

Introduction

This guide will walk the reader through the process of setting up the Health Care Staffing Nigeria web platform both locally for local development and on a web server for production using AWS EC2.

Please refer first to the README.md. Those instructions will be more concise. Please use this guide in case more clarity is needed or you intend to put platform on production.

Table of Contents

This guide has 4 parts:

1. Local Installation
2. Setting database
3. Production Installation
4. AWS services

Part 1 Local Installation

1. Install Node JS and Node Package Manager (NPM)
 - a. Node JS can be downloaded from <https://nodejs.org/en/download/>
 - b. Attempting to install Node JS will bring up an installation wizard, continue through the installation wizard until Node JS is installed.
 - i. NPM will be installed along with Node JS
2. Install MySQL
 - a. Follow the steps in part 2 of this document
3. Acquire Tiny Text Editor API Key
 - a. Can be found here : <https://www.tiny.cloud/>
4. Then go to file ./client/src/components/facility/CreatePosting.jsx and on line 132 paste the key into the empty string.
 - a. Empty string is the open and close apostrophes containing nothing
5. Complete AWS services set up, in part 4 of this document.
6. Run the runs.sh script.
 - a. On MacOS or Linux:
 - i. Go to the folder containing the project
 - ii. Right click the file the file manager
 1. In macOS go to services -> New Terminal At Folder
 2. In Linux click open terminal here
 - iii. Type ./run.sh into the terminal
 1. For first time set up with no database yet add “-ide”
 - a. For example: ./run.sh -ide
 2. For consecutive set up simply use -l
 - a. For example: ./run.sh -l
 - i. -i argument to install node dependencies
 - ii. -d argument to reset the database with sample data
 - iii. -c argument to clean the database
 - iv. -p argument to update the database procedures
 - v. -e argument to set up environmental variables (should only be run once at first start up)
 - b. When ./run.sh -ide is ran
 - i. “Enter password:” should appear on the next line, if MySQL was given a password in part 2, enter it here, otherwise just hit enter.
 1. The script will ask for a password 5 times (1 for each SQL script)
 - ii. Then it will prompt you for various environment values
 1. For node server port enter 4000
 2. In the database section enter
 - a. Localhost for host
 - b. Root for user
 - c. The mysql password from step 2 for password
 - d. cmg_staffing_nigeria for
 - c. For the Cognito and Bucket information use the settings set up in step 3.

Part 2 Database

How to initialize the local database:

1. download MySQL:

If you have homebrew enter "brew install mysql" into the command line

Otherwise go to <https://dev.mysql.com/doc/mysql-getting-started/en/#mysql-getting-started-installing> and download mysql

2. Start the mySQL server

If installed on homebrew run "brew services start mysql"

Otherwise follow the guide in the install link

3. Now run the sql file.

On MacOS or Linux run: "mysql -uroot -p < \$(pwd)/server/database/initializeDB.sql" from the main directory

When a password is requested, you can just hit enter.

On Windows run "mysql -uroot -p < %cd%/server/database/InitializeDB.sql"

Part 3 Production Set Up

To deploy to production, this guide will discuss using AWS EC2 Service.

This guide assumes the project is currently stored in a github repository, if that is not the case

This guide was heavily inspired by a guide by Jason Watmore with changes to use MySQL and to initialize the environmental variables for the server. For additional pointers, consider referring to that guide located here:

<https://jasonwatmore.com/post/2019/11/18/react-nodejs-on-aws-how-to-deploy-a-mern-stack-app-to-amazon-ec2>

Before proceeding please complete AWS Set Up.

1. Complete the AWS Set Up in Part 4
2. On the EC2 Dashboard, go to Launch Instance and click Launch Instance
3. Select An Ubuntu AMI
4. Select the instance type
 - a. For just getting the site up and running, t2.micro can work, although if the platform receives more traffic, a more powerful instance may improvement performance
5. Edit Network Settings
6. Add Security group rule
 - a. Add an http and https connection type
 - i. This will require adding two new security group rules
7. Click Launch
8. A pop-up window will come up inquiring about a key pair
 - a. Enter the name of the key pair to whatever you would like
 - b. Click Create key pair and the key should download
9. A new page should come up, navigate to all instances
 - a. Alternatively, click Instances on the side bar to the left, and the instance should be there
10. Select the created instance and record the "Public IPv4 DNS"
11. In the terminal, navigate to the key that was downloaded
12. Execute the command "chmod 400 <key-file-name>"
13. Execute the command "ssh -i <file-path-to-key> ubuntu@<ipv4 address>"
 - a. The Ipv4 address can be directly copied from the instance page for aws
 - b. Answer yes to the following question and continue
14. Enter the following command: "sudo git clone <git-repository>"
 - a. If it asks for a password, then you will need to generate a personal access token on github and use that as the password
 - i. Generating personal access tokens are located under settings -> developer settings
15. Acquire Tiny Text Editor API Key
 - a. Can be found here : <https://www.tiny.cloud/>
16. Then go to file ./client/src/components/facility/CreatePosting.jsx and on line 132 paste the API key into the empty string.
 - a. Empty string is the open and close apostrophes containing nothing

17. Navigate into the folder and run the shell script `“./deploy_server.sh”`, the exact commands of that script are described in the note after these steps.
 - a. If the file refuses to run execute `“sudo chmod +x deploy_server.sh”`
18. Initialize database
 - a. Run `“sudo chmod +x deploy_db.sh”`
 - b. Run `“./deploy_db.sh”`
 - c. Run `“sudo chmod -x deploy_db.sh”`
19. Run the command `“./enviro.sh”`
 - a. For port number put 4000
 - b. Answer the prompts about the database configuration.
 - c. Answer the prompts about the AWS configurations
20. Navigate to Healthcare-Staffing-Nigeria/server/node
 - a. run `“sudo npm install”`
 - b. Then run `“pm2 start index.js”`
21. Navigate to Healthcare-Staffing-Nigeria/client
 - a. Run `“sudo npm install”`
22. Run `“sudo rm /etc/nginx/sites-available/default”`
23. Open a text editor with: `“sudo nano /etc/nginx/sites-available/default”`
 - a. Paste the contents of `“Healthcare-Staffing-Nigeria/nginx.txt”` into the text editor
 - b. Save the file with `ctrl + x`
24. `sudo systemctl restart nginx`

Note: The shell script file `deploy_server` initializes the environment, it runs the following commands:

- a. `curl -sL https://deb.nodesource.com/setup_16.x | sudo -E bash -`
 - i. Downloads node js and related tools
- b. `sudo apt-get install -y nodejs`
 - i. Installs nodejs
- c. `sudo apt install mysql-server`
 - i. Installs mySQL (for the database)
 - ii. When a prompt asks for confirmation enter ‘y’ to confirm
- d. `sudo systemctl status mysql`
 - i. Checks if the server is running or not
 - ii. Look for “active (running)”
- e. `sudo npm install -g pm2`
 - i. Installs pm2
 - ii. This a process manager
- f. `sudo pm2 startup systemd`
 - i. Has pm2 start automatically upon startup
- g. `sudo apt-get install -y nginx`
 - i. Installs nginx for handling https requests
- h. `sudo ufw allow OpenSSH`
 - i. Allows ssh connections through firewall
- i. `sudo ufw allow 'Nginx Full'`
 - i. Allows https and http connections through the firewall

- j. `sudo ufw --force enable`
 - i. Enables the firewall

Part 4: AWS Set Up

Go to Amazon AWS console, and search for either Cognito or S3. The following guides will assume that you navigated to the Cognito or S3 sections respectively.

Amazon Cognito

1. Go to create new user pool
2. Give the user pool a name, the name is arbitrary.
3. Then Step through the settings.
4. Choose to allow users to sign in with email only
 - a. Choose the case insensitivity option
5. Select following attributes to be required
 - a. Email
 - b. Create a custom attribute named "type"
 - i. The type should be type string
 - ii. Allow it to be mutable
 - iii. AWS should automatically insert "custom:"
6. Set the minimum password length to 8
 - a. Require numbers, special, uppercase, and lowercase characters
7. Allow users to sign themselves up
8. Allow users to recover their account via email only
9. Verify email only
10. Edit Email settings as developer desires
11. Create 2 app clients
 - a. Call the first one React Client
 - b. Call the second one Node JS client
 - c. Do not create client secret
 - d. A client id should be generated upon creating them, keep track of this id. Whenever a client id is required, it refers to one of these keys.
 - e. Record both client ID's they will be used later
12. Go to domain name.
 - a. If deploying with a domain name, select use your domain
 - b. If not, enter a domain prefix in the Amazon Cognito domain section
13. In the review page, check the setting are correct
 - a. Double check that "custom:type" exists
14. Click Create Pool
15. Upon creating User Pool, a new page will come up
16. Record the Pool ID towards the top of this page
17. That completes the guide for setting up Amazon Cognito!

After following these steps, the following should be recorded:

- Client Id for **React Client** (Step 11) and Client Id for **Node JS Server** (Step 11)

- Pool Id (Step 16)

Amazon S3

1. Create a new bucket
2. Give the bucket a name, it can be arbitrary
 - a. Be sure to record the name, it will be used later
3. Select a region this is where the data will be stored
 - a. Be sure to record the region, it will be used later
4. Click Create Bucket
5. Navigate to IAM
6. Go to the policy page
7. Click create new policy
 - a. In the service section type or select S3
 - b. Click on the dropdown arrow by Write
 - i. Select DeleteObject and PutObject
 - c. Click on the dropdown arrow by Read
 - i. Select GetObject
 - d. In the Resources section, select Add ARN
 - i. In the pop up enter the bucket name created on step 2
8. Click continue
9. On the add tags page, simply continue to the next page
10. In the review page, give the policy a name, it can be arbitrary
11. Click create policy
12. Add a new user called cmgHiringNode
 - a. We will give the node js server access to access the bucket
13. Select Access key option and continue
14. Select attach existing policies directly
15. Enter the name of your policy from step 10, in the filter section and select that policy
16. Click next:tags, this page is not relevant, so click Next: Review
17. After creating the user a "Access key ID" and "Secret access key" will be generate.
 - a. Record both keys, these keys are sensitive
 - b. Secret access key will no longer be accessible after you leave the page.
18. That completes the guide for setting up Amazon S3 Bucket

After following these steps, the following should be recorded:

- Bucket Name (Step 2)
- Bucket Region (Step 3)
- Access Key (Step 17)
- Secret Access Key (Step 17)