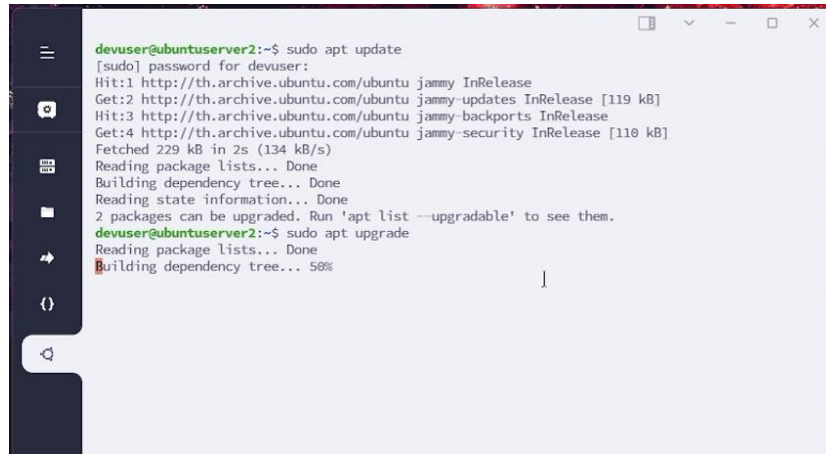


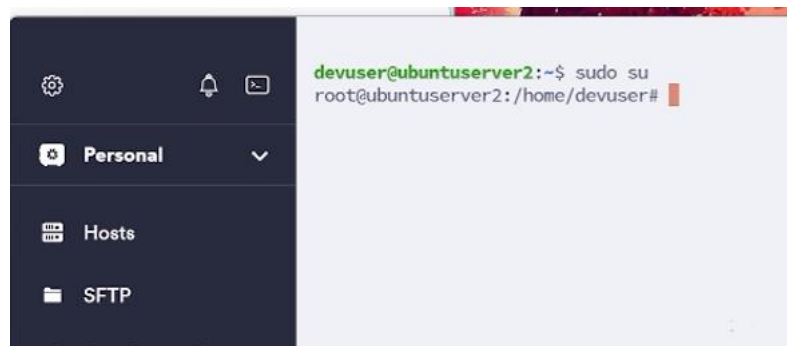
## LAB3

Step1: Update package. `[apt update && apt upgrade]`



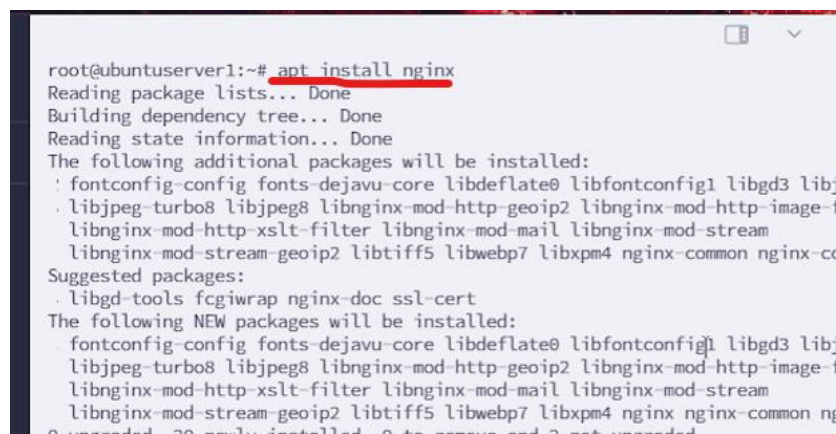
```
devuser@ubuntu-server2:~$ sudo apt update
[sudo] password for devuser:
Hit:1 http://th.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://th.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:3 http://th.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://th.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Fetched 229 kB in 2s (134 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
devuser@ubuntu-server2:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... 50%
```

Step2: Switch to the root account. `[sudo su]`



```
devuser@ubuntu-server2:~$ sudo su
root@ubuntu-server2:/home/devuser#
```

Step3: Install Nginx. `[apt install nginx]`



```
root@ubuntu-server1:~# apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjpeg-turbo8 libjpeg8 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4 nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libdeflate0 libfontconfig1 libgd3 libjpeg-turbo8 libjpeg8 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libtiff5 libwebp7 libxpm4 nginx nginx-common nginx-core
0 upgraded, 28 newly installed, 0 to remove and 2 not upgraded.
```

**Step4:** Check app list & allow Nginx & check status. [ufw app list], [ufw allow 'Nginx HTTP'], [ufw status]

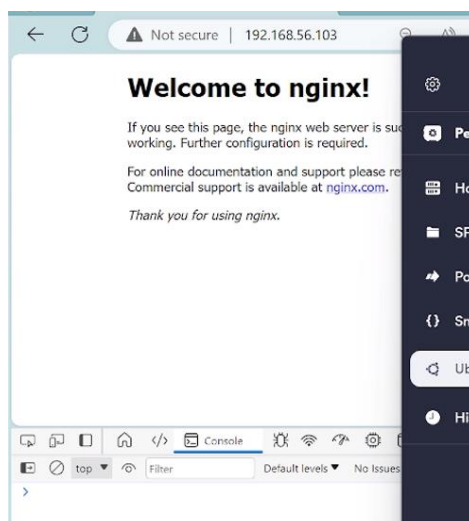
```
root@ubuntu1:~# ufw app list
Available applications:
  Nginx Full
  Nginx HTTP
  Nginx HTTPS
  OpenSSH
root@ubuntu1:~# ufw allow 'Nginx HTTP'
Rule added
Rule added (v6)
root@ubuntu1:~# ufw status
Status: active

To Action From
--
OpenSSH ALLOW Anywhere
Nginx HTTP ALLOW Anywhere
OpenSSH (v6) ALLOW Anywhere (v6)
Nginx HTTP (v6) ALLOW Anywhere (v6)
```

**Step5:** Check service status nginx. [systemctl status nginx]

```
root@ubuntu1:~# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2023-12-24 14:47:43 UTC; 1min 4s ago
     Docs: man:nginx(8)
   Process: 16093 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on;
   Process: 16094 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exi
 Main PID: 16189 (nginx)
    Tasks: 3 (limit: 4558)
   Memory: 5.3M
      CPU: 110ms
   CGroup: /system.slice/nginx.service
           └─16189 "nginx: master process /usr/sbin/nginx -g daemon on; master_proce
             └─16191 "nginx: worker process"
             └─16192 "nginx: worker process"
```

**Step6:** Enter your ip localhost on Web Browser.



Step7: Install NodeJS. [apt install nodejs]

```
root@ubuntu1604:~# apt install nodejs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

\*NVM (Node Version Manager) is a tool used to download, install, manage, and upgrade NodeJS.

Step8: Change Node version to version 20 using nvm (First of all, you must install nvm). [apt install curl], [source ~/.bashrc], [nvm install 20]

- Check Node version [node -v] Check npm version [npm -v] and Check nvm version.

[nvm -v]

```
root@ubuntu1604:~# sudo apt install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (7.81.0-1ubuntu1.15).
curl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
root@ubuntu1604:~# curl https://raw.githubusercontent.com/creationix/nvm/v0.39.7/install.sh | bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time
                                Dload  Upload   Total   Spent    \
=> Downloading nvm from git to '/root/.nvm'
=> Cloning into '/root/.nvm'...
remote: Enumerating objects: 365, done.
remote: Counting objects: 100% (365/365), done.
remote: Compressing objects: 100% (314/314), done.
remote: Total 365 (delta 43), reused 160 (delta 25), pack-reused 160
Receiving objects: 100% (365/365), 364.78 KiB | 1.96 MiB/s, done.
Resolving deltas: 100% (43/43), done.
* (HEAD detached at FETCH_HEAD)
   master
=> Compressing and cleaning up git repository

=> Appending nvm source string to /root/.bashrc
=> Appending bash_completion source string to /root/.bashrc
=> Close and reopen your terminal to start using nvm or run the following
to load it immediately:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion"
root@ubuntu1604:~# source ~/.bashrc
root@ubuntu1604:~# nvm install node
```

```
root@ubuntu1604:~#
root@ubuntu1604:~# nvm install 20
Downloading and installing node v20.10.0...
Downloading https://nodejs.org/dist/v20.10.0/node-v20.10.0-linux-x64.tar.xz
#####
Computing checksum with sha256sum
Checksums matched!
```

```
root@ubuntu1604:~# node -v
v20.10.0
root@ubuntu1604:~# npm -v
10.2.3
root@ubuntu1604:~# nvm -v
0.39.7
root@ubuntu1604:~#
```

# React

**Step1:** Make directories lab3. [mkdir -p /var/www/lab3]

- Go inside the folder lab3. [cd /var/www/lab3]

```
root@ubuntu1:~# mkdir -p /var/www/lab3
```

**Step2:** Create Project React with command. [npm create-react-app my-react-app]

- Edit file package.json (Specify a port run project 3001).

```
root@ubuntu1:/var/www/lab3# npm create-react-app my-react-app
Need to install the following packages:
create-react-app@5.0.1
Ok to proceed? (y) y
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will no
t receive security updates. Please upgrade asap.

Creating a new React app in /var/www/lab3/my-react-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

( ) reify:levn: reifyNode:node_modules/tryer Completed in 1
```

```
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"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  }
}
```

- Allow port 3001. [ufw allow 3001]

```
root@ubuntu1:~# ufw allow 3001
Skipping adding existing rule
Skipping adding existing rule (v6)
root@ubuntu1:~#
```

**Step3:** Set file nginx to set port project React.

- Copy file default in sites-available. `[cp default my-react-app]`

```
root@ubuntuserver1:/etc/nginx/sites-available# ls
default expressjs-example myapp
root@ubuntuserver1:/etc/nginx/sites-available# cp default my-react-app
```

- Set listen port **80** cause the default file will be 80 default\_server
- Set server name. **my-react-app.se-rmutl.net**
- Add location and set port **3001**

```
GNU nano 6.2 my-react-app *
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;
    }
}

# 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.
```

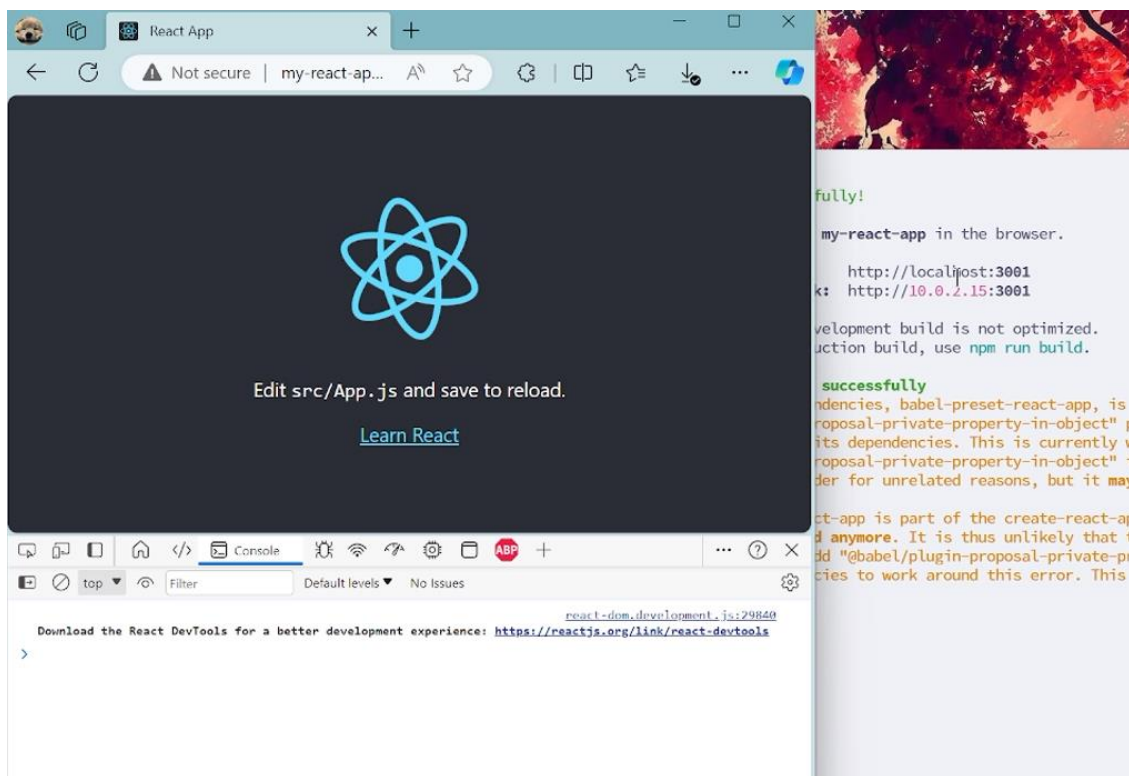
- Then restart nginx and check service it's ok. `[systemctl restart nginx]`, `[nginx -t]`

```
root@ubuntuserver1:/etc/nginx/sites-available# nano my-react-app
root@ubuntuserver1:/etc/nginx/sites-available# systemctl restart nginx
root@ubuntuserver1:/etc/nginx/sites-available# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@ubuntuserver1:/etc/nginx/sites-available# clea
```

- Then set up the host (from the outside machine).

127.0.0.1	localhost	
::1	localhost	
172.10.0.3	host.docker.internal	
172.10.0.3	gateway.docker.internal	
127.0.0.1	kubernetes.docker.internal	
192.168.56.103	myapp.se-rmutl.net	
192.168.56.103	expressjs-example.se-rmutl.net	
<u>192.168.56.103</u>	<u>my-react-app.se-rmutl.net</u>	

- Test to see if Hosts are working normally. `[npm start]`





\*PM2 is a tool that helps our NodeJS process run at all times.

Step4: Install pm2. [npm install pm2 -g]

```
root@ubuntu-server1:~# npm install pm2 -g
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.

added 157 packages in 2m

13 packages are looking for funding
  run `npm fund` for details
root@ubuntu-server1:~#
```

- Run start project react with pm2. [pm2 start --name ReactApp npm -- start]

```
13 packages are looking for funding
  run `npm fund` for details
root@ubuntu-server1:/var/www/lab3/my-react-app# pm2 start --name ReactApp npm -- start
[PM2] Starting /root/.nvm/versions/node/v20.10.0/bin/npm in fork_mode (1 instance)
[PM2] Done.
```

id	name	mode	Uptime	status	cpu	memory
1	ReactApp	fork	0	online	0%	30.6mb
0	my-react-app	fork	139	online	86.8%	69.5mb

- Check the work of pm2 to see if it works? [pm2 stop ReactApp]

The screenshot shows a web browser window with a 502 Bad Gateway error from nginx/1.18.0. In the background, a terminal window displays the following commands and output:

```
root@ubuntu-server1:/var/www/lab3/my-react-app# pm2 list
```

id	name	mode	Uptime	status	cpu	memory
1	ReactApp	fork	0	online	0%	62.1mb

```
root@ubuntu-server1:/var/www/lab3/my-react-app# pm2 stop ReactApp
[PM2] Applying action stopProcessId on app [ReactApp](ids: [ 1 ])
[PM2] [ReactApp](1) ✓
```

id	name	mode	Uptime	status	cpu	memory
1	ReactApp	fork	0	stopped	0%	0b

```
root@ubuntu-server1:/var/www/lab3/my-react-app#
```

- Check pm2 run project it's work again? `[pm2 start ReactApp]`
- Restart nginx. `[systemctl restart nginx]`
- Then check service nginx is ok. `[nginx -t]`

The screenshot shows a web browser window displaying a React App with a blue logo and the text "Edit src/App.js and save to reload". Below the browser, a terminal window is open, showing the following commands and output:

```
root@ubuntu-server1:/var/www/lab3/my-react-app# pm2 list
```

id	name	mode	watch	status	cpu	memory
1	ReactApp	fork	0	online	0%	62.1mb

```
root@ubuntu-server1:/var/www/lab3/my-react-app# pm2 stop ReactApp
[PM2] Applying action stopProcessId on app [ReactApp](ids: [ 1 ])
[PM2] [ReactApp](1) ✓
```

id	name	mode	watch	status	cpu	memory
1	ReactApp	fork	0	stopped	0%	0b

```
root@ubuntu-server1:/var/www/lab3/my-react-app# pm2 start ReactApp
[PM2] Applying action restartProcessId on app [ReactApp](ids: [ 1 ])
[PM2] [ReactApp](1) ✓
[PM2] Process successfully started
```

id	name	mode	watch	status	cpu	memory
1	ReactApp	fork	0	online	0%	19.1mb

```
root@ubuntu-server1:/var/www/lab3/my-react-app# systemctl restart nginx
root@ubuntu-server1:/var/www/lab3/my-react-app# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@ubuntu-server1:/var/www/lab3/my-react-app#
```



# NextJS

**Step1:** Create project NextJS inside lab3 with command. `[npx create-next-app@latest ubuntu-next-app]`

```
root@ubuntu-server1:/var/www/lab3# ls
my-react-app
root@ubuntu-server1:/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
```

- Edit file package.json (Specify a port run project 3002).

```
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"
  }
}
```

**Step2:** Set file nginx to set port project NextJS.

- Copy file my-react-app in sites-available. `[cp my-react-app ubuntu-next-app]`

```
root@ubuntu-server1:~# cd /etc/nginx/
conf.d/          koi-win          nginx.conf       sites-enabled/
fastcgi.conf     mime.types       proxy_params     snippets/
fastcgi_params   modules-available/ scgi_params      uwsgi_params
koi-utf          modules-enabled/  sites-available/ win-utf
root@ubuntu-server1:~# cd /etc/nginx/sites-available
root@ubuntu-server1:/etc/nginx/sites-available# ls
default expressjs-example myapp my-react-app
root@ubuntu-server1:/etc/nginx/sites-available# cp my-react-app ubuntu-next-app
```

- Set server name. `ubuntu-next-app.se-rmutl.net`
- Change app port run `3002`

```
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;
```

- Then restart nginx and allow port run 3002 `[systemctl restart nginx]`, `[ufw allow 3002]`
- Set up the host (from the outside machine).

127.0.0.1	kubernetes.docker.internal	
192.168.56.103	myapp.se-rmutl.net	
192.168.56.103	expressjs-example.se-rmutl.net	
192.168.56.103	my-react-app.se-rmutl.net	
192.168.56.103	ubuntu-next-app.se-rmutl.net	

- Test project Build project NextJS [npm run build]

```
Terminus - nextjs (Not Responding)
root@ubuntu-server1:/var/www/lab3/ubuntu-next-app# npm run build

> ubuntu-next-app@0.1.0 build
> next build

Attention: Next.js now collects completely anonymous telemetry regarding usage.
This information is used to shape Next.js' roadmap and prioritize features.
You can learn more, including how to opt-out if you'd not like to participate in this
anonymous program, by visiting the following URL:
https://nextjs.org/telemetry

▲ Next.js 14.0.4

(node:30752) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please
use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
✓ Creating an optimized production build
✓ Compiled successfully
  Linting and checking validity of types .
```

The screenshot shows a web browser window with the Next.js landing page. The page has a light blue header with the text "Get started by editing app/page.tsx". Below this is the "NEXT.js" logo, followed by a "Docs →" link and the text "Find in-depth information about Next.js features and API.". There is also a "Learn →" link and the "By Vercel" logo at the bottom.

Overlaid on the right side of the browser window is a terminal window showing the output of the "npm run build" command. The terminal output includes the same "Attention" message about telemetry, the "Next.js 14.0.4" version, and the deprecation warning. It also shows the successful completion of the build process with the following logs:

```
✓ Creating an optimized production build
✓ Compiled successfully
  Linting and checking validity of types .
```

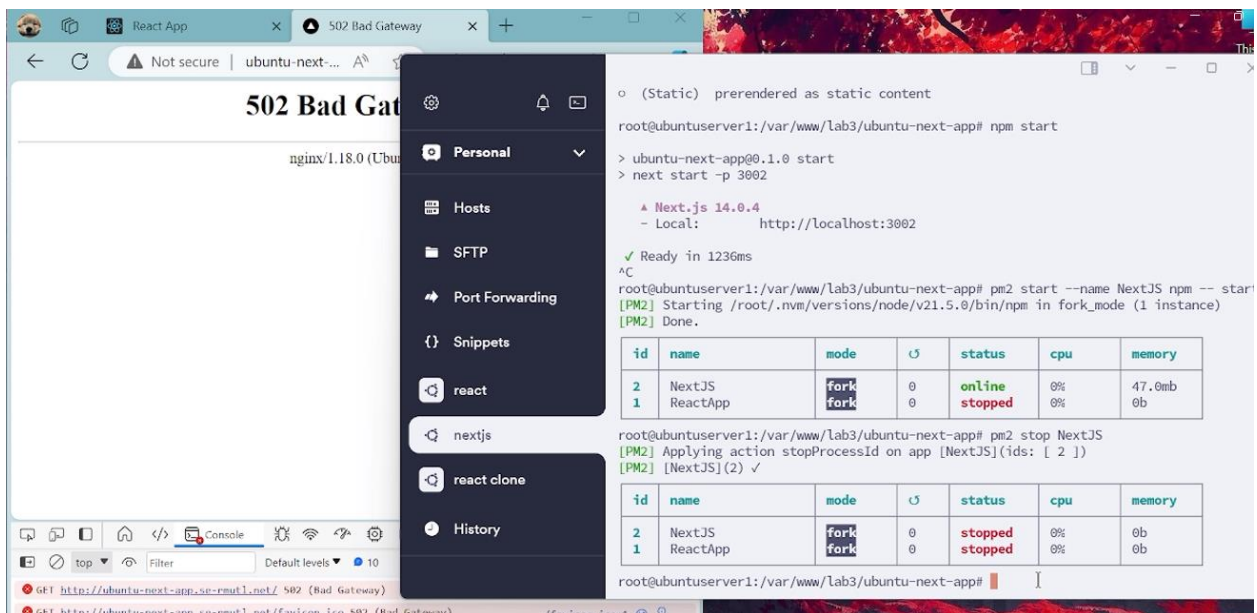
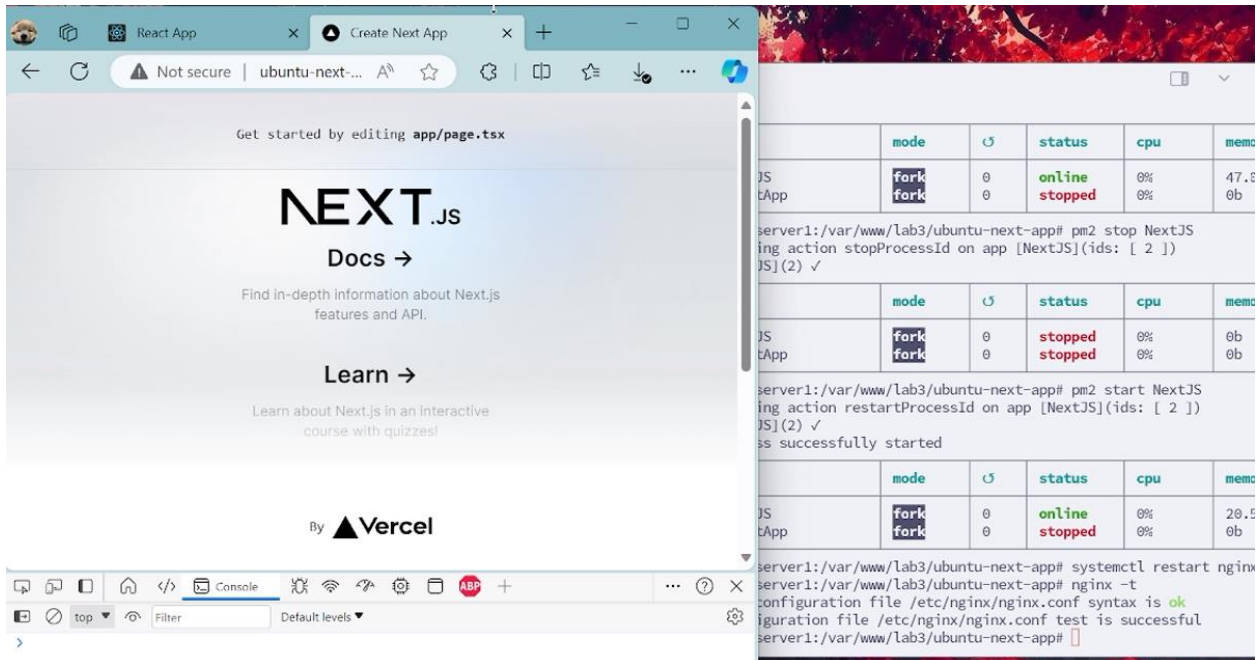
Below the build logs, the terminal shows a table of the build output:

	Size	First Load
und	5.16 kB	
JS shared by all	869 B	
38-5e061ba0d46125b1.js	26.7 kB	
d9d1056-735d320b4b8745cb.js	53.3 kB	
ain-app-b0baefae49ce51a1.js	220 B	
webpack-83ed4af7468830e7.js	1.65 kB	

Below the table, the terminal shows the text "prerendered as static content" and the command "t-app@0.1.0 start -p 3002". At the bottom, it shows the version "14.0.4" and the URL "http://localhost:3002".

Step3: Run start project with pm2 [pm2 start --name NextJS npm -- start]

- Check work of pm2 see if it works. [pm2 stop NextJS]
- Check pm2 run project it's work again. [pm2 start NextJS]
- Restart nginx and check service nginx is ok. [systemctl restart nginx], [nginx -t]



วชิรวิทย์ มุลอ้ง 66543210027-9 (เทียบโอน)