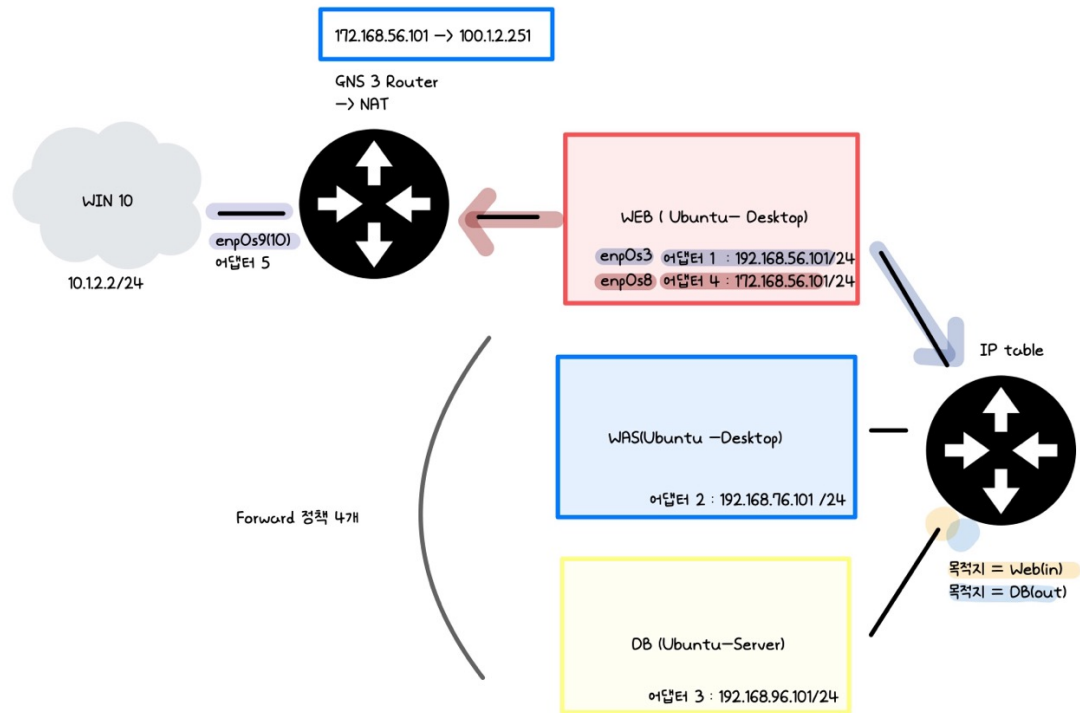


실습 -3Tier

백브리핑(논리적 그림)



1. 위 구성도에서 DB Server 에 접근하기 위한 다양한 방법을 실습 진행 합니다.
 - a. 동일한 사설 네트워크의 서브넷에서 접속하는 방법에 대해 실습 합니다.
 - b. 동일한 사설 네트워크의 다른 서브넷에서 접속하는 방법에 대해 실습 합니다.(DB에 접속 허용 주소가 계정과 맵핑되어 있습니다.)
 - c. 외부 네트워크에서 DB Server에 접속하는 방법에 대해 실습을 합니다.
 - NAT 구성으로 직접 DB에 연결
 - NAT 로 DB 에 직접 연결 하는 것이 아닌 Bestion Host 를 사용하여 연결
 - SSH 터널링 기능을 활용하여 DB에 연결
2. Web 서버를 추가하여 다음의 실습을 진행합니다.
 - a. 웹접근시 본인의 사진을 출력하는 화면을 띄워 주세요
 - b. PHP 를 사용하여 DB에 연결 후 DB 에 저장된 데이터를 조회 합니다.(필요한 패키지: apache2, php8.1, php8.1-mysql)
3. web + was + db 서버를 분리하여 동작하는 3Tier를 구현하는 실습을 진행합니다.
 - a. 메인(정적) 페이지는 개인 캐릭터 이미지
 - b. 동적 페이지는 개인 프로필 정보(DB에 저장된 정보)
 - c. DB정보는 이름, 나이, 희망 목표, 현재 본인 기술 스택을 필수 내용으로 하며 추가적으로 각 팀원에 대한 정보를 넣고 실제 조회 시 각자 개인 정보가 나오게 함.
 - d. 구성도는 지금까지 실습했던 내용을 참고하여 새롭게 구현하여 완성할 것

- 전제조건
 - WEB2Tier 실습 한 장비를 토대로 사용

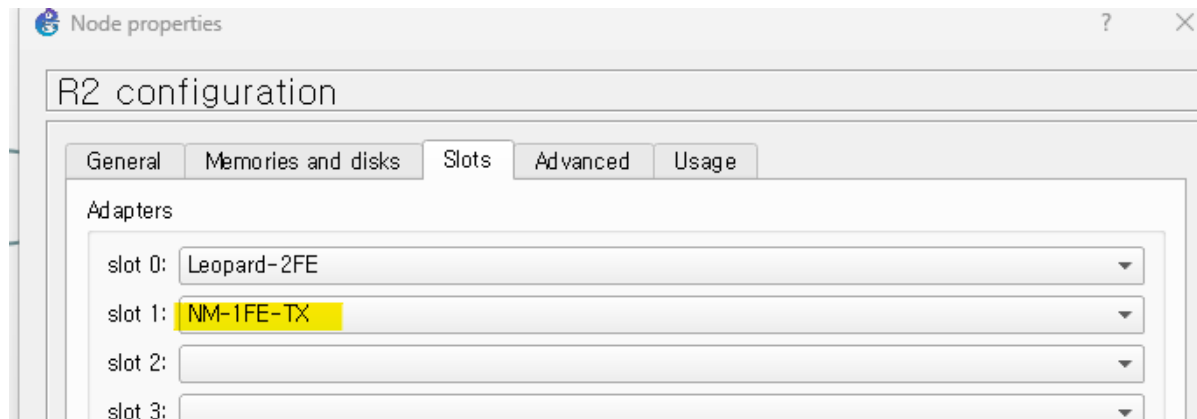
실습과정

1. 환경 설정하기

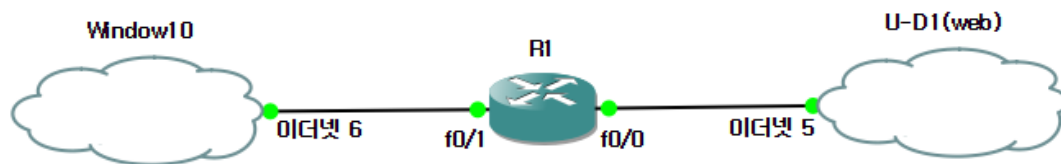
- 다음과 같이 어댑터 ip 대역 수정

이름	IPv4 접두사	IPv6 접두사	DHCP 서버
VirtualBox Host-Only Ethernet Adapter	192.168.56.101/24		사용 안 함
VirtualBox Host-Only Ethernet Adapter # 2	192.168.76.101/24		사용 안 함
VirtualBox Host-Only Ethernet Adapter # 3	192.168.96.101/24		사용 안 함
VirtualBox Host-Only Ethernet Adapter # 4	172.168.56.101/24		사용 안 함
VirtualBox Host-Only Ethernet Adapter # 5	10.1.2.1/24		사용 안 함

- R2 슬롯 추가

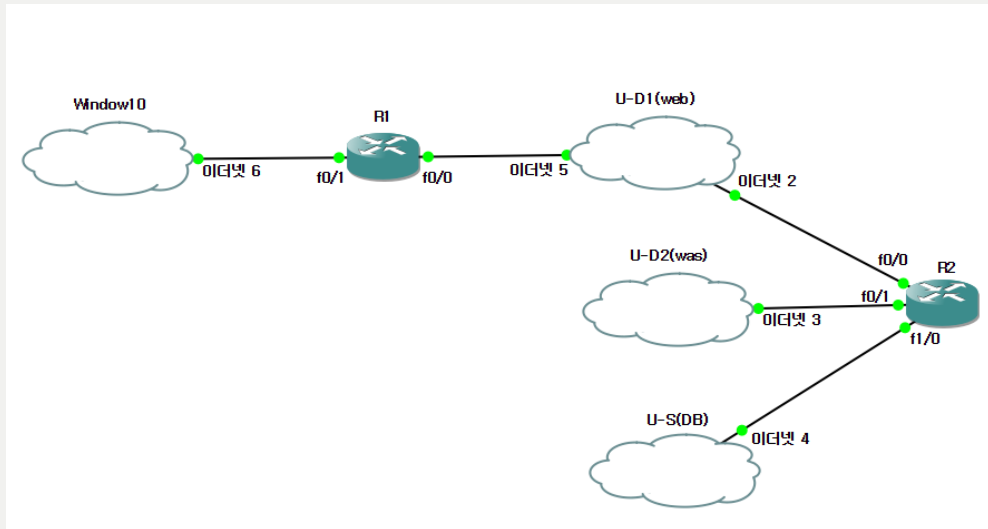


- GNS 설정하기
 - 물리적 그림은 다음과 같음





주의) 아래는 논리적 그림임. gns에는 물리적 그림을 그려줘야 함



- Router에 ip 할당하기

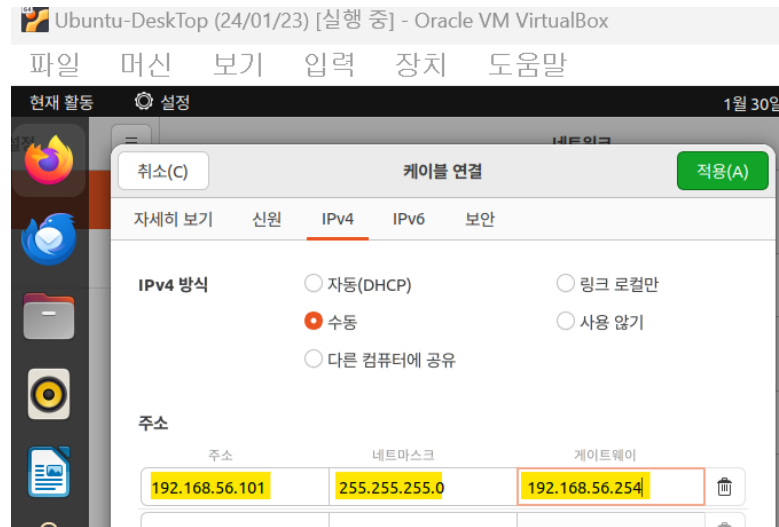
- R1

```
R1(config)#int f0/1
R1(config-if)#ip add 10.1.2.254 255.255.255.0
R1(config-if)#no sh
R1(config-if)#
R1(config-if)#int f
*Mar 1 00:03:57.099: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar 1 00:03:58.099: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
R1(config-if)#int f0/0
R1(config-if)#ip add 172.168.56.254 255.255.255.0
R1(config-if)#no sh
R1(config-if)#
R1(config-if)#do show ip
*Mar 1 00:04:18.115: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:04:19.115: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#do show ip int br
Interface                IP-Address      OK? Method Status      Protocol
FastEthernet0/0          172.168.56.254 YES manual up          up
FastEthernet0/1          10.1.2.254     YES manual up          up
R1(config-if)#
```

- 각 서버에 ip 할당& 어댑터 연결하기

- U-D1

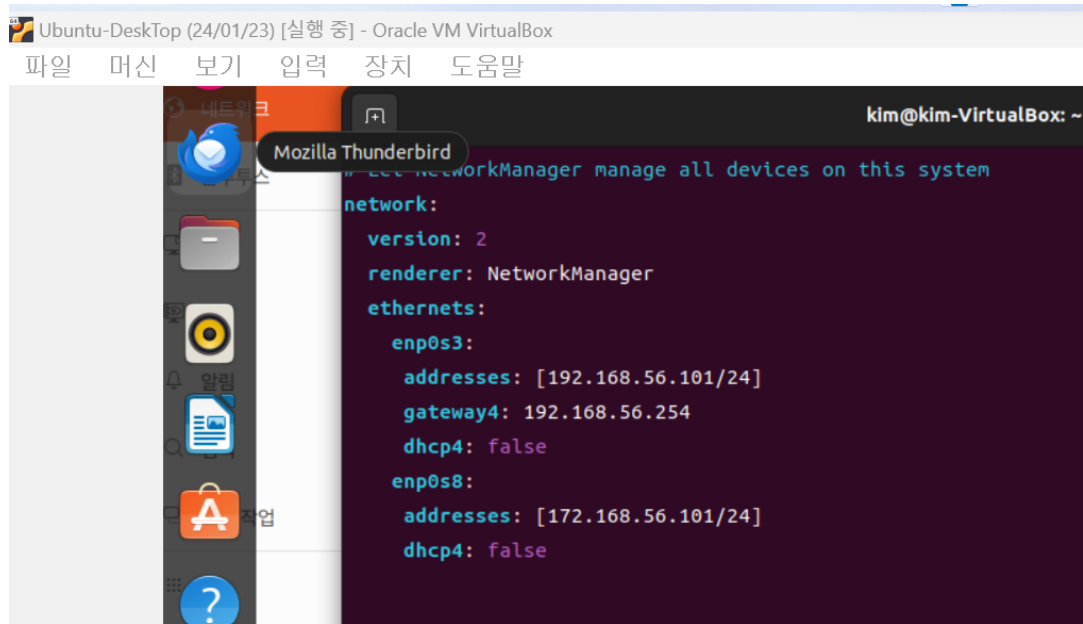
- 어댑터 1: 192.168.56.101



- 어댑터 4: 172.168.56.101

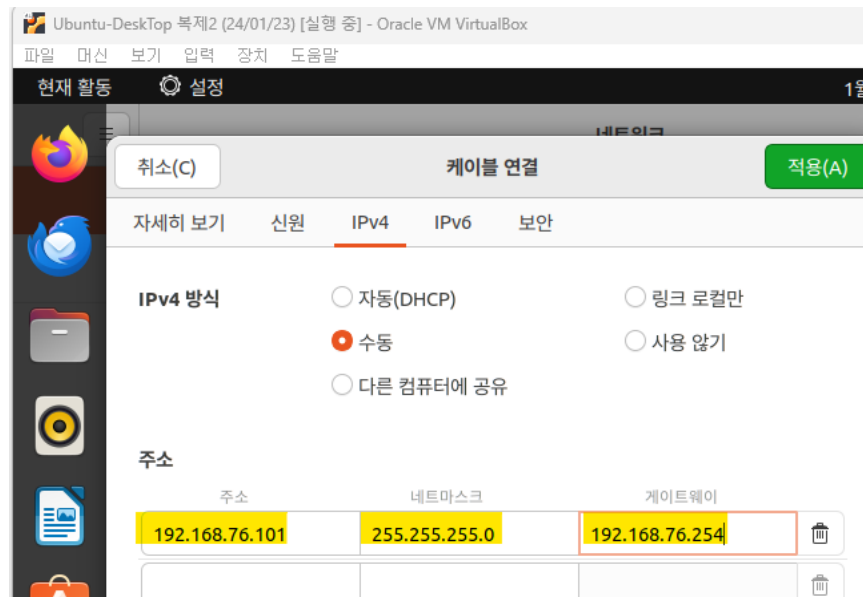


- 아래처럼 netplan으로 ip 할당해도 됨



◦ U-D2

- 어댑터 2: 192.168.76.101



◦ U-S1

- 어댑터 3: 192.168.96.101

`sudo vim /etc/netplan/00-installer-config.yaml` //netplan 파일 편집하기

```
Ubuntu-Server1 (스냅샷 1/25) [실행 중] - Oracle VM Vi  
파일  머신  보기  입력  장치  도움말  
network:  
  ethernets:  
    enp0s3:  
      addresses: [192.168.96.101/24]  
      gateway4: 192.168.96.254  
      nameservers:  
        addresses: [8.8.8.8]  
      dhcp4: no  
    enp0s8:  
      dhcp4: true  
  version: 2  
~
```



주의)

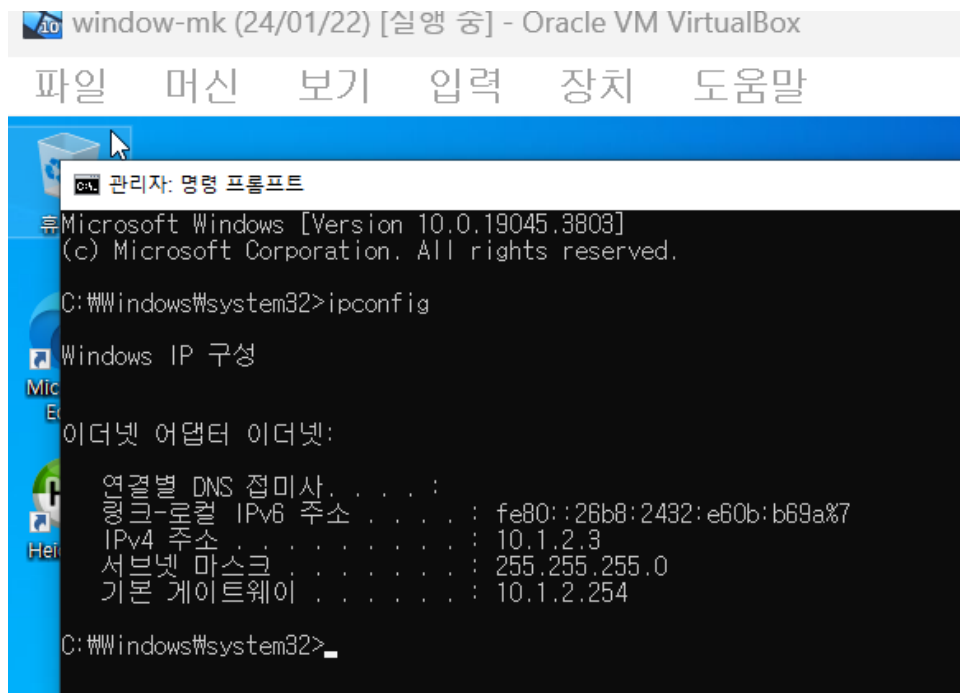
아래같이 설정 시 ubuntu server는 꺾다키면 ip 날라가니까 netplan에서 변경해주기

```
sudo ifconfig enp0s3 192.168.96.101/24 //ip 수동 할당하기
```

```
sudo netplan apply
```

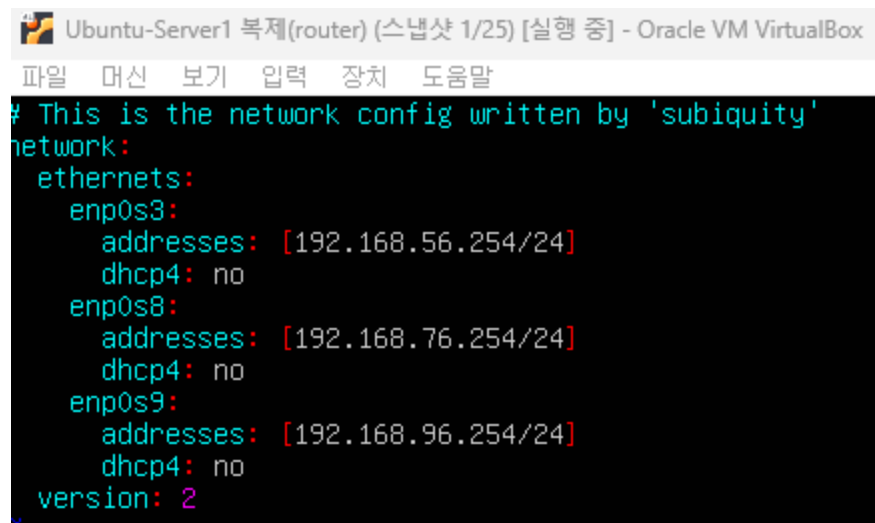
◦ Window10

- 어댑터 5: 10.1.2.3



◦ U-S2 (router 역할)

- 어댑터 1: 192.168.56.254
- 어댑터 2: 192.168.76.254
- 어댑터 3: 192.168.96.254



`sudo netplan apply`



잘못 넣은 것

Ubuntu-Server1 복제(router) (스냅샷 1/25) [실행 중] - Oracle VM VirtualBox

파일 머신 보기 입력 장치 도움말

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      addresses: [192.168.56.102/24]
      gateway4: 192.168.56.254
      dhcp4: no
    enp0s8:
      addresses: [192.168.76.102/24]
      gateway4: 192.168.76.254
      dhcp4: no
    enp0s9:
      addresses: [192.168.96.102/24]
      gateway4: 192.168.96.254
      dhcp4: no_
  version: 2
```

- ping 통신 확인하기
 - 서버끼리의 ping 통신이 안됨

2. 포워딩 설정하기

#특정 IP를 차단하려면

iptables -A INPUT -s x.x.x.x -j DROP을 실행하면 되고 (inbound)

iptables -A OUTPUT -d x.x.x.x -j DROP (outbound)

- 모든 FORWARD 규칙 삭제

```
sudo iptables -F FORWARD
```

- 기본 정책 설정 (FORWARD 체인에 있는 모든 패킷 거부)

```
sudo iptables -P FORWARD DROP
```

- 192.168.56.0/24에서 192.168.76.0/24로의 통신 허용

```
sudo iptables -A FORWARD -s 192.168.56.0/24 -d 192.168.76.0/24 -j ACCEPT
sudo iptables -A FORWARD -s 192.168.76.0/24 -d 192.168.56.0/24 -j ACCEPT
```

- 192.168.76.0/24에서 192.168.96.0/24로의 통신 허용

```
sudo iptables -A FORWARD -s 192.168.76.0/24 -d 192.168.96.0/24 -j ACCEPT
sudo iptables -A FORWARD -s 192.168.96.0/24 -d 192.168.76.0/24 -j ACCEPT
```

```
ubuntu@ubuntu:~$ sudo iptables -F FORWARD
ubuntu@ubuntu:~$
ubuntu@ubuntu:~$ sudo iptables -P FORWARD DROP
ubuntu@ubuntu:~$
ubuntu@ubuntu:~$ sudo iptables -A FORWARD -s 192.168.56.0/24 -d 192.168.76.0/24 -j ACCEPT
ubuntu@ubuntu:~$ sudo iptables -A FORWARD -s 192.168.76.0/24 -d 192.168.56.0/24 -j ACCEPT
ubuntu@ubuntu:~$
ubuntu@ubuntu:~$ sudo iptables -A FORWARD -s 192.168.76.0/24 -d 192.168.96.0/24 -j ACCEPT
ubuntu@ubuntu:~$ sudo iptables -A FORWARD -s 192.168.96.0/24 -d 192.168.76.0/24 -j ACCEPT
ubuntu@ubuntu:~$ _
```

- `sudo iptables -L` //iptables 규칙 확인

3. ping 통신 해보기

- but WEB - WAS끼리의 ping 통신이 안됨
 - 현재 U-D1⇒ 192.168.76.0/24대를 모르는 상태임
 - 10.1.2.0/24대도 모르는 상태일테니
 - static route 설정해주기
 - U-D1에서

```
kim@kim-VirtualBox:~$ sudo ip route add 10.1.2.0/24 via 192.168.56.254
[sudo] kim 암호:
kim@kim-VirtualBox:~$ sudo ip route add 192.168.76.0/24 via 192.168.56.254
kim@kim-VirtualBox:~$
```

- U-D2에서

```
kim@kim-VirtualBox:~$ sudo ip route add 192.168.56.0/24 via 192.168.76.254
[sudo] kim 암호:
kim@kim-VirtualBox:~$ sudo ip route add 192.168.96.0/24 via 192.168.76.254
```

- ping 통신 ⇒ 실패

```

kim@kim-VirtualBox:~$ route
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
default          192.168.56.254 0.0.0.0         UG    20102  0      0 enp0s3
default          172.168.56.254 0.0.0.0         UG    20103  0      0 enp0s8
link-local       0.0.0.0         255.255.0.0     U     1000   0      0 enp0s3
172.168.56.0     0.0.0.0         255.255.255.0   U     103    0      0 enp0s8
192.168.56.0     0.0.0.0         255.255.255.0   U     102    0      0 enp0s3

```

◦ 다시 ping 통신 시도(WEB-WAS) ⇒ 성공

- web → was

```

kim@kim-VirtualBox:~$ ping 192.168.76.101
PING 192.168.76.101 (192.168.76.101) 56(84) bytes of data.
64 bytes from 192.168.76.101: icmp_seq=1 ttl=63 time=1.94 ms
64 bytes from 192.168.76.101: icmp_seq=2 ttl=63 time=1.43 ms
64 bytes from 192.168.76.101: icmp_seq=3 ttl=63 time=1.74 ms

```

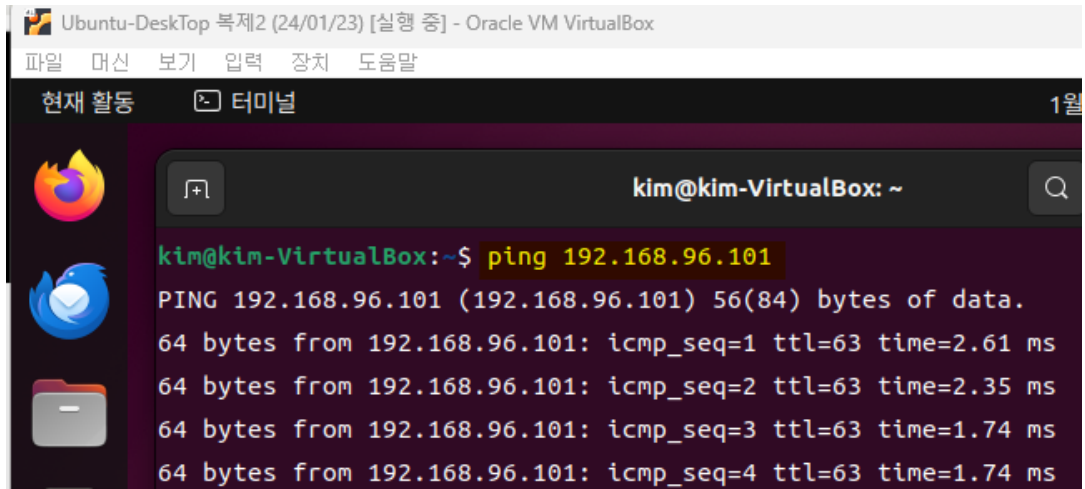
- was → web

```

kim@kim-VirtualBox:~$ ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=63 time=1.69 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=63 time=1.40 ms
^C
--- 192.168.56.101 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1006ms
rtt min/avg/max/mdev = 1.400/1.544/1.689/0.144 ms

```

- was → db



The screenshot shows a terminal window titled "Ubuntu-Desktop 복제2 (24/01/23) [실행 중] - Oracle VM VirtualBox". The terminal prompt is "kim@kim-VirtualBox: ~". The command "ping 192.168.96.101" has been entered and executed. The output shows four successful ping responses, each with 64 bytes of data, an icmp_seq number, a ttl of 63, and a response time in milliseconds.

```
kim@kim-VirtualBox:~$ ping 192.168.96.101
PING 192.168.96.101 (192.168.96.101) 56(84) bytes of data.
64 bytes from 192.168.96.101: icmp_seq=1 ttl=63 time=2.61 ms
64 bytes from 192.168.96.101: icmp_seq=2 ttl=63 time=2.35 ms
64 bytes from 192.168.96.101: icmp_seq=3 ttl=63 time=1.74 ms
64 bytes from 192.168.96.101: icmp_seq=4 ttl=63 time=1.74 ms
```

4. window → 메인(정적) 페이지 접근

<https://enginnersnack.tistory.com/15> 참고함

- 먼저 ping 통신 해보기
 - but 안되었음 ⇒ 중간에 U-D1 서버 꺼졌는데, 라우팅 설정 다 날라감
 - 라우팅 다시 해주기



iptables 불러오는 방법,,,? (찾아보기,,)

Ubuntu-Server1 복제(router) (스냅샷 1/25) [실행 중] - Oracle VM VirtualBox

```
파일  마신  보기  입력  장치  도움말
-A ufw-not-local -m addrtype --dst-type MULTICAST -j RETURN
-A ufw-not-local -m addrtype --dst-type BROADCAST -j RETURN
-A ufw-not-local -m limit --limit 3/min --limit-burst 10 -j ufw-logging-deny
-A ufw-not-local -j DROP
-A ufw-skip-to-policy-forward -j DROP
-A ufw-skip-to-policy-input -j DROP
-A ufw-skip-to-policy-output -j ACCEPT
-A ufw-track-output -p tcp -m conntrack --ctstate NEW -j ACCEPT
-A ufw-track-output -p udp -m conntrack --ctstate NEW -j ACCEPT
-A ufw-user-input -p tcp -m tcp --dport 22 -j ACCEPT
-A ufw-user-input -p tcp -m tcp --dport 80 -j ACCEPT
-A ufw-user-input -p tcp -m tcp --dport 443 -j ACCEPT
-A ufw-user-input -p udp -m udp --dport 443 -j ACCEPT
-A ufw-user-input -p tcp -m tcp --dport 53 -j ACCEPT
-A ufw-user-input -p udp -m udp --dport 53 -j ACCEPT
-A ufw-user-input -p tcp -m tcp --dport 53 -j ACCEPT
-A ufw-user-input -p udp -m udp --dport 53 -j ACCEPT
-A ufw-user-input -p tcp -m tcp --dport 2222 -j ACCEPT
-A ufw-user-input -p udp -m udp --dport 2222 -j ACCEPT
-A ufw-user-input -p tcp -m tcp --dport 3306 -j ACCEPT
-A ufw-user-input -p udp -m udp --dport 3306 -j ACCEPT
-A ufw-user-input -p tcp -m tcp --dport 30306 -j ACCEPT
-A ufw-user-input -p udp -m udp --dport 30306 -j ACCEPT
-A ufw-user-input -p tcp -m tcp --dport 22 -j ACCEPT
-A ufw-user-input -p udp -m udp --dport 22 -j ACCEPT
-A ufw-user-limit -m limit --limit 3/min -j LOG --log-prefix "[UFW LIMIT BLOCK] "
-A ufw-user-limit -j REJECT --reject-with icmp-port-unreachable
-A ufw-user-limit-accept -j ACCEPT
COMMIT
# Completed on Tue Jan 30 11:57:54 2024
ubuntu@ubuntu:~$ ipt
iptables                                iptables-nft                          iptables-save
iptables-apply                        iptables-nft-restore                  iptables-translate
iptables-legacy                      iptables-nft-save                    iptables-xml
iptables-legacy-restore              iptables-restore                      iptunnel
iptables-legacy-save                 iptables-restore-translate
ubuntu@ubuntu:~$ iptables-restore< test.txt
```

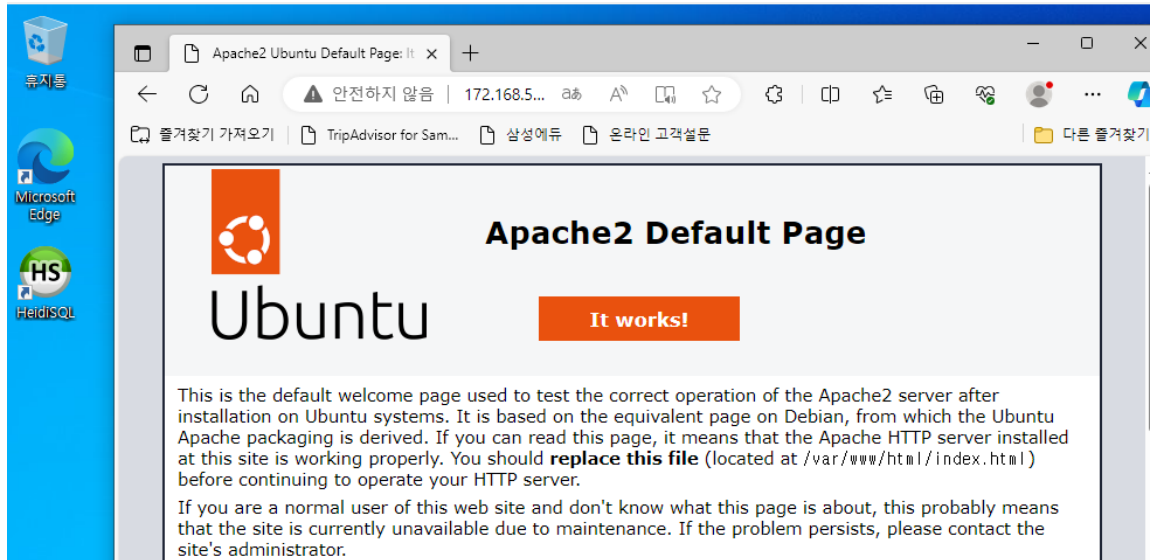
◦ ping 통신 성공

```
C:\Windows\system32>ping 172.168.56.254

Ping 172.168.56.254 32바이트 데이터 사용:
172.168.56.254의 응답: 바이트=32 시간=10ms TTL=255
172.168.56.254의 응답: 바이트=32 시간=10ms TTL=255
172.168.56.254의 응답: 바이트=32 시간=5ms TTL=255

172.168.56.254에 대한 Ping 통계:
    패킷: 보냄 = 3, 받음 = 3, 손실 = 0 (0% 손실),
    왕복 시간(밀리초):
        최소 = 5ms, 최대 = 10ms, 평균 = 8ms
Control-C
^C
C:\Windows\system32>
```

• ip로 접근하기

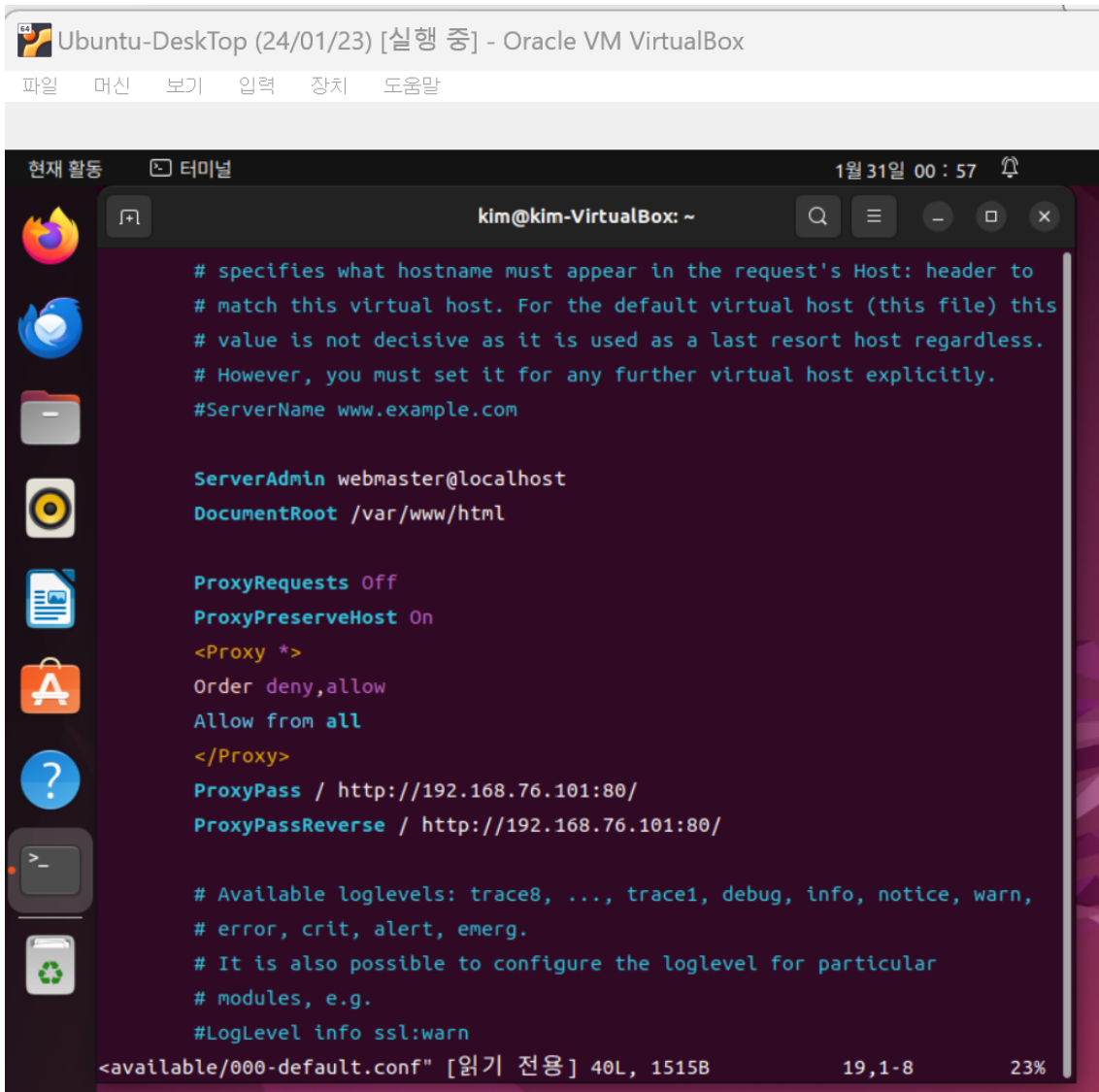


5. 동적 페이지로 접근

- `vi /etc/apache2/sites-available/000-default.conf` //apache 서버의 프록시 설정 파일에 들어가기

```
kim@kin-VirtualBox:/etc/apache2/sites-available$ sudo vim 000-default.conf
```

- 파일 수정



```
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

ProxyRequests Off
ProxyPreserveHost On
<Proxy *>
Order deny,allow
Allow from all
</Proxy>
ProxyPass / http://192.168.76.101:80/
ProxyPassReverse / http://192.168.76.101:80/

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

<available/000-default.conf" [읽기 전용] 40L, 1515B 19,1-8 23%
```


▪ ProxyRequests

- On으로 설정할 경우 ⇒ 포워드 프록시
- Off로 설정할 경우 ⇒ 리버스 프록시

▪ ProxyPreserveHost

- Apache가 원래 호스트 헤더를 백엔드 서버로 전달하도록 함
- 이는 백엔드 서버가 응용프로그램에 액세스하는 데 사용되는 주소를 인식하도록 하기 때문에 유용

 You
ProxyPreserveHost

 ChatGPT
`ProxyPreserveHost`는 아파치 웹 서버에서 사용되는 디렉티브 중 하나로, 리버스 프록시 설정에서 호스트 헤더를 보존하도록 지시합니다. 이 디렉티브를 사용하면 프록시 서버로 전달되는 요청의 호스트 헤더가 변경되지 않고 유지됩니다.

기본적으로 아파치의 리버스 프록시는 클라이언트에서 온 요청의 호스트 헤더를 사용하여 백엔드 서버로 요청을 전달합니다. 그러나 특정 상황에서는 이 헤더를 변경하고 싶지 않을 수 있습니다. 이때 `ProxyPreserveHost`를 사용하면 됩니다.


■ <proxy *> ~ </proxy>

- 프록시 서버에 대한 기본적인 접근 규칙을 정의하는 것
- 모든 IP 주소로부터의 액세스를 허용함
- 이 설정은 보안 상의 이유로 사용되는 경우가 많으며, 필요에 따라 클라이언트의 IP 주소 or 특정 IP 주소 범위에 대한 액세스 제한을 추가로 설정할 수 있음

■ ProxyPass

- 클라이언트의 요청을 프록시 서버로 전달하는 역할을 함
- ex) ProxyPass / http://localhost:8080/와 같이 설정하면
 - 클라이언트의 요청이 http://localhost:8080/으로 전달됩니다.

 You
ProxyPass /proxy01 http://10.0.20.13:80/ 해석좀

 ChatGPT
`ProxyPass` 디렉티브는 아파치 웹 서버의 모듈 중 하나인 `mod_proxy`를 사용하여 리버스 프록시를 설정하는데 사용됩니다. 주어진 구문인 `ProxyPass /proxy01 http://10.0.20.13:80/`은 다음과 같이 해석됩니다:

- 클라이언트가 `/proxy01` 경로로 요청을 보내면, 아파치 서버는 이 요청을 `http://10.0.20.13:80/` 주소로 리디렉션(프록시)합니다.

간단하게 말하면, 클라이언트가 아파치 서버의 `/proxy01` 경로로 요청을 보내면, 아파치는 해당 요청을 `http://10.0.20.13:80/` 주소로 전달하고, `10.0.20.13` 서버에서 받은 응답을 클라이언트에게 반환합니다.

이 설정은 리버스 프록시로서 작동하며, 일반적으로 클라이언트는 아파치 서버에 요청을 보내고, 아파치 서버가 해당 요청을 백엔드 서버(`10.0.20.13:80`로 지정된 서버)로 전달합니다. 이를 통해 백엔드 서버의 콘텐츠를 클라이언트에게 제공할 수 있습니다.

▪ ProxyPassReverse

- ProxyPass와 동일한 구성이어야 함
 - 프록시 서버에서 받은 응답을 클라이언트에게 전달하기 전에 응답 헤더의 내용을 조정하는 역할을 함
 - 또한 원본 서버에서 전달된 응답 헤더의 주소나 위치 정보를 프록시 서버로 조정하여 클라이언트에게 올바른 주소나 위치 정보를 제공함
 - ex) 원본 서버에서 응답 헤더에 포함된 Location: <http://localhost:8080/>와 같은 주소를 Location: <http://example.com/>로 변경하여 클라이언트에게 전달할 수 있음
- 이후 변경된 config값 적용 및 proxy mod 활성화를 위해 명령어를 입력

```
kim@kim-VirtualBox:/etc/apache2/sites-available$ sudo a2enmod proxy
Enabling module proxy.
To activate the new configuration, you need to run:
    systemctl restart apache2
kim@kim-VirtualBox:/etc/apache2/sites-available$ sudo a2enmod proxy_http
Considering dependency proxy for proxy_http:
Module proxy already enabled
Enabling module proxy_http.
To activate the new configuration, you need to run:
    systemctl restart apache2
kim@kim-VirtualBox:/etc/apache2/sites-available$ systemctl restart apache2
kim@kim-VirtualBox:/etc/apache2/sites-available$
```



You

```
a2enmod proxy
a2enmod proxy_http
systemctl restart apache2
```



ChatGPT

제공하신 명령어들은 아파치 웹 서버에서 `proxy` 모듈과 `proxy_http` 모듈을 활성화하고, 변경사항을 적용하기 위해 아파치 서버를 재시작하는 명령어들입니다.

1. `a2enmod proxy`: 이 명령어는 아파치 서버의 `proxy` 모듈을 활성화합니다. `proxy` 모듈은 리버스 프록시와 관련된 기능을 제공합니다.
2. `a2enmod proxy_http`: 이 명령어는 아파치 서버의 `proxy_http` 모듈을 활성화합니다. `proxy_http` 모듈은 HTTP 프로토콜을 사용하는 리버스 프록시 설정을 가능케 합니다.
3. `systemctl restart apache2`: 이 명령어는 아파치 서버를 재시작합니다. 설정 변경을 적용하기 위해 아파치 서버를 다시 시작하는 것이 필요합니다.

위의 명령어들을 사용하면 아파치 웹 서버에서 리버스 프록시 설정을 하기 위한 필수 모듈들을 활성화하고, 변경사항을 적용할 수 있습니다. 설정이 제대로 되었다면, 이제 아파치 서버는 클라이언트의 요청을 백엔드 서버로 전달할 수 있게 됩니다.



- Web에서 접속
 - apache를 재시작 한 후 WEB에서 본인의 IP로 접속하기
⇒ WAS 페이지가 뜸

