
opennaru
Ansible 소개
(IT Automation & Configuration Management)
KHAN
apm

IT Evolution

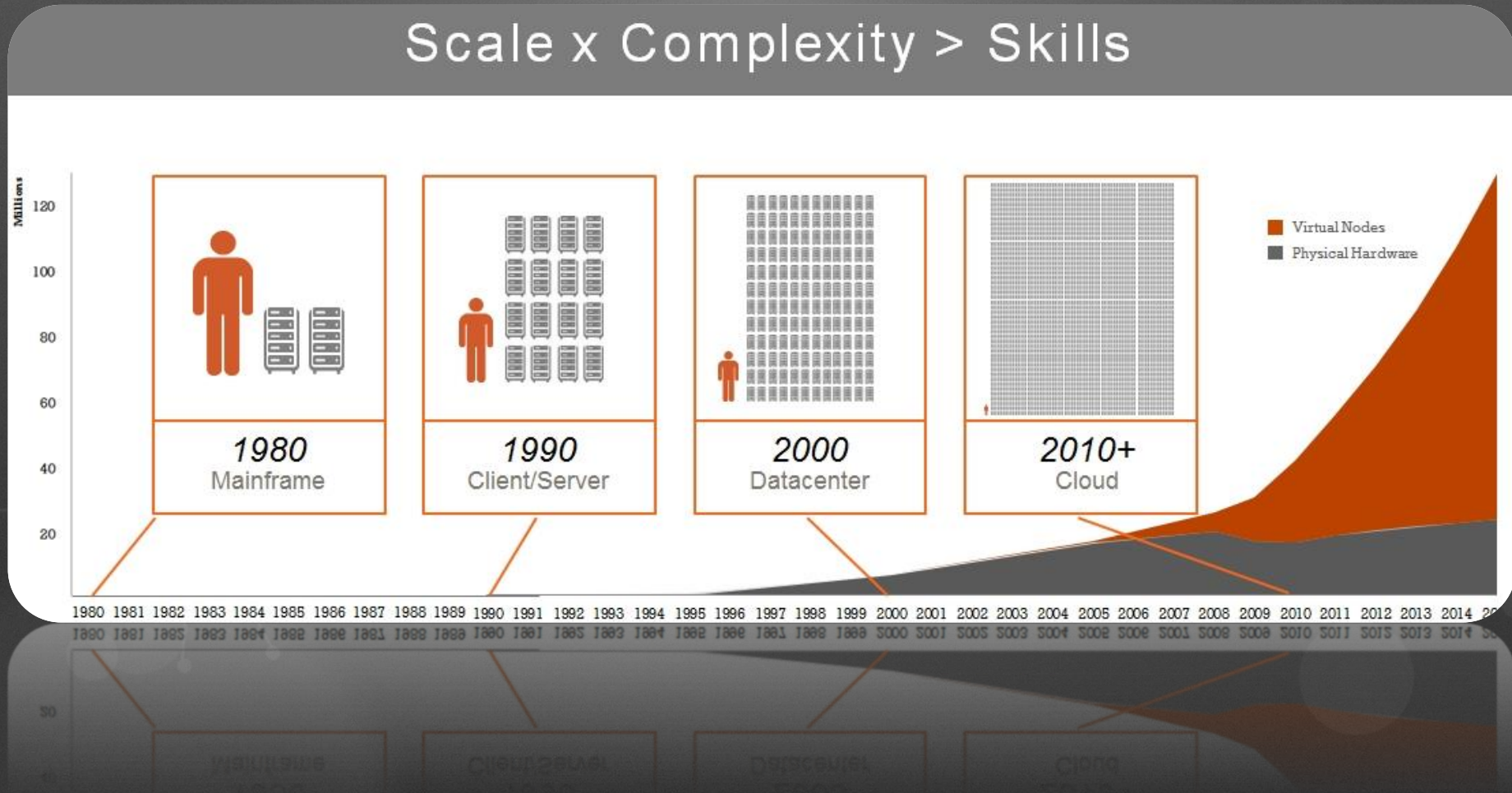
opennaru

KHAN [apm]

Development Process	Application Architecture	Deployment & Packaging	Application Infrastructure
WATERFALL	MONOLITHIC	PHYSICAL SERVERS	DATA CENTER
AGILE	N-TIER	VIRTUAL SERVERS	HOSTED
DEVOPS	MICROSERVICES	CONTAINERS	CLOUD

Increasing scale and complexity means we need admin automation

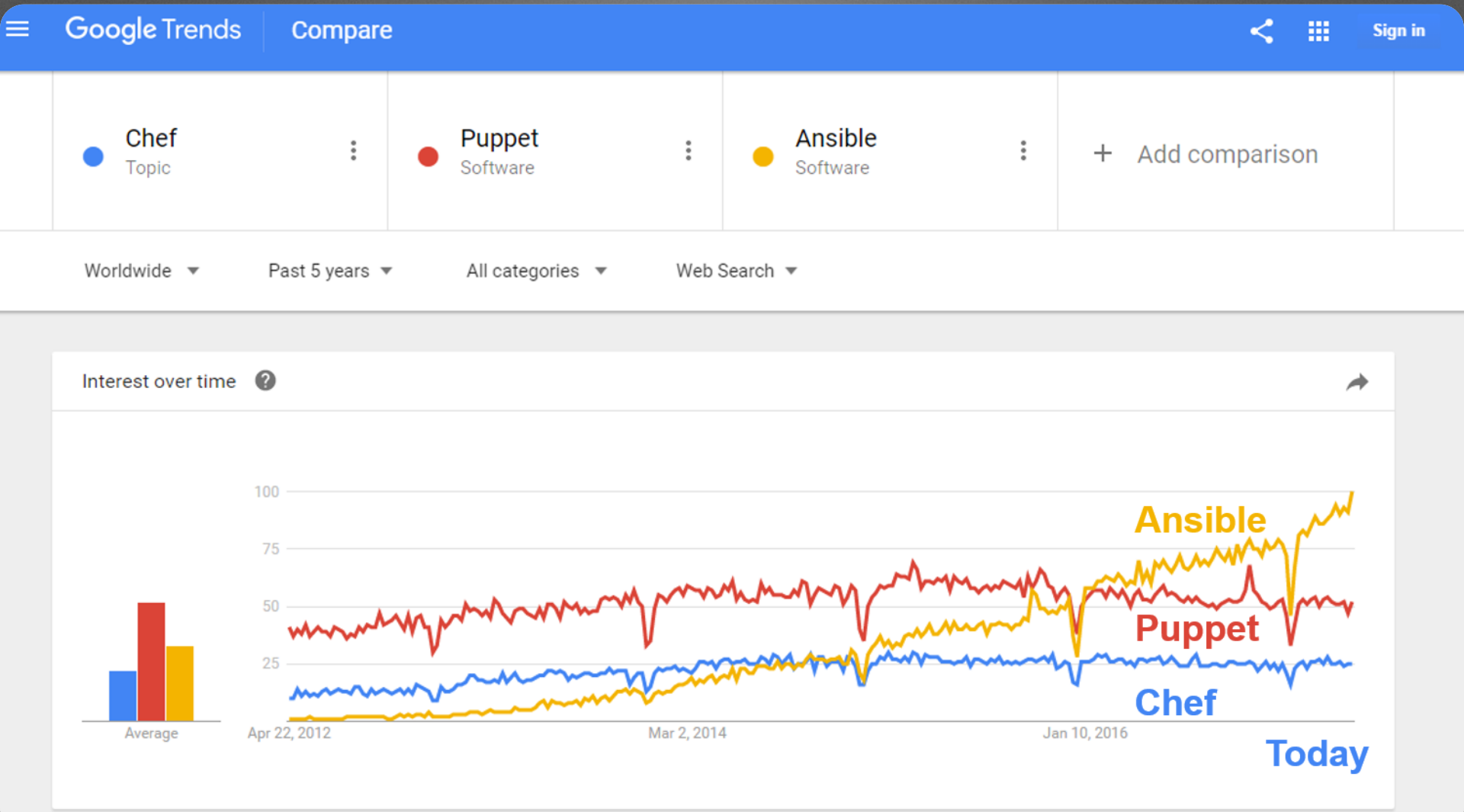
opennaru



Opscode gets more venture dough for its Chef
From - http://goo.gl/dLcjS

Global Google Trends : Ansible vs. Puppet vs. Chef

opennaru



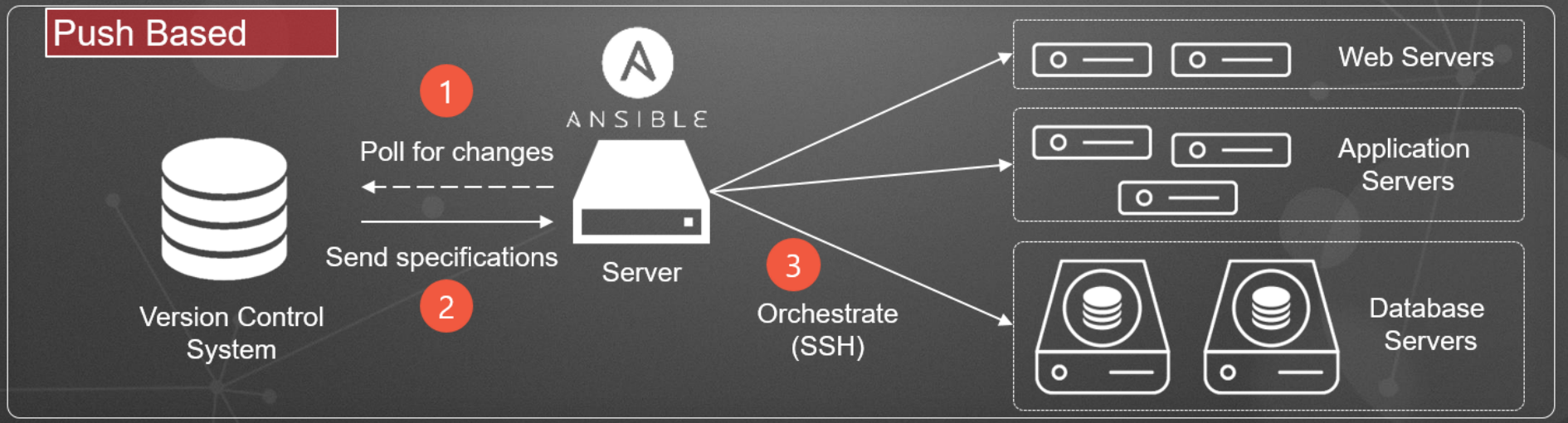
opennaru

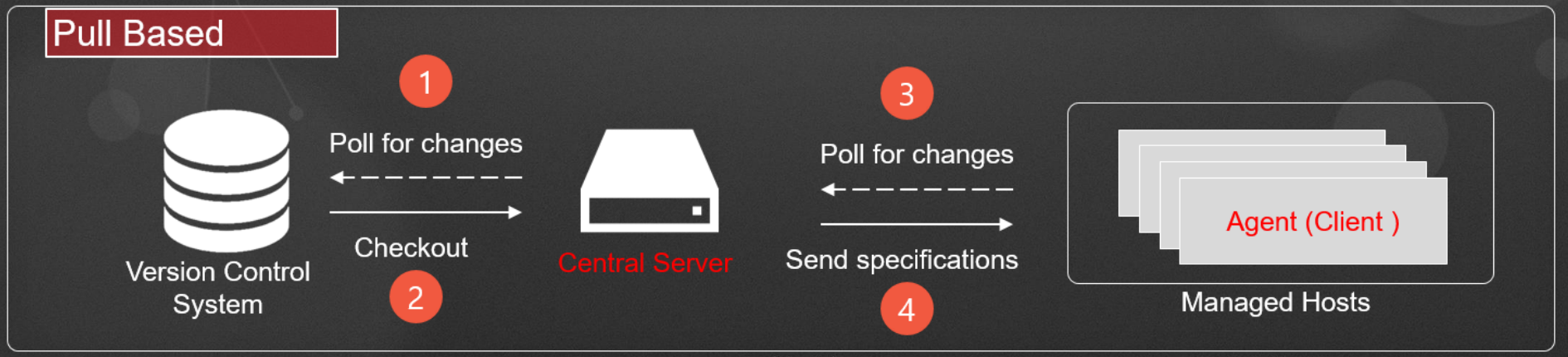
Red Hat acquired Ansible at year end 2015

Push Based vs. Pull Based

opennaru

- Puppet과 Chef는 Pull based Tool
  - 대상 서버에 있는 Agent들은 주기적은 중앙 서버에 구성정보를 확인
- Ansible은 Push based Tool





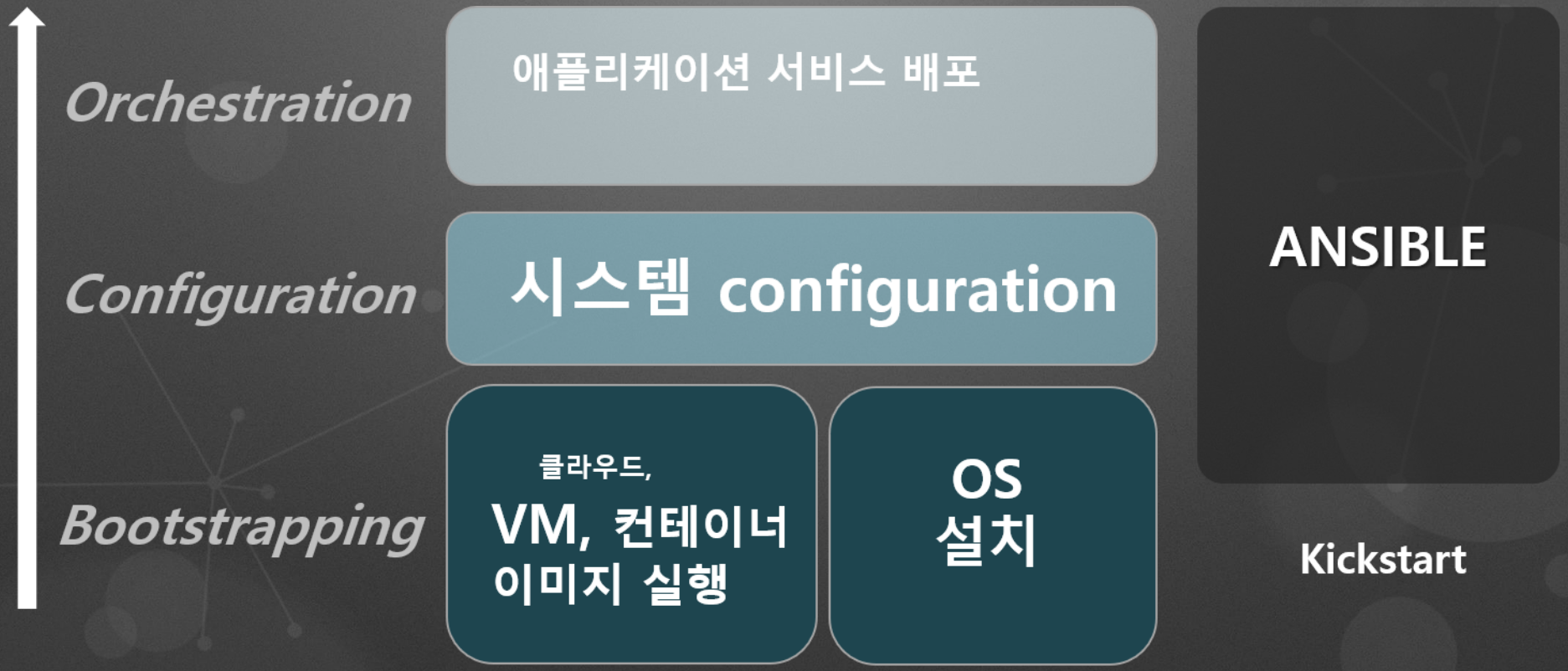
Infrastructure as Code 영역

opennaru

- Bootstrapping
  - 플랫폼 자체 또는 서버로서 OS가 이용 가능한 상태를 만드는 도구
  - 가상머신이나 컨테이너, 클라우드에 대해 API를 이용해 자원 할당
- Configuration
  - 서버에 미들웨어 설치나 각종 설정
  - 서버 구성 관리의 자동화에 해당되는 부분
- Orchestration
  - 자원과 관련된 서비스 제공
  - 구축 끝난 환경에서 필요한 작업

Provisioning Activity

opennaru



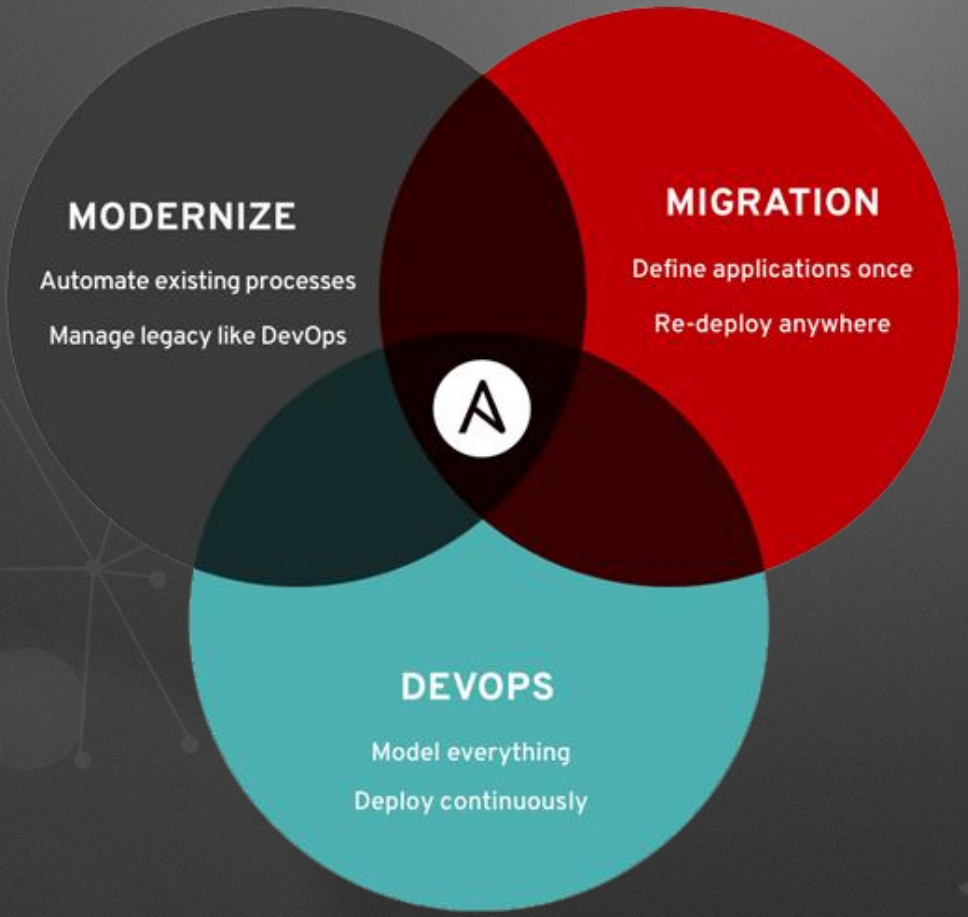
Source : Provisioning Toolchain (Velocity2010) by Lee Thompson

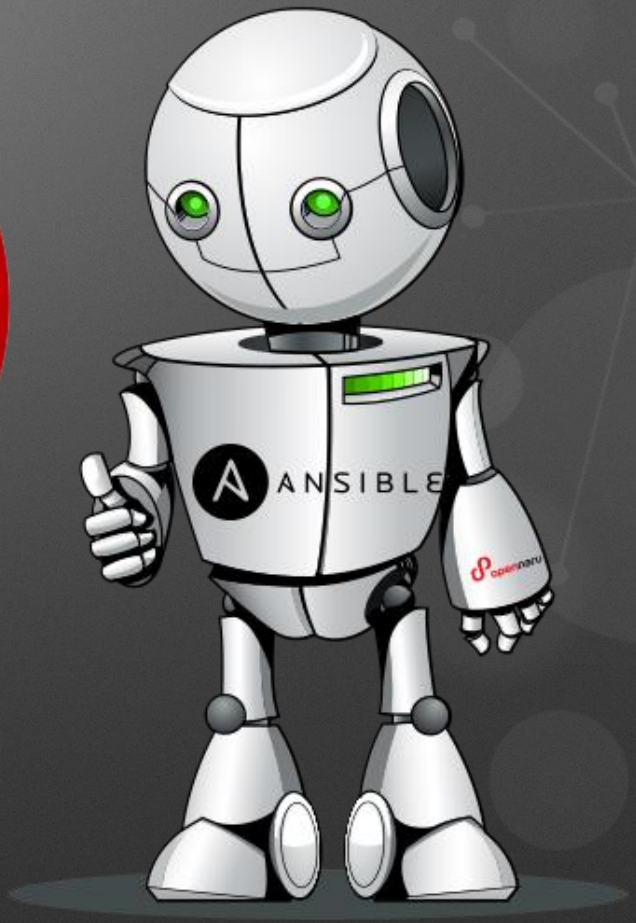
Comparison : Ansible vs. Puppet vs. Chef

opennaru

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	△	△	△	◎

opennaru





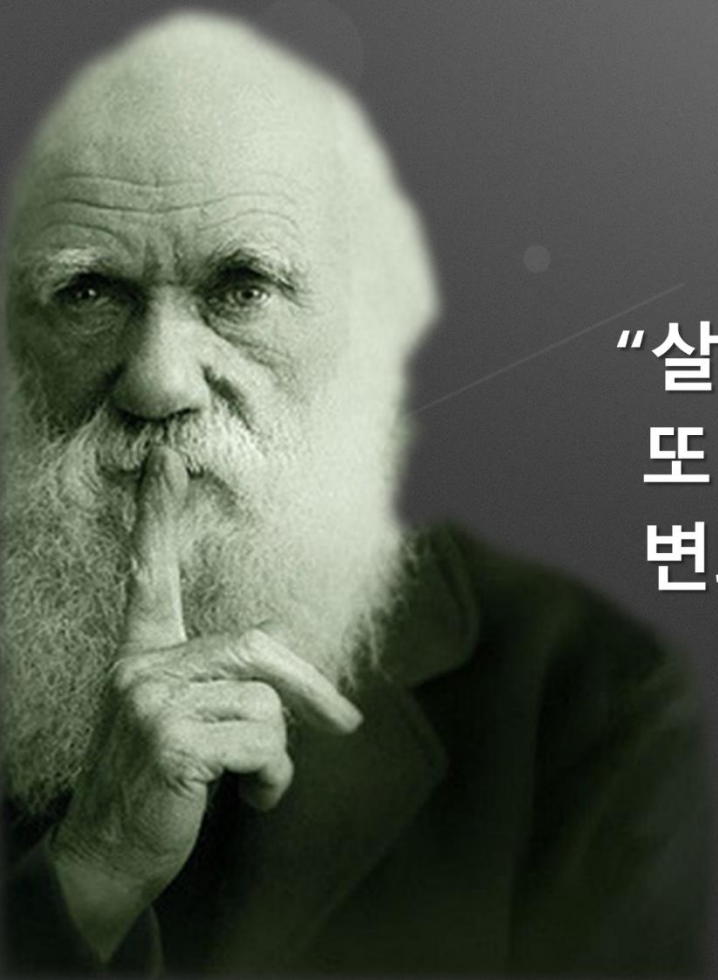
Ansible Core & Ansible Tower

opennaru

- 오픈소스로 제공되고 있는 Ansible (Core)
- 레드햇 제품으로 제공되지 않음
  (※ 2016년 9월말 기준)

- Red Hat 에서 서브스크립션으로 제공하는 제품
- 많은 Red Hat 제품과 연계되고, 오픈소스 Ansible Core에 많은 기능이 추가

opennaru



"살아 남는 종(種)은 강한 종이 아니고,
또 우수한 종도 아니다.
변화에 적응하는 종이다. "

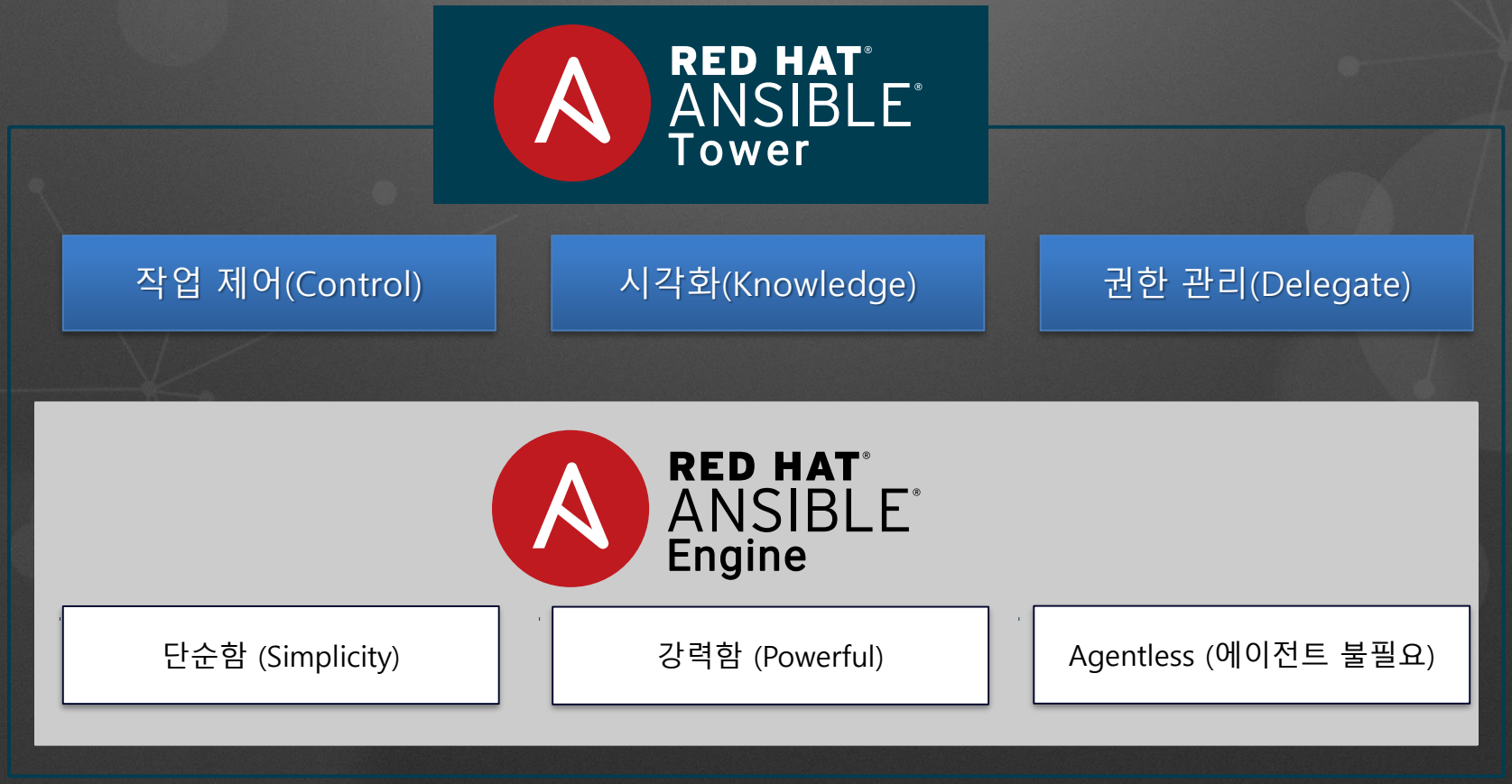
- Charles Darwin, 1809


Application Performance Management
Introduction to Ansible
KHAN
[apm]

Ansible Tower

opennaru

- Ansible Tower는 다양한 관리 기능을 제공하고
- 사용자는 Ansible Tower 통해 Ansible을 실행

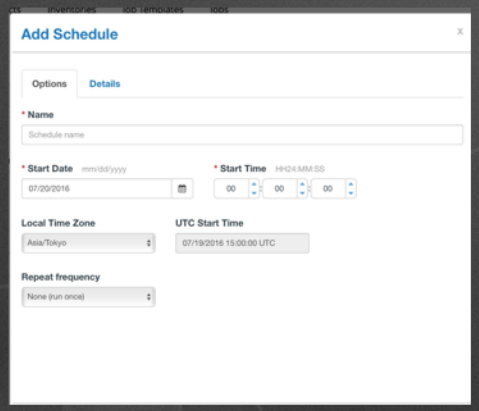


Ansible Tower

opennaru

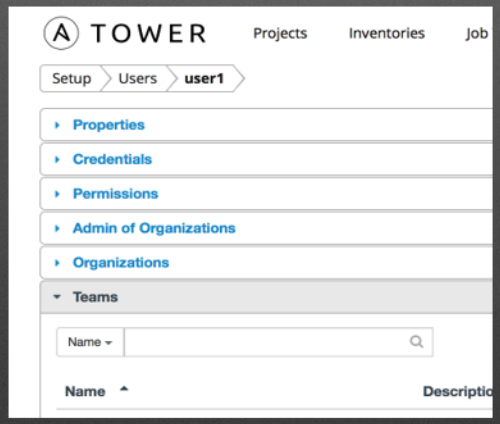
RED HAT® ANSIBLE® Tower

작업 제어(Control)



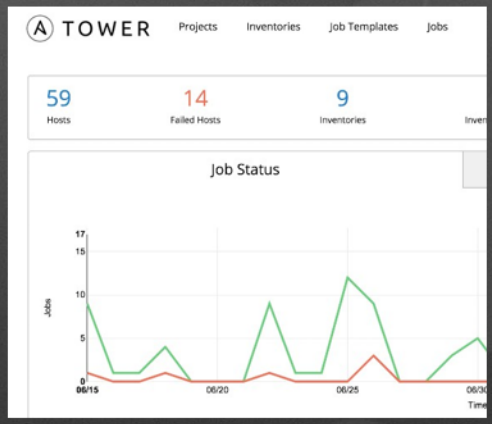
- 스케쥴에 따른 작업
- 중앙 통제 작업 / 일괄 실행
- Ad-hoc Command 실행
- 작업 추상화

권한 관리(Delegate)



- 사용자 인증 기능 (로그인)
- 사용자/팀 별 권한 설정

시각화(Knowledge)



- 대시 보드
- Job의 실행 결과
- Job Template 실행 결과
- 업데이트 실행
- Inventory/Host 실행 결과

ANSIBLE TOWER

opennaru

ACCESS CONTROL
역할 기반ACL,
LDAP과의 연계

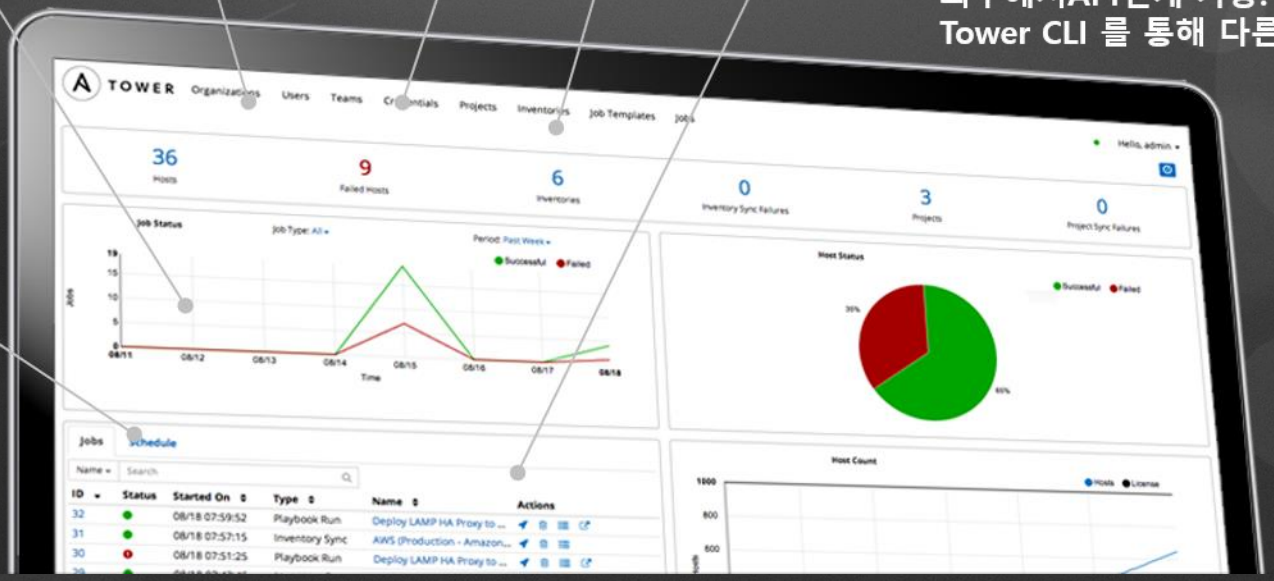
카탈로그 관리
Playbook의 종류와 대상 자원을 그래픽으로 관리

DELEGATION OF CREDENTIALS
작업 실행의 권한 관리

PUSH-BUTTON LAUNCH
작업 실행을 한 번의 클릭으로 시작

AUDITING
Ansible작업 실행 기록
를 드릴 다운 모니터

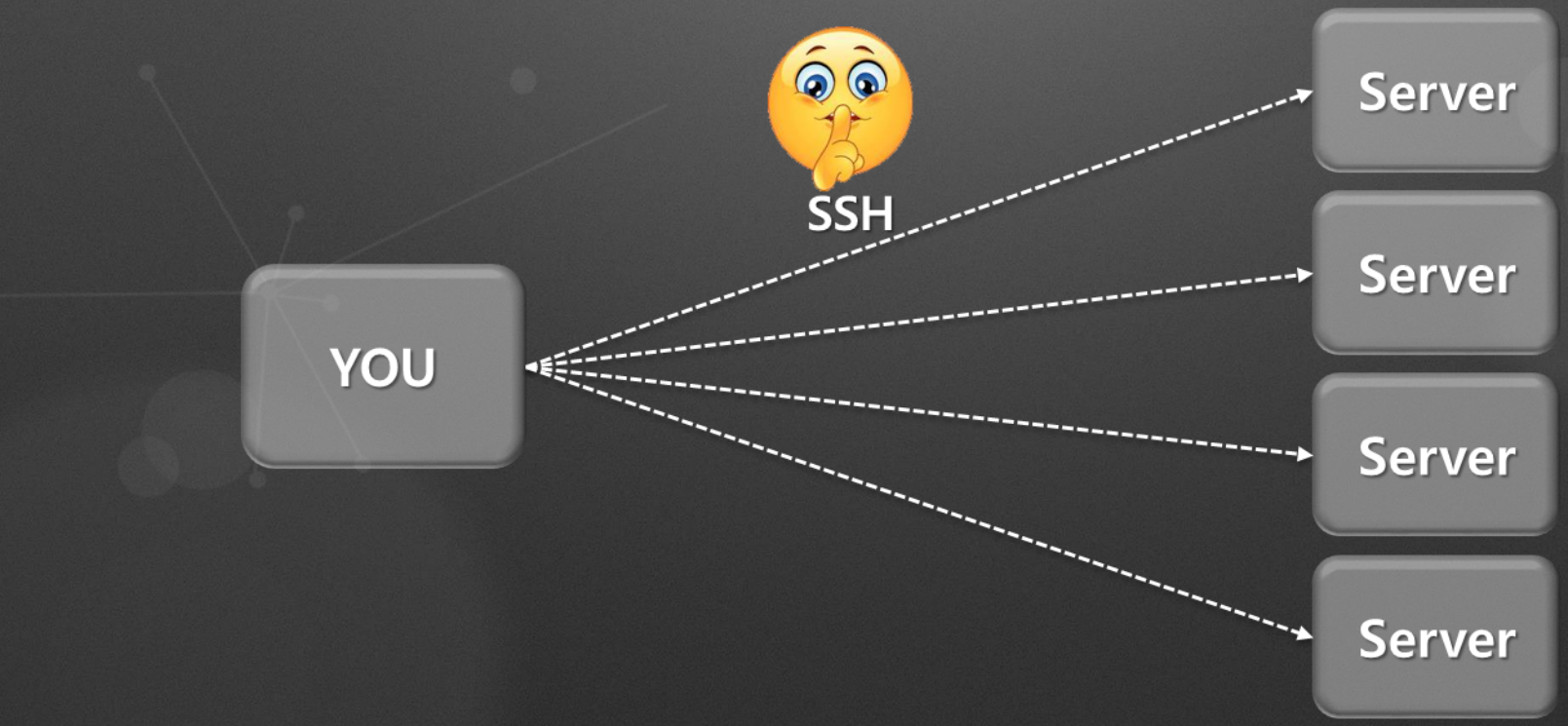
API & CLI
RESTful API 를 제공하기 때문에
외부에서API연계 가능.
Tower CLI 를 통해 다른 도구와 통합



JOB SCHEDULING
각종 작업 예약
및 자동 실행
상태 목록

Ansible works

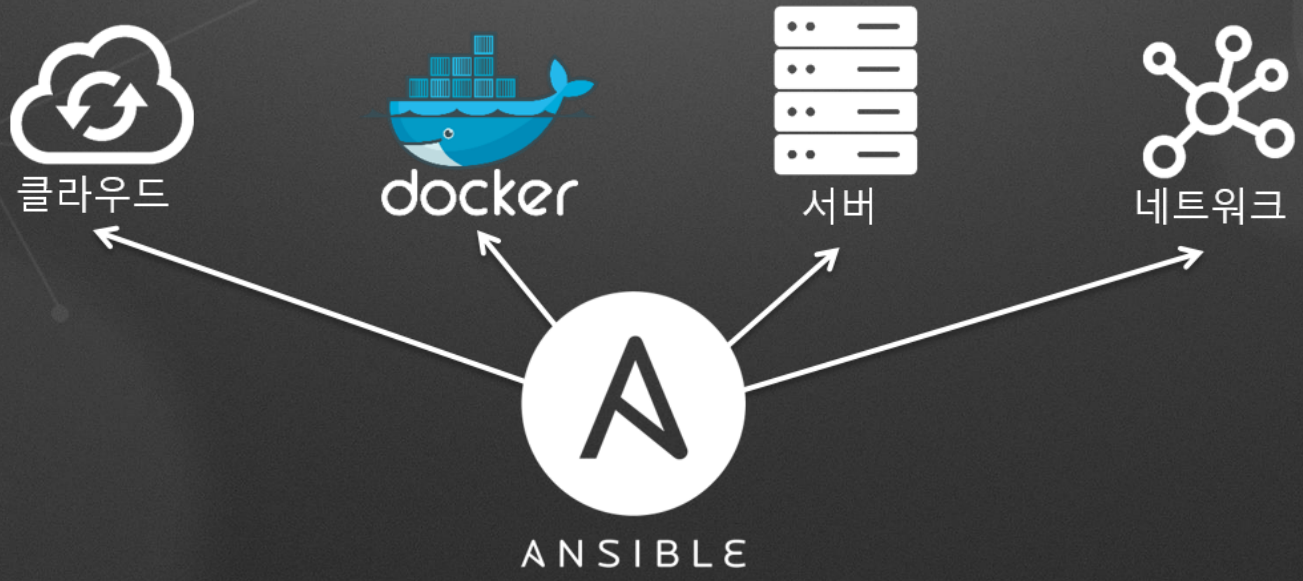
- Ansible works via SSH.
- No Master Server!
- No Agent on the Server is required.
- Configuration In YAML



What is Ansible?

opennaru

- 설치/배포
  - 패키지 인스톨, 설정 변경, 파일 전송, 서비스 시작/정지 등을 원격 조작
- 오케스트레이션
  - 서버, 네트워크, 서비스, 로드 밸런스, 방화벽(fire wall) 설정 및 배포를 자동화
- 구성 관리 도구
  - 신규시스템 구축이나 일상적인 운영 업무를 텍스트 파일화



ANSIBLE FEATURES

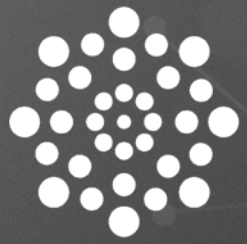
opennaru

Agentless

- 에이전트가 필요 없는 환경
- OpenSSH & WinRM 지원
- 즉각적인 사용 가능
- 높은 효율성과 보안성

Simple

- YAML 형식의 읽고 쓰기 쉬운 설정 파일
- 프로그래밍 스킬이 필요하지 않음
- 팀 간의 작업 공유가 쉬움
- 멱등성 지원
- 높은 생산성



Powerful

- 700개 이상 대다수의 서버와 네트워크 장비 지원
- 동시에 다수의 대상 서버에서 실행
- Bootstrap 부터 설정 변경까지 원스톱 실행
- 완벽한 구성 관리, 오케스트레이션, 배포

Ansible 로 자동화할 수 있는 것

opennaru

프로비저닝

- 소프트웨어 설치
- 구성/설정 변경
- 파일 전송

오케스트레이션

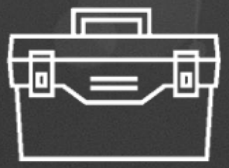
- 다양한 서버와 장비에 대한 작업 자동화
- Server / Router / Switch / FW / Load Balancer / Storage / Database /Cloud etc...

구성 관리

- 서비스 시작과 종료
- 상태 파악과 확인
- Batch 처리
- 업데이트 실행
- 보안 패치

Ansible 도입 기대효과

opennaru

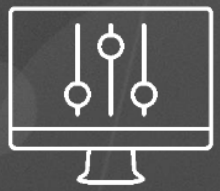


- 안전성 향상
  - 휴먼 에러 방지
  - 작업자에게 의존하지 않음 (인력 의존성 탈피)
  - 변경 이력 관리 : 누가, 언제, 무엇을?
  - 작업계획과 운영 환경의 차이 감소
- 작업 효율 향상
  - 대상 서버 수와 상관없이 구축할 수 있으며, 병렬 실행
  - 장시간 작업이나 야간 작업에 대한 인력의존성 탈피
  - 신속한 릴리스 작업
- 다른 툴과 통합하여 자동화와 효율성 향상
  - 버젼 관리툴(git, svn...)에 의한 순서/설정 관리
  - 자동 테스트 툴에 의한 환경 테스트(serverspec등)
  - 각종 CI툴과의 자동 연계 (jenkins등)
  - 모니터링 도구와 연계된 장애 대응 자동화(zabbix, nagios등)
  - Slack등과 연계해 채팅 베이스에서의 운용 작업 실행

Ansible 에 의한 오토메이션

opennaru

- **bootstrap**
  - IaaS를 위한 관리 API나 각종 커멘드를 이용하여 OS환경과 네트워크 설정
- **설정 관리**
  - **OS 설정**
    - 유저, 그룹 생성 등
  - **각종 미들웨어 설치/구성**
  - **각종 서비스와 demon 관리**
  - **어플리케이션 배포**
    - 소스 코드/빌드 애플리케이션 배포
    - 설정 파일 수정
- **오케스트레이션**
  - **여러 서버의 구성을 정리해 하나의 시스템에서 관리**
    - 부하 상황에 따른 Scale Out
    - 서비스 신규 추가나 다운을 모니터링

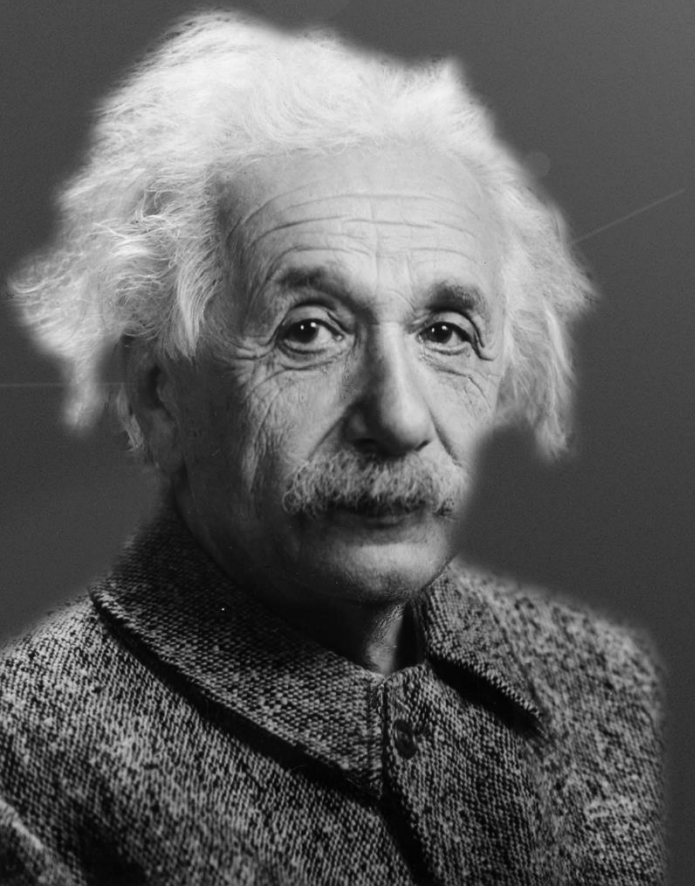


Idempotency (멱등성)

opennaru

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

opennaru



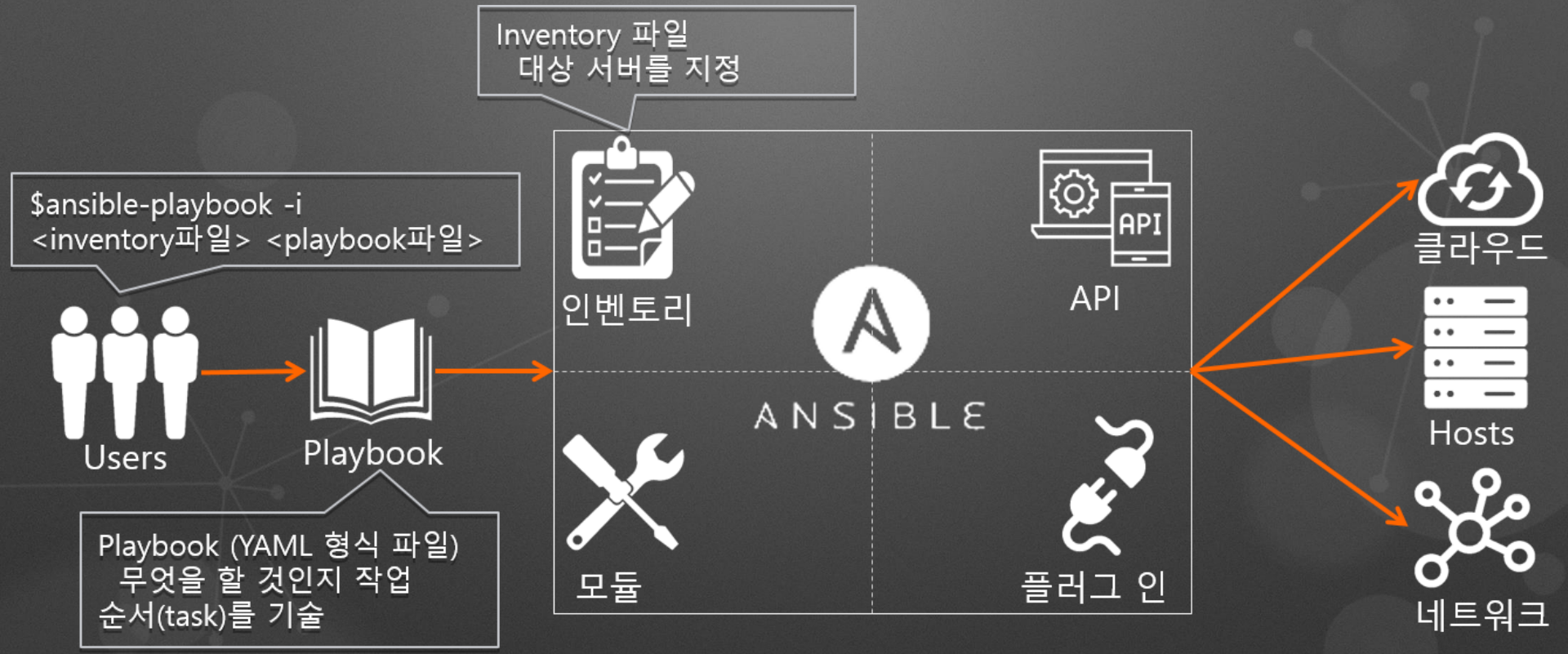
"'The measure of intelligence is the ability to change"

- Albert Einstein


Application Performance Management
Ansible Architecture
KHAN
[apm]

Ansible Architecture

opennaru



Inventory 파일

- inventory 파일은 리모트 서버에 대한 meta 데이터를 기술하는 파일
- 관리 대상 서버를 기술
  - 호스트명
  - IP주소
  - ssh 사용자명
- remote host를 Grouping 할 수 있음
- 기본 파일은 /etc/ansible/hosts
- ansible-playbook 커멘드의 i 옵션으로 지정

Playbook 예제

- ansible-playbook 커멘드의 실행
- $ ansible-playbook i inventory_file playbook.yml

```
---
- name: Apache 설치와 실행                    #Playbook의 설명
  hosts: app                                  #app 그룹
  remote_user: root                           #리모트 사용자
  vars:                                       #변수
    http_port: 80
    max_clients: 200
  tasks:                                      #실행하는 순서의 내용
  - name: httpd의 인스톨                       #실행시에 처리마다 표시되는 이름
    yum: pkg=httpd state=latest
  - name: Apache config 파일에 변수를 설정해 전개
    template: src=/srv/httpd.j2 dest=/etc/httpd.conf
  - name: httpd를 기동
    service: name=httpd state=running
```

TARGET 섹션

VARS 섹션

TASKS 섹션

TASK

모듈

실행 순서

playbook

opennaru

- playbook은 ansible의 환경 설정, 배포를 가능케 함
- YAML 문법을 사용하여 정의
- linux 기반 권한 관리 (user, group) 지원
- 하나의 playbook은 하나 또는 그 이상의 'play'를 정의하며, play의 목적은 여러 호스트들에 잘 정의된 'role'과 'task'를 매핑하는 역할을 맡음
- 그 외의 기능 들
  - 반복 (with_item, with nested, until ...)
  - 조건 분기 (when, register, ...)
  - 다른 playbook 참조 (include, role, ...)
  - 외부 정보 참조
  - 환경 변수, 파일 등 (environment, lookup, vars_prompt,...)
  - 커스텀 모듈을 이용한 확장

YAML 형식이란

- '야믈'이라 발음
- YAML은 마크업이 보다는 구조화 된 데이터를 표현하기 위한 텍스트 형식의 포맷
  - 사람이 쉽게 읽을 수 있는' 데이터 직렬화 양식
  - 마크업 언어와 다르기 때문에 읽기 쉽고, 쓰기 쉽고, 알기 쉽다
- yaml 이라는 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
- 설정 파일이나 데이터 저장 형식, 로그 파일로 자주 사용
- YAML 형식의 사양에 대해서는 http://www.yaml.org/ 를 참조

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

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

태스크(Task)란

- 정형화 된 작업의 열거
- 태스크는 관련 지을 수 있었던 모듈을 호출
- 모듈은 Python나 Bash로 기술되고 있다

```
tasks:
  - name: check install httpd
    yum: name=httpd state=latest
```

모듈(Module)이란

opennaru

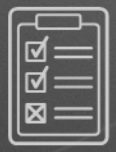
- **특정 목적을 위해 작성된 Ansible 백엔드**
- **주로 Python 으로 구현**
- **대표적인 모듈들**
  - **패키지 관리**
    - yum, apt
      지정 패키지(및 의존 패키지) 설치
  - **서비스 제어**
    - service
      서비스 시작/정지 등
  - **파일 처리**
    - File, copy, fetch, template
      파일 배포(copy, template), 파일 수집(fetch) 등
  - **커맨드 실행**
    - command, shell
      외부 커맨드 실행과 그 출력 결과 보고 등
  - **소스 코드 관리: git, subversion**

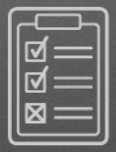
Ansible Module

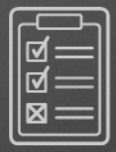
opennaru

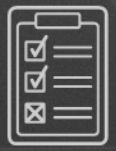
- Module : 대상 호스트에서 실행하는 라이브러리들
- 700이상 Module제공
- Ansible 커뮤니티에서 지속적으로 새로운 Module 공개

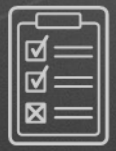

Amazon EC2 설정
50 가지 이상

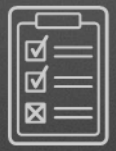

Linux 각종 설정
80 가지 이상

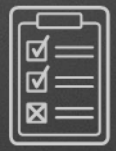

CISCO 스위치 설정
15 가지 이상


OpenStack 설정
40 가지 이상

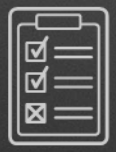

Azure 설정
18 가지 이상

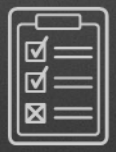

DevOps 관련 설정
10 가지 이상

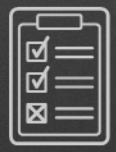

OSS DBMS 설정
7 가지 이상

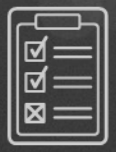

Vertica 설정
5 가지 이상

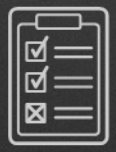

VMWare 설정
20 가지 이상

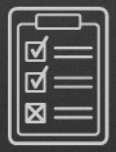

Big-IP 설정
19 가지 이상


NetApp 설정
15 가지 이상


네트워크 설정
15 가지 이상


Windows 설정
40 가지 이상


Docker 설정
8 가지 이상


그 외 다수

http://docs.ansible.com/ansible/list_of_all_modules.html

변수 정의(vars)

- 태스크 섹션 전에 vars: 섹션으로 변수를 정의

```
vars:
  hello: Hello
tasks:
  - name: Hello World
    debug: msg="{{ hello }} Ansible"
```

조건 분기 실행(when)

- 태스크에서 모듈명 다음 줄에서“when:” 을 기술하여 모듈의 실행 조건을 정의
- 지역 변수나 vars 정의된 변수에 대해 등호와 부등호를 이용하여 조건식이 true의 경우에 실행

```
tasks:
  - name: install Apache Web server
    yum: name=httpd state=latest
    when: ansible_os_family == 'RedHat'
  - name: install Apache Web server
    apt: name=apache2 state=latest
    when: ansible_os_family == 'Debian' or ₩
          ansible_os_family == 'Ubuntu'
```

루프 실행(Loops)

- 태스크에서 모듈명의 다음 줄에서 “with_000:”으로 기술하여 모듈에{{ item }} 변수를 전달

- with_items
- with_nested
- with_dict
- with_lines
- with_indexed_items
- with_ini
- with_flattened
- with_file
- with_fileglob
- with_first_found
- with_together
- with_subelements
- with_random_choice
- with_sequence

템플릿

- YAML 파일 뿐만 아니라 모든 파일에서 활용 가능
- 일반적으로 파일 확장자명을 .j2로 함
  - Index.php.j2
  - Mysql.출.j2
- Template task 일 때 jinja2가 적용 가능(copy task는 적용 안됨)

```
tasks:
  - name: deploy my.cnf
    template: src=my.cnf.j2 dest=/etc/my.cnf

filename: my.cnf.j2
[mysqld]
user = {{ mysql_user }}
port = {{ mysql_port }}
datadir = /var/lib/mysql
socket = /var/lib/mysql/mysql.sock
pid-file = /var/lib/mysqld/mysqld.pid
```


Application Performance Management
Cloud Ready System
KHAN
apm

Provisioning 제품 비교

opennaru

제품	정의	개발언어	Agent 여부	통신방법
Ansible	YAML	Python	필요없음(SSH)	JSON
Chef	DSL	Ruby	필요	REST / STOMP
Puppet	DSL	Ruby	필요	HTTP SSL

Ansible이 할 수 있는 일

opennaru

- 설치
  - OS 패키지 설치 : yum, apt-get, zypper 등
  - Language 패키지 설치 : npm, bower, gem, pip 등
- 다운로드
  - get_url, wget, git, subversion, fetch 등
- 환경설정 파일 배포
  - copy, template
- 실행
  - shell, command, task, script
- 기타
  - Cloud, Clustering, Database, Crypto, Network, Remote Management, Windows등 다양한 모듈을 제공

Ansible Playbook을 이용한 Apache 설치

apache_setup.yml

```
---
- name : install apache
  hosts : apache
  user: root

  tasks :
  - name : install httpd
    yum: name=httpd state=latest

  - name : start apache service
    service: name=httpd state=running
```

Playbook

Play

Tasks
Handler

Modules

/etc/ansible/hosts

```
[apache]
web[01:03].opennaru.com
192.168.11.3
```

Inventory

```
$ ansible-playbook apache_setup.yml
```

Install

Apache 설정 파일 변경

apache_setup.yml

```
---
- name : install apache
  hosts : apache
  user: root

  tasks :
  - name : install httpd
    yum: name=httpd state=latest

  - name : copy httpd.conf file
    template: src=httpd.conf dest=/etc/httpd/conf/httpd.conf

  - name : start apache service
    service: name=httpd state=running
```

Playbook

Play

Template Module

Apache 설정 파일 변경 – 환경변수

opennaru

httpd.conf Templates

환경변수

```
ServerRoot "{{ SVC_HTTPD_DIR }}"

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
Listen {{ HTTPD_PORT }}
```

환경변수

Ansible 제어문

opennaru

Looping

```
- name: add users
  user: name={{ item }} state=present groups=user
  with_items:
    - open
    - naru
    - admin
```

Conditional

```
- name: install apache
  apt: name=httpd state=latest
  when: ansible_distribution == 'Ubuntu'

- name: install apache
  yum: name=httpd state=latest
  when: ansible_distribution == 'RedHat'
```

Include

```
- include: test/main.yml
```

Ansible Ad-hok Task 실행

opennaru

```
$ ansible <host-pattern> [options]
```

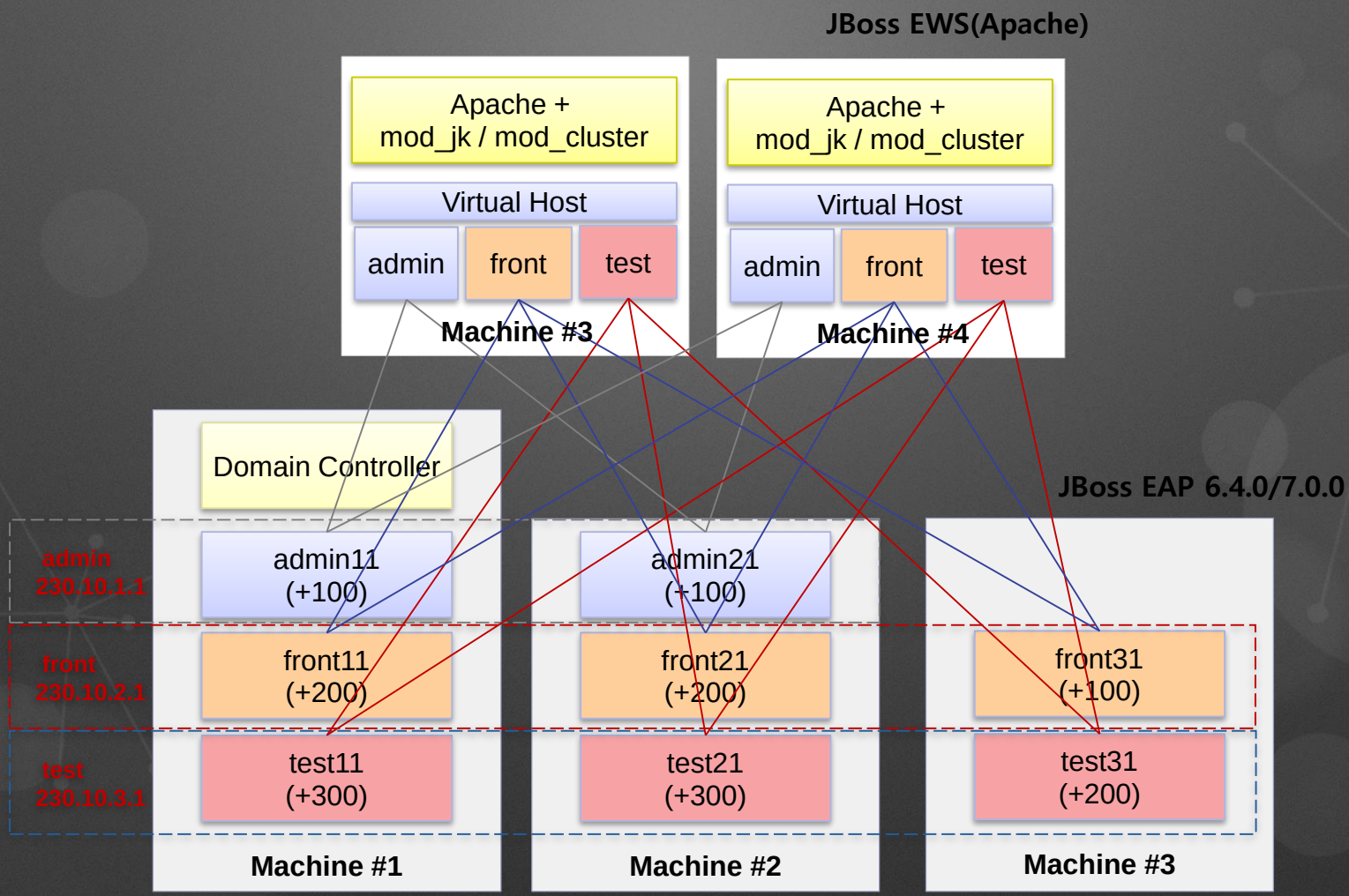
특정 Host에 명령 실행

```
$ ansible 192.168.11.3 -m ping -u root --ask-pass
SSH password:
192.168.23.14 | success >> {
    "changed": false,
    "ping": "pong"
}
```

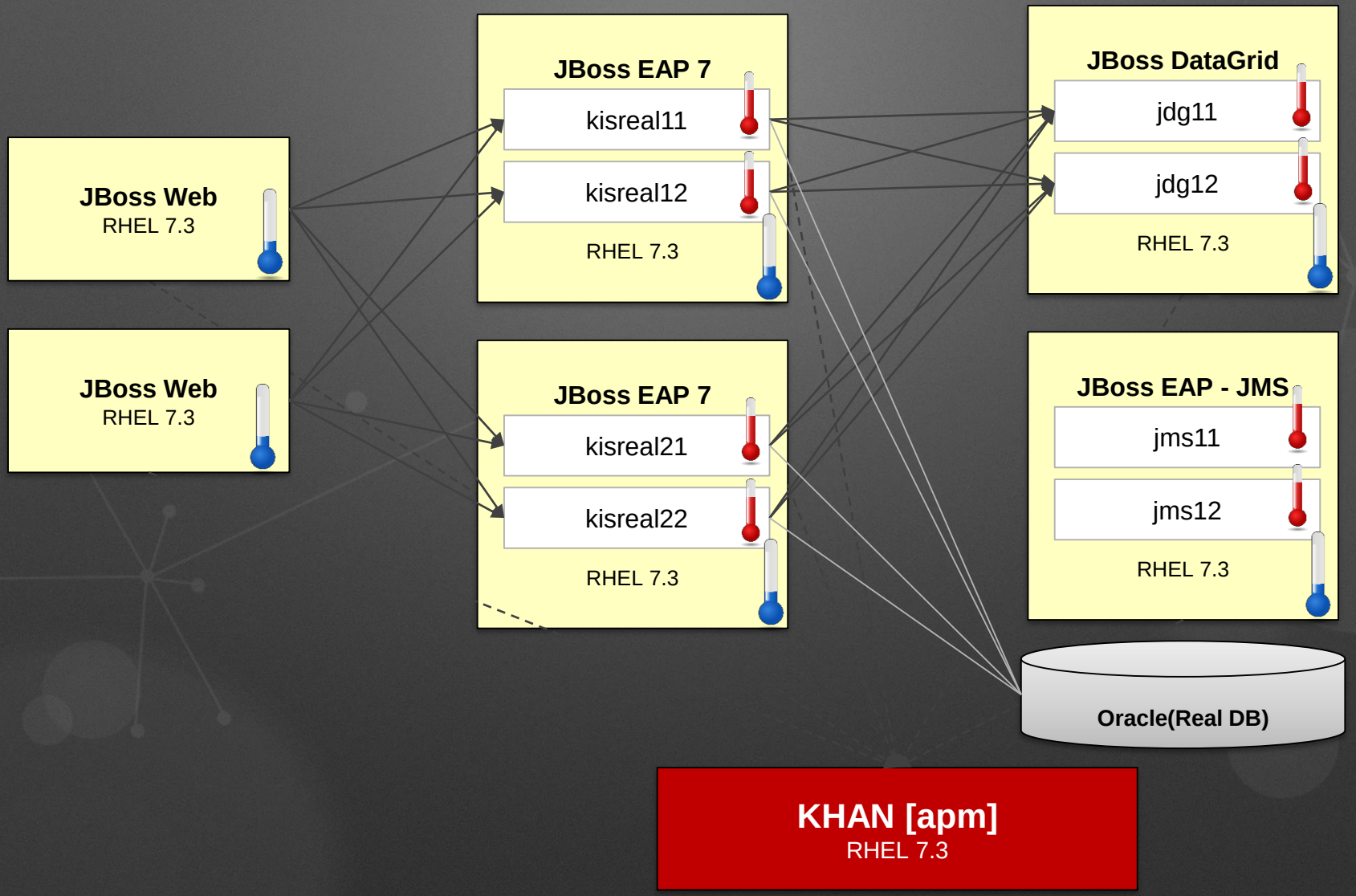
특정 Host에 명령 실행

```
$ ansible web01.opennaru.com -m command -a '/sbin
  /reboot' --ask-pass
```

실제 Web / WAS 구성 – 다양한 서비스

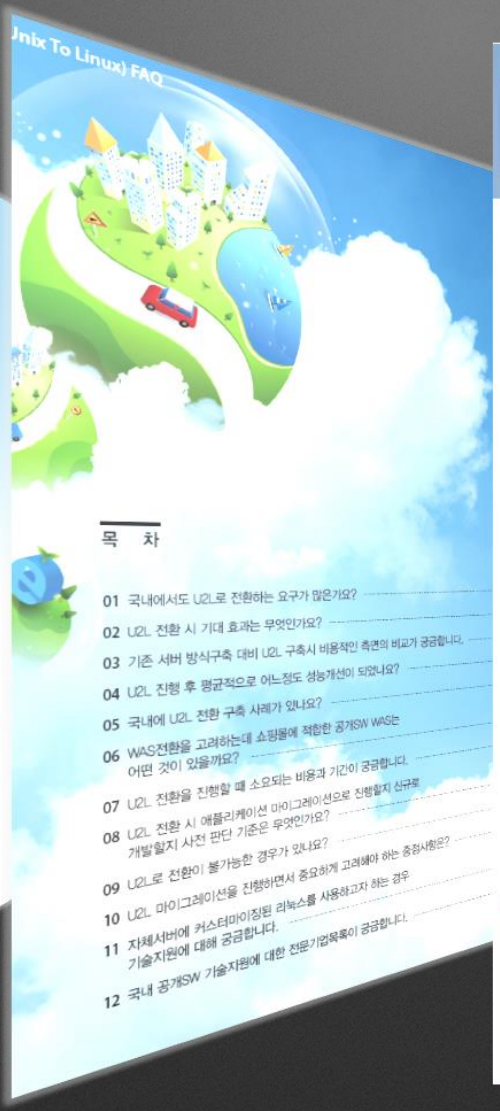


실제 Web / WAS 구성 – 다양한 구성요소




Application Performance Management
KHAN [apm] 설치/구성/튜닝
KHAN
[apm]

KHAN [apm] – Unix2Linux 전환지원



U2L(Unix To Linux) FAQ

06 WAS전환을 고려하는데 쇼핑몰에 적합한 공개SW WAS는 어떤 것이 있을까요?

- 쇼핑몰의 경우 이벤트 등 사용자가 일시에 몰리는 현상이 발생할 수 있으므로 웹서비스의 유연한 확장성이 요구되는 대표적인 분야입니다.
- 상용 벤더들의 경우 향후 로드맵을 활용할 수 없게 종속되는 경우가 일반적입니다. Jboss, Tomcat과 같은 공개SW는 표준 기술을 사용하여 다른 벤더로 쉽게 마이그레이션이 가능하며 표준화된 기술을 사용하여 향후 어느 방향으로든 나아갈 수 있는 토대가 된다는 장점이 있습니다.
  - --> (JBossEAP) 빠른 부팅시간과 라이트한 모듈구성으로 인해 WAS분야 글로벌 마켓에서 성능 또한 이미 검증된 제품입니다.
  - --> (Apache Tomcat) 서블릿과 JSP만 처리하므로 가볍고 속도가 빠른 장점을 가졌으며 오랜 시간 개발되고 자바 웹개발자들이 많이 사용하는 WAS
  - --> (Jetty) 빠른 성능과 간편한 설정으로 사용자층을 넓혀가고 있는 오픈 소스 WAS
- WEB/WAS 마이그레이션의 경우, 서블릿 등 JAVA EE의 표준스펙에 맞게 작성된 애플리케이션은 특별한 이슈 없이 진행이 가능합니다.
- JAVA EE 표준스펙의 경우는 문제가 없지만, 일부 특수 코드나 특정 WAS의 벤더 종속성이 있는 소스코드일 경우 수동으로 확인해야 하므로 전문가의 지원이 필요할 수 있습니다.
- 웹시스템 구축에 필요한 공개SW 웹서버와 WAS 등을 자동으로 구성해 주는 자동프로비저닝 솔루션(KHAN 등)을 이용하여 즉각적인 인스턴스 확장이 가능하도록 구성이 가능합니다.
  - --> 서비스 운영 중 WEB/WAS에 대한 기술지원을 받아야 할 경우에도 KHAN 등을 이용하여 모니터링 및 장애감지, 애플리케이션 문제점 분석 등이 가능하나 JVM GC 튜닝, Thread Pool 튜닝, Heap Dump 분석 등은 전문가의 지원이 필요할 수도 있습니다.

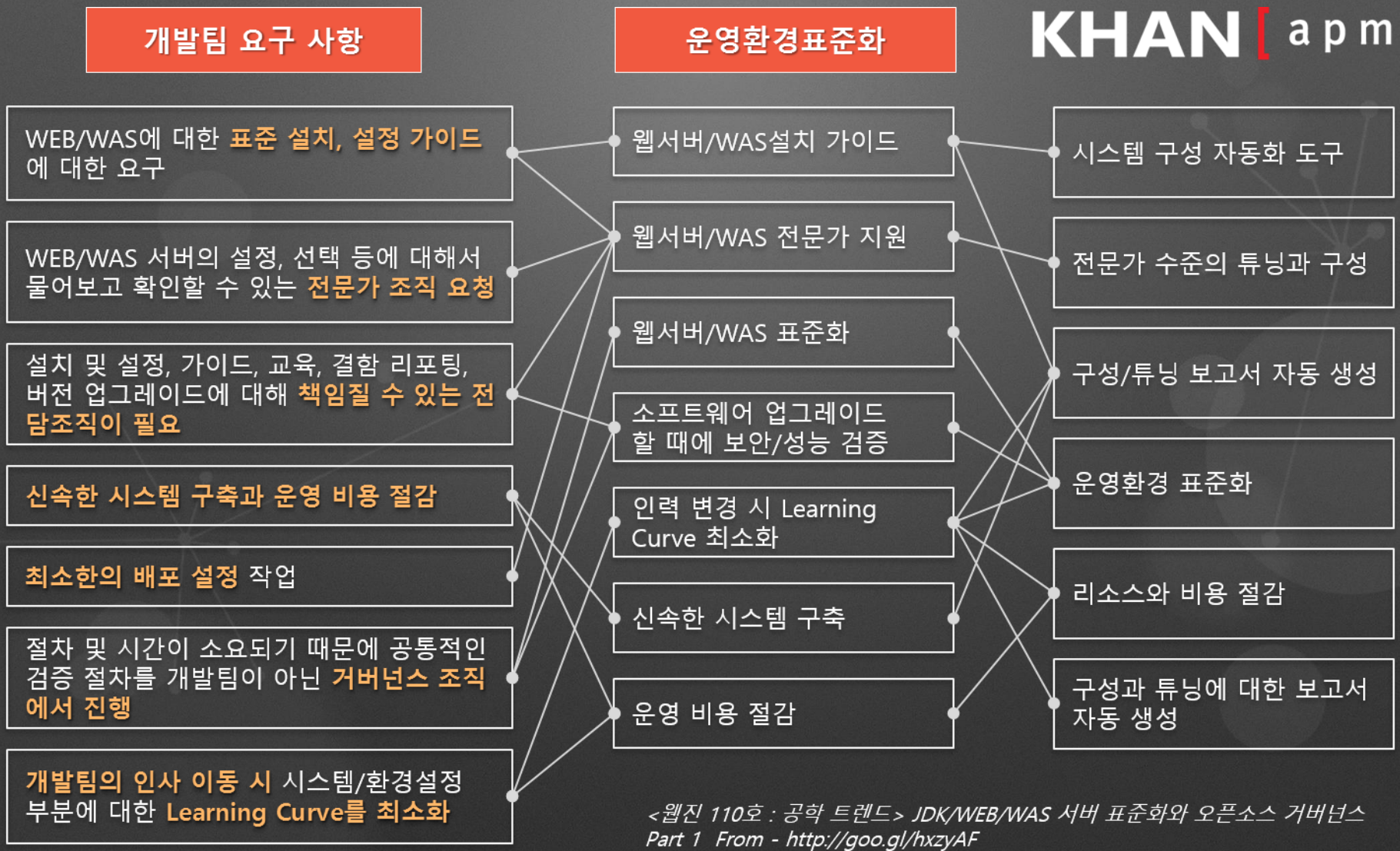
새로운 시작 – 산 너머 산

KHAN [APM] - 미들웨어 설치/구성/튜닝 자동화

운영환경표준화 - 개발팀 요구 사항

opennaru

KHAN [apm]

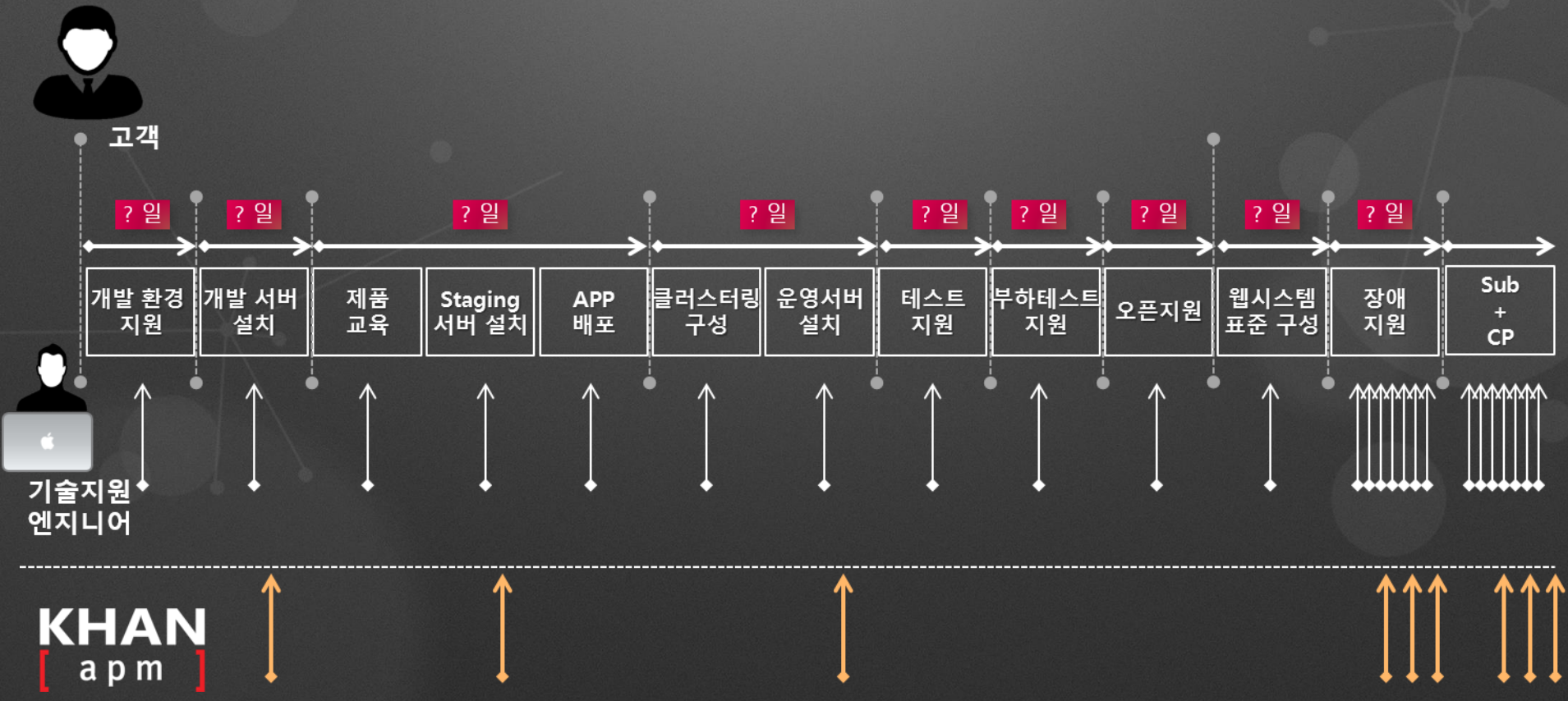


<웹진 110호 : 공학 트렌드> JDK/WEB/WAS 서버 표준화와 오픈소스 거버넌스 Part 1 From - http://goo.gl/hxzyAF

기존 기술지원과 KHAN APM 의 비교

opennaru

- 웹시스템 구축은 일정에 따라 다양한 기술 지원 요청이 발생
- 각 단계별로 전문적인 기술 지원이 필수적임
- KHAN을 사용할 경우 기존 방법 대비 On-Site 지원 횟수 감수
- 온사이트 장애 지원에 대한 횟수 감수



고객 요구사항에 따른 웹 시스템 작업

opennaru

KHAN [apm]

수 일 이상 기술 지원

고객

- 1 일: 개발 환경 지원
- 개발 서버 설치
- 1 일: 웹 서버 설치
- WAS서버 설치
- 웹 서버/ WAS서버연결
- 1 일: 클러스터링 구성
- 애플리케이션 배포 지원
- 테스트 지원
- 1 일: 부하테스트 지원
- ? 일: 오픈 지원
- ? 일: 웹 시스템 표준 구성
- ? 일: 장애 지원

기존 단순 설치	최적화된 웹시스템 구성			
웹서버 설치	웹서버 튜닝	Worker MPM 구성	mod_cluster 설정	웹서버
mod_jk 설정	웹서버 설치 보고서 작성	로깅 디렉터리 설정	웹취약점 조치	
WAS 설치	도메인 모드 구성	쓰레드 풀 튜닝	애플리케이션 배포 구성	WAS
JDK 설치	멀티 인스턴스 구성	각종 Dump 디렉터리 구성	Native 라이브러리 설정	
	JDBC 커넥션 풀 튜닝	모듈 설정	JSP 변경 체크 설정	
	모니터링 설정	Log4J 설정	WAS 서버 설치보고서 작성	
운영 계정 설정	추가 rpm 설치	블루투스 off	커널 파라미터 설정	OS
방화벽 설정	Huge Page 설정	limits.conf 설정	Cpuspped 설정	
	OS 설정 보고서 작성	운영을 위한 쉘 구성	로깅 디렉터리 설정	
배포 테스트	기본 네트워크 테스트	클러스터링 테스트	단위 테스트	테스트
	애플리케이션 배포 테스트	통합 테스트	Benchmark 테스트	
	운영관리 테스트	멀티캐스트 테스트	JDBC 테스트	

감사합니다.

KHAN
[apm]

제품이나 서비스에 관한 문의

콜 센터 : 02-469-5426 (휴대폰 : 010-2243-3394)

전자 메일 : sales@opennaru.com