

# 클라우드기초

최종석(jschoi@ssu.ac.kr)

---



# 03

## Container Platform 도구

01. Terraform & Ansible

02. Vault

03. Ceph



# Terraform & Ansible

## » Container Platform 오픈소스 목록



오픈소스 명	라이선스
kubernetes	Apache 2.0
kubeEdge	Apache 2.0
harbor	Apache 2.0
mariadb	GPL v2
keycloak	Apache 2.0
terraform	MLP 2.0
ingress-nginx-controller	Apache 2.0
istio	Apache 2.0
kubeflow	Apache 2.0
vault	MLP 2.0
rook	Apache 2.0



# Terraform & Ansible

## » IaC(Infrastructure as Code) 개념

### 스크립트를 통해 인프라 및 구성 관리를 자동화 하는 방법론

- ✓ 수동 프로세스가 아닌 코드를 통해 인프라를 관리하고 프로비저닝
- ✓ 정보 관리를 위해 별도의 인프라 세트가 필요하지 않음(Masterless)

### IaC 툴의 장점

- ✓ 물리적 환경에 대한 세부정보를 추상화 하여 중요한 코드에 집중 가능
- ✓ 비용절감
- ✓ 배포 속도 향상
- ✓ 오류 감소
- ✓ 인프라 일관성 향상
- ✓ 구성 드리프트 제거

## » Terraform이란?

HashiCorp에서 운영 중인 인프라를 손쉽게 구축하고 안전하게 변경하고,  
효율적으로 인프라의 형상을 관리할 수 있는 오픈 소스 도구



- ✓ 플랫폼에 구매 받지 않음
  - 다양한 클라우드 서비스들을 프로바이더 방식으로 제공(AWS, GCP, Azure 등)
- ✓ 변경 불가능 인프라
  - 환경에 대한 변경 사항이 적용될 때마다 현재 구성이 변경을 반영할 수 있는 새로운 구성으로 대체되고 인프라가 다시 프로비저닝됨

# Terraform & Ansible

## » Terraform 특징

### ✓ Infrastructure as Code

인프라를 코드로 정의하여 생산성과 투명성을 높일 수 있고, 정의한 코드를 쉽게 공유할 수 있어 효율적으로 협업할 수 있음

### ✓ Execution Plan

변경 계획과 변경 적용을 분리하여 변경 내용을 적용할 때 발생할 수 있는 실수를 줄일 수 있음

### ✓ Resource Graph

사소한 변경이 인프라 전체에 어떤 영향을 미칠지 미리 확인할 수 있으므로 종속성 그래프를 작성하여 이 그래프를 바탕으로 계획을 세우고, 이 계획을 적용했을 때 변경되는 인프라 상태를 확인할 수 있음

### ✓ Change Automation

여러 장소에 같은 구성의 인프라를 구축하고 변경할 수 있도록 자동화할 수 있어 인프라를 구축하는 데 드는 시간을 절약할 수 있고, 실수를 줄일 수 있음

## » Ansible 이란?

Red Hat에서 개발 중인 프로비저닝, 구성 관리, 애플리케이션 배포, 오케스트레이션 등 여러 수동 IT 프로세스를 자동화하는 오픈소스 IT 자동화 툴



- ✓ 플레이북에 실행할 구성을 선언해 놓으면, 필요시마다 자동 실행 가능  
즉, 웹서버의 구성과, DB서버의 구성을 선언해 놓으면 관리자들은 필요할 때마다 그 구성대로 서버의 설정을 배포할 수 있음

## 01

## Terraform &amp; Ansible

## » Terraform과 Ansible 비교



HashiCorp

Terraform



ANSIBLE

유형	오케스트레이션 도구	컨피규레이션 관리 도구
구문	HCL	YAML
언어	선언적	절차적
기본접근	가변 인프라	불변 인프라
생명주기 관리	지원	미지원
기능	프로비저닝, 컨피규어링	프로비저닝, 컨피규어링
Agentless	○	○
Masterless	○	○



# Terraform & Ansible

## » Terraform과 Ansible 비교



HashiCorp

Terraform



ANSIBLE

유형

오케스트레이션 도구

컨피규레이션 관리 도구

Terraform: "집을 설계하고 짓는 건축가"

→ AWS, Azure, GCP에서 서버, 네트워크 등 기반 구조를 세팅함.

Ansible: "집 안을 꾸미고 요리를 준비하는 셰프"

→ 클라우드(서버)에 소프트웨어 설치 및 환경 설정을 수행함.

# Terraform & Ansible

## » Terraform과 Ansible 비교



유형

오케스트레이션 도구

컨피규레이션 관리 도구

오케스트레이션 도구는 복잡한 자동화와 조정 작업을 처리하는 데 초점을 맞추는 작업을 이르는 용어

여러 서버, 컨테이너, 서비스 간의 작업을 자동화하고 조율  
주로 클라우드 환경에서의 리소스 관리, 애플리케이션 배포, 서비스 관리, 스케일링, 네트워킹 설정 등을 담당

컨피규레이션은 서버나 다른 IT 시스템의 설정과 관리를 자동화하는 데 중점을 둔 용어  
주로 서버 설정, 소프트웨어 설치, 시스템 업데이트, 보안 규칙 적용 등을 자동화하여 일관된 환경을 유지하고, 수작업으로 발생할 수 있는 오류를 최소화

# Terraform & Ansible

## » Terraform과 Ansible 비교



유형	오케스트레이션 도구	컨피규레이션 관리 도구
오케스트레이션 도구는 복잡한 자동화와 조정 작업을 처리하는 데 초점을 맞추는 작업을 이르는 용어 여러 서버, 컨테이너, 서비스 간의 작업을 자동화하고 조율 주로 클라우드 환경에서의 리소스 관리, 애플리케이션 배포, 서비스 관리, 스케일링, 네트워킹 설정 등을 담당	오케스트레이션 도구	컨피규레이션 관리 도구

오케스트레이션 예:

클라우드 리소스(예: 서버, 네트워크, 데이터베이스 등)를 생성, 수정, 삭제하는 데 사용

"AWS에 VM 5대를 만들어줘!" -> Terraform이 직접 리소스를 생성하고 구성.

# Terraform & Ansible

## » Terraform과 Ansible 비교



유형

오케스트레이션 도구

컨피규레이션 관리 도구

컨피규레이션은 서버나 다른 IT 시스템의 설정과 관리를 자동화하는 데 중점을 둔 용어  
주로 서버 설정, 소프트웨어 설치, 시스템 업데이트, 보안 규칙 적용 등을 자동화하여  
일관된 환경을 유지하고, 수작업으로 발생할 수 있는 오류를 최소화

이미 만들어진 리소스(서버 등)를 **설정, 관리, 배포**하는 데 사용  
예: "이 서버에 Python 설치하고, 방화벽 설정하고, 애플리케이션 배포해!" ->  
Ansible이 해당 작업들을 실행.

# Terraform & Ansible

## » Terraform과 Ansible 비교



HashiCorp

Terraform



ANSIBLE

유형

오케스트레이션 도구

컨피규레이션 관리 도구

**목적:** 오케스트레이션 도구는 서비스와 애플리케이션 간의 상호작용과 조율을 관리하는 반면, 컨피규레이션 도구는 서버나 시스템의 설정을 일관되게 유지하고 자동화하는 데 중점

**작업:** 오케스트레이션 도구는 보통 "시스템 전체"에 걸친 작업을 수행하며, 여러 컴포넌트와 서비스를 통합  
→ 인프라를 만드는 데 초점

컨피규레이션 도구는 "단일 서버"나 "단일 시스템" 수준에서 더 세밀한 설정을 제공

→ 만들어진 인프라를 설정하고 관리하는 데 초점

# Terraform & Ansible

## » Terraform과 Ansible 비교



선언적(원하는 상태 설명)	절차적(어떻게 도달할지 설명)
IaC를 작성하는데 사용	YAML 구문을 사용하여 대상 인프라에서 수행할 절차를 정의함
선언적 형식인 HCL을 사용하고 코드가 작성되는 순서는 중요하지 않음	Ansible YAML 스크립트는 <b>절차적 언어</b> 로 스크립트를 작성할 때 아래로 실행
코드는 여러 파일에 분산될 수도 있음	앤서블 스크립트는 “Ansible Playbook”에 저장
코드를 어떻게 작성하든 Terraform은 <b>종속성을 식별</b> 하고 인프라를 프로비저닝	특정 일련의 작업을 수행해야 하는 경우 플레이북에서 동일한 작업을 정의하고, 작업은 작성된 <b>순서대로</b> 수행
기존 인프라를 작성하거나 <b>코드로 쉽게 변환</b>	루트 사용자로 지정된 가상머신에 Apache 서버를 설치하려면 <b>설치 작업을 정의하기 전에 사용자 생성 단계를 작성</b> 해야 함

# Terraform & Ansible

## » Terraform과 Ansible 비교



변경 가능	변경 불가
원하는 인프라 상태를 입력으로 가져와 프로비저닝	기본적으로 변경 불가능
선언적 형식인 HCL을 사용하고 코드가 작성되는 순서는 중요하지 않음	최신 버전의 플레이북에 따라 구성 변경 사항을 일관되게 유지하고 관리
클라우드 공급자가 인프라를 재부팅 하거나 교체하지 않고 변경 사항을 구현할 수 없는 경우 인프라를 변경	변경 사항은 기본 인프라의 “교체”에 영향을 주지 않으며, 지정된 구성 요소의 구성만 복구하거나 수정
이전 인프라 구성요소를 제거하고 최신 구성 설정이 있는 새 구성 요소로 교체	Ansible에서 관리하는 서버는 수행한 구성의 변경 사항을 기록

# Terraform & Ansible

## » Terraform과 Ansible 비교



우수한 스케줄링 기능과 함께  
제공되며 사용자에게  
매우 쾌적한 환경을 제공

Docker와 통합 우수

대상 장치가  
어떻게 최종 상태가  
되었는지에 대한  
명확한 증거가 없음



보다 안전한 보안과  
ACL기능 제공

전통적인 자동화 프레임워크와  
함께 편안하게 조정되기 때문에  
숙성한 도구

논리적 증속성,  
오케스트레이션 서비스 및 상호  
연결된 응용 프로그램과 같은  
서비스에는 부적합



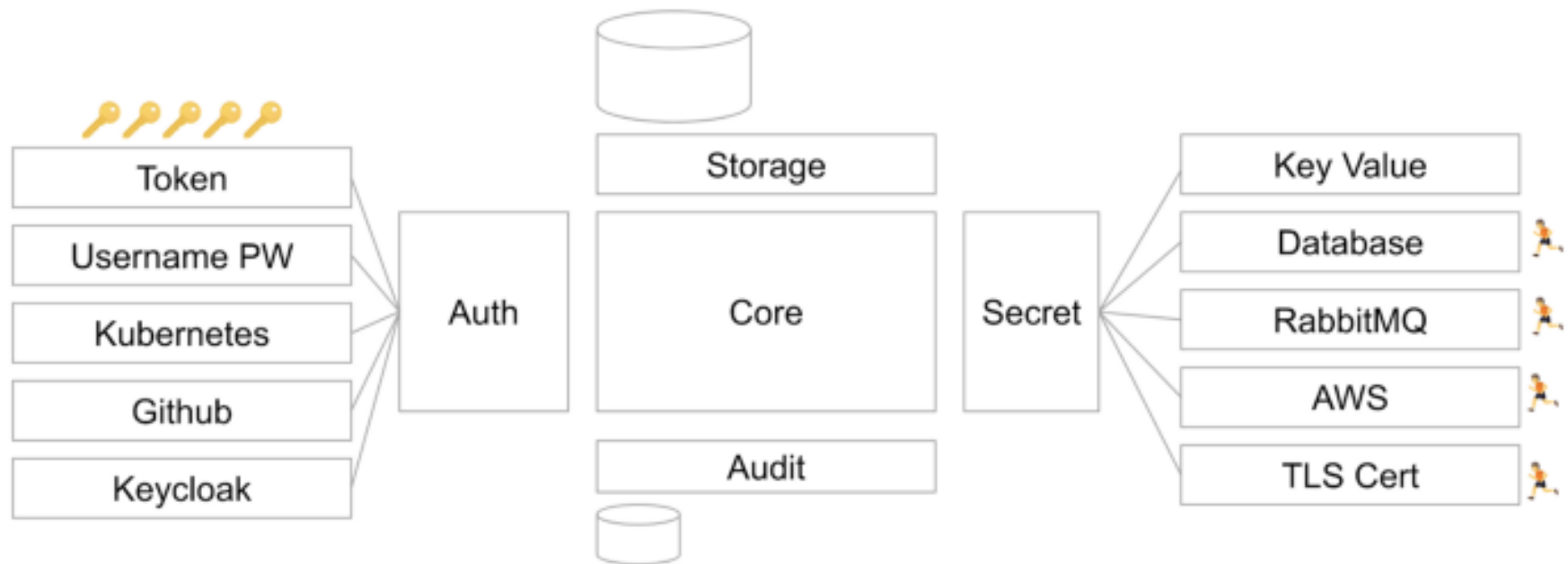
## » Vault 란?

HashiCorp에서 제공하는 크로스플랫폼 패스워드 및 인증 관리 시스템



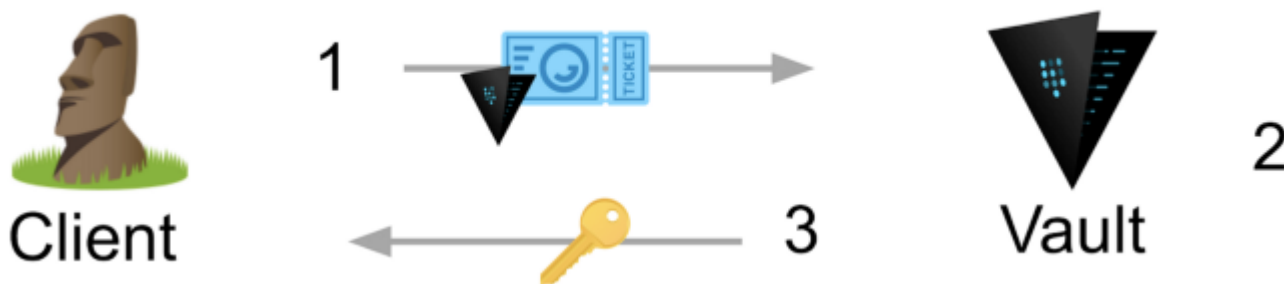
- ✓ UI, CLI, HTTP API 등의 인터페이스를 제공하고 있으며, 저장된 비밀 정보를 안전하게 사용할 수 있는 방법들을 제공
- ✓ Vault의 storage backend는 암호화된 데이터를 저장하기 위한 스토리지를 담당하며, 스토리지의 종류, 가용성 등을 책임지지 않음

## » Vault 구성요소 체계



Vault 구성 요소 체계

## » Vault 구조



1. 클라이언트는 Vault에 토큰과 시크릿 경로를 보냄
2. Vault는 클라이언트가 원하는 데이터(시크릿 정보)를 가지고 있음
3. Vault는 클라이언트의 토큰 권한을 확인, 허용되면 요청된 시크릿 정보를 반환

## » Ceph 란?

오픈소스 소프트웨어(Software Defined Storage) 스토리지 플랫폼



- ✓ 단일 분산 컴퓨터 클러스터에 object 스토리지를 구현하고 object, block 및 file level의 스토리지 기능을 제공
- ✓ Single point of failure이 없는 완전히 분산된 운영을 주소 목표로 하며 엑사바이트 수준으로 scale-out 가능

## » Ceph 특징

## 소프트웨어 정의 스토리지 (Software Defined Storage) 이점

- ✓ **비용 대비 성능 절충** : 애플리케이션이 동일한 소프트웨어 스택을 사용하여 서로 다른 하드웨어 및 내구성 구성의 성능 및 비용 절충을 선택할 수 있음
- ✓ **유연한 인터페이스** : 산업 표준 API를 선택하거나, 애플리케이션에 클라이언트 라이브러리를 내장하거나, 필요한 경우 독점 API 사용 가능
- ✓ **다양한 스토리지 구현** : object, block 및 file 추상화 전반에 걸쳐 동일한 스토리지 소프트웨어 스택을 활용하여 R&D 및 운영 비용을 절감

## » Ceph 구조



CEPH\_378927\_1017

- ✓ **Ceph OSD daemon**  
Ceph 클라이언트를 대신하여 데이터를 저장하며, 데이터 관리 기능을 수행
- ✓ **Ceph Monitor**  
Ceph 스토리지 클러스터의 현재 상태에 대한 Ceph 스토리지 클러스터 맵의 마스터 복사본을 유지
- ✓ **Ceph Manager**  
Ceph Monitor 대신 placement groups(PG), 프로세스 메타데이터 및 호스트 메타데이터에 대한 자세한 정보를 유지하여 규모에 맞게 성능을 크게 향상시킴
- ✓ **MDS(Metadata Servers)**  
클라이언트에 의한 효율적인 POSIX 명령 실행을 위해 CephFS 에서 사용하는 메타데이터를 저장