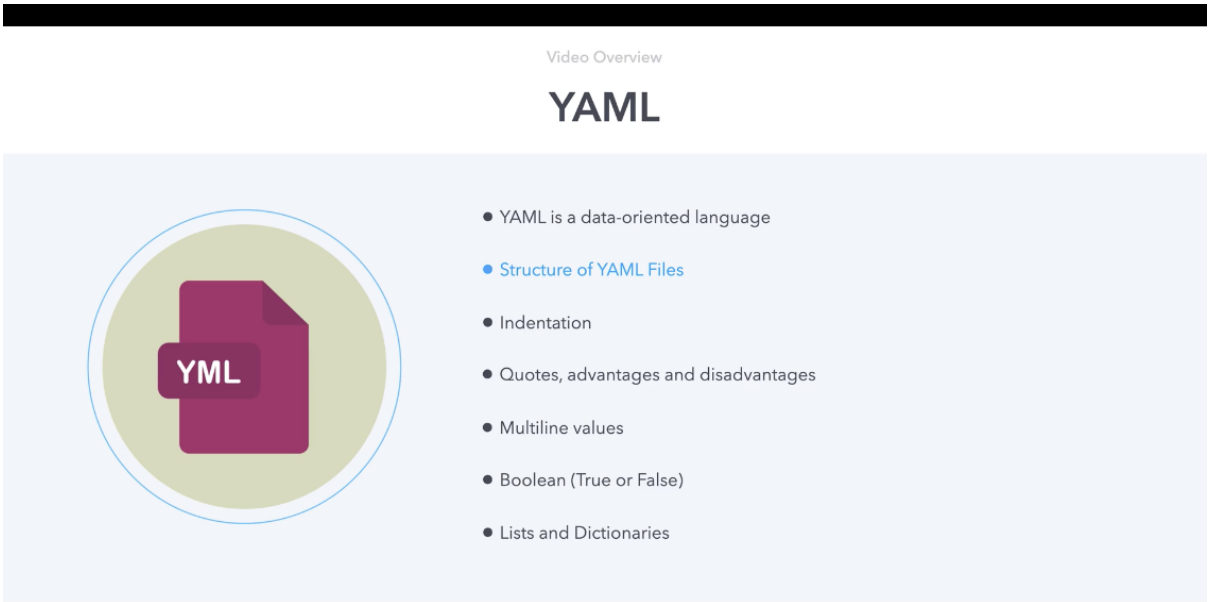
Chapter 4: Ansible Playbooks Introduction

1. YAML Ain't Markup Language (YAML)




```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Introduction/YAML/01

$ ls -l
total 8
-rwxr-xr-x 1 ansible ansible 107 Jun 24 09:08 show_yaml_python.sh
-rwxr-xr-x 1 ansible ansible 106 Jun 24 09:08 test.yaml

$ cat show_yaml_python.sh
python3 -c 'import yaml,pprint;pprint.pprint(yaml.load(open("test.yaml").read(), Loader=yaml.FullLoader))'

$ cat test.yaml
# Every YAML file should start with three dashes
---


# Every YAML file should end with three dots
...

$ ./show_yaml_python.sh
None

################################################################################

$ cd ../02
$ cat test.yaml
# Every YAML file should start with three dashes
---

example_key_1: this is a string
example_key_2: this is another string

# Every YAML file should end with three dots
...

$ ./show_yaml_python.sh
{'example_key_1': 'this is a string', 'example_key_2': 'this is another string'}

$ python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> myvar = {'example_key_1': 'this is a string', 'example_key_2': 'this is another string'}
>>>
>>> print(myvar)
{'example_key_1': 'this is a string', 'example_key_2': 'this is another string'}
>>>
>>> print(myvar['example_key_1'])
this is a string
>>> print(myvar['example_key_2'])
this is another string
>>>
```

```
#############################################################################

$ cd ../03

# 문자열을 항상 쌍따옴표로 묶을 필요는 없다.
$ cat test.yaml
# Every YAML file should start with three dashes
---

no_quotes: this is a string example
double_quotes: "this is a string example"
single_quotes: 'this is a string example'

# Every YAML file should end with three dots
...

# 파이썬은 Single Quote로 문자열을 처리한다.
$ ./show_yaml_python.sh
{'double_quotes': 'this is a string example',
 'no_quotes': 'this is a string example',
 'single_quotes': 'this is a string example'}

#############################################################################

$ cd ../04

# 문자열안에 줄바꿈 문자가 포함되어 있다.
$ cat test.yaml
# Every YAML file should start with three dashes
---

no_quotes: this is a string example\n
double_quotes: "this is a string example\n"
single_quotes: 'this is a string example\n'

# Every YAML file should end with three dots
...

# double quotes : no escape
# single quotes : escape
# no quotes : escape
$ ./show_yaml_python.sh
{'double_quotes': 'this is a string example\n',
 'no_quotes': 'this is a string example\\n',
 'single_quotes': 'this is a string example\\n'}

#############################################################################

$ cd ../05

# multiline 으로 구성된 문자열 처리 방법(|)
$ cat test.ymal
# Every YAML file should start with three dashes
---

example_key_1: |
  this is a string
  that goes over
  multiple lines

# Every YAML file should end with three dots
...

# 줄바꿈이 escape 되지 않는것을 확인할 수 있다. 즉, multiline으로 처리
$ ./show_yaml_python.sh
{'example_key_1': 'this is a string\nthat goes over\nmultiple lines\n'}

# 파이썬에서 직접 확인
$ python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print('this is a string\nthat goes over\nmultiple lines\n')
this is a string
that goes over
multiple lines

>>>

#############################################################################

$ cd ../06

$ cat test.yaml
# Every YAML file should start with three dashes
---
```

```
# multiline을 > 으로 처리할 수 있다.
example_key_1: >
  this is a string
  that goes over
  multiple lines

# Every YAML file should end with three dots
...

# 파이썬에서 직접 확인
# 줄바꿈 문자가 없어지고 마지막 라인에만 처리됨을 주의 한다.
$ ./show_yaml_python.sh
{'example_key_1': 'this is a string that goes over multiple lines\n'}

################################################################################

$ cd ../07

# >- 에서 마지막 - 은 줄바꿈문자를 제거한다.
$ cat test.yaml
# Every YAML file should start with three dashes
---

example_key_1: >-
  this is a string
  that goes over
  multiple lines

# Every YAML file should end with three dots
...

# 파이썬에서 직접 처리
$ ./show_yaml_python.sh
{'example_key_1': 'this is a string that goes over multiple lines'}

################################################################################

$ cd ../08

# integer 타입을 선언하는 방법(따옴표 없음을 주의)
$ cat test.yaml
# Every YAML file should start with three dashes
---

example_integer: 1

# Every YAML file should end with three dots
...

$ ./show_yaml_python.sh
{'example_integer': 1}

# 파이썬에 데이터 타입 확인
$ python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> myvar = {'example_integer': 1}
>>> print(myvar['example_integer'])
1
>>> print(type(myvar['example_integer']))
<class 'int'>
>>>

################################################################################

$ cd ../09

# 따옴표 처리되어 있으므로 문자열로 처리된다.
$ cat test.yaml
# Every YAML file should start with three dashes
---

example_integer: "1"

# Every YAML file should end with three dots
...
$ ./show_yaml_python.sh
{'example_integer': '1'}

################################################################################

$ cd ../10

# 불리언 처리
$ cat test.yaml
---
```

```
# Every YAML file should start with three dashes

# false, False, FALSE, no, No, NO, off, Off, OFF
# true, True, TRUE, yes, Yes, YES, on, On, ON

# n.b. n does not equal false, y does not equal true

is_false_01: false
is_false_02: False
is_false_03: FALSE
is_false_04: no
is_false_05: No
is_false_06: NO
is_false_07: off
is_false_08: Off
is_false_09: OFF
is_false_10: n
is_true_01: true
is_true_02: True
is_true_03: TRUE
is_true_04: yes
is_true_05: Yes
is_true_06: YES
is_true_07: on
is_true_08: On
is_true_09: ON
is_true_10: y

# Every YAML file should end with three dots
...

# YAML 의 n과 y는 불리언으로 처리되지 않고 문자열로 처리된다.
$ ./show_yaml_python.sh
{'is_false_01': False,
 'is_false_02': False,
 'is_false_03': False,
 'is_false_04': False,
 'is_false_05': False,
 'is_false_06': False,
 'is_false_07': False,
 'is_false_08': False,
 'is_false_09': False,
 'is_false_10': 'n',
 'is_true_01': True,
 'is_true_02': True,
 'is_true_03': True,
 'is_true_04': True,
 'is_true_05': True,
 'is_true_06': True,
 'is_true_07': True,
 'is_true_08': True,
 'is_true_09': True,
 'is_true_10': 'y'}

# 파이썬과의 구문일치를 위해 YAML에서의 불리언은 True 또는 False를 쓰는것이 일관성이 있다.

###############################################################################

$ cd ../11

# 리스트 처리
$ cat test.yaml
---
# Every YAML file should start with three dashes

- item 1
- item 2
- item 3
- item 4
- item 5

# Every YAML file should end with three dots
...

$ ./show_yaml_python.sh
['item 1', 'item 2', 'item 3', 'item 4', 'item 5']

$ python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> myvar = ['item 1', 'item 2', 'item 3', 'item 4', 'item 5']
>>>
>>> print(myvar)
['item 1', 'item 2', 'item 3', 'item 4', 'item 5']
>>> print(myvar[0])
item 1
```

```
>>> print(myvar[1])
item 2
>>> print(myvar[2])
item 3
>>> print(myvar[3])
item 4
>>> print(myvar[4])
item 5
>>> print(myvar[5])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range
>>>

#########################################################################

$ cd ../13

# 인라인으로 처리(딕셔너리)
$ cat test.yaml
---
# Every YAML file should start with three dashes

{example_key_1: example_value_1, example_key_2: example_value_2}

# Every YAML file should end with three dots
...

$ ./show_yaml_python.sh
{'example_key_1': 'example_value_1', 'example_key_2': 'example_value_2'}

# 가급적 YAML 포맷을 유지하는 것이 좋다.

#########################################################################

$ cd ../14

# 인라인으로 처리(리스트)
$ cat test.yaml
---
# Every YAML file should start with three dashes

[example_list_entry_1, example_list_entry_2]

# Every YAML file should end with three dots
...
$ ./show_yaml_python.sh
['example_list_entry_1', 'example_list_entry_2']

# 가급적 YAML 포맷을 유지하는 것이 좋다.

#########################################################################

$ cd ../15

# invalid yaml 포맷
$ cat test.yaml
---
# Every YAML file should start with three dashes

example_key_1: example_value_1
- example_list_entry_1

# Every YAML file should end with three dots
...

$ ./show_yaml_python.sh
Traceback (most recent call last):
  File "<string>", line 1, in <module>
  File "/usr/local/lib/python3.8/dist-packages/yaml/__init__.py", line 81, in load
    return loader.get_single_data()
  File "/usr/local/lib/python3.8/dist-packages/yaml/constructor.py", line 49, in get_single_data
    node = self.get_single_node()
  File "/usr/local/lib/python3.8/dist-packages/yaml/composer.py", line 36, in get_single_node
    document = self.compose_document()
  File "/usr/local/lib/python3.8/dist-packages/yaml/composer.py", line 55, in compose_document
    node = self.compose_node(None, None)
  File "/usr/local/lib/python3.8/dist-packages/yaml/composer.py", line 84, in compose_node
    node = self.compose_mapping_node(anchor)
  File "/usr/local/lib/python3.8/dist-packages/yaml/composer.py", line 127, in compose_mapping_node
    while not self.check_event(MappingEndEvent):
  File "/usr/local/lib/python3.8/dist-packages/yaml/parser.py", line 98, in check_event
    self.current_event = self.state()
  File "/usr/local/lib/python3.8/dist-packages/yaml/parser.py", line 438, in parse_block_mapping_key
    raise ParserError("while parsing a block mapping", self.marks[-1],
yaml.parser.ParserError: while parsing a block mapping
  in "<unicode string>", line 4, column 1:
```

```
    example_key_1: example_value_1
    ^
expected <block end>, but found '-'
  in "<unicode string>", line 5, column 1:
    - example_list_entry_1
    ^

########################################################################

$ cd ../16

# invalid yaml 포맷
$ cat test.yaml
---
# Every YAML file should start with three dashes

{example_key_1: example_value_1}
[example_list_entry_1]

# Every YAML file should end with three dots
...
$ ./show_yaml_python.sh
Traceback (most recent call last):
  File "<string>", line 1, in <module>
  File "/usr/local/lib/python3.8/dist-packages/yaml/__init__.py", line 81, in load
    return loader.get_single_data()
  File "/usr/local/lib/python3.8/dist-packages/yaml/constructor.py", line 49, in get_single_data
    node = self.get_single_node()
  File "/usr/local/lib/python3.8/dist-packages/yaml/composer.py", line 39, in get_single_node
    if not self.check_event(StreamEndEvent):
  File "/usr/local/lib/python3.8/dist-packages/yaml/parser.py", line 98, in check_event
    self.current_event = self.state()
  File "/usr/local/lib/python3.8/dist-packages/yaml/parser.py", line 171, in parse_document_start
    raise ParserError(None, None,
yaml.parser.ParserError: expected '<document start>', but found '['
  in "<unicode string>", line 5, column 1:
    [example_list_entry_1]
    ^

########################################################################

$ cd ../17

# valid yaml format(indentation)
$ cat test.yaml
---
# Every YAML file should start with three dashes

example_key_1:
  sub_example_key1: sub_example_value1

example_key_2:
  sub_example_key2: sub_example_value2

# Every YAML file should end with three dots
...

$ ./show_yaml_python.sh
{'example_key_1': {'sub_example_key1': 'sub_example_value1'},
 'example_key_2': {'sub_example_key2': 'sub_example_value2'}}

########################################################################

$ cd ../18

$ cat test.yaml
---
# Every YAML file should start with three dashes

example_1:
  - item_1
  - item_2
  - item_3

example_2:
  - item_4
  - item_5
  - item_6

# Every YAML file should end with three dots
...
ansible@ubuntu-c:~/diveintoansible/Ansible Playbooks, Introduction/YAML/18$ ./show_yaml_python.sh
{'example_1': ['item_1', 'item_2', 'item_3'],
 'example_2': ['item_4', 'item_5', 'item_6']}

########################################################################
```

```
$ cd ../19

$ cat test.yaml
---
# Every YAML file should start with three dashes

example_dictionary_1:
  - example_dictionary_2:
    - 1
    - 2
    - 3
  - example_dictionary_3:
    - 4
    - 5
    - 6
  - example_dictionary_4:
    - 7
    - 8
    - 9

# Every YAML file should end with three dots
...
ansible@ubuntu-c:~/diveintoansible/Ansible Playbooks, Introduction/YAML/19$ ./show_yaml_python.sh
{'example_dictionary_1': [{'example_dictionary_2': [1, 2, 3]},
                          {'example_dictionary_3': [4, 5, 6]},
                          {'example_dictionary_4': [7, 8, 9]}]
```

Quiz

1. Create a file called test.yaml, with the appropriate start and end markers, run the test utility, the output, show no none

```
---
...
```

2. Create a list of Car Manufactures, it show include:

- Aston Martin
- Fiat
- Ford
- Vauxhall

```
---
- Aston Martin
- Fiat
- Ford
- Vauxhall
...
```

3. Change each of these entries in the list, so that they are dictionaries.

```
---
- Aston Martin:
- Fiat:
- Ford:
- Vauxhall:
...
```

4. Add as key values to each manufacturer, year_founded and website

```
---
- Aston Martin:
  year_found: 1913
  website: astonmartin.com
- Fiat:
  year_found: 1899
  website: fiat.com
- Ford:
  year_found: 1903
  website: ford.com
- Vauxhall:
  year_founded: 1857
```

```
  website: vauxhall.co.uk
...
```

5. Add a key of founded_by, but with a list of the founder, or founders

```
---
- Aston Martin:
  year_found: 1913
  website: astonmartin.com
  founded_by:
    - Lionel Martin
    - Robert Bamford
- Fiat:
  year_found: 1899
  website: fiat.com
  founded_by:
    - Giovanni Agnelli
- Ford:
  year_found: 1903
  website: ford.com
  founded_by:
    - Henry Ford
- Vauxhall:
  year_founded: 1857
  website: vauxhall.co.uk
  founded_by:
    - Alexander Wilson
...
```

2. Ansible Playbooks- Breakdown of Sections

```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Introduction/Ansible Playbooks, Breakdown of Sections/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

  # Vars: variables that will apply to the play, on all target systems

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

```
  # Handlers: the list of handlers that are executed as a notify key from a task

  # Roles: list of roles to be imported into the play

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

$ cat centos_motd
Welcome to CentOS Linux - Ansible Rocks

############################################################################

# playbook

$ 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: centos
  user: root

  # Vars: variables that will apply to the play, on all target systems

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Configure a MOTD (message of the day)
      copy:
        src: centos_motd
        dest: /etc/motd

  # Handlers: the list of handlers that are executed as a notify key from a task

  # Roles: list of roles to be imported into the play

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

############################################################################

# playbook 실행

$ ansible-playbook motd_playbook.yaml

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

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

TASK [Configure a MOTD (message of the day)] ***********************************************************************************************
changed: [centos2]
changed: [centos3]
changed: [centos1]

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

############################################################################

# centos1 에서 확인

centos1 login: ansible
Password:
Welcome to CentOS Linux - Ansible Rocks
[ansible@centos1 ~]$

############################################################################

# ubuntu-c 에서 재 실행
# 변경사항이 없으므로 모두 green

$ ansible-playbook motd_playbook.yaml

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

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

```
TASK [Configure a MOTD (message of the day)] ***********************************************************************************
ok: [centos1]
ok: [centos2]
ok: [centos3]

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
```

Video Overview

Target Options

- become
- connection
- gather_facts

```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Introduction/Ansible Playbooks, Breakdown of Sections/02

$ time ansible-playbook motd_playbook.yaml

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

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

TASK [Configure a MOTD (message of the day)] ***********************************************************************************
ok: [centos2]
ok: [centos1]
ok: [centos3]

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


real    0m3.668s
user    0m1.944s
sys     0m0.820s
```

`gather_facts: False` 를 통한 수행 시간 비교

- 전체 수행 시간 감소
- 수행되는 task 갯수 감소(1개)

```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Introduction/Ansible Playbooks, Breakdown of Sections/03

$ time ansible-playbook motd_playbook.yaml

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

```
TASK [Configure a MOTD (message of the day)] ***********************************************************************************
ok: [centos3]
ok: [centos1]
ok: [centos2]

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


real    0m2.380s
user    0m1.328s
sys     0m0.551s
```

copy 모듈의 `content` 속성 사용

```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Introduction/Ansible Playbooks, Breakdown of Sections/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: centos
  user: root
  gather_facts: False

  # Vars: variables that will apply to the play, on all target systems

  # 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 CentOS Linux - Ansible Rocks
        dest: /etc/motd

  # Handlers: the list of handlers that are executed as a notify key from a task

  # Roles: list of roles to be imported into the play

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

#
# target 호스트의 파일내용을 같은 내용으로 바꿈에도 불구하고 changed 로 표시됨을 유의한다.
#

$ ansible-playbook motd_playbook.yaml

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

TASK [Configure a MOTD (message of the day)] ***********************************************************************************
changed: [centos2]
changed: [centos3]
changed: [centos1]

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

#
# centos1 호스트에서 확인
# - 로그인 배너에서 줄바꿈이 없음을 확인한다.
#

centos1 login: ansible
Password:
Last login: Mon Jun 27 02:16:10 on pts/0
Welcome to CentOS Linux - Ansible Rocks[ansible@centos1 ~]$
```

vars와 jinja2 템플릿을 사용

```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Introduction/Ansible Playbooks, Breakdown of Sections/05
```

```
$ 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: centos
  user: root
  gather_facts: False

  # Vars: variables that will apply to the play, on all target systems
  vars:
    motd: "Welcome to CentOS 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 }}"
        dest: /etc/motd

  # Handlers: the list of handlers that are executed as a notify key from a task

  # Roles: list of roles to be imported into the play

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

cli에서 변수 오버라이드 하기

```
$ ansible-playbook motd_playbook.yaml -e 'motd="Testing the modtd playbook\n"'

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

TASK [Configure a MOTD (message of the day)] ***********************************************************************************************************
changed: [centos3]
changed: [centos2]
changed: [centos1]

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

handler : task 의 notify에 의해 실행됨(task 실행 후)

```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Introduction/Ansible Playbooks, Breakdown of Sections/06

$ 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: centos
  user: root
  gather_facts: False

  # Vars: variables that will apply to the play, on all target systems
  vars:
    motd: "Welcome to CentOS 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 }}"
        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
```

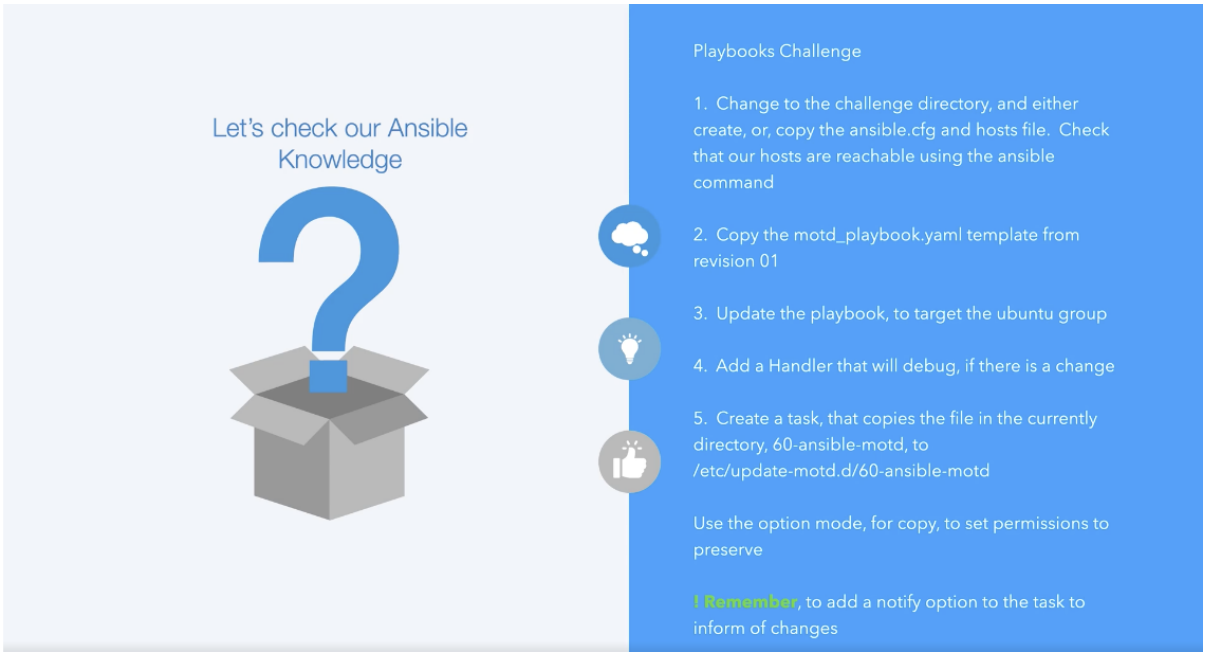
```
      debug:
        msg: The MOTD was changed

  # Roles: list of roles to be imported into the play

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

#############################################################################

$ ansible-playbook motd_playbook.yaml

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

TASK [Configure a MOTD (message of the day)] ***********************************************************************************************************
changed: [centos1]
changed: [centos3]
changed: [centos2]

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

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

#############################################################################
#
# 변경 사항이 없으므로 handler 실행 안됨
#

$ ansible-playbook motd_playbook.yaml

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

TASK [Configure a MOTD (message of the day)] ***********************************************************************************************************
ok: [centos1]
ok: [centos2]
ok: [centos3]

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

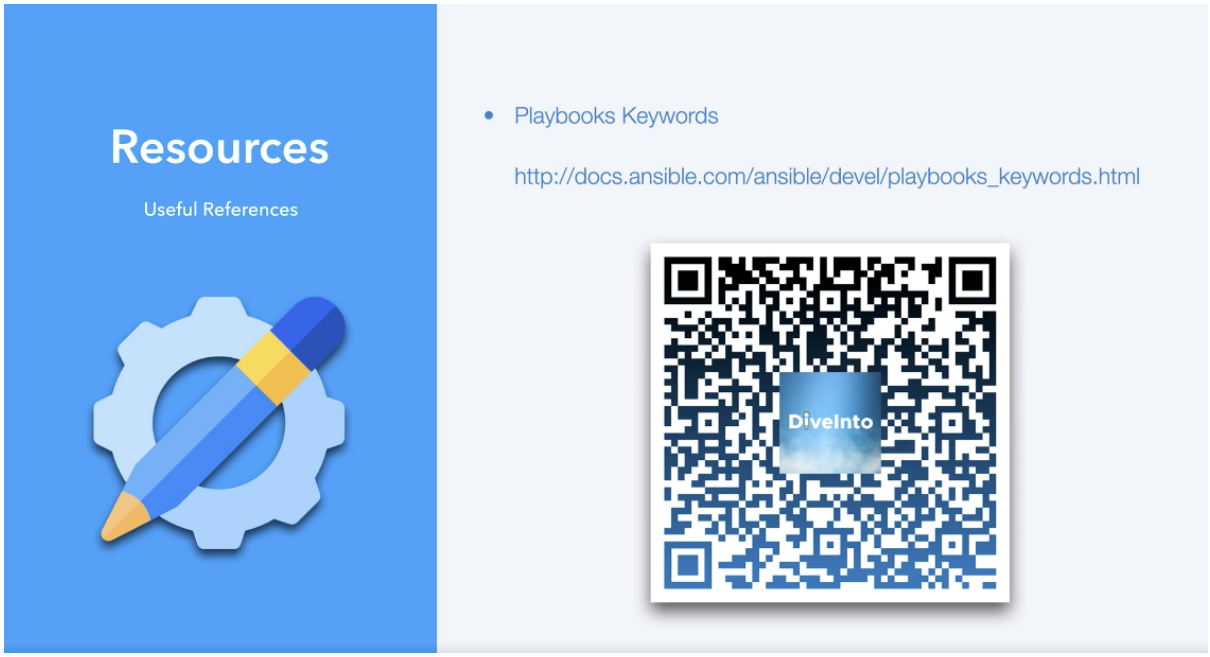
호스트의 OS 종류(배포본)에 따라 다르게 motd 설정하기. 배포본 종류를 찾기 위해 setup 모듈을 통해 값을 먼저 확인

```
$ ansible all -i centos2, -m setup | more
...
...
        "ansible_distribution": "CentOS",
        "ansible_distribution_file_parsed": true,
        "ansible_distribution_file_path": "/etc/redhat-release",
        "ansible_distribution_file_variety": "RedHat",
        "ansible_distribution_major_version": "8",
        "ansible_distribution_release": "NA",
        "ansible_distribution_version": "8.4",
        "ansible_dns": {
            "nameservers": [
                "127.0.0.11"
            ],
            "options": {
                "ndots": "0"
            }
        },
...
...

$ ansible all -i centos2,ubuntu2, -m setup | grep ansible_distribution
        "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",
        "ansible_distribution": "CentOS",
        "ansible_distribution_file_parsed": true,
        "ansible_distribution_file_path": "/etc/redhat-release",
        "ansible_distribution_file_variety": "RedHat",
        "ansible_distribution_major_version": "8",
        "ansible_distribution_release": "NA",
        "ansible_distribution_version": "8.4",

$ 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

  # Roles: list of roles to be imported into the play

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

########################################################################
#
# centos 는 앞서 변경 했으므로 이번에는 ubuntu 호스트에 대해서만 변경
#

$ ansible-playbook motd_playbook.yaml

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

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

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

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

RUNNING HANDLER [MOTD changed] *****************************************************************************************************************
ok: [ubuntu2] => {
```

```
    "msg": "The MOTD was changed"
}
ok: [ubuntu3] => {
    "msg": "The MOTD was changed"
}
ok: [ubuntu1] => {
    "msg": "The MOTD was changed"
}

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

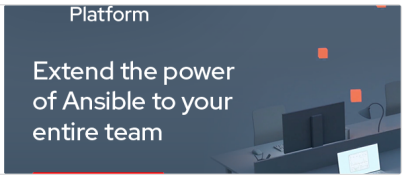




Playbook Keywords - Ansible Documentation

A dictionary that gets converted into environment vars to be provided for the task upon execution. This can ONLY be used with modules. This isn't supported for any other type of plugins nor Ansible itself nor its configuration, it just sets the variables for the code responsible for executing the task.

https://docs.ansible.com/ansible/devel/reference_appendices/playbooks_keywords.html



3. Ansible Playbooks- Variables

```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Introduction/Ansible Playbooks, Variables/01

$ cat variables_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: centos1
  gather_facts: False

  # Vars: variables that will apply to the play, on all target systems
  vars:
    example_key: example value

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Test dictionary key value
      debug:
        msg: "{{ example_key }}"

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

$ ansible-playbook variables_playbook.yaml

PLAY [centos1] *****************************************************************************************************************************************

TASK [Test dictionary key value] ***********************************************************************************************************************
ok: [centos1] => {
    "msg": "example value"
}

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

딕셔너리 vars

```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Introduction/Ansible Playbooks, Variables/02

$ cat variables_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: centos1
  gather_facts: False

  # Vars: variables that will apply to the play, on all target systems
  vars:
    dict:
      dict_key: This is a dictionary value

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Test named dictionary dictionary
      debug:
        msg: "{{ dict }}"

    - name: Test named dictionary dictionary key value with dictionary dot notation
      debug:
```

```
        msg: "{{ dict.dict_key }}"

    - name: Test named dictionary dictionary key value with python brackets notation
      debug:
        msg: "{{ dict['dict_key'] }}"

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

$ ansible-playbook variables_playbook.yaml
[WARNING]: Found variable using reserved name: dict

PLAY [centos1] ***************************************************************************************************************************

TASK [Test named dictionary dictionary] **************************************************************************************************
ok: [centos1] => {
    "msg": {
        "dict_key": "This is a dictionary value"
    }
}

TASK [Test named dictionary dictionary key value with dictionary dot notation] ***********************************************************
ok: [centos1] => {
    "msg": "This is a dictionary value"
}

TASK [Test named dictionary dictionary key value with python brackets notation] **********************************************************
ok: [centos1] => {
    "msg": "This is a dictionary value"
}

PLAY RECAP *******************************************************************************************************************************
centos1                    : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

인라인 딕셔너리

```
$ cd ../03

$ cat variables_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: centos1
  gather_facts: False

  # Vars: variables that will apply to the play, on all target systems
  vars:
    inline_dict:
      {inline_dict_key: This is an inline dictionary value}

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Test named inline dictionary dictionary
      debug:
        msg: "{{ inline_dict }}"

    - name: Test named inline dictionary dictionary key value with dictionary dot notation
      debug:
        msg: "{{ inline_dict.inline_dict_key }}"

    - name: Test named inline dictionary dictionary key value with brackets notation
      debug:
        msg: "{{ inline_dict['inline_dict_key'] }}"

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

$ ansible-playbook variables_playbook.yaml

PLAY [centos1] ***************************************************************************************************************************

TASK [Test named inline dictionary dictionary] *******************************************************************************************
ok: [centos1] => {
    "msg": {
        "inline_dict_key": "This is an inline dictionary value"
    }
}

TASK [Test named inline dictionary dictionary key value with dictionary dot notation] ****************************************************
```

```
ok: [centos1] => {
    "msg": "This is an inline dictionary value"
}

TASK [Test named inline dictionary dictionary key value with brackets notation] ***********************************************************
ok: [centos1] => {
    "msg": "This is an inline dictionary value"
}

PLAY RECAP ********************************************************************************************************************************
centos1                    : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

리스트 사용 #1

```
$ cd ../04

$ cat variables_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: centos1
  gather_facts: False

  # Vars: variables that will apply to the play, on all target systems
  vars:
    named_list:
      - item1
      - item2
      - item3
      - item4

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Test named list
      debug:
        msg: "{{ named_list }}"

    - name: Test named list first item dot notation
      debug:
        msg: "{{ named_list.0 }}"

    - name: Test named list first item brackets notation
      debug:
        msg: "{{ named_list[0] }}"

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

$ ansible-playbook variables_playbook.yaml

PLAY [centos1] ****************************************************************************************************************************

TASK [Test named list] ********************************************************************************************************************
ok: [centos1] => {
    "msg": [
        "item1",
        "item2",
        "item3",
        "item4"
    ]
}

TASK [Test named list first item dot notation] ********************************************************************************************
ok: [centos1] => {
    "msg": "item1"
}

TASK [Test named list first item brackets notation] ***************************************************************************************
ok: [centos1] => {
    "msg": "item1"
}

PLAY RECAP ********************************************************************************************************************************
centos1                    : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

리스트 사용 #2

```
$ cd ../05

$ cat variables_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: centos1
  gather_facts: False

  # Vars: variables that will apply to the play, on all target systems
  vars:
    inline_named_list:
      [ item1, item2, item3, item4 ]

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Test inline named list
      debug:
        msg: "{{ inline_named_list }}"

    - name: Test inline named list first item dot notation
      debug:
        msg: "{{ inline_named_list.0 }}"

    - name: Test inline named list first item brackets notation
      debug:
        msg: "{{ inline_named_list[0] }}"

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

$ ansible-playbook variables_playbook.yaml

PLAY [centos1] *****************************************************************************************************************************

TASK [Test inline named list] **************************************************************************************************************
ok: [centos1] => {
    "msg": [
        "item1",
        "item2",
        "item3",
        "item4"
    ]
}

TASK [Test inline named list first item dot notation] **************************************************************************************
ok: [centos1] => {
    "msg": "item1"
}

TASK [Test inline named list first item brackets notation] *********************************************************************************
ok: [centos1] => {
    "msg": "item1"
}

PLAY RECAP *********************************************************************************************************************************
centos1                    : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

외부 변수(external vars)

```
$ cd ../06

$ cat external_vars.yaml
---
external_example_key: example value

external_dict:
   dict_key: This is a dictionary value

external_inline_dict:
   {inline_dict_key: This is an inline dictionary value}

external_named_list:
   - item1
   - item2
   - item3
   - item4
```

```
external_inline_named_list:
   [ item1, item2, item3, item4 ]
...

$ cat variables_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: centos1
  gather_facts: False

  # Vars: variables that will apply to the play, on all target systems
  vars_files:
    - external_vars.yaml

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Test external dictionary key value
      debug:
        msg: "{{ external_example_key }}"

    - name: Test external named dictionary dictionary
      debug:
        msg: "{{ external_dict }}"

    - name: Test external named dictionary dictionary key value with dictionary dot notation
      debug:
        msg: "{{ external_dict.dict_key }}"

    - name: Test external named dictionary dictionary key value with brackets notation
      debug:
        msg: "{{ external_dict['dict_key'] }}"

    - name: Test external named inline dictionary dictionary
      debug:
        msg: "{{ external_inline_dict }}"

    - name: Test external named inline dictionary dictionary key value with dictionary dot notation
      debug:
        msg: "{{ external_inline_dict.inline_dict_key }}"

    - name: Test external named inline dictionary dictionary key value with brackets notation
      debug:
        msg: "{{ external_inline_dict['inline_dict_key'] }}"

    - name: Test external named list
      debug:
        msg: "{{ external_named_list }}"

    - name: Test external named list first item dot notation
      debug:
        msg: "{{ external_named_list.0 }}"

    - name: Test external named list first item brackets notation
      debug:
        msg: "{{ external_named_list[0] }}"

    - name: Test external inline named list
      debug:
        msg: "{{ external_inline_named_list }}"

    - name: Test external inline named list first item dot notation
      debug:
        msg: "{{ external_inline_named_list.0 }}"

    - name: Test external inline named list first item brackets notation
      debug:
        msg: "{{ external_inline_named_list[0] }}"


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

$ ansible-playbook variables_playbook.yaml

PLAY [centos1] *****************************************************************************************************************************************

TASK [Test external dictionary key value] **************************************************************************************************************
ok: [centos1] => {
    "msg": "example value"
}
```

```
TASK [Test external named dictionary dictionary] ********************************************************************************
ok: [centos1] => {
    "msg": {
        "dict_key": "This is a dictionary value"
    }
}

TASK [Test external named dictionary dictionary key value with dictionary dot notation] ****************************************
ok: [centos1] => {
    "msg": "This is a dictionary value"
}

TASK [Test external named dictionary dictionary key value with brackets notation] ***********************************************
ok: [centos1] => {
    "msg": "This is a dictionary value"
}

TASK [Test external named inline dictionary dictionary] *************************************************************************
ok: [centos1] => {
    "msg": {
        "inline_dict_key": "This is an inline dictionary value"
    }
}

TASK [Test external named inline dictionary dictionary key value with dictionary dot notation] **********************************
ok: [centos1] => {
    "msg": "This is an inline dictionary value"
}

TASK [Test external named inline dictionary dictionary key value with brackets notation] ****************************************
ok: [centos1] => {
    "msg": "This is an inline dictionary value"
}

TASK [Test external named list] *************************************************************************************************
ok: [centos1] => {
    "msg": [
        "item1",
        "item2",
        "item3",
        "item4"
    ]
}

TASK [Test external named list first item dot notation] *************************************************************************
ok: [centos1] => {
    "msg": "item1"
}

TASK [Test external named list first item brackets notation] ********************************************************************
ok: [centos1] => {
    "msg": "item1"
}

TASK [Test external inline named list] ******************************************************************************************
ok: [centos1] => {
    "msg": [
        "item1",
        "item2",
        "item3",
        "item4"
    ]
}

TASK [Test external inline named list first item dot notation] ******************************************************************
ok: [centos1] => {
    "msg": "item1"
}

TASK [Test external inline named list first item brackets notation] *************************************************************
ok: [centos1] => {
    "msg": "item1"
}

PLAY RECAP **********************************************************************************************************************
centos1                    : ok=13   changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

vars_prompt #1

```
$ cd ../07

$ cat variables_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: centos1
  gather_facts: False

  # Vars: variables that will apply to the play, on all target systems
  vars_prompt:
    - name: username
      private: False

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Test vars_prompt
      debug:
        msg: "{{ username }}"

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

$ ansible-playbook variables_playbook.yaml
username: James

PLAY [centos1] ***************************************************************************************************************************************

TASK [Test vars_prompt] ******************************************************************************************************************************
ok: [centos1] => {
    "msg": "James"
}

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

vars_prompt #2

```
$ cd ../08

$ cat variables_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: centos1
  gather_facts: False

  # Vars: variables that will apply to the play, on all target systems
  vars_prompt:
    - name: password
      private: True

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Test vars_prompt
      debug:
        msg: "{{ password }}"

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

$ ansible-playbook variables_playbook.yaml
password:

PLAY [centos1] ***************************************************************************************************************************************

TASK [Test vars_prompt] ******************************************************************************************************************************
ok: [centos1] => {
    "msg": "12345"
}

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

hostvars

```
$ cd ../09

$ cat variables_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: centos1
  gather_facts: True

  # Vars: variables that will apply to the play, on all target systems

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Test hostvars with an ansible fact and collect ansible_port, dot notation
      debug:
        msg: "{{ hostvars[ansible_hostname].ansible_port }}"

    - name: Test hostvars with an ansible fact and collect ansible_port, dict notation
      debug:
        msg: "{{ hostvars[ansible_hostname]['ansible_port'] }}"

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

$ ansible-playbook variables_playbook.yaml

PLAY [centos1] *****************************************************************************************************************************************

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

TASK [Test hostvars with an ansible fact and collect ansible_port, dot notation] ***********************************************************************
ok: [centos1] => {
    "msg": "2222"
}

TASK [Test hostvars with an ansible fact and collect ansible_port, dict notation] ***********************************************************************
ok: [centos1] => {
    "msg": "2222"
}

PLAY RECAP *********************************************************************************************************************************************
centos1                    : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

오픈된 port가 다를 경우 에러 발생

```
$ cd ../10

$ cat hosts
[control]
ubuntu-c ansible_connection=local

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

[centos:vars]
ansible_user=root

[ubuntu]
ubuntu[1:3]

[ubuntu:vars]
ansible_become=true
ansible_become_pass=password

[linux:children]
centos
ubuntu

$ cat variables_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
  gather_facts: True

  # Vars: variables that will apply to the play, on all target systems

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Test hostvars with an ansible fact and collect ansible_port, dot notation
      debug:
        msg: "{{ hostvars[ansible_hostname].ansible_port }}"

    - name: Test hostvars with an ansible fact and collect ansible_port, dict notation
      debug:
        msg: "{{ hostvars[ansible_hostname]['ansible_port'] }}"

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

$ ansible-playbook variables_playbook.yaml

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

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

TASK [Test hostvars with an ansible fact and collect ansible_port, dot notation] ***********************************************************
ok: [centos1] => {
    "msg": "2222"
}
fatal: [centos2]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'ansible.vars.hostvars.Ho
fatal: [centos3]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'ansible.vars.hostvars.Ho

TASK [Test hostvars with an ansible fact and collect ansible_port, dict notation] **********************************************************
ok: [centos1] => {
    "msg": "2222"
}

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

default value 지정

```
$ cd ../11

$ cat variables_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
  gather_facts: True

  # Vars: variables that will apply to the play, on all target systems

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Test hostvars with an ansible fact and collect ansible_port, dot notation, default if not found
      debug:
        msg: "{{ hostvars[ansible_hostname].ansible_port | default('22') }}"

    - name: Test hostvars with an ansible fact and collect ansible_port, dict notation, default if not found
      debug:
        msg: "{{ hostvars[ansible_hostname]['ansible_port'] | default('22') }}"

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

$ ansible-playbook variables_playbook.yaml

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

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

```
TASK [Test hostvars with an ansible fact and collect ansible_port, dot notation, default if not found] ******************************
ok: [centos1] => {
    "msg": "2222"
}
ok: [centos2] => {
    "msg": "22"
}
ok: [centos3] => {
    "msg": "22"
}

TASK [Test hostvars with an ansible fact and collect ansible_port, dict notation, default if not found] ******************************
ok: [centos1] => {
    "msg": "2222"
}
ok: [centos2] => {
    "msg": "22"
}
ok: [centos3] => {
    "msg": "22"
}

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

ansible_user(centos)

```
$ cd ../12

$ cat variables_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
  gather_facts: True

  # Vars: variables that will apply to the play, on all target systems

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Test groupvars
      debug:
        msg: "{{ ansible_user }}"

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

$ ansible-playbook variables_playbook.yaml

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

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

TASK [Test groupvars] *********************************************************************************************************************
ok: [centos1] => {
    "msg": "root"
}
ok: [centos2] => {
    "msg": "root"
}
ok: [centos3] => {
    "msg": "root"
}

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
```

ansible_user(ubuntu)

```
$ cd ../13

$ cat variables_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
  gather_facts: True

  # Vars: variables that will apply to the play, on all target systems

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Test groupvars
      debug:
        msg: "{{ ansible_user }}"

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

$ ansible-playbook variables_playbook.yaml

PLAY [ubuntu] ******************************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************
ok: [ubuntu2]
ok: [ubuntu3]
ok: [ubuntu1]

TASK [Test groupvars] **********************************************************************************************************************************
fatal: [ubuntu1]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'ansible_user' is undefin
fatal: [ubuntu2]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'ansible_user' is undefin
fatal: [ubuntu3]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'ansible_user' is undefin

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

```
$ cd ../14

$ ansible-playbook variables_playbook.yaml

PLAY [ubuntu] ******************************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************************
ok: [ubuntu2]
ok: [ubuntu3]
ok: [ubuntu1]

TASK [Test groupvars] **********************************************************************************************************************************
fatal: [ubuntu1]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'ansible_user' is undefin
fatal: [ubuntu2]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'ansible_user' is undefin
fatal: [ubuntu3]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'ansible_user' is undefin

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

$ ansible-playbook variables_playbook.yaml

PLAY [centos1] *****************************************************************************************************************************************

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

TASK [Test groupvars with an ansible fact, show that the variable is also accessible from the hostvars section] ***********************************
ok: [centos1] => {
    "msg": "root"
}

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

```
$ cd ../15

$ cat variables_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: centos1
  gather_facts: True

  # Vars: variables that will apply to the play, on all target systems

  # Tasks: the list of tasks that will be executed within the playbook
  tasks:
    - name: Test hostvars with an ansible fact and collect ansible_port, dot notation
      debug:
        msg: "{{ hostvars[ansible_hostname].ansible_port }}"

    - name: Test groupvars
      debug:
        msg: "{{ ansible_user }}"

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

$ ansible-playbook variables_playbook.yaml

PLAY [centos1] *****************************************************************************************************************************************

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

TASK [Test hostvars with an ansible fact and collect ansible_port, dot notation] ***********************************************************************
ok: [centos1] => {
    "msg": "2222"
}

TASK [Test groupvars] **********************************************************************************************************************************
ok: [centos1] => {
    "msg": "root"
}

PLAY RECAP *********************************************************************************************************************************************
centos1                    : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
########################################################################
# using extra vars in ini format
########################################################################

$ cd ../16

$ ansible-playbook variables_playbook.yaml -e extra_vars_key="'extra vars value'"

PLAY [centos1] *****************************************************************************************************************************************

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

TASK [Test extra vars] *********************************************************************************************************************************
ok: [centos1] => {
    "msg": "extra vars value"
}

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

########################################################################
# using extra vars in json format
########################################################################
$ cd ../17

$ ansible-playbook variables_playbook.yaml -e '{"extra_vars_key": "extra vars value"}'

PLAY [centos1] *****************************************************************************************************************************************
```

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

TASK [Test extra vars] *********************************************************************************************************************
ok: [centos1] => {
    "msg": "extra vars value"
}

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

###########################################################################
# using extra vars in yaml format
###########################################################################
$ cd ../16

$ ansible-playbook variables_playbook.yaml -e '{extra_vars_key: extra vars value}'

PLAY [centos1] *****************************************************************************************************************************

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

TASK [Test extra vars] *********************************************************************************************************************
ok: [centos1] => {
    "msg": "extra vars value"
}

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

###########################################################################
# passing variables as a yaml file
###########################################################################
$ cd ../17

$ ansible-playbook variables_playbook.yaml -e @extra_vars_file.yaml

PLAY [centos1] *****************************************************************************************************************************

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

TASK [Test extra vars] *********************************************************************************************************************
ok: [centos1] => {
    "msg": "extra vars value"
}

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

$ cat extra_vars_file.yaml
---
extra_vars_key: extra vars value
...

###########################################################################
# passing extra vars as a json file
###########################################################################
$ cd ../17

$ ansible-playbook variables_playbook.yaml -e @extra_vars_file.json

PLAY [centos1] *****************************************************************************************************************************

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

TASK [Test extra vars] *********************************************************************************************************************
ok: [centos1] => {
    "msg": "extra vars value"
}

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

$ cat extra_vars_file.json
{
    "extra_vars_key": "extra vars value"
}
```

Hostvars and Groupvars

Using Directories for Hostvars and Groupvars

- It is also possible to use directories for host and group variables in combination with a Yaml file
- See revisions 15,16 & 17 folders where the existing variables have been moved into host_vars and group_vars folders
- Hostvars use the directory structure -

  host_vars/hostname i.e. host_vars/ubuntu-c
- Groupvars use the directory structure -

  group_vars/group i.e. group_vars/ubuntu

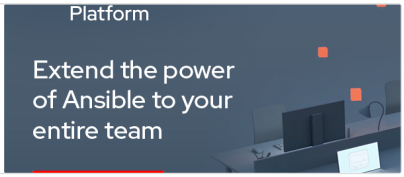
```
$ cd ../17

$ ls -l
total 20
-rwxr-xr-x 1 ansible ansible  55 Jun 24 09:08 ansible.cfg
-rwxr-xr-x 1 ansible ansible  44 Jun 24 09:08 extra_vars_file.json
-rwxr-xr-x 1 ansible ansible  41 Jun 24 09:08 extra_vars_file.yaml
drwxr-xr-x 4 ansible ansible 128 Jun 24 09:08 group_vars
drwxr-xr-x 4 ansible ansible 128 Jun 24 09:08 host_vars
-rwxr-xr-x 1 ansible ansible  95 Jun 24 09:08 hosts
-rwxr-xr-x 1 ansible ansible 555 Jun 24 09:08 variables_playbook.yaml

##############################################################################
# host_vars
##############################################################################

$ cd host_vars/
$ ls -l
total 8
-rwxr-xr-x 1 ansible ansible 27 Jun 24 09:08 centos1
-rwxr-xr-x 1 ansible ansible 34 Jun 24 09:08 ubuntu-c

$ cat ubuntu-c
---
ansible_connection: local
...
$ cat centos1
---
ansible_port: 2222
...

##############################################################################
# group_vars
##############################################################################

$ cd ../group_vars/
$ ls -l
total 8
-rwxr-xr-x 1 ansible ansible 27 Jun 24 09:08 centos
-rwxr-xr-x 1 ansible ansible 59 Jun 24 09:08 ubuntu

$ cat centos
---
ansible_user: root
...
$ cat ubuntu
---
ansible_become: true
ansible_become_pass: password
...
```

4. Ansible Playbooks- Facts

ansible.builtin.setup module - Gathers facts about remote hosts - Ansible Documentation

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

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




```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Introduction/Ansible Playbooks, Facts/01

$ ansible centos1 -m setup -a 'gather_subset=network'

$ ansible centos1 -m setup -a 'gather_subset=!all,!min,network'

$ ansible centos1 -m setup -a 'gather_subset=network' | wc -l
527
$ ansible centos1 -m setup -a 'gather_subset=!all,!min,network' | wc -l
356

$ ansible centos1 -m setup -a 'filter=ansible_memfree_mb'
centos1 | SUCCESS => {
    "ansible_facts": {
        "ansible_memfree_mb": 172,
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false
}

$ ansible centos1 -m setup -a 'filter=ansible_mem*'
centos1 | SUCCESS => {
    "ansible_facts": {
        "ansible_memfree_mb": 168,
        "ansible_memory_mb": {
            "nocache": {
                "free": 2345,
                "used": 1589
            },
            "real": {
                "free": 168,
                "total": 3934,
                "used": 3766
            },
            "swap": {
                "cached": 0,
                "free": 1023,
                "total": 1023,
                "used": 0
```

```
            }
        },
        "ansible_memtotal_mb": 3934,
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false
}

# 결괏값은 ansible_facts(딕셔너리 키)가 facts 네임스페이스의 루트에 추가된다.
$ ansible centos1 -m setup -a 'gather_subset=!all,!min,network'
centos1 | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "172.19.0.5"
        ],
        "ansible_all_ipv6_addresses": [],
        "ansible_architecture": "x86_64",
        "ansible_default_ipv4": {
            "address": "172.19.0.5",
            "alias": "eth0",
            "broadcast": "172.19.255.255",
            "gateway": "172.19.0.1",
            "interface": "eth0",
            "macaddress": "02:42:ac:13:00:05",
            "mtu": 1500,
            "netmask": "255.255.0.0",
            "network": "172.19.0.0",
            "type": "ether"
        },
        "ansible_default_ipv6": {},
        "ansible_distribution": "CentOS",
        "ansible_distribution_file_parsed": true,
        "ansible_distribution_file_path": "/etc/redhat-release",
        "ansible_distribution_file_variety": "RedHat",
        "ansible_distribution_major_version": "8",
        "ansible_distribution_release": "NA",
        "ansible_distribution_version": "8.4",
        "ansible_domain": "",
```

```
ansible@ubuntu-c:~/diveintoansible/Ansible Playbooks, Introduction/Ansible Playbooks, Facts/01$ cat facts_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: Show IP Address
      debug:
        msg: "{{ ansible_default_ipv4.address }}"

# Three dots indicate the end of a YAML document
...
ansible@ubuntu-c:~/diveintoansible/Ansible Playbooks, Introduction/Ansible Playbooks, Facts/01$
```

```
"ansible_facts": {
    "ansible_all_ipv4_addresses": [
        "172.18.0.5"
    ],
    "ansible_all_ipv6_addresses": [],
    "ansible_architecture": "x86_64",
    "ansible_default_ipv4": {
        "address": "172.18.0.5",
        "alias": "eth0",
        "broadcast": "172.18.255.255",
        "gateway": "172.18.0.1",
        "interface": "eth0",
        "macaddress": "02:42:ac:12:00:05",
        "mtu": 1500,
        "netmask": "255.255.0.0",
        "network": "172.18.0.0",
        "type": "ether"
    },
```

ansible_default_ipv4.address 키는 ansible_facts 딕셔너리에 포함되지만 ansible_default_ipv4.address 를 사용하기위해 부모 딕셔너리(ansible_facts)를 명시하지 않아도 된다.

```
$ cat facts_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: Show IP Address
      debug:
        msg: "{{ ansible_default_ipv4.address }}"

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

```
$ ansible-playbook facts_playbook.yaml

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

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

TASK [Show IP Address] *********************************************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "172.19.0.3"
}
ok: [centos1] => {
    "msg": "172.19.0.5"
}
ok: [centos2] => {
    "msg": "172.19.0.6"
}
ok: [centos3] => {
    "msg": "172.19.0.7"
}
ok: [ubuntu1] => {
    "msg": "172.19.0.4"
}
ok: [ubuntu2] => {
    "msg": "172.19.0.8"
}
ok: [ubuntu3] => {
    "msg": "172.19.0.9"
}

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
ubuntu-c                   : 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
```

Custom Facts

- Can be written in any language
- Returns a JSON structure
- (or) Returns an ini structure
- By default, expects to use /etc/ansible/facts.d

(*)default 경로인 /etc/ansible/facts.d 를 다른 경로로 바꿀 수 있다.

```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Introduction/Ansible Playbooks, Facts/02/templates

$ ls -l
total 8
-rwxr-xr-x 1 ansible ansible 45 Jun 24 09:08 getdate1.fact
-rwxr-xr-x 1 ansible ansible 41 Jun 24 09:08 getdate2.fact

# json format
```

```
$ cat getdate1.fact
#!/bin/bash
echo {\""date\"" : \""`date`\""}

$ ./getdate1.fact
{"date" : "Tue Jun 28 08:14:34 UTC 2022"}

# ini format
# [date] 로 시작하는 카테고리는 반드시 필요하다(json의 키에 대응).
$ cat getdate2.fact
#!/bin/bash
echo [date]
echo date=`date`

$ ./getdate2.fact
[date]
date=Tue Jun 28 08:16:08 UTC 2022
```

Custom Facts

- JSON Output -

  {"date" : "Day Mon DD HH:MM:SS TZ YYYY"}

- INI Output -

  [date]
  date=Day Mon DD HH:MM:SS TZ YYYY

/etc/ansible/facts.d 디렉토리 테스트

```
$ sudo mkdir -p /etc/ansible/facts.d
$ sudo cp * /etc/ansible/facts.d/
$ cd ..
$ ansible ubuntu-c -m setup | more
...
        "ansible_local": {
            "getdate1": {
                "date": "Tue Jun 28 08:20:00 UTC 2022"
            },
            "getdate2": {
                "date": {
                    "date": "Tue Jun 28 08:20:00 UTC 2022"
                }
            }
        },
...

# filter로 조회
$ ansible ubuntu-c -m setup -a 'filter=ansible_local'
ubuntu-c | SUCCESS => {
    "ansible_facts": {
        "ansible_local": {
            "getdate1": {
                "date": "Tue Jun 28 08:22:46 UTC 2022"
            },
            "getdate2": {
                "date": {
                    "date": "Tue Jun 28 08:22:46 UTC 2022"
                }
            }
        },
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false
}

##########################################################################
```

```
#
# Custom Facts를 변수로 사용하기
#

$ cd ../03
$ cat facts_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: Show IP Address
      debug:
        msg: "{{ ansible_default_ipv4.address }}"

    - name: Show Custom Fact 1
      debug:
        msg: "{{ ansible_local.getdate1.date }}"

    - name: Show Custom Fact 2
      debug:
        msg: "{{ ansible_local.getdate2.date.date }}"

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

# /etc/ansible/facts.d가 적용된 호스트는 ubuntu-c이므로 limit(-l)옵션으로
# ubuntu-c 호스트를 지정한다.
$ ansible-playbook facts_playbook.yaml -l ubuntu-c

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

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

TASK [Show IP Address] *********************************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "172.19.0.3"
}

TASK [Show Custom Fact 1] ******************************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "Tue Jun 28 08:26:32 UTC 2022"
}

TASK [Show Custom Fact 2] ******************************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "Tue Jun 28 08:26:32 UTC 2022"
}

PLAY RECAP *********************************************************************************************************************************
ubuntu-c                   : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

########################################################################
#
# hostvars 에서 사용하기
#

$ cd ../04/
$ cat facts_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: Show IP Address
      debug:
        msg: "{{ ansible_default_ipv4.address }}"

    - name: Show Custom Fact 1
      debug:
```

```
        msg: "{{ ansible_local.getdate1.date }}"

    - name: Show Custom Fact 2
      debug:
        msg: "{{ ansible_local.getdate2.date.date }}"

    - name: Show Custom Fact 1 in hostvars
      debug:
        msg: "{{ hostvars[ansible_hostname].ansible_local.getdate1.date }}"

    - name: Show Custom Fact 2 in hostvars
      debug:
        msg: "{{ hostvars[ansible_hostname].ansible_local.getdate2.date.date }}"

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

$ ansible-playbook facts_playbook.yaml -l ubuntu-c

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

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

TASK [Show IP Address] ********************************************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "172.19.0.3"
}

TASK [Show Custom Fact 1] *****************************************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "Tue Jun 28 08:30:49 UTC 2022"
}

TASK [Show Custom Fact 2] *****************************************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "Tue Jun 28 08:30:49 UTC 2022"
}

TASK [Show Custom Fact 1 in hostvars] *****************************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "Tue Jun 28 08:30:49 UTC 2022"
}

TASK [Show Custom Fact 2 in hostvars] *****************************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "Tue Jun 28 08:30:49 UTC 2022"
}

PLAY RECAP ********************************************************************************************************************************************
ubuntu-c                   : ok=6    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

#########################################################################
#
# /etc/ansible/facts.d 를 모든 호스트에 적용하기
#

$ cd ../05
$ cat facts_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: Make Facts Dir
      file:
        path: /etc/ansible/facts.d
        recurse: yes
        state: directory

    - name: Copy Fact 1
      copy:
        src: /etc/ansible/facts.d/getdate1.fact
        dest: /etc/ansible/facts.d/getdate1.fact
        mode: 0755

    - name: Copy Fact 2
      copy:
```

```
        src: /etc/ansible/facts.d/getdate2.fact
        dest: /etc/ansible/facts.d/getdate2.fact
        mode: 0755

    - name: Refresh Facts
      # refresh facts
      setup:

    - name: Show IP Address
      debug:
        msg: "{{ ansible_default_ipv4.address }}"

    - name: Show Custom Fact 1
      debug:
        msg: "{{ ansible_local.getdate1.date }}"

    - name: Show Custom Fact 2
      debug:
        msg: "{{ ansible_local.getdate2.date.date }}"

    - name: Show Custom Fact 1 in hostvars
      debug:
        msg: "{{ hostvars[ansible_hostname].ansible_local.getdate1.date }}"

    - name: Show Custom Fact 2 in hostvars
      debug:
        msg: "{{hostvars[ansible_hostname].ansible_local.getdate2.date.date }}"

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

$ ansible-playbook facts_playbook.yaml
```

/etc/ansible/facts.d 변경 테스트

```
# /etc/ansible/facts.d 존재 여부 확인

$ ansible linux -m file -a 'path=/etc/ansible/facts.d/getdate1.fact state=absent'
$ ansible linux -m file -a 'path=/etc/ansible/facts.d/getdate2.fact state=absent'

$ cd ../06
$ ls
ansible.cfg  facts.d  facts_playbook.yaml  group_vars  host_vars  hosts
$ ls facts.d/
getdate1.fact  getdate2.fact

$ cat facts_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: Make Facts Dir
      file:
        path: /home/ansible/facts.d
        recurse: yes
        state: directory
        owner: ansible

    - name: Copy Fact 1
      copy:
        src: facts.d/getdate1.fact
        dest: /home/ansible/facts.d/getdate1.fact
        owner: ansible
        mode: 0755

    - name: Copy Fact 2
      copy:
        src: facts.d/getdate2.fact
        dest: /home/ansible/facts.d/getdate2.fact
        owner: ansible
        mode: 0755
    # 디폴트 경로가 아닌 facts를 인식할 수 있도록
    # setup 모듈에 fact_path 인자를 추가
    - name: Reload Facts
```

```
      setup:
        fact_path: /home/ansible/facts.d

    - name: Show IP Address
      debug:
        msg: "{{ ansible_default_ipv4.address }}"

    - name: Show Custom Fact 1
      debug:
        msg: "{{ ansible_local.getdate1.date }}"

    - name: Show Custom Fact 2
      debug:
        msg: "{{ ansible_local.getdate2.date.date }}"

    - name: Show Custom Fact 1 in hostvars
      debug:
        msg: "{{ hostvars[ansible_hostname].ansible_local.getdate1.date }}"

    - name: Show Custom Fact 2 in hostvars
      debug:
        msg: "{{hostvars[ansible_hostname].ansible_local.getdate2.date.date }}"

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

$ ansible-playbook facts_playbook.yaml

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

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

TASK [Make Facts Dir] **************************************************************************************************************************
changed: [centos3]
changed: [ubuntu1]
changed: [centos2]
changed: [centos1]
changed: [ubuntu2]
changed: [ubuntu3]

TASK [Copy Fact 1] *****************************************************************************************************************************
changed: [centos2]
changed: [ubuntu2]
changed: [centos1]
changed: [ubuntu1]
changed: [centos3]
changed: [ubuntu3]

TASK [Copy Fact 2] *****************************************************************************************************************************
changed: [centos3]
changed: [centos1]
changed: [centos2]
changed: [ubuntu2]
changed: [ubuntu1]
changed: [ubuntu3]

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

TASK [Show IP Address] *************************************************************************************************************************
ok: [centos1] => {
    "msg": "172.19.0.5"
}
ok: [centos2] => {
    "msg": "172.19.0.6"
}
ok: [centos3] => {
    "msg": "172.19.0.7"
}
ok: [ubuntu1] => {
    "msg": "172.19.0.4"
}
ok: [ubuntu2] => {
    "msg": "172.19.0.8"
}
ok: [ubuntu3] => {
```

```
    "msg": "172.19.0.9"
}

TASK [Show Custom Fact 1] ******************************************************************************************************************
ok: [centos1] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [centos2] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [centos3] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [ubuntu1] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [ubuntu2] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [ubuntu3] => {
    "msg": "Tue Jun 28 08:48:28 UTC 2022"
}

TASK [Show Custom Fact 2] ******************************************************************************************************************
ok: [centos1] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [centos2] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [centos3] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [ubuntu1] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [ubuntu2] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [ubuntu3] => {
    "msg": "Tue Jun 28 08:48:28 UTC 2022"
}

TASK [Show Custom Fact 1 in hostvars] ******************************************************************************************************
ok: [centos1] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [centos2] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [centos3] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [ubuntu1] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [ubuntu2] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [ubuntu3] => {
    "msg": "Tue Jun 28 08:48:28 UTC 2022"
}

TASK [Show Custom Fact 2 in hostvars] ******************************************************************************************************
ok: [centos1] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [centos2] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [centos3] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [ubuntu1] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [ubuntu2] => {
    "msg": "Tue Jun 28 08:48:27 UTC 2022"
}
ok: [ubuntu3] => {
    "msg": "Tue Jun 28 08:48:28 UTC 2022"
}

PLAY RECAP *********************************************************************************************************************************
centos1                    : ok=10   changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=10   changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=10   changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
ubuntu1                    : ok=10   changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu2                    : ok=10   changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3                    : ok=10   changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

# /home/ansible/facts.d 디렉토리 삭제
# state=abset 파라미터 주의!
$ ansible linux -m file -a 'path=/home/ansible/facts.d state=absent'
```

5. Templating with Jinja2

5-1. The Jinja2 Template Language

- double currly braces syntax

5-2. if/elif/else statements

- `{# … -#}` : comments
- `{% … -%}` : statements
- `{% … %}` : end of statements
- `{% … is defind -%}` statements

```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Introduction/Templating with Jinja2/01

################################################################################
#
# if statements
#

$ cat jinja2_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: Ansible Jinja2 if
      debug:
        msg: >
             --== Ansible Jinja2 if statement ==--

             {# If the hostname is ubuntu-c, include a message -#}
```

```
             {% if ansible_hostname == "ubuntu-c" -%}
                  This is ubuntu-c
             {% endif %}

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


$ ansible-playbook jinja2_playbook.yaml

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

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

TASK [Ansible Jinja2 if] *******************************************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "--== Ansible Jinja2 if statement ==--\nThis is ubuntu-c\n"
}
ok: [centos1] => {
    "msg": "--== Ansible Jinja2 if statement ==--\n"
}
ok: [centos2] => {
    "msg": "--== Ansible Jinja2 if statement ==--\n"
}
ok: [centos3] => {
    "msg": "--== Ansible Jinja2 if statement ==--\n"
}
ok: [ubuntu1] => {
    "msg": "--== Ansible Jinja2 if statement ==--\n"
}
ok: [ubuntu2] => {
    "msg": "--== Ansible Jinja2 if statement ==--\n"
}
ok: [ubuntu3] => {
    "msg": "--== Ansible Jinja2 if statement ==--\n"
}

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
ubuntu-c                   : 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

#########################################################################
#
# if/elif statements
#

$ cd ../02
$ cat jinja2_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: Ansible Jinja2 if elif
      debug:
        msg: >
             --== Ansible Jinja2 if elif statement ==--

             {% if ansible_hostname == "ubuntu-c" -%}
                This is ubuntu-c
             {% elif ansible_hostname == "centos1" -%}
                This is centos1 with it's modified SSH Port
             {% endif %}

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

```
$ ansible-playbook jinja2_playbook.yaml

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

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

TASK [Ansible Jinja2 if elif] **************************************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "--== Ansible Jinja2 if elif statement ==--\nThis is ubuntu-c\n"
}
ok: [centos1] => {
    "msg": "--== Ansible Jinja2 if elif statement ==--\nThis is centos1 with it's modified SSH Port\n"
}
ok: [centos2] => {
    "msg": "--== Ansible Jinja2 if elif statement ==--\n"
}
ok: [centos3] => {
    "msg": "--== Ansible Jinja2 if elif statement ==--\n"
}
ok: [ubuntu1] => {
    "msg": "--== Ansible Jinja2 if elif statement ==--\n"
}
ok: [ubuntu2] => {
    "msg": "--== Ansible Jinja2 if elif statement ==--\n"
}
ok: [ubuntu3] => {
    "msg": "--== Ansible Jinja2 if elif statement ==--\n"
}

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
ubuntu-c                   : 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

#########################################################################
#
# if/elif/else statements
#

$ cd ../03
$ cat jinja2_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: Ansible Jinja2 if elif else
      debug:
        msg: >
             --== Ansible Jinja2 if elif else statement ==--

             {% if ansible_hostname == "ubuntu-c" -%}
                This is ubuntu-c
             {% elif ansible_hostname == "centos1" -%}
                This is centos1 with it's modified SSH Port
             {% else -%}
                This is good old {{ ansible_hostname }}
             {% endif %}

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

$ ansible-playbook jinja2_playbook.yaml

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

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

```
ok: [ubuntu1]
ok: [centos1]
ok: [centos2]
ok: [ubuntu-c]
ok: [ubuntu2]
ok: [ubuntu3]

TASK [Ansible Jinja2 if elif else] *********************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "--== Ansible Jinja2 if elif else statement ==--\nThis is ubuntu-c\n"
}
ok: [centos1] => {
    "msg": "--== Ansible Jinja2 if elif else statement ==--\nThis is centos1 with it's modified SSH Port\n"
}
ok: [centos2] => {
    "msg": "--== Ansible Jinja2 if elif else statement ==--\nThis is good old centos2\n"
}
ok: [centos3] => {
    "msg": "--== Ansible Jinja2 if elif else statement ==--\nThis is good old centos3\n"
}
ok: [ubuntu1] => {
    "msg": "--== Ansible Jinja2 if elif else statement ==--\nThis is good old ubuntu1\n"
}
ok: [ubuntu2] => {
    "msg": "--== Ansible Jinja2 if elif else statement ==--\nThis is good old ubuntu2\n"
}
ok: [ubuntu3] => {
    "msg": "--== Ansible Jinja2 if elif else statement ==--\nThis is good old ubuntu3\n"
}

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
ubuntu-c                   : 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

#########################################################################
#
# if is defined
#

$ cd ../04
$ cat jinja2_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: Ansible Jinja2 if variable is defined ( where variable is not defined )
      debug:
        msg: >
             --== Ansible Jinja2 if variable is defined ( where variable is not defined ) ==--

             {% if example_variable is defined -%}
                example_variable is defined
             {% else -%}
                example_variable is not defined
             {% endif %}

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

$ ansible-playbook jinja2_playbook.yaml

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

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

TASK [Ansible Jinja2 if variable is defined ( where variable is not defined )] *************************************************************
```

```
ok: [ubuntu-c] => {
    "msg": "--== Ansible Jinja2 if variable is defined ( where variable is not defined ) ==--\nexample_variable is not defined\n"
}
ok: [centos1] => {
    "msg": "--== Ansible Jinja2 if variable is defined ( where variable is not defined ) ==--\nexample_variable is not defined\n"
}
ok: [centos2] => {
    "msg": "--== Ansible Jinja2 if variable is defined ( where variable is not defined ) ==--\nexample_variable is not defined\n"
}
ok: [centos3] => {
    "msg": "--== Ansible Jinja2 if variable is defined ( where variable is not defined ) ==--\nexample_variable is not defined\n"
}
ok: [ubuntu1] => {
    "msg": "--== Ansible Jinja2 if variable is defined ( where variable is not defined ) ==--\nexample_variable is not defined\n"
}
ok: [ubuntu2] => {
    "msg": "--== Ansible Jinja2 if variable is defined ( where variable is not defined ) ==--\nexample_variable is not defined\n"
}
ok: [ubuntu3] => {
    "msg": "--== Ansible Jinja2 if variable is defined ( where variable is not defined ) ==--\nexample_variable is not defined\n"
}

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
ubuntu-c                   : 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

#####################################################################
#
# variable set
#

$ cd ../05/
$ cat jinja2_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: Ansible Jinja2 if variable is defined ( where variable is defined )
      debug:
        msg: >
             --== Ansible Jinja2 if variable is defined ( where variable is defined ) ==--

             {% set example_variable = 'defined' -%}
             {% if example_variable is defined -%}
                example_variable is defined
             {% else -%}
                example_variable is not defined
             {% endif %}

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

$ ansible-playbook jinja2_playbook.yaml

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

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

TASK [Ansible Jinja2 if variable is defined ( where variable is defined )] *****************************************************
ok: [ubuntu-c] => {
    "msg": "--== Ansible Jinja2 if variable is defined ( where variable is defined ) ==--\nexample_variable is defined\n"
}
ok: [centos1] => {
    "msg": "--== Ansible Jinja2 if variable is defined ( where variable is defined ) ==--\nexample_variable is defined\n"
}
ok: [centos2] => {
```

```
    "msg": "--== Ansible Jinja2 if variable is defined ( where variable is defined ) ==--\nexample_variable is defined\n"
}
ok: [centos3] => {
    "msg": "--== Ansible Jinja2 if variable is defined ( where variable is defined ) ==--\nexample_variable is defined\n"
}
ok: [ubuntu1] => {
    "msg": "--== Ansible Jinja2 if variable is defined ( where variable is defined ) ==--\nexample_variable is defined\n"
}
ok: [ubuntu2] => {
    "msg": "--== Ansible Jinja2 if variable is defined ( where variable is defined ) ==--\nexample_variable is defined\n"
}
ok: [ubuntu3] => {
    "msg": "--== Ansible Jinja2 if variable is defined ( where variable is defined ) ==--\nexample_variable is defined\n"
}

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
ubuntu-c                   : 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
```

5-3. for loops

```
$ cd ../06
$ cat jinja2_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: Ansible Jinja2 for statement
      debug:
        msg: >
             --== Ansible Jinja2 for statement ==--

             {% for entry in ansible_interfaces -%}
                Interface entry {{ loop.index }} = {{ entry }}
             {% endfor %}

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

$ ansible-playbook jinja2_playbook.yaml

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

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

TASK [Ansible Jinja2 for statement] ********************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "--== Ansible Jinja2 for statement ==--\nInterface entry 1 = ip6tnl0\nInterface entry 2 = tunl0\nInterface entry 3 = lo\nIn
}
ok: [centos1] => {
    "msg": "--== Ansible Jinja2 for statement ==--\nInterface entry 1 = lo\nInterface entry 2 = tunl0\nInterface entry 3 = ip6tnl0\nIn
}
ok: [centos2] => {
    "msg": "--== Ansible Jinja2 for statement ==--\nInterface entry 1 = tunl0\nInterface entry 2 = lo\nInterface entry 3 = ip6tnl0\nIn
}
ok: [centos3] => {
    "msg": "--== Ansible Jinja2 for statement ==--\nInterface entry 1 = lo\nInterface entry 2 = eth0\nInterface entry 3 = tunl0\nInter
}
ok: [ubuntu1] => {
    "msg": "--== Ansible Jinja2 for statement ==--\nInterface entry 1 = eth0\nInterface entry 2 = ip6tnl0\nInterface entry 3 = tunl0\n
}
ok: [ubuntu2] => {
```

```
    "msg": "--== Ansible Jinja2 for statement ==--\nInterface entry 1 = ip6tnl0\nInterface entry 2 = tunl0\nInterface entry 3 = lo\nIn
}
ok: [ubuntu3] => {
    "msg": "--== Ansible Jinja2 for statement ==--\nInterface entry 1 = lo\nInterface entry 2 = tunl0\nInterface entry 3 = ip6tnl0\nIn
}

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
ubuntu-c                   : 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 ../07
$ cat jinja2_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: Ansible Jinja2 for range
      debug:
        msg: >
             --== Ansible Jinja2 for range

             {% for entry in range(1, 11) -%}
                {{ entry }}
             {% endfor %}

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

$ ansible-playbook jinja2_playbook.yaml

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

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

TASK [Ansible Jinja2 for range] ************************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "--== Ansible Jinja2 for range\n1\n2\n3\n4\n5\n6\n7\n8\n9\n10\n"
}
ok: [centos1] => {
    "msg": "--== Ansible Jinja2 for range\n1\n2\n3\n4\n5\n6\n7\n8\n9\n10\n"
}
ok: [centos2] => {
    "msg": "--== Ansible Jinja2 for range\n1\n2\n3\n4\n5\n6\n7\n8\n9\n10\n"
}
ok: [centos3] => {
    "msg": "--== Ansible Jinja2 for range\n1\n2\n3\n4\n5\n6\n7\n8\n9\n10\n"
}
ok: [ubuntu1] => {
    "msg": "--== Ansible Jinja2 for range\n1\n2\n3\n4\n5\n6\n7\n8\n9\n10\n"
}
ok: [ubuntu2] => {
    "msg": "--== Ansible Jinja2 for range\n1\n2\n3\n4\n5\n6\n7\n8\n9\n10\n"
}
ok: [ubuntu3] => {
    "msg": "--== Ansible Jinja2 for range\n1\n2\n3\n4\n5\n6\n7\n8\n9\n10\n"
}

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
ubuntu-c                   : 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
```

5-4. break and continue

- jinja2 extension 필요 : `jinja2.ext.loopcontrols`

```
$ cd ../08/
$ cat jinja2_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: Ansible Jinja2 for range, reversed (simulate while greater 5)
      debug:
        msg: >
             --== Ansible Jinja2 for range, reversed (simulate while greater 5) ==--

             {% for entry in range(10, 0, -1) -%}
                {% if entry == 5 -%}
                   {% break %}
                {% endif -%}
                {{ entry }}
             {% endfor %}

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

$ ansible-playbook jinja2_playbook.yaml

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

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

TASK [Ansible Jinja2 for range, reversed (simulate while greater 5)] ***********************************************************************************
ok: [ubuntu-c] => {
    "msg": "--== Ansible Jinja2 for range, reversed (simulate while greater 5) ==--\n10\n9\n8\n7\n6\n"
}
ok: [centos1] => {
    "msg": "--== Ansible Jinja2 for range, reversed (simulate while greater 5) ==--\n10\n9\n8\n7\n6\n"
}
ok: [centos2] => {
    "msg": "--== Ansible Jinja2 for range, reversed (simulate while greater 5) ==--\n10\n9\n8\n7\n6\n"
}
ok: [centos3] => {
    "msg": "--== Ansible Jinja2 for range, reversed (simulate while greater 5) ==--\n10\n9\n8\n7\n6\n"
}
ok: [ubuntu1] => {
    "msg": "--== Ansible Jinja2 for range, reversed (simulate while greater 5) ==--\n10\n9\n8\n7\n6\n"
}
ok: [ubuntu2] => {
    "msg": "--== Ansible Jinja2 for range, reversed (simulate while greater 5) ==--\n10\n9\n8\n7\n6\n"
}
ok: [ubuntu3] => {
    "msg": "--== Ansible Jinja2 for range, reversed (simulate while greater 5) ==--\n10\n9\n8\n7\n6\n"
}

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
ubuntu-c                   : 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
```

ansible.cfg

```
[defaults]
inventory = hosts
host_key_checking = False
jinja2_extensions = jinja2.ext.loopcontrols
```

continue

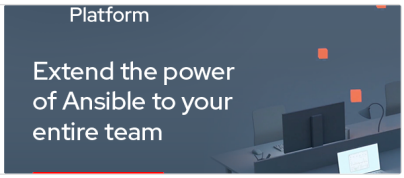
```
$ cd ../09/
$ cat jinja2_playbook.yaml
---
# YAML documents begin with the document separator ---

# 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: Ansible Jinja2 for range, reversed (continue if odd)
      debug:
        msg: >
             --== Ansible Jinja2 for range, reversed (continue if odd) ==--

             {% for entry in range(10, 0, -1) -%}
                {% if entry is odd -%}
                   {% continue %}
                {% endif -%}
                {{ entry }}
             {% endfor %}

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

$ cat ansible.cfg
[defaults]
inventory = hosts
host_key_checking = False
jinja2_extensions = jinja2.ext.loopcontrols

$ ansible-playbook jinja2_playbook.yaml

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

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

TASK [Ansible Jinja2 for range, reversed (continue if odd)] ********************************************************************************************
ok: [ubuntu-c] => {
    "msg": "--== Ansible Jinja2 for range, reversed (continue if odd) ==--\n10\n8\n6\n4\n2\n"
}
ok: [centos1] => {
    "msg": "--== Ansible Jinja2 for range, reversed (continue if odd) ==--\n10\n8\n6\n4\n2\n"
}
ok: [centos2] => {
    "msg": "--== Ansible Jinja2 for range, reversed (continue if odd) ==--\n10\n8\n6\n4\n2\n"
}
ok: [centos3] => {
    "msg": "--== Ansible Jinja2 for range, reversed (continue if odd) ==--\n10\n8\n6\n4\n2\n"
}
ok: [ubuntu1] => {
    "msg": "--== Ansible Jinja2 for range, reversed (continue if odd) ==--\n10\n8\n6\n4\n2\n"
}
ok: [ubuntu2] => {
    "msg": "--== Ansible Jinja2 for range, reversed (continue if odd) ==--\n10\n8\n6\n4\n2\n"
}
ok: [ubuntu3] => {
    "msg": "--== Ansible Jinja2 for range, reversed (continue if odd) ==--\n10\n8\n6\n4\n2\n"
}

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
```

```
ubuntu-c                   : 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
```

Using filters to manipulate data - Ansible Documentation

Filters let you transform JSON data into YAML data, split a URL to extract the hostname, get the SHA1 hash of a string, add or multiply integers, and much more. You can use the Ansible-specific filters documented here to manipulate your data, or use any of the standard filters shipped with Jinja2 - see the list of built-in

https://docs.ansible.com/ansible/latest/user_guide/playbooks_filters.html




```
$ cd ../10
$ cat jinja2_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: Ansible Jinja2 filters
      debug:
        msg: >
             ---=== Ansible Jinja2 filters ===---

             --== min [1, 2, 3, 4, 5] ==--

             {{ [1, 2, 3, 4, 5] | min }}

             --== max [1, 2, 3, 4, 5] ==--

             {{ [1, 2, 3, 4, 5] | max }}

             --== unique [1, 1, 2, 2, 3, 3, 4, 4, 5, 5] ==--

             {{ [1, 1, 2, 2, 3, 3, 4, 4, 5, 5] | unique }}

             --== difference [1, 2, 3, 4, 5] vs [2, 3, 4] ==--

             {{ [1, 2, 3, 4, 5] | difference([2, 3, 4]) }}

             --== random ['rod', 'jane', 'freddy'] ==--

             {{ ['rod', 'jane', 'freddy'] | random }}

             --== urlsplit hostname ==--

             {{ "http://docs.ansible.com/ansible/latest/playbooks_filters.html" | urlsplit('hostname') }}

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

$ cat ansible.cfg
[defaults]
inventory = hosts
host_key_checking = False
jinja2_extensions = jinja2.ext.loopcontrols

$ ansible-playbook jinja2_playbook.yaml

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

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

TASK [Ansible Jinja2 filters] **************************************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "---=== Ansible Jinja2 filters ===---\n--== min [1, 2, 3, 4, 5] ==--\n1\n--== max [1, 2, 3, 4, 5] ==--\n5\n--== unique [1,
}
```

```
ok: [centos1] => {
    "msg": "---=== Ansible Jinja2 filters ===---\n--== min [1, 2, 3, 4, 5] ==--\n1\n--== max [1, 2, 3, 4, 5] ==--\n5\n--== unique [1,
}
ok: [centos2] => {
    "msg": "---=== Ansible Jinja2 filters ===---\n--== min [1, 2, 3, 4, 5] ==--\n1\n--== max [1, 2, 3, 4, 5] ==--\n5\n--== unique [1,
}
ok: [centos3] => {
    "msg": "---=== Ansible Jinja2 filters ===---\n--== min [1, 2, 3, 4, 5] ==--\n1\n--== max [1, 2, 3, 4, 5] ==--\n5\n--== unique [1,
}
ok: [ubuntu1] => {
    "msg": "---=== Ansible Jinja2 filters ===---\n--== min [1, 2, 3, 4, 5] ==--\n1\n--== max [1, 2, 3, 4, 5] ==--\n5\n--== unique [1,
}
ok: [ubuntu2] => {
    "msg": "---=== Ansible Jinja2 filters ===---\n--== min [1, 2, 3, 4, 5] ==--\n1\n--== max [1, 2, 3, 4, 5] ==--\n5\n--== unique [1,
}
ok: [ubuntu3] => {
    "msg": "---=== Ansible Jinja2 filters ===---\n--== min [1, 2, 3, 4, 5] ==--\n1\n--== max [1, 2, 3, 4, 5] ==--\n5\n--== unique [1,
}

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
ubuntu-c                   : 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

#########################################################################
#
# template 모듈 사용
#

$ cd ../11
$ cat jinja2_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: Jinja2 template
      template:
        src: template.j2
        dest: "/tmp/{{ ansible_hostname }}_template.out"
        trim_blocks: true
        mode: 0644

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

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

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

TASK [Ansible Jinja2 filters] **************************************************************************************************************
ok: [ubuntu-c] => {
    "msg": "---=== Ansible Jinja2 filters ===---\n--== min [1, 2, 3, 4, 5] ==--\n1\n--== max [1, 2, 3, 4, 5] ==--\n5\n--== unique [1,
}
ok: [centos1] => {
    "msg": "---=== Ansible Jinja2 filters ===---\n--== min [1, 2, 3, 4, 5] ==--\n1\n--== max [1, 2, 3, 4, 5] ==--\n5\n--== unique [1,
}
ok: [centos2] => {
    "msg": "---=== Ansible Jinja2 filters ===---\n--== min [1, 2, 3, 4, 5] ==--\n1\n--== max [1, 2, 3, 4, 5] ==--\n5\n--== unique [1,
}
ok: [centos3] => {
    "msg": "---=== Ansible Jinja2 filters ===---\n--== min [1, 2, 3, 4, 5] ==--\n1\n--== max [1, 2, 3, 4, 5] ==--\n5\n--== unique [1,
}
ok: [ubuntu1] => {
    "msg": "---=== Ansible Jinja2 filters ===---\n--== min [1, 2, 3, 4, 5] ==--\n1\n--== max [1, 2, 3, 4, 5] ==--\n5\n--== unique [1,
}
ok: [ubuntu2] => {
    "msg": "---=== Ansible Jinja2 filters ===---\n--== min [1, 2, 3, 4, 5] ==--\n1\n--== max [1, 2, 3, 4, 5] ==--\n5\n--== unique [1,
```

```
}
ok: [ubuntu3] => {
    "msg": "--=== Ansible Jinja2 filters ===---\n--== min [1, 2, 3, 4, 5] ==--\n1\n--== max [1, 2, 3, 4, 5] ==--\n5\n--== unique [1,
}

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
ubuntu-c                   : 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

$ cat /tmp/ubuntu-c_template.out
--== Ansible Jinja2 if statement ==--

This is ubuntu-c

--== Ansible Jinja2 if elif statement ==--

This is ubuntu-c

--== Ansible Jinja2 if elif else statement ==--

This is ubuntu-c

--== Ansible Jinja2 if variable is defined ( where variable is not defined ) ==--

example_variable is not defined

--== Ansible Jinja2 if varible is defined ( where variable is defined ) ==--

example_variable is defined

--== Ansible Jinja2 for statement ==--

IP Address entry 1 = 172.19.0.3

--== Ansible Jinja2 for range

1
2
3
4
5
6
7
8
9
10

--== Ansible Jinja2 for range, reversed (simulate while greater 5) ==--

10
9
8
7
6

--== Ansible Jinja2 for range, reversed (continue if odd) ==--

10
8
6
4
2

---=== Ansible Jinja2 filters ===---

--== min [1, 2, 3, 4, 5] ==--

1

--== max [1, 2, 3, 4, 5] ==--

5

--== unique [1, 1, 2, 2, 3, 3, 4, 4, 5, 5] ==--

[1, 2, 3, 4, 5]

--== difference [1, 2, 3, 4, 5] vs [2, 3, 4] ==--

[1, 5]

--== random ['rod', 'jane', 'freddy'] ==--
```

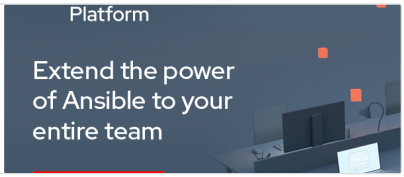
```
jane

--== urlsplit hostname ==--

docs.ansible.com
```

template 모듈의 `trim_blocks` parameter

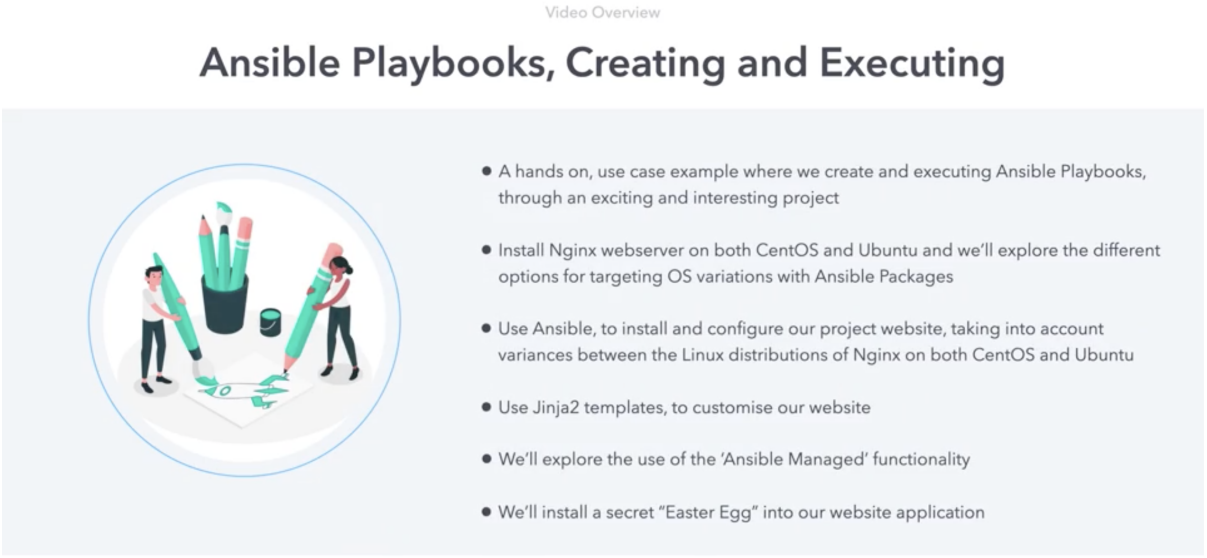
ansible.builtin.template module - Template a file out to a target host - Ansible Documentation

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/template_module.html#parameter-trim_blocks



6. Ansible Playbooks- Creating and Executing



6-1. Playbooks Creating and Executing Challenge

6-1-1. Configure our hosts to target the linux group

```
$ pwd
/home/ansible/diveintoansible/Ansible Playbooks, Introduction/Ansible Playbooks, Creating and Executing/solution/01

$ cat nginx_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

  # 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'

  # Handlers: the list of handlers that are executed as a notify key from a task
```

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

$ ansible-playbook nginx_playbook.yaml

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

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

TASK [Install EPEL] ****************************************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
changed: [centos1]
changed: [centos2]
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
```

6-1-2. Centos/RHEL uses yum or it's successor dnf, for package installation. Install the package named epel-release using either the yum or dnf module

Give the tasks a name of 'Install Epel'

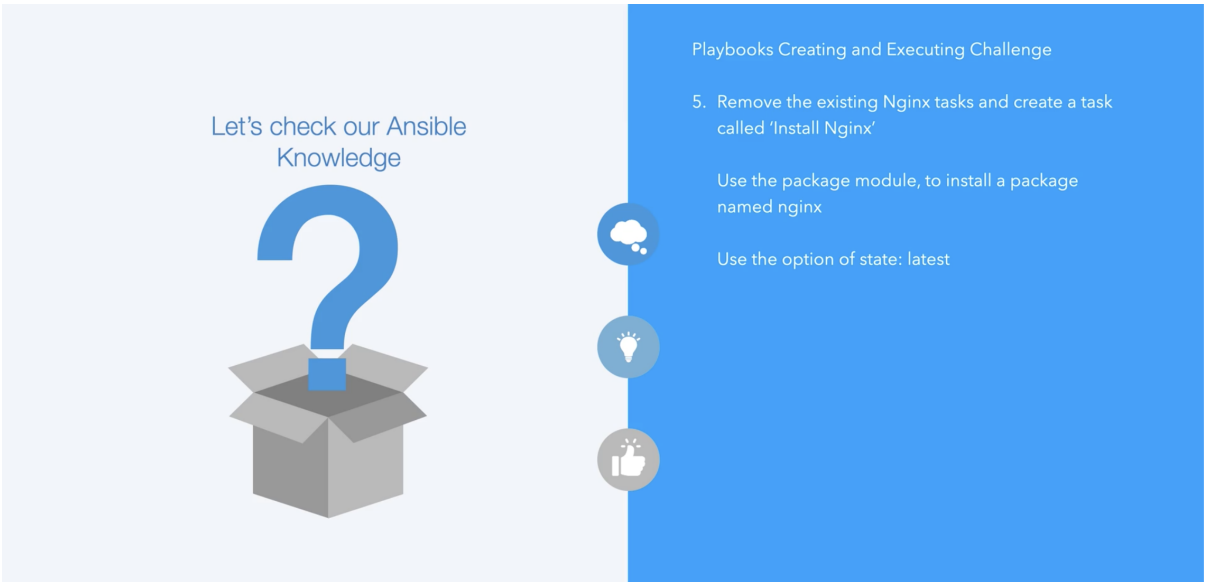
Use the following options -

`update_cache : yes`
`state : lates`

Target only the CentOS hosts for this using the fact ansible_distribution with a value of CentOS

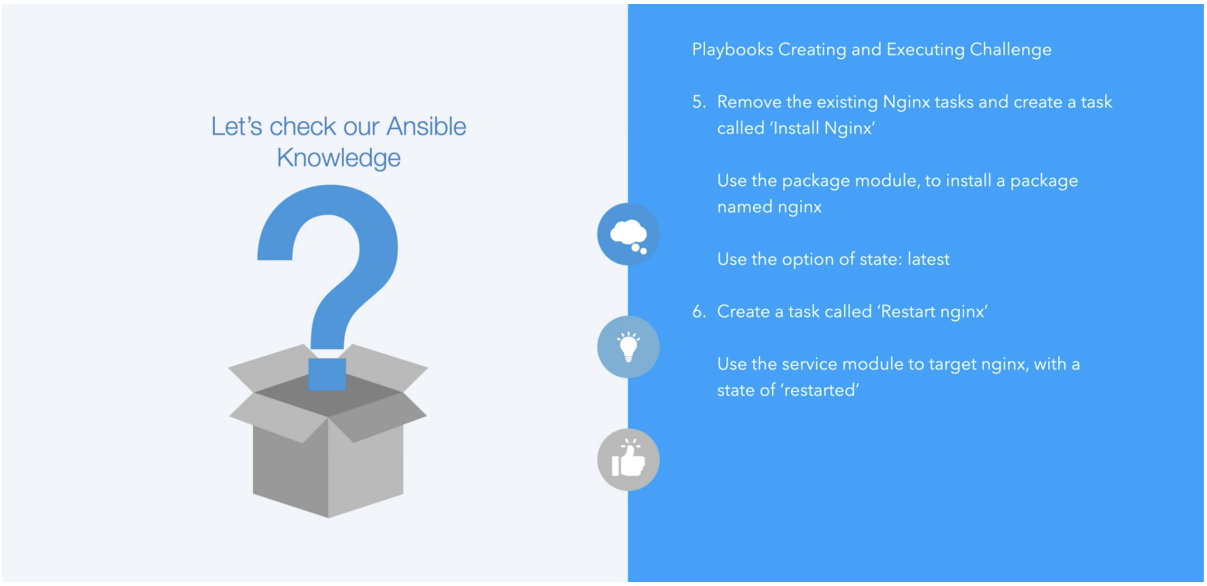

```
$ cd .../02

$ cat nginx_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

  # 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 CentOS
      yum:
        name: nginx
        update_cache: yes
        state: latest
      when: ansible_distribution == 'CentOS'

    - name: Install Nginx Ubuntu
      apt:
        name: nginx
        update_cache: yes
        state: latest
      when: ansible_distribution == 'Ubuntu'

  # Handlers: the list of handlers that are executed as a notify key from a task

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

$ ansible-playbook nginx_playbook.yaml

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

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

```
ok: [ubuntu3]

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

TASK [Install Nginx CentOS] ****************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
changed: [centos1]
changed: [centos2]
changed: [centos3]

TASK [Install Nginx Ubuntu] ****************************************************************************************************************
skipping: [centos1]
skipping: [centos2]
skipping: [centos3]
changed: [ubuntu3]
changed: [ubuntu2]
changed: [ubuntu1]

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




```
$ cd ../03/

$ cat nginx_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

  # 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
```

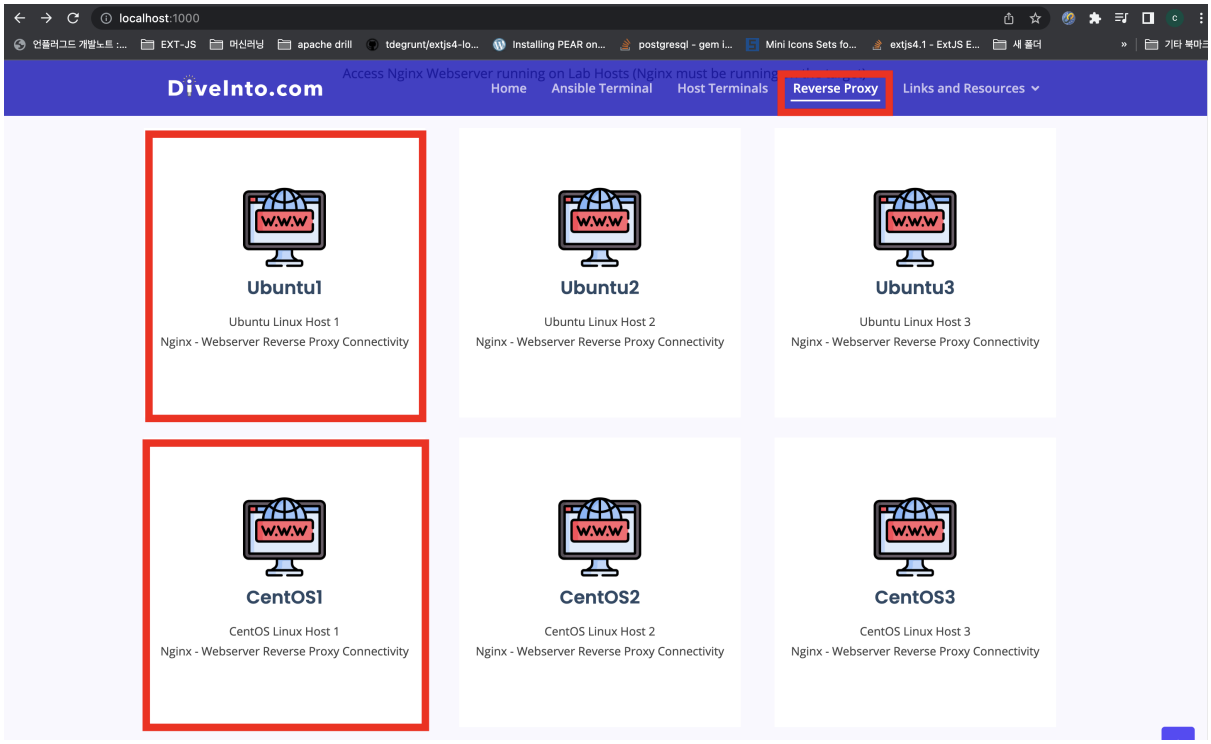
```
        update_cache: yes
        state: latest
      when: ansible_distribution == 'CentOS'

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

  # Handlers: the list of handlers that are executed as a notify key from a task

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

$ ansible-playbook nginx_playbook.yaml

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

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

TASK [Install EPEL] ****************************************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
ok: [centos3]
ok: [centos1]
ok: [centos2]

TASK [Install Nginx] ***************************************************************************************************************************************
ok: [centos1]
ok: [centos2]
ok: [centos3]
ok: [ubuntu1]
ok: [ubuntu2]
ok: [ubuntu3]

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

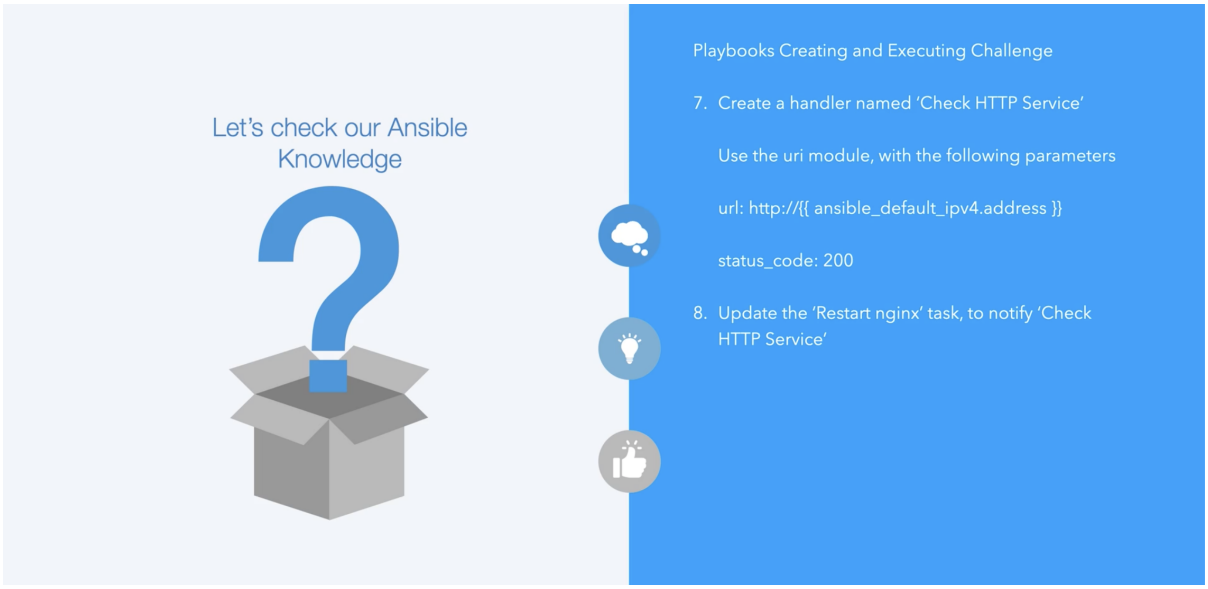



```
$ cd ../04/

$ cat nginx_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

  # 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

  # Handlers: the list of handlers that are executed as a notify key from a task

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

$ ansible-playbook nginx_playbook.yaml

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

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

TASK [Install EPEL] ****************************************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
ok: [centos2]
ok: [centos1]
ok: [centos3]

TASK [Install Nginx] ***************************************************************************************************************************************
ok: [centos3]
ok: [centos2]
ok: [centos1]
ok: [ubuntu1]
ok: [ubuntu2]
ok: [ubuntu3]

TASK [Restart nginx] ***************************************************************************************************************************************
changed: [centos2]
changed: [ubuntu2]
changed: [centos3]
changed: [ubuntu1]
changed: [centos1]
changed: [ubuntu3]

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
ubuntu1                    : ok=3    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu2                    : ok=3    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu3                    : ok=3    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

nginx 확인



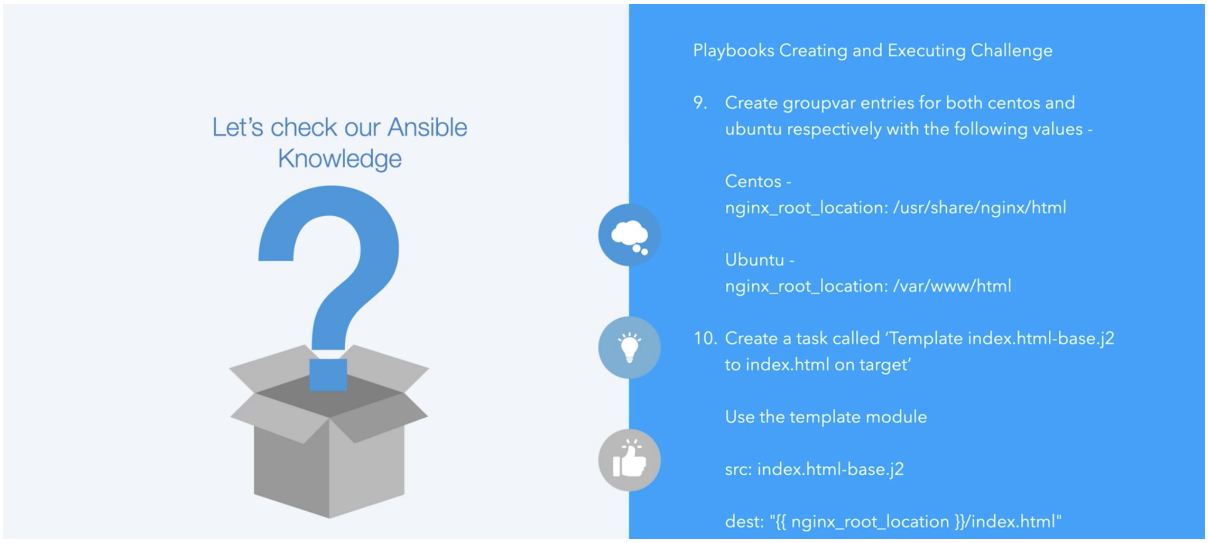



```
$ cd ../05
$ cat nginx_playbook.yaml
---
# YAML documents begin with the document separator ---

# 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

  # 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
      notify: Check HTTP Service

  # Handlers: the list of handlers that are executed as a notify key from a task
  handlers:
    - name: Check HTTP Service
      uri:
        url: http://{{ ansible_default_ipv4.address }}
        status_code: 200

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

$ ansible-playbook nginx_playbook.yaml

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

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

TASK [Install EPEL] ********************************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
ok: [centos3]
ok: [centos1]
ok: [centos2]

TASK [Install Nginx] *******************************************************************************************************************************
ok: [centos2]
ok: [centos3]
ok: [centos1]
ok: [ubuntu2]
ok: [ubuntu1]
ok: [ubuntu3]

TASK [Restart nginx] *******************************************************************************************************************************
changed: [centos1]
changed: [centos2]
changed: [ubuntu2]
changed: [ubuntu1]
changed: [centos3]
changed: [ubuntu3]

RUNNING HANDLER [Check HTTP Service] ***************************************************************************************************************
ok: [centos2]
ok: [centos1]
ok: [centos3]
ok: [ubuntu1]
ok: [ubuntu2]
ok: [ubuntu3]

PLAY RECAP *****************************************************************************************************************************************
centos1                    : ok=5    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=5    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=5    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=4    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu2                    : ok=4    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu3                    : ok=4    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

centos 와 ubuntu 의 nginx root

- centos : /usr/share/nginx/html
- ubuntu : /var/www/html




```
$ cd ../06/
$ cat nginx_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

  # 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
      notify: Check HTTP Service

    - name: Template index.html-base.j2 to index.html on target
      template:
        src: index.html-base.j2
        dest: "{{ nginx_root_location }}/index.html"
        mode: 0644

  # Handlers: the list of handlers that are executed as a notify key from a task
  handlers:
    - name: Check HTTP Service
      uri:
        url: http://{{ ansible_default_ipv4.address }}
        status_code: 200

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

group_vars

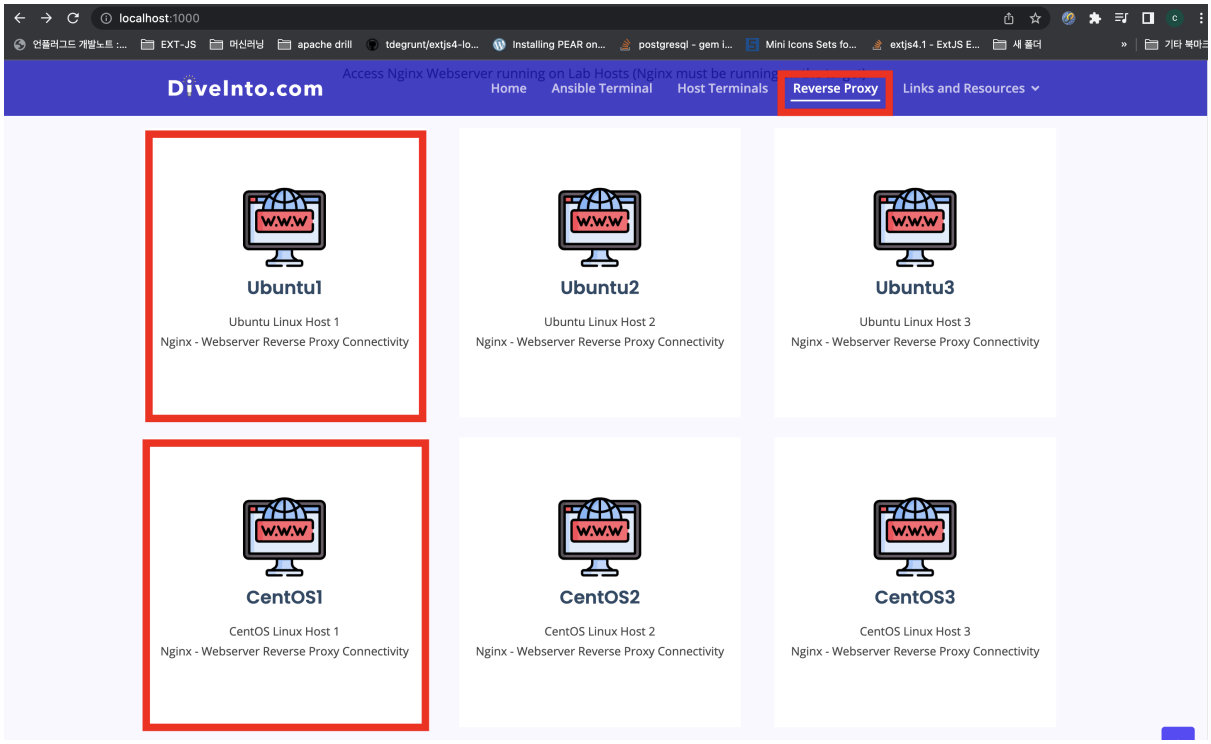
```
---
ansible_user: root
```

```
nginx_root_location: /usr/share/nginx/html
...
```

```
---
ansible_become: true
ansible_become_pass: password
nginx_root_location: /var/www/html
...
```

index.html-base.js2 ⇒ templates/index.html-base.j2

```
$ ansible-playbook nginx_playbook.yaml

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

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

TASK [Install EPEL] ****************************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
ok: [centos1]
ok: [centos2]
ok: [centos3]

TASK [Install Nginx] ***************************************************************************************************************************
ok: [centos2]
ok: [centos1]
ok: [centos3]
ok: [ubuntu1]
ok: [ubuntu2]
ok: [ubuntu3]

TASK [Restart nginx] ***************************************************************************************************************************
changed: [centos3]
changed: [centos1]
changed: [ubuntu2]
changed: [ubuntu1]
changed: [centos2]
changed: [ubuntu3]

TASK [Template index.html-base.j2 to index.html on target] *************************************************************************************
changed: [centos3]
changed: [centos2]
changed: [ubuntu1]
changed: [ubuntu2]
changed: [centos1]
changed: [ubuntu3]

RUNNING HANDLER [Check HTTP Service] ***********************************************************************************************************
ok: [centos2]
ok: [centos1]
ok: [centos3]
ok: [ubuntu1]
ok: [ubuntu2]
ok: [ubuntu3]

PLAY RECAP *************************************************************************************************************************************
centos1                    : ok=6    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=6    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=6    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=5    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu2                    : ok=5    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu3                    : ok=5    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

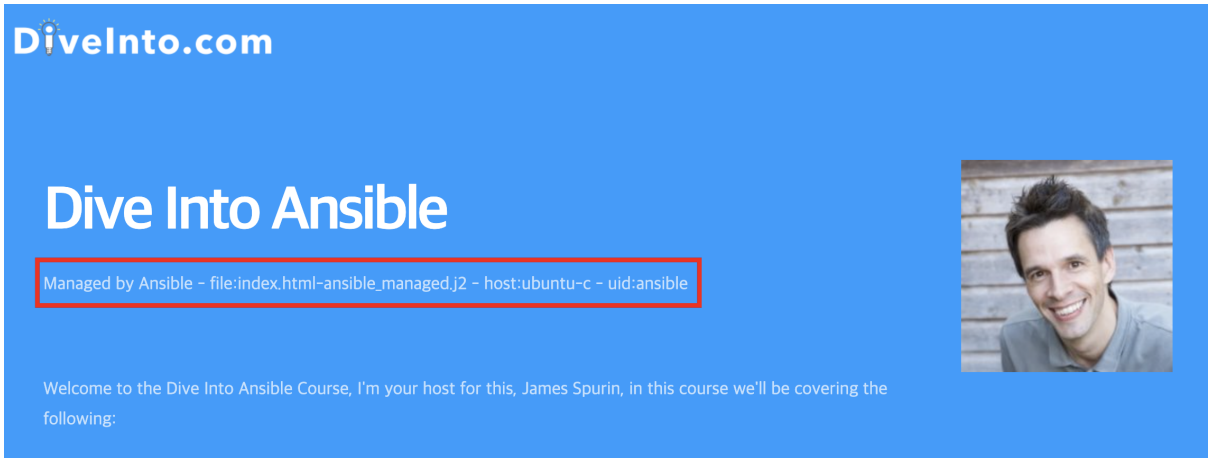


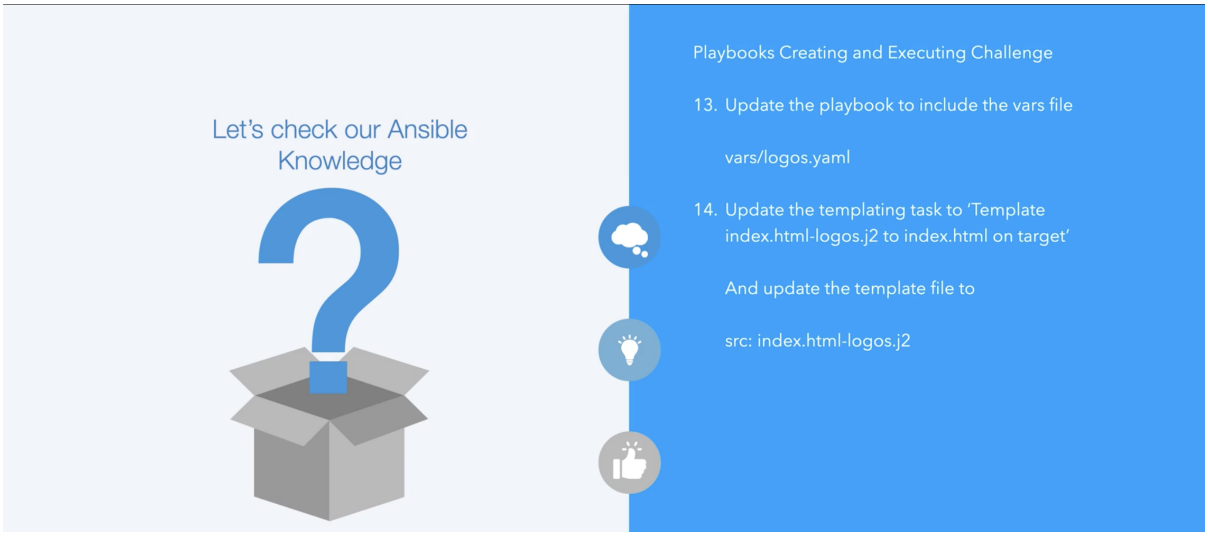

```
$ cd ../07
$ cat ansible.cfg
[defaults]
inventory = hosts
host_key_checking = False
ansible_managed = Managed by Ansible - file:{file} - host:{host} - uid:{uid}
```

```
<h1>Dive Into Ansible</h1>
<p>{{ ansible_managed }}</p>
<br>
```

```
$ ansible-playbook nginx_playbook.yaml

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

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

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

TASK [Install EPEL] ************************************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
ok: [centos3]
ok: [centos2]
ok: [centos1]

TASK [Install Nginx] ***********************************************************************************************************************************
ok: [centos2]
ok: [centos3]
ok: [centos1]
ok: [ubuntu2]
ok: [ubuntu1]
ok: [ubuntu3]

TASK [Restart nginx] ***********************************************************************************************************************************
changed: [centos2]
changed: [ubuntu1]
changed: [centos1]
changed: [centos3]
changed: [ubuntu2]
changed: [ubuntu3]

TASK [Template index.html-ansible_managed.j2 to index.html on target] **********************************************************************************
changed: [centos1]
changed: [centos2]
changed: [centos3]
changed: [ubuntu1]
changed: [ubuntu2]
changed: [ubuntu3]

RUNNING HANDLER [Check HTTP Service] *******************************************************************************************************************
ok: [centos1]
ok: [centos2]
ok: [centos3]
ok: [ubuntu2]
ok: [ubuntu1]
ok: [ubuntu3]

PLAY RECAP *********************************************************************************************************************************************
centos1                    : ok=6    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=6    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=6    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=5    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu2                    : ok=5    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu3                    : ok=5    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```





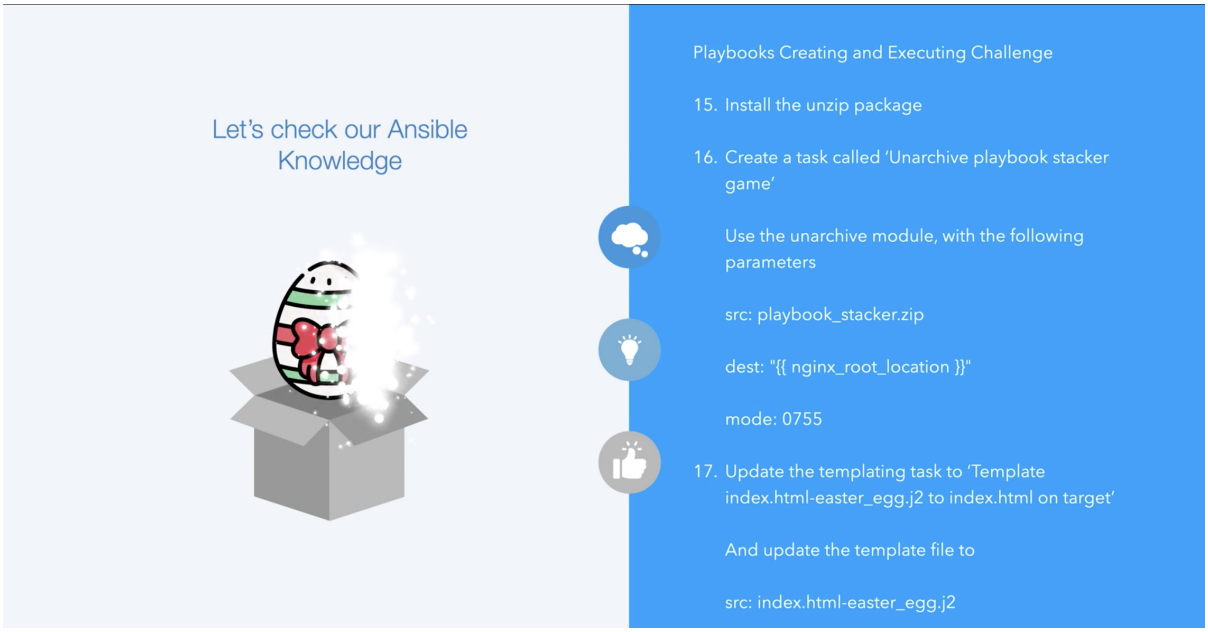

```
$ cd ../08
```

```
---
# 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:
    - vars/logos.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: 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
      notify: Check HTTP Service

    - name: Template index.html-logos.j2 to index.html on target
      template:
        src: index.html-logos.j2
        dest: "{{ nginx_root_location }}/index.html"
        mode: 0644

  # Handlers: the list of handlers that are executed as a notify key from a task
  handlers:
    - name: Check HTTP Service
      uri:
        url: http://{{ ansible_default_ipv4.address }}
        status_code: 200

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

```
$ ansible-playbook nginx_playbook.yaml

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

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

TASK [Install EPEL] ****************************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
ok: [centos3]
ok: [centos2]
ok: [centos1]

TASK [Install Nginx] ***************************************************************************************************************************
ok: [centos2]
ok: [centos1]
ok: [centos3]
ok: [ubuntu2]
ok: [ubuntu1]
ok: [ubuntu3]

TASK [Restart nginx] ***************************************************************************************************************************
changed: [centos3]
changed: [centos1]
changed: [ubuntu2]
changed: [centos2]
changed: [ubuntu1]
changed: [ubuntu3]

TASK [Template index.html-logos.j2 to index.html on target] ************************************************************************************
changed: [ubuntu2]
changed: [centos3]
changed: [centos2]
changed: [ubuntu1]
changed: [centos1]
changed: [ubuntu3]

RUNNING HANDLER [Check HTTP Service] ***********************************************************************************************************
ok: [centos3]
ok: [ubuntu2]
ok: [centos1]
ok: [centos2]
ok: [ubuntu1]
ok: [ubuntu3]

PLAY RECAP *************************************************************************************************************************************
centos1                    : ok=6    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=6    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=6    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=5    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu2                    : ok=5    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu3                    : ok=5    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

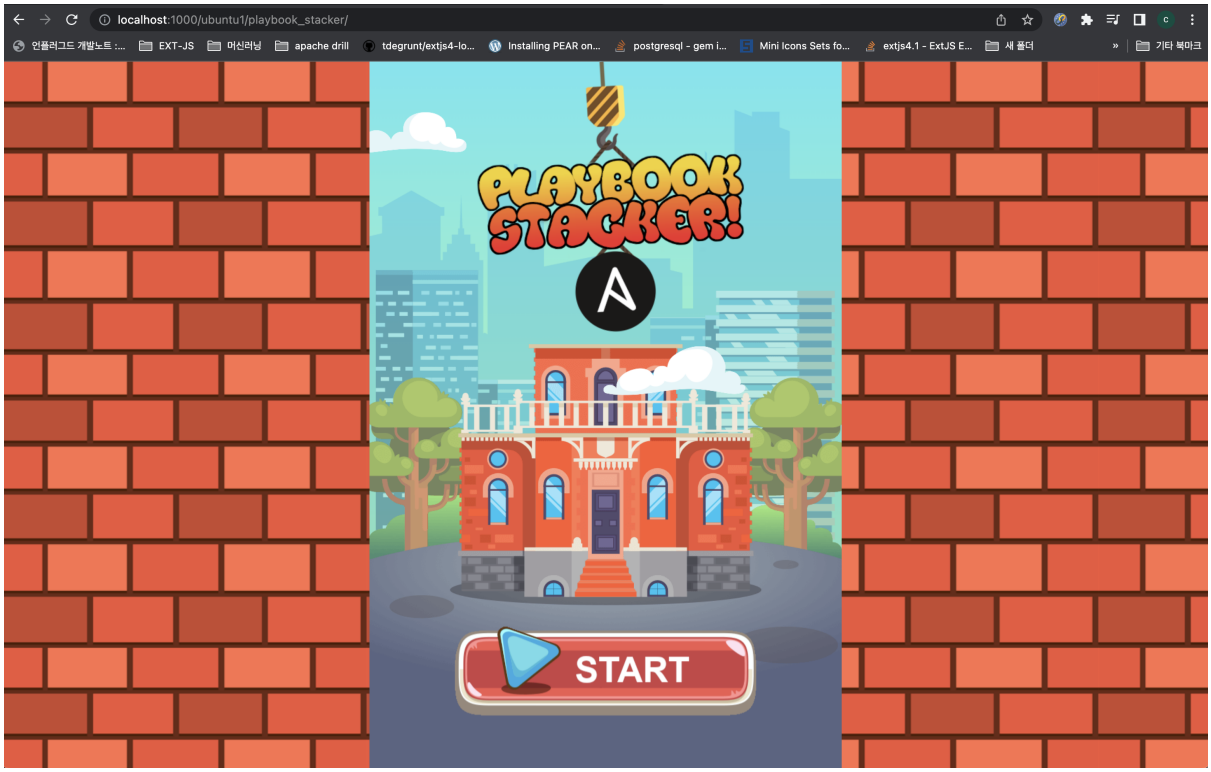
```
---
centos_logo: '......'
ubuntu_logo: '......'
...
```

```
<img src="
    {%- if ansible_distribution == 'CentOS' -%}
    {{ centos_logo }}
    {%- elif ansible_distribution == 'Ubuntu' -%}
    {{ ubuntu_logo }}
    {%- endif %}" class="intro" alt="mainpic">
```




```
$ cd ../09
$ cat nginx_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_files:
    - vars/logos.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: 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
      notify: Check HTTP Service

    - name: Template index.html-easter_egg.j2 to index.html on target
      template:
        src: index.html-easter_egg.j2
        dest: "{{ nginx_root_location }}/index.html"
        mode: 0644

    - name: Install unzip
      package:
        name: unzip
        state: latest

    - name: Unarchive playbook stacker game
      unarchive:
        src: playbook_stacker.zip
        dest: "{{ nginx_root_location }}"
        mode: 0755

  # Handlers: the list of handlers that are executed as a notify key from a task
```

```
  handlers:
    - name: Check HTTP Service
      uri:
        url: http://{{ ansible_default_ipv4.address }}
        status_code: 200

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

$ ansible-playbook nginx_playbook.yaml

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

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

TASK [Install EPEL] ****************************************************************************************************************************
skipping: [ubuntu1]
skipping: [ubuntu2]
skipping: [ubuntu3]
ok: [centos1]
ok: [centos2]
ok: [centos3]

TASK [Install Nginx] ***************************************************************************************************************************
ok: [centos1]
ok: [centos2]
ok: [centos3]
ok: [ubuntu2]
ok: [ubuntu1]
ok: [ubuntu3]

TASK [Restart nginx] ***************************************************************************************************************************
changed: [centos3]
changed: [centos1]
changed: [centos2]
changed: [ubuntu1]
changed: [ubuntu2]
changed: [ubuntu3]

TASK [Template index.html-easter_egg.j2 to index.html on target] *******************************************************************************
changed: [centos1]
changed: [centos2]
changed: [centos3]
changed: [ubuntu1]
changed: [ubuntu2]
changed: [ubuntu3]

TASK [Install unzip] ***************************************************************************************************************************
ok: [centos2]
ok: [centos3]
ok: [centos1]
ok: [ubuntu2]
ok: [ubuntu1]
ok: [ubuntu3]

TASK [Unarchive playbook stacker game] *********************************************************************************************************
changed: [centos1]
changed: [centos3]
changed: [ubuntu1]
changed: [centos2]
changed: [ubuntu2]
changed: [ubuntu3]

RUNNING HANDLER [Check HTTP Service] ***********************************************************************************************************
ok: [ubuntu1]
ok: [centos2]
ok: [centos3]
ok: [centos1]
ok: [ubuntu2]
ok: [ubuntu3]

PLAY RECAP *************************************************************************************************************************************
centos1                    : ok=8    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos2                    : ok=8    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
centos3                    : ok=8    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu1                    : ok=7    changed=3    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu2                    : ok=7    changed=3    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
ubuntu3                    : ok=7    changed=3    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```


localhost:1000/ubuntu1/playbook_stacker/
PLAYBOOK STACKER!
START