



Vagrant

최초 작성일: 24/02/26

최종 제출일: 24/02/27

김민경

0. Vagrant란

0-1. 개념

- 가상화 환경을 관리하고 프로비저닝하는 도구
 - 개발자들이 동일한 개발 환경을 구축하고, 다른 운영 체제나 호스트에서 동일한 개발 환경을 실행할 수 있게 해줌
- VirtualBox, VMware 등과 같은 가상화 소프트웨어와 함께 사용 가능
- Vagrantfile이라는 파일을 작성하여 가상 머신의 설정 및 프로비저닝을 관리하며, 이 파일은 버전 관리 시스템에 저장하여 다른 개발자와 공유 가능함

0-2. Vagrantfile

- **Vagrant.configure("2")**
 - Vagrantfile의 버전 명시, Vagrantfile의 맨 처음에 위치해야함
- **do |config|**
 - Vagrant 설정 정의하는 블록 열어줌, 이 블록 안에서 Vagrant 설정을 지정하는 데 사용되는 변수
 - do 뒤의 수직 막대 (|): 블록에서 사용할 변수를 정의하는데 사용
- **config.vm.box = "ubuntu/jammy64"**
 - 가상 머신 이미지 선택 / 앞에는 OS 타입
 - <https://app.vagrantup.com/boxes/search> 참조
- **config.vm.box_check_update = false**
 - Vagrant가 가상 머신에 할당된 Box 이미지를 업데이트할 때 자동으로 확인하는 것을 비활성화하는 설정
- **config.vm.network "forwarded_port", guest: 80, host: 8080**
 - 가상 머신에 대한 포트 포워딩을 설정하는 라인
 - guest 속성에는 가상 머신 안의 포트 번호를 지정하고, host 속성에는 호스트 컴퓨터에서의 포트 번호를 지정
 - ex, 호스트 컴퓨터에서 브라우저를 열어 <http://localhost:8080>으로 접속시 가상 머신 안의 80번 포트에 접속

- **config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"**
 - 호스트와 게스트 간에 포트 포워딩을 설정하는 데 사용됨. 이 경우에는 게스트의 80번 포트를 호스트의 8080번 포트로 포워딩
 - 추가로 host_ip 옵션을 사용하여 호스트의 IP 주소를 지정하여 게스트의 포트에 대한 외부 접근을 비활성화할 수 있음
- **config.vm.network "private_network", ip: "192.168.33.10"**
 - Vagrantfile에서 가상 머신에 네트워크를 추가하는 데 사용
 - private_network는 호스트 컴퓨터와 가상 머신 간에만 통신할 수 있는 전용 네트워크를 만듦
 - 가상 머신에 고정 IP 주소를 할당할 수 있으며, 이를 통해 호스트와 가상 머신 간의 통신이 쉽게 가능
- **config.vm.network "public_network"**
 - Vagrant에서 사용 중인 가상 머신을 호스트의 공용 네트워크 인터페이스에 연결하도록 구성
 - 외부에서 가상 머신에 직접 연결할 수 있음
- **config.vm.synced_folder "../..../Dropbox", "/dropbox/"**
 - 공유를 생성할 폴더 위치를 지정해줌 첫번째 인자(기준은 vagrantfile 위치)
- **config.vm.provider "virtualbox" do |vb|**
 - Vagrantfile에서 VirtualBox 공급자(provider)를 설정하는 부분
 - 가상화 솔루션을 사용하여 생성된 가상 머신의 설정을 구성할 수 있음. vb 변수를 사용하여 가상 머신의 속성, 예를 들어 GUI 여부, 메모리 및 CPU 수 등을 지정할 수 있음
- **vb.memory = "2048"**
 - VM 메모리 설정
- **vb.cpus = "4"**
 - VM의 CPU 설정

Vagrant란? (velog.io).

1. 문제 1번

Vagrant로 vm를 생성하여 기본환경을 구성하여 테스트하기

다음의 조건에 맞는 vm를 생성하기

운영체제 사양 및 정보

종류	값	비고
CPU	1	
Memory	1G	
Adapter 1	10.0.2.15	NAT 2222(host) -> 22(guest)
Adapter 2	192.168.33.10/24	Private Network
hostname	mz-server	
username	mzuser	
password	mzuser	

1-1. Vagrantfile 스크립트

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/focal64"

  # VM의 호스트 이름 설정
  config.vm.hostname = "mz-server"

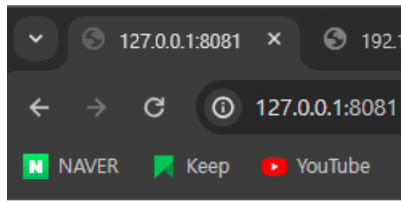
  # VM에 할당할 CPU와 메모리 설정
  config.vm.provider "virtualbox" do |vb|
    vb.name = "mz-server"
    vb.cpus = 1
    vb.memory = "1024" # 메모리를 1GB로 설정
  end

  # 네트워크 설정
  # NAT 네트워크 설정, 포트 포워딩
  config.vm.network "forwarded_port", guest: 22, host: 2222, id: "ssh", auto_cor

  # 프라이빗 네트워크 설정, 고정 IP 할당
  config.vm.network "private_network", ip: "192.168.33.10"

  # VM이 처음 실행될 때 사용자 이름과 비밀번호를 설정하는 스크립트
  config.vm.provision "shell", inline: <<-SHELL
    useradd -m -s /bin/bash mzuser || true # 사용자가 이미 존재하면 에러를 무시
    echo 'mzuser:mzuser' | chpasswd # 비밀번호 설정
  SHELL
end
```

1-2. 결과 확인하기



BABO WebServer

2. 문제 2번

우분투 운영체제에 아파치와 Nginx를 구성하여 정적 웹 서비스를 위한 서버 구성하기

가상머신에 할당된 사양

종류	값
CPU	1
Memory	1G
NIC 1	NAT
NIC 2	Private Network

가상머신 구성 정보

구분	값
hostname	mz-web-server
username	mzuser
password	mzuser
ip address	192.168.33.10

2-1. Vagrantfile 스크립트

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/focal64"
  config.vm.hostname = "mz-web-server"
  config.vm.network "private_network", ip: "192.168.33.10"

  config.vm.synced_folder ".", "/var/www/html"
  config.vm.synced_folder ".", "/var/www/apache2"

  config.vm.provision "shell", inline: <<-SHELL

  sudo apt-get update
  sudo apt-get install -y nginx apache2

  sudo sed -i 's/Listen 80/Listen 8080/g' /etc/apache2/ports.conf
  sudo sed -i 's/<VirtualHost *:80>/<VirtualHost *:8080>/g' /etc/apache2/sites-enabled/000-default.conf

  sudo a2ensite /etc/apache2/000-default.conf
end
```

```

sudo systemctl restart apache2

sudo systemctl stop nginx
sudo fuser -k 80/tcp
sudo systemctl start nginx

sudo useradd mzuser -m -s /bin/bash || true
echo "mzuser:mzuser" | sudo chpasswd

echo '' | sudo tee /var/www/html/index.html

echo '' | sudo tee -a /var/www/ap
SHELL
end

```

1) apache 포트포워딩 해당 부분

```

sudo sed -i 's/Listen 80/Listen 8080/g' /etc/apache2/ports.conf
sudo sed -i 's/<VirtualHost *:80>/<VirtualHost *:8080>/g' /etc/apache2/sites-available

```

- 여기서 apache는 80포트 → 8080포트
- 따라서 VM에서 포트포워딩 설정해줘야 함

비트	프로토콜	호스트 IP	호스트 포트	게스트 IP	게스트 포트
1	TCP	10.0.2.15	8082		22
2	TCP	127.0.0.1	2200		80

2) nginx 포트포워딩 해당 부분

```

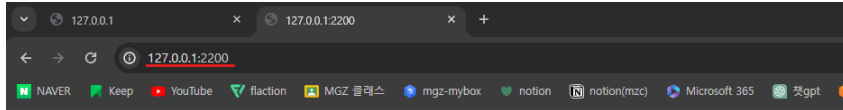
sudo fuser -k 80/tcp

```

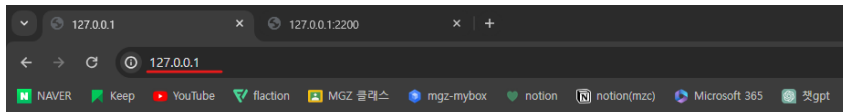
- 여기서 nginx는 80포트를 사용한다는 뜻임

2-2. 결과 확인하기

1) apache로 접속 (2200포트)

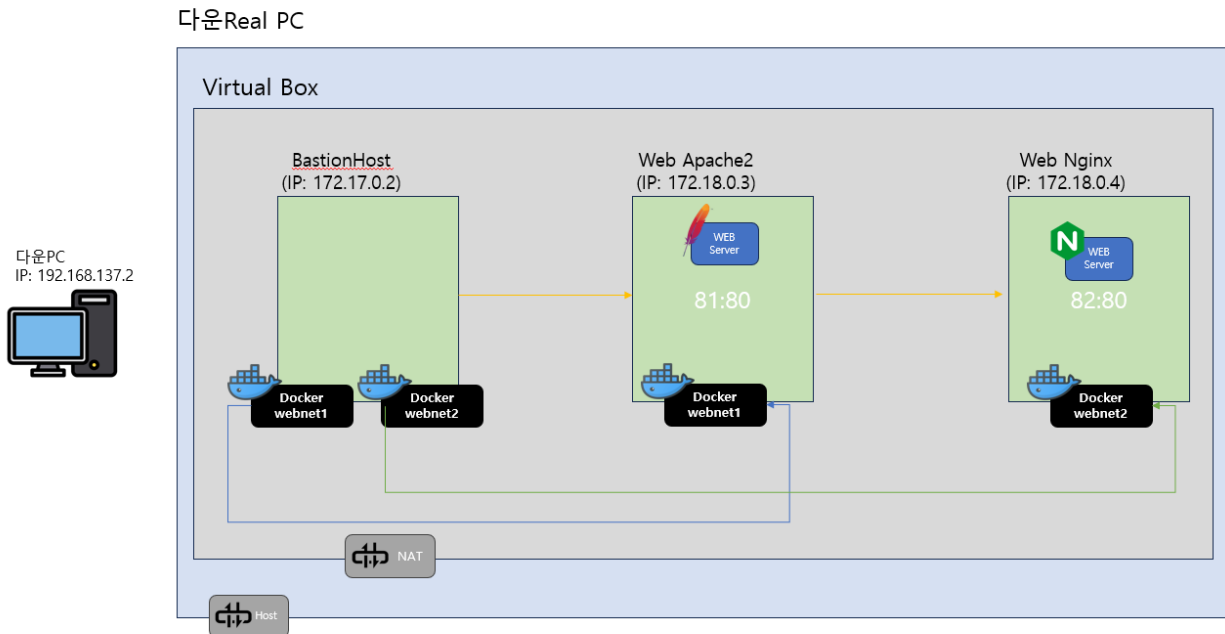


2) nginx로 접속 (80포트)



3. 문제 3번

문제 1번과 문제 2번에서의 실습을 컨테이너로 만들어 3개의 서버 간 통신이 되게 만들기



3-1. Vagrantfile 스크립트

- 초기 스크립트

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/focal64"
  config.vm.box_version = "20240223.0.0"

  config.vm.provision "shell", inline: <<-SHELL
    # 도커 설치
    curl -fsSL https://get.docker.com -o get-docker.sh # 도커 설치 스크립트를 다운로드
    sh get-docker.sh # 다운로드한 Docker 설치 스크립트를 실행하여 Docker를 설치
    usermod -aG docker vagrant # 현재 사용자(vagrant)를 docker 그룹에 추가하여 Docker를 사용할 수 있도록 함
    echo 'sudo su -' >> .bashrc # vagrant 사용자의 .bashrc 파일에 "sudo su -"를 추가하여 root 권한을 얻기 위해 설정

    # 도커 브릿지 네트워크 생성
    docker network create webnet1
    docker network create webnet2

    # 아파치 컨테이너 생성 및 네트워크 연결
    docker run -itd --name apache --network=webnet1 -p 81:80 ubuntu:latest /bin/bash

    # Nginx 컨테이너 생성 및 네트워크 연결
    docker run -d --name nginx --network=webnet2 -p 82:80 nginx:latest

    # 베스트첸 호스트 컨테이너 생성
```

```

    docker run -d --name besthost --network=webnet1 -p 2225:22 --network=webr
SHELL

config.vm.provider "virtualbox" do |vb|
  vb.memory = "4096"
  vb.cpus = "2"
end

config.vm.synced_folder "./htmls", "/dockerhtmls"
config.vm.network "private_network", ip: "192.168.50.13"
config.vm.provision "shell", inline: <<-SHELL
SHELL
end

```

- Vagrantfile 내용을 위의 스크립트로 바꾸기
- cmd에서 **vagrant up** 해서 VM 생성하기
- ⇒ 하지만 오류 뜸

```

PS D:\TEST\Wt4> vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
There are errors in the configuration of this machine. Please fix
the following errors and try again:

vm:
* The host path of the shared folder is missing: /Users/minky/.VirtualBox/htmls

```

- Vagrant가 호스트의 `./htmls` 경로를 찾을 수 없어서 발생함

```
config.vm.synced_folder "./htmls", "/dockerhtmls"
```

- 이를 해결하기 위해서는 호스트 시스템에 `./htmls` 경로를 생성하고 코드를 아래와 같이 수정하기

```
C:\Users\minky\VirtualBox>mkdir htmls
```

```
config.vm.synced_folder "C:/Users/minky/.VirtualBox/htmls", "/dockerhtm
```

- 최종 스크립트

```

Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/focal64"
  config.vm.box_version = "20240223.0.0"

  config.vm.provision "shell", inline: <<-SHELL
    # 도커 설치
    curl -fsSL https://get.docker.com -o get-docker.sh # 도커 설치 스크립트를 다

```



```

sh get-docker.sh # 다운로드한 Docker 설치 스크립트를 실행하여 Docker를 설치
usermod -aG docker vagrant # 현재 사용자(vagrant)를 docker 그룹에 추가하여 D
echo 'sudo su -' >> .bashrc # vagrant 사용자의 .bashrc 파일에 "sudo su -"

# 도커 브릿지 네트워크 생성
docker network create webnet1
docker network create webnet2

# 아파치 컨테이너 생성 및 네트워크 연결
docker run -itd --name apache --network=webnet1 -p 81:80 ubuntu:latest /k

# Nginx 컨테이너 생성 및 네트워크 연결
docker run -d --name nginx --network=webnet2 -p 82:80 nginx:latest

# 베스트첸 호스트 컨테이너 생성
docker run -d --name besthost --network=webnet1 -p 2225:22 --network=webnet2 -p 81:80 nginx:latest
SHELL

config.vm.provider "virtualbox" do |vb|
  vb.memory = "4096"
  vb.cpus = "2"
end

config.vm.synced_folder "C:/Users/minky/.VirtualBox/htmls", "/dockerhtmls"
config.vm.network "private_network", ip: "192.168.50.13"
config.vm.provision "shell", inline: <<-SHELL
SHELL
end

```

3-2. 결과 확인하기 (도커)

- 도커 컨테이너 생성확인

```

root@ubuntu-focal:~# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
c897b150eb6a   ubuntu:latest  "sleep infinity"        19 minutes ago Up 19 minutes 0.0.0.0:2225->22/tcp, :::2225->22/tcp besthost
0576293b94c3   nginx:latest   "/docker-entrypoint. ..." 19 minutes ago Up 19 minutes 0.0.0.0:82->80/tcp, :::82->80/tcp  nginx
aeac2d032bd0   ubuntu:latest  "/bin/bash"             19 minutes ago Up 19 minutes 0.0.0.0:81->80/tcp, :::81->80/tcp  apache
root@ubuntu-focal:~#

```

3-3. 결과 확인하기 (웹)

- 웹 접속 확인

```

root@ubuntu-focal:~# curl localhost:82
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@ubuntu-focal:~#

```

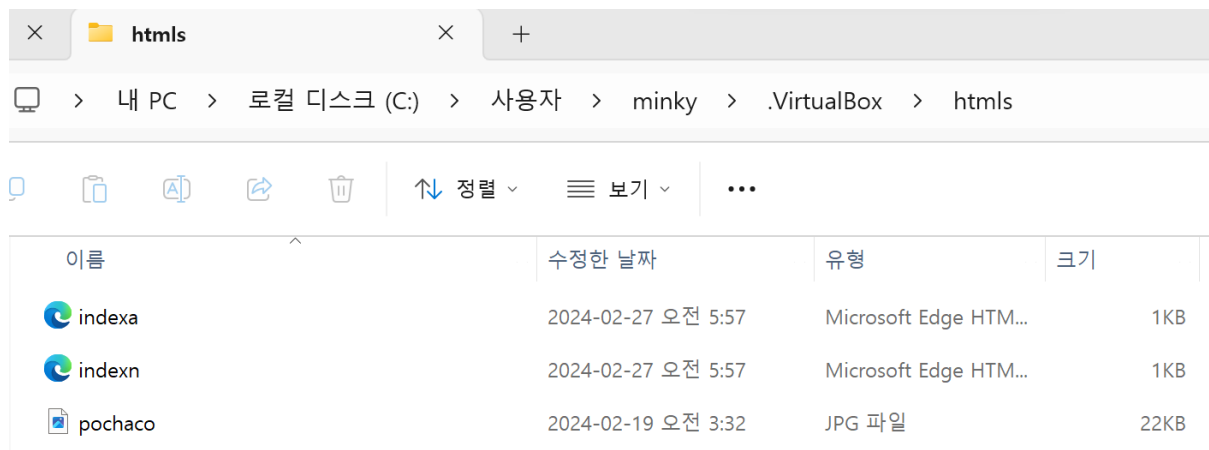
- 아피치 서버는 ubuntu 이미지 기반이기 때문에 apache2를 추가로 설치해야 함

docker exec -it apache /bin/bash //apache 컨테이너에 접속하기

apt-get update

apt-get install apache2

- 공유폴더인 htmls 폴더에 apache와 nginx에 띄울 index.html과 이미지 옮기기



- indexa.html (apache에 띄울 index.html)
- indexn.html (nginx에 띄울 index.html)
- cmd에서도 확인하기

```

root@ubuntu-focal:/dockerhtmls# ll
total 29
drwxrwxrwx  1 vagrant vagrant    0 Feb 26 21:02 ./
drwxr-xr-x 21 root    root    4096 Feb 26 20:00 ../
-rwxrwxrwx  1 vagrant vagrant   82 Feb 26 20:57 indexa.html*
-rwxrwxrwx  1 vagrant vagrant   81 Feb 26 20:57 indexn.html*
-rwxrwxrwx  1 vagrant vagrant 22219 Feb 18 18:32 pochaco.jpg*

```

- indexa.html을 apache 서버로 옮기기

```

root@ubuntu-focal:/dockerhtmls# docker cp /dockerhtmls/indexa.html apache:var/www/html
Successfully copied 2.05kB to apache:var/www/html
root@ubuntu-focal:/dockerhtmls#
root@ubuntu-focal:/dockerhtmls#
root@ubuntu-focal:/dockerhtmls# docker cp /dockerhtmls/pochaco.jpg apache:var/www/html
Successfully copied 24.1kB to apache:var/www/html

```

- 옮긴 후에 indexa.html → index.html로 복사하기

```

root@ubuntu-focal:/dockerhtmls# docker exec -it apache /bin/bash
root@aeac2d032bd0:/# cd /var/www/html
root@aeac2d032bd0:/var/www/html# ls
index.html  indexa.html  pochaco.jpg
root@aeac2d032bd0:/var/www/html# cp indexa.html index.html
root@aeac2d032bd0:/var/www/html# ls
index.html  indexa.html  pochaco.jpg
root@aeac2d032bd0:/var/www/html# cat index.html
<h2>Welcome to Minkyeong server! It's apache sever! </h2>
root@aeac2d032bd0:/var/www/html#

```

- indexa.html 지우기

```

root@aeac2d032bd0:/var/www/html# ls
index.html  indexa.html  pochaco.jpg
root@aeac2d032bd0:/var/www/html# rm indexa.html
root@aeac2d032bd0:/var/www/html# ls
index.html  pochaco.jpg

```

- curl 접속 확인하기

```

root@ubuntu-focal:/dockerhtmls# curl localhost:81
curl: (56) Recv failure: Connection reset by peer

```

⇒ 하지만 오류 뜸

```
root@ubuntu-focal:/dockerhtmls# curl localhost:81
curl: (56) Recv failure: Connection reset by peer
```

- 이 오류는 일반적으로 서버와의 연결이 설정되었지만 원격 서버에서 해당 연결을 닫은 경우 발생되었다고 함. 아직 구체적인 원인은 모르겠음
- 해결방법

apt-get update

apt-get install openssl //openssl 설치하기

```
root@aeac2d032bd0:/# apt-get install openssl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssl is already the newest version (3.0.2-0ubuntu1.14).
openssl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

- 이미 설치되어 있는 것으로 보이긴 함

service apache2 restart

- 다시 curl 접속하기

```
root@ubuntu-focal:/dockerhtmls# curl localhost:81
<h2>Welcome to MinKyeong server! It's apache sever! </h2>
root@ubuntu-focal:/dockerhtmls#
```

- 동일한 과정으로 indexa.html을 nginx 서버로 옮기기
 - nginx에서는 /usr/share/nginx/html로 옮기기

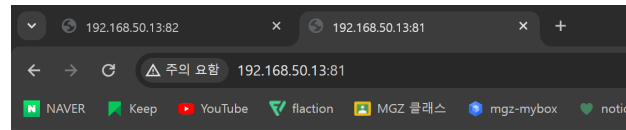
```
root@ubuntu-focal:/dockerhtmls# docker cp /dockerhtmls/indexn.html nginx:/usr/share/nginx/html
Successfully copied 2.05kB to nginx:/usr/share/nginx/html
root@ubuntu-focal:/dockerhtmls#
root@ubuntu-focal:/dockerhtmls# docker cp /dockerhtmls/pochaco.jpg nginx:/usr/share/nginx/html
Successfully copied 24.1kB to nginx:/usr/share/nginx/html
root@ubuntu-focal:/dockerhtmls#
```

- curl 접속 확인하기

```
root@ubuntu-focal:/dockerhtmls# curl localhost:82
<h2>Welcome to MinKyeong server! It's nginx sever! </h2>
root@ubuntu-focal:/dockerhtmls#
```

- 외부접속 확인하기

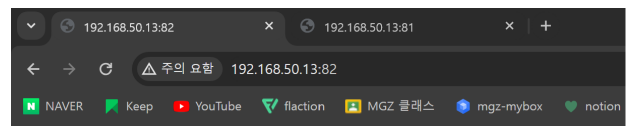
- apache



Welcome to Minkyeong server! It's apache sever!



- nginx



Welcome to Minkyeong server! It's nginx sever!

