How to setup server (node.js + react)

1. **Upload the source code**

* Clone the git repository
  + Move to home directory

$ cd ~

* + Clone it

$ git clone https://github.com/saroopmathur/appadmin.git

1. **PostgreSQL (Database)**

* Installing PostgreSQL
  + To install PostgreSQL, first refresh your server’s local package index

$ sudo apt update

* + Then, install the Postgres package along with a -contrib package

$ sudo apt install postgresql postgresql-contrib

* Creating a New Database
  + You can create the appropriate database with the createdb command

$ sudo -u postgres createdb acm\_db

Or

sudo -u postgres psql

postgres=# create database acm\_db;

* Importing Data from Backup
  + Move to the folder where backup file is

$ cd ~/appadmin/backend

* + To import data from backkup

$ sudo -u postgres psql acm\_db < acm\_db.sql

-

CREATE EXTENSION IF NOT EXISTS pgcrypto;

**CREATE** EXTENSION IF **NOT** **EXISTS** citext;

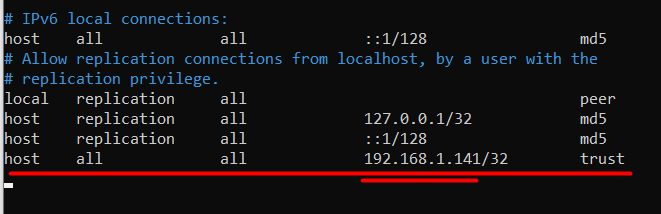
* Changing Password for postgres
  + To change default password for postgres user

$ sudo -u postgres psql acm\_db

acm\_db=# ALTER USER postgres WITH PASSWORD ‘postgres’;

* Allowing access from remote (optional)
  + Add the source IP address to postgresql configuration file

$ sudo nano /etc/postgresql/12/main/pg\_hba.conf



* Access postgres (password: “postgres”)

sudo -u postgres psql acm\_db -h localhost -W

1. **Nginx (Web Server)**

* Installing Nginx
  + We can install Nginx using the apt packaging system

$ sudo apt update

$ sudo apt install nginx

* Enabling HTTPS
  + Install LetsEncryp certbot

$ sudo apt install certbot python3-certbot-nginx

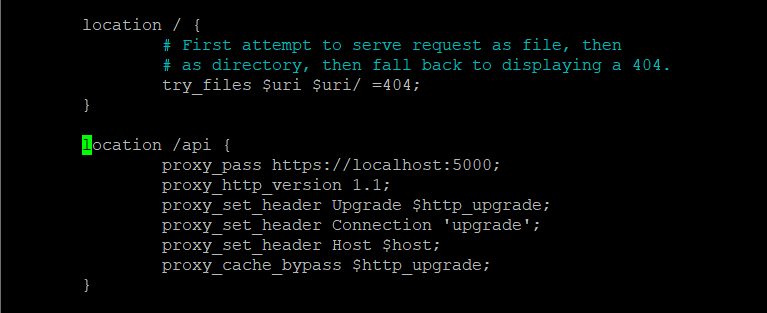
* + Obtaining a SSL Certificate

$ sudo certbot --nginx -d demo2.eclipz.live

sudo certbot --nginx -d webadmin.xpresstrust.com --register-unsafely-without-email

* Configuring as to serve the frontend and backend
  + Add a location for the proxy

$ sudo nano /etc/nginx/sites-available/default



proxy\_pass <https://localhost:5000>;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection ‘upgrade’;

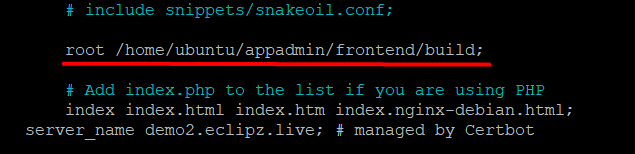
proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

* + Change the document root

$ sudo nano /etc/nginx/sites-available/default

root /home/ubuntu/appadmin/frontend/build;



**#Correct**

**location / {**

proxy\_pass [http://localhost:500](http://localhost:5000)1;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection ‘upgrade’;

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

**}**

* + Restart nginx configuration

$ sudo systemctl restart nginx

**# Checking server error logs**

sudo tail -n 20 /var/log/nginx/error.log

1. **Node.js (backend, frontend)**

* Installing node.js
  + Use PPA (Personal Package Archive)

$ cd ~

$ curl -sL [https://deb.nodesource.com/setup\_16.x -o nodesource\_setup.sh](https://deb.nodesource.com/setup_16.x%20-o%20nodesource_setup.sh)

$ sudo bash nodesource\_setup.sh

$ sudo apt-get install -y nodejs

* Preparing backend and frontend modules
  + Install npm modules for backend

$ cd ~/appadmin/backend

$ npm install

# getting build from frontend with command

sudo apt remove cmdtest

sudo apt remove yarn

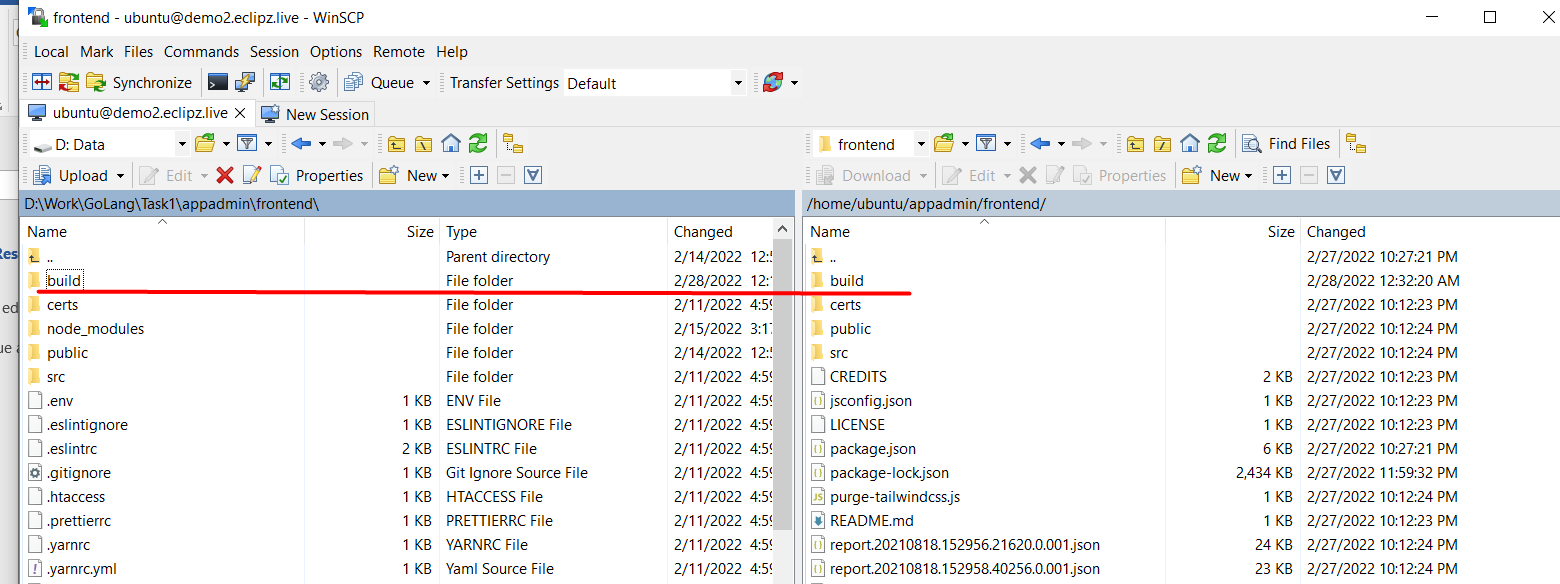
sudo npm install -g yarn

Cd ~/appadmin/frontend

Npm run build

# Can upload via ftp for build folder

* + Upload the locally built frontend files



* Using pm2 for managing the backend service
  + Installing pm2

$ sudo npm install pm2 -g

* + Running backend

$ cd ~/appadmin/backend

$ chmod 755 ./index.js

$ pm2 start ./index.js

* Updating code on frontend and backend
  + Build the frontend
  + Push to git repository
  + Pull at the server
  + Run 'pm2 restart 0'

UPDATE code from git.

cd appadmin

git pull

<Input git username and git token.>

Done

* These commands are for git pull issue.

git stash --include-untracked

git reset --hard

git clean -fd

git pull

-- auto start node app after reboot

sudo apt-get install cron

crontab -e

editor 🡪

@reboot sh -c 'cd **/home/ubuntu/appadmin/backend** && pm2 start index.js'