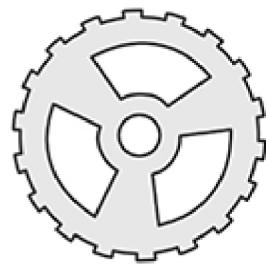


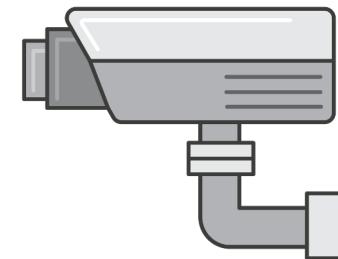
AWS Well-Architected Framework

Well-Architected Framework은 무엇인가요?

WAF라 함은 고수준의 가이드와 베스트 프랙티스를 고객 여러분에게 제공하여 보안성, 안정성, 성능 효율성, 비용 최적화, 운영 우수성이 보장되는 어플리케이션을 AWS Cloud상에서 유지할 수 있게 지원합니다.



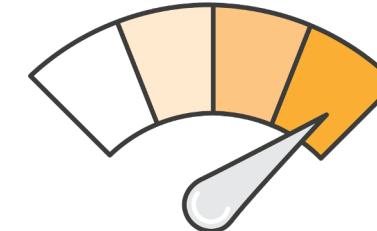
운영 우수성



보안



안정성



성능 효율성

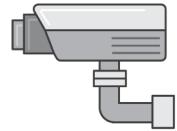


비용 최적화



AWS Well-Architected

Well Architected 영역



보안 (Security) : AWS 클라우드 상의 데이터 및 자산을 안전하게 보호하기 위한 모범 사례가 적용되어 있습니까?



안정성 (Reliability) : 시스템/애플리케이션 아키텍쳐가 장애, 업무 증가 및 기타 이벤트에 능동적으로 대처할 수 있습니까?



성능 효율화 (Performance Efficiency) : 시스템 리소스들이 최적의 성능을 낼 수 있도록 설계되어 있습니까?



비용 최적화 (Cost Optimization) : 비용을 줄일 수 있는 방법들을 고려하고 있습니까?



운영 고도화(Operational Excellence) : 운영중인 시스템을 모니터링 하고, 지원체계를 끊임없이 개선할 수 있습니까?



보안 (Security)

보안 디자인 원칙

- **강력한 신원 기반 구현**

최소한의 권한 원칙을 구현하고 각 상호 작용에 대한 적절한 권한을 부여하여 직무 분리

- **추적 로그 설정**

모든 변경 사항과 작업 실시간 모니터링 로깅 및 감사하고 자동 응답 및 조치

- **모든 계층에 보안 적용**

인프라 경계에서의 보안 뿐 아니라 모든 계층 (Edge, VPC, ELB, EC2, OS 및 애플리케이션)
보안 적용 및 제어

- **보안 모범 사례 자동 적용**

최적화된 이미지를 사용하고 인프라를 버전 관리된 템플릿을 통하여 정의

보안 디자인 원칙

- **전송 중 및 유휴 시 데이터 보호**

민감도 수준으로 데이터를 분류, 필요시 선택적 암호화, 토큰화 및 액세스 제어

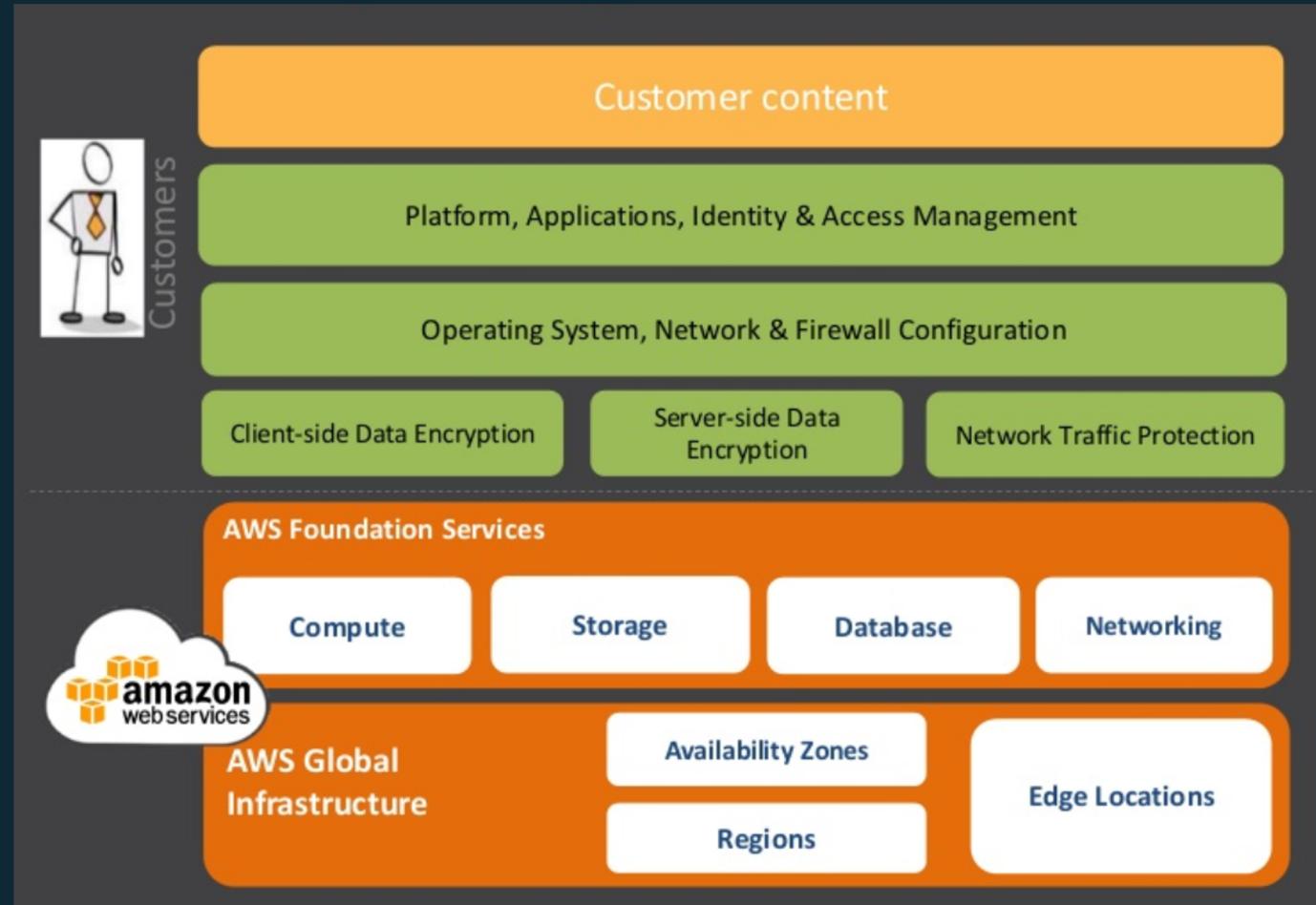
- **사람의 실수 미연에 방지**

사람의 실수 최소화를 위해 데이터의 직접 액세스 또는 수동 처리 최소화

- **보안 이벤트에 대비**

사고 대응 시뮬레이션을 실행하고 자동화된 도구를 사용하여 탐지, 조사 및 복구 속도 향상

책임 공유 모델



고객은 Cloud 내 (**in cloud**)에 자신의 서비스에 대한 보안을 유지할 책임을 집니다.

AWS는 Cloud 자체 (**of cloud**)의 보안을 유지할 책임을 집니다.

AWS의 보안 관련 서비스



Areas	Key Services
Identity and Access Management	 IAM  MFA token  AWS Organizations
Detective Controls	 AWS CloudTrail  AWS Config  Amazon CloudWatch  Amazon GuardDuty  Amazon Inspector
Infrastructure Protection	 Amazon VPC  AWS WAF  AWS Shield
Data Protection	 Elastic Load Balancing  Amazon EBS  Amazon S3  Amazon RDS  Amazon DynamoDB  AWS KMS
Incident Response	 IAM  AWS CloudFormation

1. ID 및 액세스 관리 > AWS IAM

AWS 서비스 및 리소스에 대한 액세스를 고객이 직접 제어할 수 있도록 **AWS IAM** 제공

Account Owner ID (루트 계정)

- 모든 서비스에 대한 접근
- 빌링 접근
- 콘솔과 API에 대한 접근
- 고객 지원 (Customer Support)에 대한 접근

AWS Account Owner (루
트 계정)

IAM Users, Groups and Roles

- 특정 서비스에 대한 접근
- 콘솔, API에 대한 접근
- 고객 지원 (Customer support에 대한 접근



AWS IAM Users,
Groups, Roles

Temporary Security Credentials

- 특정 서비스에 대한 접근
- 콘솔, API에 대한 접근



Temporary
Security Credentials

2. 탐지 제어 > CloudTrail

AWS CloudTrail은 AWS 관리 콘솔, AWS SDK, CLI 및 기타 AWS 서비스를 통해 수행된 작업을 비롯하여 모든 AWS 계정 활동의 이벤트 기록을 제공

The screenshot shows the AWS CloudTrail API activity history interface. The top navigation bar includes 'AWS', 'Services', 'Edit', and user information for 'Sivakanth Mundru' in 'Frankfurt'. The main content area has a search bar and filter options ('Select attribute', 'Enter lookup value', 'Time range'). A table lists three events:

	Event time	User name	Event name	Resource type	Resource name
▶	2015-09-26, 09:03:46 AM	sivakant	StartLogging	Trail	ITAuditandOpsTrail
▶	2015-09-26, 08:59:00 AM	sivakant	CreateTrail	Trail and 1 more	ITAuditandOpsTrail and 1 more
▼	2015-09-26, 08:58:20 AM	sivakant	DeleteTrail	Trail	Default

Below the table, detailed event information is shown for the DeleteTrail event:

AWS access key	AKIAIPQ7JPO3XFCDZWQ	Event source	clouptrail.amazonaws.com
AWS region	eu-central-1	Event time	2015-09-26, 08:58:20 AM
Error code		Request ID	684cf09c-6467-11e5-9440-b37c019595ed
Event ID	8336db53-daa5-4df3-9f23-8897905138a3	Source IP address	23.20.234.35
Event name	DeleteTrail	User name	sivakant

The 'Resources Referenced (1)' section lists 'Default' and 'Trail', with a 'View event' button. At the bottom, another CreateTrail event is listed.

No more events



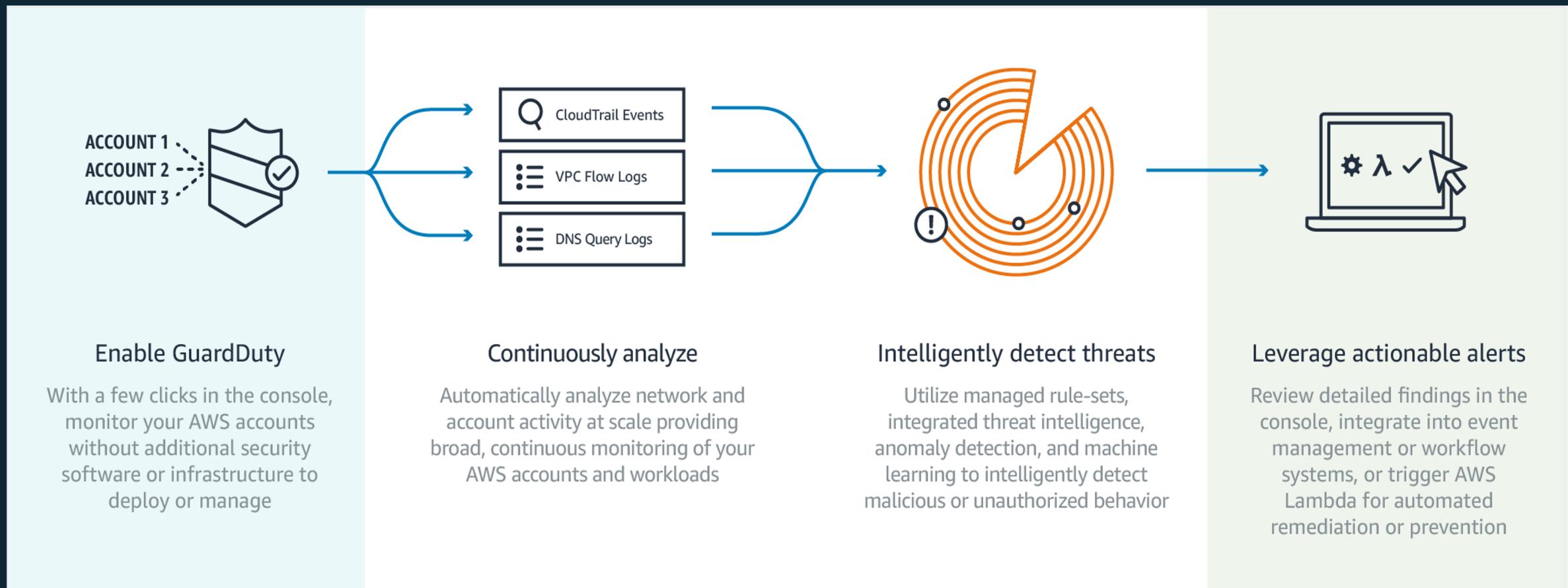
2. 탐지 제어 > AWS Config

- **Config** : AWS리소스의 변경사항을 추적하고 감사
- **Config Rules** : 해당 변경 사항이 기준 정책에 위반될 때, 대응 규칙 실행 (경보, 차단 등 AWS Lambda활용)



2. 탐지 제어 – AWS GuardDuty

- 특이한 API 호출 또는 계정 침해의 가능성을 나타내는 잠재적 활동을 모니터링
- CloudTrail, VPC Flow Logs 및 DNS Logs에서 로그를 수집, 분석하여 지능형 위협 탐지 제공



3. 인프라 보호 > 네트워크 및 호스트 보호 > 디도스 방어

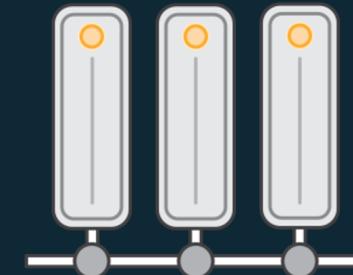
- AWS에 적용된 디도스 방어 체계



AWS 글로벌
인프라에 적용



상시 운영, 외부 라우팅
없는 신속한 방어



여분의 AWS 데이터
센터 인터넷 연결성

3. 인프라 보호 > 네트워크 및 호스트 보호 > AWS Shield



- AWS Shield standard

레이어 3/4 보호

자동 탐지 및 대응

가장 흔한 공격유형에 대한 방어
(SYN/UDP Floods, Reflection Attacks, 등)

AWS 서비스에 밀접한 결합

레이어 7 보호

레이어 7 디도스 공격 대응을 위해 AWS WAF 활용

셀프서비스 및 사용량 과금

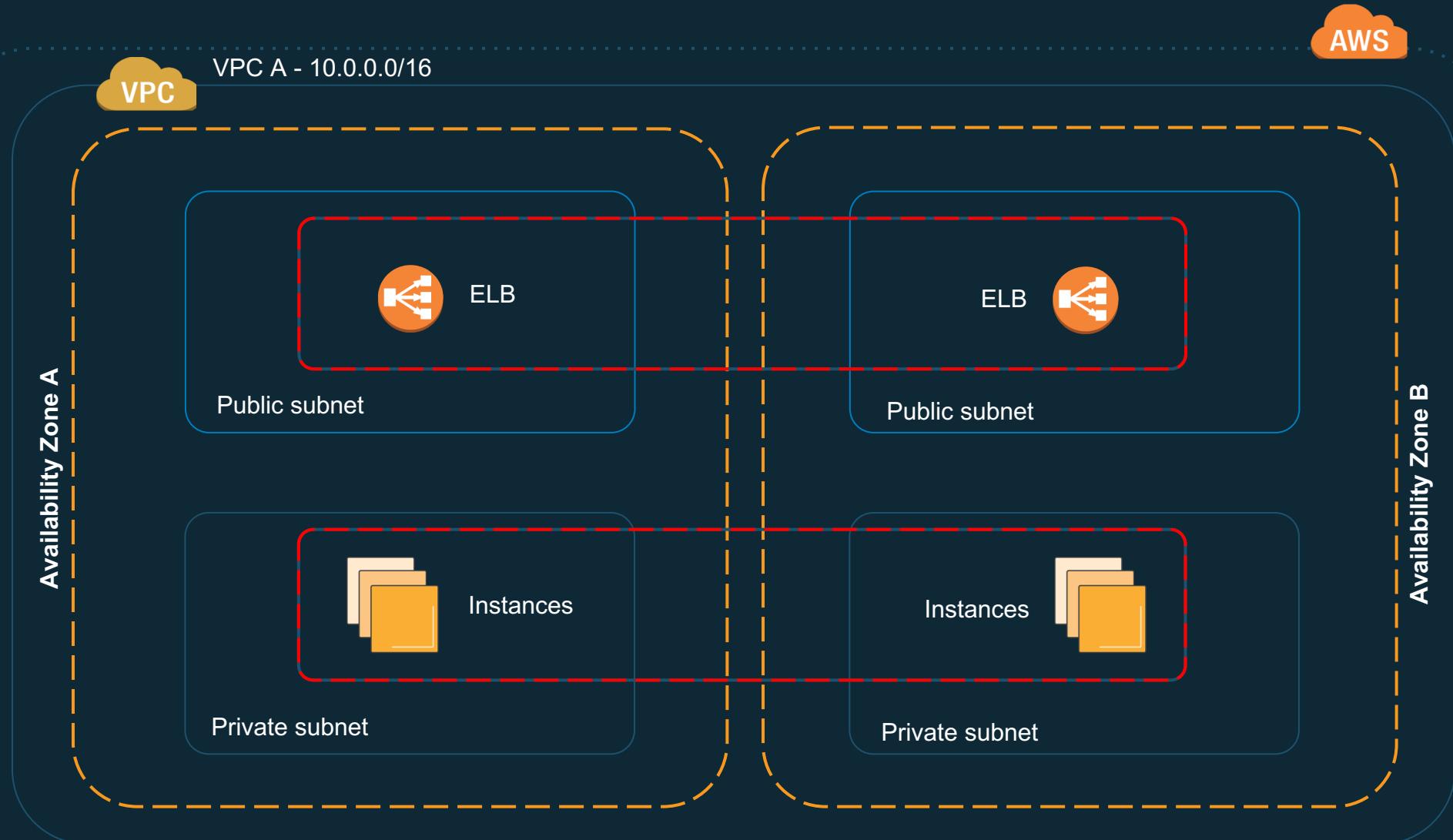
3. 인프라 보호 > 네트워크 및 호스트 보호 > AWS Shield



- AWS Shield Advanced

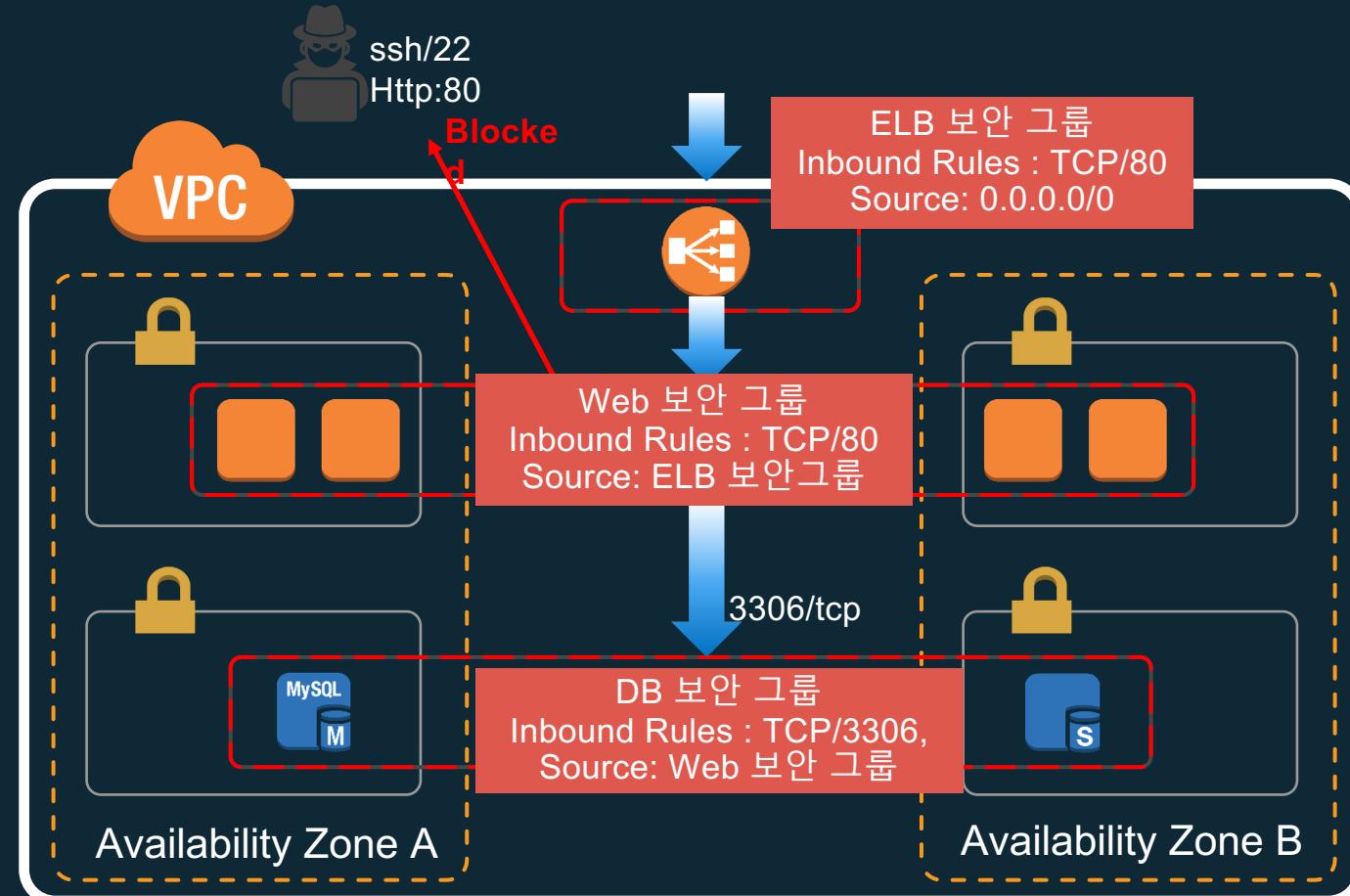


3. 인프라 보호 > 네트워크 및 호스트 보호 > VPC



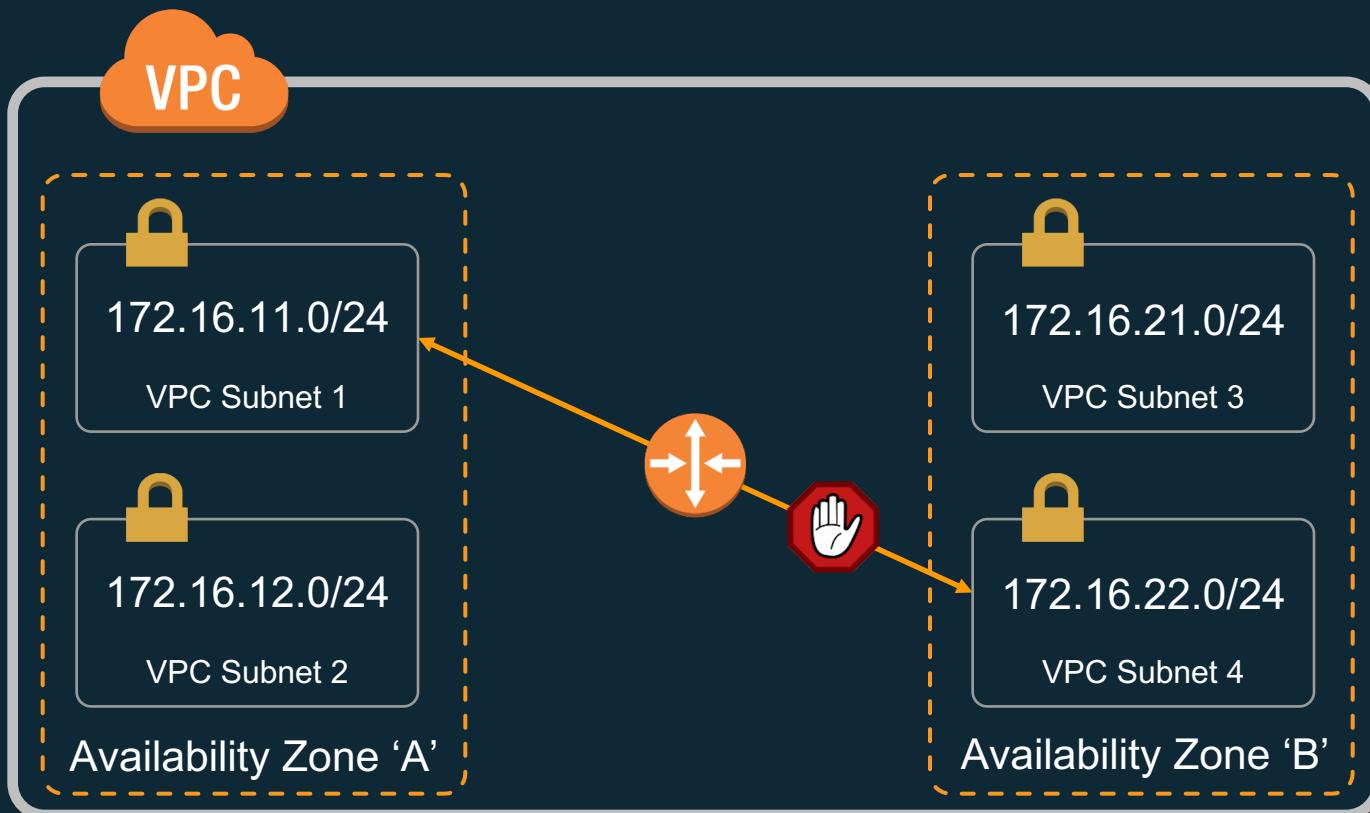
3. 인프라 보호 > 네트워크 및 호스트 보호 > 보안 그룹

- Inbound로 들어온 트래픽은 Outbound 모두 허용됨 (stateful)
- 각 티어에서 필수로 필요한 트래픽만 특정 소스에서 허용 권장



3. 인프라 보호 > 네트워크 및 호스트 보호 > NACL

- 인바운드, 아웃바운드 를 모두 설정 필수 (stateless)
- 특정 서브넷에서의 접근 원천 봉쇄 가능



subnet-a4f4f5cd Demo-public-2a						
Network ACL: acl-2f949946						
Inbound:						
Rule #	Type	Protocol	Port Range / ICMP Type	Source	Allow / Deny	
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW	
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY	
Outbound:						
Rule #	Type	Protocol	Port Range / ICMP Type	Destination	Allow / Deny	
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW	
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY	

3. 인프라 보호 > 네트워크 및 호스트 보호 > VPC Flow logs

- 별도 에이전트 없이 ENI, 서브넷 또는 VPC별로 로깅이 가능
- CloudWatch logs에 기록
- CloudWatch 알람과 연동하여 특이 증상 발생시 알람 설정 권장

Interface	Source IP	Source port	Protocol	Packet	Accept or reject
AWS account	Event Data				
▶ 2 41747	eni-b30b9cd5	119.147.115.32	10.1.1.179	6000	22 6 1 40 1442975475 1442975535 REJECT OK
▼ 2 41747	eni-b30b9cd5	169.54.233.117	10.1.1.179	21188	80 6 1 40 1442975535 1442975595 REJECT OK
▼ 2 41747	eni-b30b9cd5	212.7.209.6	10.1.1.179	3389	3389 6 1 40 1442975596 1442975655 REJECT OK
▼ 2 41747	eni-b30b9cd5	189.134.227.225	10.1.1.179	39664	23 6 2 120 1442975656 1442975716 REJECT OK
▼ 2 41747	eni-b30b9cd5	77.85.113.238	10.1.1.179	0 0 1 1	100 1442975656 1442975716 REJECT OK
▼ 2 41747	eni-b30b9cd5	10.1.1.179	198.60.73.8	512	123 17 1 76 1442975776 1442975836 ACCEPT OK

3. 인프라 보호 > 시스템 보안 구성 및 유지 보수 > Amazon Inspector

- EC2내 OS, 애플리케이션의 취약점 점검 및 규정 준수를 자동화하는 Agent 기반 보안 서비스인 **Amazon Inspector**를 통해 취약점이 경감된 최신 상태 유지 권장
- 다양한 Rule Package를 기반으로 취약점을 분석하고 적발된 내역에 대한 경감 조치와 대시 보드 제공
 - CVE (common vulnerabilities and exposures)
 - CIS (Center for Internet Security)
 - Runtime Behavior Analysis
 - Amazon security best practices – Network, Authentication, OS Configuration, Application

3. 인프라 보호 > 시스템 보안 구성 및 유지 보수 > AWS System manager

- 소프트웨어 인벤토리 자동 수집, OS 패치 관리, 운영 체제 구성을 도와주는 **System Manager**를 통해 시스템 구성을 정의하고, 추적, 유지 권장 (SSM Agent 설치)

배포, 구성 및 원격관리



Run Command



스테이트 매니저

공유 기능



유지 관리 기간



파라미터 스토어

추적 및 업데이트



인벤토리



패치 매니저



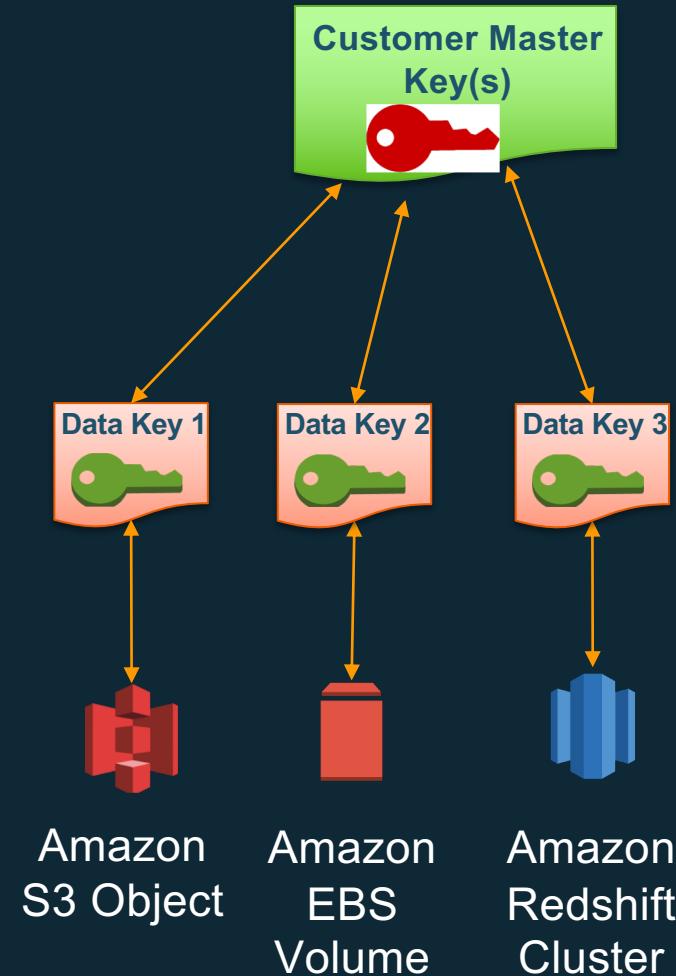
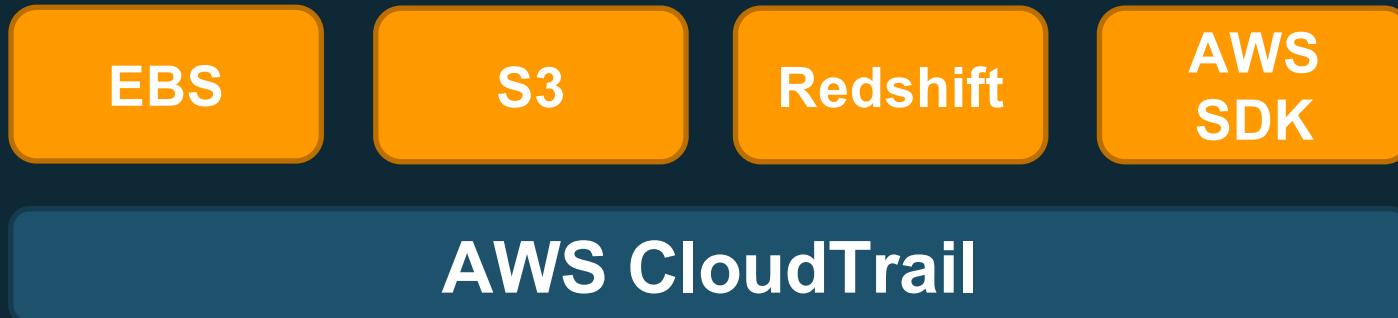
자동화

4. 데이터 보호 > 전송 중, 유휴 데이터 암호화



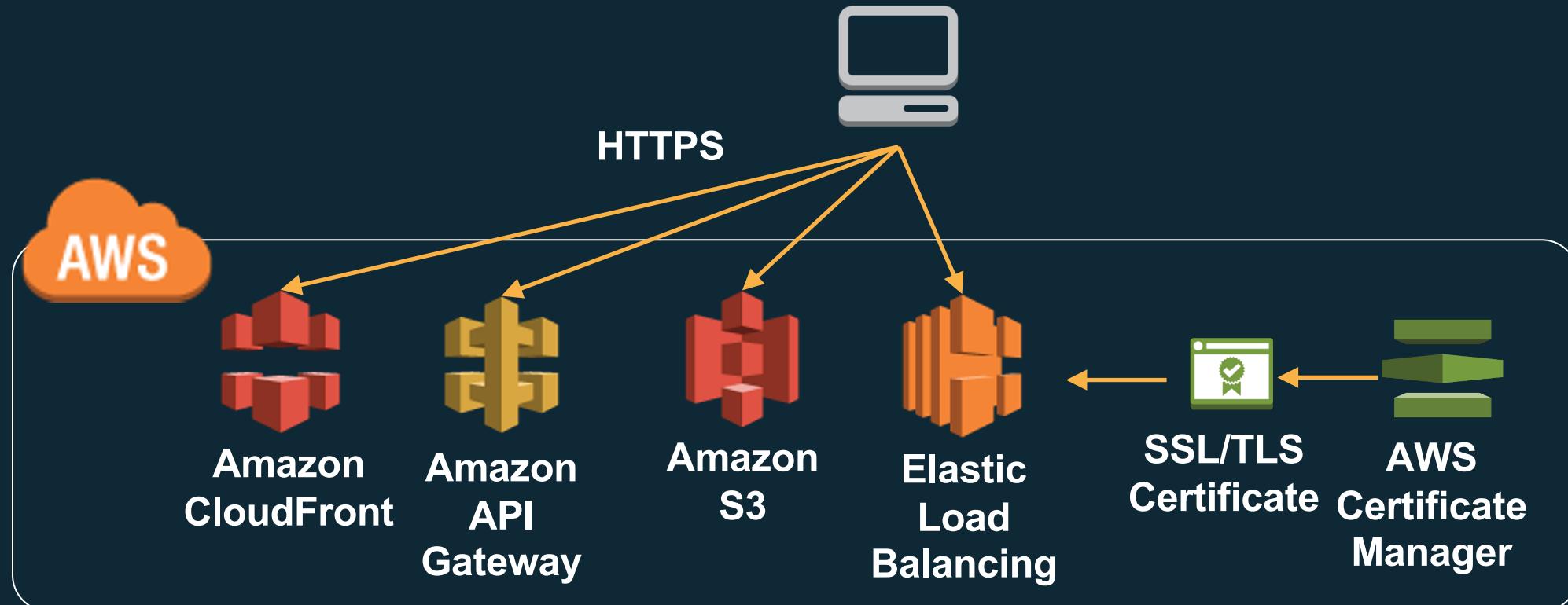
4. 데이터 보호 > AWS KMS를 활용한 키 관리

- 암호화키를 안전하게 생성/보관/관리 해주는 중앙 집중 관리 형 키 관리 서비스



4. 데이터 보호 > ACM을 통해 SSL/TSL 인증서 배포 및 관리

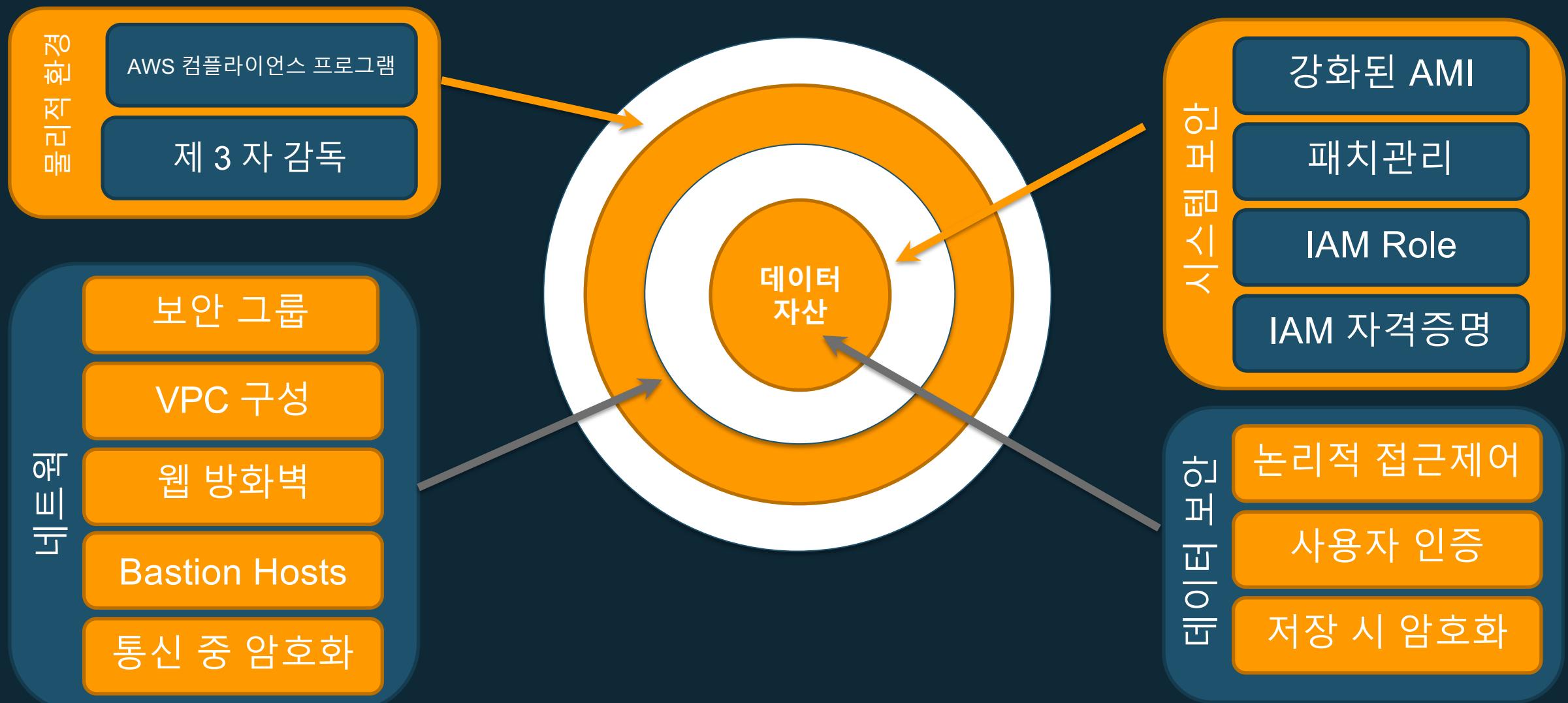
- AWS 내부 리소스에 사용할 공인 및 사설 SSL 인증서를 무료로 배포하고 자동화된 관리 및 갱신을 지원하는 ACM 활용 권장



5. 사고 대응

- 성숙한 예방 및 탐지 통제가 있어도 여전히 보안 사고의 잠재적 영향 대응 및 완화 위한 프로세스 마련 필수
- 사고 발생시 격리, 원인 분석을 수행함에 운영 환경 영향 최소화된 아키텍처 권장
 - 사고 발생시 적절한 사람이 사건을 격리하고 조사하기 위한 **도구, 엑세스** 권한 필요
 - 조사 중에 수행되어야 하는 작업 **자동화**
ex) 사고 발생 시 해당 인스턴스 보안 그룹 자동 변경 및 격리, AutoScaling으로 운영 영향 최소화
- 운영 환경 **CloudFormation 템플릿**을 미리 확보하여 빠른 재현 및 복구
- 적시에 조사 및 복구를 수행할 수 있도록 일상적인 사고 대응 훈련, **GameDay** 수행 권장

AWS는 모든 Layer에서의 보안 요소를 제공!



참고자료

<http://security.aws-korea.com/>



안정성
(Reliability)

안정성 디자인 원칙

- **복구 절차를 반드시 테스트하십시오**

다양한 형태의 장애상황을 만들고 복구하는 시나리오를 자동화하고, 이를 통하여 복구시 문제를 미리 파악하고 대처함으로써, 실제 문제 상황시의 복구 시나리오가 문제 없이 수행되도록 합니다.

- **문제 발생시 자동으로 복구될 수 있도록 시스템을 구성하십시오**

중요 요소들에 대한 모니터링을 수행하여, 임계치에 도달하였을 때 자동으로 필요한 절차를 수행하도록 시스템을 구성합니다. 이는 장애에 대한 감지, 추적, 조치 뿐만 아니라 더 나아가서 장애를 사전에 예방할 수 있도록 합니다.

- **시스템의 가용성을 높이기 위하여 수평 확장이 가능하도록 시스템을 구성하십시오**

하나의 큰 자원을 사용하는 것보다, 여러 개의 작은 자원들을 함께 사용하는 것이 장애 발생시 피해를 줄일 수 있어서 바람직합니다. 이 때, 여러 개의 작은 자원들은 공통된 장애 요소를 가지고 있지 않도록 구성합니다.

- **시스템 용량을 예측하려 하지 마십시오**

시스템의 용량을 초과하는 부하가 유입되는 경우 (종종 DDOS의 목적이 되기도 합니다), 자원의 고갈로 인하여 시스템이 동작하지 않게 됩니다. 클라우드에서는 유입되는 업무량과 시스템의 사용량을 모니터링하여 자원을 동적으로 할당하거나 제거하여 최적의 자원이 사용되도록 구성하여야 합니다.

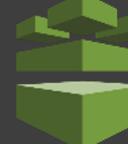
- **변경 관리를 자동화하십시오**

인프라 변경은 자동화를 통해 이루어져야 합니다. 관리되어야 하는 변경사항은 모두 자동화의 대상입니다.

AWS의 안정성 관련 서비스



Amazon CloudWatch

Areas	Key Services
Foundations	 IAM  Amazon VPC
Change Management	 AWS CloudTrail  AWS Config  AWS CodePipeline  AWS CodeDeploy
Failure Management	 AWS CloudFormation

안정성 – 기본 지식

9's

가용성(Availability) 이란 서버와 네트워크, 프로그램 등의 정보 시스템이 정상적으로 사용 가능한 정도로, 정상 동작 시간 / 총 시간에 대한 퍼센트로 표현됩니다.

강한 의존성에 대한 가용성 계산

각각 99.99%의 가용성을 가진 3개 시스템이 강한 의존성을 가진 경우,

$$99.99\% * 99.99 * 99.99\% = \text{약 } 99.97\%$$

중복된 구성요소에 대한 가용성 계산

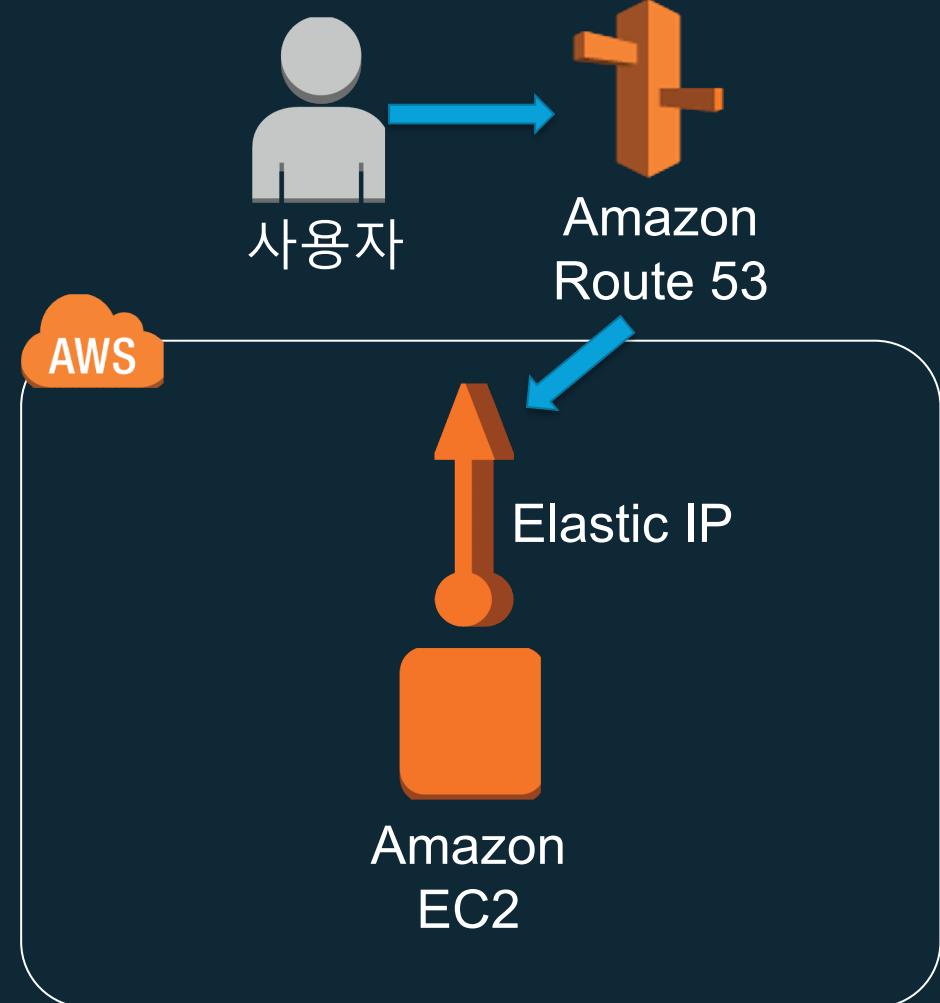
각각 99.9%의 가용성을 가진 구성요소 2개를 활용해 중복 구성한 경우,

$$100\% - (0.1\% * 0.1\%) = 99.999\%$$

가용성	최대 중단	애플리케이션 예시
99%	3일 15시간	Batch, ETL
99.9%	8시간 45분	사내 도구 (knowledge management, project tracking)
99.95%	4시간 22분	전자 상거래, POS
99.99%	52분	비디오, 방송 시스템
99.999%	5분	ATM 거래, 통신 시스템

아키텍처 진화 : 최소 구성

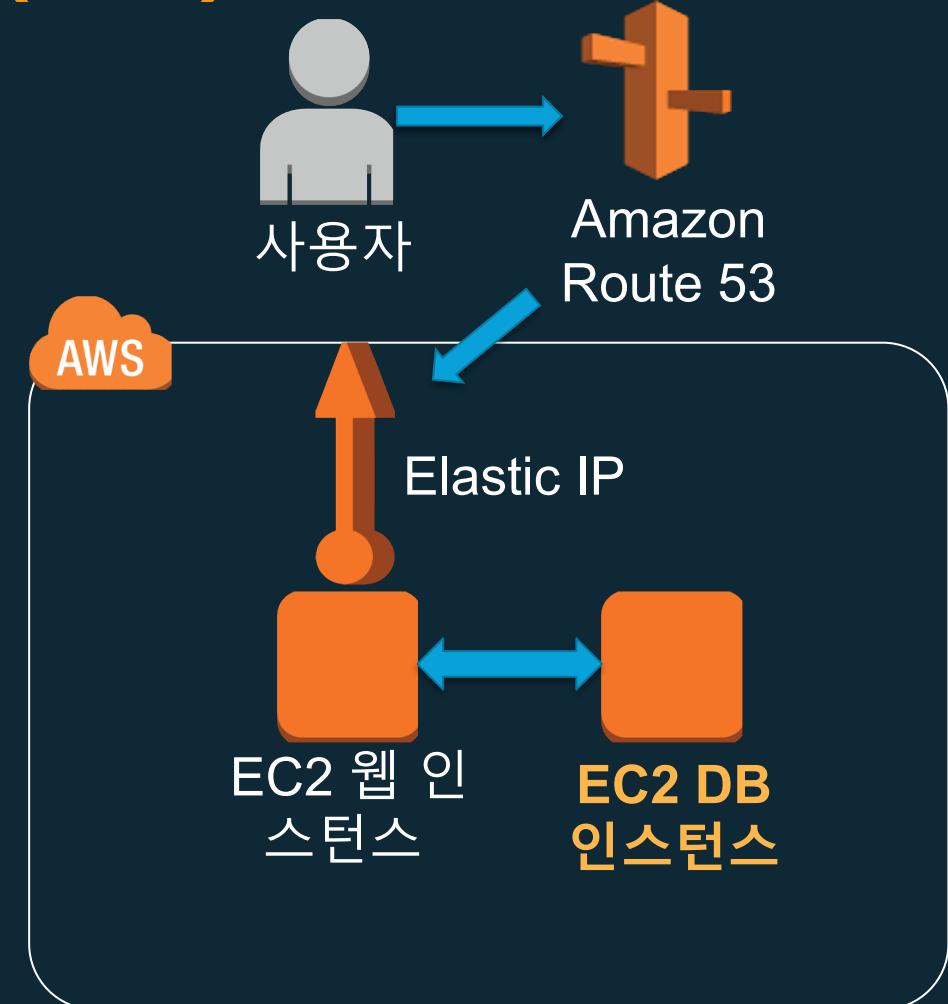
- 이중화 없음
- 장애 시 페일 오버 불가
- 한 곳에 모든 것이 존재



아키텍처 진화 : 2 tier 구성 (1/2)

먼저, 기능에 따라 인스턴스의 역할을 나눈다!

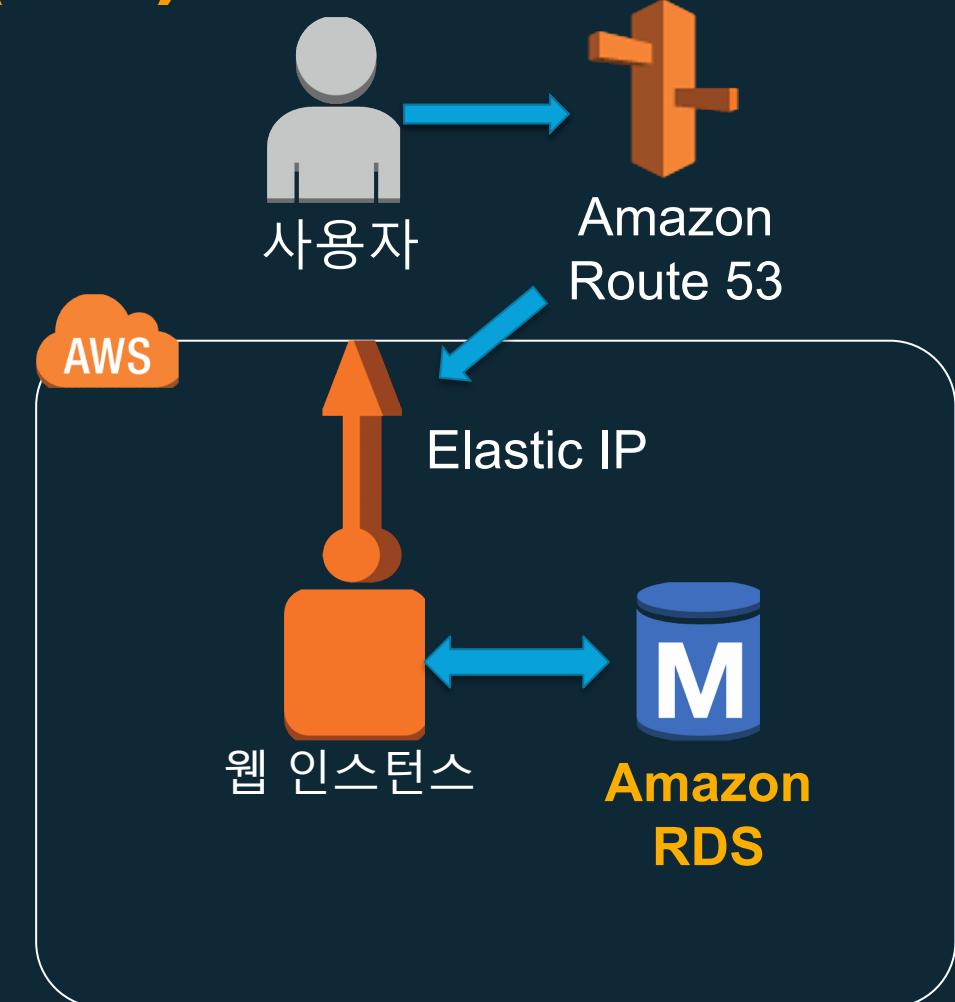
- 웹 서버용 EC2 인스턴스
- DB용 EC2 인스턴스
 - 직접 EC2에 DB 운영
 - DB 관리형 서비스 사용



아키텍처 진화 : 2 tier 구성 (2/2)

편리한 DB 운영을 위해
Amazon RDS 이용!

DB 백업, 패치 관리 대신 코
어 서비스 개발에 집중!



아키텍처 진화 : Multi-AZ 구성

Elastic Load Balancing

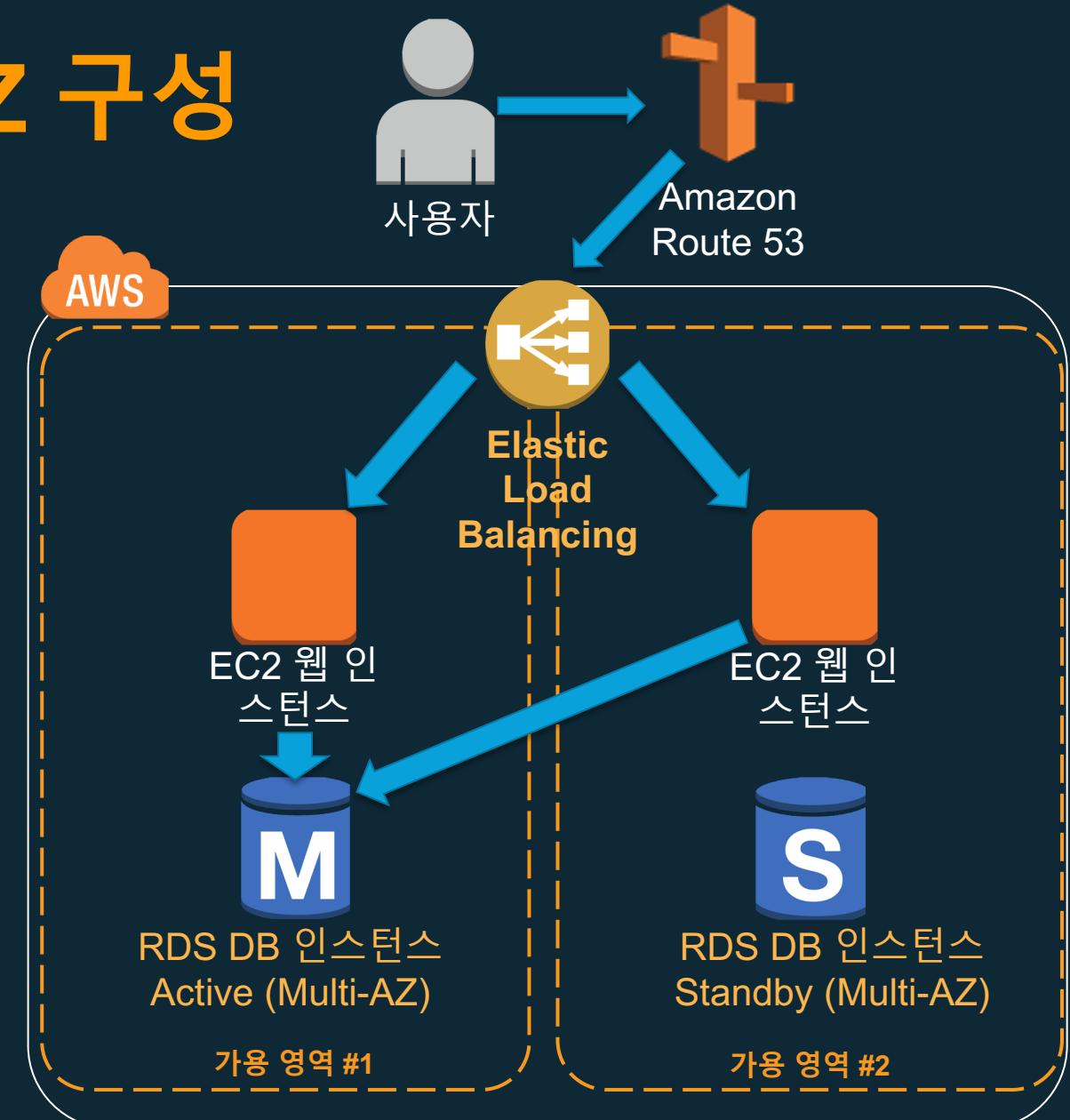
확장성 높은 부하 분산 서비스

Multi-AZ 서버 구성

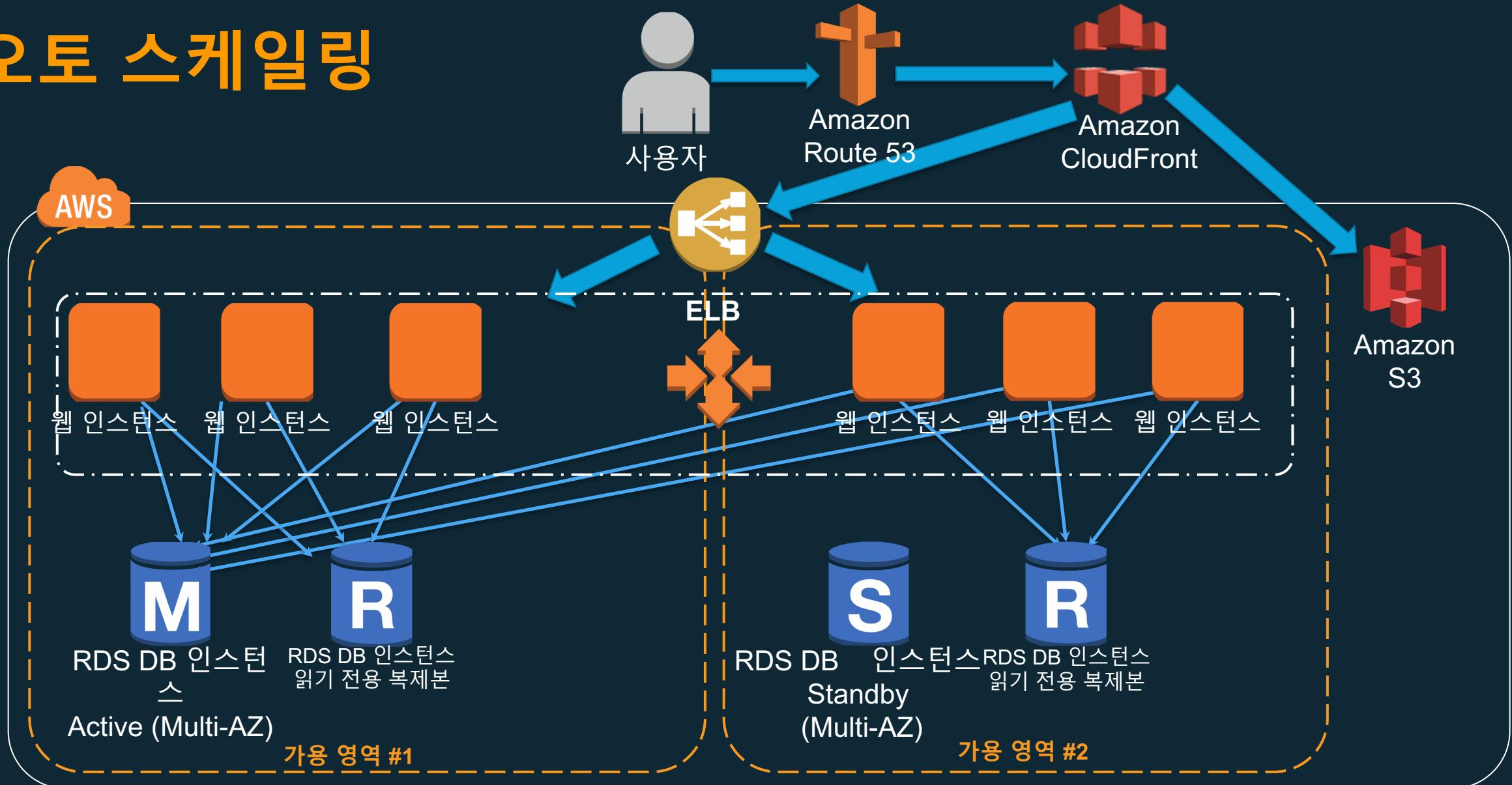
가용 영역을 통한 고가용성 확보

데이터베이스 이중화

RDS의 Active-Standby 복제본
Multi-AZ로 구성



오토 스케일링



1. 기본 요건 | Best Practices

- On-premise 환경에서는 네트워크 및 시스템 자원의 양을 미리 충분하게 준비하여야 합니다.
- AWS 클라우드를 사용하는 경우, 이러한 네트워크 및 시스템 자원의 기본 요건은 AWS가 관리하게 됩니다. AWS의 각 서비스의 limit를 파악하고, 사용중인 자원이 limit에 근접하는 경우 미리 조정하여야 합니다.

1. 기본 요건 | 제한 관리



AWS Trusted Advisor



Amazon CloudWatch

1. 기본 요건 | 네트워크 토폴로지 계획



Amazon VPC



AWS Direct Connect



VPN Gateway

2. 변경 관리 | Best Practices

- On-premise 환경에서의 변경 관리는 모든 과정이 수동으로 이루어지며, 프로세스가 잘 설계되어야만 합니다.
- AWS 클라우드를 사용하는 경우, 시스템의 상황이 모니터링되고 중요 요소들에 대한 자동화된 관리를 수행할 수 있습니다.
- 이러한 자동화된 관리에 대한 사용자 권한은 계획을 통하여 관리되어야 하며, 변경 내역은 자동으로 기록되어야 합니다.

2. 변경 관리 | 수요 변화에 따른 대처



Auto Scaling

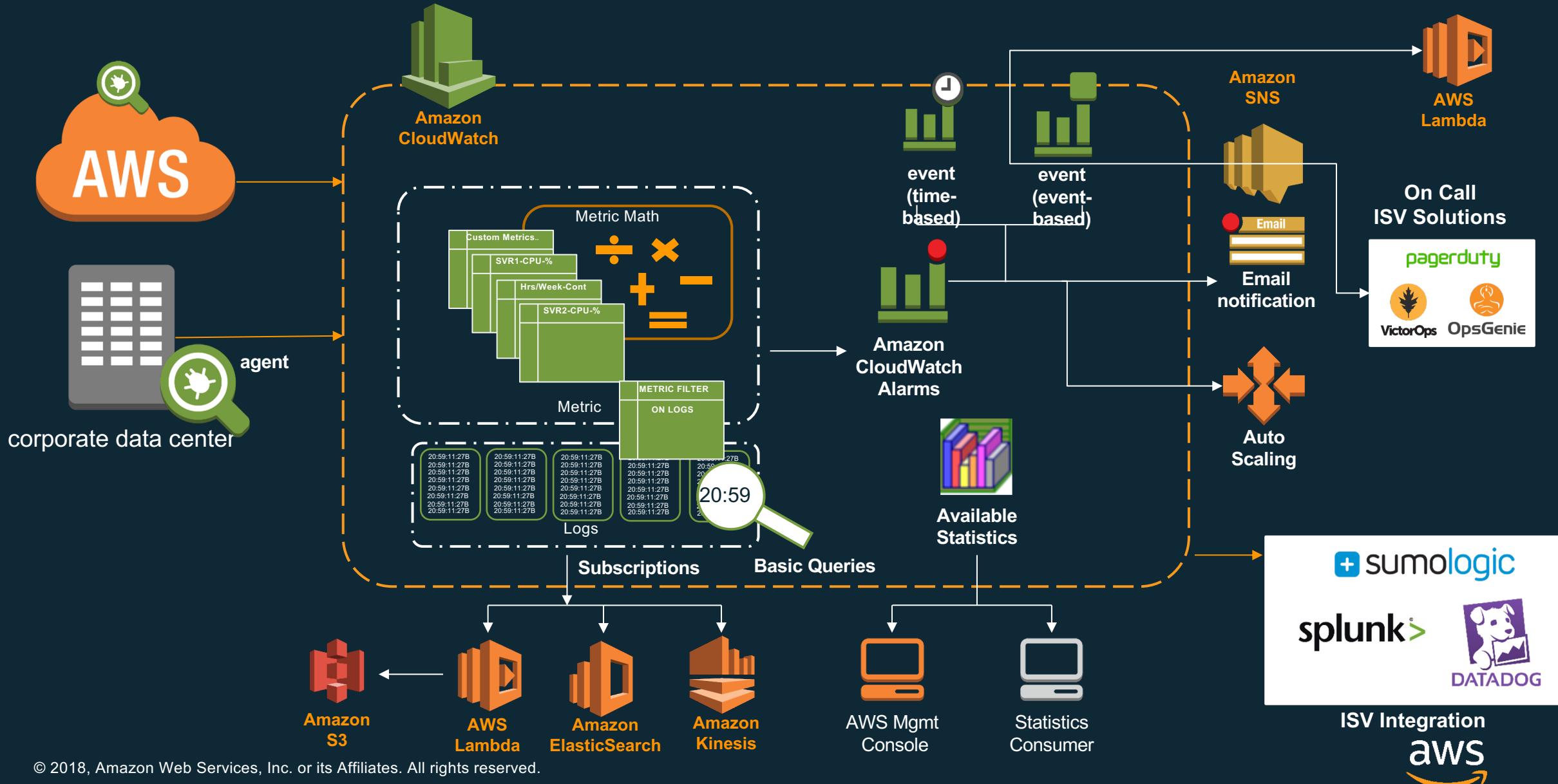


Amazon CloudWatch



Elastic Load
Balancing

2. 변경 관리 | 모니터링 - CloudWatch와 AWS 서비스 통합



2. 변경 관리 | 변경 관리 수행



AWS CloudFormation

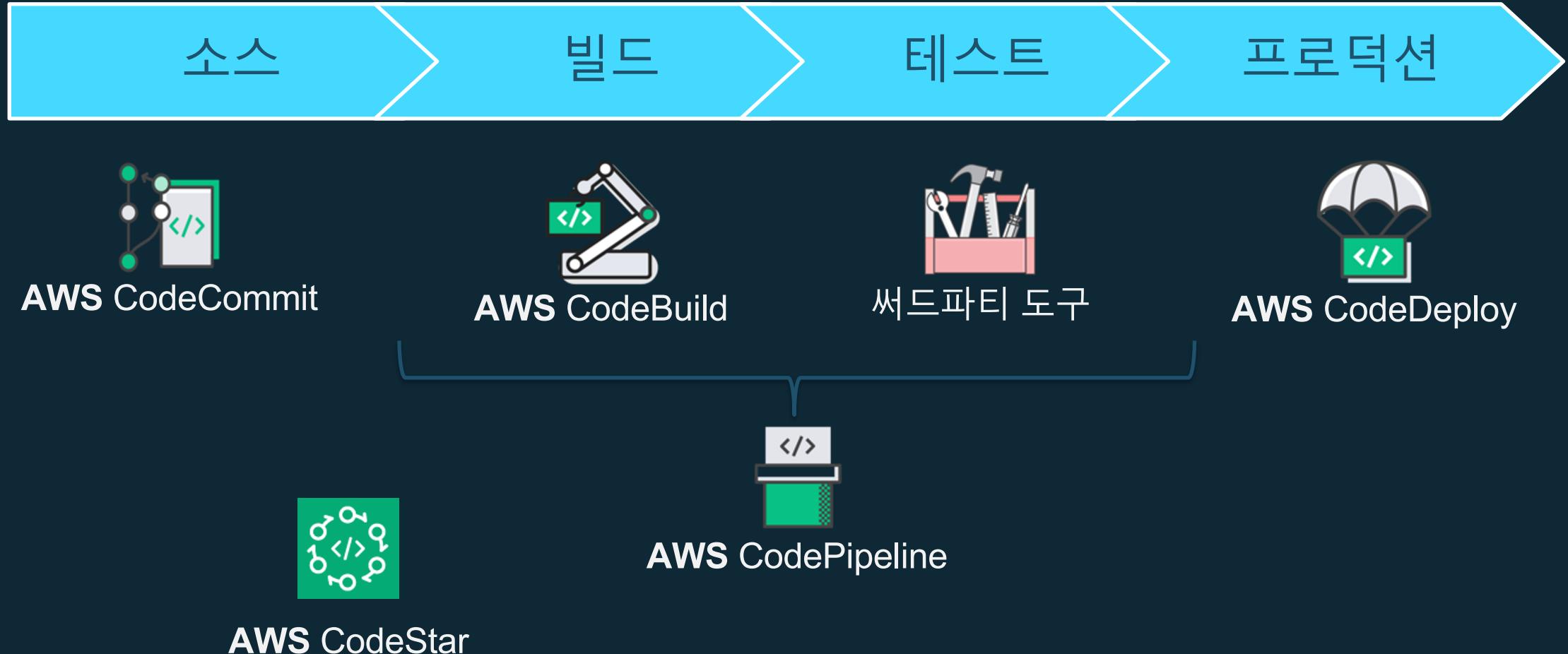


AWS CodePipeline

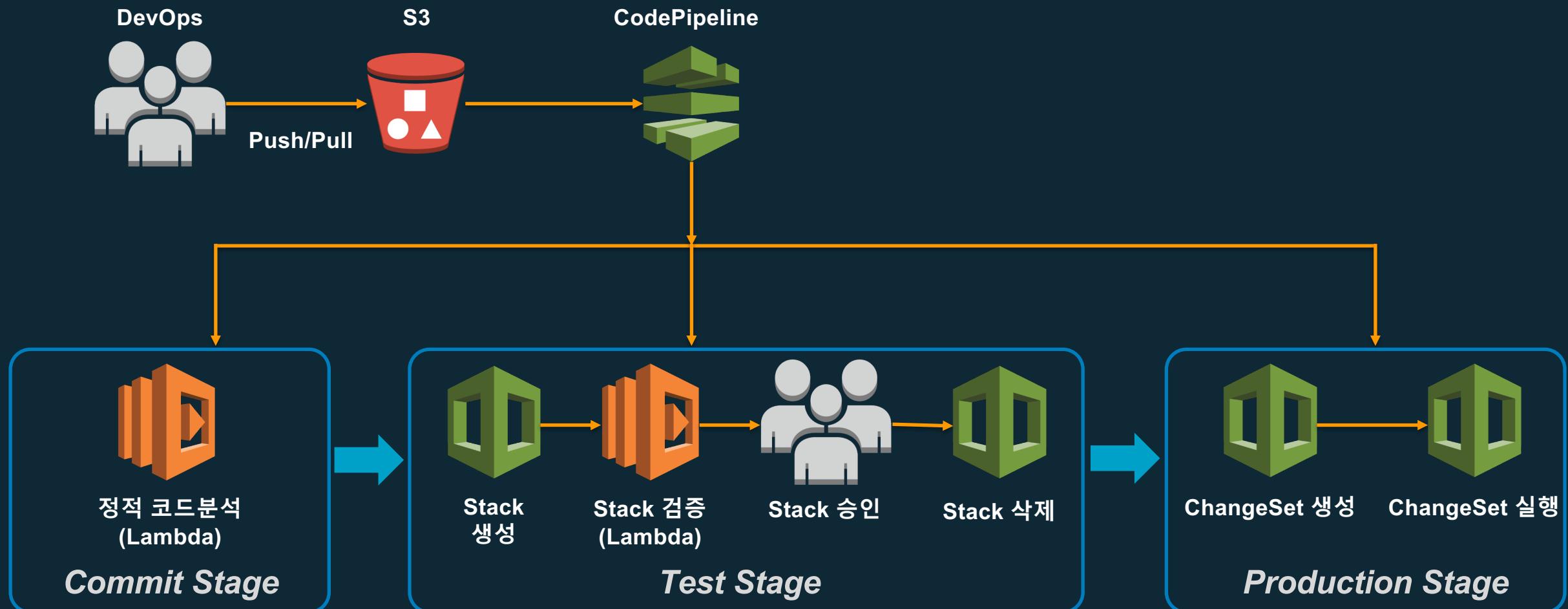


AWS CodeDeploy

AWS 코드 서비스



거버넌스 자동화



<https://aws.amazon.com/blogs/devops/implementing-devsecops-using-aws-codepipeline/>

3. 장애 관리 | Best Practices

- 어떠한 환경에서도 장애는 발생할 수 있습니다. 장애를 감지하고 조치하며, 다시 발생하지 않도록 예방하는 것은 매우 중요합니다.
- AWS에서는 데이터를 모니터링하여 상황에 자동으로 반응하도록 시스템을 구성할 수 있습니다. 또한 운영 시스템에 장애가 발생하였을 때, 이를 바로 분석하고 조치하는 것 보다는 장애가 발생한 시스템을 교체하고, 장애의 원인을 추후 분석하는 것이 바람직합니다.
- 이러한 장애 분석/복구에 대한 테스트를 수행하여, 운영 시스템에서도 문제없이 동작하도록 준비합니다.

3. 장애 관리 | 구성 요소 실패에 대한 내성



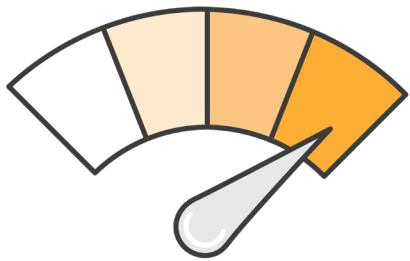
AWS SDKs



Elastic Load
Balancing



Amazon SQS



성능 효율성 (Performance Efficiency)

성능 효율성 디자인 원칙

- **최신 서비스 채택**

구현하기 어려운 최신 기술(NoSQL, EMR, AI/ML 등)도 서비스 형태로 쉽게 사용

- **즉시적 글로벌 서비스**

여러 리전과 CDN에 서비스를 배포하여 최소의 비용으로 최고 품질의 서비스를 글로벌하게 제공

- **서비스 아키텍처 사용**

서버 운영 부담, 확장성, 가용성 걱정 없이 완전 관리형 서비스를 이용해서 쉽게 웹 호스팅 또는 API 제작

- **더 자주 새로운 아이디어를 실험**

가상화 되고 자동화된 리소스를 통해 여러 종류의 인스턴스, 구성을 사용하여 쉽고 빠르게 그리고 반복적으로 실험

- **적절한 기술 접근법 사용**

데이터베이스, 스토리지 선택 시 데이터 액세스 패턴을 고려하고, 머신러닝 학습과 추론용 컴퓨팅 파워를 고려

성능 효율성 정의 | 1. 선택 – 컴퓨팅



서비스

Amazon EC2



Amazon ECS



AWS Lambda

스케일

VM

Task

Function

추상화

H/W

OS

Run-time

선택

서버부터 스토리지,
네트워크, OS까지
직접 설정

어플리케이션과 확장
성을 제어

빠르게 코드를 작성하
고 배포

성능 효율성 정의 | 1. 선택 – 컴퓨팅



워크로드

- General purpose
- Burstable
- Compute intensive
- Memory intensive
- Storage (High I/O)
- Dense storage
- GPU compute
- Graphics intensive

기능

- Choice of processor (AWS, Intel, AMD)
- Fast processors (up to 4.0 GHz)
- High memory footprint (up to 12 TiB)
- Instance storage (HDD and NVMe)
- Accelerated computing (GPUs and FPGA)
- Networking (up to 100 Gbps)
- Bare Metal
- Size (Nano to 32xlarge)

옵션

- Elastic Block Store
- Elastic Graphics
- Elastic Inference

거의 모든 워크로드와
비지니스 요구사항 중
족을 위해서

175
인스턴스 타입
인스턴스 세대 *추가적인 기능
인스턴스 패밀리
인스턴스 사이즈
인스턴스 타입



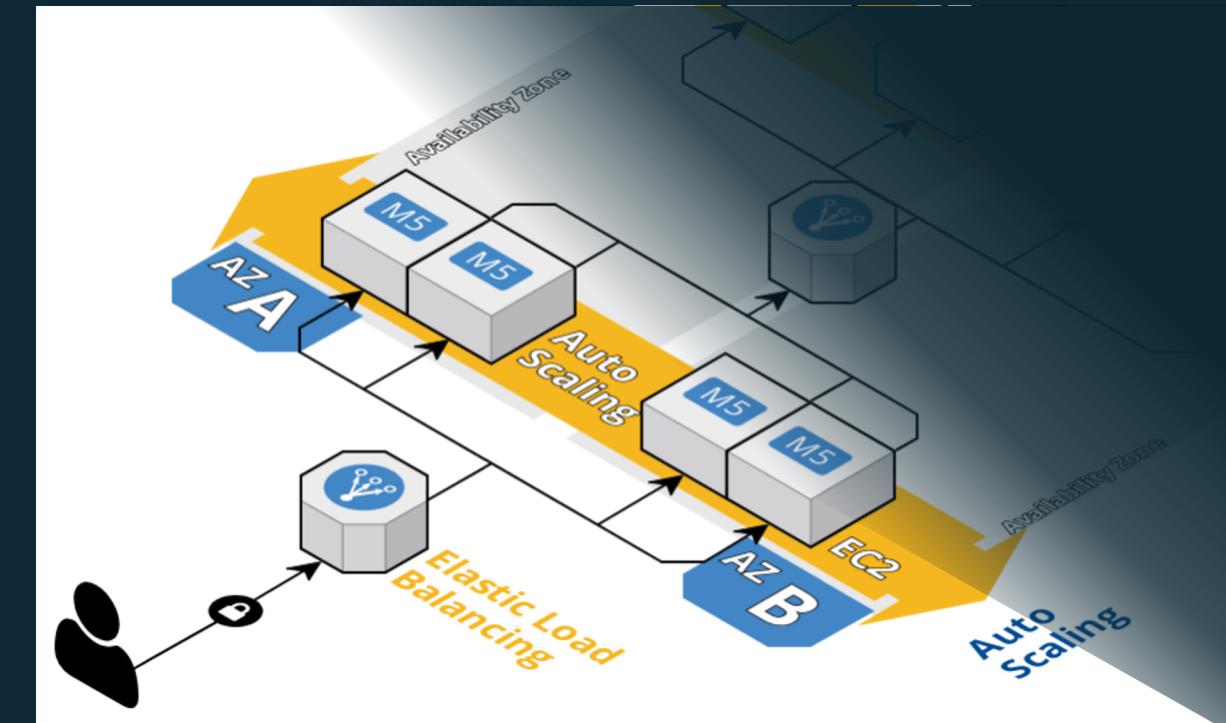
성능 효율성 정의 | 1. 선택 – 컴퓨팅



탄력적 컴퓨팅의 핵심 서비스는



Auto
Scaling



성능 효율성 정의 | 1. 선택 – 스토리지



Instance Store



Amazon
Elastic
Block
Store



Amazon
Elastic
File System



Amazon S3



Amazon
Glacier

블록

파일

객체

지연시간

Lowest, Consistent

Low, Consistent

Low-latency

Minutes to Hours

처리량

Single

Multiple

Web scale

High

공유 여부

Mounted on a instance

Many clients

Many clients

No

성능 효율성 정의 | 1. 선택 – 스토리지



	SSD (Solid State Drive)		HDD (Hard Disk Drive)	
볼륨 유형	프로비저닝된 IOPS SSD (io1)	범용 SSD (gp2)	처리량 최적화 HDD (st1)	콜드 HDD(sc1)
사용 사례	<ul style="list-style-type: none">I/O 집약적 NoSQL 및 관계형 데이터베이스	<ul style="list-style-type: none">부트 볼륨중소규모 DB개발 및 테스트	<ul style="list-style-type: none">빅 데이터데이터 웨어하우스로그 처리	<ul style="list-style-type: none">빈번하게 접근하지 않는 콜드 데이터
API 이름	io1	gp2	st1	sc1
볼륨 크기	4GB – 16TB	1GB – 16TB	500GB – 16TB	500GB – 16TB
볼륨당 최대 IOPS**	64,000	16,000	500	250
볼륨당 최대 처리량	1,000MB/초	250MB/초	500MB/초	250MB/초

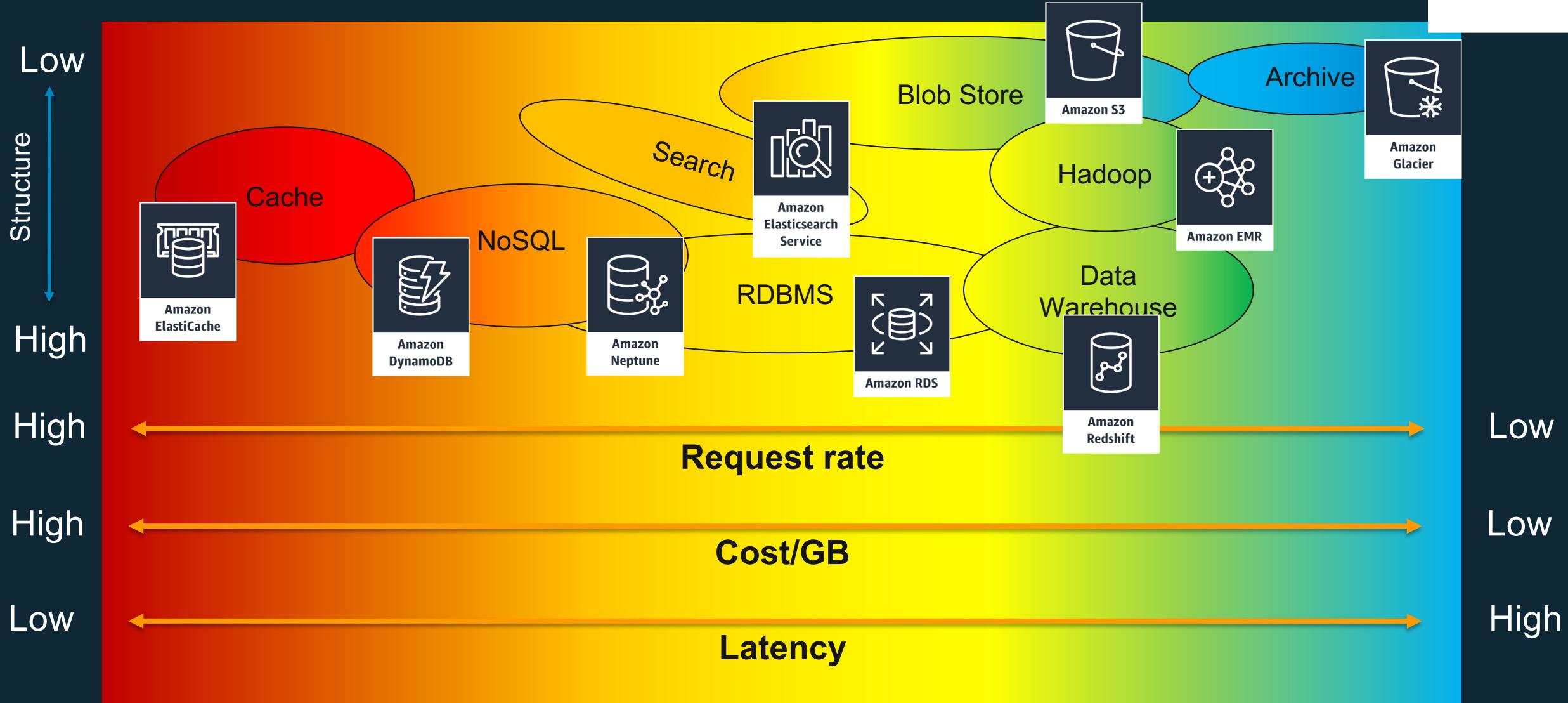
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html>

성능 효율성 정의 | 1. 선택 – 데이터베이스



데이터 셋 크기	GB 또는 TB	무제한	GB	PB
데이터 모델	관계형	문서	Key-Value 또는 문서	관계형
시멘틱스	ACID	독립적인 테이블	Transient	Unenforced constraints
쿼리 정합성	중간에서 높음	제한 없음	높음	낮음
확장성	인스턴스	읽기/쓰기량	인스턴스 및 클러스터 크기	인스턴스 및 클러스터 크기

성능 효율성 정의 | 1. 선택 – 데이터베이스



성능 효율성 정의 | 1. 선택 - 네트워크



22개 리전
69개 사용영역

지리적으로 인접한
곳에서 컨텐츠 제공

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Amazon
Route 53



Amazon
CloudFront



AWS Direct
Connect

Global CDN
160개 지점

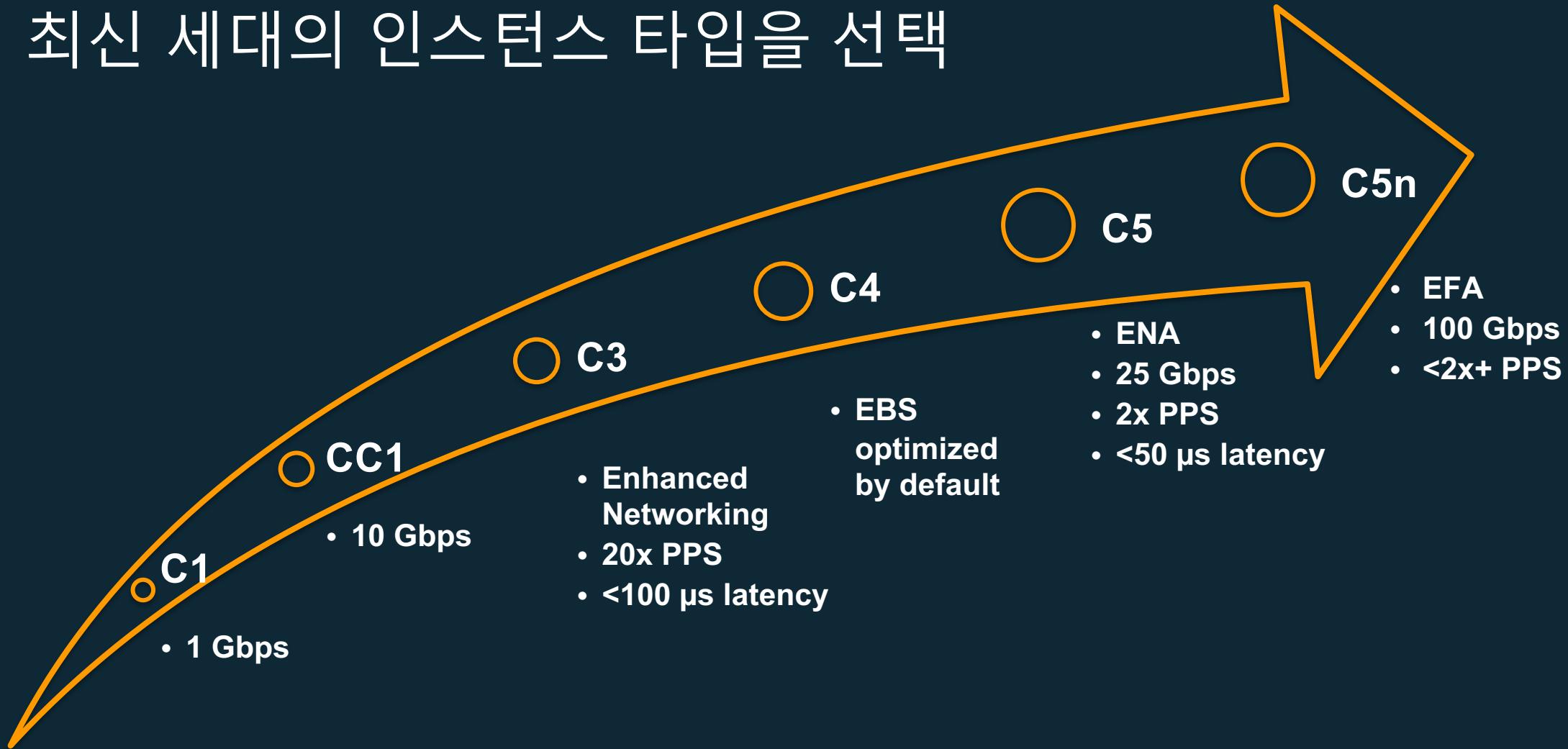
42개국 84개 도시에서
205개의 엣지 로케이션과
11개의 리전별 엣지 캐시

aws

성능 효율성 정의 | 1. 선택 - 네트워크



최신 세대의 인스턴스 타입을 선택



성능 효율성 정의 | 2. 검토 – 벤치마킹



코드 기반의 인프라 작성



AWS
CloudFormation

CI/CD를 위한 배포 파이프라인 구축, 성능 테스트 자동화



AWS
CodeCommit



AWS
CodeBuild



AWS
CodeDeploy



AWS
CodePipeline

성능 효율성 정의 | 2. 검토 – 부하 테스트



잘 정의된 지표를 이용한 CloudWatch 데이터 수집



Amazon
CloudWatch

직접 벤치마킹 테스트를 만들거나 업계 표준 테스트를 사용



alarm

프로덕션과 같은 환경으로 공격적인 부하 테스트



Spot
Instance

유저 스토리에 기반하여 케이스를 테스트

SaaS/Marketplace 솔루션 활용



성능 효율성 정의 | 3. 모니터링 – 단계



1. 생성 – 모니터링 범위, 지표, 임계치 확인
2. 집계 – 다양한 소스로부터 오는 데이터를 수집
3. 실시간 처리 및 알림 – 문제 인식 및 대응
4. 저장 – 데이터 관리 및 보존 정책

1. 분석 – 대쉬보드, 리포트 작성 및 통계



Amazon CloudWatch



Amazon S3

성능 효율성 정의 | 3. 모니터링 - 단계



클라우드 리소스 및 애플리케이션의 완벽한 가시성

- 응용 프로그램 모니터링
- 성능 변화에 대응
- 리소스 활용 최적화
- 운영 상태에 대한 통일된 뷰 얻기



생산, 분석 된 어플리케이션 분석 및 디버깅

- 성능 병목 현상 파악
- 근본 원인을 해결하십시오.
- 사용자 요청 추적
- 간단하고 복잡한 응용 프로그램

성능 효율성 정의 | 4. 트레이드오프 - 설명



이슈 상황	기술	활용	개선	관련 서비스		
읽기 부하	캐싱 읽기 복제	메모리 (공간)	시간	 Amazon CloudFront	 Amazon ElastiCache	 RDS DB Instance Read Replica
쓰기 부하	샤딩 파티셔닝	사이즈 복잡도	시간		 Amazon DynamoDB	
대용량	압축	시간	공간	 Amazon CloudFront	 Amazon Redshift	 AWS Snowball*
대량 요청	버퍼링	공간 시간	효율성	 Amazon SQS	 Amazon Kinesis	



비용 최적화 (Cost)

비용 최적화 디자인 원칙

- **데이터 센터 운영에 필요한 비용 제거**
 - 데이터 센터 내의 IT 인프라 관련 작업 제거
 - 고객 사업 프로젝트에만 집중 가능
- **소비 모델 채택**
 - 실제로 소비하는 컴퓨팅 리소스에 대해서만 지불
 - 정교한 예측이 아닌 비즈니스 요구 사항에 따른 사용량 증감
- **전체 효율 측정**
 - 시스템의 비즈니스 산출물과 이에 관련된 비용 측정
 - 측정 값을 이용하여 비즈니스 산출물을 증가시키고 비용은 절감할 수 있는 이점 확인
- **비용 분석 및 부과**
 - 클라우드는 시스템 사용 및 비용을 정확하게 파악할 수 있는 기능 제공
 - 개별 비즈니스 소유자에게 IT 비용을 투명하게 부과 가능, 투자 수익률(ROI) 측정 및 리소스를 최적화하고 비용을 절감할 수 있는 기회 제공
- **관리형 서비스를 사용하여 소유 비용 절감**
 - 이메일 전송이나 데이터베이스 관리 작업과 같은 서버 운영 유지 및 관리의 부담 제거
 - 관리형 서비스는 클라우드 규모에서 운영되기 때문에 트랜잭션 또는 서비스당 비용이 저렴

비용 최적화 전략

1

Right Architecture

Monolithic vs MSA

Host-base vs Container-base

Server-base vs Serverless-base

2

Right Service

IaaS

PaaS

SaaS

3

Right Size

Right Config

Right Volume

Right Scale Model

4

Right Cost Model

On-demand

Reserved Instance

Spot Instance

최적의 자원을 선택해야 합니다.



EBS 프로비
저닝된
IOPS
“io1”



EBS
범용
“gp2”



EBS
처리량에 최
적화된 “st1”



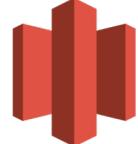
EBS
Cold HDD
“sc1”



S3
Standard



S3
IA



Glacier



Amazon
RDS



Amazon
DynamoDB



Amazon
ElastiCache

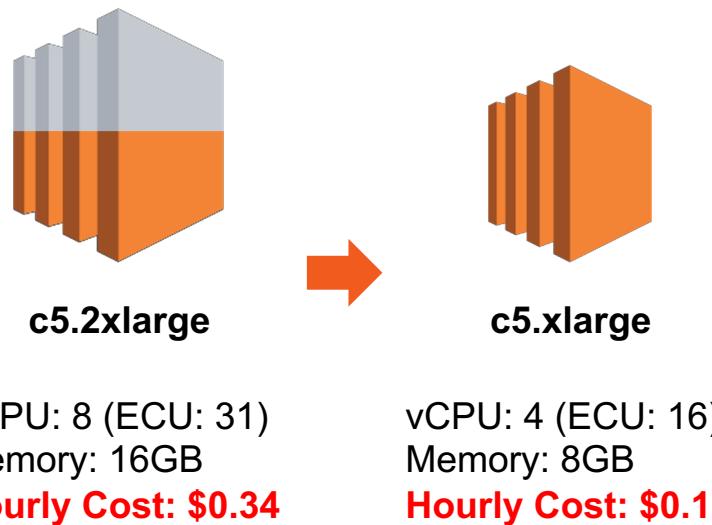


Amazon
Redshift

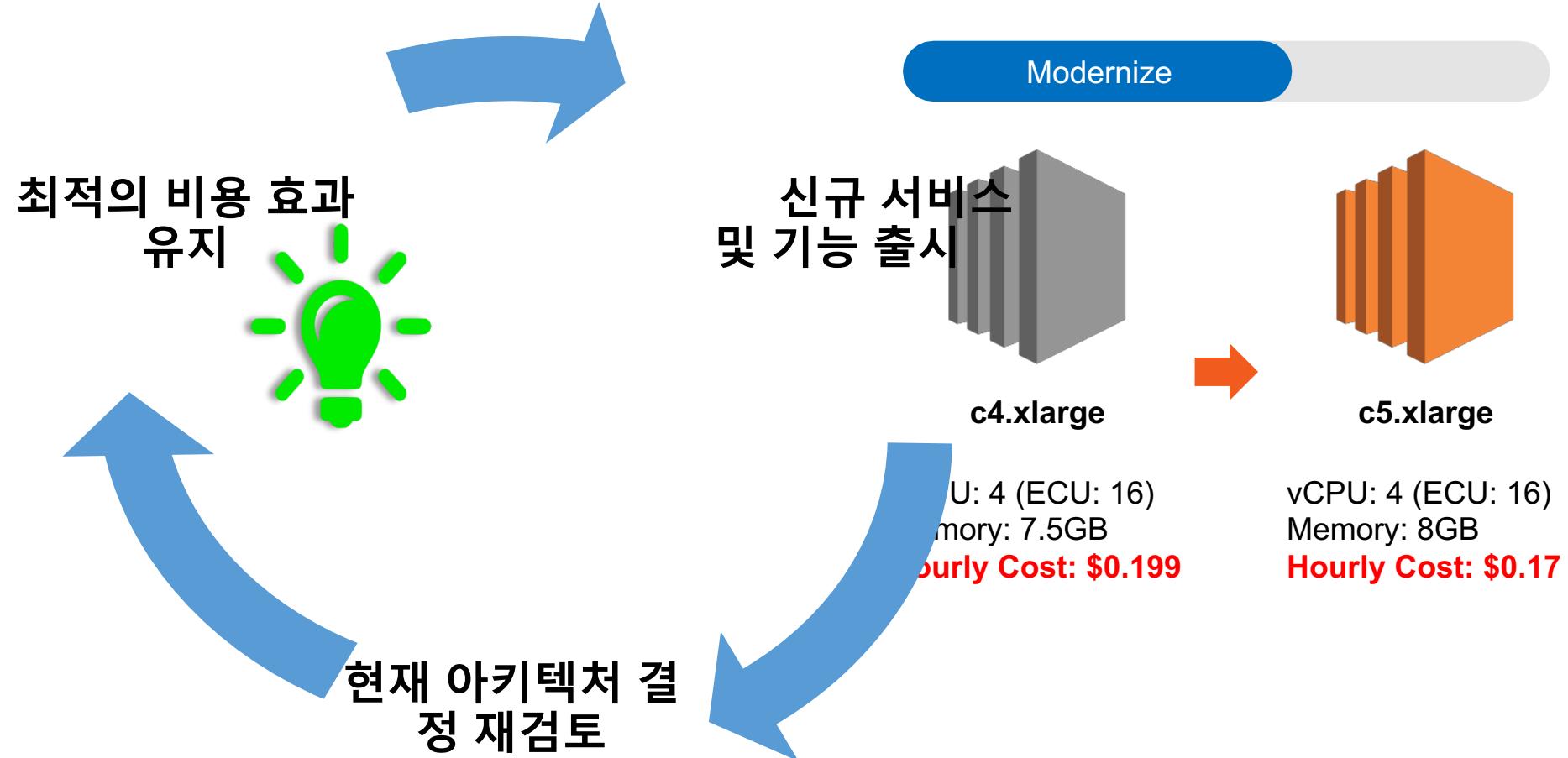
적합한 크기의 자원을 투입해야 합니다.

Performance Pattern 분석을 통해 현재 Workload 에 적합한 Type, Size 로 변경

Downsize



최신 자원을 투입해야 합니다.



장기적으로 사용할 리소스는 장기 계약으로 구매합니다.

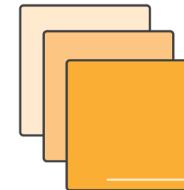
•최대 75% 이상 할인 혜택*
(+ 용량 예약)

약정 기간

1년
3년

RI 제공 AWS 서비스들*

Amazon EC2
Amazon RDS
Amazon DynamoDB
Amazon Redshift
Amazon ElastiCache
Amazon CloudFront**



* 특정 AWS 서비스, 크기, 유형 및 리전에 따라 다를 수 있음
** 예약 용량

Savings Plans (SP)

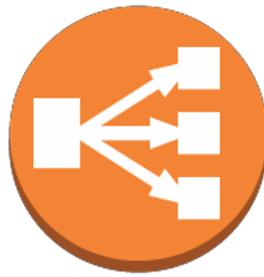
항목	Reserved Instance (RI)	Savings Plans (SP)
유형	Standard RI, Convertible RI	EC2 Instance SP, Compute SP
워크로드	안정적인 워크로드	안정적인 워크로드
대상	EC2, RDS, ElastiCache, Redshift, Elasticsearch	<ul style="list-style-type: none"> - EC2 Instance SP: EC2 Instance Family - Compute SP: EC2, Fargate, lambda
유연성	<ul style="list-style-type: none"> - Size (Linux OS only) - Instance Type, OS, Tenancy : Convertible RI 만 <p>* Instance Type 변경 시, 수동으로 계약 변경 (Convertible RI)</p>	<ul style="list-style-type: none"> - Size, OS, Tenancy - Instance Type, Region, EC2, Fagate : Compute SP 만 <p>* Instance Type 변경 시, 자동으로 계약 변경 (Compute SP)</p>
약정	약정 기간	1년, 3년
	약정 단위	<p>시간 당 인스턴스 용량</p> <ul style="list-style-type: none"> * 할인 후 금액 기준 * 최소 \$0.001/hr commit * 연단위, 월단위 금액 commit 아님
할인율	<ul style="list-style-type: none"> - Standard : 최대 72% - Convertible: 최대 66% 	<ul style="list-style-type: none"> - EC2 Instance SP: Standard RI 할인율과 동일 - Compute SP : Convertible RI 할인율과 동일

Savings Plans (SP)

항목	Reserved Instance (RI)	Savings Plans (SP)
결제 옵션	No Upfront, Partial Upfront, All Upfront	RI와 동일한 결제 옵션 제공
구매 방법	각 서비스(EC2, RDS 등) 별 페이지에서 구매 * RI 추천 페이지와 RI 구매 페이지 분리	Cost Explorer 및 EC2 페이지에서 구매 가능 (API로도 구매 가능) * SP 추천 페이지와 SP 구매 페이지를 하나로
할인 공유	Payer 계정 별로 공유 가능. 공유 대상 계정 지정 가능.	RI와 동일한 기능 제공 불가
취소 및 환불	불가	* 단 SP 발표 직전에 RI 구매한 경우, one-time 성으로 case open하여 취소 처리 - 1년 약정: 30일 이내, 3년 약정: 60일 이내
제약 사항		- 중국 리전(Beijing, Ningxia) 지원하지 않음 - Volume Discount 없음
기타		- 기존 RI를 SP로 변환 불가 - RI 할인 우선 적용 → SP 할인 적용 - EDP 할인 중복 가능 - RI와 분리되어 별도의 row로 제공 - SP 소유 계정과 적용 계정 구분 (기존 RI와 동일)

<https://aws.amazon.com/ko/blogs/aws/new-savings-plans-for-aws-compute-services/>

사용하지 않는 리소스는 제거합니다.



ELB



EIP



EBS

EC2 에 연결되지 않은 ELB

EC2 에 연결되지 않은 EIP

EC2 에 연결되지 않은 EBS

Stop 상태인 EC2 에 연결된 ELB

Stop 상태인 EC2 에 연결된 EIP

Stop 상태인 EC2 에 연결된 EBS

Auto Scaling 이 실시되지 않은 ELB

non-production 인스턴스는 사용하는 순간에만 켭니다.

개발/테스트 환경의 인스턴스 끄기

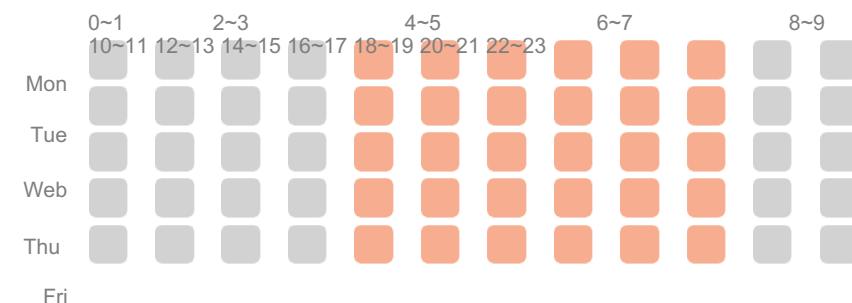
- 개발 및 테스트 환경이나 프로덕션 환경이 아닌 곳에서 항상 켜 있는 인스턴스를 찾아 끄기

“월 720 시간” vs. “월 160 업무 시간”

꺼 놓기만 해도, **80%** 절약

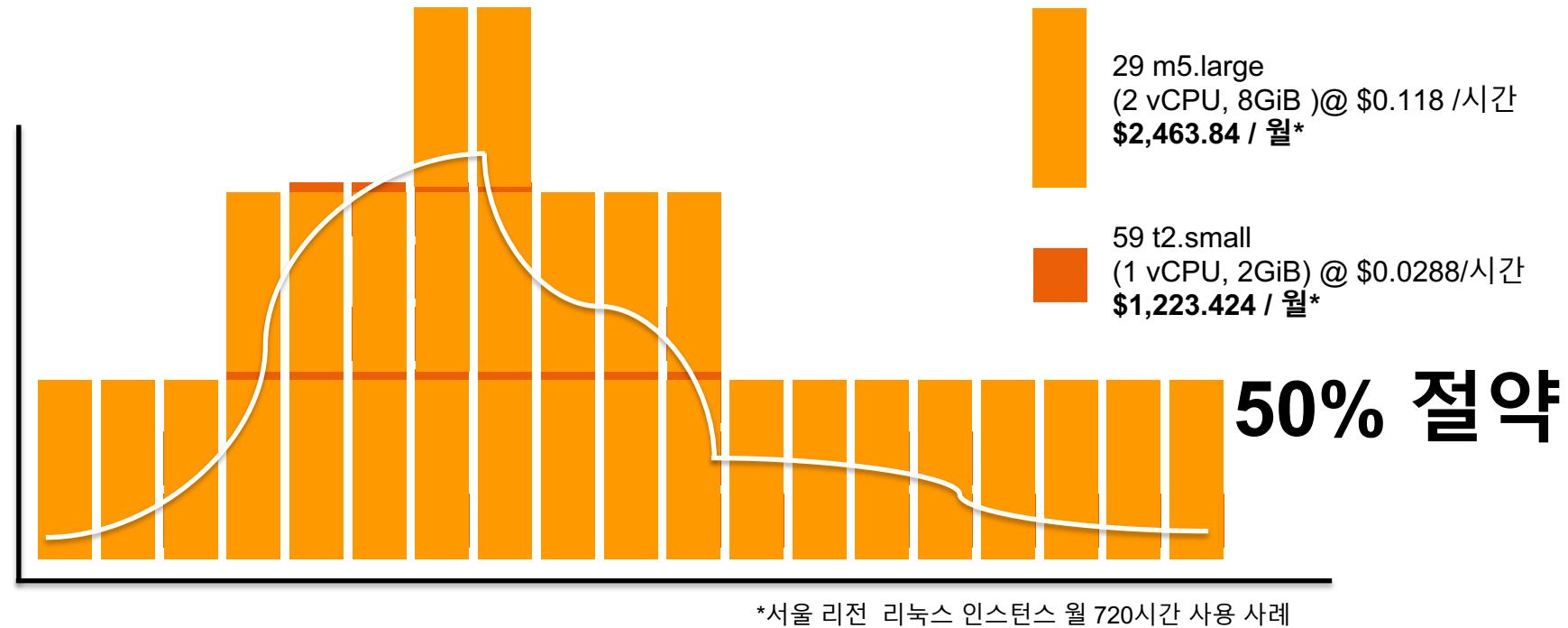
Scheduling

항상 켜져있는 Non-Production Instance 를
사용 하는 시간에만 켜기



AutoScale은 최대한 작은 인스턴스로 구성

작은 인스턴스 vs. 큰 인스턴스



비용 효율적인 리소스를 위한 모범 사례



관리형 서비스

이미 있는 기능을 다시 발명하지 마십시오!

- 가능하면 AWS가 제공하는 애플리케이션 서비스를 사용하세요

- 이메일
- 큐잉
- 트랜스코딩
- 검색
- 데이터베이스
- 모니터링
- 메트릭
- 로깅
- 컴퓨트



AWS Lambda



Amazon SNS



Amazon CloudSearch



Amazon SQS



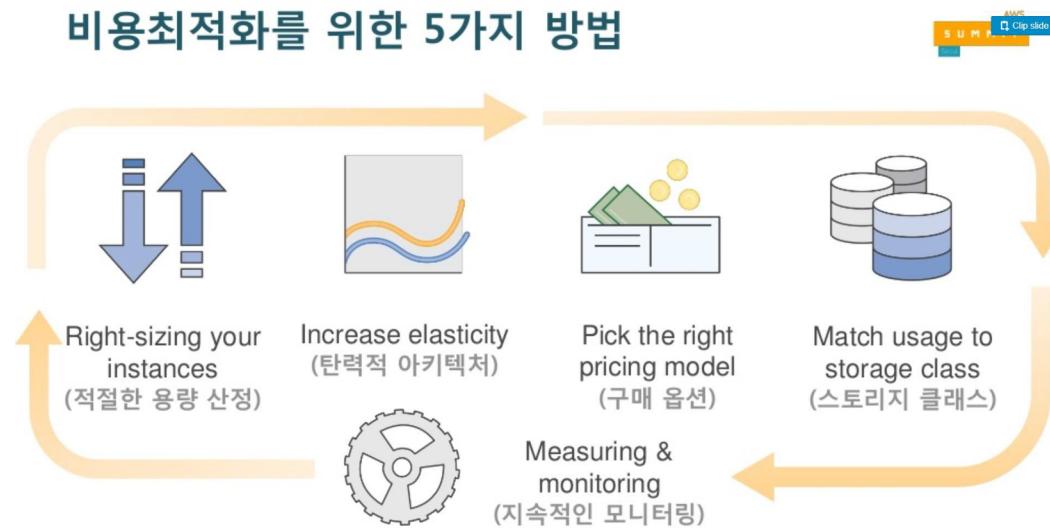
Amazon SES



Amazon SWF



Amazon Elastic
Transcoder



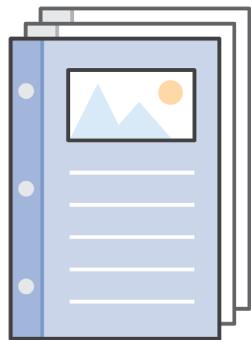
비용 최적화는

1회성으로 끝나서는 안되며

지속적인 모니터링을 통해

자원 및 비용의 비효율성을 수정해
가야함

이를 위한 인력과 시간이 필요함



운영 우수성 (Operational Excellence)

운영 우수성 디자인 원칙 (1/2)

- **코드를 통한 운영**

클라우드에서는 어플리케이션 코드를 위해 사용하였던 엔지니어링 원칙을 인프라를 포함하여 적용할 수 있습니다. 워크로드를 코드로 정의하고 수정함으로써 이벤트에 대응하는 운영을 자동화하고 휴먼에러를 최소화하십시오.

- **문서 Annotation(주석)**

온 프레미스 환경에서는 수작업으로 문서가 작성되어 변경사항을 업데이트하기가 쉽지 않았습니다. 클라우드에서는 빌드작업 후 문서생성을 어노테이션을 통해 자동화할 수 있고, 코드화된 운영 (모니터링 등)에서 이를 활용할 수 있습니다.

- **변경을 작게, 자주, 원복 가능하도록**

콤포넌트들이 정기적으로 업데이트될 수 있도록 업무를 설계하십시오. 작은 증분의 변경은 실패시에도 쉽게 원복할 수 있습니다.

운영 우수성 디자인 원칙 (2/2)

- 운영 절차를 수시로 개선

운영절차에서 개선기회를 찾으십시오. 정기적 게임데이(Game day)를 통해 효과적으로 운영 절차를 리뷰하고 검증할 수 있습니다.

- 실패 예측

프리모템(Pre-mortem)을 실행하여 잠재 장애요인을 찾으십시오. 실패 시나리오를 테스트하고 영향에 대한 이해를 검증하십시오. 정기적으로 게임데이를 실행하여 가상 이벤트에 대한 대응을 검증하십시오.

- 모든 운영실패를 통한 개선

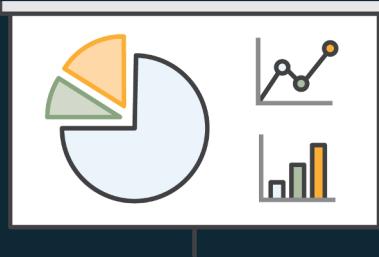
Lessons learned를 통해 개선사항을 찾고 팀과 전체 조직에 공유하십시오.

운영 우수성



준비 (Prepare)

- 비즈니스 목표와 정렬된 운영업무
- 모니터링기반 업무 표준화
- 이벤트/장애 대응 위한 테스트 설계
- 업무를 코드화



운영 (Operate)

- 성공 운영을 위한 측정 지표와 기준선
- 이벤트를 모니터링하고 프로세스 연결
- 적절한 권한과 에스컬레이션
- 근본원인 분석



진화 (Evolve)

- 정기적 개선기회 평가
- 피드백루프를 포함한 개선사이클
- Lessons Learned의 공유/ 상호분석

1. 준비 | 모범사례

운영업무 설계시 비즈니스 목표 반영

- 비즈니스 목표를 사업, 개발, 운영팀 모두와 공유
- 모니터링을 기반으로 하는 표준화된 운영업무 설계

운영업무와 변경이 제품으로 연결되도록 하는 운영 평가체계

- 표준을 준수하는지 체크, Runbook, Playbook 점검
- 실행에 앞서 이벤트와 장애에 대한 대처를 테스트
- 인공적인 장애나 게임데이와 같은 이벤트를 통해 운영 대응을 훈련

운영업무를 코드화하고 모니터링하기

- 코드기반으로 템플릿화된 일관성있는 환경구성 적용
- 운영시스템으로부터 다양한 로그를 수집할 수 있는 체계 구축

2. 운영 | 모범사례

성공적인 운영을 위한 측정지표 준비

- 배포, 인시던트 대응과 같은 운영업무내용이 포함된 지표
- 기준선 정의, 분석 및 개선사항 모색

운영 이벤트 관리

- 알려진 이벤트에 대한 Runbook, 알려지지 않은 이벤트에 대한 Playbook을 준비하고 비즈니스 영향에 따른 이벤트 대응 우선순위를 정의하기
- 이벤트 대응을 알람 및 실행 프로세스와 연결
- 이벤트 대응 인력이 적절한 권한을 가지고 있는지, 또는 에스컬레이션 프로세스가 있는지 확인하기
- 대시보드와 Notification을 통한 운영업무 현황 공유

문제에 대한 근본원인을 확인

- 근본원인 확인을 통해 이벤트 재발행을 감소시키고 다른 팀들과 공유하기

3. 진화 | 모범사례

지속적이고 점진적인 개선을 위한 업무 사이클 준비

- 운영업무와 운영절차에서 정기적으로 개선 기회를 평가하고 우선순위화하기
- 절차에 피드백 루프를 포함시키기

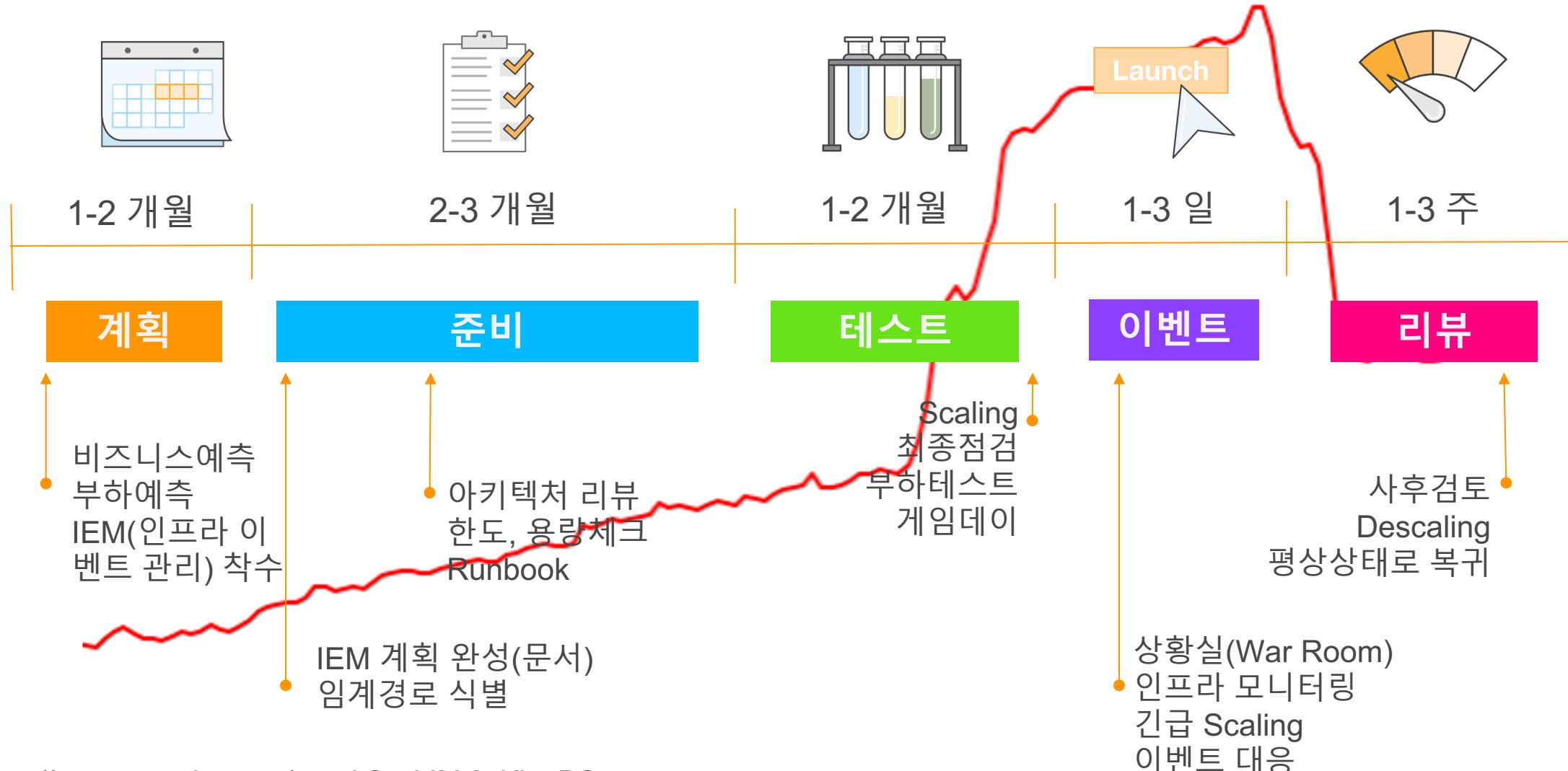
Lessons Learned를 다른 팀과 공유하기

- Lessons Learned의 트랜드를 분석 및 팀간 상호분석 실행
- 공유된 사례를 통해 변화를 실행으로 옮기기

배포활동의 결과 활용한 개선기회

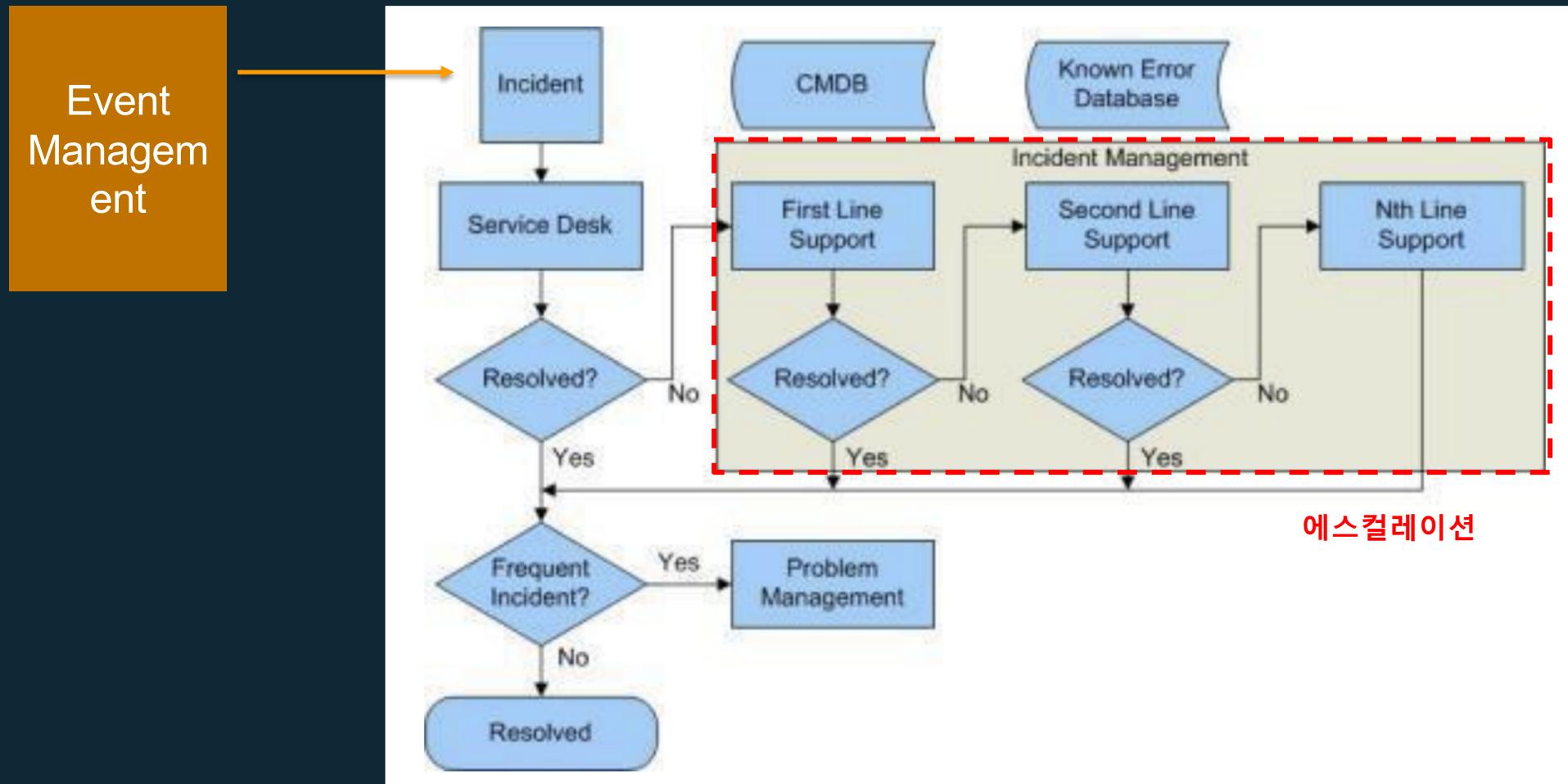
- 운영과 개발활동을 통합한 지표를 활용하고 사업과 고객성과에 대한 영향을 분석
- 다른 팀과 분석결과 교차확인

아마존.com 프라임 데이 운영사례



https://www.youtube.com/watch?v=bIMt0_KLmBQ

2. 운영 | ITIL 인시던트 관리



<https://d1.awsstatic.com/whitepapers/itil-event-management-in-the-cloud.pdf>

2. 운영 | ITIL 문제관리

Incident Management □ 복구

Problem Management □ 근본원인 제거 (재발 방지)

□ RCA(Root Cause Analysis) 원칙

- 정확한 문제 정의 (5 why)
- 정상상태부터 실패까지의 타임라인 추적 (관련된 원인요소 찾기)
- 표면적인 원인과 근본원인 구별
- 근본원인 제거와 다른 문제 예측에 활용

https://en.wikipedia.org/wiki/Root_cause_analysis

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.





Cloud Adopted Application

Cloud Adopted Application 1/2

- **자체 복구를 위한 디자인**

분산 시스템에서는 오류가 발생합니다.

오류가 발생하면 자체 복구되도록 응용 프로그램을 디자인하십시오.

- **모두 중복으로 구성**

단일 실패 지점을 피하도록 응용 프로그램에 중복성을 구축합니다.

- **조정 최소화**

확장성을 위해 응용 프로그램 서비스 간의 조정을 최소화합니다.

- **규모 확장을 위한 디자인**

수요에 따라 새 인스턴스를 추가하거나 제거하여 규모 확장이 가능하도록 응용 프로그램을 디자인합니다.

- **한도에 맞춘 분할**

분할을 사용하여 데이터베이스, 네트워크 및 계산 한도를 해결합니다.

Cloud Adopted Application 2/2

- **운영을 위한 디자인**

운영 팀에 필요한 도구가 포함되도록 응용 프로그램을 디자인합니다.

- **관리되는 서비스 사용**

가능하면 IaaS(Infrastructure as a Service)보다 PaaS(Platform as a Service)를 사용합니다.

- **작업에 가장 적합한 데이터 저장소 사용**

데이터에 가장 적합한 저장소 기술과 사용 방법을 선택합니다.

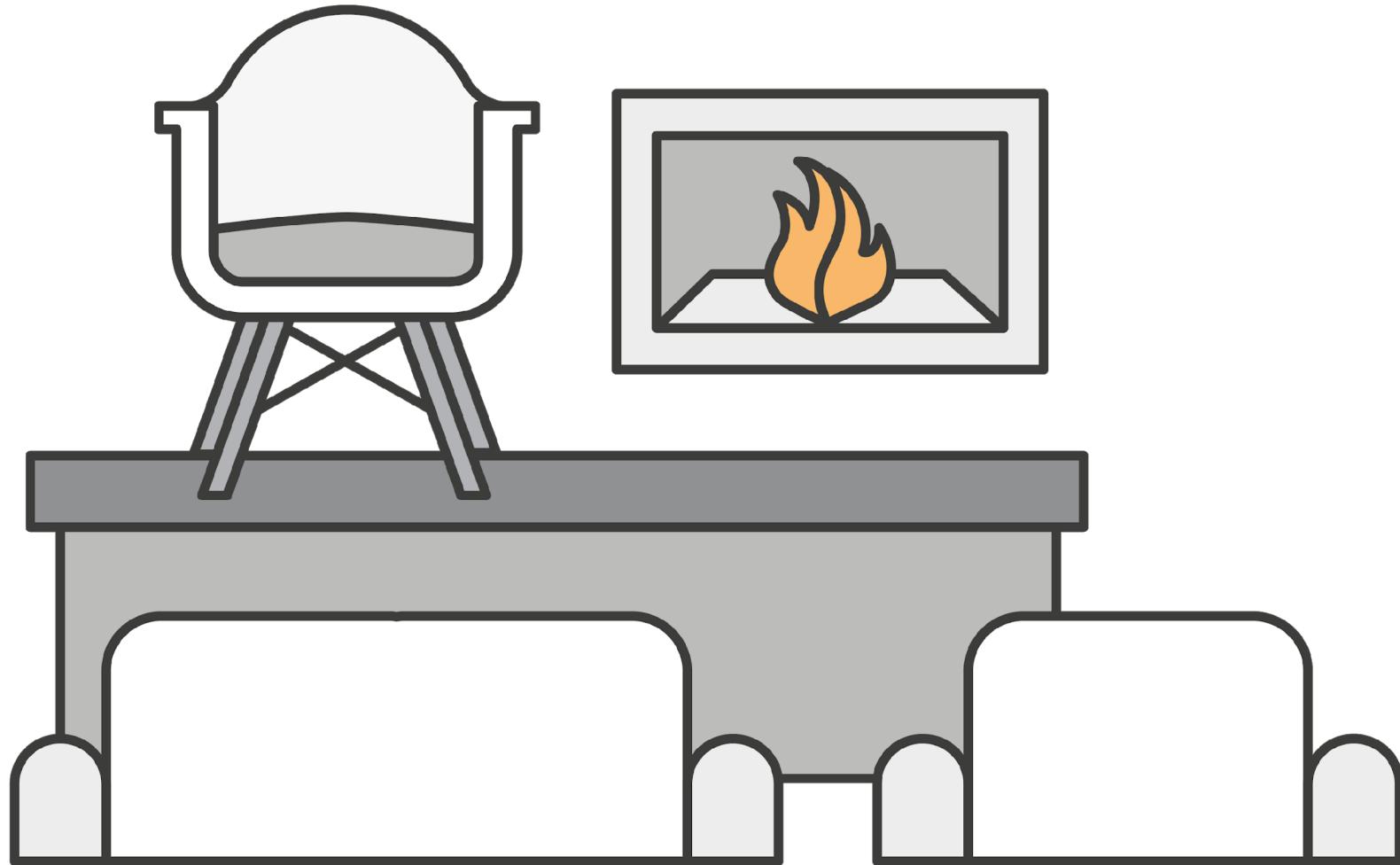
- **진화를 위한 디자인**

모든 성공적인 응용 프로그램은 시간에 따라 변화합니다. 혁신적인 디자인은 지속적인 혁신의 핵심입니다.

- **비즈니스 요구 사항에 맞게 구축**

모든 디자인 결정은 비즈니스 요구 사항에 맞춰 정당화되어야 합니다.

Q & A



Disclaimer

All information, guidance and materials (collectively, “Information”) provided to you in connection with the Program are for informational purposes only. You are solely responsible for making your own independent assessment of the Information and your use of AWS’s products or services. Neither this document nor any other Information provided to you creates any warranties (express or implied), representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. Neither this document nor any other Information provided to you are part of, nor do they modify, any agreements between you and AWS.