

# ANSIBLE SEMINAR #2

---



- 지난시간 ANSIBLE 소개와 특징, 설치에 대해 설명했습니다.
  - 이번시간에는 실제 사용을 통해서 관리하는 방법을 익히도록 하겠습니다.
- 

1. Mission #1 다음의 표를 참고하여 hosts 파일에 그룹명 seminar로 등록합니다.

SERVER	IP	PORT	ID	PASSWORD
Ansible #1	203.230.100.59	22002	wisoft	wisoft123
Ansible #2	203.230.100.59	22003	wisoft	wisoft123
Ansible #3	203.230.100.59	22004	wisoft	wisoft123

2. Mission #2 Ansible 명령을 활용하여 seminar 그룹에 ping을 확인합니다.

3. Mission #3 Ansible을 활용하여 'df -h'를 실행해 봅니다.

4. Mission #4 sudo 권한을 이용하여 'update', 'upgrade' 진행합니다.

---

# - START ANSIBLE -

## YAML

- YAML(야믈)은 XML, C, 파이썬, 펄, RFC2822에서 정의된 e-mail 양식에서 개념을 얻어 만들어진 '사람이 쉽게 읽을 수 있는' 데이터 직렬화 양식이다. 2001년에 클라크 에반스가 고안했고, Ingy dot Net 및 Oren Ben-Kiki와 함께 디자인했다.
- YAML이라는 이름은 "YAML은 마크업 언어가 아니다 (YAML Ain't Markup Language)" 라는 재귀적인 이름에서 유래되었다. 원래 YAML의 뜻은 "또 다른 마크업 언어 (Yet Another Markup Language)"였으나, YAML의 핵심은 문서 마크업이 아닌 데이터 중심에 있다는 것을 보여주기 위해 이름을 바꾸었다. 오늘날 XML이 데이터 직렬화에 주로 쓰이기 시작하면서, 많은 사람들이 YAML을 '가벼운 마크업 언어'로 사용하려 하고 있다.

### yaml 표기법

```
# Site
title: Connecting My Life
subtitle: Seong Won's Stoy
description:
author: Seong Won
language: ko-KR
timezone: Asia/Seoul
```

## JINJA



- YAML은 정적인 파일이기 때문에 파이썬 개발자들은 템플릿을 활용하여 생명력을 불어넣음
- 파이썬의 템플릿 언어

## Inventory File

- 리모트 서버에 대한 meta 데이터를 기술하는 파일입니다.
- 위치
  - /etc/ansible/hosts
  - MacOS X -> /usr/local/etc/ansible/hosts
- 따로 인벤토리 파일을 쓸 수 있는 -i 옵션을 사용할 수 있습니다.

- 원격 호스트를 그룹화 하여 관리할 수 있습니다.

```
ansible seminar -m ping (-m 모듈이름)
ansible seminar -a "df -g" (-a 모듈 파라미터)
ansible beta -i hosts -m command -a '/sbin/reboot' (따로 만든 hosts파일 사용시)
```

## Template

- yml 파일 뿐 아니라 모든 파일에서 활용 가능 (if, for문과 같이 variable을 넣을 수 있음)
- 관례상 파일 확장자명을 .j2로 함
  - index.php.j2
  - mysql.cnf.j2
- template task 일때 jinja2가 적용가능 (copy task는 적용안됨)

```
ex) mysql 배포한다고 가정할때...
[client]
default-character-set = utf8
---
[client]
{% if utf8 %}
#apply utf8
default-character-set = utf8
{% endif %}
```

## Playbook

- playbookd은 ansible의 환경설정, 배포를 가능하게 함
- yaml 문법을 채용하여 정책을 기술
- linux 기반 권한 관리 (user, group) 지원
- Play의 목적은 여러 호스트들에게 잘 정의된 'role'과 'task'를 매핑하는 역할을 합니다.
- 하나의 playbook은 하나 또는 하나 이상의 'play'를 가질 수 있습니다.

host - 적용할 호스트의 이름 또는 그룹명 (필수)

```
hosts: all
hosts: hostname
hosts: grupname
hosts: grupname1, grupname2
hosts: groupname1, hostname2
hosts: naver.com
```

sudo - 디폴트는 false

```
sudo = true
```

user - 디폴트는 root

```
user: username
```

## vars

```
group_name: wisoft
web:
  play: 192.168.0.25
  tomcat: 192.168.0.26
config: ssl.conf
config_path: /apache/$config
```

tasks에서 사용

```
{{group_name}}
{{web.play}}, {{web.tomcat}}
{{config}}
{{config_path}}
```

more with var

```
is_ubuntu: '{{ansible_distribution}}' == 'ubuntu'
is_debian: '{{ansible_distribution}}' == 'debian'
```

vars file - 변수 파일 지정

```
vars_files:
  - /vars/vars_file.yml
  - /vars/${hostname}.yml
```

## Task

- ansible module을 호출하는 단위 (필수)

task 종류 - 간단 task : name / action

```
- name : check server's aliveness (주석과 같이 사용할 수 있음)
  action : ping
```

task 종류 - Ansible 모듈 이용 task 사용가능

```
- copy: src=/src/myfiles/foo.conf dest=/etc/foo.conf owner=foo group=foo mode="u=rw, g=r, o=r"
< name없이 리모트 서버로 설정 가능 >
```

more with task

- include - 변수 값을 지정해서 include yaml 파일로 넘길 수 있습니다.

```
include: task/add_user.yml user=www group=www
```

- when (conditional)

```
- apt: name=$item state=latest
  with_items:
    - ntp
  when: ansible_distribution == 'Ubuntu'

- yum: name=$item state=latest
  with_items:
    - ntp
  when: ansible_distribution == 'CentOS'
```

event 발생 (task -> handler)

- task

```
- name: Hello PHP script
  template: src=index.php.j2 dest=/var/www/
  index.php mode=0666
  notify: Restart Apache
```

- handler

```
handler:
- name: Request Apache
  service: name=apache2 state=restarted
```

< 어떤 파일이 수정이 될 경우 자동으로 재시작 될 수 있도록 작성 할 수 있음 >

## role

- structure 기본 단위로서 설치, 사용이 가능
  - apache
  - redis
  - java

- jenkins
- tomcat

- remtoe hosts 연결없이 local로 동작할 수 있음

ansible 명령어

```
ansible-playbook playbook.yml --connection=local
```

yaml 파일

```
-hosts: 127.0.0.1
  connection: local
```

# ANSIBLE PLAYBOOK

- 이제 Ansible을 활용하여 원격 서버에 프로그램을 설치해보도록 하겠습니다.

## 1. Server 연결 playbook

```
$ vi test.yml

---
- name: Connect Test
  hosts: all
  remote_user: root
  # remote_user: user
  # sudo: yes // 다음과 같은 표기 방법은 더이상 사용되지 않습니다.

$ ansible-playbook -i hosts test.yml

PLAY [Connect Test] *****
*****

TASK [Gathering Facts] *****
*****
ok: [203.230.100.59]

PLAY RECAP *****
*****
203.230.100.59          : ok=1    changed=0    unreachable=0    failed=0
```

## 2. apt를 활용하여 프로그램 설치 배포

```
$ vi install_apr.yml
```

```
---
- hosts: all
  become: true
  become_method: sudo
  tasks:
    - name: install packages
      apt: name={{item}} state=latest update_cache=yes
      with_items:
        - nmap
        - htop
        - build-essential
        - mysql-server
```

```
$ ansible-playbook -i hosts install_apr.yml
```

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

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

```
ok: [203.230.100.59]
```

```
TASK [install packages] *****/***
*****
```

```
ok: [203.230.100.59] => (item=[u'nmap'])
```

```
PLAY RECAP *****
*****
```

```
203.230.100.59 : ok=2 changed=0 unreachable=0 failed=0
```