Deploy Mern Stack Project

AWS-EC2

It took me 3 days to figure out this problem.

Deploying only Nodejs app is very very easy as compared to Deploying MERN

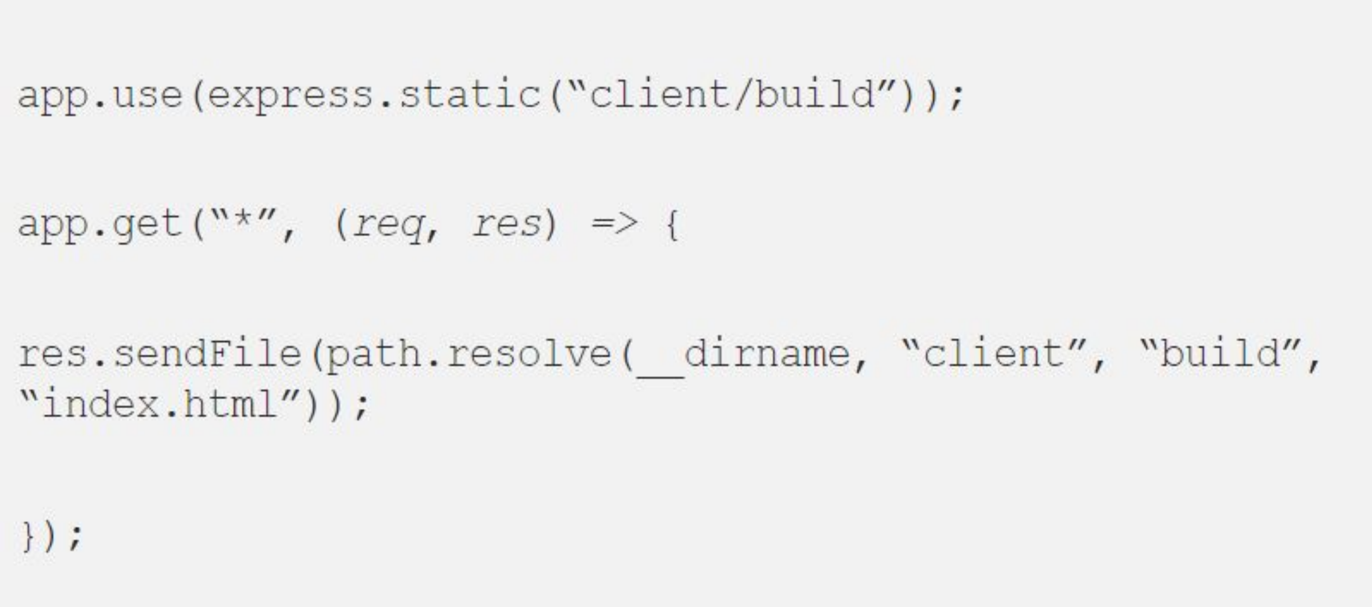
App using AWS-EC2 with NGINX

Some code snippets and mistakes that we generally cause.

1) if(node.env==prod)...

   This line is only used in case of heroku. Make sure to remove this line and add below lines

2)



```
app.use(express.static("client/build"));

app.get("*", (req, res) => {

res.sendFile(path.resolve(__dirname, "client", "build",
"index.html"));

});
```

3) No proxy will be used in react

4) use 3000||process.env.PORT

5) we will discuss later…

steps

```
sudo apt-get update
```

```
sudo apt-get install -y build-essential openssl libssl-dev pkg-config
```

```
sudo apt-get install -y nodejs
```

```
sudo apt-get install npm -y
sudo npm cache clean -f
```

```
sudo npm install -g n
```

```
sudo n stable
```

```
sudo apt-get install nginx -y
```

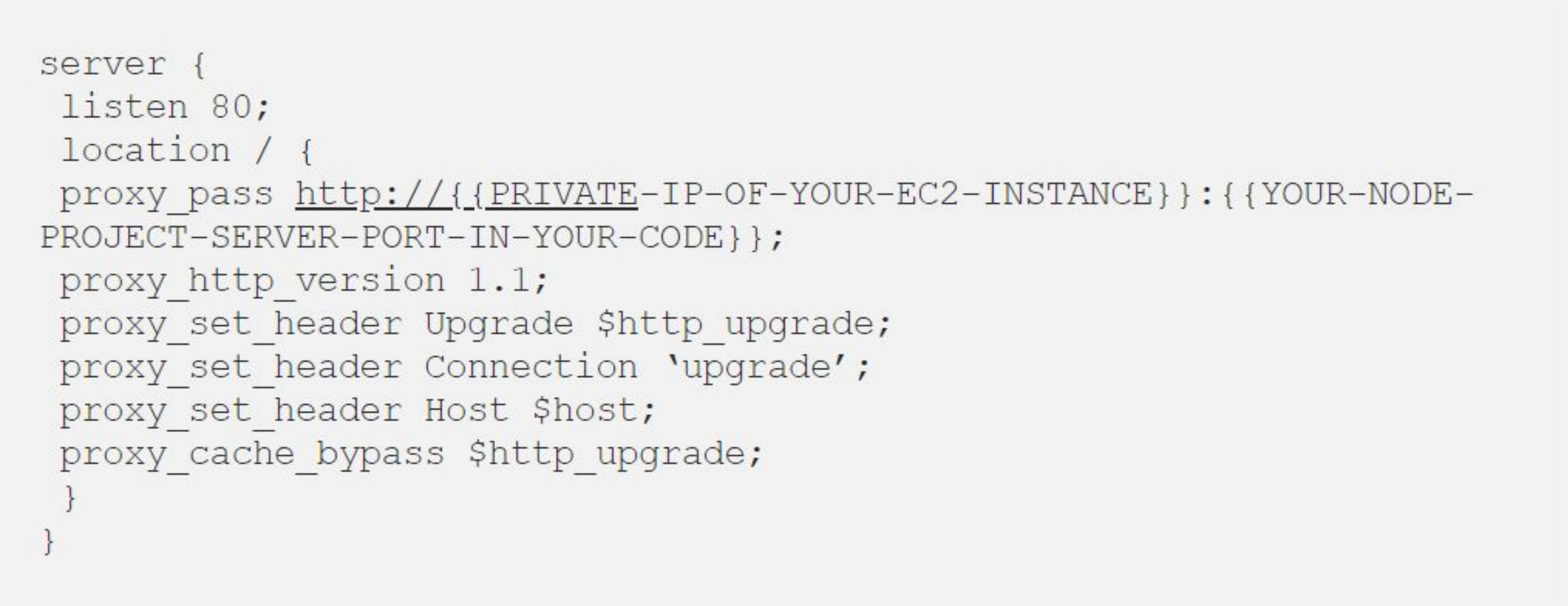
```
cd /var/www
```

- `sudo git clone {{your project file path on github/git}i.e http://githds...git}`
- **Link your project with Nginx**
- Type the following commands into your Ubuntu Server’s terminal,
- `cd /etc/nginx/sites-available`
- `sudo vim {{your cloned repository’s name}}`

- Add the following code to the file **(editing the placeholder text off-course)** and save the file by pressing the **esc** key and then typing **:wq** and then pressing the **return** key to save and close the vim editor.

```
server {
listen 80;
location / {
proxy_pass http://{{PRIVATE-IP-OF-YOUR-EC2-INSTANCE}}:{{YOUR-NODE-PROJECT-SERVER-PORT-IN-YOUR-CODE}};
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;
}
}
```

Like this:



```
server {
 listen 80;
 location / {
 proxy_pass http://{{PRIVATE-IP-OF-YOUR-EC2-INSTANCE}}:{{YOUR-NODE-
PROJECT-SERVER-PORT-IN-YOUR-CODE}};
 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;
  }
}
```

Link the **sites-enabled** with **sites-available** by typing the following in your Ubuntu server terminal,

```
sudo ln -s /etc/nginx/sites-available/{{your cloned
repository’s name}} /etc/nginx/sites-enabled/{{your
cloned repository’s name}}
```

Now type the following to remove the default file from the **Nginx sites-enabled** and **sites-available** directory,

```
sudo rm default
```

```
sudo rm /etc/nginx/sites-enabled/default
```

If you want to setup mongodb::

Don’t use if you are using atlas

https://itnext.io/deploy-a-mongodb-expressjs-reactjs-nodejs-mern-stack-web-application-on-aws-ec2-2a0d8199a682

Setup MongoDB (Note: Only follow this if you are not using an external MongoDB service like Mongo Atlas etc.)

Type the following in your Ubuntu Server’s terminal,

```
sudo apt-key adv — keyserver hkp://keyserver.ubuntu.com:80 — recv
9DA31620334BD75D9DCB49F368818C72E52529D4
echo “deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu
xenial/mongodb-org/4.0 multiverse” | sudo tee
/etc/apt/sources.list.d/mongodb-org-4.0.list


sudo apt install udo


sudo apt-get update


sudo apt-get install -y mongodb-org --allow-unauthenticated
```

Create a directory where your database files will be stored by typing the following in your Ubuntu Server's terminal,

```
sudo mkdir /data

sudo mkdir /data/db
```

Now start your MongoDB by typing the following in your Ubuntu Server's terminal,

```
sudo service mongod start
```

To configure automatic start of MongoDB when the server starts type the following in your Ubuntu Server's terminal,

```
sudo service mongod start
```

To configure automatic start of MongoDB when the server starts type the following in your Ubuntu Server’s terminal,

```
sudo systemctl enable mongod && sudo systemctl start mongod
```

Now for server we need pm2

Install pm2

Type the following in your Ubuntu server’s terminal to install pm2 globally,

```
sudo npm install pm2 -g
```

Now install node i.e. npm install for server

```
1.  cd /var/www/
2.  sudo chown –R ubuntu {{your cloned repository’s name}}
3.  cd {{your cloned repository’s name}}
4.  sudo npm install
```

Now install node i.e. npm install for client

```
1. cd client
2. sudo npm install
3. sudo npm run build
```

Everything’s setup!

Start pm2

```
1.  cd /var/www/{{your cloned repository’s name}}
2.  pm2 start server.js
```

Start nginx

```
1.  sudo service nginx stop && sudo service nginx start
```

Important commands

Nginx -t :::::: status

Pm2 status app.js

Pm2 restart app.js

Remember : reverse proxy is always private ipv4 address

Reference:
https://itnext.io/deploy-a-mongodb-expressjs-reactjs-nodejs-mern-stack-web-application-on-aws-ec2-2a0d8199a682

Thankyou