오픈소스 SW

개발도구 활용

과제 2

20185112 김나은

```
naeunkim@naeunkim-virtual-machine:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 upgraded, 8 newly installed, 0 to remove and 77 not upgraded.
Need to get 1,918 kB of archives.
```

sudo apt install apache2를 입력하여 아파치 설치한다.

```
naeunkim@naeunkim-virtual-machine:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
  OpenSSH
naeunkim@naeunkim-virtual-machine:~$ sudo ufw allow 'Apache'
Rules updated
Rules updated (v6)
```

외부에서 접속을 허가할 수 있도록 허가 가능한 목록을 먼저 확인한다.

```
naeunkim@naeunkim-virtual-machine:~$ sudo ufw allow 'Apache'
Skipping adding existing rule
Skipping adding existing rule (v6)
```

기존 룰에 아파치를 허가하고

```
naeunkim@naeunkim-virtual-machine:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache                     ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)

naeunkim@naeunkim-virtual-machine:~$
```

sudo ufw status를 통해 Apache가 허가 되었음을 확인 할 수있다.

```
naeunkim@naeunkim-virtual-machine:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
     Active: active (running) since Mon 2022-09-26 13:55:20 KST; 9min ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 5425 (apache2)
      Tasks: 55 (limit: 4584)
     Memory: 4.9M
        CPU: 121ms
     CGroup: /system.slice/apache2.service
             ├─5425 /usr/sbin/apache2 -k start
             ├─5427 /usr/sbin/apache2 -k start
             └─5428 /usr/sbin/apache2 -k start

 9월  26 13:55:20 naeunkim-virtual-machine systemd[1]: Starting The Apache HTTP >
 9월  26 13:55:20 naeunkim-virtual-machine apachectl[5424]: AH00558: apache2: Co>
 9월  26 13:55:20 naeunkim-virtual-machine systemd[1]: Started The Apache HTTP S>
lines 1-16/16 (END)
```

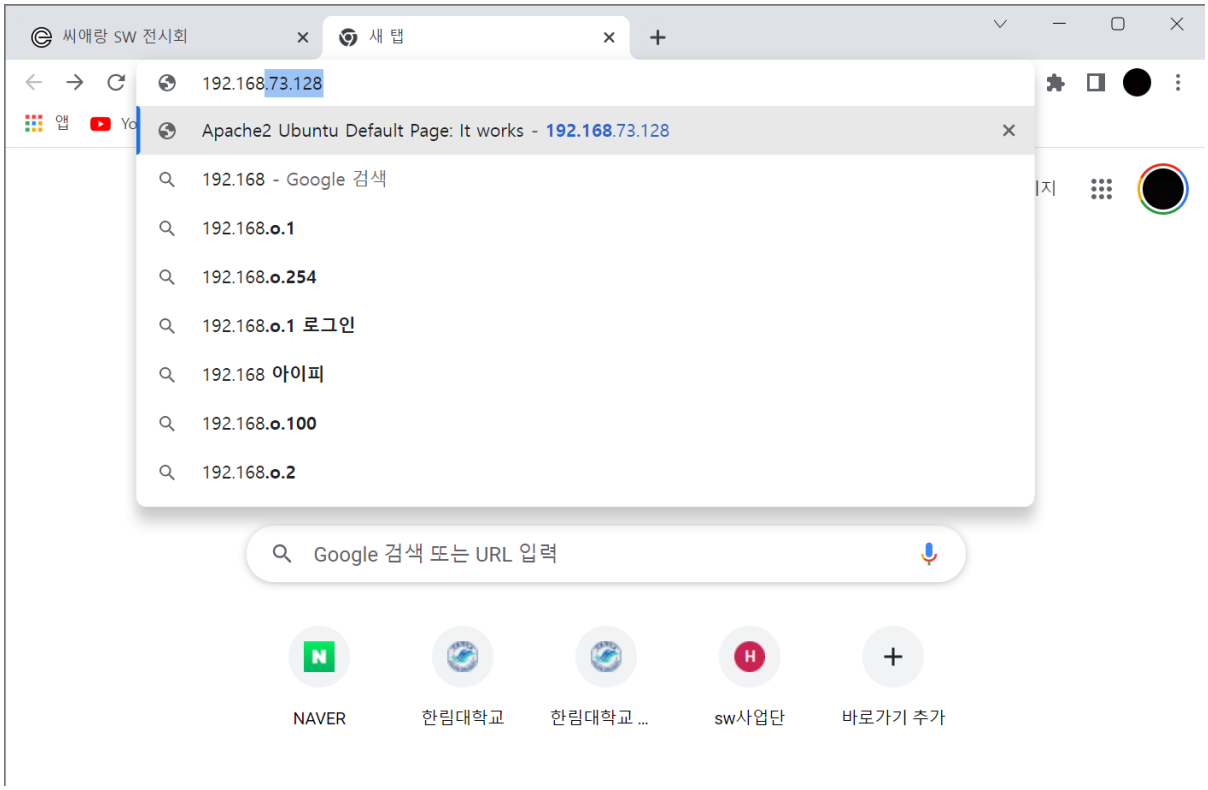
systemctl을 통해 현재 Apache에 대한 서비스 상태를 확인한다.

```
naeunkim@naeunkim-virtual-machine:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:cd:4f:c6:11  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.73.128  netmask 255.255.255.0  broadcast 192.168.73.255
        inet6 fe80::476a:a1b1:d931:8c7d  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:de:11:6b  txqueuelen 1000  (Ethernet)
        RX packets 64067  bytes 93255747 (93.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18328  bytes 1317374 (1.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 238  bytes 24095 (24.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 238  bytes 24095 (24.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

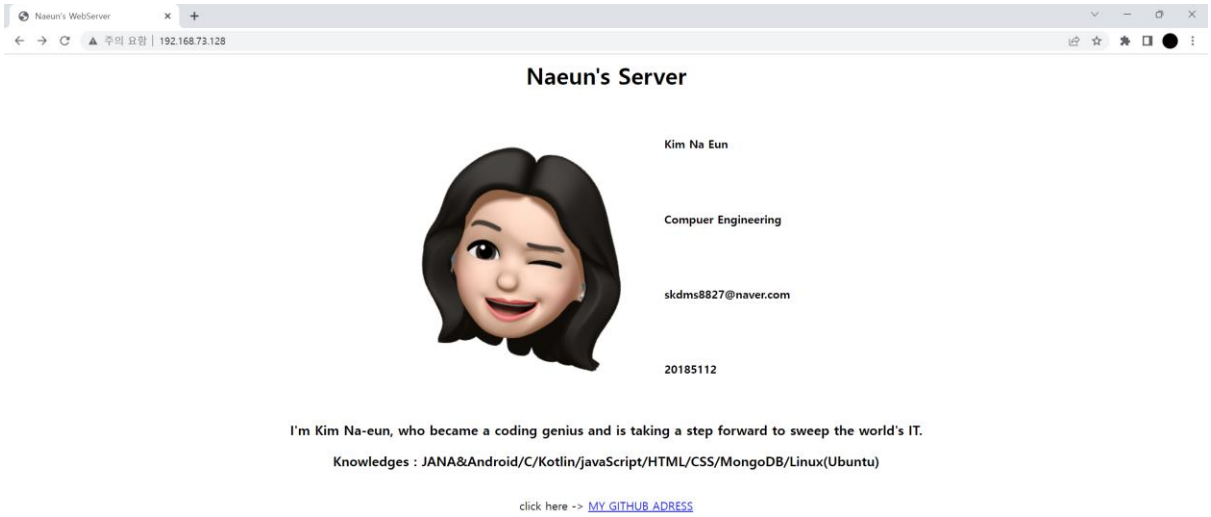
Ifconfig 명령어를 통해 현재 ip주소를 확인하고



ip 주소를 통해 아파치에 접속한다.


씨애랑 SW 전시회
Apache2 Ubuntu Default Page:
주의 요함 | 192.168.73.128
Apache2 Default Page
Ubuntu
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled
| `-- *.conf
|-- sites-enabled
| `-- *.conf
apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
Configuration files in the mods-enabled/, conf-enabled/ and sites-enabled/ directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers a2enmod, a2dismod, a2ensite, a2dissite, and a2enconf, a2disconf . See their respective man pages for detailed information.

아파치 웹서버 설치 성공!!




```
 GNU nano 6.2                                                index.html
<html><head><body>
<title>Naeun's WebServer</title>
<style>
div{ text-aligen:center;}
</style>
<h1>Naeun's Server</h1>
<div>
<table id="me">
<tr>
<td rowspan="4"><div id="imgbox"><img src="https://eunjios.github.io/assets/author_icon.png"></div></td>
<td><strong>Kim Na Eun</strong></td>
</tr>
<tr>
<td><strong>Compuer Engineering</strong></td>
</tr>
<tr>
<td><strong>skdms8827@naver.com</strong></td>
</tr>
<tr>
<td><strong>20185112</strong></td>
</tr>
</table>
</div>
<div>
<h3>I'm Kim Na-eun, who became a coding genius and is taking a step forward to sweep the world's IT.</h3>
<h3>Knowledges : JANA&Android/C/Kotlin/javaScript/HTML/CSS/MongoDB/Linux(Ubuntu)</h3>
<br><span>click here -></span>
<a href="GITHUB : https://github.com/skdms8827">MY GITHUB ADRESS</a>
</div>
</div></body></head></html>
```

html><head><body>

<title>Naeun's WebServer</title>

<style>

body{ text-align:center;}

table { margin-left:500px;}

</style>

<h1>Naeun's Server</h1>

<table id="me">

<tr>

<td rowspan="4"> <img src="https://eunjios.github.io/assets/author_icon.png"> </td>

<td>Kim Na Eun</td>

</tr>

<tr>

<td>Compuer Engineering</td>

</tr>

<tr>

<td>skdms8827@naver.com</td>

</tr>

<tr>

<td>20185112</td>

</tr>

</table>

<h3>I'm Kim Na-eun, who became a coding genius and is taking a step forward to sweep the world's IT.</h3>

<h3>Knowledges : JANA&Android/C/Kotlin/javaScript/HTML/CSS/MongoDB/Linux(Ubuntu)</h3>

click here ->

MY GITHUB ADRESS

</body></head></html>

```
naeunkim@naeunkim-virtual-machine:~$ sudo apt update
[sudo] password for naeunkim:
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://kr.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://kr.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [13.1 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [12.1 kB]
Get:6 http://kr.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:7 http://kr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [92.7 kB]
Get:8 http://kr.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [247 kB]
Get:9 http://kr.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:10 http://kr.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [12.3 kB
]
Fetched 702 kB in 5s (144 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
77 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

먼저 Repository 업데이트한다.

```
naeunkim@naeunkim-virtual-machine:~$ sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20211016).
ca-certificates set to manually installed.
lsb-release is already the newest version (11.1.0ubuntu4).
lsb-release set to manually installed.
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 77 not upgraded.
Need to get 194 kB of archives.
After this operation, 453 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://kr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.81.0-1ubuntu1.4 [
94 kB]
Fetched 194 kB in 2s (110 kB/s)
Selecting previously unselected package curl.
(Reading database ... 199225 files and directories currently installed.)
Preparing to unpack .../curl_7.81.0-1ubuntu1.4_amd64.deb ...
Unpacking curl (7.81.0-1ubuntu1.4) ...
Setting up curl (7.81.0-1ubuntu1.4) ...
Processing triggers for man-db (2.10.2-1) ...
naeunkim@naeunkim-virtual-machine:~$
```

```
naeunkim@naeunkim-virtual-machine:~$ sudo mkdir -p /etc/apt/keyrings
naeunkim@naeunkim-virtual-machine:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sud
o gpg --dearmor -o /etc/apt/keyrings/docker.gpg
naeunkim@naeunkim-virtual-machine:~$ echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download
.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

필요한 여러 패키지와 공식키를 설치한다.

```
naeunkim@naeunkim-virtual-machine:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io do
cker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras docker-scan-plugin git git-man liberror-perl libslirp0 pigz
  slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email
  git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin
  docker-scan-plugin git git-man liberror-perl libslirp0 pigz slirp4netns
0 upgraded, 12 newly installed, 0 to remove and 77 not upgraded.
Need to get 113 MB of archives.
After this operation, 444 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.6.8-1 [28
1 MB]
Get:2 http://kr.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:3 http://kr.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:4 http://kr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.
```

최종 도커 설치!

```
naeunkim@naeunkim-virtual-machine:/var/www/html$ whoami
naeunkim
naeunkim@naeunkim-virtual-machine:/var/www/html$ sudo usermod -aG docker naeunkim
naeunkim@naeunkim-virtual-machine:/var/www/html$ groups naeunkim
naeunkim : naeunkim adm cdrom sudo dip plugdev lpadmin lxd sambashare docker
naeunkim@naeunkim-virtual-machine:/var/www/html$
```

도커 그룹에 나를 추가하여 도커를 사용할 수 있도록 추가한다. Docker 그룹에 나를 추가함으로써 sudo 명령어를 사용하지 않고 자유롭게 사용할 수 있다.

```
naeunkim@naeunkim-virtual-machine:/var/www/html$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:62af9efd515a25f84961b70f973a798d2eca956b1b2b026d0a4a63a3b0b6a3f2
Status: Image is up to date for hello-world:latest
docker.io/library/hello-world:latest
naeunkim@naeunkim-virtual-machine:/var/www/html$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
httpd         latest    f2789344c573   13 days ago     145MB
nginx         latest    2d389e545974   13 days ago     142MB
hello-world   latest    feb5d9fea6a5   12 months ago   13.3kB
```

Hello-world, httpd, nginx를 사용하기 위해 다운 받는다.

```
naeunkim@naeunkim-virtual-machine:/var/www/html$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
httpd         latest    f2789344c573   13 days ago     145MB
nginx         latest    2d389e545974   13 days ago     142MB
hello-world   latest    feb5d9fea6a5   12 months ago   13.3kB
naeunkim@naeunkim-virtual-machine:/var/www/html$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

다운받은 hello-world 실행해본다.

```
naeunkim@naeunkim-virtual-machine:/var/www/html$ docker run --name me -d -p 80:80 nginx
4138bd8783f04f42cc7c388303c47bc55249f20ab2f5f8b59bc84e0bb25eff49
```

hello-world를 다운 받은 방법과 동일하게 pull로 nginx와 httpd를 설치한다.

(이전에 설치한 적이 있어 pull하지 않았다.)

현재 80번포트로 들어오는 것을 nginx의 80포트로 연결하고

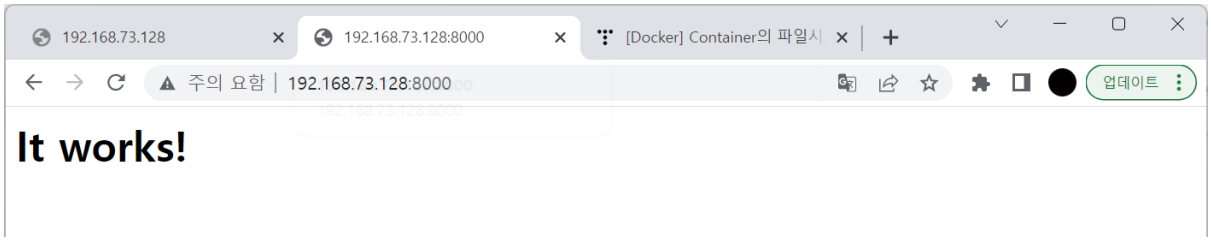
그 이름은 me로 했다.

연결 성공!

이번에는 httpd를 이용하여 접속해본다.

```
naeunkim@naeunkim-virtual-machine:/$ docker run --name httpd8000 -d -p 8000:80 httpd
3bc6d3ef50a102ca269ad9bf34801842f4abbe888883fb091ab6c3846ee7329d
```

httpd8000이라는 이름을 가지고 8000번 포트로 들어오는 것을 80번포트로 연결한 컨터이너를 생성한다.



위 사진가 같은 결과가 나온다면 연결 성공!!

It works 말고 전의 자기소개로 수정했던 html로 수정한다.

하지만 index.html의 위치를 몰라 구글링을 통해 찾을 수 있었다.

```
naeunkim@naeunkim-virtual-machine:/var/www/html$ docker exec -it httpd8000 /bin/bash
root@3bc6d3ef50a1:/usr/local/apache2# cd /usr/local/apache2/htdocs/
root@3bc6d3ef50a1:/usr/local/apache2/htdocs# ls
index.html
```

내가 만든 httpd8000 컨테이너로 들어간다.

이때 -it을 통해 한번만 연결하는 것이 아니라 지속적으로 가능하게 해준다.

컨테이너 내부로 들어와서 index.html이 있는 /usr/local/apache2/htdocs/로

이동하면 Index.html 파일을 찾을 수 있다.

```
root@3bc6d3ef50a1:/usr/local/apache2/htdocs# nano index.html
bash: nano: command not found
root@3bc6d3ef50a1:/usr/local/apache2/htdocs# apt update
Get:1 http://deb.debian.org/debian bullseye InRelease [116 kB]
Get:2 http://deb.debian.org/debian-security bullseye-security InRelease [48.4 kB]
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 Packages [8184 kB]
Get:5 http://deb.debian.org/debian-security bullseye-security/main amd64 Packages [186 kB]
Get:6 http://deb.debian.org/debian bullseye-updates/main amd64 Packages [6344 B]
Fetched 8585 kB in 2s (3489 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
```

nano 에디터를 통해 index.html 파일을 수정하려고 했지만 nano가 설치되어 있지 않았다. apt를 업데이트하고 nano를 설치해본다.

.

```
root@3bc6d3ef50a1:/usr/local/apache2/htdocs# apt install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libncursesw6
Suggested packages:
  hunspell
The following NEW packages will be installed:
  libncursesw6 nano
0 upgraded, 2 newly installed, 0 to remove and 1 not upgraded.
Need to get 789 kB of archives.
After this operation, 3021 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://deb.debian.org/debian bullseye/main amd64 libncursesw6 amd64 6.2+20201114-2 [132 kB]
Get:2 http://deb.debian.org/debian bullseye/main amd64 nano amd64 5.4-2+deb11u1 [656 kB]
Fetched 789 kB in 0s (7466 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libncursesw6:amd64.
(Reading database ... 9121 files and directories currently installed.)
Preparing to unpack .../libncursesw6_6.2+20201114-2_amd64.deb ...
Unpacking libncursesw6:amd64 (6.2+20201114-2) ...
Selecting previously unselected package nano.
Preparing to unpack .../nano_5.4-2+deb11u1_amd64.deb ...
Unpacking nano (5.4-2+deb11u1) ...
Setting up libncursesw6:amd64 (6.2+20201114-2) ...
Setting up nano (5.4-2+deb11u1) ...
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/editor.1.gz because associated file /us
itor) doesn't exist
update-alternatives: using /bin/nano to provide /usr/bin/pico (pico) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/pico.1.gz because associated file /usr/
```

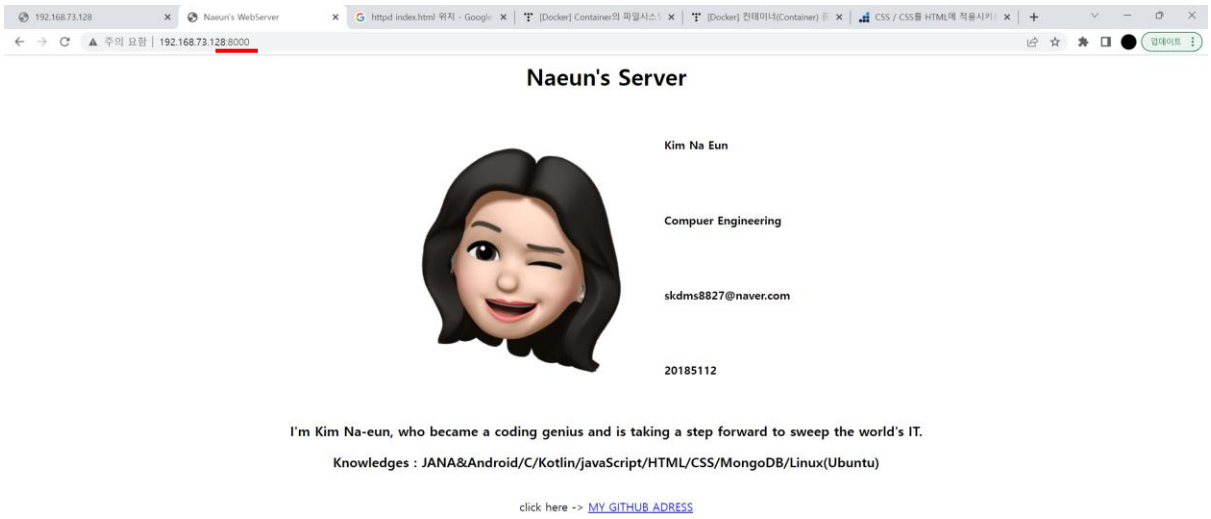
나노 설치!

```
root@3bc6d3ef50a1:/usr/local/apache2/htdocs# nano index.html
```

```
  GNU nano 5.4                                                  index.html
<html><head><body>
<title>Naeun's WebServer</title>
<style>
div{ text-aligen:center;}
</style>
<h1>Naeun's Server</h1>
<div>
<table id="me">
<tr>
<td rowspan="4"><div id="imgbox"><img src="https://eunjios.github.io/assets/author_icon.png"></div></td>
<td><strong>Kim Na Eun</strong></td>
</tr>
<tr>
<td><strong>Compuer Engineering</strong></td>
</tr>
<tr>
<td><strong>skdms8827@naver.com</strong></td>
</tr>
<tr>
<td><strong>20185112</strong></td>
</tr>
</table>
</div>
<div>
<h3>I'm Kim Na-eun, who became a coding genius and is taking a step forward to sweep the world's IT.</h3>
<h3>Knowledges : JANA&Android/C/Kotlin/javaScript/HTML/CSS/MongoDB/Linux(Ubuntu)</h3>
<br><span>click here -></span>
<a href="GITHUB : https://github.com/skdms8827">MY GITHUB ADRESS</a>
</div>
</div></body></head></html>
```

전의 과제에서 이용했던 index.html 코드를 복사해 붙여넣었다.

이제 아까 만들었던 8000포트로 접속해본다.



8000포트로 접속! 그리고 성공!!