

# ANSIBLE SEMINAR #1

---

1. ANSIBLE INTRODUCE
  2. ANSIBLE Special Feature
  3. How to Install?
    - Mac OSX
    - Debian / Ubuntu
    - Free BSD
    - yum
  4. Getting Started
-



# ANSIBLE

본 세미나는 Ansible 공식 홈페이지와 국내 문서를 바탕으로 제작되었습니다

이번 시간에는 Ansible의 소개 / 특징 / 설치법에 관한 정보가 포함되어 있습니다.  
실 사용법과 서버 관리에 대한 부분은 다음 시간에 이어서 설명하도록 하겠습니다.

---

## 1. ANSIBLE INTRODUCE

### 1) Ansible이란?

- \* 시스템 환경 설정 및 애플리케이션 배포 자동화 플랫폼
- \* 에이전트가 없는 구조, 에이전트 관리에 신경을 쓰지 않아도 됨.
- \* SSH를 통해서 통신함.
- \* Provision & configuration management tool
  - \* Provision : 시스템 구성을 위한 일련의 활동을 말함
- \* Python Github project 중 상위 랭킹 (6위)
- \* 오픈 소스 버전 (GPL)
- \* Enterprise 버전이 존재하며 ansible 오픈 소스에 UI 와 일부 utility 추가한 상용 버전

## 2) Ansible을 선택 하는 이유

- \* SSH 통신을 사용하기 때문에 빠른 provision 이 가능합니다.
- \* 추후 상용 환경에서 사용할 때 agent 기반이면 방화벽 이슈, agent 데몬 관리라는 불편한 점이 존재합니다
  - agent 방식의 장점도 물론 존재합니다. 확장성, 대규모 provision을 할 경우 매우 효과적입니다.
  - 대신 서버와 통신하는 부분이 고도화되기 때문에 빠르고 간단한 provision을 할 수 없습니다.
- \* 자동 배포 환경이 쉽습니다.
- \* 개발 가능성이 높은 오픈소스 입니다.
- \* 먹등성을 제공합니다.

### 먹등성이란 ?

연산을 여러 번 적용하더라도 결과가 달라지지 않는 성질을 먹등성 이라고 합니다.

### Ansible에서 먹등성이란?

여러 번 ansible 툴을 사용하더라도 동일한 결과값을 나올 수 있도록 제공되는 형태여야 합니다. 매번 다른 결과가 나오거나 예러가 나온다면 비 먹등성하다고 할수 있습니다.

Ansible 툴의 거의 대부분의 모듈은 먹등성을 제공합니다. 또한 먹등성을 제공하기 위해서 조건절을 제공하고 있습니다.

예를 들면, 처음 ansible 스크립트를 실행후 다시 실행을 하면 상황에 따라서는 파일이 append가 될 수 있습니다. 그러나 먹등성의 원칙은 언제나 실행은 해도 결과가 동일하게 나옵니다. 또한 파일/디렉터리를 생성 또는 삭제하는 'create', 'remove:'

같은 ansible 모듈을 실행 할 때 'when;' 조건절을 이용할수 있습니다. 대부분의 ansible 모듈이 먹등성을 보장한다는 의미는 상태를 파악할 수 있다는 의미를 가지게 됩니다.

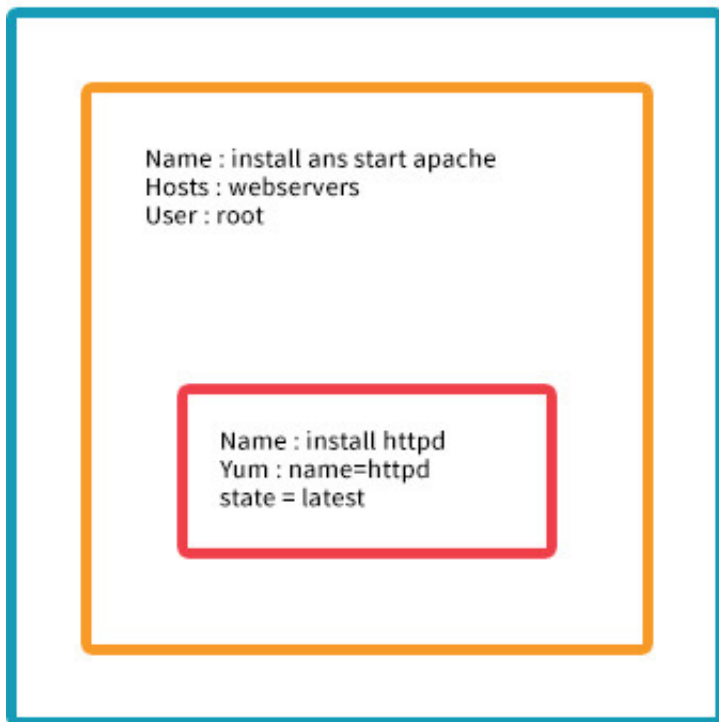
## 2. ANSIBLE Special Feature

- Ansible의 환경설정, 배포를 가능케 하는 언어입니다.
- 리모트 서버에 접속해서 무언가를 시행시키는 정책을 기술합니다.
- Yaml 문법으로 기술되어 있으며, 고급 단계에서는 로드밸런서를 모니터링 하는 복잡한 환경에서 사용할 수 있도록 함

니다.

- playbook은 하나 또는 하나 이상의 'play'를 두게 됩니다.
  - Play의 목적은 여러 호스트들에게 잘 정의된 'role'과 'task'를 매핑하는 역할을 합니다.
  - Task는 ansible 모듈의 호출을 의미합니다.
  - Role을 좀 더 편하게 관리하기 위해서 미리 정의된 yaml파일을 include하는 것이 가능합니다.
- host inventory파일에 정의한 서버 그룹별로 각각 나누어 provision 할 수 있도록 할 수 있습니다.
- 서버당 디렉토리를 나눠서 각각의 설정 정보가 정의된 파일을 읽어 설치하게 됩니다.

## PLAYBOOK



● Playbook      ● Play      ● Tasks

## Inventory

리모트 서버에 대한 meta 데이터를 기술하는 파일입니다. Ansible에서는 inventory 파일에는 yaml을 적용하지 않았습니다. 기본 파일은 /etc/ansible/hosts를 읽게 하거나, 따로 inventory 파일을 만들고 옵션을 주어 동작하게 할 수 있습니다.

만약 고정 ip를 가지고 있고 hosts 파일 안에 들어가 있지 않는 서버가 있다면 설정 파일을 만들수 있고 테스트 환경을 만들때 유용합니다.

## 3. How to install?

### - MAC

```
$ brew search ansible
$ brew install ansible
```

---

## - Debian / Ubuntu

```
$ deb http://ppa.launchpad.net/ansible/ansible/ubuntu trusty main
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 93C4A3FD7BB9C367
$ sudo apt-get update
$ sudo apt-get install ansible
```

---

```
$ sudo apt-get install software-properties-common
$ sudo apt-add-repository ppa:ansible/ansible
$ sudo apt-get update
$ sudo apt-get install ansible
```

---

## - Free BSD

```
$ sudo pkg install ansible
or
$ sudo make -C /usr/ports/sysutils/ansible install
```

---

## - Yum

```
# install the epel-release RPM if needed on CentOS, RHEL, or Scientific Linux
$ sudo yum install ansible

$ git clone git://github.com/ansible/ansible.git --recursive
$ cd ./ansible
$ make rpm
$ sudo rpm -Uvh ./rpm-build/ansible-*.noarch.rpm
```

---

# 4. Getting Start

- ansible 버전을 다음과 같이 확인합니다.

```
$ ansible --version
```

```
ansible 2.1.1.0
  config file = /etc/ansible/ansible.cfg
  configured module search path = Default w/o overrides
```

- 호스트와의 연결을 확인하기 위해서 ping 명령어를 입력합니다.

```
$ ansible all -m ping
```

- 당연히 호스트에 대한 정보를 입력하지 않았기 때문에 확인할 수 없다는 명령어가 나오게 됩니다.
- localhost에서는 연결이 되는지 확인해 보겠습니다.

```
$ ansible localhost -m ping
```

```
localhost | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

```
$ ansible localhost -a "/bin/echo hello"
```

```
localhost | SUCCESS | rc=0 >>
hello
```

## 관리 대상 목록 (Inventory)

Ansible에서는 관리 대상 목록을 Inventory라고 합니다. 이 대상 목록은 디폴트로 /etc/ansible/hosts에 정의되어 있습니다.

하지만 -i 옵션을 이용하여 명령행에서 지정할 수도 있습니다. 이것은 고정된 것 뿐만 아니라 동적 관리 대상 목록 (Dynamic Inventory)을 이용할 수 있습니다.

- hosts파일의 일반적인 경로는 다음과 같습니다.
  - Ubuntu -> /etc/ansible/hosts
  - MacOS X -> /usr/local/etc/ansible
- hosts파일이 없으면 생성합니다.
- hosts파일의 구성은 다음과 같습니다.

```
mail.example.com
```

```
[webservers] // 그룹을 의미합니다.
```

```
foo.example.com
```

```
badwolf.example.com:5309 // 만약 SSH 포트가 디폴트가 아닐경우 다음과 같이 포트를 지정합니다.
```

```
[dbservers]
```

```
one.example.com
```

```
two.example.com
```

```
three.example.com
```

더 자세한 내용은 URL : [https://mcchae.gitbooks.io/ansible\\_ko/content/ansible\\_c18c\\_ac1c.html](https://mcchae.gitbooks.io/ansible_ko/content/ansible_c18c_ac1c.html)

## SSH접속 설정하기

- ansible은 SSH기반으로 동작하기 때문에 사전 연결설정은 필수입니다.
- 테스트 서버와 통신을 위하여 hosts파일을 다음과 같이 수정합니다.

```
[ubuntu]
```

```
203.230.100.61:22001 ansible_user=wisoft
```

```
ansible all -m ping
```

```
203.230.100.61 | UNREACHABLE! => {
```

```
    "changed": false,
```

```
    "msg": "Failed to connect to the host via ssh: Permission denied (publickey,password).\r\n",
```

```
    "unreachable": true
```

```
}
```

- 위와 같은 오류메시지가 나온다면 SSH설정이 되어 있지 않아, 정상적으로 서버에 접속하지 못하는 상황이 발생합니다.
- 이를 해결하기 위해서 잠시 SSH통신 방식에 대해 살펴보도록 하겠습니다.

## SSH KEY란?

서버에 접속할 때 비밀번호 대신 key를 제출하는 방식입니다.

SSH KEY는 언제 사용하는가?

- \* 비밀번호 보다 높은 수준의 보안을 필요로 할때
- \* 로그인 없이 자동으로 서버에 접속 할 때

## SSH KEY가 동작하는 방식



SSH Client

Private Key



SSH Server

Public Key

SSH Key는 공개키(public key)와 비공개 키(private key)로 이루어지는데 이 두개의 관계를 이해하는 것이 SSH Key를 이해하는데 핵심이다. 키를 생성하면 공개키와 비공개키가 만들어진다. 이 중에 비공개키는 로컬 머신에 위치해야 하고, 공개키는 리모트 머신에 위치해야 한다.

(로컬 머신은 SSH Client, 원격 머신은 SSH Server가 설치된 컴퓨터를 의미한다.)

**SSH 접속을 시도하면 SSH Client가 로컬 머신의 비공개키와 원격 머신의 비공개키를 비교해서 둘이 일치하는지를 확인한다.**

출처 : 생활코딩

## SSH KEY 설정 방법

### - Client

- SSH KEY가 설정되어 있지 않다는 것을 가정하고 진행하도록 하겠습니다.



```

$ ssh-keygen -t rsa
$ Generating public/private rsa key pair.
$ Enter file in which to save the key (/home/axl/.ssh/id_rsa):
$ Enter passphrase (empty for no passphrase): <Type the passphrase>
$ Enter same passphrase again: <Type the passphrase>
Your identification has been saved in /home/axl/.ssh/id_rsa.
Your public key has been saved in /home/axl/.ssh/id_rsa.pub.
The key fingerprint is:
0b:fa:3c:b8:73:71:bf:58:57:eb:2a:2b:8c:2f:4e:37

$ ls -al ~/.ssh/
total 24
drwx-----  5 seongwonlee  staff   170  4 26 01:35 .
drwxr-xr-x+ 30 seongwonlee  staff  1020  4 26 02:06 ..
-rw-----  1 seongwonlee  staff  1675  4 26 01:33 id_rsa          // 개인키
-rw-r--r--  1 seongwonlee  staff   422  4 26 01:33 id_rsa.pub      // 공유키
-rw-r--r--  1 seongwonlee  staff   706  4 26 02:36 known_hosts    // 현재까지 연결된 ho
sts 키

$ scp ~/.ssh/id_rsa.pub USER_ID@SERVER_IP:id_rsa.pub
//$ scp -P PORT_NUMBER ~ ~/.ssh/id_rsa.pub USER_ID@SERVER_IP:id_rsa.pub

```

## - Server

- ansible을 통해서 관리하고자 하는 서버에 접속하여 다음과 같이 설정합니다.

```

$ ls -l | grep id_rsa_pub
-rw-r--r-- 1 wisoft wisoft 422 4월 26 01:35 id_rsa.pub
$ cat ~/id_rsa.pub >> ~/.ssh/authorized_keys.           // '>>' 표사는 목적지 파일에 덧붙여쓰기를 말합니다.

```

## SSH를 통해 접속하기

- 클라이언트에서 다음과 같이 SSH에 접속합니다.

```
$ ssh USER_ID@SERVER_IP
```

```
Welcome to Ubuntu 17.04 (GNU/Linux 4.10.0-19-generic x86_64)
```

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage
```

```
Ubuntu 12.04 LTS end-of-life is April 25, 2017 -- Upgrade your Precise systems!
```

```
$ sudo do-release-upgrade -m server
```

```
0 packages can be updated.
0 updates are security updates.
```

```
*** System restart required ***
```

```
Last login: Tue Apr 25 01:33:32 2017 from 192.168.0.28
```

- 위와 같이 비밀번호 입력없이 접속에 성공한다면 모든 설정이 완료됩니다.

## SSH 접속시 오류가 발생한다면?

```
$ ssh -v USER_ID@SERVER_IP
```

- -vv, -vvv 옵션을 사용하면, 더 자세한 내용을 확인할 수 있습니다.

## ansible 접속하기

- 위에 모든 설정을 완료했다면 ansible을 활용해 보도록 하겠습니다.

```
$ ansible ubuntu -m ping
```

```
192.168.0.28 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

```
$ ansible ubuntu -a "/bin/echo hello"
```

```
192.168.0.28 | SUCCESS | rc=0 >>
hello
```

```
$ ansible ubuntu -s -a "apt install nmap -y" --ask-become-pass
```

```
192.168.0.28 | SUCCESS | rc=0 >>
Reading package lists...
Building dependency tree...
```

```
Reading state information...
The following packages were automatically installed and are no longer required:
  linux-headers-4.8.0-22 linux-headers-4.8.0-22-generic
  linux-image-4.8.0-22-generic linux-image-extra-4.8.0-22-generic
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  nmap
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 4,708 kB of archives.
After this operation, 21.7 MB of additional disk space will be used.
Get:1 http://kr.archive.ubuntu.com/ubuntu yakkety/main amd64 nmap amd64 7.12-1 [4,
708 kB]
Fetched 4,708 kB in 1s (4,405 kB/s)
Selecting previously unselected package nmap.
(Reading database ... 241901 files and directories currently installed.)
Preparing to unpack .../archives/nmap_7.12-1_amd64.deb ...
Unpacking nmap (7.12-1) ...
Setting up nmap (7.12-1) ...
Processing triggers for man-db (2.7.5-1) ...
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

$ ansible ubuntu -s -a "nmap localhost" --ask-become-pass

192.168.0.28 | SUCCESS | rc=0 >>

Starting Nmap 7.12 ( https://nmap.org ) at 2017-04-26 03:47 KST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000020s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
631/tcp    open  ipp

Nmap done: 1 IP address (1 host up) scanned in 1.62 seconds
```