Chapter 3: Ansible Architecture and Design

1. Ansible Configuration

/etc/ansible/ansible.cfg

```
ansible@ubuntu-c:~$ ansible --version
ansible [core 2.12.3]
  config file = None
  configured module search path = ['/home/ansible/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.8/dist-packages/ansible
  ansible collection location = /home/ansible/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.8.10 (default, Nov 26 2021, 20:14:08) [GCC 9.3.0]
  jinja version = 3.0.3
  libyaml = True
ansible@ubuntu-c:~$ su -
Password:
root@ubuntu-c:~# mkdir /etc/ansible
root@ubuntu-c:~# touch /etc/ansible/ansible.cfg
root@ubuntu-c:~# exit
logout
ansible@ubuntu-c:~$ ansible --version
ansible [core 2.12.3]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ansible/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.8/dist-packages/ansible
  ansible collection location = /home/ansible/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.8.10 (default, Nov 26 2021, 20:14:08) [GCC 9.3.0]
  jinja version = 3.0.3
  libyaml = True
```

~/.ansible.cfg

```
ansible@ubuntu-c:~$ cd ~
ansible@ubuntu-c:~$ pwd
/home/ansible
ansible@ubuntu-c:~$ touch .ansible.cfg
ansible@ubuntu-c:~$ ansible --version
ansible [core 2.12.3]
  config file = /home/ansible/.ansible.cfg
  configured module search path = ['/home/ansible/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.8/dist-packages/ansible
  ansible collection location = /home/ansible/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.8.10 (default, Nov 26 2021, 20:14:08) [GCC 9.3.0]
  jinja version = 3.0.3
  libyaml = True
ansible@ubuntu-c:~$ cd /tmp/
ansible@ubuntu-c:/tmp$ ansible --version
ansible [core 2.12.3]
```

```
  config file = /home/ansible/.ansible.cfg
  configured module search path = ['/home/ansible/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.8/dist-packages/ansible
  ansible collection location = /home/ansible/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.8.10 (default, Nov 26 2021, 20:14:08) [GCC 9.3.0]
  jinja version = 3.0.3
  libyaml = True
ansible@ubuntu-c:/tmp$
```

./ansible.cfg(current directory)

```
ansible@ubuntu-c:/tmp$ cd ~
ansible@ubuntu-c:~$ ls
diveintoansible
ansible@ubuntu-c:~$ mkdir testdir
ansible@ubuntu-c:~$ cd testdir/
ansible@ubuntu-c:~/testdir$ ansible --version
ansible [core 2.12.3]
  config file = /home/ansible/.ansible.cfg
  configured module search path = ['/home/ansible/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.8/dist-packages/ansible
  ansible collection location = /home/ansible/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.8.10 (default, Nov 26 2021, 20:14:08) [GCC 9.3.0]
  jinja version = 3.0.3
  libyaml = True
ansible@ubuntu-c:~/testdir$ touch ansible.cfg
ansible@ubuntu-c:~/testdir$ ansible --version
ansible [core 2.12.3]
  config file = /home/ansible/testdir/ansible.cfg
  configured module search path = ['/home/ansible/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.8/dist-packages/ansible
  ansible collection location = /home/ansible/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.8.10 (default, Nov 26 2021, 20:14:08) [GCC 9.3.0]
  jinja version = 3.0.3
  libyaml = True
```

ANSIBLE_CONFIG(Environment Variable, with a filename target)

```
ansible@ubuntu-c:~$ touch this_is_my_example_ansible.cfg
ansible@ubuntu-c:~$ export ANSIBLE_CONFIG=/home/ansible/this_is_my_example_ansible.cfg
ansible@ubuntu-c:~$ ansible --version
ansible [core 2.12.3]
  config file = /home/ansible/this_is_my_example_ansible.cfg
  configured module search path = ['/home/ansible/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.8/dist-packages/ansible
  ansible collection location = /home/ansible/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.8.10 (default, Nov 26 2021, 20:14:08) [GCC 9.3.0]
  jinja version = 3.0.3
  libyaml = True
ansible@ubuntu-c:~$
```

정리

```
ansible@ubuntu-c:~$ unset ANSIBLE_CONFIG
ansible@ubuntu-c:~$ sudo rm /etc/ansible/ansible.cfg
[sudo] password for ansible:
ansible@ubuntu-c:~$ sudo rmdir /etc/ansible/
ansible@ubuntu-c:~$ rm ~/.ansible.cfg
```

2. Ansible Inventories

2-1. Ansible Inventories

```
$ pwd
/home/ansible/diveintoansible/Ansible Architecture and Design/Inventories/01
$ ping centos1
PING centos1 (172.19.0.6) 56(84) bytes of data.
64 bytes from centos1.diveinto.io (172.19.0.6): icmp_seq=1 ttl=64 time=0.211 ms
64 bytes from centos1.diveinto.io (172.19.0.6): icmp_seq=2 ttl=64 time=0.082 ms
64 bytes from centos1.diveinto.io (172.19.0.6): icmp_seq=3 ttl=64 time=0.071 ms
^C
--- centos1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2011ms
rtt min/avg/max/mdev = 0.071/0.121/0.211/0.063 ms
$ ls -l
total 8
-rw-r--r-- 1 ansible ansible 29 Jun 24 09:08 ansible.cfg
-rw-r--r-- 1 ansible ansible 14 Jun 24 09:08 hosts
$ cat ansible.cfg
[defaults]
inventory = hosts
$ cat hosts
[all]
centos1
$ rm -rf /home/ansible/.ssh/known_hosts
```

fingerprint 테스트

```
$ rm -rf /home/ansible/.ssh/known_hosts
$ ansible all -m ping
The authenticity of host 'centos1 (172.19.0.6)' can't be established.
ECDSA key fingerprint is SHA256:GXYBZcsdVBb7fJRuEtJPX+6UlaRLCRZOqokUpURtTf4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? ^C [ERROR]: User interrupted execution
```

`ANSIBLE_HOST_KEY_CHECKING=False`

```
$ ANSIBLE_HOST_KEY_CHECKING=False ansible all -m ping
centos1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
```

ansible.cfg 에 host_key_checking = False 추가

```
$ cd ../02
$ pwd
/home/ansible/diveintoansible/Ansible Architecture and Design/Inventories/02
$ cat ansible.cfg
[defaults]
inventory = hosts
host_key_checking = False
$ rm -rf /home/ansible/.ssh/known_hosts
$ ansible all -m ping
centos1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
```

```
$ cd ../03
$ pwd
/home/ansible/diveintoansible/Ansible Architecture and Design/Inventories/03
$ cat hosts
[centos]
centos1
centos2
centos3

[ubuntu]
ubuntu1
ubuntu2
ubuntu3
$ ansible all  -m ping
ubuntu1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
centos1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
centos2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
centos3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}

# group 으로
$ ansible centos -m ping
$ ansible ubuntu -m ping
# '*' 패턴
$ ansible '*' -m ping
# 1라인으로 출력
$ ansible all -m ping -o
ubuntu1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
ubuntu2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
```

```
centos3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
centos1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
centos2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
ubuntu3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
```

그룹명으로 inventory host 목록 출력하기

```
$ ansible centos --list-hosts
  hosts (3):
    centos1
    centos2
    centos3
$ ansible ubuntu --list-hosts
hosts (3):
    ubuntu1
    ubuntu2
    ubuntu3
$ ansible all --list-hosts
hosts (6):
    centos1
    centos2
    centos3
    ubuntu1
    ubuntu2
    ubuntu3
```

host 이름 지정

```
$ ansible centos1 --list-hosts
hosts (1):
    centos1
$ ansible centos1 -m ping -o
centos1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
```

Pattern 매칭(Regular Expression)

```
$ ansible ~.*3 --list-hosts
  hosts (2):
    centos3
    ubuntu3
```

2-2. Provide Ansible connectivity to our centos hosts via root

```
$ cd ../04
$ pwd
/home/ansible/diveintoansible/Ansible Architecture and Design/Inventories/04
$ cat ansible.cfg
[defaults]
inventory = hosts
host_key_checking = False

$ cat hosts
[centos]
centos1 ansible_user=root
centos2 ansible_user=root
centos3 ansible_user=root

[ubuntu]
ubuntu1
ubuntu2
ubuntu3

$ id
uid=1000(ansible) gid=1000(ansible) groups=1000(ansible),27(sudo)

$ ansible all -m command -a 'id' -o
ubuntu1 | CHANGED | rc=0 | (stdout) uid=1000(ansible) gid=1000(ansible) groups=1000(ansible),27(sudo)
ubuntu2 | CHANGED | rc=0 | (stdout) uid=1000(ansible) gid=1000(ansible) groups=1000(ansible),27(sudo)
centos2 | CHANGED | rc=0 | (stdout) uid=0(root) gid=0(root) groups=0(root)
centos3 | CHANGED | rc=0 | (stdout) uid=0(root) gid=0(root) groups=0(root)
centos1 | CHANGED | rc=0 | (stdout) uid=0(root) gid=0(root) groups=0(root)
ubuntu3 | CHANGED | rc=0 | (stdout) uid=1000(ansible) gid=1000(ansible) groups=1000(ansible),27(sudo)
```

2-3. Provide Ansible connectivity to our ubuntu hosts via sudo

```
$ cd ../05
$ pwd
/home/ansible/diveintoansible/Ansible Architecture and Design/Inventories/05

$ cat ansible.cfg
[defaults]
inventory = hosts
host_key_checking = False

$ cat hosts
[centos]
centos1 ansible_user=root
centos2 ansible_user=root
centos3 ansible_user=root

[ubuntu]
ubuntu1 ansible_become=true ansible_become_pass=password
ubuntu2 ansible_become=true ansible_become_pass=password
ubuntu3 ansible_become=true ansible_become_pass=password

$ ansible all -m ping -o
ubuntu2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
centos3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
ubuntu1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
ubuntu3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}

$ ansible all -a 'id' -o
centos2 | CHANGED | rc=0 | (stdout) uid=0(root) gid=0(root) groups=0(root)
ubuntu1 | CHANGED | rc=0 | (stdout) uid=0(root) gid=0(root) groups=0(root)
ubuntu2 | CHANGED | rc=0 | (stdout) uid=0(root) gid=0(root) groups=0(root)
centos1 | CHANGED | rc=0 | (stdout) uid=0(root) gid=0(root) groups=0(root)
centos3 | CHANGED | rc=0 | (stdout) uid=0(root) gid=0(root) groups=0(root)
ubuntu3 | CHANGED | rc=0 | (stdout) uid=0(root) gid=0(root) groups=0(root)
```

2-4. Inventory host variables(hostvars)

`ansible ssh connection default port ⇒ 22`

```
centos1:
    hostname: centos1
    container_name: centos1
    #image: spurin/diveintoansible:centos
    image: spurin/diveintoansible:centos-sshd-2222
    ports:
     #- ${CENTOS1_PORT_SSHD}:22
     - ${CENTOS1_PORT_SSHD}:2222
     - ${CENTOS1_PORT_TTYD}:7681
    privileged: true
    volumes:
     - ${CONFIG}:/config
     - ${ANSIBLE_HOME}/shared:/shared
     - ${ANSIBLE_HOME}/centos1/ansible:/home/ansible
     - ${ANSIBLE_HOME}/centos1/root:/root
    networks:
     - diveinto.io
```

```
$ docker-compose rm
$ docker-compose up
```

ansible-c 호스트 접속

```
$ cd /home/ansible/diveintoansible/Ansible Architecture and Design/Inventories/05
$ ansible all -m ping -o
centos1 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host centos1 port 22: Connection refused
ubuntu3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
ubuntu2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
ubuntu1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
centos3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
```

변경된 포트를 사용하도록 hosts 파일 변경(ansible_port=2222)

```
$ cd ../06
$ cat hosts
[centos]
centos1 ansible_user=root ansible_port=2222
centos2 ansible_user=root
centos3 ansible_user=root

[ubuntu]
ubuntu1 ansible_become=true ansible_become_pass=password
ubuntu2 ansible_become=true ansible_become_pass=password
ubuntu3 ansible_become=true ansible_become_pass=password

$ ansible all -m ping -o
ubuntu2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
ubuntu1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
centos1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
ubuntu3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
```

`호스트:<port>` 형태 변경도 가능

```
$ cd ../06
$ cat hosts
[centos]
centos1:2222 ansible_user=root
centos2 ansible_user=root
centos3 ansible_user=root

[ubuntu]
ubuntu1 ansible_become=true ansible_become_pass=password
ubuntu2 ansible_become=true ansible_become_pass=password
ubuntu3 ansible_become=true ansible_become_pass=password

$ ansible all -m ping -o
ubuntu2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
ubuntu1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
centos2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
ubuntu3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
```

ansible local connection

```
$ cd ../08
$ cat hosts
[control]
ubuntu-c ansible_connection=local

[centos]
centos1 ansible_user=root ansible_port=2222
centos2 ansible_user=root
centos3 ansible_user=root

[ubuntu]
ubuntu1 ansible_become=true ansible_become_pass=password
ubuntu2 ansible_become=true ansible_become_pass=password
ubuntu3 ansible_become=true ansible_become_pass=password

$ ansible all -m ping -o
centos1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
ubuntu-c | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
centos3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
ubuntu1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
ubuntu2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
ubuntu3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
```

2-5. Simplification of Inventory with ranges

```
$ cd ../09
$ cat hosts
[control]
ubuntu-c ansible_connection=local

[centos]
```

```
centos1 ansible_user=root ansible_port=2222
centos[2:3] ansible_user=root

[ubuntu]
ubuntu[1:3] ansible_become=true ansible_become_pass=password

$ ansible all --list-hosts
hosts (7):
    ubuntu-c
    centos1
    centos2
    centos3
    ubuntu1
    ubuntu2
    ubuntu3
```

centos 그룹에서 ansible_user=root 가 반복되어 있다.

2-6. Inventory group variables(groupvars)

```
$ cd ../10
$ cat hosts
[control]
ubuntu-c ansible_connection=local

[centos]
centos1 ansible_port=2222
centos[2:3]

[centos:vars]
ansible_user=root

[ubuntu]
ubuntu[1:3]

[ubuntu:vars]
ansible_become=true
ansible_become_pass=password

$ ansible all -m ping -o
ubuntu-c | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
centos2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
ubuntu1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
ubuntu2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
ubuntu3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
```

2-7. Inventory children groups

```
$ cd ../11/
$ cat hosts
[control]
ubuntu-c ansible_connection=local

[centos]
centos1 ansible_port=2222
centos[2:3]

[centos:vars]
ansible_user=root

[ubuntu]
ubuntu[1:3]

[ubuntu:vars]
ansible_become=true
ansible_become_pass=password

[linux:children]
centos
ubuntu

$ ansible linux -m ping -o
centos2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
centos3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
centos1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
ubuntu1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
```

```
ubuntu2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
ubuntu3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
```

```
$ cd ../12
$ cat hosts
[control]
ubuntu-c ansible_connection=local

[centos]
centos1 ansible_port=2222
centos[2:3]

[centos:vars]
ansible_user=root

[ubuntu]
ubuntu[1:3]

[ubuntu:vars]
ansible_become=true
ansible_become_pass=password

[linux:children]
centos
ubuntu

[all:vars]
ansible_port=1234

$ ansible linux -m ping -o
centos2 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host centos2 port 1234: Connection refused
centos3 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host centos3 port 1234: Connection refused
ubuntu1 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host ubuntu1 port 1234: Connection refused
ubuntu2 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host ubuntu2 port 1234: Connection refused
ubuntu3 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host ubuntu3 port 1234: Connection refused
centos1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon

$ ansible all -m ping -o
centos2 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host centos2 port 1234: Connection refused
centos3 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host centos3 port 1234: Connection refused
ubuntu1 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host ubuntu1 port 1234: Connection refused
ubuntu2 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host ubuntu2 port 1234: Connection refused
ubuntu3 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host ubuntu3 port 1234: Connection refused
ubuntu-c | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
```

all 그룹의 vars 의 값이 전역이므로 cenots1, localhost를 제외한 나머지 호스트에서 ping 테스트가 실패함을 알 수 있다.

```
$ cd ../13
$ cat hosts
[control]
ubuntu-c ansible_connection=local

[centos]
centos1 ansible_port=2222
centos[2:3]

[centos:vars]
ansible_user=root

[ubuntu]
ubuntu[1:3]

[ubuntu:vars]
ansible_become=true
ansible_become_pass=password

[linux:children]
centos
ubuntu

[linux:vars]
ansible_port=1234

$ ansible linux -m ping -o
centos2 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host centos2 port 1234: Connection refused
ubuntu1 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host ubuntu1 port 1234: Connection refused
centos3 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host centos3 port 1234: Connection refused
ubuntu2 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host ubuntu2 port 1234: Connection refused
```

```
ubuntu3 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host ubuntu3 port 1234: Connection refused
centos1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
```

host 파일을 yaml 로 구성하기

```
$ cd ../14

$ cat ansible.cfg
[defaults]
inventory = hosts.yaml
host_key_checking = False

$ cat hosts.yaml
---
control:
  hosts:
    ubuntu-c:
      ansible_connection: local
centos:
  hosts:
    centos1:
      ansible_port: 2222
    centos2:
    centos3:
  vars:
    ansible_user: root
ubuntu:
  hosts:
    ubuntu1:
    ubuntu2:
    ubuntu3:
  vars:
    ansible_become: true
    ansible_become_pass: password
linux:
  children:
    centos:
    ubuntu:
...
```

- YAML 파일의 시작은 ---(3개의 대쉬)
- YAML 파일의 끝은 …(3개의 마침표)

```
$ ansible all -m ping -o
ubuntu-c | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
ubuntu1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
centos2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
ubuntu2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
ubuntu3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
```

YAML 포맷을 JSON 포맷으로 바꾸기

```
$ cd ../15
$ python3 -c 'import sys, yaml, json; json.dump(yaml.load(sys.stdin, Loader=yaml.FullLoader), sys.stdout, indent=4)' < hosts.yaml >
hosts.json

$ cat hosts.json
{
    "control": {
        "hosts": {
            "ubuntu-c": {
                "ansible_connection": "local"
            }
        }
    },
    "centos": {
        "hosts": {
            "centos1": {
                "ansible_port": 2222
            },
            "centos2": null,
            "centos3": null
        },
        "vars": {
            "ansible_user": "root"
```

```
        }
    },
    "ubuntu": {
        "hosts": {
            "ubuntu1": null,
            "ubuntu2": null,
            "ubuntu3": null
        },
        "vars": {
            "ansible_become": true,
            "ansible_become_pass": "password"
        }
    },
    "linux": {
        "children": {
            "centos": null,
            "ubuntu": null
        }
    }
}

# ansible.cfg에서 인벤토리파일을 json 으로 설정
$ cat ansible.cfg
[defaults]
inventory = hosts.json
host_key_checking = False

$ ansible all -m ping -o
ubuntu-c | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping":
 "pong"}
centos1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping":
 "pong"}
ubuntu1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping":
 "pong"}
ubuntu2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
ubuntu3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
```

커맨드라인에서 inventory 파일 지정하기(디폴트는 hosts).

```
$ cd ../16

$ cat ansible.cfg
[defaults]
inventory = hosts
host_key_checking = False

$ ansible all -i hosts.yaml --list-hosts
  hosts (7):
    ubuntu-c
    centos1
    centos2
    centos3
    ubuntu1
    ubuntu2
    ubuntu3

$ ansible all -i hosts.json --list-hosts
  hosts (7):
    ubuntu-c
    centos3
    centos2
    centos1
    ubuntu1
    ubuntu2
    ubuntu3

$ ansible all -i hosts --list-hosts
  hosts (7):
    ubuntu-c
    centos1
    centos2
    centos3
    ubuntu1
    ubuntu2
    ubuntu3
```

커맨드 라인에서 변수 오버라이드 하기

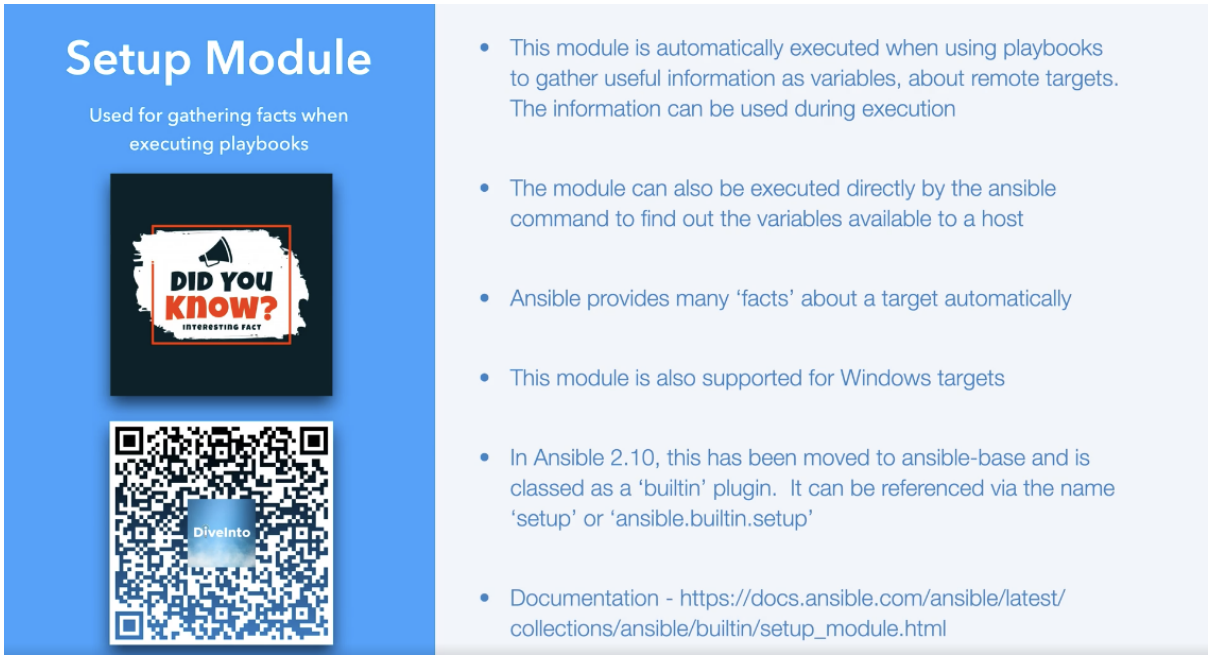
```
$ ansible --help
usage: ansible [-h] [--version] [-v] [-b] [--become-method BECOME_METHOD] [--become-user BECOME_USER] [-K | --become-password-file BEC
               [--list-hosts] [-l SUBSET] [-P POLL_INTERVAL] [-B SECONDS] [-o] [-t TREE] [--private-key PRIVATE_KEY_FILE] [-u REMOTE_U
               [--ssh-common-args SSH_COMMON_ARGS] [--sftp-extra-args SFTP_EXTRA_ARGS] [--scp-extra-args SCP_EXTRA_ARGS] [--ssh-extra-
               [-k | --connection-password-file CONNECTION_PASSWORD_FILE] [-C] [--syntax-check] [-D] [-e EXTRA_VARS] [--vault-id VAULT
               [--ask-vault-password | --vault-password-file VAULT_PASSWORD_FILES] [-f FORKS] [-M MODULE_PATH] [--playbook-dir BASEDIR
               [-a MODULE_ARGS] [-m MODULE_NAME]
               pattern

Define and run a single task 'playbook' against a set of hosts

positional arguments:
  pattern               host pattern

optional arguments:
  --ask-vault-password, --ask-vault-pass
                        ask for vault password
  --become-password-file BECOME_PASSWORD_FILE, --become-pass-file BECOME_PASSWORD_FILE
                        Become password file
  --connection-password-file CONNECTION_PASSWORD_FILE, --conn-pass-file CONNECTION_PASSWORD_FILE
                        Connection password file
  --list-hosts          outputs a list of matching hosts; does not execute anything else
  --playbook-dir BASEDIR
                        Since this tool does not use playbooks, use this as a substitute playbook directory.This sets the relative pat
                        group_vars/ etc.
  --syntax-check        perform a syntax check on the playbook, but do not execute it
  --task-timeout TASK_TIMEOUT
                        set task timeout limit in seconds, must be positive integer.
  --vault-id VAULT_IDS  the vault identity to use
  --vault-password-file VAULT_PASSWORD_FILES, --vault-pass-file VAULT_PASSWORD_FILES
                        vault password file
  --version             show program's version number, config file location, configured module search path, module location, executabl
  -B SECONDS, --background SECONDS
                        run asynchronously, failing after X seconds (default=N/A)
  -C, --check           don't make any changes; instead, try to predict some of the changes that may occur
  -D, --diff            when changing (small) files and templates, show the differences in those files; works great with --check
  -K, --ask-become-pass
                        ask for privilege escalation password
  -M MODULE_PATH, --module-path MODULE_PATH
                        prepend colon-separated path(s) to module library (default=~/.ansible/plugins/modules:/usr/share/ansible/plugi
  -P POLL_INTERVAL, --poll POLL_INTERVAL
                        set the poll interval if using -B (default=15)
  -a MODULE_ARGS, --args MODULE_ARGS
                        The action's options in space separated k=v format: -a 'opt1=val1 opt2=val2'
  -e EXTRA_VARS, --extra-vars EXTRA_VARS
                        set additional variables as key=value or YAML/JSON, if filename prepend with @
  -f FORKS, --forks FORKS
                        specify number of parallel processes to use (default=5)
  -h, --help            show this help message and exit
  -i INVENTORY, --inventory INVENTORY, --inventory-file INVENTORY
                        specify inventory host path or comma separated host list. --inventory-file is deprecated
  -k, --ask-pass        ask for connection password
  -l SUBSET, --limit SUBSET
                        further limit selected hosts to an additional pattern
  -m MODULE_NAME, --module-name MODULE_NAME
                        Name of the action to execute (default=command)
  -o, --one-line        condense output
  -t TREE, --tree TREE  log output to this directory
  -v, --verbose         verbose mode (-vvv for more, -vvvv to enable connection debugging)

Privilege Escalation Options:
  control how and which user you become as on target hosts

  --become-method BECOME_METHOD
                        privilege escalation method to use (default=sudo), use `ansible-doc -t become -l` to list valid choices.
  --become-user BECOME_USER
                        run operations as this user (default=root)
  -b, --become          run operations with become (does not imply password prompting)

Connection Options:
  control as whom and how to connect to hosts

  --private-key PRIVATE_KEY_FILE, --key-file PRIVATE_KEY_FILE
                        use this file to authenticate the connection
  --scp-extra-args SCP_EXTRA_ARGS
                        specify extra arguments to pass to scp only (e.g. -l)
  --sftp-extra-args SFTP_EXTRA_ARGS
                        specify extra arguments to pass to sftp only (e.g. -f, -l)
  --ssh-common-args SSH_COMMON_ARGS
                        specify common arguments to pass to sftp/scp/ssh (e.g. ProxyCommand)
  --ssh-extra-args SSH_EXTRA_ARGS
                        specify extra arguments to pass to ssh only (e.g. -R)
  -T TIMEOUT, --timeout TIMEOUT
                        override the connection timeout in seconds (default=10)
  -c CONNECTION, --connection CONNECTION
```

```
                        connection type to use (default=smart)
  -u REMOTE_USER, --user REMOTE_USER
                        connect as this user (default=None)

Some actions do not make sense in Ad-Hoc (include, meta, etc)

$ ansible linux -m ping -e 'ansible_port=22' -o
centos1 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host centos1 port 22: Connection refused
ubuntu1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
ubuntu3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos3 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
ubuntu2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": false,"ping": "pong"}
centos2 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon

$ ansible linux -m ping -e 'ansible_port=2222' -o
centos2 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host centos2 port 2222: Connection refused
ubuntu1 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host ubuntu1 port 2222: Connection refused
centos3 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host centos3 port 2222: Connection refused
ubuntu2 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host ubuntu2 port 2222: Connection refused
ubuntu3 | UNREACHABLE!: Failed to connect to the host via ssh: ssh: connect to host ubuntu3 port 2222: Connection refused
centos1 | SUCCESS => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": false,"ping": "pon
```

3. Ansible Modules

3-1. Ansible Modules

3-2. The setup module



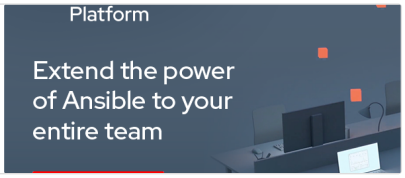

```
$ pwd
/home/ansible/diveintoansible/Ansible Architecture and Design/Modules

$ ansible centos1 -m setup
```

3-3. The file module

ansible.builtin.file module - Manage files and file properties - Ansible Documentation

Note This module is part of ansible-core and included in all Ansible installations. In most cases, you can use the short module name even without specifying the collections: keyword. However, we recommend you use the FQCN for easy linking to the module documentation and to avoid conflicting with other collections that

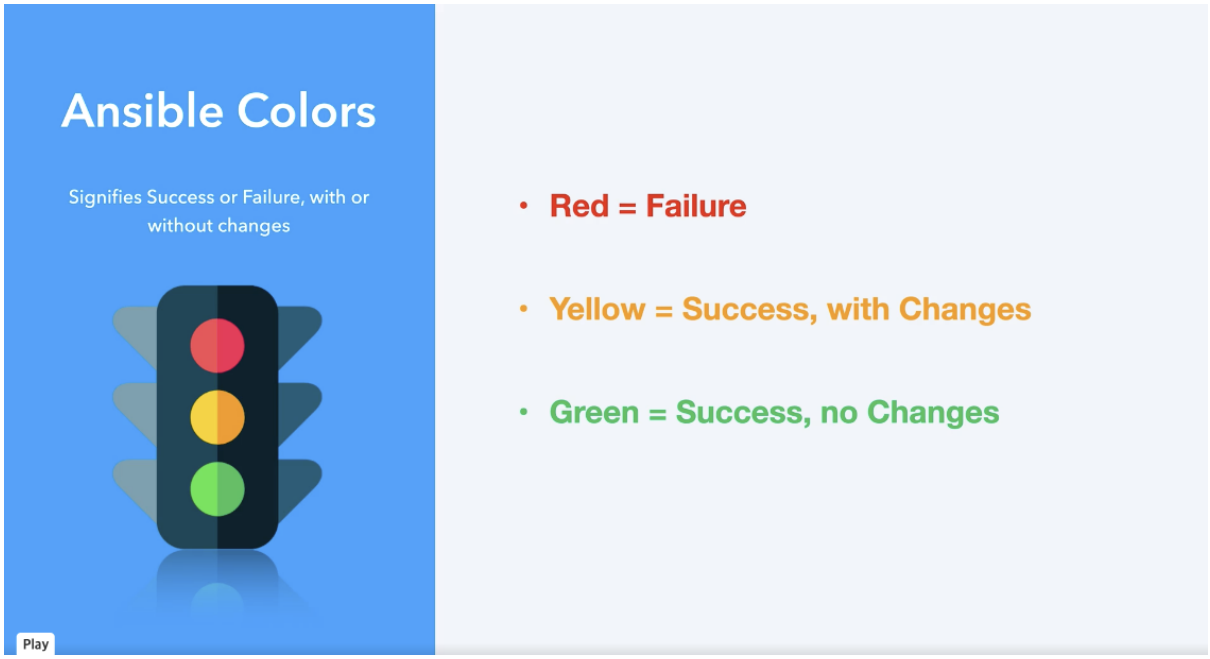
https://docs.ansible.com/ansible/latest/collections/ansible/builtin/file_module.html




```
$ ansible all -m file -a 'path=/tmp/test state=touch'
centos1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": true,
    "dest": "/tmp/test",
    "gid": 0,
    "group": "root",
    "mode": "0644",
    "owner": "root",
    "size": 0,
    "state": "file",
    "uid": 0
}
centos2 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": true,
    "dest": "/tmp/test",
    "gid": 0,
    "group": "root",
    "mode": "0644",
    "owner": "root",
    "size": 0,
    "state": "file",
    "uid": 0
}
ubuntu-c | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "dest": "/tmp/test",
    "gid": 1000,
    "group": "ansible",
    "mode": "0664",
    "owner": "ansible",
    "size": 0,
    "state": "file",
    "uid": 1000
}
centos3 | CHANGED => {
```

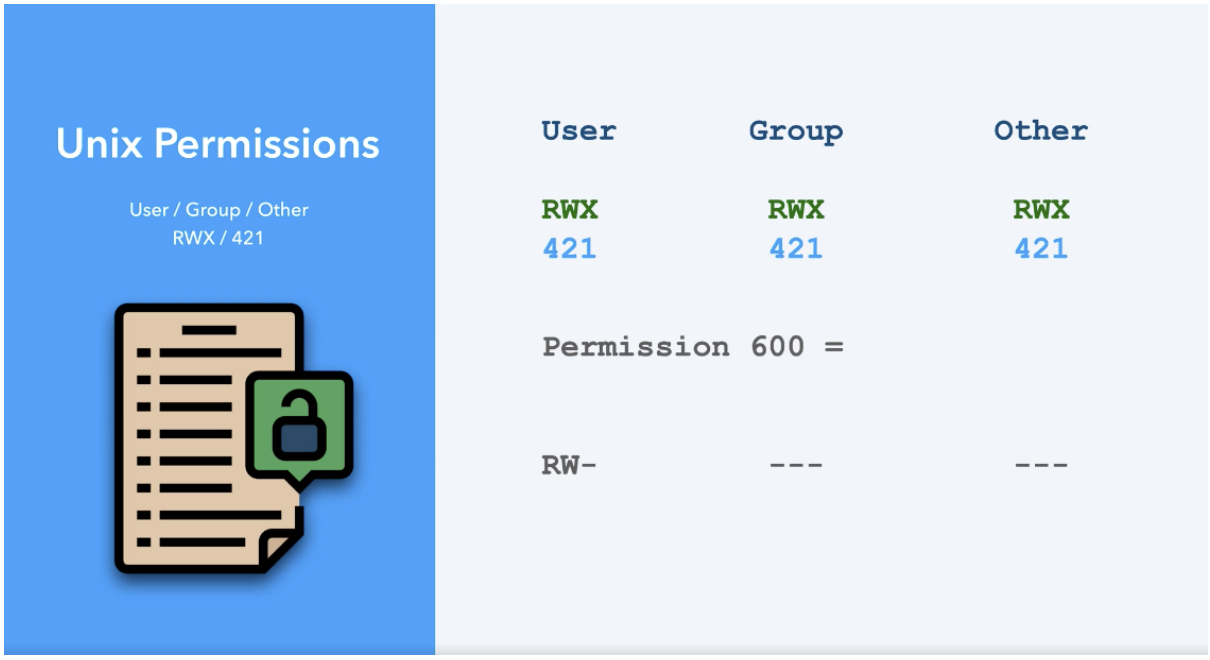
```
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": true,
    "dest": "/tmp/test",
    "gid": 0,
    "group": "root",
    "mode": "0644",
    "owner": "root",
    "size": 0,
    "state": "file",
    "uid": 0
}
ubuntu1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "dest": "/tmp/test",
    "gid": 0,
    "group": "root",
    "mode": "0644",
    "owner": "root",
    "size": 0,
    "state": "file",
    "uid": 0
}
ubuntu2 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "dest": "/tmp/test",
    "gid": 0,
    "group": "root",
    "mode": "0644",
    "owner": "root",
    "size": 0,
    "state": "file",
    "uid": 0
}
ubuntu3 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "dest": "/tmp/test",
    "gid": 0,
    "group": "root",
    "mode": "0644",
    "owner": "root",
    "size": 0,
    "state": "file",
    "uid": 0
}

$ ls -l /tmp/test
-rw-rw-r-- 1 ansible ansible 0 Jun 25 03:34 /tmp/test
$ ssh centos2 ls -althr /tmp/test
-rw-r--r-- 1 root root 0 Jun 25 03:34 /tmp/test
```

3-4. Color notation use during Ansible execution




```
$ ansible all -m file -a 'path=/tmp/test state=touch'
ubuntu-c | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "dest": "/tmp/test",
    "gid": 1000,
    "group": "ansible",
    "mode": "0664",
    "owner": "ansible",
    "size": 0,
    "state": "file",
    "uid": 1000
}
centos2 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": true,
    "dest": "/tmp/test",
    "gid": 0,
    "group": "root",
    "mode": "0644",
    "owner": "root",
    "size": 0,
    "state": "file",
    "uid": 0
}
centos1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": true,
    "dest": "/tmp/test",
    "gid": 0,
    "group": "root",
    "mode": "0644",
    "owner": "root",
    "size": 0,
    "state": "file",
    "uid": 0
}
centos3 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": true,
    "dest": "/tmp/test",
    "gid": 0,
    "group": "root",
    "mode": "0644",
    "owner": "root",
```

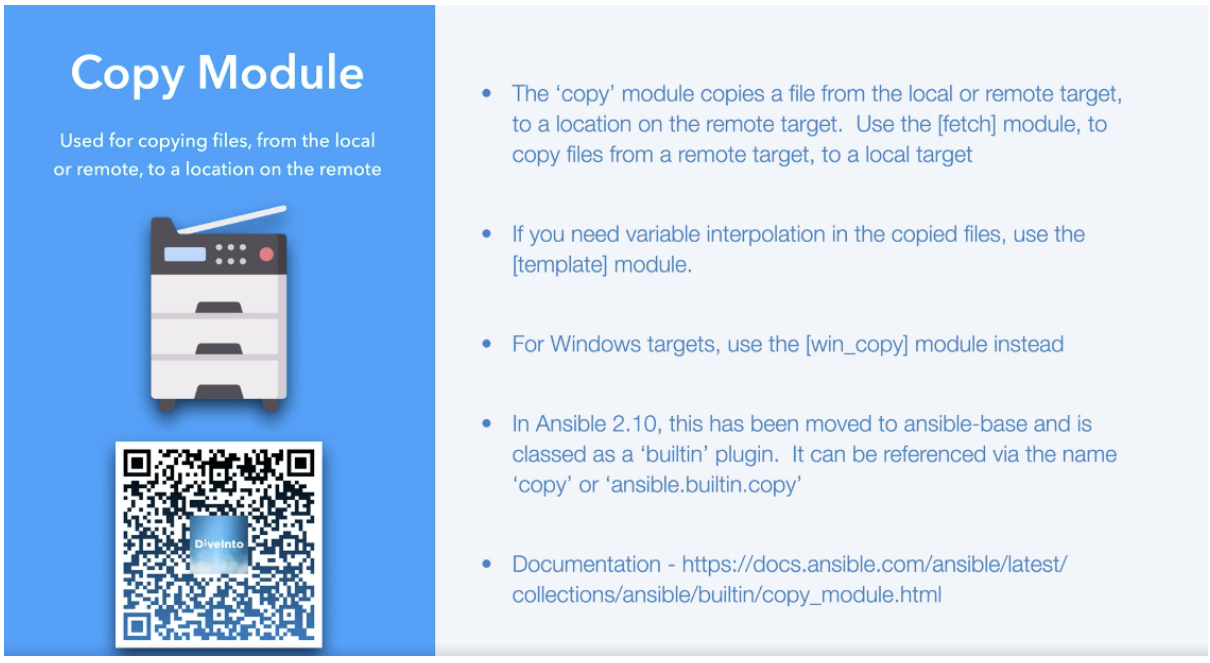
```
    "size": 0,
    "state": "file",
    "uid": 0
}
ubuntu1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "dest": "/tmp/test",
    "gid": 0,
    "group": "root",
    "mode": "0644",
    "owner": "root",
    "size": 0,
    "state": "file",
    "uid": 0
}
ubuntu2 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "dest": "/tmp/test",
    "gid": 0,
    "group": "root",
    "mode": "0644",
    "owner": "root",
    "size": 0,
    "state": "file",
    "uid": 0
}
ubuntu3 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "dest": "/tmp/test",
    "gid": 0,
    "group": "root",
    "mode": "0644",
    "owner": "root",
    "size": 0,
    "state": "file",
    "uid": 0
}
```


```
$ ansible all -m file -a 'path=/tmp/test state=file mode=600'
```

3-5. Idempotence

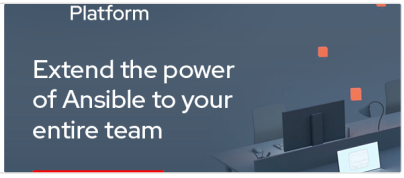
3-6. The copy module



ansible.builtin.copy module - Copy files to remote locations - Ansible Documentation

Note This module is part of ansible-core and included in all Ansible installations. In most cases, you can use the short module name even without specifying the collections: keyword. However, we recommend you use the FQCN for easy linking to the module documentation and to avoid conflicting with other collections that

https://docs.ansible.com/ansible/latest/collections/ansible/builtin/copy_module.html



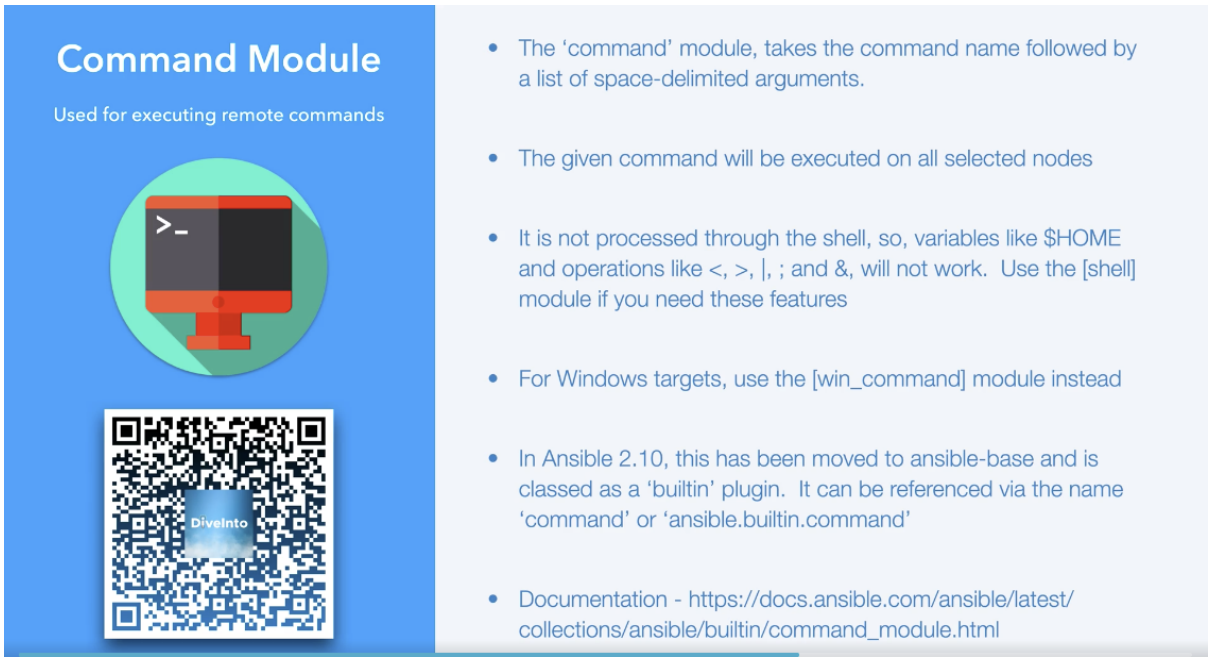

```
$ touch /tmp/x
$ ansible all -m copy -a 'src=/tmp/x dest=/tmp/x'
```

Copy on the remote target using a remote source, to a remote destination(supports recursive copying)

`remote_src=yes`

```
$ touch /tmp/x
$ ansible all -m copy -a 'remote_src=yes src=/tmp/x dest=/tmp/y'
```

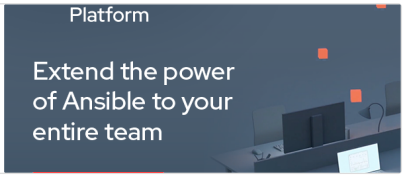
3-7. The command module



ansible.builtin.command module - Execute commands on targets - Ansible Documentation

This module is part of ansible-core and included in all Ansible installations. In most cases, you can use the short module name even without specifying the collections: keyword. However, we recommend you use the FQCN for easy linking to the module documentation and to avoid conflicting with other collections that may

https://docs.ansible.com/ansible/latest/collections/ansible/builtin/command_module.html




```
$ ansible all -m command -a 'hostname' -o
ubuntu-c | CHANGED | rc=0 | (stdout) ubuntu-c
centos2 | CHANGED | rc=0 | (stdout) centos2
centos1 | CHANGED | rc=0 | (stdout) centos1
centos3 | CHANGED | rc=0 | (stdout) centos3
ubuntu1 | CHANGED | rc=0 | (stdout) ubuntu1
ubuntu2 | CHANGED | rc=0 | (stdout) ubuntu2
ubuntu3 | CHANGED | rc=0 | (stdout) ubuntu3

$ ansible all -a 'hostname' -o
centos2 | CHANGED | rc=0 | (stdout) centos2
ubuntu-c | CHANGED | rc=0 | (stdout) ubuntu-c
centos3 | CHANGED | rc=0 | (stdout) centos3
ubuntu1 | CHANGED | rc=0 | (stdout) ubuntu1
centos1 | CHANGED | rc=0 | (stdout) centos1
ubuntu2 | CHANGED | rc=0 | (stdout) ubuntu2
ubuntu3 | CHANGED | rc=0 | (stdout) ubuntu3

$ ansible all -a 'touch /tmp/test_command_module creates=/tmp/test_command_module'
ubuntu-c | CHANGED | rc=0 >>

centos1 | CHANGED | rc=0 >>

centos2 | CHANGED | rc=0 >>

centos3 | CHANGED | rc=0 >>

ubuntu2 | CHANGED | rc=0 >>

ubuntu3 | CHANGED | rc=0 >>

ubuntu1 | CHANGED | rc=0 >>
```

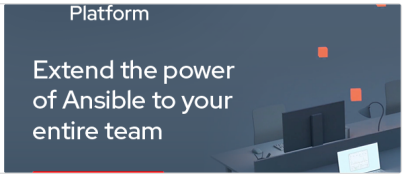
```
$ ansible all -a 'touch /tmp/test_command_module creates=/tmp/test_command_module'
ubuntu-c | SUCCESS | rc=0 >>
skipped, since /tmp/test_command_module existsDid not run command since '/tmp/test_command_module' exists
ubuntu1 | SUCCESS | rc=0 >>
skipped, since /tmp/test_command_module existsDid not run command since '/tmp/test_command_module' exists
centos1 | SUCCESS | rc=0 >>
skipped, since /tmp/test_command_module existsDid not run command since '/tmp/test_command_module' exists
centos2 | SUCCESS | rc=0 >>
skipped, since /tmp/test_command_module existsDid not run command since '/tmp/test_command_module' exists
centos3 | SUCCESS | rc=0 >>
skipped, since /tmp/test_command_module existsDid not run command since '/tmp/test_command_module' exists
ubuntu2 | SUCCESS | rc=0 >>
skipped, since /tmp/test_command_module existsDid not run command since '/tmp/test_command_module' exists
ubuntu3 | SUCCESS | rc=0 >>
skipped, since /tmp/test_command_module existsDid not run command since '/tmp/test_command_module' exists

$ ansible all -a 'rm /tmp/test_command_module removes=/tmp/test_command_module'
ubuntu-c | CHANGED | rc=0 >>

centos2 | CHANGED | rc=0 >>

centos1 | CHANGED | rc=0 >>

centos3 | CHANGED | rc=0 >>

ubuntu1 | CHANGED | rc=0 >>

ubuntu2 | CHANGED | rc=0 >>

ubuntu3 | CHANGED | rc=0 >>

$ ansible all -a 'rm /tmp/test_command_module removes=/tmp/test_command_module'
ubuntu1 | SUCCESS | rc=0 >>
skipped, since /tmp/test_command_module does not existDid not run command since '/tmp/test_command_module' does not exist
centos1 | SUCCESS | rc=0 >>
skipped, since /tmp/test_command_module does not existDid not run command since '/tmp/test_command_module' does not exist
centos2 | SUCCESS | rc=0 >>
skipped, since /tmp/test_command_module does not existDid not run command since '/tmp/test_command_module' does not exist
centos3 | SUCCESS | rc=0 >>
skipped, since /tmp/test_command_module does not existDid not run command since '/tmp/test_command_module' does not exist
ubuntu-c | SUCCESS | rc=0 >>
skipped, since /tmp/test_command_module does not existDid not run command since '/tmp/test_command_module' does not exist
ubuntu2 | SUCCESS | rc=0 >>
skipped, since /tmp/test_command_module does not existDid not run command since '/tmp/test_command_module' does not exist
ubuntu3 | SUCCESS | rc=0 >>
skipped, since /tmp/test_command_module does not existDid not run command since '/tmp/test_command_module' does not exist
```

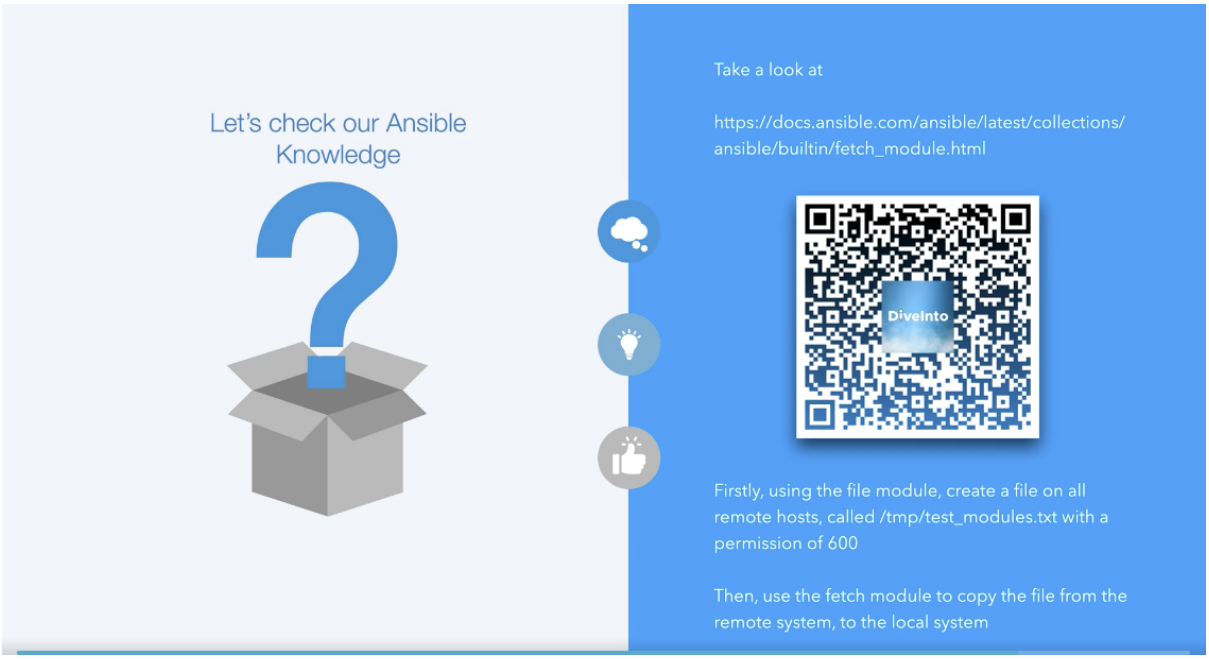
creates 와 remove 파라미터 참고

ansible.builtin.command module - Execute commands on targets - Ansible Documentation

This module is part of ansible-core and included in all Ansible installations. In most cases, you can use the short module name even without specifying the collections: keyword. However, we recommend you use the FQCN for easy linking to the module documentation and to avoid conflicting with other collections that may

https://docs.ansible.com/ansible/latest/collections/ansible/builtin/command_module.html#parameters






```
$ ansible all -m file -a 'path=/tmp/test_module.txt state=touch mode=600' -o
ubuntu-c | CHANGED => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": true,"dest": "/tmp/test_modul
centos1 | CHANGED => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": true,"dest": "/tmp
centos2 | CHANGED => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": true,"dest": "/tmp
centos3 | CHANGED => {"ansible_facts": {"discovered_interpreter_python": "/usr/libexec/platform-python"},"changed": true,"dest": "/tmp
ubuntu1 | CHANGED => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": true,"dest": "/tmp/test_module
ubuntu2 | CHANGED => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": true,"dest": "/tmp/test_module
ubuntu3 | CHANGED => {"ansible_facts": {"discovered_interpreter_python": "/usr/bin/python3"},"changed": true,"dest": "/tmp/test_module

$ ansible all -m fetch -a 'src=/tmp/test_module.txt dest=/tmp' -o
ubuntu-c | CHANGED => {"changed": true,"checksum": "da39a3ee5e6b4b0d3255bfef95601890afd80709","dest": "/tmp/ubuntu-c/tmp/test_module.t
centos1 | CHANGED => {"changed": true,"checksum": "da39a3ee5e6b4b0d3255bfef95601890afd80709","dest": "/tmp/centos1/tmp/test_module.txt
centos3 | CHANGED => {"changed": true,"checksum": "da39a3ee5e6b4b0d3255bfef95601890afd80709","dest": "/tmp/centos3/tmp/test_module.txt
centos2 | CHANGED => {"changed": true,"checksum": "da39a3ee5e6b4b0d3255bfef95601890afd80709","dest": "/tmp/centos2/tmp/test_module.txt
ubuntu1 | CHANGED => {"changed": true,"checksum": "da39a3ee5e6b4b0d3255bfef95601890afd80709","dest": "/tmp/ubuntu1/tmp/test_module.txt
ubuntu2 | CHANGED => {"changed": true,"checksum": "da39a3ee5e6b4b0d3255bfef95601890afd80709","dest": "/tmp/ubuntu2/tmp/test_module.txt
ubuntu3 | CHANGED => {"changed": true,"checksum": "da39a3ee5e6b4b0d3255bfef95601890afd80709","dest": "/tmp/ubuntu3/tmp/test_module.txt
```

3-8. Ansible-doc

```
$ ansible-doc file
$ ansible-doc fetch
```