**Frontend server setup procedure**

**Step 1: Prepare Your AWS Environment**

1. **Log in to AWS:**
   * Log in to your AWS Management Console.
2. **Create an EC2 Instance:**
   * Go to the **EC2 Dashboard**.
   * Click **Launch Instance** and configure the following:
     + **AMI:** Use a Linux-based AMI like **Amazon Linux 2** or **Ubuntu 20.04**.
     + **Instance Type:** Choose at least **t2.medium** (or larger for production).
     + **Key Pair:** Create or select an existing key pair.
     + **Security Group:** Open these ports:
       - **22** for SSH
       - **3000**

**Verify the Key and Permissions**

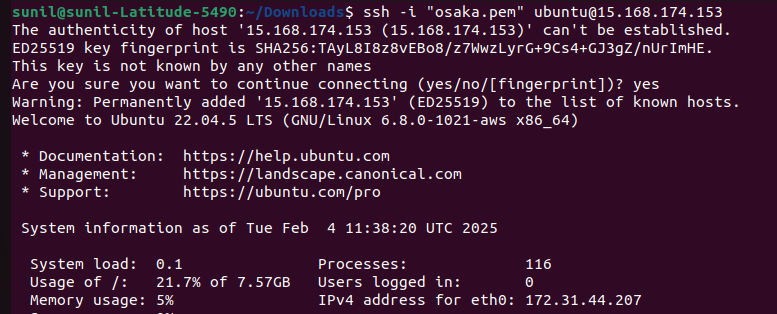
1. **Ensure the key file is correct:**
   * Double-check that osaka.pem is the private key associated with the EC2 instance.
2. **Ensure the key file has the correct permissions:** Run the following command to set the correct permissions:

**Chmod 400 osaka.pem**

**Access the EC2 Instances:**

* SSH into the EC2 instances using

Example: -

ssh -i your-key.pem ec2-user@<instance-ip>

Use the command:

sudo apt update: - Update the package

sudo apt install git: - Git installation

A screenshot of a computer program

Description automatically generatedTo clone the front-end server repo

git –version: - Verify the installation.

**Make installation:**

Sudo apt install make

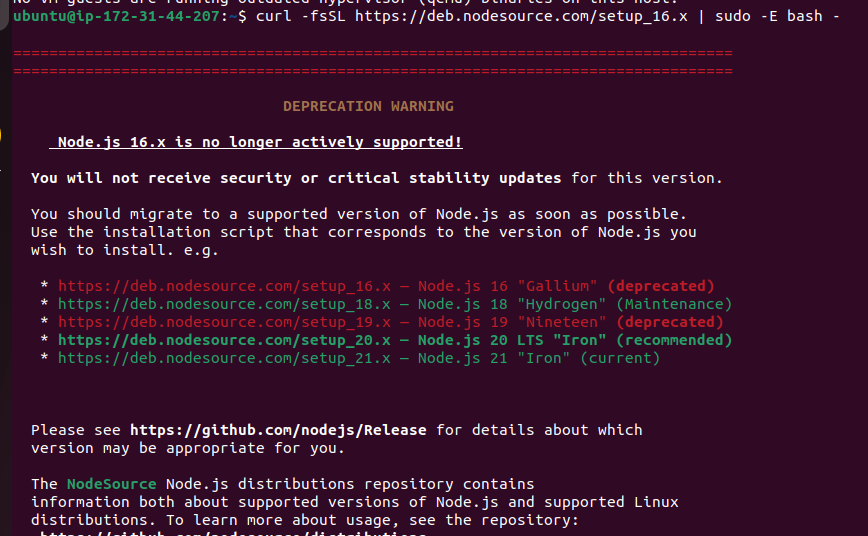
A computer screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated Sudo apt install make-guile

**Install Node.js and npm from the NodeSource repository:** - First, install the NodeSource repository to get the latest version of Node.js



Install Node.js and npm:-

A screenshot of a computer program

Description automatically generatedsudo apt install -y nodejs

Verify the installation: -

node -v

npm -v

A screenshot of a computer program

Description automatically generated

Now clone the frontend sever repo

git clone <repo>

A computer screen shot of a computer code

Description automatically generated

Once the repo is cloned to your server then verify the file then go inside the file, commands are mentioned below:

ls: - for to list the folder

cd < file name>

A computer code with numbers and symbols

Description automatically generated

For building the application, we can use make command

make build

Once we run the command then we will see the results, which is mentioned below

A computer code with many small letters

Description automatically generated with medium confidence

A building in the dark

Description automatically generated

The build folder has been generated successfully, and the application is ready to be deployed.

If you're planning to deploy your app, you can follow the instructions provided to serve the build locally for testing

Commands:

A screenshot of a computer screen

Description automatically generatedsudo npm install -g serve

serve -s buildA black screen with white text

Description automatically generated

The steps have been completed on the server side. The app should now be running, but we have made a small change in the security groups on AWS.

Here are the steps.

1st go to the security group which is connected to your frontend server.

A screenshot of a computer

Description automatically generated

2nd check the inbound rules

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated3rd click on the add rules

4th, select "Custom TCP" from the drop-down menu as shown in the picture. Then, enter the port “3000” on which your frontend server is running, select "Anywhere" as the source, and save the rule.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedNow, refresh the page and open any browser. In the search bar, enter the public IP of your frontend server followed by port number 3000, as shown in the picture.

Now, you should be able to see that the front-end server is up and accessible.