



ANSIBLE

# 1장

Ansible 소개 및 설치

# 전체 내용

Ansible 소개

Ansible  
설치

Ansible  
기본 사용

# Ansible 소개

Ansible이란?

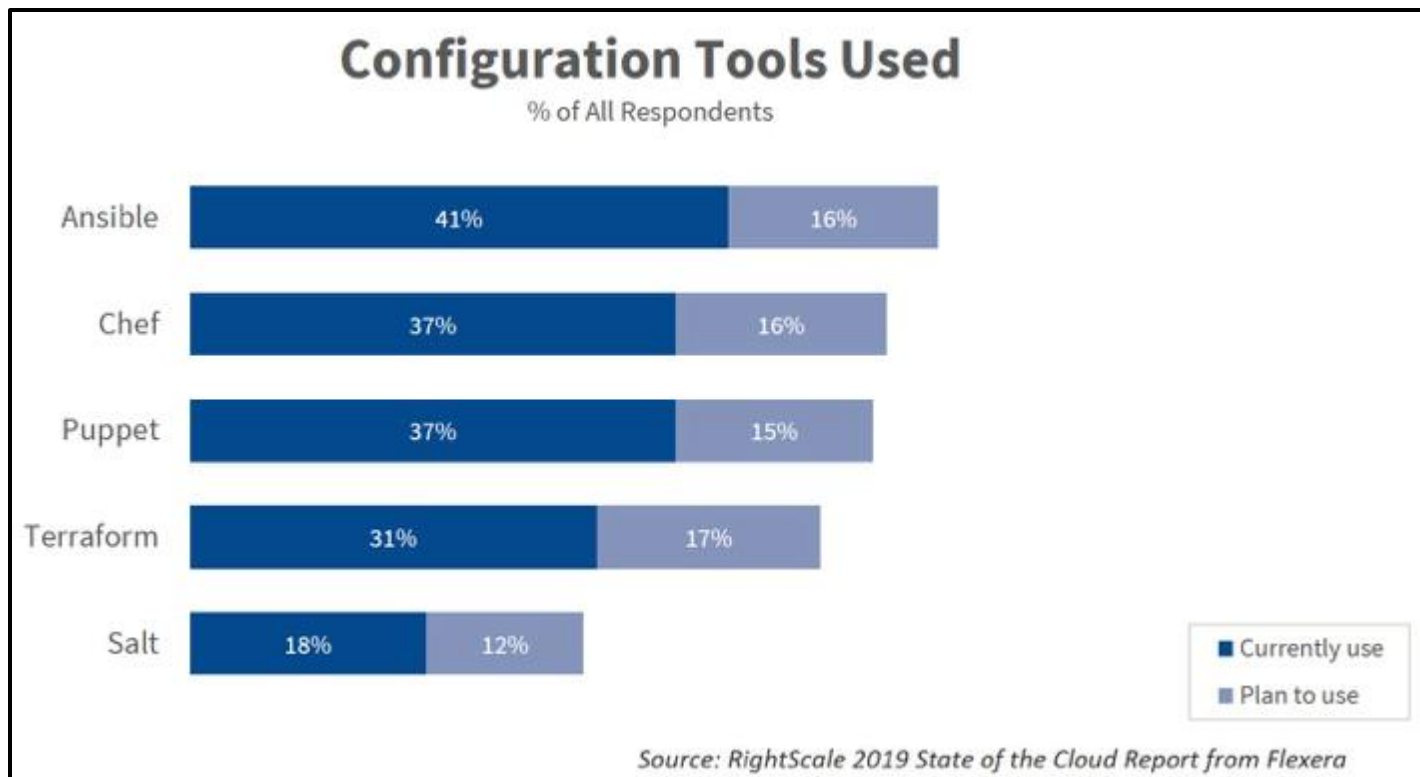
Ansible 장점

- **Ansible이란?**

- 주요 Configuration Management Tools

- 출처: RightScale 2019 State of the Cloud Report from Flexera

- <https://resources.flexera.com/web/media/documents/rightscale-2019-state-of-the-cloud-report-from-flexera.pdf>



- **Ansible이란?**

- Ansible은 **Open Source 자동화 플랫폼**이다

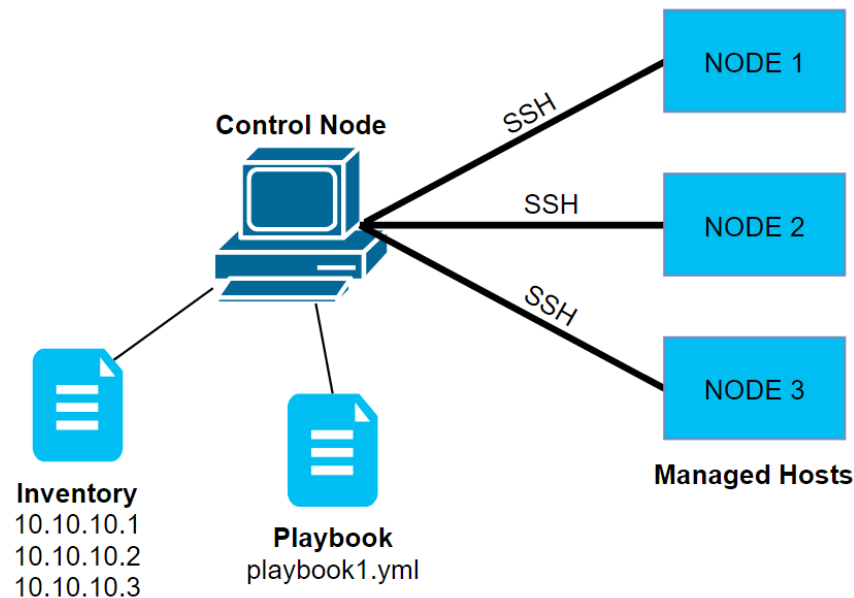
- 구성 관리, 애플리케이션 배포, 작업 자동화 및 IT orchestration을 하는데 도움이 된다
- 매우 단순하여 설치 및 구성이 쉽고 효율적이고 매우 강력하다
- 다양한 종류의 Linux 배포판 Server들을 자동으로 관리하는 도구
  - WinRM을 사용하는 Windows Server도 가능하다
- 수동적이고 반복적으로 진행했던 IT Infra 관리(배포 및 구성 관리)를 손쉽게 자동화하여 생산성을 높이는 도구

## • **Ansible**이란?

- Ansible은 **Open Source 자동화 플랫폼**이다
  - "곡괭이로 작은 밭을 가는 농부" vs "트랙터로 대규모 밭을 가는 농부"
    - 빨리 일을 끝낸 후에는 농약을 뿌리고 주기적으로 스프링쿨러로 물을 주기도 한다
  - 앞 선 작업 후에 그 다음 작업을 하도록 **업무 흐름을 자동화**한다
  - **반복적인 작업을 자동화**한다
    - 일정한 시간이 되면 정해진 작업을 자동으로 할 수 있다
    - 사람으로 인한 **운영의 오류를 최소화**한다
    - 서비스 자동화, 운영 자동화, 배포 자동화, 네트워크 자동화, 보안 및 Compliance 관리

## • Ansible이란?

- DevOps도구로서 **IT인프라를 쉽게 관리하기 위한 도구**
  - Code로서 인프라를 구축하기 때문에 Infrastructure as Code라고 불린다
  - 시스템 구성시 다양한 애플리케이션을 환경에 맞게 설치하고 관리해야 하는데 ansible은 이러한 일련의 작업들을 자동화하는데 도움을 주는 도구이다



## • **Ansible 장점**

- 관리되는 컴퓨터에 **Agent**를 설치할 필요가 없음
  - 관리하는 **Ansible 서버에만** Ansible을 설치한다
  - Agent 관리, 추가 방화벽 관리 및 데몬 관리 **불필요**
  - Ansible server의 시스템 요구조건이 매우 낮다
  - SSH/WinRM와 Python을 사용하여 관리되는 컴퓨터에 연결한다
  - SSH를 사용하기 때문에 관리를 위한 배포가 매우 빠르다
- 구매 비용도 없고 Open Source라는 것
  - 자발적으로 많은 Community가 운영되고 있음
- **구성 관리**를 위해 많은 Module 보유
  - 구성관리를 위해 750여개의 내장된 모듈 이용 가능



## • **Ansible 장점**

- 멱등성(idempotency) 제공
  - 수학 연산을 여러 번 적용하더라도 결과 값이 달라지지 않는 성질
  - 재수행 시 기 수행된 코드는 반복 수행되지 않음
- 성능이 좋다
- 설치가 쉽고 단순하여 배우기가 쉽다
  - Ansible은 구성 파일에 YAML Syntax를 사용한다
  - Playbook을 사용할 때 Yaml 파일을 사용하게 되는데, 내용이 영어 문장과 비슷하여 구성 내용이 쉽게 이해 된다
  - YAML 파일을 이용하여 Infrastructure as Code 구현
- Bash Script의 한계를 극복

- **Why do we use Ansible?**

1. Ansible is **free and Open Source**
2. **Agentless** on managed nodes
3. System requirement of **control node is very low**
4. Ansible uses **SSH for the operations**
5. Lightweight and Quick Deployment
6. Ansible users **YAML syntax in configuration files**
7. Larger community
8. Developed **in Python**

- **Ansible 주요 요소**

- **Inventory:** 자동화 대상(managed node)
  - 자동화 관리 대상 컴퓨터를 의미
  - 호스트 개별 또는 그룹별로 구분하여 작업 가능
- **Module:** 자동화 기능이 내장된 패키지
  - Ansible 내장 자동화 module
  - Server, Storage, Network, Cloud, Database 모듈 등 다양함
  - 2020년 7월 기준 약 1,600여개의 모듈 제공(계속 추가 중)
- **Playbook:** 자동화 코드
  - 자동화 기능을 수행하는 Code Set
  - EC2 인스턴스 생성 Playbook, 보안설정 Playbook 등등
- **Ansible Galaxy:** Playbook 공유 Hub
  - Playbook 공식 공유 Hub로서 Playbook 재사용 및 참고에 유용함
  - Open Source code 공유

- **Ansible에 사용되는 주요 용어(Terminology)**

1. Playbooks
2. Task
3. Tasks
4. Actions
5. Facts
6. Host
7. Groups
8. Handlers
9. Ad-hoc
10. Inventory

11. Modules
12. Library
13. Notify
14. Orchestration
15. Push mode
16. Pull mode
17. Roles
18. Templates
19. When (optional)
20. YAML

# Ansible 소개

- **Ansible이 지원되는 OS**

- 참고: <https://bit.ly/3vbDASv>
- Control node
  - Python 2.7 / Python 3.5 이상 요구
  - Red Hat, Debian, CentOS, macOS, BSD 등등(##Windows는 지원 안 함)
- Managed node
  - Python 2.6 / Python 3.5 이상 요구
  - SSH가 되는 모든 OS
  - Cisco 장비 포함



# Ansible 설치

Ansible Lab 환경 구성

Ansible 설치

Ansible 기본 사용

# Ansible 설치

- Ansible Lab 환경 구성
  - Ansible Server(Control node) 1대
    - CentOS 7.x
    - Ansible 설치하기
  - Ansible Node(Managed node) 4대
    - CentOS 7.x: 3대
    - Ubuntu Server 18.04: 1대
    - 각 노드는 기본 설치만 하면 된다(Agent 설치 필요 없음)
  - 인증 방식
    - root 계정에 대하여 Public Key, Private Key로 인증하기
  - 기타
    - root 계정에 암호: 1
    - Ubuntu는 root 계정이 Remote로 접속할 수 있도록 환경 구성 필요
    - 단지 SSH로 접속하여 관리

# Ansible 설치

- Ansible Server 설치하기
  - CentOS 7.x 설치하기
    - root 계정에 대한 암호: 1
    - selinux 및 firewalld 중지
    - yum update
    - hostname: ansible
- 관리되는 Node 설치하기
  - 먼저 설치한 Ansible Server를 3대 복제하기
    - hostname: node1, node2, node3
  - Ubuntu Server 18.04 설치하기
    - adminuser 계정에 대한 암호: 1
    - firewalld 중지
    - apt update
    - hostname: node4



# Ansible 설치

- Ubuntu Server 18.04 구성하기
  - root 계정 사용하도록 설정하기
    - `sudo passwd -u root (##-u: unlock)`
  - root 계정 암호("1"로 설정) 지정하기
    - `sudo passwd root`
  - 참고: root 계정에 대한 암호 삭제 및 잠그기
    - `sudo passwd -dl root (##-d: delete, -l: lock)`
  - root 계정으로 원격에서 접속하도록 허용하기
    - `su root`
    - `vi /etc/ssh/sshd_config`

```
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
PermitRootLogin yes
#StrictModes yes
```
    - PermitRootLogin yes를 추가한다

# Ansible 설치

- hosts 파일 구성하기

- 각 노드에 접속하기 위해 IP Address를 사용하기 보다는 hostname을 사용하면 편리하다
- ansible 서버에서 다음과 같이 작업하여 각 node에 복사하기

- vi /etc/hosts

```
127.0.0.1    localhost lo
::1          localhost lo
192.168.56.102 ansible
192.168.56.103 node1
192.168.56.104 node2
192.168.56.105 node3
192.168.56.106 node4
```

- scp /etc/hosts node1:/etc/hosts
  - scp /etc/hosts node2:/etc/hosts
  - scp /etc/hosts node3:/etc/hosts
  - scp /etc/hosts node4:/etc/hosts

# Ansible 설치

- root 계정에 대한 Public key, Private key 인증 구성
  - 원격 컴퓨터에 접속하여 인증할 때 암호 방식이 아닌 Private key, Public key 방식으로 하면 편리하고 보안성도 좋다
    - 각 사용자 별로 Key를 만들어서 원격 컴퓨터에 복사해둔다
- root 계정에 대한 Private key와 Public key 생성하기
  - **whoami** (##root 계정인 것을 확인)
  - **ssh-keygen** (##private key, public key 생성)
- root 계정의 public key를 원격 컴퓨터에 복사하기
  - **ssh-copy-id root@node1**
  - **ssh-copy-id root@node2**
  - **ssh-copy-id root@node3**
  - **ssh-copy-id root@node4**
- 원격 컴퓨터에 접속하기
  - **ssh node1** (##성공)

# Ansible 설치

- Ansible 서버(ansible) 구성하기
  - ansible 서버에 **ansible** 패키지 설치하기
    - yum install **ansible** (##실패)
    - yum install **epel-release** -y
      - ##ansible 패키지가 저장된 repository 추가하기
    - **yum install ansible** -y
  - ansible 도움말 확인하기
    - **ansible --help**
  - ansible 버전 확인하기
    - **ansible --version**
    - **python --version**

```
[root@ansible ~]# ansible --version
ansible 2.9.27
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.
  ansible python module location = /usr/lib/
  executable location = /usr/bin/ansible
  python version = 2.7.5 (default, Nov 16 2015, 12:24:44)
```

# Ansible 설치

- Ansible 서버(ansible) 구성하기
  - ansible로 시작하는 명령어 확인하기
    - **compgen -c | grep ansible** (##여러 개의 유용한 명령어가 있다)
      - **ansible**
      - **ansible-playbook**
      - **ansible-doc**
      - **ansible-pull**
      - **ansible-vault**
      - **ansible-config**
      - **ansible-connection**
      - **ansible-console**
      - **ansible-galaxy**
      - **ansible-inventory**

# Ansible 기본 사용

- Ansible 서버 구성 파일 확인하기

- ansible 구성 파일

- **ansible --version**

- **cd /etc/ansible**

- **ls -l**

```
[root@ansible ansible]# ls -l
total 28
-rw-r--r-- 1 root root 19985 Jan 16 06:55 ansible.cfg
-rw-r--r-- 1 root root 1016 Jan 16 06:55 hosts
```

- /etc/ansible/ansible.cfg (##환경설정 파일)

```
# some basic default values...

#inventory      = /etc/ansible/hosts
#library        = /usr/share/my_modules/
#module_utils   = /usr/share/my_module_utils/
```

- /etc/ansible/hosts (##Ansible이 접속하는 호스트-Managed nodes)

# Ansible 기본 사용

- Inventory 설정하기

- inventory 뜻

- 기업에서 관리하는 **재고 품목**

- ansible에서는 관리해야 하는 호스트들을 말한다(Managed nodes)

- ansible server가 관리할 호스트들의 목록

- 여기에 group별로 ip address 및 host name으로 등록

- 그룹별 지정하기: **[nginx], [httpd], [centos], [ubuntu]** 등등

- **vi /etc/ansible/hosts**

- 우측 그림처럼 입력한다

- 그룹별로 구분하기: [ ] 사용

- **Managed node 확인하기**

- **ansible-inventory --list -y**

- **cat /etc/ansible/hosts**

```
## db-[99:101]-node.example.com
[centos]
node1
node2
node3

[ubuntu]
node4

[nginx]
node1
node2

[httpd]
node3
```

# Ansible 기본 사용

- Inventory 설정하기

- /etc/ansible/hosts 외에 그냥 파일을 만들어서도 가능하다
- docker를 실행중인 호스트 파일 생성
  - **vi /etc/ansible/docker.host**  
node1  
node3
- 이렇게 생성한 후 -i 옵션으로 사용할 수 있다
  - **-i: --inventory-file**
  - **ansible all -i /etc/ansible/docker.host -m ping**
  - ## 모든 호스트 중에서 docker.host에 기록된 호스트에게만 ping을 한다
  - **-m: --module-name**



# Ansible 기본 사용

- Inventory 확인하기
  - ansible server가 관리할 **모든 호스트**
    - **ansible**
    - **ansible all --list-hosts** (##all: 모든 호스트)
- 특정한 기능으로 묶은 그룹 호스트들(관리 대상 그룹 확인)
  - **cat /etc/ansible/hosts**
  - **ansible-inventory --list -y**
  - **ansible centos --list-hosts**
  - **ansible ubuntu --list-hosts**
  - **ansible nginx --list-hosts**
  - **ansible httpd --list-hosts**

# Ansible 기본 사용

- Known\_hosts 에 등록하기
  - 본격적으로 관리하기 전에 host들과 통신하여 목록에 추가한다(연결 확인하기)
    - **ansible all -m ping** (##-m: module)
      - ##여기서 실패한(Failed) 사람은 public key, private key 설정을 하지 않았기 때문이다. 다음과 같이 password를 입력하여 다시 진행
    - **ansible all -m ping -k** (##-k: password 입력하기)
    - **ansible centos -m ping** (##특정한 호스트그룹만 연결하기)
    - **ansible centos:ubuntu:httpd -m ping** (##여러 개의 호스트그룹에 연결)
  - known\_hosts 확인하기
    - **cat /root/.ssh/known\_hosts**

```
[root@ansible ~]# cat /root/.ssh/known_hosts
node1,192.168.56.103 ecdsa-sha2-nistp256 AAAA
/7s+TwjE3x4p/BJWwnufHYyjEdhDjXoVJaIXXPgazbLm3
node2,192.168.56.104 ecdsa-sha2-nistp256 AAAA
/7s+TwjE3x4p/BJWwnufHYyjEdhDjXoVJaIXXPgazbLm3
node3,192.168.56.105 ecdsa-sha2-nistp256 AAAA
/7s+TwjE3x4p/BJWwnufHYyjEdhDjXoVJaIXXPgazbLm3
node4,192.168.56.107 ecdsa-sha2-nistp256 AAAA
```

# Ansible 기본 사용

- 관리 대상별로 작업하기
  - 관리 대상을 그룹별로 확인하기
    - **cat /etc/ansible/hosts**
- 관리대상 그룹별로 작업하기
  - **ansible centos -m ping**
  - **ansible ubuntu -m ping**
  - **ansible nginx -m ping**
  - **ansible httpd -m ping**
- **ansible all -i /etc/ansible/docker.host -m ping**

# Ansible 기본 사용

- 관리 대상별로 작업하기
  - nginx 그룹(node1, node2)에 nginx 패키지 설치하기
    - **ansible nginx -m yum -a "name=nginx state=present"**
      - ##-a: argument
      - ##present 대신 installed 사용 가능
    - **ansible nginx -m service -a "name=nginx state=started"**
    - **curl node1 ; curl node2**
  - httpd 그룹(node3)에 httpd 패키지 설치하기
    - **ansible httpd -m yum -a "name=httpd state=installed"**
    - **ansible httpd -m service -a "name=httpd state=started"**
    - **curl node3**
    - ## Package를 설치할 때는 굳이 state를 사용하지 않아도 된다
      - **ansible httpd -m yum -a "name=tree"**
      - **ansible httpd -m shell -a "tree -L 2 /home"**

# Ansible 기본 사용

- 관리 대상별로 작업하기
  - nginx 그룹(node1, node2)에 nginx 서비스 중지하기
    - **ansible nginx -m service -a "name=nginx state=stopped"**
    - **curl node1**
  - httpd 그룹(node3)에 httpd 서비스 중지하기
    - **ansible httpd -m service -a "name=httpd state=stopped"**
    - **curl node3**
- 서비스 시작하기
  - **ansible nginx -m service -a "name=nginx state=started"**
  - **ansible httpd -m service -a "name=httpd state=started"**
  - **curl node1**
  - **curl node3**

# Ansible 기본 사용

- 관리 대상별로 작업하기
  - nginx 그룹(node1, node2)에 nginx 패키지 삭제하기
    - **ansible nginx -m yum -a "name=nginx state=absent"**
    - **curl node1 ; curl node2**
    - **##absent 대신 removed 사용 가능**
  - httpd 그룹(node3)에 httpd 패키지 삭제하기
    - **ansible httpd -m yum -a "name=httpd state=removed"**
    - **curl node3**

# Ansible 기본 사용

- 관리 대상별로 작업하기
  - docker.list에 호스트에게 **shell** 명령어 실행하기
    - **ansible all -i /etc/ansible/docker.host -m shell -a "uptime"**
      - ## module 중에서 shell을 사용하면 ansible의 각종 mode에 대한 argument를 사용하는 대신 linux 명령어를 사용하는 것이다.
    - **ansible all -i /etc/ansible/docker.host -m shell -a "df -h"**
      - ## disk 용량 확인하기
    - **ansible all -i /etc/ansible/docker.host -m shell -a "free -h"**
      - ## 메모리 상태 확인하기