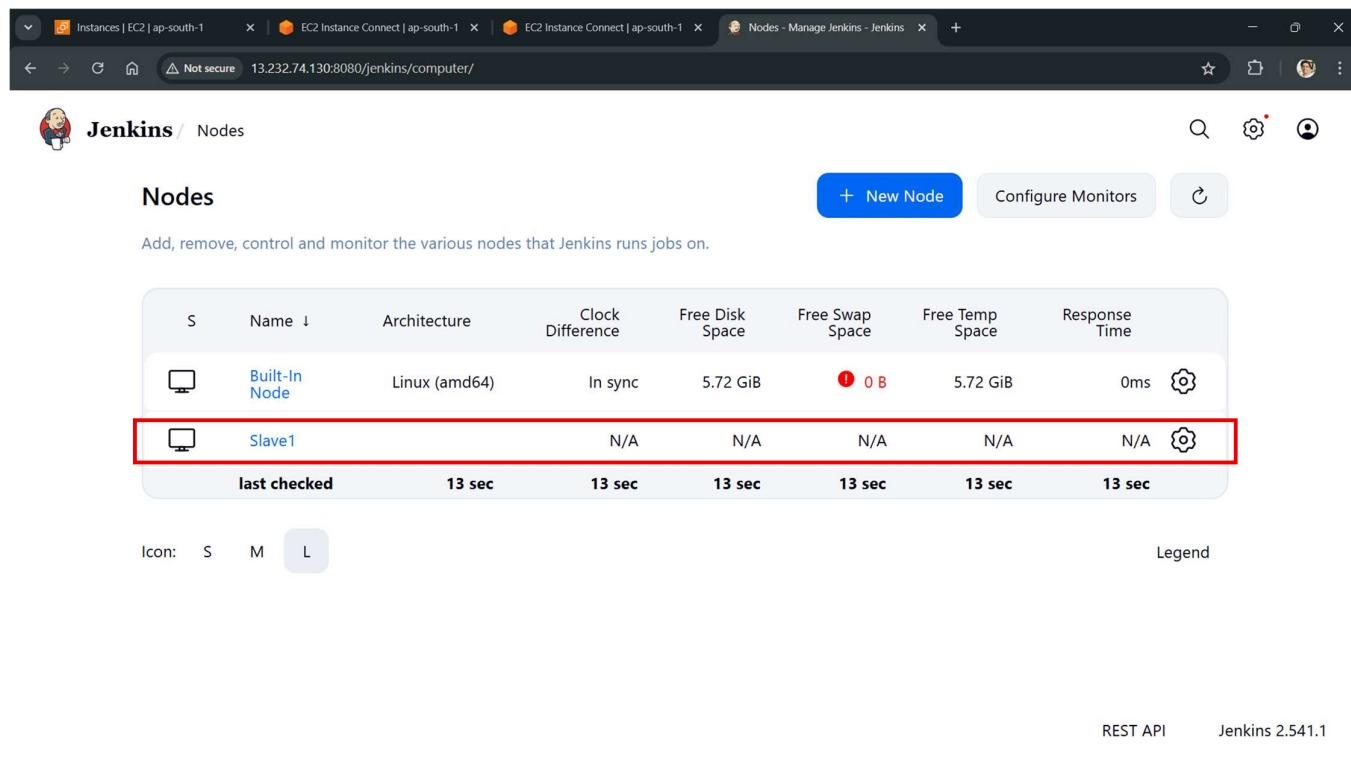
```
[root@ip-172-31-38-225 target]# aws --version
aws-cli/2.32.22 Python/3.9.25 Linux/6.1.159-182.297.amzn2023.x86_64 source/x86_64.amzn.2023
[root@ip-172-31-38-225 target]#
[root@ip-172-31-38-225 target]#
[root@ip-172-31-38-225 target]# aws configure
AWS Access Key ID [*****QRF5]: AWS Secret Access Key [*****WRMY]: Default region
name [ap-south-1]: Default output format [text]: [root@ip-172-31-38-225 target]#
[root@ip-172-31-38-225 target]#
[root@ip-172-31-38-225 target]# aws s3 cp LoginWebApp.war s3://vel-bucket-manish01022026
upload: ./LoginWebApp.war to s3://vel-bucket-manish01022026/LoginWebApp.war
[root@ip-172-31-38-225 target]#
[root@ip-172-31-38-225 target]#
[root@ip-172-31-38-225 target]#
```

Below the terminal window, the instance name **i-075d51099e0b88c53 (Linux Jenkins Master)** is displayed. At the bottom, there are links for **CloudShell**, **Feedback**, **Console Mobile App**, **Privacy**, **Terms**, and **Cookie preferences**, along with the copyright notice: © 2026, Amazon Web Services, Inc. or its affiliates.

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’:



The screenshot shows the Jenkins 'Nodes' page. At the top, there's a 'New Node' button and a 'Configure Monitors' button. Below them is a table of nodes. The 'Slave1' node is highlighted with a red box. The table has columns: S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. The 'Slave1' node has a status of 'last checked', a clock difference of '13 sec', and a response time of '13 sec'.

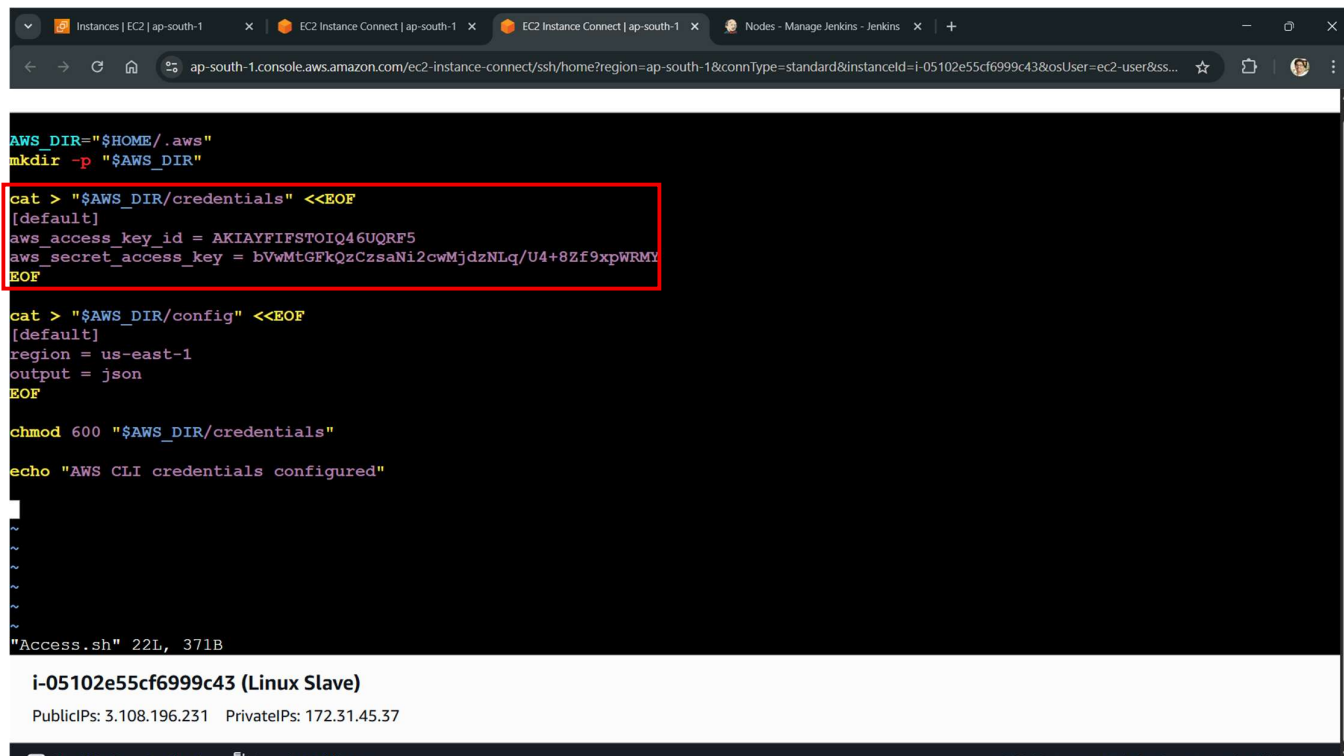
| S | Name | Architecture | Clock Difference | Free Disk Space | Free Swap Space | Free Temp Space | Response Time |
|---|---------------|---------------|------------------|-----------------|-----------------|-----------------|---------------|
| | Built-in Node | Linux (amd64) | In sync | 5.72 GiB | 0 B | 5.72 GiB | 0ms |
| | Slave1 | | N/A | N/A | N/A | N/A | N/A |

last checked 13 sec 13 sec 13 sec 13 sec 13 sec 13 sec

Icon: S M L Legend

REST API Jenkins 2.541.1

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’:



The screenshot shows a terminal window with the following commands and output:

```
AWS_DIR="$HOME/.aws"
mkdir -p "$AWS_DIR"

cat > "$AWS_DIR/credentials" <<EOF
[default]
aws_access_key_id = AKIAYFIFSTOIQ46UQRF5
aws_secret_access_key = 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
EOF

cat > "$AWS_DIR/config" <<EOF
[default]
region = us-east-1
output = json
EOF

chmod 600 "$AWS_DIR/credentials"

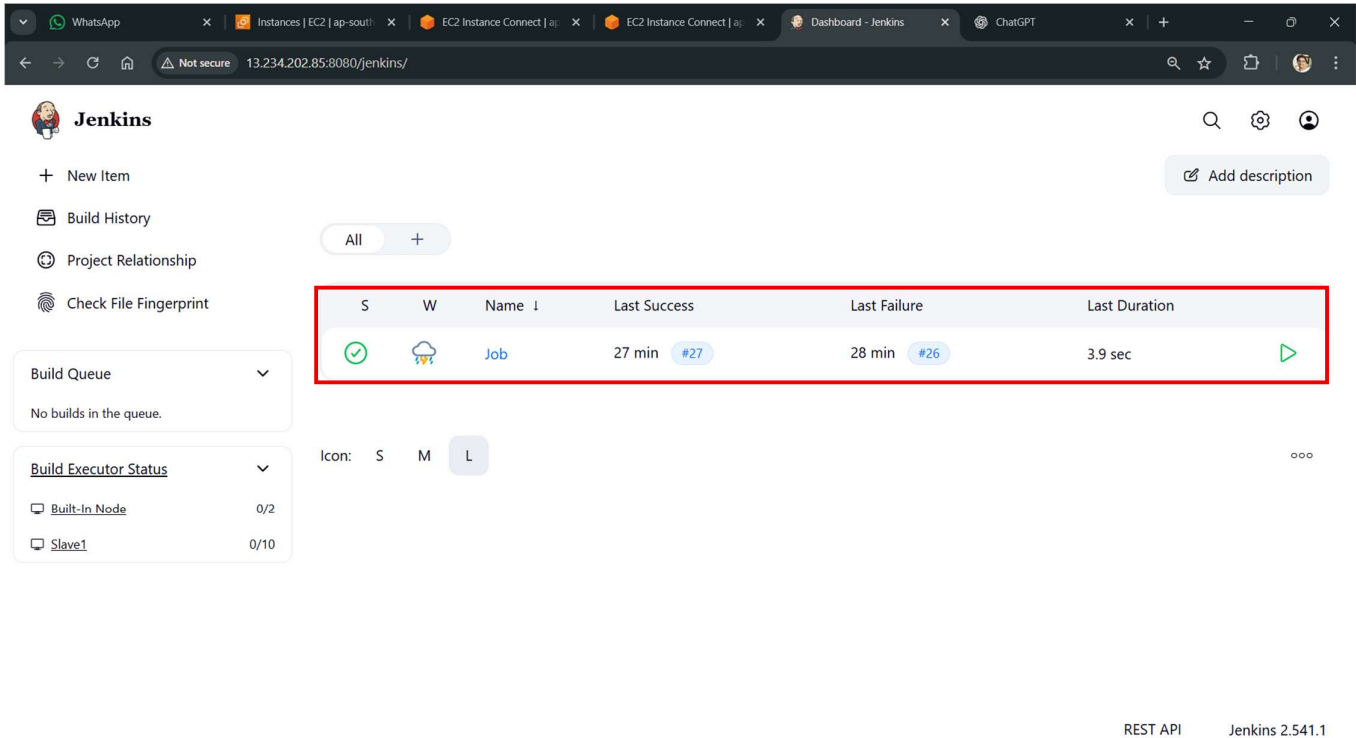
echo "AWS CLI credentials configured"
```

"Access.sh" 22L, 371B

i-05102e55cf6999c43 (Linux Slave)

PublicIPs: 3.108.196.231 PrivateIPs: 172.31.45.37

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

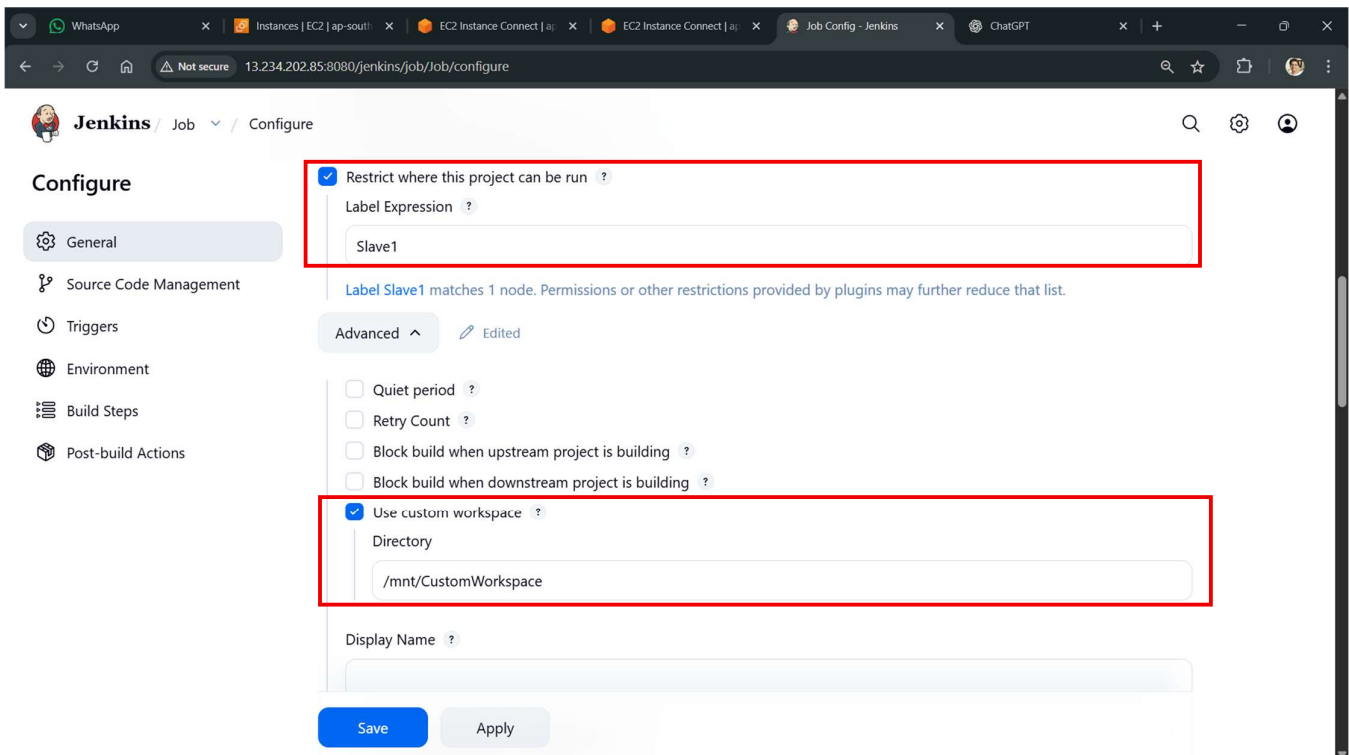


The screenshot shows the Jenkins Dashboard at the URL `13.234.202.85:8080/jenkins/`. The dashboard includes a sidebar with navigation links: New Item, Build History, Project Relationship, and Check File Fingerprint. The main area displays a table of jobs. A red box highlights the following table:

| S | W | Name ↓ | Last Success | Last Failure | Last Duration |
|---|---|--------|--------------|--------------|---------------|
| ✓ | ☁ | Job | 27 min #27 | 28 min #26 | 3.9 sec |

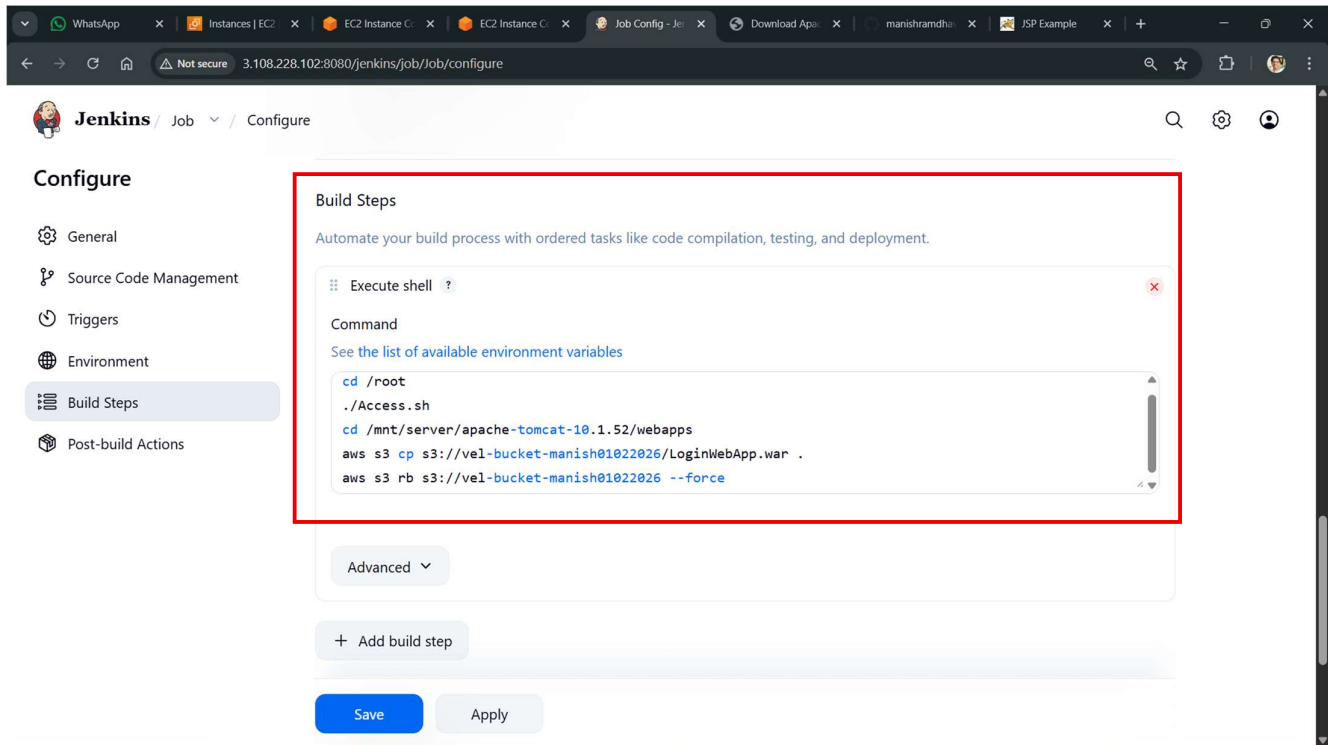
Below the table, there is a section for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (Built-In Node: 0/2, Slave1: 0/10). The bottom right corner shows 'REST API' and 'Jenkins 2.541.1'.

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



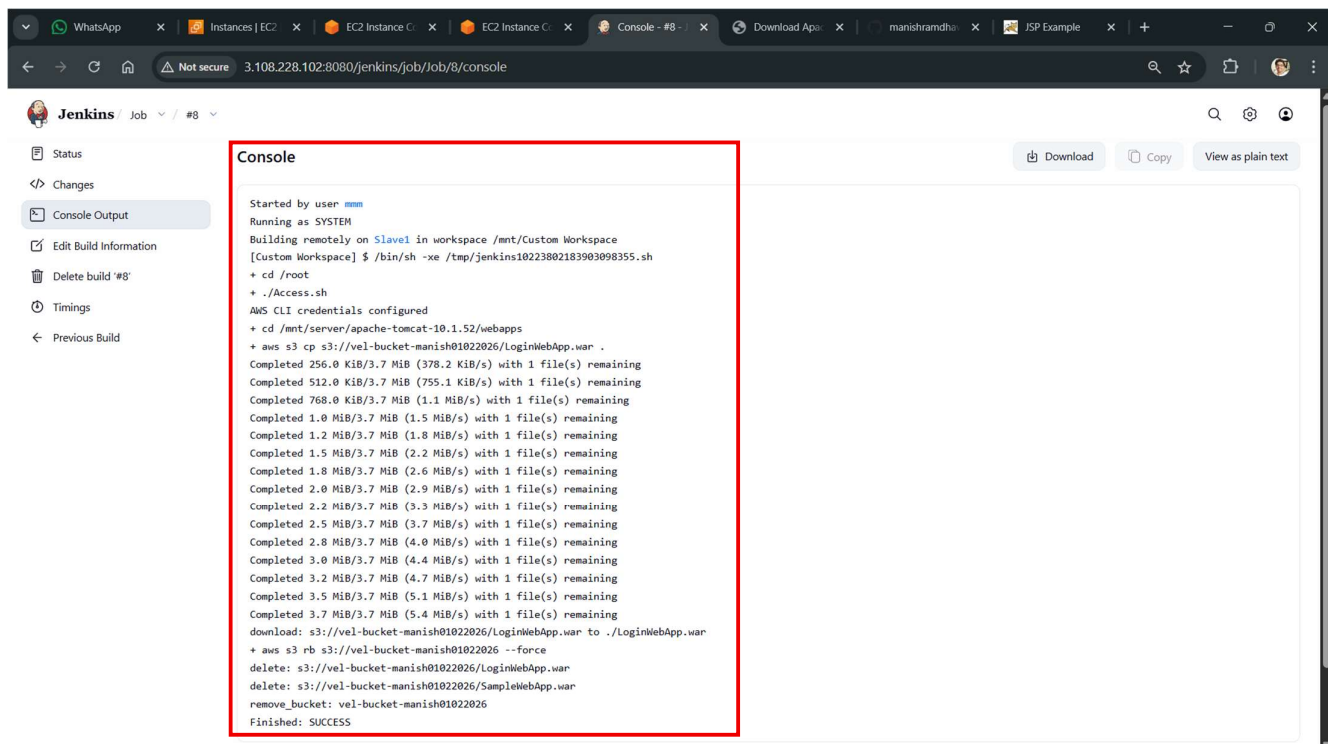
The screenshot shows the Jenkins Job Configuration page at the URL `13.234.202.85:8080/jenkins/job/Job/configure`. The left sidebar shows the configuration categories: General, Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. The 'General' tab is selected. A red box highlights the 'Restrict where this project can be run' section, which is checked and set to 'Slave1'. Below this, another red box highlights the 'Use custom workspace' section, which is also checked and set to '/mnt/CustomWorkspace/'. The 'Display Name' field is empty. At the bottom, there are 'Save' and 'Apply' buttons.

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**:

