

Lab. 실습환경 구축하기

설치 전 과정

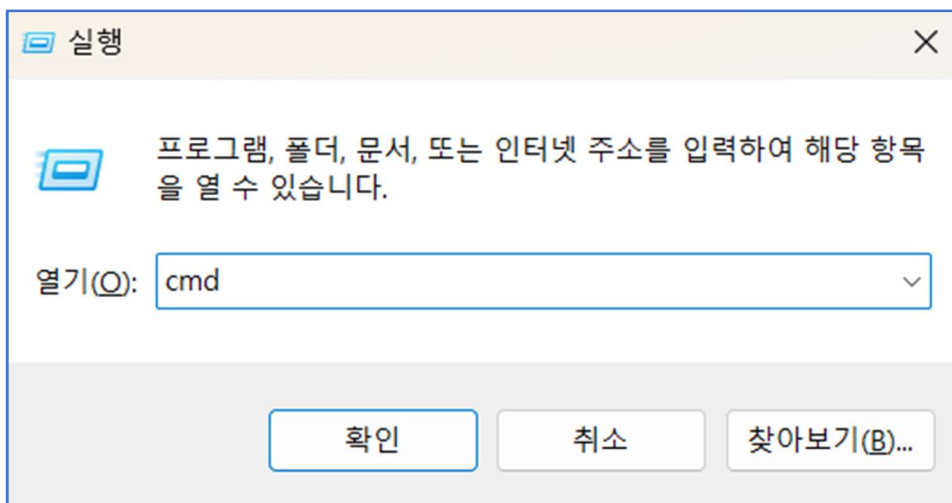
1. 권장 실습 환경

- ① Windows 11
- ② 한글 계정명 사용 금지
- ③ C Drive 여유공간 50GB 이상
- ④ RAM 16GB 이상 권장
- ⑤ 안정적인 Network 환경(가급적 유선연결)

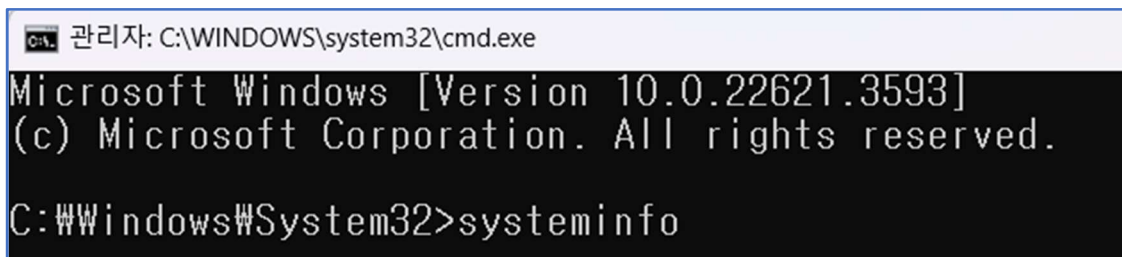
2. Hyper-V 설정 확인

※ 실습용 Windows 시스템에 Hyper-V 기능이 활성화되어 있으면 실습 불가

- ① Windows Key + r > 실행 창 > cmd > Ctrl + Shift + Enter



- ② **systeminfo**

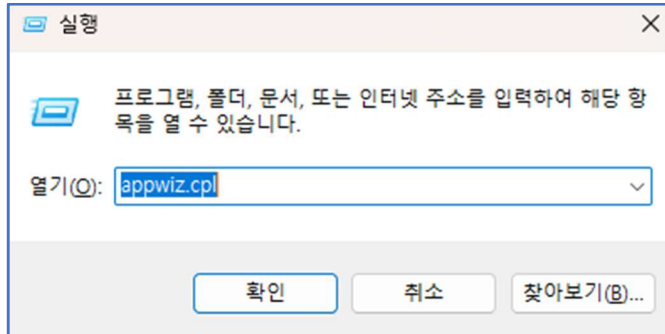


③ 출력 결과 확인

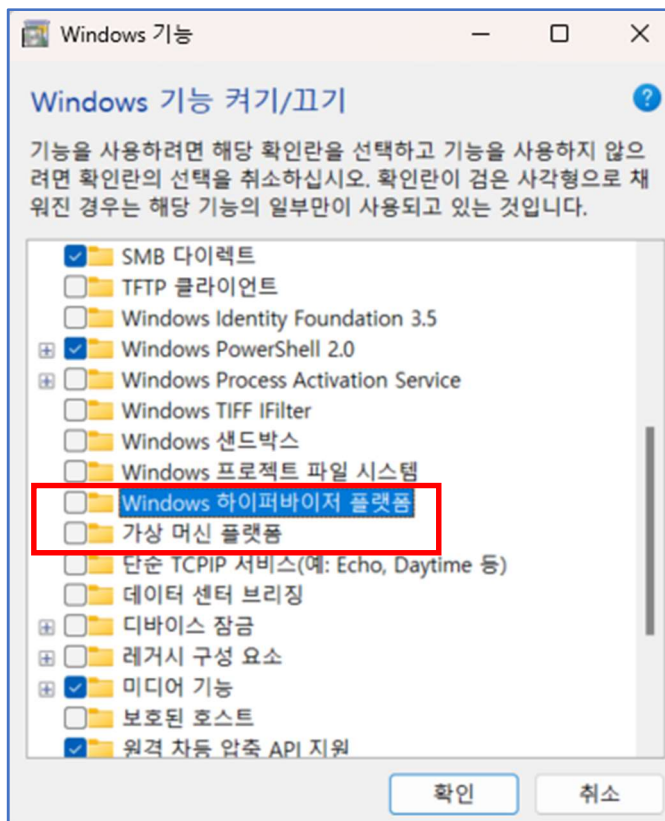
1) 추가 작업 필요

Hyper-V 요구 사항: 상태: 미디어 연결이 끊어짐
하이퍼바이저가 검색되었습니다. Hyper-V에 필요한 기능이 표시되지 않습니다.

- 실행 창에 [appwiz.cpl] 입력 > [Windows 기능 켜기/끄기]

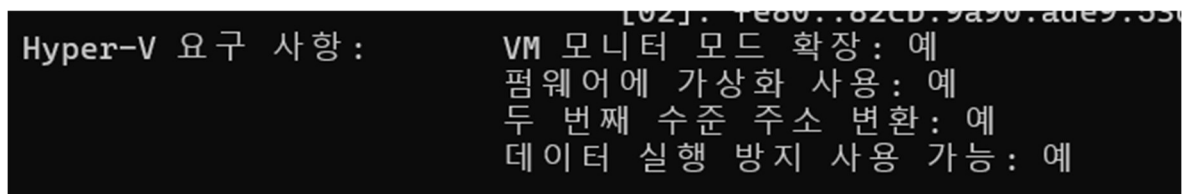


- 아래 그림과 같이 [Windows 하이퍼바이저 플랫폼]과 [가상 머신 플랫폼]을 체크 해제한다.



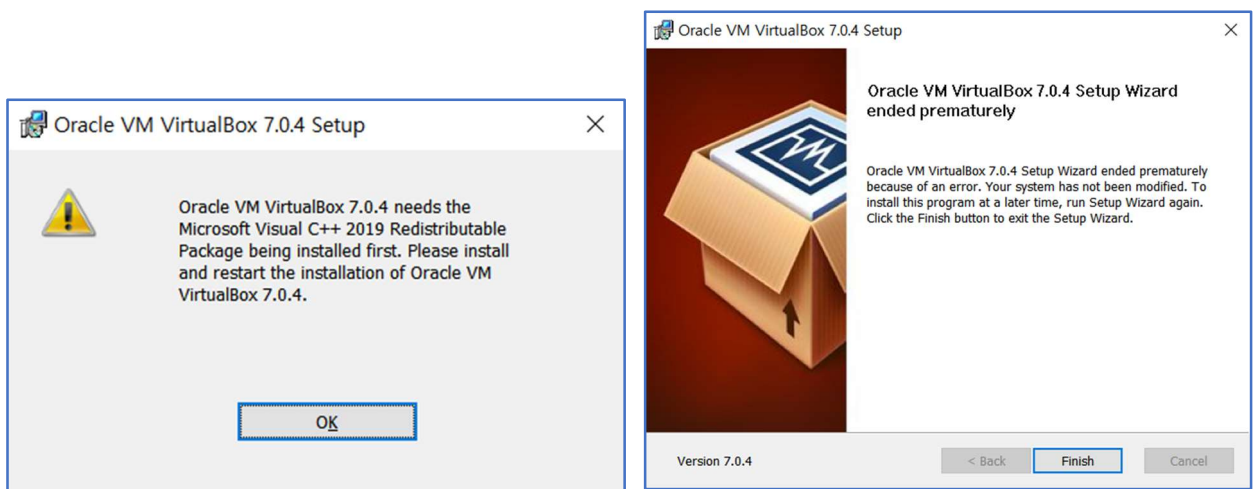
- [확인] 버튼을 클릭하고 시스템을 재부팅한다.

2) 정상인 경우



3. Oracle VirtualBox 설치

- ① VirtualBox 다운로드 및 설치 (VMware, Hyper-V 제거 필요)
- ② <https://www.virtualbox.org>, VirtualBox-7.0.6-155176-Win.exe
- ③ 설치파일 다운로드 후 관리자 권한으로 실행 (한글 계정명 사용 금지)
- ④ Microsoft Visual C++ 2019 Redistributable Package Error 발생 시



- Microsoft Visual C++ 2019 Redistributable Package 설치 후 진행

<https://learn.microsoft.com/en-US/cpp/windows/latest-supported-vc-redist?view=msvc-170>

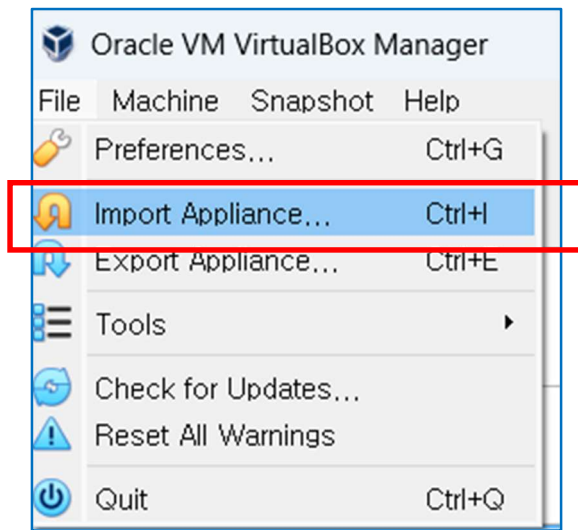
- VC_redist.x64.exe 설치 후 다시 VirtualBox 설치할 것

⑤ VirtualBox 호스트키 조합 설정

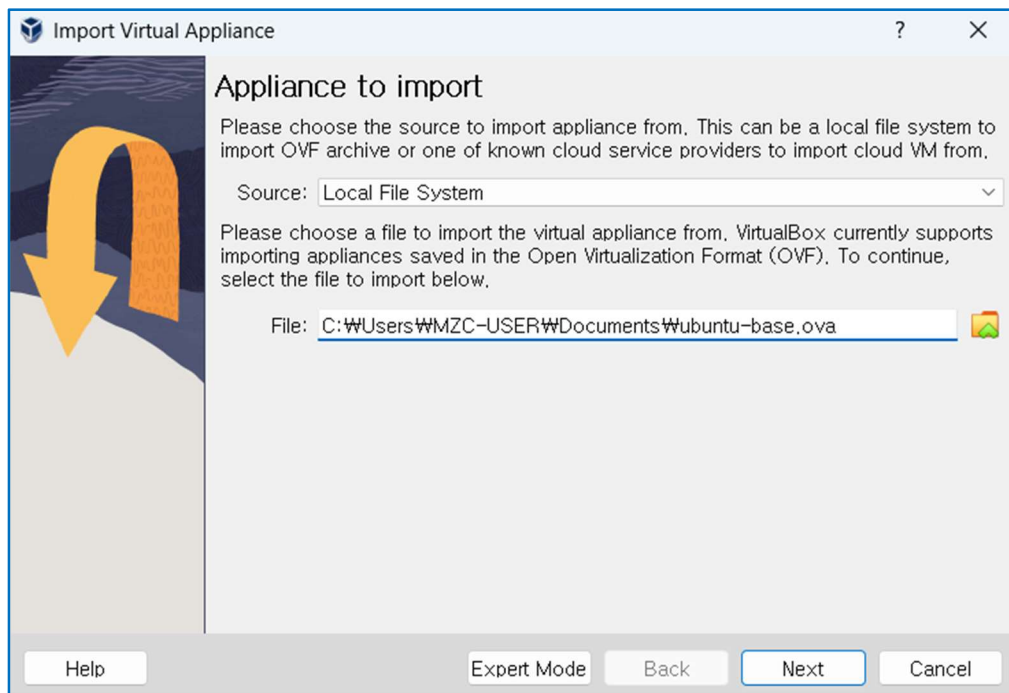
- 파일 > 환경 설정 > 입력 > 가상 머신
- 호스트 키 조합 > **F12**

4. Ubuntu VM Import – Ubuntu Linux Server 22.04 LTS

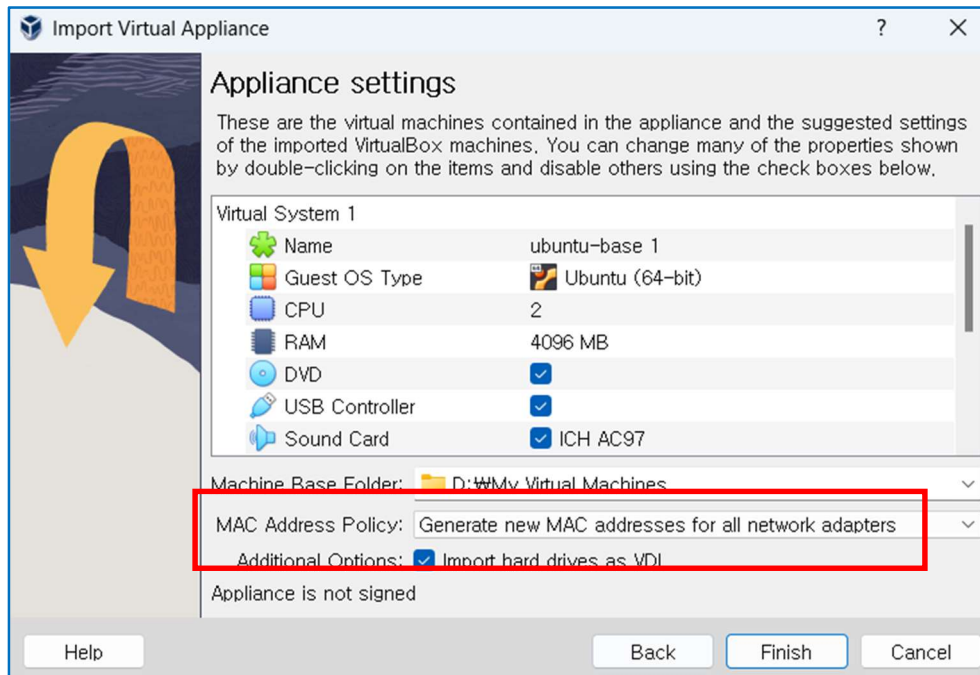
- ① [파일] > [가상 시스템 가져오기(Import Appliance)] > 파일



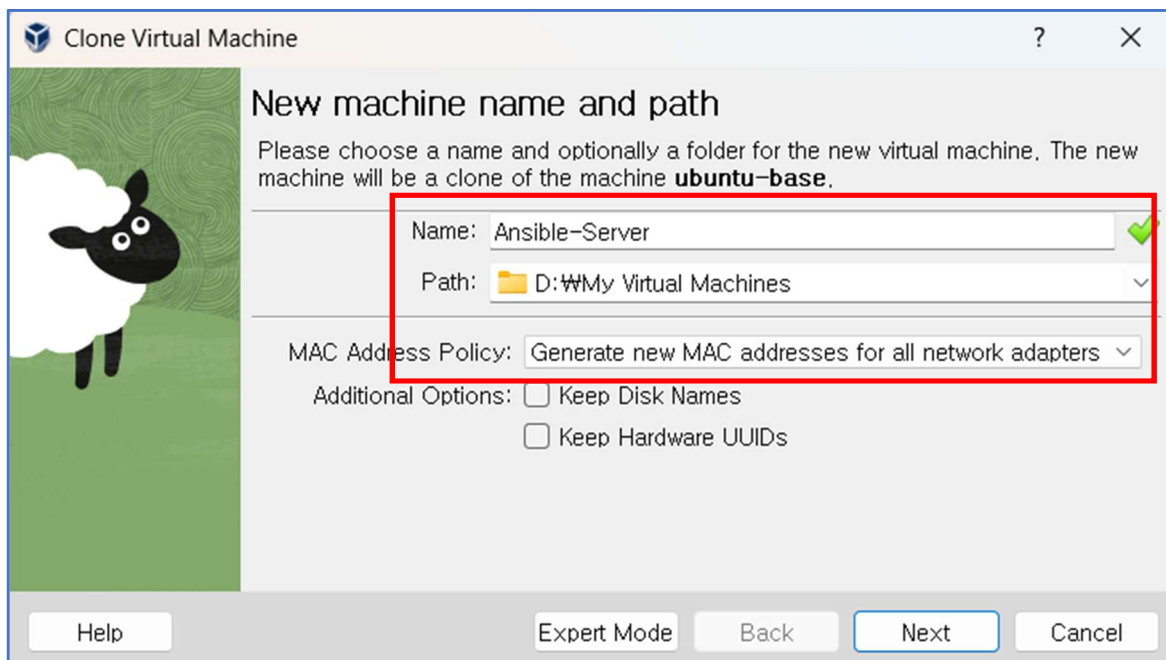
- ② "ubuntu-base.ova" 선택 > [열기] > [다음]



- ③ [Appliance settings] 창에서, [MAC 주소 정책(MAC Address Policy)] : [모든 네트워크 어댑터의 새 MAC 주소 생성(Generate new MAC addresses for all network adapters)] → 반드시 선택하여 변경할 것 > [Finish]



5. Import한 ubuntu-base 가상 머신을 복제한다. “ubuntu-base” 마우스 오른쪽 버튼을 클릭하여 [복제 (Clone)]를 클릭한다. 다음과 같이 [Name]은 “Ansible-Server”로 입력하고, 반드시, [MAC Address Policy] 항목은 “Generate new MAC addresses for all network adapters”를 선택한다. 그리고 [Next] 버튼을 클릭한다.



6. [Clone type] 창에서 [Full clone]을 선택하고 [Finish] 버튼을 클릭하여 새 VM을 생성한다.

7. VirtualBox - NatNetwork 설정(192.168.137.0/24)

- ① [도구(Tools)] > [network] > [만들기(Create)] > NatNetwork
- ② [General Options] > [IP4 Prefix] : 192.168.137.0/24 > [적용(Apply)]
- ③ [포트 포워딩(Port Forwarding)] > 새 포트 포워딩 규칙 추가(+)
 - [이름] : ubuntu
 - [호스트 IP] : 192.168.56.1
 - [호스트 포트] : 50
 - [게스트 IP] : 192.168.137.50
 - [게스트 포트] : 22

General Options		Port Forwarding				
IPv4		IPv6				
Name	Protocol	Host IP	Host Port	Guest IP	Guest Port	
Ansible-Server	TCP	192,168,56,1	50	192,168,137,50	22	

- ④ [적용(Apply)] 버튼 클릭

8. Ubuntu VM IP 설정 및 Hostname 변경하기

- ① 로그인 후, 00-installer-config.yaml 파일 수정

vi /etc/netplan/00-installer-config.yaml ← 각 라인의 들여쓰기는 반드시 2칸이다.

```
# This is the network config written by 'subiquity'
network:
  renderer: NetworkManager
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.137.50/24
      routes:
        - to: default
          via: 192.168.137.1
      nameservers:
        addresses: [8.8.8.8,8.8.7.7]
  version: 2
```

netplan try

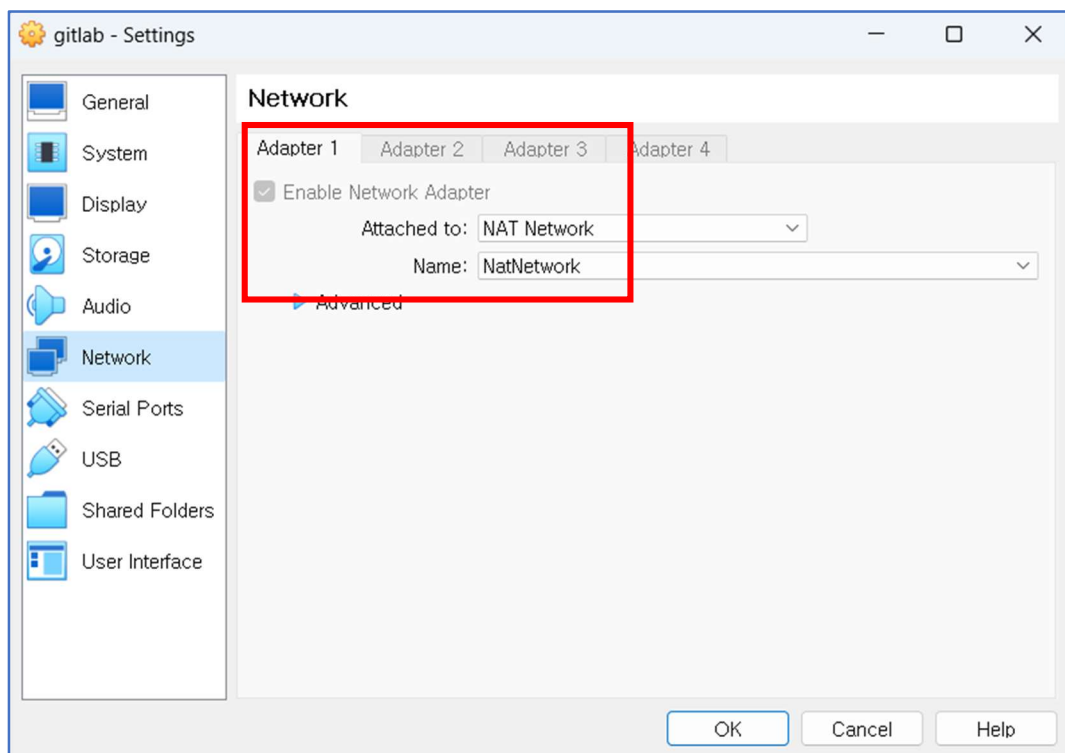
```
# netplan apply
```

```
# ip a
```

```
ubuntu@ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:17:b9:cc brd ff:ff:ff:ff:ff:ff
    inet 192.168.137.50/24 brd 192.168.137.255 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe17:b9cc/64 scope link
        valid_lft forever preferred_lft forever
```

② Ubuntu VM Network Adapter 변경하기

- Ubuntu VM > [Settings] > [Network] > [Adapter1] > [Attached to] : NAT에서 NAT Network로 변경
- 이름이 NatNetwork로 변경 확인 [OK] 버튼 클릭하여 적용



③ 호스트 이름은 다음의 명령으로 변경한다.

```
# hostnamectl set-hostname ansible-server
```

④ 호스트 이름 변경 후 확인한다.

```
# hostname
```



```
root@ubuntu:~# hostname
ansible-server
root@ubuntu:~#
```

- ⑤ /etc/hosts 파일 수정하기

```
# vi /etc/hosts
```

```
127.0.0.1 localhost
127.0.1.1 ansible-server
192.168.137.50 ansible-server.example.com
```

- ⑥ Network Test를 위해 다음의 명령을 수행한다.

```
# apt update
```

- ⑦ 만일 오류가 발생하면 다음의 명령을 수행한다.

```
# vi /etc/resolv.conf
```

```
nameserver 8.8.8.8
```

```
# This is /run/systemd/resolve/stub-resolv.conf managed by man:systemd-resolved(8).
# Do not edit.
#
# This file might be symlinked as /etc/resolv.conf. If you're looking at
# /etc/resolv.conf and seeing this text, you have followed the symlink.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs should typically not access this file directly, but only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
# different way, replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.
```

```
nameserver 8.8.8.8
options edns0 trust-ad
search
```


- ⑧ 저장하면 바로 적용됨

```
# ping -c 4 www.google.com
```

```
ubuntu@ubuntu:~$ ping -c 4 www.google.com
PING www.google.com (142.250.206.196) 56(84) bytes of data:
64 bytes from kix07s07-in-f4.1e100.net (142.250.206.196): icmp_seq=1 ttl=112 time=58.8 ms
64 bytes from kix07s07-in-f4.1e100.net (142.250.206.196): icmp_seq=2 ttl=112 time=56.5 ms
64 bytes from kix07s07-in-f4.1e100.net (142.250.206.196): icmp_seq=3 ttl=112 time=56.9 ms
64 bytes from kix07s07-in-f4.1e100.net (142.250.206.196): icmp_seq=4 ttl=112 time=57.0 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3050ms
rtt min/avg/max/mdev = 56.507/57.304/58.766/0.866 ms
ubuntu@ubuntu:~$ _
```

- ⑨ Network Test를 위해 다음의 명령을 다시 수행한다.

```
# apt update
```

```
root@ubuntu:~# apt update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:2 http://kr.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://kr.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1,475 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [254 kB]
Get:6 http://kr.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:7 http://kr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,687 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [854 kB]
Get:9 http://kr.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [313 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [165 kB]
Get:11 http://kr.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,076 kB]
Get:12 http://kr.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [247 kB]
Get:13 http://kr.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [67.1 kB]
Get:14 http://kr.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [27.2 kB]
Fetched 6,548 kB in 6s (1,013 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
81 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ubuntu:~#
```

9. **Ansible-Server**와 동일한 순서로 다음과 같이 관리 노드가 될 VM 4대를 생성한다. **Ansible-Server**와 달리 관리 노드의 VM의 하드웨어 사양은 다음과 같다.

- ① Memory : 2GB
- ② CPU : 1
- ③ Network Adapter : NatNetwork

10. **Ansible-Server**를 포함하여 관리 노드 VM의 각 Network 설정은 다음과 같다.

서버이름	ansible-server	servera	serverb	serverc	serverd
Hostname	ansible-server.example.com	servera.example.com	serverb.example.com	serverc.example.com	serverd.example.com
IP Address	192.168.137.50	192.168.137.51	192.168.137.52	192.168.137.53	192.168.137.54

11. 각각의 관리노드는 다음과 같은 순서대로 설정한다.

- ① IP Address 수정 → # **vi /etc/netplan/00-00-installer-config.yaml**
- ② 서버이름 변경 → # **hostnamectl set-hostname {각 관리노드의 서버이름}**
- ③ Hostname 수정 → # **vi /etc/hosts**
- ④ Name Server 변경 → # **vi /etc/resolv.conf**
- ⑤ # **apt update** 및 # **ping -c 4 www.google.com**
- ⑥ VirtualBox에서 다음과 같이 Port Forwarding 설정한다.

General Options		Port Forwarding				
IPv4		IPv6				
Name	Protocol	Host IP	Host Port	Guest IP	Guest Port	
servera	TCP	192.168.56.1	51	192.168.137.51	22	
serverb	TCP	192.168.56.1	52	192.168.137.52	22	
serverc	TCP	192.168.56.1	53	192.168.137.53	22	

12. Ubuntu VM 종료 후 "**Ansible-Server**" 머신의 [시작] > [Headless Start] 선택

13. SSH Client Tool로 VSCode를 사용한다. VSCode를 설치 후 다음 **Extension**을 설치한다.

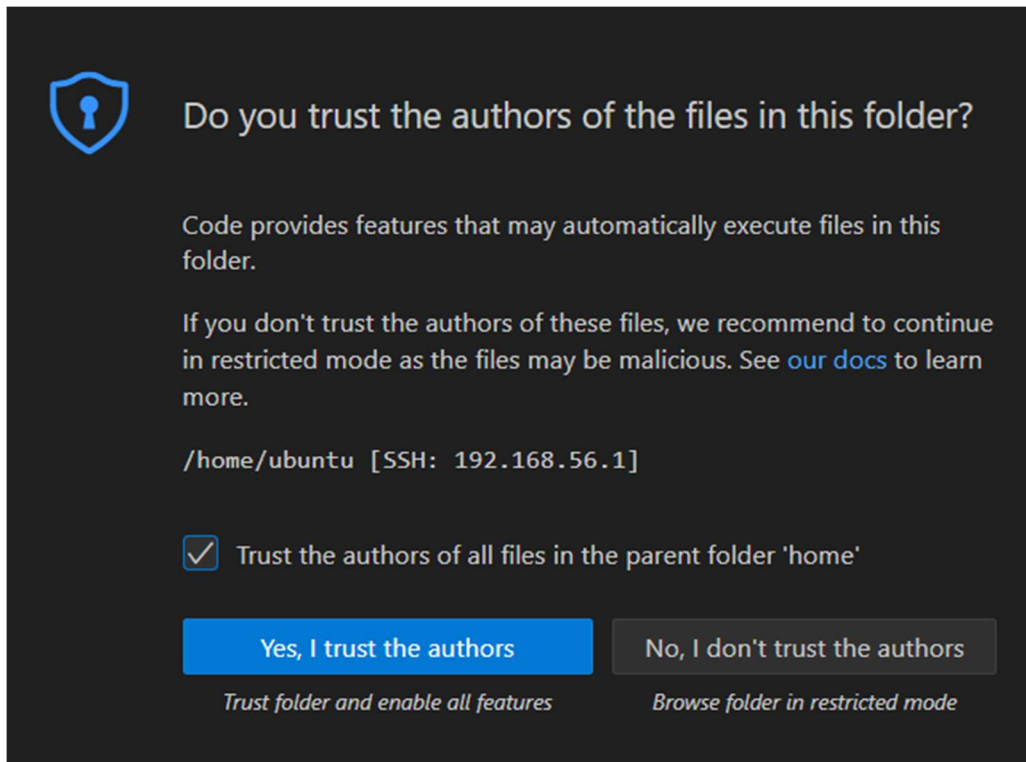
- ① **Remote Development**
- ② **yaml**
- ③ **ansible**

14. 먼저, VSCode로 **Ansible-Server**에 원격 연결한다.

- ① **[+Add New SSH Host]**
- ② ssh [ubuntu@192.168.56.1:50](ssh://ubuntu@192.168.56.1:50)
- ③ C:\Users\W{컴퓨터계정}\.ssh\config
- ④ Linux > Continue > {패스워드}

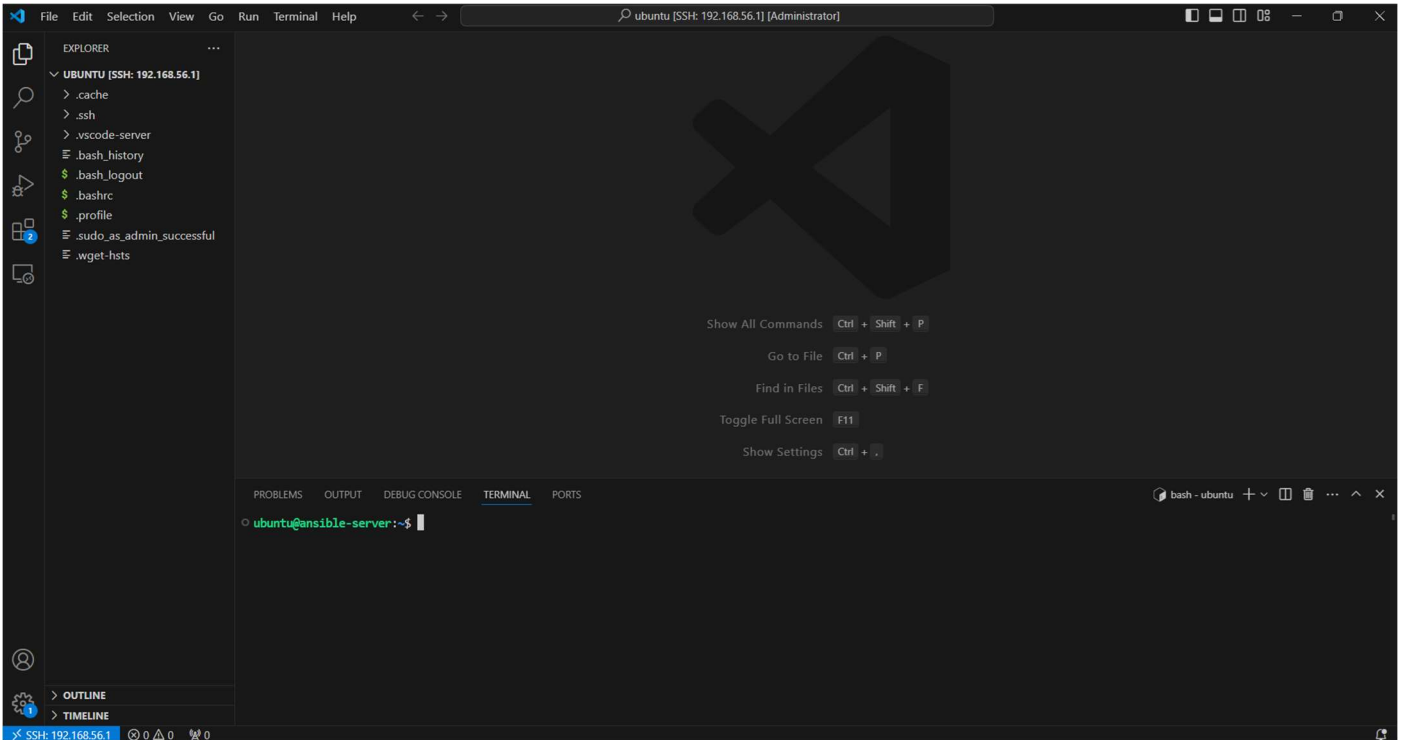
15. VSCode가 원격으로 Ansible-Server에 연결됐으면 VSCode의 좌측 패널의 **[Explorer]**을 클릭하고 **[Open Folder]** 버튼을 클릭한다. 기본 경로는 **/home/ubuntu/**이다. **[OK]** 버튼을 클릭한다.

- ① Linux > {패스워드} > **[Trust the authors of all files in the parent folder "home"]** 체크
- ② **[Yes, I trust the authors]** 버튼 클릭

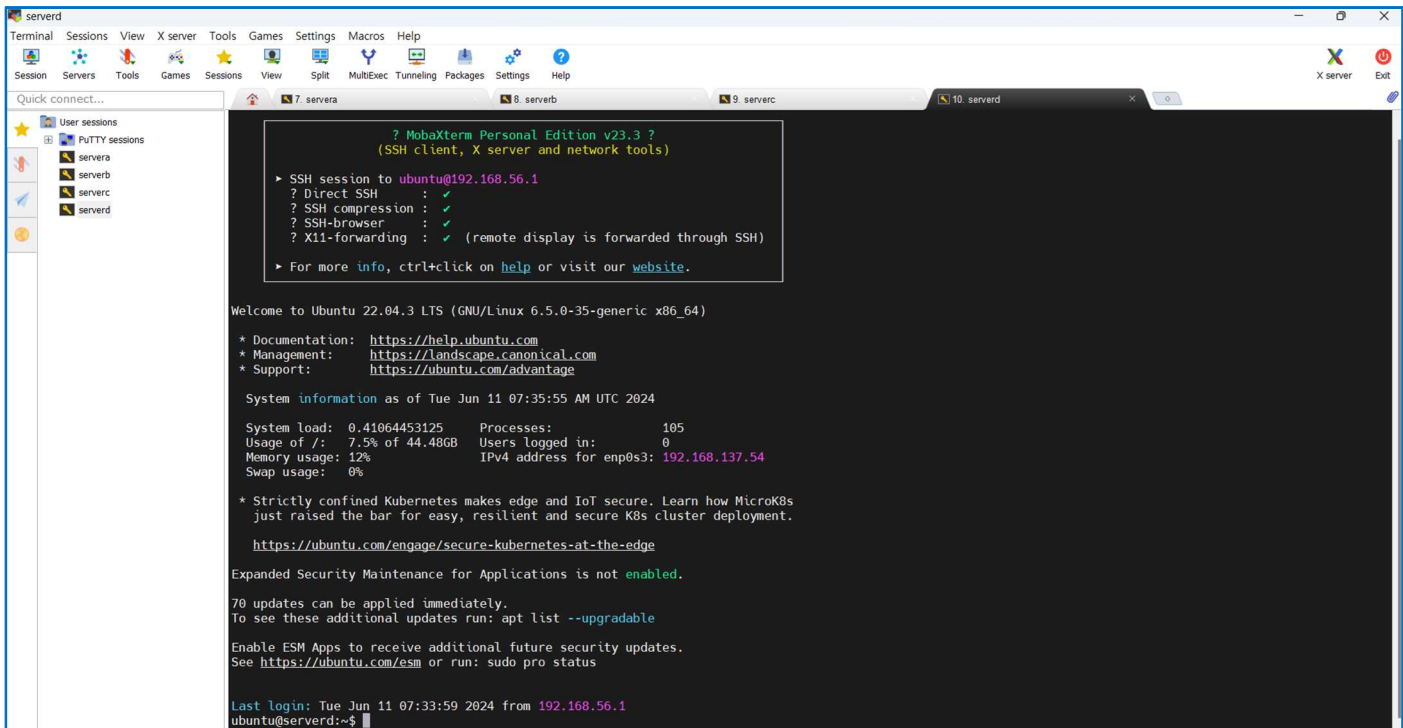


16. VSCode의 메뉴 중 **[Terminal]** > **[New Terminal]**를 클릭하여 새 터미널을 오픈한다.

17. 지금까지 VSCode를 이용한 설정의 그림은 다음과 같다.



18. 각각의 관리노드의 SSH 연결은 MobaXterm을 사용한다. 각각 관리노드를 연결하면 다음과 같다.



19. Visual Studio Code를 사용하여 연결한 **Ansible-Server**에서 다음과 같이 host 파일을 설정한다.

```
● ubuntu@ansible-server:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 ansible-server

192.168.137.50 ansible-server.example.com
192.168.137.51 servera.example.com
192.168.137.52 serverb.example.com
192.168.137.53 serverc.example.com
192.168.137.54 serverd.example.com
○ ubuntu@ansible-server:~$
```

20. 각각의 관리노드에 대해 ping 테스트를 수행하여 제대로 각 관리노드를 찾아가는지 확인한다.

```
$ ping -c 4 servera.example.com
```

```
$ ping -c 4 serverb.example.com
```

```
$ ping -c 4 serverc.example.com
```

```
$ ping -c 4 serverd.example.com
```

Ansible-Server 설정하기

1. 다음과 같이 Ansible을 Ansible-Server에 설치한다. (ref

https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html#installing-ansible-on-ubuntu)

```
$ sudo apt update
```

```
$ sudo apt install software-properties-common
```

```
$ sudo add-apt-repository --yes --update ppa:ansible/ansible
```

```
$ sudo apt install ansible
```

2. Ansible이 정상적으로 설치되었는지 확인한다.

```
$ ansible --version
```

```
ubuntu@ansible-server:~$ ansible --version
ansible [core 2.16.7]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
ubuntu@ansible-server:~$
```

3. /etc/hosts 파일을 열고 다음과 같이 각 hostname을 입력한다.

```
$ sudo vi /etc/hosts
```

```
192.168.137.50    ansible-server.example.com
```

```
192.168.137.51    servera.example.com
```

```
192.168.137.52    serverb.example.com
```

```
192.168.137.53    serverc.example.com
```

```
192.168.137.54    serverd.example.com
```

```
127.0.0.1 localhost
```

```
127.0.1.1 ansible-server
```

```
192.168.137.50    ansible-server.example.com
```

```
192.168.137.51    servera.example.com
```

```
192.168.137.52    serverb.example.com
```

```
192.168.137.53    serverc.example.com
```

```
192.168.137.54    serverd.example.com
```

4. **Ansible-server**에서 현재 계정(ubuntu)의 홈디렉토리에 **workspace**라는 작업 폴더를 생성한다.

```
$ mkdir workspace
```

5. **workspace** 디렉토리에 다음과 같이 **inventory**파일과 **ansible.cfg** 파일을 vi 에디터를 사용하여 생성한다.

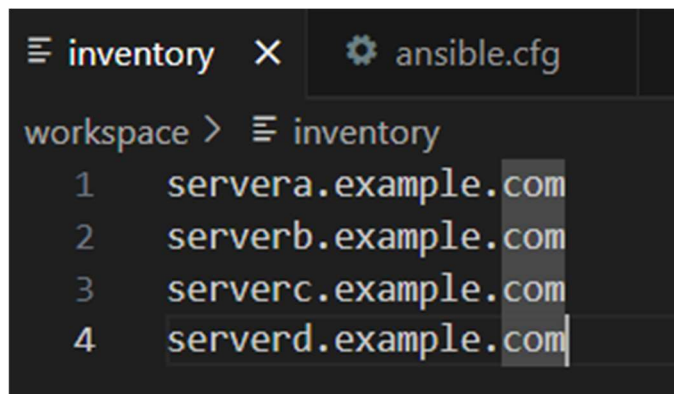
```
$ vi inventory
```

```
servera.example.com
```

```
serverb.example.com
```

```
serverc.example.com
```

```
serverd.example.com
```




```
$ vi ansible.cfg

[defaults]

inventory = ./inventory

remote_user = ubuntu

ask_pass = true

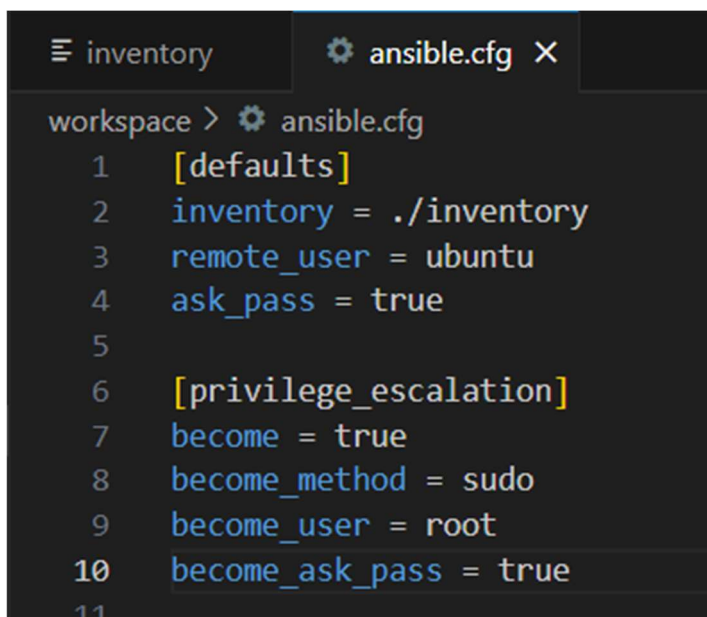

[privilege_escalation]

become = true

become_method = sudo

become_user = root

become_ask_pass = true
```



```
workspace > ansible.cfg
1  [defaults]
2  inventory = ./inventory
3  remote_user = ubuntu
4  ask_pass = true
5
6  [privilege_escalation]
7  become = true
8  become_method = sudo
9  become_user = root
10 become_ask_pass = true
11
```

6. **ansible** 명령을 통해 Ansible-Server에서 각각의 관리노드를 제대로 처리할 수 있는지 다음의 명령으로 확인해 보자. 현재 config file은 위에서 방금 생성한 ansible.cfg 임을 확인할 수 있다.

```
$ ansible --version
```

```
● ubuntu@ansible-server:~/workspace$ ansible --version
ansible [core 2.16.7]
  config file = /home/ubuntu/workspace/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
```

7. **ansible** 명령을 통해 각각의 관리노드의 IP 주소를 읽어오는 명령이다. 모든 관리노드의 IP Address를 읽어온다. 성공하면 제대로 **Control-Node**와 **Management-Node**를 설치한 것이다.

```
$ ansible all -m command -a 'ip addr'
```

```
● ubuntu@ansible-server:~/workspace$ ansible all -m command -a 'ip addr'
SSH password:
BECOME password[defaults to SSH password]:
serverb.example.com | CHANGED | rc=0 >>
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:4b:09:86 brd ff:ff:ff:ff:ff:ff
    inet 192.168.137.52/24 brd 192.168.137.255 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe4b:986/64 scope link
        valid_lft forever preferred_lft forever
serverd.example.com | CHANGED | rc=0 >>
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:7e:fa:79 brd ff:ff:ff:ff:ff:ff
    inet 192.168.137.54/24 brd 192.168.137.255 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe7e:fa79/64 scope link
        valid_lft forever preferred_lft forever
serverc.example.com | CHANGED | rc=0 >>
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1c:8b:c9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.137.53/24 brd 192.168.137.255 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
```