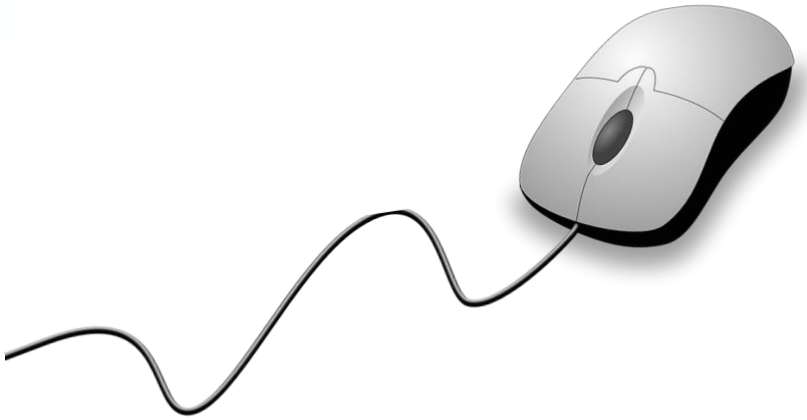
공개SW 솔루션 설치 & 활용 가이드

시스템SW > 자원관리



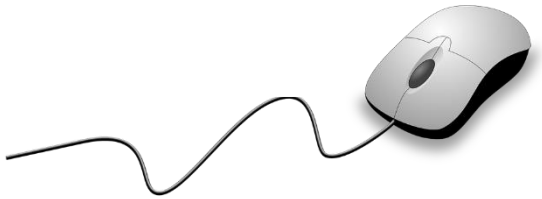
ANSIBLE

제대로 배워보자

How to Use Open Source Software

Open Source Software Installation & Application Guide

nipa 공개SW역량프라자



CONTENTS

1. 개요

소개	• 서버를 시작할 때 미리 설정 파일에 따라 소프트웨어 설치 및 설정을 자동으로 수행함 • 대규모 컴퓨터 클러스터 를 구축 할 때 시간을 단축하고 오류 감소에 도움 • 구성 관리 뿐만 아니라 오케스트레이션 및 소프트웨어 배포 기능을 가짐 • 안정적이고 신뢰성이 높으며, 확장성이 좋음		
주요기능	• 프로비저닝 • 오케스트레이션 • 빌드 및 배포 자동화 등		
대분류	• 시스템 SW	**소분류**	• 자원관리
라이선스 형태	• GNU General Public License	**사전설치 솔루션**	• Python2(2.7) or Python3(3.5 and higher)
실행 하드웨어	• Red Hat, Debian, CentOS, macOS, any of the BSDs • Windows isn't supported for the control machine	**버전**	• v2.6.7 (2018년 10월 기준)
특징	• SIMPLE – 알기 쉬운 자동화 스크립트, 텍스트 기반 플레이북을 사용 기존 형상관리 솔루션 사용 가능 • POWERFUL – Linux, Windows, UNIX 지원, 어플리케이션 자동 배포, 환경 설정 자동 관리 • AGENTLESS – Push-based 방식이며 별도 에이전트 불필요, OpenSSH와 WinRM 사용		
보안취약점	• SSH, WinRM 툴 사용으로 매니지먼트 노드들을 컨트롤 하기에 해당 툴 자체에 취약점이 발생하지 않는 이상 별도 보안취약점은 없음		
개발회사/커뮤니티	• Red Hat / https://www.ansible.com/community		
공식 홈페이지	• https://www.ansible.com		

2. 기능요약

- Ansible의 주요 기능

주요기능	상세내용
프로비저닝	소프트웨어 설치 및 업데이트 구성/설정 변경 파일 전송
어플리케이션 배포	다수의 서버에 어플리케이션 자동배포 전체 어플리케이션을 대상으로 손쉬운 수명 주기 관리 개발부터 상용화까지 자동화 가능
os 자동 설치	가상화 클라우드 기반의 VM을 대상으로 OS 설치 지원 템플릿 기반 VM 생성 Vcenter, Open Stack, AWS, Azure, GCP 등 다양한 가상환경 지원
빌드 및 배포 자동화	CI/CD 환경을 손쉽게 구성 서비스 개발 및 상용화를 빠르게 구현할 수 있는 자동화 플랫폼 어플리케이션 빌드부터 테스트까지 다른 빌드 솔루션과 연동 가능
보안 및 컴플라이언스	Ansible을 통해 보안 정책과 컴플라이언스 수립 시스템 생성 시부터 보안 적용 표준화를 통한 강제성 부여
오케스트레이션	가상머신 생성부터 서비스 제공까지 또는 어플리케이션 롤링 업그레이드 구현

3. 실행환경

•Ansible 요구 사항

- **Python 2 (버전 2.7) 또는 Python 3 (버전 3.5 이상)이 설치된 모든 시스템**
- Windows는 제어 시스템에 지원 안됨
- **Red Hat, Debian, CentOS, macOS, any of the BSDs** 등이 포함됨
- Ansible 커뮤니티 버전의 경우 Python 소프트웨어 외 별도 하드웨어 스펙 요구사항은 없음

4. 설치 및 실행

세부 목차

4. 설치 및 실행

4.1 Ansible Install

- **4.1.1 Git repository에서 Control Machine으로 Ansible 다운로드**

```
[root@localhost ~]# git clone git://github.com/ansible/ansible.git
Cloning into 'ansible'...
remote: Enumerating objects: 245, done.
remote: Counting objects: 100% (245/245), done.
remote: Compressing objects: 100% (211/211), done.
remote: Total 377154 (delta 163), reused 34 (delta 33), pack-reused 376909
Receiving objects: 100% (377154/377154), 125.24 MiB | 4.36 MiB/s, done.
Resolving deltas: 100% (247620/247620), done.
```

- **4.1.2 ansible 환경 적용**

```
[root@localhost ansible]# source ./hacking/env-setup –q
[root@localhost ansible]# echo $PATH
/root/ansible/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

5. 기능소개

세부 목차

5.1.1 ansible

- 용도
  - 단일 작업 실행
- 사용법
  - [root@localhost ansible]# ansible <host-pattern> [options]
- 사용 옵션
  - URL : https://docs.ansible.com/ansible/latest/cli/ansible.html

5.1.2 ansible-config

- 용도
  - Ansible 설정파일 표시
- 사용법
  - [root@localhost ansible]# ansible-config [view|dump|list] [--help] [options] [ansible.cfg]
- 사용 옵션
  - URL : https://docs.ansible.com/ansible/latest/cli/ansible-config.html#

5.1.3 ansible-console

- 용도
  - 선택한 인벤토리에 대해 작업 실행 가능한 REPL 생성
- 사용법
  - [root@localhost ansible]# ansible-console [<host-pattern>] [options]
- 사용 옵션
  - URL : https://docs.ansible.com/ansible/latest/cli/ansible-console.html

5.1.4 ansible-doc

- 용도
  - 사용 가능한 모듈에 대한 정보 표시
- 사용법
  - [root@localhost ansible]# ansible-doc [-l|-F|-s] [options] [-t <plugin type>] [plugin]
- 사용 옵션
  - URL : https://docs.ansible.com/ansible/latest/cli/ansible-doc.html

5.1.4 ansible-galaxy

- 용도

  - Ansible 공유 저장소 관리

- 사용법

  - [root@localhost ansible]# ansible-galaxy [delete|import|info|init|install|list|login|remove|search|setup] [--help] [options] ...

- 사용 옵션

  - URL : https://docs.ansible.com/ansible/latest/cli/ansible-galaxy.html

5.1.5 ansible-inventory

- 용도

  - 구성된 인벤토리를 표시

- 사용법

  - [root@localhost ansible]# ansible-inventory [options] [host|group]

- 사용 옵션

  - URL : https://docs.ansible.com/ansible/latest/cli/ansible-inventory.html

5.1.6 ansible-playbook

- 용도
  - 다중 작업 실행
- 사용법
  - [root@localhost ansible]# ansible-playbook [options] playbook.yml [playbook2 ...]
- 사용 옵션
  - URL : https://docs.ansible.com/ansible/latest/cli/ansible-playbook.html

5.1.7 ansible-pull

- 용도
  - VCS repo에서 playbook을 가져와 로컬 호스트에서 실행
- 사용법
  - [root@localhost ansible]# ansible-pull -U <repository> [options] [<playbook.yml>]
- 사용 옵션
  - URL : https://docs.ansible.com/ansible/latest/cli/ansible-pull.html

5.1.8 ansible-vault

- 용도

  - Ansible 데이터 파일 암호화/복호화

- 사용법

  - [root@localhost ansible]# ansible-vault [create|decrypt|edit|encrypt|encrypt_string|rekey|view] [options] [vaultfile.yml]

- 사용 옵션

  - URL : https://docs.ansible.com/ansible/latest/cli/ansible-vault.html

6. 활용예제

세부 목차

6.1.1 패스워드 없이 ssh 통신을 하기위한 ssh public key copy

[root@localhost ~]# ssh-copy-id -i ~/.ssh/id_rsa.pub user@management-host

6.1.2 프로젝트 디렉토리 생성

[root@localhost ~]# mkdir test-project

[root@localhost ~]# cd test-project

6.1.3 인벤토리 생성

[root@localhost test-project]# vi inventory

[web]

management-host1

[was]

management-host2

...

6.1.4 설정파일 생성

[root@localhost test-project]# vi ansible.cfg

```
[defaults]
inventory=./inventory
remote_user=user
private_key_file=~/.ssh/id_rsa

[privilege_escalation]
become=True
become_method=sudo
become_user=root
become_ask_pass=False
```

6.2.1 ad-hoc을 이용한 서버 ping 체크

[root@localhost test-project]# ansible all –m ping

전체 인벤토리 지정 all, ping 모듈 사용

management-host1 | SUCCESS => {

"changed": false,

"ping": "pong"

}

management-host2 | SUCCESS => {

"changed": false,

"ping": "pong"

}

…

6.2.2 데몬 설치 여부 확인 및 설치

[스크립트 작성]

[root@localhost test-project]# vi install.yml

```
---
## 앞서 생성한 인벤토리의 호스트 그룹명 [web] 지정
- hosts: web


  tasks:
  ## service facts 체크
  - name: service gather facts
    service_facts:

  ## 수집한 facts 정보로 services 내에 httpd가 있는지 확인 후 없다면 yum 모듈을 이용하여 최신 httpd 설치
  - name: httpd service install
    yum:
      name: httpd
      state: latest
    when: "'httpd' not in services"
```

[플레이북 실행]

[root@localhost test-project]# ansible-playbook install.yml

PLAY [web] **

TASK [service gather facts] **
ok: [management-host1]

TASK [httpd service install] ***
changed: [management-host1]

PLAY RECAP **
management-host1 : ok=2 changed=1 unreachable=0 failed=0 skipped=0

인벤토리 [web] 호스트 그룹에 대한 TASK 정상 적용 확인, changed의 경우 managed 호스트 서버에 변경 사항이 있을 경우 출력(httpd 설치)

ok : 변경사항 없음
changed : 변경사항 있음
unreachable : 연결 실패
failed : 실패
skipped : 플레이북 내 skip 롤이 있는 경우 출력

[실행결과 확인]

[root@managed-host ~]# rpm -qa | grep httpd

httpd-2.4.6-88.el7.x86_64

httpd-tools-2.4.6-88.el7.x86_64

[root@managed-host ~]# systemctl status httpd

● httpd.service - The Apache HTTP Server

Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)

Active: inactive (dead)

Docs: man:httpd(8)

man:apachectl(8)

6.2.3 서비스 기동 상태 확인 및 기동

[스크립트 작성]

[root@localhost test-project]# vi service_check.yml

```
---
## 앞서 생성한 인벤토리의 호스트 그룹명 [web] 지정
- hosts: web


  tasks:
  ## service facts 체크
  - name: service not started check
    command: systemctl status httpd
    register: result
    ignore_errors: yes

  ## 수집한 facts 정보로 httpd 서비스가 기동중이지 않다면 기동
  - name: httpd service start
    systemd:
      name: httpd
      state: started
    when: "result.rc != 0"
```

[플레이북 실행]

[root@localhost test-project]# ansible-playbook service_check.yml

```
PLAY [web] *********************************************************************************************************

TASK [service not started check] ***********************************************************************************
ok: [management-host1]

TASK [httpd service start] *****************************************************************************
changed: [management-host1]

PLAY RECAP *********************************************************************************************************
management-host1            : ok=2    changed=1    unreachable=0    failed=0    skipped=0
```

ok : 변경사항 없음
changed : 변경사항 있음
unreachable : 연결 실패
failed : 실패
skipped : 플레이북 내 skip 롤이 있는 경우 출력

[실행결과 확인]

```
[root@managed-host ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2018-11-23 17:25:41 KST; 1s ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 15482 (httpd)
   Status: "Processing requests..."
   CGroup: /system.slice/httpd.service
           ├─15482 /usr/sbin/httpd -DFOREGROUND
           ├─15483 /usr/sbin/httpd -DFOREGROUND
           ├─15484 /usr/sbin/httpd -DFOREGROUND
           ├─15485 /usr/sbin/httpd -DFOREGROUND
           ├─15486 /usr/sbin/httpd -DFOREGROUND
           └─15487 /usr/sbin/httpd -DFOREGROUND
```

6.2.4 환경설정 파일 배포

[httpd 설정 파일 배포]

[root@localhost test-project]# vi deploy.yml

```
---
## 앞서 생성한 인벤토리의 호스트 그룹명 [web] 지정
- hosts: web


  tasks:
  ## 설정 파일 배포, src: 배포파일, dest: 교체파일
  - name: deploy httpd.conf
    copy:
      src: /tmp/httpd.conf
      dest: /etc/httpd/conf/httpd.conf
      owner: root
      group: root
      mode: 0644


  ## httpd 재시작
 - name: httpd service restart
    systemd:
      name: httpd
      state: restart
```

[플레이북 실행]

[root@localhost test-project]# ansible-playbook deploy.yml

```
PLAY [web] *********************************************************************************************************

TASK [deploy httpd.conf] *******************************************************************************************
changed : [management-host1]

TASK [httpd service restart] ***************************************************************************************
changed: [management-host1]

PLAY RECAP *********************************************************************************************************
management-host1            : ok=1    changed=2    unreachable=0    failed=0    skipped=0
```

ok : 변경사항 없음
changed : 변경사항 있음
unreachable : 연결 실패
failed : 실패
skipped : 플레이북 내 skip 롤이 있는 경우 출력

[실행결과 확인]

[root@managed-host ~]# systemctl status httpd

● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active **(running)** since Fri 2018-11-23 17:25:41 KST; 1s ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 15482 (httpd)
   Status: "Processing requests..."
   CGroup: /system.slice/httpd.service
           ├─15482 /usr/sbin/httpd -DFOREGROUND
           ├─15483 /usr/sbin/httpd -DFOREGROUND
           ├─15484 /usr/sbin/httpd -DFOREGROUND
           ├─15485 /usr/sbin/httpd -DFOREGROUND
           ├─15486 /usr/sbin/httpd -DFOREGROUND
           └─15487 /usr/sbin/httpd -DFOREGROUND

7. FAQ

Q **Windows장비도 컨트롤이 가능한가요?**

A 가능합니다. WinRM을 이용하여 Linux장비의 SSH와 마찬가지로 통신하여 컨트롤 합니다. Windows 전용 모듈을 이용하면 보안 패치, 디스크 증설 등 많은 작업을 자동화 할 수 있습니다.

Q **Ansible은 어떤 환경을 지원하나요?**

A 베어메탈, Private Cloud, Public Cloud 등을 관리할 수 있는 모듈을 별도로 제공합니다. 대부분의 환경에서 사용 가능하며 인스턴스 생성, 컨트롤, Dynamic Inventory를 통한 호스트 노드들을 적게는 수백, 많게는 수천대를 컨트롤 할 수 있습니다.

8. 용어정리

용어	설 명
Inventory	관리 대상 서버 리스트
Modules	host에 특정 action을 수행하는 패키지화 된 스크립트
playbook	변수 및 task를 관리 호스트에 수행하기 위해 정의된 파일
YAML	playbook 작성에 필요한 언어
Plug-in	확장 기능(email, logging 등)
Custom module	사용자가 직접 작성한 모듈
REPL	대화형 환경(Read-Eval-Print Loop), CLI

Open Source Software Installation & Application Guide

nipa 공개SW역량프라자