### FiverrMERNProj Setup

#### Step 1

Setup AWS EC2 instance with Ubuntu 18.04.2 LTS and http , https in the firewall

Setup Elastic IP to EC2 <a href="https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html">https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html</a>

Login to Instance via SSH

RUN "sudo -i" without quotes to switch to root

#### Step 2

#### **Install mongodb**

```
apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 9ECBEC467F0CEB10 apt-get install mongodb -y
```

#### Install libs required for nodejs and nodejs

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

#### Install npm, MERN Stack and Forever to run application as service

```
apt-get install npm -y

npm cache clean -f

npm install -g n

n stable

npm install -g mern-cli

npm install forever --global
```

#### Start mongodb service

service mongodb start

#### **Install git and clone the project**

```
apt-get install git
```

git clone <a href="https://github.com/DrewAtlas/FiverrMERNProj.git">https://github.com/DrewAtlas/FiverrMERNProj.git</a>

#### Setup Dependencies for project and run it

#### App1

cd /root/FiverrMERNProj/fvApp1

npm install

forever start nodeapp1start.js

#### App2

cd /root/FiverrMERNProj/fvApp2

npm install

forever start nodeapp2start.js

#### Check if node applications are running in the background

netstat -tulpn

tcp	0	0 0.0.0.0:3100	0.0.0.0:*	LISTEN	26047/node
tcp	0	0 0.0.0.0:4100	0.0.0.0:*	LISTEN	26080/node

#### **Install Nginx and setup reverse proxy**

apt-get install nginx -y

#For reverse proxy delete all contents in /etc/nginx/sites-available/default and add below config

Copy default config from repo nginx config

#### Finally restart nginx and test the urls

service nginx restart

#### **Results**

#### **Default Website**

http://3.13.57.74/

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to <u>nginx.org</u>. Commercial support is available at <u>nginx.com</u>.

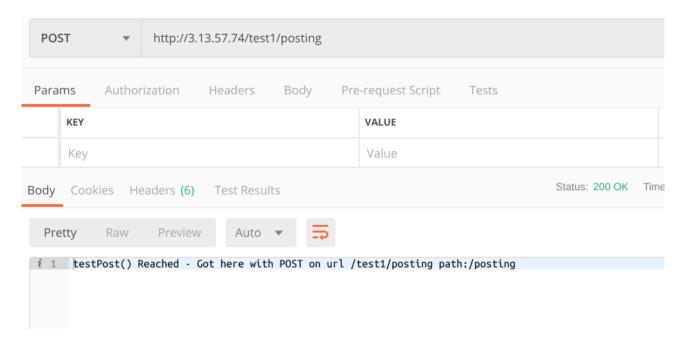
Thank you for using nginx.

#### App1

http://3.13.57.74/app1/test1/getting



testGet() Reached - Got here with GET on url /test1/getting path:/getting

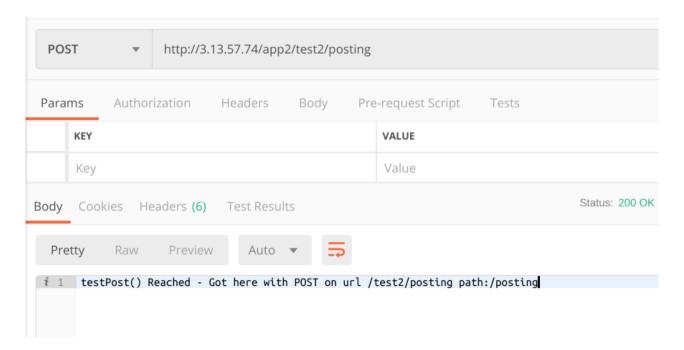


App2

#### http://3.13.57.74/app2/test2/getting



#### http://3.13.57.74/app2/test2/posting



Please note we need a domain name in order to use SSL without certificate warnings, I am submitting the order since its one day delivery and we need to work together to get the requirements fullfilled for SSL setup. I will give you ssl configurations for same once you setup a domain name and point it to IP 3.13.57.74

#### Mongo connection string to be used and its output

```
// Set up mongoose connection const dbBase = process.env.MONGOOSE_URL || "mongodb://aUser:aUserPw@127.0.0.1/"; // Put the database name and parameters separate so that I can use a different DB later... const dbName = process.env.MONGOOSE_DBNAME || "Demo"; const dbParams = process.env.MONGOOSE_PARAMS || "?retryWrites=true";
```

```
root@ip-172-31-30-29:~/FiverrMERNProj/fvApp1# npm start

> fvapp1@1.0.0 start /root/FiverrMERNProj/fvApp1
> node nodeapp1start.js

Trace|MS1100|Mongoose connection started

EZSS1000| NodeTest1 listening for Requests on port: 3100

Trace|MS1040|Mongoose connection: Connected to mongodb://aUser:aUserPw@127.0.0.1/Demo?retryWrites=true

Trace|MS1060|Mongoose Connection Open
```

#### How to access mongodb and create database username and password for database access

Type mongo in shell to access mongo shell

```
NongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.3
Server has startup warnings:
2019-06-19720:07:37.949+0000 I STORAGE [initandlisten]
2019-06-19720:07:39.083+0000 I CONTROL [initandlisten]
```

Run below code in mongo shell to create database with user password for the access of database

**Note:** replace dbname dbuser and dbpassword values as needed

#### use dbname

```
db.createUser({
    user: 'dbuser',
    pwd: 'dbpassword',
    roles: [{ role: 'readWrite', db:'dbname'}]
})
```

#### **SSL** config for nginx

#### Generate self signed cert and key

 $root@ip-172-31-30-29: \sim \# \ openssl\ req\ -x509\ -nodes\ -days\ 365\ -newkey\ rsa: 2048\ -keyout\ /etc/nginx/cert.key\ -out\ /etc/nginx/cert.crt$ 

Generating a RSA private key
.....+++++
writing new private key to '/etc/nginx/cert.key'

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

----

Country Name (2 letter code) [AU]:US

State or Province Name (full name) [Some-State]:USA

Locality Name (eg, city) []:USA

Organization Name (eg, company) [Internet Widgits Pty Ltd]:FiverrMERN

Organizational Unit Name (eg, section) []:IT

Common Name (e.g. server FQDN or YOUR name) []:3.13.57.74

Email Address []:srivishnu.k90@gmail.com

# Create new config file name default-ssl in /etc/nginx/sites-available/ add symbolic link to /etc/nginx/sites-enabled/default-ssl and

touch /etc/nginx/sites-available/default

ln -s /etc/nginx/sites-available/default-ssl /etc/nginx/sites-enabled/default-ssl

#include the default-ssl config in the git repo modify server\_name and the path to ssl certificate and key file

restart nginx