AWS 클라우드로 천만명 웹 서비스 확장하기

윤석찬, AWS 테크에반젤리스트

@channyun

목차

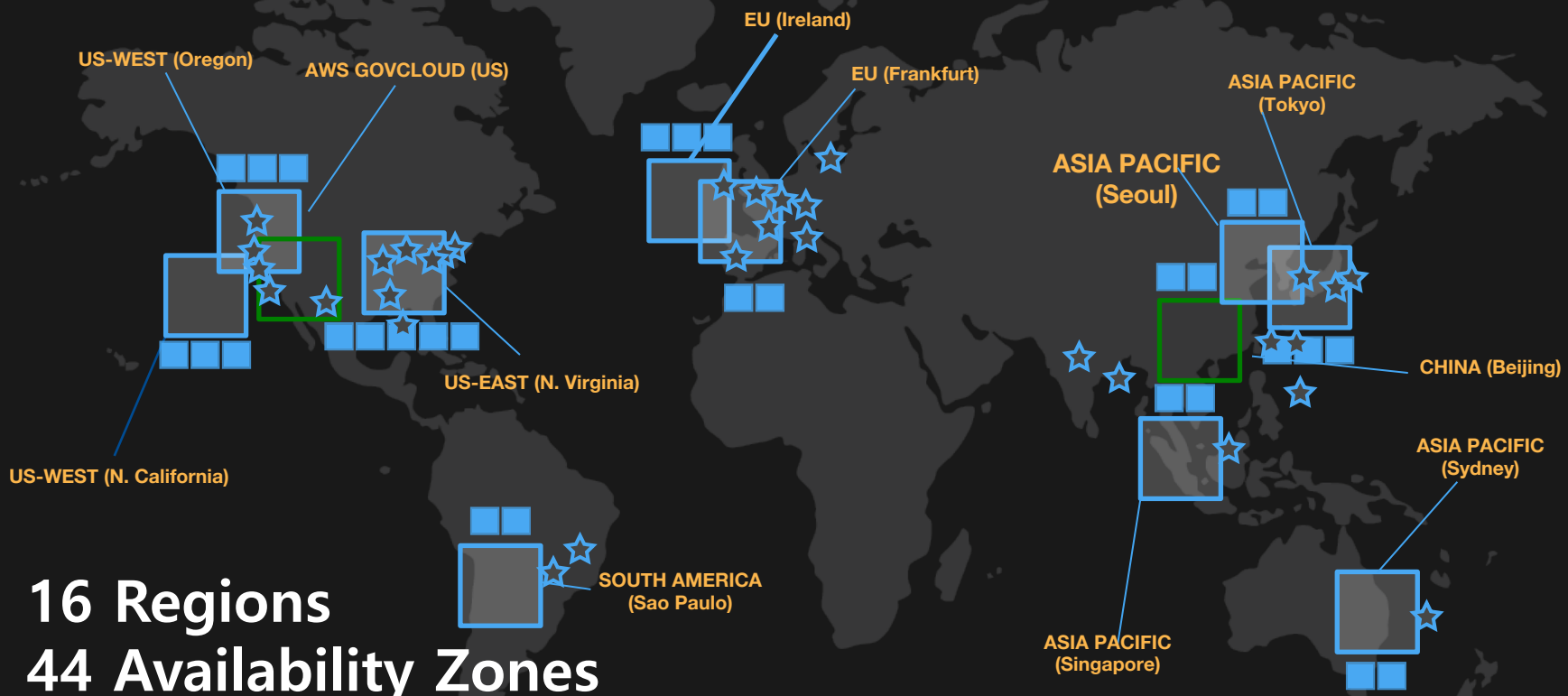
- **AWS 기본 지식 3가지**
- **사용자 증가에 따른 클라우드 확장 방법**
  - 스케일 업을 통한 수직적 확장
  - 부하 분산 및 멀티 AZ를 통한 수평적 확장
  - 높은 성능과 가용성을 위한 고급 아키텍처
  - 오토 스케일링
- **클라우드 네이티브 전략**
  - 인프라 자동화 및 모니터링
  - 새로운 흐름: 서버 없는(Serverless) 아키텍처
- **마무리**


?
아마존은 왜 클라우드 컴퓨팅 서비스를 시작 했나?
Since 2006

멋진 클라우드 아키텍처를 위한

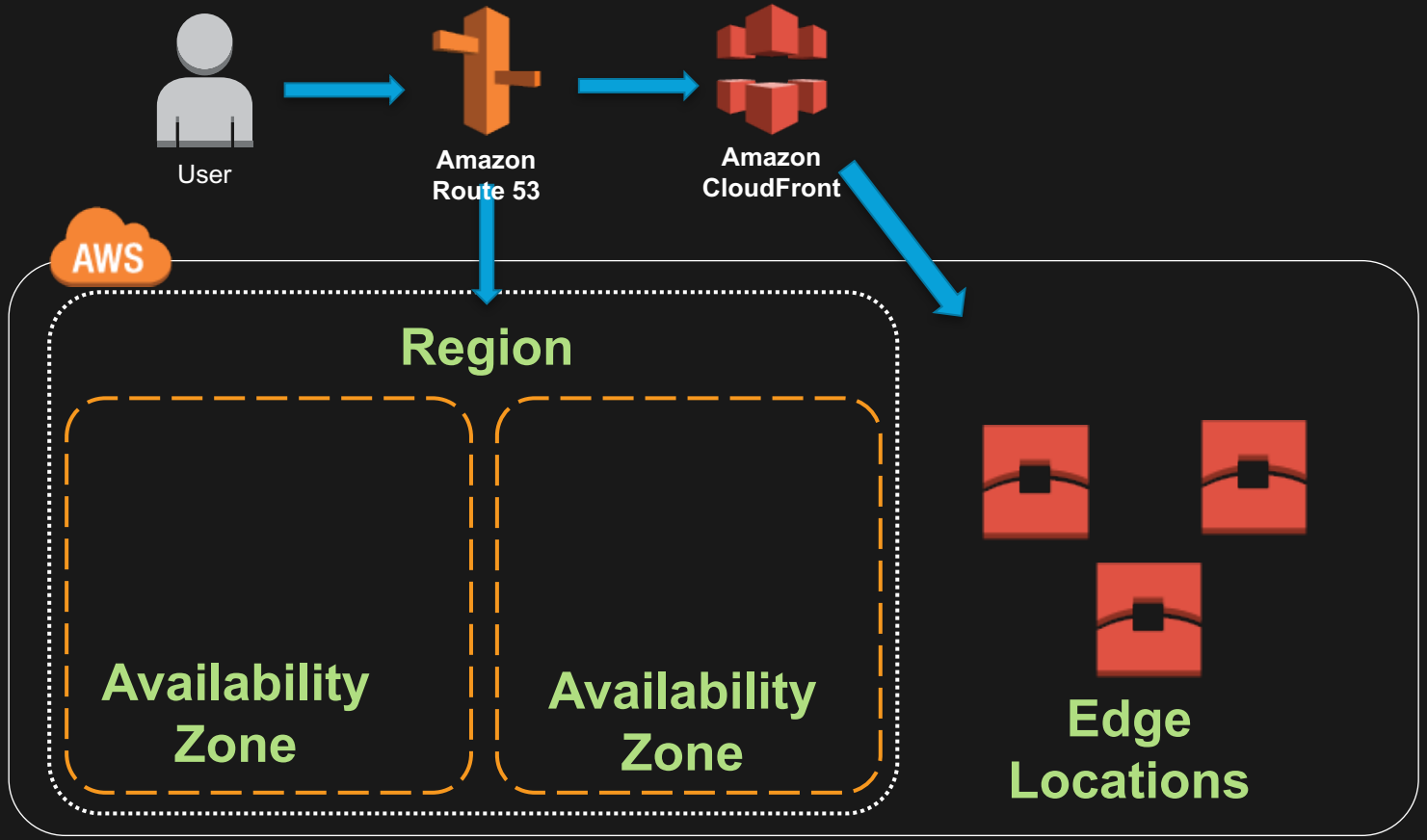
AWS 기본 지식 3가지

1 고가용성 글로벌 클라우드 인프라



16 Regions
44 Availability Zones
79 Edge Locations

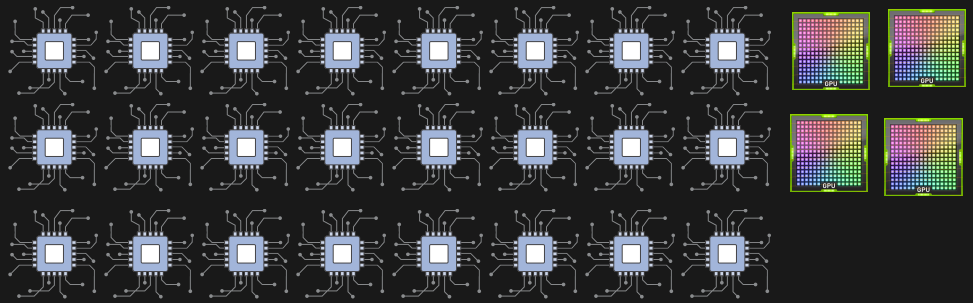
1 고가용성 글로벌 클라우드 인프라



2 언제나 원하는 대로 사용한 만큼

vCPU 32

GPU 4



x 50

vCPU 1,600

GPU 200

g2.8xlarge

= $2.6 per hour

(버지니아 기준)

g2.8xlarge x 50

= $130 per hour

스팟인스턴스를 쓴다면?

= $13 per hour

2 언제나 원하는 대로 사용한 만큼

```
$aws ec2-run-instances ami-b232d0db
      --instance-count 50
      --instance-type g2.8xlarge
      --region us-east-1
```

```
$aws ec2-stop-instances
      i-10a64379 i-10a64280 ...
```

3 AWS 클라우드의 다양하고 폭넓은 서비스

고객 및 기술 지원

- 고객지원
- 프로페셔널 컨설팅
- 파트너 생태계
- 교육 및 인증
- 솔루션 아키텍트
- 영업 지원
- 보안 및 빌링 시스템

하이브리드 환경

- 네트워크 통합
- 전용 회선 연결
- 통합 인증
- 통합 앱 개발 환경
- 데이터 백업
- 통합 리소스 관리

마켓 플레이스

비지니스 앱 | 비지니스 인텔리전스 | DevOps 도구 | 보안 | 네트워킹 | 데이터베이스 | 스토리지

데이터 분석

- 데이터웨어 하우스
- 비지니스 인텔리전스
- Hadoop/ Spark
- 실시간 데이터 분석
- 실시간 데이터 저장
- 머신 러닝
- Elastic Search 서비스

앱 서비스

- 알람 및 큐 서비스
- 워크 플로우
- 풀텍스트 검색
- Email 전송
- 동영상 변환

모바일 서비스

- API Gateway
- 모바일 허브
- 모바일 인증
- 기기 동기화
- 모바일 분석
- 모바일 앱 테스트
- 푸시 알림

개발 및 운영 도구

- 원 클릭 앱 개발
- DevOps 리소스 관리
- 앱 라이프사이클 관리 도구
- 콘테이너 서비스
- 클라우드 함수
- 리소스 템플릿

IoT

- 규칙 엔진
- 디바이스 쉐도우
- 디바이스 SDK
- 디바이스 게이트웨이
- 레지스트리

엔터프라이즈

- 가상 데스크톱
- 기업용 공유 도구
- 기업용 이메일
- 백업 및 복구

보안 및 규정 준수

사용자 인증 관리 | 서비스 접근 제어 | 암호 키 관리 및 저장 | 모니터링 로그 | 기업 내 규정 준수 | 웹 방화벽 | 기업 자원 설정 및 보고 | 리소스 사용량 및 감사

핵심 서비스

- 컴퓨팅 — VMs, Auto-scaling, Load Balancing, Containers, Cloud functions
- 스토리지 — Object, Blocks, File, Archival, Import/Export
- 콘텐츠 배포 (CDN)
- 데이터베이스 — Relational, NoSQL, Caching, Migration
- 네트워킹 — VPC, DX, DNS

인프라

글로벌 리전 (Region) | 가용 영역 (AZ) | 콘텐츠 배포 지점(Edges)

3 AWS 클라우드의 다양하고 폭넓은 서비스

비지니스 요구 사항에 맞는

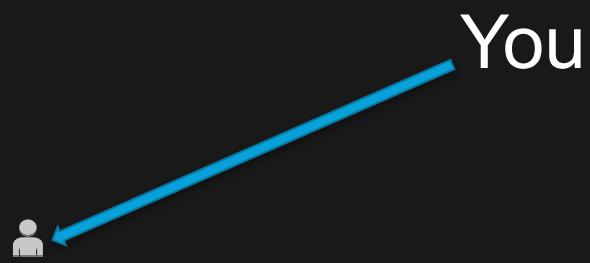
70여개 이상의 서비스 조립을 통해 유연한 활용 가능

AWS 활용 = Building Block 조립

이제 시작해볼까요?

첫날! 첫 사용자!

1 user



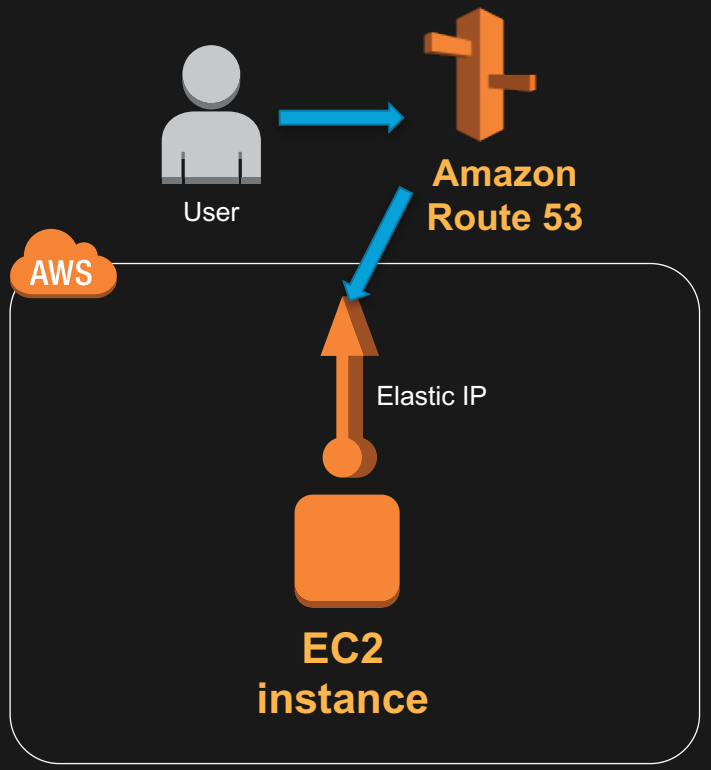
첫날! 첫 사용자 1명

Amazon Route 53

- 도메인 네임 서비스
- 글로벌 부하 분산
- AWS Cloud에 최적화

Amazon Elastic Compute Cloud

- 하나의 인스턴스(가상 서버) 구성
  - 웹 서버
  - 데이터베이스
  - 시스템 운영
  - 파일 저장....



Demo: Amazon Elastic Compute Cloud

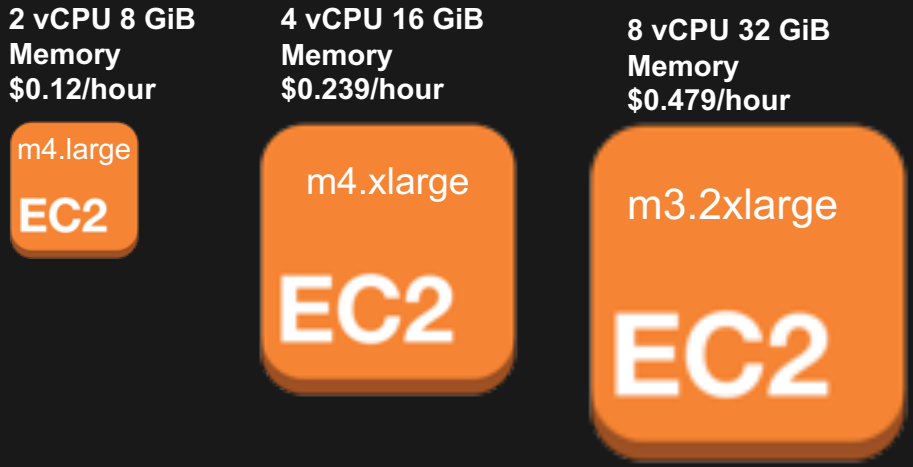
서버 트래픽↗ "좀 더 성능 좋은 인스턴스!"

스케일업(Scale-Up)

손 쉽게 인스턴스 타입을
변경함으로서 서비스 확장 가능

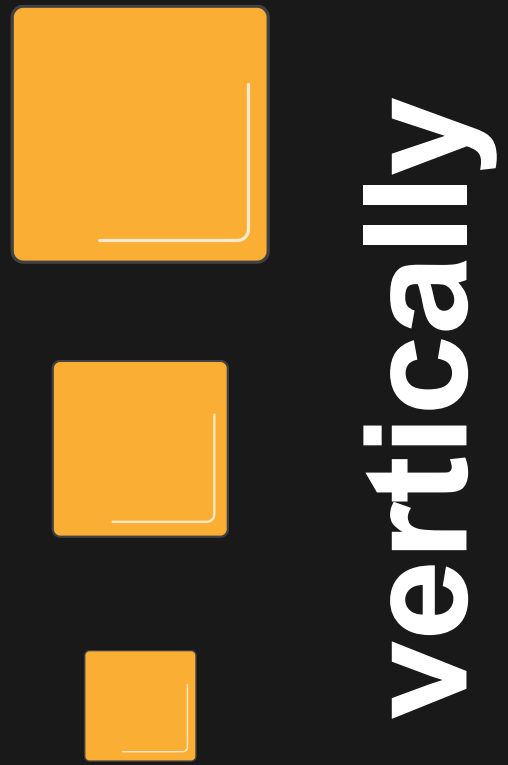
다양한 인스턴스 타입

- CPU기반
- 메모리 기반
- I/O 기반
- 스토리지 기반

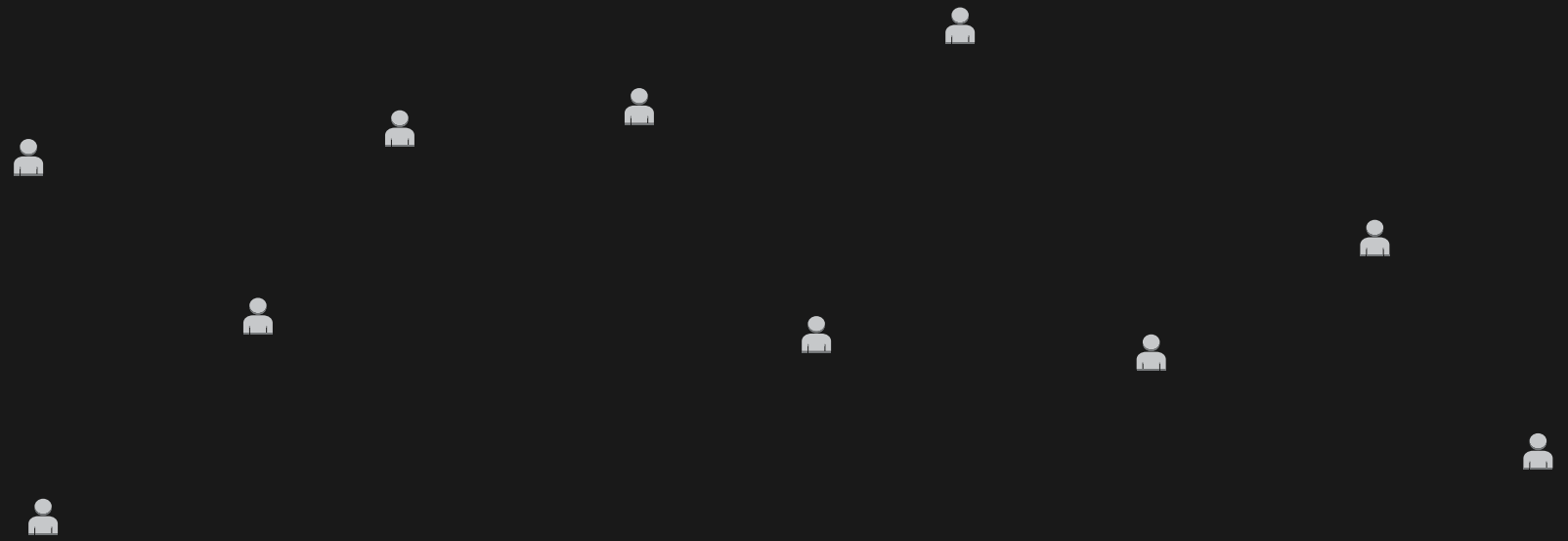


서울 리전 기준 http://aws.amazon.com/ko/ec2/pricing

손쉬운 수직적 확장 가능



Users >10

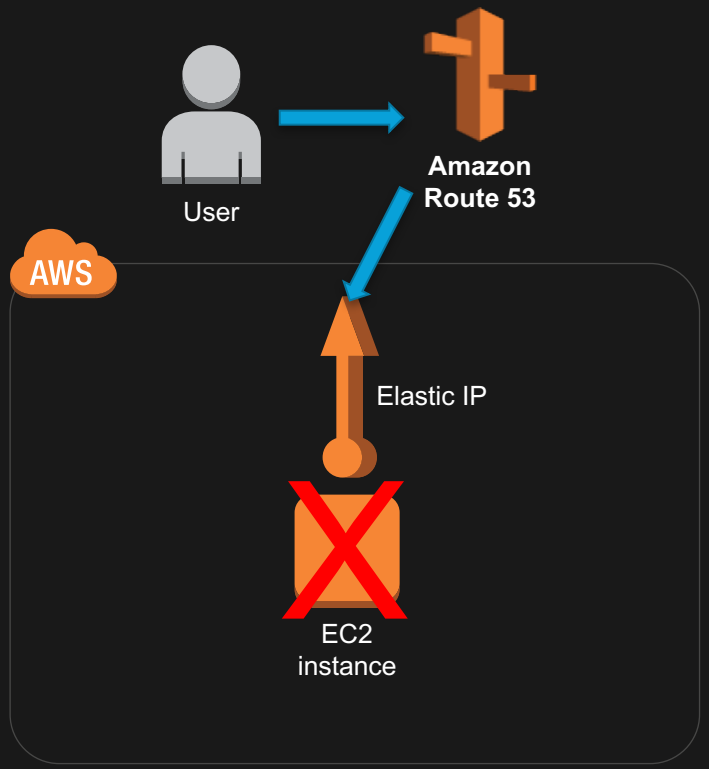


사용자 10+

문제점

- No Failover
- No Redundancy

"한 바구니에 계란을
모두 담지 말자!"

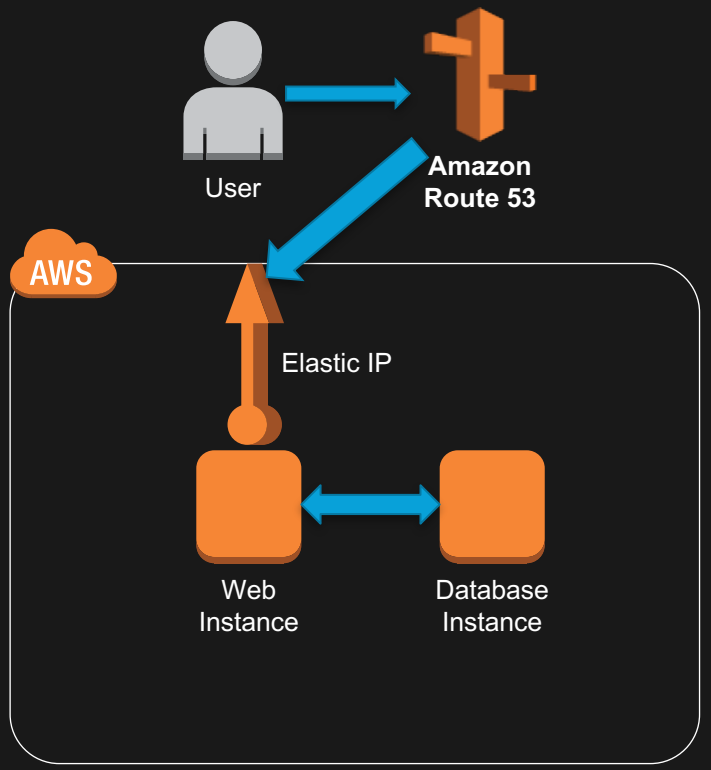


사용자 10+

해법 I

기능에 따라 인스턴스의 역할을 나눈다!

- 웹 서버용 인스턴스
- DB용 인스턴스



데이터베이스 선택 사항

직접 운영

Amazon EC2
직접 설치

원하는 데이터
베이스를 선택
하여 운영

라이센스/백업
등 직접 운영

AWS 관리 서비스 선택

Amazon RDS

MySQL
Microsoft SQL
Oracle
PostgreSQL
MariaDB
Amazon Aurora

라이센스/백업등
선택적 제공

Amazon
DynamoDB

SSD 스토리지기반
NoSQL 서비스
빠른 처리속도

중단없는 확장성 및
관리 필요 없음

Amazon
Redshift

대용량 병렬
페타바이트급
데이터웨어 서비스

빠르고 강력한
확장성 제공

RDB? NoSQL?

> 한해에 5 TB 이상이신가요?

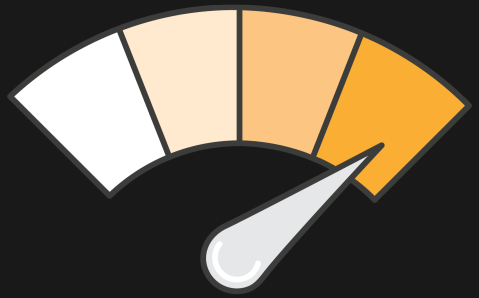
갑작스럽게 데이터가 증가하시나요?

OK!

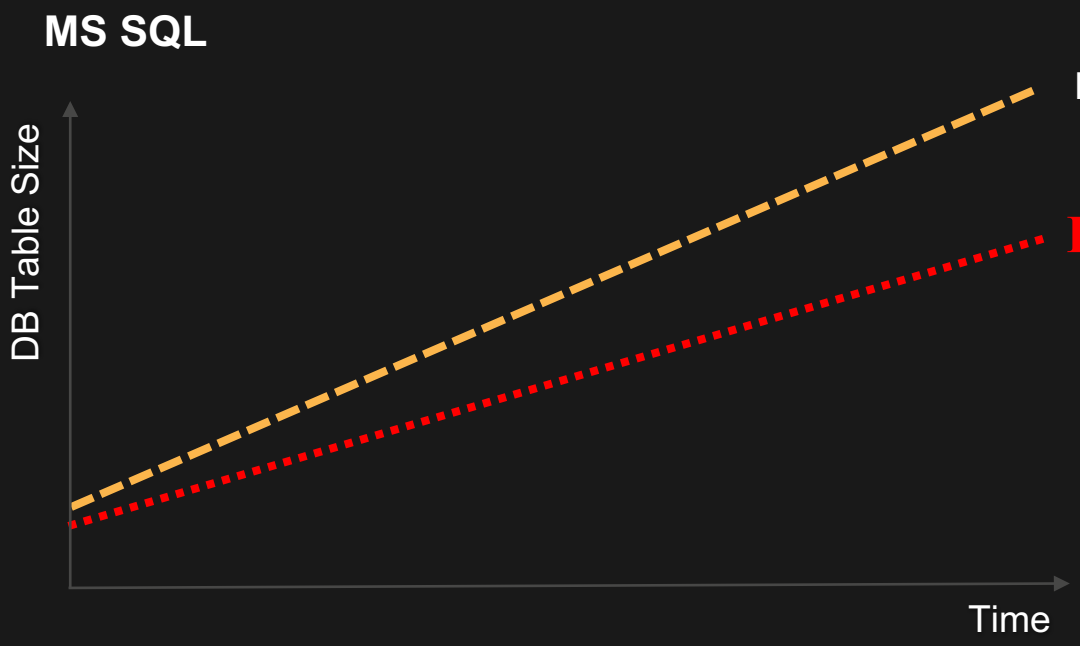
NoSQL을 고려하세요.

Amazon DynamoDB

- ✓ 손 쉬운 관리형 NoSQL 서비스
- ✓ 일관된 응답시간 – 10밀리초 미만
- ✓ 읽기 및 쓰기 용량 제어 가능
- ✓ 무제한 스토리지 제공
- ✓ 프로비저닝한 용량 만큼만 과금



고객 사례: 모바일 게임 데이터베이스

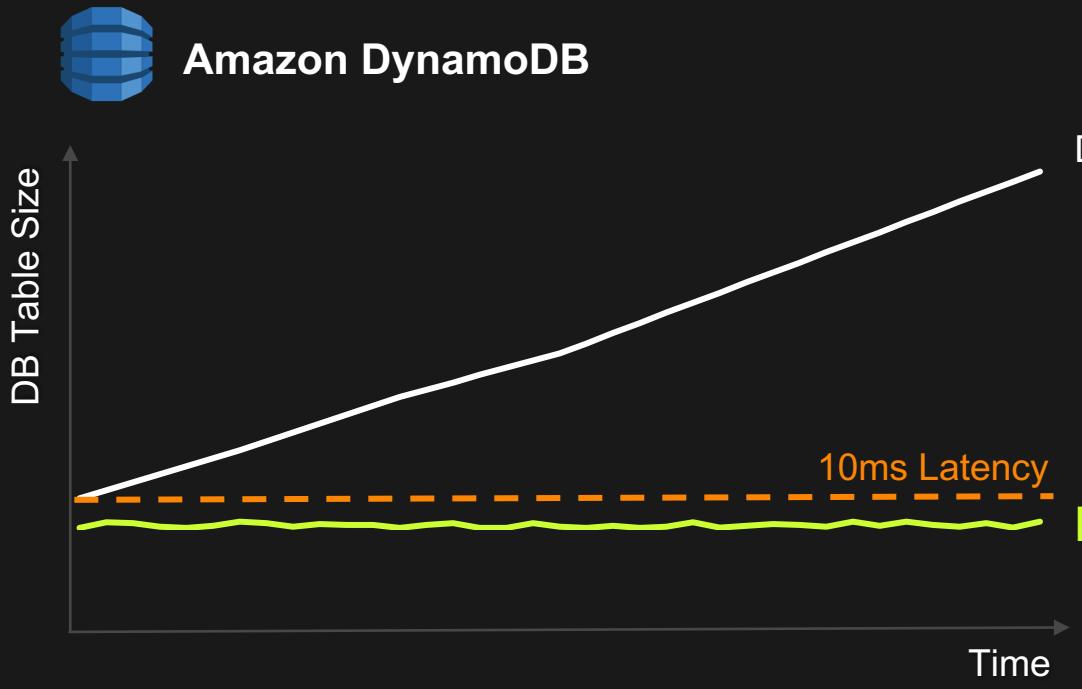


DB Table Size의 증가

DB latency 증가

NEXON

고객 사례: HIT에서 DynamoDB 적용



NEXON

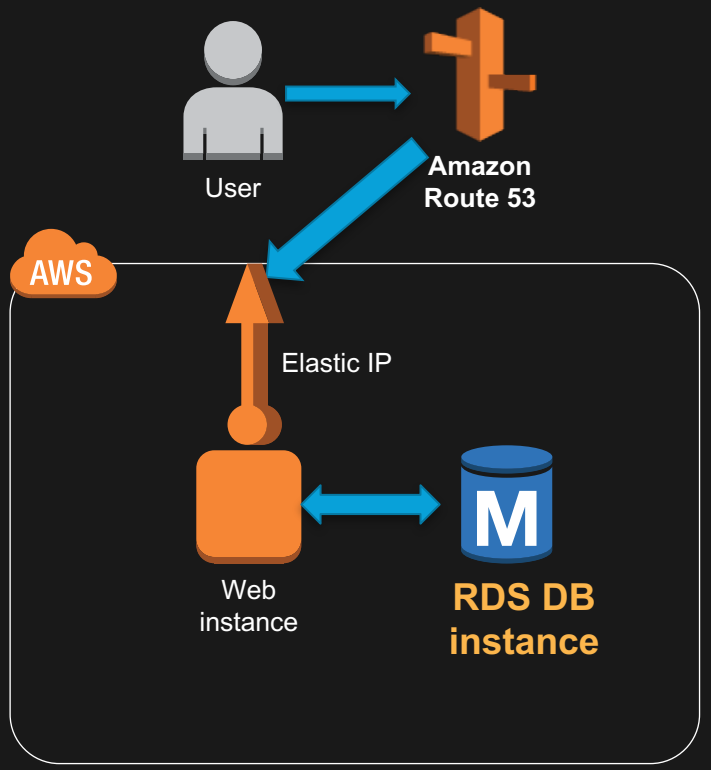
Users >100

사용자 >100

해법 II

기능에 따라 인스턴스의
역할을 나눈다!

편리한 DB 운영을 위해
Amazon RDS 선택!

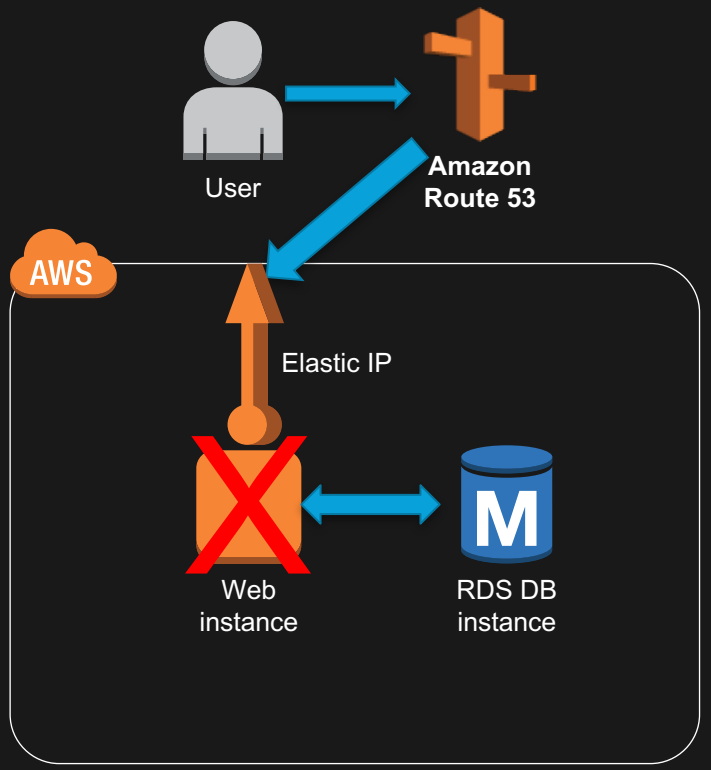


사용자 >100

문제점

만약 웹 서버에 문제가 생긴다면?

- 장애 복구의 어려움
- 증설의 어려움




Users >1,000

사용자 > 1000+

Elastic Load Balancing

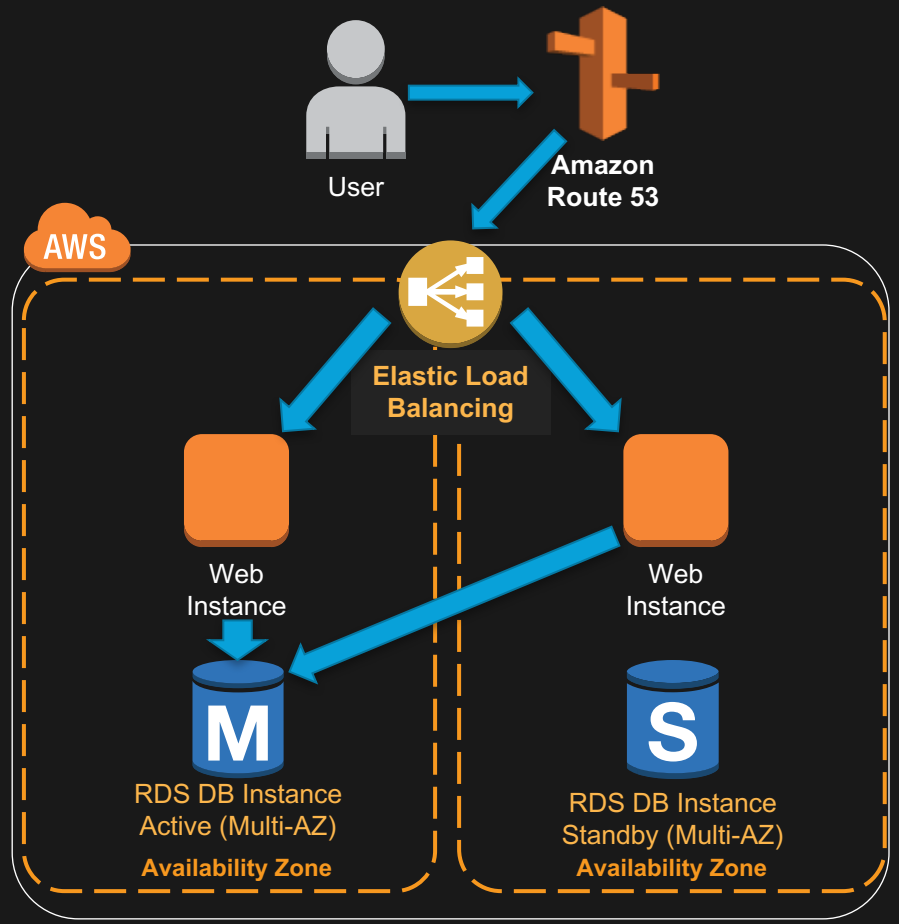
확장성 높은 부하 분산 서비스

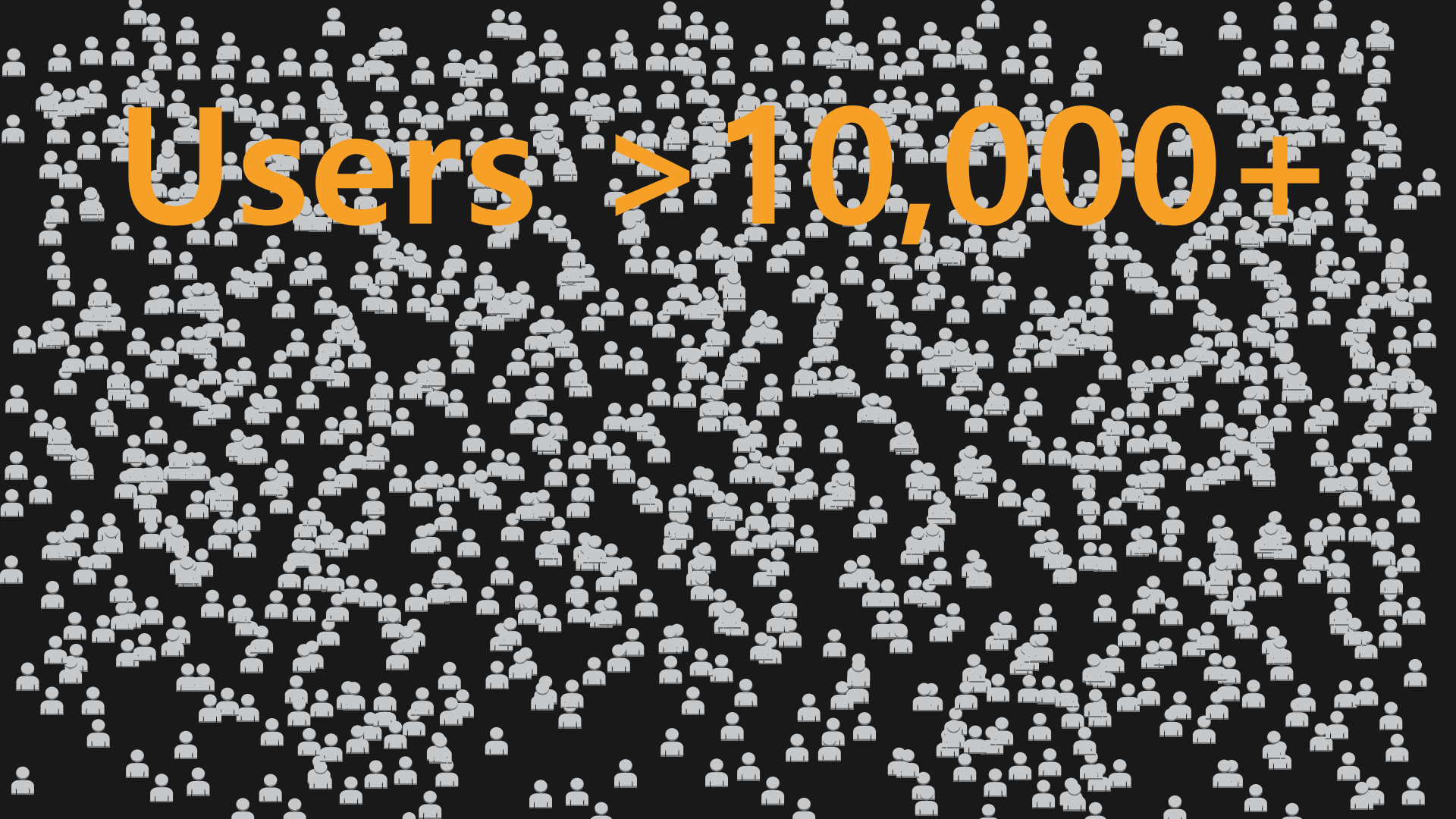
Multi-AZ 서버 구성

가용 영역을 통한 고가용성 확보

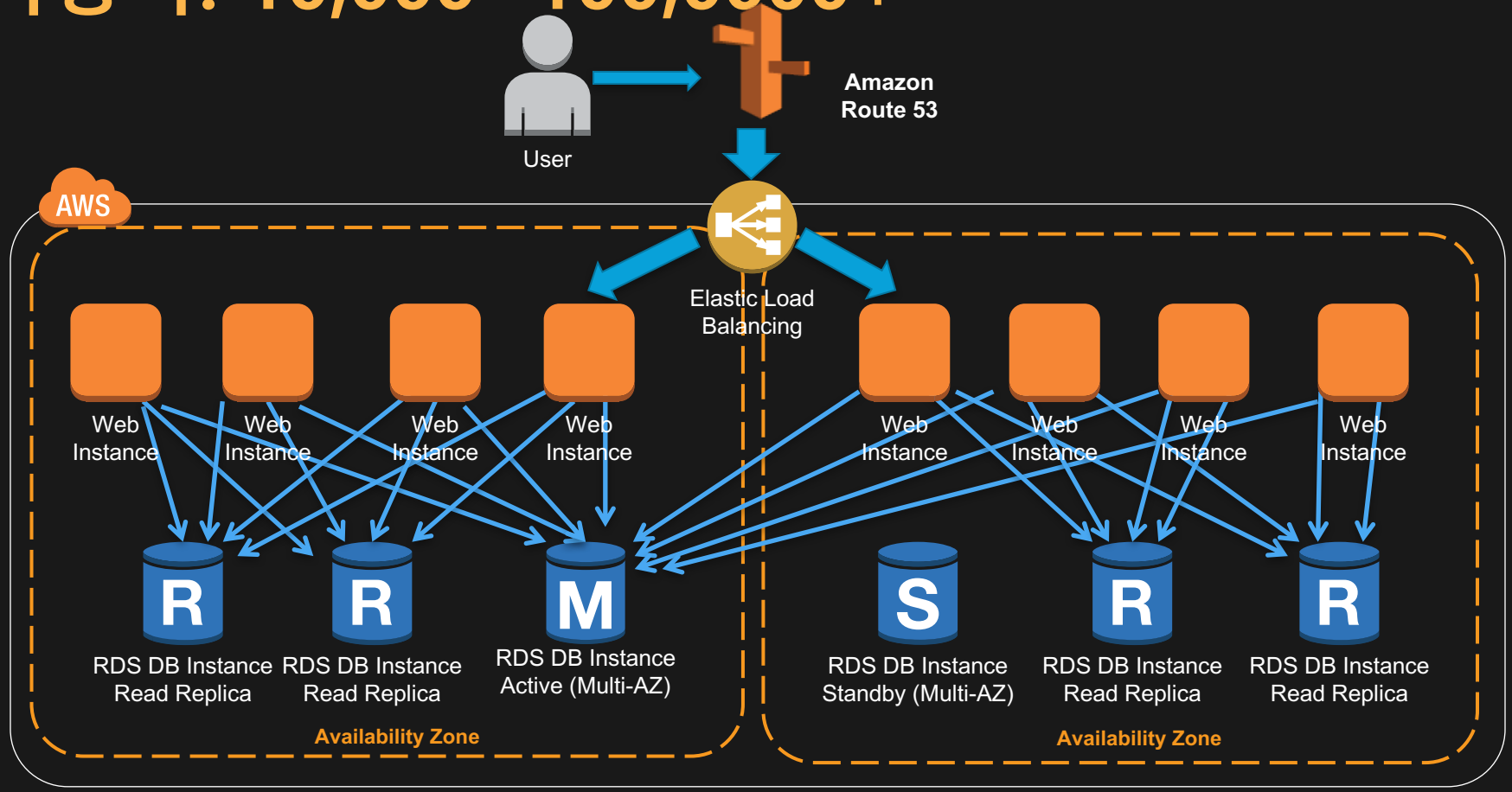
데이터베이스 이중화

RDS의 마스터-스탠바이를 Multi-AZ에 구성

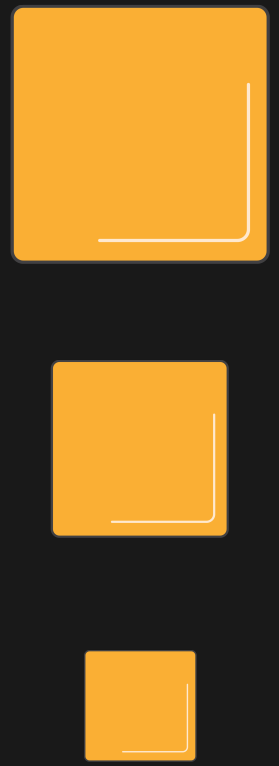



Users >10,000+

사용자: 10,000 -100,0000+



horizontally



vertically

높은 성능과 가용성을 위한
고급 클라우드 아키텍처

1. 성능을 위한 로드 분산

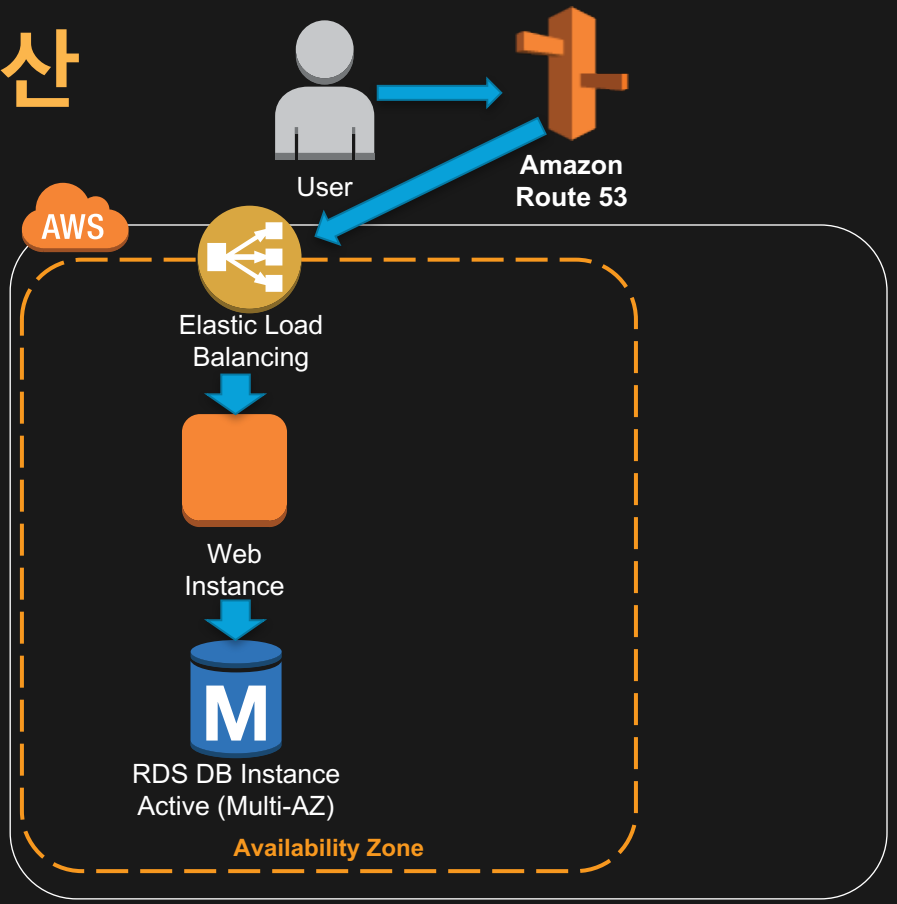
웹 서버 및 데이터베이스의
로드 분산

왜 해야 하나?

- 애플리케이션에 집중
- 가벼운 배포 가능
- 비용 절감 가능

무엇을?

- 정적 및 동적 콘텐츠
- DB에 부담되는 핫 아이템

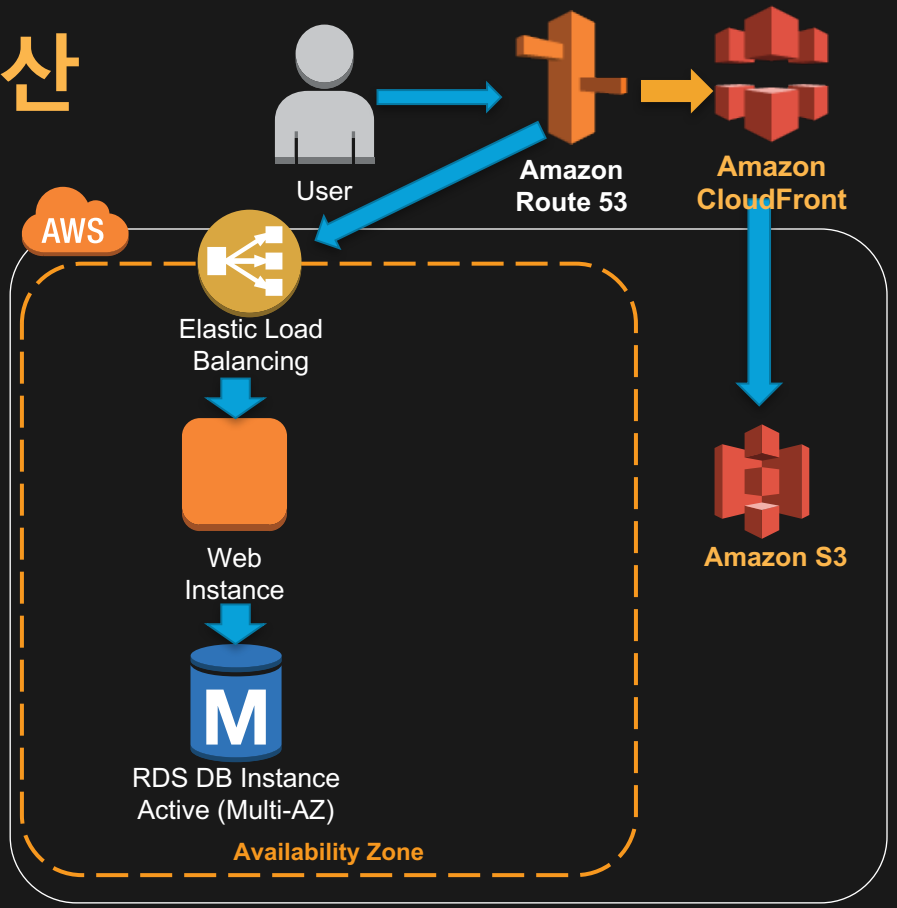


1. 성능을 위한 로드 분산

방법 1

정적 콘텐츠 Amazon S3 및 CloudFront로 이전

- CSS/JS 파일 및 사용자 업로드 이미지 등
- 무제한 저장소 및 콘텐츠 배포 네트워크 활용 가능

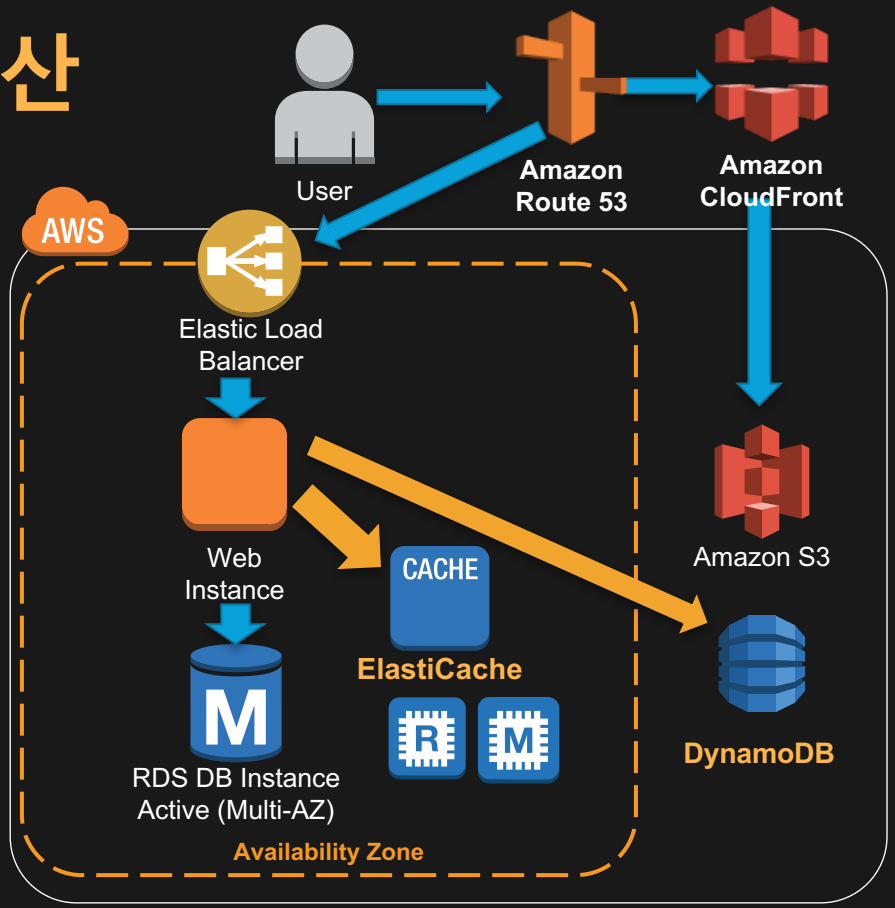


1. 성능을 위한 로드 분산

방법 2

Amazon ElasticCache 및 DynomoDB 활용

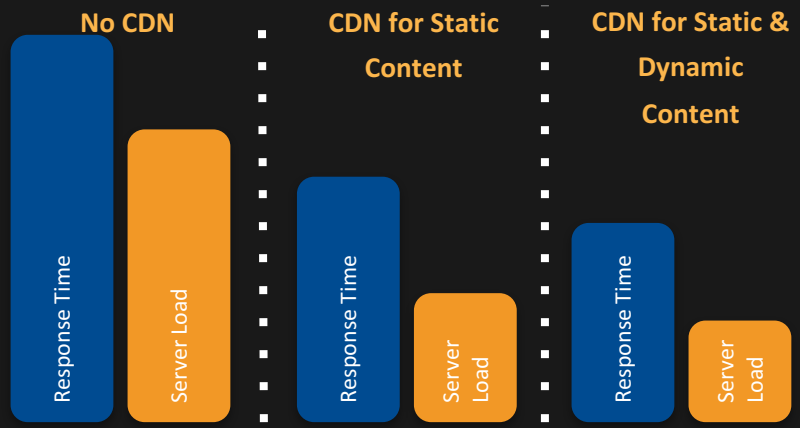
- 자주 사용하거나 업데이트가 잦은 데이터는 NoSQL 서비스나 인 메모리 기반 캐시 서비스 (Memcached, Redis) 활용

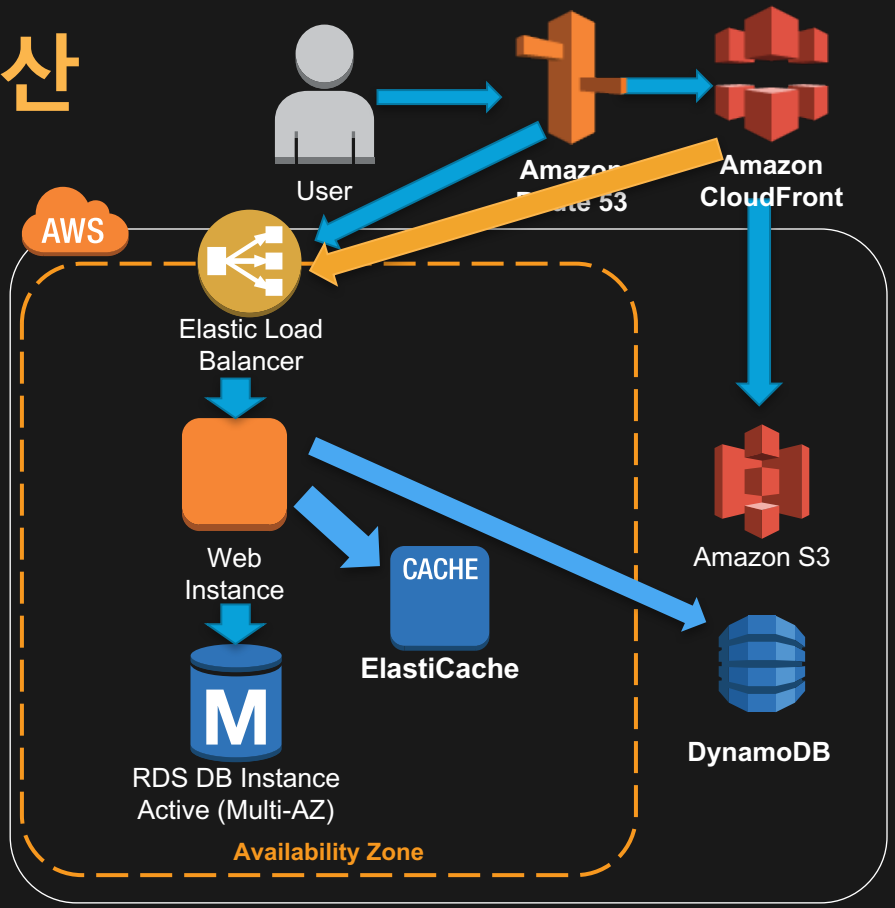


1. 성능을 위한 로드 분산

방법 3

동적 콘텐츠에 대해서도 Amazon CloudFront 활용





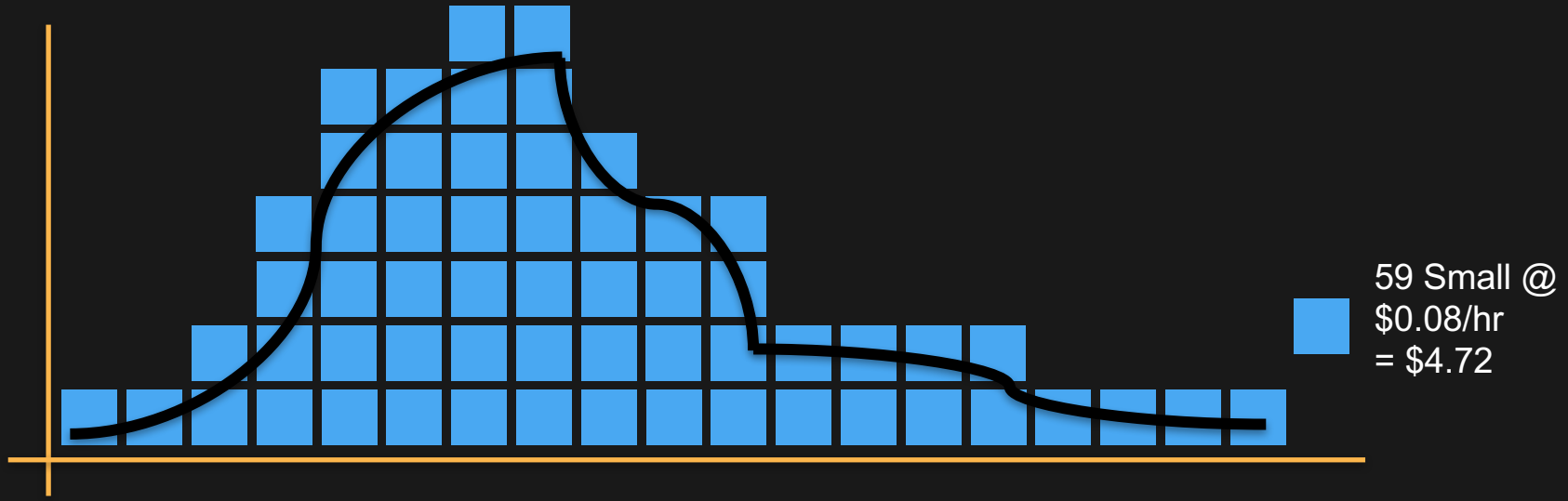
이제 가용성 높은 아키텍처를 위해

오토 스케일링을

적용해 봅시다!

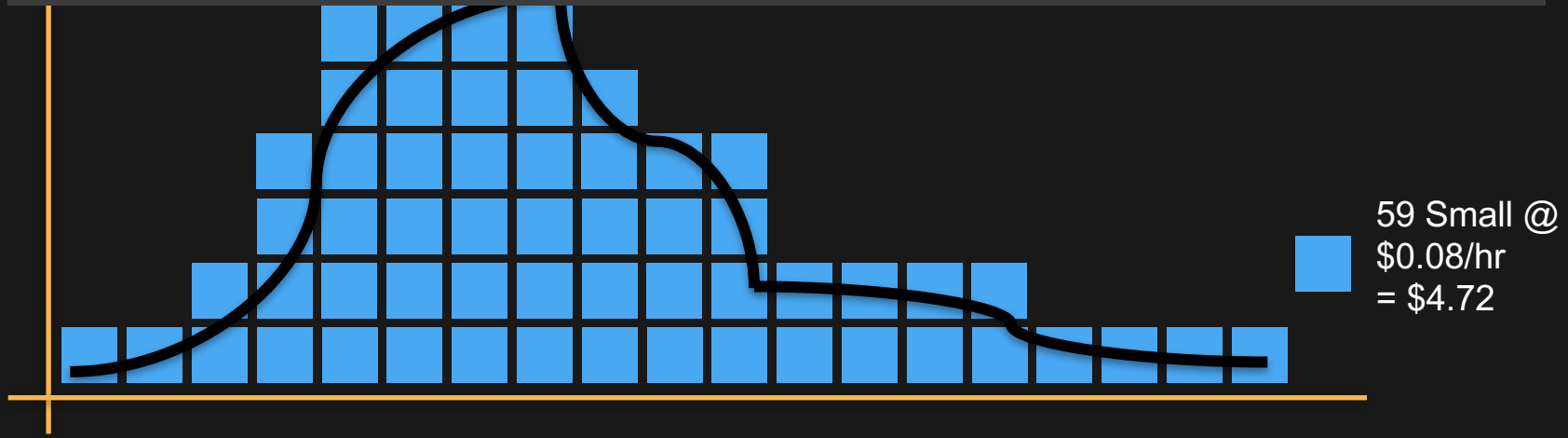
2. 가용성을 위한 오토스케일링

- ✓ CPU 사용률 혹은 네트워크 트래픽에 따라 자동 스케일-인/아웃
- ✓ 수요 곡선에 따라 인스턴스 사용 가능하므로 비용 절감 가능



2. 가용성을 위한 오토스케일링

```
$ aws autoscaling create-auto-scaling-group
 --auto-scaling-group-name MyGroup
 --launch-configuration-name MyConfig
 --min-size 1
 --max-size 10
 --availability-zones ap-northeast-2a, ap-northeast-2b
```

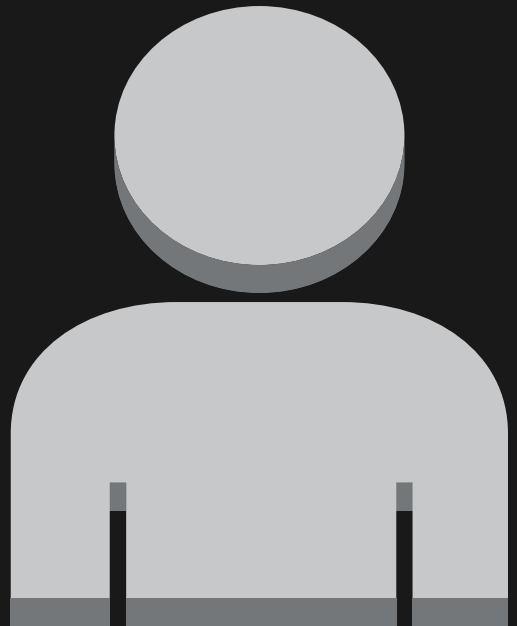


디스패치 오토 스케일링 사례

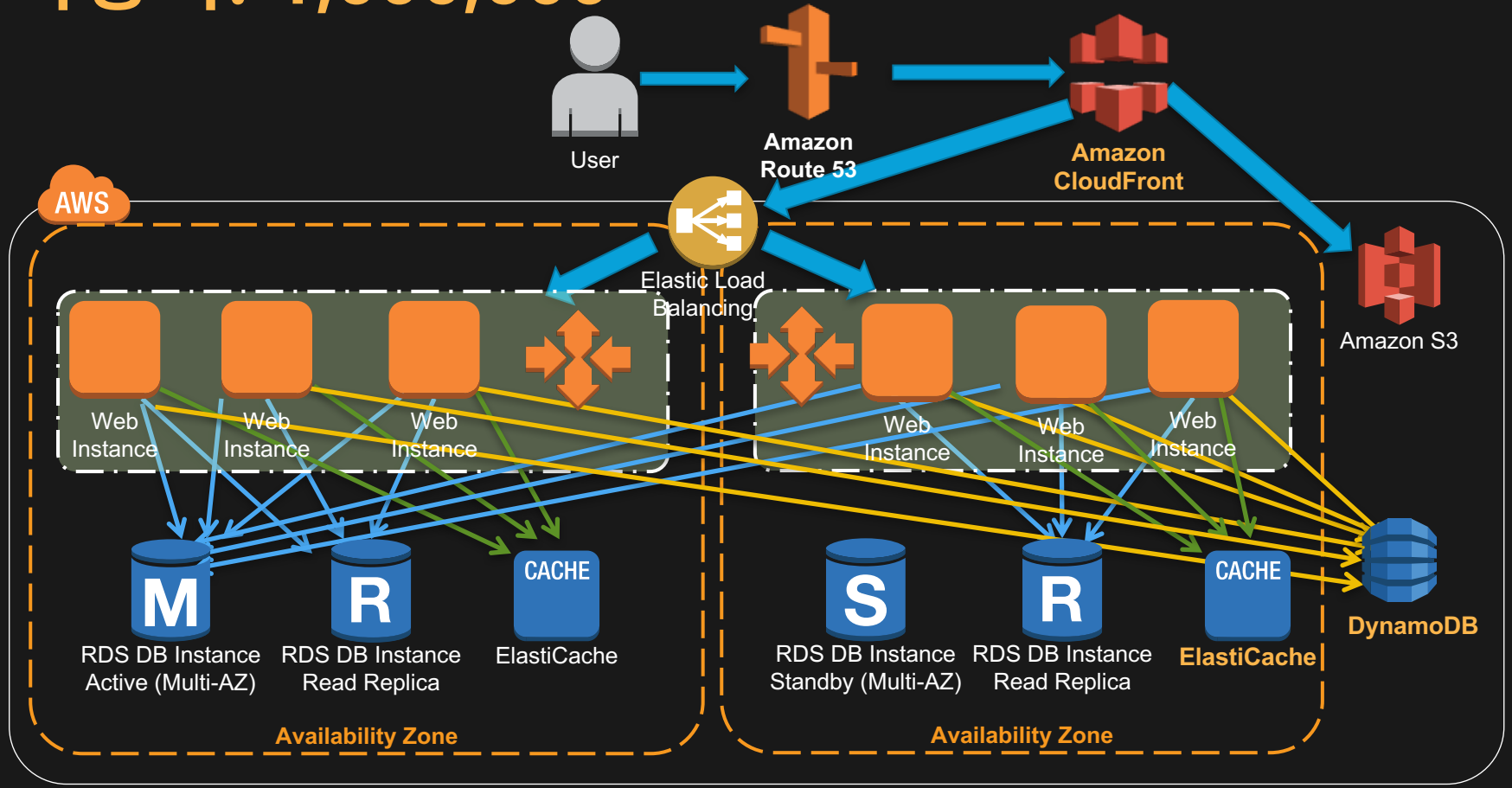
[단독] 강용석, 홍콩 스캔들 증거…그가 답해야 할 의혹 5

[단독] "우리 지금 만나"… 아이유·장기하, 뮤지션 커플의 탄생

Users >1,000,000



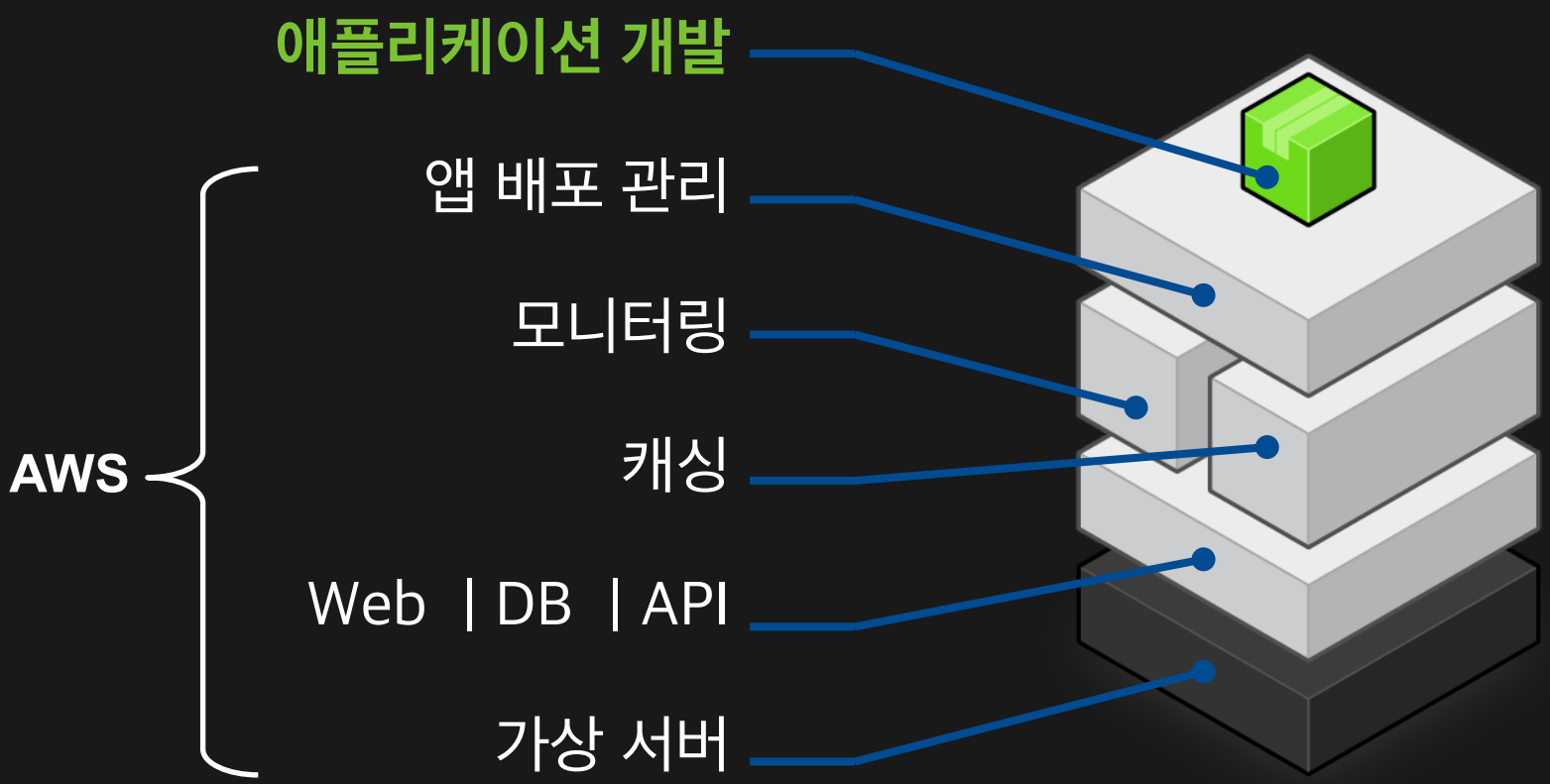
사용자: 1,000,000+


User
Amazon Route 53
Amazon CloudFront
AWS
Elastic Load Balancing
Amazon S3
Web Instance
Web Instance
Web Instance
Web Instance
Web Instance
Web Instance
M
R
CACHE
S
R
CACHE
RDS DB Instance Active (Multi-AZ)
RDS DB Instance Read Replica
ElastiCache
RDS DB Instance Standby (Multi-AZ)
RDS DB Instance Read Replica
ElastiCache
DynamoDB
Availability Zone
Availability Zone

비지니스에만 집중하기 위한

클라우드 네이티브 전략

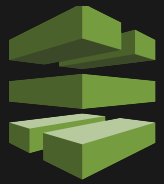
여러분이 필요한 것만 집중합시다!



1. 인프라 및 서비스 앱 배포 자동화

코딩 → 빌드 → 테스트 → 배포 → 구축 → 모니터링

AWS
CodeCommit



AWS
CodePipeline

AWS
CodeDeploy

AWS
CloudFormation

AWS
CloudWatch

AWS OpsWorks

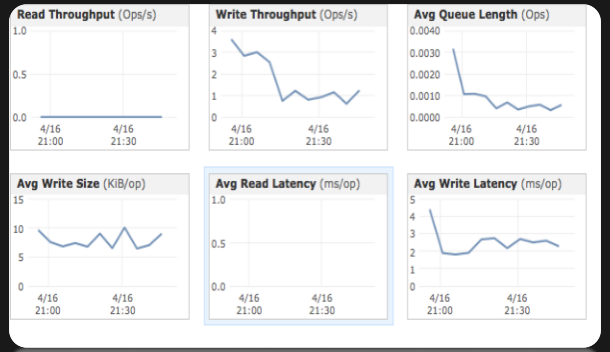
AWS Elastic Beanstalk

편리성

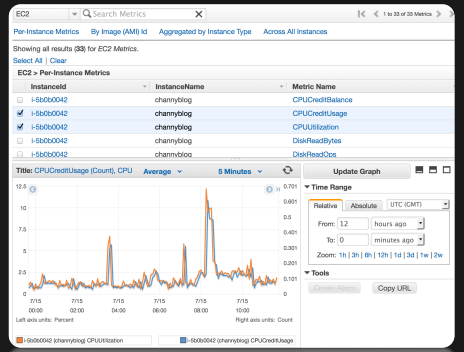
Demo: Amazon Elastic Beanstalk

2. AWS 자원 모니터링 및 로그 분석

AWS
CloudWatch



Host Level Metrics



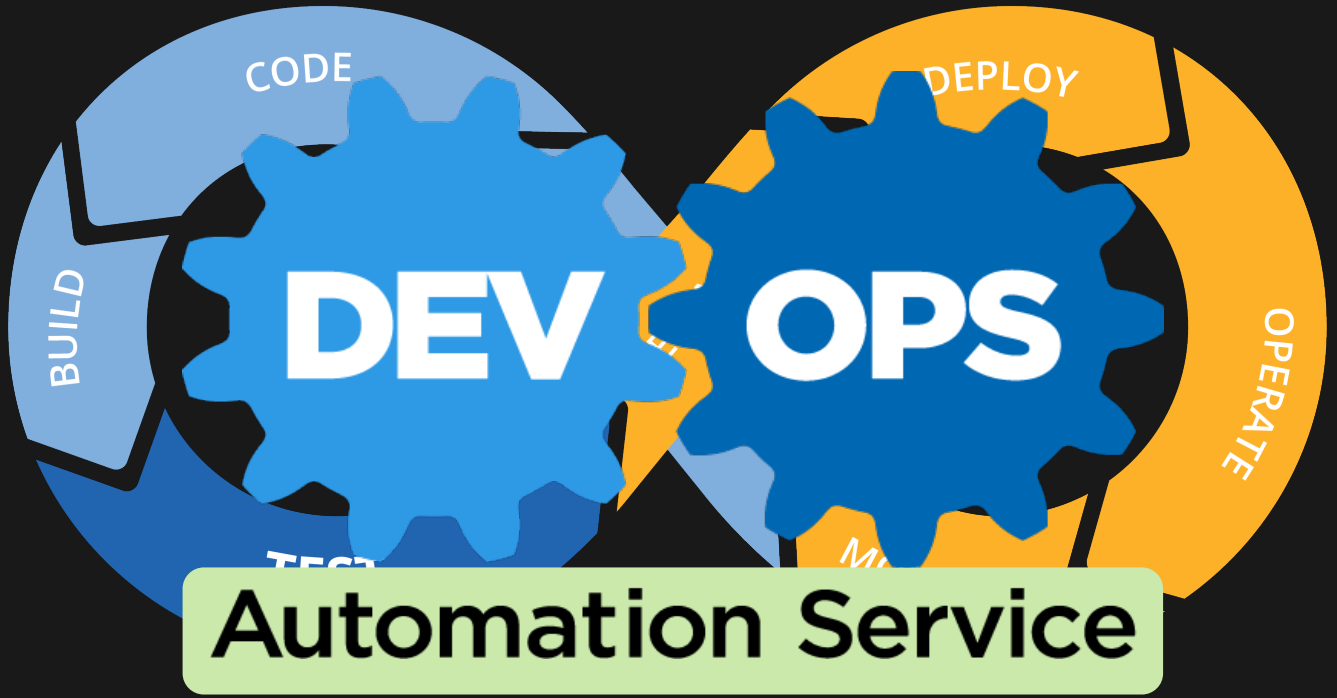
Aggregate Level Metrics

Amazon
Elasticsearch
Service

kibana

\+

서드파티
도구 활용


CODE
BUILD
DEPLOY
OPERATE
DEV
OPS
Automation Service

3. 서비스 재활용

AWS의 다양한 애플리케이션 서비스를 이용하여 개발 비용 감소!

Don’t reinvent the Wheel:

- ✓ 텍스트 검색 구현
- ✓ 메시지 큐 서비스 구현
- ✓ 대량 이메일 전송
- ✓ 워크플로 기반 구현
- ✓ 동영상 인코딩

3. 서비스 재활용

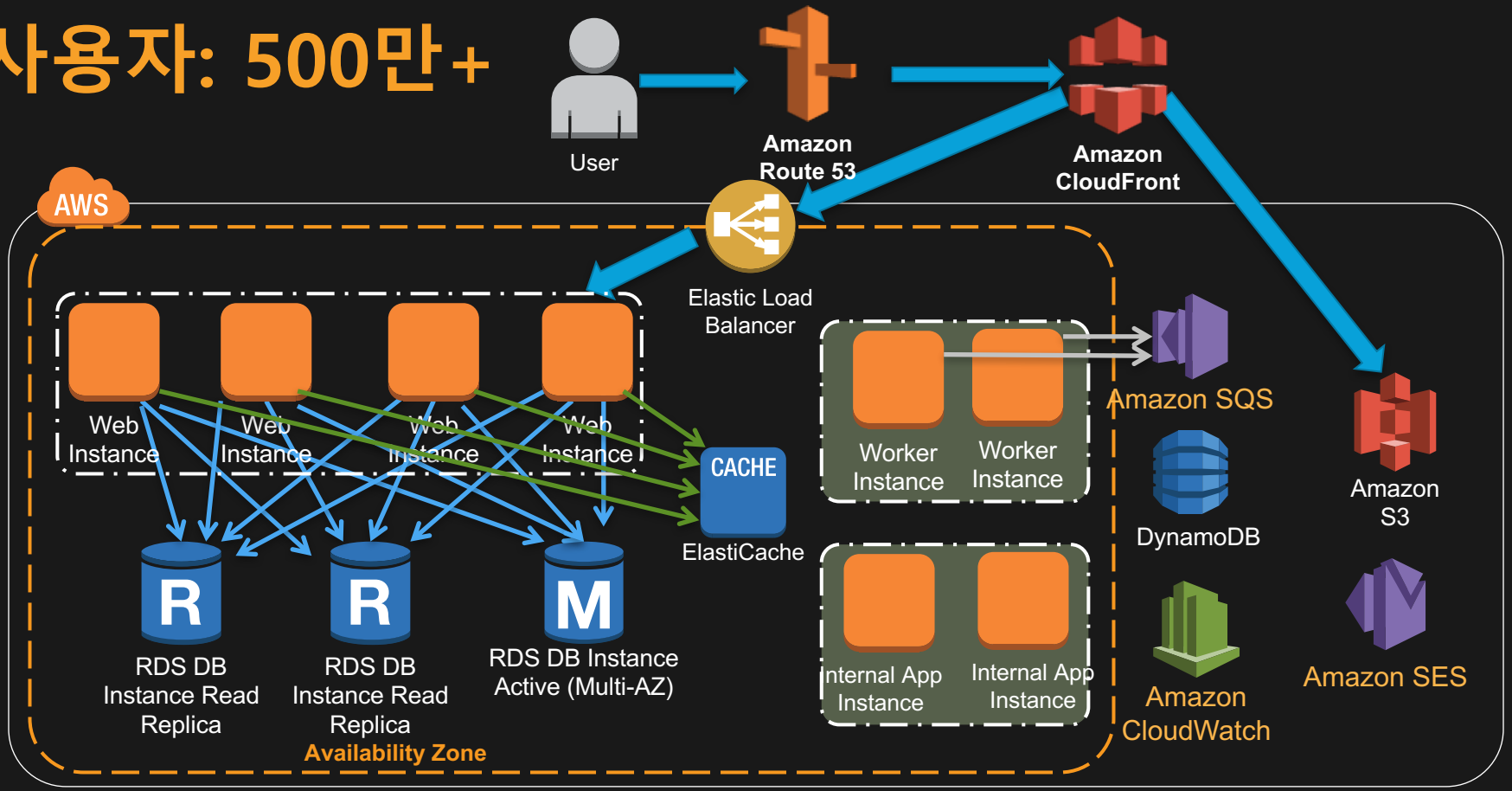
AWS의 다양한 애플리케이션 서비스를 이용하여 개발 비용 감소!

모바일도 또 만들지 말자:

- ✓ 소셜 로그인 및 모바일 인증
- ✓ 모바일 앱 데이터 분석
- ✓ 푸시 노티피케이션
- ✓ 모바일 앱 테스트
- ✓ 빠른 모바일 앱 개발
- ✓ IoT 기기간 통신 및 AWS 연동

Users >5,000,000

사용자: 500만+


User
Amazon Route 53
Amazon CloudFront
AWS
Elastic Load Balancer
Web Instance
Web Instance
Web Instance
Web Instance
CACHE
ElastiCache
Worker Instance
Worker Instance
Amazon SQS
DynamoDB
Amazon S3
R
R
M
RDS DB Instance Read Replica
RDS DB Instance Read Replica
RDS DB Instance Active (Multi-AZ)
Internal App Instance
Internal App Instance
Amazon CloudWatch
Amazon SES
Availability Zone

고객 사례: 비트앱

Mobile Client

Signed-URL

Route 53

CloudFront

Origin-Access Identity

AWS Tokyo region

AWS

Amazon S3
Static Contents
& Logs

DynamoDB

Ansible

CloudWatch

Redshift

Elastic Load Balancing

Availability Zone #1

API Servers

Workers & MQ

ElastiCache

RDS DB

Availability Zone #2

API Servers

Workers & MQ servers

ElastiCache

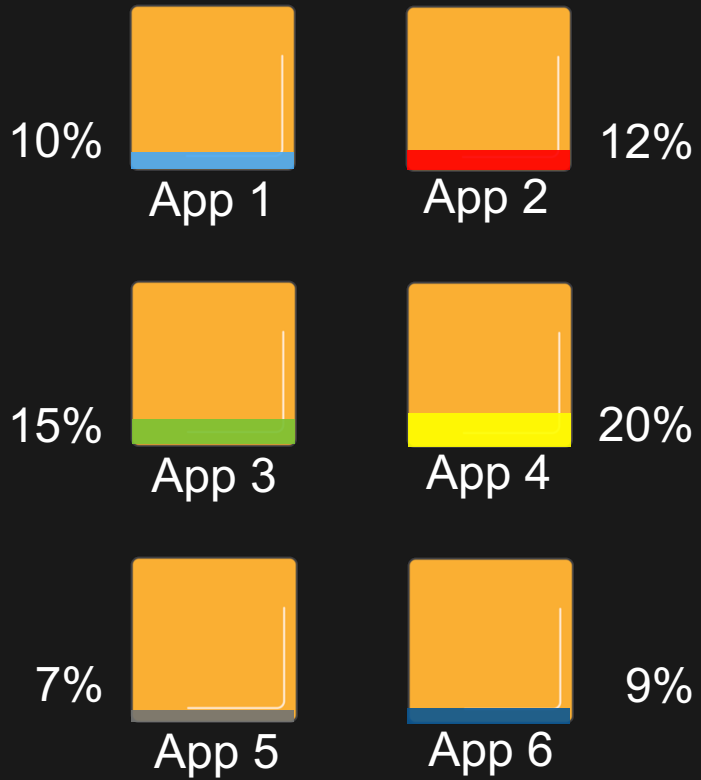
RDS (Multi-AZ)

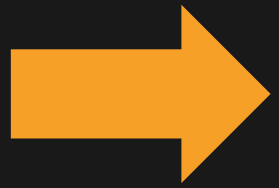
THE BEATPACKING COMPANY

Dr. Werner Vogels, Amazon.com CTO

re:Invent 2015 Keynote

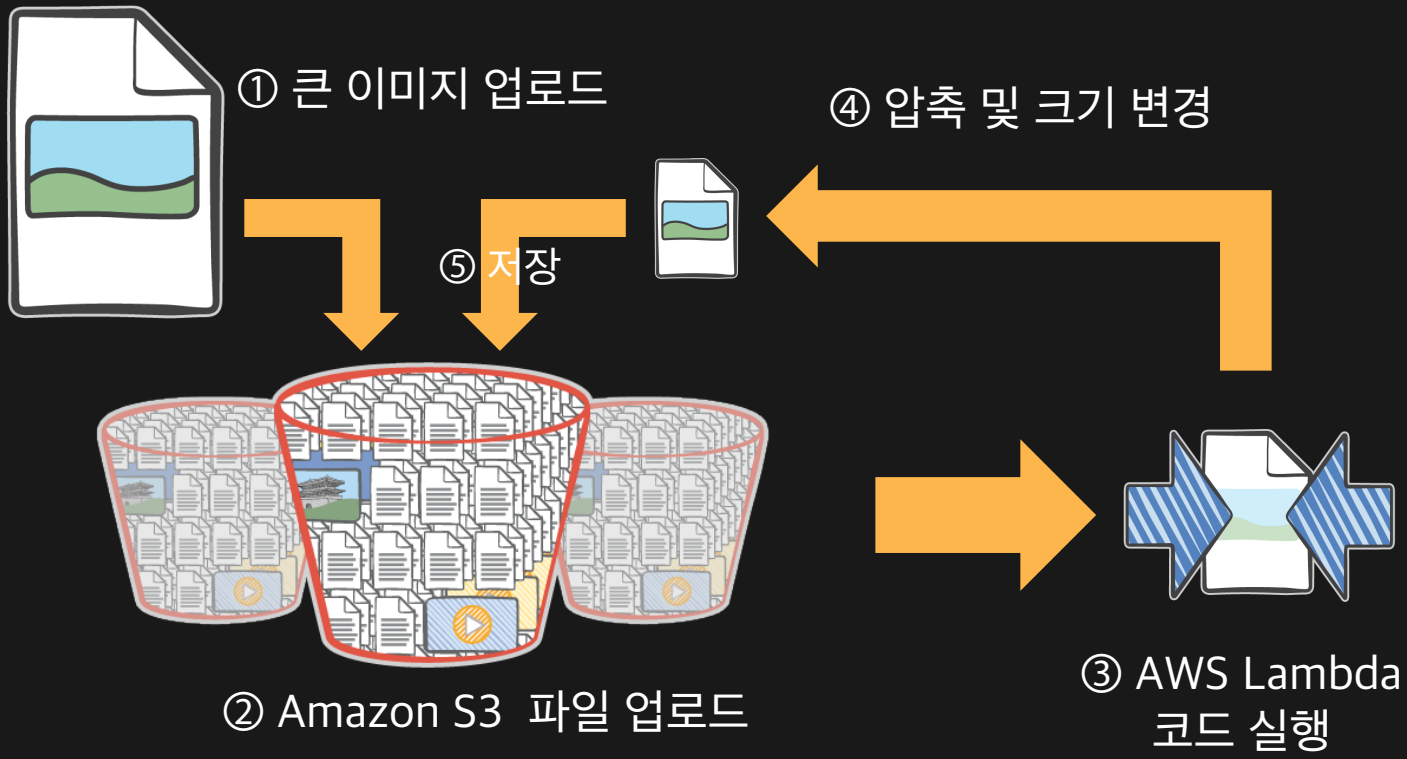
1. 콘테이너를 통한 EC2 인스턴스 효율화





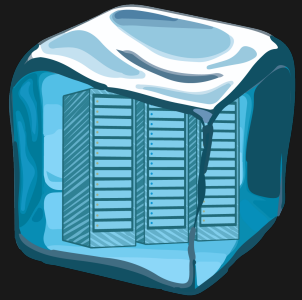
Amazon EC2
Container Service

2. 이벤트 기반 AWS Lambda 함수 실행



3. 클라우드 컴퓨팅 기술의 변화

Weeks



On-Premises

Minutes

Amazon EC2

Seconds

Amazon EC2 Containers Services

Milliseconds

AWS Lambda

4. 마이크로서비스(Microservices)

- 큰 서비스를 작게 분리하여 느슨하게 연결
  (Loosely Coupling)
- 작은 서비스간 API로 인터페이스 및 통신
  (Bounded Context)
- 자율적이고 책임 있는 팀으로 기술 변화에 빠르게 대응 가능
  (DevOps)

Serverless Architecture 빌딩 블럭

Amazon API Gateway

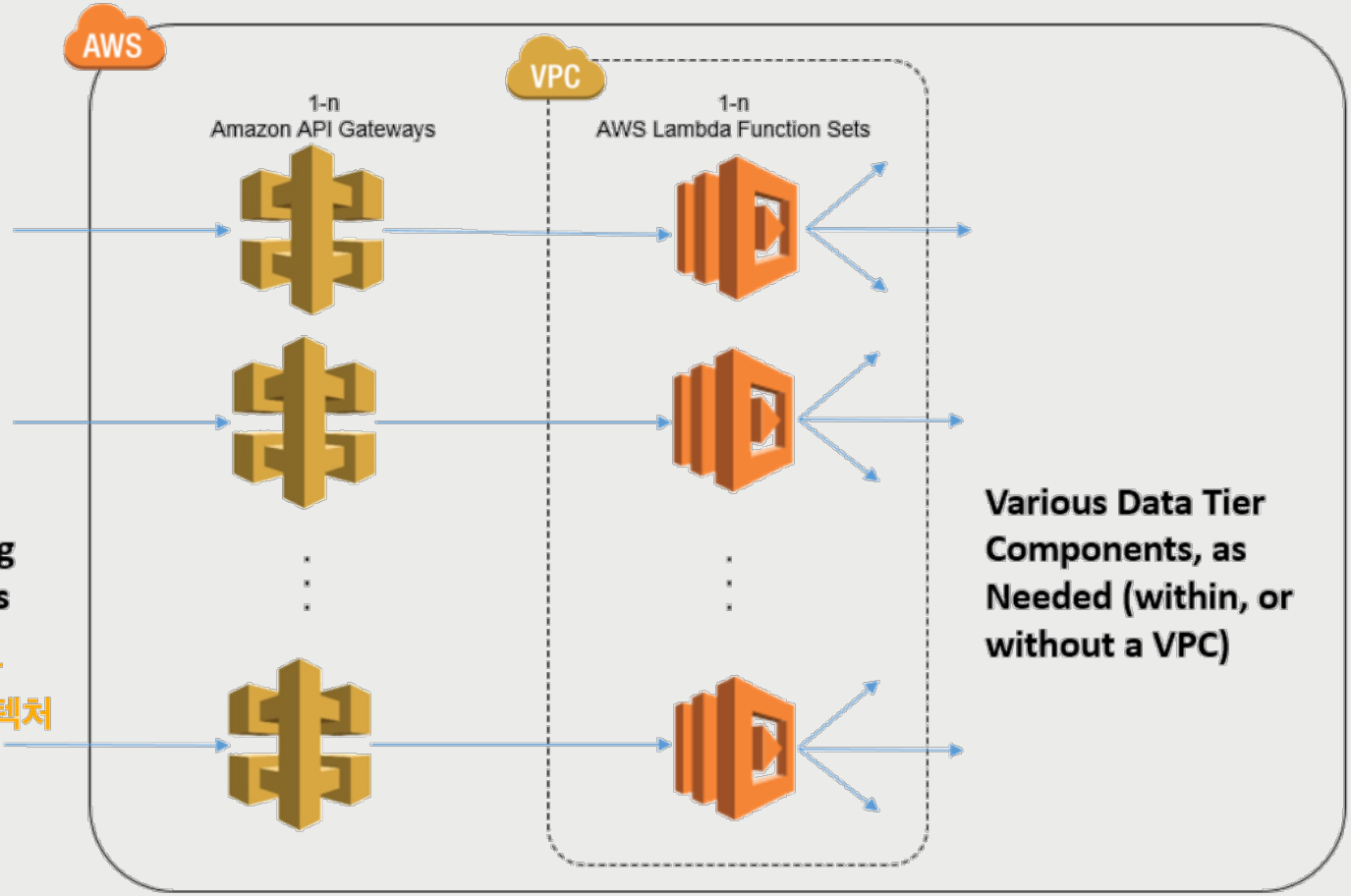
서비스간 API
인증/캐싱 용이

AWS Lambda

이벤트를 기반한
코딩 함수 실행

Amazon Simple Queue Service

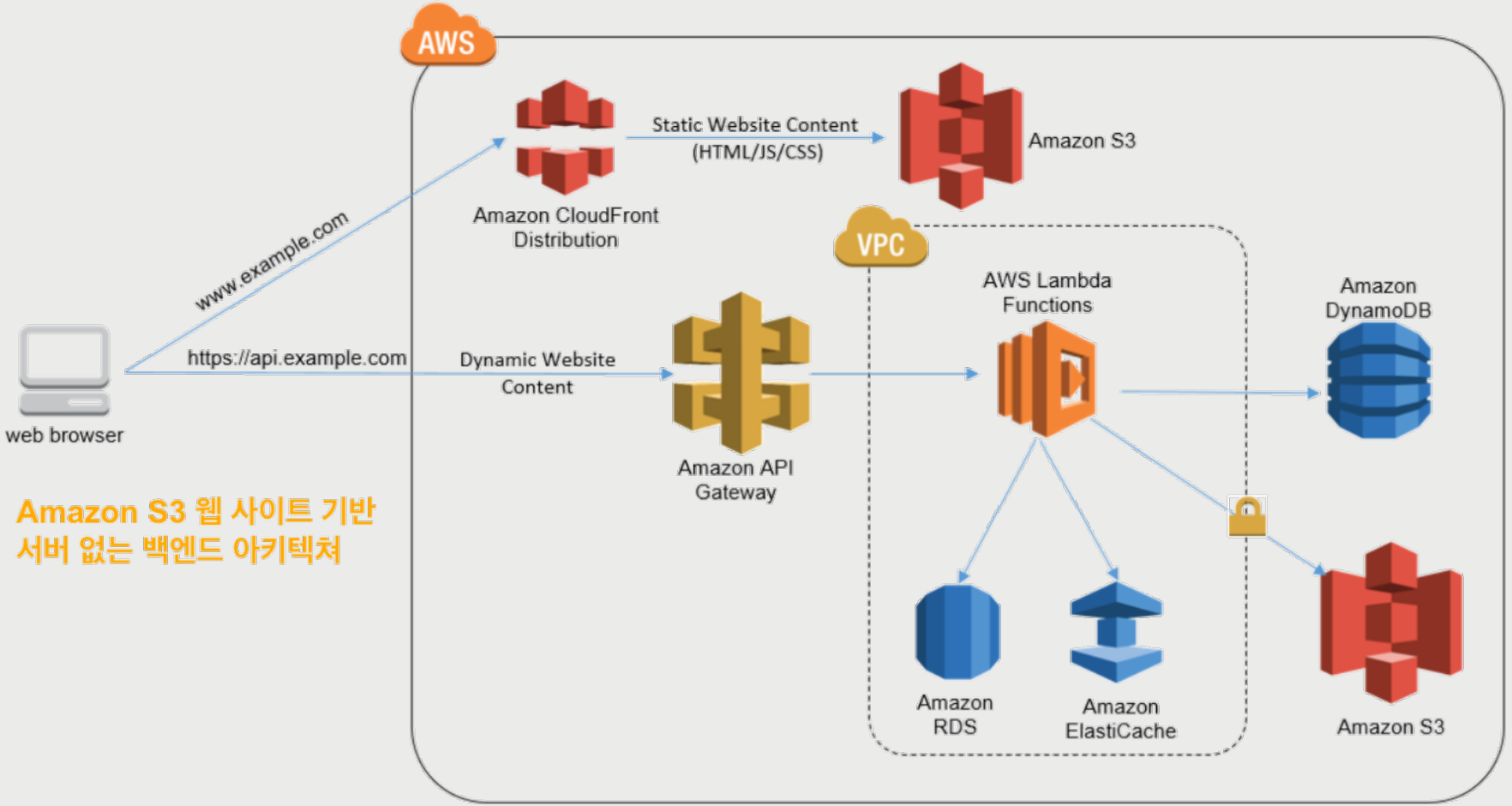
서비스간
메시지 큐 관리



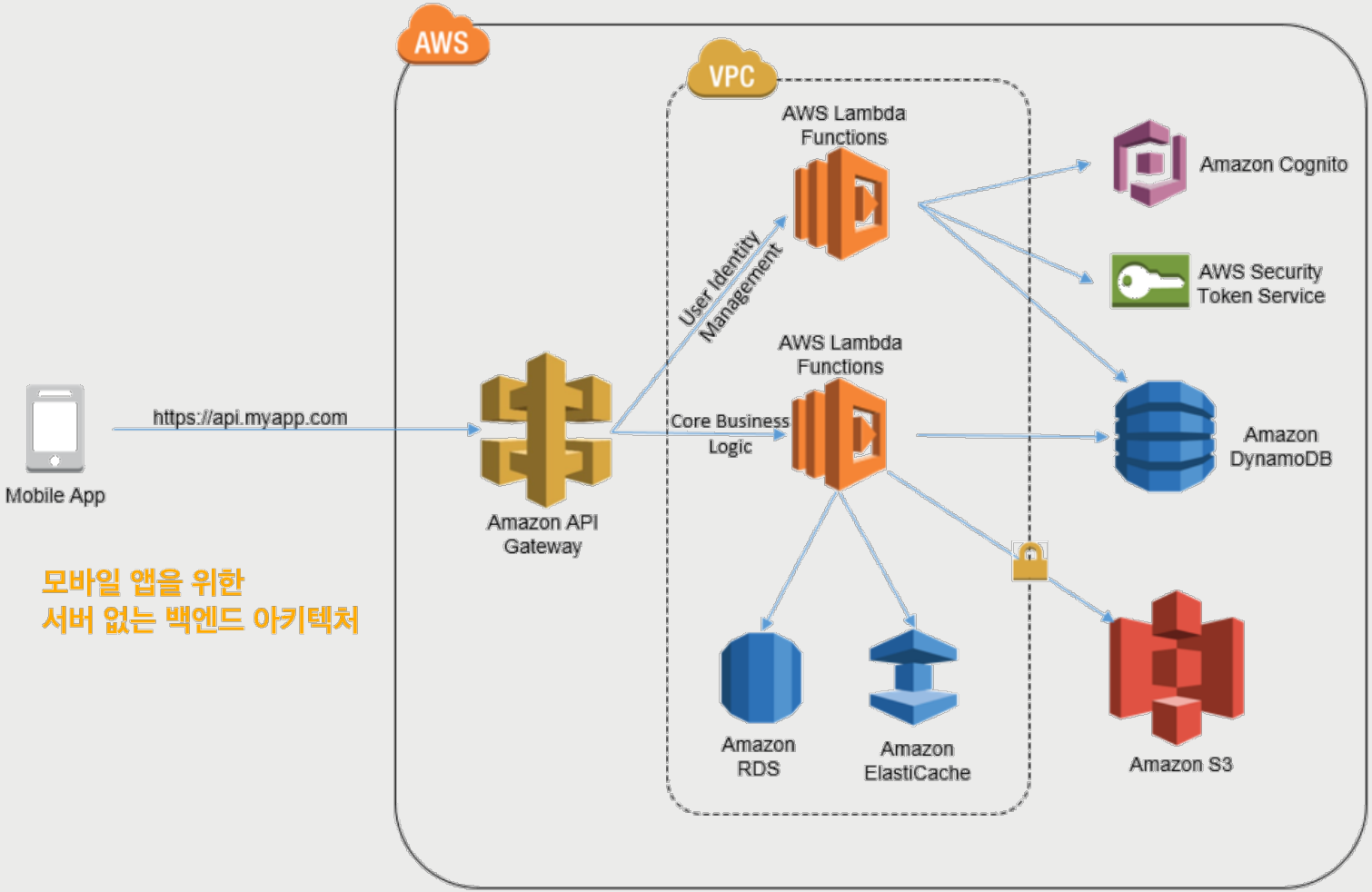
Various Clients, Potentially Including Other Microservices

마이크로서비스를 위한 서버 없는 백엔드 아키텍처

Various Data Tier Components, as Needed (within, or without a VPC)

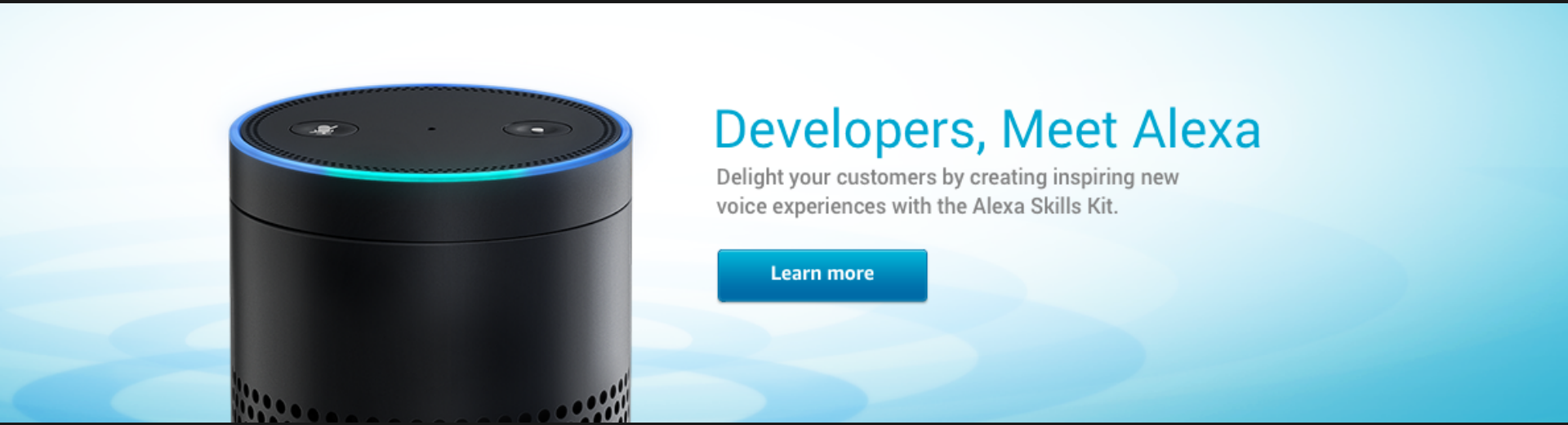


Amazon S3 웹 사이트 기반
서버 없는 백엔드 아키텍쳐



모바일 앱을 위한
서버 없는 백엔드 아키텍처

Demo: API Gateway + AWS Lambda + Alexa Skillt Kit



Users >10,000,000

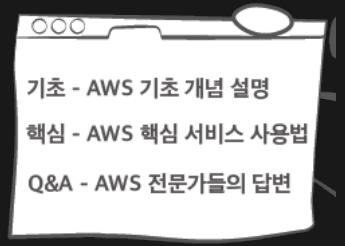
천만 사용자를 위한 십계명

1. Multi-AZ 아키텍처로 고가용성 확보
2. 부하 분산을 통해 성능 극대화
3. 아키텍처 전 영역에서 증설 감안
4. 내외부의 캐시를 적극 활용
5. 오토스케일링 사용, 두번 사용!

천만 사용자를 위한 십계명

5. DevOps를 통해 운영을 자동화
6. 한 곳에서 모니터링/통계/분석
7. 직접 만들지 말고 AWS 서비스 활용
8. 애플리케이션 성능 튜닝 필요
9. 마이크로 서비스로 전환
10. **클라우드 네이티브- 돈과 시간 절약**

What's Next?



AWS 온라인 세미나

http://bit.ly/awskr-webinar

AWS 기초 동영상 강의

https://opentutorials.org/module/1946

What's Next?

무료 온라인 실습 하기

EC2/ELB/RDS/S3/CloudFront/AutoScaling/Beanstalk

https://www.qwiklab.com/lab_catalogue

What's Next?

"아마존 웹 서비스" 검색 후

"AWSKRUG" 검색 후

한국 사용자모임 커뮤니티 가입

더 궁금한 점은...

"윤석찬" 검색 후