HANSUNG UNIVERSITY

클라우드컴퓨팅

AWS Service - Security

Sung-Dong Kim,
School of Computer Engineering,
Hansung University

What to study

- 공동 책임 모델 (Shared Responsibility Model: SRM)
- AWS CloudTrail
- Other Services

공동 책임 모델

공동 책임 모델

고객 책임

고객 데이터

플랫폼, 애플리케이션, 자격 증명 및 액세스 관리

운영 체제, 네트워크, 방화벽 구성

클라이언트 측 데이터 암호화 및 데이터 무결성 인증

서버 측 암호화 (파일 시스템 및/또는 데이터)

네트워크 트래픽 보호(암호화, 무결성, 자격 증명)

AWS의 책임

AWS 기초 서비스

컴퓨팅 | 스토리지 | 데이터베이스 | 네트워킹

AWS 글로벌 인프라

리전 | 가용 영역 | 엣지 로케이션

intel

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

공동 책임 모델

AWS의 책임: 클라우드의 보안

- 데이터 센터의 물리적 보안
- 하드웨어 및 소프트웨어 인프라: 스토리지 폐기, 호스트 OS 액세스 로깅 및 감사
- 네트워크 인프라: 침입 탐지
- 가상화 인프라: 인스턴스 격리

공동 책임 모델

고객의 책임: 클라우드에서의 보안

- EC2 instance 운영 체제: 패치 적용, 유지 관리 등
- 애플리케이션: 암호, 역할 기반 액세스 등
- 보안 그룹 구성
- OS 또는 호스트 기반 방화벽: 침입 탐지 또는 차단 시스템
- 네트워크 구성
- 계정 관리: 각 사용자에 대한 로그인 및 권한 설정

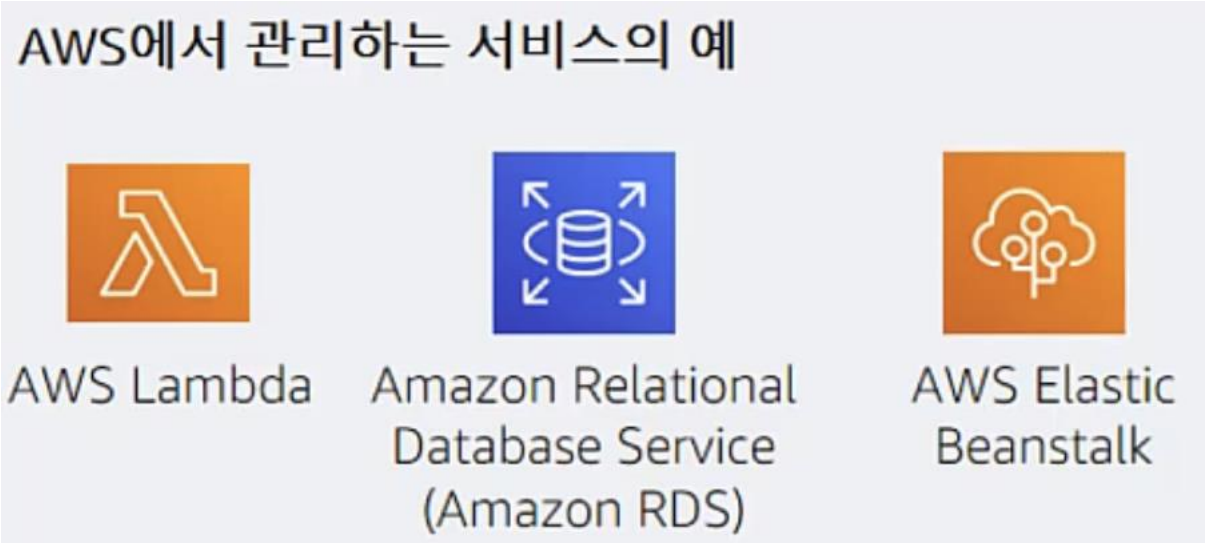
공동 책임 모델

서비스 특성 및 보안 책임

- IaaS
  - 고객은 네트워킹 및 스토리지 설정을 보다 유연하게 구성할 수 있음
  - 보안의 더 많은 측면을 고객이 관리해야 함
  - 액세스 제어를 고객이 구성

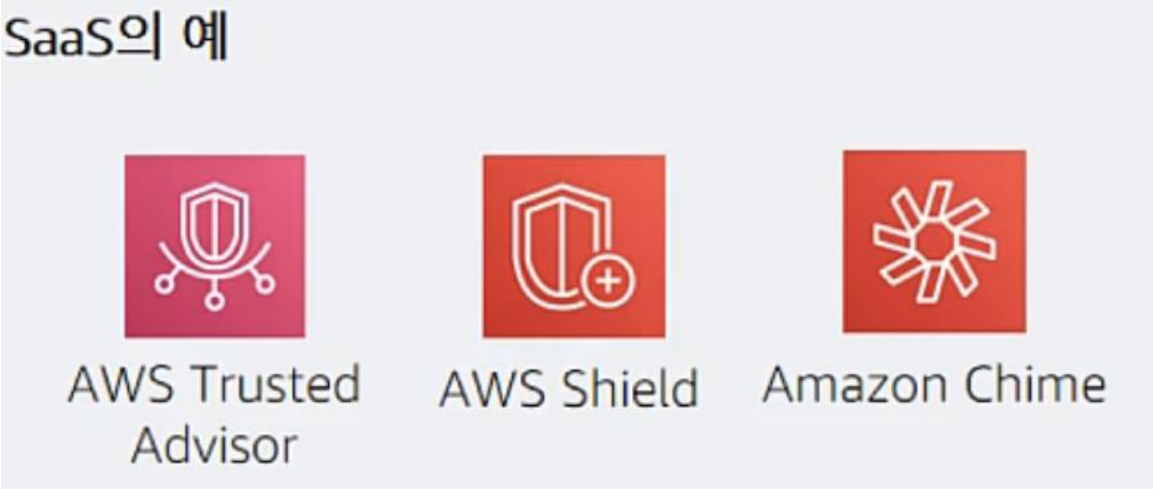
공동 책임 모델

- PaaS
  - 고객이 기본 인프라를 관리할 필요 없음
  - 운영체제, 데이터베이스 패치 적용, 방화벽 구성 및 재해 복구를 AWS가 처리
  - 고객은 코드 또는 데이터 관리에 집중



공동 책임 모델

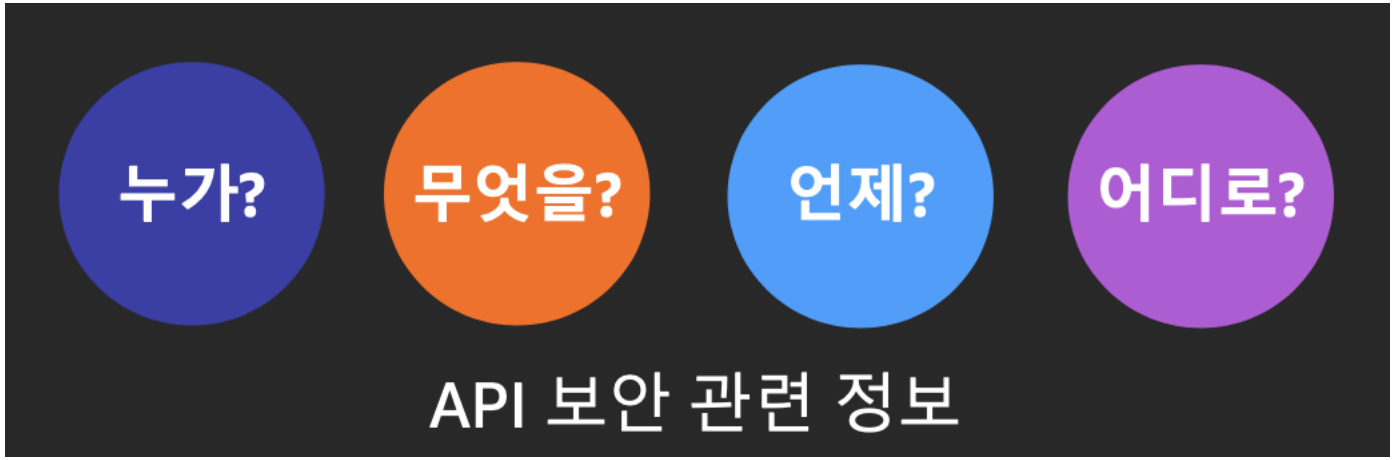
- SaaS
  - 소프트웨어가 중앙에서 호스팅 됨
  - 구독 모델 또는 종량 과금제로 라이선스가 부여됨
  - 일반적으로 웹 브라우저, 모바일 앱 또는 API를 통해 액세스 함
  - 고객은 서비스를 지원하는 인프라를 관리할 필요 없음



AWS CloudTrail

AWS CloudTrail

- AWS 계정의 사용자 활동 및 API 사용 추적
- 지속적으로 사용자 활동을 모니터링하고 API 호출을 기록
- 규정 준수 감사, 보안 분석, 문제 해결에 유용
- 로그 파일은 S3 버킷으로 전송됨



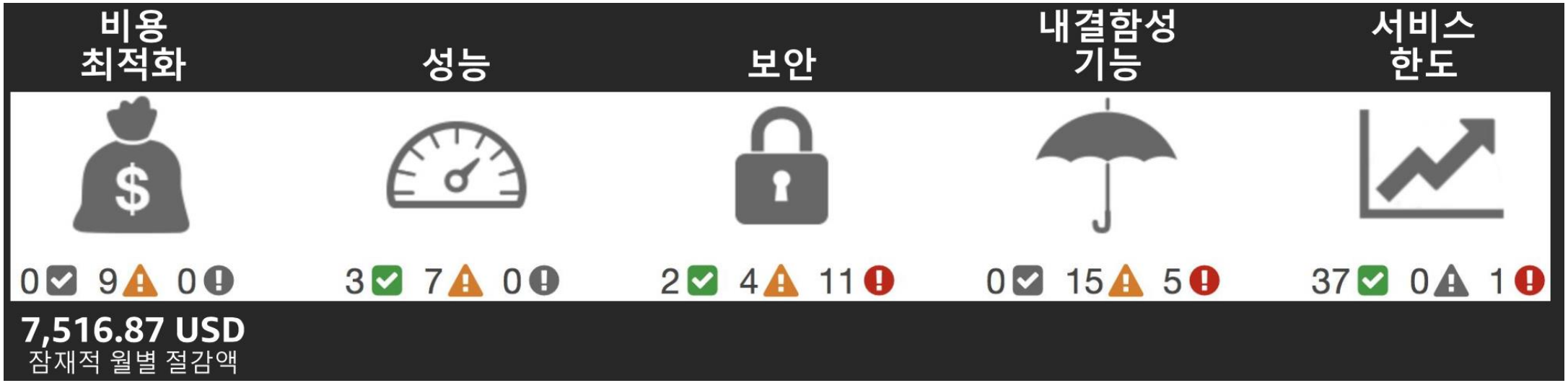
AWS CloudTrail

- 기본 AWS CloudTrail 이벤트 기록은 기본적으로 활성화 되어 있으며 무료임
  - 최근 90일 간의 계정 활동에 대한 모든 관리 이벤트 데이터 포함
- CloudTrail 액세스
  - AWS Management Console → CloudTrail 선택
  - [Event history (이벤트 기록)]
- 90일이 지난 로그 및 이벤트 알림 기능 활성화 → 추적 생성
  - CloudTrail 콘솔 추적 페이지에서 [Create trail(추적 생성)] 선택
  - 이름 지정, 모든 리전 적용, 로그 저장을 위한 새 S3 버킷 생성
  - S3 버킷에 대한 액세스 권한 구성

Other Services

AWS Trusted Advisor

- 비용 절감, 성능 개선, 보안 강화에 도움이 되는 지침을 제공하는 서비스



AWS Key Management Service (AWS KMS)

- 암호화 키를 생성하고 관리
- AWS 서비스 및 애플리케이션 전체의 암호화 사용 제어
- AWS CloudTrail과 통합할 경우 모든 키 사용 로깅
- FIPS (Federal Information Processing Standard) 140-2에서 검증한 HSM (하드웨어 보안 모듈)을 사용하여 키 보호

Amazon Cognito

- 웹 및 모바일 애플리케이션에 사용자 가입, 로그인 및 액세스 제어를 추가
- 수백만 명의 사용자까지 확장
- Facebook, Google, Amazon과 같은 소셜 자격 증명 공급자와 SAML (Security Assertion Markup Language) 2.0을 사용한 Microsoft Active Directory와 같은 엔터프라이즈 자격 증명 공급자를 사용한 로그인 지원

AWS Shield

- DDoS (분산 서비스 거부 공격) 방어를 위한 관리형
- AWS에서 실행되는 애플리케이션 보호
- AWS Shield Standard는 추가 비용 없이 활성화 됨
- AWS Shield Advanced는 선택형 유료 서비스
- 애플리케이션 다운타임과 지연시간을 최소화 할 수 있음

데이터 암호화

유휴 데이터 암호화

- 보안 키로 데이터 인코딩
- 보안 키가 있는 사용자만 데이터 디코딩
- AWS KMS에서 보안 키 관리

전송 데이터 암호화

- TLS (전송 계층 보안)
- AWS Certificate Manager: TLS 인증서 배포, 관리 및 갱신 방법 제공
- 보안 HTTP (=HTTPS)은 보안 터널 생성

HANSUNG UNIVERSITY

클라우드컴퓨팅

AWS Service - Cost-Billing

Sung-Dong Kim,
School of Computer Engineering,
Hansung University

What to study

- AWS 요금 모델
- 총 소유 비용 (Total Cost of Ownership: TCO)
- AWS Billing Services

AWS 요금 모델

AWS 요금 모델

AWS의 3가지 기본적인 비용 동인

컴퓨팅

- 시간/초당 청구*
- 인스턴스 유형에 따라 다름

*Linux 전용

스토리지

- 일반적으로 GB당 청구

데이터 전송

- 아웃바운드 요금은 집계하여 청구
- 인바운드는 무료 (일부 예외)
- 일반적으로 GB당 청구

AWS 요금 모델

AWS의 요금 지불

사용한 만큼만 지불

예약 시 더 적은
비용 지불

사용량과 AWS
규모가 증가하면 비용
절감액도 증가

총 소유 비용

총 소유 비용

- TCO (Total Cost of Ownership: 총 소유 비용)는 시스템의 직접 및 간접 비용을 확인하는 데 도움이 되는 재정적 추정
- Why?
  - on-premise와 AWS에서 전체 인프라 환경 또는 특정 워크로드를 실행할 때의 비용을 비교하기 위해
  - 클라우드로 이전하기 위한 예산을 책정하고 비즈니스 사례를 만들기 위해

총 소유 비용

고려 사항

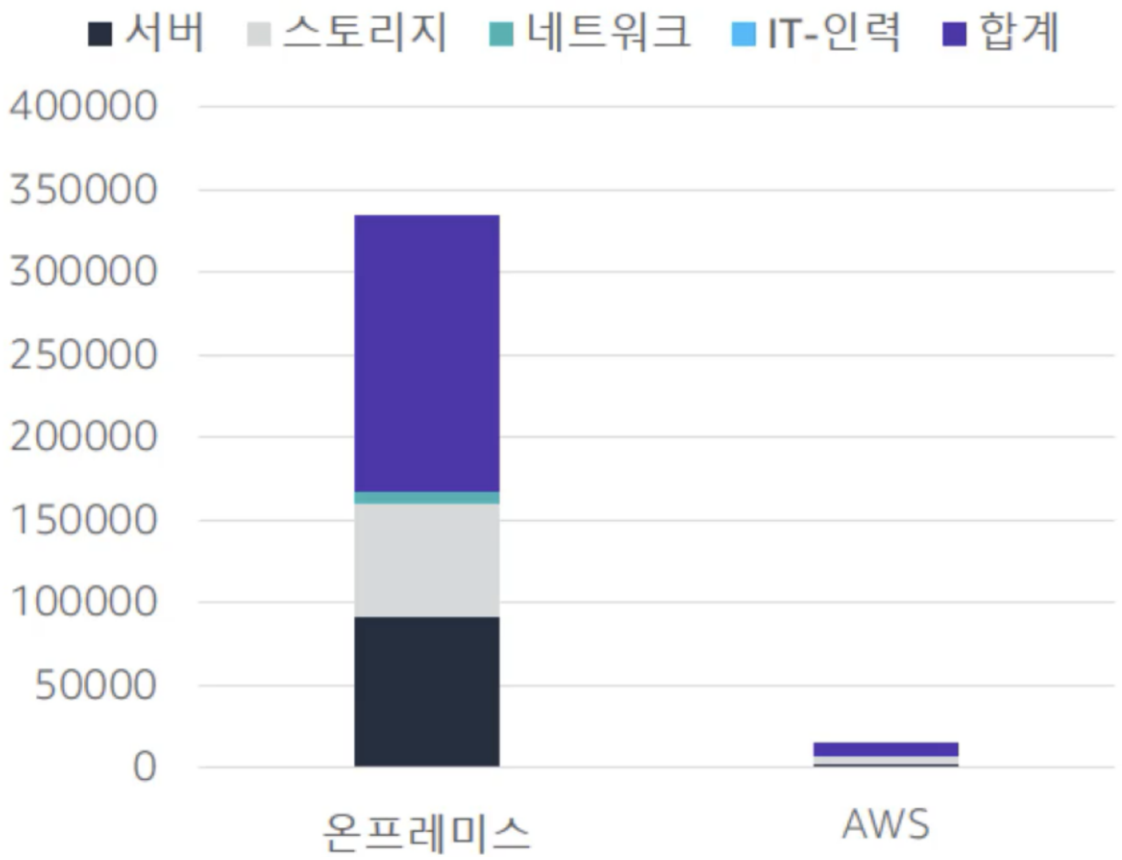
1	서버 비용	하드웨어: 서버, 랙 섀시의 PDU(배전 장치), TOR(Top-of-Rack) 스위치(및 유지 관리)	소프트웨어: OS(운영 체제), 가상화 라이선스(및 유지 관리)	시설 비용		
				공간	전력	냉각
2	스토리지 비용	하드웨어: 스토리지 디스크, SAN(Storage Area Network) 또는 FC(파이버 채널) 스위치	스토리지 관리 비용	시설 비용		
				공간	전력	냉각
3	네트워크 비용	네트워크 하드웨어: LAN(Local Area Network) 스위치, 로드 밸런서 대역폭 비용	네트워크 관리 비용	시설 비용		
				공간	전력	냉각
4	IT 인건비	서버 관리 비용				

총 소유 비용

온프레미스와의 비교

- AWS로 이전 시, 연간 최대 96% 비용 절감
- 3년 총 절감액 = 159,913 USD

3-Year Total Cost of Ownership		
	On-Premises	AWS
Server	$91,922	$2,547
Storage	$67,840	$4,963
Network	$7,660	$-------
IT – Labor	$ -------- --	$-------
Total	$167, 422	$7,509



총 소유 비용

월 사용량 계산기

- 월별 비용 추정
- 월별 비용 절감 기회 파악
- 템플릿을 사용하여 서비스 및 배포 모델 비교
- https://calculator.s3.amazonaws.com/index.html

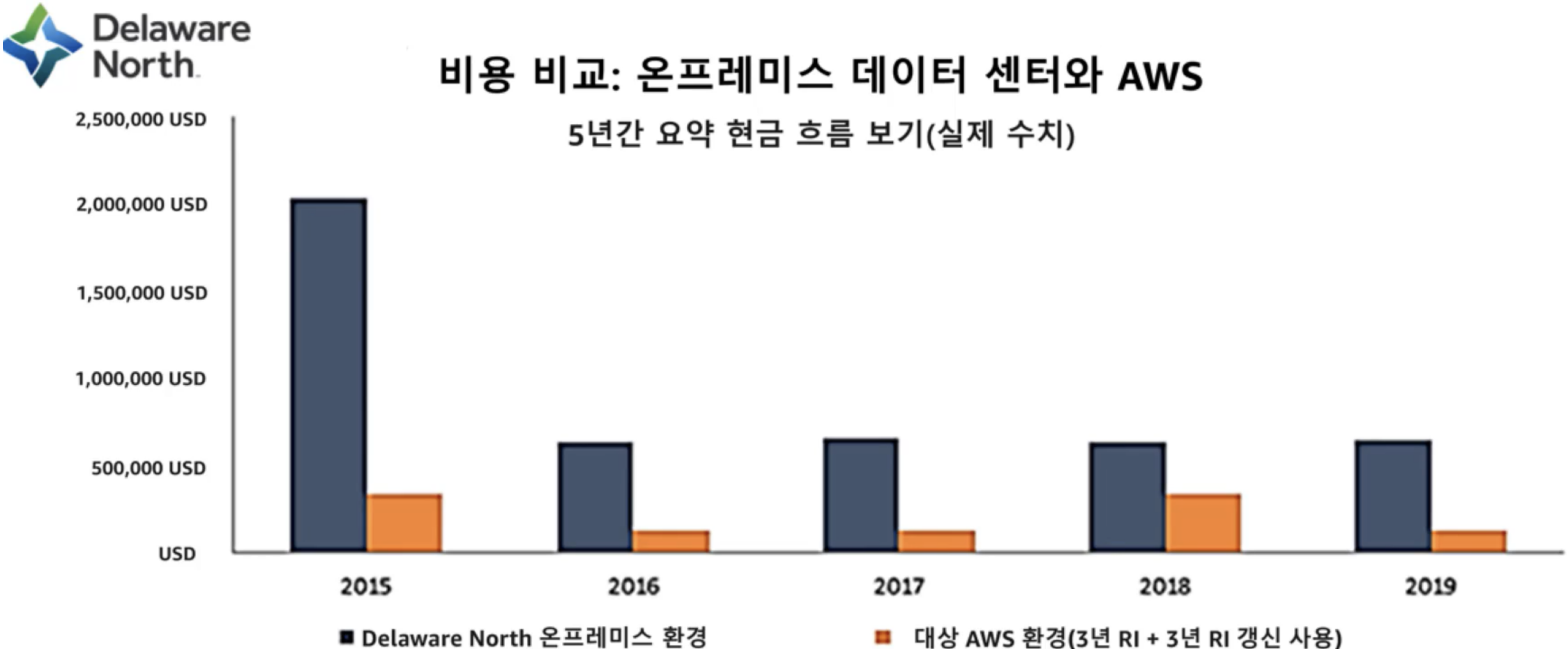
총 소유 비용

HANSUNG UNIVERSITY

Delaware North

- 배경
  - 200개가 넘는 지역에서 성장하는 글로벌 기업
  - 5억 명의 고객, 연간 매출 3억 USD
- 문제점
  - 새로운 솔루션을 신속하게 배포하기 위한 요구 사항 충족
  - 오래된 장비를 지속적으로 업그레이드
- 기준
  - 모든 워크로드를 처리하기 위한 광범위한 솔루션 보유
  - 비용 절감 및 효율성 개선을 위해 프로세스를 수정할 수 있는 기능
  - ROI (투자 수익) 달성
- 솔루션
  - 온프레미스 데이터 센테에서 AWS로 이동
    - 205개의 서버(90%)제거
    - 거의 모든 애플리케이션을 AWS로 이전
  - 3년 Amazon EC2 예약 인스턴스 사용

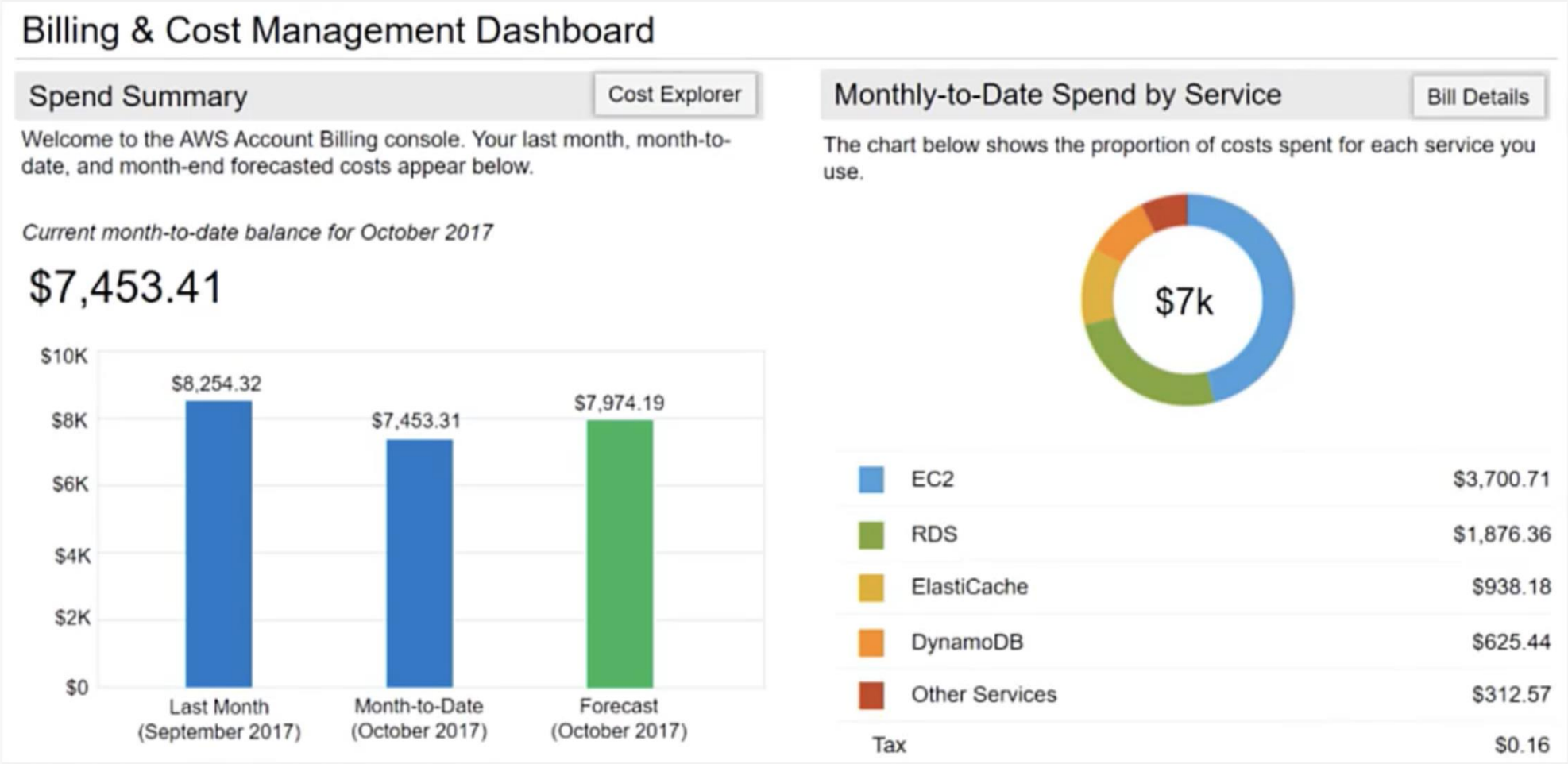
총 소유 비용


Delaware North
비용 비교: 온프레미스 데이터 센터와 AWS
5년간 요약 현금 흐름 보기(실제 수치)
2,500,000 USD
2,000,000 USD
1,500,000 USD
1,000,000 USD
500,000 USD
USD
2015
2016
2017
2018
2019
Delaware North 온프레미스 환경
대상 AWS 환경(3년 RI + 3년 RI 갱신 사용)

AWS Billing Services

AWS Billing

AWS 결제 대시보드



AWS Billing

도구

AWS 예산

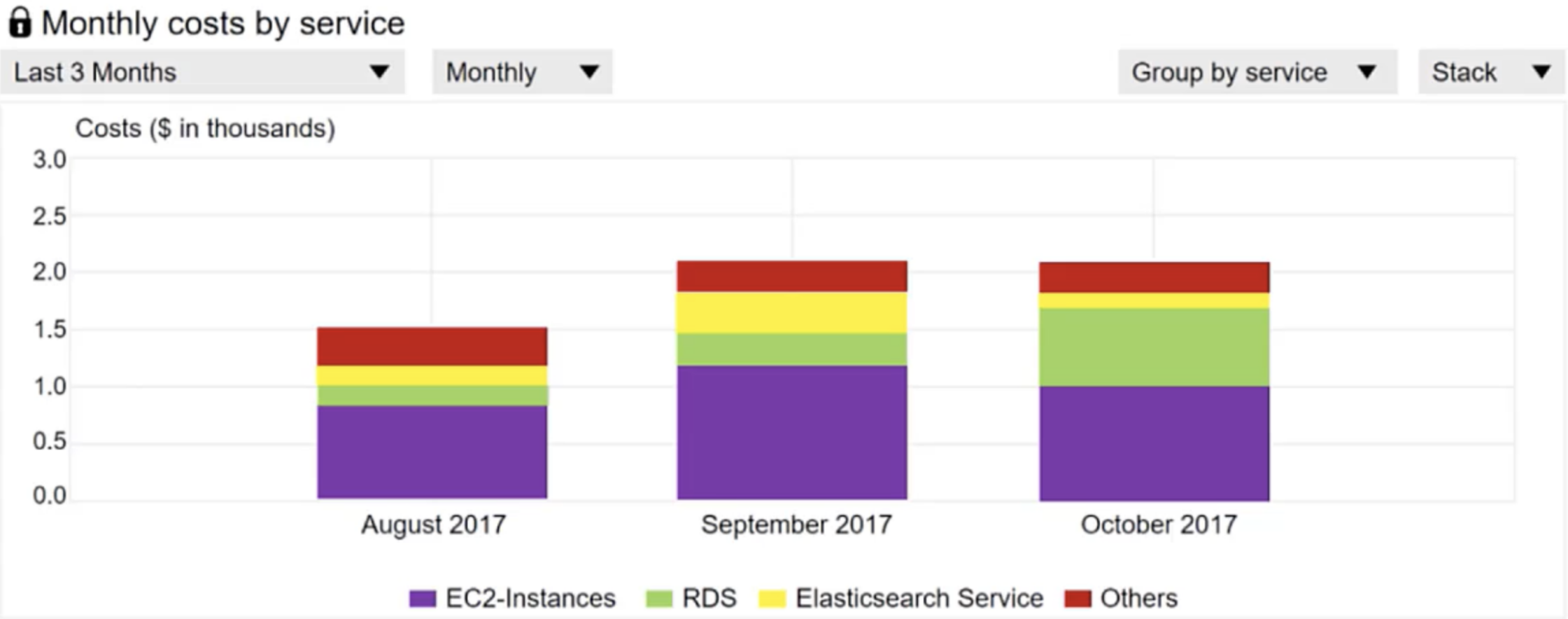
AWS 비용 및 사용 보고서

AWS Cost Explorer

AWS Billing

HANSUNG UNIVERSITY

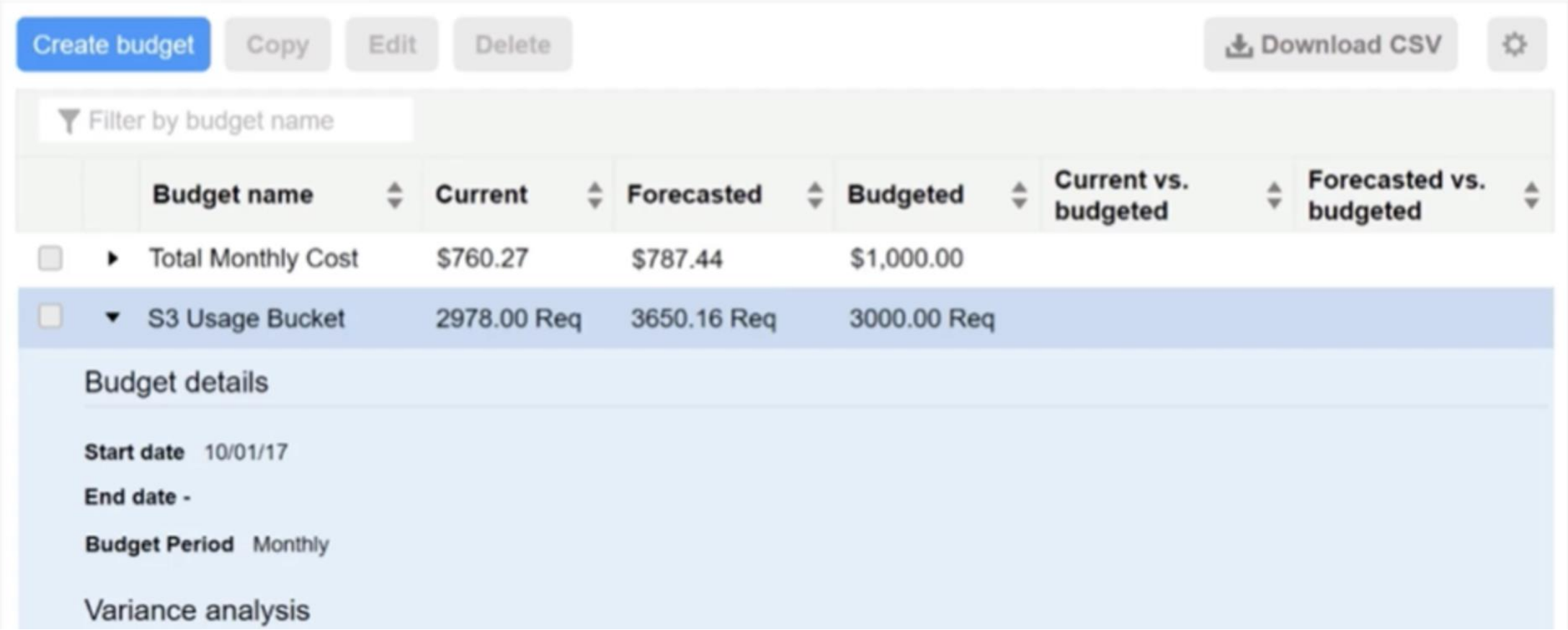
Cost Explorer



AWS Billing

HANSUNG UNIVERSITY

예산 (budgets)



AWS Billing

HANSUNG UNIVERSITY

비용 및 사용 보고서

Product Code	Usage Type	Operation	Availability Zone	Usage Amount	Currency Code	Line Item Description
Amazon S3	Requests – Tier 1	ListAllMyBuckets		2	USD	$0.00 per request – PUT, COPY, POST, LIST under the global free tier
Amazon EC2	USW2-Boxusage:t2.micro	Runinstnaces:0002	us-west-2a	1	USD	$0.00 per Windows t2.micro instance-hour under monthly free tier
Amazon S3	Requests – Tier 1	ListAllMyBuckets		2	USD	$0.00 per request – PUT, COPY, POST, LIST under the global free tier
Amazon EC2	USW2-Boxusage:t2.micro	Runinstnaces:0002	us-west-2a	1	USD	$0.00 per Windows t2.micro instance-hour under monthly free tier
Amazon S3	Requests – Tier 1	ListAllMyBuckets		2	USD	$0.00 per request – PUT, COPY, POST, LIST under the global free tier
Amazon S3	Requests – Tier 1	ListAllMyBuckets		2	USD	$0.00 per request – PUT, COPY, POST, LIST under the global free tier

AWS 이론 4
- Load Balancing

Sung-Dong Kim,

School of Computer Engineering,

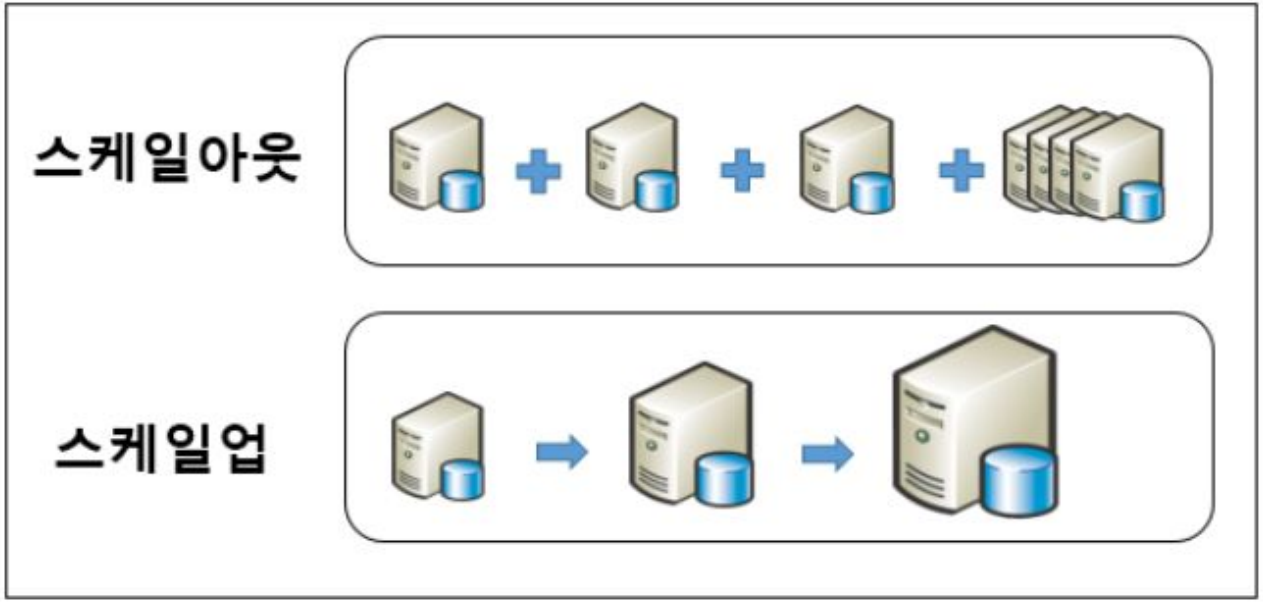
Hansung University

What

- 네트워크 기술의 일종
- 네트워크 트래픽을 하나 이상의 서버나 장비로 분산하는 기술
- 로드 밸런서(load balancer): 로드 밸런싱을 수행하는 SW, HW

Scale up/out (1)

- 웹 트래픽 증가에 대한 처리 방식



https://m.blog.naver.com/islove8587/220548900044

Scale up/out (2)

- Scale up
  - 더 높은 성능을 가진 서버로 upgrade
  - 비용, 가용성 문제
- Scale out
  - 저렴한 여러 노드로 cluster 구성
  - 가용성이 높음
  - 로드 밸런싱

로드 밸런싱 방식

- round robin: server로의 session 연결을 순차적으로 수행
- hash: client와 server 간에 연결된 session을 계속 유지
- least connection: 가장 작은 session을 보유한 server로 session을 연결
- response time
  - resource와 connection의 차이가 있는 환경
  - 빠른 응답시간을 제공하는 서버로 session 연결

AWS ELB

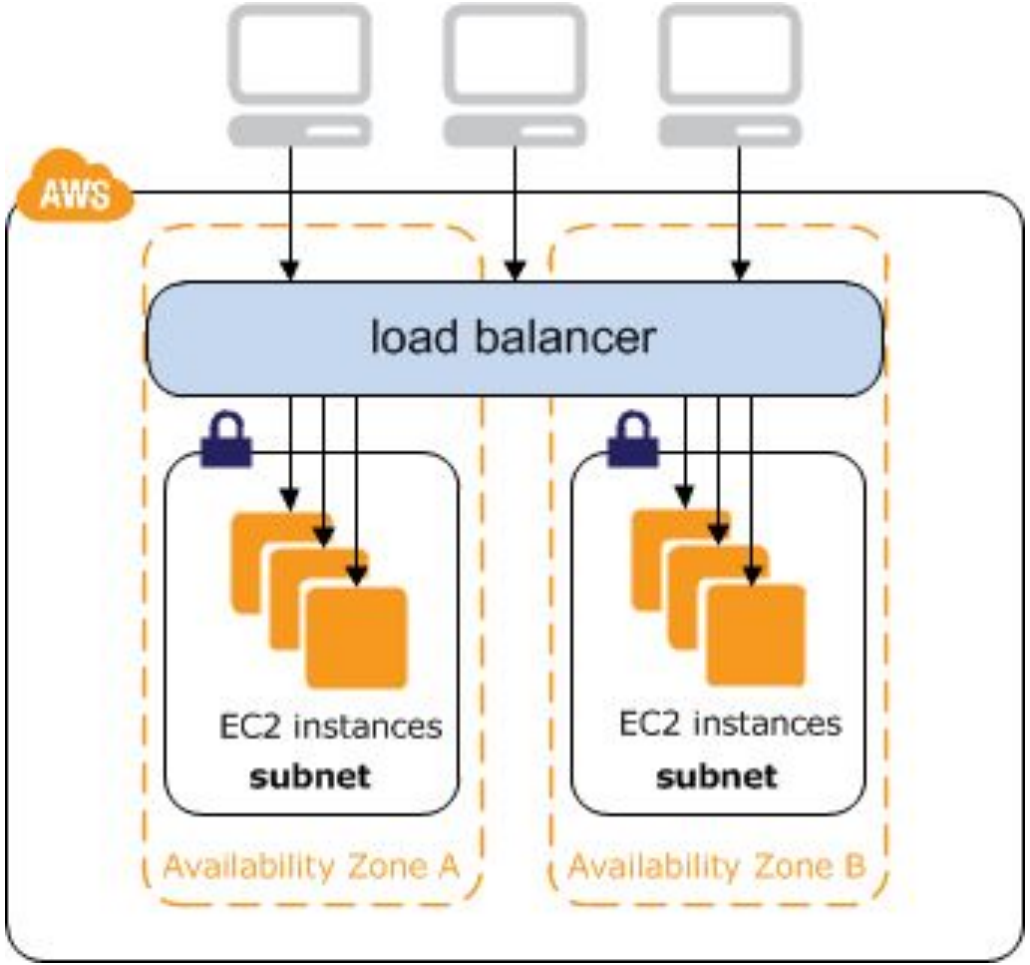
- EC2 instance, container, IP address 같은 여러 대상에 대해 수신 application 또는 network traffic을 여러 AZ에 배포
- workload를 다수의 computing resource로 분산
- application의 가용성, 내결함성 향상

특징 (1)

- health check (상태 확인)
  - ELB와 연결된 instance의 연결 상태를 수시로 체크
  - HTTP, HTTPS: 웹 페이지 접속 시도에 대한 응답 코드(200) 정상 반환 여부 확인
- sticky session
  - client가 처음 접속한 서버로 계속 연결시켜줌
  - 처음 연결된 client에 별도의 HTTP 기반의 쿠키 값 생성, 이용

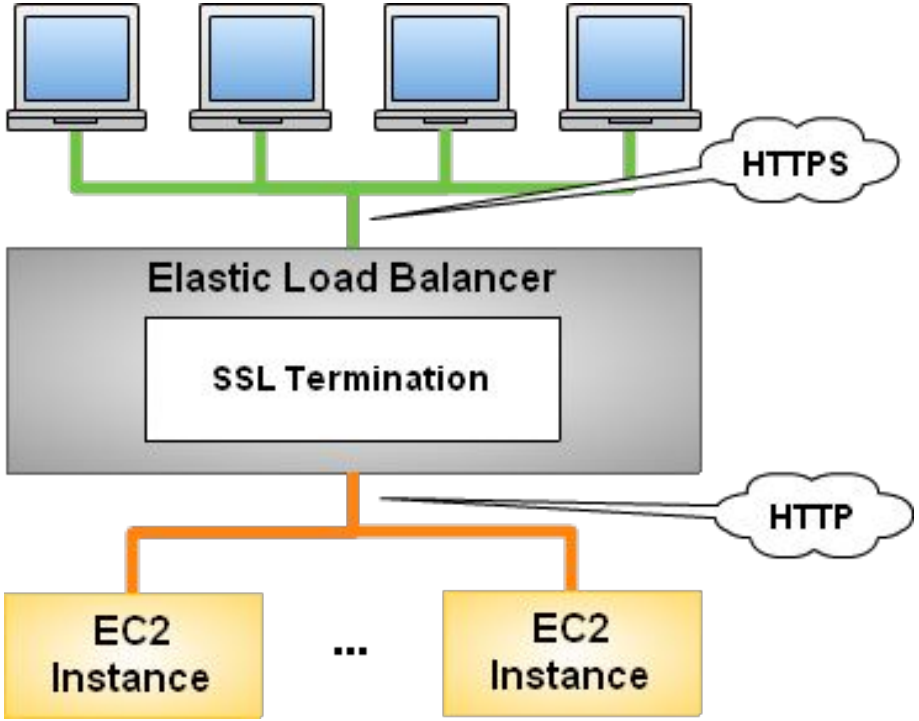
특징 (2)

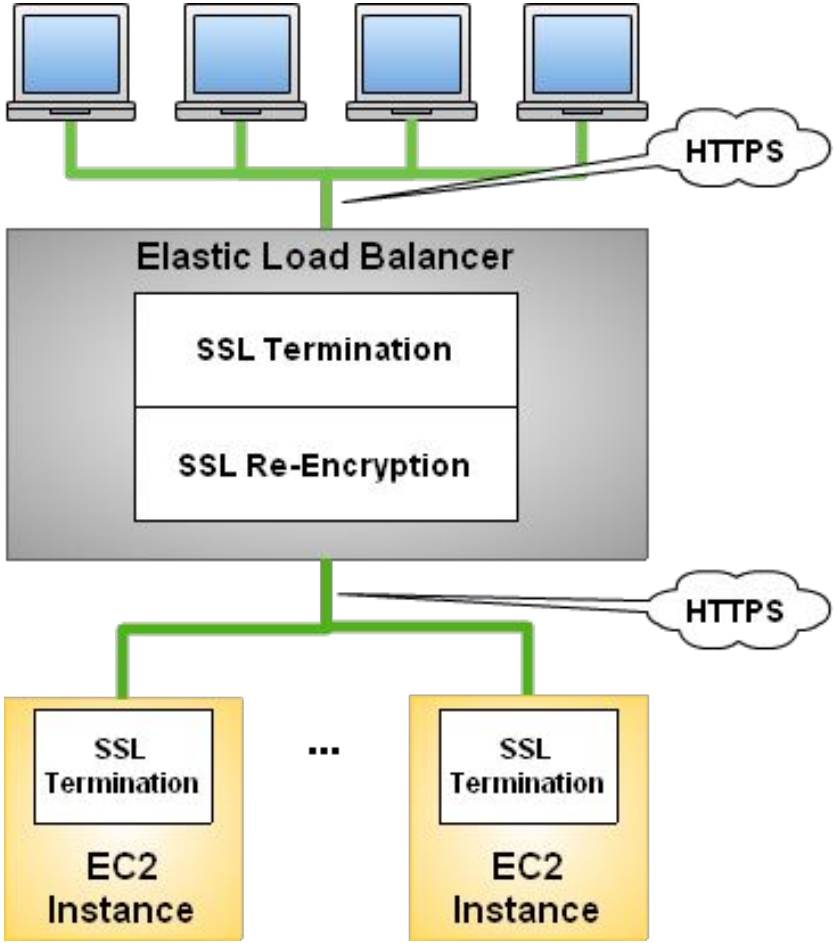
- 고가용성 구성
  - Route53 등의 AWS의 다른 서비스와의 연계를 통해 가용성 서비스 제공


AWS
load balancer
EC2 instances
subnet
EC2 instances
subnet
Availability Zone A
Availability Zone B

특징 (3)

- SSL termination 및 보안 기능
  - HTTPS와 같은 암호화 통신: 웹 서버에 별도의 공인인증서 필요
  - 개별 EC2 instace에서 SSL 암호화 및 복호화를 직접 처리
  - ELB의 SSL termination: 개별 instance에 SSL 인증서를 직접 설치할 필요 없음


HTTPS
Elastic Load Balancer
SSL Termination
HTTP
EC2 Instance
...
EC2 Instance


HTTPS
Elastic Load Balancer
SSL Termination
SSL Re-Encryption
HTTPS
SSL Termination
EC2 Instance
...
SSL Termination
EC2 Instance

종류

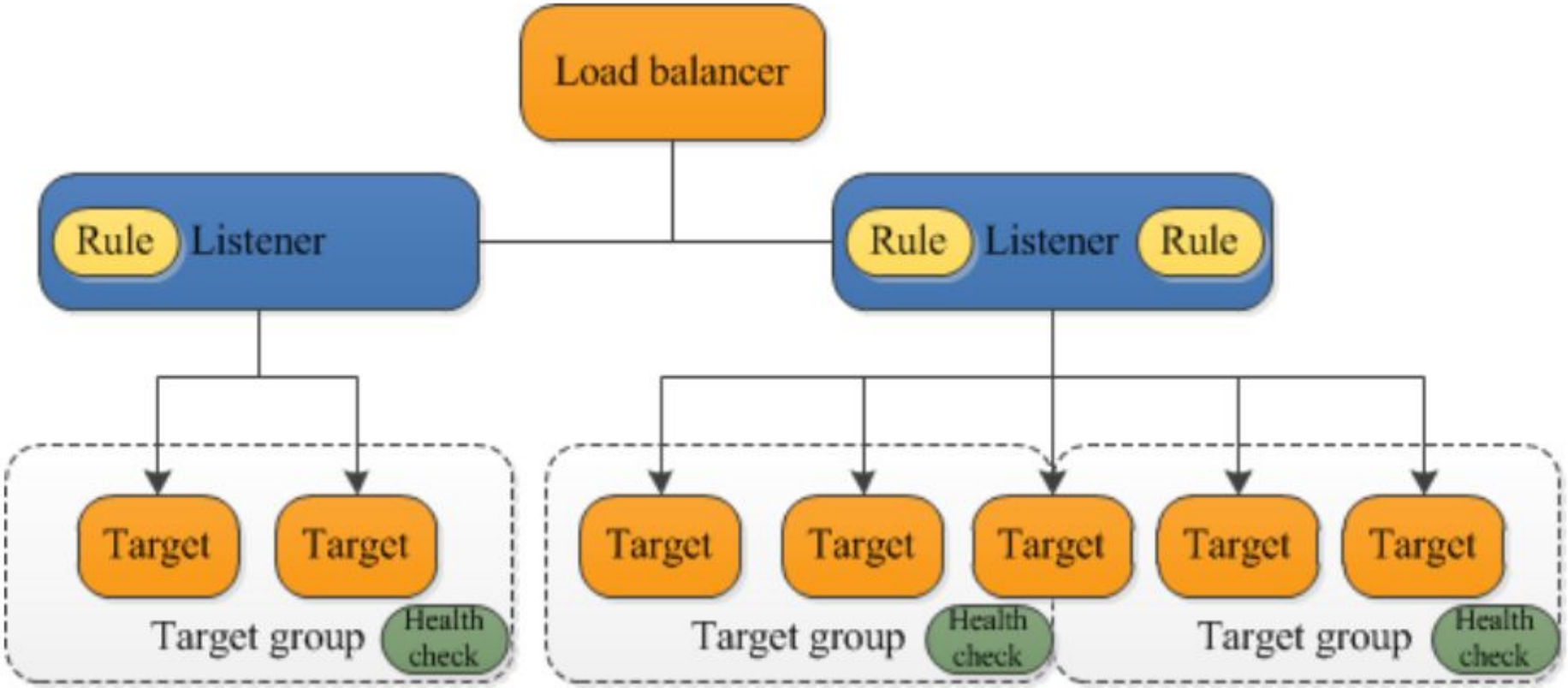
- Application Load Balancer
- Network Load Balancer
- Classic Load Balancer

Classic/Network Load Balancer

- classic load balancer
  - EC2-classic 네트워크 내에 구축된 application 대상
  - OSI 4-layer (transport layer), 3-layer (network layer)에서 작동
- network load balancer
  - OSI 4-layer (transport layer)에서 작동, TCP 트래픽 로드 밸런싱
  - 짧은 지연 시간

Application Load Balancer (1)

- load balancer: single point of contact for clients
- listener
  - clinet의 접속 요청 검사 → target group으로 보냄
  - rule을 정의: target group, condition, prority
- target group: 등록된 target으로 요청을 보냄 (protocol, port 이용)


Load balancer
Rule
Listener
Rule
Listener
Rule
Target
Target
Target group
Health check
Target
Target
Target
Target group
Health check
Target
Target
Target group
Health check

Application Load Balancer (2)

- OSI layer의 7번째 application layer에서 작동: HTTP/HTTPS
- request → listner rule 결정 → target group에서 target 결정

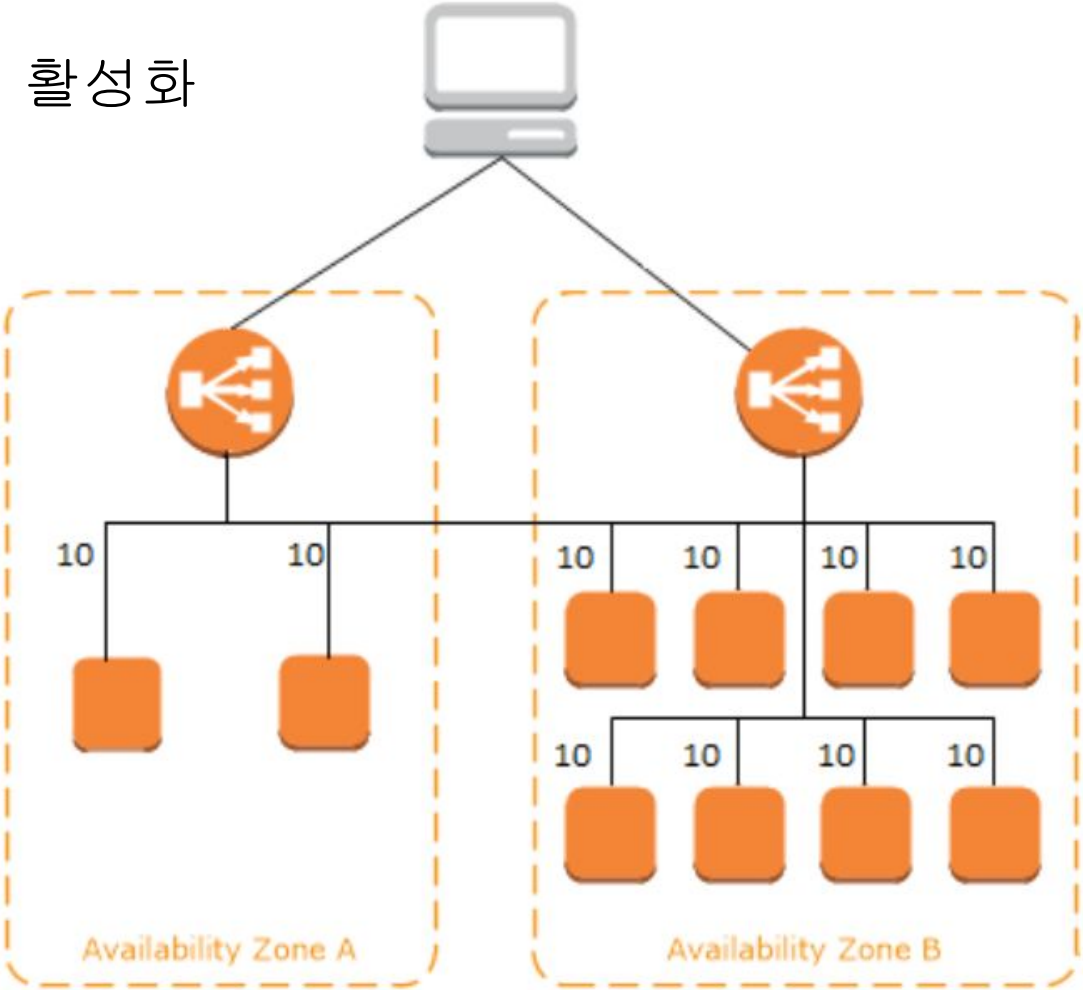
작동 방식

- client에서의 traffic을 받아, target group의 instance에 요청을 라우팅
- target group instance의 상태 모니터링
- listner
  - incomming traffic을 허용
  - protocol + port number

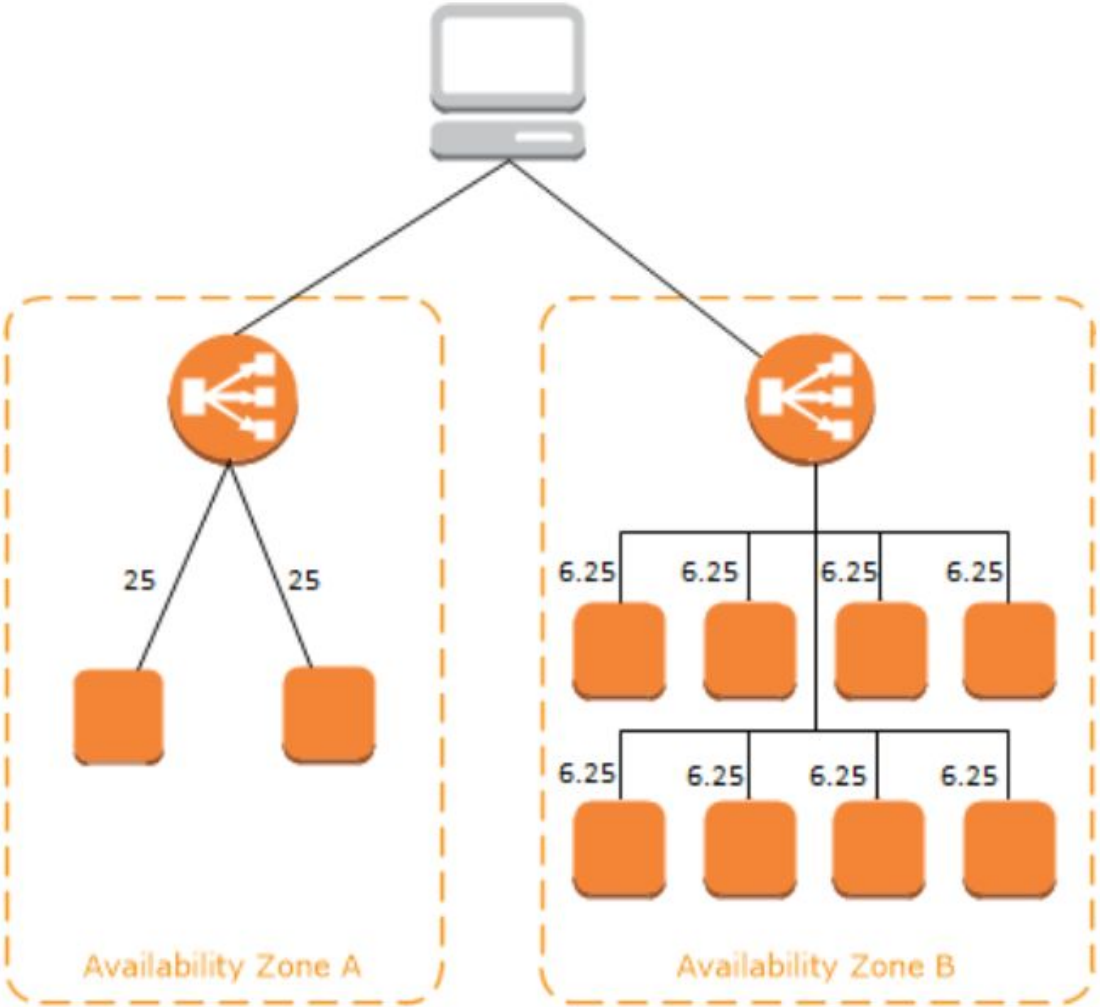
가용 영역 및 로드 밸런서 노드

- 여러 개의 가용 영역 활성화: 각 AZ에 하나 이상의 대상 등록
- 교차 영역 로드 밸런싱
  - 모든 AZ에 등록된 대상으로 트래픽 분산
  - Application load balancer에서는 항상 활성화 상태
  - Network load balancer에서는 기본적으로 비활성화 상태

교차 영역 로드 밸런싱 활성화

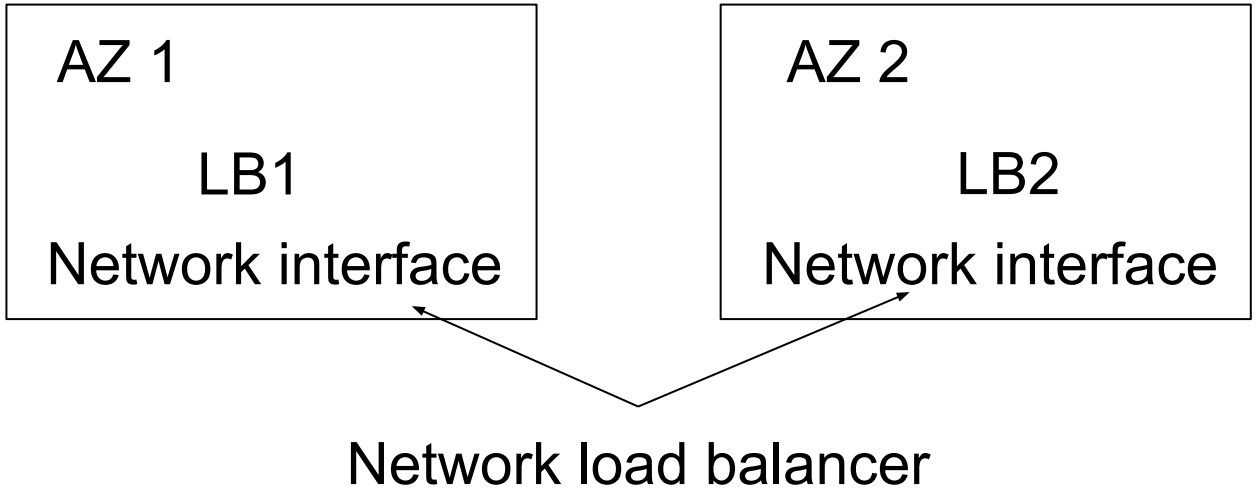


교차 영역 로드 밸런싱 비활성화



라우팅 요청 (1)

- client → DNS server → domain 이름 해석 → LB에 요청을 보냄: LB ← static IP from NI



라우팅 요청 (2)

- client의 요청을 받은 LB는 상태가 양호한 대상을 선택 → private IP를 사용하여 요청 전송
- 라우팅 알고리즘
  - priority에 따라 listner의 규칙 평가 후 규칙 결정
  - round-robin algorithm으로 대상 그룹에서 대상 선택

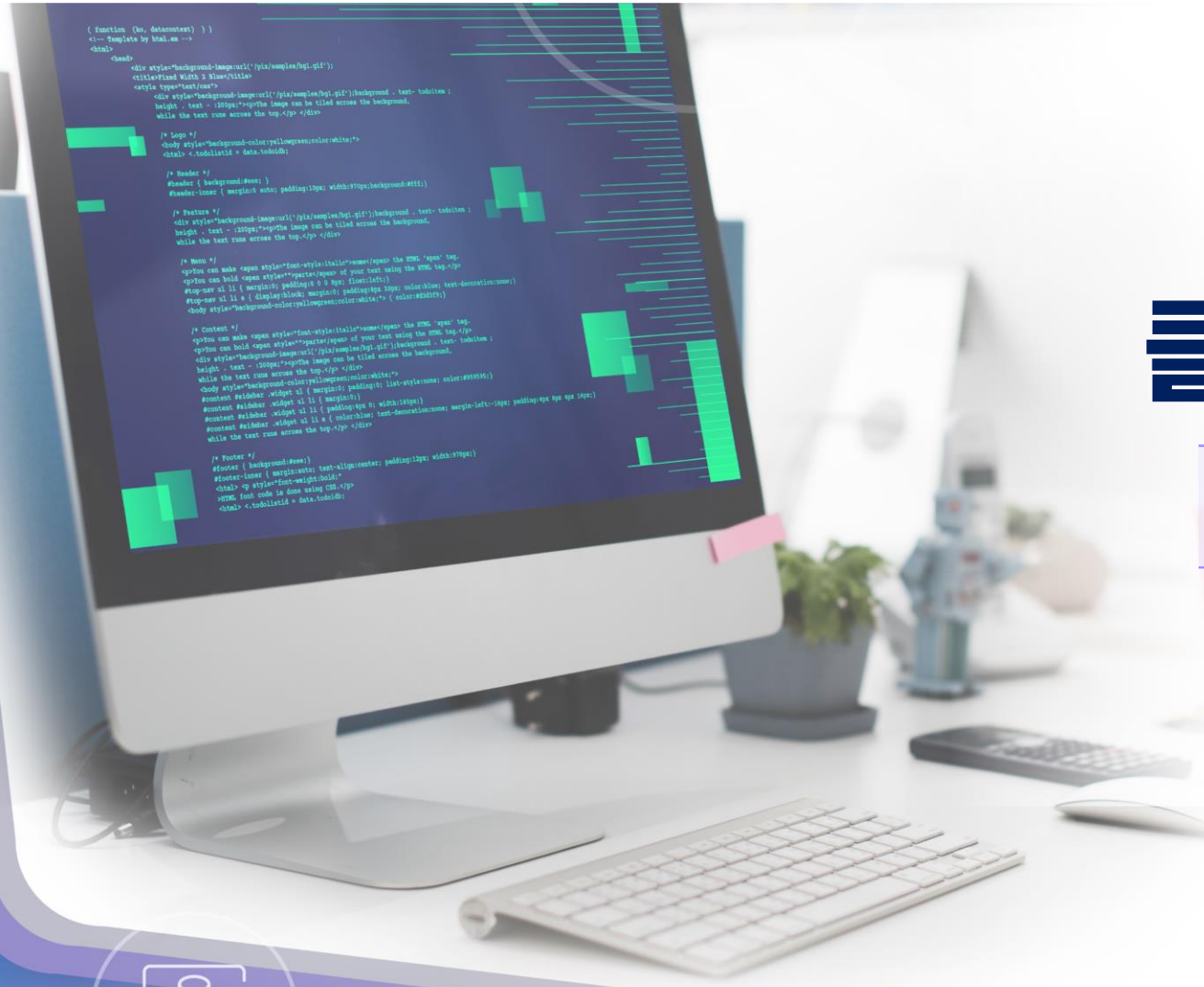
로드 밸런서 체계 (1)

- 내부 로드 밸런서 vs 인터넷 경계 로드 밸런서
  - private IP를 사용하여 요청을 라우팅
  - public IP가 없는 target instance도 요청을 수신할 수 있음

로드 밸런서 체계 (2)

- 인터넷 경계 로드 밸런서
  - public IP, private IP
  - VPC 내부 및 internet을 통해 client의 요청을 라우팅
- 내부 로드 밸런서
  - private IP
  - VPC 내부만 접속

HANSUNG UNIVERSITY



클라우드컴퓨팅

AWS Practice - Static Website Hosting on S3

Sung-Dong Kim,
School of Computer Engineering,
Hansung University

What to study

- Introduction
- Steps

Introduction

Introduciton

정적 웹 사이트

- HTML, Javascript, 이미지, 동영상 및 기타 파일을 웹 사이트 방문자에게 제공
- 애플리케이션 코드를 포함하지 않음

Introduciton

수행할 수 있는 작업

- Amazon S3를 사용하여 static website 배포
  - S3: website용 문서, 정적 콘텐츠용 스토리지 제공
- Amazon Route 53: domai과 website 연결
- Amazon CloudFront
  - 웹 사이트가 신속하게 로드될 수 있도록 지원
  - 사용자와 가까운 위치에서 콘텐츠 호스팅: CDN 생성

Steps

버킷 생성과 파일 업로드

- Management Console → S3
- 버킷 생성: ksdwebsite
- 파일 업로드
  - index.html, error.html
  - balloon.png
  - others
  - file url을 이용한 접속 확인

Web Server 기능 설정

HANSUNG UNIVERSITY

- 버킷 → 속성 → 정적 웹 사이트 호스팅 → 활성화
  - 호스팅 유형: 정적 웹 사이트 호스팅
  - 인덱스 문서: index.html
  - 오류 문서: error.html

정적 웹 사이트 호스팅

이 버킷을 사용하여 웹 사이트를 호스팅하거나 요청을 리디렉션합니다. 자세히 알아보기

편집

정적 웹 사이트 호스팅

비활성됨

Web Server 기능 설정

웹사이트 접근을 위한 권한 (permission) 설정

- S3 public access setting 변경: 버킷 → 권한

퍼블릭 액세스 차단(버킷 설정)

퍼블릭 액세스는 ACL(액세스 제어 목록), 버킷 정책, 액세스 지점 정책 또는 모두를 통해 버킷 및 객체에 부여됩니다. 모든 S3 버킷 및 객체에 대한 퍼블릭 액세스가 차단되었는지 확인하려면 [모든 퍼블릭 액세스 차단]을 활성화합니다. 이 설정은 이 버킷 및 해당 액세스 지점에만 적용됩니다. AWS에서는 [모든 퍼블릭 액세스 차단]을 활성화하도록 권장하지만, 이 설정을 적용하기 전에 퍼블릭 액세스가 없어도 애플리케이션이 올바르게 작동하는지 확인합니다. 버킷 또는 내부 객체에 어느 정도 수준의 퍼블릭 액세스가 필요한 경우 특정 스토리지 사용 사례에 맞게 아래 개별 설정을 사용자 지정할 수 있습니다. 자세히 알아보기

편집

모든 퍼블릭 액세스 차단

활성화

▶ 이 버킷의 개별 퍼블릭 액세스 차단 설정

모든 퍼블릭 액세스 차단

비활성

Web Server 기능 설정

HANSUNG UNIVERSITY

웹사이트 접근을 위한 권한 (permission) 설정

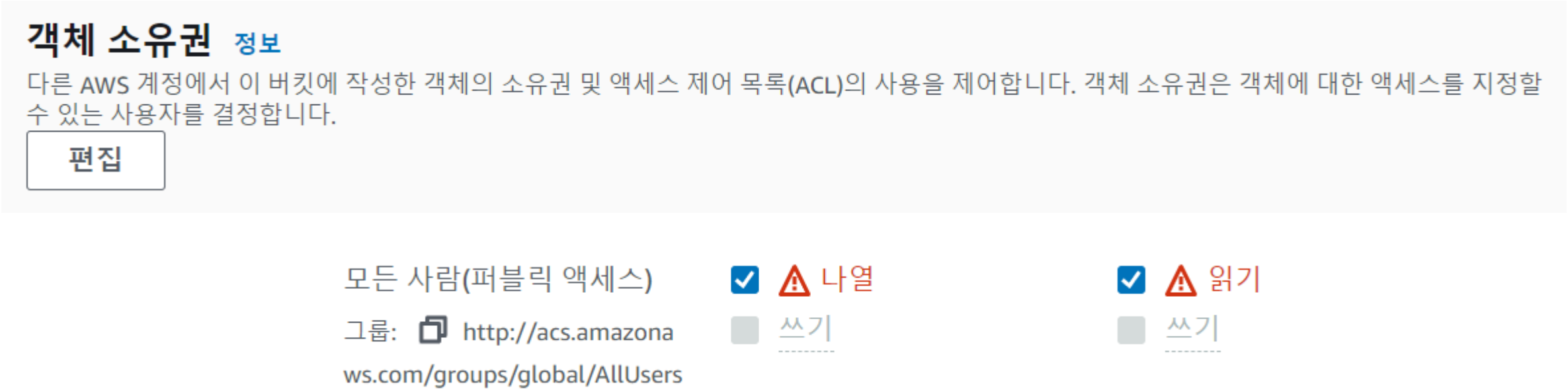
- 버킷 정책 (bucket policy) 추가

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::ksdwebsite/*"
            ]
        }
    ]
}
```

Web Server 기능 설정

웹사이트 접근을 위한 권한 (permission) 설정

- 객체 액세스 제어 목록 (object access control list: ACL)을 이용하여 객체에 대한 public READ 권한 부여
- S3 → 버킷 → 권한 → 객체 소유권 → ACL 활성화
- ACL 편집



참고

- https://docs.aws.amazon.com/AmazonS3/latest/user guide/WebsiteHosting.html