Deploying Angular calculator Application using nodejs and npm (javascript) in nginx:

Prefix:

This document provides a detailed explanation of the step-by-step process to build and deploy a JavaScript application using a two-server architecture. The first server, referred to as the **Build Server**, is responsible for compiling the source code, installing dependencies, and generating the production-ready build artifacts. The second server, known as the **Deployment Server**, hosts and runs the application, ensuring it is accessible to end users. The deployment process involves securely transferring the build artifacts from the Build Server to the Deployment Server, configuring the environment, and verifying successful application startup. This setup ensures a clear separation of build and deployment responsibilities, improving maintainability and scalability.

Pre-requisites:

Software Requirements:

*Node.js & npm installed

*Build scripts defined in package.json (e.g., npm run build)

Pre-deployment Steps:

*Run npm install to install dependencies

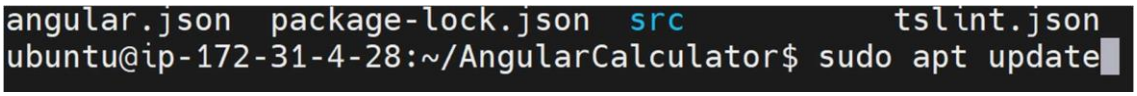
*Run npm run build to create production build

Step 1

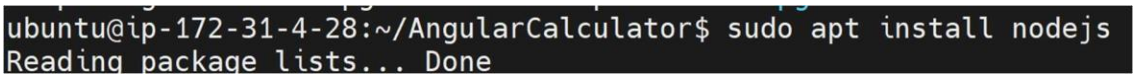
*Launch 2 servers (1 Build server 1Deploy server)

Build server:-

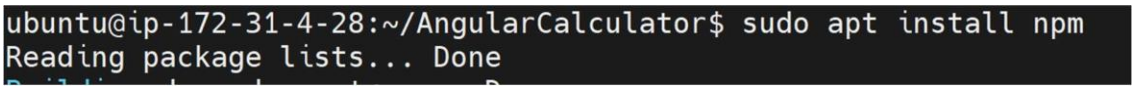
Sudo apt update



Sudo apt install nodejs



Sudo apt install npm



Sudo apt install -g @angular/cli

(This command installs the Angular CLI (Command Line Interface) globally on your system using npm (Node Package Manager).

curl -fsSL https://raw.githubusercontent.com/nvm- sh/nvm/master/install.sh | bash

(To **download and install NVM**, a tool that lets you easily **install, manage, and switch between multiple Node.js versions** on the same machine.)

nvm install 20

(is used to **install Node.js version 20** on your system using **NVM (Node Version Manager)**

```
ubuntu@ip-172-31-4-28:~/AngularCalculator$ npm install
```

export NODE_OPTIONS=--openssl-legacy-provider

(is used to **resolve compatibility issues** between **Node.js (v17 and above)** and older versions of **Webpack** or other JavaScript build tools that rely on deprecated OpenSSL algorithms.)

```
ubuntu@ip-172-31-4-28:~/AngularCalculator$ export NODE_OPTIONS=--openssl-legacy-provider
```

Ng build

Dist folder created is created
(To **convert Angular source code** (TypeScript, HTML, CSS) into **optimized JavaScript, HTML, and CSS files** that can be deployed on a web server)

```
ubuntu@ip-172-31-4-28:~/AngularCalculator$ ls
README.md     dist  node_modules       package.json  tsconfig.json
angular.json  e2e   package-lock.json  src           tslint.json
```

STEP 2

Deploy server

nginx Install

(Deployment

server)

```
ubuntu@ip-172-31-9-81:~$ sudo apt install nginx -y
Reading package lists... Done
```

Nginx exposed

Step 3

Give permissions:

sudo chown -R www-data:www-data /var/www/html

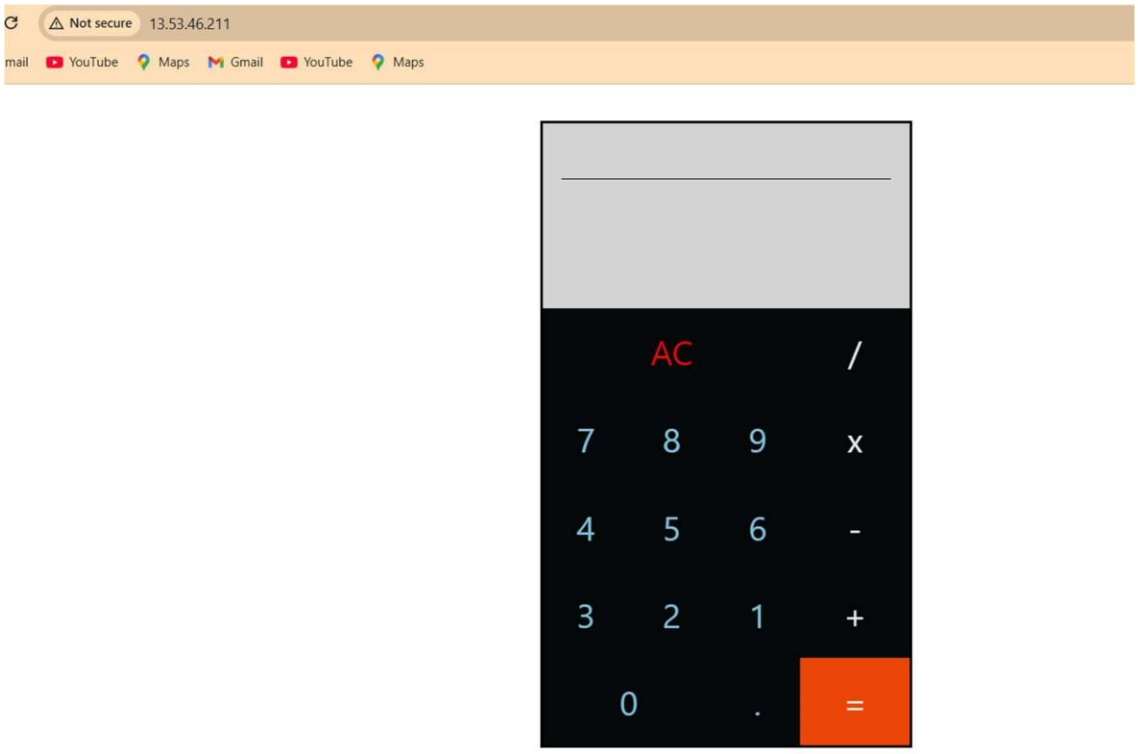
(Changes the ownership of /var/www/html and its contents to www-data so that Nginx can access and serve the files.)

sudo chmod -R 755 /var/www/html

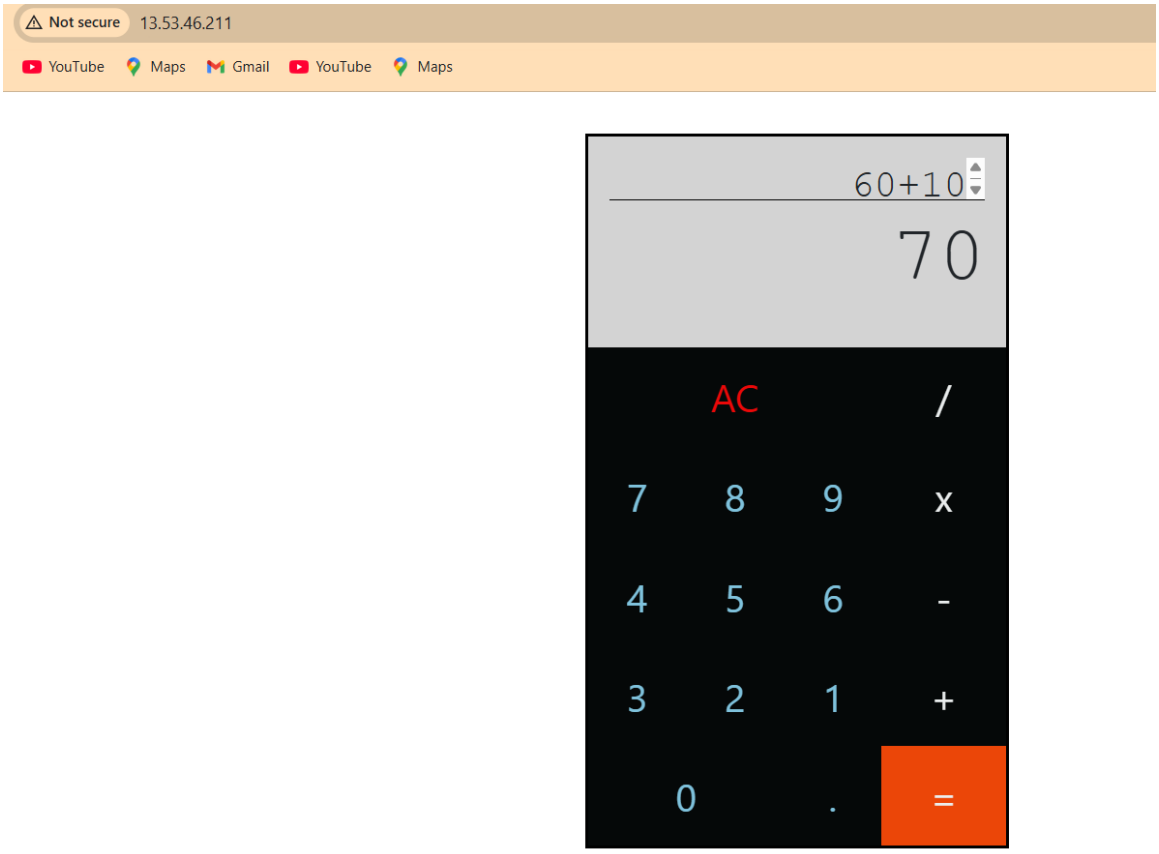
(These commands are used to **set ownership and permissions** for the web server directory (/var/www/html) where your application will be deployed)

```
ubuntu@ip-172-31-9-81:~$ sudo chown -R www-data:www-data /var/www/html
ubuntu@ip-172-31-9-81:~$ sudo chmod -R 755 /var/www/html
```

Expose the app



App is working



Overall Purpose:

The project aims to **build and deploy a JavaScript web application** using a two-server setup, with a **Build Server** for compiling code and a **Deployment Server** for hosting. It ensures the application is **optimized, secure, and accessible** to end users.