01 왜 테라폼인가? 데브옵스란

1 데브옵스의 등장

2 코드형 인프라란?

코드형 일프라(Infrastructure as Code, IaC)란 코드를 작성 및 실행하여 인프라를 생성, 배포, 수정, 정리하는것을 말한다.

코드형 인프라 도구의 범주

- 애드혹 스크립트
- 구성 관리 도구
- 서버 템플릿 도구
- 오케스트레이션 도구
- 프로비전 도구

2.1 애드혹 스크립트

자동화하는 가장 간단한 방법은 애드혹 스크립트를 사용하는 것이다.

수행할 작업을 단계별로 나누고 배시, 루비, 파이썬 등 선호하는 언어를 사용하여 각 단계를 코드로 정의하고 작성된 스크립트를 서버에서 수동으로 실행하는 것이다.

```
# 스크립트 예제
# apt-get 캐시를 업데이트
sudo apt-get update
```

```
# PHP와 아파치 설치
sudo apt-get install -y php apache2

# 깃 리포지터리에서 코드를 다운로드
sudo git clone https://github.com/brilis98/php-app.git /var/www/html/app

# 아파치 웹 서버 시작
sudo service apache2 start
```

2.2 구성 관리 도구

셰프, 퍼핏, 앤서블, 솔트스택 등은 모두 구성관리 도구로써 대상 서버에 소프트웨어를 설치하고 관리하도록 설계되어 있다.

```
- name: Update the apt-get cache
  apt:
    update_cache: yes

- name: Install PHP
  apt:
    name: php

- name: Install Apache
  apt:
    name: apache2

- name: Copy the code from the repository
  git: repo=https://github.com/brilis98/php-app.git dest=/var/www/html/app

-name: Start Apache
  service: name=apache2 state=started enable=yes
```

코딩규칙

앤서블은 문서화, 파일 레이아웃, 명확하게 이름 붙여진 매개 변수, 시크릿 관리 등을 포함하는 일관되고 예측 가능한 구조를 제공한다.

멱등성

실행 횟수에 관계없이 올바르게 동작하는 것을 말한다.

분산형 구조

에드혹 스크립트는 단일 로컬 머신에서만 실행되도록 설계되었다. 반면 구성 관리 도구는 원격의 수많은 서버를 관리하기 위해 설계되었다.

예를 들어 5대의 서버에 web-server.yml 롤을 적용하기 위해서는 면저 5대의 서버의 IP주소가 포함된 host 파일을 생성한다.

```
[web server]
11.11.11.11
11.11.11.12
11.11.11.13
11.11.11.14
11.11.11.15

# 다음으로 앤서블 플레이북을 정의한다.
-hosts: web server
  roles:
  - webserver

# 마지막으로 플레이북을 실행한다.
ansible-playbook playbook.yml
```

2.3 서버 템플릿 도구

구성 관리 도구의 대안으로 도커(Docker), 패커(Packer), 베이크런트(Vagrant)와 같은 서버 템플릿 도구도 인기가 높아지고 있다.

여러 서버를 시작하고 각각 동일한 코드를 실행하여 서버를 구성하는 기존 방식과 다르게, 서버 템플릿 도구는 운영 체제, 소프트웨어, 파일 및 기타 필요한 모든 내용을 포함하고 있는 스냅숏으로 이미지를 생성한다.

이미지 작업을 위한 도구

가상 머신

가상 머신(Virtual Machine, VM)은 하드웨어를 포함한 전체 컴퓨터 시스템을 에뮬레이트한다.

VM웨어, 버추얼박스 또는 패러럴즈와 같은 하이퍼바이저를 사용하여 CPU, 메모리, 하드 드라이브, 네트워크를 가상화한다. 하이퍼바이저에서 실행되는 모든 Vm 이미지는 가상화된 하드웨어로만 볼 수 있다.

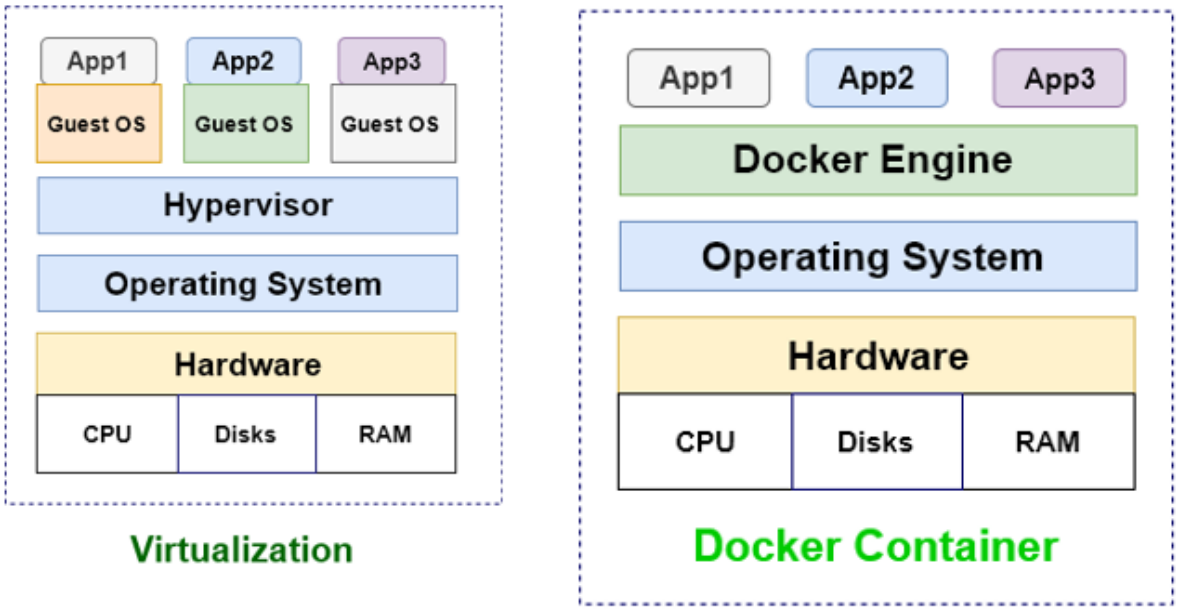
단점은 모든 하드웨어를 가상화하고 다른 VM과도 완전히 분리했기 때문에 VM마다 별도의 CPU, 메모리, 리소스가 할당되는 오버헤드가 발행한다.

컨테이너

컨테이너(Container)는 OS의 사용자 공간을 에뮬레이트한다. 도커, 코어OS의 rkt 또는 크라이오와 같은 컨테이너 엔진을 실행하여 격리된 프로세스, 메모리, 마운트 지점, 네트워킹을 만든다.

컨테이너 엔진에서 실행되는 컨테이너는 그 자체의 사용자 공간만 볼 수 있다. 즉, 호스트 시스템 및 다른 컨테이너와는 격리되어 개인 컴퓨터, QA 서버, 실제 운영 환경 등 모든 환경에서 정확히 동일하게 실행된다는 장점이 있다.

단점은 단일 서버에서 실행되는 모든 컨테이너가 해당 서버의 OS 커널과 하드웨어를 공유하므로 VM을 사용하는 것만큼의 격리 및 보안 수준을 당성하기가 훨씬 어려울 수 있다.



2.4 오케스트레이션 도구

서버 템플릿 도구는 VM이나 컨테이너를 생성하기에 더없이 좋은 도구이지만 이를 어떻게 관리하는가도 중요하다.

VM과 컨테이너를 하드웨어에 효율적으로 배포하기

롤링 배포, 블루-그린 배포, 카나리 배포 전략을 사용하여 기존의 VM이나 컨테이너를 효율적으로 업데이터하거나 롤백하기

VM과 컨테이너의 상태를 모니터링하고 비정상적인 부분을 자동으로 대체하기

발생하는 트래픽에 따라 VM과 컨테이너의 수를 늘리거나 줄이기

VM과 컨테이너의 트래픽을 분산하기

서로 다른 네트워크에 있더라도 VM과 컨테이너가 서로 식별하고 통신할 수 있게 하기

이 작업들을 처리하기 위해 쿠버네티스(Kubernetes), 마라톤/메소스)Marathon/Mesos), 아마존 엘라스틱 컨테이너 서비스(Amazon Elastic Container Service, Amazon ECS), 도커 스웜(Docker Swarm), 노마드(Nomad) 같은 오케스트레이션 도구가 필요하다.

2.5 프로비전 도구

구성 관리, 서버 템플릿 및 오케스트레이션 도구가 각 서버에서 실행되는 코드를 정의한다면 테라폼, 클라우드포메이션, 오픈스택 히트와 같은 프로비전 도구는 서버 자체를 생성한다.

프로비전 도구를 사용하면 서버만 생성하는 것이 아니라 데이터베이스, 캐시, 로드 밸런서, 큐, 모니터링, 서브넷 구성, 방화벽 설정, 라우팅 규칙 설정, SSL 인증서 등 인프라에 관한 거의 모든 부분을 프로비저닝할 수 있다.