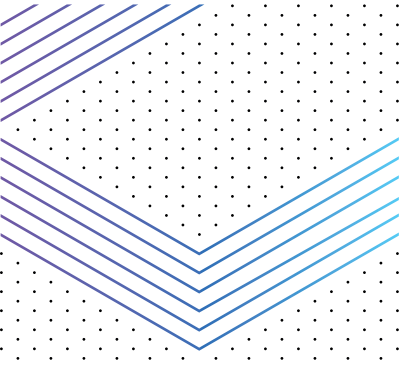


모든 애플리케이션과 인프라를 지원하는

# 멀티/하이브리드 클라우드 운영 모델








하시코프는 모든비즈니스를 위한  
 종속되지 않은 클라우드 운영 모델을 적용하고  
 디지털 혁신 전략이 성공할 수 있도록 지원합니다.



온프레미스  
 멀티/하이브리드  
 클라우드 통합 운영 아키텍처

## | OVERVIEW

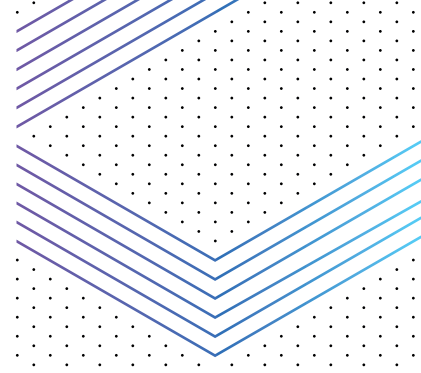
				
2012년 설립	2,000+ 임직원	2,100+ 엔터프라이즈 고객	100M 연간 다운로드	19개 국가

## | CUSTOMER





# 지금까지의 IT 프로젝트의 고민과 해결



- ▶▶ **빠르게** 프로젝트를 진행할 수 있나?
- 🔒 **보안과 정책 위반**이 없는가?
- 🔄 **비용 효율적인 방법**인가?

현재 IT의 문제점	Challenge	Issue
	서비스를 위한 서버자원 수동 설치/변경 서버자원이 필요할 때 마다 인프라 담당자가 필요 스펙을 전달 받아 그때그때 수동으로 생성	작업속도와 품질저하 / 작업이력과 노하우에 대한 관리가 안됨
	작업을 위한 접속 비밀정보 공유(엑셀/메모장) 서버접속 /애플리케이션 배포/연결을 위한 액세스 키와 P/W등을 메모장/암호 대장 등으로 관리&공유	코드상에 있는 P/W, 클라우드 접속정보 노출 및 외부 액세스 허용
	개발된 애플리케이션 수동 배포 인프라 상태확인 및 서버 접속을 통해 매번 수동으로 설치 해야함	매번 자원상태 파악 및 장애 대응을 위해 대기 / 수동작업을 통한 시간소요
	배포 서비스 연결확인 수백개의 서비스를 IP기반으로 연결하고 추적/관리 중	서비스가 늘어날 수록, 연결관리와 상태파악이 힘들어 짐
	이기종 클라우드 관리 인프라마다 작업방법이 다름	클라우드마다 적합한 작업 담당자와 아키텍처를 고민해야 함

## HashiCorp의 해결 방안

**Web**  
OS VM  
aws

**WAS**  
OS VM  
[Icons]

**DB**  
[Icons]

**Terraform** **Packer**

클라우드 리소스의 생성/관리/삭제를 코드로 자동화  
클라우드 종속 없는 일관된 운영 방법론 적용

APP, ID&PW, X, [Icons]

인프라 운영자 SM/MSP (운영)

**Vault**

Access Key/Password 등  
모든 접속정보(Credential)의 중앙화 & 자동화

SI (개발) → Build → 애플리케이션

[Code Icon] → [Build Arrow] → [APP Icon]

**Nomad**

개발된 애플리케이션 자동 배포

[Network Icon]

네트워크

**Consul**

배포 서비스 연결 확인  
서비스 디스커버리와 서비스메시

HashiCorp, KOREA

Contact-kr@hashicorp.com

3

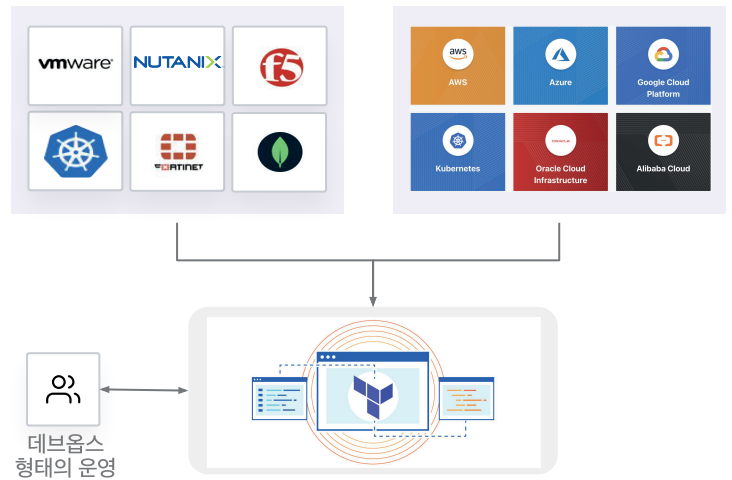
# 멀티/하이브리드 운영 노하우 자산화

프로비저닝, 컴플라이언스 준수, 관리를 위한  
IaC(Infrastructure as Code) 활용

## 멀티/하이브리드 프로비저닝

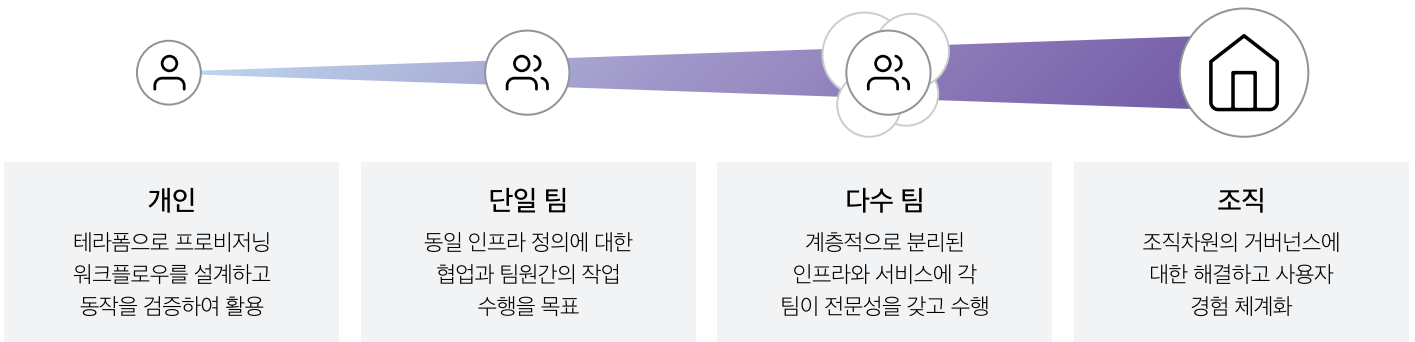
클라우드 환경의 도입은 기존 정적 인프라에서 탈피해 동적 인프라를 프로비저닝하고 관리하는 방식으로 전환이 필요합니다. 특히 다수의 클라우드 제공서비스와 기존 온프레미스 환경을 불변성(Immutability)과 일시성(Ephemerality)을 수용하고 여러 환경에 걸쳐 배포가 가능해야 한다는 것을 의미 합니다.

- 단일 관리시스템으로 쉽게 모든 클라우드 및 레가시 인프라스트럭처 관리 및 배포
- 모든 작업을 표준화 및 자동화하여, 실수 없는 신속한 배포 운영 환경 구축



## 운영 노하우 자산화와 DevOps 실현

인프라 환경을 프로비저닝하기 위해 작업자는 설계, 구축, 운영의 노하우를 코드화 하여 재사용 가능한 모듈 레지스트리를 구성합니다. IaC 도구에 익숙하지 않은 사용자도 검증된 템플릿에 의해 쉽고 빠르게 인프라를 구축하며 최고 수준의 인프라의 질을 보장할 수 있습니다.



## 솔루션 이점

- **협업 가능한 IaC Infrastructure as Code로 팀 생산성 향상**  
협업 기능을 지원하는 인프라 자동화 기능을 통해 IT 운영팀은 IaC를 안전하게 구축, 변경, 공유할 수 있습니다.
- **클라우드 컴플라이언스와 관리를 통한 위험요소 제거 및 클라우드 비용 절감**  
기업 정책을 코드화된 방식으로 구현할 수 있는 인프라 자동화를 통해 보안, 컴플라이언스, 운영 노하우를 적용할 수 있습니다.
- **셀프서비스형 인프라 On-Demand를 통한 개발 민첩성 향상**  
승인된 IaC 모듈 라이브러리를 통해 개발자들은 선호하는 워크플로우(VCS, CI/CD, ITSM)에서 자체 인프라를 프로비저닝 할 수 있습니다.

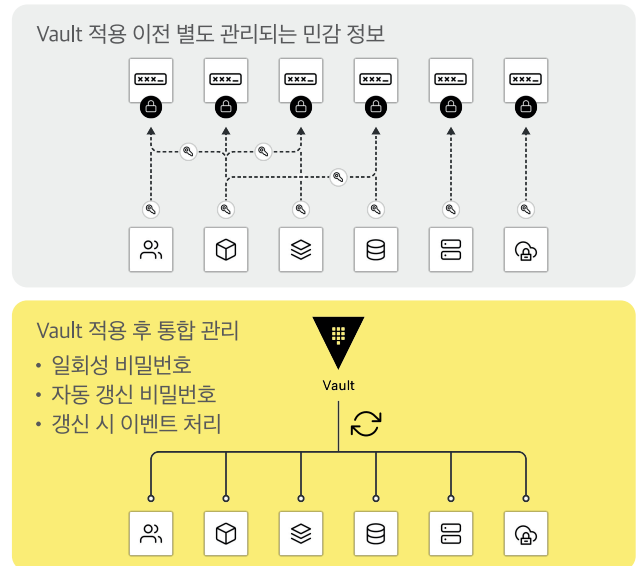
# 시크릿 중앙관리와 안전한 자동갱신

인프라 환경 전반에 걸친 안전한 접근과 민감한 데이터 보호로 시크릿 정보의 중앙 관리와 자동화된 갱신

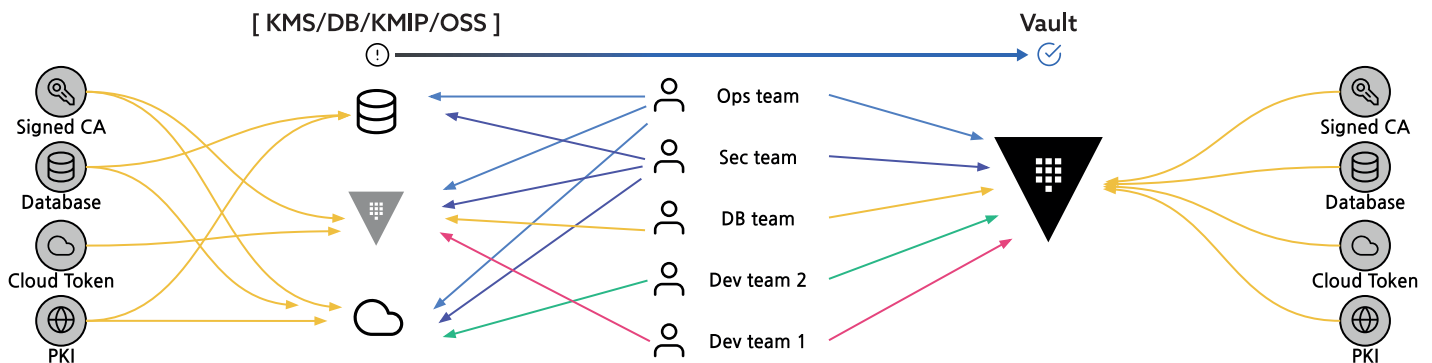
## 다양한 민감 정보 관리

인프라 환경 전반에서 발생하는 흩어져 있는 기밀 정보와 데이터를 안전하게 보호할 수 있는 온프레미스와 클라우드에 대한 보안 기반을 제공합니다.

- SSH 암호, 클라우드 접속 정보, 데이터 베이스의 접속 정보 등을 최대한 짧게 유지하여 더이상 관리자 및 작업자의 로컬 환경에 존재하지 않도록 합니다.
- 전반적인 시스템과 애플리케이션에 연동되어 패스워드와 인증서 같은 민감 정보를 노출하지 않고 사용 가능하고 정책에 따라 교체하고 갱신합니다.



## 일관된 중앙관리로 조직 전반에 제공



- 사용 목적과 관리를 위해 클러스터 분리가 필요 없으므로 민감 정보에 대한 단일 관리 포인트를 제공합니다.
- 다중 클러스터의 HA 구성 및 DR 구성으로 안정적인 시크릿 서비스를 제공합니다.
- 보안 작업을 위해 전체 시스템에 적용되어야 하는 작업의 효율성이 높아지고 Dev, Sec, Ops 팀 각각을 위한 단일 접근 포인트가 제공됩니다.

## 솔루션 이점

- 데이터 유출 위험 감소**  
 민감한 데이터를 단일 워크플로우 및 API를 통해 중앙에서 관리하고, 암호화 키를 이용하여 전송 중 저장된 민감한 데이터를 보호합니다.
- 보안 침해 위험 감소**  
 보안을 중앙 집중화 하여 하드코딩된 정적 인증 정보를 제거합니다. 또한 신뢰할 수 있는 인증 기반으로 철저하게 사용자와 애플리케이션의 접근을 제어 합니다.
- 생산성 증가**  
 운영팀과 개발팀이 인프라와 애플리케이션 전반에서 빌드와 배포 프로세스 상에 자동으로 보안을 적용하고, 단일 워크플로우를 통해 민감한 데이터를 노출 없이 사용합니다.

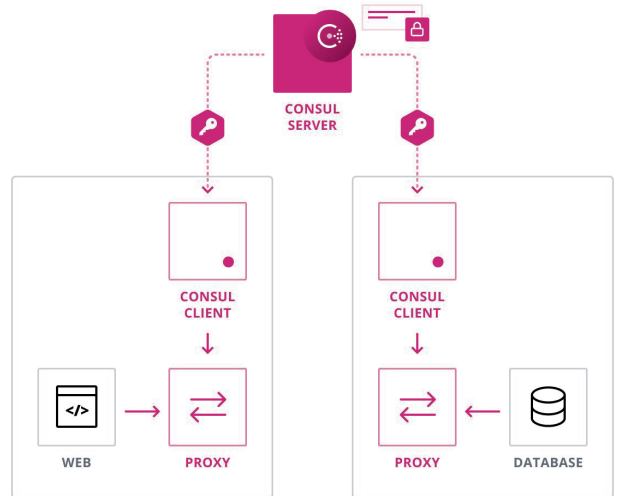
# Zero Trust 와 MSA 기반 구성을 한번에

멀티/하이브리드 환경의 안전한 서비스 연결과  
 레거시/마이크로 서비스 아키텍처 모두를 지원하는 동적인 네트워킹

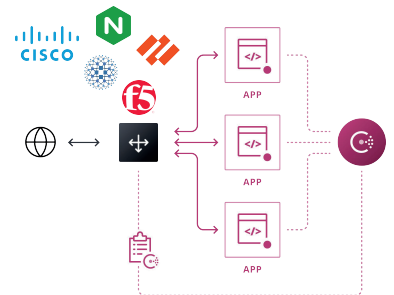
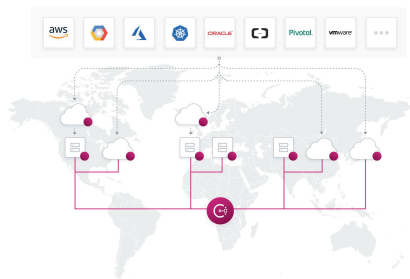
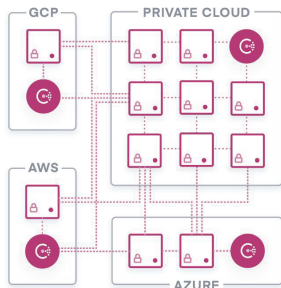
## Zero Trust 환경 대응

기존 인프라 환경과 동적인 서비스의 증가와 클라우드 환경을 고려하여 더 이상 기존 통제 방식이 통용되지 않는, 신뢰할 수 없는 환경에 놓인 우리의 상황을 “Zero Trust” 라 이야기합니다. 이런 환경 속에서도 우리가 소유한 서비스와 인프라 들은 의도한대로 안전하게 연결하고 관리되어야 합니다.

- 서비스 ID 기반 네트워킹 지원
- 양방향 인증 지원
- 서비스 연결의 인가
- 짧은 인증서 서명과 자동 갱신
- 항상 암호화된 통신 지원
- 서비스 등록과 해지의 감사와 로깅



## 멀티/하이브리드 클라우드 통합 MSA 환경 지원



### 동적으로 증가/감소하는 서비스의 자동 추적

레거시와 클라우드 서비스 전반에 걸쳐 모든 애플리케이션의 실시간 상태와 IP:Port를 확인하여 개발 편의성 증가

### 다중 플랫폼 멀티/하이브리드 클라우드

mTLS를 사용하여 프라이빗-클라우드의 서비스 간 트래픽을 보호하고 MSA 형태의 애플리케이션을 지원

### 인프라/네트워크/구성정보 자동 갱신 및 변경

동적인 서비스의 연결 설정 관련 작업을 자동화하여 애플리케이션 제공을 가속화

## 솔루션 이점

- 서비스 레지스트리와 탐색 기능을 통한 복원력 향상**  
 클라우드 팀과 애플리케이션 팀은 실행 중인 애플리케이션과 서비스에 대한 가시성을 통해 장애 발생 시 정상적인 서비스 트래픽의 경로를 재지정하여 기존 대비 고가용성을 보장 합니다.
- 네트워크 미들웨어 자동화를 통한 생산성 증대**  
 네트워크 미들웨어에 서비스 변경 사항을 적용하고 네트워크 작업을 자동화 합니다. 개발자들은 새로운 서비스의 교체, 확장, 축소의 동작을 수행합니다.
- 서비스 메시지를 통한 제로 트러스트 네트워킹 구현으로 위험제거와 비용감소**  
 서비스 ID 기반의 보안 정책과 기본 적용되는 양방향 인증으로 서비스 간 트래픽을 보호합니다. 관리해야 하는 보안 규칙이 100배 가량 줄어들고 손쉽게 일관된 정책을 적용합니다.

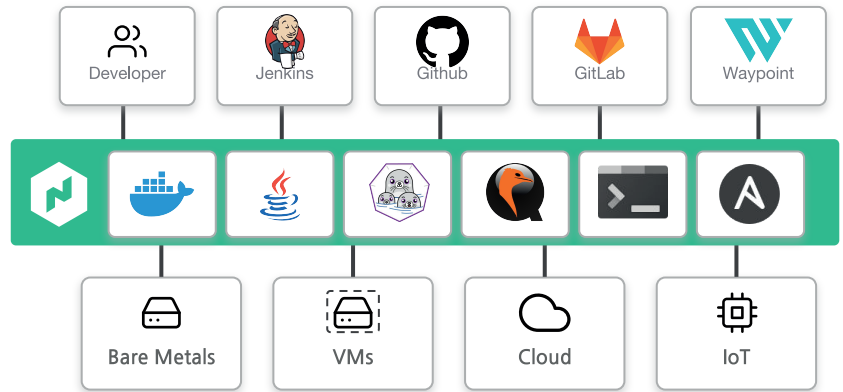
# 간단한 고성능 오케스트레이터

멀티/하이브리드 환경에서 컨테이너와 기존 애플리케이션 모두에 자동 복구와 확장, 그리고 자원 최적화로 비용 절약

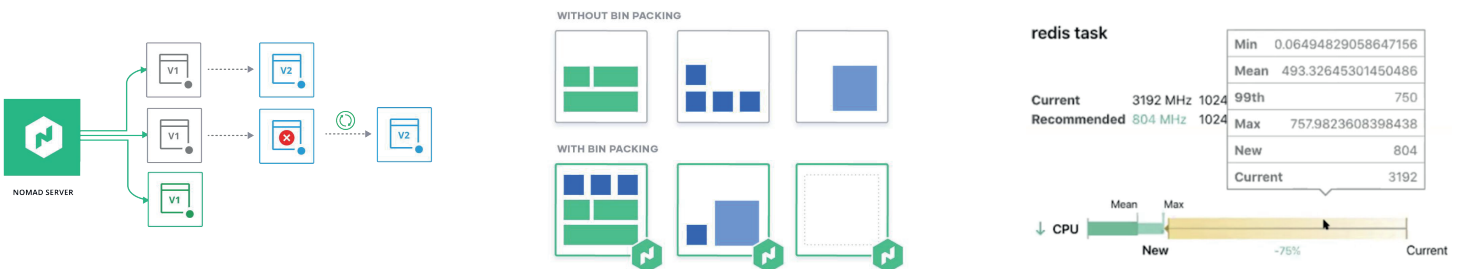
## 통합 배포 환경

온프레미스와 클라우드 전반에 컨테이너, 기존 애플리케이션 모두를 배포하고 관리하는 간단하고 유연한 워크로드 오케스트레이터입니다. 운영팀과 개발팀 모두에게 쉽고 간단한 DevOps 환경을 제공합니다.

- 컨테이너 전용 스케줄러
- Java 전용 지원
- 배치잡 및 스케줄링
- 윈도우 환경 지원
- AI/ML 워크로드
- 서버 전반에 공통 스크립트 수행



## 서비스 효율성 향상 및 인프라 리소스 최적화



### 자동 복구 / 확장

무중단 배포전략을 제공합니다. 서비스 배포 및 오류시 자동 복구 및 자동 확장 기능을 제공하여 고가용성과 고성능을 유지합니다.

### Bin-Packing

인프라 환경에 대한 리소스 활용도를 높여 호스트의 리소스 낭비를 최소화 하고 클라우드에서의 서버 자원을 최소화 하여 비용을 최적화 합니다.

### 리소스 최적화

예측하기 어려운 리소스 소비 수준에 맞게 애플리케이션 자원 크기를 조정하여 지출을 줄이고 규모에 맞게 지능적으로 비용을 절감합니다.

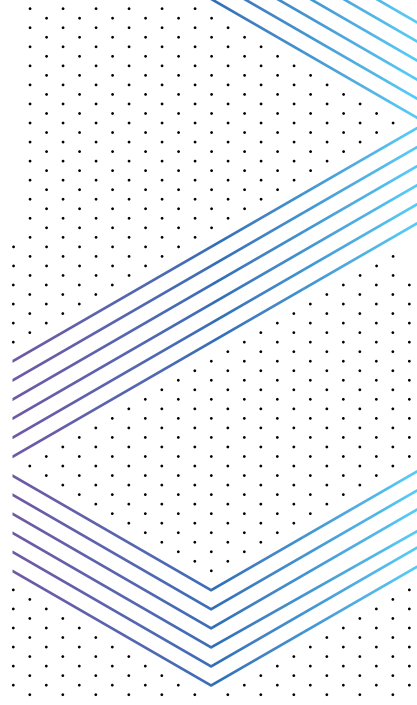
## 솔루션 이점

- 단순한 컨테이너 오케스트레이션으로 빠른 가치 실현 및 운영 오버헤드 절감

Nomad는 적은 리소스를 사용하는 가벼운 단일 프로세스로 실행되어 온프레미스 환경과 모든 클라우드에 쉽게 배포될 수 있으며, 운영 담당자와 개발자 모두에게 가장 단순한 접근성을 제공하여 기업의 가치 실현 시간을 단축합니다. 단순한 아키텍처로 운영 담당자와 개발자 모두에게 익숙한 사용자 경험을 제공하여 모든 규모에서 컨테이너를 손쉽게 구축, 확장, 관리 가능합니다.

- 컨테이너화 되지 않은 애플리케이션을 포함한 오케스트레이션으로 최소한의 변경으로 실행 환경 현대화

컨테이너뿐만 아니라 Windows, Java, VM, Docker 등 다양한 워크로드를 지원합니다. 기존 워크로드 및 CI/CD 파이프라인에 연결할 수 있으며, 리소스 가용성, 자동 복구, 자동 확장 애플리케이션 업그레이드 전략을 제공하여 아키텍처와 애플리케이션을 앵고도 오케스트레이션의 이점을 제공합니다.



[www.hashicorp.com](http://www.hashicorp.com)  
[contact-kr@hashicorp.com](mailto:contact-kr@hashicorp.com)