

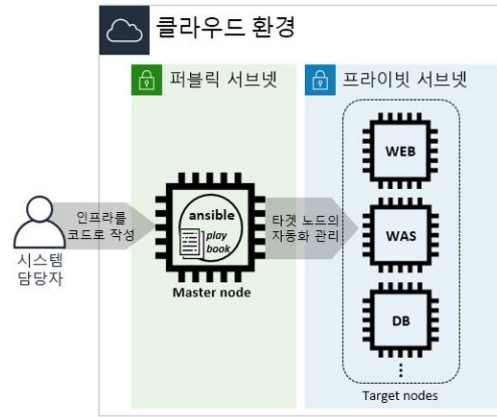


ANSIBLE

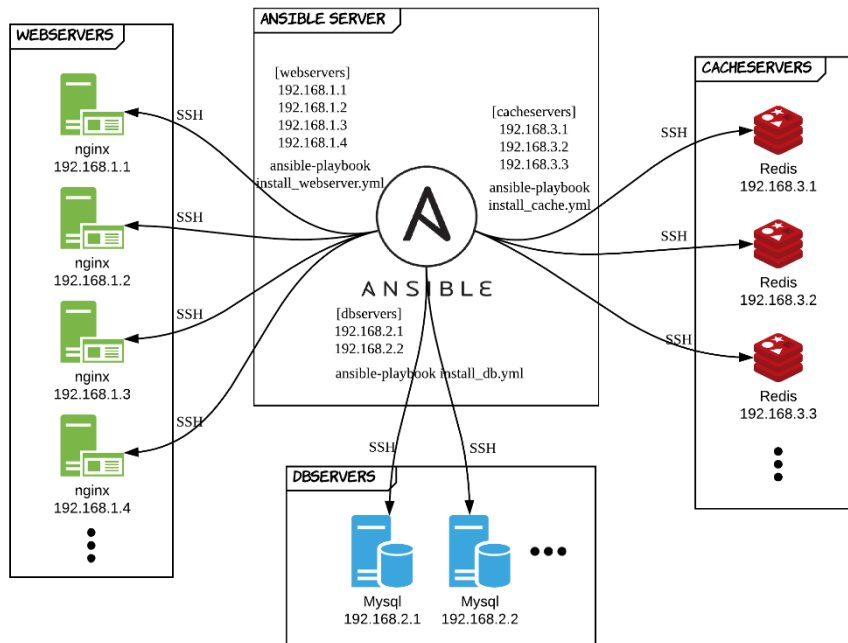
# *Ansible*을 이용한 시스템 구성관리

Mont.Kim

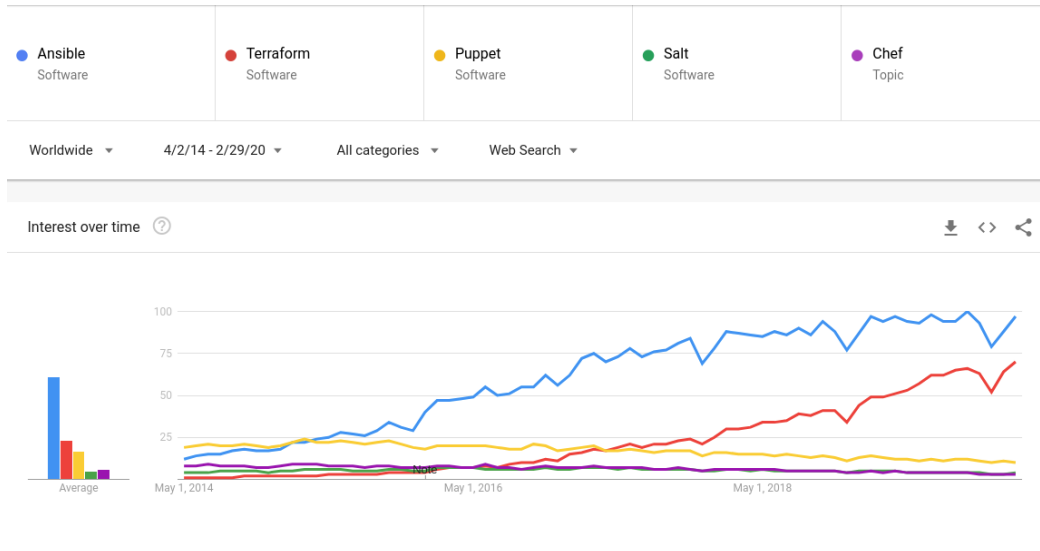
# Ansible



- Infrastructure As Code
- 단순성, 사용편의성
- Yaml기반 Playbook
- Client에 Agent 설치 불필요
- OpenSSH 기반 연결
- Host(Inventory)와 Role 기반 작업 할당



# Ansible



## Stack Overflow Trends

See how technologies have trended over time based on use of their tags since 2008, when Stack Overflow was founded. Enter up to 15 tags to compare growth and decline.

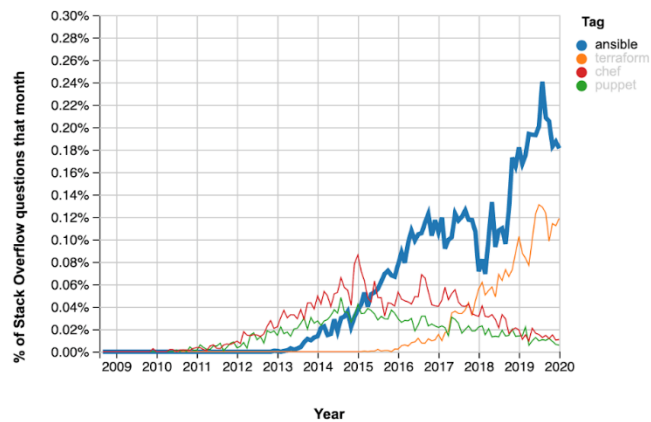
Tags:

ansible x terraform x puppet x chef x

Don't know what tags to look at? Try one of our presets:

- Most Popular Languages (TIOBE Index for May 2017)
- Operating Systems
- Mobile Operating Systems
- Javascript Frameworks
- Smaller Javascript Frameworks
- Closed-source Browser Plugins
- Data Science and Big Data
- Apache Open-source Projects

For more on this tool and what you can learn from it, see our [blog post](#).



제품	Puppet	Chef	Salt	Ansible
업체	Puppet Labs	Opscode	SaltStack	AnsibleWorks
출시	2005 년	2009 년	2011 년	2012 년
프로그래밍 언어	Ruby	Ruby/ Erlang (서버)	Python	Python
도입 사례	◎	◎	△	◎
정보량	○	◎	△	◎
확장성	◎	◎	◎	◎
공유 저장소	Puppet Forge	Opscode Community		Ansible Galaxy
Web UI	◎	◎	○	◎ (Ansible Tower)
정의 파일	자체 DSL (Ruby 기반)	YAML	자체 DSL (Python 기반)	YAML
에이전트 설치	필요	필요	필요	불필요
시스템 복잡성	△	△	△	◎

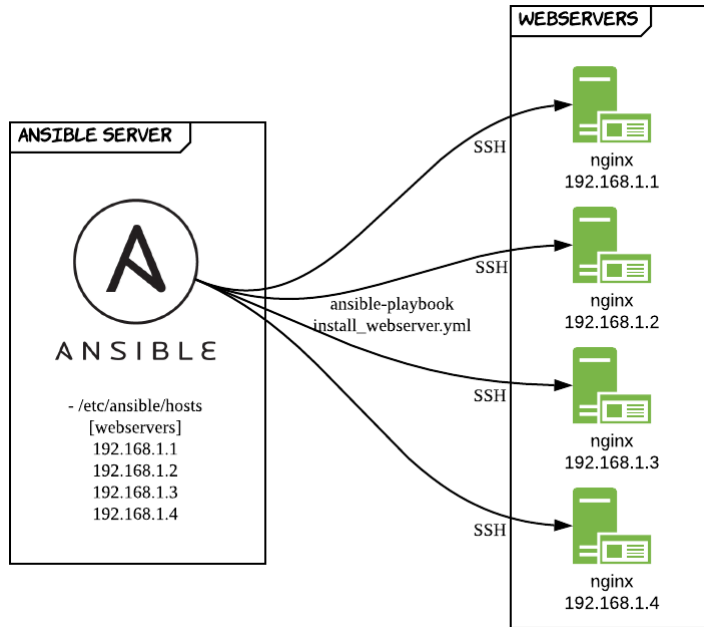
Github Star : 6.5K

6.8K

12K

52K

# Host & Role



```
block: |  
  [Management]  
  192.168.32.144  
  [K8S-Master]  
  192.168.32.150  
  [Worker]  
  192.168.32.151  
  192.168.32.152
```

Host를 기준으로 작업 할당

Role에 맞게 Task 진행

```
---  
- name: install nginx on the client  
  hosts: webservers  
  become: yes  
  
  tasks:  
    - name: install epel-release  
      action: "{{ ansible_pkg_mgr }} name=epel-release state=latest"  
    - name: install nginx web server  
      action: "{{ ansible_pkg_mgr }} name=nginx state=present"  
    - name: start nginx web server  
      service: name=nginx state=started
```



# Module

- Ansible 코드의 실행 단위
- Task에서 목적을 가지는 기능집합
- ping, shell, apt, service, user, copy 등 약 2,000개 모듈 존재

```
changed: false,
"ping": "pong"
}
root@ansible-server:/home/vagrant# ansible all -m shell -a free -k
SSH password:
192.168.32.152 | SUCCESS | rc=0 >>
      total      used         free       shared  buff/cache   available
Mem:    4039628    131648    3466088         7736     441892     3678104
Swap:   1003516         0     1003516

192.168.32.151 | SUCCESS | rc=0 >>
      total      used         free       shared  buff/cache   available
Mem:    4039628    131828    3301136         7680     606664     3675464
Swap:   1003516         0     1003516

192.168.32.150 | SUCCESS | rc=0 >>
      total      used         free       shared  buff/cache   available
Mem:    8168036    158012    7335480         8012     674544     7755368
Swap:   1003516         0     1003516
```

```
- name: Install CA-certificates, tzdata, perl
  apt:
    name:
      - ca-certificates
      - tzdata
      - perl
    state: present
```



# *Ansible With Vagrant*



ANSIBLE

+



VAGRANT

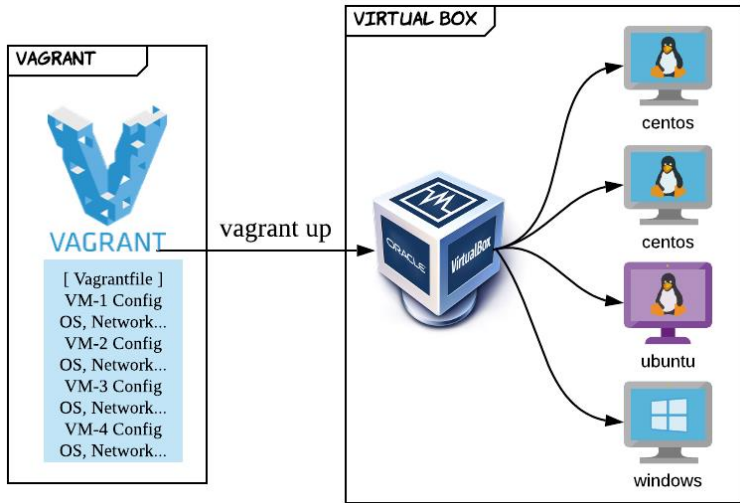
인프라 관리 도구

이미지 프로비저닝

Sandbox 환경에서의 개발



# Vagrant



## 가상 시스템 환경 관리 도구

### Boxes 기반 이미지 관리

Vagrant는 이미 만들어진 VM 이미지를 사용하여 복제합니다. 이 이미지를 Vagrant에서 Boxes라고 합니다. 사용 할 Boxes를 지정하여 Vagrantfile을 만드는것이 Vagrant 시작입니다.

### Provision 요구사항에 맞는 구축

Provisioning는 시스템에 소프트웨어를 자동으로 설치하고 구성을 변경하는 등 준비 작업을 의미합니다.

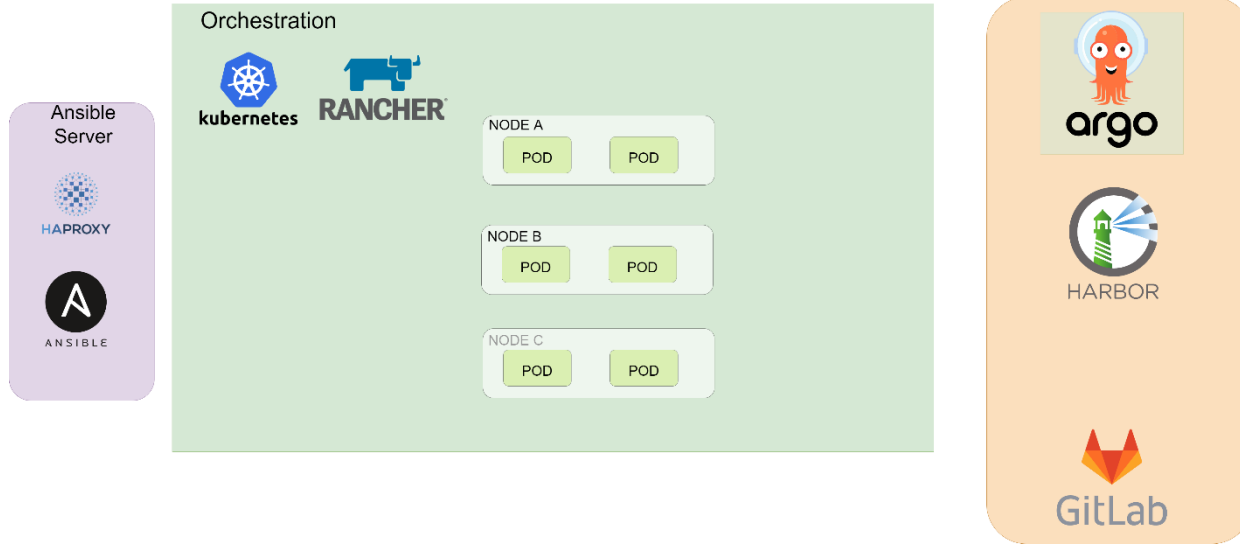
```
#=====#
# Ansible Server #
#=====#
# config.vm.define에 아래에 vm.box를 구분하기 위해 do |cfg| 를 추가함. end로 닫아야 함
config.vm.define "ansible-server" do |cfg|
  cfg.vm.box = "bento/ubuntu-18.04"

  # 가상머신 프로바이더 지정. 가상 머신의 이름 지정
  cfg.vm.provider "vmware_desktop" do |v|
    v.cpus = 2
    v.memory = 4096
  end

  # 포트가 중복적으로 사용될 경우 변경되도록 auto_correct를 true로 설정
  cfg.vm.host_name = "ansible-server"

  cfg.ssh.insert_key = false
  cfg.vm.network "public_network", ip: "192.168.32.143"
```

# Demo - Structure



노드	vCPU/RAM	IP	기타
Ansible Server	2 Core 4 Gi	192.168.31.143	HAPROXY
Management	4 Core 8 Gi	192.168.31.144	Harbor, Gitlab
K8S master	4 Core 8 Gi	192.168.31.150	Rancher, ArgoCD
worker1	2 Core 8 Gi	192.168.31.151	
worker2	2 Core 8 Gi	192.168.31.152	

playbook	역할	위치
main	RKE 설치	master
	RKE 클러스터링	worker
	Rancher 배포	master
	Harbor 설치	mgmt
	Haproxy 설치	ansible-server
argocd	ArgoCD 설치	master
gitlab	Gitlab 설치	mgmt



# Demo prerequisite

Install Provisioning Tool

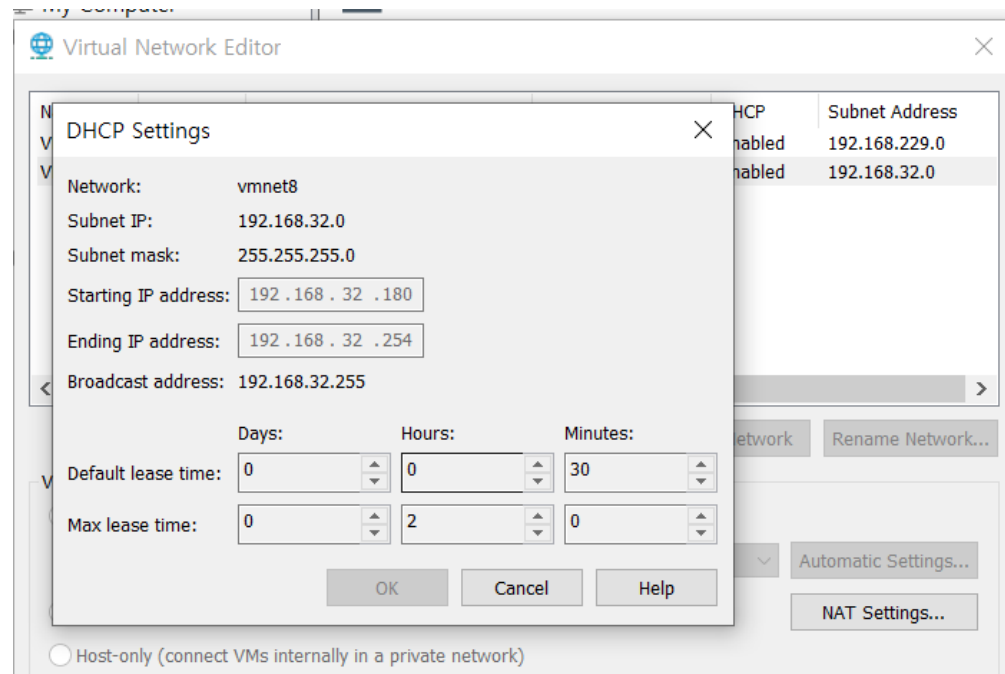
- VMWare(추천), VirtualBox

Provisioning Tool NetworkManager

- 192.168.32.XX 대역 사용

HOST

- Haproxy, harbor, gitlab domain 등록



```
# 38.25.63.10 x.acme.com # x client host
192.168.32.143 haproxy.osckorea.com devharbor.osckorea.com gitlab.osckorea.com

# localhost name resolution is handled within DNS itself.
# 127.0.0.1 localhost
# ::1 localhost
```

# Demo - Vagrant

## Install Vagrant

- <https://www.vagrantup.com/downloads>

## Install Vmware addon

- <https://www.vagrantup.com/docs/providers/vmware/vagrant-vmware-utility>

## Command on CMD

- vagrant plugin install vagrant-vmware-desktop
- vagrant version

## Provision(압축해제한 경로 Ansible/)

- vagrant up

## Access

- vagrant ssh ansible-server

```
PS C:\Users\Mont\Downloads\ansible> vagrant up
Bringing machine 'k8s-master' up with 'vmware_desktop' provider...
Bringing machine 'node1' up with 'vmware_desktop' provider...
Bringing machine 'node2' up with 'vmware_desktop' provider...
Bringing machine 'mgmt' up with 'vmware_desktop' provider...
Bringing machine 'ansible-server' up with 'vmware_desktop' provider...

PS C:\Users\Mont\Downloads\ansible> vagrant ssh ansible-server
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-163-generic x86_64)

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

System information as of Sun Feb 13 02:03:09 UTC 2022

System load: 0.13          Processes:            117
Usage of /:   2.8% of 61.80GB  Users logged in:     0
Memory usage: 4%          IP address for eth0: 192.168.32.154
Swap usage:  0%           IP address for eth1: 192.168.32.143

This system is built by the Bento project by Chef Software
More information can be found at https://github.com/chef/bento
vagrant@ansible-server:~$
```



## Demo - Ansible

Try ping (password : vagrant)

- `ans all -m ping -k`

Play main.yml on playbook folder

- `anp main.yml -k`

Play argocd playbook

- `anp argocd.yml -k`

Play gitlab playbook

- `anp gitlab.yml -k`

```
vagrant@ansible-server:~/playbook$ ans all -m ping -k
SSH password:
192.168.32.152 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
vagrant@ansible-server:~/playbook$ anp main.yml -k
SSH password:

PLAY [Install RKE on K8S Master] *****

vagrant@ansible-server:~/playbook$ anp argocd.yml -k
SSH password:

PLAY [Transfer and execute a script for nginx.] *****

TASK [Gathering Facts] *****
vagrant@ansible-server:~/playbook$ anp gitlab.yml -k
SSH password:

PLAY [Install Gitlab on Management] *****
```

