

Ansible의 이해와 활용

김용환

- 배경 (서버)

- 빠듯한 개발 일정

- 기획은 자주 바뀌고 개발자는 바뀐 기획을 따라가기 바쁨

- > QA는 기능이 개발되었는지 확인하고 싶어함

- > 개발자는 QA의 압박을 받음

- > QA가 개발자에게 개발이 완료되고 Alpha서버에 배포시 연락부탁

- > 배포 전 날 (D-1) QA에게 테스트를 부탁함

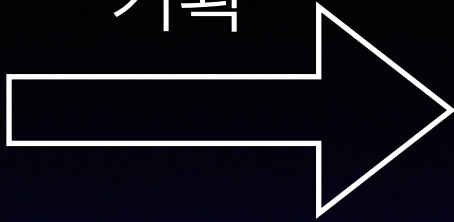
- > QA는 밤새 개발자와 함께 D-day를 맞이함

- QA Engineer는 json/rest, web 기반의 functional test를 SOAP UI, Jmeter 툴 또는 Code로 Local/Alpha에서 테스트 희망 (Local 테스트 환경만 있다면 Functional Test는 가능)

- Local 테스트 환경을 만들어주어 QA 엔지니어가 계속 자동화할 수 있도록 함

Project Starts

기획



개발

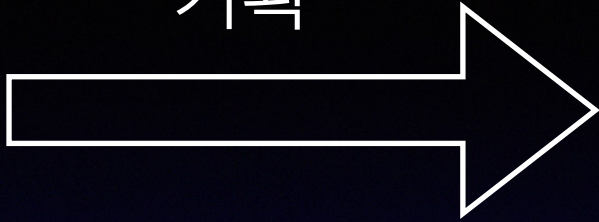


테스트

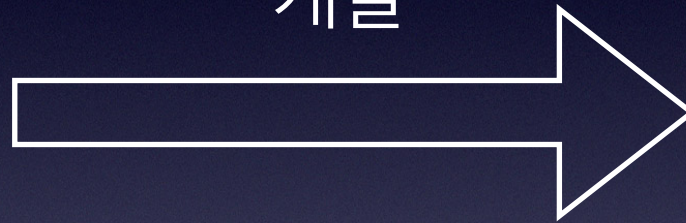


Real World

기획



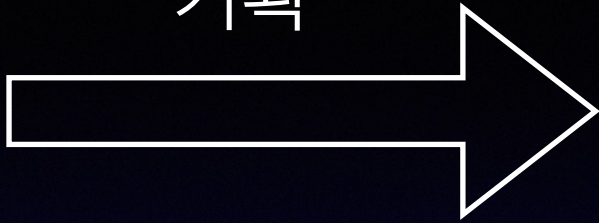
개발



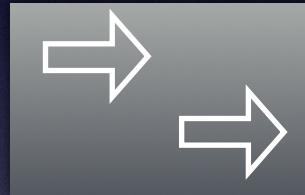
테스트

개발자와 QA간의 좋은 호흡

기획

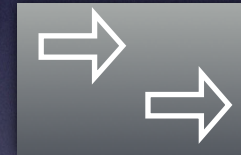


개발 & 테스트



phase#1

개발 & 테스트

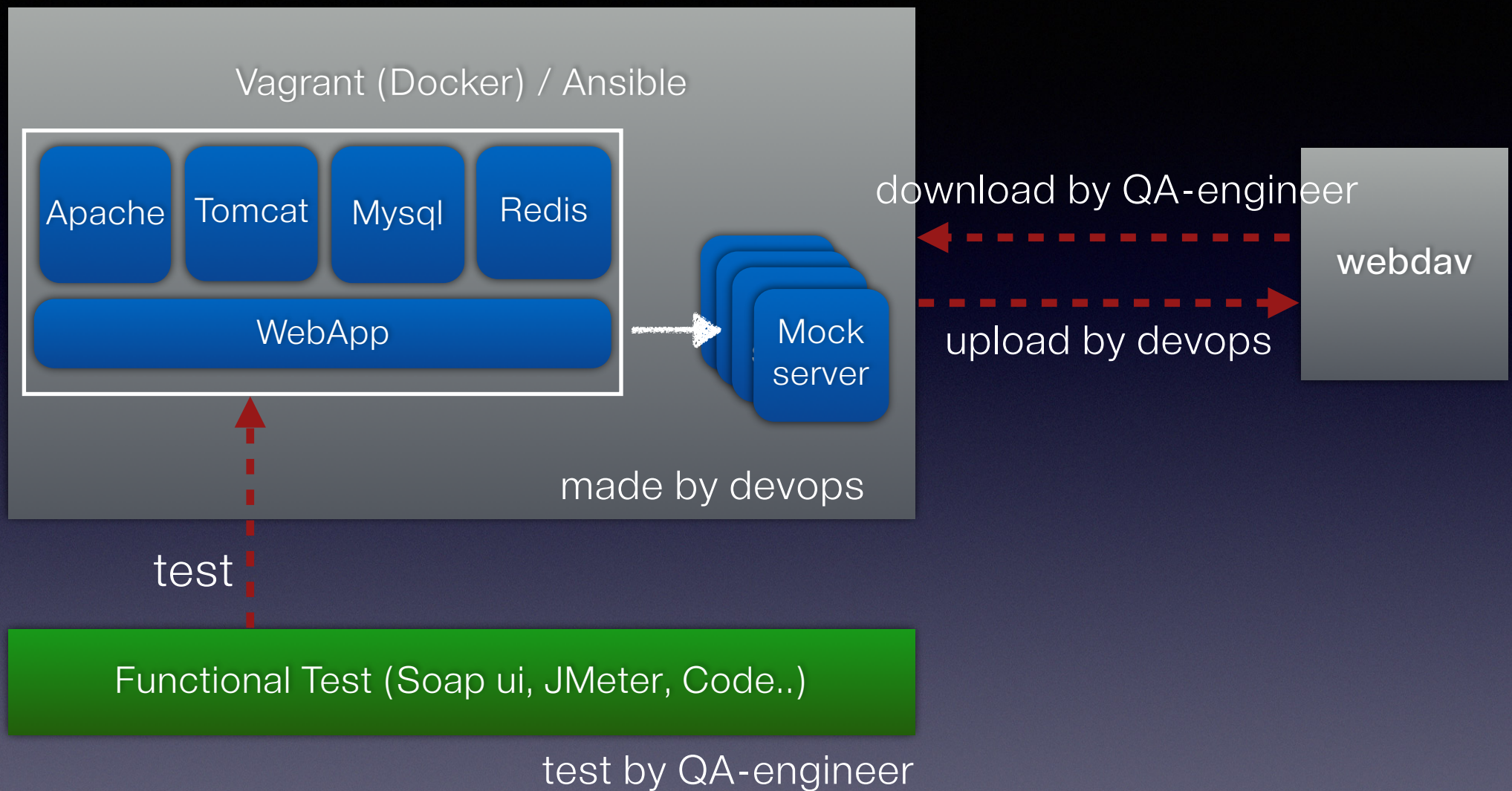


phase#2

개발 & 테스트



phase#3



- 아까 그림은 테스트 환경이 Image Box 기반 배포시스템과 비슷한데?? (docker, vagrant)
- 그러면, 가능해지는 것들?
 - 특정 Branch 소스 commit 될 때 마다 테스트 Image Box 생성할 수 있도록 해서 QA Engineer가 테스트할 수 있도록 구성
 - 게다가 master 소스 commit 될 때 Continuous Integration (jenkins) / Acceptance Test 진행 후 Cloud 용 Box가 생성, 배포
 - 매일 새벽마다 Regression Test를 진행하여 master 소스 및 모든 branch의 정합성 체크 가능

- Image Box 를 만들어주는 툴은 무엇으로 정할까?
 - 간단하면서 쉬운 툴인 Ansible로 사용하기
(DSL 기반의 provision툴인 chef, puppet,
capistrano 또는 python의 fab 대신 쉬운 툴로 선택)

- Ansible
 - 테스트 환경 구축 (개발자들의 다양한 OS)
 - 설치 / 배포 서버가 없는 경우
 - 서버 추가할 때 (ex. tomcat, jenkins)
 - 디렉토리 zip 또는 hard copy, soft copy
 - Ruby 기반(Chef, Puppet, Capistrano)의 provision tool을 잘 모르는 경우

Demo

<https://github.com/knight1128/ansible-simple>

- ansible 실행 명령어

- ansible

```
$ ansible
```

```
Usage: ansible <host-pattern> [options]
```

- ansible-playbook

```
$ ansible-playbook
```

```
Usage: ansible-playbook playbook.yml
```


Ansible 이란?

- Provision & configuration management tool
- Python Github project 중 상위 랭킹 (6위)

The screenshot shows the GitHub repository for Ansible. At the top, it says 'ansible / ansible' with 579 watches, 7,550 stars, and 2,325 forks. Below this, a description states: 'Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications— automate in a language that approaches plain English, using SSH, with no agents to install on remote systems. <http://ansible.com/>'. Statistics show 12,117 commits, 20 branches, 50 releases, and 872 contributors. A merge pull request #8988 from bbasleeper/bugfix_yum_module is highlighted. A list of recent commits follows, including 'Add error checking to ansible-galaxy to verify the roles_path is valid' by mpdehaan 4 days ago, 'Update test strategies guide.' 4 months ago, 'Updating PR testing document with manual branch checkout steps' 2 days ago, 'Add documentation about bin_ansible_callbacks setting. Standard...' 3 days ago, 'Modules are almost always written now by mutiple authors, so do...' 3 days ago, 'Various tests using datafiles are being moved into the integration t...' 7 months ago, 'Merge pull request #8853 from kvar/devel' a day ago, 'Fix a typo on is_update call inside latest function' 14 hours ago, 'Make debuild happy by including a date in debian/changelog' a month ago, 'Tweaking logical assignment of vm_metadata in abiquo inventory ...' 11 days ago, 'Fix quoting issues in lineinfile for individual single-quotes' a day ago, 'Updated pr_duplicate ticket stub' 19 days ago, and 'Updated DEB build workflow' 3 months ago.

<https://github.com/ansible/ansible>

The screenshot shows a list titled 'Top 400 Python Projects in Github'. The list includes the following projects:

1. **django** (git) - The Web framework for perfectionists with deadlines. 10531 stars. Love it
2. **requests** (git) - Python HTTP Requests for Humans™. 10368 stars. Love it
3. **httpie** (git) - Command line HTTP client, human-friendly cURL replacement. 10281 stars. Love it
4. **flask** (git) - A microframework based on Werkzeug, Jinja2 and good intentions. 10273 stars. Love it
5. **tornado** (git) - Tornado is a Python web framework and asynchronous networking library, originally developed at FriendFeed. 7489 stars. Love it
6. **ansible** (git) - Ansible is a radically simple IT orchestration engine that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications— automate in a language that approaches plain English, using SSH, with no agents to install on remote systems. 6843 stars. Love it

<http://pythonhackers.com/open-source/>

- Enterprise 버전은 따로 존재. ansible 오픈 소스에 UI 와 일부 utility 추가한 상용 버전
<http://www.ansible.com/home>

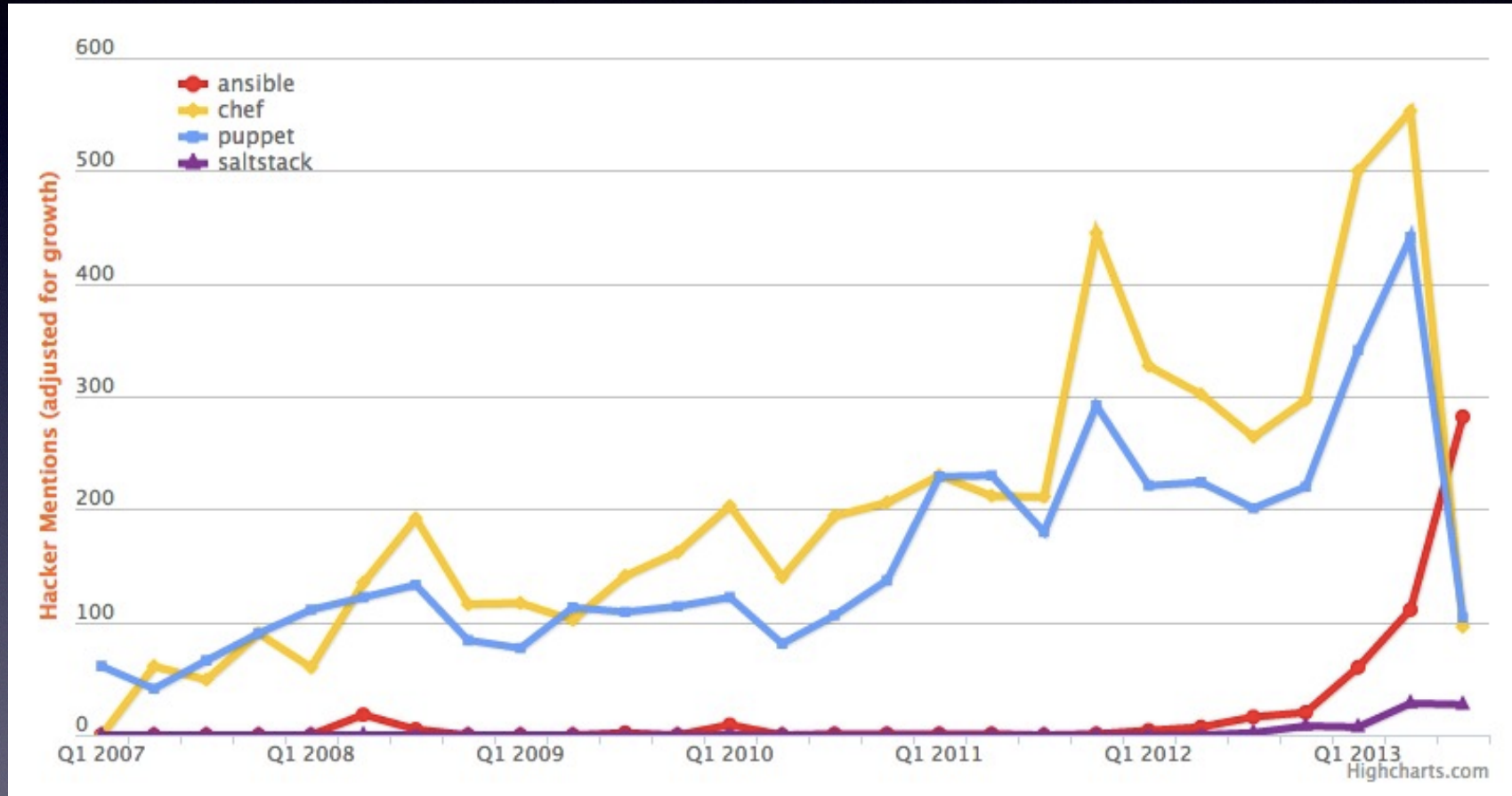
- 오픈 소스 버전 (GPL)
<https://github.com/ansible/ansible/>

- 레퍼런스



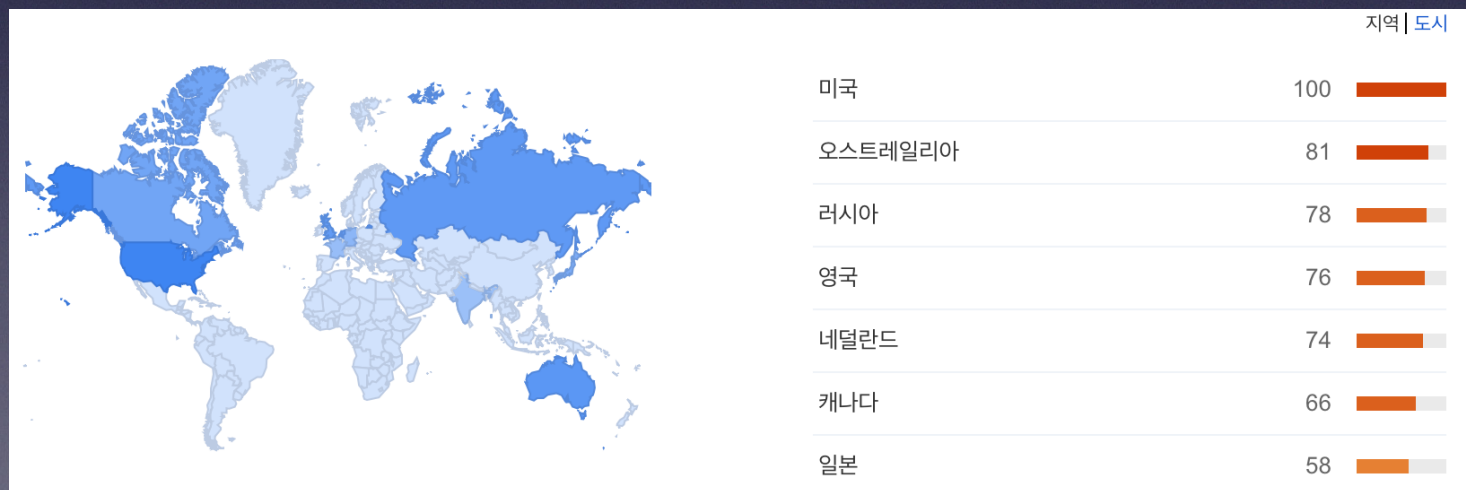
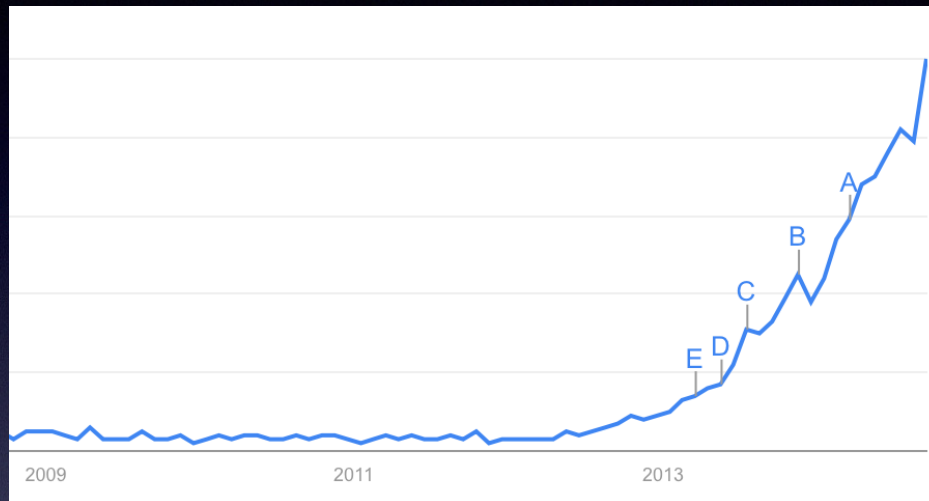
- ebay's ci solution with Apache Messos 에 ansible 적용
 - <http://www.ebaytechblog.com/2014/04/04/delivering-ebays-ci-solution-with-apache-mesos-part-i>
- open stack쪽과 긴밀함
 - <https://github.com/openstack-ansible/openstack-ansible>

- 해커 뉴스(Hacker News) 분석을 보면, Ansible이 많이 Mention되고 있음



<http://redmonk.com/sograzy/2013/12/06/configuration-management-2013/>

- 구글 트렌드에서 Ansible 검색결과, Ansible 에 대한 많은 관심도를 가지고 있음



<https://www.google.co.kr/trends/explore?q=ansible>

ansible	개발언어	정의	Agent/SSH여부	통신방법	github 활동성
puppet	ruby	DSL	Agent	http ssl	중
chef	ruby	DSL	Agent	Rest / STOMP	중
<u>ansible</u>	python	YAML	SSH	json	상

- Ad-hoc 지원
- Simple
- SSH (非 agent)
- Not DSL (非 Ruby 개발자)
- 병렬 provisioning 지원
- plug-and-play(module, plugin 사용 및 개발 가능)

- 멱등성 (Idempotency)
 - 여러 번 적용해도 결과는 바뀌지 않는다.
 - 바뀌는 것이 없으면 당연히 배포되어도 바뀌지 않는다.
 - 바뀌는 부분이 있으면 그 부분만 반영된다.
- Ansible
 - 대부분이 멱등성을 제공한다.
 - 멱등성을 제공하지 부분(모듈)
 - shell, command, file module

Ansible 이해

- 할 수 있는 일
 - 설치 : apt-get, yum, homebrew...
 - 환경 설정 파일 및 스크립트 배포 : copy, template...
 - 다운로드 : get_url, git, subversion...
 - 실행 : shell, task...
- 결과
 - ok, failed, changed, unreachable

- 설치

- SSH 기반이라 Server에 Ansible을 설치하면 완료

- brew install ansible

- pip install ansible

- apt-get / yum

- But, Remote Host(managed node)에서 python 2.4 이하의 버전을 사용시에는 simplejson 모듈을 설치해야 한다.

- pip install simplejson

- YAML (<http://yaml.org>)
 - ‘야믈’이라 발음
 - 사람이 쉽게 읽을 수 있는' 데이터 직렬화 양식
 - yaml이라는 이름은 ‘YAML ain't Markup Language’ (YAML은 마크업 언어가 아니다.) 라는 재귀적인 이름에서 유래되었으나, ‘Yet Another Markup Language’ (또 다른 마크업 언어) 이 현재 공식적인 약자
 - yaml 은 마크업보다는 데이터 중심, 즉 serialization format 이라는 의미를 강조하고 있다.
 - ruby, python 개발자는 property 파일로 활용하고 있으나, c,c++,java개발자에게는 자주 쓰이지 않아 모를 수 있음. 참고로 Spring 에서도 지원

```
'[  
  "apple",  
  {  
    "bar": ["baz", "kwa", 3.0, 5]  
  }  
'
```

json



```
- apple  
- bar:  
  - baz  
  - kwa  
  - 3.0  
  - 5
```

YAML



- Jinja2

- <http://jinja.pocoo.org/docs/>

- python template 언어

- 하드 코딩 없이 변수 명으로 사용 가능

```
{{ server_name }}
```

- 변수가 정의되었는지 확인(verification)

```
{% if utf8 %} .. {% endif %}  
{% if lb.master is not defined %}  
..  
{% endif %}
```




- more with jinja2

- default

```
vol="{{ item.data_volume | default('False') }}"
```

- simple loop

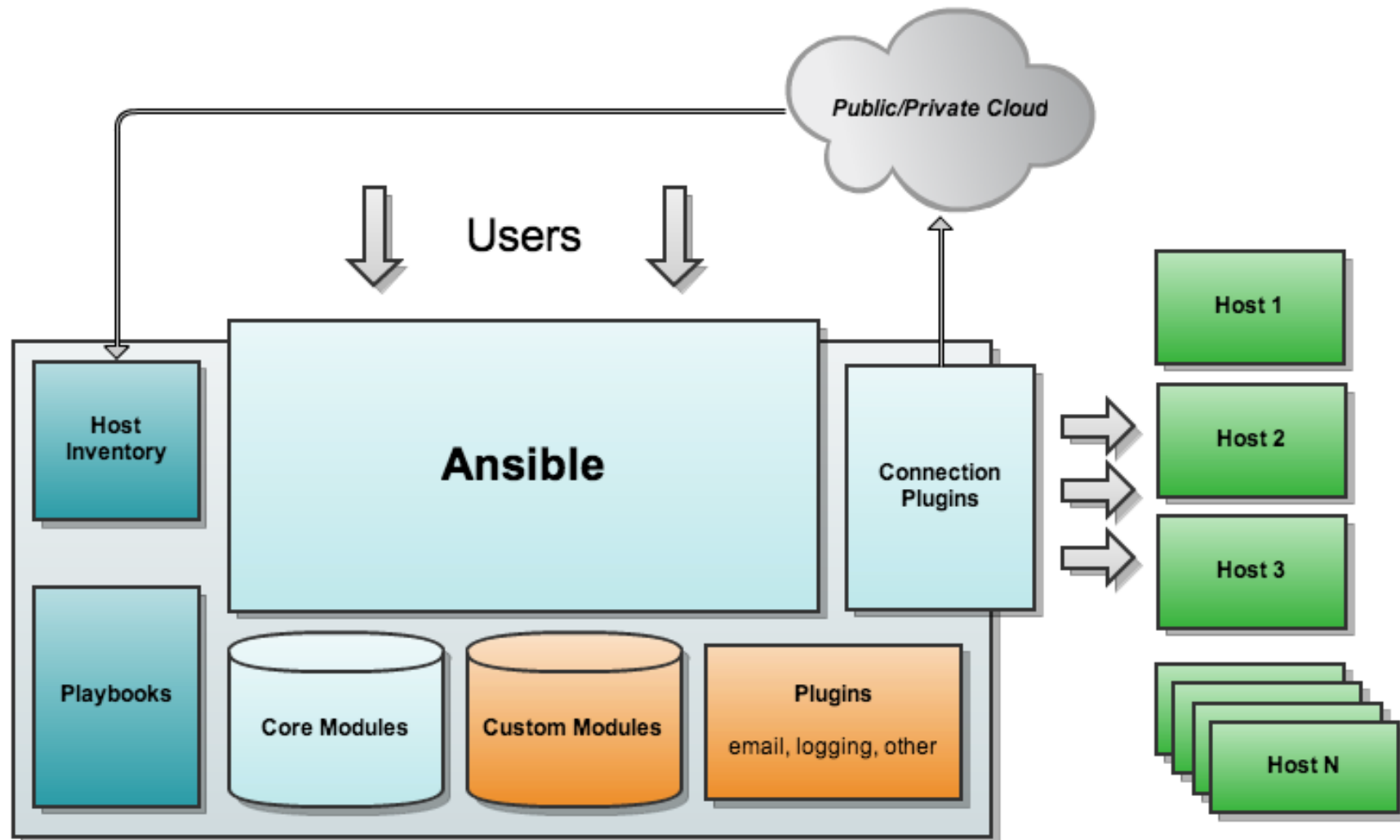
```
- name: add users
  user: name={{ item }} state=present groups=www
  with_items:
    - testuser1
    - testuser2
```

- for loop

```
{% for host in groups['db_servers'] %}
  {{ host }}
{% endfor %}
```

- combining

```
{{ users | join(', ', attribute='username') }}
```

<http://terry.im/wiki/terry/Ansible.html>

- inventory file
 - inventory 파일은 리모트 서버에 대한 meta 데이터를 기술 하는 파일
 - 기본 파일은 /etc/ansible/hosts이나, 따로 inventory 파일을 쓸 수 있는 옵션(-i)을 줄 수 있음
 - remote host를 Grouping 할 수 있음

- more with inventory file

- /etc/ansible/hosts 또는 임의의 파일 (hosts)에 정의

- adhoc 지원

```
-m <module_name>  
-a <module_parameter>
```

- (만약 /etc/ansible/hosts 존재)

```
$ ansible production -m ping
```

- (만약 따로 만든 hosts 파일 사용시)

```
$ ansible beta -i hosts -m command -a '/sbin/reboot'
```

- yum, apt-get 설치 가능

```
$ ansible production -m yum -a "name=acme state=installed"
```

```
[alpha]  
192.168.1.1  
[beta]  
192.168.1.2  
[production]  
192.168.1.3  
192.168.1.4
```


- template

- YAML 파일 뿐 아니라 모든 파일에서 활용 가능
- 관례상 파일 확장자명을 .j2로 함
index.php.j2
mysql.cnf.j2
- template task 일때 jinja2가 적용 가능(copy task
는 적용안됨)

```
[client]
default-character-set = utf8
```



```
[client]
{% if utf8 %}
# apply utf8
default-character-set = utf8
{% endif %}
```


- playbook
 - playbook은 ansible의 환경 설정, 배포를 가능케 함
 - yaml 문법을 채용하여 정책을 기술
 - linux 기반 권한 관리 (user, group) 지원
 - conditional (when), variables, loops(with_items), include, 결과 저장(register) 지원
 - 하나의 playbook은 하나 또는 하나 이상의 'play'를 두며, play의 목적은 여러 호스트들에 잘 정의된 'role'과 'task'를 매핑하는 역할을 맡음

- host - 적용할 호스트의 이름 또는 그룹명 (필수)

```
hosts: all
```

```
hosts: hostname
```

```
hosts: groupname
```

```
hosts: groupname1, groupname2
```

```
hosts: groupname1, hostname2
```

```
hosts: commerce*.apache.org
```

- sudo - 디폴트는 false

```
sudo = true
```

- user - 디폴트는 root

```
user: remoteuser
```


- vars 선언

```
group_name: rabbit
web:
  play: 192.168.1.1
  tomcat: 192.168.1.
config: ssl.conf
config_path: /apache/$config
```

tasks 에서 사용



```
${group_name}
${web.play}, ${web.tomcat}
${config}
${config_path}
```


- more with var

```
is_ubuntu: '${ansible_distribution}' == 'ubuntu'  
is_debian: '${ansible_distribution}' == 'debian'
```

- vars files - 변수 파일 지정

```
vars_files:  
  - /vars/vars_file.yml  
  - /vars/${hostname}.yml
```


- task - ansible module을 호출하는 단위 (필수)

- task 종류

- 간단 task : name / action

- name : check server's aliveness.
action : ping

- Ansible 모듈 이용 task

- copy: src=/srv/myfiles/foo.conf dest=/etc/foo.conf owner=foo group=foo mode="u=rw,g=r,o=r"

- more with task

- item을 이용한 복수 작업

- name: add several users

- user: name={{ item }} state=present

- groups=www

- with_items:

- testuser1

- testuser2

- more with task

- include - 변수 값을 지정해서 include yaml 파일로 넘길 수 있음

- **include:** tasks/add_user.yml user=www group=www

- when (conditional)

- apt: name=\$item state=latest

- with_items:

- ntp

- when:** ansible_distribution == 'Ubuntu'

- yum: name=\$item state=lates

- with_items:

- ntp

- when:** ansible_distribution == 'CentOS'

- more with task

- event 발생 (task -> handler)

- task

- name: Copy Hello World PHP script
template: src=index.php.j2 dest=/var/www/
index.php mode=0664
notify: Restart Apache

- handler

- handlers:
 - name: Restart Apache
service: name=apache2 state=restarted

- more with handler

- include

- `include: handlers/main.yml`

- role : structure 기본 단위로서 설치, 사용이 가능
 - apache
 - redis
 - java
 - jenkins
 - tomcat
- (예, vars, tasks 디렉토리가 존재하면 role/vars/main.yml, role/tasks/main.yml 을 가장 먼저 찾음)

- remote host(managed node) 연결없이 local 동작하게 할 수 있음

- ansible 명령

```
ansible-playbook playbook.yml --connection=local
```

- yaml 파일

```
- hosts: 127.0.0.1  
  connection: local
```

- 이외 다양하고 쓸만한 module이 굉장히 많음

Demo Apache+PHP

<https://github.com/knight1128/ansible-apache-php>

- Vagrant + Example Playbook
 - Vagrantfile

```
# -*- mode: ruby -*-  
# vi: set ft=ruby :  
  
# Vagrantfile API/syntax version. Don't touch unless you know what you're doing!  
VAGRANTFILE_API_VERSION = "2"  
  
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|  
  config.vm.box = "precise64"  
end
```


playbook

```
- name: Install Apache and PHP
hosts: all
sudo : yes
vars:
  server_name: Apatch
```

play

tasks:

- name: 1. Install Apache
apt: name=apache2 state=present
- name: 2. Install PHP5
apt: name=php5 state=present
- name: 3. Install Apache PHP5 module
apt: name=libapache2-mod-php5 state=present
- name: 4. Copy Hello World PHP script
template: src=index.php.j2 dest=/var/www/index.php mode=0664
notify: Restart Apache

handlers:

- name: Restart Apache
service: name=apache2 state=restarted

task & handler

index.php.j2

```
<html>
  <head>
    <title>Ansible Test App</title>
  </head>
  <body>
    <br>
    Server name is {{ server_name }}.
    <a href=http://{{ ansible_eth0.ipv4.address }}/index.html>Homepage</a>
    <a href=http://{{ ansible_eth0["ipv4"]["address"] }}/index.html>Homepage</a>
    </br>
  <?php
    Print "Hello, World! I am configured Ansible Test App: ";
    echo exec('hostname');
  ?>
</body>
</html>
```



```
$ ansible-playbook -i hosts playbook.yml
```

```
PLAY [Install Apache] *****
```

```
GATHERING FACTS *****
```

```
ok: [127.0.0.1]
```

```
TASK: [1. Install Apache] *****
```

```
ok: [127.0.0.1]
```

```
TASK: [2. Install PHP5] *****
```

```
ok: [127.0.0.1]
```

```
TASK: [3. Install Apache PHP5 module] *****
```

```
ok: [127.0.0.1]
```

```
TASK: [4. Copy Hello World PHP script] *****
```

```
changed: [127.0.0.1]
```

```
NOTIFIED: [Restart Apache] *****
```

```
changed: [127.0.0.1]
```

```
PLAY RECAP *****
```

```
127.0.0.1 : ok=6 changed=2 unreachable=0 failed=0
```


DEMO

(Jenkins+Plugins)

<https://github.com/knight1128/ansible-jenkins>


```
$ cat playbooks/setup.yml
```

```
---  
- hosts: all  
  sudo: True  
  roles:  
    - common  
    - jdk  
    - jenkins
```



```
|— LICENCE
|— README.md
|— Vagrantfile
|— hosts
|— playbooks
    |— common
        |   |— tasks
        |   |   |— main.yml
        |   |   └─ packages.yml
        |   └─ vars
        |       └─ main.yml
    |— jdk
        |   └─ tasks
        |       └─ main.yml
    |— jenkins
        |   |— files
        |   |   └─ jenkins-ci.org.key
        |   |— handlers
        |   |   └─ main.yml
        |   |— main.yml
        |   |— tasks
        |   |   |— jenkins.yml
        |   |   |— main.yml
        |   |   |— plugins.yml
        |   |   └─ repo.yml
        |   |— templates
        |   |   └─ hudson.tasks.Mailer.xml.j2
        |   └─ vars
        |       └─ main.yml
    |— setup.yml
    └─ vars
        └─ main.yml
```


사용 소감

- Ansible은 계속 진화 중...
 - 하위 버전에 쓰던 모듈을 사용 못할 때 있음 (only_if..)
- Python 진영의 밀어주는 속도가 좋고 플러그인과 모듈이 많이 생기고 있음
- 단계가 많고 복잡한 Webapp provision의 경우는 안어울 수도 있음. 그러나 Continuous Integration 성이나 테스트 환경에 어울릴 듯 함

- playbook 작성시 팁
 - `${role}` 단위로 해야 편함 (main.yml을 자동으로 검색하고 없으면 에러 발생)
 - `${role}/tasks/main.yml`
 - role 단위로 정의할 때는 복수형으로 작성
 - 복수형 - `${role}/tasks`
예) common/tasks, jdk/tasks
 - 최대한 멍등성이 지원되게 (Continuous Delivery!!)
 - register/when를 잘 활용하여 잘 활용하면 이득
 - 하나의 yml 파일에 넣지 말고 역할에 맞게 나누어 Simple하게
 - role과 include를 잘 나누어 사용
 - 디버그는 syntax-check, verbose, register, debug를 활용

- Test support (검증 수단)
 - stat, action, shell module 활용하여 fail, assert 할 수 있다.
- 클라우드 환경 지원
 - amazon, open stack, google cloud

- 아쉬운 점

- multiline comment 지원 안 함
- 실시간 provision vs 클라우드 형태의 provision
- interactive shell 을 core에서 지원하지 않음
 - <https://github.com/dominis/ansible-shell>
모듈에서 지원

Good Practice Reference

- <http://docs.ansible.com/>
- <https://www.reinteractive.net/posts/167-ansible-real-life-good-practices>
- <http://www.slideshare.net/SarahZelechowski/ansible-how-to-get-more-sleep-and-require-less-coffee>

감사합니다.

Q & A