



ANSIBLE

Vagrant

■ Vagrant

- 2010.03 Mitchell Hashimoto, Opensource Project → HashiCorp
- Ruby로 개발되었으며 Linux, FreeBSD, OSX, Windows 지원
- 가상화 인스턴스 관리 SW → **Provisioning**
 - **VirtualBox (기본 프로바이더)**
 - VMware
 - KVM
 - Linux Container(LXC)
 - Docker
 - AWS
 - Azure

■ Vagrant

■ Provisioning

- 사용자의 요구에 맞게 시스템 자원을 할당, 배치, 배포한 다음 필요 시 **시스템을 즉시 사용**할 수 있는
상태로 미리 준비



- Hostname:
 - ansible-server
 - ansible-node01, node02
- IP
 - 172.20.10.10
 - 172.20.10.11
 - 172.20.10.12



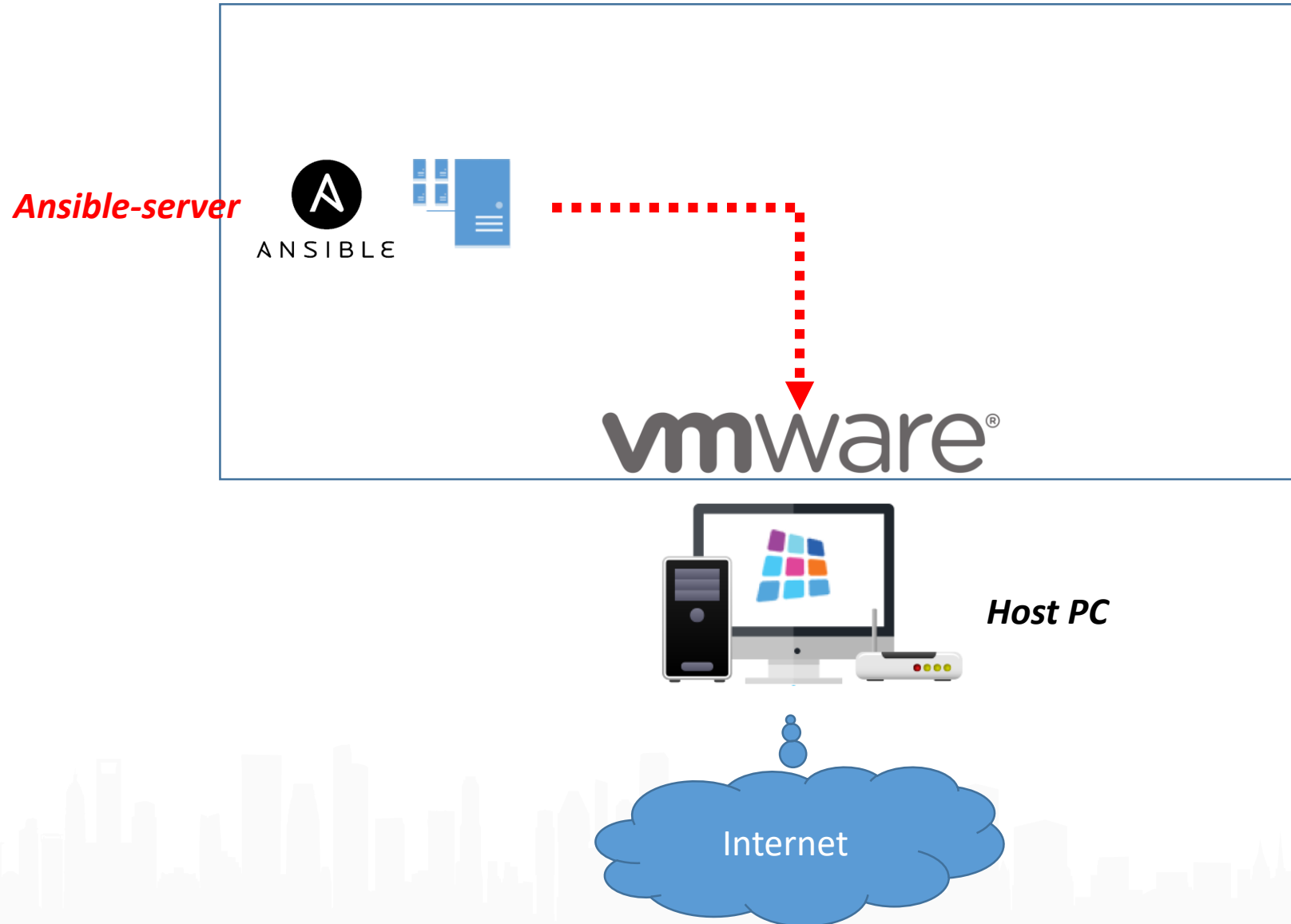
- Hostname:
 - ansible-node03
- IP
 - 172.20.10.13



- Hostname:
 - ansible-node04
- IP
 - 172.20.10.14

■ Vagrant

■ 실습 환경의 구성



■ Vagrant

■ Vagrant 동작

- 1) Box 이미지 다운로드
- 2) Box 이미지를 이용하여 프로젝트 생성
- 3) 프로젝트 최상위 디렉토리에 위치한 Vagrantfile 파일 수정
- 4) 프로젝트의 가상 인스턴스 시작
- 5) 가상 이미지 접속 및 작업
- 6) 가상 이미지 종료

■ Box 이미지

- Box 파일 조회
 - <https://app.vagrantup.com/boxes/search>

■ Vagrant

■ Vagrant 명령어

- `vagrant -v`
- `vagrant status`
- `vagrant global-status`
- `vagrant init` → Provisioning 하는 스크립트 생성
- `vagrant up` → Vagrantfile 을 읽어 Provisioning 진행
- `vagrant halt` → Host 종료
- `vagrant destroy` → Host 삭제
- `vagrant suspend`
- `vagrant resume`
- `vagrant reload`
- `vagrant ssh` → Host 접속
- `vagrant provision` → Host 설정 변경 적용

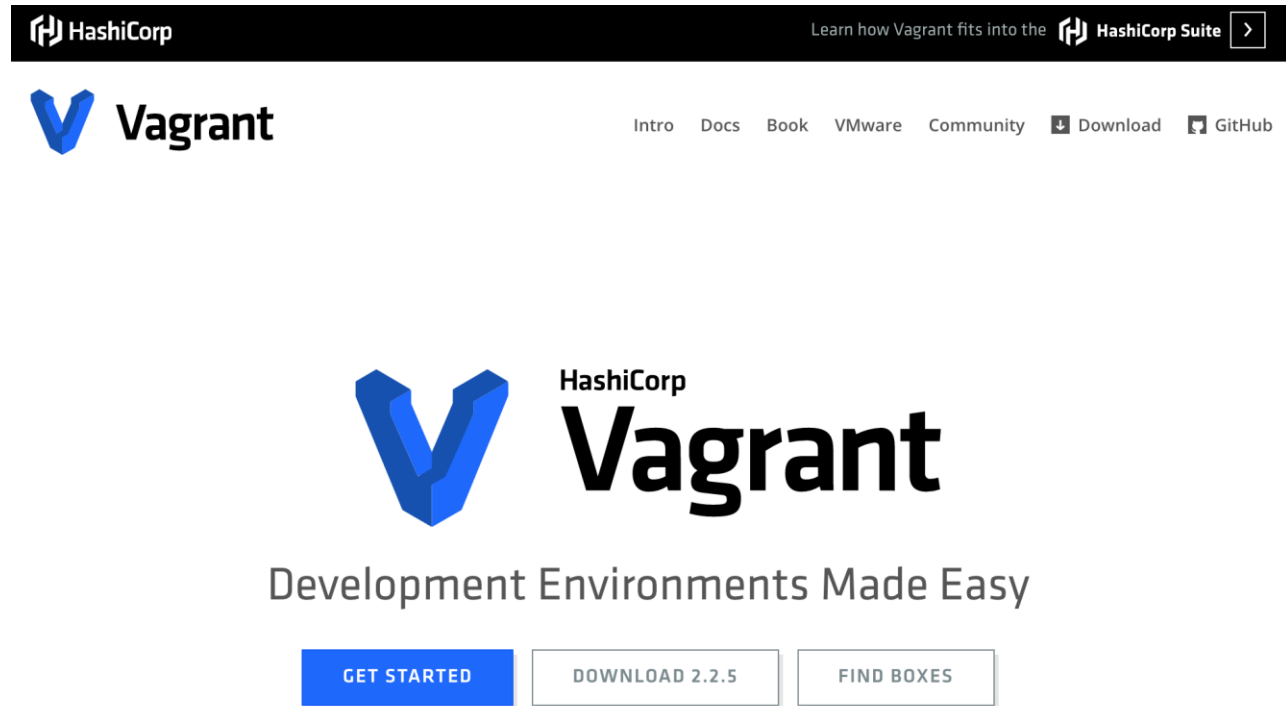
■ Vagrant

- Editor 설치

- Visual Studio Code or Notepad++ ...

- Vagrant 설치

- <https://www.vagrantup.com/>



■ Vagrant

■ Box 다운로드

- `$ mkdir ~/Work`
- `$ cd ~/Work`
- `$ vagrant init`

```
dowon@DOWON-MacBook > ~/Desktop/Work/vagrant > vagrant init
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
dowon@DOWON-MacBook > ~/Desktop/Work/vagrant > |
```

■ Vagrantfile 수정

- Ruby 언어로 작성
- <https://app.vagrantup.com/boxes/search> 에서 사용할 이미지 검색

■ Vagrant

■ Vagrantfile 수정

- 15 line

```
13 # Every Vagrant development environment requires a box. You can search for  
14 # boxes at https://vagrantcloud.com/search.  
15 config.vm.box = "base"
```

centos / 7 Vagrant box

How to use this box with Vagrant:

Vagrantfile [New](#)

```
Vagrant.configure("2") do |config|  
  config.vm.box = "centos/7"  
end
```

Vagrant

Vagrant plugin 설치

- \$ vagrant plugin install vagrant-vbguest → VirtualBox
- \$ vagrant plugin install vagrant-vmware-desktop → VMware
- ex) \$ vagrant plugin install vagrant-vmware-desktop → VMware → **license 필요**

```
✗ down@DOWON-MacBook ~/Desktop/Work/vagrant$ vagrant plugin install vagrant-vmware-desktop
Installing the 'vagrant-vmware-desktop' plugin. This can take a few minutes...
Fetching: vagrant-vmware-desktop-2.0.3.gem (100%)
Installed the plugin 'vagrant-vmware-desktop (2.0.3)'!
down@DOWON-MacBook ~/Desktop/Work/vagrant$ vagrant plugin list
vagrant-vmware-desktop (2.0.3, global)
down@DOWON-MacBook ~/Desktop/Work/vagrant$ vagrant
A valid license is required to run the Vagrant VMware
provider. Please visit http://www.vagrantup.com to purchase
a license. Once you purchase a license, you can install it
using `vagrant plugin license`.
```

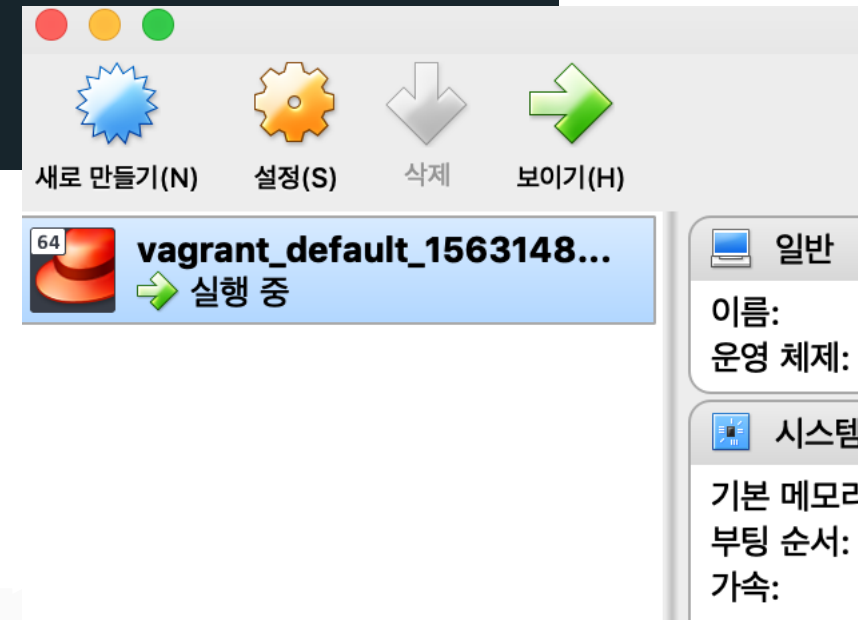
■ Vagrant

■ Vagrant up

- \$ vagrant up

```
VirtualBox Guest Additions: Look at /var/log/vboxadd-setup.log to find out what went wrong
VirtualBox Guest Additions: Starting.
Redirecting to /bin/systemctl start vboxadd.service
Redirecting to /bin/systemctl start vboxadd-service.service
Unmounting Virtualbox Guest Additions ISO from: /mnt
==> default: Checking for guest additions in VM...
==> default: Rsyncing folder: /Users/down/Desktop/Work/vagrant/ => /vagrant
down@DOWON-MacBook ~/Desktop/Work/vagrant
```

```
✗ down@DOWON-MacBook ~/Desktop/Work/vagrant vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'centos/7' could not be found. Attempting to find and install...
default: Box Provider: virtualbox
default: Box Version: >= 0
==> default: Loading metadata for box 'centos/7'
default: URL: https://vagrantcloud.com/centos/7
==> default: Adding box 'centos/7' (v1905.1) for provider: virtualbox
```



- VirtualBox에서 확인

■ Vagrant

■ Vagrant up

- \$ varant ssh
- Ansible 은 미설치

■ Vagrantfile 수정

```
8 Vagrant.configure("2") do |config|
9   config.vm.define:"ansible-server" do |cfg|
10     cfg.vm.box = "centos/7"
11     cfg.vm.provider:virtualbox do |vb|
12       vb.name="Ansible-Server(KTDS)"
13     end
14     cfg.vm.host_name="ansible-server"
15     cfg.vm.synced_folder ".", "/vagrant", disabled: true
16     cfg.vm.network "public_network", ip: "172.20.10.10"
17     cfg.vm.network "forwarded_port", guest: 22, host: 19210, auto_correct:
      false, id: "ssh"
18     cfg.vm.provision "shell", path: "bootstrap.sh"
19   end
20 end
```

■ Vagrant

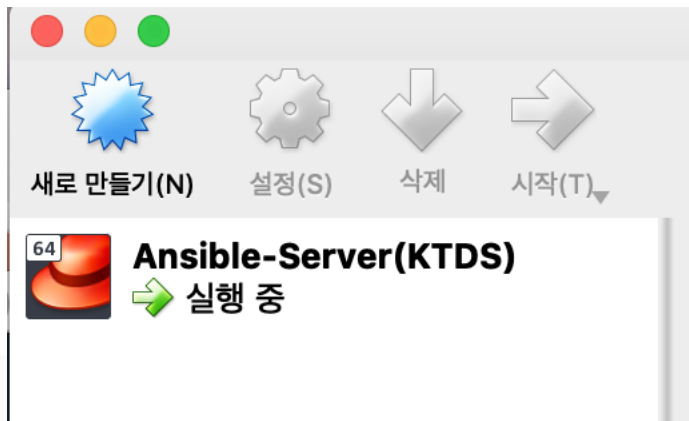
■ bootstrap.sh 파일 생성

- Vagrant script 파일

```
1  #!/usr/bin/env bash
2
3  yum install epel-release -y
4  yum install ansible -y
```

■ Vagrant up

- 기존 VM 이미지 삭제 후 재 실행



Vagrant

Vagrant ssh

- `$ vagrant ssh ansible-server`
- `[vagrant@ansible-server]$ ansible`

```
dowon@DOWON-MacBook ~/Desktop/Work/vagrant/ansible/01.centos$ vagrant ssh ansible-server
[vagrant@ansible-server ~]$
[vagrant@ansible-server ~]$
[vagrant@ansible-server ~]$
[vagrant@ansible-server ~]$ ansible
usage: ansible [-h] [--version] [-v] [-b] [--become-method BECOME_METHOD]
               [--become-user BECOME_USER] [-K] [-i INVENTORY] [--list-hosts]
               [-l SUBSET] [-P POLL_INTERVAL] [-B SECONDS] [-o] [-t TREE] [-k]
               [--private-key PRIVATE_KEY_FILE] [-u REMOTE_USER]
               [-c CONNECTION] [-T TIMEOUT]
               [--ssh-common-args SSH_COMMON_ARGS]
               [--sftp-extra-args SFTP_EXTRA_ARGS]
               [--scp-extra-args SCP_EXTRA_ARGS]
               [--ssh-extra-args SSH_EXTRA_ARGS] [-C] [--syntax-check] [-D]
               [-e EXTRA_VARS] [--vault-id VAULT_IDS]
               [--ask-vault-pass | --vault-password-file VAULT_PASSWORD_FILES]
               [-f FORKS] [-M MODULE_PATH] [--playbook-dir BASEDIR]
               [-a MODULE_ARGS] [-m MODULE_NAME]
               pattern
ansible: error: too few arguments
[vagrant@ansible-server ~]$
```

■ Playbook

- Vagrantfile 수정
 - Ansible_env_ready.yml 파일 실행 추가
- bootstrap.sh 파일 수정
 - vim plugin 관련 파일이 저장될 디렉토리 생성
 - vim과 bash 환경 설정 파일 생성
- Ansible_env_ready.yml 파일
 - yum을 통해 vim-enhanced 설치
 - yum을 통해 git 설치
 - pathogen.vim 다운로드
 - vim-ansible-yaml 다운로드 (git clone)
 - vim의 환경 설정 파일 수정 (.vimrc)
 - bash의 환경 설정 파일 수정 (.bashrc)

■ Playbook

■ Vagrantfile 파일 실행

- `$ vagrant destroy ansible-server`
- `$ vagrant up` (or `$ vagrant provision ansible-server`)

```
ansible-server: TASK [Configure vimrc] *****
ansible-server: ok: [localhost] => (item=set number)
ansible-server: ok: [localhost] => (item=execute pathogen#infect())
ansible-server: ok: [localhost] => (item=syntax on)
ansible-server:
ansible-server: TASK [Configure Bashrc] *****
ansible-server: ok: [localhost] => (item=alias vi='vim')
ansible-server: ok: [localhost] => (item=alias ans='ansible')
ansible-server: changed: [localhost] => (item=alias anp='ansible-playbook')
ansible-server:
ansible-server: PLAY RECAP *****
```

- `$ vagrant ssh ansible-server`
- `[vagrant@ansible-server]$ ls`
- `[vagrant@ansible-server]$ ans`
- `[vagrant@ansible-server]$ anp`

■ Ansible Node 추가

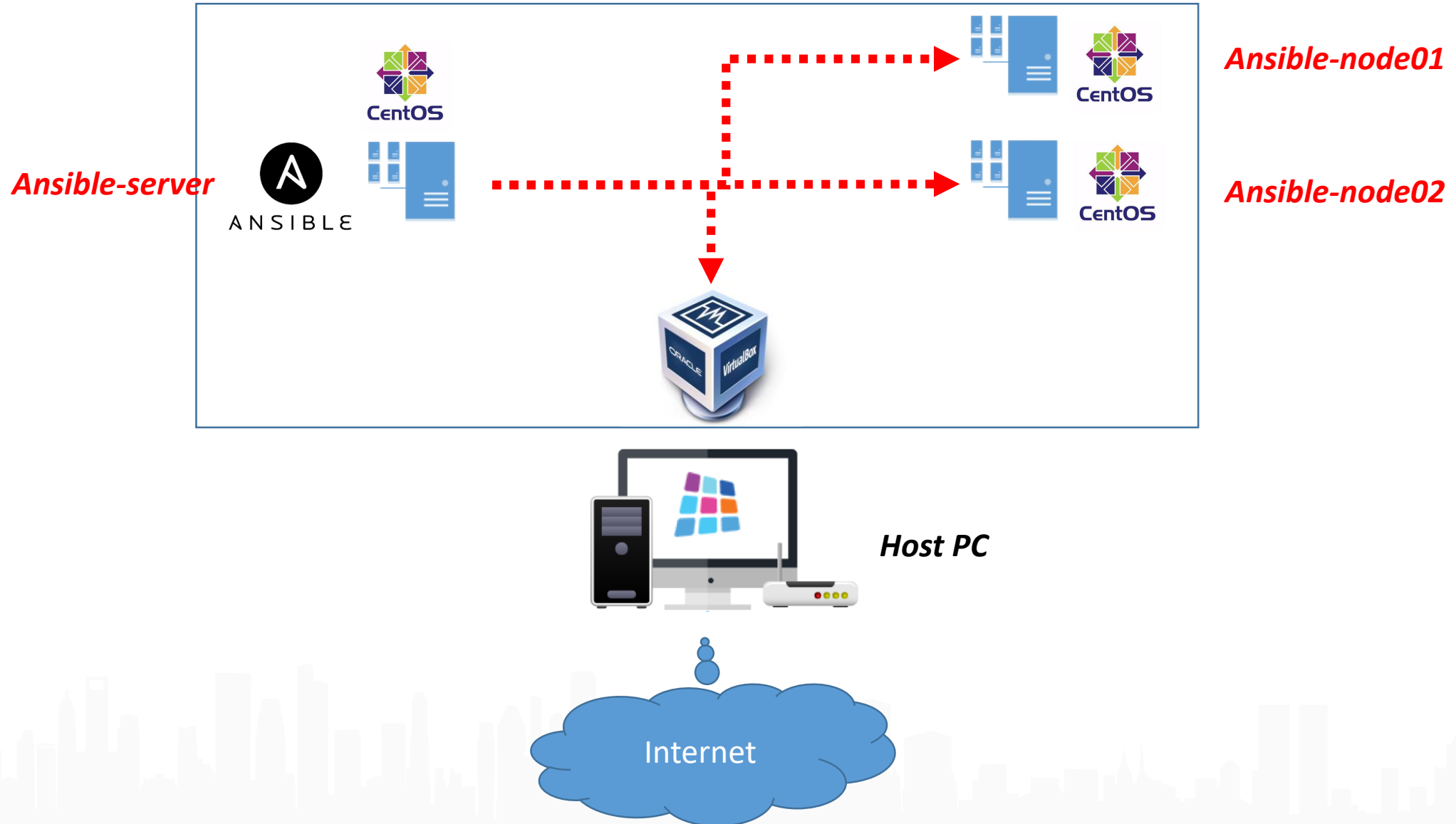
■ 다양한 os 지원

- CentOS
- ubuntu
- Windows



■ Ansible Node 추가

■ 실습 환경의 구성



■ Ansible Node 추가

■ Vagrantfile 추가

1) Ansible Node 2개 추가 (CentOS)

2) 구성 파일 작성

- Vagrant Host 이름 작성
- VirtualBox 설정 변경
 - VirtualBox에서 구분하는 Host 이름 작성
 - CPU와 메모리 크기 변경
- 가상머신의 Host 이름 변경
- 공유 디렉토리 사용하지 않음
- 인터넷에 연결되는 IP 설정
- Host PC의 포트 변경 (port forwarding)
- Ansible Server에서 **add_ssh_auth.sh** 파일 실행 (vagrant 유저)

■ Ansible Node 추가

- *add_ssh_auth.sh* 파일 추가
 - *sshpass* 를 이용해 Ansible Server 에 Ansible Node 의 *ssh_key* 를 등록
- *Ansible_env_ready.yml* 파일 수정
 - */etc/hosts* 서버와 노드 등록
 - */etc/ansible/hosts* 에 앤서블 통해 관리할 노드 등록
 - *yum* 을 통해 앤서블 서버에 *sshpass* 설치

■ Ansible Node 추가

■ add_ssh_auth.sh 파일 추가

```
1  #! /usr/bin/env bash
2
3  # ssh key 생성
4  sshpass -p vagrant ssh -T -o StrictHostKeyChecking=no vagrant@node01
5  sshpass -p vagrant ssh -T -o StrictHostKeyChecking=no vagrant@node02
```

■ Ansible_env_ready.yml 파일 수정

```
6  tasks:
7    - name: Add "/etc/hosts"
8      blockinfile: |
9        dest=/etc/hosts
10       content="
11         172.20.10.10 server
12         172.20.10.11 node01
13         172.20.10.12 node02"
```

```
15  - name: Add "/etc/ansible/hosts"
16    blockinfile: |
17      dest=/etc/ansible/hosts
18      content="
19        [CentOS]
20        node01
21        node02"
22
23  - name: Install sshpass for Authentication
24    yum:
25      name: sshpass
26      state: present
27
```

■ Ansible Node 추가

■ Vagrantfile 파일 수정

```
7 # Ansible-Node01
8 config.vm.define:"ansible-node01" do |cfg|
9   cfg.vm.box = "centos/7"
10  cfg.vm.provider:virtualbox do |vb|
11    vb.name="Ansible-Node01(KTDS)"
12    vb.customize ["modifyvm", :id, "--cpus", 1]
13    vb.customize ["modifyvm", :id, "--memory", 1024]
14  end
15  cfg.vm.host_name="ansible-node01"
16  cfg.vm.synced_folder ".", "/vagrant", disabled: true
17  cfg.vm.network "public_network", ip: "172.20.10.11"
18  cfg.vm.network "forwarded_port", guest: 22, host: 19211, auto_correct: false, id: "ssh"
19 end
20
21 # Ansible-Node02
22 config.vm.define:"ansible-node02" do |cfg|
23   cfg.vm.box = "centos/7"
24   cfg.vm.provider:virtualbox do |vb|
25     vb.name="Ansible-Node02(KTDS)"
26     vb.customize ["modifyvm", :id, "--cpus", 1]
27     vb.customize ["modifyvm", :id, "--memory", 1024]
28   end
29   cfg.vm.host_name="ansible-node02"
30   cfg.vm.synced_folder ".", "/vagrant", disabled: true
31   cfg.vm.network "public_network", ip: "172.20.10.12"
32   cfg.vm.network "forwarded_port", guest: 22, host: 19212, auto_correct: false, id: "ssh"
33 end
```

■ Ansible Node 추가

- `$ vagrant destroy ansible-server`
- `$ vagrant up`
- `$ vagrant ssh ansible-server`
- `[vagrant@ansible-server]$ cat .ssh/known_hosts`
- `[vagrant@ansible-server]$ vi /etc/hosts`
- `[vagrant@ansible-server]$ vi /etc/ansible/hosts`
- `[vagrant@ansible-server]$ ans all -m ping -k`

```
[vagrant@ansible-server ~]$ ans all -m ping -k
SSH password:
node01 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).",
  "unreachable": true
}
node02 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).",
  "unreachable": true
}
```

■ Ansible Node 통신 문제

■ Ansible Server와 Ansible Node 간의 통신 문제

```
[vagrant@ansible-server ~]$ ping node01
PING node01 (172.20.10.11) 56(84) bytes of data.
64 bytes from node01 (172.20.10.11): icmp_seq=1 ttl=64 time=0.425 ms
64 bytes from node01 (172.20.10.11): icmp_seq=2 ttl=64 time=0.425 ms
^C
--- node01 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 0.425 ms
rtt min/avg/max/standard deviation = 0.425/0.425/0.425/0.000
[vagrant@ansible-server ~]$ ansible all -m ping -k
SSH password:
node01 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
node02 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```

SSH Server
sshd

authroized_keys

config

no → yes

■ Ansible Node 통신 문제

■ Ansible Server → Ansible Playbook

```
6 tasks:
7   - name: PasswordAuthentication change from no to yes
8     replace:
9       dest=/etc/ssh/sshd_config
10      regexp='PasswordAuthentication no'
11      replace='PasswordAuthentication yes'
12      backup=yes
13   - name: sshd restart to apply "PasswordAuthentication"
14     service:
15       name: sshd
16       state: restarted
```

■ Ansible Node → Bash Shell

```
3 now=$(date +"%m_%d_%Y")
4 cp /etc/ssh/sshd_config /etc/ssh/sshd_config_${now}.backup
5 sed -i -e 's/PasswordAuthentication no/PasswordAuthentication yes/g' /etc/ssh/sshd_config
6 systemctl restart sshd
```

■ Ansible Node 통신 문제

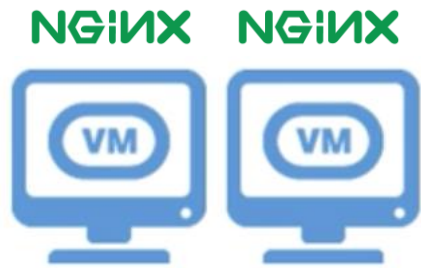
■ \$ vagrant provision

```
✗ down@DOWON-MacBook ~/Desktop/Work/vagrant$ vagrant provision
==> ansible-node01: Running provisioner: shell...
ansible-node01: Running: /var/folders/l6/ywd_lbxx2z3618qtqx7902ph0000gn/T/vagrant-shell20190716-29009-b3gq9y.sh
ansible-node01: date: extra operand '%m_%d_%Y'
ansible-node01: Try 'date --help' for more information.
ansible-node01: sed: -e expression #1, char 53: unknown option to `s'
==> ansible-node02: Running provisioner: shell...
ansible-node02: Running: /var/folders/l6/ywd_lbxx2z3618qtqx7902ph0000gn/T/vagrant-shell20190716-29009-qdjvhy.sh
ansible-node02: date: extra operand '%m_%d_%Y'
ansible-node02: Try 'date --help' for more information.
ansible-node02: sed: -e expression #1, char 53: unknown option to `s'
==> ansible-server: Running provisioner: shell...
ansible-server: Running: /var/folders/l6/ywd_lbxx2z3618qtqx7902ph0000gn/T/vagrant-shell20190716-29009-1hu2xzy.sh
ansible-server: Loaded plugins: fastestmirror
```

■ \$ vagrant ssh ansible-server

```
down@DOWON-MacBook ~/Desktop/Work/vagrant$ vagrant ssh ansible-server
[vagrant@ansible-server ~]$ sudo vi /etc/ssh/sshd_config
[vagrant@ansible-server ~]$
[vagrant@ansible-server ~]$
[vagrant@ansible-server ~]$ clear
[vagrant@ansible-server ~]$ ans all -m ping -k
SSH password:
node01 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```

■ Nginx 배포 테스트



■ 설치



■ 파일 전송



■ 서비스 재시작



■ 삭제

■ Nginx 배포 테스트

- `$ vagrant ssh ansible-server`
- `[vagrant@ansible-server]$ vi nginx_install.yml`

```
2  - name: install nginx on CentOS
3    hosts: CentOS
4    gather_facts: no
5    become: yes
6
7    tasks:
8      - name: install epel-release
9        yum: name=epel-release state=latest
10     - name: install nginx web server
11       yum: name=nginx state=present
12     - name: upload default index.html for web server
13       get_url: url=https://www.nginx.com dest=/usr/share/nginx/html/ mode=0644
14     - name: start nginx web server
15       service: name=nginx state=started
```

- `[vagrant@ansible-server]$ vi nginx_remove.yml`

```
2  - name: remove nginx on CentOS
3    hosts: CentOS
4    gather_facts: no
5    become: yes
6
7    tasks:
8      - name: remove epel-release
9        yum: name=epel-release state=absent
10     - name: remove nginx web server
11       yum: name=nginx state=absent
```

■ Nginx 배포 테스트

- `[vagrant@ansible-server]$ anp nginx_install.yml -k`

```
PLAY [install nginx on CentOS] *****

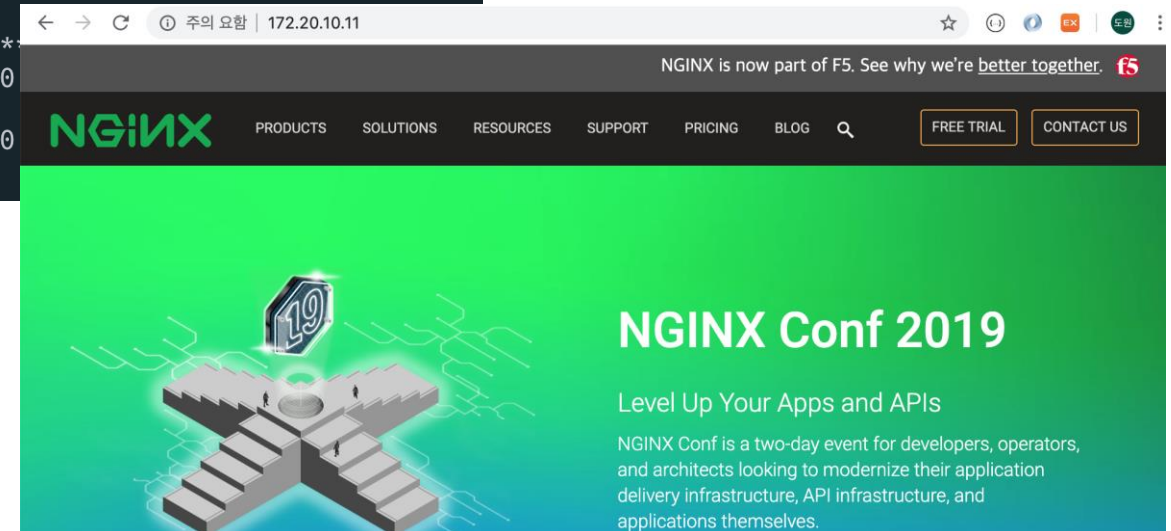
TASK [install epel-release] *****
changed: [node01]
changed: [node02]

TASK [install nginx web server] *****
changed: [node02]
changed: [node01]

TASK [upload default index.html for web server] *****
changed: [node01]
changed: [node02]

TASK [start nginx web server] *****
changed: [node02]
changed: [node01]

PLAY RECAP *****
node01                : ok=4    changed=4    unreachable=0    failed=0
ignored=0
node02                : ok=4    changed=4    unreachable=0    failed=0
ignored=0
```



■ Nginx 배포 테스트

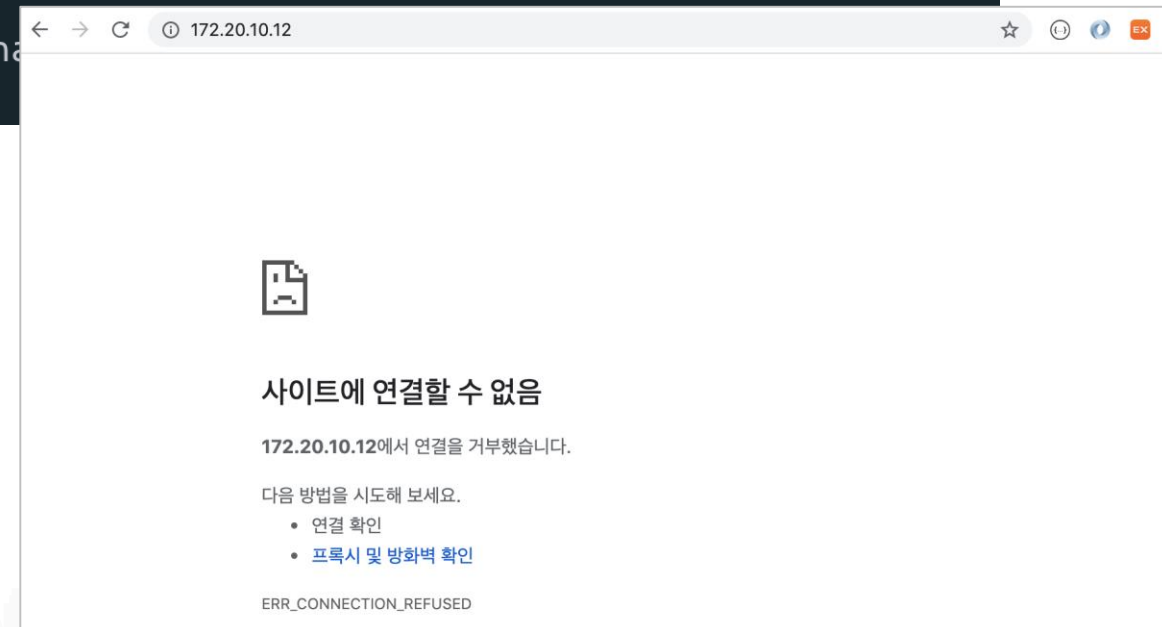
- `[vagrant@ansible-server]$ anp nginx_remove.yml -k`

```
PLAY [remove nginx on CentOS] *****

TASK [remove epel-release] *****
changed: [node01]
changed: [node02]

TASK [remove nginx web server] *****
changed: [node01]
changed: [node02]

PLAY RECAP *****
node01                : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
node02                : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```



■ 시간대 변경

- `[vagrant@ansible-server]$ date`

```
[vagrant@ansible-server ~]$ date
2019. 07. 15. (월) 23:27:10 UTC
[vagrant@ansible-server ~]$
```

- `[vagrant@ansible-server]$ vi timezone.yml`

```
2  - name: setup timezone
3    hosts: CentOS
4    gather_facts: no
5    become: yes
6
7    tasks:
8      - name: set timezone to Asia/Seoul
9        timezone: name=Asia/Seoul
```

```
[vagrant@ansible-node01 ~]$ date
2019. 07. 16. (화) 08:32:10 KST
[vagrant@ansible-node01 ~]$
```



- `[vagrant@ansible-server]$ ansible-playbook -k timezone.yml`

```
PLAY [setup timezone] *****
TASK [set timezone to Asia/Seoul] *****
changed: [node01]
changed: [node02]

PLAY RECAP *****
node01          : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
node02          : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```