

66543210020-4

LAB 3

Reactjs

```
devuser@server:~$ cd /var/www
devuser@server:/var/www$ sudo mkdir lab3
devuser@server:/var/www$ sudo chown -R $USER:$USER /var/www/lab3
devuser@server:/var/www$ sudo nano /etc/apache2/sites-available/lab3.conf
devuser@server:/var/www$ cd lab3
devuser@server:/var/www/lab3$ npx create-react-app my-react-app
```

- สร้าง directory lab3
- จากนั้นกำหนด owner ให้ user ใช้ lab3 ได้
- จากนั้น cd เข้าโฟลเดอร์
- เริ่มสร้างโฟลเดอร์ reactjs

```
npm start
  Starts the development server.

npm run build
  Bundles the app into static files for production.

npm test
  Starts the test runner.

npm run eject
  Removes this tool and copies build dependencies, configuration files
  and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd my-react-app
  npm start

Happy hacking!
devuser@server:/var/www/lab3$
```

พอได้แล้วลองทดสอบว่ารันได้ไหม

Compiled successfully!

You can now view **my-react-app** in the browser.

Local: http://localhost:3000

On Your Network: http://10.211.55.4:3000

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled **successfully**

^C

devuser@server:/var/www/lab3/my-react-app\$ nano package.json

devuser@server:/var/www/lab3/my-react-app\$

จากนั้นออกมา แล้วเปลี่ยนหมายเลข port จาก 3000 เป็น 3001

โดยใช้คำสั่ง nano package.json

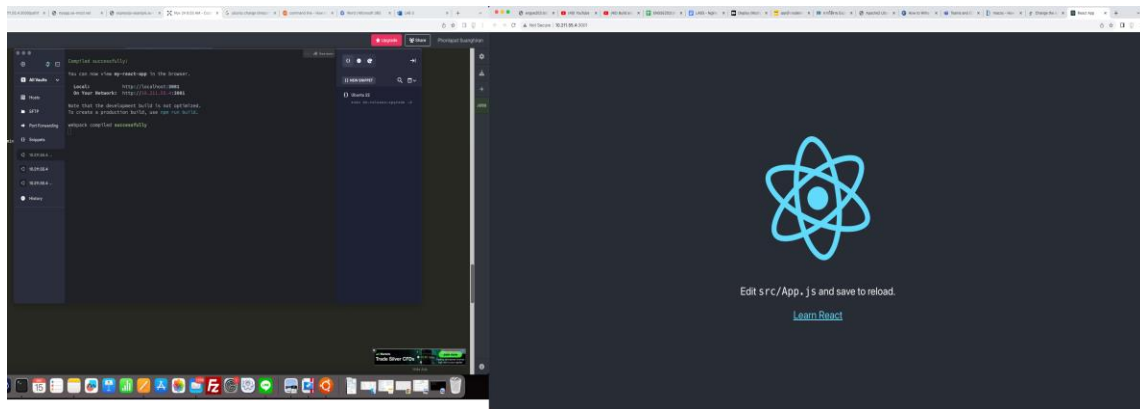
```
GNU nano 6.2 package.json
{
  "name": "my-react-app",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.17.0",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "PORT=3001 react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  }
}
```

เติมในช่อง **start** ด้านล่าง เข้าไปเพิ่ม **PORT=3001** ไปยังด้านหน้า **react-scripts**

*** ก่อนจะรัน ***

- จากนั้นใช้คำสั่ง `ufw allow 3001` อนุญาตให้รันโดยไม่ติด firewall ได้

จากนั้นออกมา ลองทดสอบว่ารันได้ไหม



เสร็จแล้ว จากนั้นมา `cd /etc/nginx/sites-available`

ทำการก๊อปปี้ตัว default ที่ใช้ได้มา

```
devuser@server:/etc/nginx/sites-available$ sudo cp myapp my-react-app
devuser@server:/etc/nginx/sites-available$ sudo nano my-react-app
```

แล้วเข้าไปกำหนดเลข port และชื่อโดเมน

```
GNU nano 6.2 my-react-app *
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 80;
    listen [::]:80;
    server_name my-react-app.se-rmutl.net;
    location / {
        proxy_pass http://127.0.0.1:3001; # !!! - change to your app port
        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;
    }

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    #
    # See the nginx documentation for more details and configuration
    # options.
}
```

เสร็จแล้วก็ restart nginx ใหม่ แล้ว test ดูว่าผ่านไหม

จากนั้นก็เข้า terminal

```
devuser@server:/etc/nginx/sites-available$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
devuser@server:/etc/nginx/sites-available$ sudo systemctl restart nginx
devuser@server:/etc/nginx/sites-available$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

เปิด terminal แล้ว cd เข้าไปที่ /etc/hosts

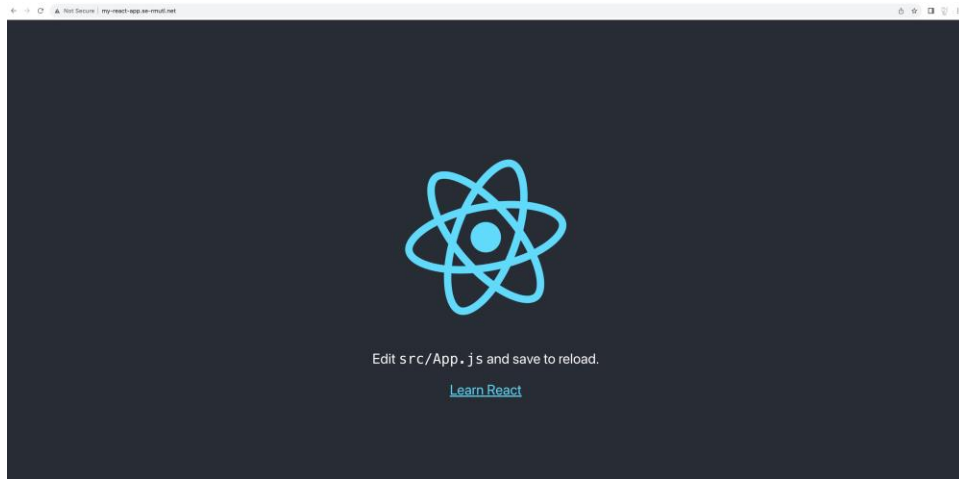
เข้าไปเพิ่มโดเมนใน ip เพื่อให้ mac สามารถเปิดโดเมนนี้ได้



```
UW PICO 5.09                               File: /etc/hosts

##
# Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting. Do not change this entry.
##
127.0.0.1        localhost engse203.lab1
255.255.255.255 broadcasthost
::1             localhost
10.211.55.4      myapp.se-rmutl.net expressjs-example.se-rmutl.net
10.211.55.4      my-react-app.se-rmutl.net ubuntu-next-app.se-rmutl.net
10.211.55.4      ubuntu-linux-lab.shared ubuntu-linux-lab #prl_hostonly shared

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Pg   ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where is  ^V Next Pg   ^U UnCut Text ^T To Spell
```



http://my-react-app.se-rmutl.net/

Nextjs

```
devuser@server:~$ cd /var/www/lab3
devuser@server:/var/www/lab3$ npx create-next-app@latest ubuntu-next-app
Need to install the following packages:
create-next-app@14.0.4
Ok to proceed? (y) y
✓ Would you like to use TypeScript? ... No / Yes
✓ Would you like to use ESLint? ... No / Yes
✓ Would you like to use Tailwind CSS? ... No / Yes
✓ Would you like to use `src/` directory? ... No / Yes
✓ Would you like to use App Router? (recommended) ... No / Yes
✓ Would you like to customize the default import alias (@/*)? ... No / Yes
Creating a new Next.js app in /var/www/lab3/ubuntu-next-app.

Using npm.

Initializing project with template: app-tw

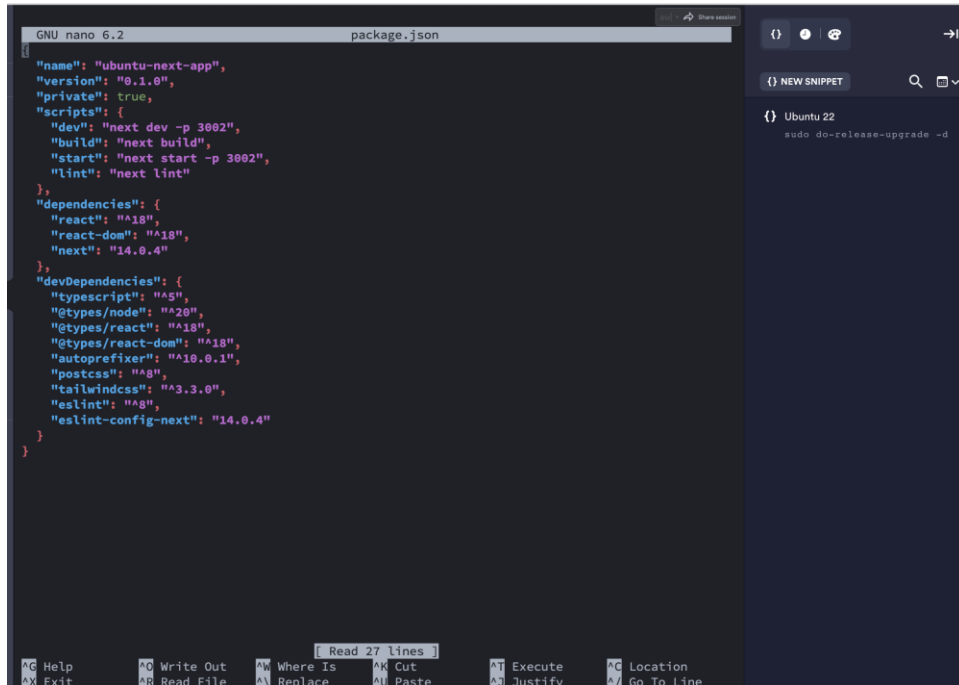
Installing dependencies:
- react
- react-dom
- next

Installing devDependencies:
- typescript
- @types/node
- @types/react
- @types/react-dom
- autoprefixer
- postcss
- tailwindcss
- eslint
- eslint-config-next

( ) :: idealTree:ubuntu-next-app: sill idealTree buildDeps
```

สร้างเฟลเตอร์ Nextjs

Nano เข้า package.json ของ Nextjs เพื่อเข้าไปเปลี่ยนเลข Port



```
GNU nano 6.2 package.json
{
  "name": "ubuntu-next-app",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "dev": "next dev -p 3002",
    "build": "next build",
    "start": "next start -p 3002",
    "lint": "next lint"
  },
  "dependencies": {
    "react": "^18",
    "react-dom": "^18",
    "next": "14.0.4"
  },
  "devDependencies": {
    "typescript": "^5",
    "@types/node": "^20",
    "@types/react": "^18",
    "@types/react-dom": "^18",
    "autoprefixer": "^10.0.1",
    "postcss": "^8",
    "tailwindcss": "^3.3.0",
    "eslint": "^8",
    "eslint-config-next": "14.0.4"
  }
}
```

เปลี่ยนเลข port เป็น 3002 ใน package.json ของ Nextjs

- จากนั้นใช้คำสั่ง `ufw allow 3002` อนุญาตให้รันโดยไม่ติด firewall ได้

ลองทดสอบว่ารันได้ไหม

```
devuser@server:/var/www/lab3/ubuntu-next-app$ nano package.json
devuser@server:/var/www/lab3/ubuntu-next-app$ npm run dev

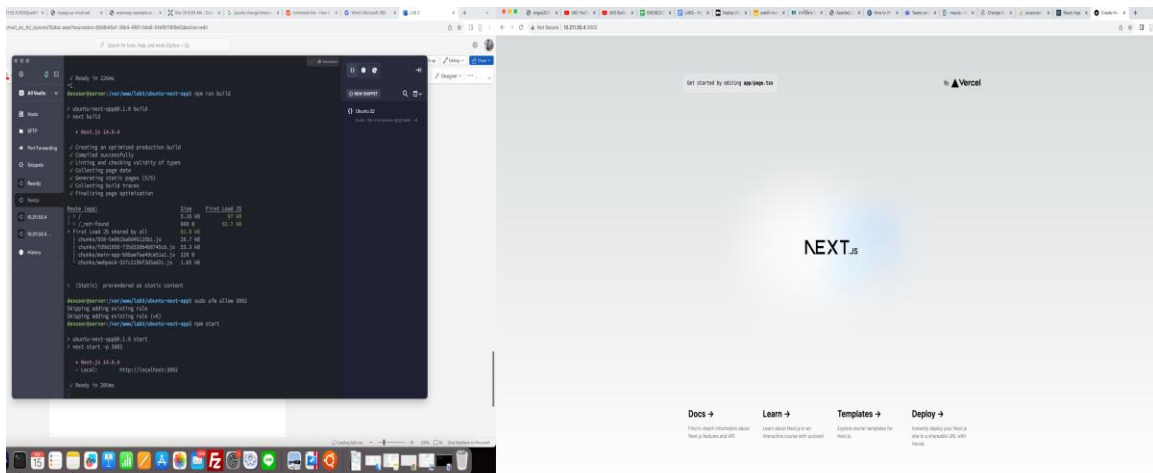
> ubuntu-next-app@0.1.0 dev
> next dev -p 3002

  ▲ Next.js 14.0.4
  - Local:      http://localhost:3002

✓ Ready in 2.6s
^C

devuser@server:/var/www/lab3/ubuntu-next-app$ sudo ufw allow 3002
[sudo] password for devuser:
Rule added
Rule added (v6)

devuser@server:/var/www/lab3/ubuntu-next-app$ npm run build
```

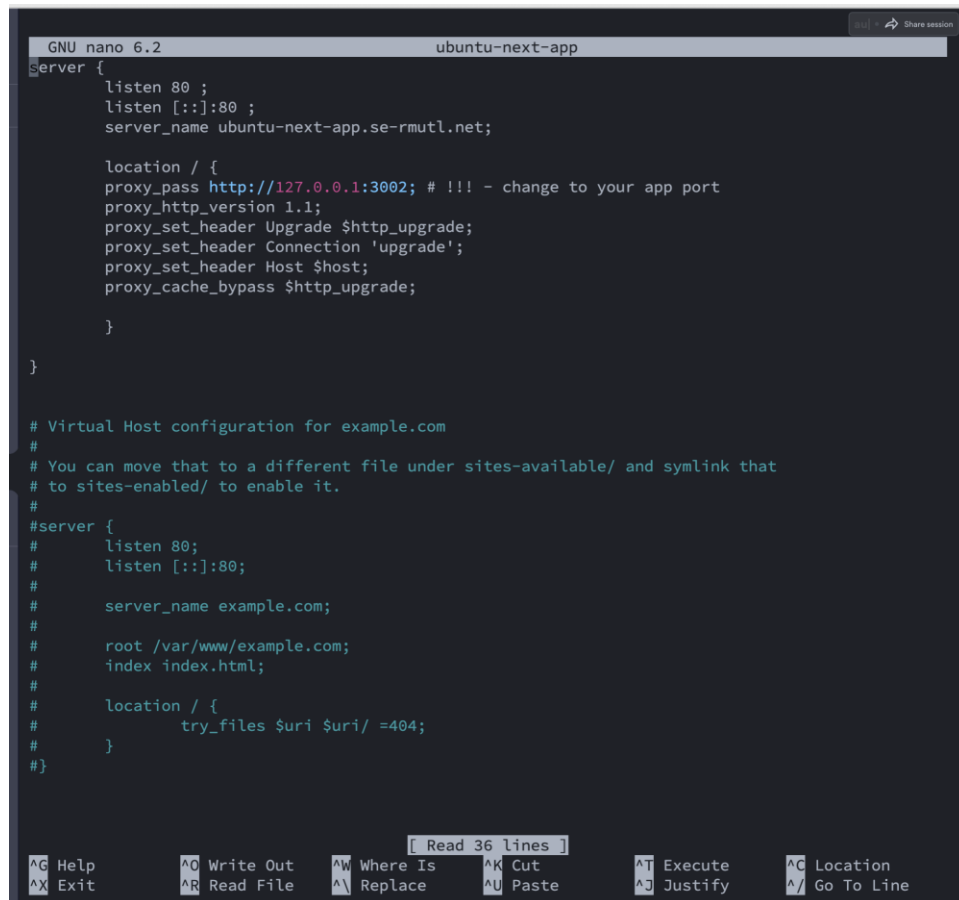


จากนั้นมา `cd /etc/nginx/sites-available`

ทำการก๊อปปี้ตัว default ที่ใช้ได้มา

```
devuser@server:/etc/nginx/sites-available$ sudo cp myapp ubuntu-next-app
devuser@server:/etc/nginx/sites-available$ sudo nano ubuntu-next-app
```


แล้วเข้าไปกำหนดเลข port และชื่อโดเมน



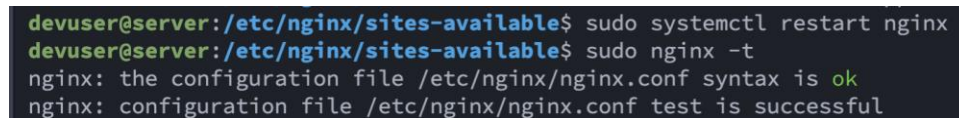
```
GNU nano 6.2 ubuntu-next-app
server {
    listen 80 ;
    listen [::]:80 ;
    server_name ubuntu-next-app.se-rmutl.net;

    location / {
        proxy_pass http://127.0.0.1:3002; # !!! - change to your app port
        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;
    }
}

# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/ and symlink that
# to sites-enabled/ to enable it.
#
#server {
#    listen 80;
#    listen [::]:80;
#
#    server_name example.com;
#
#    root /var/www/example.com;
#    index index.html;
#
#    location / {
#        try_files $uri $uri/ =404;
#    }
#}
```

เสร็จแล้วก็ restart nginx ใหม่ แล้ว test ดูว่าผ่านไหม

จากนั้นก็เข้า terminal



```
devuser@server:/etc/nginx/sites-available$ sudo systemctl restart nginx
devuser@server:/etc/nginx/sites-available$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

เปิด terminal แล้ว cd เข้าไปที่ /etc/hosts

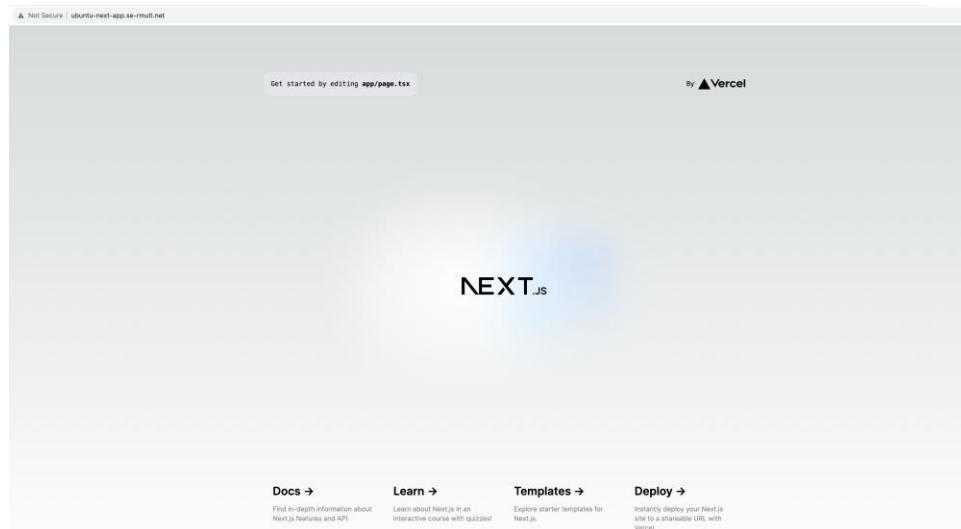
เข้าไปเพิ่มโดเมนใน ip เพื่อให้ mac สามารถเปิดโดเมนนี้ได้

```
rocket — pico — sudo — 80x24
UW PICO 5.09 File: /etc/hosts

##
# Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting. Do not change this entry.
##
127.0.0.1    localhost engse203.lab1
255.255.255.255 broadcasthost
::1         localhost
10.211.55.4  myapp.se-rmutl.net expressjs-example.se-rmutl.net
10.211.55.4  my-react-app.se-rmutl.net ubuntu-next-app.se-rmutl.net
10.211.55.4  ubuntu-linux-lab.shared ubuntu-linux-lab #prl_hostonly shared

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Pg  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where is  ^V Next Pg  ^U UnCut Text ^T To Spell
```

จากนั้นลองรันในชื่อโดเมนล่าสุด



<http://ubuntu-next-app.se-rmutl.net/>

PM2

ทำการติดตั้ง pm2 ในไฟล์

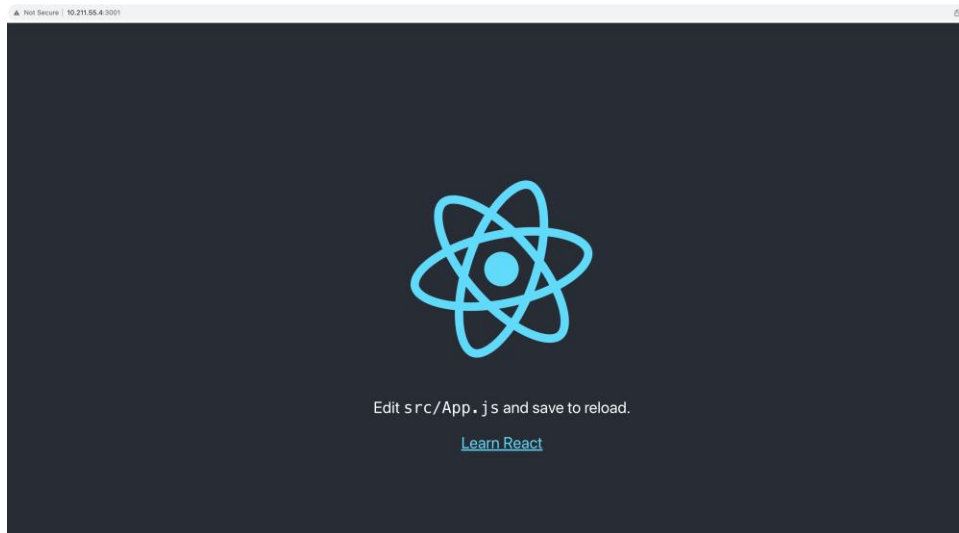
จากนั้นทำการ cd เข้าโฟลเดอร์ของไฟล์ที่จะรัน

```
devuser@server:/var/www/lab3$ cd my-react-app
devuser@server:/var/www/lab3/my-react-app$ pm2 start --name my-react-app npm -- start
```

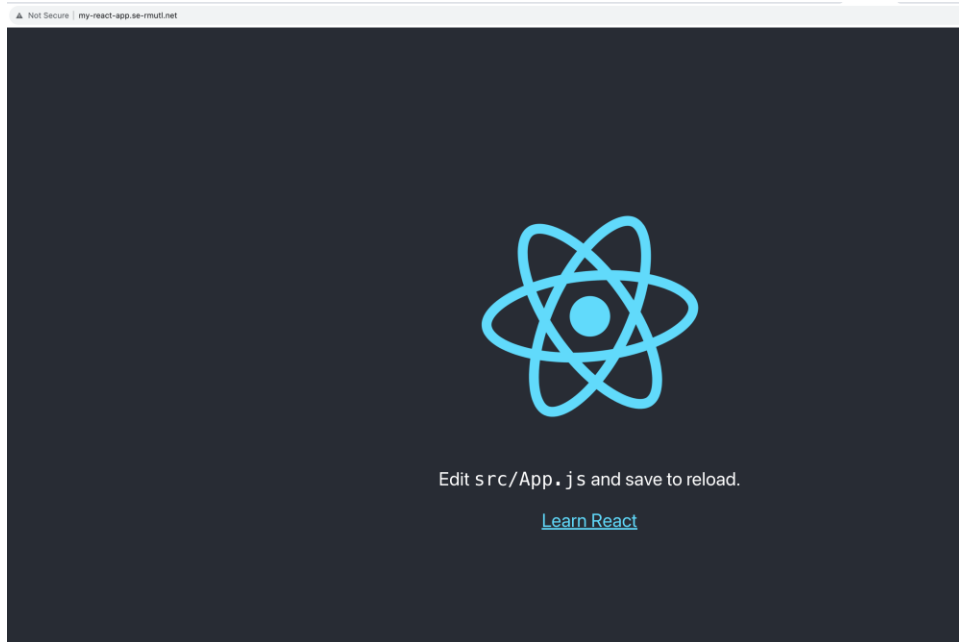
แล้วทำการลองรันด้วยคำสั่งของ pm2

```
[PM2] Starting /usr/bin/npm in fork_mode (1 instance)
[PM2] Done.
```

id	name	mode	๗	status	cpu	memory
0	my-react-app	fork	0	online	0%	16.9mb



<http://10.211.55.4:3001/>

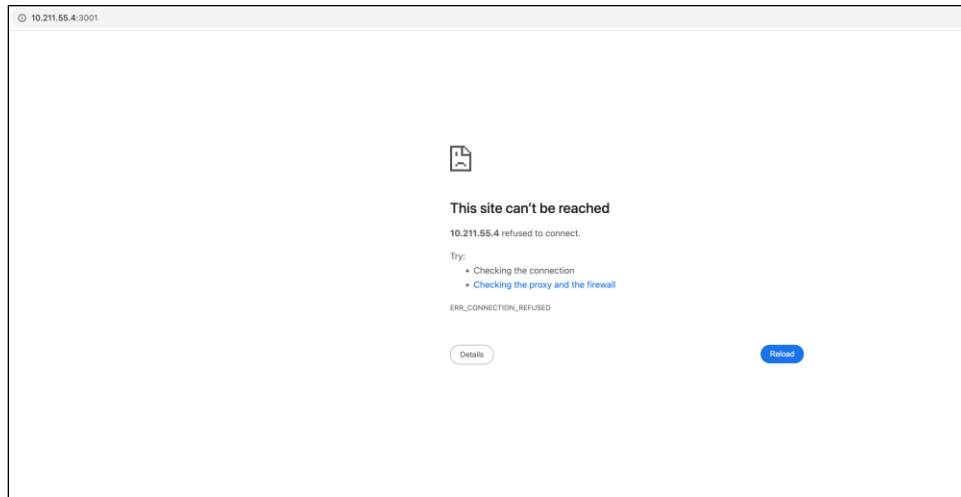


<http://my-react-app.se-rmutl.net/>

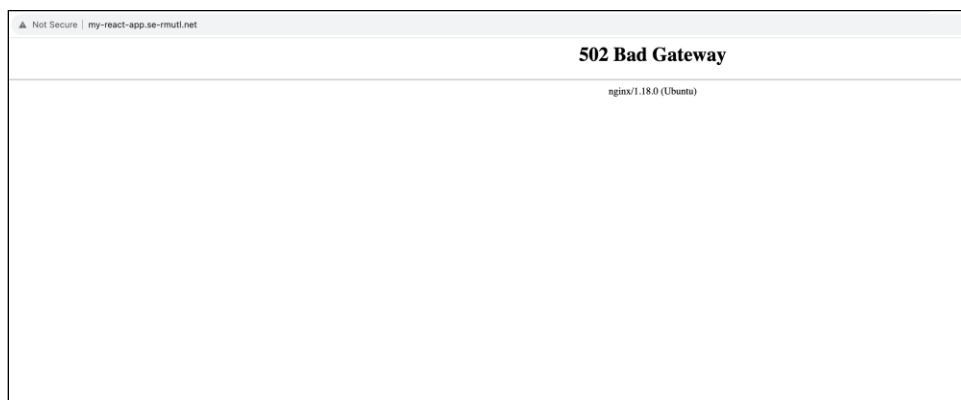
จากนั้นลองทดสอบหยุดรัน

```
devuser@server:/var/www/lab3/my-react-app$ pm2 stop my-react-app
[PM2] Applying action stopProcessId on app [my-react-app](ids: [ 0 ])
[PM2] [my-react-app](0) ✓
```

id	name	mode	๓	status	cpu	memory
0	my-react-app	fork	0	stopped	0%	0b



<http://10.211.55.4:3001/>



<http://my-react-app.se-rmutl.net/>

จากนั้นลองเปลี่ยนชื่อโดยใช้คำสั่ง

```
devuser@server:/var/www/lab3/my-react-app$ pm2 start my-react-app --name "Reactjs"
```

แสดงผลในตาราง pm2

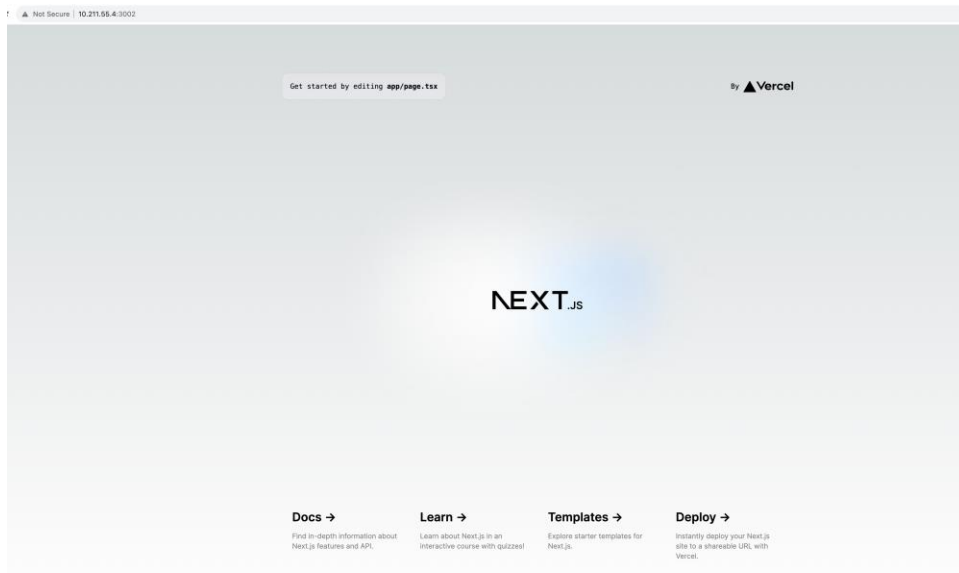
0	Reactjs	fork	2	stopped	0%	0b
---	---------	------	---	---------	----	----

ทำกับ NextJs ด้วยเหมือนกัน

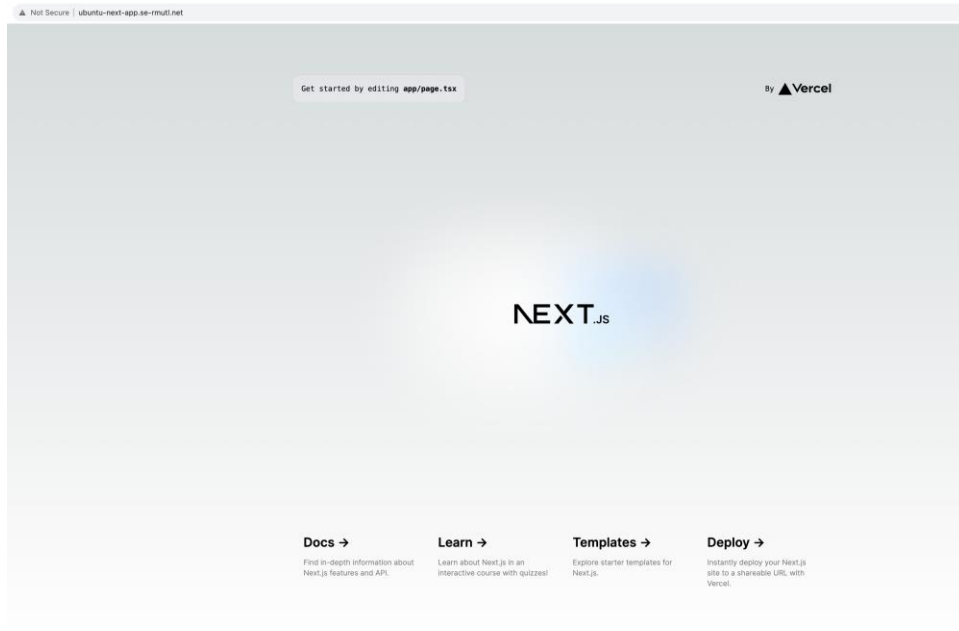
ผลลัพธ์ที่ได้

Start

id	name	mode	๗	status	cpu	memory
2	Nextjs	fork	1	online	0%	61.5mb
๐	Reactjs	fork	2	stopped	0%	๐b



http://10.211.55.4:3002/

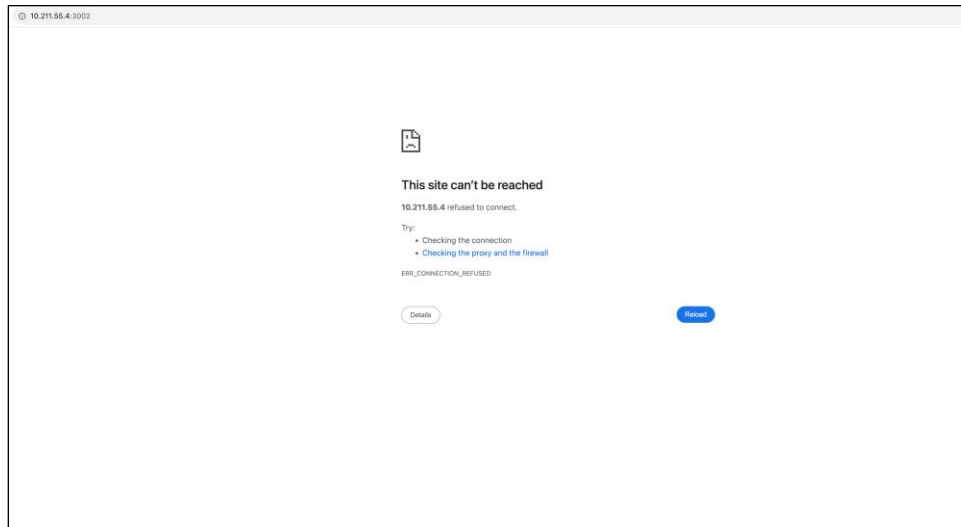


<http://ubuntu-next-app.se-rmutl.net/>

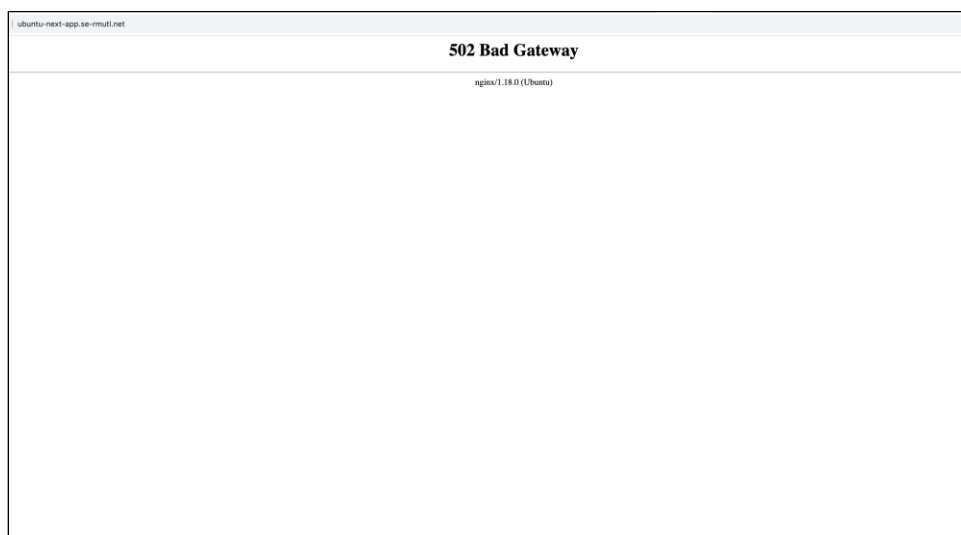
Stop

```
devuser@server:/var/www/lab3/my-react-app$ pm2 stop Nextjs
[PM2] Applying action stopProcessId on app [Nextjs](ids: [ 2 ])
[PM2] [Nextjs](2) ✓
```

id	name	mode	⌵	status	cpu	memory
2	Nextjs	fork	1	stopped	0%	0b
0	Reactjs	fork	2	stopped	0%	0b



<http://10.211.55.4:3002/>



<http://ubuntu-next-app.se-rmutl.net/>

คำสั่ง Delete

```
devuser@server:/var/www/lab3/my-react-app$ pm2 delete Nextjs  
[PM2] Applying action deleteProcessId on app [Nextjs](ids: [ 2 ])  
[PM2] [Nextjs](2) ✓
```

id	name	mode	๗	status	cpu	memory
0	Reactjs	fork	2	stopped	0%	0b