



AWS Cloud Day in Busan



AWS Cloud Day in Busan



천만 사용자를 위한
AWS 클라우드 아키텍처 진화하기

Donghyun Lee

Public Sector Solutions Architect
Amazon Web Services



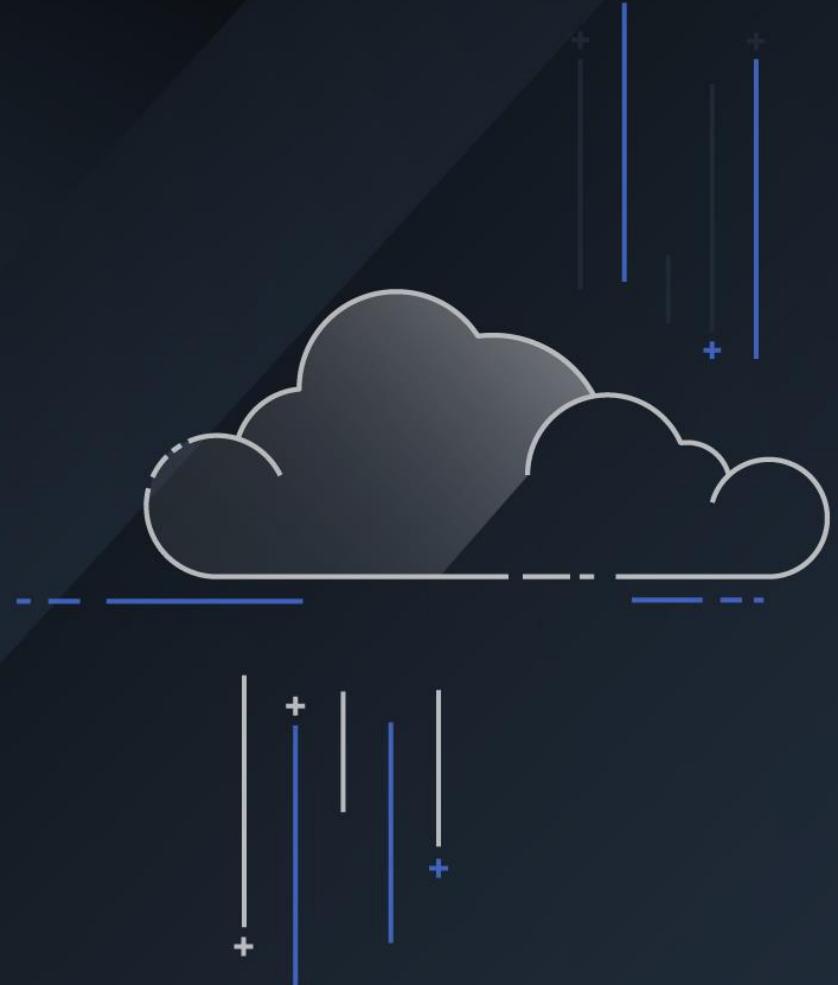
사용자 증가에 따른 클라우드 아키텍처 진화

- 수직적 스케일 업을 통한 확장
- 멀티 AZ 기반 수평적 확장과 부하 이동
- 오토 스케일링
- 모니터링 및 자동화
- 마이크로 서비스

AWS Cloud Day in Busan

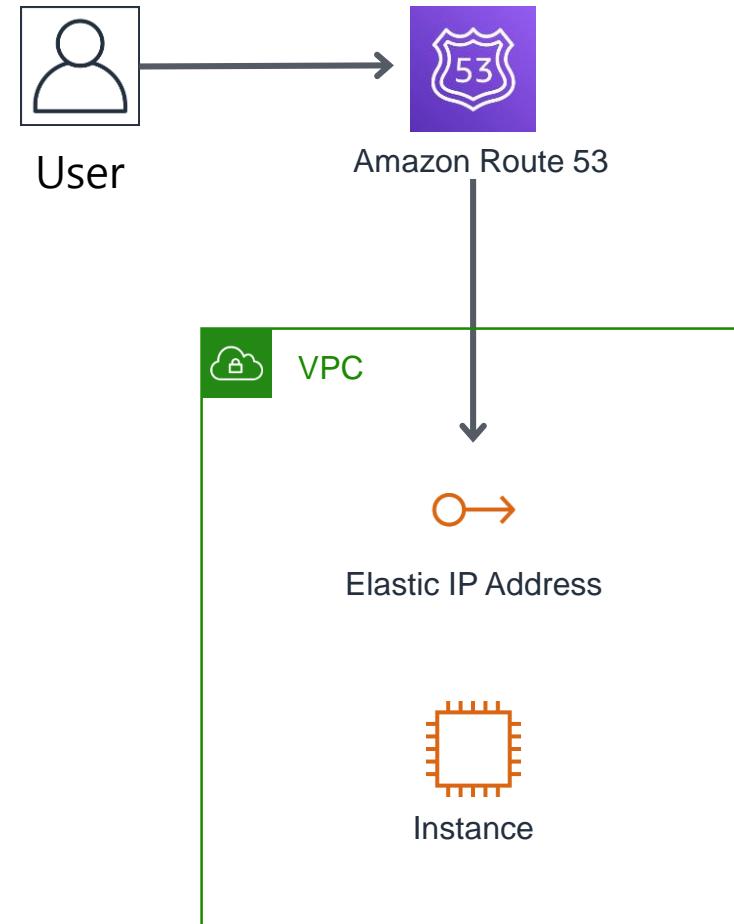


천만 사용자를 위한 아키텍처
하나씩 발전시켜 봅시다

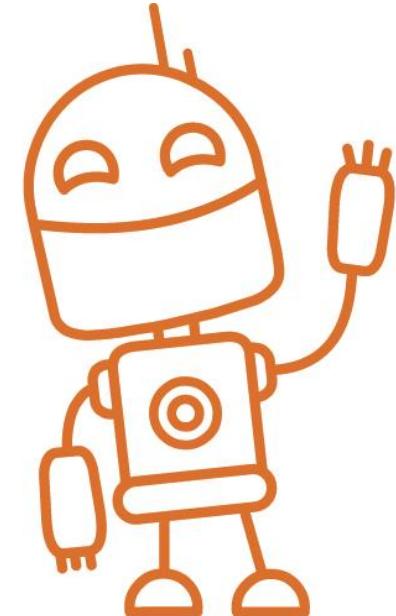


사용자 수 = 1

- 페일오버 필요 없음
- 이중화 필요 없음



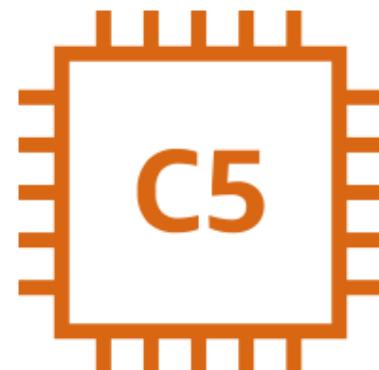
- 미리 설치된 컴퓨팅, 스토리지, 네트워크를 포함하는 7개의 계획에서 선택
- 저렴하고 예측가능한 가격
- 수 초 안에 완전구성된 서버 생성
- 직관적인 Lightsail 콘솔에서 관리
- AWS 서비스들과 접목해서 확장
- Lightsail API 와 AWS CLI 로 자동화



가장 단순한 접근

- 더 높은 CPU, 메모리, I/O 인스턴스 타입
- EBS PIOPS를 이용 고려
- 인스턴스 사이즈를 변경하기 쉬움

언젠가는 결국 한계에 봉착



c5.9xlarge

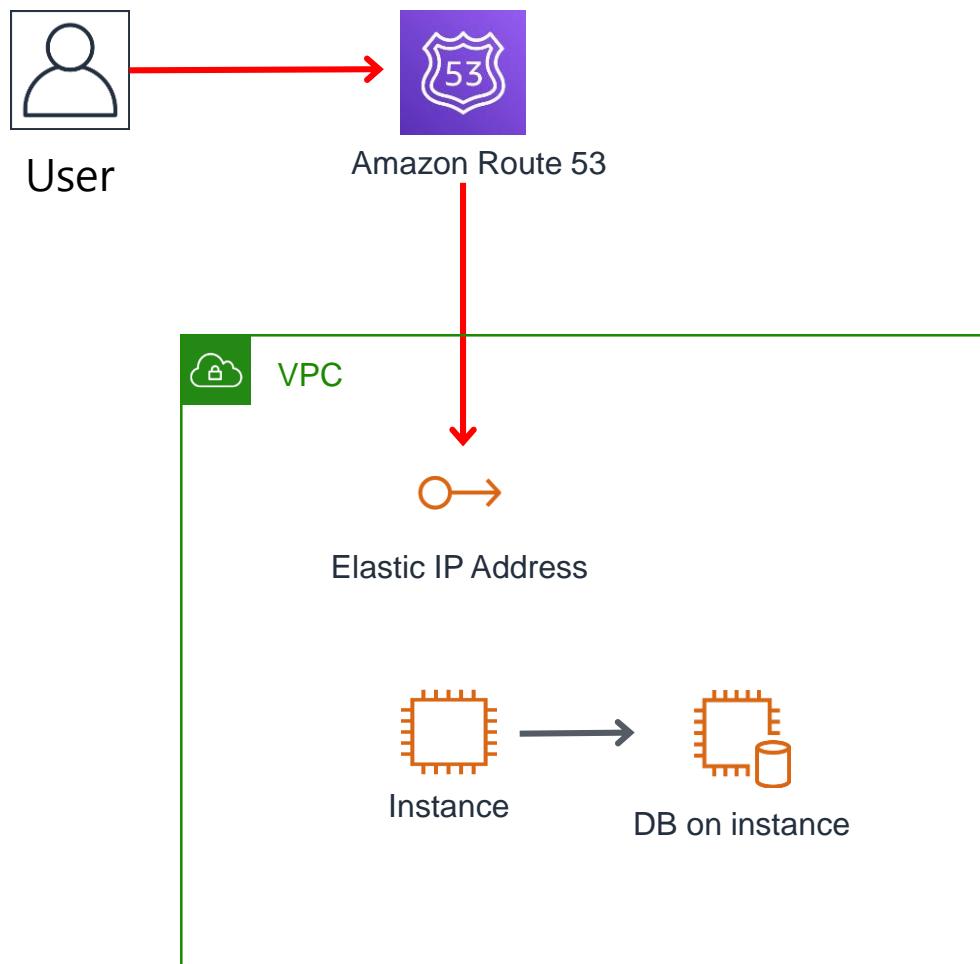


m5.2xlarge

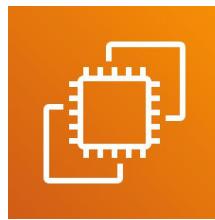


t3.nano

사용자 수 > 1



Self-managed



Amazon EC2



Amazon RDS

Fully managed



Amazon DynamoDB



Amazon
Neptune



Amazon DocumentDB
(MongoDB 호환)

가장 많이 선호하는 데이터베이스 엔진을 갖춘 관계형 데이터베이스

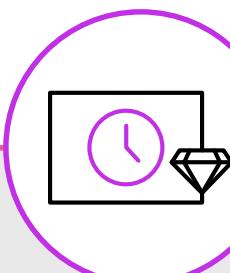


관리 용이성



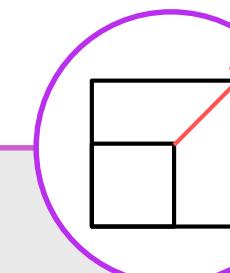
하드웨어 준비, 소프트웨어 설치
등의 관리 작업 불필요

가용성 및 내구성



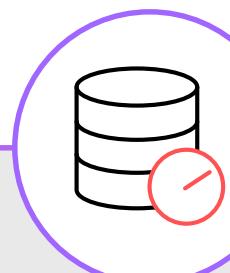
다중 AZ를 통한 동기식 복제,
자동화된 백업, 스냅샷, 장애 조치

뛰어난 확장성



몇 번의 클릭만으로도
다운타임 없이
컴퓨팅 및 스토리지 확장

빠른 성능과 보안



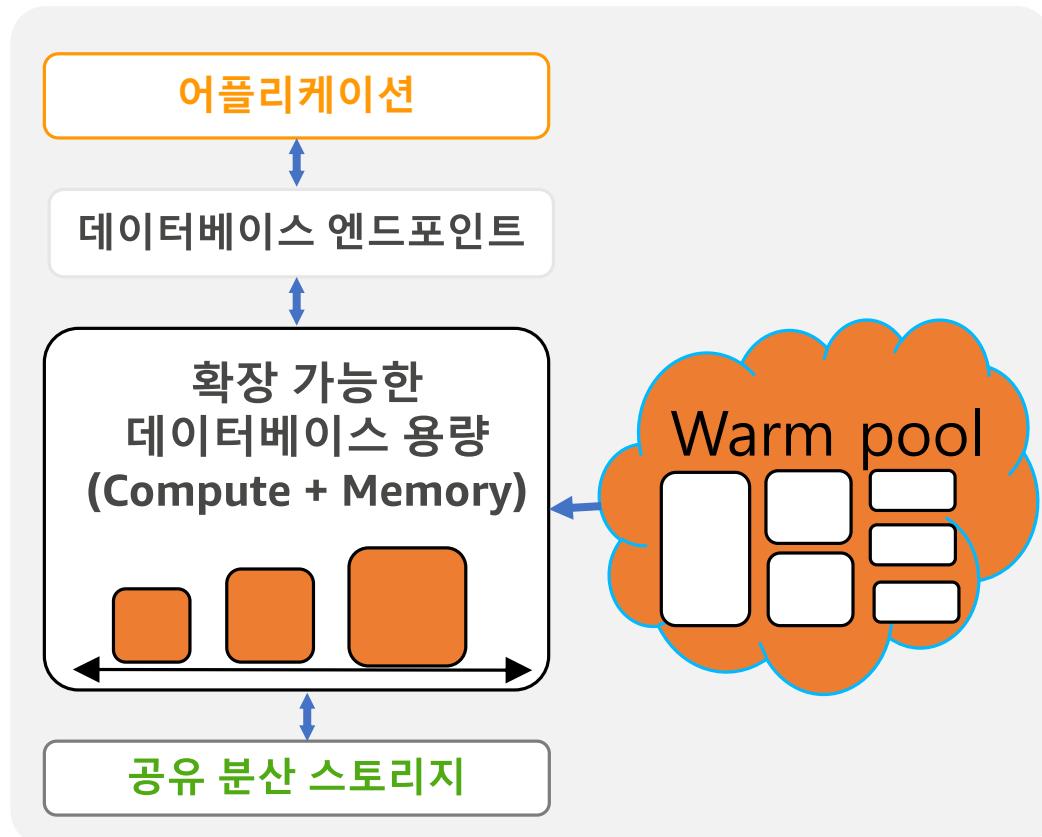
SSD 스토리지 및 성능 향상된
I/O 보장,
저장 및 전송 중 암호화 지원



Amazon Aurora

- MySQL 또는 Postgres 호환
- 자동으로 스토리지 확장 (최대 64 TB)
- 최대 15 개의 읽기 전용 복제본
- Amazon S3로 지속적인(증분) 백업
- 3개의 AZ에 6개의 데이터 복제
- Serverless 옵션

- 가변적인 업무를 위한 온디맨드 & 오토 스케일링이 가능한 DB



필요할 때 바로 시작하고 불필요시 바로 셧다운

자동으로 인스턴스 확장

실제 사용 데이터베이스 용량(ACU) 초단위로 과금

new PostgreSQL 호환 지원

NoSQL을 쓸 것인가 말 것인가?

AWS Cloud Day in Busan



먼저 SQL 데이터베이스부터
시작해봅시다

- 시장에서 잘 알려진 기술
- 유용한 코드, 커뮤니티, 책, 도구들
- 수백만 사용자 수를 확보하더라도, SQL DB는 유용
- 확장성을 위해 명확한 패턴들

데이터가
정말
많다구요?



**MY DATA IS
HUGE**

memegenerator.net

연간 데이터 크기 > 5 TB

많은 부하에 데이터까지 크다구요?

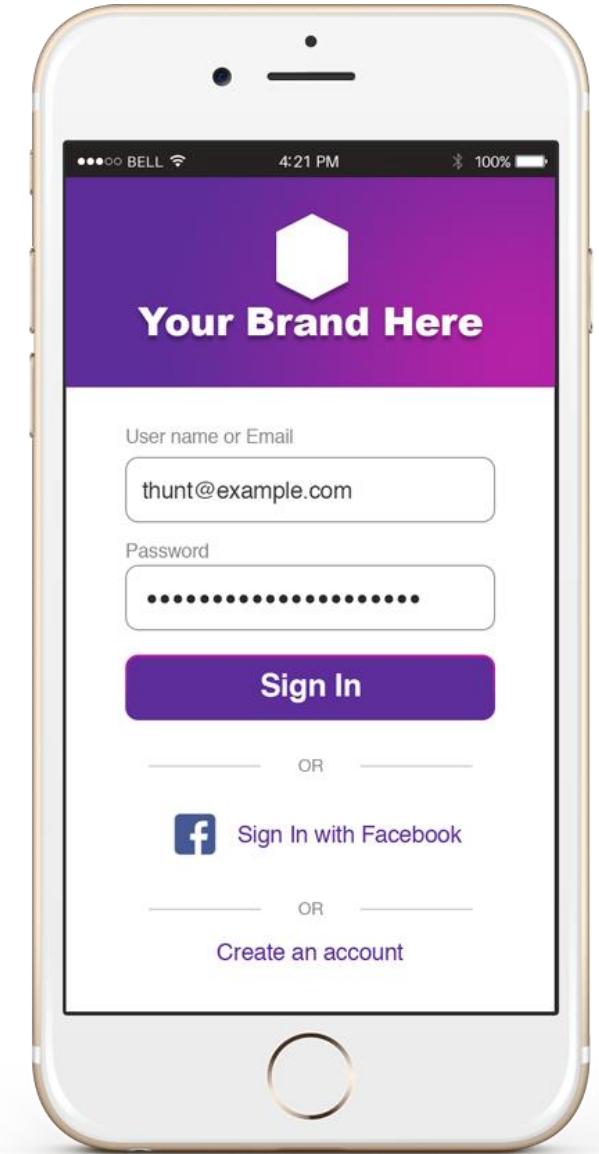
OK!

이제 NoSQL이 필요할 것 같네요

- 굉장히 낮은 자연시간을 요구하는 어플리케이션
- 메타데이터 지향 데이터 셋트
- 비-관계적인 데이터
- 스키마-리스 데이터 구조가 필요(스키마가 없을 때 더 개발하기 쉬움)
- 빠른 데이터 수집 (초당 수천 건의 레코드)
- 대량의 데이터가 발생 (TB단위)

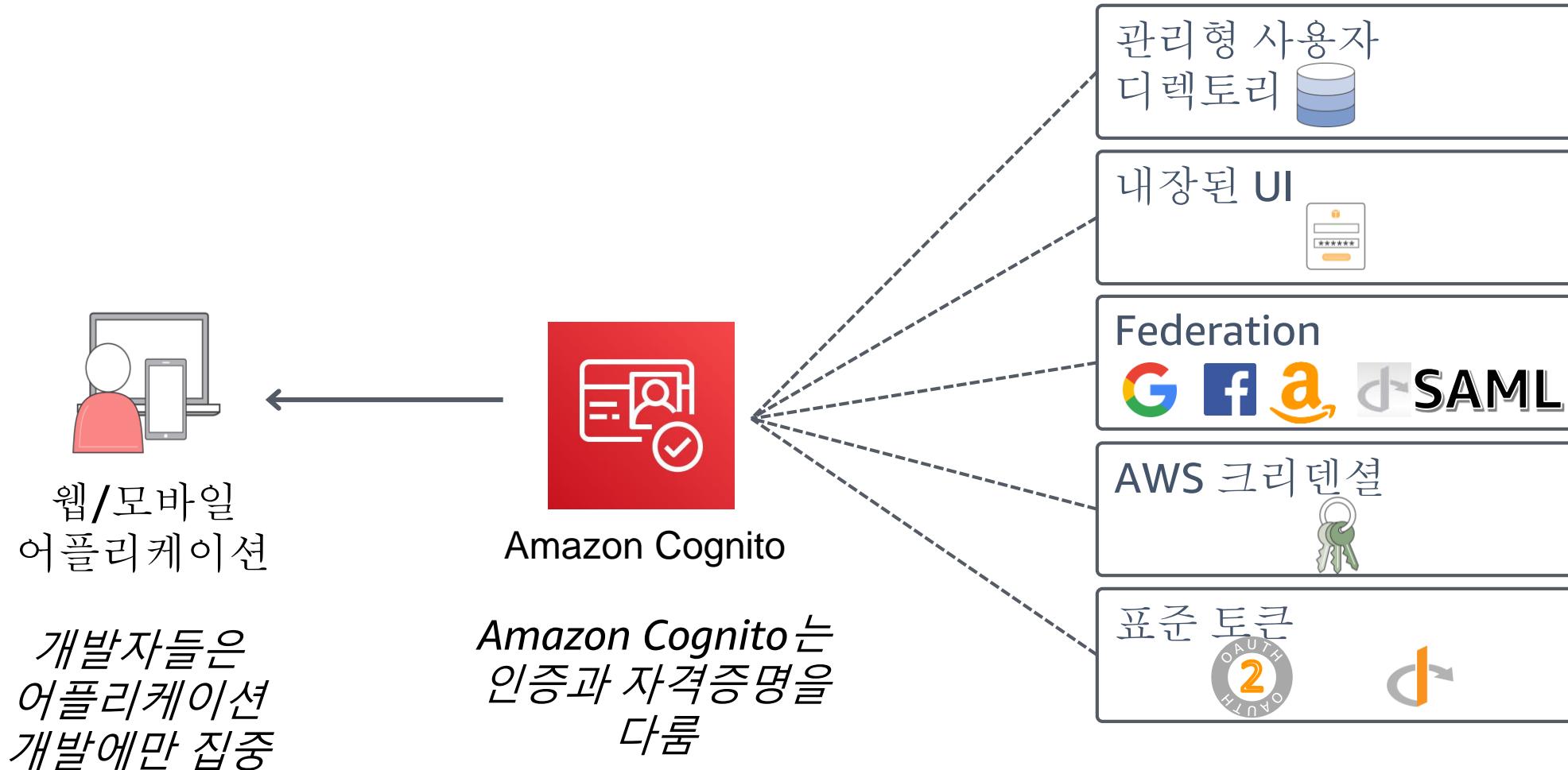
사용자 수 > 1

등록, 로그인 등등



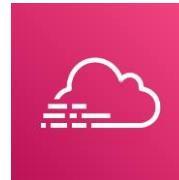
Amazon Cognito 가 있습니다!

AWS Cloud Day in Busan





AWS Identity and Access
Management (IAM)



AWS CloudTrail



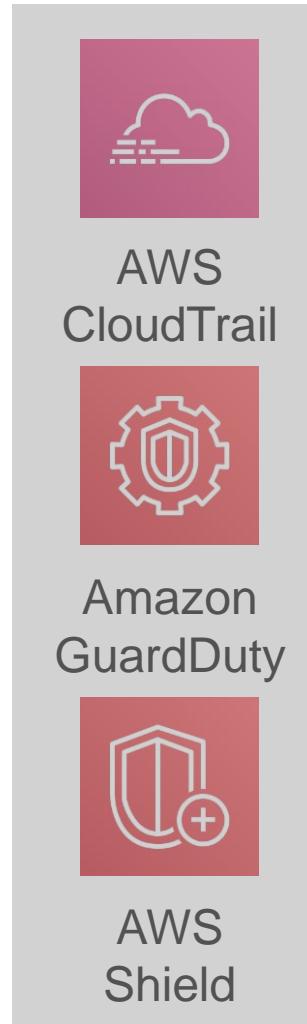
AWS IAM



Amazon GuardDuty



AWS Shield



AWS
Secrets Manager



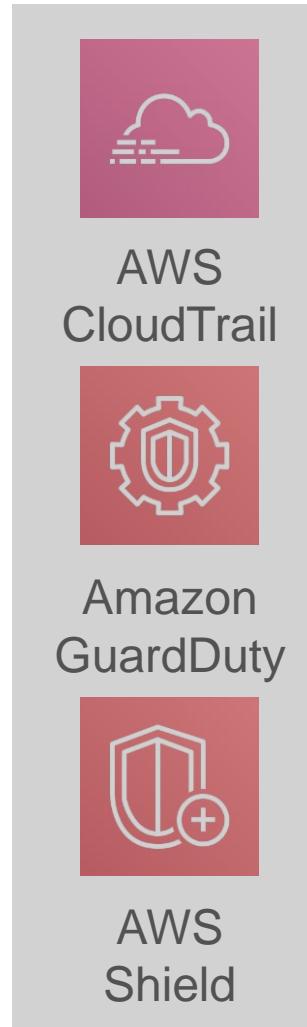
AWS
Certificate Manager



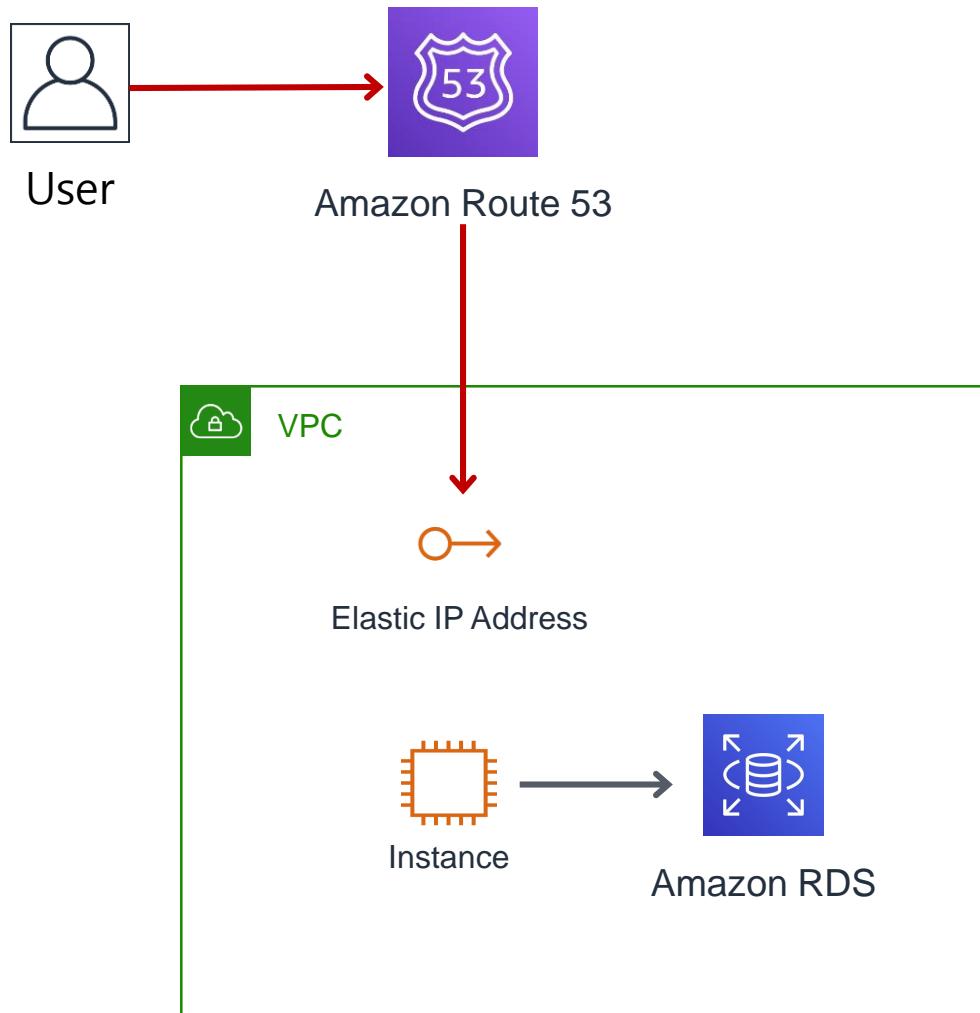
AWS WAF



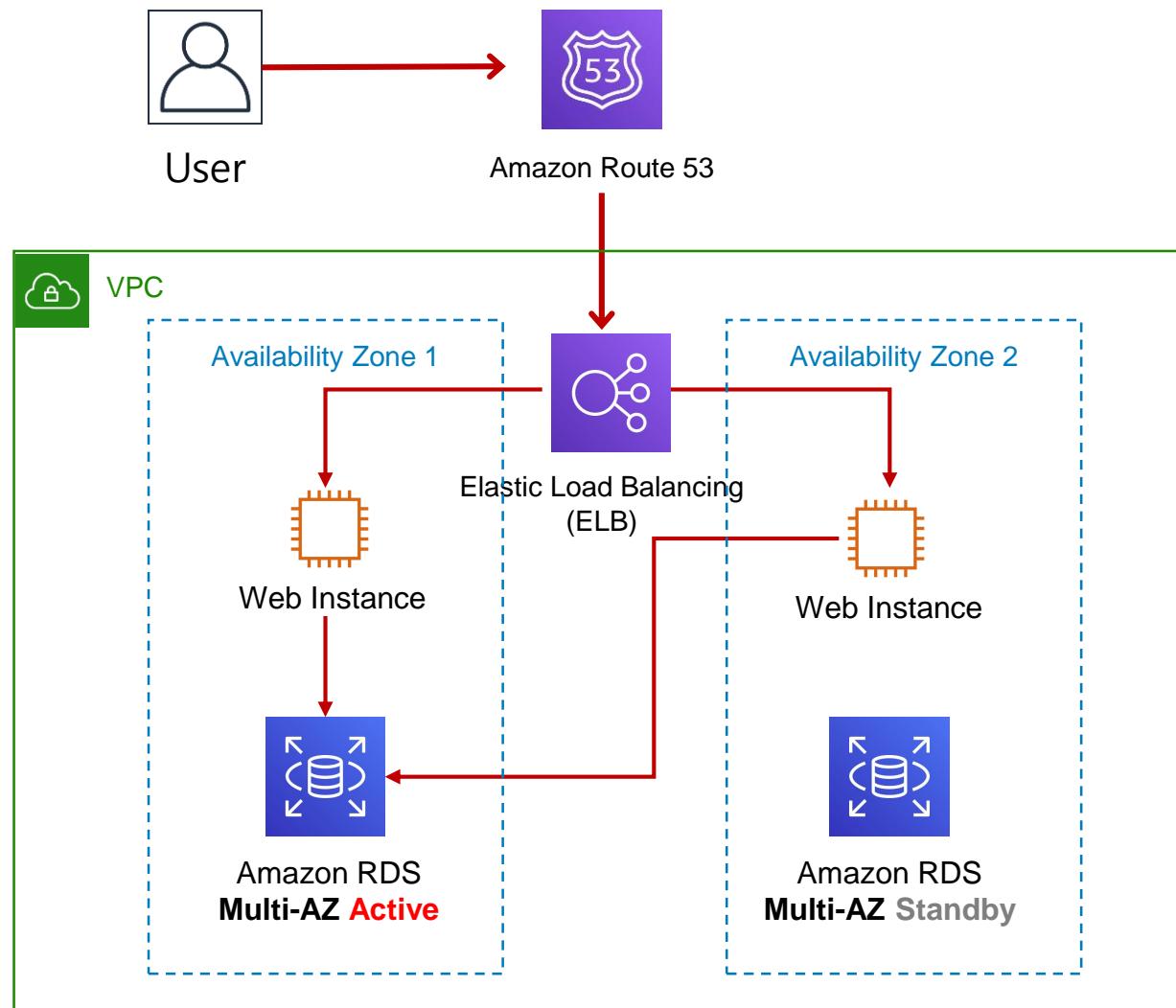
AWS
Key Management Service



사용자 수 > 100



사용자 수 > 1000





Elastic Load Balancing



Application
Load Balancer



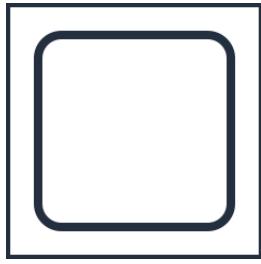
Network Load
Balancer



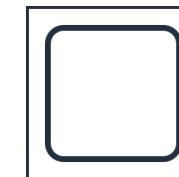
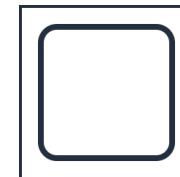
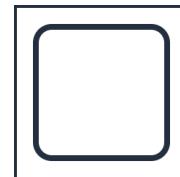
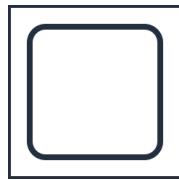
Classic Load
Balancer



- 고가용성
- 1 ~ 65535 포트 사용 가능
- 헬스 체크
- 세션 유지
- 모니터링/로깅
- 컨텐츠 기반 라우팅
- WebSocket
- HTTP/2

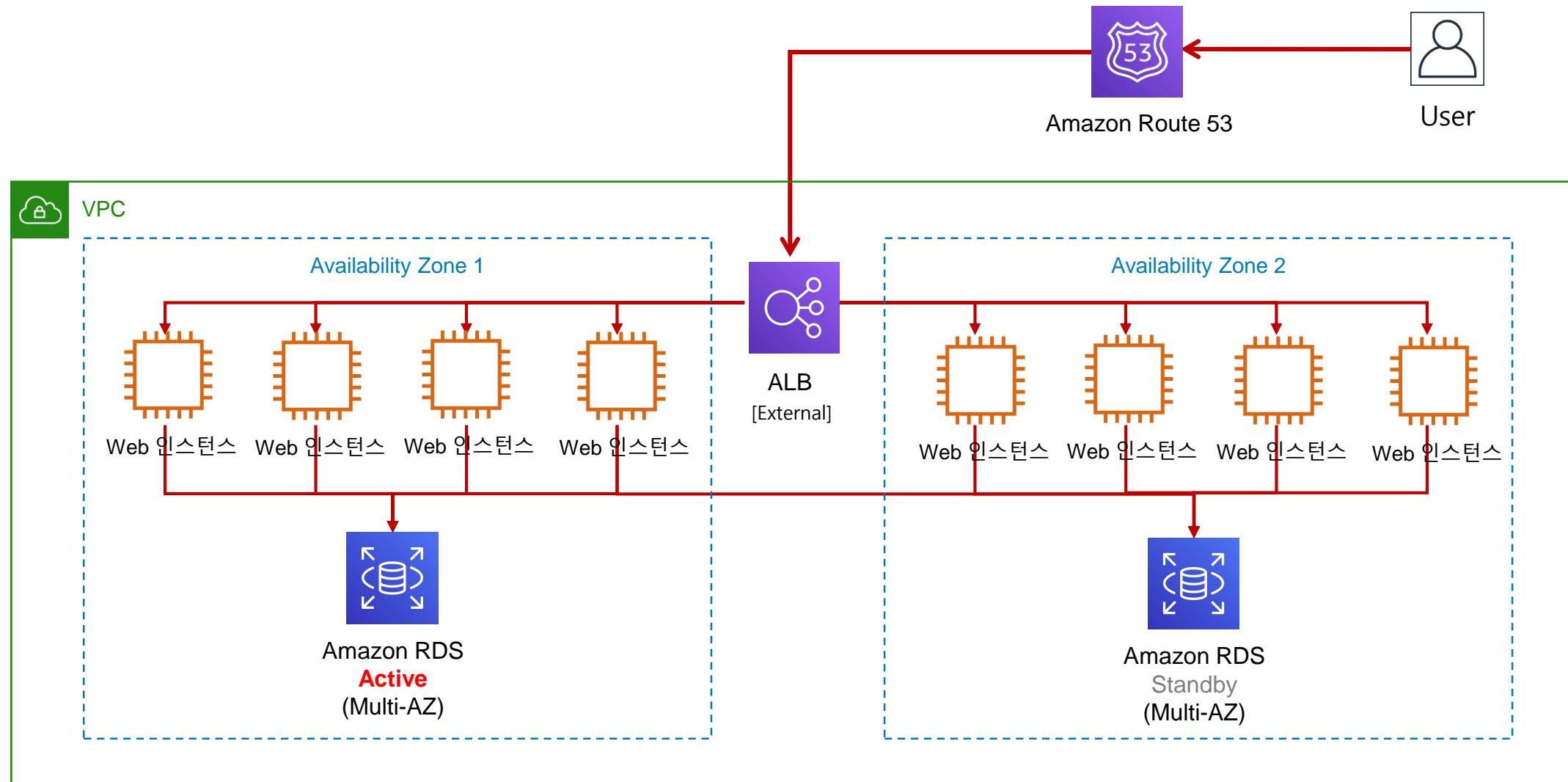


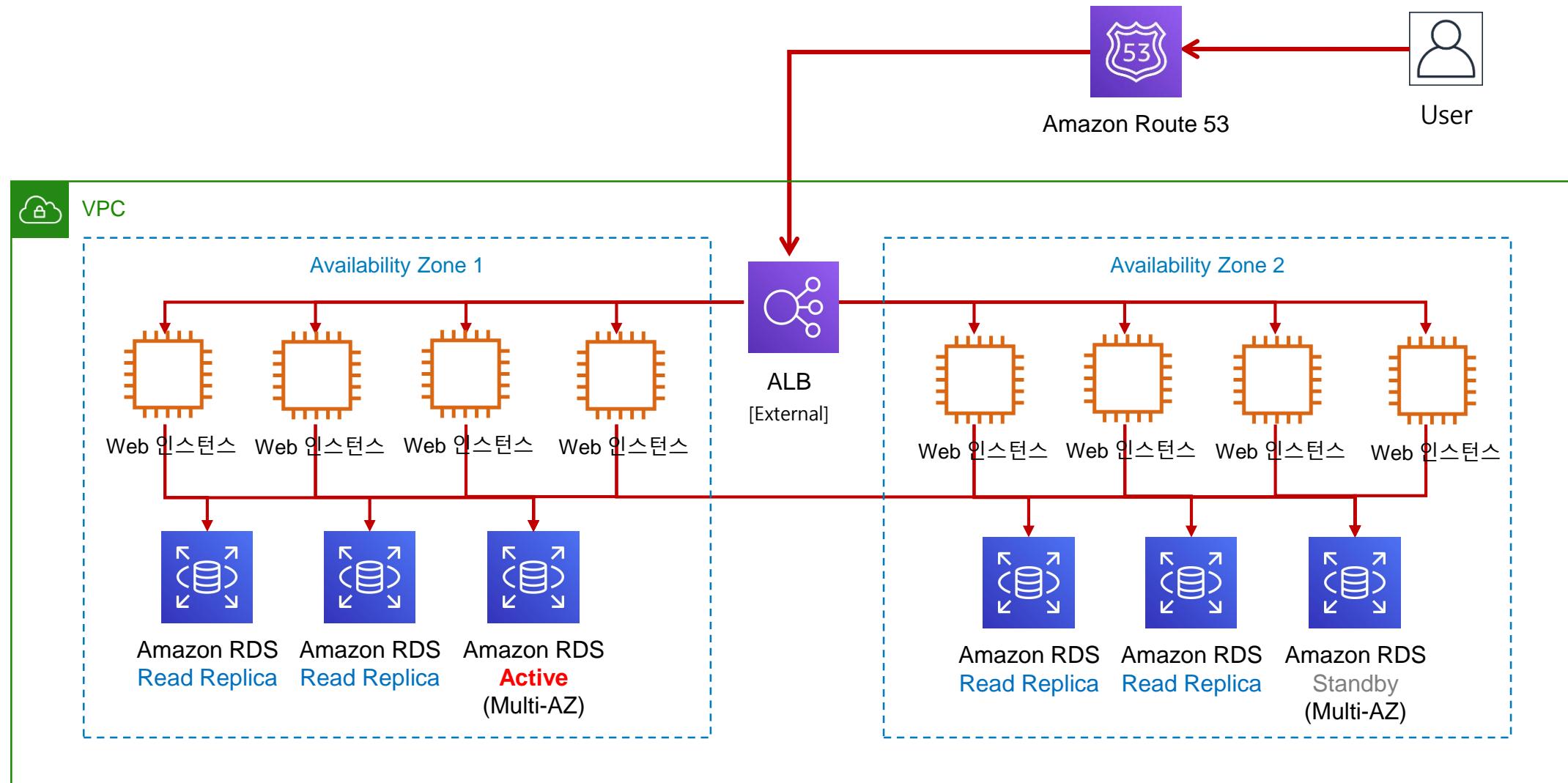
수평적으로 확장



수직적으로 확장

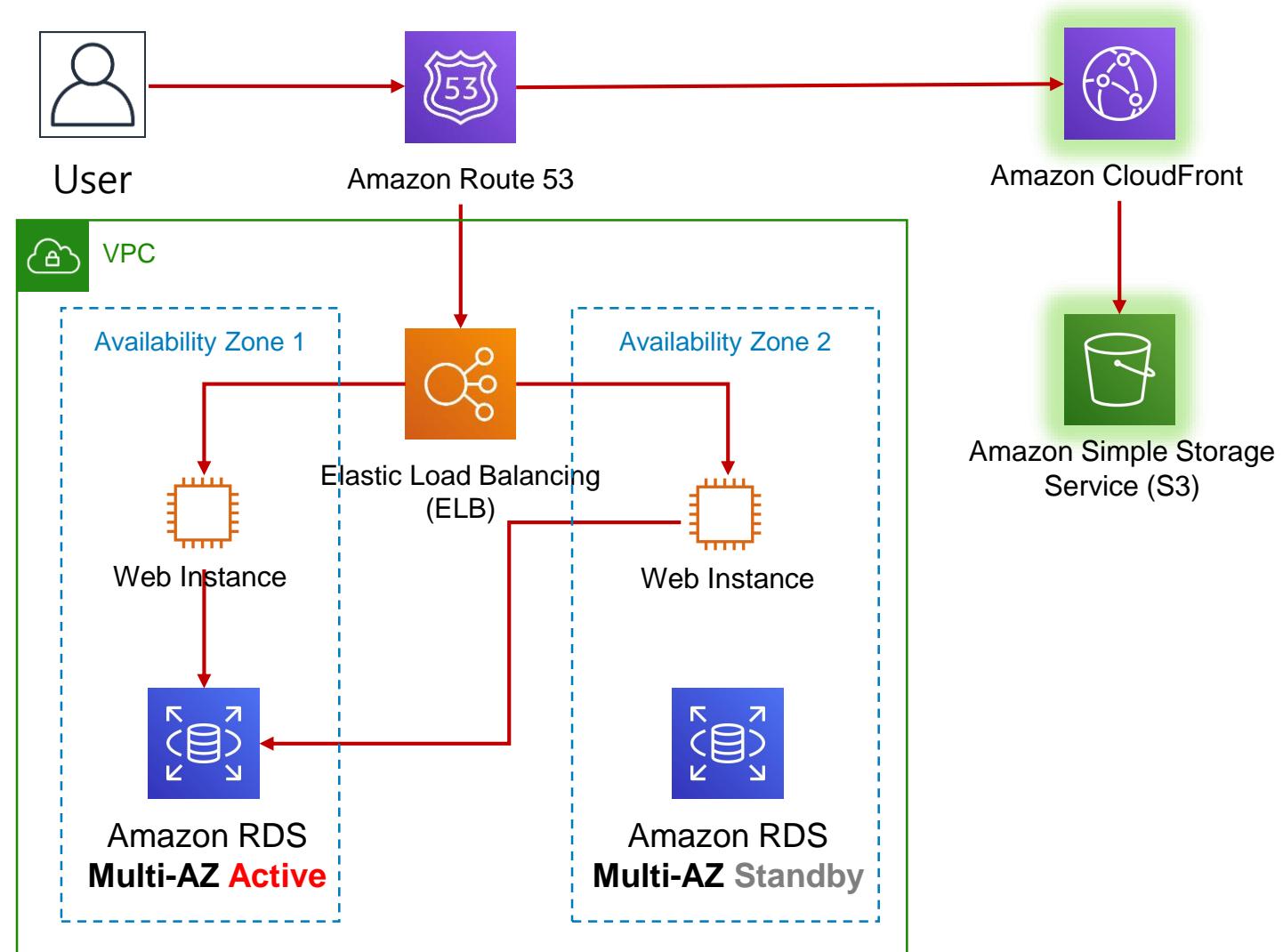
사용자 수 > 10,000





정적 컨텐츠는 S3와 Cloudfront를 활용!

AWS Cloud Day in Busan





Amazon Simple Storage
Service (S3)

- 오브젝트 기반 스토리지
- 높은 내구성 (99.99999999%)
- 정적 자산에 최적화
- 무한 확장 가능
- 오브젝트 당 5TB 까지 지원
- 전송 중 / 저장 시 암호화 지원

- 빠른 컨텐츠 전달을 위해 캐싱
- 오리진 측의 부담을 덜어줌
- 동적 / 정적 컨텐츠 / 스트리밍 비디오
- 사용자 SSL 인증서



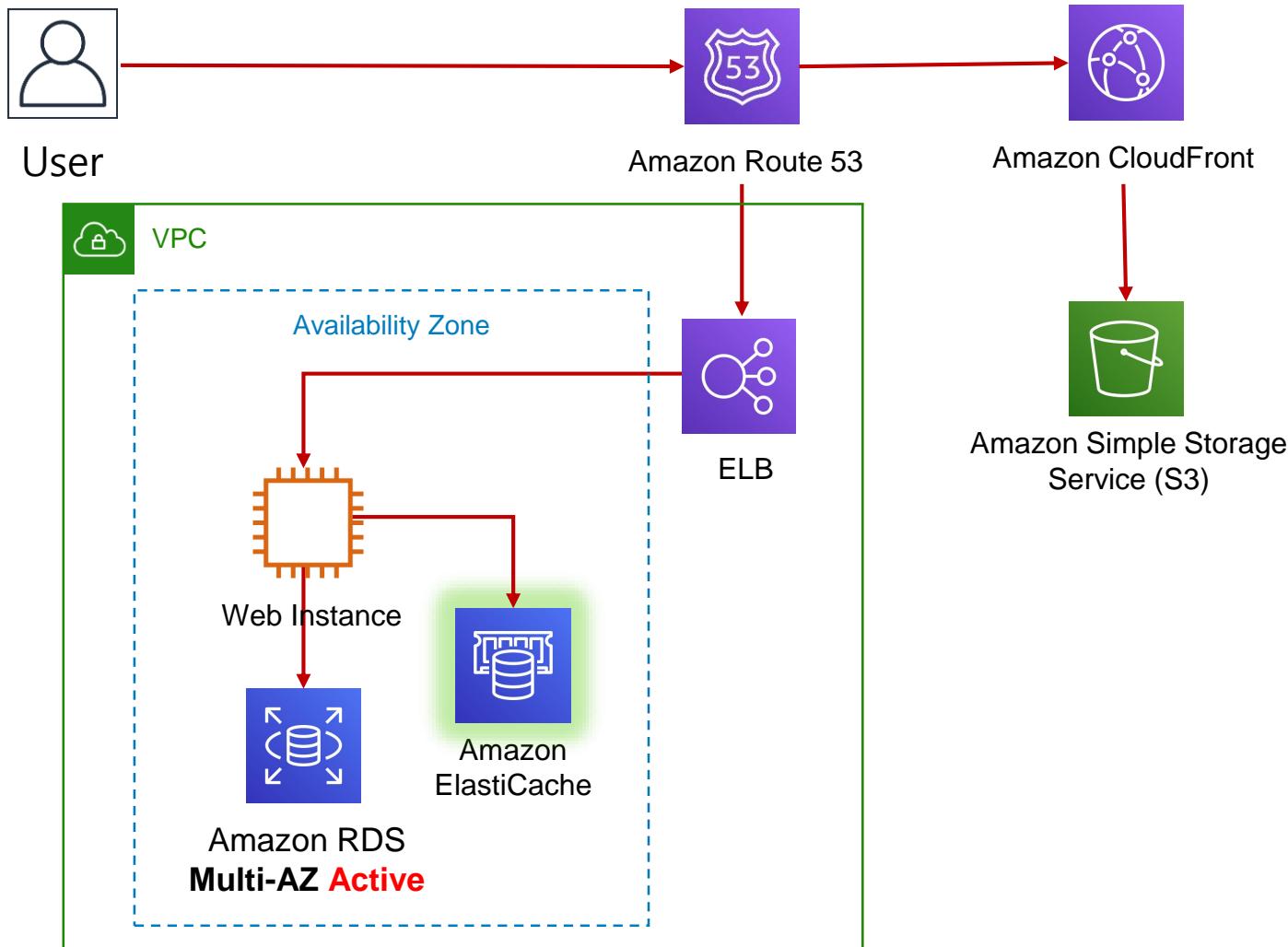
Amazon CloudFront

- 낮은 TTL (0초까지 설정가능)
- AWS에 최적화



좀 더 많은 부하를 주변으로 옮겨봅시다

AWS Cloud Day in Busan



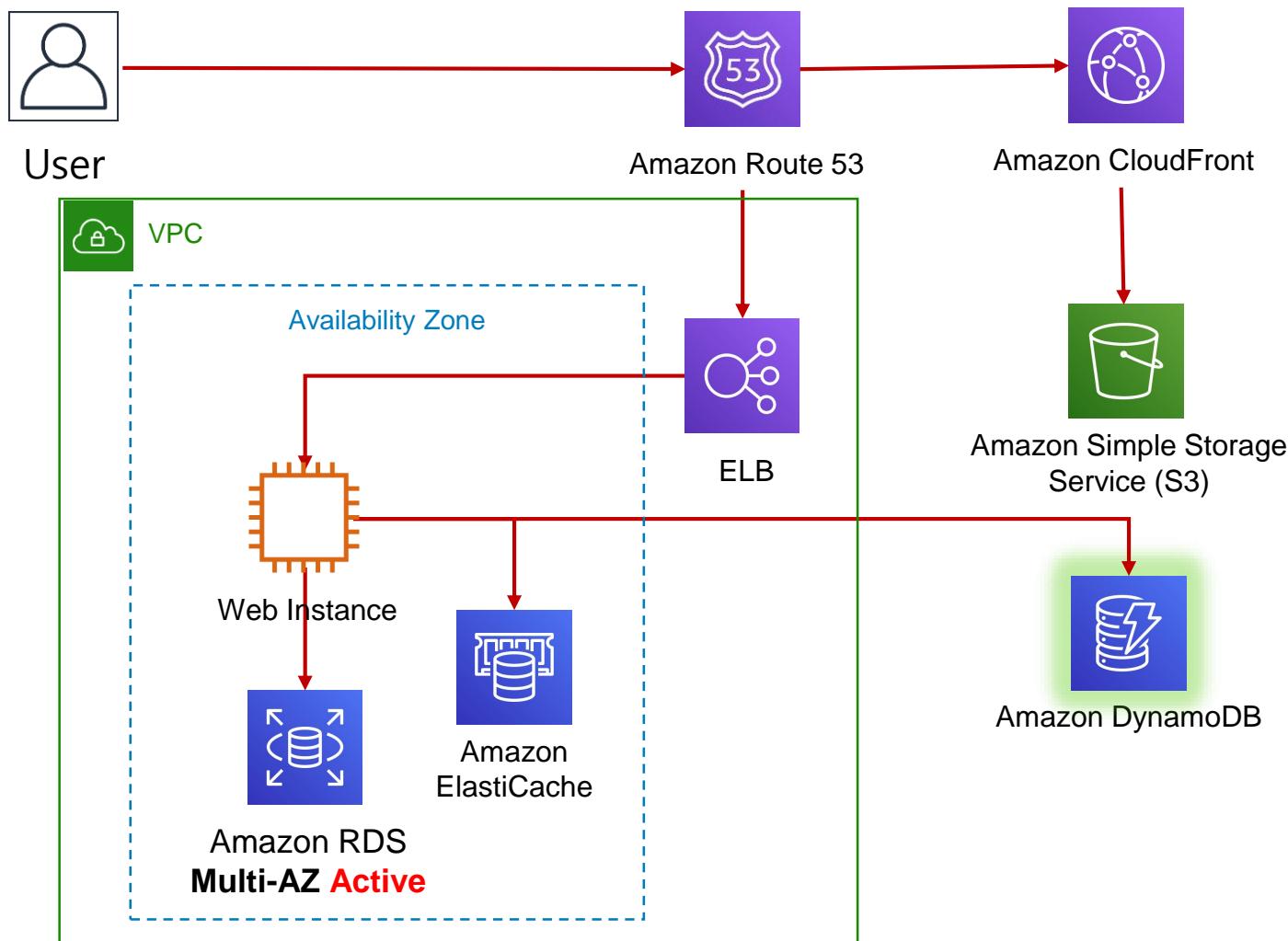


Amazon ElastiCache

- 관리형 Memcached 또는 Redis
- 한 개에서 여러 개의 노드로 확장
- 자가 복구 (문제 인스턴스 교체)
- 한 자리수 ms 속도 (일반적으로)
- Memcached로 하나의 AZ에서 서비스
- Redis에서 여러 AZ에서 서비스

더 많은 부하를 분산해봅시다

AWS Cloud Day in Busan





Amazon DynamoDB

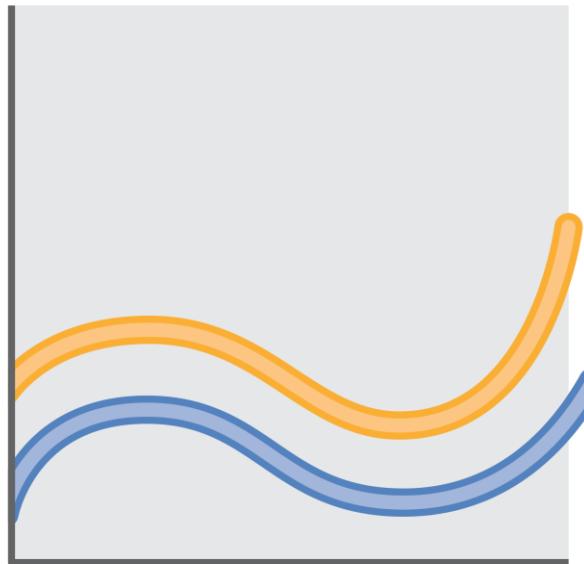
- 완전 관리형 NoSQL 데이터베이스
- 프로비저닝된 스루풋
- 빠르고 예상 가능한 성능
- 완전 분산, 내결함성
- 아이템 당 최대 400 KB, JSON 지원
- Time-to-live (TTL)
- 스트림 / 트리거 역할
- AWS 어플리케이션 오토스케일링
- 글로벌 테이블

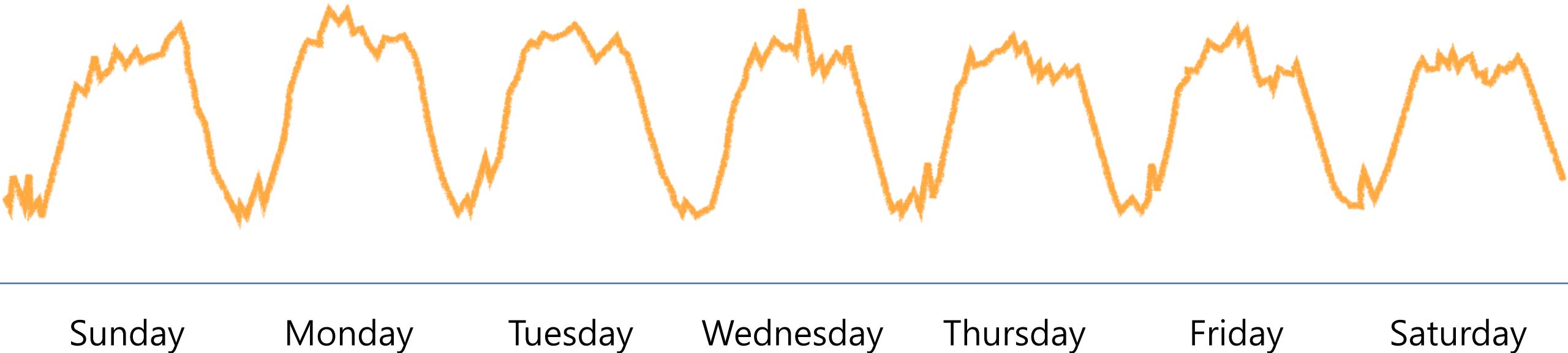
AWS Cloud Day in Busan



확장성을 높여보겠습니다







Sunday

Monday

Tuesday

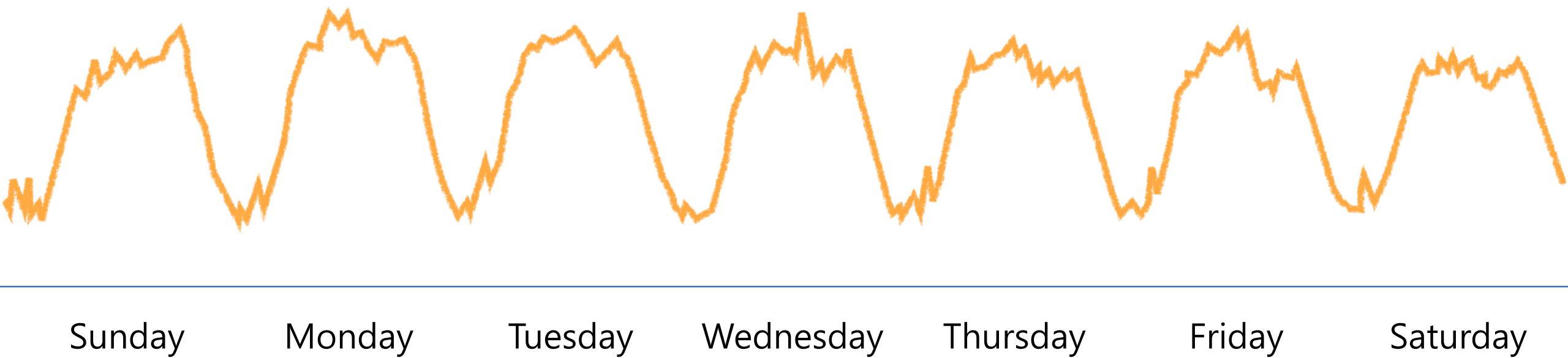
Wednesday

Thursday

Friday

Saturday

미리 준비한 컴퓨팅 용량



Sunday

Monday

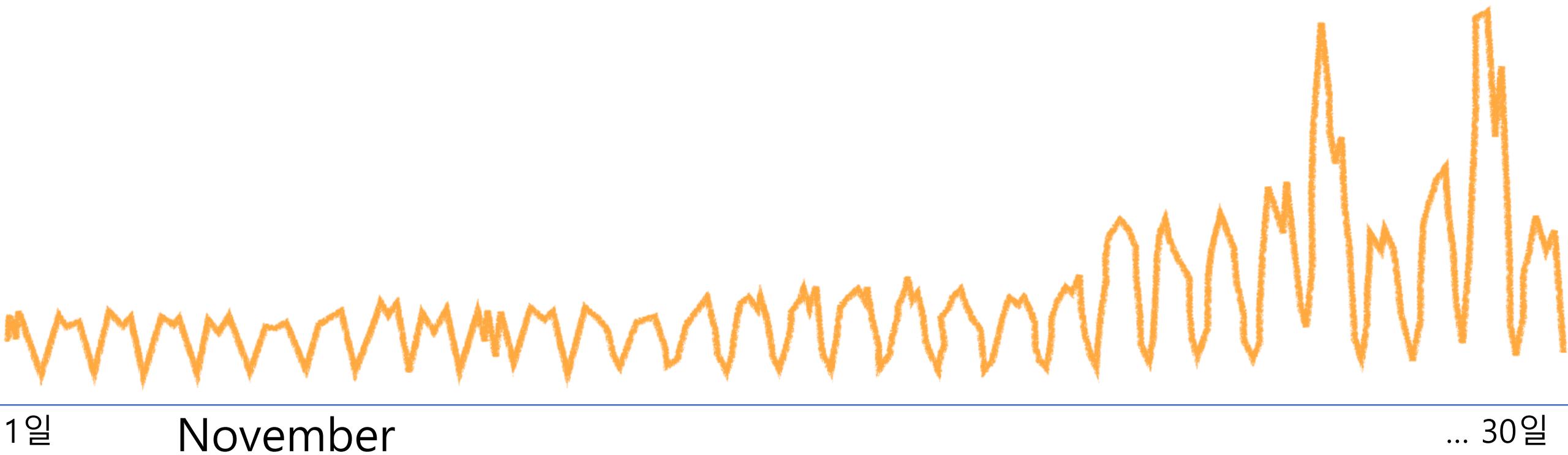
Tuesday

Wednesday

Thursday

Friday

Saturday



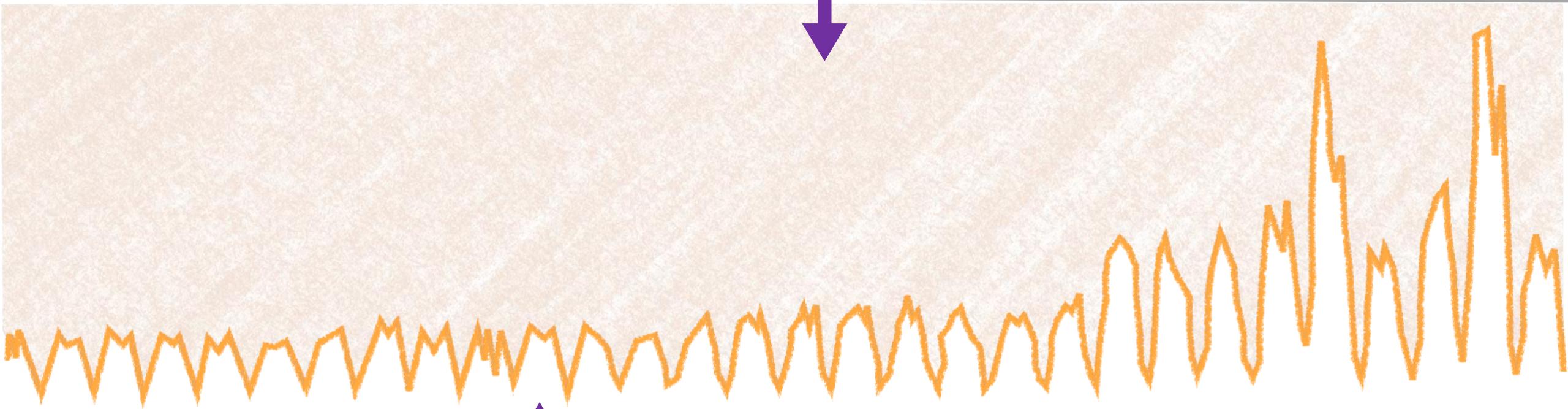
미리 준비한 컴퓨팅 용량



November

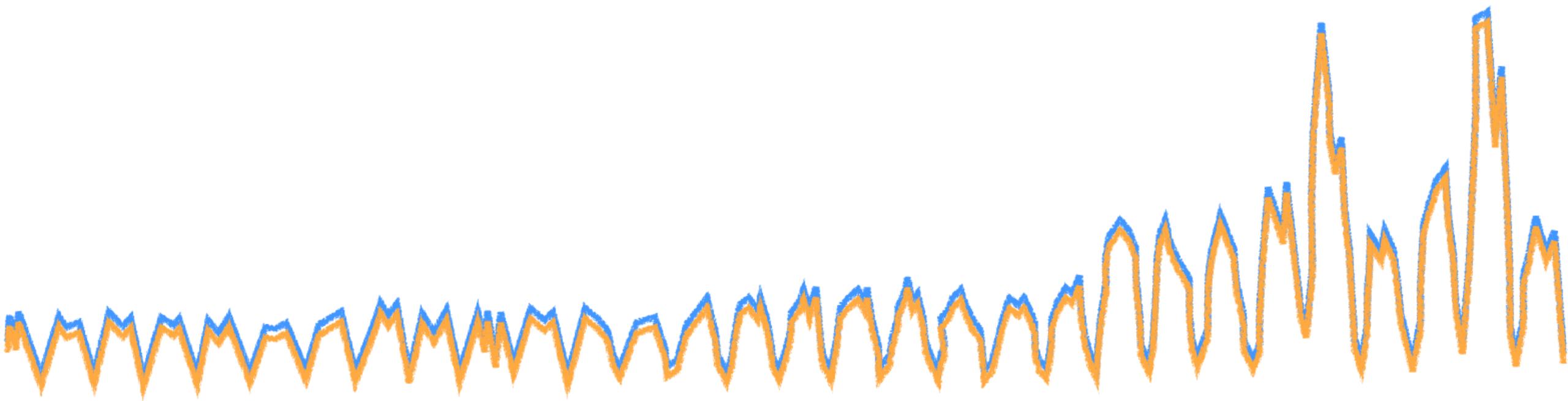
미리 준비한 컴퓨팅 용량

76%



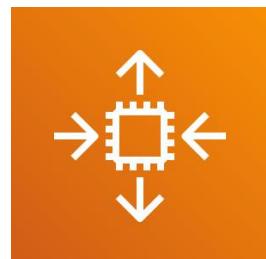
November

24%



November

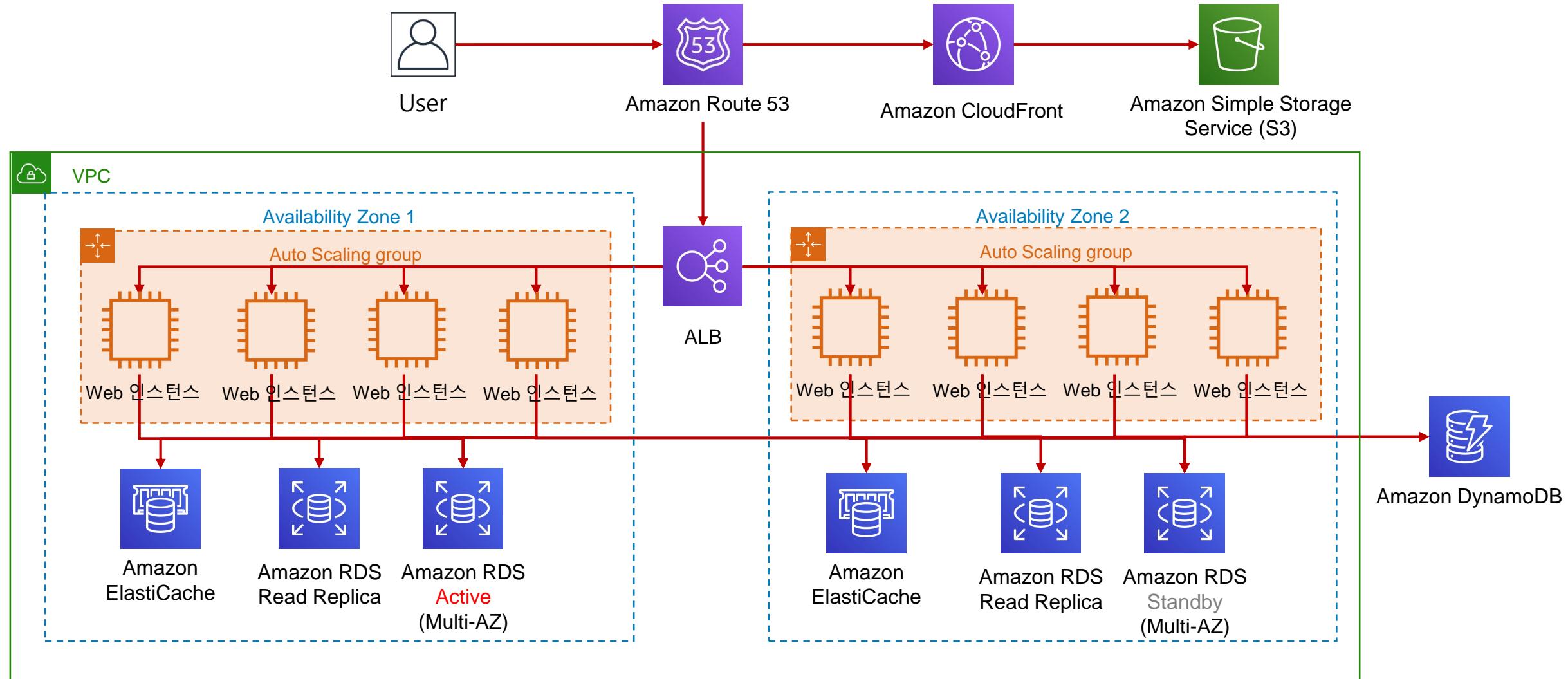
- 컴퓨터 클러스터의 자동 크기 조절
- Pool 사이즈의 최소/최대 값을 지정
- Amazon CloudWatch 지표 기반 스케일링
- 온디맨드 또는 스팟 인스턴스

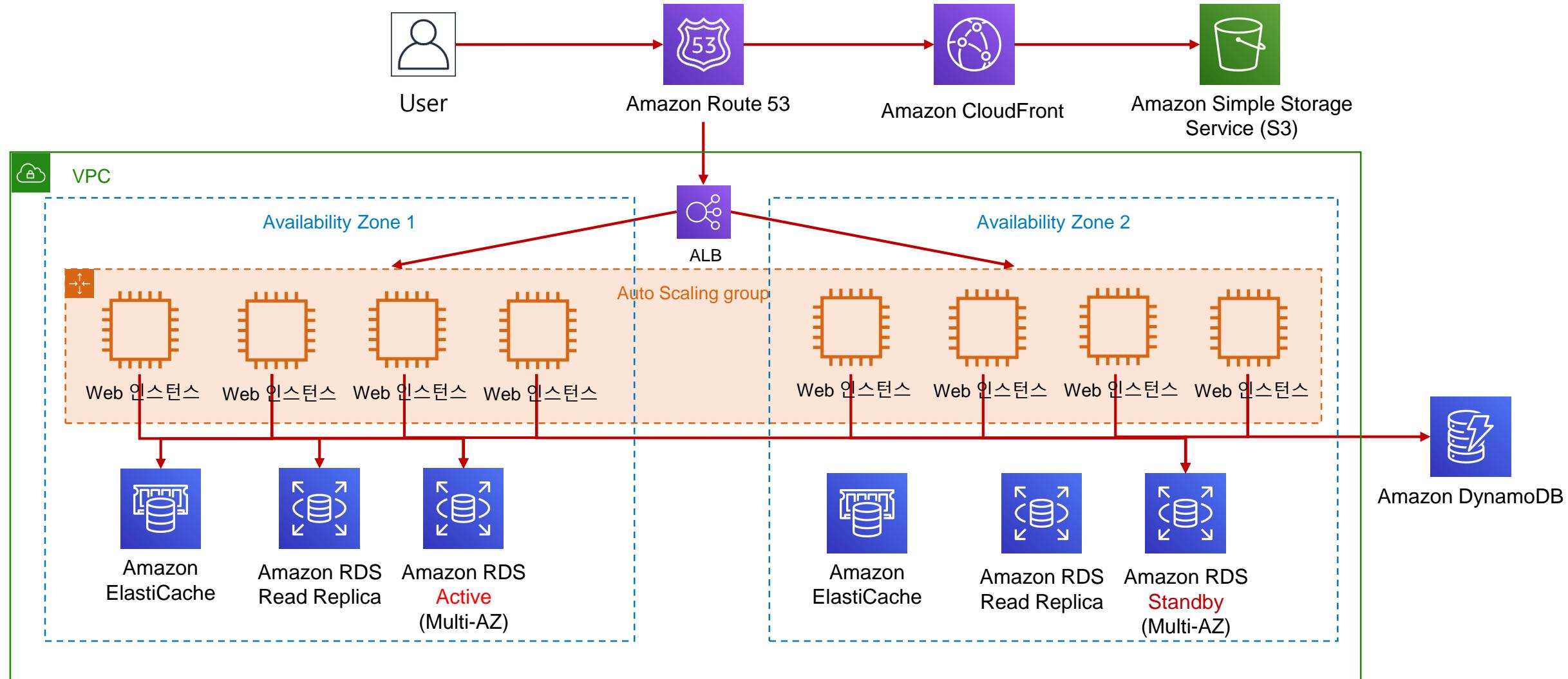


Amazon EC2 Auto Scaling

```
aws autoscaling create-auto-scaling-group  
--auto-scaling-group-name MyGroup  
--launch-configuration-name MyConfig  
--min-size 4  
--max-size 200  
--availability-zones us-west-2c, us-west-2b
```

사용자 수 > 500,000

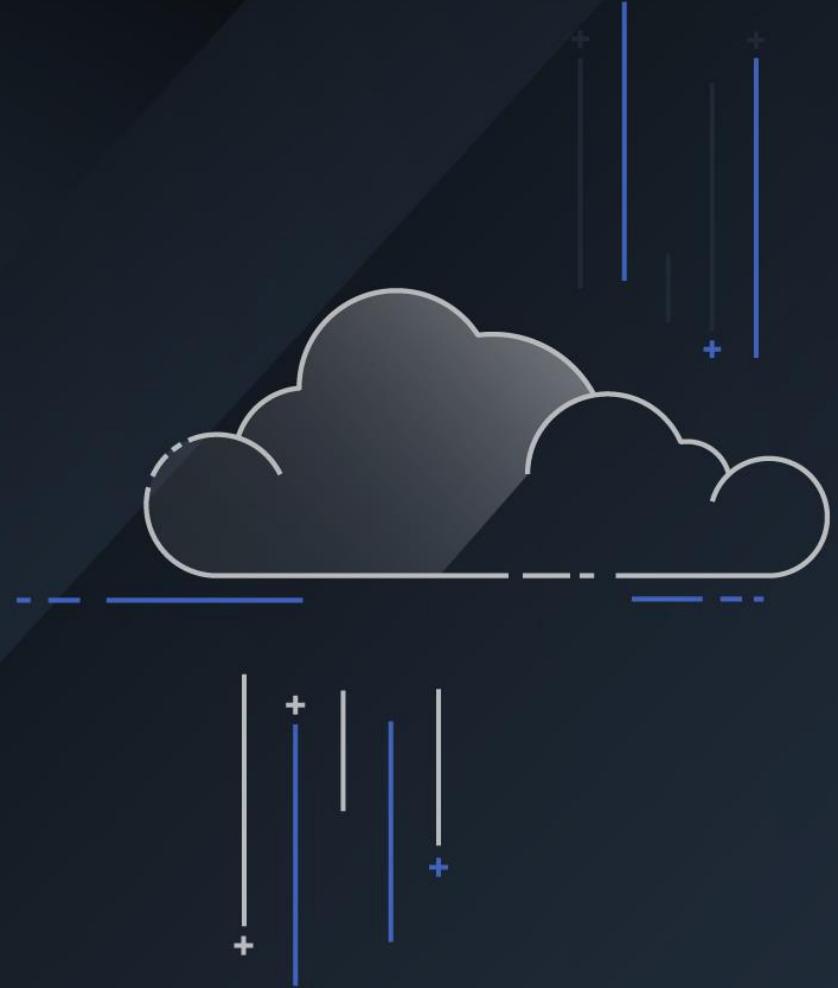




AWS Cloud Day in Busan



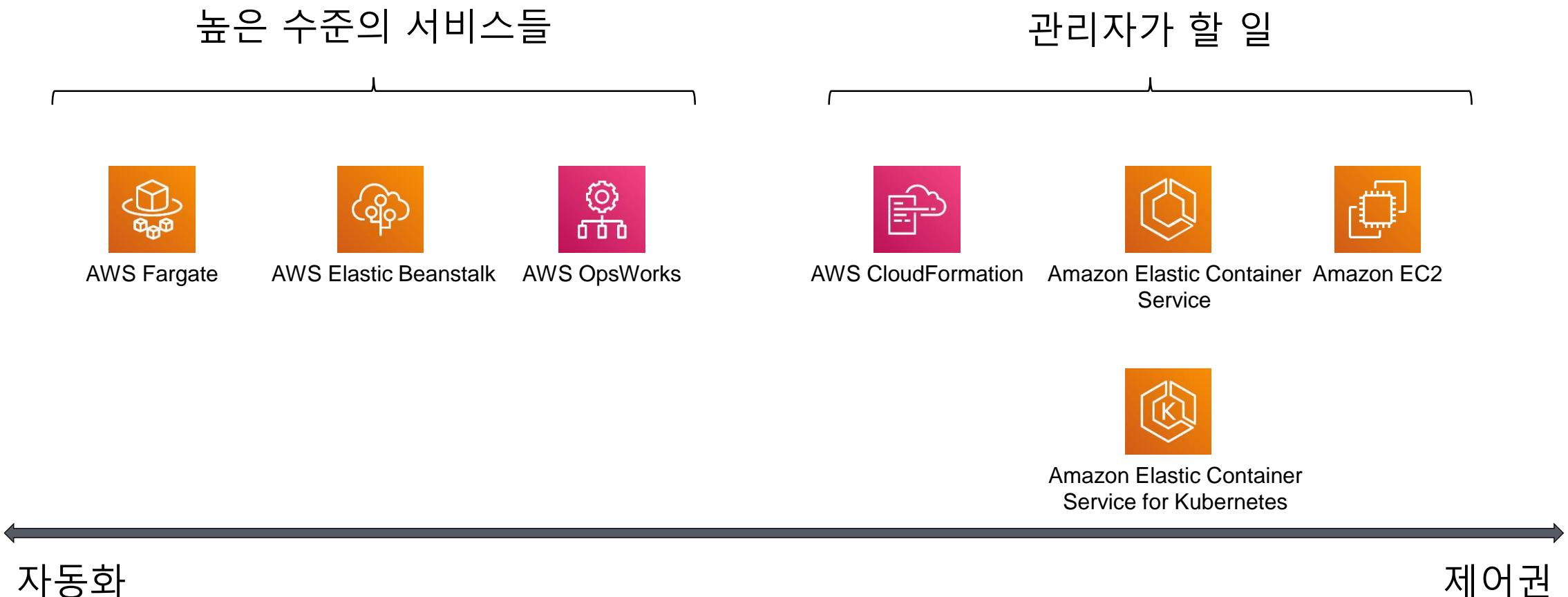
자동화를 적용합시다



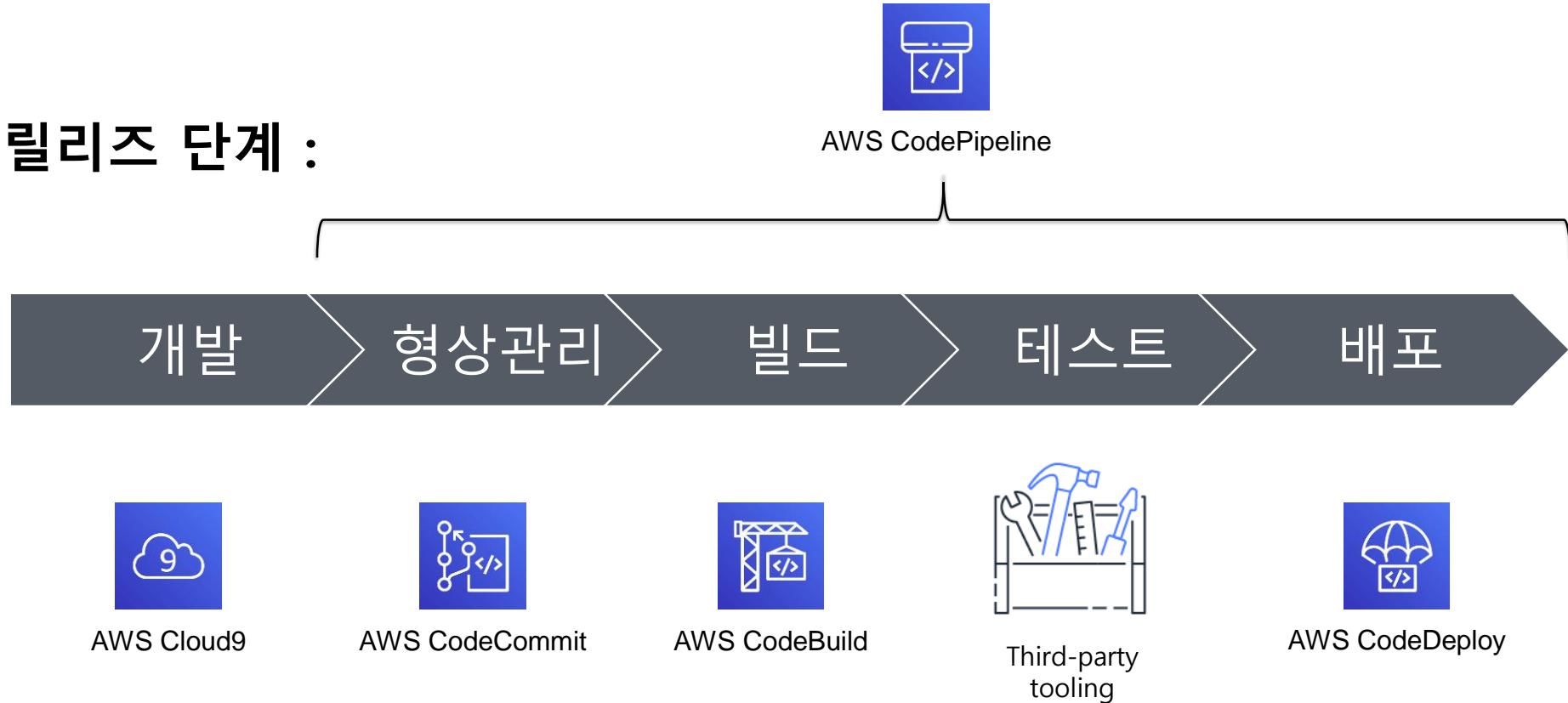


AWS
Systems Manager

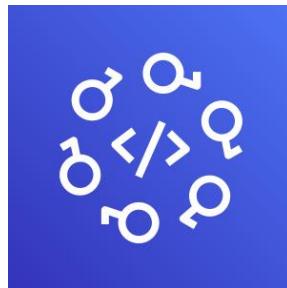
- AWS 클라우드 뿐만 아니라 온프레미스도 지원
- 관리자의 작업을 자동화
- OS 쉘 접근 (베스천 서버 없이)
- 저렴한 가격



소프트웨어 릴리즈 단계 :

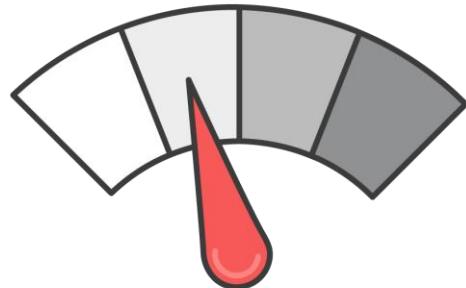


AWS 상에서 어플리케이션을 빠르게 개발, 빌드, 배포



AWS CodeStar

- AWS에서 개발을 수 분 내에 시작
- 보안을 지키며 팀 간 협업
- 소프트웨어 배포 관리가 쉬워짐
- 여러 프로젝트 템플릿 선택 가능



- 모니터링, 지표, 로깅
- 각 서비스/컴포넌트의 성능을 최대한
끌어내기 위해 노력해야합니다
- 고객들의 목소리를 들어봅시다



Amazon CloudWatch

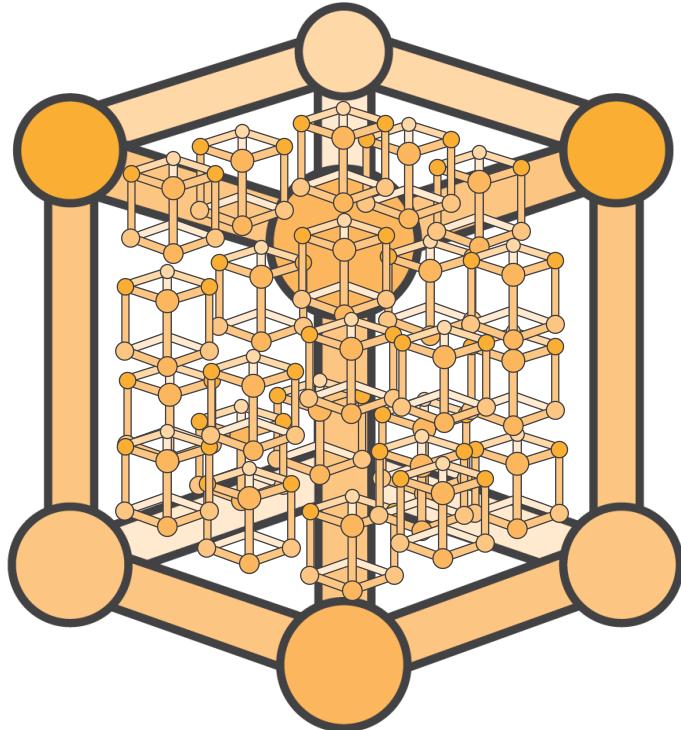
- 수집 : 지표와 로그들
- 모니터링 : 알람과 대시보드
- 액션 : 오토 스케일링과 이벤트
- 분석 : 추이와 지표 계산
- 컴플라이언스와 보안

AWS Cloud Day in Busan



Microservice로 나아갑니다

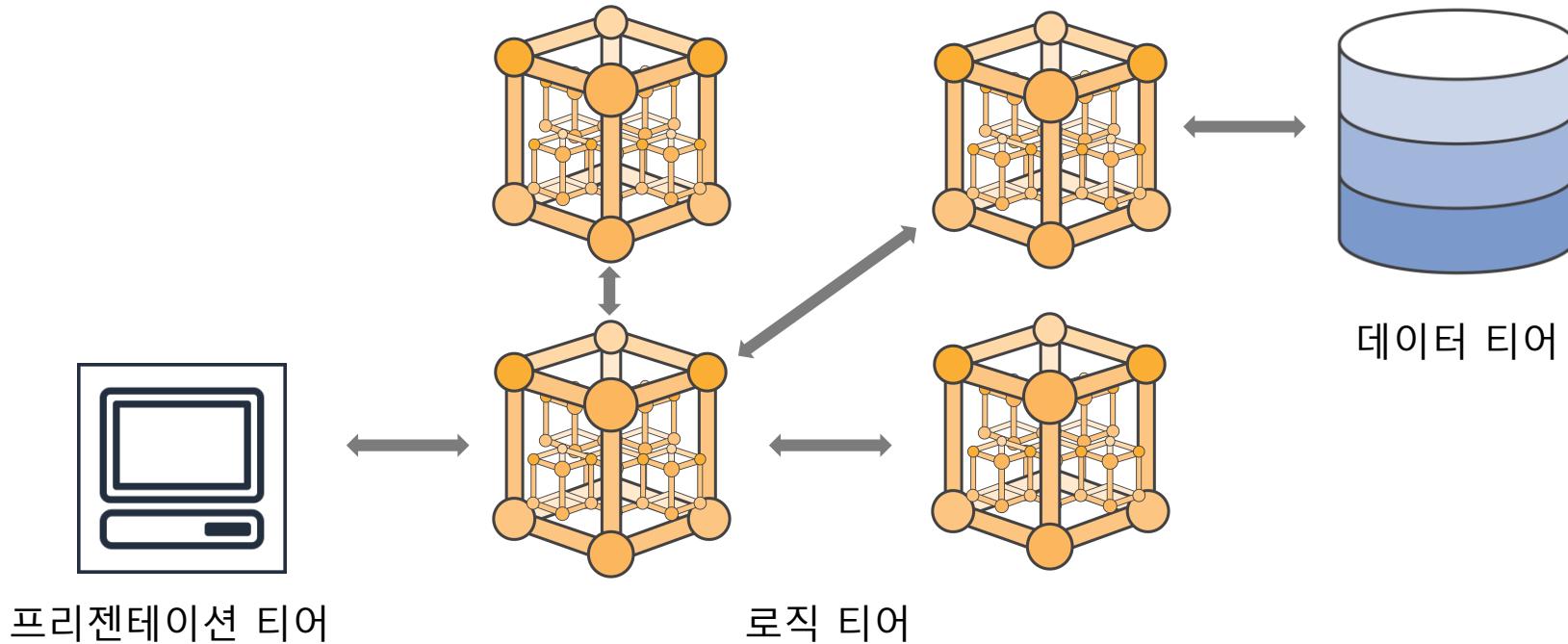


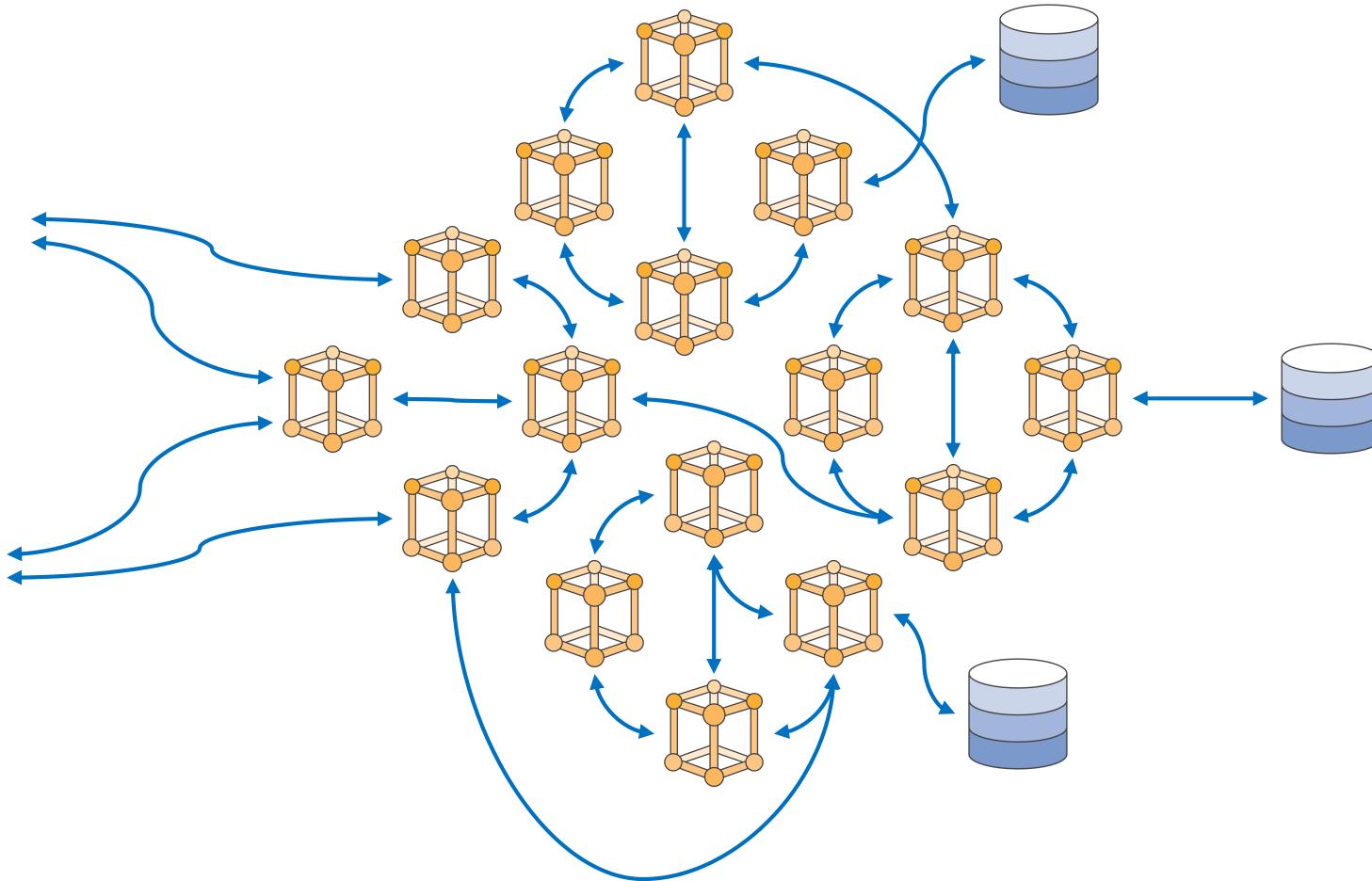


사용자 인터페이스

비즈니스 로직

데이터 접근





바퀴(Wheel)를 재발명하지 맙시다

- API
- Queuing
- Transcoding
- 검색
- 데이터베이스
- 모니터링
- 로깅
- 컴퓨트
- 머신 러닝



Amazon API Gateway



Amazon Simple
Notification Service



Amazon Elasticsearch
Service



Amazon Simple Queue
Service



AWS Fargate



AWS Lambda



Amazon Simple Email
Service (SES)



AWS Step Functions



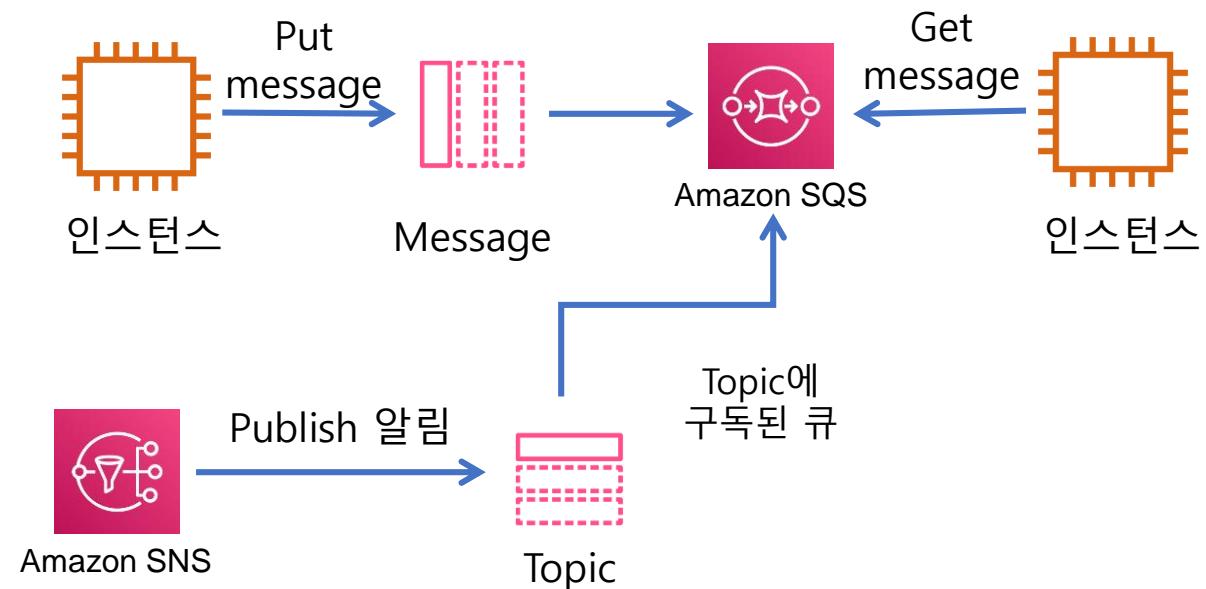
AWS Elemental
MediaConvert



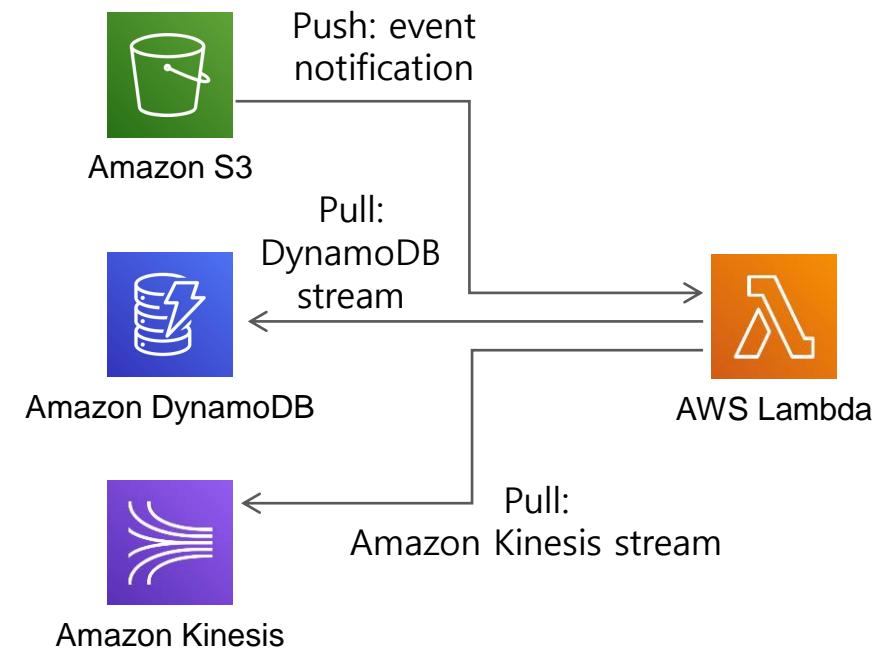
Amazon SageMaker

Loose coupling – Amazon SQS 과 Amazon SNS **AWS Cloud Day in Busan**

- 신뢰성 (multi-AZ)
- 확장 가능 (무제한 메시징)
- 안전함 (큐 인증)
- 간단함 (간단한 API들)
- FIFO 도 지원합니다

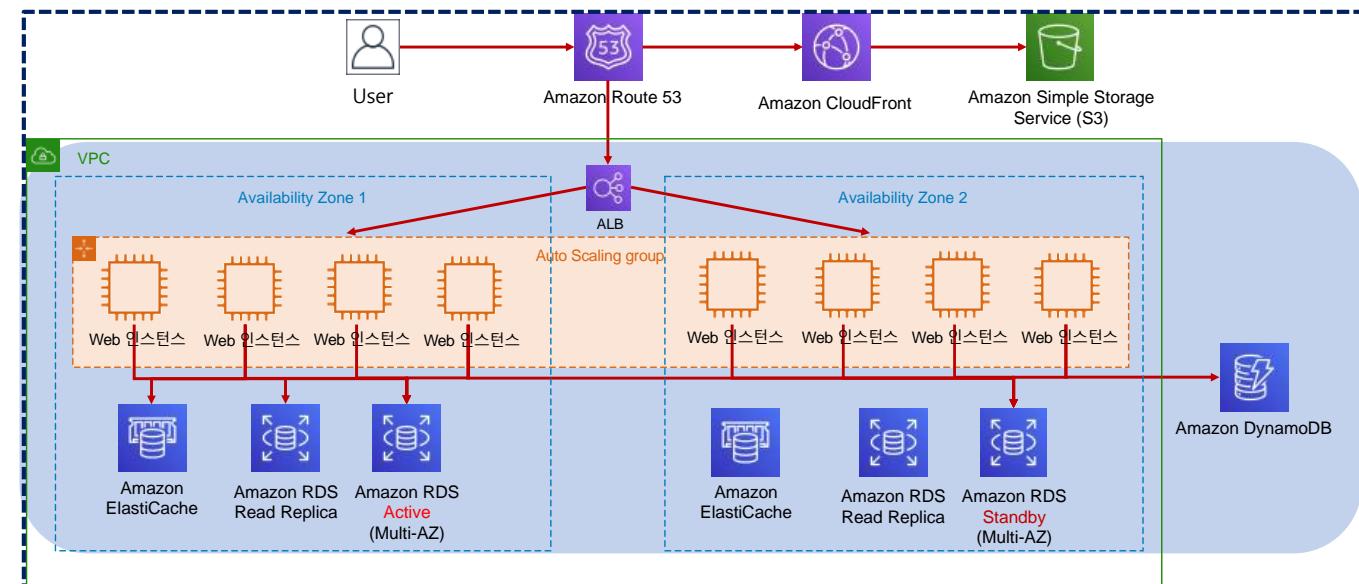
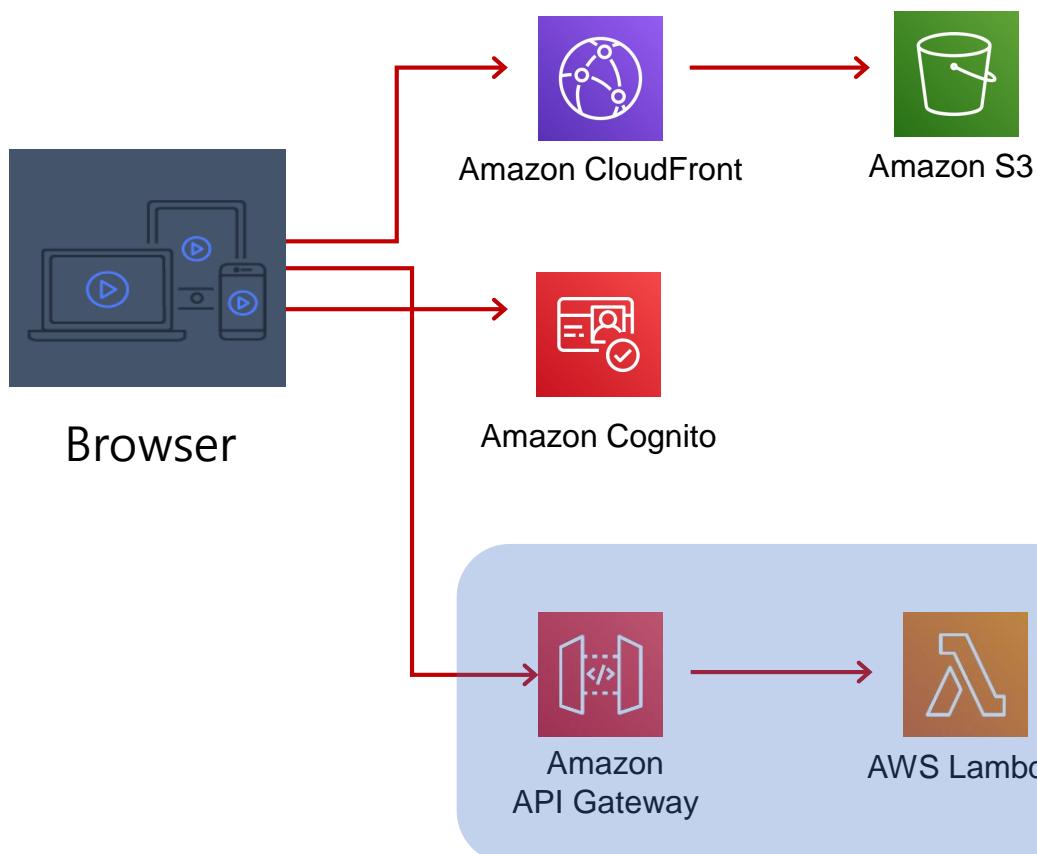


- 이벤트가 발생하면 함수가 실행됨
- Node.js (JavaScript), Java, Python, C#
- Serverless
- 내부적으로 스케일링



Serverless 웹 어플리케이션

AWS Cloud Day in Busan





AWS X-Ray

- 성능의 병목 원인과 에러 파악
- 어플리케이션의 특정 서비스 이슈를 파악
- 어플리케이션의 사용자에 대한 영향도 파악
- 어플리케이션의 서비스 호출 그래프 시각화

Service map

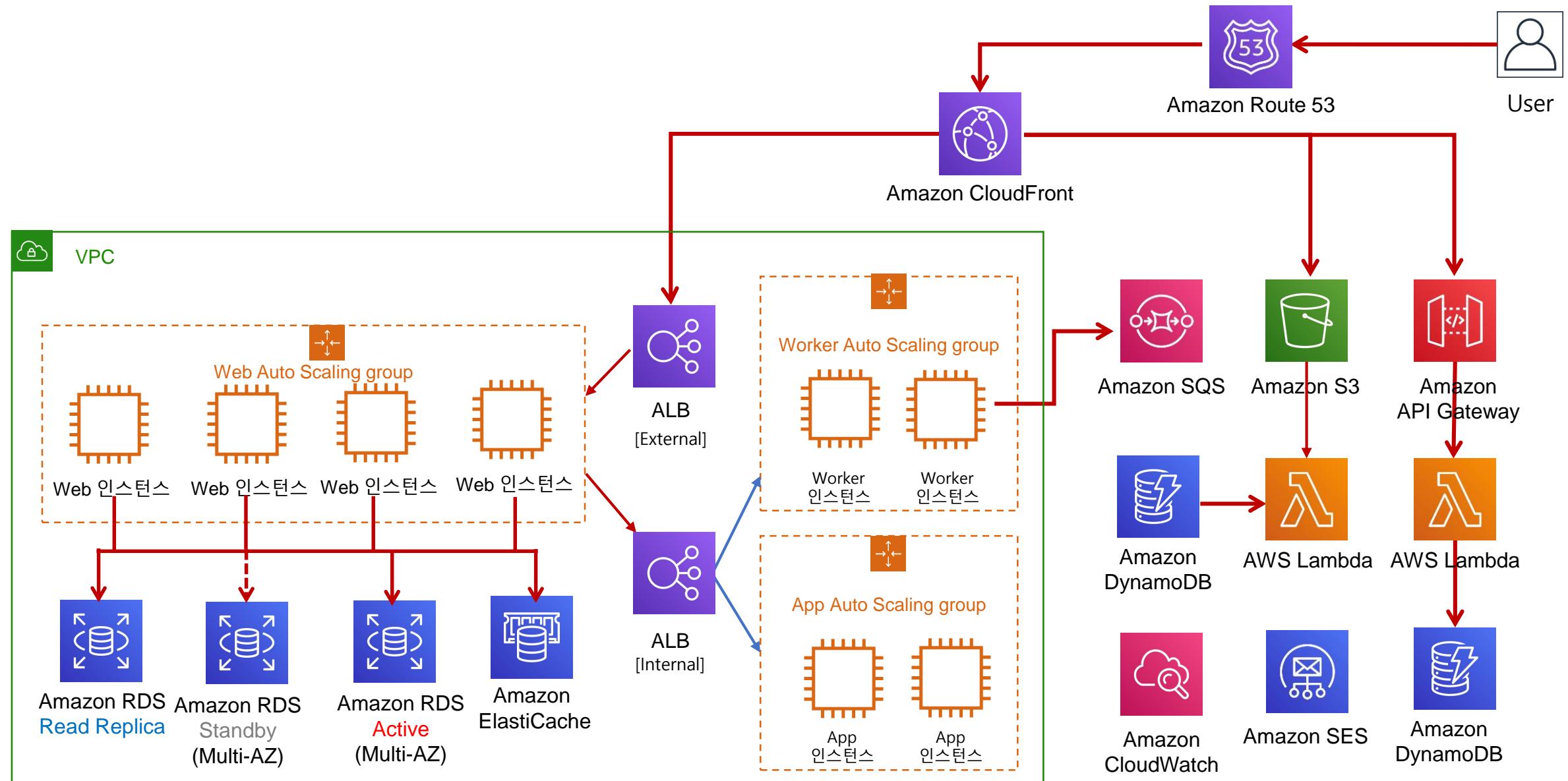
Updated on 2016/11/30 09:05:26 (UTC -08:00)



사용자 수 >1,000,000

1백만이 넘어가면 앞서 언급된 것들이 더욱 필요해집니다

- Multi-AZ
- 계층 사이에 Elastic Load Balancing(ELB)
- 오토 스케일링
- Service oriented architecture (SOA)
- 컨텐츠 전송의 최적화 (Amazon S3/CloudFront)
- DB의 캐시



데이터베이스 이슈들?

해결책

- 역할 별로 여러 DB로 쪼개기
- 샤딩 — 여러 DB인스턴스로 하나의 데이터 세트를 쪼개기
- 어떤 기능은 다른 타입의 DB로 이관 (NoSQL, Graph 등)

1천만 사용자를 위한 아키텍처,

빠르게 리뷰해보겠습니다

- Multi-AZ를 적극적으로 활용
- 확장성이 뛰어난 서비스들을 활용 - ELB, Amazon S3, AWS Lambda, Amazon SNS, Amazon SQS, AWS Step Functions 등
- 모든 계층에 이중화 적용
- 인프라의 안쪽과 바깥쪽에서 데이터를 캐시
- 인프라에 자동화 툴들을 이용

- 적절한 지표/모니터링/로깅을 하고 있는지 확인
- 각 계층을 개별 서비스로 분리 (SOA)
- 오토 스케일링 이용
- 바퀴를 다시 발명하지 마세요
- 적절한 때가 되면 NoSQL로 이행

- 어플리케이션을 더 정교하게 튜닝
- 더 많은 영역에서 마이크로서비스 적용
- 멀티 AZ에서 멀티 리전으로 진화
- 전체 스택에 대해서 깊은 분석 진행
- 가능하다면 언제나 서비스 적용

아래 링크의 글들을 읽어보세요

aws.amazon.com/documentation

aws.amazon.com/architecture

aws.amazon.com/well-architected

aws.amazon.com/solutions

aws.amazon.com/quickstart

AWS 사용 시작하기

aws.amazon.com/free

forums.aws.amazon.com

aws.amazon.com/answers

aws.amazon.com/premiumsupport

여러분의 피드백을 기다립니다!

세션 종료 후,
AWS Cloud Day in Busan 모바일 앱과 QR코드를 통해
강연평가 및 **설문조사**에 참여하시고
AWS만의 기념품을 받아가세요.
여러분의 소중한 의견 부탁 드립니다.



▲
AWS Cloud Day in Busan
설문 페이지

AWS Cloud Day in Busan



감사합니다.

