**Steps for setup.**

**Software required:**

To login into server you need private key access: name of private key is

ssh\_blockchain\_london.pem

**ssh\_blockchain\_london.pem FILE CAN BE found in the production.zip folder.**

Give 400 permissions by running

**sudo chmod 400 Prototype.pem**

**ssh -i Prototype.pem ubuntu@** 3.131.82.71

**You will login into the server via ssh shell.**

**Run below commands**

sudo apt install nginx -y

sudo apt install unzip -y

sudo apt update

sudo apt install nodejs -y

sudo apt install npm –y //tool to download load packages

sudo npm cache clean -f

sudo npm install -g n

sudo n stable

curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -

echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list

sudo apt update

sudo apt install elasticsearch

sudo npm install -g ganache-cli

ganache-cli -m "myth like bonus scare over problem client lizard pioneer submit female collect" --db network\_bc\_ganache1 -p 7545

ganache-cli -m "blanket endless master abandon praise spoil people mixed sauce impulse toss auto" --db network\_bc\_ganache2 -p 7546

Moving the code to server.

We have entire code in production.zip, on unzipping it has 3 folders: backend and blockchain, nginx folder.

**production**

**backend** (this has code for frontend and rest api for io sensors data and other urls for front end dashboard)

**blockchain** (connects to blockchain and adds data to 2 blockchains and also saves data in elastic search)

nginx (which has ssl keys and nginx conf files)

TO transfer files to server use scp.

**scp –i Prototype.pem production.zip ubuntu@**3.131.82.71**:/home/ubuntu**

Then in server after ssh, u can find production.zip

Run below commands,

**unzip production.zip**

**cd blockchain**

Once you run both blockchains, you need to edit .env file in blockchain, .env file has below variables,

production=true

EXPRESS\_IP=127.0.0.1

EXPRESS\_PORT=8040

NODE\_URL1="http://127.0.0.1:7546"

DEPLOYED\_CONTRACT\_ADDRESS\_1="0xbF68875C14cB22d85D1819CD1554757775832295"

ACCOUNT\_ADDRESS\_1="0x7482EDb31Cb9be3f55E1D2CF3A4c38Fd059c07D2"

PRIVATE\_KEY\_GANACHE\_1="dc5ec7262b6b164bc5d2990b75e68c56e740b1749467f0f249c51d6e843993d8"

NODE\_URL2="http://127.0.0.1:7545"

**DEPLOYED\_CONTRACT\_ADDRESS\_2**="0x532Cc9fE390deB6685007Ee87afAC28D29EC553e"

ACCOUNT\_ADDRESS\_2="0x8FfcAbcA373861f87F90F152d9a650A131C1273F"

PRIVATE\_KEY\_GANACHE\_2="4baeb65a0797359664c9263fb8776d1ddf3d741ae958cf9e9ddc208533221e92"

MAX\_TEMPETATURE=20

MAX\_HUMIDITY=58

You need to edit DEPLOYED\_CONTRACT\_ADDRESS\_1 (blockchain network 1 smart contract deployed address)

DEPLOYED\_CONTRACT\_ADDRESS\_2 (blockchain 2)

Now you need to edit the deployed contract address.

Lets deploy smart contracts, run below commands, to deploy on network 1

npx hardhat run scripts/deploy.js --network ganache1

NOW, the terminal will return output for contract address which looks like

**“0x…….”,**

**Copy the contract address and save it under** DEPLOYED\_CONTRACT\_ADDRESS\_1

SImilarly deploy contract on network 2 using below command

npx hardhat run scripts/deploy.js --network ganache2

**Copy the contract address and save it under** DEPLOYED\_CONTRACT\_ADDRESS\_2

**Now stop blockchains, press control C. (DO NOT PRESS CONTROl Z at all)**

TO run blockchains networks in background run below commands, so there will run forever until you stop them by killing process.

**Start the blockchain**

nohup ganache-cli -m "myth like bonus scare over problem client lizard pioneer submit female collect" --db network\_bc\_ganache1 -p 7545 &

nohup ganache-cli -m "blanket endless master abandon praise spoil people mixed sauce impulse toss auto" --db network\_bc\_ganache2 -p 7546 &

TO start blockchain node service, u need to run app.js

Run node app.js

To run in background forever use

nohup node app.js &

This app.js will talk with blockchains and elastic search.

Return back to home (using cd .. run 2 times and check home directory using pwd. Pwd must return /home/ubuntu)

Now configure nginx, to ensure admin or usercan access

<https://blockchain.venkateshupadrista.com>

Production folder which you moved earlier will have nginx folder too.

Go to production/nginx

cd production/nginx

sudo rm -rf /etc/nginx/nginx.conf

sudo mv nginx.conf /etc/nginx

sudo mv ssl /etc/nginx/

sudo systemctl restart nginx

Now ngnx is configured with your domain.<https://blockchain.venkateshupadrista.com>

If domain is changed, then you need to edit nginx.conf, at

server\_name blockchain.venkateshupadrista.com;

if ($http\_x\_forwarded\_proto = "http") {

return 301 https://blockchain.venkateshupadrista.com$request\_uri;

}

Edit the server name and replace [blockchain.venkateshupadrista.com](https://blockchain.venkateshupadrista.com)

(domain) with any new domain.

And

Replace return 301 <https://blockchain.venkateshupadrista.com>$request\_uri;

With your own domain.

Ensure you change ssl certificates accordingly, you need to generate ssl certificates for the new domain, full pem chain is required, (full public keys from root to website public cert pem keys required)

Save them exactly as private.key for **private key** and for full pem chain (public cert) save it as **ca\_fullpem.pem**

Now go to backend folder in production.

cd ..

cd backend

Run node app.js

To run in background forever use

nohup node app.js &

Now the entire configuration is ready, login from the website.

Use the below credentials for login.

<https://blockchain.venkateshupadrista.com>

[testuser@gmail.com](mailto:testuser@gmail.com)

QWERTY1234!!@@##