Deploying .Net Core Api, Service and React App on Ubuntu with NGINX

Installing Requirements

Development computer :

- ssh (to connect to deployment server remotely)
- scp (to copy your build application to deployment server remotely)

Suggestions : you can use *git bash*

Deployment server :

- openssh-server (to be connected remotely with ssh)

```
sudo apt-get install openssh-server
sudo systemctl status ssh
```

- ufw (to configure firewall)

```
sudo apt-get install ufw
```

- Nginx (proxy server)

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

- Dotnet core runtime (to run built core app)

```
wget
https://packages.microsoft.com/config/ubuntu/18.04/packages-microsoft-prod.
deb -O packages-microsoft-prod.deb
sudo dpkg -i packages-microsoft-prod.deb
sudo apt-get update;
sudo apt-get install -y apt-transport-https
sudo apt-get update
sudo apt-get install -y aspnetcore-runtime-3.1
```

- Mssql-server (mssql DB)

```
wget -qO- https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
sudo add-apt-repository "$(wget -qO-
https://packages.microsoft.com/config/ubuntu/18.04/mssql-server-2019.list)"
sudo apt-get update
sudo apt-get install -y mssql-server
sudo /opt/mssql/bin/mssql-conf setup
```

- Npm (for react)

```
curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
sudo apt install nodejs
```

- Mssql-tools (optional)

```
curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
curl https://packages.microsoft.com/config/ubuntu/18.04/prod.list | sudo
tee /etc/apt/sources.list.d/msprod.list
sudo apt-get update
sudo apt-get install mssql-tools unixodbc-dev
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash_profile
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc
source ~/.bashrc
```

Configure firewall

```
sudo ufw enable
sudo ufw allow ssh
sudo ufw allow "Nginx Full"
sudo ufw allow "Nginx HTTP"
sudo ufw allow "Nginx HTTPS"
```

Assumptions

- Development
    - Windows 10
    - All projects in *D:* drive

- Deployment
    - ubuntu 18.04
    - All our built folder resides in `/var/www/cefalo-all-apps`

Building Applications

(N.B. please everytime go to your specified project first)

- Backend Api

```
dotnet publish -c Release -o D:/cefalo-all-apps/cefalo-hrportal-api
```

- Sync Service

```
dotnet publish -c Release -o
D:/cefalo-all-apps/cefalo-hrportal-sync-service
```

- Frontend React app
    - in src\config.json.js file please paste this content

```
const BASE_URL = "https://hrportal.cefalolab.com";
export const config = {
REACT_APP_GOOGLE_CLIENT_ID:"17849993230-4hgid40cv298seg554vc9kofv3aqoi8s.apps.googleusercontent.com",
API_BASE_URL: `${BASE_URL}/api`, PROFILE_IMAGE_BASE_URL:
`${BASE_URL}/api/avatars`,
};
```

```
npm run build
```

- move content of your build folder to *D:/cefalo-all-apps/cefalo-hrportal-ui*

Secure Copy to server

(N.B. go to d: drive first)

```
scp -r cefalo-all-apps  username@ip:/var/www/
```

Server

- Establish remote connection

```
ssh server_username@server_ip_address
```

- Change permission of of our working directory (giving all access , you can control access)

```
sudo chmod -R  777 /var/www/cefalo-all-apps
```

- Make required directory for Api storage

```
cd /var/www/cefalo-all-apps/cefalo-hrportal-api
mkdir -p App_Data/TemporaryUploads
sudo chmod -R  777 App_Data/TemporaryUploads
mkdir -p Media/avatars
sudo chmod -R  777 Media/avatars
```

- Copy Photos to Avatars folder (from ~/photos)

```
Cp -rm ~/photos/. Media/avatars/
```

- If you copy all profile pictures in avatars folder and then run `sudo chmod -R  777 Media/avatars`

- Nginx Configuration as reverse proxy :

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

Then paste this and save

```
server {
    listen              5001 default_server;
    server_name         www.example.com;

    location /api {
        proxy_pass         http://localhost:5000;
        proxy_http_version 1.1;
        proxy_set_header   Upgrade $http_upgrade;
        proxy_set_header   Connection keep-alive;
        proxy_set_header   Host $host;
        proxy_cache_bypass $http_upgrade;
        proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header   X-Forwarded-Proto $scheme;
    }

    location / {
        root /var/www/cefalo-all-apps/cefalo-hrportal-ui;
        index index.html index.htm;
        location / {
             try_files $uri /index.html;
        }
    }
}
```

Check if nginx configuration is okey

```
sudo nginx -t
```

- Creating api service :

```
sudo nano /etc/systemd/system/cefalo-hrportal-api.service
```

Then paste this and save

```
[Unit]
Description=Example .NET Web API App running on Ubuntu

[Service]
WorkingDirectory=/var/www/cefalo-all-apps/cefalo-hrportal-api
ExecStart=/usr/bin/dotnet
```

```
/var/www/cefalo-all-apps/cefalo-hrportal-api/Cefalo.AttendanceManagement.Api.dll
Restart=always
RestartSec=10
KillSignal=SIGINT
SyslogIdentifier=dotnet-example
User=www-data
Environment=ASPNETCORE_ENVIRONMENT=Production
Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false

[Install]
WantedBy=multi-user.target
```

- Creating Sync Service :

```
Sudo nano /etc/systemd/system/cefalo-hrportal-sync-service.service
```

```
[Unit]
Description=attendace data sync

[Service]
Type=notify
WorkingDirectory=/var/www/cefalo-all-apps/cefalo-hrportal-sync-service/
ExecStart=/usr/bin/dotnet
/var/www/cefalo-all-apps/cefalo-hrportal-sync-service/Cefalo.AttendanceSyncService.dll
Restart=always
RestartSec=10
[Install]
WantedBy=multi-user.target
```

- Now Run all Services that eventually run your applications

```
sudo systemctl daemon-reload
sudo systemctl restart cefalo-hrportal-api.service
cefalo-hrportal-sync-service.service nginx.service mssql-server.service
```

- Check all our service status

```
sudo systemctl status cefalo-hrportal-api.service
```

```
cefalo-hrportal-sync-service.service nginx.service mssql-server.service
```

Customization

- Change the database connection string as your need in `AppSettings.json`
- point to api url from your front end application in `config.json.js`

References

https://docs.microsoft.com/en-us/aspnet/core/host-and-deploy/linux-nginx?view=aspnetcore-3.1