Chapter 5: Ansible Playbooks - Deep Dive

1. Modules in Ansible Playbooks

Using Different Modules with Ansible

Video Overview

Playbook Modules

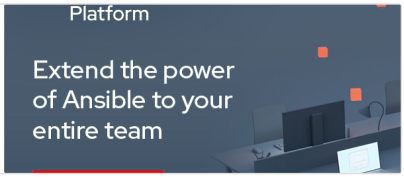
- set_fact
- pause
- prompt
- wait_for
- assemble
- add_host
- group_by
- fetch

1-1. set_fact

- used for gathering facts when executing playbooks
- dynamically add or change facts during execution

ansible.builtin.set_fact module - Set host variable(s) and fact(s). - Ansible Documentation

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




```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Deep Dive/Playbook Modules
$ cd 01

$ cat set_fact_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu3,centos3

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Set a fact
      set_fact:
        our_fact: Ansible Rocks!

    - name: Show custom fact
      debug:
        msg: "{{ our_fact }}"

# Three dots indicate the end of a YAML document
...

$ ansible-playbook set_fact_playbook.yaml

PLAY [ubuntu3,centos3] ********************************************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************************************
ok: [centos3]
ok: [ubuntu3]

TASK [Set a fact] *************************************************************************************************************************************
```

```
ok: [ubuntu3]
ok: [centos3]

TASK [Show custom fact] ********************************************************************************************************************************
ok: [ubuntu3] => {
    "msg": "Ansible Rocks!"
}
ok: [centos3] => {
    "msg": "Ansible Rocks!"
}

PLAY RECAP *********************************************************************************************************************************************
centos3                    : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
$ cd ../02/
$ cat set_fact_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu3,centos3

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Set a fact
      set_fact:
        our_fact: Ansible Rocks!
        ansible_distribution: "{{ ansible_distribution | upper }}"

    - name: Show our_fact
      debug:
        msg: "{{ our_fact }}"

    - name: Show ansible_distribution
      debug:
        msg: "{{ ansible_distribution }}"

# Three dots indicate the end of a YAML document
...

$ ansible-playbook set_fact_playbook.yaml

PLAY [ubuntu3,centos3] *********************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************
ok: [centos3]
ok: [ubuntu3]

TASK [Set a fact] **************************************************************************************************************************************
ok: [ubuntu3]
ok: [centos3]

TASK [Show our_fact] ***********************************************************************************************************************************
ok: [ubuntu3] => {
    "msg": "Ansible Rocks!"
}
ok: [centos3] => {
    "msg": "Ansible Rocks!"
}

TASK [Show ansible_distribution] ***********************************************************************************************************************
ok: [ubuntu3] => {
    "msg": "UBUNTU"
}
ok: [centos3] => {
    "msg": "CENTOS"
}

PLAY RECAP *********************************************************************************************************************************************
centos3                    : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
$ cd ../03
$ cat set_fact_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu3,centos3

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Set our installation variables for CentOS
      set_fact:
        webserver_application_port: 80
        webserver_application_path: /usr/share/nginx/html
        webserver_application_user: root
      when: ansible_distribution == 'CentOS'

    - name: Set our installation variables for Ubuntu
      set_fact:
        webserver_application_port: 8080
        webserver_application_path: /var/www/html
        webserver_application_user: nginx
      when: ansible_distribution == 'Ubuntu'

    - name: Show pre-set distribution based facts
      debug:
        msg: "webserver_application_port:{{ webserver_application_port }} webserver_application_path:{{ webserver_application_path }}

# Three dots indicate the end of a YAML document
...

$ ansible-playbook set_fact_playbook.yaml

PLAY [ubuntu3,centos3] ***********************************************************************************************************************

TASK [Gathering Facts] ***********************************************************************************************************************
ok: [centos3]
ok: [ubuntu3]

TASK [Set our installation variables for CentOS] *********************************************************************************************
skipping: [ubuntu3]
ok: [centos3]

TASK [Set our installation variables for Ubuntu] *********************************************************************************************
ok: [ubuntu3]
skipping: [centos3]

TASK [Show pre-set distribution based facts] **************************************************************************************************
ok: [ubuntu3] => {
    "msg": "webserver_application_port:8080 webserver_application_path:/var/www/html webserver_application_user:nginx"
}
ok: [centos3] => {
    "msg": "webserver_application_port:80 webserver_application_path:/usr/share/nginx/html webserver_application_user:root"
}

PLAY RECAP ***********************************************************************************************************************************
centos3                    : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu3                    : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

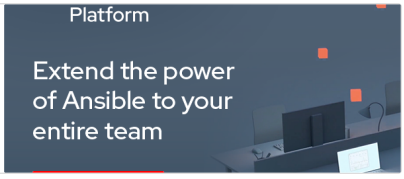
1-2. pause

- pause playbook for a set amount of time or until a prompt is acknowledged

ansible.builtin.pause module - Pause playbook execution - 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/pause_module.html




```
$ cd ../04
$ cat pause_playbook.yaml
---
# YAML documents begin with the document separator ---
```

```
# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu3,centos3

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Pause our playbook for 10 seconds
      pause:
        seconds: 10

# Three dots indicate the end of a YAML document
...

$ ansible-playbook pause_playbook.yaml

PLAY [ubuntu3,centos3] *********************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [centos3]
ok: [ubuntu3]

TASK [Pause our playbook for 10 seconds] ***************************************************************************************************
Pausing for 10 seconds
(ctrl+C then 'C' = continue early, ctrl+C then 'A' = abort)
ok: [ubuntu3]

PLAY RECAP *********************************************************************************************************************************
centos3                    : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
$ cd ../05/
$ cat pause_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu3,centos3

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Prompt user to verify before continue
      pause:
        prompt: Please check that the webserver is running, press enter to continue

# Three dots indicate the end of a YAML document
...

$ ansible-playbook pause_playbook.yaml

PLAY [ubuntu3,centos3] *********************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [centos3]
ok: [ubuntu3]

TASK [Prompt user to verify before continue] ***********************************************************************************************
[Prompt user to verify before continue]
Please check that the webserver is running, press enter to continue:
^Mok: [ubuntu3]

PLAY RECAP *********************************************************************************************************************************
centos3                    : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
$ cd ../06
$ ls
ansible.cfg  group_vars  host_vars  hosts  run_webserver_playbook.yaml  wait_for_playbook.yaml
```

```
$ cat run_webserver_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu3,centos3

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Install EPEL
      yum:
        name: epel-release
        update_cache: yes
        state: latest
      when: ansible_distribution == 'CentOS'

    - name: Install Nginx
      package:
        name: nginx
        state: latest

    - name: Restart nginx
      service:
        name: nginx
        state: restarted

# Three dots indicate the end of a YAML document
...

$ cat wait_for_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu3,centos3

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Wait for the webserver to be running on port 80
      wait_for:
        port: 80

# Three dots indicate the end of a YAML document
...

$ ansible-playbook wait_for_playbook.yaml

PLAY [ubuntu3,centos3] *********************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [centos3]
ok: [ubuntu3]

TASK [Wait for the webserver to be running on port 80] *************************************************************************************
ok: [centos3]
ok: [ubuntu3]

PLAY RECAP *********************************************************************************************************************************
centos3                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

$ ansible centos3 -m service -a "name=nginx state=stopped"
$ ansible-playbook wait_for_playbook.yaml &
$ ansible centos3 -m service -a "name=nginx state=started"
```

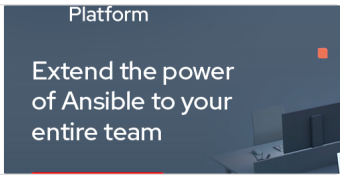
1-3. assemble

- Assemble configuration files from fragments
- Allows configuration files to be broken into segments and concatenated to form a destination file
- Great to use, when application or tool, require it’s configuration as a single file, but, you wish to manage it as separate entities

ansible.builtin.assemble module - Assemble configuration files from fragments - 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 have the same

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




```
$ cd ../07/

$ cat conf.d/defaults
## Defaults

Port 22
Protocol 2
ForwardX11 yes
GSSAPIAuthentication no

$ cat conf.d/centos1
## Custom for centos1
Host centos1
  User root
  Port 2222

$ cat assemble_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Assemble conf.d to sshd_config
      assemble:
        src: conf.d
        dest: sshd_config

# Three dots indicate the end of a YAML document
...

$ ansible-playbook assemble_playbook.yaml

PLAY [ubuntu-c] ***********************************************************************************************************************************

TASK [Gathering Facts] ****************************************************************************************************************************
ok: [ubuntu-c]

TASK [Assemble conf.d to sshd_config] *************************************************************************************************************
ok: [ubuntu-c]

PLAY RECAP ****************************************************************************************************************************************
ubuntu-c                   : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

$ cat sshd_config
## Custom for centos1
Host centos1
  User root
  Port 2222

## Defaults

Port 22
Protocol 2
ForwardX11 yes
GSSAPIAuthentication no

$ ssh -F sshd_config centos1
Last login: Thu Jun 30 00:36:09 2022 from 172.19.0.3
[root@centos1 ~]# exit
```

1-4. Add Host

1-4-1. add_host

- Dynamically add targets to our running playbooks

- Create new inventory groups and targets, on the fly
- Great, for when a resource is created during execution and you wish to include it, in your playbook execution

ansible.builtin.add_host module - Add a host (and alternatively a group) to the ansible-playbook in-memory inventory - Ansible Documentation

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

```
$ cd ../08
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Deep Dive/Playbook Modules/08

$ cat add_host_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Add centos1 to adhoc_group
      add_host:
        name: centos1
        groups: adhoc_group1, adhoc_group2

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: adhoc_group1

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Ping all in adhoc_group1
      ping:

# Three dots indicate the end of a YAML document
...

$ ansible-playbook add_host_playbook.yaml

PLAY [ubuntu-c] ***********************************************************************************************************************************

TASK [Gathering Facts] ****************************************************************************************************************************
ok: [ubuntu-c]

TASK [Add centos1 to adhoc_group] *****************************************************************************************************************
changed: [ubuntu-c]

PLAY [adhoc_group1] *******************************************************************************************************************************

TASK [Gathering Facts] ****************************************************************************************************************************
ok: [centos1]

TASK [Ping all in adhoc_group1] *******************************************************************************************************************
ok: [centos1]

PLAY RECAP ****************************************************************************************************************************************
centos1                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu-c                   : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

$ cd ../09
$ cat add_host_playbook.yaml
---
# YAML documents begin with the document separator ---

- hosts: ubuntu-c

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
```

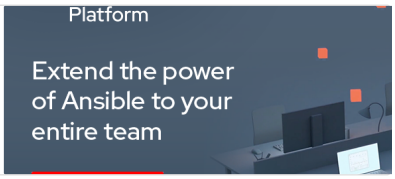
```
    - name: Add centos1 to adhoc_group
      add_host:
        name: centos1
        groups: adhoc_group1, adhoc_group2

- hosts: adhoc_group1

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Ping all in adhoc_group1
      ping:

# Three dots indicate the end of a YAML document
...

$ ansible-playbook add_host_playbook.yaml

PLAY [ubuntu-c] ****************************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************
ok: [ubuntu-c]

TASK [Add centos1 to adhoc_group] **********************************************************************************************************************
changed: [ubuntu-c]

PLAY [adhoc_group1] ************************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************
ok: [centos1]

TASK [Ping all in adhoc_group1] ************************************************************************************************************************
ok: [centos1]

PLAY RECAP *********************************************************************************************************************************************
centos1                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu-c                   : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

1-4-2. group_by

- Create groups, based on facts
- Utilize facts to dynamically create, associated groups

ansible.builtin.group_by module - Create Ansible groups based on facts - Ansible Documentation

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




```
$ cd ../10
$ cat group_by_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: all

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Create group based on ansible_distribution
      group_by:
        key: "custom_{{ ansible_distribution | lower }}"

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: custom_centos

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Ping all in custom_centos
```

```
      ping:

# Three dots indicate the end of a YAML document
...

$ ansible-playbook group_by_playbook.yaml

PLAY [all] ********************************************************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************************************
ok: [ubuntu-c]
ok: [centos3]
ok: [centos2]
ok: [ubuntu1]
ok: [ubuntu2]
ok: [ubuntu3]
ok: [centos1]

TASK [Create group based on ansible_distribution] *****************************************************************************************************
changed: [ubuntu-c]
changed: [centos1]
changed: [centos2]
changed: [centos3]
changed: [ubuntu1]
changed: [ubuntu2]
changed: [ubuntu3]

PLAY [custom_centos] **********************************************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************************************
ok: [centos1]
ok: [centos2]
ok: [centos3]

TASK [Ping all in custom_centos] **********************************************************************************************************************
ok: [centos1]
ok: [centos3]
ok: [centos2]

PLAY RECAP ********************************************************************************************************************************************
centos1                    : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu-c                   : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

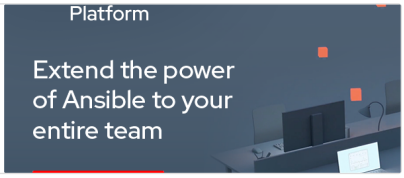
1-4-3. fetch

- capture files
- capture files from remote hosts and targets

ansible.builtin.fetch module - Fetch files from remote nodes - 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/fetch_module.html




```
$ cd ../11
$ cat fetch_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: centos

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  # destination에 해당 디렉토리가 없을 경우 생성
  tasks:
    - name: Fetch /etc/redhat-release
      fetch:
        src: /etc/redhat-release
        dest: /tmp/redhat-release
```

```
# Three dots indicate the end of a YAML document
...

$ ansible-playbook fetch_playbook.yaml

PLAY [centos] ******************************************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************************
ok: [centos3]
ok: [centos1]
ok: [centos2]

TASK [Fetch /etc/redhat-release] ***********************************************************************************************************************************
changed: [centos1]
changed: [centos3]
changed: [centos2]

PLAY RECAP *********************************************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

# 로컬 호스트에서 생성된 디렉토리 및 파일 확인

$ cat /tmp/redhat-release/centos1/etc/redhat-release
CentOS Linux release 8.4.2105
$ cat /tmp/redhat-release/centos2/etc/redhat-release
CentOS Linux release 8.4.2105
$ cat /tmp/redhat-release/centos3/etc/redhat-release
CentOS Linux release 8.4.2105
```

2. Dynamic Inventories

How to use them and how to create them

Recap

Inventories

Dynamic Inventory Key Requirements

- Needs to be an executable file. Can be written in any language providing that it can be executed from the command line
- Accepts the command line options of --list and --host hostname
- Returns a JSON encoded dictionary of inventory content when used with --list
- Returns a basic JSON encoded dictionary structure for --host hostname

```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Deep Dive/Dynamic Inventories/01

$ ./inventory.py
usage: inventory.py [-h] [--list] [--host HOST]
```

▼ inventory.py

```
#!/usr/bin/env python3

'''
Dynamic inventory for Ansible in Python
'''

# Use print functionality from Python 3 for compatibility
from __future__ import print_function

import argparse
import logging
```

```
# Attempt to import json, if it fails, import simplejson
try:
    import json
except ImportError:
    import simplejson as json

# Inherit from object for Python 2/3 compatibility
class Inventory(object):

    # Constructor
    def __init__(self, include_hostvars_in_list):

        # Configure logger
        #self.configure_logger()

        # Capture and store include_hostvars_in_list
        self.include_hostvars_in_list = include_hostvars_in_list

        # Capture the script command line arguments
        parser = argparse.ArgumentParser()
        parser.add_argument('--list', action='store_true',
                            help='list inventory')
        parser.add_argument('--host', action='store',
                            help='show HOST variables')
        self.args = parser.parse_args()

        # If not called with --host or --list, show usage and exit
        if not (self.args.list or self.args.host):
            parser.print_usage()
            raise SystemExit

        # Capture and store the inventory
        self.define_inventory()

        # When called with --list, print the inventory
        if self.args.list:
            self.print_json(self.list())

        # If called with --host, print host information
        elif self.args.host:
            self.print_json(self.host())

    def define_inventory(self):
        self.groups = {
            "centos": {
                "hosts": ["centos1", "centos2", "centos3"],
                "vars": {
                    "ansible_user": 'root'
                }
            },
            "control": {
                "hosts": ["ubuntu-c"],
            },
            "ubuntu": {
                "hosts": ["ubuntu1", "ubuntu2", "ubuntu3"],
                "vars": {
                    "ansible_become": True,
                    "ansible_become_pass": 'password'
                }
            },
            "linux": {
                "children": ["centos", "ubuntu"],
            }}

        self.hostvars = {
            'centos1': {
                'ansible_port': 2222
            },
            'ubuntu-c': {
                'ansible_connection': 'local'
            }
        }

    # Pretty print JSON
    def print_json(self, content):
        print(json.dumps(content, indent=4, sort_keys=True))

    # Return inventory dictionary
    def list(self):

        #self.logger.info('list executed')

        # If include_hostvars_in_list is True, merge the hostvars
        # as _meta data
        if self.include_hostvars_in_list:
            merged = self.groups
```

```
            merged['_meta'] = {}
            merged['_meta']['hostvars'] = self.hostvars
            return merged

        # Otherwise, return the groups without hostvars
        else:
            return self.groups

    # Return host dictionary
    def host(self):

        #self.logger.info('host executed for {}'.format(self.args.host))

        # If the requested hosts exists in hostvars, return it
        if self.args.host in self.hostvars:
            return self.hostvars[self.args.host]

        # Otherwise, return an empty list
        else:
            return {}

    # Logger, for debugging as stdout is used by the script
    def configure_logger(self):
        self.logger = logging.getLogger('ansible_dynamic_inventory')
        self.hdlr = logging.FileHandler('/var/tmp/ansible_dynamic_inventory.log')
        self.formatter = logging.Formatter('%(asctime)s %(levelname)s %(message)s')
        self.hdlr.setFormatter(self.formatter)
        self.logger.addHandler(self.hdlr)
        self.logger.setLevel(logging.DEBUG)

# Call the Inventory class constructor (__init__)
# Pass include_hostsvars_in_list as True to include hostvars
# as _meta data in list output
Inventory(include_hostvars_in_list=False)
```

```
$ ./inventory.py --list
{
    "centos": {
        "hosts": [
            "centos1",
            "centos2",
            "centos3"
        ],
        "vars": {
            "ansible_user": "root"
        }
    },
    "control": {
        "hosts": [
            "ubuntu-c"
        ]
    },
    "linux": {
        "children": [
            "centos",
            "ubuntu"
        ]
    },
    "ubuntu": {
        "hosts": [
            "ubuntu1",
            "ubuntu2",
            "ubuntu3"
        ],
        "vars": {
            "ansible_become": true,
            "ansible_become_pass": "password"
        }
    }
}

$ ./inventory.py --host centos1
{
    "ansible_port": 2222
}

$ ./inventory.py --host centos2
{}

$ ansible all -i inventory.py --list-hosts
  hosts (7):
    ubuntu-c
    centos1
    centos2
```

```
    centos3
    ubuntu1
    ubuntu2
    ubuntu3

$ ansible all -i inventory.py -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": "pon
centos1 | 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"}
```

debugging dynamic inventories

```
$ cd ../02

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

$ cat /var/tmp/ansible_dynamic_inventory.log
2022-07-01 01:40:20,591 INFO list executed
2022-07-01 01:40:20,618 INFO host executed for ubuntu1
2022-07-01 01:40:20,643 INFO host executed for centos3
2022-07-01 01:40:20,668 INFO host executed for ubuntu2
2022-07-01 01:40:20,694 INFO host executed for ubuntu-c
2022-07-01 01:40:20,718 INFO host executed for ubuntu3
2022-07-01 01:40:20,743 INFO host executed for centos2
2022-07-01 01:40:20,770 INFO host executed for centos1
```

많은 호스트를 실행할 때 시간 측정(performance)

```
$ cd ../03

$ for i in {1..1000}
> do
> echo \'fake{i}\'\,
> done | tr "\n" " "

# inventory.py에 fake 호스트가 추가되었는지 확인

$ tail -f /var/tmp/ansible_dynamic_inventory.log &
$ time ansible all -i inventory.py --list-hosts
fake992
    fake993
    fake994
    fake995
    fake996
    fake997
    fake998
    fake999
    fake1000
    centos1
    centos2
    centos3
    ubuntu1
    ubuntu2
    ubuntu3
2022-07-01 01:59:36,183 INFO host executed for fake918
2022-07-01 01:59:36,211 INFO host executed for fake651

real    0m28.216s
user    0m17.729s
sys     0m6.467s
```

성능 개선(_meta)

```
$ cd ../04
$ cat inventory.py | tail -1
Inventory(include_hostvars_in_list=True)
```

```
$ time ansible all -i inventory.py --list-hosts
    fake996
    fake997
    fake998
    fake999
    fake1000
    centos1
    centos2
    centos3
    ubuntu1
    ubuntu2
    ubuntu3

real    0m1.256s
user    0m0.349s
sys     0m0.182s

# 백그라운드 작업 중지
$ jobs
[1]   Running                 tail -f /var/tmp/ansible_dynamic_inventory.log &  (wd: ~/diveintoansible/Ansible Playbooks, Deep Dive/Dy
[2]-  Running                 tail -f /var/tmp/ansible_dynamic_inventory.log &  (wd: ~/diveintoansible/Ansible Playbooks, Deep Dive/Dy
[3]+  Running                 tail -f /var/tmp/ansible_dynamic_inventory.log &
$ kill %1
$ kill %2
$ kill %3
```

3. Register and When

For registering information and Acting Upon Certain Conditions


```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Deep Dive/Register and When/01

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

$ cat register_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
```

```
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Exploring register
      command: hostname -s
      register: hostname_output

# Three dots indicate the end of a YAML document
...

$ ansible-playbook register_playbook.yaml

PLAY [linux] ***********************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************
ok: [centos3]
ok: [centos2]
ok: [centos1]
ok: [ubuntu2]
ok: [ubuntu1]
ok: [ubuntu3]

TASK [Exploring register] **********************************************************************************************************************
changed: [centos1]
changed: [centos2]
changed: [ubuntu2]
changed: [centos3]
changed: [ubuntu1]
changed: [ubuntu3]

PLAY RECAP *************************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

How to use register

```
$ cd ../02
$ cat register_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Exploring register
      command: hostname -s
      register: hostname_output

    - name: Show hostname_output
      debug:
        var: hostname_output

# Three dots indicate the end of a YAML document
...

$ ansible-playbook register_playbook.yaml

PLAY [linux] ***********************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************
ok: [centos3]
ok: [centos1]
ok: [centos2]
ok: [ubuntu1]
ok: [ubuntu2]
ok: [ubuntu3]

TASK [Exploring register] **********************************************************************************************************************
changed: [centos1]
changed: [centos3]
changed: [centos2]
```

```
changed: [ubuntu1]
changed: [ubuntu2]
changed: [ubuntu3]

TASK [Show hostname_output] ***************************************************************************************************************************
ok: [centos1] => {
    "hostname_output": {
        "changed": true,
        "cmd": [
            "hostname",
            "-s"
        ],
        "delta": "0:00:00.002214",
        "end": "2022-07-01 04:01:55.235723",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-07-01 04:01:55.233509",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "centos1",
        "stdout_lines": [
            "centos1"
        ]
    }
}
ok: [centos2] => {
    "hostname_output": {
        "changed": true,
        "cmd": [
            "hostname",
            "-s"
        ],
        "delta": "0:00:00.001904",
        "end": "2022-07-01 04:01:55.246238",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-07-01 04:01:55.244334",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "centos2",
        "stdout_lines": [
            "centos2"
        ]
    }
}
ok: [centos3] => {
    "hostname_output": {
        "changed": true,
        "cmd": [
            "hostname",
            "-s"
        ],
        "delta": "0:00:00.002056",
        "end": "2022-07-01 04:01:55.246522",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-07-01 04:01:55.244466",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "centos3",
        "stdout_lines": [
            "centos3"
        ]
    }
}
ok: [ubuntu1] => {
    "hostname_output": {
        "changed": true,
        "cmd": [
            "hostname",
            "-s"
        ],
        "delta": "0:00:00.002883",
        "end": "2022-07-01 04:01:55.254202",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-07-01 04:01:55.251319",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "ubuntu1",
        "stdout_lines": [
            "ubuntu1"
        ]
```

```
    }
}
ok: [ubuntu2] => {
    "hostname_output": {
        "changed": true,
        "cmd": [
            "hostname",
            "-s"
        ],
        "delta": "0:00:00.004475",
        "end": "2022-07-01 04:01:55.276663",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-07-01 04:01:55.272188",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "ubuntu2",
        "stdout_lines": [
            "ubuntu2"
        ]
    }
}
ok: [ubuntu3] => {
    "hostname_output": {
        "changed": true,
        "cmd": [
            "hostname",
            "-s"
        ],
        "delta": "0:00:00.001904",
        "end": "2022-07-01 04:01:55.500582",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-07-01 04:01:55.498678",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "ubuntu3",
        "stdout_lines": [
            "ubuntu3"
        ]
    }
}

PLAY RECAP *********************************************************************************************************************************************
centos1                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Access register variables

```
$ cd ../03
$ cat register_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Exploring register
      command: hostname -s
      register: hostname_output

    - name: Show hostname_output
      debug:
        var: hostname_output.stdout

# Three dots indicate the end of a YAML document
...

$ ansible-playbook register_playbook.yaml

PLAY [linux] *******************************************************************************************************************************************
```

```
TASK [Gathering Facts] *********************************************************************************************************************
ok: [centos1]
ok: [centos2]
ok: [ubuntu2]
ok: [centos3]
ok: [ubuntu1]
ok: [ubuntu3]

TASK [Exploring register] ******************************************************************************************************************
changed: [centos1]
changed: [centos2]
changed: [ubuntu1]
changed: [ubuntu2]
changed: [centos3]
changed: [ubuntu3]

TASK [Show hostname_output] ****************************************************************************************************************
ok: [centos1] => {
    "hostname_output.stdout": "centos1"
}
ok: [centos2] => {
    "hostname_output.stdout": "centos2"
}
ok: [centos3] => {
    "hostname_output.stdout": "centos3"
}
ok: [ubuntu1] => {
    "hostname_output.stdout": "ubuntu1"
}
ok: [ubuntu2] => {
    "hostname_output.stdout": "ubuntu2"
}
ok: [ubuntu3] => {
    "hostname_output.stdout": "ubuntu3"
}

PLAY RECAP *********************************************************************************************************************************
centos1                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Varaible ways to use register

```
$ cd ../04
$ cat register_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Exploring register
      command: hostname -s
      when: ansible_distribution == "CentOS" and ansible_distribution_major_version == "8"

# Three dots indicate the end of a YAML document
...

$ ansible-playbook register_playbook.yaml

PLAY [linux] *******************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [centos1]
ok: [ubuntu2]
ok: [centos2]
ok: [ubuntu1]
ok: [ubuntu3]
ok: [centos3]

TASK [Exploring register] ******************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
```

```
skipping: [ubuntu3]
changed: [centos2]
changed: [centos1]
changed: [centos3]

PLAY RECAP *********************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu2                    : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu3                    : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

# Adhoc 과 filter로 확인
$ ansible ubuntu1 -m setup -a filter='ansible_distribution*'
ubuntu1 | SUCCESS => {
    "ansible_facts": {
        "ansible_distribution": "Ubuntu",
        "ansible_distribution_file_parsed": true,
        "ansible_distribution_file_path": "/etc/os-release",
        "ansible_distribution_file_variety": "Debian",
        "ansible_distribution_major_version": "20",
        "ansible_distribution_release": "focal",
        "ansible_distribution_version": "20.04",
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false
}
```

```
$ cd ../05
$ cat register_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Exploring register
      command: hostname -s
      when: ( ansible_distribution == "CentOS" and ansible_distribution_major_version == "8" ) or
            ( ansible_distribution == "Ubuntu" and ansible_distribution_major_version == "20" )

# Three dots indicate the end of a YAML document
...

$ ansible-playbook register_playbook.yaml

PLAY [linux] *******************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [centos2]
ok: [centos3]
ok: [centos1]
ok: [ubuntu1]
ok: [ubuntu2]
ok: [ubuntu3]

TASK [Exploring register] ******************************************************************************************************************
changed: [centos1]
changed: [ubuntu1]
changed: [centos3]
changed: [centos2]
changed: [ubuntu2]
changed: [ubuntu3]

PLAY RECAP *********************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
$ cd ../06
$ cat register_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Exploring register
      command: hostname -s
      when: ( ansible_distribution == "CentOS" and ansible_distribution_major_version | int >= 8 ) or
            ( ansible_distribution == "Ubuntu" and ansible_distribution_major_version | int >= 20 )

# Three dots indicate the end of a YAML document
...

$ ansible-playbook register_playbook.yaml

PLAY [linux] ***********************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************
ok: [centos2]
ok: [ubuntu1]
ok: [centos1]
ok: [ubuntu2]
ok: [centos3]
ok: [ubuntu3]

TASK [Exploring register] **********************************************************************************************************************
changed: [centos1]
changed: [ubuntu1]
changed: [centos3]
changed: [ubuntu2]
changed: [centos2]
changed: [ubuntu3]

PLAY RECAP *************************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
$ cd ../07
$ cat register_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Exploring register
      command: hostname -s
      when:
        - ansible_distribution == "CentOS"
        - ansible_distribution_major_version | int >= 8

# Three dots indicate the end of a YAML document
...

$ ansible-playbook register_playbook.yaml

PLAY [linux] ***********************************************************************************************************************************
```

```
TASK [Gathering Facts] *********************************************************************************************************************
ok: [centos3]
ok: [centos1]
ok: [centos2]
ok: [ubuntu1]
ok: [ubuntu2]
ok: [ubuntu3]

TASK [Exploring register] ******************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
changed: [centos3]
changed: [centos1]
changed: [centos2]

PLAY RECAP *********************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu2                    : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu3                    : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

```
$ cd ../08
$ cat register_when_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Exploring register
      command: hostname -s
      when:
        - ansible_distribution == "CentOS"
        - ansible_distribution_major_version | int >= 8
      register: command_register

    - name: Show register
      debug:
        var: command_register

# Three dots indicate the end of a YAML document
...

$ ansible-playbook register_when_playbook.yaml

PLAY [linux] *******************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [centos1]
ok: [centos3]
ok: [ubuntu1]
ok: [centos2]
ok: [ubuntu2]
ok: [ubuntu3]

TASK [Exploring register] ******************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
changed: [centos1]
changed: [centos3]
changed: [centos2]

TASK [Show register] ***********************************************************************************************************************
ok: [centos1] => {
    "command_register": {
        "changed": true,
        "cmd": [
            "hostname",
            "-s"
        ],
```

```
        "delta": "0:00:00.001550",
        "end": "2022-07-01 04:32:45.169364",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-07-01 04:32:45.167814",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "centos1",
        "stdout_lines": [
            "centos1"
        ]
    }
}
ok: [centos2] => {
    "command_register": {
        "changed": true,
        "cmd": [
            "hostname",
            "-s"
        ],
        "delta": "0:00:00.002878",
        "end": "2022-07-01 04:32:45.185190",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-07-01 04:32:45.182312",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "centos2",
        "stdout_lines": [
            "centos2"
        ]
    }
}
ok: [centos3] => {
    "command_register": {
        "changed": true,
        "cmd": [
            "hostname",
            "-s"
        ],
        "delta": "0:00:00.001411",
        "end": "2022-07-01 04:32:45.180217",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-07-01 04:32:45.178806",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "centos3",
        "stdout_lines": [
            "centos3"
        ]
    }
}
ok: [ubuntu1] => {
    "command_register": {
        "changed": false,
        "skip_reason": "Conditional result was False",
        "skipped": true
    }
}
ok: [ubuntu2] => {
    "command_register": {
        "changed": false,
        "skip_reason": "Conditional result was False",
        "skipped": true
    }
}
ok: [ubuntu3] => {
    "command_register": {
        "changed": false,
        "skip_reason": "Conditional result was False",
        "skipped": true
    }
}

PLAY RECAP *********************************************************************************************************************************
centos1                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

```
$ cd ../09
$ cat register_when_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Exploring register
      command: hostname -s
      when:
        - ansible_distribution == "CentOS"
        - ansible_distribution_major_version | int >= 8
      register: command_register

    - name: Install patch when changed
      yum:
        name: patch
        state: present
      when: command_register.changed

# Three dots indicate the end of a YAML document
...

$ ansible-playbook register_when_playbook.yaml

PLAY [linux] ***********************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************
ok: [centos2]
ok: [centos3]
ok: [ubuntu1]
ok: [centos1]
ok: [ubuntu2]
ok: [ubuntu3]

TASK [Exploring register] **********************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
changed: [centos1]
changed: [centos3]
changed: [centos2]

TASK [Install patch when changed] **************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
changed: [centos2]
changed: [centos1]
changed: [centos3]

PLAY RECAP *************************************************************************************************************************************
centos1                    : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
ubuntu2                    : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
ubuntu3                    : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

```
$ cd ../10
$ cat register_when_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux
```

```
  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Exploring register
      command: hostname -s
      when:
        - ansible_distribution == "CentOS"
        - ansible_distribution_major_version | int >= 8
      register: command_register

    - name: Install patch when changed
      yum:
        name: patch
        state: present
      when: command_register is changed

# Three dots indicate the end of a YAML document
...

$ ansible-playbook register_when_playbook.yaml

PLAY [linux] ***********************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************
ok: [centos1]
ok: [centos2]
ok: [centos3]
ok: [ubuntu2]
ok: [ubuntu1]
ok: [ubuntu3]

TASK [Exploring register] **********************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
changed: [centos1]
changed: [centos3]
changed: [centos2]

TASK [Install patch when changed] **************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
ok: [centos1]
ok: [centos3]
ok: [centos2]

PLAY RECAP *************************************************************************************************************************************
centos1                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
ubuntu2                    : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
ubuntu3                    : ok=1    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

```
$ cd ../11
$ cat register_when_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Exploring register
      command: hostname -s
      when:
        - ansible_distribution == "CentOS"
        - ansible_distribution_major_version | int >= 8
      register: command_register

    - name: Install patch when changed
      yum:
        name: patch
        state: present
```

```
      when: command_register is changed

    - name: Install patch when skipped
      apt:
        name: patch
        state: present
      when: command_register is skipped

# Three dots indicate the end of a YAML document
...

$ ansible-playbook register_when_playbook.yaml

PLAY [linux] ***********************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************
ok: [centos1]
ok: [ubuntu1]
ok: [centos2]
ok: [ubuntu2]
ok: [centos3]
ok: [ubuntu3]

TASK [Exploring register] **********************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
changed: [centos2]
changed: [centos1]
changed: [centos3]

TASK [Install patch when changed] **************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
ok: [centos1]
ok: [centos2]
ok: [centos3]

TASK [Install patch when skipped] **************************************************************************************************************
skipping: [centos1]
skipping: [centos2]
skipping: [centos3]
ok: [ubuntu2]
ok: [ubuntu1]
ok: [ubuntu3]

PLAY RECAP *************************************************************************************************************************************
centos1                    : ok=3    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
centos2                    : ok=3    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
centos3                    : ok=3    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

4. Using Loops

Exploring the multitude of ways for using Loops in Ansible

4-1. with_items

```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Deep Dive/Looping/01

$ cat motd_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Vars: variables that will apply to the play, on all target systems
  vars:
    motd_centos: "Welcome to CentOS Linux - Ansible Rocks\n"
    motd_ubuntu: "Welcome to Ubuntu Linux - Ansible Rocks\n"

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Configure a MOTD (message of the day)
      copy:
        content: "{{ motd_centos }}"
        dest: /etc/motd
      notify: MOTD changed
      when: ansible_distribution == "CentOS"

    - name: Configure a MOTD (message of the day)
      copy:
        content: "{{ motd_ubuntu }}"
        dest: /etc/motd
      notify: MOTD changed
      when: ansible_distribution == "Ubuntu"

  # Handlers: the list of handlers that are executed as a notify key from a task
  handlers:
    - name: MOTD changed
      debug:
        msg: The MOTD was changed

# Three dots indicate the end of a YAML document
...

$ cd ../02
$ cat motd_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Configure a MOTD (message of the day)
      copy:
        content: "Welcome to {{ ansible_distribution }} Linux - Ansible Rocks\n"
        dest: /etc/motd
      notify: MOTD changed

  # Handlers: the list of handlers that are executed as a notify key from a task
  handlers:
    - name: MOTD changed
      debug:
        msg: The MOTD was changed

# Three dots indicate the end of a YAML document
...

$ cd ../03
$ cat motd_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
```

```
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Configure a MOTD (message of the day)
      copy:
        content: "Welcome to {{ item }} Linux - Ansible Rocks!\n"
        dest: /etc/motd
      notify: MOTD changed
      with_items: [ 'CentOS', 'Ubuntu' ]
      when: ansible_distribution == item

  # Handlers: the list of handlers that are executed as a notify key from a task
  handlers:
    - name: MOTD changed
      debug:
        msg: The MOTD was changed

# Three dots indicate the end of a YAML document
...

$ ansible-playbook motd_playbook.yaml

PLAY [linux] ***********************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************
ok: [centos3]
ok: [centos1]
ok: [ubuntu1]
ok: [centos2]
ok: [ubuntu2]
ok: [ubuntu3]

TASK [Configure a MOTD (message of the day)] ***************************************************************************************************
skipping: [ubuntu1] => (item=CentOS)
skipping: [ubuntu2] => (item=CentOS)
changed: [centos2] => (item=CentOS)
skipping: [centos2] => (item=Ubuntu)
changed: [ubuntu2] => (item=Ubuntu)
changed: [centos3] => (item=CentOS)
skipping: [centos3] => (item=Ubuntu)
skipping: [ubuntu3] => (item=CentOS)
changed: [ubuntu1] => (item=Ubuntu)
changed: [centos1] => (item=CentOS)
skipping: [centos1] => (item=Ubuntu)
changed: [ubuntu3] => (item=Ubuntu)

RUNNING HANDLER [MOTD changed] *****************************************************************************************************************
ok: [centos2] => {
    "msg": "The MOTD was changed"
}
ok: [ubuntu2] => {
    "msg": "The MOTD was changed"
}
ok: [centos3] => {
    "msg": "The MOTD was changed"
}
ok: [ubuntu1] => {
    "msg": "The MOTD was changed"
}
ok: [centos1] => {
    "msg": "The MOTD was changed"
}
ok: [ubuntu3] => {
    "msg": "The MOTD was changed"
}

PLAY RECAP *************************************************************************************************************************************
centos1                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

$ cd ../04
$ cat motd_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux
```

```
  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Configure a MOTD (message of the day)
      copy:
        content: "Welcome to {{ item }} Linux - Ansible Rocks!\n"
        dest: /etc/motd
      notify: MOTD changed
      with_items:
        - CentOS
        - Ubuntu
      when: ansible_distribution == item

  # Handlers: the list of handlers that are executed as a notify key from a task
  handlers:
    - name: MOTD changed
      debug:
        msg: The MOTD was changed

# Three dots indicate the end of a YAML document
...

$ ansible-playbook motd_playbook.yaml

PLAY [linux] ***********************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************
ok: [centos1]
ok: [ubuntu1]
ok: [centos3]
ok: [ubuntu2]
ok: [centos2]
ok: [ubuntu3]

TASK [Configure a MOTD (message of the day)] ***************************************************************************************************
skipping: [ubuntu1] => (item=CentOS)
skipping: [ubuntu2] => (item=CentOS)
ok: [ubuntu2] => (item=Ubuntu)
ok: [centos1] => (item=CentOS)
skipping: [centos1] => (item=Ubuntu)
ok: [centos2] => (item=CentOS)
skipping: [centos2] => (item=Ubuntu)
ok: [centos3] => (item=CentOS)
skipping: [centos3] => (item=Ubuntu)
skipping: [ubuntu3] => (item=CentOS)
ok: [ubuntu1] => (item=Ubuntu)
ok: [ubuntu3] => (item=Ubuntu)

PLAY RECAP *************************************************************************************************************************************
centos1                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

$ cd ../05
$ cat user_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Creating user
      user:
        name: "{{ item }}"
      with_items:
        - james
        - hayley
        - lily
        - anwen

# Three dots indicate the end of a YAML document
...

$ ansible-playbook user_playbook.yaml

PLAY [linux] ***********************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************
ok: [centos1]
```

```
ok: [ubuntu1]
ok: [ubuntu2]
ok: [centos3]
ok: [ubuntu3]
ok: [centos2]

TASK [Creating user] ***********************************************************************************************************************************
changed: [centos1] => (item=james)
changed: [ubuntu1] => (item=james)
changed: [centos2] => (item=james)
changed: [centos3] => (item=james)
changed: [ubuntu2] => (item=james)
changed: [centos1] => (item=hayley)
changed: [ubuntu1] => (item=hayley)
changed: [centos3] => (item=hayley)
changed: [centos2] => (item=hayley)
changed: [ubuntu2] => (item=hayley)
changed: [centos1] => (item=lily)
changed: [ubuntu1] => (item=lily)
changed: [centos3] => (item=lily)
changed: [centos2] => (item=lily)
changed: [ubuntu2] => (item=lily)
changed: [centos1] => (item=anwen)
changed: [ubuntu1] => (item=anwen)
changed: [centos3] => (item=anwen)
changed: [centos2] => (item=anwen)
changed: [ubuntu2] => (item=anwen)
changed: [ubuntu3] => (item=james)
changed: [ubuntu3] => (item=hayley)
changed: [ubuntu3] => (item=lily)
changed: [ubuntu3] => (item=anwen)

PLAY RECAP *********************************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

$ ssh centos3 tail -5 /etc/passwd
ansible:x:1000:1000::/home/ansible:/bin/bash
james:x:1001:1001::/home/james:/bin/bash
hayley:x:1002:1002::/home/hayley:/bin/bash
lily:x:1003:1003::/home/lily:/bin/bash
anwen:x:1004:1004::/home/anwen:/bin/bash

# Remove user

$ cd ../06
$ cat user_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Removing user
      user:
        name: "{{ item }}"
        state: absent
      with_items:
        - james
        - hayley
        - lily
        - anwen

# Three dots indicate the end of a YAML document
...

$ ansible-playbook user_playbook.yaml

PLAY [linux] *******************************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************
ok: [centos3]
ok: [ubuntu2]
ok: [ubuntu1]
ok: [centos1]
ok: [centos2]
ok: [ubuntu3]
```

```
TASK [Removing user] ***********************************************************************************************************************
changed: [centos2] => (item=james)
changed: [centos1] => (item=james)
changed: [ubuntu1] => (item=james)
changed: [ubuntu2] => (item=james)
changed: [centos3] => (item=james)
changed: [centos2] => (item=hayley)
changed: [ubuntu1] => (item=hayley)
changed: [ubuntu2] => (item=hayley)
changed: [centos1] => (item=hayley)
changed: [centos3] => (item=hayley)
changed: [centos2] => (item=lily)
changed: [ubuntu1] => (item=lily)
changed: [ubuntu2] => (item=lily)
changed: [centos3] => (item=lily)
changed: [centos1] => (item=lily)
changed: [ubuntu2] => (item=anwen)
changed: [centos2] => (item=anwen)
changed: [ubuntu1] => (item=anwen)
changed: [centos3] => (item=anwen)
changed: [centos1] => (item=anwen)
changed: [ubuntu3] => (item=james)
changed: [ubuntu3] => (item=hayley)
changed: [ubuntu3] => (item=lily)
changed: [ubuntu3] => (item=anwen)

PLAY RECAP *********************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

4-2. with_dict

user 생성시 코멘트 추가

```
$ cd ../07/
$ cat user_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Creating user
      user:
        name: "{{ item.key }}"
        comment: "{{ item.value.full_name }}"
      with_dict:
        james:
          full_name: James Spurin
        hayley:
          full_name: Hayley Spurin
        lily:
          full_name: Lily Spurin
        anwen:
          full_name: Anwen Spurin

# Three dots indicate the end of a YAML document
...

$ ansible-playbook user_playbook.yaml

PLAY [linux] *******************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [centos1]
ok: [centos2]
ok: [centos3]
ok: [ubuntu2]
ok: [ubuntu1]
ok: [ubuntu3]
```

```
TASK [Creating user] ***********************************************************************************************************************
changed: [centos1] => (item={'key': 'james', 'value': {'full_name': 'James Spurin'}})
changed: [centos2] => (item={'key': 'james', 'value': {'full_name': 'James Spurin'}})
changed: [ubuntu1] => (item={'key': 'james', 'value': {'full_name': 'James Spurin'}})
changed: [ubuntu2] => (item={'key': 'james', 'value': {'full_name': 'James Spurin'}})
changed: [centos3] => (item={'key': 'james', 'value': {'full_name': 'James Spurin'}})
changed: [centos2] => (item={'key': 'hayley', 'value': {'full_name': 'Hayley Spurin'}})
changed: [centos1] => (item={'key': 'hayley', 'value': {'full_name': 'Hayley Spurin'}})
changed: [ubuntu1] => (item={'key': 'hayley', 'value': {'full_name': 'Hayley Spurin'}})
changed: [centos3] => (item={'key': 'hayley', 'value': {'full_name': 'Hayley Spurin'}})
changed: [ubuntu2] => (item={'key': 'hayley', 'value': {'full_name': 'Hayley Spurin'}})
changed: [centos2] => (item={'key': 'lily', 'value': {'full_name': 'Lily Spurin'}})
changed: [centos1] => (item={'key': 'lily', 'value': {'full_name': 'Lily Spurin'}})
changed: [ubuntu1] => (item={'key': 'lily', 'value': {'full_name': 'Lily Spurin'}})
changed: [centos3] => (item={'key': 'lily', 'value': {'full_name': 'Lily Spurin'}})
changed: [ubuntu2] => (item={'key': 'lily', 'value': {'full_name': 'Lily Spurin'}})
changed: [centos2] => (item={'key': 'anwen', 'value': {'full_name': 'Anwen Spurin'}})
changed: [centos1] => (item={'key': 'anwen', 'value': {'full_name': 'Anwen Spurin'}})
changed: [ubuntu1] => (item={'key': 'anwen', 'value': {'full_name': 'Anwen Spurin'}})
changed: [centos3] => (item={'key': 'anwen', 'value': {'full_name': 'Anwen Spurin'}})
changed: [ubuntu2] => (item={'key': 'anwen', 'value': {'full_name': 'Anwen Spurin'}})
changed: [ubuntu3] => (item={'key': 'james', 'value': {'full_name': 'James Spurin'}})
changed: [ubuntu3] => (item={'key': 'hayley', 'value': {'full_name': 'Hayley Spurin'}})
changed: [ubuntu3] => (item={'key': 'lily', 'value': {'full_name': 'Lily Spurin'}})
changed: [ubuntu3] => (item={'key': 'anwen', 'value': {'full_name': 'Anwen Spurin'}})

PLAY RECAP *********************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

$ ssh centos3 tail -5 /etc/passwd
ansible:x:1000:1000::/home/ansible:/bin/bash
james:x:1001:1001:James Spurin:/home/james:/bin/bash
hayley:x:1002:1002:Hayley Spurin:/home/hayley:/bin/bash
lily:x:1003:1003:Lily Spurin:/home/lily:/bin/bash
anwen:x:1004:1004:Anwen Spurin:/home/anwen:/bin/bash

$ cd ../08/
$ cat user_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Removing user
      user:
        name: "{{ item.key }}"
        comment: "{{ item.value.full_name }}"
        state: absent
      with_dict:
        james:
          full_name: James Spurin
        hayley:
          full_name: Hayley Spurin
        lily:
          full_name: Lily Spurin
        anwen:
          full_name: Anwen Spurin

# Three dots indicate the end of a YAML document
...

$ ansible-playbook user_playbook.yaml

PLAY [linux] *******************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [centos2]
ok: [centos3]
ok: [ubuntu1]
ok: [ubuntu2]
ok: [centos1]
ok: [ubuntu3]

TASK [Removing user] ***********************************************************************************************************************
changed: [centos1] => (item={'key': 'james', 'value': {'full_name': 'James Spurin'}})
```

```
changed: [centos3] => (item={'key': 'james', 'value': {'full_name': 'James Spurin'}})
changed: [ubuntu1] => (item={'key': 'james', 'value': {'full_name': 'James Spurin'}})
changed: [centos2] => (item={'key': 'james', 'value': {'full_name': 'James Spurin'}})
changed: [ubuntu2] => (item={'key': 'james', 'value': {'full_name': 'James Spurin'}})
changed: [centos3] => (item={'key': 'hayley', 'value': {'full_name': 'Hayley Spurin'}})
changed: [centos1] => (item={'key': 'hayley', 'value': {'full_name': 'Hayley Spurin'}})
changed: [centos2] => (item={'key': 'hayley', 'value': {'full_name': 'Hayley Spurin'}})
changed: [ubuntu1] => (item={'key': 'hayley', 'value': {'full_name': 'Hayley Spurin'}})
changed: [ubuntu2] => (item={'key': 'hayley', 'value': {'full_name': 'Hayley Spurin'}})
changed: [centos1] => (item={'key': 'lily', 'value': {'full_name': 'Lily Spurin'}})
changed: [centos2] => (item={'key': 'lily', 'value': {'full_name': 'Lily Spurin'}})
changed: [centos3] => (item={'key': 'lily', 'value': {'full_name': 'Lily Spurin'}})
changed: [ubuntu1] => (item={'key': 'lily', 'value': {'full_name': 'Lily Spurin'}})
changed: [ubuntu2] => (item={'key': 'lily', 'value': {'full_name': 'Lily Spurin'}})
changed: [centos1] => (item={'key': 'anwen', 'value': {'full_name': 'Anwen Spurin'}})
changed: [centos2] => (item={'key': 'anwen', 'value': {'full_name': 'Anwen Spurin'}})
changed: [ubuntu1] => (item={'key': 'anwen', 'value': {'full_name': 'Anwen Spurin'}})
changed: [centos3] => (item={'key': 'anwen', 'value': {'full_name': 'Anwen Spurin'}})
changed: [ubuntu2] => (item={'key': 'anwen', 'value': {'full_name': 'Anwen Spurin'}})
changed: [ubuntu3] => (item={'key': 'james', 'value': {'full_name': 'James Spurin'}})
changed: [ubuntu3] => (item={'key': 'hayley', 'value': {'full_name': 'Hayley Spurin'}})
changed: [ubuntu3] => (item={'key': 'lily', 'value': {'full_name': 'Lily Spurin'}})
changed: [ubuntu3] => (item={'key': 'anwen', 'value': {'full_name': 'Anwen Spurin'}})

PLAY RECAP *********************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

4-3. with_subelements

```
$ cd ../09
$ cat user_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Creating user
      user:
        name: "{{ item.1 }}"
        comment: "{{ item.1 | title }} {{ item.0.surname }}"
      with_subelements:
        - family:
            surname: Spurin
            members:
             - james
             - hayley
             - lily
             - anwen
        - members

# Three dots indicate the end of a YAML document
...

$ ansible-playbook user_playbook.yaml

PLAY [linux] *******************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [centos2]
ok: [centos3]
ok: [centos1]
ok: [ubuntu1]
ok: [ubuntu2]
ok: [ubuntu3]

TASK [Creating user] ***********************************************************************************************************************
changed: [centos1] => (item=[{'surname': 'Spurin'}, 'james'])
changed: [ubuntu2] => (item=[{'surname': 'Spurin'}, 'james'])
changed: [centos3] => (item=[{'surname': 'Spurin'}, 'james'])
changed: [ubuntu1] => (item=[{'surname': 'Spurin'}, 'james'])
changed: [centos2] => (item=[{'surname': 'Spurin'}, 'james'])
```

```
changed: [centos1] => (item=[{'surname': 'Spurin'}, 'hayley'])
changed: [ubuntu2] => (item=[{'surname': 'Spurin'}, 'hayley'])
changed: [centos3] => (item=[{'surname': 'Spurin'}, 'hayley'])
changed: [ubuntu1] => (item=[{'surname': 'Spurin'}, 'hayley'])
changed: [centos2] => (item=[{'surname': 'Spurin'}, 'hayley'])
changed: [centos1] => (item=[{'surname': 'Spurin'}, 'lily'])
changed: [ubuntu2] => (item=[{'surname': 'Spurin'}, 'lily'])
changed: [centos3] => (item=[{'surname': 'Spurin'}, 'lily'])
changed: [ubuntu1] => (item=[{'surname': 'Spurin'}, 'lily'])
changed: [centos2] => (item=[{'surname': 'Spurin'}, 'lily'])
changed: [ubuntu2] => (item=[{'surname': 'Spurin'}, 'anwen'])
changed: [centos1] => (item=[{'surname': 'Spurin'}, 'anwen'])
changed: [centos3] => (item=[{'surname': 'Spurin'}, 'anwen'])
changed: [ubuntu1] => (item=[{'surname': 'Spurin'}, 'anwen'])
changed: [centos2] => (item=[{'surname': 'Spurin'}, 'anwen'])
changed: [ubuntu3] => (item=[{'surname': 'Spurin'}, 'james'])
changed: [ubuntu3] => (item=[{'surname': 'Spurin'}, 'hayley'])
changed: [ubuntu3] => (item=[{'surname': 'Spurin'}, 'lily'])
changed: [ubuntu3] => (item=[{'surname': 'Spurin'}, 'anwen'])

PLAY RECAP *********************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

$ ssh centos3 tail -5 /etc/passwd
james:x:1001:1001:James Spurin:/home/james:/bin/bash
hayley:x:1002:1002:Hayley Spurin:/home/hayley:/bin/bash
lily:x:1003:1003:Lily Spurin:/home/lily:/bin/bash
anwen:x:1004:1004:Anwen Spurin:/home/anwen:/bin/bash
```

```
$ cd. ../10
$ cat user_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Creating user
      user:
        name: "{{ item.1 }}"
        comment: "{{ item.1 | title }} {{ item.0.surname }}"
      with_subelements:
        -
          - surname: Spurin
            members:
              - james
              - hayley
              - lily
              - anwen
          - surname: Darlington
            members:
              - freya
          - surname: Jalba
            members:
              - ana
          - surname: Angne
            members:
              - abhishek
          - surname: Mahmood
            members:
              - sara
        - members

# Three dots indicate the end of a YAML document
...

$ ansible-playbook user_playbook.yaml

PLAY [linux] *******************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
```

```
ok: [centos2]
ok: [centos3]
ok: [centos1]
ok: [ubuntu2]
ok: [ubuntu1]
ok: [ubuntu3]

TASK [Creating user] ***********************************************************************************************************************************
ok: [centos1] => (item=[{'surname': 'Spurin'}, 'james'])
ok: [centos3] => (item=[{'surname': 'Spurin'}, 'james'])
ok: [centos2] => (item=[{'surname': 'Spurin'}, 'james'])
ok: [ubuntu1] => (item=[{'surname': 'Spurin'}, 'james'])
ok: [ubuntu2] => (item=[{'surname': 'Spurin'}, 'james'])
ok: [centos1] => (item=[{'surname': 'Spurin'}, 'hayley'])
ok: [centos2] => (item=[{'surname': 'Spurin'}, 'hayley'])
ok: [ubuntu1] => (item=[{'surname': 'Spurin'}, 'hayley'])
ok: [centos3] => (item=[{'surname': 'Spurin'}, 'hayley'])
ok: [ubuntu2] => (item=[{'surname': 'Spurin'}, 'hayley'])
ok: [centos1] => (item=[{'surname': 'Spurin'}, 'lily'])
ok: [ubuntu1] => (item=[{'surname': 'Spurin'}, 'lily'])
ok: [centos2] => (item=[{'surname': 'Spurin'}, 'lily'])
ok: [centos3] => (item=[{'surname': 'Spurin'}, 'lily'])
ok: [ubuntu2] => (item=[{'surname': 'Spurin'}, 'lily'])
ok: [ubuntu1] => (item=[{'surname': 'Spurin'}, 'anwen'])
ok: [centos1] => (item=[{'surname': 'Spurin'}, 'anwen'])
ok: [centos3] => (item=[{'surname': 'Spurin'}, 'anwen'])
ok: [centos2] => (item=[{'surname': 'Spurin'}, 'anwen'])
ok: [ubuntu2] => (item=[{'surname': 'Spurin'}, 'anwen'])
changed: [centos1] => (item=[{'surname': 'Darlington'}, 'freya'])
changed: [ubuntu1] => (item=[{'surname': 'Darlington'}, 'freya'])
changed: [centos3] => (item=[{'surname': 'Darlington'}, 'freya'])
changed: [centos2] => (item=[{'surname': 'Darlington'}, 'freya'])
changed: [ubuntu2] => (item=[{'surname': 'Darlington'}, 'freya'])
changed: [centos1] => (item=[{'surname': 'Jalba'}, 'ana'])
changed: [ubuntu1] => (item=[{'surname': 'Jalba'}, 'ana'])
changed: [centos3] => (item=[{'surname': 'Jalba'}, 'ana'])
changed: [centos2] => (item=[{'surname': 'Jalba'}, 'ana'])
changed: [ubuntu2] => (item=[{'surname': 'Jalba'}, 'ana'])
changed: [centos1] => (item=[{'surname': 'Angne'}, 'abhishek'])
changed: [ubuntu1] => (item=[{'surname': 'Angne'}, 'abhishek'])
changed: [centos2] => (item=[{'surname': 'Angne'}, 'abhishek'])
changed: [centos3] => (item=[{'surname': 'Angne'}, 'abhishek'])
changed: [ubuntu2] => (item=[{'surname': 'Angne'}, 'abhishek'])
changed: [centos1] => (item=[{'surname': 'Mahmood'}, 'sara'])
changed: [ubuntu1] => (item=[{'surname': 'Mahmood'}, 'sara'])
changed: [centos2] => (item=[{'surname': 'Mahmood'}, 'sara'])
changed: [centos3] => (item=[{'surname': 'Mahmood'}, 'sara'])
changed: [ubuntu2] => (item=[{'surname': 'Mahmood'}, 'sara'])
ok: [ubuntu3] => (item=[{'surname': 'Spurin'}, 'james'])
ok: [ubuntu3] => (item=[{'surname': 'Spurin'}, 'hayley'])
ok: [ubuntu3] => (item=[{'surname': 'Spurin'}, 'lily'])
ok: [ubuntu3] => (item=[{'surname': 'Spurin'}, 'anwen'])
changed: [ubuntu3] => (item=[{'surname': 'Darlington'}, 'freya'])
changed: [ubuntu3] => (item=[{'surname': 'Jalba'}, 'ana'])
changed: [ubuntu3] => (item=[{'surname': 'Angne'}, 'abhishek'])
changed: [ubuntu3] => (item=[{'surname': 'Mahmood'}, 'sara'])

PLAY RECAP *********************************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

# 확인
$ ssh centos3 tail -8 /etc/passwd
james:x:1001:1001:James Spurin:/home/james:/bin/bash
hayley:x:1002:1002:Hayley Spurin:/home/hayley:/bin/bash
lily:x:1003:1003:Lily Spurin:/home/lily:/bin/bash
anwen:x:1004:1004:Anwen Spurin:/home/anwen:/bin/bash
freya:x:1005:1005:Freya Darlington:/home/freya:/bin/bash
ana:x:1006:1006:Ana Jalba:/home/ana:/bin/bash
abhishek:x:1007:1007:Abhishek Angne:/home/abhishek:/bin/bash
sara:x:1008:1008:Sara Mahmood:/home/sara:/bin/bash

# 사용자 비밀번호 입력
$ cd ../11
$ cat user_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
```

```
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Creating user
      user:
        name: "{{ item.1 }}"
        comment: "{{ item.1 | title }} {{ item.0.surname }}"
        # https://docs.ansible.com/ansible/latest/plugins/lookup/password.html
        password: "{{ lookup('password', '/dev/null length=15 chars=ascii_letters,digits,hexdigits,punctuation') | password_hash('sha5
      with_subelements:
        -
          - surname: Spurin
            members:
             - james
             - hayley
             - lily
             - anwen
          - surname: Darlington
            members:
             - freya
          - surname: Jalba
            members:
             - ana
          - surname: Angne
            members:
             - abhishek
          - surname: Mahmood
            members:
             - sara
        - members

# Three dots indicate the end of a YAML document
...

$ ansible-playbook user_playbook.yaml

PLAY [linux] ***********************************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************************
ok: [centos2]
ok: [centos3]
ok: [centos1]
ok: [ubuntu1]
ok: [ubuntu2]
ok: [ubuntu3]

TASK [Creating user] ***************************************************************************************************************************************
changed: [centos1] => (item=[{'surname': 'Spurin'}, 'james'])
changed: [centos2] => (item=[{'surname': 'Spurin'}, 'james'])
changed: [centos3] => (item=[{'surname': 'Spurin'}, 'james'])
changed: [ubuntu1] => (item=[{'surname': 'Spurin'}, 'james'])
changed: [ubuntu2] => (item=[{'surname': 'Spurin'}, 'james'])
changed: [centos2] => (item=[{'surname': 'Spurin'}, 'hayley'])
changed: [centos1] => (item=[{'surname': 'Spurin'}, 'hayley'])
changed: [ubuntu2] => (item=[{'surname': 'Spurin'}, 'hayley'])
changed: [centos3] => (item=[{'surname': 'Spurin'}, 'hayley'])
changed: [ubuntu1] => (item=[{'surname': 'Spurin'}, 'hayley'])
changed: [centos1] => (item=[{'surname': 'Spurin'}, 'lily'])
changed: [centos2] => (item=[{'surname': 'Spurin'}, 'lily'])
changed: [ubuntu2] => (item=[{'surname': 'Spurin'}, 'lily'])
changed: [centos3] => (item=[{'surname': 'Spurin'}, 'lily'])
changed: [ubuntu1] => (item=[{'surname': 'Spurin'}, 'lily'])
changed: [centos1] => (item=[{'surname': 'Spurin'}, 'anwen'])
changed: [centos2] => (item=[{'surname': 'Spurin'}, 'anwen'])
changed: [centos3] => (item=[{'surname': 'Spurin'}, 'anwen'])
changed: [ubuntu1] => (item=[{'surname': 'Spurin'}, 'anwen'])
changed: [ubuntu2] => (item=[{'surname': 'Spurin'}, 'anwen'])
changed: [centos1] => (item=[{'surname': 'Darlington'}, 'freya'])
changed: [centos2] => (item=[{'surname': 'Darlington'}, 'freya'])
changed: [centos3] => (item=[{'surname': 'Darlington'}, 'freya'])
changed: [ubuntu1] => (item=[{'surname': 'Darlington'}, 'freya'])
changed: [ubuntu2] => (item=[{'surname': 'Darlington'}, 'freya'])
changed: [centos1] => (item=[{'surname': 'Jalba'}, 'ana'])
changed: [centos2] => (item=[{'surname': 'Jalba'}, 'ana'])
changed: [centos3] => (item=[{'surname': 'Jalba'}, 'ana'])
changed: [ubuntu1] => (item=[{'surname': 'Jalba'}, 'ana'])
changed: [ubuntu2] => (item=[{'surname': 'Jalba'}, 'ana'])
changed: [centos1] => (item=[{'surname': 'Angne'}, 'abhishek'])
changed: [centos2] => (item=[{'surname': 'Angne'}, 'abhishek'])
changed: [centos3] => (item=[{'surname': 'Angne'}, 'abhishek'])
changed: [ubuntu1] => (item=[{'surname': 'Angne'}, 'abhishek'])
changed: [ubuntu2] => (item=[{'surname': 'Angne'}, 'abhishek'])
changed: [centos1] => (item=[{'surname': 'Mahmood'}, 'sara'])
changed: [centos2] => (item=[{'surname': 'Mahmood'}, 'sara'])
changed: [centos3] => (item=[{'surname': 'Mahmood'}, 'sara'])
changed: [ubuntu2] => (item=[{'surname': 'Mahmood'}, 'sara'])
```

```
changed: [ubuntu1] => (item=[{'surname': 'Mahmood'}, 'sara'])
changed: [ubuntu3] => (item=[{'surname': 'Spurin'}, 'james'])
changed: [ubuntu3] => (item=[{'surname': 'Spurin'}, 'hayley'])
changed: [ubuntu3] => (item=[{'surname': 'Spurin'}, 'lily'])
changed: [ubuntu3] => (item=[{'surname': 'Spurin'}, 'anwen'])
changed: [ubuntu3] => (item=[{'surname': 'Darlington'}, 'freya'])
changed: [ubuntu3] => (item=[{'surname': 'Jalba'}, 'ana'])
changed: [ubuntu3] => (item=[{'surname': 'Angne'}, 'abhishek'])
changed: [ubuntu3] => (item=[{'surname': 'Mahmood'}, 'sara'])

PLAY RECAP *********************************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

$ ssh root@centos3 tail -8 /etc/shadow
james:$6$gyinbsG/DTwOj/B5$5mnunqyc7oA5cSJlg3TUbFRGvBT0G5vhYR25VFUJ2UiXUXiUzBCQJ4COByMy83uGEfqcDTO3x2FIHR7csVvAl.:19174:0:99999:7:::
hayley:$6$.OlZ7XJ4TXP53qEB$2mhkXd.BkIF3qsFM4l2l/cH7ytiHbkUZuiZgs7Astq5D/pziWsdigoCcnBajo2p8FIyGGT06N5N4khkyrid0K/:19174:0:99999:7:::
lily:$6$aBYeCrhPQZh7Wfeg$L4DBw/KiBRUvTCDBTQN8i.H2b0pmXZmGIzL3eECmkxzcORTE.ngL6BmcPpeph2rnS4pV3BEIMga9HM2SjYb.s/:19174:0:99999:7:::
anwen:$6$8c0NFjhI3D1BKNTb$GF5hSy4mdgNx7ZB.LOpjeBFSTzG7KF3XYpxiyAEyIUYpWNRQwGbPC5ad2qYJeNPY4vTiPnPRYhBfIA1xLy1RU/:19174:0:99999:7:::
freya:$6$4X0dQzCVnRTrrqPq$BqZbjZo93VuIIuEH3SpGSc4u2cwQAPDWD6yxygtfFc20QjhrNFxB9/hYbgpFO3ZXCyy1dBGA.eoOs98Lm/06b1:19174:0:99999:7:::
ana:$6$jmXSs/ecsfDzhSxg$PdPo7e7khy9SAM.DngGHHYzgpd2QQBv0RLJXD.XxCReVBThtFVS6CWx9nlQ.7vHMgt6idfcjbbxnumEvt9FOq/:19174:0:99999:7:::
abhishek:$6$kO.zv9XjXUL9GJwi$kr/Ft7MCId5DvE2JathS25kI5f4ohNX6MD6boICCIdRzEOmfaC20Kf9d1IqOg.ClTkWlwGkHBLkLYeguPpwUt.:19174:0:99999:7:::
sara:$6$f8mWKREn3pm5G9EU$dwEwYfvyINEoJRYo/Tg6Ufqq/e.5hA9VAybo42irSQhpoed/rJ00hXCaIdZZBKD9yHx42Sb6YjPNNGvgPBiL60:19174:0:99999:7:::
```

create user directory

```
$ cd ../12
$ cat user_directories.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Creating user directories
      file:
        dest: "/home/{{ item.0 }}/{{ item.1 }}"
        owner: "{{ item.0 }}"
        group: "{{ item.0 }}"
        state: directory
      with_nested:
        - [ james, hayley, freya, lily, anwen, ana, abhishek, sara ]
        - [ photos, movies, documents ]

# Three dots indicate the end of a YAML document
...

$ ansible-playbook user_directories.yaml

$ ssh centos3 -l root ls -altr /home/james
total 32
-rw-r--r-- 1 james james  376 Jan 12  2021 .bashrc
-rw-r--r-- 1 james james  141 Jan 12  2021 .bash_profile
-rw-r--r-- 1 james james   18 Jan 12  2021 .bash_logout
drwxr-xr-x 1 root  root  4096 Jul  1 08:15 ..
drwxr-xr-x 2 james james 4096 Jul  1 08:33 photos
drwxr-xr-x 2 james james 4096 Jul  1 08:34 movies
drwxr-xr-x 2 james james 4096 Jul  1 08:34 documents
drwx------ 5 james james 4096 Jul  1 08:34 .
```

4-4. with_together

```
$ cd ../13
$ cat user_directories.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-
```

```
  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Creating user directories
      file:
        dest: "/home/{{ item.0 }}/{{ item.1 }}"
        owner: "{{ item.0 }}"
        group: "{{ item.0 }}"
        state: directory
      with_together:
        - [ james, hayley, freya, lily, anwen, ana, abhishek, sara ]
        - [ tech, psychology, acting, dancing, playing, japanese, coffee, music ]

# Three dots indicate the end of a YAML document
...

$ ansible-playbook user_directories.yaml

PLAY [linux] ***********************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************
ok: [centos1]
ok: [centos2]
ok: [ubuntu2]
ok: [centos3]
ok: [ubuntu1]
ok: [ubuntu3]

TASK [Creating user directories] ***************************************************************************************************************
changed: [centos2] => (item=['james', 'tech'])
changed: [ubuntu2] => (item=['james', 'tech'])
changed: [ubuntu1] => (item=['james', 'tech'])
changed: [centos3] => (item=['james', 'tech'])
changed: [centos1] => (item=['james', 'tech'])
changed: [centos2] => (item=['hayley', 'psychology'])
changed: [ubuntu1] => (item=['hayley', 'psychology'])
changed: [centos1] => (item=['hayley', 'psychology'])
changed: [ubuntu2] => (item=['hayley', 'psychology'])
changed: [centos3] => (item=['hayley', 'psychology'])
changed: [centos2] => (item=['freya', 'acting'])
changed: [ubuntu1] => (item=['freya', 'acting'])
changed: [centos3] => (item=['freya', 'acting'])
changed: [centos1] => (item=['freya', 'acting'])
changed: [ubuntu2] => (item=['freya', 'acting'])
changed: [centos2] => (item=['lily', 'dancing'])
changed: [ubuntu1] => (item=['lily', 'dancing'])
changed: [centos1] => (item=['lily', 'dancing'])
changed: [centos3] => (item=['lily', 'dancing'])
changed: [ubuntu2] => (item=['lily', 'dancing'])
changed: [centos2] => (item=['anwen', 'playing'])
changed: [ubuntu1] => (item=['anwen', 'playing'])
changed: [centos1] => (item=['anwen', 'playing'])
changed: [centos3] => (item=['anwen', 'playing'])
changed: [ubuntu2] => (item=['anwen', 'playing'])
changed: [centos2] => (item=['ana', 'japanese'])
changed: [ubuntu1] => (item=['ana', 'japanese'])
changed: [centos1] => (item=['ana', 'japanese'])
changed: [centos3] => (item=['ana', 'japanese'])
changed: [ubuntu2] => (item=['ana', 'japanese'])
changed: [centos2] => (item=['abhishek', 'coffee'])
changed: [ubuntu1] => (item=['abhishek', 'coffee'])
changed: [centos1] => (item=['abhishek', 'coffee'])
changed: [centos3] => (item=['abhishek', 'coffee'])
changed: [ubuntu2] => (item=['abhishek', 'coffee'])
changed: [centos2] => (item=['sara', 'music'])
changed: [ubuntu1] => (item=['sara', 'music'])
changed: [centos1] => (item=['sara', 'music'])
changed: [centos3] => (item=['sara', 'music'])
changed: [ubuntu2] => (item=['sara', 'music'])
changed: [ubuntu3] => (item=['james', 'tech'])
changed: [ubuntu3] => (item=['hayley', 'psychology'])
changed: [ubuntu3] => (item=['freya', 'acting'])
changed: [ubuntu3] => (item=['lily', 'dancing'])
changed: [ubuntu3] => (item=['anwen', 'playing'])
changed: [ubuntu3] => (item=['ana', 'japanese'])
changed: [ubuntu3] => (item=['abhishek', 'coffee'])
changed: [ubuntu3] => (item=['sara', 'music'])

PLAY RECAP *************************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
ubuntu3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

$ ssh centos3 -l root ls -altr /home/james
total 36
-rw-r--r-- 1 james james  376 Jan 12  2021 .bashrc
-rw-r--r-- 1 james james  141 Jan 12  2021 .bash_profile
-rw-r--r-- 1 james james   18 Jan 12  2021 .bash_logout
drwxr-xr-x 1 root  root  4096 Jul  1 08:15 ..
drwxr-xr-x 2 james james 4096 Jul  1 08:33 photos
drwxr-xr-x 2 james james 4096 Jul  1 08:34 movies
drwxr-xr-x 2 james james 4096 Jul  1 08:34 documents
drwxr-xr-x 2 james james 4096 Jul  1 08:53 tech
drwx------ 6 james james 4096 Jul  1 08:53 .
```

with_file

```
$ cd ../14
$ cat ssh_key_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Create authorized key
      authorized_key:
        user: james
        key: "{{ item }}"
      with_file:
        - /home/ansible/.ssh/id_rsa.pub

# Three dots indicate the end of a YAML document
...

$ ansible-playbook ssh_key_playbook.yaml

PLAY [linux] ***********************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************
ok: [centos1]
ok: [centos3]
ok: [centos2]
ok: [ubuntu1]
ok: [ubuntu2]
ok: [ubuntu3]

TASK [Create authorized key] *******************************************************************************************************************
changed: [centos1] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqW12MwGIUSuS+GfGd9c0hqUwzt/G0MsNqsrq6d0HNzWHBEh4kkTvCTQMAxktYQ+/PRx
changed: [centos3] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqW12MwGIUSuS+GfGd9c0hqUwzt/G0MsNqsrq6d0HNzWHBEh4kkTvCTQMAxktYQ+/PRx
changed: [centos2] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqW12MwGIUSuS+GfGd9c0hqUwzt/G0MsNqsrq6d0HNzWHBEh4kkTvCTQMAxktYQ+/PRx
changed: [ubuntu1] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqW12MwGIUSuS+GfGd9c0hqUwzt/G0MsNqsrq6d0HNzWHBEh4kkTvCTQMAxktYQ+/PRx
changed: [ubuntu2] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqW12MwGIUSuS+GfGd9c0hqUwzt/G0MsNqsrq6d0HNzWHBEh4kkTvCTQMAxktYQ+/PRx
changed: [ubuntu3] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqW12MwGIUSuS+GfGd9c0hqUwzt/G0MsNqsrq6d0HNzWHBEh4kkTvCTQMAxktYQ+/PRx

PLAY RECAP *************************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

# 접속 테스트(without password)
$ ssh centos3 -l james
[james@centos3 ~]$ exit

#########################################################################
#
# multiple keys
#
#########################################################################
$ cd ../15
# generate another key
$ ssh-keygen -f custom_key
$ cat ssh_key_playbook.yaml
---
# YAML documents begin with the document separator ---
```

```
# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Create authorized key
      authorized_key:
        user: james
        key: "{{ item }}"
      with_file:
        - /home/ansible/.ssh/id_rsa.pub
        - custom_key.pub

# Three dots indicate the end of a YAML document
...

$ ansible-playbook ssh_key_playbook.yaml

PLAY [linux] ***********************************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************************
ok: [centos2]
ok: [centos1]
ok: [ubuntu1]
ok: [centos3]
ok: [ubuntu2]
ok: [ubuntu3]

TASK [Create authorized key] *******************************************************************************************************************************
ok: [centos1] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqW12MwGIUSuS+GfGd9c0hqUwzt/G0MsNqsrq6d0HNzWHBEh4kkTvCTQMAxktYQ+/PRxF7BXM
ok: [centos2] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqW12MwGIUSuS+GfGd9c0hqUwzt/G0MsNqsrq6d0HNzWHBEh4kkTvCTQMAxktYQ+/PRxF7BXM
ok: [centos3] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqW12MwGIUSuS+GfGd9c0hqUwzt/G0MsNqsrq6d0HNzWHBEh4kkTvCTQMAxktYQ+/PRxF7BXM
ok: [ubuntu1] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqW12MwGIUSuS+GfGd9c0hqUwzt/G0MsNqsrq6d0HNzWHBEh4kkTvCTQMAxktYQ+/PRxF7BXM
ok: [ubuntu2] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqW12MwGIUSuS+GfGd9c0hqUwzt/G0MsNqsrq6d0HNzWHBEh4kkTvCTQMAxktYQ+/PRxF7BXM
changed: [centos1] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCbm3Qtosc52xxL4S4Z+Mrs670Kd6MJOQt61dtyQCDee2KKB83wpGRkQVkvBVF7jxU4Qa
changed: [centos2] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCbm3Qtosc52xxL4S4Z+Mrs670Kd6MJOQt61dtyQCDee2KKB83wpGRkQVkvBVF7jxU4Qa
changed: [centos3] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCbm3Qtosc52xxL4S4Z+Mrs670Kd6MJOQt61dtyQCDee2KKB83wpGRkQVkvBVF7jxU4Qa
changed: [ubuntu1] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCbm3Qtosc52xxL4S4Z+Mrs670Kd6MJOQt61dtyQCDee2KKB83wpGRkQVkvBVF7jxU4Qa
changed: [ubuntu2] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCbm3Qtosc52xxL4S4Z+Mrs670Kd6MJOQt61dtyQCDee2KKB83wpGRkQVkvBVF7jxU4Qa
ok: [ubuntu3] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCqW12MwGIUSuS+GfGd9c0hqUwzt/G0MsNqsrq6d0HNzWHBEh4kkTvCTQMAxktYQ+/PRxF7BXM
changed: [ubuntu3] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCbm3Qtosc52xxL4S4Z+Mrs670Kd6MJOQt61dtyQCDee2KKB83wpGRkQVkvBVF7jxU4Qa

PLAY RECAP *************************************************************************************************************************************************
centos1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

# ssh 접속 테스트
$ ssh -i custom_key centos3 -l james
[james@centos3 ~]$
```

4-5. with_sequence … many other loops … with_random_choice

```
$ cd ../16
$ cat directory_sequence.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Create sequence directories
      file:
        dest: "/home/james/sequence_{{ item }}"
        state: directory
      with_sequence: start=0 end=100 stride=10

# Three dots indicate the end of a YAML document
...
```

```
$ ansible-playbook directory_sequence.yaml
$ ssh centos3 -l root ls -altrh /home/james
total 88K
-rw-r--r--  1 james james  376 Jan 12  2021 .bashrc
-rw-r--r--  1 james james  141 Jan 12  2021 .bash_profile
-rw-r--r--  1 james james   18 Jan 12  2021 .bash_logout
drwxr-xr-x  1 root  root  4.0K Jul  1 08:15 ..
drwxr-xr-x  2 james james 4.0K Jul  1 08:33 photos
drwxr-xr-x  2 james james 4.0K Jul  1 08:34 movies
drwxr-xr-x  2 james james 4.0K Jul  1 08:34 documents
drwxr-xr-x  2 james james 4.0K Jul  1 08:53 tech
-rw-------  1 james james    5 Jul  1 09:23 .bash_history
drwx------  2 james james 4.0K Jul  1 09:26 .ssh
drwxr-xr-x  2 root  root  4.0K Jul  1 09:30 sequence_0
drwxr-xr-x  2 root  root  4.0K Jul  1 09:30 sequence_10
drwxr-xr-x  2 root  root  4.0K Jul  1 09:30 sequence_20
drwxr-xr-x  2 root  root  4.0K Jul  1 09:30 sequence_30
drwxr-xr-x  2 root  root  4.0K Jul  1 09:30 sequence_40
drwxr-xr-x  2 root  root  4.0K Jul  1 09:30 sequence_50
drwxr-xr-x  2 root  root  4.0K Jul  1 09:30 sequence_60
drwxr-xr-x  2 root  root  4.0K Jul  1 09:30 sequence_70
drwxr-xr-x  2 root  root  4.0K Jul  1 09:30 sequence_80
drwxr-xr-x  2 root  root  4.0K Jul  1 09:30 sequence_90
drwxr-xr-x  2 root  root  4.0K Jul  1 09:30 sequence_100
drwx------ 18 james james 4.0K Jul  1 09:30 .
```

```
$ cd ../17
$ cat directory_sequence.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Create sequence directories
      file:
        dest: "{{ item }}"
        state: directory
      with_sequence: start=0 end=100 stride=10 format=/home/james/sequence_%d

# Three dots indicate the end of a YAML document
...

$ ansible-playbook directory_sequence.yaml
```

```
$ cd ../18
$ cat hex_directory_sequence_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Create hex sequence directories
      file:
        dest: "{{ item }}"
        state: directory
      with_sequence: start=0 end=16 stride=1 format=/home/james/hex_sequence_%x

# Three dots indicate the end of a YAML document
...

$ ansible-playbook hex_directory_sequence_playbook.yaml
```

```
$ cd ../19
$ cat count_directory_sequence_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Create hex sequence directories
      file:
        dest: "{{ item }}"
        state: directory
      with_sequence: count=5 format=/home/james/count_sequence_%x

# Three dots indicate the end of a YAML document
...

$ ansible-playbook count_directory_sequence_playbook.yaml
$ ssh centos3  -l root ls -altrh /home/james
drwxr-xr-x  2 root  root  4.0K Jul  1 09:41 count_sequence_1
drwxr-xr-x  2 root  root  4.0K Jul  1 09:41 count_sequence_2
drwxr-xr-x  2 root  root  4.0K Jul  1 09:41 count_sequence_3
drwxr-xr-x  2 root  root  4.0K Jul  1 09:41 count_sequence_4
drwxr-xr-x  2 root  root  4.0K Jul  1 09:41 count_sequence_5
```

```
$ cd ../20
$ cat random_choice_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Create random directory
      file:
        dest: "/home/james/{{ item }}"
        state: directory
      with_random_choice:
        - "google"
        - "facebook"
        - "microsoft"
        - "apple"

# Three dots indicate the end of a YAML document
...

$ ansible-playbook random_choice_playbook.yaml
$ $ ssh centos3 -l root ls -altrh /home/james
```

4-6. until

```
$ cd ../21
$ cat random.sh
#!/bin/bash
echo $((1 + RANDOM % 10))
$ ./random.sh
4

$ cat until_playbook.yaml
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
```

```
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Run a script until we hit 10
      script: random.sh
      register: result
      retries: 100
      until: result.stdout.find("10") != -1
      # n.b. the default delay is 5 seconds
      delay: 1

# Three dots indicate the end of a YAML document
...

$ ansible-playbook until_playbook.yaml
```

5. Asynchronous, Serial, and Parallel Approaches

Asynchronous, Serial, and Parallel Performance Enhancements for Playbook Execution

5-1. Linear Task

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  # Over Total 30 Seconds
  tasks:
    - name: Task 1
      command: /bin/sleep 5

    - name: Task 2
      command: /bin/sleep 5

    - name: Task 3
      command: /bin/sleep 5

    - name: Task 4
      command: /bin/sleep 5
```

```
    - name: Task 5
      command: /bin/sleep 5

    - name: Task 6
      command: /bin/sleep 5

# Three dots indicate the end of a YAML document
...
```

```
$ time ansible-playbook slow_playbook.yaml

real    1m7.166s
user    0m4.073s
sys     0m3.157s
```

5-2. Improve Linear Task Performance

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Task 1
      command: /bin/sleep 5
      when: ansible_hostname == 'centos1'

    - name: Task 2
      command: /bin/sleep 5
      when: ansible_hostname == 'centos2'

    - name: Task 3
      command: /bin/sleep 5
      when: ansible_hostname == 'centos3'

    - name: Task 4
      command: /bin/sleep 5
      when: ansible_hostname == 'ubuntu1'

    - name: Task 5
      command: /bin/sleep 5
      when: ansible_hostname == 'ubuntu2'

    - name: Task 6
      command: /bin/sleep 5
      when: ansible_hostname == 'ubuntu3'

# Three dots indicate the end of a YAML document
...
```

```
$ time ansible-playbook slow_playbook.yaml

real    0m35.458s
user    0m1.894s
sys     0m1.403s
```

5-2. Further Improve Linear Task Performance

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux
```

```
  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  # async : 10 => wait at least 10 seconds
  # poll : 1 => poll status every 1 seconds
  tasks:
    - name: Task 1
      command: /bin/sleep 5
      when: ansible_hostname == 'centos1'
      async: 10
      poll: 1

    - name: Task 2
      command: /bin/sleep 5
      when: ansible_hostname == 'centos2'
      async: 10
      poll: 1

    - name: Task 3
      command: /bin/sleep 5
      when: ansible_hostname == 'centos3'
      async: 10
      poll: 1

    - name: Task 4
      command: /bin/sleep 5
      when: ansible_hostname == 'ubuntu1'
      async: 10
      poll: 1

    - name: Task 5
      command: /bin/sleep 5
      when: ansible_hostname == 'ubuntu2'
      async: 10
      poll: 1

    - name: Task 6
      command: /bin/sleep 5
      when: ansible_hostname == 'ubuntu3'
      async: 10
      poll: 1

# Three dots indicate the end of a YAML document
...
```

```
$ time ansible-playbook slow.yaml

real    0m41.390s
user    0m2.736s
sys     0m2.333s
```

5-2. Further Improve Linear Task Performance (Fire and Forget)

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Task 1
      command: /bin/sleep 5
      when: ansible_hostname == 'centos1'
      async: 10
      poll: 0

    - name: Task 2
      command: /bin/sleep 5
      when: ansible_hostname == 'centos2'
      async: 10
      poll: 0

    - name: Task 3
      command: /bin/sleep 5
      when: ansible_hostname == 'centos3'
```

```
      async: 10
      poll: 0

    - name: Task 4
      command: /bin/sleep 30
      when: ansible_hostname == 'ubuntu1'
      async: 10
      poll: 0

    - name: Task 5
      command: /bin/sleep 5
      when: ansible_hostname == 'ubuntu2'
      async: 10
      poll: 0

    - name: Task 6
      command: /bin/sleep 5
      when: ansible_hostname == 'ubuntu3'
      async: 10
      poll: 0

# Three dots indicate the end of a YAML document
...
```

```
$ time ansible-playbook slow_playbook.yaml

real    0m4.663s
user    0m1.386s
sys     0m1.333s

# 백그라운드 작업이 있는지 확인
$ ps -ef | grep ssh
```

30초 동안 대기하는 task가 있음에도 전체 수행 시간이 30초가 걸리지 않은 것은 모든 task가 백그라운드로 실행되고 있기 때문이다. 즉, task를 실행만하고 task 종료까지 기다리지 않는다.

5-3. 결괏값 받기

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Task 1
      command: /bin/sleep 5
      when: ansible_hostname == 'centos1'
      async: 10
      poll: 0
      register: result1

    - name: Task 2
      command: /bin/sleep 5
      when: ansible_hostname == 'centos2'
      async: 10
      poll: 0
      register: result2

    - name: Task 3
      command: /bin/sleep 5
      when: ansible_hostname == 'centos3'
      async: 10
      poll: 0
      register: result3

    - name: Task 4
      command: /bin/sleep 30
      when: ansible_hostname == 'ubuntu1'
      async: 60
      poll: 0
      register: result4

    - name: Task 5
```

```
      command: /bin/sleep 5
      when: ansible_hostname == 'ubuntu2'
      async: 10
      poll: 0
      register: result5

    - name: Task 6
      command: /bin/sleep 5
      when: ansible_hostname == 'ubuntu3'
      async: 10
      poll: 0
      register: result6

    - name: Show registered context
      debug:
        var: result1

    - name: Show registered context as jinja2
      debug:
        msg: "{{ result1 }}"

# Three dots indicate the end of a YAML document
...
```

```
$ ansible-playbook slow_playbook.yaml
TASK [Show registered context] ********************************************************************************************************
ok: [centos1] => {
    "result1": {
        "ansible_job_id": "271988051557.2734",
        "changed": true,
        "failed": 0,
        "finished": 0,
        "results_file": "/root/.ansible_async/271988051557.2734",
        "started": 1
    }
}
ok: [centos2] => {
    "result1": {
        "changed": false,
        "skip_reason": "Conditional result was False",
        "skipped": true
    }
}
ok: [centos3] => {
    "result1": {
        "changed": false,
        "skip_reason": "Conditional result was False",
        "skipped": true
    }
}
ok: [ubuntu1] => {
    "result1": {
        "changed": false,
        "skip_reason": "Conditional result was False",
        "skipped": true
    }
}
ok: [ubuntu2] => {
    "result1": {
        "changed": false,
        "skip_reason": "Conditional result was False",
        "skipped": true
    }
}
ok: [ubuntu3] => {
    "result1": {
        "changed": false,
        "skip_reason": "Conditional result was False",
        "skipped": true
    }
}

TASK [Show registered context as jinja2] **********************************************************************************************
ok: [centos1] => {
    "msg": {
        "ansible_job_id": "271988051557.2734",
        "changed": true,
        "failed": 0,
        "finished": 0,
        "results_file": "/root/.ansible_async/271988051557.2734",
        "started": 1
    }
}
ok: [centos2] => {
```

```
    "msg": {
        "changed": false,
        "skip_reason": "Conditional result was False",
        "skipped": true
    }
}
ok: [centos3] => {
    "msg": {
        "changed": false,
        "skip_reason": "Conditional result was False",
        "skipped": true
    }
}
ok: [ubuntu1] => {
    "msg": {
        "changed": false,
        "skip_reason": "Conditional result was False",
        "skipped": true
    }
}
ok: [ubuntu2] => {
    "msg": {
        "changed": false,
        "skip_reason": "Conditional result was False",
        "skipped": true
    }
}
ok: [ubuntu3] => {
    "msg": {
        "changed": false,
        "skip_reason": "Conditional result was False",
        "skipped": true
    }
}
```

5-4. Caputer Job ID

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Vars: variables that will apply to the play, on all target systems
  vars:
    jobids: []

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Task 1
      command: /bin/sleep 5
      when: ansible_hostname == 'centos1'
      async: 10
      poll: 0
      register: result1

    - name: Task 2
      command: /bin/sleep 5
      when: ansible_hostname == 'centos2'
      async: 10
      poll: 0
      register: result2

    - name: Task 3
      command: /bin/sleep 5
      when: ansible_hostname == 'centos3'
      async: 10
      poll: 0
      register: result3

    - name: Task 4
      command: /bin/sleep 30
      when: ansible_hostname == 'ubuntu1'
      async: 60
      poll: 0
      register: result4

    - name: Task 5
```

```
      command: /bin/sleep 5
      when: ansible_hostname == 'ubuntu2'
      async: 10
      poll: 0
      register: result5

    - name: Task 6
      command: /bin/sleep 5
      when: ansible_hostname == 'ubuntu3'
      async: 10
      poll: 0
      register: result6

    - name: Capture Job IDs
      set_fact:
        jobids: >
                {% if item.ansible_job_id is defined -%}
                  {{ jobids + [item.ansible_job_id] }}
                {% else -%}
                  {{ jobids }}
                {% endif %}
      with_items: "{{ [ result1, result2, result3, result4, result5, result6 ] }}"

    - name: Show Job IDs
      debug:
        var: jobids

# Three dots indicate the end of a YAML document
...
```

```
$ ansible-playbook slow_playbook.yaml
TASK [Show Job IDs] ************************************************************************************************************************************
ok: [centos1] => {
    "jobids": [
        "579226060160.2816"
    ]
}
ok: [centos2] => {
    "jobids": [
        "656414258913.2816"
    ]
}
ok: [centos3] => {
    "jobids": [
        "216438455124.2918"
    ]
}
ok: [ubuntu1] => {
    "jobids": [
        "477515077495.3535"
    ]
}
ok: [ubuntu2] => {
    "jobids": [
        "498437601132.3535"
    ]
}
ok: [ubuntu3] => {
    "jobids": [
        "162322874555.3535"
    ]
}
```

5-5. 모든 프로세스가 종료될때까지 기다리기

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Vars: variables that will apply to the play, on all target systems
  vars:
    jobids: []

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
```

```
  tasks:
    - name: Task 1
      command: /bin/sleep 5
      when: ansible_hostname == 'centos1'
      async: 10
      poll: 0
      register: result1

    - name: Task 2
      command: /bin/sleep 5
      when: ansible_hostname == 'centos2'
      async: 10
      poll: 0
      register: result2

    - name: Task 3
      command: /bin/sleep 5
      when: ansible_hostname == 'centos3'
      async: 10
      poll: 0
      register: result3

    - name: Task 4
      command: /bin/sleep 30
      when: ansible_hostname == 'ubuntu1'
      async: 60
      poll: 0
      register: result4

    - name: Task 5
      command: /bin/sleep 5
      when: ansible_hostname == 'ubuntu2'
      async: 10
      poll: 0
      register: result5

    - name: Task 6
      command: /bin/sleep 5
      when: ansible_hostname == 'ubuntu3'
      async: 10
      poll: 0
      register: result6

    - name: Capture Job IDs
      set_fact:
        jobids: >
                {% if item.ansible_job_id is defined -%}
                  {{ jobids + [item.ansible_job_id] }}
                {% else -%}
                  {{ jobids }}
                {% endif %}
      with_items: "{{ [ result1, result2, result3, result4, result5, result6 ] }}"

    - name: Show Job IDs
      debug:
        var: jobids

    - name: 'Wait for Job IDs'
      async_status:
         jid: "{{ item }}"
      with_items: "{{ jobids }}"
      register: jobs_result
      until: jobs_result.finished
      retries: 30

# Three dots indicate the end of a YAML document
...
```

```
$ ansible-playbook slow_playbook.yaml

TASK [Wait for Job IDs] ********************************************************************************************************************************
FAILED - RETRYING: [centos1]: Wait for Job IDs (30 retries left).
FAILED - RETRYING: [ubuntu1]: Wait for Job IDs (30 retries left).
FAILED - RETRYING: [ubuntu2]: Wait for Job IDs (30 retries left).
FAILED - RETRYING: [centos2]: Wait for Job IDs (30 retries left).
FAILED - RETRYING: [centos3]: Wait for Job IDs (30 retries left).
changed: [centos1] => (item=28115682377.2898)
FAILED - RETRYING: [ubuntu1]: Wait for Job IDs (29 retries left).
changed: [centos2] => (item=541260706108.2898)
changed: [centos3] => (item=710184711615.3000)
changed: [ubuntu2] => (item=636546076142.3621)
changed: [ubuntu3] => (item=242272575222.3621)
FAILED - RETRYING: [ubuntu1]: Wait for Job IDs (28 retries left).
FAILED - RETRYING: [ubuntu1]: Wait for Job IDs (27 retries left).
```

```
FAILED - RETRYING: [ubuntu1]: Wait for Job IDs (26 retries left).
FAILED - RETRYING: [ubuntu1]: Wait for Job IDs (25 retries left).
changed: [ubuntu1] => (item=265810576809.3621)
PLAY RECAP *********************************************************************************************************************************
centos1                    : ok=5    changed=2    unreachable=0    failed=0    skipped=5    rescued=0    ignored=0
centos2                    : ok=5    changed=2    unreachable=0    failed=0    skipped=5    rescued=0    ignored=0
centos3                    : ok=5    changed=2    unreachable=0    failed=0    skipped=5    rescued=0    ignored=0
ubuntu1                    : ok=5    changed=2    unreachable=0    failed=0    skipped=5    rescued=0    ignored=0
ubuntu2                    : ok=5    changed=2    unreachable=0    failed=0    skipped=5    rescued=0    ignored=0
ubuntu3                    : ok=5    changed=2    unreachable=0    failed=0    skipped=5    rescued=0    ignored=0


real    0m36.834s
user    0m2.794s
sys     0m2.408s
```

5-6. 모든 호스트에서 프로세스가 종료될때까지 기다리기

tasks에서 when 파라미터 제거된 상태

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Vars: variables that will apply to the play, on all target systems
  vars:
    jobids: []

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Task 1
      command: /bin/sleep 5
      async: 10
      poll: 0
      register: result1

    - name: Task 2
      command: /bin/sleep 5
      async: 10
      poll: 0
      register: result2

    - name: Task 3
      command: /bin/sleep 5
      async: 10
      poll: 0
      register: result3

    - name: Task 4
      command: /bin/sleep 30
      async: 60
      poll: 0
      register: result4

    - name: Task 5
      command: /bin/sleep 5
      async: 10
      poll: 0
      register: result5

    - name: Task 6
      command: /bin/sleep 5
      async: 10
      poll: 0
      register: result6

    - name: Capture Job IDs
      set_fact:
        jobids: >
                {% if item.ansible_job_id is defined -%}
                  {{ jobids + [item.ansible_job_id] }}
                {% else -%}
                  {{ jobids }}
                {% endif %}
      with_items: "{{ [ result1, result2, result3, result4, result5, result6 ] }}"
```

```
    - name: Show Job IDs
      debug:
        var: jobids

    - name: 'Wait for Job IDs'
      async_status:
         jid: "{{ item }}"
      with_items: "{{ jobids }}"
      register: jobs_result
      until: jobs_result.finished
      retries: 30

# Three dots indicate the end of a YAML document
...
```

```
$ time ansible-playbook slow_playbook.yaml
TASK [Wait for Job IDs] ********************************************************************************************************************************
changed: [ubuntu1] => (item=837390999361.4356)
changed: [centos3] => (item=77244715408.3469)
changed: [centos1] => (item=417933949514.3367)
changed: [ubuntu2] => (item=622501484699.4236)
changed: [centos2] => (item=651916695919.3383)
FAILED - RETRYING: [centos1]: Wait for Job IDs (30 retries left).
FAILED - RETRYING: [ubuntu1]: Wait for Job IDs (30 retries left).
FAILED - RETRYING: [centos2]: Wait for Job IDs (30 retries left).
FAILED - RETRYING: [centos3]: Wait for Job IDs (30 retries left).
FAILED - RETRYING: [ubuntu2]: Wait for Job IDs (30 retries left).
changed: [centos1] => (item=75520434025.3387)
changed: [centos3] => (item=310718158809.3489)
changed: [ubuntu1] => (item=438622398659.4382)
changed: [ubuntu2] => (item=364754420160.4262)
changed: [centos2] => (item=221120668036.3403)
changed: [centos1] => (item=188421221034.3407)
changed: [ubuntu1] => (item=17707337556.4408)
changed: [centos3] => (item=549513690747.3509)
changed: [ubuntu2] => (item=373043238078.4288)
changed: [centos2] => (item=666954980274.3423)
FAILED - RETRYING: [ubuntu1]: Wait for Job IDs (30 retries left).
FAILED - RETRYING: [centos1]: Wait for Job IDs (30 retries left).
FAILED - RETRYING: [ubuntu2]: Wait for Job IDs (30 retries left).
FAILED - RETRYING: [centos3]: Wait for Job IDs (30 retries left).
FAILED - RETRYING: [centos2]: Wait for Job IDs (30 retries left).
FAILED - RETRYING: [ubuntu1]: Wait for Job IDs (29 retries left).
FAILED - RETRYING: [centos1]: Wait for Job IDs (29 retries left).
FAILED - RETRYING: [ubuntu2]: Wait for Job IDs (29 retries left).
FAILED - RETRYING: [centos2]: Wait for Job IDs (29 retries left).
FAILED - RETRYING: [centos3]: Wait for Job IDs (29 retries left).
FAILED - RETRYING: [centos1]: Wait for Job IDs (28 retries left).
FAILED - RETRYING: [ubuntu1]: Wait for Job IDs (28 retries left).
FAILED - RETRYING: [centos2]: Wait for Job IDs (28 retries left).
FAILED - RETRYING: [ubuntu2]: Wait for Job IDs (28 retries left).
FAILED - RETRYING: [centos3]: Wait for Job IDs (28 retries left).
FAILED - RETRYING: [centos1]: Wait for Job IDs (27 retries left).
FAILED - RETRYING: [ubuntu1]: Wait for Job IDs (27 retries left).
FAILED - RETRYING: [ubuntu2]: Wait for Job IDs (27 retries left).
FAILED - RETRYING: [centos2]: Wait for Job IDs (27 retries left).
FAILED - RETRYING: [centos3]: Wait for Job IDs (27 retries left).
changed: [centos1] => (item=395754096840.3427)
changed: [ubuntu1] => (item=88236858064.4434)
changed: [ubuntu2] => (item=502355462905.4314)
changed: [centos2] => (item=105484992880.3443)
changed: [centos3] => (item=811065360901.3529)
changed: [centos1] => (item=573312451439.3447)
changed: [centos2] => (item=560279781250.3463)
changed: [ubuntu1] => (item=329701604629.4460)
changed: [ubuntu2] => (item=333617664025.4340)
changed: [centos3] => (item=866953301170.3549)
changed: [centos1] => (item=623071418062.3467)
changed: [ubuntu1] => (item=746923610112.4486)
changed: [ubuntu2] => (item=333228827443.4366)
changed: [centos3] => (item=725385647546.3569)
changed: [centos2] => (item=470532173600.3483)
changed: [ubuntu3] => (item=707060316768.4092)
changed: [ubuntu3] => (item=65904580740.4118)
changed: [ubuntu3] => (item=679234802446.4144)
changed: [ubuntu3] => (item=375781167583.4170)
changed: [ubuntu3] => (item=644137489573.4196)
changed: [ubuntu3] => (item=832651670204.4222)

PLAY RECAP *********************************************************************************************************************************************
centos1                    : ok=10   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=10   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=10   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=10   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
ubuntu2                    : ok=10   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=10   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0


real    0m38.198s
user    0m5.155s
sys     0m6.592s
```

5-7. Ansible Execution Default Strategy - Linear Execution

앤서블의 실행은 기본적으로 선형적으로 수행되며 타스크를 하나씩 순차적으로 실행한다. 따라서, 아래의 플레이북의 실행 시간은 (호스트수 * 5초 * 타스크수 = 6 * 5 * 6 = 90 seconds) 90초 걸릴것으로 예상된다.

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux
  gather_facts: false

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Task 1
      command: /bin/sleep 5

    - name: Task 2
      command: /bin/sleep 5

    - name: Task 3
      command: /bin/sleep 5

    - name: Task 4
      command: /bin/sleep 5

    - name: Task 5
      command: /bin/sleep 5

    - name: Task 6
      command: /bin/sleep 5

# Three dots indicate the end of a YAML document
...
```

```
$ ansible-playbook slow_playbook.yaml

real    1m4.444s
user    0m3.679s
sys     0m2.464s
```

하지만, 예상과는 달리 하나의 타스크당 대략 10초 정도 소요되었다. 이는 각 타스크별로 프로세스가 포크되어 5개씩 수행되고 있기 때문이다.

Ansible Default Forks = 5

We recently starting using Ansible to help perform software upgrades on the large number of Juniper EX-4300 and EX-2300 switches in our environment. Like the vast majority of organizations our downtime windows are extremely short and unfortunately the element of human error is usually greater than the

https://blog.michaelfmcnamara.com/2022/04/ansible-default-forks-5/

5-8. Task Fork 조정

```
[defaults]
inventory = hosts
host_key_checking = False
forks=6
```

Ansible Playbooks, Deep Dive/Asynchronous, Serial, Parallel/10/ansible.cfg

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux
  gather_facts: false

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Task 1
      command: /bin/sleep 5

    - name: Task 2
      command: /bin/sleep 5

    - name: Task 3
      command: /bin/sleep 5

    - name: Task 4
      command: /bin/sleep 5

    - name: Task 5
      command: /bin/sleep 5

    - name: Task 6
      command: /bin/sleep 5

# Three dots indicate the end of a YAML document
...
```

```
$ time ansible-playbook slow_playbook.yaml

real    0m33.427s
user    0m1.885s
sys     0m2.149s
```

5-9. Task Batch #1

serial 파라미터를 수정하면 수행되는 타스크의 배치 사이즈를 조정할 수 있다. 여기서는 2로 정하여 두개로 나누어 수행되도록 한다.

```
[defaults]
inventory = hosts
host_key_checking = False
forks=6
```

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux
  gather_facts: false
  serial: 2

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Task 1
      command: /bin/sleep 1

    - name: Task 2
      command: /bin/sleep 1

    - name: Task 3
      command: /bin/sleep 1

    - name: Task 4
      command: /bin/sleep 1
```

```
    - name: Task 5
      command: /bin/sleep 1

    - name: Task 6
      command: /bin/sleep 1

# Three dots indicate the end of a YAML document
...
```

```
$ ansible-playbook serial_playbook.yaml
```

5-10. Task Batch #2

```
[defaults]
inventory = hosts
host_key_checking = False
forks=6
```

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux
  gather_facts: false
  serial:
    - 1
    - 2
    - 3

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Task 1
      command: /bin/sleep 1

    - name: Task 2
      command: /bin/sleep 1

    - name: Task 3
      command: /bin/sleep 1

    - name: Task 4
      command: /bin/sleep 1

    - name: Task 5
      command: /bin/sleep 1

    - name: Task 6
      command: /bin/sleep 1

# Three dots indicate the end of a YAML document
...
```

5-11. Task Batch #3

```
[defaults]
inventory = hosts
host_key_checking = False
forks=6
```

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
```

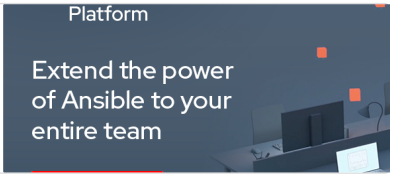
```
  hosts: linux
  gather_facts: false
  serial:
    - 16%
    - 34%
    - 50%

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Task 1
      command: /bin/sleep 1

    - name: Task 2
      command: /bin/sleep 1

    - name: Task 3
      command: /bin/sleep 1

    - name: Task 4
      command: /bin/sleep 1

    - name: Task 5
      command: /bin/sleep 1

    - name: Task 6
      command: /bin/sleep 1

# Three dots indicate the end of a YAML document
...
```

5-12. Free Strategy

ansible.builtin.free strategy - Executes tasks without waiting for all hosts - Ansible Documentation

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

https://docs.ansible.com/ansible/2.9/plugins/strategy/free.html



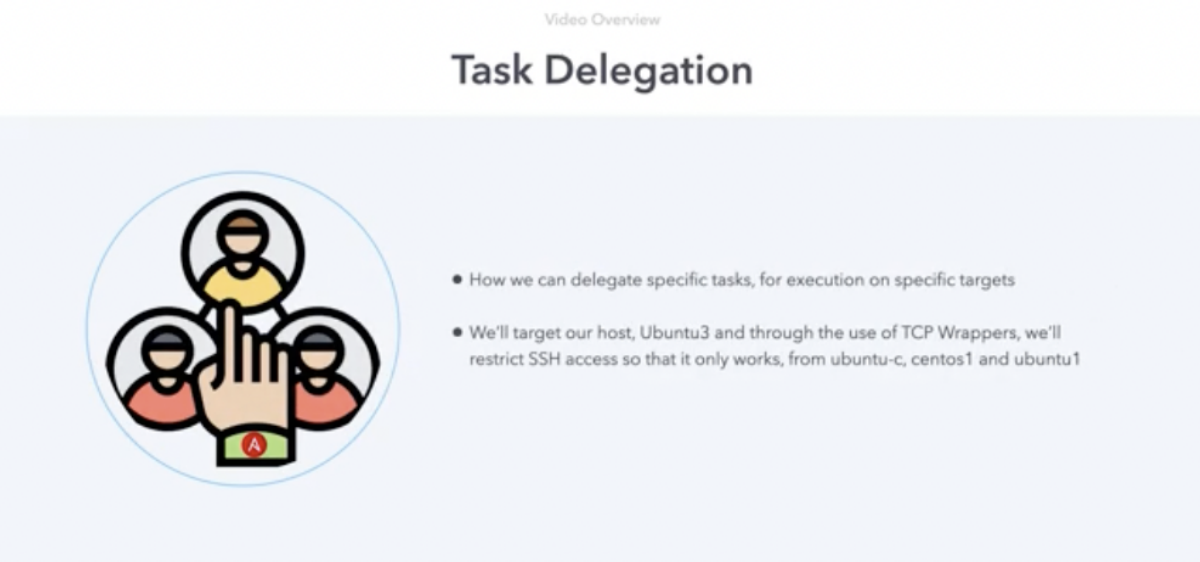

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux
  gather_facts: false

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Task 1
      command: "/bin/sleep {{ 10 |random}}"

    - name: Task 2
      command: "/bin/sleep {{ 10 |random}}"

    - name: Task 3
      command: "/bin/sleep {{ 10 |random}}"

    - name: Task 4
      command: "/bin/sleep {{ 10 |random}}"

    - name: Task 5
      command: "/bin/sleep {{ 10 |random}}"

    - name: Task 6
      command: "/bin/sleep {{ 10 |random}}"

# Three dots indicate the end of a YAML document
...
```

```
---
# YAML documents begin with the document separator ---
```

```
# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux
  gather_facts: false
  strategy: free

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Task 1
      command: "/bin/sleep {{ 10 |random}}"

    - name: Task 2
      command: "/bin/sleep {{ 10 |random}}"

    - name: Task 3
      command: "/bin/sleep {{ 10 |random}}"

    - name: Task 4
      command: "/bin/sleep {{ 10 |random}}"

    - name: Task 5
      command: "/bin/sleep {{ 10 |random}}"

    - name: Task 6
      command: "/bin/sleep {{ 10 |random}}"

# Three dots indicate the end of a YAML document
...
```

6. Task Delegation

Delegation of Tasks to Specific Target for Execution



6-1. Public 키 배포

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c
  gather_facts: False
```

```
  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Generate an OpenSSH keypair for ubuntu3
      openssh_keypair:
        path: ~/.ssh/ubuntu3_id_rsa

-

  # Hosts: where our play will run and options it will run with
  hosts: linux
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Copy ubuntu3 OpenSSH keypair with permissions
      copy:
        owner: root
        src: "{{ item.0 }}"
        dest: "{{ item.0 }}"
        mode: "{{ item.1 }}"
      with_together:
        - [ ~/.ssh/ubuntu3_id_rsa, ~/.ssh/ubuntu3_id_rsa.pub ]
        - [ "0600", "0644" ]

-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu3
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Add public key to the ubuntu3 authorized_keys file
      authorized_key:
        user: root
        state: present
        key: "{{ lookup('file', '~/.ssh/ubuntu3_id_rsa.pub') }}"

# Three dots indicate the end of a YAML document
...
```

6-2. SSH 접속 테스트

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Generate an OpenSSH keypair for ubuntu3
      openssh_keypair:
        path: ~/.ssh/ubuntu3_id_rsa

-

  # Hosts: where our play will run and options it will run with
  hosts: linux
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Copy ubuntu3 OpenSSH keypair with permissions
      copy:
        owner: root
        src: "{{ item.0 }}"
        dest: "{{ item.0 }}"
        mode: "{{ item.1 }}"
      with_together:
        - [ ~/.ssh/ubuntu3_id_rsa, ~/.ssh/ubuntu3_id_rsa.pub ]
        - [ "0600", "0644" ]
```

```
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu3
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Add public key to the ubuntu3 authorized_keys file
      authorized_key:
        user: root
        state: present
        key: "{{ lookup('file', '~/.ssh/ubuntu3_id_rsa.pub') }}"

-

  # Hosts: where our play will run and options it will run with
  hosts: all
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Check that ssh can connect to ubuntu3 using the ssh tool
      command: ssh -i ~/.ssh/ubuntu3_id_rsa -o BatchMode=yes -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null root@ubuntu3
      changed_when: False
      ignore_errors: True

# Three dots indicate the end of a YAML document
...
```

SSH 옵션 설명

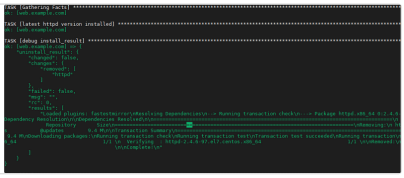
미친푸우 • 2018. 11. 26. 14:02 이것 이후의 (다음의 Host키워드가 나타날 때까지의) 설정 항목을, 여기서 지정된 패턴의 어떤 것인가에 매치 하는 호스트인 만큼 제한합니다. 패턴중에서는 '*' (와)과 '?' 하지만 와일드 카드로서 사용할 수 있습니다. 단독의 '*' (은)는, 모든 호스트에 대한 디폴트로서 사용할 수 있습니다.

https://blog.naver.com/PostView.naver?blogId=kkh0879&logNo=221406404173&parentCategoryNo=1&categoryNo=&viewDate=&isShowPopularPosts=false&from=postList

13장. 작업 제어 구현 - 오류처리

ansible에서는 각 task의 return code를 평가하여 task의 성공 여부를 판단한다. 일반적으로 task에서 하나가 실패하는 즉시 ansible은 해당 호스트의 나머지 play를 중단하고 종료된다. 하지만 작업이 실패한다 하더라도 play를 계속할 수 있어야 한다. 예를들어 특정 작업이 실패할 것으로 예상하고 몇가지 다른 작업을 조건부로 실행하여 복구하려고 할

https://watch-n-learn.tistory.com/86



6-3. delegate_to 를 사용하여 원격호스트에서 task 수행

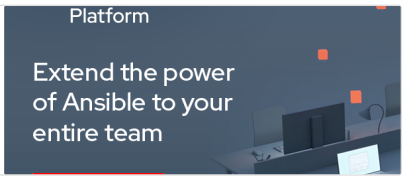
```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Generate an OpenSSH keypair for ubuntu3
      openssh_keypair:
        path: ~/.ssh/ubuntu3_id_rsa

-

  # Hosts: where our play will run and options it will run with
  hosts: linux
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Copy ubuntu3 OpenSSH keypair with permissions
      copy:
        owner: root
```

```
        src: "{{ item.0 }}"
        dest: "{{ item.0 }}"
        mode: "{{ item.1 }}"
      with_together:
        - [ ~/.ssh/ubuntu3_id_rsa, ~/.ssh/ubuntu3_id_rsa.pub ]
        - [ "0600", "0644" ]

-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu3
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Add public key to the ubuntu3 authorized_keys file
      authorized_key:
        user: root
        state: present
        key: "{{ lookup('file', '~/.ssh/ubuntu3_id_rsa.pub') }}"

-

  # Hosts: where our play will run and options it will run with
  hosts: all
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Check that ssh can connect to ubuntu3 using the ssh tool
      command: ssh -i ~/.ssh/ubuntu3_id_rsa -o BatchMode=yes -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null root@ubuntu3
      changed_when: False
      ignore_errors: True

-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c, centos1, ubuntu1
  # Serial is important as we are writing to a single file
  serial: 1

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Add host to /etc/hosts.allow for sshd
      lineinfile:
        path: /etc/hosts.allow
        line: "sshd: {{ ansible_hostname }}.diveinto.io"
        create: True
      delegate_to: ubuntu3

-

  # Hosts: where our play will run and options it will run with
  hosts: all
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Check that ssh can connect to ubuntu3 using the ssh tool
      command: ssh -i ~/.ssh/ubuntu3_id_rsa -o BatchMode=yes -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null root@ubuntu3
      changed_when: False
      ignore_errors: True

# Three dots indicate the end of a YAML document
...
```

ansible.builtin.lineinfile module - Manage lines in text files - 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 lineinfile 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

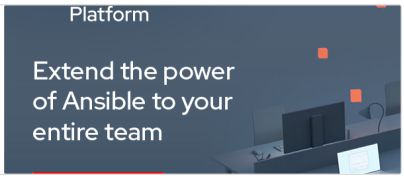
https://docs.ansible.com/ansible/latest/collections/ansible/builtin/lineinfile_module.html



Controlling where tasks run: delegation and local actions - Ansible Documentation

Controlling where tasks run: delegation and local actions

https://docs.ansible.com/ansible/latest/user_guide/playbooks_delegation.html#delegating-tasks



Linux의 TCP 래퍼(/etc/hosts.allow 및 /etc/hosts.deny) 이해 - GeekPeach.net

TCP 래퍼는 들어오는 네트워크 트래픽의 기본 트래픽 필터링을 제공합니다. 액세스 wrapped 다른 시스템의 Linux 서버에서 실행되는 네트워크 서비스를 허용하거나 거부할 수 있습니다. TCP 래핑된 서비스는 libwrap.a 라이브러리에 대해 컴파일된 서비스입니다. ldd 명령을 사용하여 네트워크 서비스가 libwrap.a에 연결되어 있는지 확인합니다. 다음 예에서는 sshd 서비스의 절대 경로 이름을 확인한 다음 grep 명령을 사용하여 libwrap 라이브러리를 검색하여 sshd [...]

https://geekpeach.net/ko/linux%EC%9D%98-tcp-%EB%9E%98%ED%8D%BC-etc-hosts-allow-%EB%B0%8F-etc-hosts-deny-%EC%9D%B4%ED%95%B4

delegate_to: ubuntu3 를 통해 hosts 에 지정된 호스트(ubuntu-c, centos1, ubuntu1)의 tcp 래퍼가 추가된다.

```
ansible@ubuntu3:~$ cat /etc/hosts.allow
# /etc/hosts.allow: list of hosts that are allowed to access the system.
#                   See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:    ALL: LOCAL @some_netgroup
#             ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#

sshd: ubuntu-c.diveinto.io
sshd: centos1.diveinto.io
sshd: ubuntu1.diveinto.io
```

6-4. sshd TCP deny 래퍼 추가

모든 호스트에서의 ssh 연결을 차단. 단, delegate_to 를 통해 추가된 호스트의 연결운 허가

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Generate an OpenSSH keypair for ubuntu3
      openssh_keypair:
        path: ~/.ssh/ubuntu3_id_rsa

-

  # Hosts: where our play will run and options it will run with
  hosts: linux
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Copy ubuntu3 OpenSSH keypair with permissions
      copy:
        owner: root
        src: "{{ item.0 }}"
        dest: "{{ item.0 }}"
        mode: "{{ item.1 }}"
      with_together:
        - [ ~/.ssh/ubuntu3_id_rsa, ~/.ssh/ubuntu3_id_rsa.pub ]
        - [ "0600", "0644" ]

-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu3
  gather_facts: False
```

```
  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Add public key to the ubuntu3 authorized_keys file
      authorized_key:
        user: root
        state: present
        key: "{{ lookup('file', '~/.ssh/ubuntu3_id_rsa.pub') }}"

-

  # Hosts: where our play will run and options it will run with
  hosts: all
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Check that ssh can connect to ubuntu3 using the ssh tool
      command: ssh -i ~/.ssh/ubuntu3_id_rsa -o BatchMode=yes -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null root@ubuntu3
      changed_when: False
      ignore_errors: True

-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c, centos1, ubuntu1
  # Serial is important as we are writing to a single file
  serial: 1

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Add host to /etc/hosts.allow for sshd
      lineinfile:
        path: /etc/hosts.allow
        line: "sshd: {{ ansible_hostname }}.diveinto.io"
        create: True
      delegate_to: ubuntu3

-

  # Hosts: where our play will run and options it will run with
  hosts: all
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Check that ssh can connect to ubuntu3 using the ssh tool
      command: ssh -i ~/.ssh/ubuntu3_id_rsa -o BatchMode=yes -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null root@ubuntu3
      changed_when: False
      ignore_errors: True

-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu3
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Drop SSH connectivity from everywhere else
      lineinfile:
        path: /etc/hosts.deny
        line: "sshd: ALL"
        create: True

-

  # Hosts: where our play will run and options it will run with
  hosts: all
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Check that ssh can connect to ubuntu3 using the ssh tool
      command: ssh -i ~/.ssh/ubuntu3_id_rsa -o BatchMode=yes -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null root@ubuntu3
      changed_when: False
      ignore_errors: True

# Three dots indicate the end of a YAML document
...
```

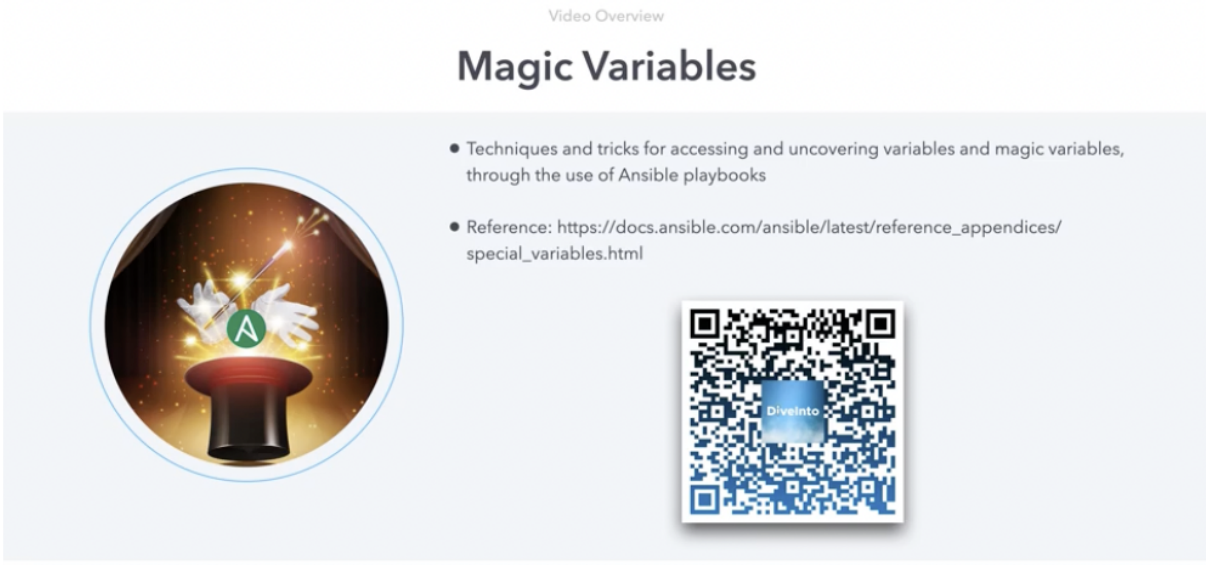
6-5. hosts.allow 에 추가된 호스트 제거하기

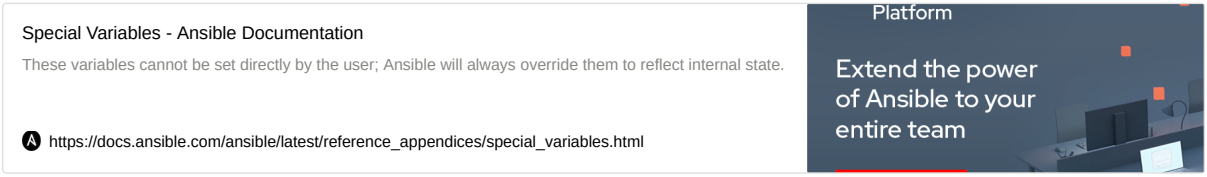
```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Generate an OpenSSH keypair for ubuntu3
      openssh_keypair:
        path: ~/.ssh/ubuntu3_id_rsa

-

  # Hosts: where our play will run and options it will run with
  hosts: linux
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Copy ubuntu3 OpenSSH keypair with permissions
      copy:
        owner: root
        src: "{{ item.0 }}"
        dest: "{{ item.0 }}"
        mode: "{{ item.1 }}"
      with_together:
        - [ ~/.ssh/ubuntu3_id_rsa, ~/.ssh/ubuntu3_id_rsa.pub ]
        - [ "0600", "0644" ]

-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu3
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Add public key to the ubuntu3 authorized_keys file
      authorized_key:
        user: root
        state: present
        key: "{{ lookup('file', '~/.ssh/ubuntu3_id_rsa.pub') }}"

-

  # Hosts: where our play will run and options it will run with
  hosts: all
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Check that ssh can connect to ubuntu3 using the ssh tool
      command: ssh -i ~/.ssh/ubuntu3_id_rsa -o BatchMode=yes -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null root@ubuntu3
      changed_when: False
      ignore_errors: True

-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c, centos1, ubuntu1
  # Serial is important as we are writing to a single file
  serial: 1

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Add host to /etc/hosts.allow for sshd
      lineinfile:
        path: /etc/hosts.allow
        line: "sshd: {{ ansible_hostname }}.diveinto.io"
        create: True
```

```
      delegate_to: ubuntu3

-

  # Hosts: where our play will run and options it will run with
  hosts: all
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Check that ssh can connect to ubuntu3 using the ssh tool
      command: ssh -i ~/.ssh/ubuntu3_id_rsa -o BatchMode=yes -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null root@ubuntu3
      changed_when: False
      ignore_errors: True

-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu3
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Drop SSH connectivity from everywhere else
      lineinfile:
        path: /etc/hosts.deny
        line: "sshd: ALL"
        create: True

-

  # Hosts: where our play will run and options it will run with
  hosts: all
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Check that ssh can connect to ubuntu3 using the ssh tool
      command: ssh -i ~/.ssh/ubuntu3_id_rsa -o BatchMode=yes -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null root@ubuntu3
      changed_when: False
      ignore_errors: True

-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu-c, centos1, ubuntu1
  # Serial is important as we are writing to a single file
  serial: 1

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Remove specific host entries in /etc/hosts.allow for sshd
      lineinfile:
        path: /etc/hosts.allow
        line: "sshd: {{ ansible_hostname }}.diveinto.io"
        state: absent
      delegate_to: ubuntu3

-

  # Hosts: where our play will run and options it will run with
  hosts: ubuntu3
  gather_facts: False

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Allow SSH connectivity from everywhere
      lineinfile:
        path: /etc/hosts.deny
        line: "sshd: ALL"
        state: absent

# Three dots indicate the end of a YAML document
...
```

7. Magic Variables

A Special Set of Reserved Ansible Variables





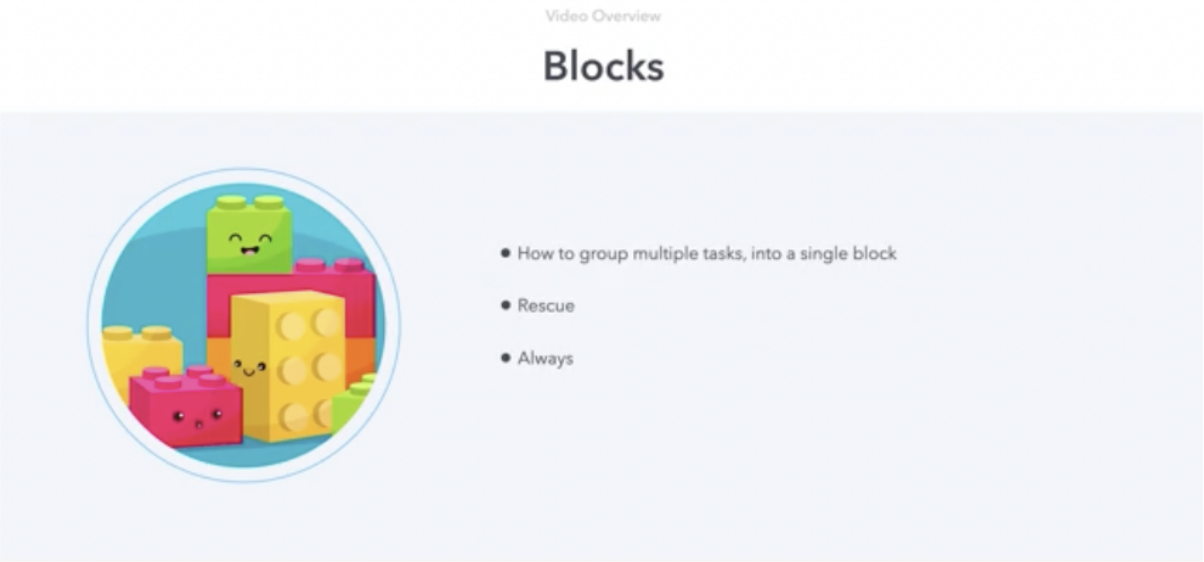

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: all

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: Using template, create a remote file that contains all variables available to the play
      template:
        src: templates/dump_variables
        dest: /tmp/ansible_variables

    - name: Fetch the templated file with all variables, back to the control host
      fetch:
        src: /tmp/ansible_variables
        dest: "captured_variables/{{ ansible_hostname }}"
        flat: yes

    - name: Clean up left over files
      file:
        name: /tmp/ansible_variables
        state: absent

# Three dots indicate the end of a YAML document
...
```

```
PLAYBOOK VARS (Ansible vars):

{{ vars | to_nice_yaml }}
```

```
$ ansible-playbook dump_vars_playbook.yaml
$ ls captured_variables/
centos1  centos2  centos3  ubuntu-c  ubuntu1  ubuntu2  ubuntu3
```

8. Blocks

For Structured Task Execution in ‘Blocks’



8-1. Simple Block

여러개의 task를 하나의 블록으로 묶어서 처리

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: A block of modules being executed
      block:
        - name: Example 1
          debug:
            msg: Example 1

        - name: Example 2
          debug:
            msg: Example 2

        - name: Example 3
          debug:
            msg: Example 3

# Three dots indicate the end of a YAML document
...
```

8-2. target 호스트 필터링

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
```

```
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:
    - name: A block of modules being executed
      block:
        - name: Example 1 CentOS only
          debug:
            msg: Example 1 CentOS only
          when: ansible_distribution == 'CentOS'

        - name: Example 2 Ubuntu only
          debug:
            msg: Example 2 Ubuntu only
          when: ansible_distribution == 'Ubuntu'

        - name: Example 3 with items
          debug:
            msg: "Example 3 with items - {{ item }}"
          with_items: ['x', 'y', 'z']

# Three dots indicate the end of a YAML document
...
```

8-3. Error Handling

python-dnspython 패키지는 centos에서는 설치가 실패한다. rescue 에서는 실패한 작업을 롤백한다.

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:

    - name: Install patch and python-dns
      block:
        - name: Install patch
          package:
            name: patch

        - name: Install python-dnspython
          package:
            name: python-dnspython

      rescue:
        - name: Rollback patch
          package:
            name: patch
            state: absent

        - name: Rollback python-dnspython
          package:
            name: python-dnspython
            state: absent

      always:
        - debug:
            msg: This always runs, regardless

# Three dots indicate the end of a YAML document
...
```

dnspython

dnspython

https://www.dnspython.org/

9. Vault

9-1. --ask-vault-pass

```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Deep Dive/Vault/01

$ cat group_vars/ubuntu
---
ansible_become: true
...

# 패스워드가 지정되어 있지 않으므로 실패한다.
$ ansible ubuntu -m ping -o
ubuntu1 | FAILED! => {"msg": "Missing sudo password"}
ubuntu3 | FAILED! => {"msg": "Missing sudo password"}
ubuntu2 | FAILED! => {"msg": "Missing sudo password"}

# ansible-valut를 사용하여 패스워드를 암호화한다.
$ ansible-vault encrypt_string --ask-vault-pass --name 'ansible_become_pass' 'password'
New Vault password: <vaultpass>
Confirm New Vault password: <vaultpass>
ansible_become_pass: !vault |
          $ANSIBLE_VAULT;1.1;AES256
          32393333633431326438386432653864323037623562643964366338333163383062373537386337
          34656361343738653139393530313233643736323038346 60a666637346536643633306438636630
          61323732353933353434396336393234386566376432633239623831346431306131383133383633
          37656461313034653 30a366530336238626564343663336232393263356264393434336137613431
          3633
Encryption successful

# 암호화된 정보를 group_vars/ubuntu 파일에 추가한다.
$ cat group_vars/ubuntu
---
ansible_become: true
ansible_become_pass: !vault |
          $ANSIBLE_VAULT;1.1;AES256
          32393333633431326438386432653864323037623562643964366338333163383062373537386337
          34656361343738653139393530313233643736323038346 60a666637346536643633306438636630
          61323732353933353434396336393234386566376432633239623831346431306131383133383633
          37656461313034653 30a366530336238626564343663336232393263356264393434336137613431
          3633
...

# 암호화된 패스워드를 입력할 수 있도록 flag를 사용한다.(--ask-vault-pass)
$ ansible --ask-vault-pass ubuntu -m ping -o
Vault password:
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"}
```

9-2. external vault

```
external_vault_var: Example External Vault Var
```

파일 암호화

```
$ ansible-vault encrypt external_vault_vars.yaml
New Vault password: <vaultpass>
Confirm New Vault password: <vaultpass>
Encryption successful
```

암호화된 파일 확인

```
$ cat external_vault_vars.yaml
$ANSIBLE_VAULT;1.1;AES256
63653537636231343735343466326634333323064316637373936303539663538393364343564633
63336133316263316366636166653230383032363832383703533333437633135373863303232633
34386564353739336138643031376663343134656162613562656135353166666266633961656538
36663362313835663640a333238643665633063386132633237393862343136303363646439393061
32313632336539336136636337393339363336303632363363373437636535366161663433386532
39353063666163313561613965383233373334303630326535390
```

```
---
# YAML documents begin with the document separator ---

# The minus in YAML this indicates a list item.  The playbook contains a list
# of plays, with each play being a dictionary
-

  # Hosts: where our play will run and options it will run with
  hosts: linux

  # Vars: variables that will apply to the play, on all target systems
  vars_files:
     - external_vault_vars.yaml

  # Tasks: the list of tasks that will be executed within the play, this section
  # can also be used for pre and post tasks
  tasks:

    - name: Show external_vault_var
      debug:
        var: external_vault_var

# Three dots indicate the end of a YAML document
...
```

```
$ ansible-playbook --ask-vault-pass vault_playbook.yaml
Vault password:

PLAY [linux] ***********************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************
ok: [centos1]
ok: [centos2]
ok: [ubuntu2]
ok: [ubuntu1]
ok: [centos3]
ok: [ubuntu3]

TASK [Show external_vault_var] *****************************************************************************************************************
ok: [centos1] => {
    "external_vault_var": "Example External Vault Var"
}
ok: [centos2] => {
    "external_vault_var": "Example External Vault Var"
}
ok: [centos3] => {
    "external_vault_var": "Example External Vault Var"
}
ok: [ubuntu1] => {
    "external_vault_var": "Example External Vault Var"
}
ok: [ubuntu2] => {
```

```
    "external_vault_var": "Example External Vault Var"
}
ok: [ubuntu3] => {
    "external_vault_var": "Example External Vault Var"
}

PLAY RECAP *********************************************************************************************************************************************
centos1                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

9-3. decrypt/encrypt

```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Deep Dive/Vault/02

# decrypt
$ ansible-vault decrypt external_vault_vars.yaml
Vault password:
Decryption successful

$ cat external_vault_vars.yaml
external_vault_var: Example External Vault Var

# encrypt
$ ansible-vault encrypt external_vault_vars.yaml
New Vault password:
Confirm New Vault password:
Encryption successful
```

9-4. rekey : change vault password

```
$ ansible-vault rekey external_vault_vars.yaml
Vault password: <vaultpass>
New Vault password: <vaultpass2>
Confirm New Vault password: <vaultpass2>
Rekey successful

$ ansible-vault view external_vault_vars.yaml
Vault password: <vaultpass2>
external_vault_var: Example External Vault Var
```

9-5. generate vault key from file

```
$ echo vaultpass2 > password_file
$ ansible-vault view --vault-password-file password_file external_vault_vars.yaml
external_vault_var: Example External Vault Var
```

9-6. prompt/read password

```
# password 입력
$ ansible-vault view --vault-id @prompt external_vault_vars.yaml
Vault password (default):
external_vault_var: Example External Vault Var

# 파일로부터 password 읽어오기
$ ansible-vault view --vault-id @password_file external_vault_vars.yaml
external_vault_var: Example External Vault Var

# vault 패스워드 변경
$ rm password_file

$ ansible-vault decrypt external_vault_vars.yaml
Vault password:
Decryption successful

# vault에 이름 지정
$ ansible-vault encrypt --vault-id vars@prompt external_vault_vars.yaml
New vault password (vars): <varpass>
```

```
Confirm new vault password (vars): <varpass>
Encryption successful

# vars로 지정된 이름을 확인할 수 있다.
$ cat external_vault_vars.yaml
$ANSIBLE_VAULT;1.2;AES256;vars
6338353934313537323539376132363138373430336239373661663030303939363762383139653 1
3434333839633365663462303034316430653466636432300a6164393839393064386166666343864
3862633232666235383330336439336432313264396537623236363266353833323366353132316 4
3235653166373665350a396263653962316464383734653733363564623137633635376336616637
3063383362373639376438383631663364643664623032636466383561396232346634636330343 2
6236366164323962373530666663326631333936393536616233

# ssh password
$ ansible-vault encrypt_string --vault-id ssh@prompt --name 'ansible_become_pass' 'password'
New vault password (ssh): <sshpass>
Confirm new vault password (ssh): <sshpass>
ansible_become_pass: !vault |
          $ANSIBLE_VAULT;1.2;AES256;ssh
          6264303331303335366133396663393666316265623330396332316265303234356535333061323 1
          6435663866313866623435383438343036323462363931310a613030613065633463366561396435
          3535313334633132313539623761383663326364316430626238303466363737336263326366646 4
          3738326431643064370a356665343761373636646561313636626465646666635303634653732623 9
          3766
Encryption successful

# group_vars/ubuntu 수정
$ cat group_vars/ubuntu
---
ansible_become: true
ansible_become_pass: !vault |
          $ANSIBLE_VAULT;1.2;AES256;ssh
          6264303331303335366133396663393666316265623330396332316265303234356535333061323 1
          6435663866313866623435383438343036323462363931310a613030613065633463366561396435
          3535313334633132313539623761383663326364316430626238303466363737336263326366646 4
          3738326431643064370a356665343761373636646561313636626465646666635303634653732623 9
          3766
...

# playbook 실행
$ ansible-playbook --vault-id vars@prompt --vault-id ssh@prompt vault_playbook.yaml
Vault password (vars): <varpass>
Vault password (ssh): <sshpass>

# encrypt playbook
$ ansible-vault encrypt --vault-id playbook@prompt vault_playbook.yaml
New vault password (playbook): <playbookpass>
Confirm new vault password (playbook): <playbookpass>
Encryption successful

# view encrypted playbook
$ cat vault_playbook.yaml
$ANSIBLE_VAULT;1.2;AES256;playbook
3932363062303138613461383637353564303131303931643038656466656466393363633866 6
3132353764383335626166373531353337366636346139350a666463336432386438323230313131
6162343333653438636338363738356338396462376431653961386164303930316462353138663 2
6361303133396131630a323862623930343534373461346435323533326333323463333637383761
6165316261343934653634646234326639656330336137333863613063393330366564306262623 0
3434643236373330653362353232646434626332393238333637383431623538336263333737353 3
3730383134333066663631613633663364383363343466303831303162343735643132316239373 5
6436613364386431633666383463306463633465666633646139363263353236393831373162366 2
6338333065353030393839336239613034393963653566393265303236663231656133616383363 3
6239353662306139343638643139393561646637373031666431353032333532323463653637643 4
6235623335306532373535343661643130343066626666383839613032313337386165373934366 1
6239363037663665313034353630393464653033643766333932325653564613936386364356234 61
3066616631653765643836333564353439656632653332326339363466643063303138623030313 5
3533363830323030343931613861643634643332666635643362343266666136396230646462666 5
3362623264373966343637356136353338633461313866626534373465386534383035623833653 7
3438656436376363316665336165333438343165313131613766306531336638346333346533386 5
6461386665323463663366656132333064376664373437356464393965643166636635353736303 5
6566373938313931366339633939363663623461626566653234316235656638623135656566343 6
6334306130666661663664666132343266631623134316165386139656265303736643763653930
6639353934616538646261653135131316638623134393332386333396138393237653830306331
6263313963636539643436393839313236346334633737566393231383339633462343534656234
6134656238616536393031356437383463326265383035643965366666643738636432333265643 6
3664333339393664303033663339366263336334643632653235383536333233613936613038306 2
3535633032373632396130346437323933653838656138366666656665613265653162383136234
6234313638396334666534613230616361666136637323863356338363964376638663765326538
3138386237393034653630326532353731373162393032393565633163623164376439306383039
3631313062643038366165636135237656365336531346466326237663462663730353632343534
3532306166323332633033363530643539306462356635373463663033331663732623738653533 9
3130303261386334386537343739663030346164343336373564356538323964633137623966666 5
3462313933363338613564388353338376333396531646461616333333039623461626636636633
6362396138386563388383331316336653065636635939656230303161623666656633323237393 8
3564383734396566434646634326136623430396139356332396366313737333239343162383935
3366626634626139303061636334383730653463663132663238663864616433313161343364343 7
3433373232323833396337303333303165653235336230393323235336665343135653761306535
```

```
6265363132396163333532313938313664373961336361383362646236623231636130623761383039373932323361653563663065316638343233616464633834313664666362313530393635323837
3034

# playbook 실행
$ ansible-playbook --vault-id vars@prompt --vault-id ssh@prompt --vault-id playbook@prompt vault_playbook.yaml
Vault password (vars): <varspass>
Vault password (ssh): <sshpass>
Vault password (playbook): <plabookpass>
```