

실전! 시스템 관리자를 위한 Ansible

Development Process



WATERFALL



AGILE



DEVOPS



Application Architecture



MONOLITHIC



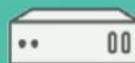
N-TIER



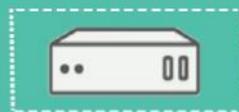
MICROSERVICES



Deployment & Packaging



PHYSICAL SERVERS



VIRTUAL SERVERS



CONTAINERS



Application Infrastructure



DATA CENTER



HOSTED

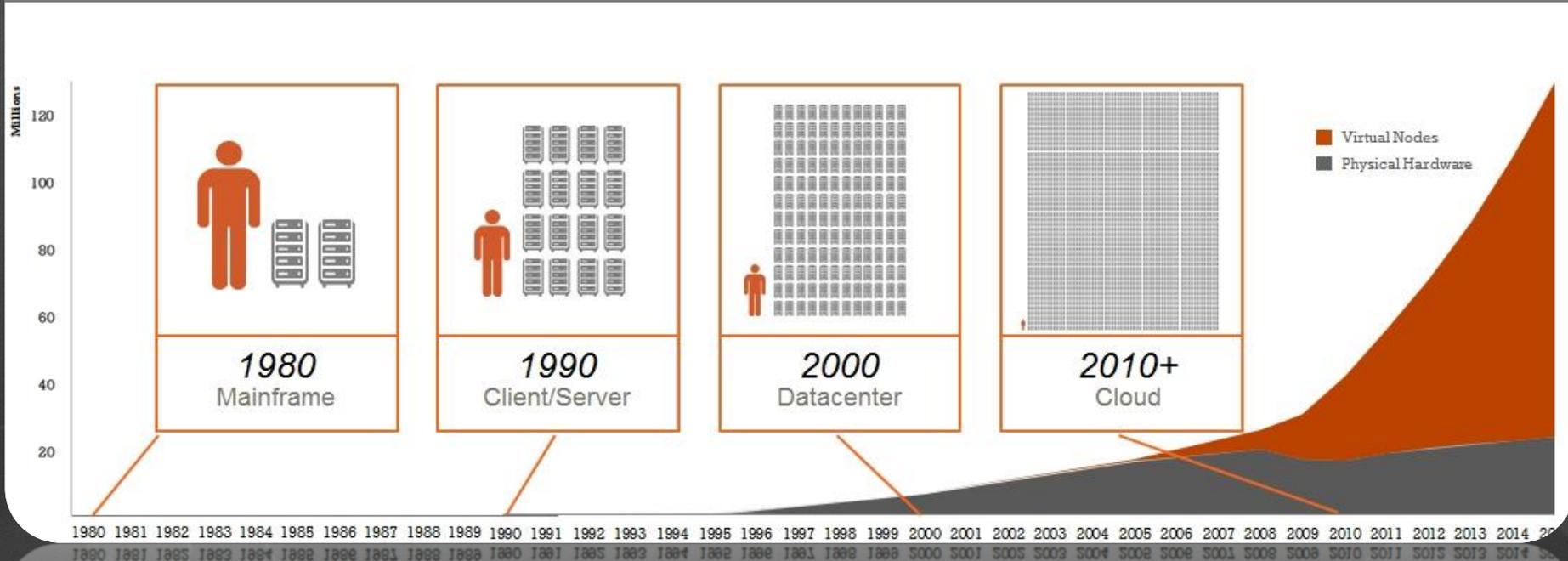


CLOUD



Increasing scale and complexity means we need admin automation

Scale x Complexity > Skills



Opscode gets more venture dough for its Chef

From - <http://goo.gl/dLcjS>



Apache HTTPD를 설치(3대 머신)



간략한 보안 조치(3대 머신)

Application Performance Management

Ansible 소개

KHAN
a p m
g b w

WHAT IS ANSIBLE AUTOMATION?

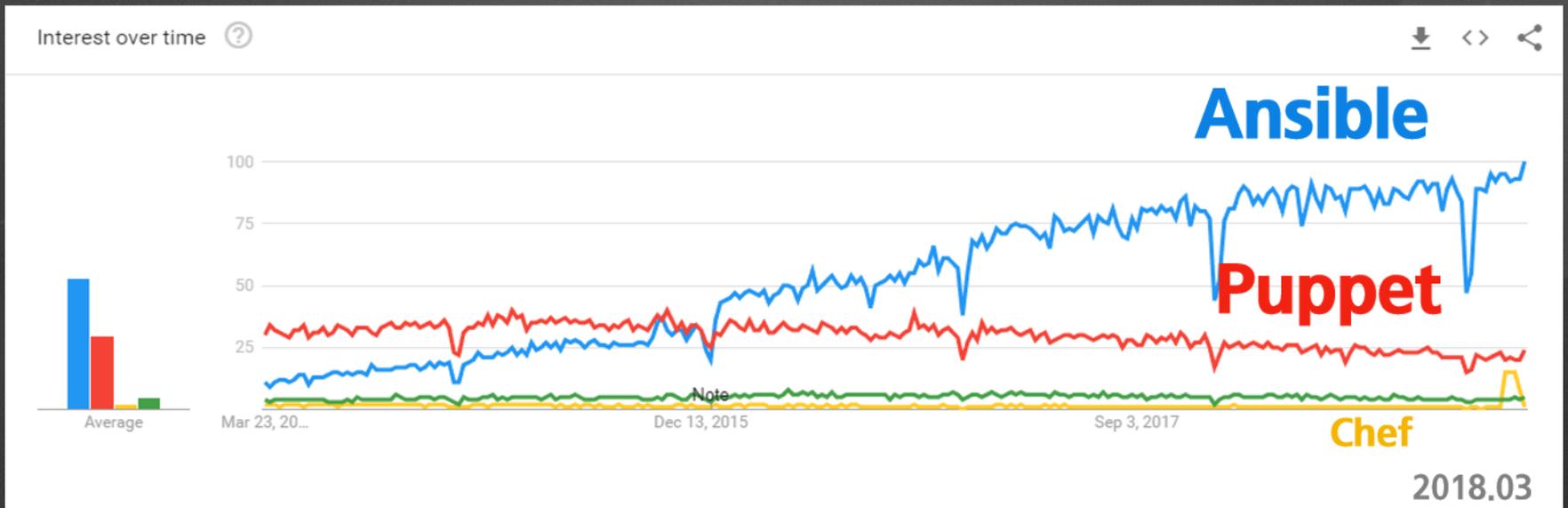


Red Hat acquired Ansible at year end 2015

- *Ansible*은 클라우드 프로비저닝, 구성 관리, 응용 프로그램 배포, 서비스 내
 - 오케스트레이션 및 기타 많은 IT 요구 사항을 자동화하는
 - 간단한 **IT 자동화 엔진**입니다.

“원격 프로토콜을 이용해서 배포, 구성 및 오케스트레이션을 제공하는 자동화 도구”

Global Google Trends : Ansible vs. Puppet vs. Chef



Ansible 비교



제품	Puppet	Chef	Ansible
벤더	Puppet Labs	Opscode	Red Hat (이전 Ansible.Inc)
출시	2005 년	2009 년	2012 년
구현 언어	Ruby	Ruby/Erlang (서버)	Python
라이선스	Apache License Ver2.0	Apache License Ver2.0	GNU Public License Ver3
도입 사례	○	◎	◎
정보량	○	◎	◎
확장성	◎	◎	◎
Web UI	◎	◎	◎ (Ansible Tower)
정의 파일	자체 DSL (Ruby 기반)	YAML	YAML
코드 관리	manifest	recipe	playbook
에이전트 설치	Pull (Agent)	Pull (Agent)	Push (Agentless)
시스템 복잡성	△	△	◎

ANSIBLE FEATURES



Agentless

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

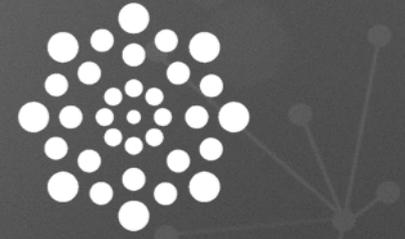
Architecture



Simple

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

Playbook

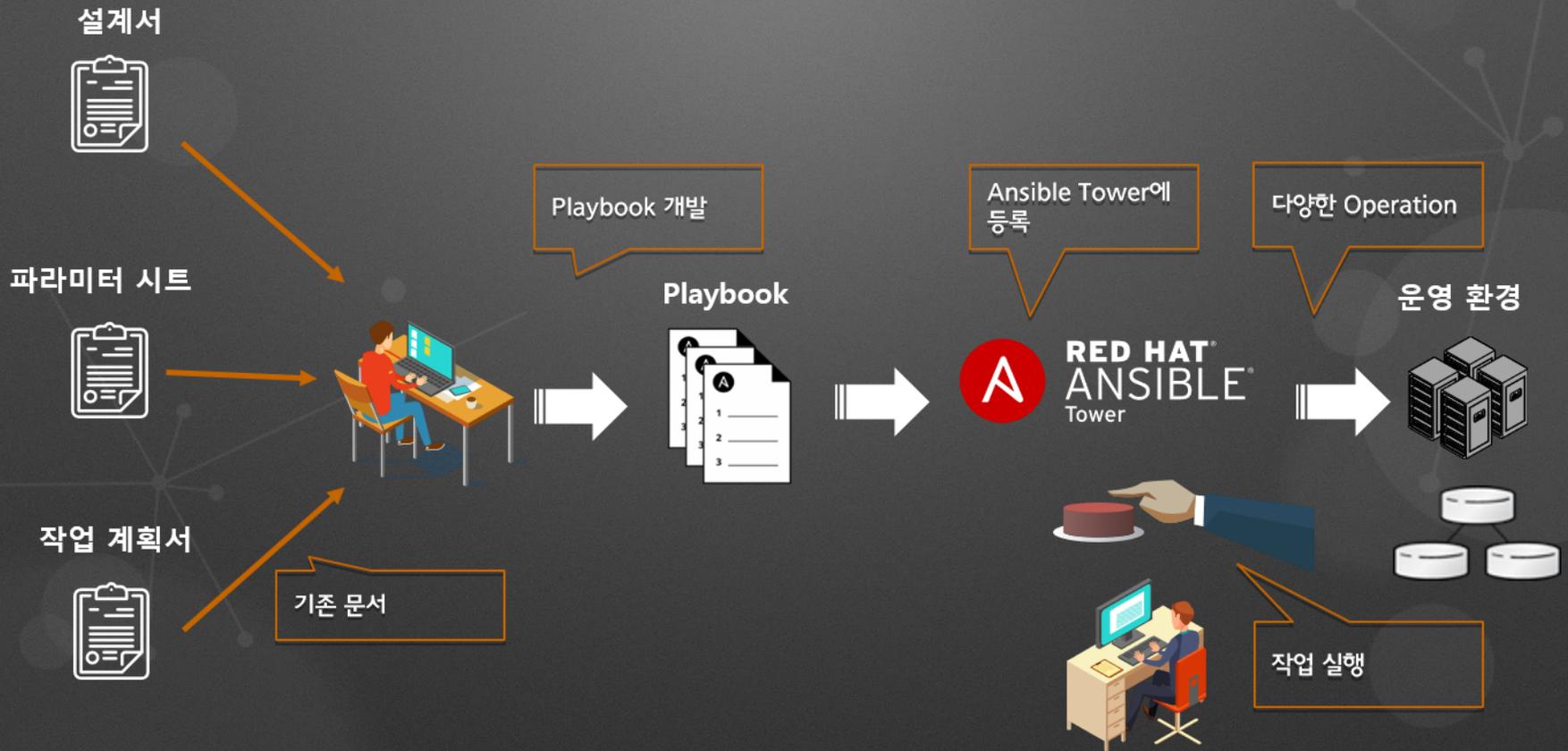


Powerful

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

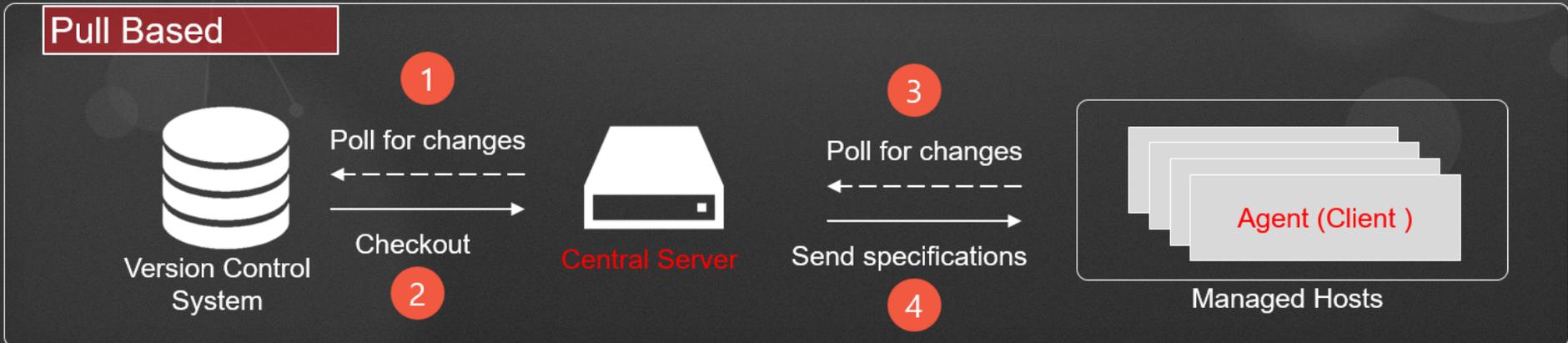
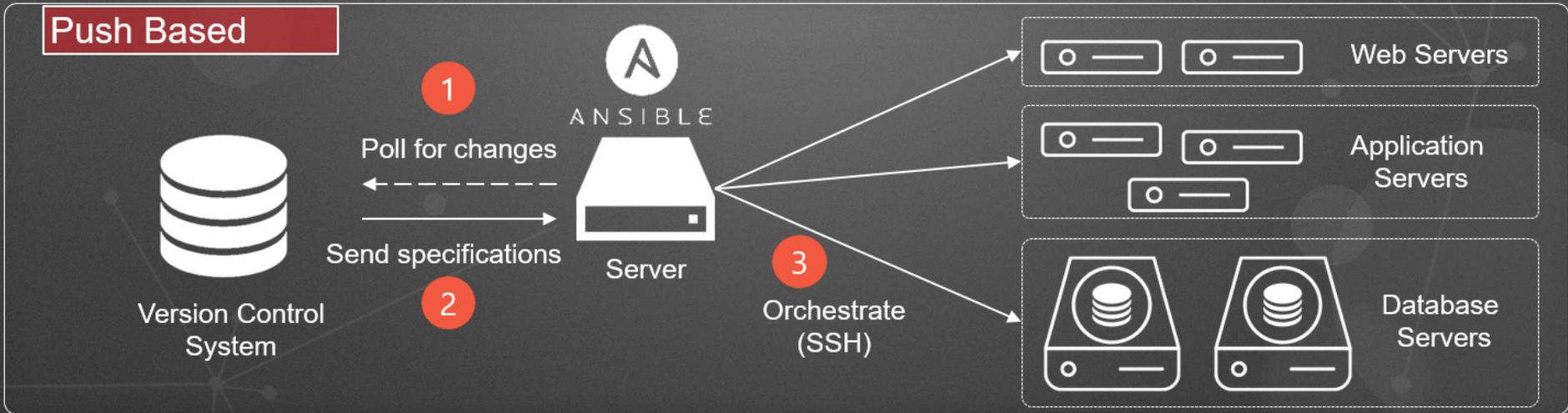
Modules

Ansible 에서의 자동화의 흐름



Push Based vs. Pull Based

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



Ansible Core & Ansible Tower



ANSIBLE

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



**ANSIBLE
TOWER**
by Red Hat®

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



- Ansible AWX
 - Ansible Tower의 오픈소스 Upstream 버전
- GUI 관리 환경 제공

Ansible 로 자동화할 수 있는 것



프로비저닝

- 소프트웨어 설치
- 구성/설정 변경
- 파일 전송



오케스트레이션

- 다양한 서버와 장비에 대해서 자동으로 순서 실행
- Server / Router / Switch / FW / Load Balancer / Storage / Database /Cloud etc...



구성 관리

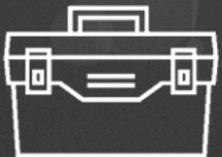
- 상태 확인
- Batch 처리
- 업데이트 실행
- 보안 패치



- 안전성 향상
 - 휴먼 에러 방지
 - 작업자에게 의존하지 않음 (인력 의존성 탈피)
 - 변경 이력 관리 : 누가, 언제, 무엇을?
 - 작업계획과 운영 환경의 차이 감소



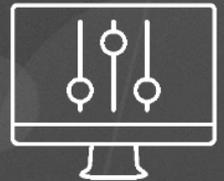
- 작업 효율 향상
 - 대상 서버 수와 상관없이 구축할 수 있으며, 병렬 실행
 - 장시간 작업이나 야간 작업에 대한 인력의존성 탈피
 - 신속한 릴리스 작업



- 다른 툴과 통합하여 자동화와 효율성 향상
 - 버전 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

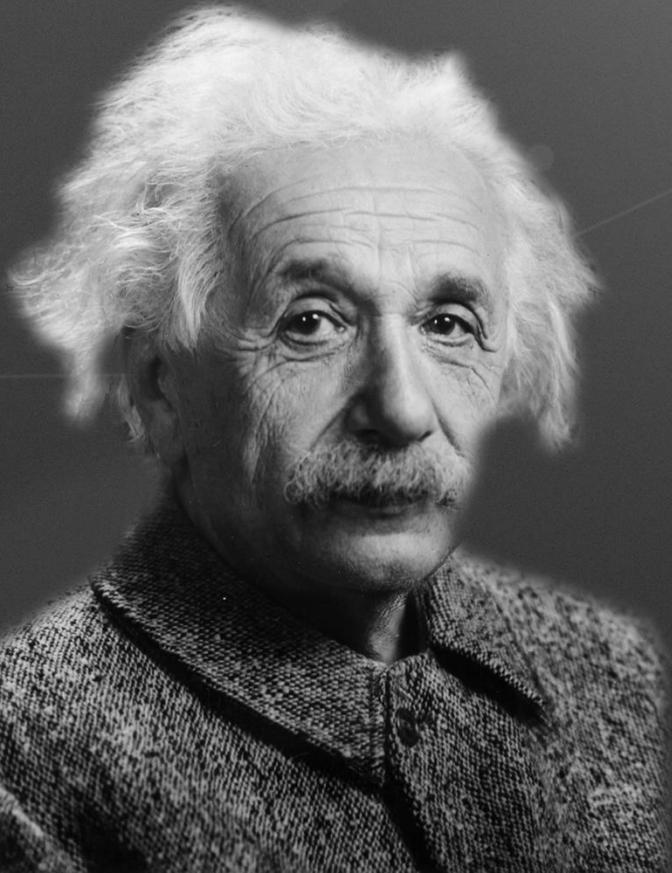
Ansible 에 의한 오토메이션

- **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 어플리케이션 배포
 - 설정 파일 수정
- **오케스트레이션**
 - 여러 서버의 구성을 정리해 하나의 시스템에서 관리
 - 부하 상황에 따른 Scale Out
 - 서비스 신규 추가나 다운을 모니터링



Idempotency (멱등성)

- 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



**“The measure of intelligence is
the ability to change”**

- *Albert Einstein*

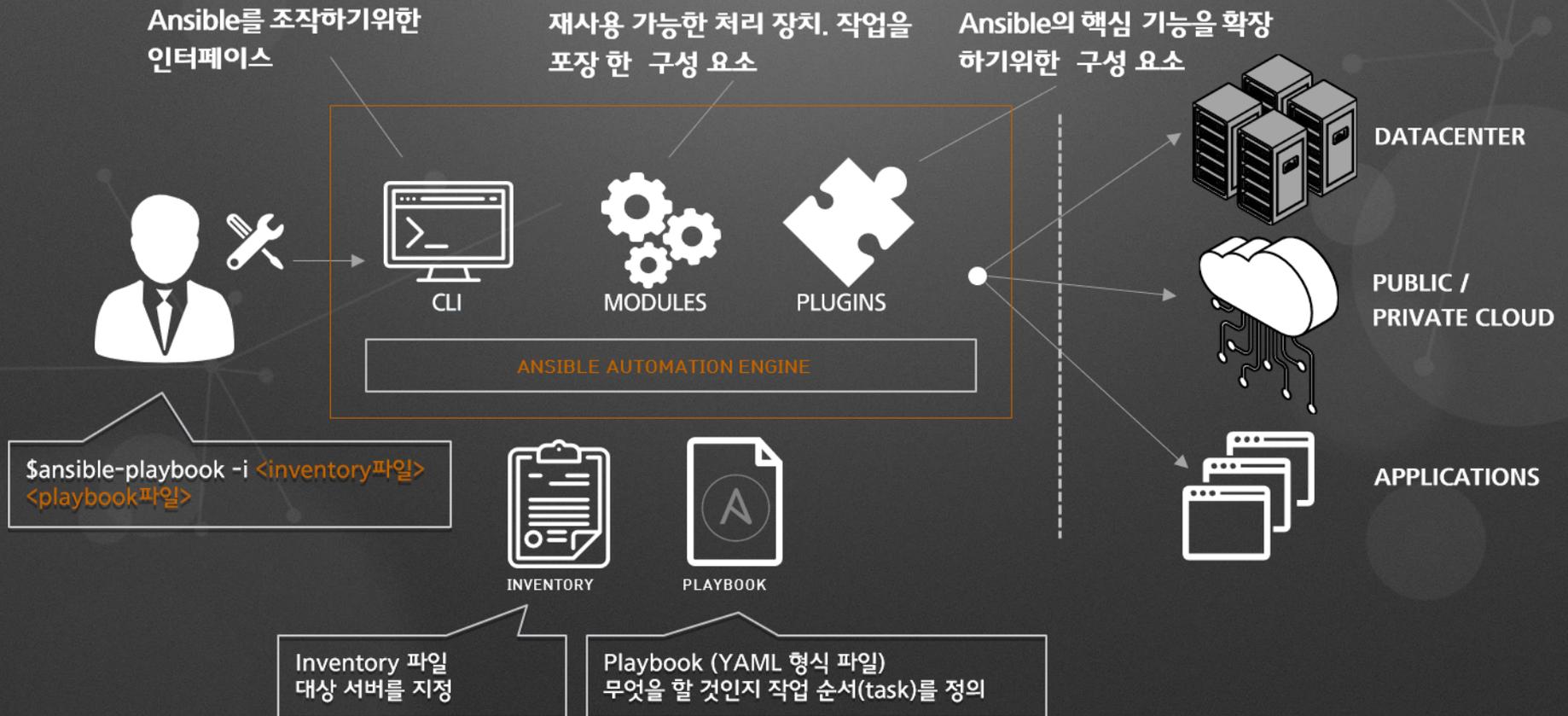
Application Performance Management

Ansible Architecture



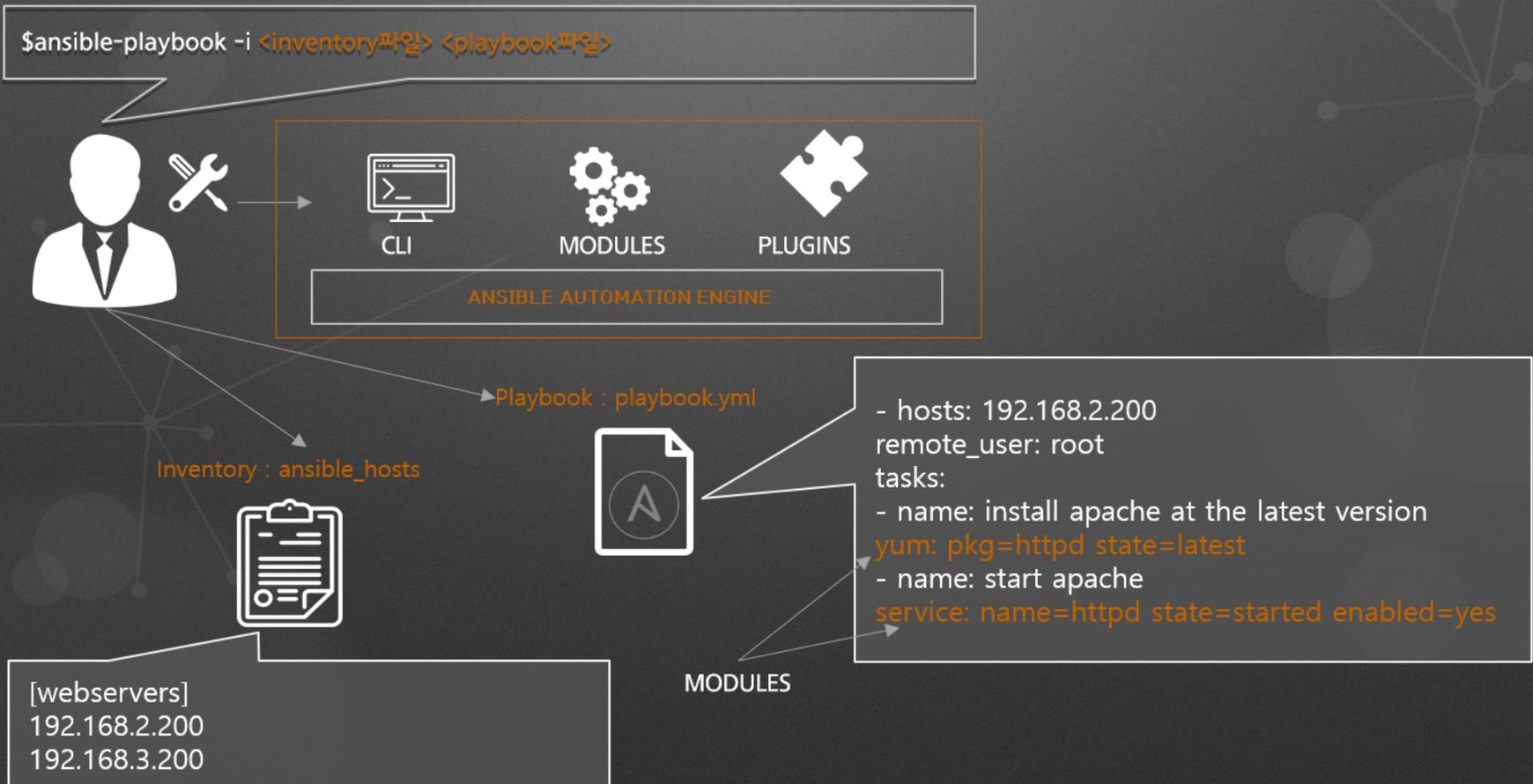
Ansible 핵심 컴포넌트

- 대상 호스트에 Python 필요 (Python 2.4 이상)
- 사용자는 inventory 파일과 Playbook 파일 만 작성



Ansible 사용 방법

- 사용자는 inventory 파일과 Playbook 파일 만 작성
- “192.168.2.200” 서버에 대해서 root 사용자로 yum 을 사용해 Apache Httpd 를 설치 하고 Apache Httpd 을 실행

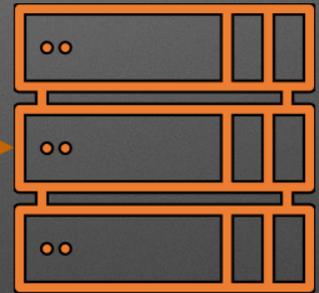


ansible-playbook 커맨드 실행

1 \$ ansible-playbook -i ansible_hosts playbook.yml



SSH



3

```
# yum install httpd
# service httpd start
# chkconfig httpd on
(OR systemctl enabled httpd)=0
```

구성되는 서버
대상 서버에는 에이전트는 필요 없으나
Python이 설치 되어야함.

```
PLAY [192.168. 2.200]
*****

GATHERING FACTS
*****
ok: [192.168. 2.20]
:
:
PLAY RECAP
*****
192.168. 2.20 : ok=2  changed=2  unreachable=0  failed=0
```

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

```
[db]
db-1.example.com
db-2.example.com
db-3.example.com
```

```
[app]
app-1.example.com
app-2.example.com
```

그룹

Playbook 예제

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

TARGET
섹션

VARS
섹션

TASKS
섹션

모듈

```

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

실행 순서 ↓

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
- yam1 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
- bar:
  - baz
  - kwa
  - 3.0
  - 5
  
```

YAML

태스크(Task)란

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

```
tasks:
```

```
- name: check install httpd
```

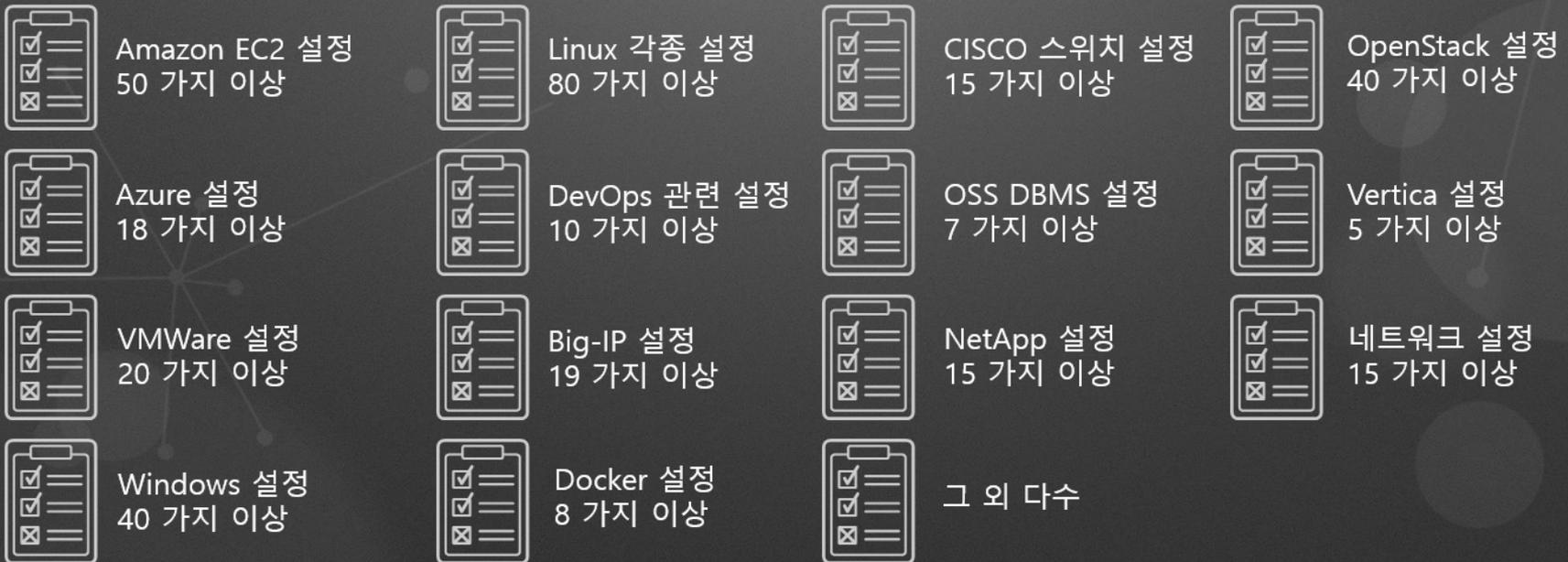
```
  yum: name=httpd state=latest
```

모듈(Module)이란

- 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

- Module : 대상 호스트에서 실행하는 라이브러리들
- 2,000 개 이상의 Module을 제공
- Ansible 커뮤니티에서 지속적으로 새로운 Module 공개



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 }}"
```

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.conf.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/mysql/mysql.pid
```

Ansible이 할 수 있는 일

- 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, Windows 등 다양한 모듈을 제공

Ansible Playbook을 이용한 Apache 설치

Playbook

Play

Task Handler

Module

Inventory

Install

```
apache_setup.y  
_ml
```

```
- 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
```

```
/etc/ansible/hosts
```

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

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

apache_setup.yml

Playbook

Play

Template Module

```
- 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
```

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

Looping

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

Array Loop



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'
```

Conditional



Include

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

Include



Ansible Ad-hoc Task 실행

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

```
$ 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
```

특정 Host에 명령 실행

Application Performance Management

Ansible 적용 효과

KHAN

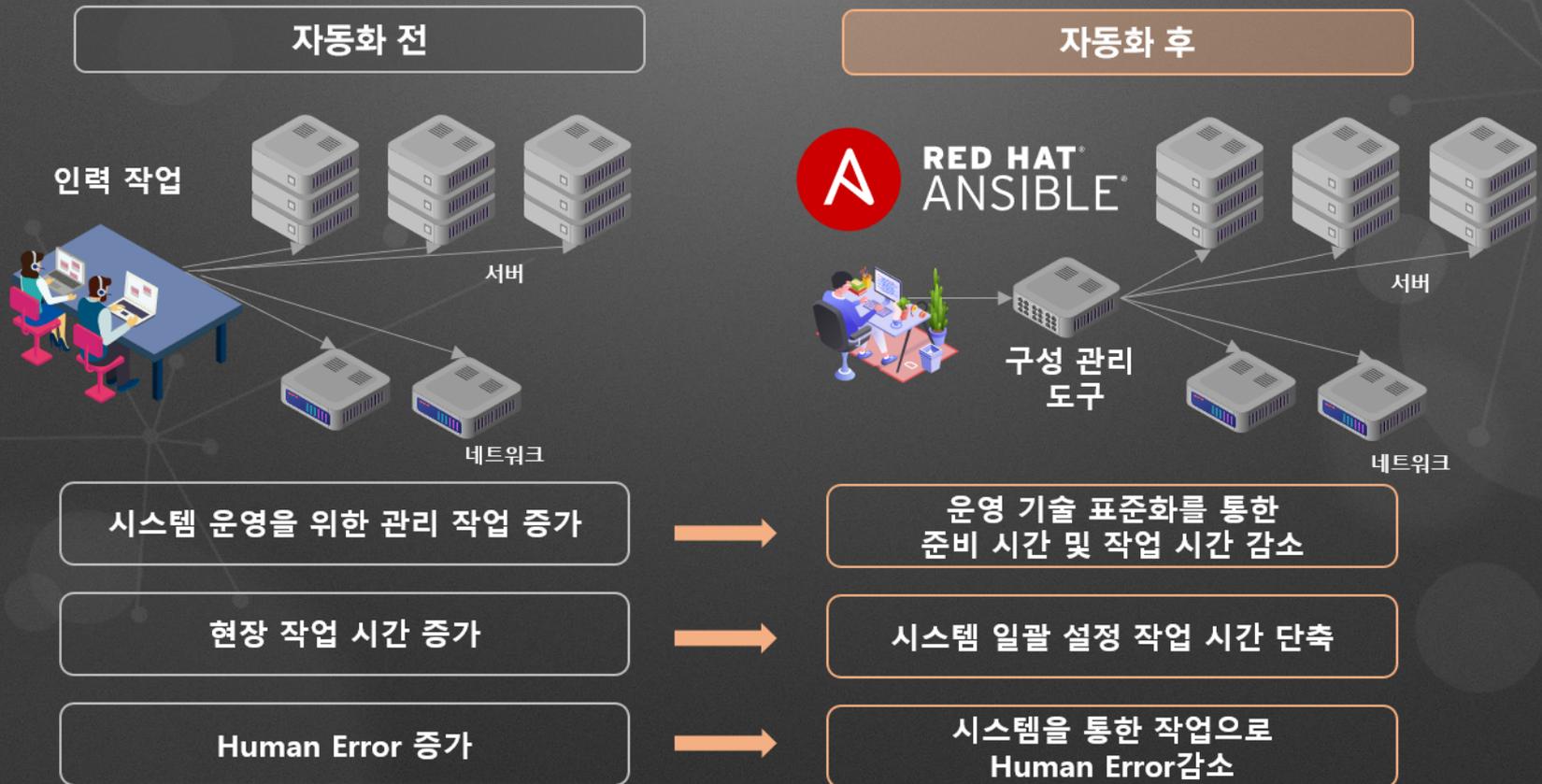
apm

gbw

KHAN

Ansible 을 통한 IT 인프라 운영 자동화

- IT 인프라의 대규모화, 가상화 및 고도화에 따라 IT 장비에 대한 환경설정 및 정보 취합이 복잡하고 어려움
- 작업 계획시간과 현장 작업 시간의 증가와 휴먼 에러의 증가

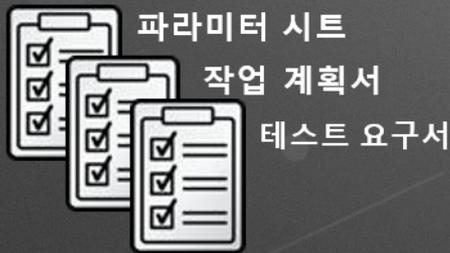


Ansible 자동화 추진 방안

- Start Small Think Big

단순한 작업들을 대체
(개발이 쉬운 부분부터 일부분 자동화)

대체



Playbook



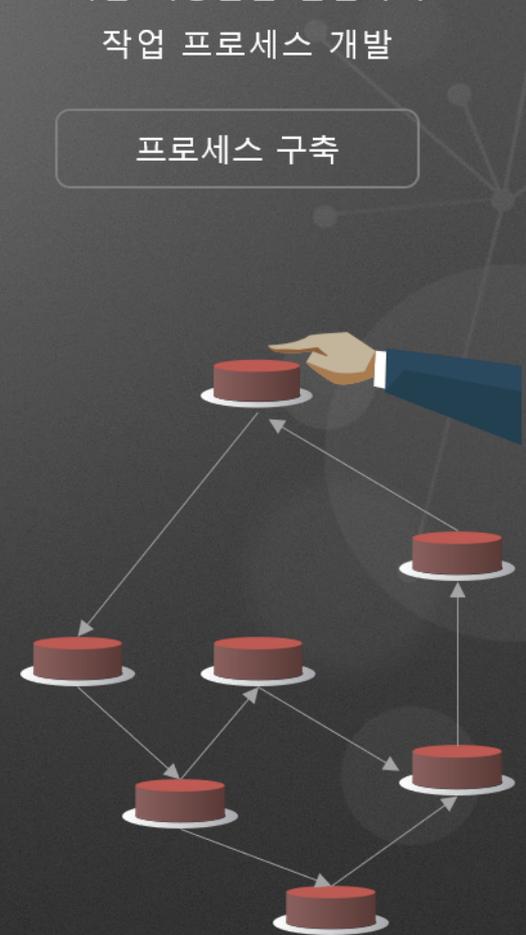
자동화를 통하여
운영 작업 위탁 (버튼)

기능화 (서비스화)



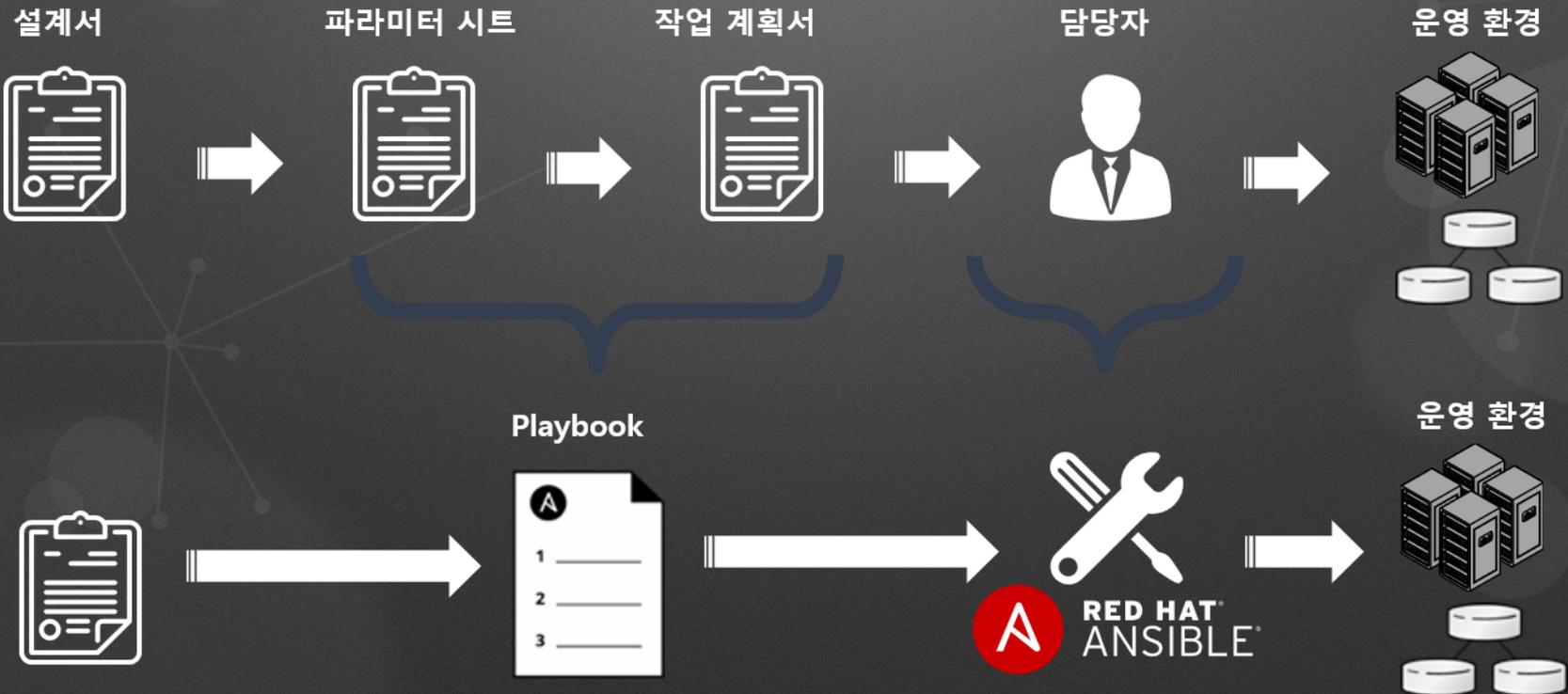
작은 기능들을 연결하여
작업 프로세스 개발

프로세스 구축



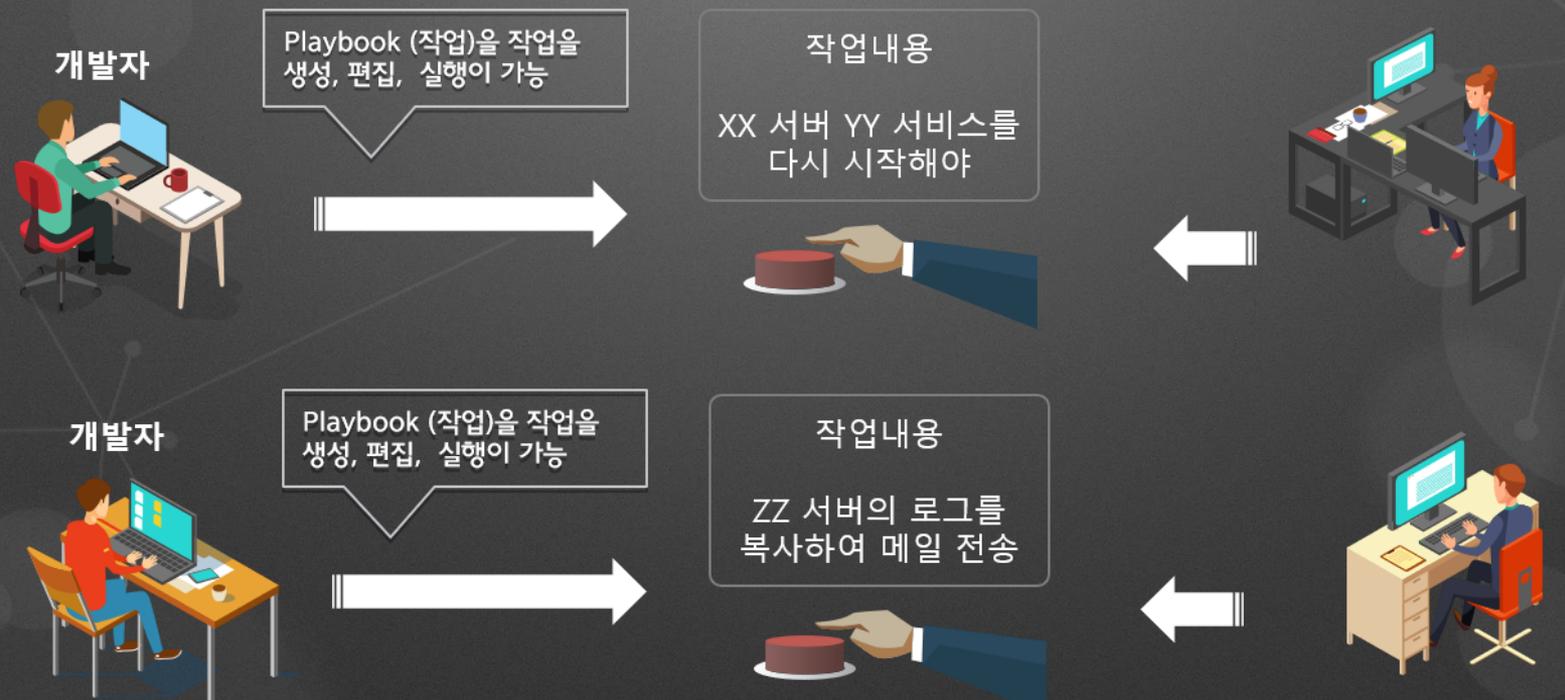
Ansible 을 통한 작업 계획서 대체

- 현재 운영 작업들을 Playbook 로 정의하여 자동화
 - Playbook 은 실행 가능한 파라미터 시트 겸 가이드



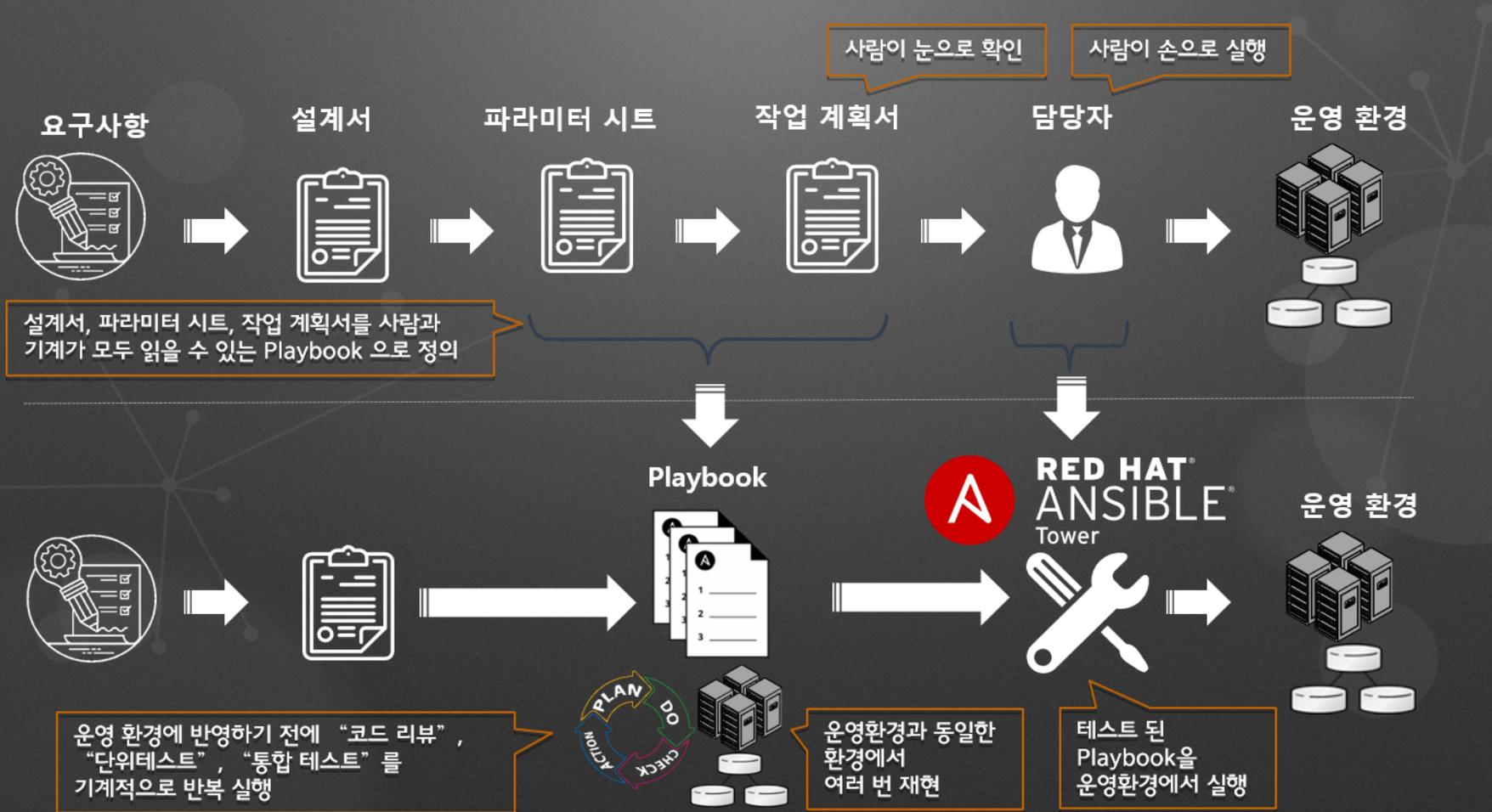
운영 작업 위임 (서비스 화)

- 개발된 Playbook 에 대한 실행 권한을 운영자에게 위임
 - 서비스를 재 시작할 수 있는 권한을 운영자에게 위임
 - 서버 로그를 메일로 전달하는 권한을 운영자에게 위임



Ansible 을 통한 전체 작업의 효율화

- Ansible Playbook으로 Code를 정의하고 검증 후 운영 환경에서 반영



애플리케이션을 위한 인프라 제공

- 클라우드 환경에서 앤서블을 통한 인프라 제공의 효과

작업	시간 (분)
가상머신 생성	30
스토리지/네트워킹 추가	30
팀 (작업) 전환 시 대기 시간	120
운영체제 설치	60
설치 후 대기	60
운영체제 구성	90
응용프로그램 설치	90
응용프로그램 구성	45
팀 (작업) 전환 시 대기 시간	120
보안 구성 및 검사	90
실제 작업 시간	5 시간 15분
전체 시간	10 시간


 앤서블 오케스트레이션

작업	시간 (분)
가상머신 생성	2
스토리지/네트워킹 추가	3
팀 (작업) 전환 시 대기 시간	120
운영체제 설치	2
설치 후 대기	60
운영체제 구성	1
응용프로그램 설치	2
응용프로그램 구성	1
팀 (작업) 전환 시 대기 시간	120
보안 구성 및 검사	2
실제 작업 시간	0 분
전체 시간	13 분

} 운영 자동화
 (인프라 프로비저닝)

} 형상 관리
 + 운영 자동화

} 구성 진단

Application Performance Management

KHAN [apm] 설치/구성/튜닝

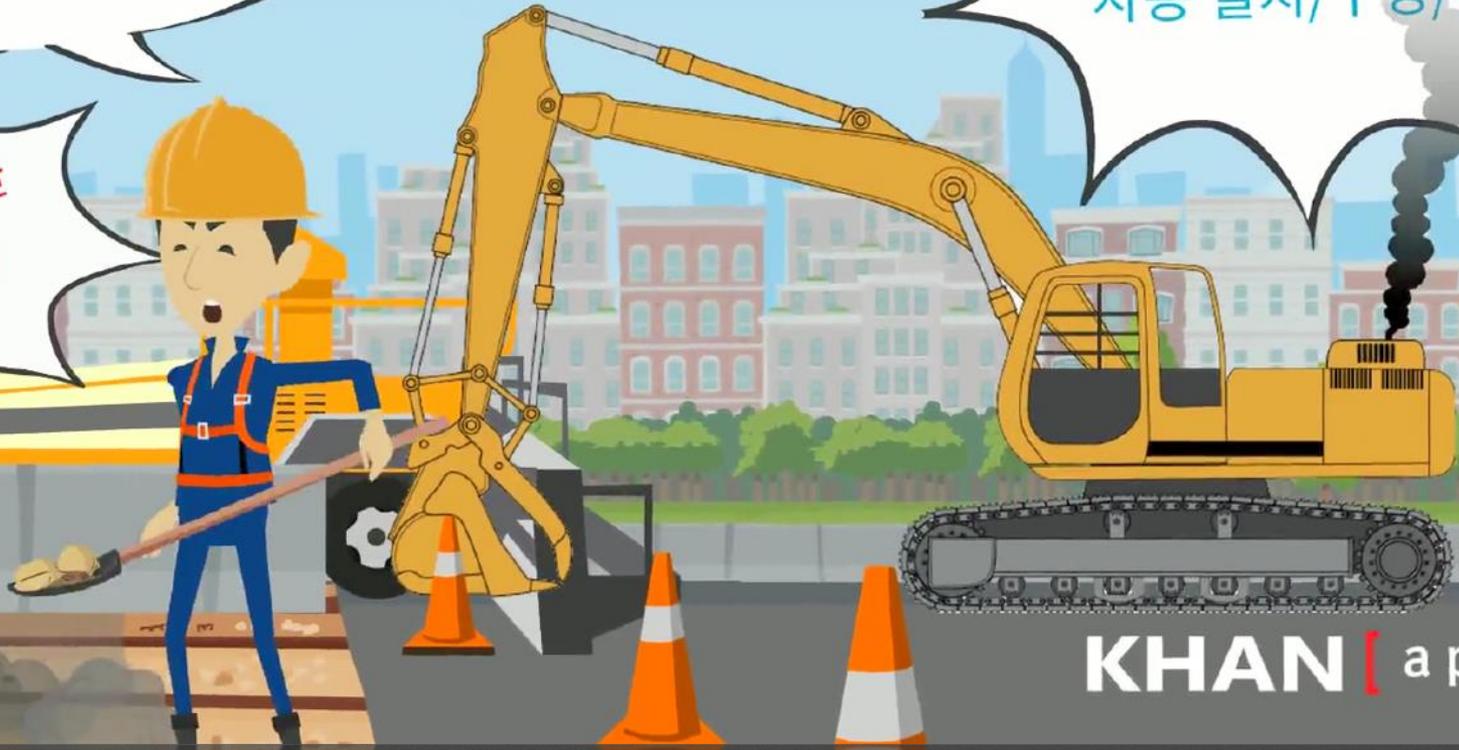
KHAN
[a p m]

KHAN [APM] - 미들웨어 설치/구성/튜닝 자동화

자꾸 바꾸라고 그러지.... 째
하루 종일 삽질만 하네....

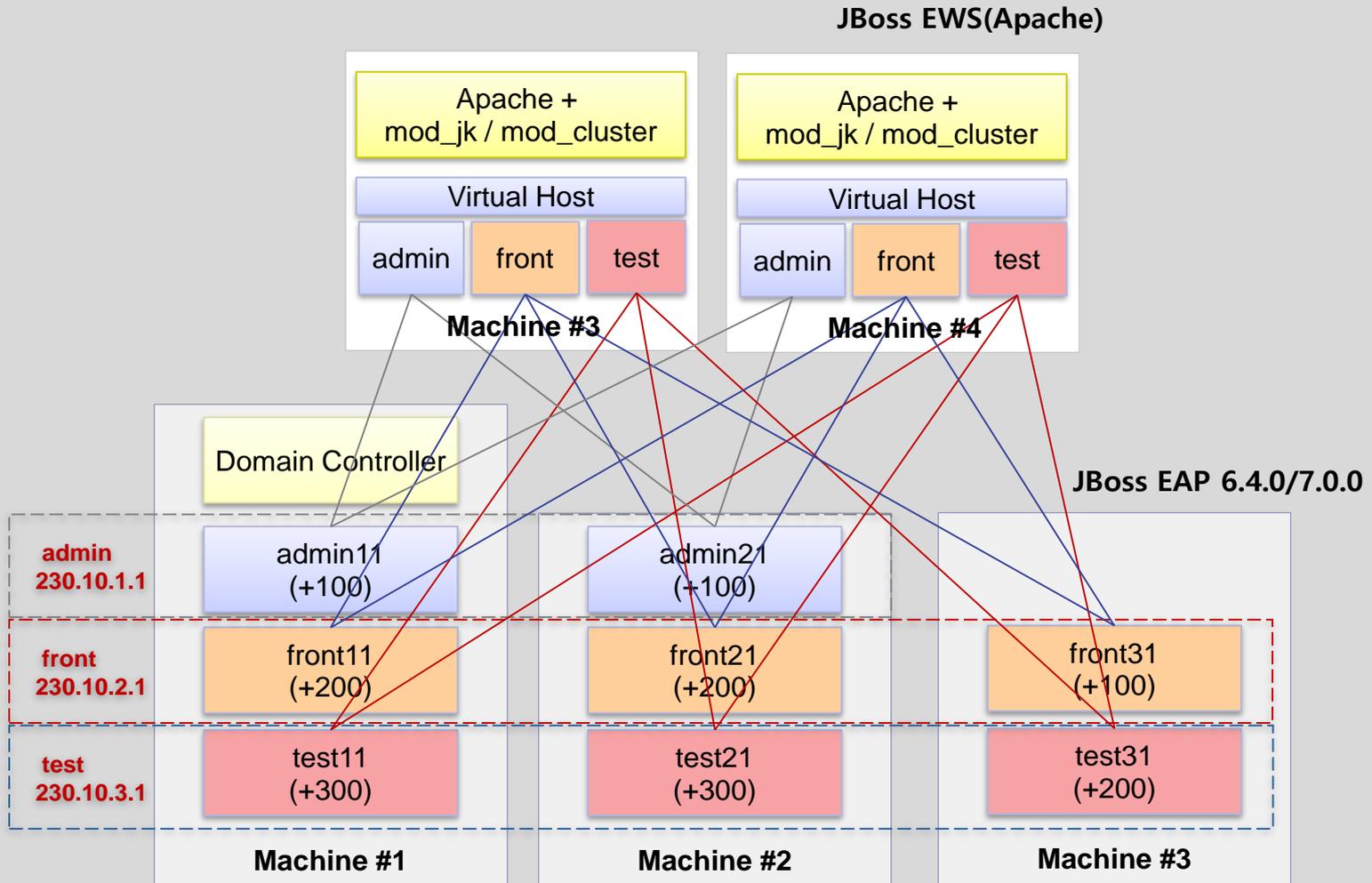
수작업에 의한
설치/구성/튜닝

KHAN [apm]에 의한
자동 설치/구성/튜닝



KHAN [a p m]

실제 Web / WAS 구성 - 다양한 서비스



KHAN [apm] – Ansible playbook 명령 정의



- JBoss EAP Domain Mode
- JBoss EAP Standalone Mode
- JBoss EWS Tomcat
- Apache HTTPD
- JBoss
- JBoss
- JBoss
- JBoss

설치 구성 화면

Filename	WAS Type	Web Type	Date	Size	Edit
hosts-apache-httpd-jboss-eap.sample	jboss-eap	apache-httpd	2014-03-27_21-02-47	14289	[Edit]
hosts-apache-httpd-tomcat.sample	apache-tomcat	apache-httpd	2014-03-27_21-02-47	9675	[Edit]
hosts-ews-httpd-jboss-eap.sample	jboss-eap	jboss-ews	2014-03-27_21-02-47	14391	[Edit]
hosts-ews2-httpd-tomcat.sample	jboss-ews2-tomcat	jboss-ews2-httpd	2014-03-27_21-02-47	9857	[Edit]
hosts-jboss-datagrid.sample	jboss-datagrid		2014-03-27_21-02-47	6040	[Edit]
test	jboss-eap	jboss-ews	2014-03-27_15-52-38	12476	[Edit]

GUI 설치구성 화면

WAS 인스턴스 구성

IP 주소	SSH 포트 Bind Address	Pem 파일
192.168.0.11	22	
인스턴스 이름	포트 요격셋	그룹명
admin11	100	admin
		스탠드얼론 프로파일
		standalone-full-ha.xml
		standalone-ha.xml

IP 주소	SSH 포트 Bind Address	Pem 파일
192.168.0.12	22	
인스턴스 이름	포트 요격셋	그룹명
admin21	100	admin
		스탠드얼론 프로파일
		standalone-full-ha.xml
		standalone-ha.xml

WAS 인스턴스 구성



KHAN
[a p m]

제품이나 서비스에 관한 문의

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

전자 메일 : sales@opennaru.com