

게임 서버 오케스트레이션

Windows / Linux 를 모두 지원하는
범용 애플리케이션 실행 오케스트레이터

- 
1. Nomad 개요
 2. 고객 사례

01

Nomad 개요



애플리케이션 다운타임을
최소화 하고 싶으십니까?

컨테이너로의 전환을
고민하고 계십니까?

서비스 부하에 따른 동적인 확장과
축소 기능이 필요한가요?

쿠버네티스... 변화에 대응할
사람은 준비되어있나요?

Dev와 Ops 모두에게 도움이되는
배포관리 툴을 검토하고 있습니까?

하이브리드, IoT, 멀티 클라우드, 거기에
레거시까지 관리해야 합니까?



—

컨테이너, 레거시, 배치 서비스를 배포와 관리

유연하고, 가볍고, 탁월한 성능에
쉽기까지한 오케스트레이터

리소스를 최적화하고 배포를
자동화하여 인프라 운영 비용 절감



기존 실행 노하우

Dev와 Ops는 기존 관리하던 방식으로
자동화만을 더하여 학습 부담을 줄입니다.

컨테이너 전환 체계 수립

레거시와 컨테이너 모두를 단일
플랫폼/단일 방식으로 실행하고
관리합니다.

소규모부터 대규모까지 지원

Kubernetes는 대규모 서비스를 가정한
플랫폼으로 조직 구성원과 인프라에
급격한 변화를 야기합니다.

Cloud-Native 형태의 전환을 쉽고
간단하게 시작하고, 축적된 노하우를 향후
Kubernetes 환경까지 이어갑니다.

Nomad 활용 범위



간단한 컨테이너 오케스트레이터

프로덕션 환경에서 엔터프라이즈
컨테이너를 쉽게 배포, 관리 및
확장할 수 있습니다.



비-컨테이너 앱을 컨테이너 처럼 오케스트레이션

재작성 없이 컨테이너화되지
않은 애플리케이션을
현대화합니다.



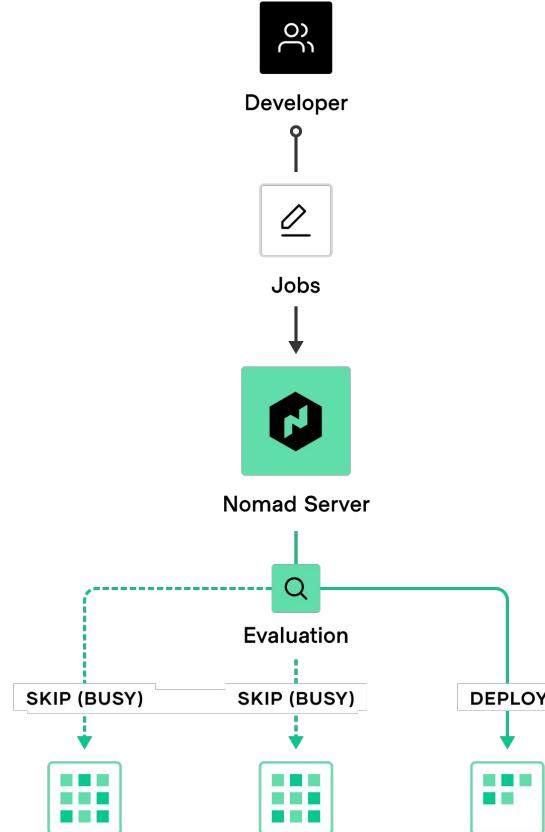
배치 워크로드 지원과 클러스터 관리

Nomad는 배치 서비스를 실행하고
ML, AI, 데이터 사이언스 같은 성능
집약적 워크로드를 지원하는 GPU
지원을 제공합니다.



자동화된 배포 워크플로우

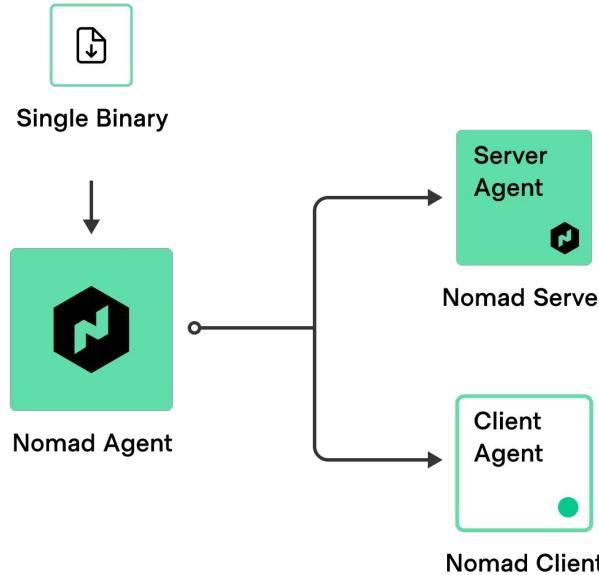
- 빠른 검증과 운영에 대한 배포 전환으로 Time to Market 과 Time to Value를 실현합니다.
- Nomad에 애플리케이션을 배포하기 위해 개발자를 온보딩하는데 1~2시간이면 충분합니다.
- 개발자와 운영자의 비율을 100:1 수준으로 운영 오버헤드를 줄입니다.





가볍고 쉬운 하나의 바이너리 파일로 구성

실행과 CLI 역할을 모두 수행합니다. (Windows/Linux/Mac/BSD)



Nomad 서버는 스케줄링을 위한 컨트롤
플레인을 구성

\$.nomad -config=server.hcl

Nomad 클라이언트는 서버에 등록된
노드에서 실행되어 할당될 작업을 감시하고
작업을 실행

\$.nomad -config=client.hcl



직관적인 배포 규칙

job

애플리케이션 배포 규칙을 정의

_ group

동일한 Nomad 클라이언트에 함께 배치되어야 하는 Task를 정의

_ task

커맨드, 서비스, 애플리케이션, 컨테이너같은 실행에 대한 정의

The screenshot shows a terminal window titled "TERMINAL". The code displayed is a Nomad job definition:

```
job "my_job" {
    region = "us"
    datacenters = ["us-west-1", "us-east-1"]

    type = "service"

    group "web" {
        count = 5
    }

    task "frontend" {
        driver = "docker"

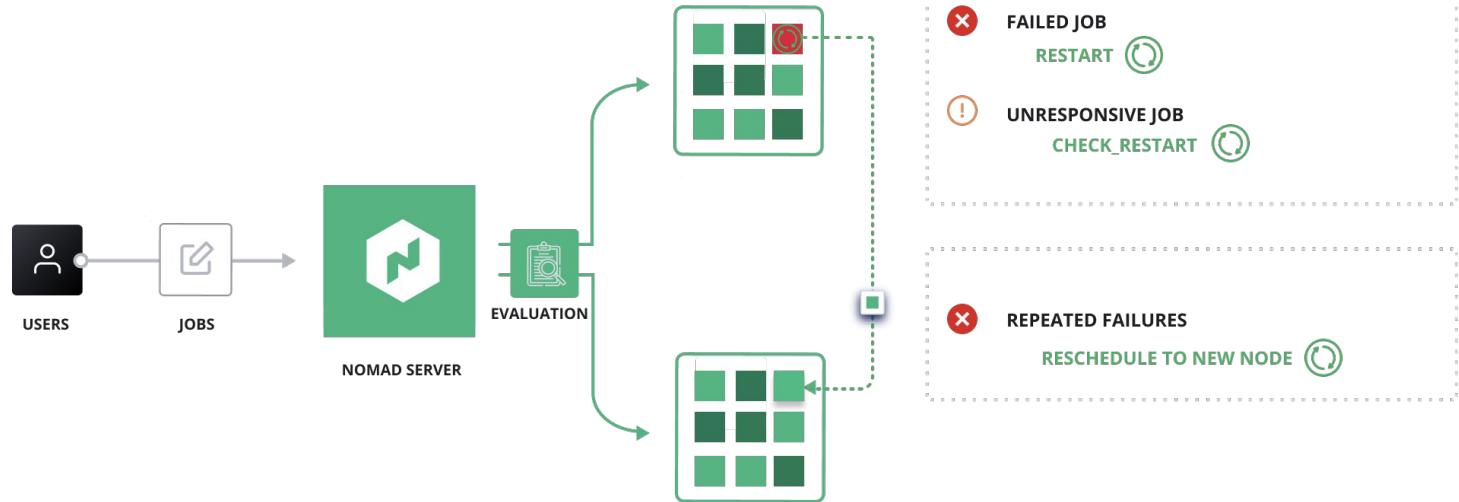
        config {
            image = "hashicorp/web-frontend"
        }

        resources {
            cpu      = 500 # MHz
            memory  = 128 # MB
            network {
                mbits = 100
                port "http" {}
                port "https" {
                    static = 443
                }
            }
        }
    }
}
```

애플리케이션 복원력 Resiliency



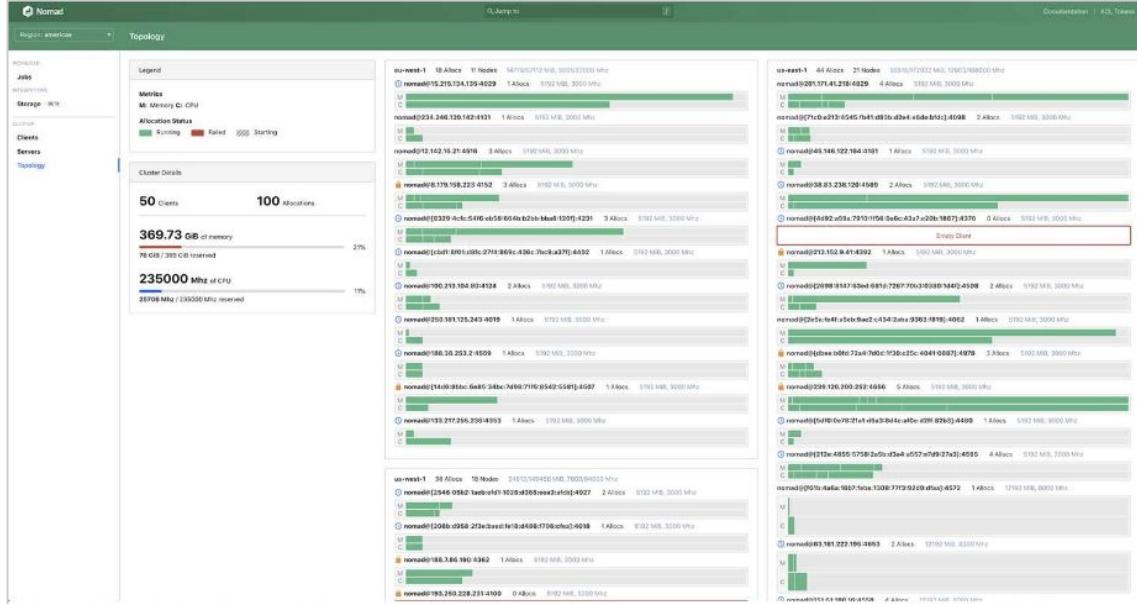
- **Restart** : Nomad Job에서 실패하고 응답하지 않는 Task 동작을 자동으로 다시시작합니다.
- **Reschedule** : 특정 Nomad 클라이언트에서 반복적으로 실패한 실행은 다른 클라이언트로 자동 재예약하는 전략을 수행합니다.





Topology

모든 데이터 센터, 노드 및 할당 정보와 해당 리소스 사용률을 확인합니다. 사용자는 이를 활용하여 클러스터 용량을 직관적으로 이해하고, 애플리케이션 배포가 노드에 분산되는 방식을 관찰하고, 노드 장애시 폭발 반경을 낮추기 위해 최적이 아닌 할당 배치를 사전에 식별 할 수 있습니다.

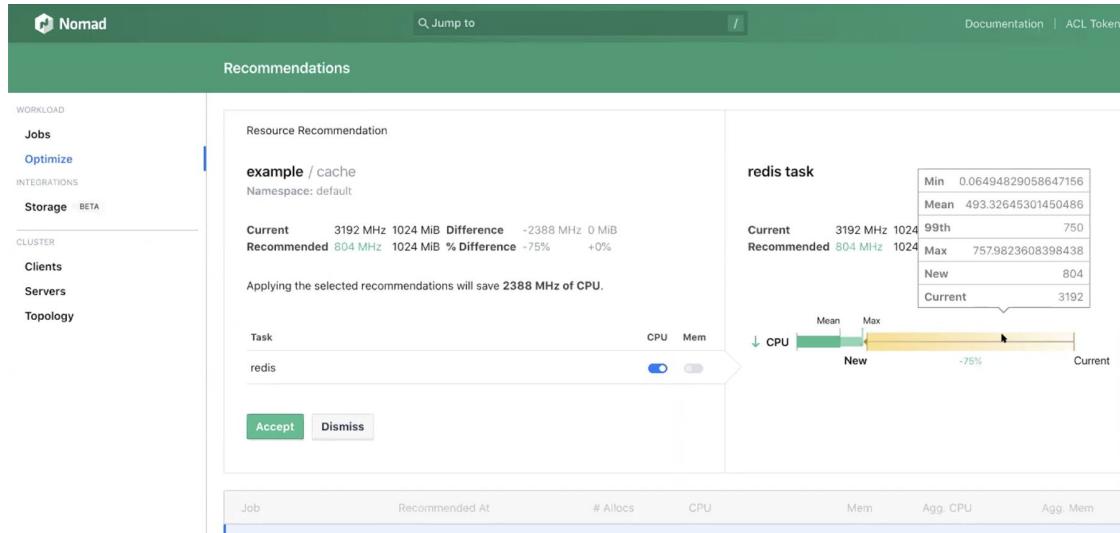




Dynamic Application Sizing

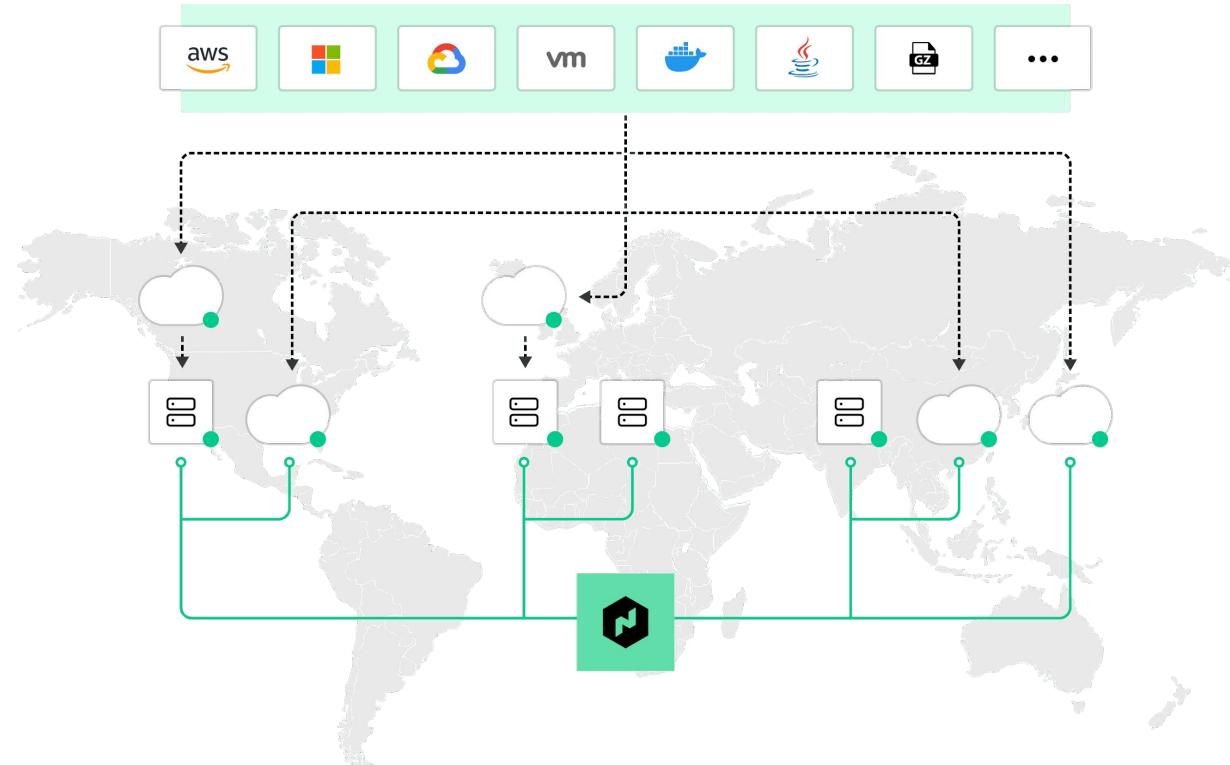
Nomad가 가장 효율적인 리소스 소비 수준에 맞게 애플리케이션 자원 크기를 조정합니다.

지출을 줄이고 규모에 맞게 지능적으로 중단없이 비용을 절감합니다.





Federation이 있는 단일 모니터링 및 컨트롤 플레인 제공





멀티 클러스터, 멀티 클라우드 Federation

멀티/하이브리드 클라우드를 현실화 시키는 역할을 수행

- Federation은 분산 시스템에 중앙 집중화를 실현합니다. 하시코프의 Nomad 조직이 지속적으로 추구한 비전입니다.
- Federation을 사용하면 여러 클라우드 및 데이터센터에 애플리케이션을 배포하고 관리하기 위한 단일 통합 모니터링 및 컨트롤 플레인을 사용할 수 있습니다.



— 02

고객 사례



TOP 15 PC GAMES OF 2023

www.activeplayer.io



Rank	-	Game	Concurrent	30 Days
1		Fortnite	1,608,914	30,950,114.00
2		Roblox	1,112,777	22,506,614.40
3		Minecraft	1,126,889	17,501,147.00
4		League of Legends	1,252,512	14,000,060.79
5		Counter Strike 2	588,012	2,223,337.86



HashiConf

The slide features a large Epic Games logo on the left. In the center, there is a video player showing a man with a beard and glasses speaking, wearing a black t-shirt with the Unreal Engine logo. To the right, there is a text section and a Fortnite game screenshot. The text reads: "What is **Fortnite**? One of the largest video games in the world. It has numerous game modes and even live events like concerts." Below this, it says "500M+ Accounts". The Fortnite screenshot shows two characters in colorful skins playing on a grassy landscape with mountains in the background.

What is **Fortnite**?

One of the largest video games in the world. It has numerous game modes and even live events like concerts.

500M+ Accounts

@EPICGAMES

A screenshot from the video game Fortnite, showing two characters in dynamic poses, one holding a large weapon, against a backdrop of a mountainous landscape under a clear blue sky.

[Cooking with Nomad: Powering the Fortnite creator ecosystem](#)



- 유연한 오케스트레이션 지원 (배포 큐잉, API 지원)
- 실행을 위한 라이프사이클 지원
 - Prestart → Main → Poststart → Poststop
- Docker 파라미터 관리
- 격리된 파일 시스템 (보안)
 - 각 Job 프로세스는 격리
 - 동일 Job내의 프로세스간에는 공유
- 매우 간단한 + Windows 환경 지원
- 2000+개 이상의 노드 관리
 - 클라우드 비용 최적화
 - 인스턴스 확장/축소



ROBLOX

Roblox는 어린이와 청소년에게 가장 인기 있는 게임 회사 중 하나입니다. 회사가 빠르게 성장하고 발전함에 따라 Nomad는 Roblox가 전용 Windows 서버에서 컨테이너로 성공적으로 마이그레이션하고 글로벌 게임 플랫폼을 쉽고 안정적으로 확장할 수 있도록 지원했습니다. Nomad를 통해 이 플랫폼은 월간 활성 사용자 수가 3천만 명에서 1억 명 이상으로 확장되었습니다.

On-prem

Containers

Windows

Gaming



Case Study:



Challenge



Solution



Results

Roblox는 Windows에서 Linux로의 컨테이너 마이그레이션을 지원하고, 애플리케이션 배포를 가속화하고, 1억 명의 활성 사용자로 확장하면서 성능을 보장하기 위한 오퍼레이터가 필요했습니다.

온프레미스 전용 서버를 사용한 수동 워크플로

- 수작업으로 최대 30,000대의 서버 관리
- PowerShell 스크립트를 통한 수동 스케줄링
- 서버/애플리케이션 상태의 블랙박스

느린 배포 이슈

- 새 애플리케이션을 배포하는 데 8주
- 리소스 용량 추가에 12주

높은 비용

- 연간 5백만 달러 이상의 Windows 라이선스 청구서



Case Study:

ROBLOX



Challenge



Solution



Results

Nomad 선택의 이유

- 운영 간소화

사용 편의성과 단일 경량 바이너리로 배포 및 운영할 수 있어 가치 실현 시간을 단축할 수 있습니다. Roblox는 Nomad 클러스터를 설정하고 애플리케이션을 배포하는 데 단 4일밖에 걸리지 않았습니다.

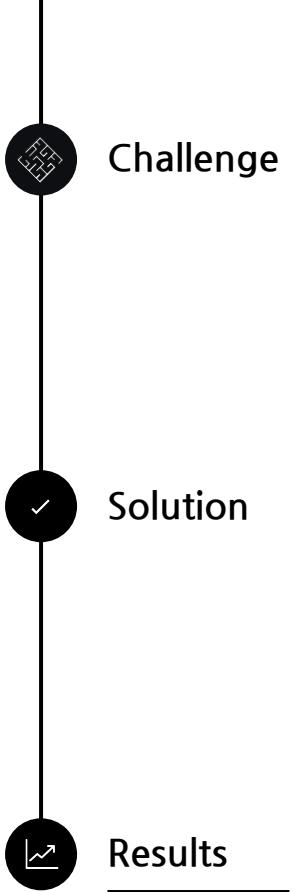
- 유연한 워크로드 지원

Nomad는 Windows 및 Linux 애플리케이션을 원활하게 배포할 수 있는 단일 오케스트레이터로서 원활한 마이그레이션을 지원합니다.



Case Study:

ROBLOX



Challenge

1억 MAU
200개 이상의 국가

11,000+ 서버
베어메탈 및 AWS에서 20개의
Nomad 클러스터

Solution

400+ 개발자
Nomad에 배포하는 개발자 수

4 SREs
Nomad를 운영하는 담당자 수

Results

~ 8 분
애플리케이션을 전 세계에
배포하는데 소요되는 시간

리소스 사용률 +200%
동일한 하드웨어에서 기존 게임 서버
프로세스를 두 배 이상으로 사용
(30,000 → 11,000)



“ Nomad는 단일 오케스트레이터로 마이그레이션 전부터 완료 까지 전 과정에서 Windows와 Linux 워크로드를 모두 원활하게 배포할 수 있었습니다. …

Kubernetes도 검토의 대상이였습니다. 다른 사람들이 떠든다고해서 "나도 써야하나?"라는 생각에 사로잡히지 마세요, 실제 필요한 수준에서 운영하는 것은 다른 이야기입니다.

Rob Cameron | Director of Cloud Services | Roblox



505
GAMES®

505 Games는 Control, 쿠킹 마마, 페이데이 2, 스나이퍼 엘리트 III, 테라리아, 호肯, 코지마 히데오의 데스 스트랜딩 등
다수의 수상 경력에 빛나는 타이틀을 보유한 글로벌 비디오 게임 퍼블리셔입니다. Nomad는 이 회사가 글로벌 규모의 최신
게임 경험을 지원하기 위해 인프라를 현대화하는 데 도움을 주었습니다.

Edge Deployment

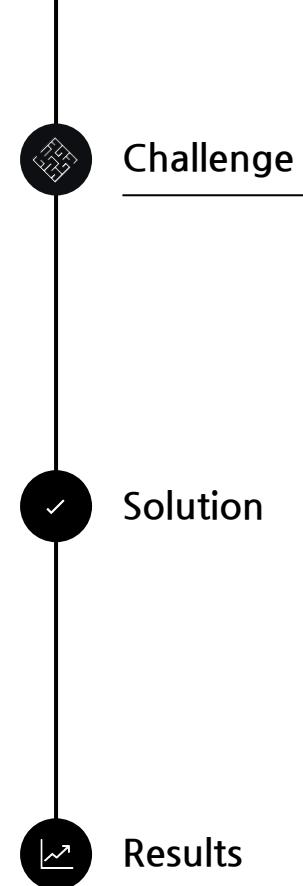
On-prem

Gaming

N Nomad
505GAMES



Case Study:



Challenge

최신 비디오 게임에 필요한 규모와 가동 시간을 지원할 수 없는 기존 인프라

- Ansible 플레이북을 사용한 기존 워크플로
- 느리고, 버그가 많고, 고통스러운 프로세스
- 애플리케이션의 상태에 대한 가시성 없음
- 내결함성 없음 - 비정상적인 서버 수동 재부팅
- 새로운 애플리케이션 버전에 대한 번거로운 롤아웃 프로세스
- 느린 개발 속도

505 Games는 사용 편의성이 높고, 온프레미스 베어메탈 환경을 최고 수준으로 지원하며, 글로벌 규모로 쉽게 운영할 수 있는 오케스트레이터가 필요했습니다.



Case Study:



Challenge



Solution



Results

Nomad 선택의 이유

- **유연한 워크로드 지원**

비-컨테이너에 대한 Nomad의 기능을 통해 회사는 컨테이너화할 필요 없이 tarball, zip, 기타 레거시 애플리케이션을 오케스트레이션할 수 있습니다.

- **단일 클러스터 토플로지를 사용한 글로벌 엣지 배포**

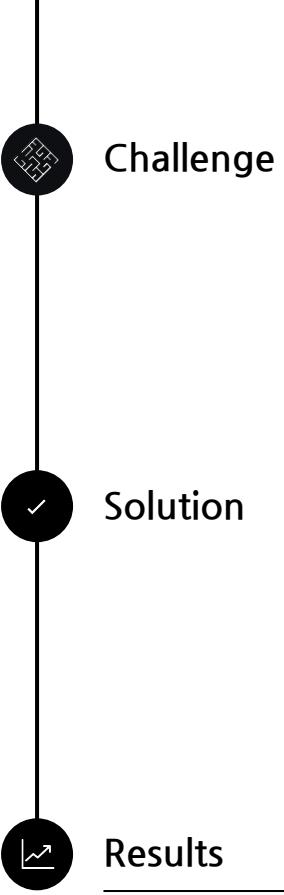
Nomad의 고유한 단순성 덕분에 전 세계 엣지에 노드를 배포하고 필요에 따라 특정 지역을 동적으로 확장하면서도 단순한 단일 클러스터 토플로지를 유지할 수 있습니다.

- **탁월한 온프레미스 지원**

Nomad는 온프레미스에서 사용하기 쉬워 매우 낮은 오버헤드로 운영 및 관리할 수 있습니다.



Case Study:



Challenge

10만 MAU
Nomad가 제공하는 게임
플랫폼에서 지원

단일 클러스터
글로벌 분산 시스템

Solution

4+ 국가
Nomad가 분산된 리전

1 SRE
Nomad를 운영하는 담당자 수

Results



“ 정말 멋진 점은 단 3대의 서버로 구성된 작은 Nomad 클러스터에서 400개의 노드를 사용할 수 있고, 그 서버 노드에서 18%의 CPU 정도만 사용한다는 것입니다. Nomad 배포 능력으로, 100만 컨테이너 챌린지를 훨씬 뛰어넘을 수 있다는 것을 알 수 있습니다.

Joshua Clausen | Online and Platform Service Director | 505 Games



고맙습니다

contact-kr@hashicorp.com

www.hashicorp.com