



INNOVATE

MODERN APPLICATIONS EDITION

26 OCT, 2023

Graviton 프로세서와 Karpenter로 극한까지 EKS 비용 최적화

구동수, 박진이
솔루션즈 아키텍트, AWS

Agenda

1. 쿠버네티스 클러스터 운영 상 비용 과제
2. AWS Graviton으로 비용 최적화
3. Karpenter로 비용 최적화
4. 정리

쿠버네티스 클러스터 운영 상 비용 과제

고객들이 당면한 비용 과제들

KUBERNETES에서의 비용과 성능



팀과 부서간 비용 할당



Chargeback/showback



리포팅, 예산 예측,
그리고 비용 최적화

늘어나는 Kubernetes 운영의 어려움

KUBERNETES에서의 비용과 성능

89% 기업에서는 Kubernetes를 운영 중...

68% 기업은 Kubernetes 지출을 모니터링하지 않거나 월별 추정치에 의존

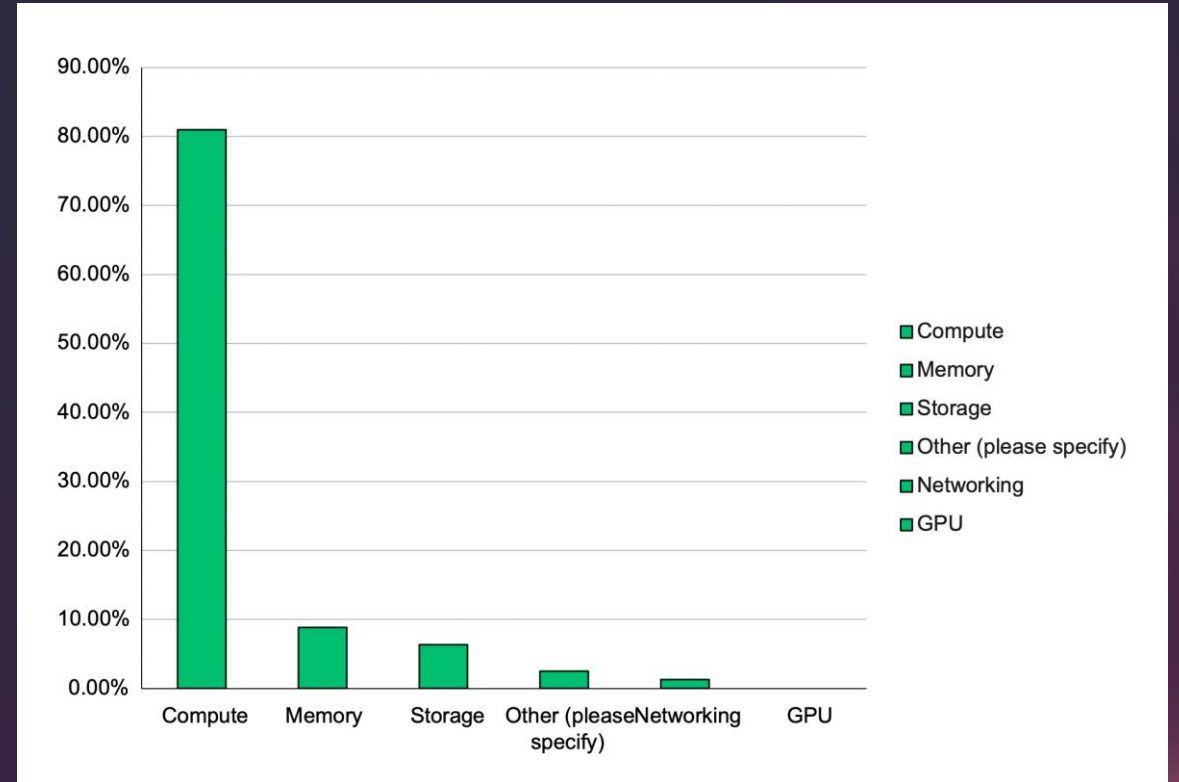
68% 기업에서는 Kubernetes 비용이 상승하고 있으며,
해당 조사대상의 50%는 연간 20%의 증가를 경험.

Source: CNCF Annual Survey, FinOps for Kubernetes by FinOps Foundation

Kubernetes 클러스터에서의 비용 요인들

KUBERNETES에서의 비용과 성능

1. 클러스터의 컴퓨트와 메모리 > 90%
 - a. 애플리케이션
 - b. 활용도가 낮은 리소스들
 - c. Kubernetes 오버헤드, 데몬셋 등
2. 클러스터의 스토리지와 네트워킹
3. 3rd 소프트웨어 라이선스
4. 관리형 Kubernetes 컨트롤 플레인



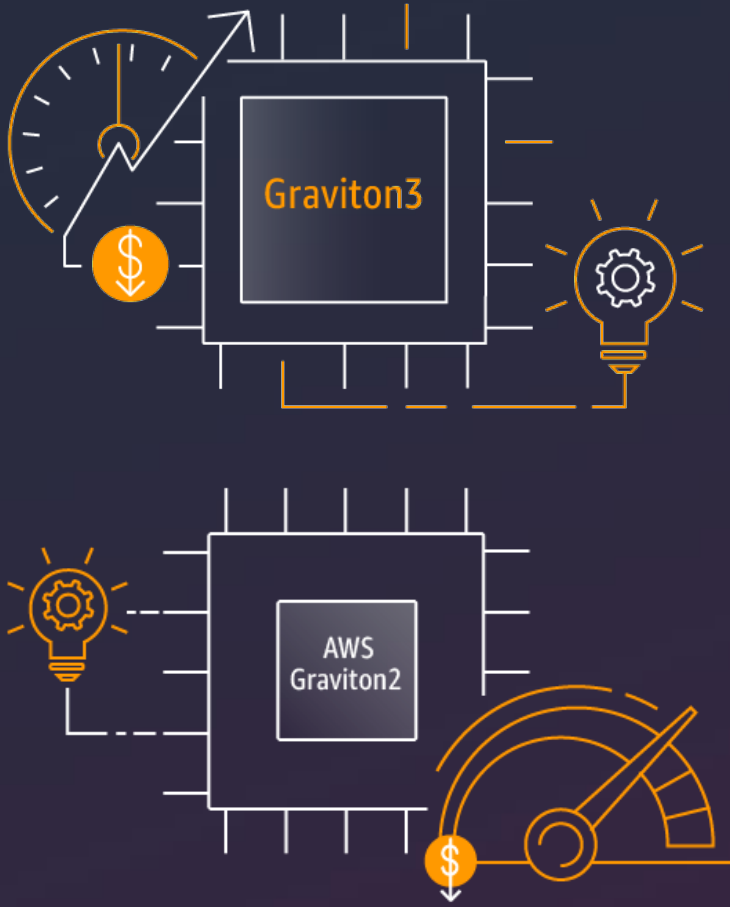
Source: FinOps for Kubernetes by FinOps Foundation

AWS Graviton으로 비용 최적화

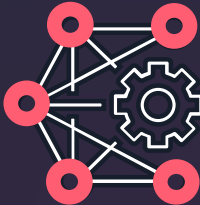


AWS Graviton 이란?

AWS GRAVITON을 통한 KUBERNETES 컴퓨트 최적화



64-bit Arm 프로세서 코어를 갖춘
맞춤형 AWS 실리콘



클라우드 중심 워크로드를 위한
최적화

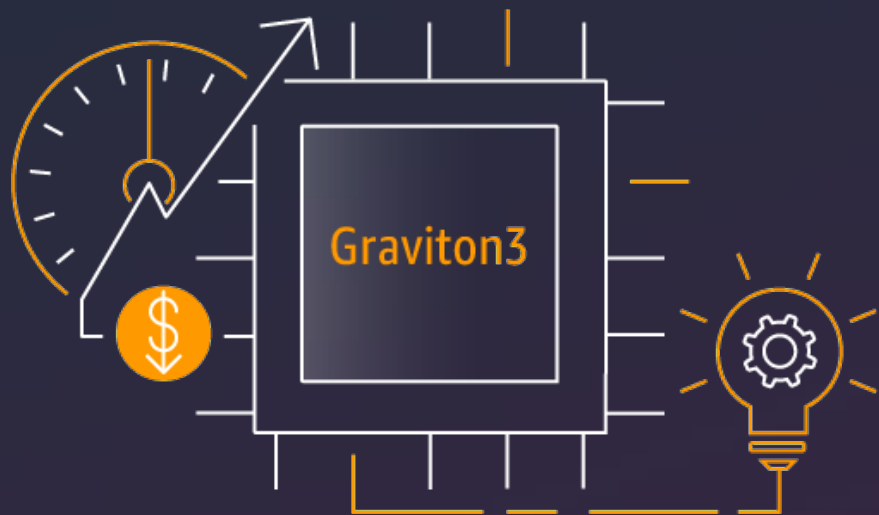


고객을 대신하여 신속하게 혁신, 빌드
및 반복 작업 수행

AWS Graviton3

Now Available!

AWS GRAVITON을 통한 KUBERNETES 컴퓨트 최적화



Graviton2 대비 최대 25% 향상된 성능

Graviton2 대비 최대 2배 높은 부동소수점,
암호화 성능 및 최대 3배 향상된 ML 성능

동급 EC2 인스턴스 대비 60% 더 높은
에너지 효율성

컴퓨팅 집약적 워크로드를 위한 최고의
가격 대비 성능을 제공하는 C7g 인스턴스

Amazon EKS 와 AWS Graviton을 사용하는 방법

AWS GRAVITON을 통한 KUBERNETES 컴퓨트 최적화

Amazon EKS managed node groups

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: eks-graviton-cluster
  region: us-west-2
managedNodeGroups:
  - name: mng-arm-c7
    instanceType: c7g.xlarge
    desiredCapacity: 2
```

Amazon EKS with Karpenter

```
apiVersion: karpenter.sh/v1alpha5
kind: Provisioner
metadata:
  name: default
spec:
  requirements:
    - key: "kubernetes.io/arch"
      operator: In
      values: ["arm64"]
```

AWS Graviton 및 Amazon EKS 고려 사항

AWS GRAVITON을 통한 KUBERNETES 컴퓨트 최적화

Remember: 애플리케이션 프로그래밍 언어와 에코시스템이 멀티 아키텍처를 지원해야 합니다(예: Go, Rust)

- Graviton에서는 ARM, x86 인스턴스 유형을 혼합하여 운영하고, 애플리케이션을 적절한 인스턴스 유형에서 실행할 수 있도록 ARM 호환 컨테이너 이미지, 이상적으로 멀티 아키텍처(ARM64, AMD64) 이미지가 필요합니다.
- 멀티 아키텍처 컨테이너 이미지는 Amazon Elastic Container Registry (ECR)에서 완벽하게 지원 됩니다.

삼성전자 고객 사례

AWS GRAVITON 도입으로 AMAZON EKS 클러스터 비용 최적화

→ API서버의 응답속도 개선

예상보다 결과가 좋다.

APPLICATION 특성에 따라

ARM이 더 좋은 퍼포먼스를 낼 수 있다.

응답속도 개선	
P50	5.6%
P75	4.5%
P90	48.2%

→ 15%

인스턴스 비용 절감

동일 CLASS 대비 15% 정도 저렴하므로

비용 절감 효과를 낼 수 있다.

	인스턴스 타입	인스턴스 비용(\$)
교체 전	C5.2xlarge	\$0.34
교체 후	C7g.2xlarge	\$0.29
	절감 비율	15%



Karpenter로 비용 최적화



Karpenter – Kubernetes를 위한 지능형 컴퓨팅

KARPENTER를 이용해 KUBERNETES 컴퓨팅 비용을 자동으로 최적화



Karpenter는 **고성능의 지능형**
Kubernetes 컴퓨트 프로비저닝
및 관리 솔루션



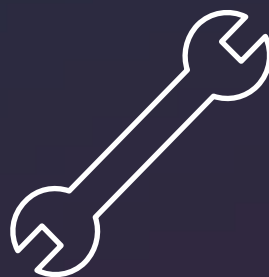
Karpenter를 통해 Kubernetes와
Amazon EC2 간의 긴밀한 통합으로
AWS에서의 이점 극대화

Karpenter를 사용해야하는 이유

KARPENTER를 이용해 KUBERNETES 컴퓨팅 비용을 자동으로 최적화



애플리케이션 부하 변경에
수십 초 이내에 대응해
가용성 향상



Kubernetes 네이티브 CRD로
GitOps 워크플로를 지원해
운영 오버헤드 최소화



지능적으로
인스턴스 유형을 선택해
더 낮은 컴퓨팅 비용으로
Pod를 통합

Karpenter를 통한 컴퓨트 비용 절감

KARPENTER를 이용해 KUBERNETES 컴퓨팅 비용을 자동으로 최적화

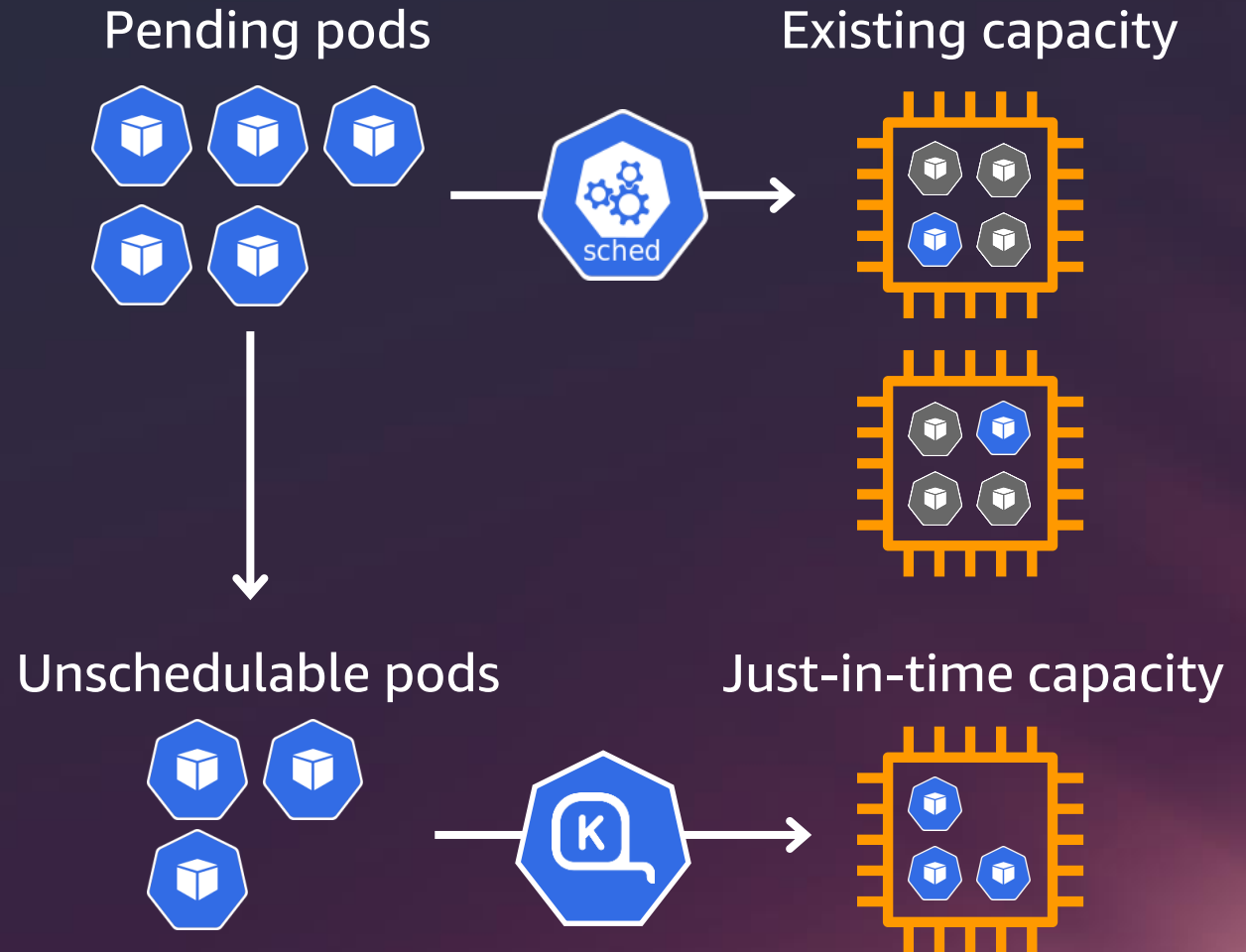
- 지능형의 동적인 인스턴스 유형 선택
- 자동 워크로드 Consolidation 기능
- **EC2 Spot** 및 **AWS Graviton** 와 같은 AWS 컴퓨트 서비스 및 기능의 Kubernetes 네이티브 통합
- 일관성 있는 더 빠른 노드 구동시간을 통해 시간/비용 낭비 최소화
- 운영 오버헤드 감소로 비즈니스 가치 제공에 집중 가능

Karpenter 동작 방식

KARPENTER를 이용해 KUBERNETES 컴퓨팅 비용을 자동으로 최적화

Karpenter는 **Kubernetes 스케줄러** 및 **컴퓨트 제공자**와 함께 동작

- 최적의 컴퓨팅 리소스를 동적으로 선택
- 사용되지 않는 노드의 종료
- Kubernetes 네이티브 스케줄링 제약사항을 이용하여 가용성 요구사항 달성

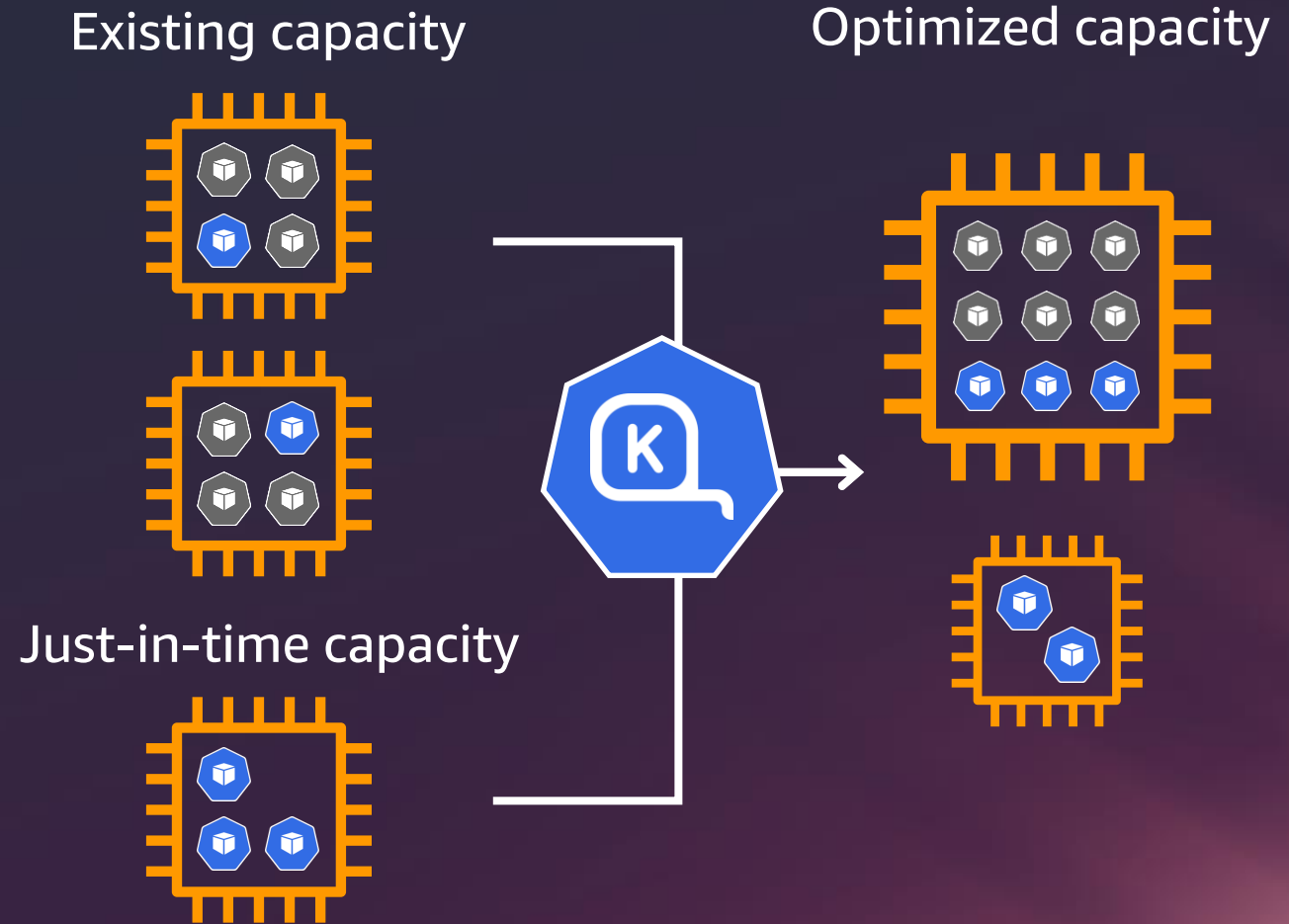


Karpenter Consolidation 기능

KARPENTER를 이용해 KUBERNETES 컴퓨팅 비용을 자동으로 최적화

Karpenter의 **Consolidation** 기능은 시간 경과에 따라 클러스터 효율성을 증대

- 구동 중인 Pod를 클러스터 내 사용률이 낮은 컴퓨팅 리소스에 재스케줄링
- 사용률이 낮은 노드를 대체하기 위해 보다 비용 효율적인 컴퓨팅 리소스를 새롭게 구동



Karpenter Consolidation 기능 동작 방식

KARPENTER를 이용해 KUBERNETES 컴퓨팅 비용을 자동으로 최적화

1. Consolidation poll 주기동안 대기
2. 대상 노드를 식별
 - a. Consolidation 기능이 활성화 된 Provisioner에 의해 구동된 노드
 - b. `do-not-consolidate` 어노테이션이 설정 되지 않은 노드
 - c. 초기화가 완료되었으며, reports Ready이며, 모든 확장 리소스가 등록된 노드
3. 중단 비용을 기준으로 노드 정렬
4. 최저 중단 비용을 가진 노드가 존재하지 않을 경우의 스케줄링을 시뮬레이션

기존 노드에 Pod 스케줄링 가능 → 노드 삭제

더 저렴한 노드에 Pod 스케줄링 가능 → 노드 축소

Pod가 동일 혹은 더 비싼 노드를 요구 → 기존 유지

Amazon EKS 및 Karpenter를 통한 비용 절감

KARPENTER를 이용해 KUBERNETES 컴퓨팅 비용을 자동으로 최적화

Karpenter는 **AWS Graviton** 및 **EC2 Spot**와의 통합을 간소화하여 클러스터 활용률을 높이고 비용을 절감

```
kind: Provisioner
spec:
  consolidation:
    enabled: true
  requirements:
    - key: karpenter.sh/capacity-type
      operator: In
      values: ["spot", "on-demand"]
    - key: kubernetes.io/arch
      operator: In
      values: ["amd64", "arm64"]
```



무신사 고객 사례

KARPENTER를 이용해 KUBERNETES 컴퓨팅 비용을 자동으로 최적화

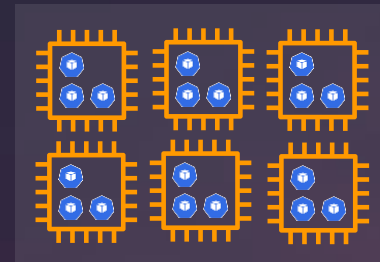
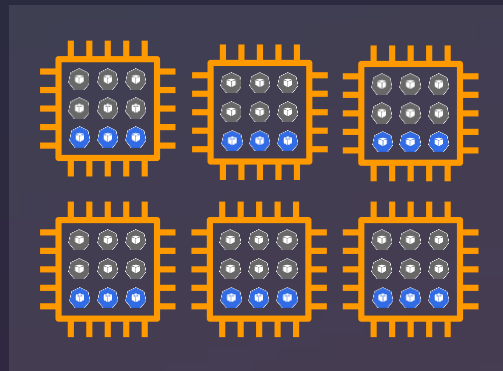


노드 크기 최적화

12XLARGE 노드 6대 → 8XLARGE 노드 6대

기존의 느린 확장속도를 개선하여

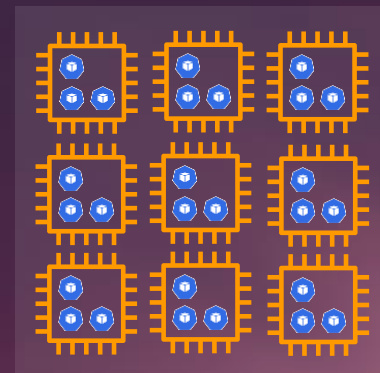
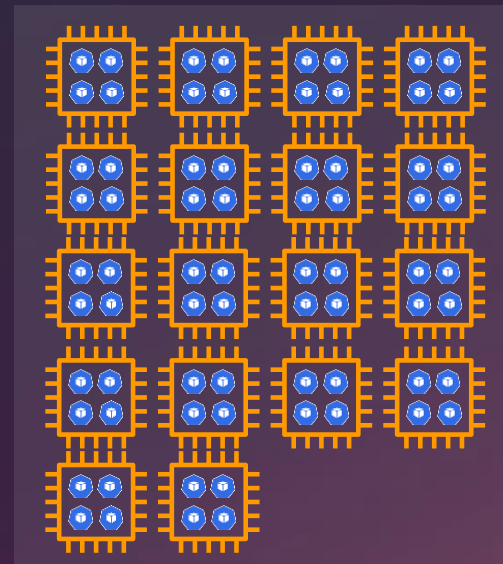
보수적인 노드 사이징을 해소 및 비용 절감



노드 개수/활용률 최적화

ELASTICSEARCH 운영 노드를 18개 → 9개 축소

노드 당 리소스 할당량 40% → 90%로 상승 (REQUESTS 기준)



정리



앱 현대화 리소스 허브

AWS가 제공하는 앱 현대화에 관한 다양한 자료들을 통해 더욱 심층적으로 학습해 보세요!

현대화 여정을 계획하고 실행하는데 유용한 AWS의 다양한 자료들을 앱 현대화 리소스 허브에서 확인해 보세요.

- AWS기반 현대적 애플리케이션 구축 자료
- 클라우드 현대화의 비즈니스 가치
- 이벤트 드리븐 아키텍처 소개 및 가이드
- 풀스택 웹 앱 및 모바일 앱 개발 가이드
- 총 소유 비용 비교: 서버리스 기술과 서버 기반 기술 비교



<https://bit.ly/modern-apps-aws>

리소스 허브 방문하기



AWS 교육 및 자격증

AWS 스킬 빌더, 600개 이상의 무료 디지털 콘텐츠를 만나보세요!

30개 이상의 AWS 솔루션에 대한 디지털 셀프 학습 계획 및 Ramp-Up 가이드 등 다양한 학습 리소스를 제공하여 여러분에게 가장 필요한 클라우드 기술과 서비스에 집중해서 학습하실 수 있습니다.

- 여러분의 진도에 맞춰 원하는 목표를 달성하세요.
- 학습 계획에 따라 여러분의 지식과 기술을 발전시키세요.
- AWS 자격증으로 여러분의 클라우드 역량을 증명하시기 바랍니다.



<https://aws.amazon.com/ko/training/digital/>

클라우드 기술 습득하기



AWS 파트너와 참여

AWS 파트너와 함께 클라우드로의 여정을 가속화하고 AWS가 제공하는 모든 서비스를 최대한 활용해 보세요.

- **혁신** – 조직을 위한 혁신적이고 비용 효율적인 확장 가능한 클라우드 솔루션 및 기능으로 최첨단 기술 변화에 보조를 맞출 수 있습니다.
- **전문성** – 전략적이고 경험이 풍부한 전문가들이 귀사의 비즈니스 성장을 지원하는 획기적이고 적절하며 신뢰할 수 있는 솔루션을 제고합니다.
- **글로벌 범위** – 소프트웨어, 하드웨어, 및 서비스 전반에 걸쳐 신뢰할 수 있는 AWS 파트너의 글로벌 커뮤니티를 만나볼 수 있습니다.



[Connect with an AWS Partner](#)

적합한 AWS 파트너 찾아보기

AWS Innovate – 앱 현대화 특집에 참석해 주셔서 감사합니다.

저희가 준비한 강연, 어떻게 보셨나요?
더 나은 세미나를 위하여 **설문을 꼭 작성해 주세요!**



aws-korea-marketing@amazon.com



twitter.com/AWSKorea



facebook.com/amazonwebservices.ko



youtube.com/AWSKorea



linkedin.com/company/amazon-web-services



twitch.tv/aws

Thank you!