<Ansible>

Ansible 은 **여러 개의 서버를 효율적으로 관리하기 이해 고안된 환경 구성 자동화 오픈 소스 도구**이다.

여러 서버를 구성할 때 사용하는 가장 기본적인 방식은 shell script를 만들어서 돌리는 방식이나, 서버의 대수가 많아지고 동시에 환경을 구성해야 하는 일이 발생한다면 기존의 shell script로는 한계가 있다.

이를 위해 고안된 개념이 **Infrastructure as a Code이며 환경의 배포와 구성을 규격화 된 코드로 정의해 사용하는 것을 의미**한다.

Ansible은 이러한 개념을 바탕으로 생성된 툴이다. Python으로 개발되었고 Yaml언어를 통해 정의할 수 있다.

1. 특징

1) Agentless

Chef/Puppet과 같은 기존 IaC 솔루션들은 원격 서버에 에이전트를 설치할 필요가 있었다. 따라서 명령을 내려주는 Controller 서버와 원격 서버에 설치된 Agent들이 명령을 주고받는 방식으로 동작했다.

그러나 **앤서블(Ansible)은 SSH를 기반으로 원격 서버에 명령을 전달하기 때문에 에이전트가 필요 없다.**

Agent가 필요 없다는 건, 각 원격 서버에 접속해서 agent를 설치해줄 필요가 없다는 것이다. agent 설치 단계를 제거하여 인프라 구축을 더 자동화에 가깝게 만든 것이다.

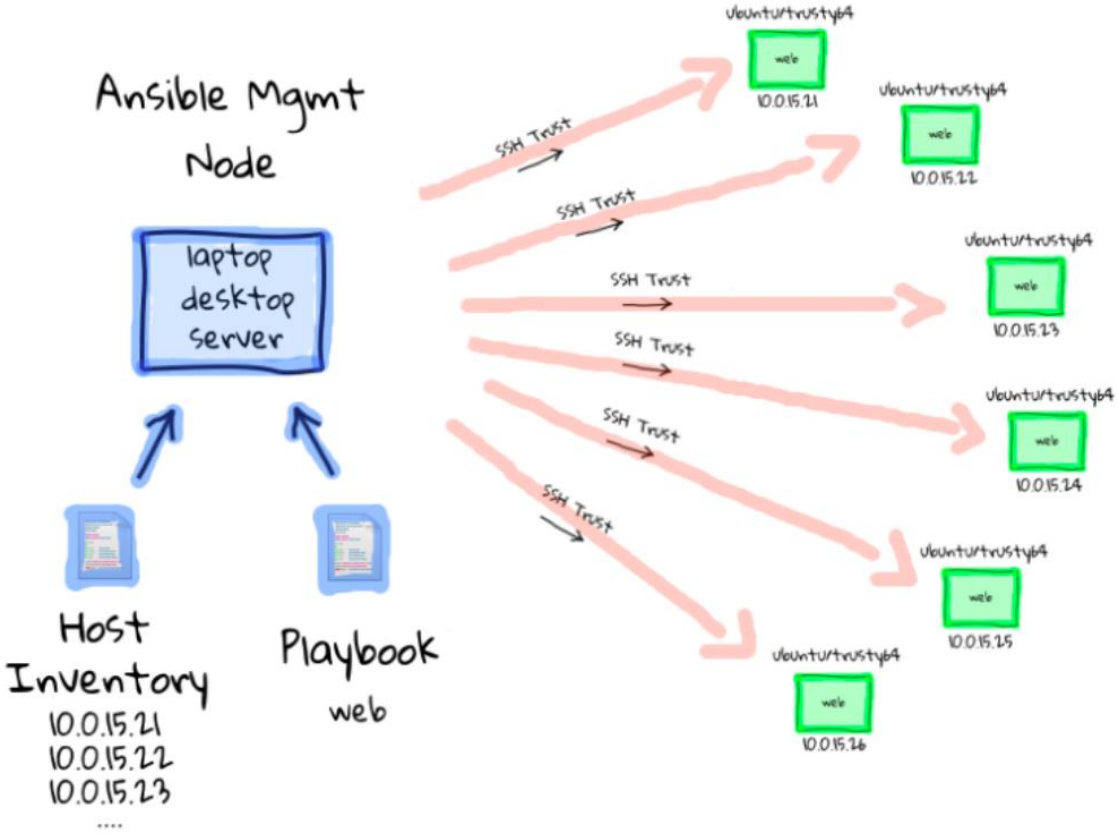
2) 접근 용이성

앤서블은 명령 모음집(playbook)을 YAML형식의 파일로 관리한다. Yaml 형식의 파일로 관리하며 명령어들을 모아서 한번에 처리 할 수 있다.

3) 멱등성 (idempotence)

멱등성이란 여러번 수행해도 같은 결과를 뱉는 성질을 말한다. 앤서블은 YAML로 관리되는 Playbook을 여러 번 수행하더라도 언제나 같은 결과가 나올 수 있도록 여러가지 관리를 한다.

2. Architecture & Component



1) Control & Managed Node

에이전트 없이 SSH 데몬으로 통신만 가능하다면 Ansible로 관리할 수 있다. 다만 Python 베이스로 ansible이 동작하기 때문에 control과 Managed 노드 모두 python 이 설치되어 있어야 한다.

2) Component

- **Module**

미리 정의해둔 실행 단위 이다. 다양한 역할의 모듈이 존재하고 단일 모듈을 호출해서 사용할 수도 있으며 Playbook에서 여러 다른 모듈을 조합해서 사용할 수도 있다.

- **Task**

Ansible의 작업 단위이다. 각 Tasks는 모듈의 집합이다.

- **Playbook**

Tasks들을 실행 순서대로 저장해 놓은 작업들의 리스트이다. YAML 형태로 작성된다.

- **Inventory**

관리되는 노드들의 목록이 담긴 파일이다. /etc/ansible/hosts 파일에 원격 서버들 목록이 저장되어 있다. Ansible은 Inventory 파일을 참고해서 Playbook을 실행한다.

3. Hosts 파일 편집 및 Test

Control Node 와 Managed Node에 python3을 설치하고, Control Node에 Ansible 설치한다. Control Node와 Managed Node의 ssh 통신을 확인한다.

#vi /etc/ansible/hosts 파일을 편집한다.

```
[all]
172.31.10.32 #suejin

[all:vars]
ansible_python_interpreter=/usr/bin/python3
```

1) Ping Test

```
ansible@test03:~$ ansible -m ping all
172.31.10.32 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

-m : 모듈 이름

All : hosts 파일에서 설정한 리스트의 이름

2) Raw command Test

모듈 말고 리눅스 명령어를 실행하고 싶을 때에는 raw를 사용한다.

```
ansible@test03:~$ ansible -m raw -a 'uptime' all
172.31.10.32 | CHANGED | rc=0 >>
 04:36:42 up 26 min,  2 users,  load average: 0.02, 0.11, 0.10
Shared connection to 172.31.10.32 closed.
```

4. Playbook 만들어 보기

1) Inventory 생성:

vim hosts.inv

```
[suejin]
172.31.10.32

[suejin:vars]
ansible_python_interpreter=/usr/bin/python3
```

2) Playbook 작성:

#vim playbook-test.yaml

```
- hosts: suejin
  become: yes
  tasks:
  - name: Install packages
    apt:
      name:
      - ntpdate
      state: latest
      cache_valid_time: 3600
```

Hosts: playbook 이 적용될 Node 리스트의 이름

Become: 아래의 task를 sudo권한으로 실행할 거면 yes

Name: task의 이름 (해당 task가 무슨 역할을 하는지 적는다)

Apt : task 에서 실행할 모듈의 이름, 위 예시에서는 ntpdate라는 패키지를 실행하기 위한 모듈 apt를 적었다.

3) Playbook 실행

Reference

1. 앤서블 architecture – 앤서블 철저입문
2. 특징 - https://wickso.me/ansible/basic/
3. 구조와 설치방법 - https://gruuuuu.github.io/ansible/ansible-basic/
4. Playbook - https://medium.com/@wintonjkt/ansible-101-getting-started-1daaff872b64
5. https://www.techtarget.com/searchitoperations/definition/Ansible