

## Lab5. Azure Portal에서 Linux Server 가상머신에 Apache Web Server 올리기

### 1. 목적

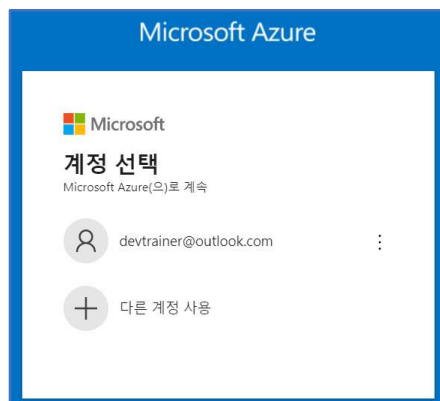
앞의 Lab4에서 Azure Portal에서 생성한 Windows Server 가상머신을 통해 Microsoft Web Server를 설치하고 간단한 홈페이지를 작성해서 웹브라우저에서 확인했다. 이제 같은 Azure Portal에서 Lab3에서 생성한 Linux Server 가상머신을 이용해서 Apache Web Server를 설치하고 간단한 웹페이지를 만들어본다.

### 2. 사전 준비물

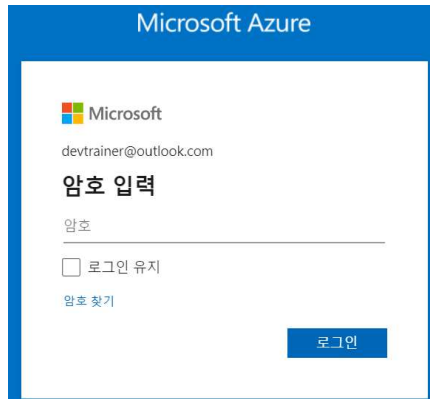
- Azure 체험 계정

### 3. Azure Portal에 연결하기

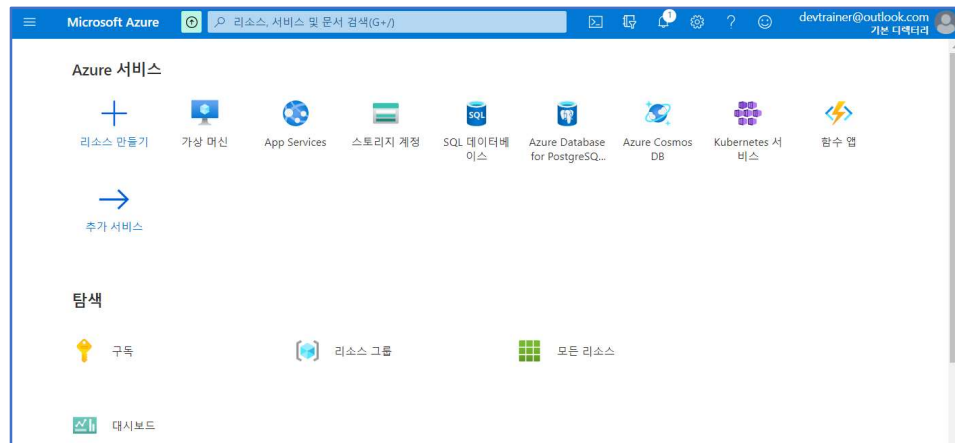
- A. 웹 브라우저를 열고 Azure Portal 사이트에 접속한다. <https://portal.azure.com/#home>



- B. 본인의 계정을 선택하고 암호를 입력하여 로그인한다.



C. Azure Portal 사이트에 들어왔다.

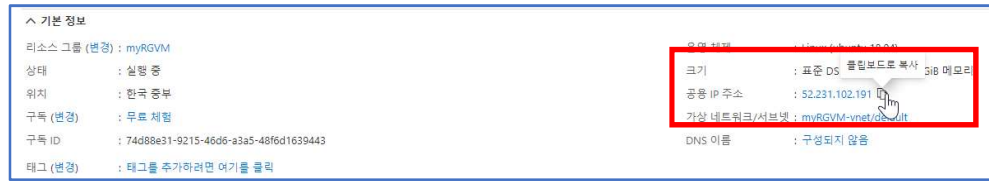


D. [Azure 서비스]의 [가상 머신] 아이콘을 클릭하여 가상 머신 목록으로 가보자. 기존에 생성했던 myVM을 확인할 수 있다.

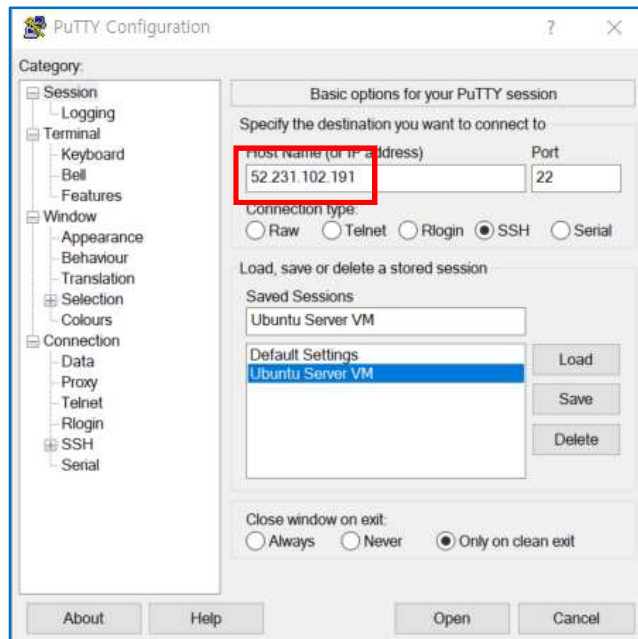


4. Linux Server 가상 머신에 연결하기

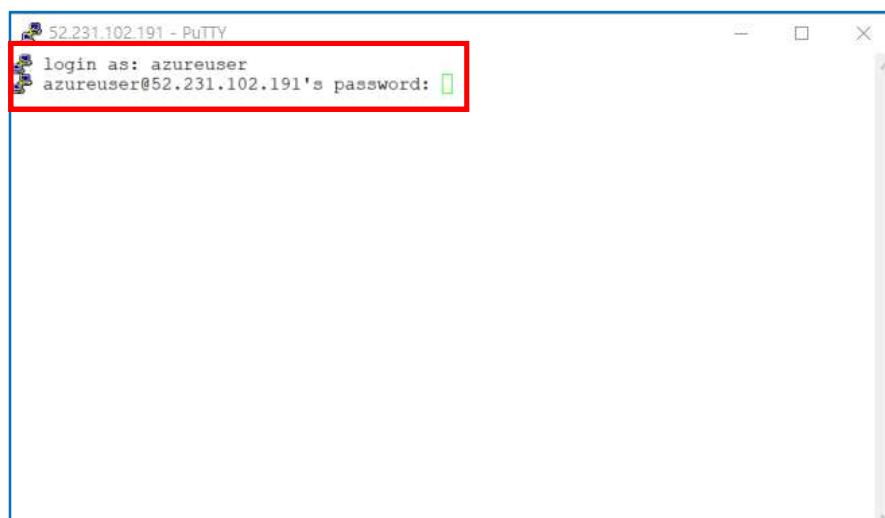
A. 가상 머신 [개요] 페이지에서 [공용 IP 주소]를 복사한다.



- B. PuTTY 프로그램을 연다. 방금 복사한 [공용 IP 주소]를 [Host Name]에 붙여 넣고 [Open] 버튼을 클릭한다.



- C. [login as:]에 관리자계정 **azureuser**를 **azureuser**에 대한 패스워드 **Pa\$\$wOrd1234**를 입력하여 Linux Server 가상 머신에 연결한다.



D. 연결에 성공했다.

```
azureuser@myVM: ~
* Support:      https://ubuntu.com/advantage

System information as of Fri May  7 02:18:44 UTC 2021

System load:  0.0      Processes:      108
Usage of /:   4.5% of 28.90GB  Users logged in:  0
Memory usage: 5%      IP address for eth0: 10.0.0.4
Swap usage:   0%

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azureuser@myVM:~$
```

## 5. 웹 서버 설치 및 테스트하기

A. 이 실습에서 **Apache Web Server**를 설치하고 **Apache Welcome** 페이지가 정상적으로 뜨는지 확인한다. 먼저, 다음의 명령을 사용해서 최신 **apt list**를 업데이트 한다.

**\$ sudo apt update**

```
azureuser@myVM: ~
Get:20 http://azure.archive.ubuntu.com/ubuntu bionic-backports/universe Translat
ion-en [4588 B]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [16
99 kB]
Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [31
8 kB]
Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packag
es [305 kB]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-
en [40.7 kB]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages
[1124 kB]
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en
[253 kB]
Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packag
es [19.2 kB]
Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-
en [4412 B]
Fetched 22.8 MB in 5s (4482 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
7 packages can be upgraded. Run 'apt list --upgradable' to see them.
azureuser@myVM:~$
```

B. 명령 프롬프트에서 다음의 명령어를 사용해서 **Apache Web Server**를 설치한다. 계속 설치를 진행할 것인가 묻는 곳에서 'y'를 넣고 Enter key를 누르거나 기본값으로 'Y'에 맞춰져 있기 때문에 그냥 Enter key를 넣어도 된다.

**\$ sudo apt install apache2**

```
azureuser@myVM: ~  
$ sudo apt install apache2  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following package was automatically installed and is no longer required:  
  linux-headers-4.15.0-142  
Use 'sudo apt autoremove' to remove it.  
The following additional packages will be installed:  
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1  
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 ssl-cert  
Suggested packages:  
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom  
  openssl-blacklist  
The following NEW packages will be installed:  
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1  
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 ssl-cert  
0 upgraded, 10 newly installed, 0 to remove and 7 not upgraded.  
Need to get 1729 kB of archives.  
After this operation, 6985 kB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

C. 설치가 완료되었다.

```
azureuser@myVM: ~  
$ sudo systemctl enable apache2  
Enabling module negotiation.  
Enabling module setenvif.  
Enabling module filter.  
Enabling module deflate.  
Enabling module status.  
Enabling module reqtimeout.  
Enabling conf charset.  
Enabling conf localized-error-pages.  
Enabling conf other-vhosts-access-log.  
Enabling conf security.  
Enabling conf serve-cgi-bin.  
Enabling site 000-default.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.  
Processing triggers for libc-bin (2.27-3ubuntu1.4) ...  
Processing triggers for systemd (237-3ubuntu0.46) ...  
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...  
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...  
Processing triggers for ureadahead (0.100.0-21) ...  
azureuser@myVM: ~$
```

D. Apache Web Server가 제대로 설치됐는지 확인해 보자. 다음의 명령어를 입력한다.

**\$ apache2 -v**

```
azureuser@myVM: ~  
$ apache2 -v  
Server version: Apache/2.4.29 (Ubuntu)  
Server built: 2020-08-12T21:33:25  
azureuser@myVM: ~$
```

- E. 설치된 Apache Web Server가 제대로 구동되는지 확인해보자. 다음의 명령어로 확인할 수 있다.

\$ netstat -ntlp

```
azureuser@myVM: ~
azureuser@myVM:~$ netstat -ntlp
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN
-
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN
-
tcp6       0      0 :::80                  :::*                    LISTEN
-
tcp6       0      0 :::22                  :::*                    LISTEN
-
azureuser@myVM:~$
```

- F. Linux Server의 Apache Web Server는 현재 잘 실행되고 있는 것을 알 수 있다. 이전 Lab인 Lab4에서도 다뤘지만, 여러분들이 웹브라우저로 확인하려면 **80번 포트**가 **[인바운드 포트 규칙]**에 포함되어 있어야 한다. 일단 **PutTY** 프로그램을 닫고, 가상 머신 **[개요]** 페이지에서 좌측 메뉴의 **[네트워킹]**을 클릭한다. **[인바운드 포트 규칙]**도 보이고 **[NIC 공용 IP]**도 확인할 수 있다. **[인바운드 포트 규칙]** 목록에서 **80번** 포트를 찾는다. 혹시 **80번** 포트가 없으면 **[인바운드 포트 규칙 추가]** 파란색 버튼을 클릭한다.



- G. 각 항목의 기본값은 그대로 놓고 **[서비스]**를 **HTTP**로 맞추면 자동으로 **[대상 포트 범위]**가 **80**으로, **[프로토콜]**은 **TCP**로, **[작업]**은 **허용**으로 설정된다. **[이름]**을 **Port\_80**으로 설정하고 **[추가]** 파란색 버튼을 클릭하여 **80번** 포트를 **[인바운드 포트 규칙]**에 추가한다.

**인바운드 보안 규칙 추가**
✕

myVM-nsg

소스 ①

Any

원본 포트 범위 \* ①

\*

대상 주소 ①

Any

서비스 ①

HTTP

대상 포트 범위 ①

80

프로토콜

☐ Any
☒ TCP
☐ UDP
☐ ICMP

작업

☒ 허용
☐ 거부

우선 순위 \* ①

310

이름 \*

Port\_80

설명

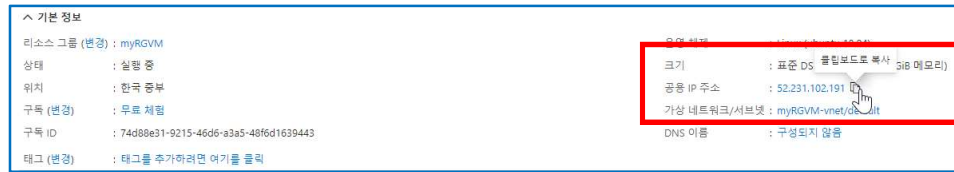
추가

취소

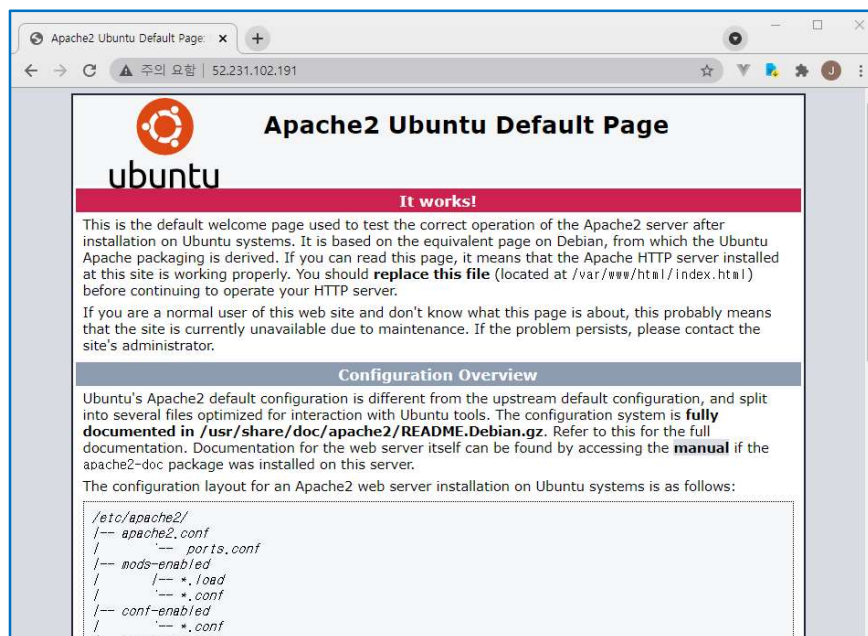
H. 방금 추가한 80번 포트를 [인바운드 포트 규칙]에서 확인할 수 있다.

인바운드 포트 규칙							
<div> <div> <div>인바운드 포트 규칙</div> <div>아웃바운드 포트 규칙</div> <div>액셀러레이션 보안 그룹</div> <div>부하 분산</div> </div> <div> <div> <div>네트워크 보안 그룹 myVM-nsg (네트워크 인터페이스에 연결됨: myvm935)</div> <div>영향 0개 서비스, 1개 네트워크 인터페이스</div> </div> <div>인바운드 포트 규칙 추가</div> </div> </div>							
우선 순위	이름	포트	프로토콜	소스	대상 주소	작업	
300	SSH	22	TCP	모두	모두	허용	...
310	Port_80	80	TCP	모두	모두	허용	...
65000	AllowVnetInBound	모두	모두	VirtualNetwork	VirtualNetwork	허용	...
65001	AllowAzureLoadBalancerInBound	모두	모두	AzureLoadBalancer	모두	허용	...
65500	DenyAllInBound	모두	모두	모두	모두	거부	...

- I. 다시 [개요] 페이지로 돌아와서 [공용 IP 주소]를 복사한다.



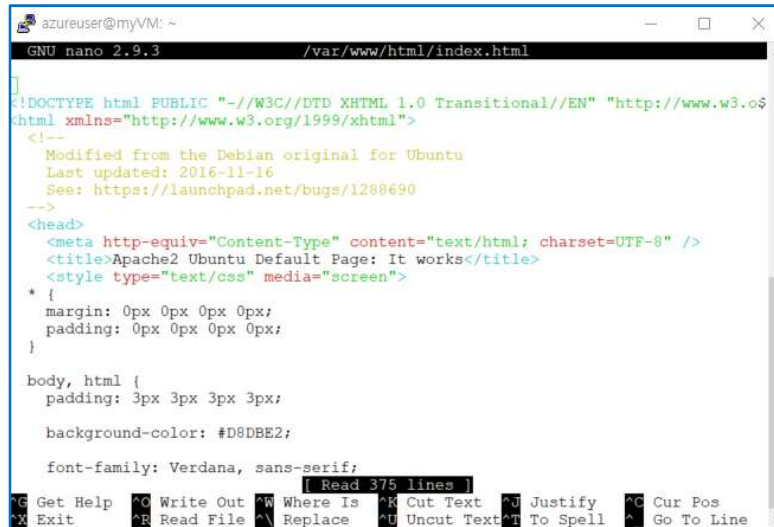
- J. 방금 복사한 [공용 IP 주소]를 여러분의 웹 브라우저를 열고 주소창에 복사한 주소를 붙여 넣는다. 그러면 아래 그림과 같이 **Apache Web Server**의 **Welcome** 화면을 확인하게 될 것이다.



- K. 만일 위의 그림을 편집해서 여러분이 원하는 홈페이지를 만들려면 다음과 같은 작업을 수행한다. 다시 **PuTTY**로 가상머신을 연결한 다음, 명령 프롬프트에서 다음의 명령을 수행한다.

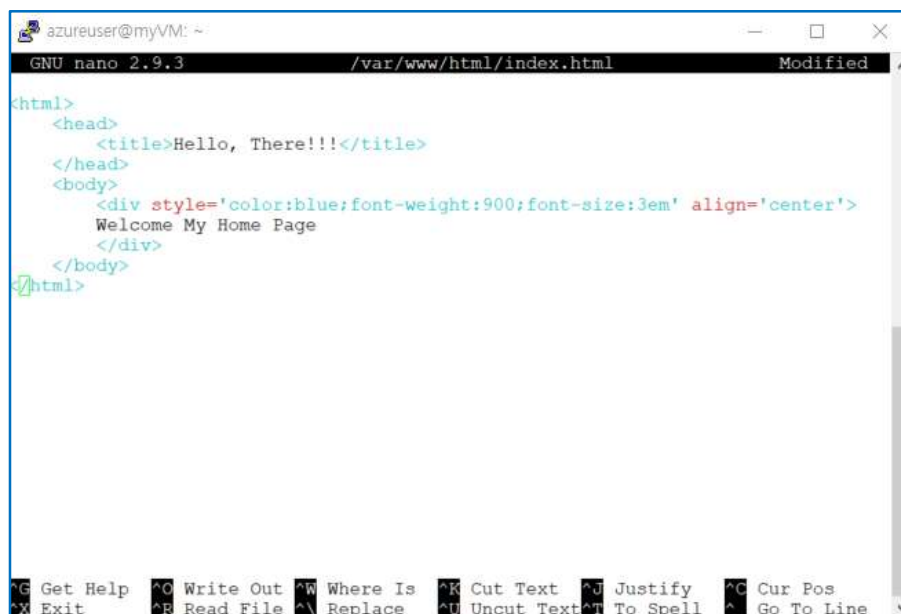
```
$ sudo nano /var/www/html/index.html
```





```
azureuser@myVM: ~  
GNU nano 2.9.3 /var/www/html/index.html  
[  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/1999/xhtml">  
<html xmlns="http://www.w3.org/1999/xhtml">  
  <!--  
    Modified from the Debian original for Ubuntu  
    Last updated: 2016-11-16  
    See: https://launchpad.net/bugs/1288690  
  -->  
  <head>  
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
    <title>Apache2 Ubuntu Default Page: It works</title>  
    <style type="text/css" media="screen">  
      * {  
        margin: 0px 0px 0px 0px;  
        padding: 0px 0px 0px 0px;  
      }  
      body, html {  
        padding: 3px 3px 3px 3px;  
        background-color: #D8DBE2;  
        font-family: Verdana, sans-serif;  
      }  
    </style>  
  </head>  
  <body>  
    <div style="border: 1px solid #D8DBE2; padding: 5px 10px 5px 10px;">  
      <div style="text-align: center; padding: 5px 0 5px 0;">  
        <h1>It Works!</h1>  
      </div>  
    </div>  
  </body>  
</html>
```

- L. Nano 라는 프로그램을 사용하여 편집한다. 이미 **index.html**에는 수 많은 글자들이 있기 때문에 제일 위쪽에 커서를 갖다 놓고, 키보드의 **Alt + a**를 눌러서 블록지정을 시작하고 키보드의 아래 화살표를 계속 눌러서 블록지정을 제일 마지막까지 한다. 그리고 나서 키보드의 **Ctrl + k**를 눌러서 블록지정되어 있는 글자들을 모두 삭제한다. 그리고 다음과 같이 간단한 HTML 코드를 입력한다.



```
azureuser@myVM: ~  
GNU nano 2.9.3 /var/www/html/index.html Modified  
<html>  
  <head>  
    <title>Hello, There!!!</title>  
  </head>  
  <body>  
    <div style='color:blue;font-weight:900;font-size:3em' align='center'>  
      Welcome My Home Page  
    </div>  
  </body>  
</html>
```

- M. **Index.html**을 저장하기 위해 **Ctrl + o**를 누른다. 그러면 **"File Name to Write : /var/www/html/index.html"** 이렇게 메시지가 나오면 Enter Key를 눌러서 저장한다.
- N. Nano 프로그램을 빠져나오기 위해 키보드의 **Ctrl + x**를 누른다.

- O. 여러분의 웹 브라우저를 열고 Linux Server의 **퍼블릭 IPv4 주소**를 다시 입력한다. 또는 이미 웹 브라우저가 열려 있는 상태이면 Refresh 버튼을 눌러서 페이지를 다시 로딩한다.

