Deployment Guide

Prerequisite:

1. Ubuntu OS (We will be using EC2 service from AWS)
    a. NGINX
       Link to installation guide:
       https://ubuntu.com/tutorials/install-and-configure-nginx#2-installing-nginx
    b. Node.js
       Link to installation guide:
       https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-ubuntu-20-04
    c. PM2
       Link to installation guide:
       https://pm2.keymetrics.io/
    d. GIT
       Link to installation guide:
       https://www.digitalocean.com/community/tutorials/how-to-install-git-on-ubuntu-20-04
2. MongoDB (We will be using DOCUMENTDB service from AWS)
    a. We will create a DocumentDB cluster in AWS using the https://eu-west-1.console.aws.amazon.com/docdb/home?region=eu-west-1#cluster-create
    b. After creating DocumentDB cluster in AWS, we will get a connection string which we have to add into the environment variables of our backend application.
       Sample connection string::

```
mongodb://sarafuio:<insertYourPassword>@docdb-2023-03-09-16-03-46.cluster-c87an2pwav72.eu-west-1.docdb.amazonaws.com:27017/?replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false
```

3. QLDB (We will be using QLDB service from AWS)
    a. We will create a QLDB instance in AWS using the https://eu-west-1.console.aws.amazon.com/qldb/home?region=eu-west-1#create-ledger
    b. After creating QLDB instance in AWS, we will get a ledger name which we have to add into the environment variables of our backend application.

Step to deploy Backend and Frontend:

Step 1: Open link to the git project: **https://github.com/kfarrelly/medicalv2.git**

Step 2: Clone the project and checkout the branch bdev-changes

→ git clone **https://github.com/kfarrelly/medicalv2.git** -b bdev-changes

Step 3: Install node_modules via NPM

Frontend dependencies installation ::

→ cd <PROJECT_ROOT>

→ npm install

Backend dependencies installation ::

→ cd <PROJECT_ROOT>/backend

→ npm install

Step 4: Create build using the command

Frontend::

→ cd <PROJECT_ROOT>

→ npm run build

Backend::

→ cd <PROJECT_ROOT>/backend

→ npm run start

Step 5: Serve the created builds

Frontend:: We need to configure NGINX server to serve the created build, we can use the following documentation.
https://www.digitalocean.com/community/tutorials/how-to-set-up-nginx-server-blocks-virtual-hosts-on-ubuntu-16-04

Backend:: We can manage the backend node process using PM2, so after creating the build we need to serve it through the pm2.

Command:: pm2 start npm –name <PROCESS_NAME> –cwd <ABSOLUTE_PATH_OF_PROEJECT> – run start

Note*: ‘ – ’ stands for double hyphens.

Sample Command:: pm2 start npm –name “sarafuio-api” –cwd “/var/www/sarafuio/backend” – run start

Now your api is live on the port written in .env and can be access as <SERVER_IP>:<PORT>

Now we need to create a virtual server block for backend application on nginx server config and add reverse proxy configuration for the api domain using the following guide.
https://www.digitalocean.com/community/tutorials/how-to-set-up-nginx-server-blocks-virtual-hosts-on-ubuntu-16-04

***Note:** **The .env file is committed and saved in the project repository. In the .ENV file the values of LEDGER_NAME and MONGO_DB will be updated.**

DOTENV (.env) file:

AccessKey=AKIAIWFI4OEFQNUIDQ2A

SecretKey=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
Region=eu-west-1
Port=10000
LEDGER_NAME=bdev-medichain
MONGO_DB=mongodb://sarafuio:A3H7Vgn2c3g2@docdb-2023-03-09-16-03-46.cluster-c87an2pwav72.eu-west-1.docdb.amazonaws.com:27017/MedicineDB?replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false