**Setting Up Node.js Application with Jenkins**

**Introduction**

This document provides a step-by-step guide to setting up a Node.js application with Jenkins for automated building, testing, and deployment. It includes prerequisites, configuration, and deployment details.

**1. Prerequisites**

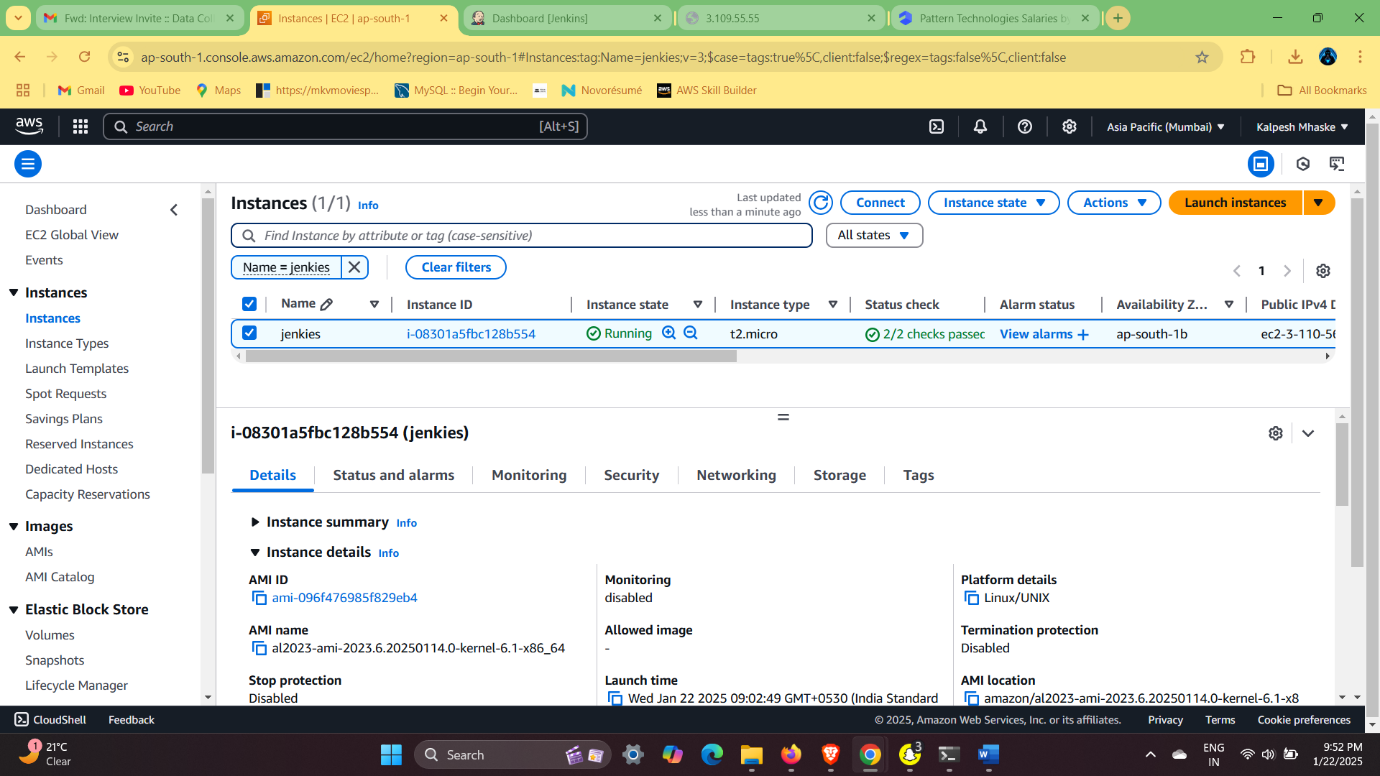
* **Jenkins Installed**: Ensure Jenkins is installed and running on a server.
* **Node.js and npm Installed**: Verify Node.js and npm are installed on the build server.
* **Git Repository**: A Git repository containing the Node.js application code.
* **Docker Installed**: If deploying using Docker.

Software and Tools

**Software and Tools**

* Jenkins Plugins:
  + **Git Plugin**
  + **Pipeline Plugin**
  + **NodeJS Plugin** (optional for managing Node.js within Jenkins)

1)Launch Jenkins EC2(Jenkins-Server)



2) Make A Jenkins Volume & Docker Jenkins Container

> mkdir jenkiesvolume

> sudo su

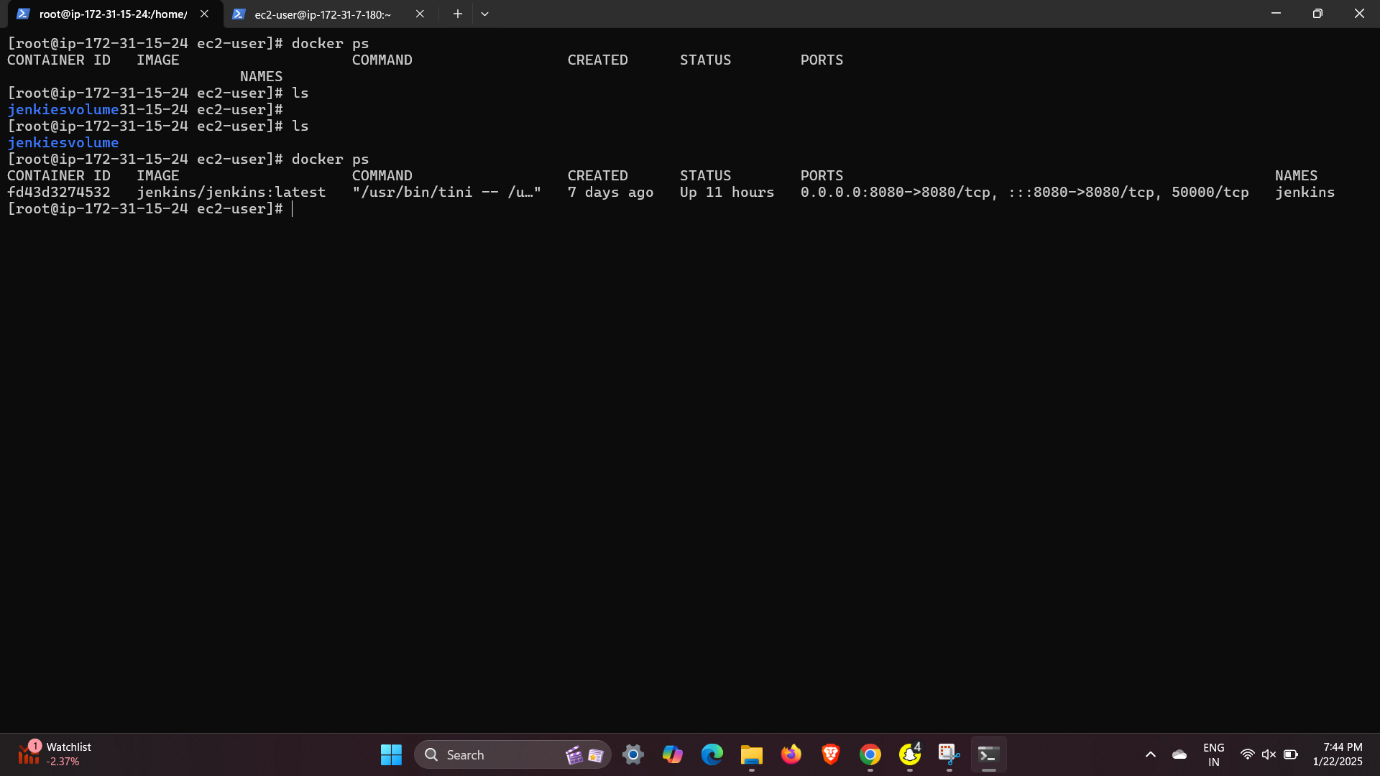
> sudo yum install docker -y

> sudo service docker start

2) Make A Docker Container

* docker run --name jenkins -v /home/ec2-user/jenkinsvolume:/var/jenkins\_home -p 8080:8080 jenkins/Jenkins
* Docker Ps

Note- Add Port number 8080 in Security Group



* Paste IP on Browser

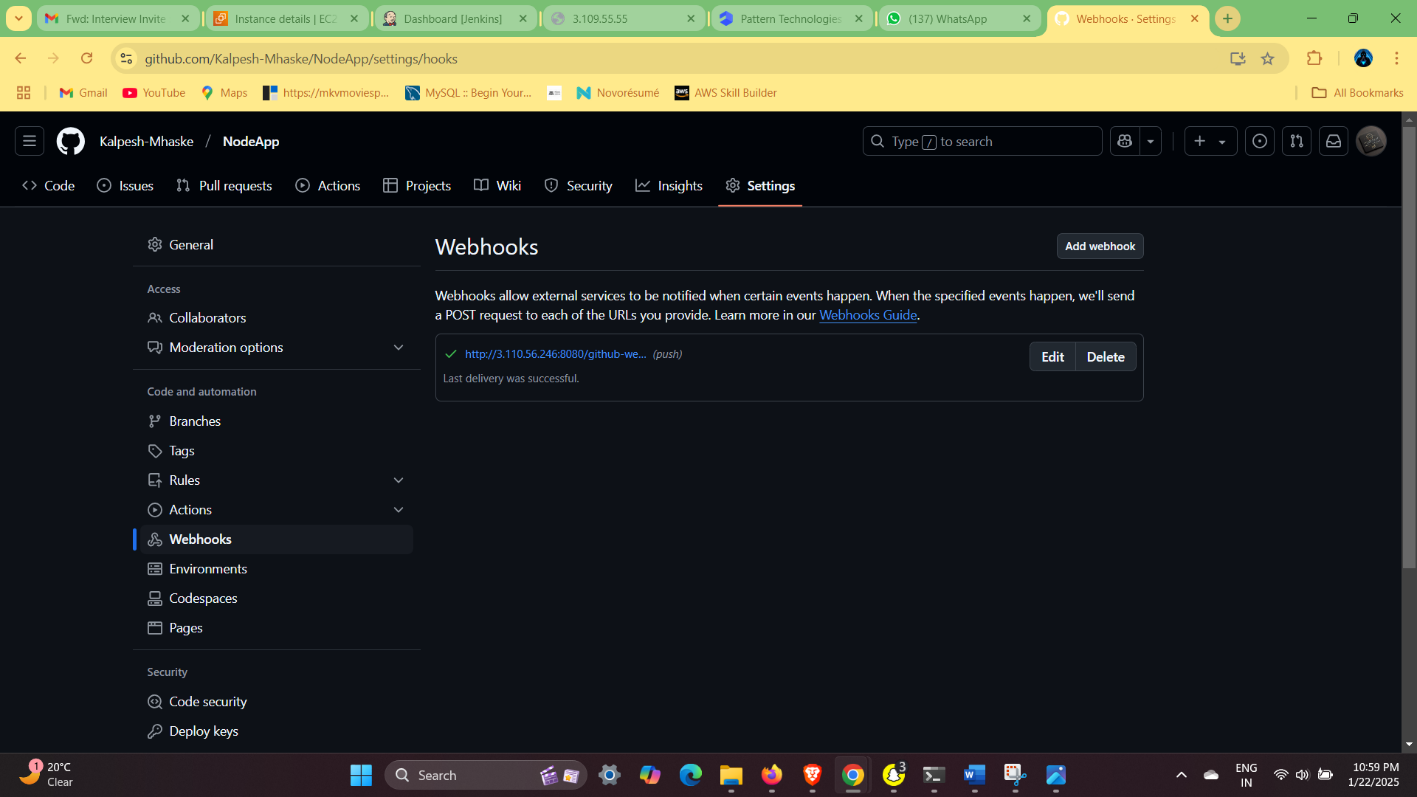
# For Generating Jenkins login Password

* docker exec -it jenkins cat /var/jenkins\_home/secrets/initialAdminPassword

- Go-to Jenkins Dashboard

**3) Automating Builds with Webhooks**

* **Configure a webhook in your Git repository to trigger builds on code push.**
* **Add Jenkins URL-IP to Github repository Webhook to access code**

****

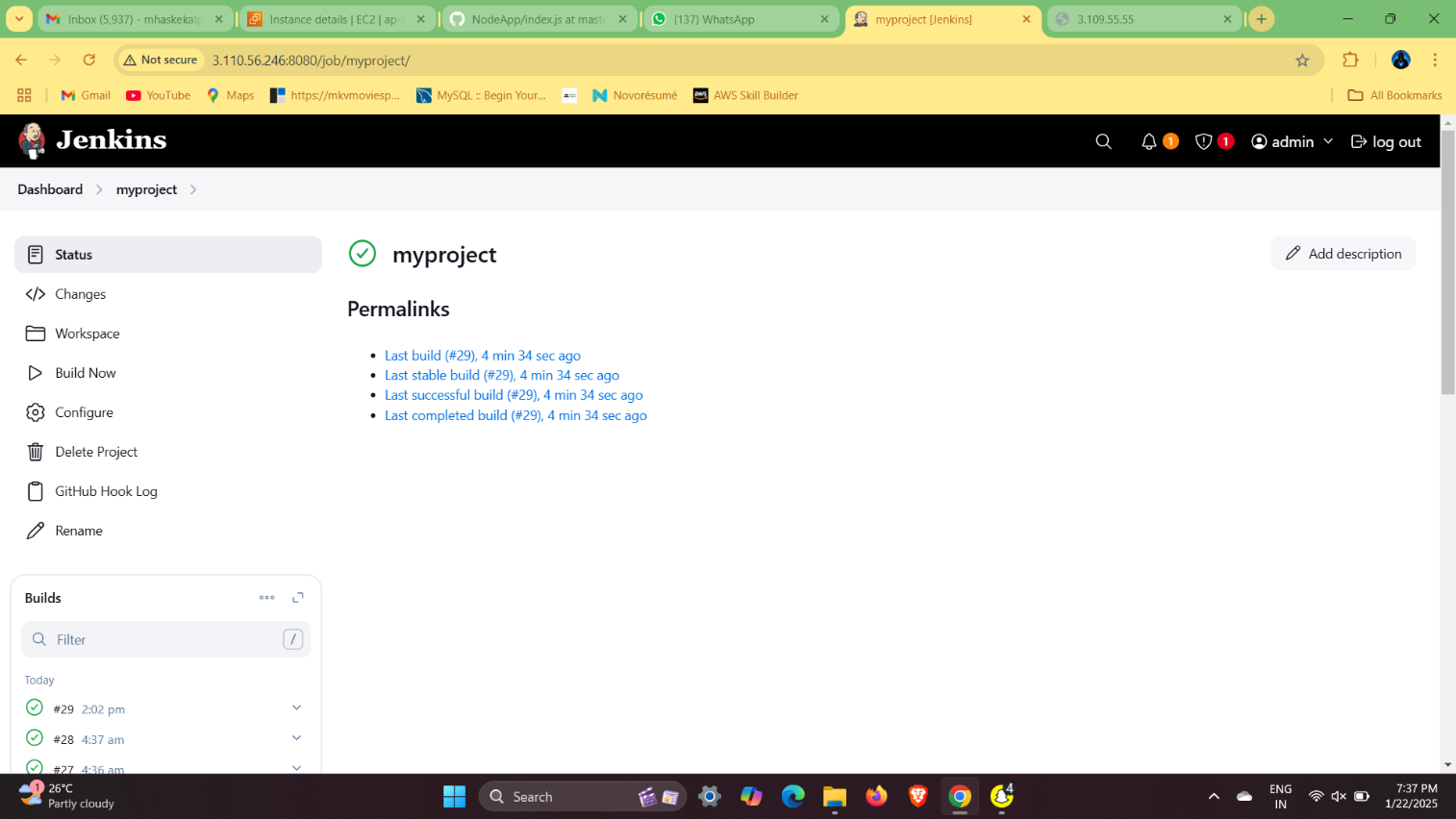
**Creating Jenkins Configuration**

**1) Create Job: myproject**

**Purpose:** Could be a custom project; adapt the configuration based on its purpose.

**Configuration:**

1. **SCM:**
   * Verify the Git repository and branch.
2. **Build Triggers:**
   * Same triggers (polling or webhook).
3. **Build Steps:**
   * Define specific commands or scripts needed for this project.
4. **Post-Build Actions:**
   * Set up notifications or further steps.



**2) Create Job: NodeAppBuild**

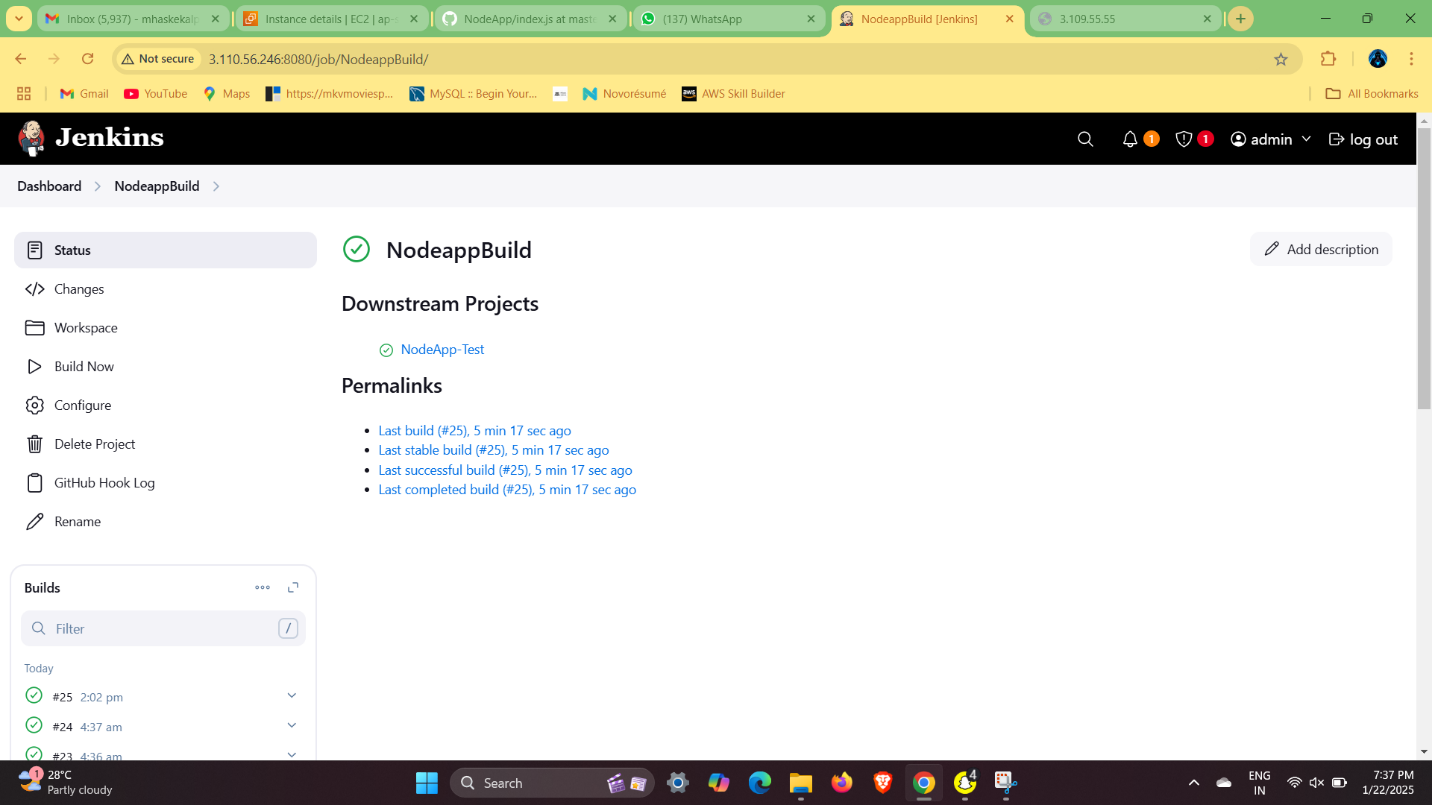
**Purpose:** Handles building the application.

**Configuration:**

1. **SCM:**
   * Add the Git repository as the source.
2. **Build Triggers:**
   * Same as the other jobs.
3. **Build Steps:**
   * Run build commands:

On Sh

* npm install
* npm run build



**3)Create Job: NodeApp-Test**

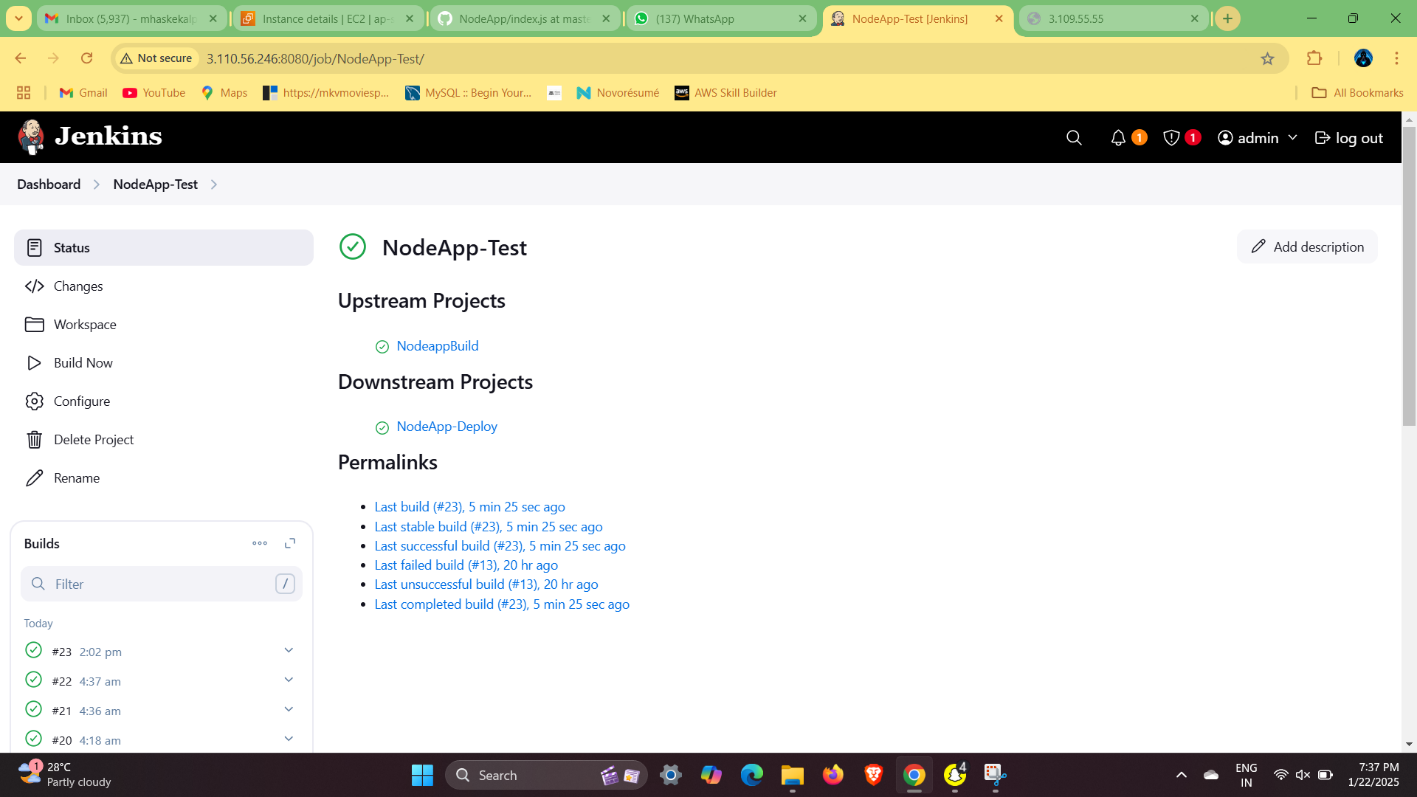
**Purpose:** Runs tests for the application.

**Configuration:**

1. **SCM:**
   * Ensure the correct repository is selected.
2. **Build Triggers:**
   * Enable build triggers like Poll SCM or webhook.
3. **Build Steps:**
   * Run test commands:

On Sh

* npm install
* npm test



**4) Create Job: NodeApp-Deploy**

**Purpose:** Handles deployment of the Node.js application.

**Configuration:**

1. **Source Code Management (SCM):**
   * Ensure the repository URL is correct.
   * Add credentials if required.
2. **Build Triggers:**
   * Use GitHub webhook or Poll SCM:

H/5 \* \* \* \* (Every 5 minutes)

1. **Build Steps:**
   * Add deployment commands, e.g., copy files to the server:

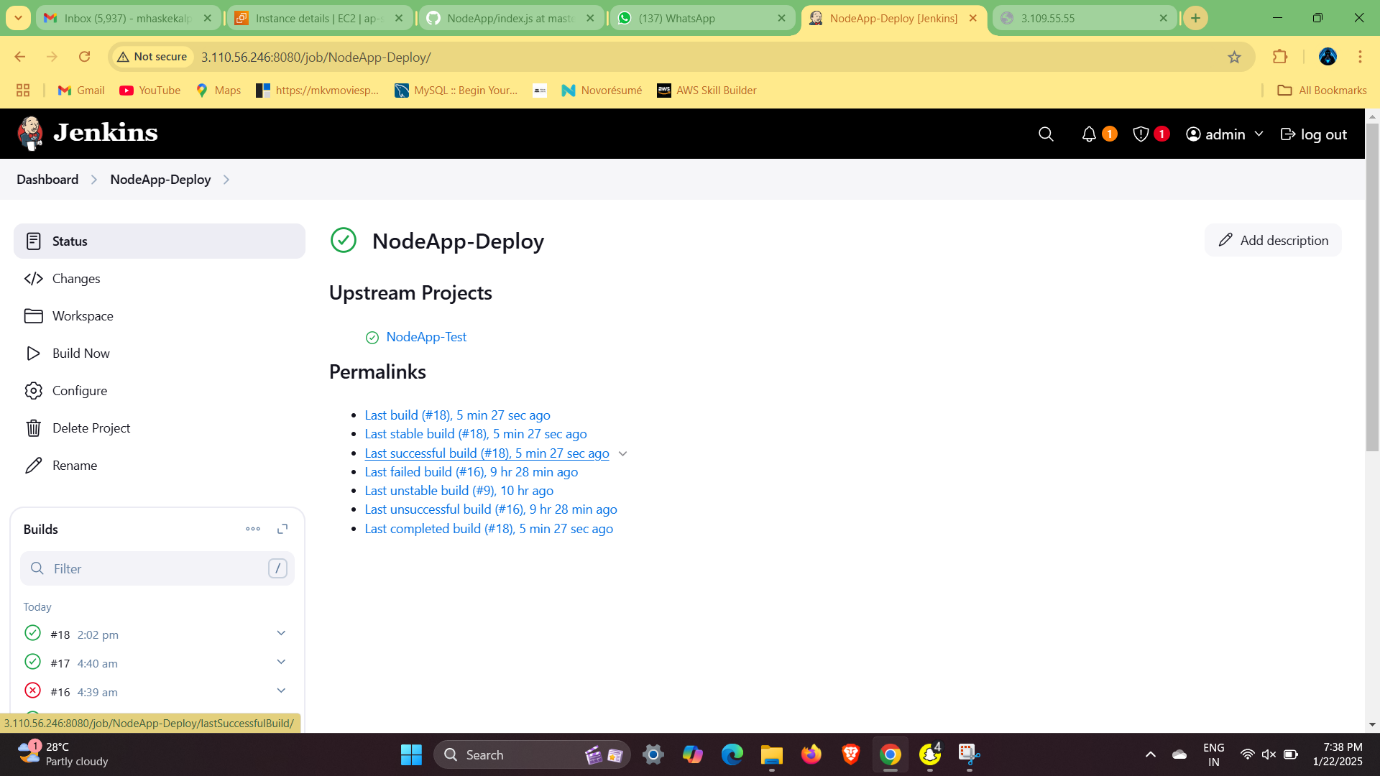
bash

scp -r dist/\* user@server:/path/to/deploy

* + Restart the server or application:

On Sh-

* ssh user@server 'pm2 restart app'



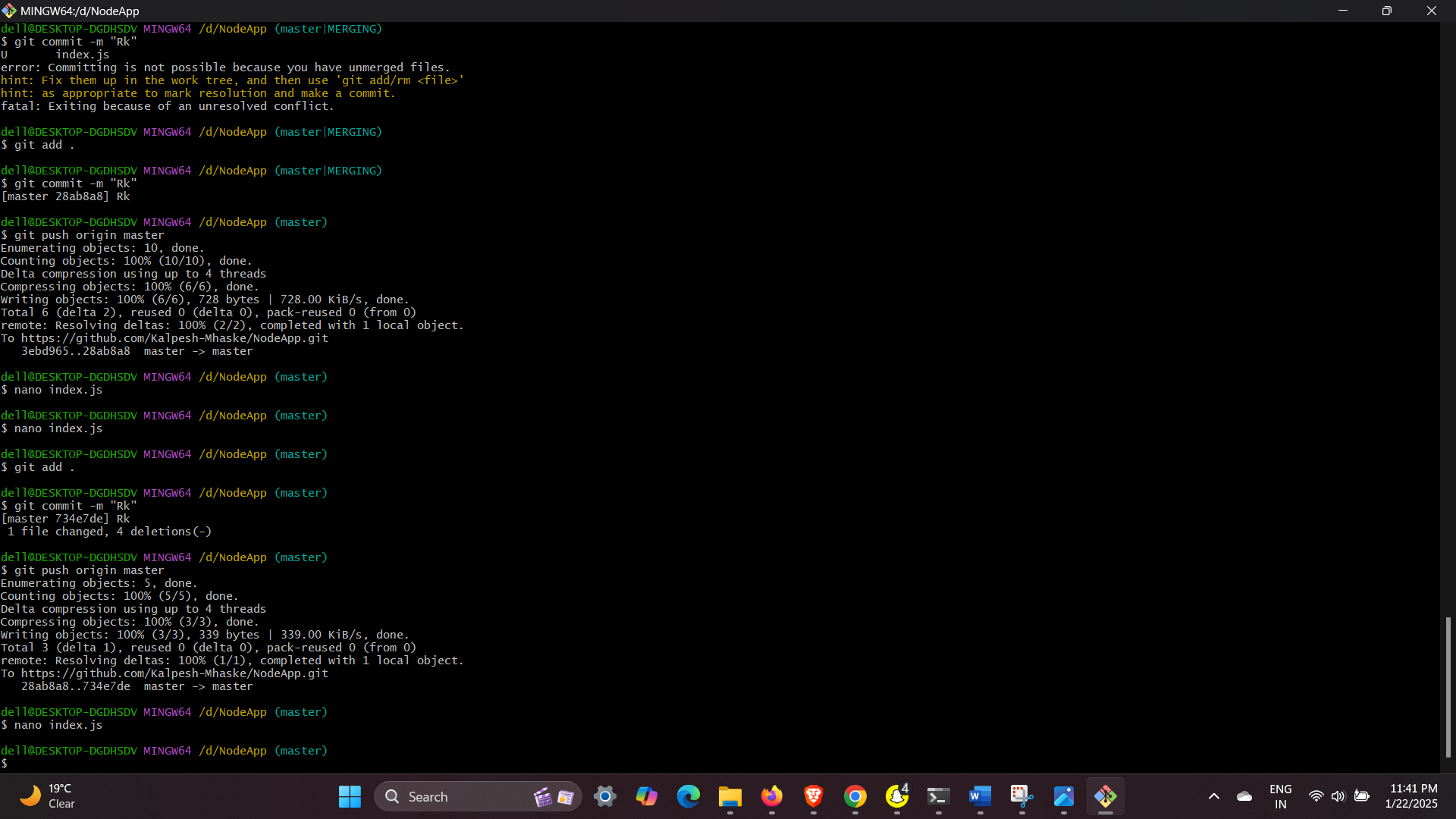
**4) Configure NodeJS Plugin**

* Go to **Manage Jenkins > Global Tool Configuration**.
* Add a NodeJS installation under **NodeJS Installations**.
* In your Jenkins job, use the NodeJS environment in the **Build Environment** section

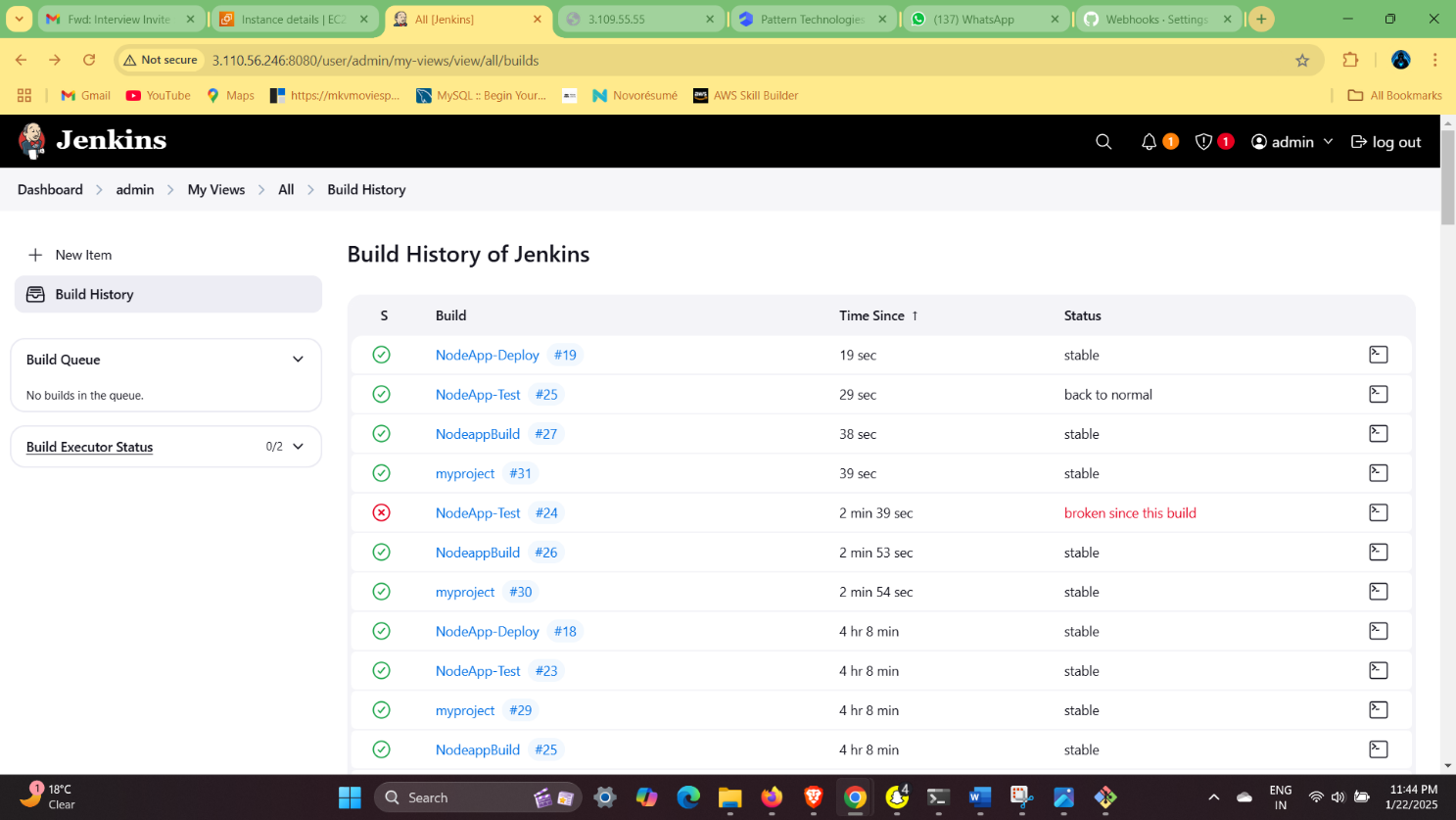
**5)Test the Pipeline**

1. **Trigger the Build:**
   * Click **Build Now** in the Jenkins job dashboard.
2. **Monitor the Pipeline:**
   * Check logs for each stage in the **Console Output**.

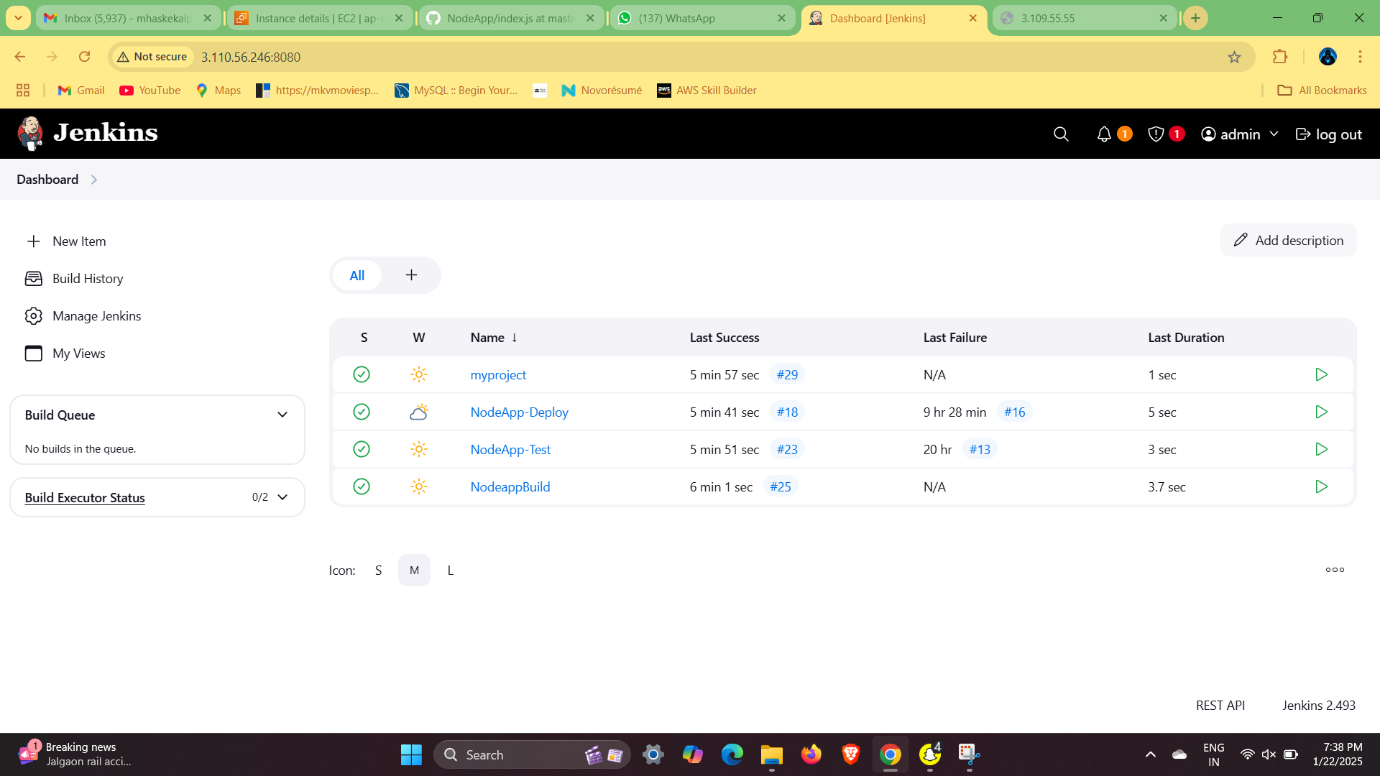
**6) Managing Code from local Machine with Git-Bash**



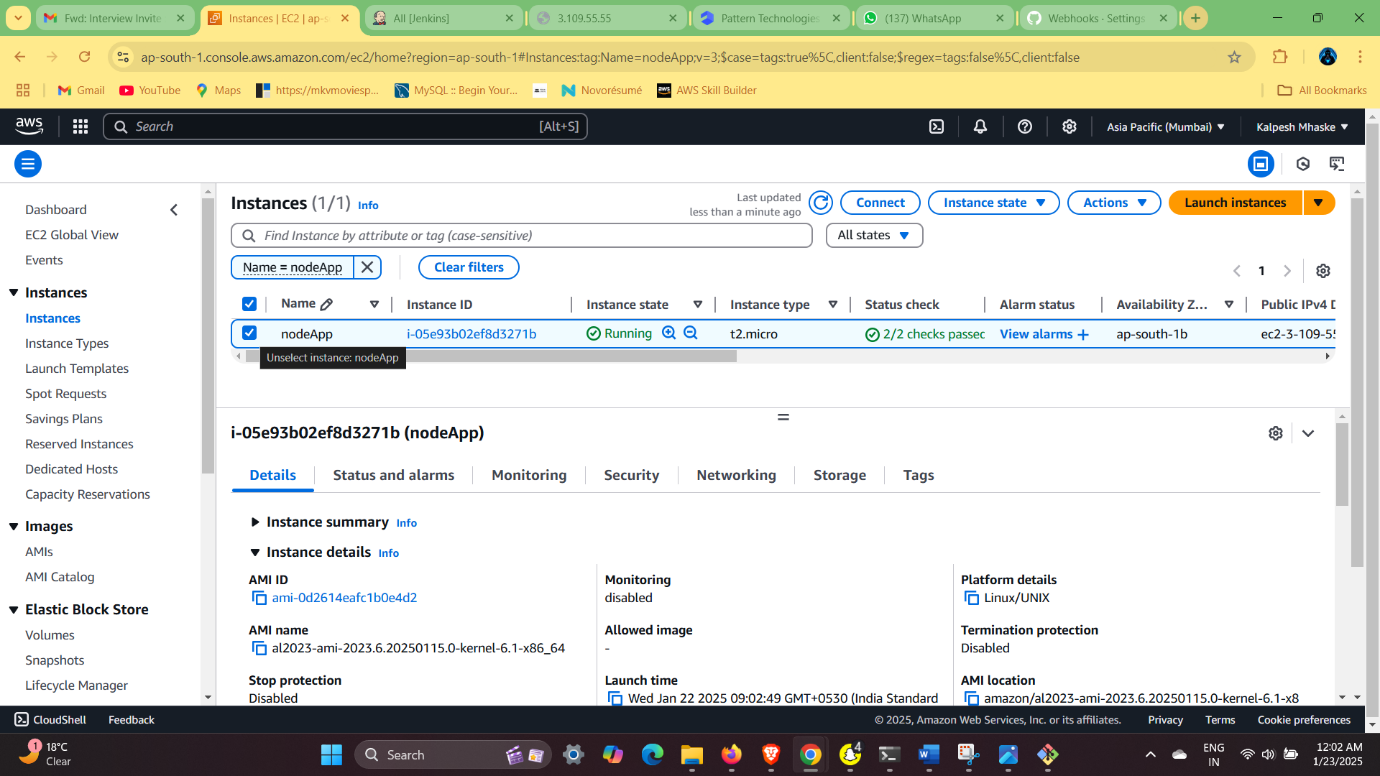
**7) NodeApp Build- Test-Deploy**

****

**All -Jobs**

****

**8) NodeApp Setup on EC2 (Live-Server)**

****

1. Install Node.js

> sudo yum install nodejs

> node -v

*Check Node.js version to confirm installation.*

2. Install Git

> sudo yum install git -y

*Ensures Git is installed for cloning repositories.*

3. Clone the NodeApp Repository

> git clone <https://github.com/Kalpesh-Mhaske/NodeApp.git>

> cd NodeApp

4. Install Dependencies

> npm install

*Installs all necessary packages listed in package.json.*

5. Run the Application

> node index.js

*Runs the Node.js application.*

NGINX Setup and Configuration

1. Install NGINX

> sudo yum install nginx

2. Edit the NGINX Configuration

> sudo nano /etc/nginx/nginx.conf

*Ensure NGINX is configured to proxy requests to your Node.js app (e.g., port 3000). Example configuration:*

[ location / {

proxy\_pass <http://localhost:3000>; ]

3. Start and Reload NGINX

> service nginx start

> sudo service nginx reload

*Starts the NGINX service and reloads the configuration.*

4. Test Connectivity

> curl http://localhost:3000

*Verify the Node.js app is running and accessible through the configured port.*

Process Management with PM2

1. Install PM2 Globally

> sudo npm install -g pm2

*PM2 ensures the app runs in the background and restarts on failure.*

2. Start the App with PM2

> pm2 start index.js

*Start the application and let PM2 handle process management.*

Optional Checks

1. Check Docker Installation

> docker --version

*Verify if Docker is installed on the system.*

2. Check Current Directory

pwd

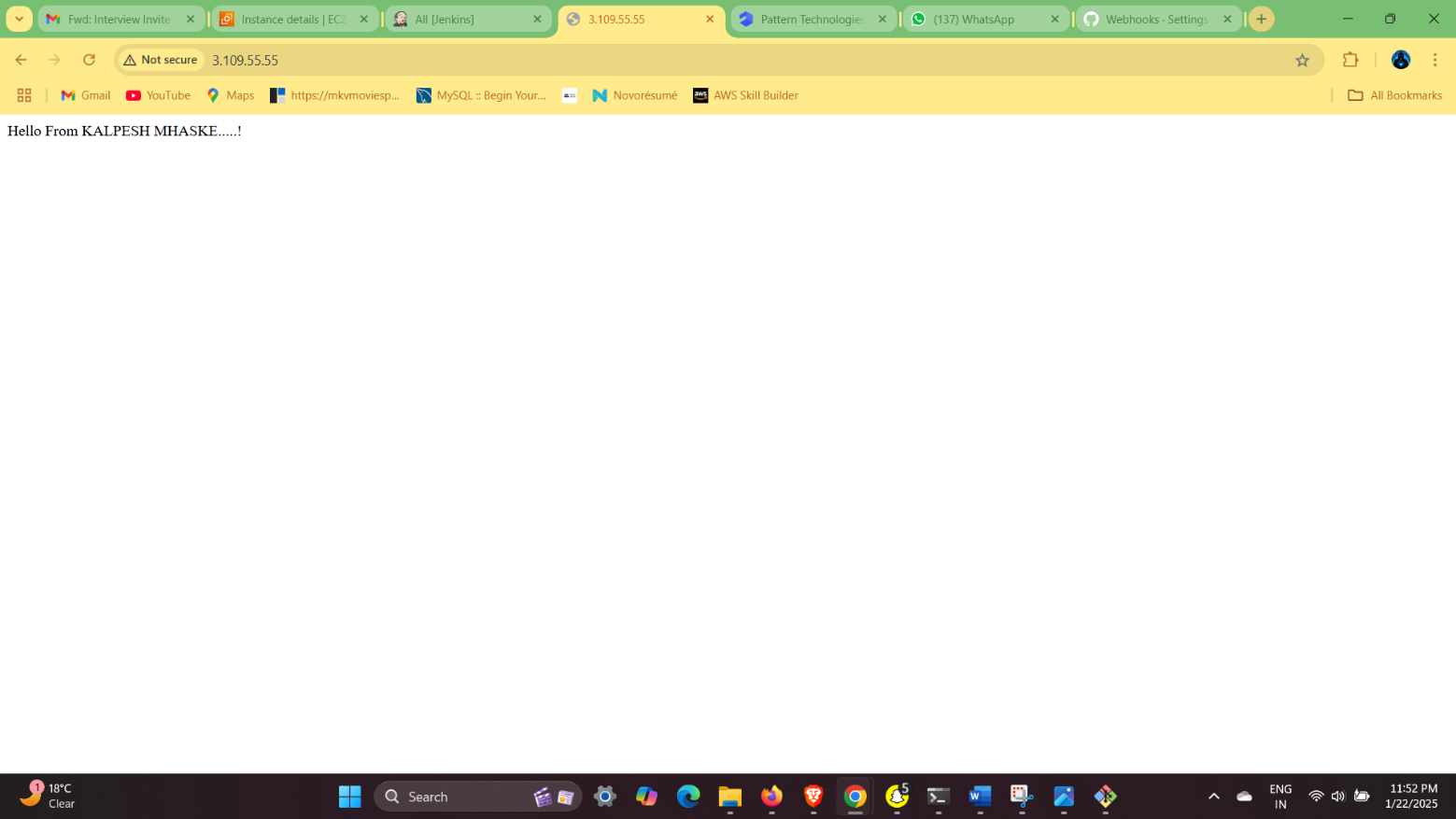
ls

*Useful for navigating and managing files.*

3. Access NodeApp Directory

> cd NodeApp

**9) Final Ouput of Our App**

****

Notes

* AWS Security Group: Ensure the correct port (3000) is open for incoming traffic.
* PM2 Persistence: Use pm2 save to save the process list and pm2 startup to configure auto-start on reboot.