

Chapter 6: Structuring Ansible Playbooks

1. Using Includes and Imports

- [1-1. include_tasks](#)
- [1-2. import_tasks](#)
- [1-3. import vs include](#)
- [1-4. how to applied 'when' statement](#)
- [1-4. import_playbook](#)

2. Using Tags

- [2-1. Using Tags](#)
- [2-2. Segmentation with Tags](#)
- [2-3. Execution with Tags](#)
- [2-4. Skipping with Tags](#)
- [2-5. Playbook Tags](#)
- [2-6. Special Tags](#)
- [2-7. Tag Inheritance](#)

3. Using Roles

- [3-1. Using Role](#)
- [3-1. The Role Structure](#)
- [3-3. How to create Roles with Ansible Galaxy](#)
- [3-4. How to move an existing project, to a Role](#)
- [3-5. Role Execution](#)
- [3-6. Role Parameters](#)
- [3-7. Role Dependencies](#)

1. Using Includes and Imports

Video Overview

Using Includes & Imports



- include_tasks
- import_tasks
- Static vs Dynamic
- import_playbook

1-1. include_tasks

```
---
# 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: Play 1 - Task 1
    debug:
      msg: Play 1 - Task 1

  - include_tasks: play1_task2.yaml

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

```

```

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

- name: Play 1 - Task 2
  debug:
    msg: Play 1 - Task 2

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

```

1-2. import_tasks

```

---
# 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: Play 1 - Task 1
      debug:
        msg: Play 1 - Task 1

    - import_tasks: play1_task2.yaml

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

```

```

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

- name: Play 1 - Task 2
  debug:
    msg: Play 1 - Task 2

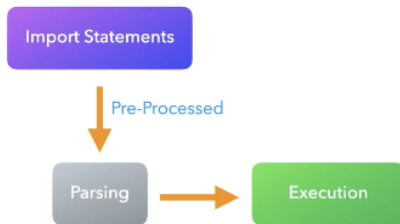
# Three dots indicate the end of a YAML document
...

```

1-3. import vs include

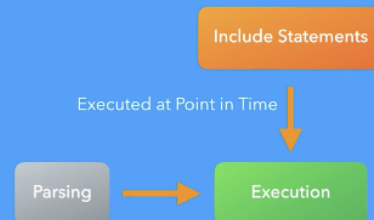
Static

- Processed as the playbook is parsed
- 'when' statement applies to **all** individual tasks at task point of execution



Dynamic

- Processed at playbook execution
- 'when' statement applies to **all** tasks at initial point of execution



1-4. how to applied 'when' statement

```

---
# 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

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

  - debug:
      msg: ===== Testing include_tasks =====

  # include_tasks is dynamic
  #
  # The when statement is executed once, if the condition is met, all
  # tasks are executed
  - include_tasks: include_tasks.yaml
    when: include_tasks_var is not defined

  - debug:
      msg: ===== Testing import_tasks =====

  # import_tasks is static
  #
  # Each task that in the include will be independently executed against
  # the when condition
  - import_tasks: import_tasks.yaml
    when: import_tasks_var is not defined

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

```

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

- set_fact:
    include_tasks_var: foo

- name: 2nd Task
  debug:
    msg: 2nd Task

- name: 3rd Task
  debug:
    msg: 3rd Task
  
```

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

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

- set_fact:
    import_tasks_var: foo

- name: 2nd Task
  debug:
    msg: 2nd Task

- name: 3rd Task
  debug:
    msg: 3rd Task

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

```
$ ansible-playbook include_import_tasks_playbook.yaml
```

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

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

TASK [debug] *****
ok: [centos1] => {
  "msg": "===== Testing include_tasks ====="
}

TASK [include_tasks] *****
included: /home/ansible/diveintoansible/Structuring Ansible Playbooks/Using Include and Import/03/include_tasks.yaml for centos1

TASK [set_fact] *****
ok: [centos1]

TASK [2nd Task] *****
ok: [centos1] => {
  "msg": "2nd Task"
}

TASK [3rd Task] *****
ok: [centos1] => {
  "msg": "3rd Task"
}

TASK [debug] *****
ok: [centos1] => {
  "msg": "===== Testing import_tasks ====="
}

TASK [set_fact] *****
ok: [centos1]

TASK [2nd Task] *****
skipping: [centos1]

TASK [3rd Task] *****
skipping: [centos1]

PLAY RECAP *****
centos1                : ok=8    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

1-4. import_playbook

```
---
# 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

# import_playbook is static
#
# Each task that is in the playbook in the import will be independently executed against
# the when condition
```

```
- import_playbook: imported_playbook.yaml
  when: import_playbook_var is not defined

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

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

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

  # Hosts: where our play will run and options it will run with
  hosts: centos1

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

    - set_fact:
        import_playbook_var: true

    - debug:
        msg: Playbook executed

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

import_playbook 은 런타임에 when 파라미터가 동작하므로 플레이북을 실행시키면 디버그 메시지가 출력되지 않는다. 즉, import 된 플레이북이 실행되지 않는다. `when: import_playbook_var is not defined` 을 `when: import_playbook_var is defined` 으로 변경해도 결과는 동일하다.

```
$ ansible-playbook import_playbook.yaml

PLAY [centos1] *****

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

TASK [set_fact] *****
ok: [centos1]

TASK [debug] *****
skipping: [centos1]

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

2. Using Tags

Tags



- Using Tags
- Segmentation with Tags
- Execution with Tags
- Skipping with Tags
- Playbook Tags
- Special Tags
- Tag Inheritance

2-1. Using Tags

```

---
# 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'
    tags:
      - install-epel

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

  - name: Restart nginx
    service:
      name: nginx
      state: restarted
    notify: Check HTTP Service
    tags:
      - restart-nginx

  - 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
    tags:
      - deploy-app

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

```

```

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

# 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
...

```

```

# centos 에서만 실행
$ ansible-playbook nginx_playbook.yaml --tags "install-epel"

# 한개 이상의 태그 실행
$ ansible-playbook nginx_playbook.yaml --tags "install-epel,restart-nginx"

# 배포
$ ansible-playbook nginx_playbook.yaml --tags "deploy-app"

# 일부 태그를 제외하고 실행
$ ansible-playbook nginx_playbook.yaml --skip-tags "deploy-app"

# 모든 태그 실행
$ ansible-playbook nginx_playbook.yaml --tags "all"

```

2-2. Segmentation with Tags

```

---
# 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
tags:
  - webapp

# 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'
  tags:
    - install-epel

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

- name: Restart nginx
  service:
    name: nginx
    state: restarted
  notify: Check HTTP Service
  tags:
    - restart-nginx

- 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
  tags:
    - deploy-app

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

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

# 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
...
```

```

# perform gathering facts
$ ansible-playbook nginx_playbook.yaml --tags "webapp"

# does not perform gathering facts
$ ansible-playbook nginx_playbook.yaml --tags "install-nginx"
```

2-3. Execution with Tags

```

---
# 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
#
# Empty play, to ensure that facts are gathered and available when tags
# are called, against individual plays (where a play tag exists)
-

  # Hosts: where our play will run and options it will run with
  # playbook 실행시 tag를 지정해도 앞선 예제와 달리 gathering facts를 무조건 수행하게된다.
  hosts: linux

-

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

  # 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'
      tags:
        - install-epel

    - name: Install Nginx
      package:
        name: nginx
        state: latest
      tags:
        - install-nginx
```



```

- name: Restart nginx
  service:
    name: nginx
    state: restarted
  notify: Check HTTP Service
  tags:
    - restart-nginx

- 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
  tags:
    - deploy-app

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

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

# 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
...

```

2-2 결과 달리 gathering facts가 실행된다.
\$ ansible-playbook nginx_playbook.yaml --tags "install-nginx"

2-4. Skipping with Tags

2-5. Playbook Tags

2-6. Special Tags

Special Tags



- **tagged**: only tasks that have a tag, will be run
- **untagged**: run tasks that do not have a tag, always does not count as a tag here
- **all**: essentially this is a no-op operation, by default, ansible is running with --tags all

```

---
# 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
#
# Empty play, to ensure that facts are gathered and available when tags
# are called, against individual plays (where a play tag exists)
-

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

-

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

  # 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'
      tags:
        - install-epel

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

    - name: Restart nginx
      service:
        name: nginx
        state: restarted
      notify: Check HTTP Service
      tags:
        - always

    - 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
      tags:
        - deploy-app

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

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

  # 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
...

```

```
# install-nginx 태그만 지정했지만, always 태그로 지정된 타스크도 실행이 된다.
$ ansible-playbook nginx_playbook.yaml --tags "install-nginx"

# install-nginx 태그만 실행하고 always 태그는 제외할 수 있다.
$ ansible-playbook nginx_playbook.yaml --tags "install-nginx" --skip-tags "always"

# 태깅된 타스크만 실행
$ ansible-playbook nginx_playbook.yaml --tags "tagged"

# 태깅되지 않은 타스크 실행(여기서는 always 태그만 실행됨)
$ ansible-playbook nginx_playbook.yaml --tags "untagged"

# 모든 태깅된 타스크 실행(special tag포함)
$ ansible-playbook nginx_playbook.yaml --tags "all"
```

2-7. Tag Inheritance

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

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

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

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

    - include_tasks: include_tasks.yaml
      tags:
        - include_tasks

    - import_tasks: import_tasks.yaml
      tags:
        - import_tasks

- import_playbook: import_playbook.yaml
  tags:
    - import_playbook

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

```
$ for tag in include_tasks import_tasks import_playbook; do echo ===== Testing ${tag} =====; ansible-playbook include_
===== Testing include_tasks =====

PLAY [ubuntu3] *****

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

TASK [include_tasks] *****
included: /home/ansible/diveintoansible/Structuring Ansible Playbooks/Using Tags/05/include_tasks.yaml for ubuntu3

PLAY [centos1] *****

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

===== Testing import_tasks =====

PLAY [ubuntu3] *****

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

TASK [debug] *****
ok: [ubuntu3] => {
  "msg": "Import tasks executed"
}

PLAY [centos1] *****

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

```

===== Testing import_playbook =====

PLAY [ubuntu3] *****

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

PLAY [centos1] *****

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

TASK [debug] *****
ok: [centos1] => {
  "msg": "Import playbook executed"
}

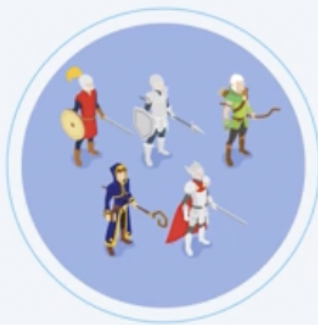
PLAY RECAP *****
centos1      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu3     : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

3. Using Roles

Video Overview

Roles



- Using Roles
- The Role Structure
- How to create Roles with Ansible Galaxy
- How to move an existing project, to a Role
- Role Execution
- Role Parameters
- Role Dependencies

3-1. Using Role

Roles



- Larger Projects are made easier to manage
- Roles are grouped, with logical structure, making them easier to share
- Roles can be written, to specific requirements, for example a web server role, a DNS role or a patching role
- Roles can be developed independently, in parallel by different entities
- Templates, Vars and Files, have designated directories and inclusion is simplified
- Roles, can have dependencies on other roles, therefore providing automated inclusion

3-1. The Role Structure

Roles



```
example-role/
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml
```

```
$ pwd
/home/ansible/diveintoansible/Structuring Ansible Playbooks/Using Roles/01

$ tree
.
├── ansible.cfg
├── files
│   └── playbook_stackerc.zip
├── group_vars
│   ├── centos
│   └── ubuntu
├── host_vars
│   ├── centos1
│   └── ubuntu-c
├── hosts
├── nginx_playbook.yaml
├── templates
│   ├── index.html-ansible_managed.j2
│   ├── index.html-base.j2
│   ├── index.html-easter_egg.j2
│   └── index.html-logos.j2
```

```
|   └─ index.html.j2
└─ vars
    └─ logos.yaml

5 directories, 14 files
```

3-3. How to create Roles with Ansible Galaxy

```
$ pwd
/home/ansible/diveintoansible/Structuring Ansible Playbooks/Using Roles/02

$ ansible-galaxy init nginx
- Role nginx was created successfully

$ tree nginx -a
nginx
├─ .travis.yml
├─ README.md
├─ defaults
│   └─ main.yml
├─ files
├─ handlers
│   └─ main.yml
├─ meta
│   └─ main.yml
├─ tasks
│   └─ main.yml
├─ templates
├─ tests
│   ├── inventory
│   └─ test.yml
└─ vars
    └─ main.yml

8 directories, 9 files

$ vi nginx_playbook.yaml nginx/handlers/main.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

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

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

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

```
$ mv templates/* nginx/templates/
$ rm -rf templates
$ mv files/* nginx/files/
$ rm -rf files
$ cat vars/logos.yaml >> nginx/vars/main.yml
$ rm -rf vars
```

```
---
# 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
#
# Empty play, to ensure that facts are gathered and available when tags
# are called, against individual plays (where a play tag exists)
-

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

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

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

```

```

---
# tasks file for nginx
- name: Install EPEL
  yum:
    name: epel-release
    update_cache: yes
    state: latest
  when: ansible_distribution == 'CentOS'
  tags:
    - install-epel

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

- name: Restart nginx
  service:
    name: nginx
    state: restarted
  notify: Check HTTP Service
  tags:
    - always

- 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
  tags:
    - deploy-app

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

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

```

```

---
# handlers file for nginx
- name: Check HTTP Service
  uri:
    url: http://{{ ansible_default_ipv4.address }}
    status_code: 200

```

```

ansible@ubuntu-c:~/diveintoansible/Structuring Ansible Playbooks/Using Roles/02$ tree . -a
.
├── ansible.cfg
├── group_vars
│   ├── centos
│   └── ubuntu
├── host_vars
│   ├── centos1
│   └── ubuntu-c
├── hosts
├── nginx
│   ├── .travis.yml
│   ├── README.md
│   ├── defaults
│   │   └── main.yml
│   ├── files
│   │   └── playbook_stacker.zip
│   ├── handlers
│   │   └── main.yml
│   ├── meta
│   │   └── main.yml
│   ├── tasks
│   │   └── main.yml
│   ├── templates
│   │   ├── index.html-ansible_managed.j2
│   │   ├── index.html-base.j2
│   │   ├── index.html-easter_egg.j2
│   │   ├── index.html-logos.j2
│   │   └── index.html.j2
│   ├── tests
│   │   ├── inventory
│   │   └── test.yml
│   └── vars
│       └── main.yml
└── nginx_playbook.yaml

```

3-4. How to move an existing project, to a Role

```

$ cd /home/ansible/diveintoansible/Structuring Ansible Playbooks/Using Roles/04

# Create Role 'webapp'
$ ansible-galaxy init webapp
- Role webapp was created successfully

$ mv nginx/templates/* webapp/templates/

$ mv nginx/files/* webapp/files
$ cp webapp/vars/main.yml nginx/vars/
$ cp nginx/vars/main.yml webapp/vars/

```

```

---
# tasks file for nginx
- name: Install EPEL
  yum:
    name: epel-release
    update_cache: yes
    state: latest
  when: ansible_distribution == 'CentOS'
  tags:
    - install-epel

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

- name: Restart nginx
  service:
    name: nginx
    state: restarted
  notify: Check HTTP Service

```



```
tags:
  - always
```

```
---
# tasks file for webapp
- name: Template index.html.j2 to index.html on target
  template:
    src: index.html-easter_egg.j2
    dest: "{{ nginx_root_location }}/index.html"
    mode: 0644
  tags:
    - deploy-app

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

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

```
$ mv nginx_playbook.yaml nginx_webapp_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

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

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

```
$ nginx-playbook nginx_webapp_playbook.yaml
```

3-5. Role Execution

```
$ cd ../06

# path is for ubuntu, not for cenots.
# It will fail on centos
$ cat webapp/defaults/main.yaml
---
# defaults file for webapp
target_dir: /var/www/html

$ ansible-playbook nginx_webapp_playbook.yaml
TASK [webapp : Template index.html.j2 to index.html on target] *****
fatal: [centos3]: FAILED! => {"changed": false, "checksum": "67b833f56dbd8345b493c42f2d1372d8348fc44f", "msg": "Destination directory
ok: [ubuntu2]
fatal: [centos1]: FAILED! => {"changed": false, "checksum": "67b833f56dbd8345b493c42f2d1372d8348fc44f", "msg": "Destination directory
fatal: [centos2]: FAILED! => {"changed": false, "checksum": "67b833f56dbd8345b493c42f2d1372d8348fc44f", "msg": "Destination directory
ok: [ubuntu1]
ok: [ubuntu3]
```

3-6. Role Parameters

```

---
# 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

# Roles: list of roles to be imported into the play
roles:
  - nginx
  - { role: webapp, target_dir: "{%- if ansible_distribution == 'CentOS' -%}/usr/share/nginx/html{% elif ansible_distribution == 'U

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

```

3-7. Role Dependencies

- nginx role을 삭제했다.

```

---
# 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

# Roles: list of roles to be imported into the play
roles:
  - { role: webapp, target_dir: "{%- if ansible_distribution == 'CentOS' -%}/usr/share/nginx/html{% elif ansible_distribution == 'U

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

```

메타정보에 dependency 에 Role을 추가한다.(- nginx)

```

galaxy_info:
  author: your name
  description: your role description
  company: your company (optional)

# If the issue tracker for your role is not on github, uncomment the
# next line and provide a value
# issue_tracker_url: http://example.com/issue/tracker

# Choose a valid license ID from https://spdx.org - some suggested licenses:
# - BSD-3-Clause (default)
# - MIT
# - GPL-2.0-or-later
# - GPL-3.0-only
# - Apache-2.0
# - CC-BY-4.0
license: license (GPL-2.0-or-later, MIT, etc)

min_ansible_version: 2.1

# If this a Container Enabled role, provide the minimum Ansible Container version.
# min_ansible_container_version:

#
# Provide a list of supported platforms, and for each platform a list of versions.
# If you don't wish to enumerate all versions for a particular platform, use 'all'.
# To view available platforms and versions (or releases), visit:
# https://galaxy.ansible.com/api/v1/platforms/
#
# platforms:
# - name: Fedora
#   versions:
#   - all
#   - 25
# - name: SomePlatform
#   versions:

```

```
# - all
# - 1.0
# - 7
# - 99.99

galaxy_tags: []
# List tags for your role here, one per line. A tag is a keyword that describes
# and categorizes the role. Users find roles by searching for tags. Be sure to
# remove the '[]' above, if you add tags to this list.
#
# NOTE: A tag is limited to a single word comprised of alphanumeric characters.
#       Maximum 20 tags per role.

dependencies:
- nginx
# List your role dependencies here, one per line. Be sure to remove the '[]' above,
# if you add dependencies to this list.
```