



TEAM CBT

2022 TEAM PROJECT

AWS Multi Region

재해 대비 멀티 리전 구성

2022. 5. 20

01

팀 소개

introduce

- 팀 구성

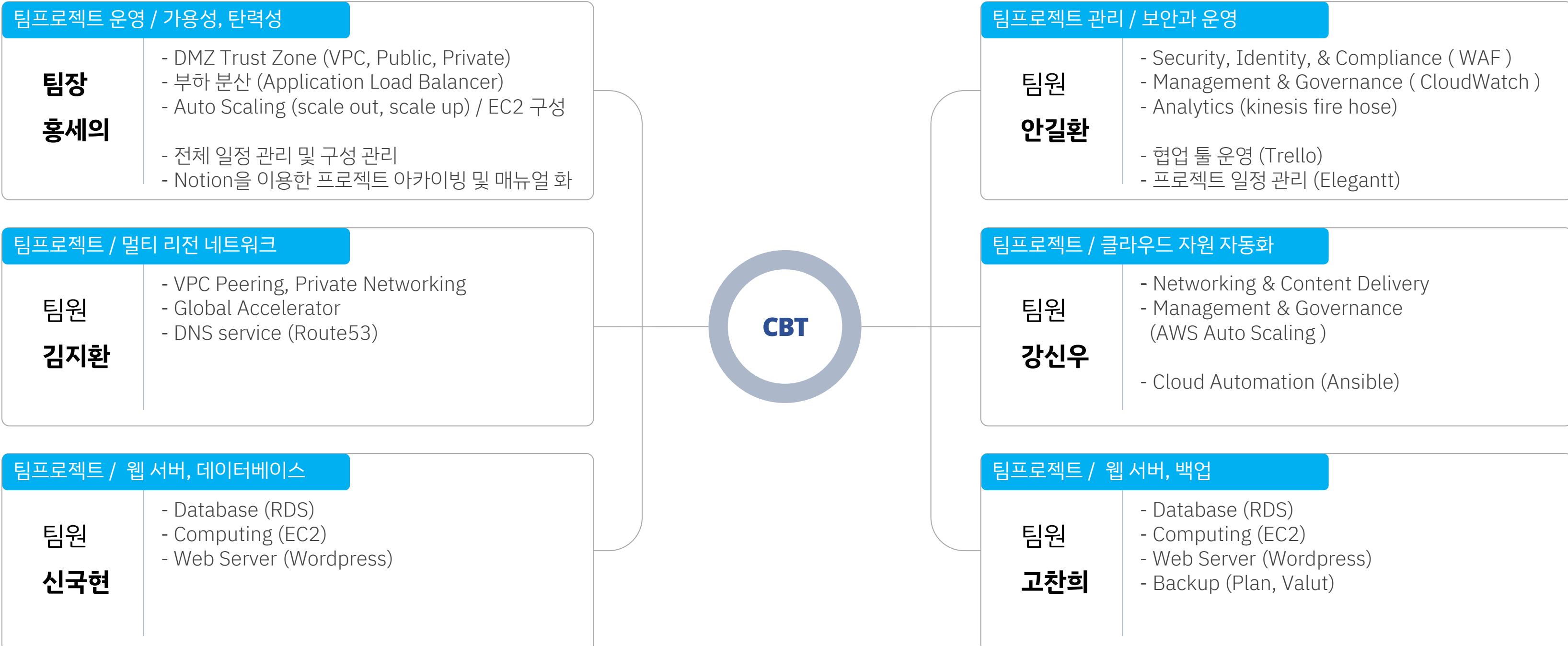
팀 소개

팀 구성

인프라 설계와 재해대비를 구성하기 위한 팀 구성이며,
성공적인 프로젝트를 위해 노하우를 아끼지 않겠습니다.

TEAM CBT

재해대비 멀티 리전 구성



02

프로젝트 개요

- AWS 도입의 이유
- 언론에 비친 AWS
- AWS 비즈니스 연속성 (BCP)
- 목표 대상 (Pilot Light)
- 재해 대비 옵션 설계
- 재해 대비 옵션 설계 (구성도)

프로젝트 개요



Q. 기업들이 AWS를 도입하는 이유가 무엇일까요?

민첩성, 탄력성, 비용절감, 몇 분 만에 전 세계에 배포

고객이 AWS를 선택하는 이유는 AWS가 제공하는 기능이 경쟁사 대비 많고, 고객과 파트너를 위해 가장 활발한 최대 규모 커뮤니티를 보유하고 있기 때문이다. 운영과 보안 측면에서 검증된 전문성을 갖췄다는 점, 비즈니스 혁신이 빠른 속도로 이뤄지며 특히 머신러닝과 AI, IoT, 서비스 컴퓨팅 등 신규 분야 혁신이 빠르다는 점도 AWS의 차별화된 경쟁력이다.

언론에 비친 AWS

한겨레

아마존 AWS, 6일 밤 또 장애...게임·배달 서비스 등 마비

천호성 기자 +구독

등록 :2022-02-07 11:50 수정 :2022-02-07 11:54

f t TALK ⚡ ★ 🔍 가+

| 20여분 간 일부 서버에서 장애 발생

 6일 저녁 아마존 클라우드 서비스에서 장애가 발생해 배달의민족과 오딘 등 아마존 클라우드를 기반으로 제공되는 앱 이용자들이 불편을 겪었다.

“AP-NORTHEAST-2 리전의 단일 가용 영역에 있는 **일부 EC2 인스턴스** (클라우드 서비스의 일종)의 **인터넷 연결에 영향을 미치는 문제가 발생했다**”

(클라우드 서비스의 일종)의 인터넷 연결에 영향을 미치는 문제가 발생했다”고 밝혔다. AP-NORTHEAST-2는 아마존 웹서비스의 서울 데이터센터다. 아마존 웹서비스는 “현재 문제가 해결됐고 연결이 복원됐다”고 덧붙였다.

이번 장애로 국내 음식배달 앱과 온라인 게임 등의 접속이 끊겼다. 아마존웹서비스는 지난해 12월에도 세차례 장애가 발생했다.

천호성 기자 rieux@hani.co.kr

언론에 비친 AWS

Byline Network AWS 고객들은 왜 멀티리전 DR을 하지 않았을까

심재석 | 2018년 11월 26일

22일 아마존의 클라우드 서비스인 아마존웹서비스(AWS)의 장애는 우리에게 적지 않은 충격을 줬다. 쿠팡, 배달의민족, 업비트, 코인원, 마켓컬리, 푹(POOQ), 야놀자, 다방, 나이키 등 주요 온라인 서비스가 모두 마비되었다. 이 서비스들이 주다대서 인터넷을 통해 서비스를 제공하는 그 자체가 온라인 서비스다.

장애 원인은 AWS의 네트워크 기반으로 서비스를 운영하는 구조 때문이었다. 네트워크 기반으로 서비스를 운영하는 구조는 네트워크 장애로 인해 서비스가 전면 중단되는 경우가 많다.

제거한 내용입니다. 이번 장애로 국내에서 AWS의 위상이 드러났다. 수많은 서비스가 AWS 기반으로 구동되고 있음이 나타났다.

쿠팡의 문제는 안정성이다. AWS가 멈추자 한국의 주요 인터넷 서비스가 멈췄다.

문제는 안정성이다. AWS가 멈추자 한국의 주요 인터넷 서비스가 멈쳤다. 7.7 디도스와 같은 대규모 해킹공격보다 더 큰 피해가 발생했다.

“이번 AWS 장애는 한국 리전에서만 벌어졌다. 일본 등 다른 리전을 이용하는 기업은 장애에서 예외였다”면서

“다른 리전을 복수로 이용했다면 이와 같은 큰 피해는 입지 않았을 것”이라고 말했다.

많은 전문가들은 이번 AWS 장애 이후 ‘멀티 리전 DR(복수의 리전에 DR 시스템을 두는 것)’의 중요성을 이야기 한다. 한 IT전문가는 “이번 AWS 장애는 한국 리전에서만 벌어졌다. 일본 등 다른 리전을 이용하는 기업은 장애에서 예외였다”면서 “다른 리전을 복수로 이용했다면 이와 같은 큰 피해는 입지 않았을 것”이라고 말했다.

“멀티 리전 DR이 필요하다는 것은 알지만, 비용이 두 배가 들기 때문에 당장 하기 어렵다”

프로젝트 개요



비즈니스 연속성 계획 (BCP)

- A. AWS를 사용하여도
장애 / 재해 대비 시스템 구축은 필요하다.

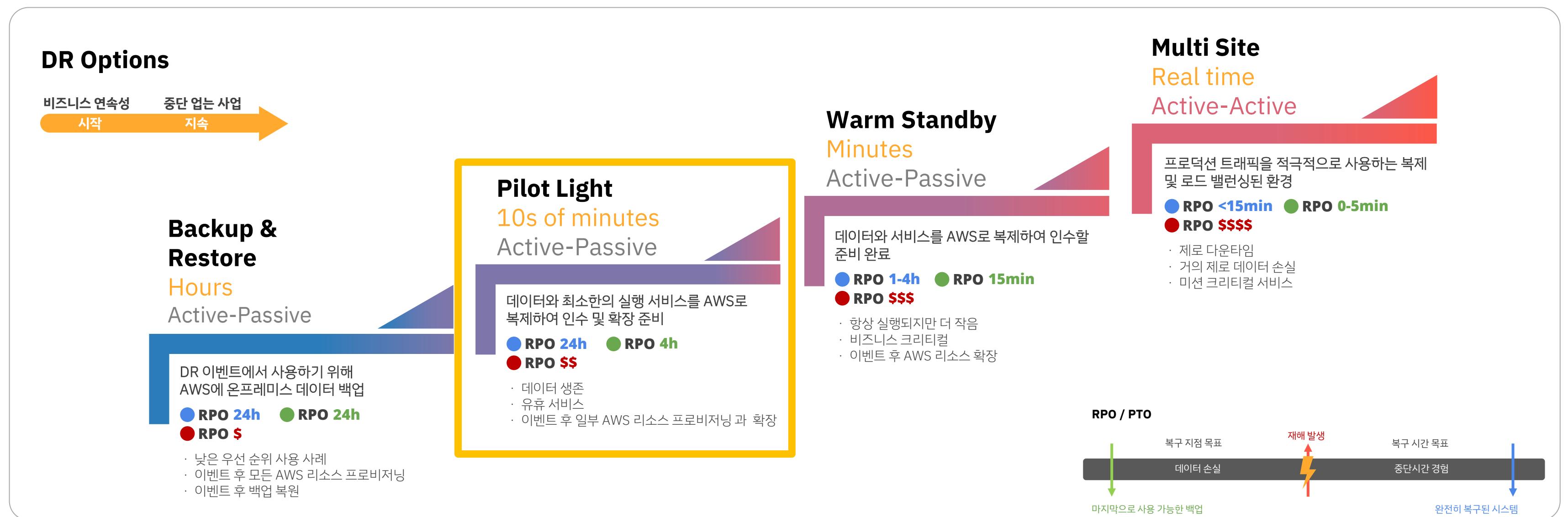
AWS 리전에서 AWS 리전으로
애플리케이션 탄력성을 높이고 여러 AWS 리전의 애플리케이션 복구에 AWS DRS를 사용하여 AWS
기반 애플리케이션의 가용성 목표를 충족할 수 있습니다.

프로젝트 개요



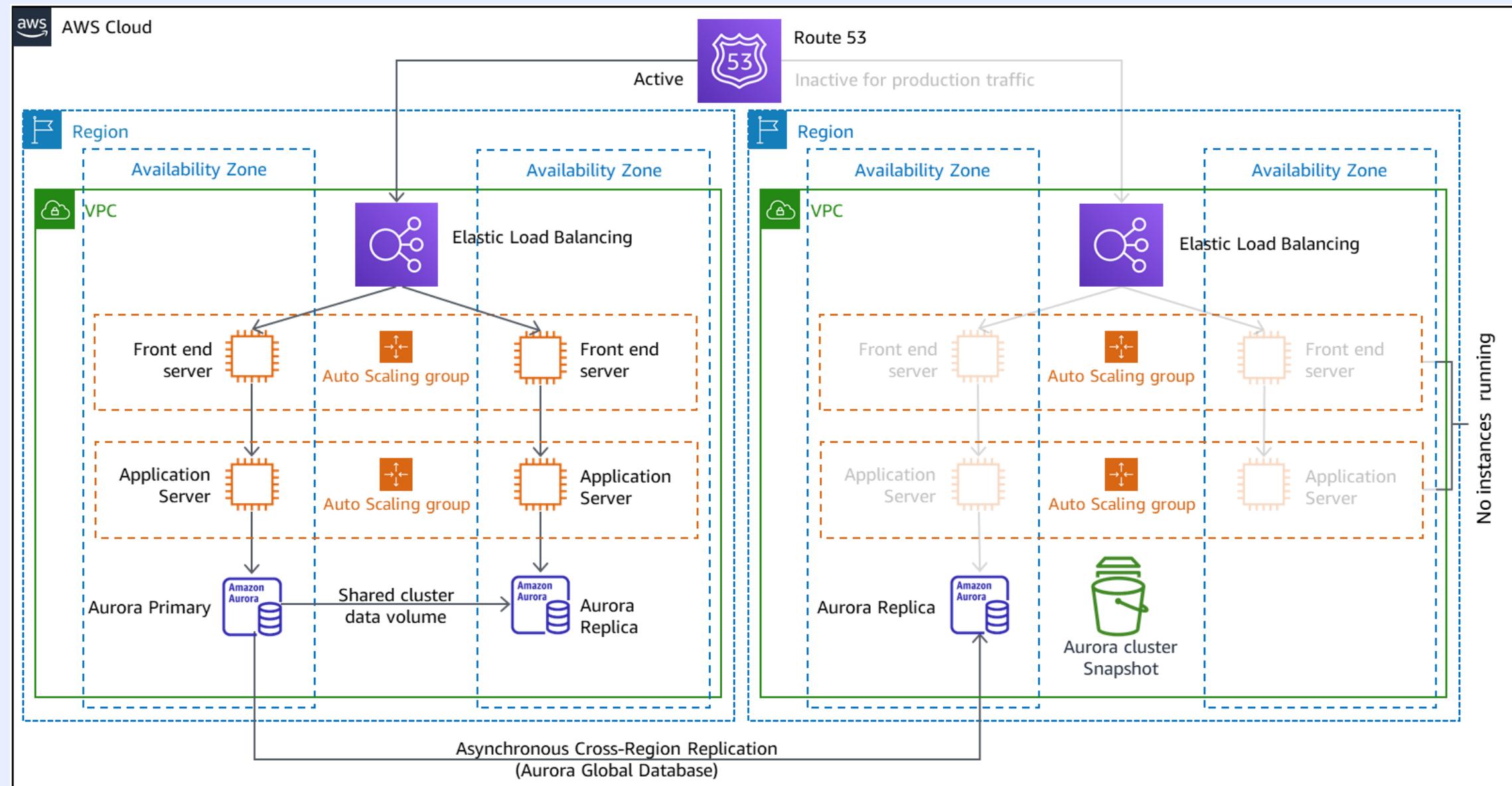
비즈니스 연속성 계획 (BCP)

모든 비즈니스의 "catch all"인 단일 재해 복구 솔루션은 없습니다. 재해 복구(DR)는 DNA와 매우 유사한 고유한 구성입니다. RTO(복구 시간 목표) 및 RPO(복구 지점 목표)를 연속성 요구 사항에 따라 고유하게 만드는 많은 요소가 있습니다. 완전히 기술에 초점을 맞추지 않은 소규모 비즈니스는 24 시간의 RTO로 괜찮을 수 있으며, **대형 온라인 회사는 가능한 한 빨리** 서버 인프라를 다시 온라인에 연결해야 합니다. 어떤 시점까지 데이터를 사용할 수 있을 것으로 예상하십니까? 일부 회사는 최대 24 시간까지 데이터 손실을 허용하고 다른 회사는 그렇지 않습니다.



프로젝트 개요

목표 대상 (Pilot Light)



1. Data Critical한 데이터를 복제하거나 미러링 하도록 EC2 인스턴스 또는 RDS 인스턴스 설정
2. AWS에서 지원되는 모든 사용자 custom 소프트웨어 패키지를 사용할 수 있는지 확인.
3. 빠른 복구가 필요한 주요 서버의 AMI를 생성해 유지
4. 이러한 서버를 정기적으로 실행하고 테스트한 후 소프트웨어 업데이트 및 Config 변경 사항 적용
5. AWS 리소스 프로비저닝 자동화를 고려.

프로젝트 개요

재해 대비 옵션 설계

aws 의 여러 자원들을 활용

Multi Region, RDS, Global Accelerator, VPC Peering, Route53

- Networking & Content Delivery
DMZ Trust Zone (VPC, Public, Private)
Application Load Balancer
VPC Peering : Private Networking
Global Accelerator
DNS service (Route53)

- Management & Governance
AWS Auto Scaling : scale out, scale up
CloudWatch

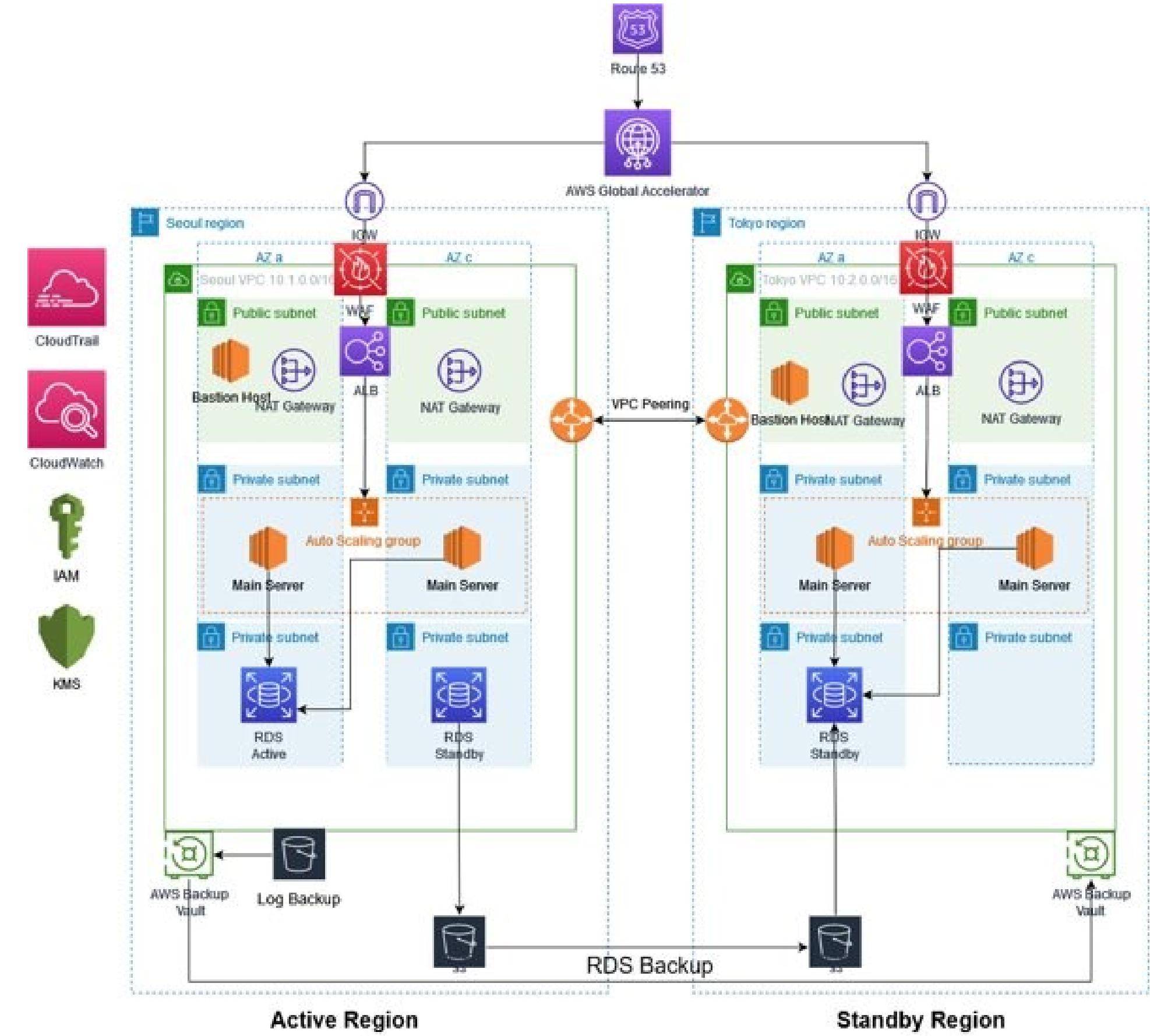
- Cloud Automation (Ansible)

- Database (RDS)
- Computing (EC2 Web Server)
- Backup (Plan, Valut)

- Security, Identity, & Compliance (WAF)
- Analytics (kinesis fire hose)

프로젝트 개요

재해 대비 옵션 설계 (구성도)



CONTENTS

01

서버

높은 가용성과 탄력성을 가지는
웹 서버의 구축

- 강신우 : EC2 부하분산, 자동확장
Ansible을 활용한 네트워크 구성 자동화
- 신국현 : Wordpress, RDS DB 구성과 백업

02

네트워크

다중 지역(국가) 간 네트워크 연결과
호스팅 서비스

- 김지환 : VPC Peering, Route53
Global Accelerator

03

보안, 운영

웹서버의 취약점에 대비하고
증적에 대한 보관

- 안길환 : WAF, Cloudwatch,
Kinesis Firehose

04

백업

재해대비 멀티 리전
AWS 백업

- 고찬희 : 백업 계획, 규칙, 볼트
백업 복원

05

총평

프로젝트 평가

- 홍세의 : 현재가치, 향후방향,
프로젝트 진행 사항
질의응답



01

서버

높은 가용성과 탄력성을 가지는 웹 서버의 구축

- 강신우
- 신국현

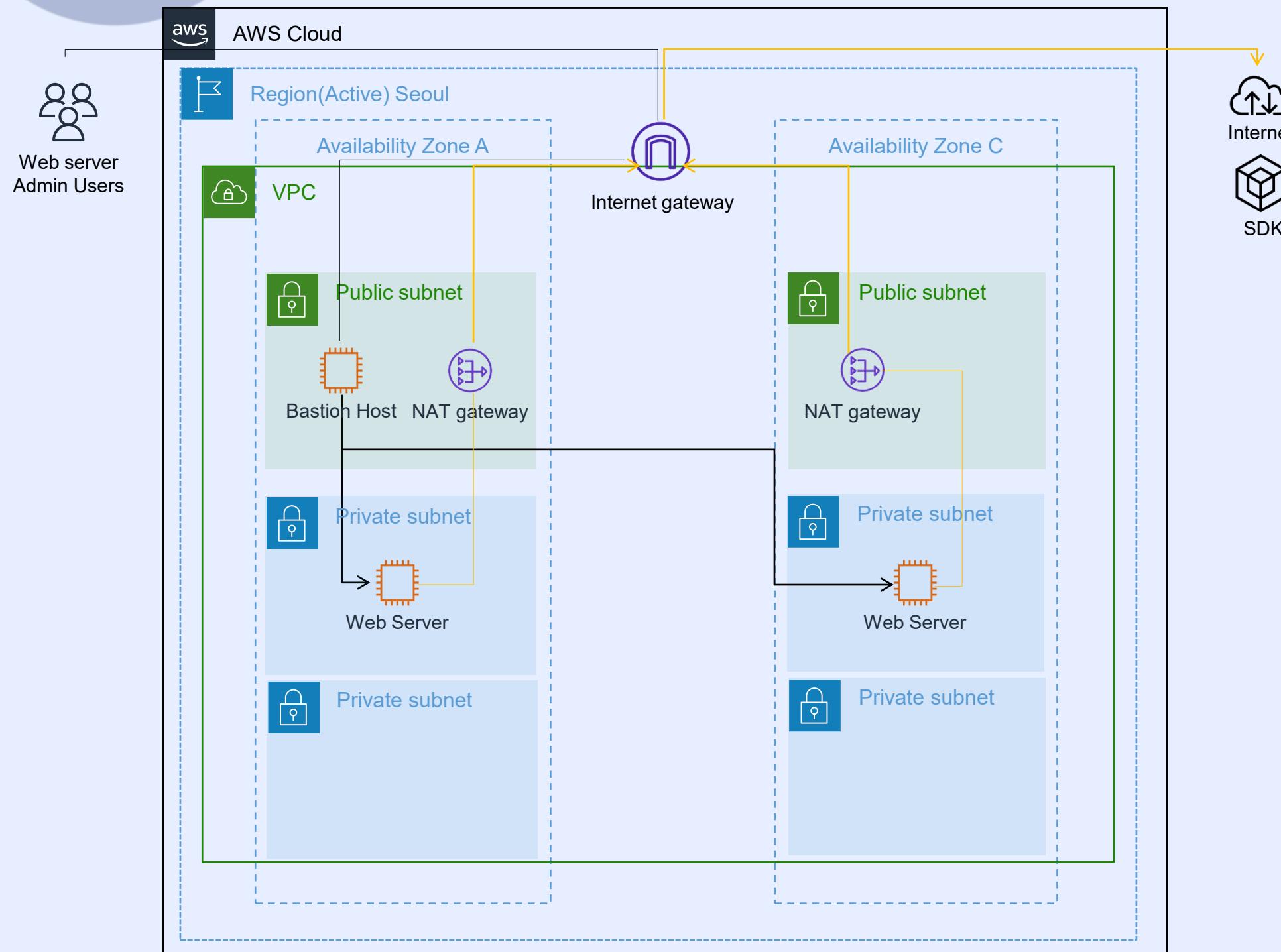
서버

강신우

EC2 부하분산, 자동확장

서버

정적 안정성 향상을 위한 분산 설계



1. 가용영역을 나누는 이유는?

한 쪽 물리적 서버에서 장애가 발생하더라도 전체 시스템이 정상적으로 작동시키기 위한 설계
가장 기초적인 시작은 가용 영역별 구분

2. Public / Private Subnet을 나누는 이유는?

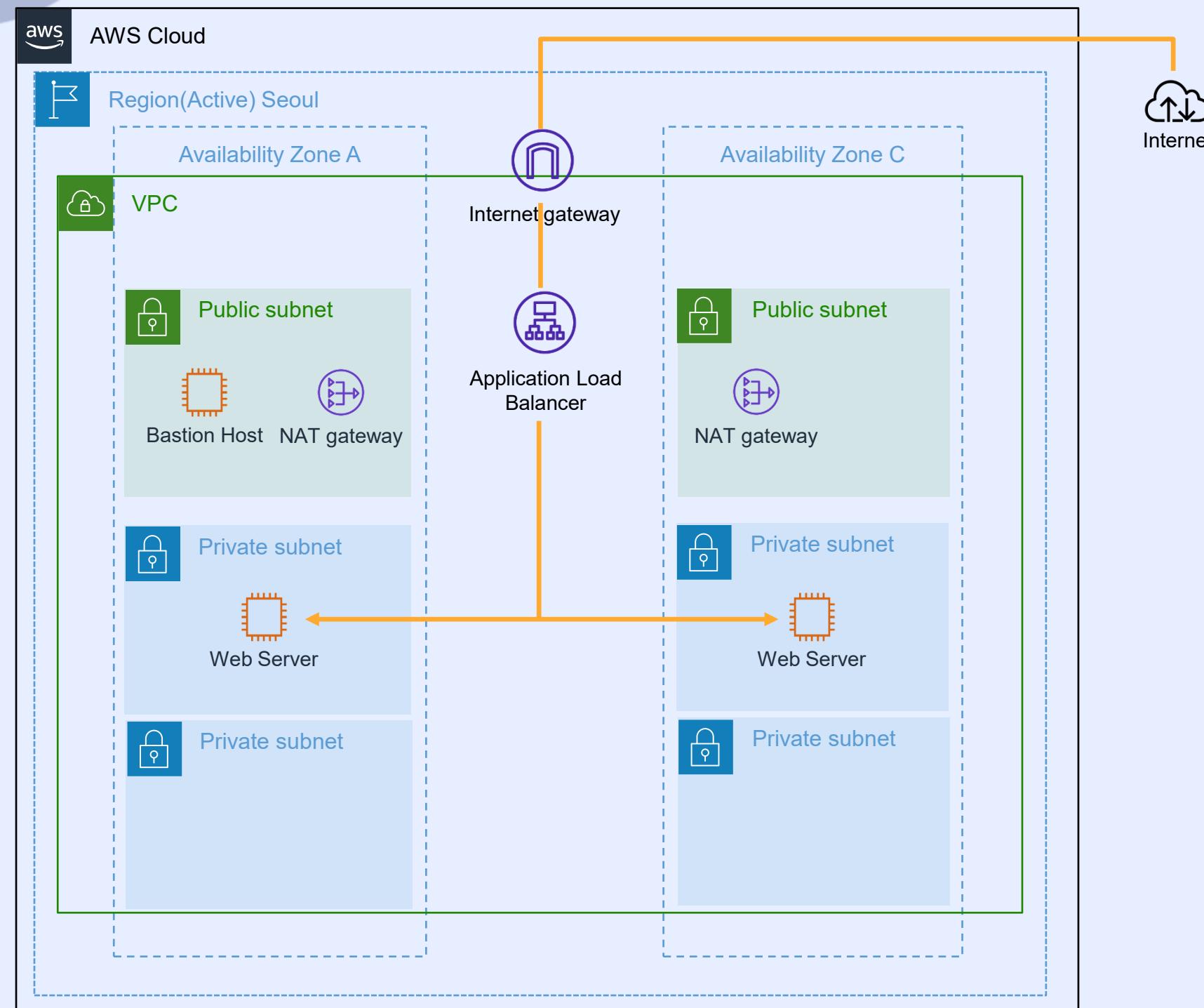
보안성 향상, 외부(인터넷)에 노출되는 내부 면적의 최소화
DMZ Trust Zone의 [내부]와 [외부]를 담당

3. DMZ Zone을 구성하여 보안 향상

- NAT Gateway
- Bastion Host

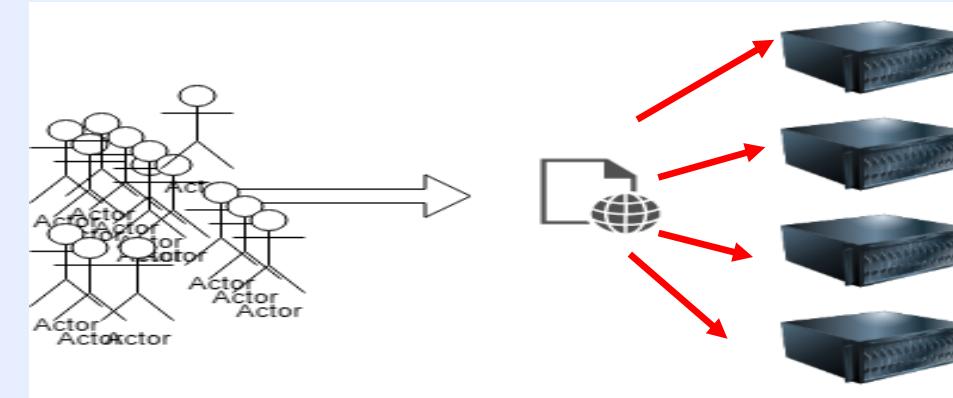
서버

부하 분산의 필요성



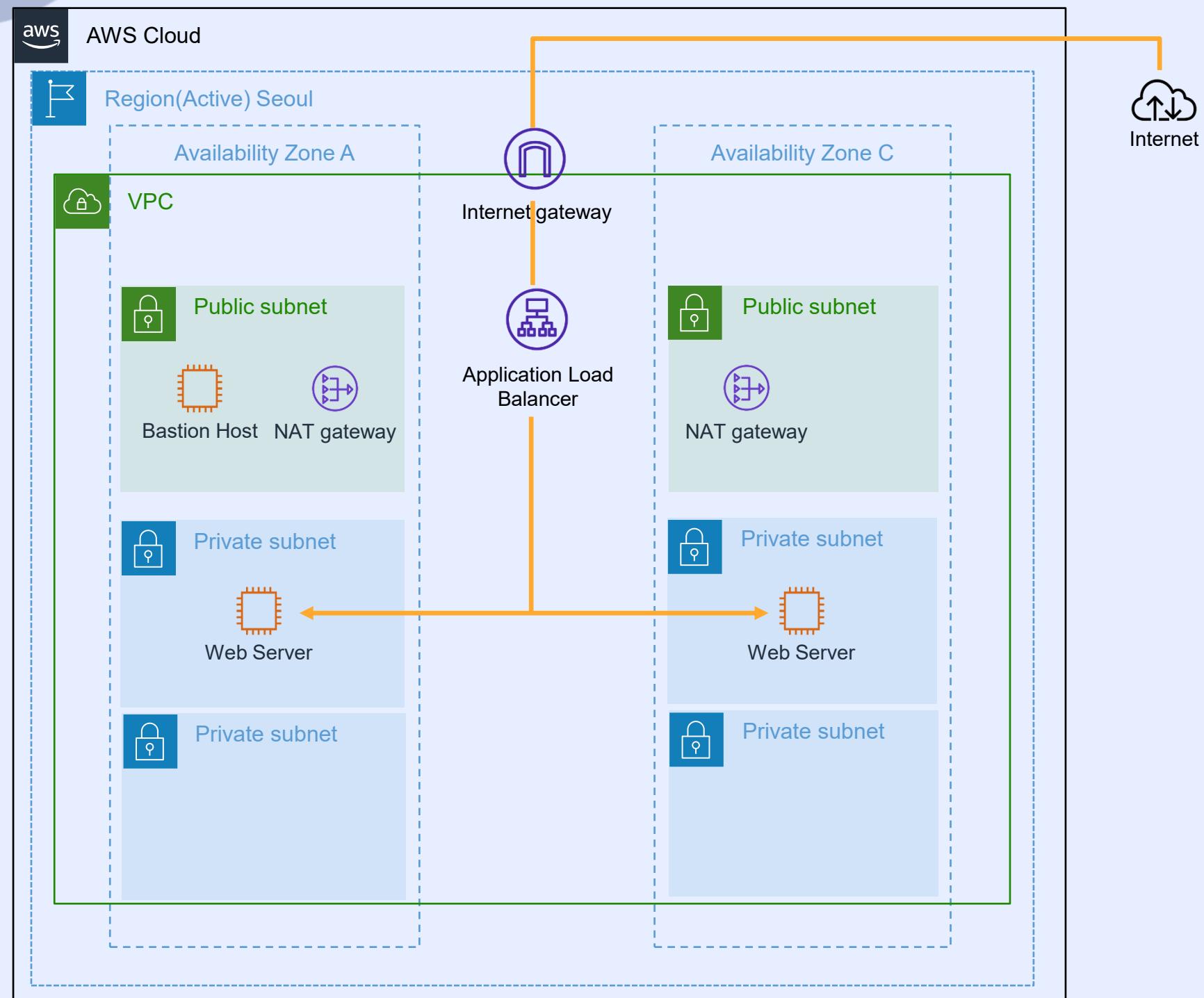
1. 실제 구성하는 서버는 한대로 동작하지 않는다
2. 서버 한대의 처리 용량은 한계가 있으므로
3. 부하분산 Load Balancer의 역할

적절하게 부하분산을 통해 여러서버가 사용자를 담당함



서버

Target Group 생성



1. Auto Scaling으로 만들어진 인스턴스에 대한
로드 밸런싱

대상 유형 선택

인스턴스

- 특정 VPC 내의 인스턴스에 대한 로드 밸런싱을 지원합니다.
- Facilitates the use of Amazon EC2 Auto Scaling to manage and scale your EC2 capacity.

2. 별도로 인스턴스를 지정하지 않고
Target Group 생성한다.

3. HTTP PORT

지금 만드는 웹페이지의 경우 포트를 http를 사용하였습니다.
대상유형에서 EC2를 선택후 추후에 오토스케일링으로 만들어지는
인스턴스를 지정할 것이므로 별도로 타겟은 지정하지 않고
넘어갑니다.

0개 선택됨

선택한 인스턴스를 위한 포트
선택한 인스턴스로 트래픽을 라우팅하기 위한 포트입니다.

80

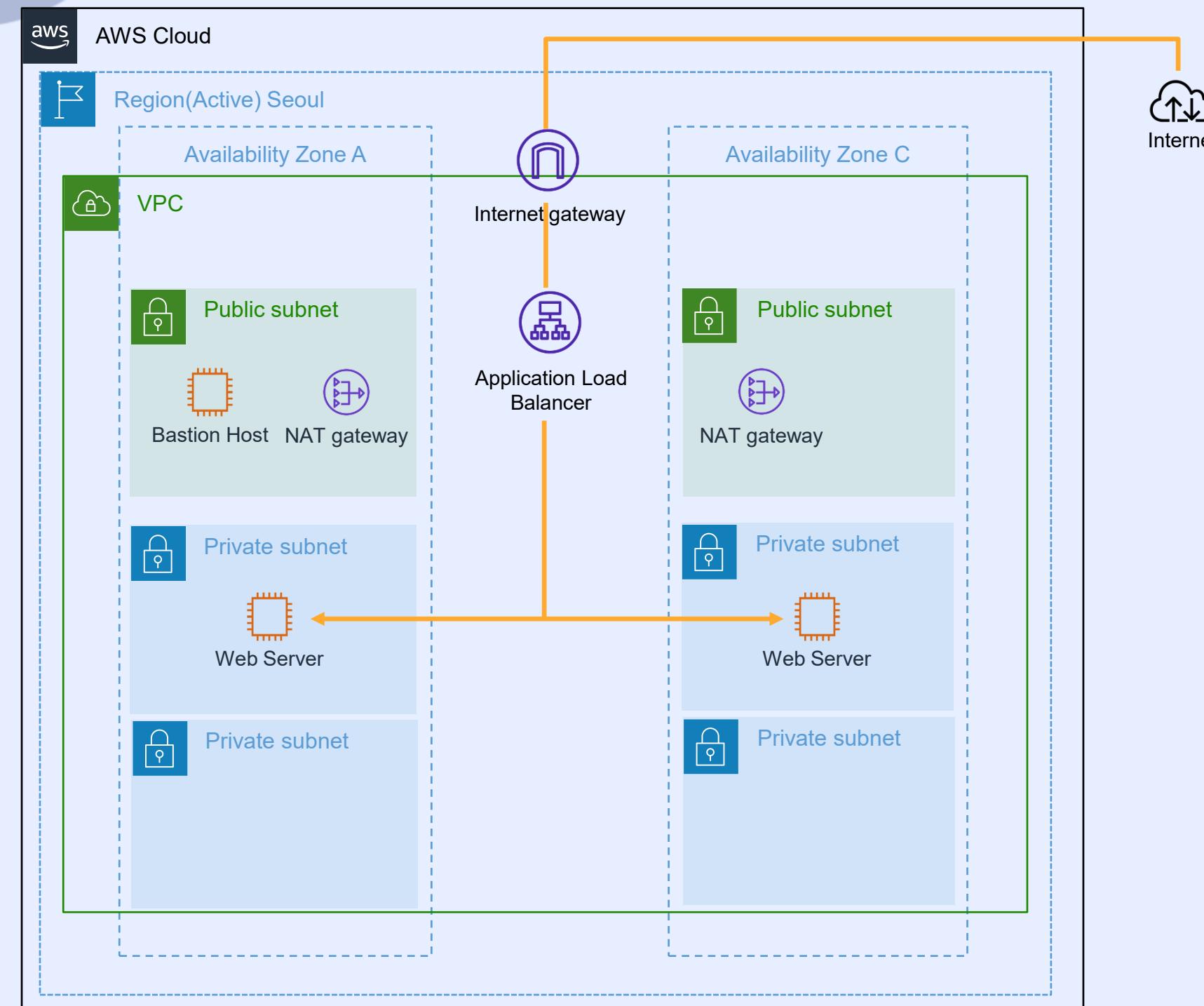
1-65535(쉼표로 여러 포트 구분)

아래에 보류 중인 것으로 포함

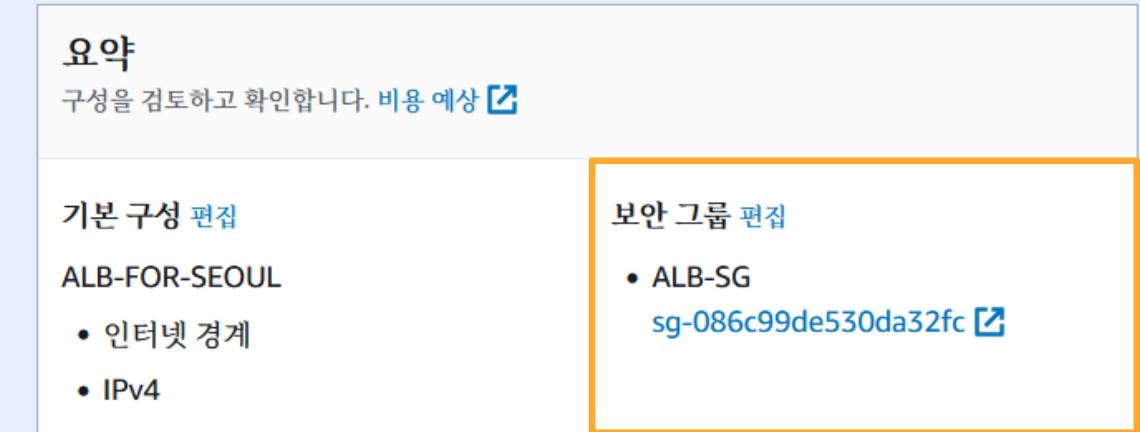
Auto Scaling 으로 활성화 된
인스턴스는 유동적이 됨

서버

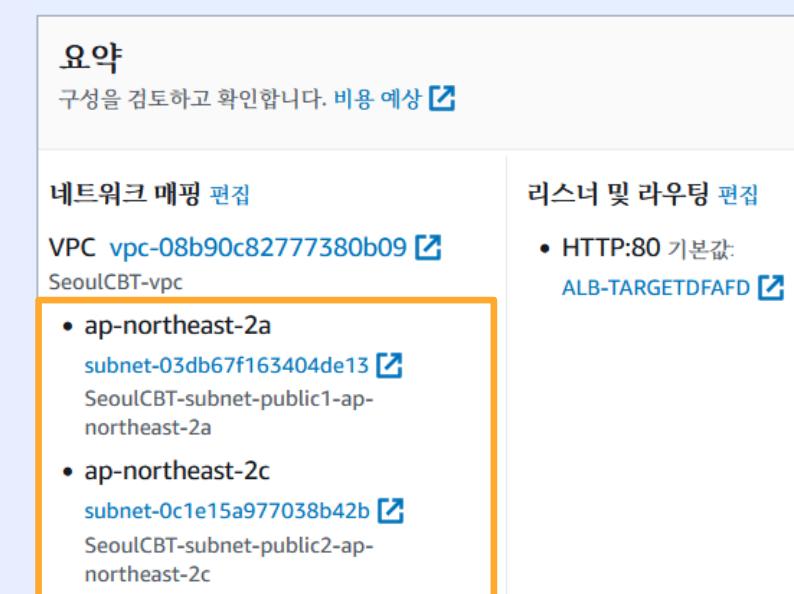
Application Load Balancer 구축



1. ALB는 인터넷과 연결되므로 SG에서 HTTP 접속을 Allow 해준다

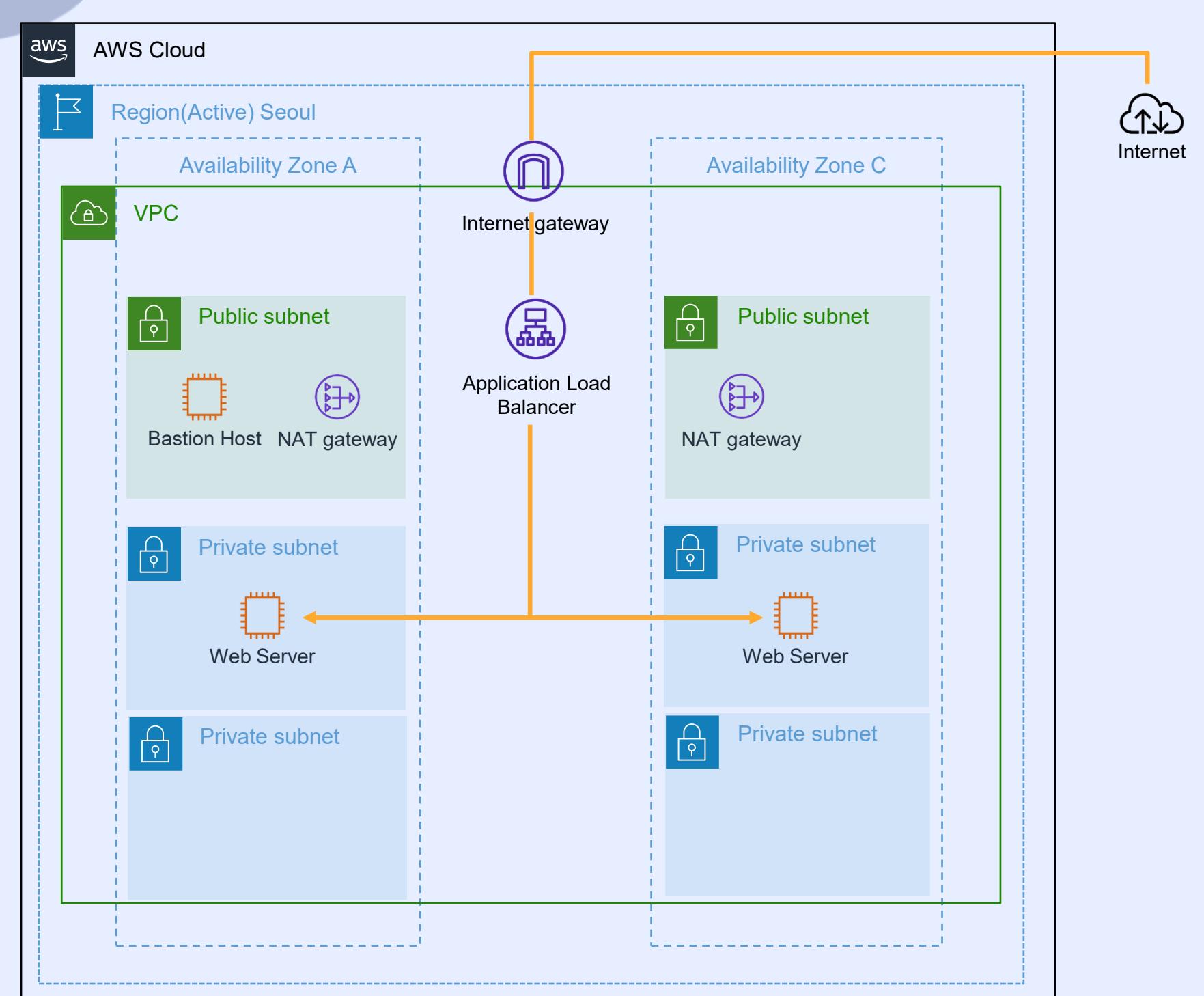


2. 가용영역 a와 c로 연결을 부하분산 하기 위해 public subnet 1, 2 지정



서버

Application Load Balancer Sticky Session



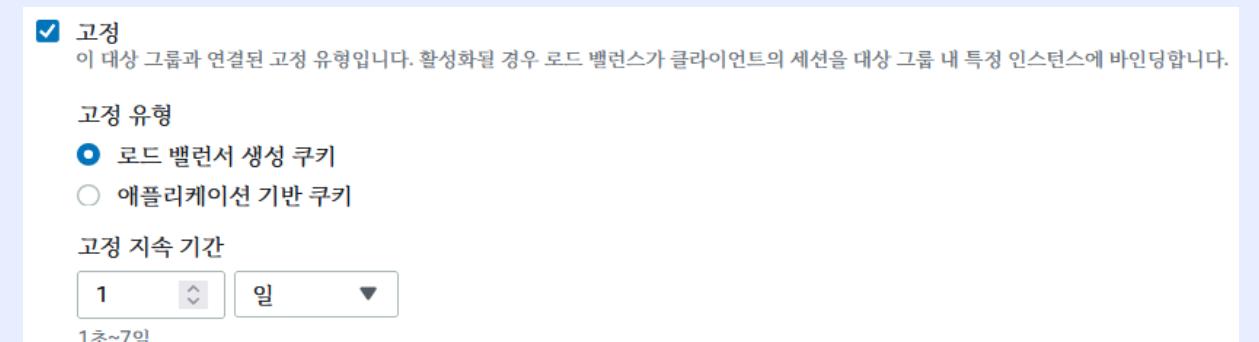
로그인 후 연결된 페이지가
로그인이 되어 있지 않다면?

a 가용영역에서 로그인 후 다른 페이지로 이동할 경우
ALB에 의해 b 가용영역의 서버로 연결될 수 있음

→ 이런 경우 a 가용영역에 있는 웹서버로 로그인 정보를 전송
했지만 ALB에 의해 b 가용영역으로 연결되었기 때문에

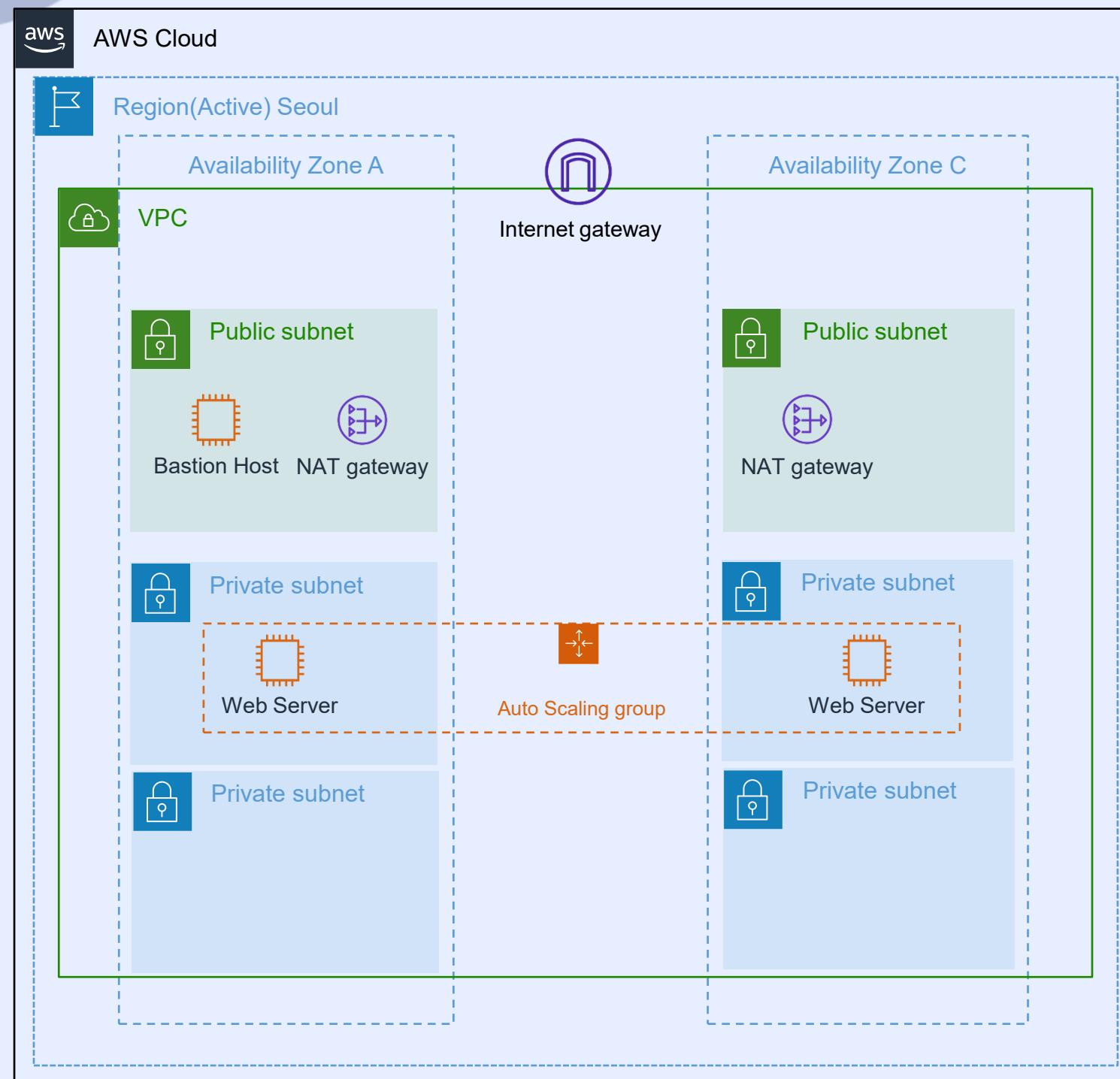
사용자 입장에서는 로그인이 초기화 된 것처럼 보인다.

→ 쿠키로 세션을 유지하는 방법으로 해결 할 수 있다.



서버

자동 인스턴스 생성 위한 AMI & Launch Template 구성

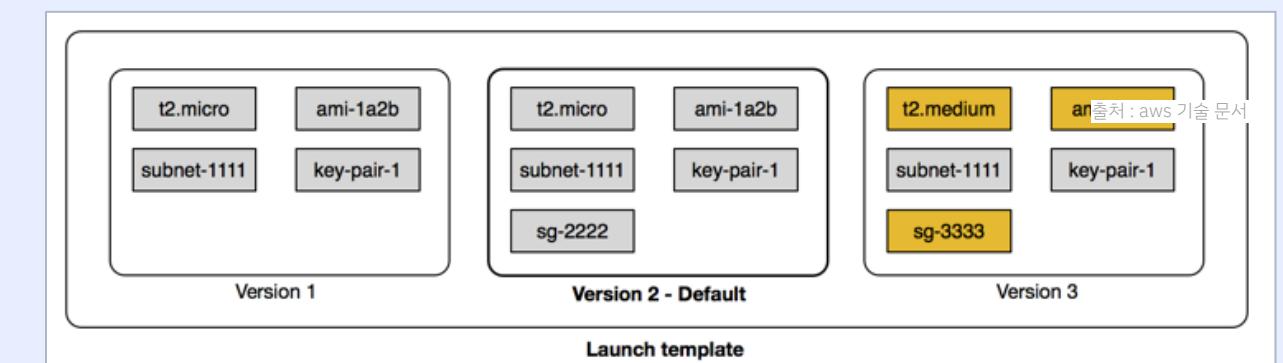


시작 템플릿과 시작구성

시작 템플릿은 기존에 만들어 둔 AMI를 바탕으로 인스턴스 시작에 필요한 구성 정보(네트워크 설정, 키 페어 등)를 지정하여 시작 파라미터를 저장하는 기능이다.

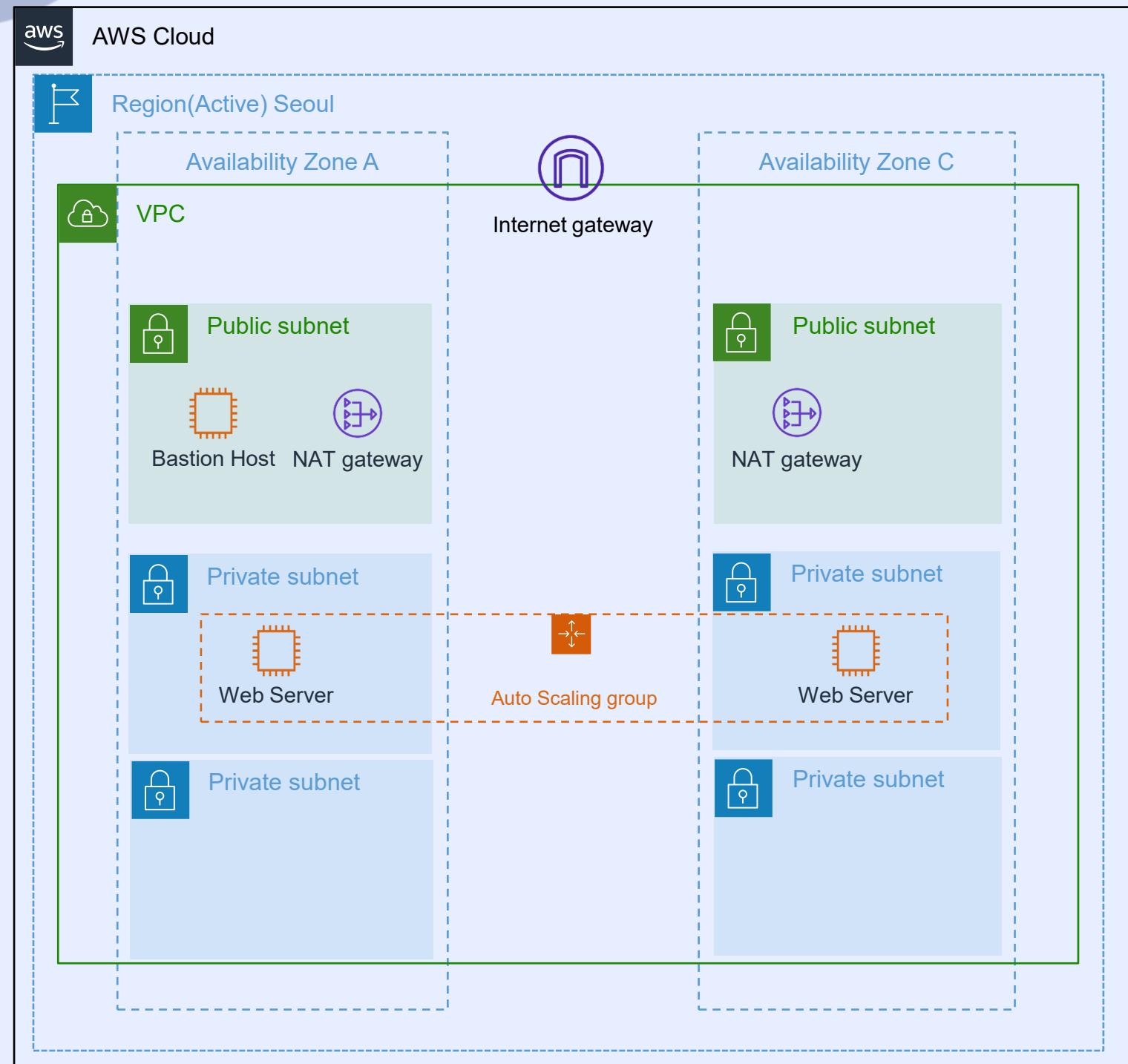
→ Auto Scaling의 수평 확장에 사용

여기서 오토스케일링에서 인스턴스 생성을 위해 **시작 템플릿과 시작구성**이 있는데, 가장 큰 차이점은 **시작템플릿은 버전관리가 가능**하다는 점에 있습니다. 반면에 **시작구성은 구성이 바뀌면 아예 새로운 시작구성을 만들어야** 합니다.



서버

Auto Scaling

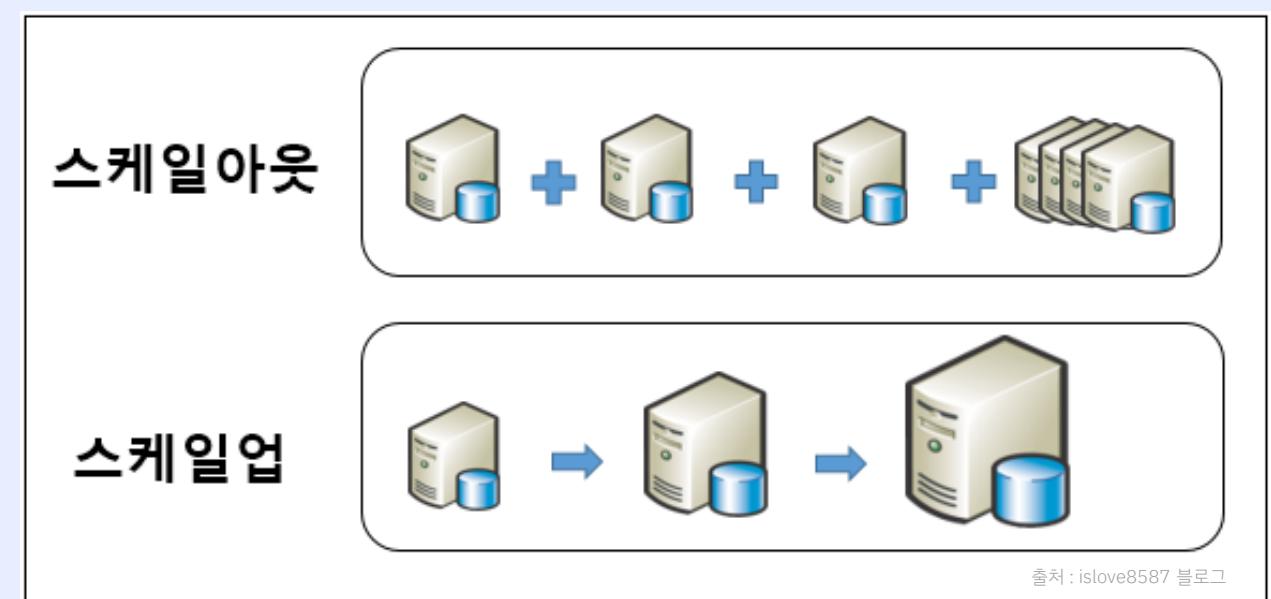


수평적 확장과 수직적 확장

AWS 서비스에서
Auto Scaling은 스케일 아웃으로 수평적 확장을 합니다.

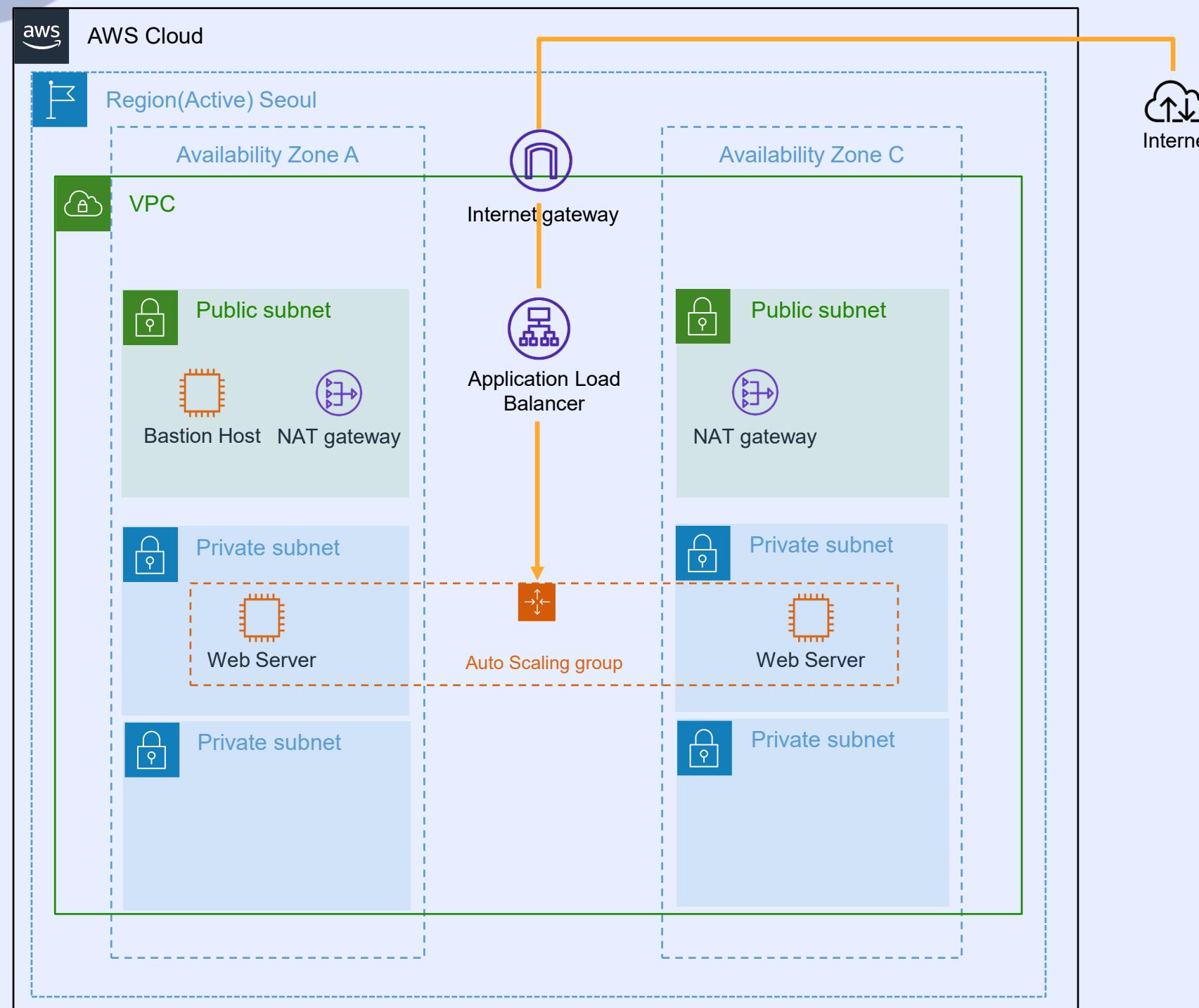
즉, 동일한 성능의 컴퓨팅 자원을 여러대를 사용하여 늘어나는 부하에 대처 합니다.

별도로 스케일업의 경우 한 서버의 컴퓨팅 자원의 성능을 높여서 부하에 대처하는 방식입니다.

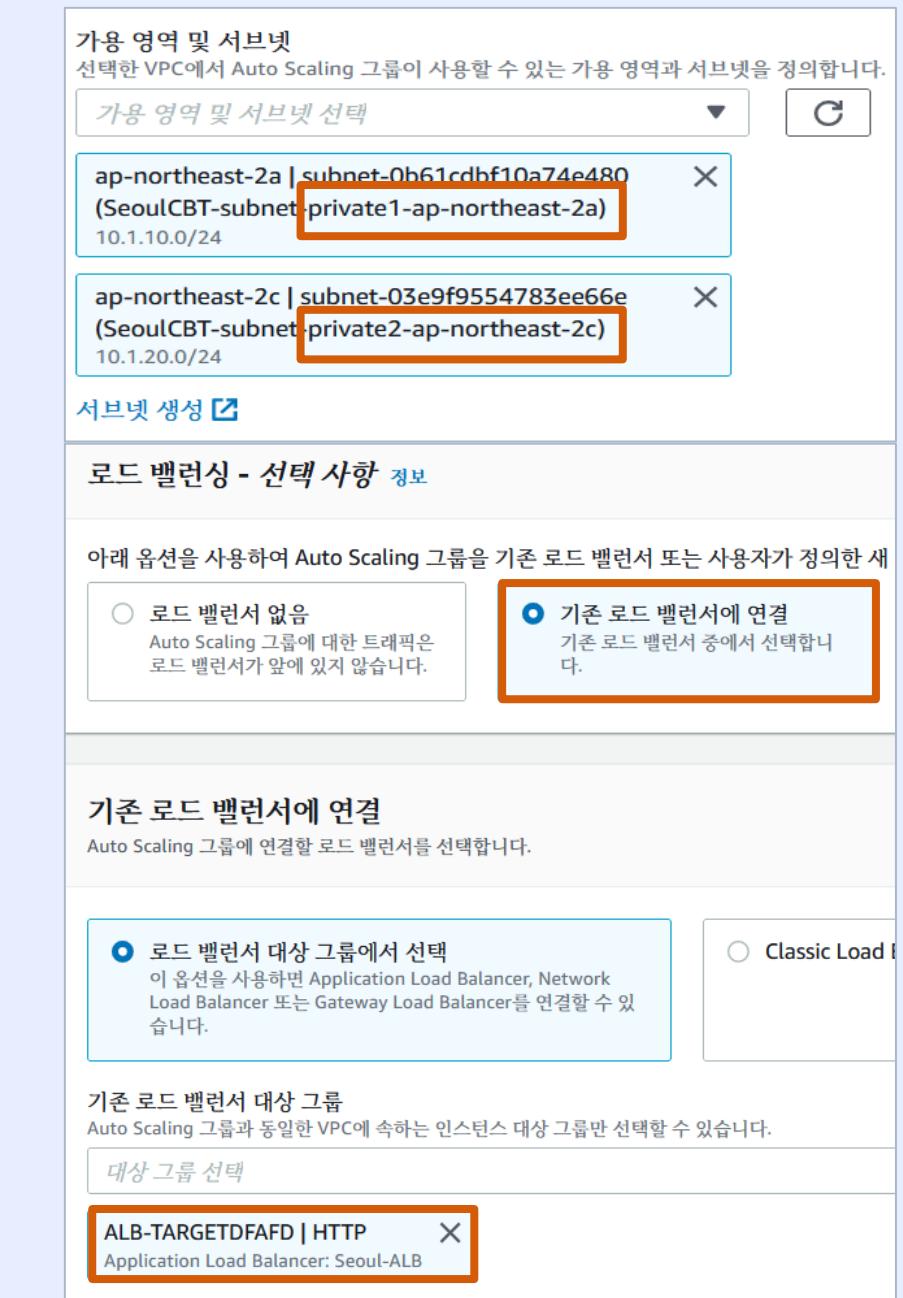


서버

Auto Scaling



로드밸런서를 연결할때, 대상그룹을 연결하여
기존로드밸런서를 연결함을 볼 수 있습니다.



서버

Auto Scaling

구성할 서버의 수를 지정

그룹 크기 - 선택 사항 정보

원하는 용량을 변경하여 Auto Scaling은 한도 범위 내에 있어야 합니다.

원하는 용량
2

최소 용량
2

최대 용량
4

어떤 것을 기준으로 조정할지 선택

조정 정책 - 선택 사항

조정 정책을 사용하여 수요의 변화를 충족하도록 Auto Scaling 그룹의 크기를 조정합니다.

● 대상 추적 조정 정책
원하는 결과를 선택하고 조정 정책에 따라 필요한 경우 용량을 추가 및 제거하여 해당 결과를 달성합니다.

조정 정책 이름
Target Tracking Policy

지표 유형

- ▶ 평균 CPU 사용률
- ▶ 평균 네트워크 입력(바이트)
- ▶ 평균 네트워크 출력(바이트)
- ▶ 대상당 Application Load Balancer 요청 수

확대 정책만 생성하려면 축소 비활성화

지표(메트릭)을 통해 용량을 조절하는 대상추적 조정 정책을 이용할 수 있습니다.

서버

Auto Scaling

작업기록을 통해 **부하가 늘어나거나 줄어들** 때 인스턴스 개수가 증감하는 것도 볼 수 있습니다.

인스턴스 (2)						
<input type="button" value="C"/> 작업 ▾						
<input type="button" value="C"/> 인스턴스 필터링						
	인스턴스 ID	수명 주기	인스턴스 유형	가중치 기반 ...	시작 템플릿/구성	가용 영역
<input type="checkbox"/>	i-01d5251357ed8862a	InService	t2.micro	-	WEB-SERVER-TEMPLA	ap-northeast-2a
<input type="checkbox"/>	i-0cd68d82273c57d62	InService	t2.micro	-	WEB-SERVER-TEMPLA	ap-northeast-2c

작업 기록 (2)						
<input type="button" value="C"/>						
<input type="button" value="C"/> 활동 기록 필터링						
상태	설명	원인		시작 시간	종료 시간	
Successful	Launching a new EC2 instance: i-01d5251357ed8862a	At 2022-05-10T12:24:25Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2022-05-10T12:24:26Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.		2022 5월 10, 09:24:28 오후 +09:00	2022 5월 10, 09:25:00 오후 +09:00	
Successful	Launching a new EC2 instance: i-0cd68d82273c57d62	At 2022-05-10T12:24:25Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2022-05-10T12:24:26Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.		2022 5월 10, 09:24:27 오후 +09:00	2022 5월 10, 09:24:59 오후 +09:00	

서버

강신우

Ansible을 활용한 네트워크 구성 자동화

서버

Ansible의 특장점



AWS 배포를 자동화할 수 있는 오픈 소스 툴

자동화 시나리오를 사용하여 응용 프로그램과 서비스를 정의, 배치 및 관리 가능함
한 번에 설정을 정의가 가능하며, 서로 다른 환경에서 일치하게 배치 가능

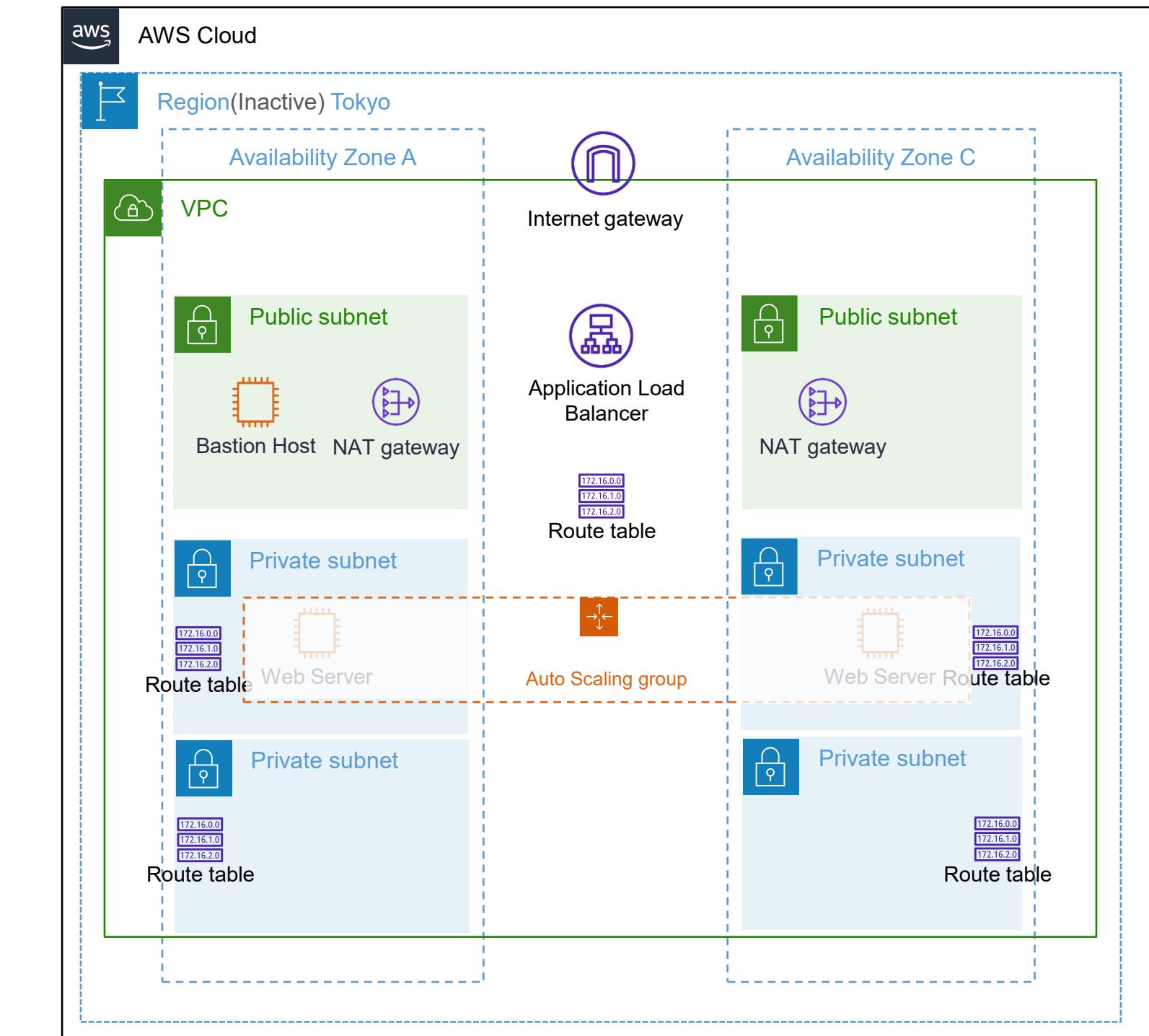
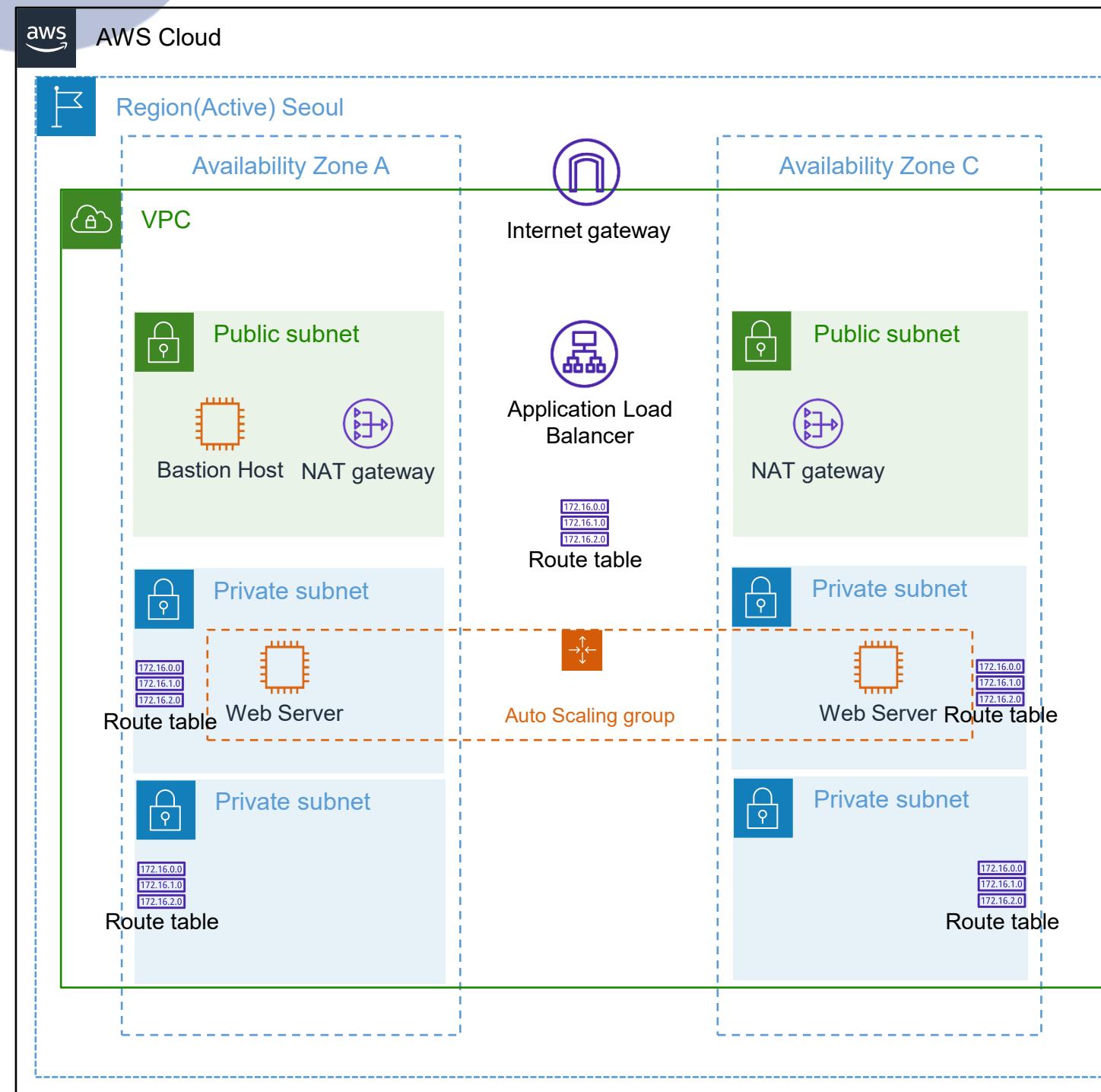
오토스케일링 환경까지 자동화를 구성하고자 자동화 툴로 ANSIBLE을 사용 하였으며

Ansible을 사용하게 된 이유로는

설정을 한번에 정의 가능할 뿐만 아니라, 서로 다른 환경에서 일관성있는 구성을 할 수 있다는 장점이 있습니다.
또한 오픈 소스라는 점, 자동화를 위한 에이전트를 시스템에 둘 필요가 없다는 것도 장점으로 꼽히고 있습니다.

서버

Ansible을 활용한 네트워크 구성 자동화



서버

Ansible을 활용한 네트워크 구성 자동화

VPC 생성 부분의 코드입니다.

엔서블 코드의 경우 서울리전의 코드를 예시로 들었습니다.



Amazon Virtual Private Cloud
(Amazon VPC)

CBT-VPC-seoul	vpc-0dfe8d65824f50c23	Available	10.1.0.0/16
<pre>---</pre>			
<pre>- name: AWS configuration using Ansible hosts: localhost vars: tasks: - name: Create a CBT-VPC-seoul ec2_vpc_net: name: CBT-VPC-seoul cidr_block: 10.1.0.0/16 region: ap-northeast-2 state: present register: vpc_result_seoul</pre>			

서버

Ansible을 활용한 네트워크 구성 자동화

클라우드 포메이션과 차이

인터넷게이트생성 부분에 어느 VPC에 붙일지 한꺼번에 지정할 수 있다는 점입니다.



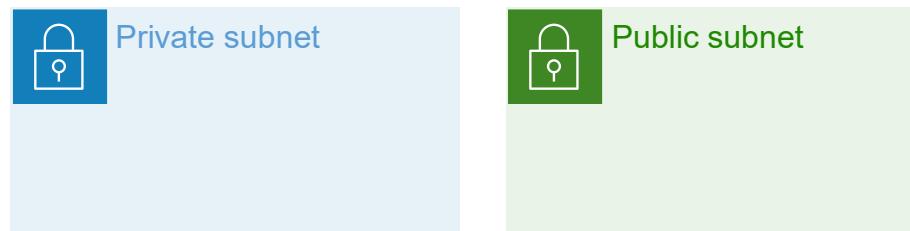
Internet gateway

CBT-IGW-seoul	igw-0e4f3f673dcfa3bad	Attached
<pre>- name: Create a CBT-IGW-seoul ec2_vpc_igw: vpc_id: "{{ vpc_result_seoul.vpc.id }}" region: ap-northeast-2 state: present tags: Name: CBT-IGW-seoul register: igw_seoul</pre>		

서버

Ansible을 활용한 네트워크 구성 자동화

map_public은 public IP를 할당할 것인가의 옵션으로
이 옵션이 켜져있는 곳에 EC2를 생성시 자동으로 public Ip 가 할당 됩니다..



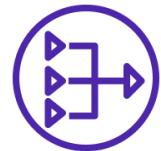
Subnet Name	Subnet ID	Status	VPC ID	CIDR Range
CBT-Public-SN-1-seoul	subnet-0634d67dcf95ef9f3	Available	vpc-0dfe8d65824f50c23 CBT...	10.1.1.0/24
CBT-Public-SN-2-seoul	subnet-03b6721e9fb810f2d	Available	vpc-0dfe8d65824f50c23 CBT...	10.1.2.0/24
CBT-Private-SN-1-seoul	subnet-0f145c56c22b974ac	Available	vpc-0dfe8d65824f50c23 CBT...	10.1.3.0/24
CBT-Private-SN-2-seoul	subnet-0a3dfdc909152c383	Available	vpc-0dfe8d65824f50c23 CBT...	10.1.4.0/24
CBT-Private-SN-3-seoul	subnet-06e8d659c50caf2e5	Available	vpc-0dfe8d65824f50c23 CBT...	10.1.5.0/24
CBT-Private-SN-4-seoul	subnet-0a409e35b026336be	Available	vpc-0dfe8d65824f50c23 CBT...	10.1.6.0/24

```
- name: Create a CBT-Public-SN-1-seoul
  ec2_vpc_subnet:
    cidr: 10.1.1.0/24
    vpc_id: "{{ vpc_result_seoul.vpc.id }}"
    region: ap-northeast-2
    az: ap-northeast-2a
    map_public: yes
    state: present
    tags:
      Name: CBT-Public-SN-1-seoul
  register: public_subnet_result_seoul_1
```

서버

Ansible을 활용한 네트워크 구성 자동화

Ansible 코드를 재 실행했을 때 서브넷 중복적으로
NAT Gateway 생성을 막기 위해 `if_exist_do_not_create` 옵션을 주었습니다.



NAT gateway

Name	NAT gateway ID	Connectivit...	State
CBT-NAT_gateway-2-seoul	nat-08384fa9323e77092	Public	Available
CBT-NAT_gateway-1-seoul	nat-09d54659ff6eb31c5	Public	Available


```
- name: Create a CBT-NAT_gateway-1-seoul
  amazon.aws.ec2_vpc_nat_gateway:
    state: present
    subnet_id: "{{ public_subnet_result_seoul_1.subnet.id }}"
    wait: true
    region: ap-northeast-2
    tags:
      Name: CBT-NAT_gateway-1-seoul
      if_exist_do_not_create: true
    register: nat_gateway_1_seoul
```

서버

Ansible을 활용한 네트워크 구성 자동화

앤서블 재 실행시 이미 있는 라우트 테이블이 또 생성되지 않게 하기 위해
Lookup:tag를 사용하여 같은 tag의 라우트 테이블이 중복 생성되지 않게 하였습니다.

172.16.0.0
172.16.1.0
172.16.2.0

Route tabl

CBT-Public-RT-1-seoul	rtb-01b9c930f6837aef4	2 subnets
CBT-Private-RT-1-seoul	rtb-0c53a15249a69a239	2 subnets
CBT-Private-RT-2-seoul	rtb-0ae71d1572710d34a	2 subnets


```
- name: Create a CBT-Public-RT-seoul
  ec2_vpc_route_table:
    vpc_id: "{{ vpc_result_seoul.vpc.id }}"
    region: ap-northeast-2
    state: present
    lookup: tag
    tags:
      Name: CBT-Public-RT-1-seoul
    subnets:
      - "{{ public_subnet_result_seoul_1.subnet.id}}"
      - "{{ public_subnet_result_seoul_2.subnet.id}}"
    routes:
      - dest: 0.0.0.0/0
        gateway_id: "{{ igw_seoul.gateway_id }}"
```

서버

Ansible을 활용한 네트워크 구성 자동화

베스천 호스트의 보안그룹 코드를 가져왔습니다.
룰 부분에 허용 아이피 대역이나 보안그룹 이름등의 설정을 할 수 있습니다.

Security group

```
- name: Create a CBT-SG-seoul
  ec2_group:
    name: CBT-SG-seoul
    vpc_id: "{{ vpc_result_seoul.vpc.id }}"
    region: ap-northeast-2
    state: present
    description: allow ssh
    tags:
      Name: CBT-SG-seoul
    rules:
      - proto: tcp
        ports:
          - 22
        cidr_ip: 0.0.0.0/0
```

서버

Ansible을 활용한 네트워크 구성 자동화

이 부분에서 유의미하게 볼 파라미터로는 `exact_count`가 있습니다. 비교적 최근 버전에 추가된 파라미터로 같은 태그의 EC2를 정확히 1개 생성하고 다음번 앤서블 실행에도 원하는 수량의 인스턴스 개수를 유지할 수 있습니다.



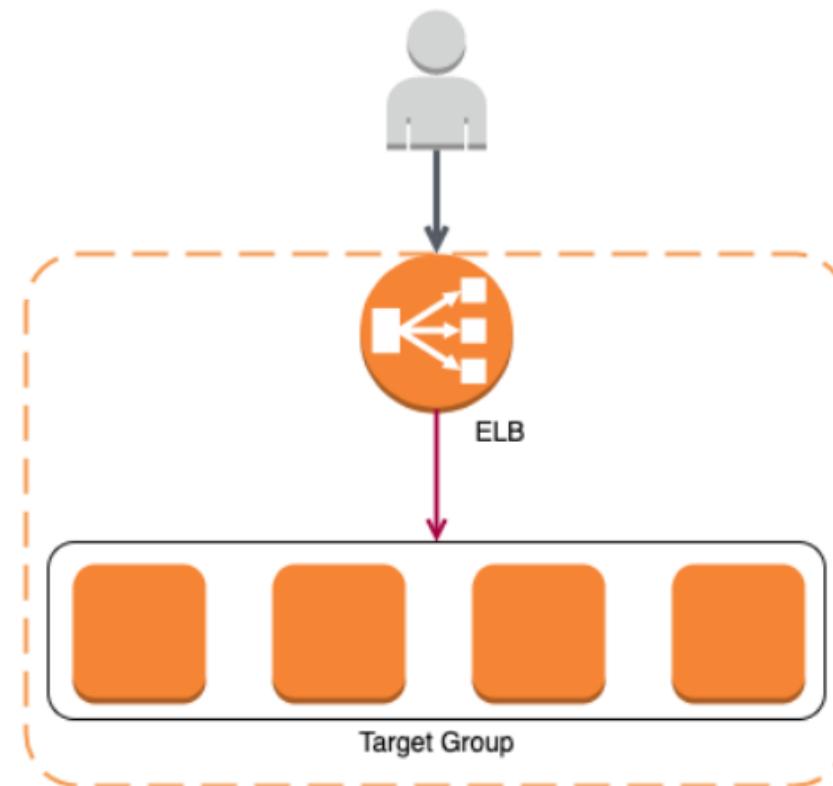
Amazon Elastic Compute
Cloud (Amazon EC2)

CBT-bastion_EC2_seoul	i-04d0b9ed31028e1bd	Running
<pre>- name: Create a CBT-bastion-EC2-seoul amazon.aws.ec2_instance: key_name: CBT-KEY-seoul name: CBT-bastion_EC2_seoul region: ap-northeast-2 instance_type: t2.micro image_id: ami-0225bc2990c54ce9a security_group: CBT-SG-seoul wait: yes exact_count: 1 vpc_subnet_id: "{{ public_subnet_result_seoul_1.subnet.id }}" network: assign_public_ip: true</pre>		

서버

Ansible을 활용한 네트워크 구성 자동화

타겟그룹에 관련된 코드입니다.

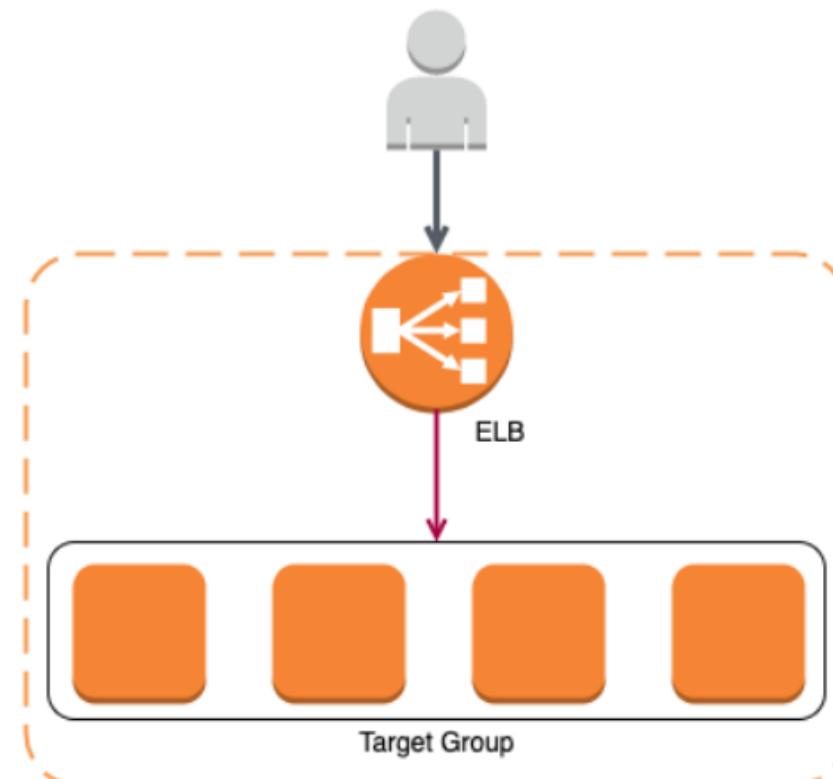


```
- name: Create a CBT-TG-seoul with a default health check
  community.aws.elb_target_group:
    name: CBT-TG-seoul
    protocol: http
    port: 80
    vpc_id: "{{ vpc_result_seoul.vpc.id }}"
    state: present
    region: ap-northeast-2
    register: CBT_TG_seoul
```

서버

Ansible을 활용한 네트워크 구성 자동화

로드밸런서에 대한 코드로 타겟그룹이 파라미터로 들어갔으며
인터넷과 직접 연결된 로드밸런서로 internet-facing 옵션을 주었습니다.

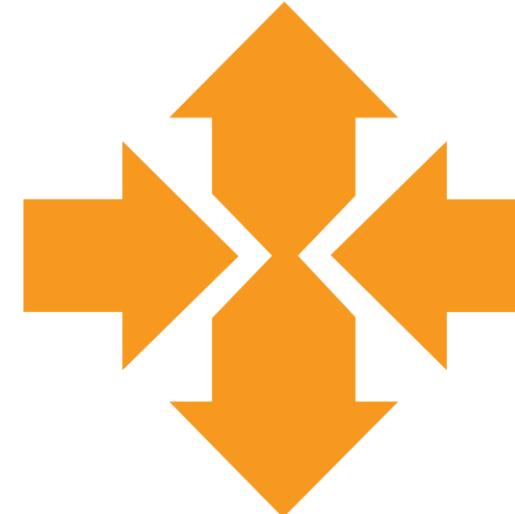


```
- name: Create a CBT-ALB-seoul
  elb_application_lb:
    name: CBT-ALB-seoul
    subnets:
      - "{{ public_subnet_result_seoul_1.subnet.id }}"
      - "{{ public_subnet_result_seoul_2.subnet.id }}"
    security_groups: "{{ CBT_ALB_SG_seoul.group_id }}"
    scheme: internet-facing
    region: ap-northeast-2
    listeners:
      - Protocol: HTTP
        Port: 80
        DefaultActions:
          - Type: forward
            TargetGroupName: CBT-TG-seoul
    state: present
    register: CBT_ALB_seoul
```

서버

Ansible을 활용한 네트워크 구성 자동화

복구를 하거나 새로운 리전에 환경을構성을 할 때 사용할 수 있도록 코드를 짜두어서 시작 구성은 넣었습니다.

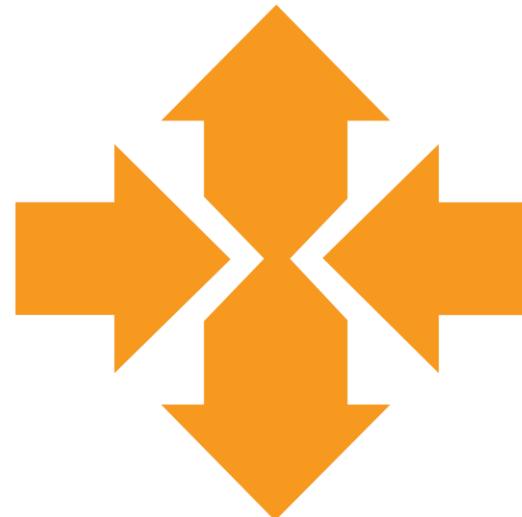


```
- name: create CBT-Launch Template-Seoul
  community.aws.ec2_launch_template:
    name: "CBT-LT-seoul"
    image_id: "ami-0225bc2990c54ce9a"
    key_name: CBT-KEY-seoul
    instance_type: t2.micro
    region: ap-northeast-2
    security_group_ids: [ "{{ CBT_ASG_SG_seoul.group_id }} " ]
    tags:
      Name: CBT-Main-Server-seoul
```

서버

Ansible을 활용한 네트워크 구성 자동화

여기서 주의해야 할 점은 오토스케일링 그룹 코드의 경우 ALB를 연결해줄 때 target 그룹의 arn을 파라미터로 넣어줘야 한다는 것이었습니다. 별도의 로드밸런서 파라미터가 있는데 이것의 경우 클래식 로드밸런서를 사용할 때 쓰입니다.



```
- name: Create CBT-ASG-seoul
  community.aws.ec2_asg:
    name: CBT-ASG-seoul
    target_group_arns: [ "{{ CBT_TG_seoul.target_group_arn }}" ]
    availability_zones: [ 'ap-northeast-2a', 'ap-northeast-2c' ]
    launch_template:
      version: '1'
      launch_template_name: 'CBT-LT-seoul'
    region: ap-northeast-2
    min_size: 2
    max_size: 4
    desired_capacity: 2
    vpc_zone_identifier:
      - "{{ private_subnet_result_seoul_3.subnet.id }}"
      - "{{ private_subnet_result_seoul_4.subnet.id }}
```

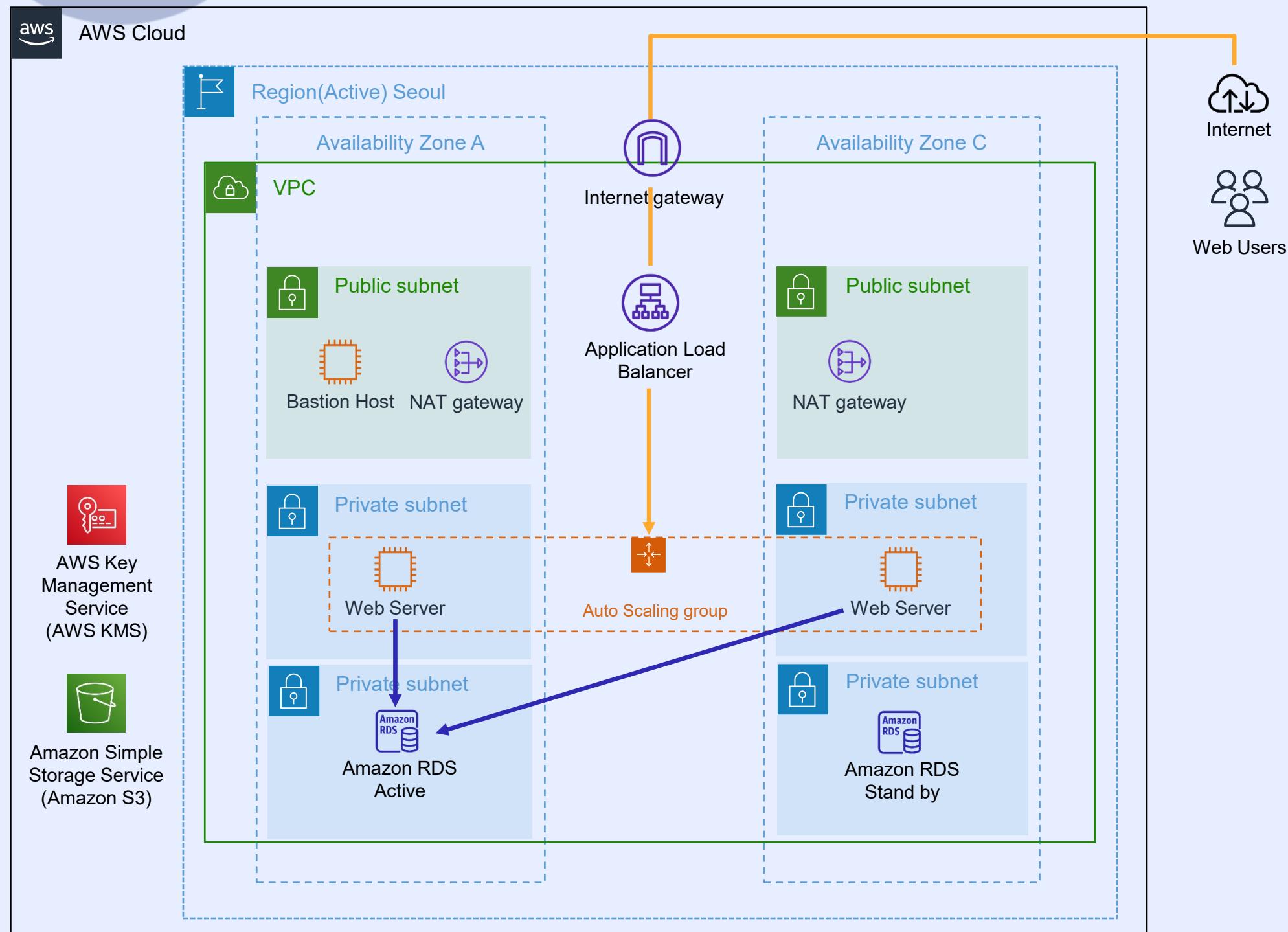
서버

신국현

Wordpress, RDS DB 구성과 백업

서버

구성 목표



1. RDS 소개

2. Wordpress 와 RDS DB : MySQL 구축



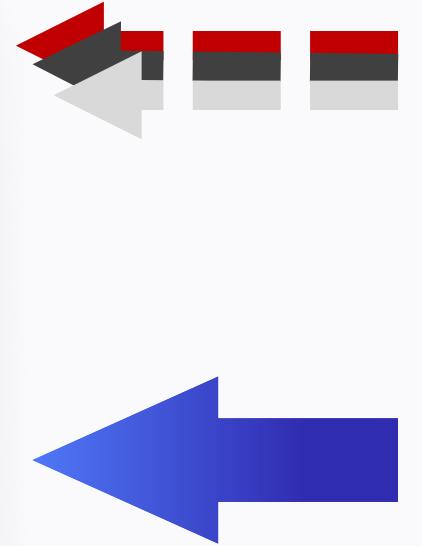
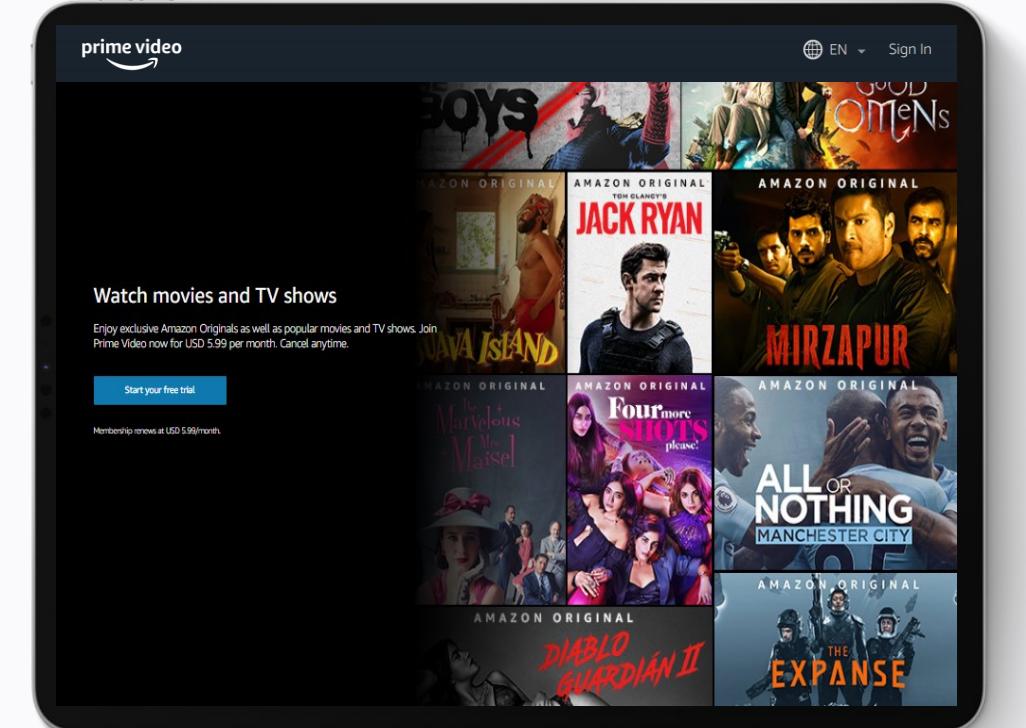
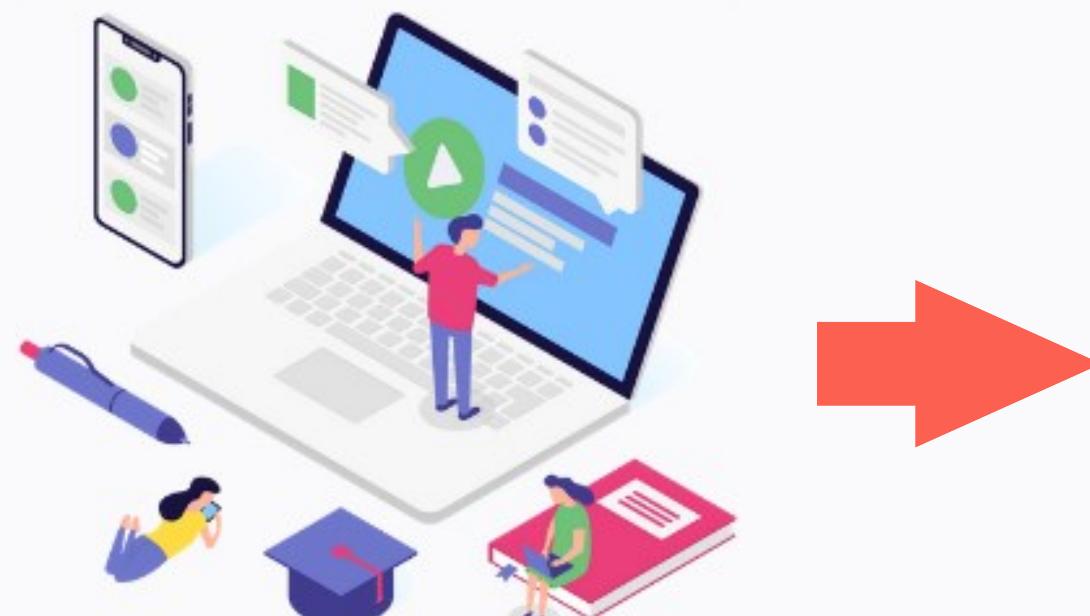
3. Multi-AZ의 Failover 매커니즘

4. S3 Snapshot export 가능

서버

웹서버와 DB 단일서버 구축방식

비 효율적인 DB in EC2



DB instance



Amazon Relational Database Service (Amazon RDS)

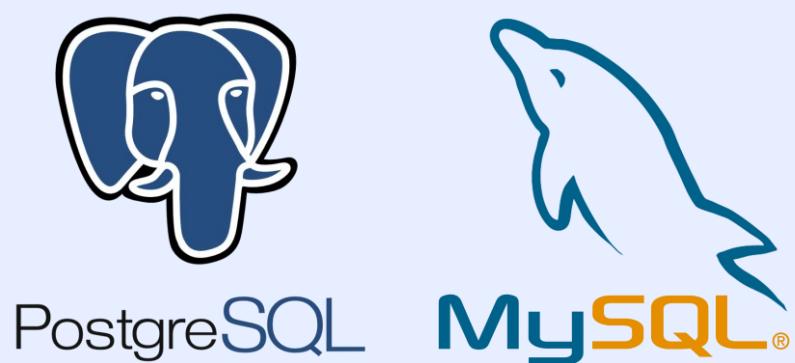
서버

DB 및 RDS 소개



(Amazon RDS)

Amazon Relational Database Service

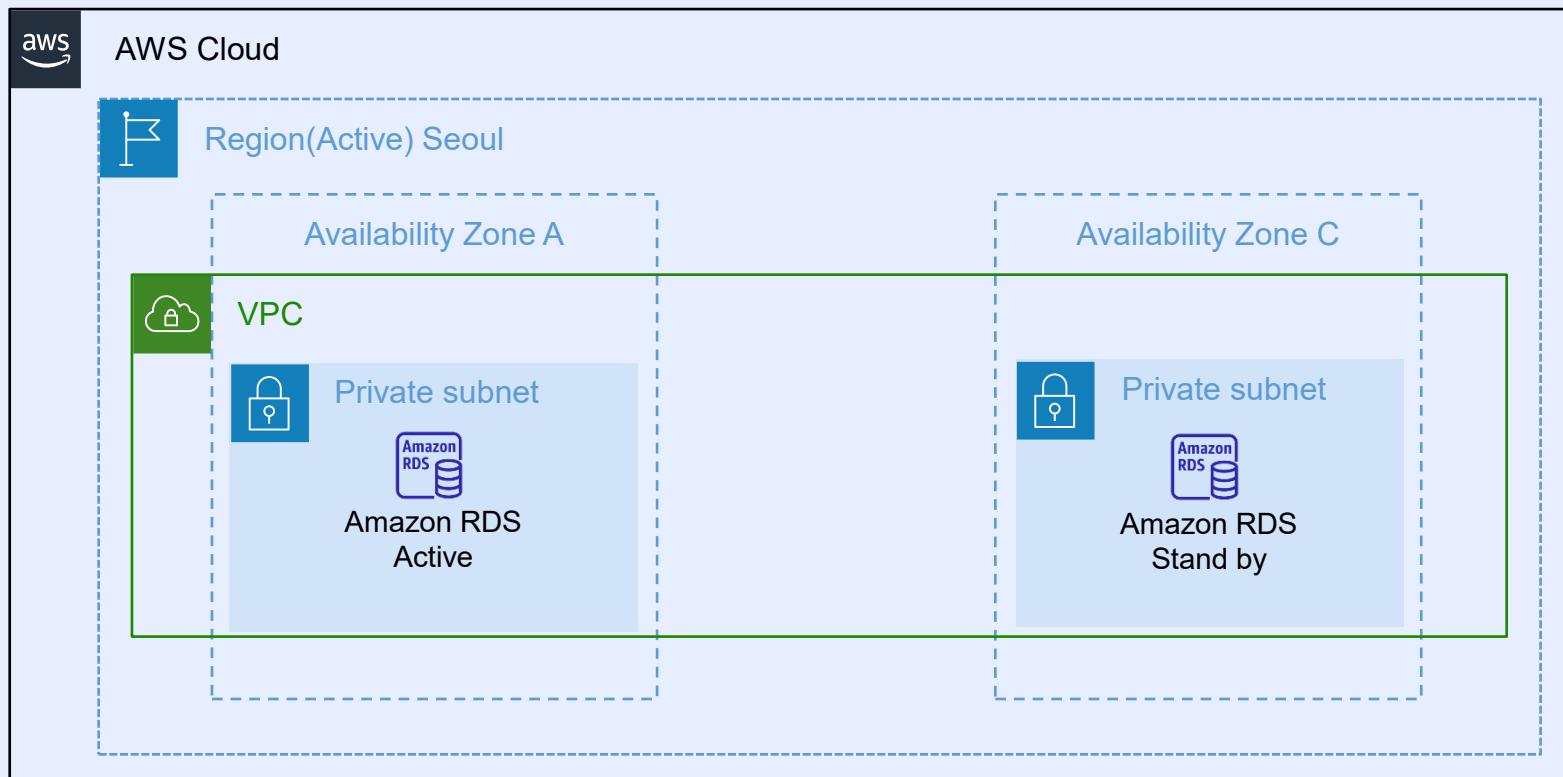


Amazon RDS (Relational Database Service)

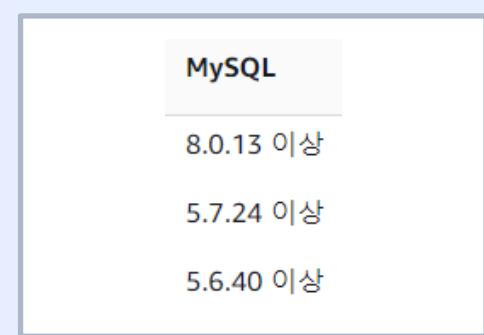
- AWS 의 관계형 데이터베이스 서비스
- 간편하게 설정, 운영 및 확장 할 수 있다.
- H/W 프로비저닝, 설정, 패치 및 백업과 같은 시간 소모적인 관리 작업을 자동화가 가능하다.
- 사용자가 애플리케이션에 집중하여 빠른 성능, 고가용성, 보안 및 호환성을 제공할 수 있도록 지원해주는 서비스이다.

서버

Wordpress 구축 – RDS DB



MySQL 엔진



S3 내보내기 지원 버전

1. Multi-AZ가 지원되는 프로덕션으로 생성

2. 자동으로 Active와 Stand by 가 나누게 된다

3. S3 내보내기가 지원되는 버전 선택

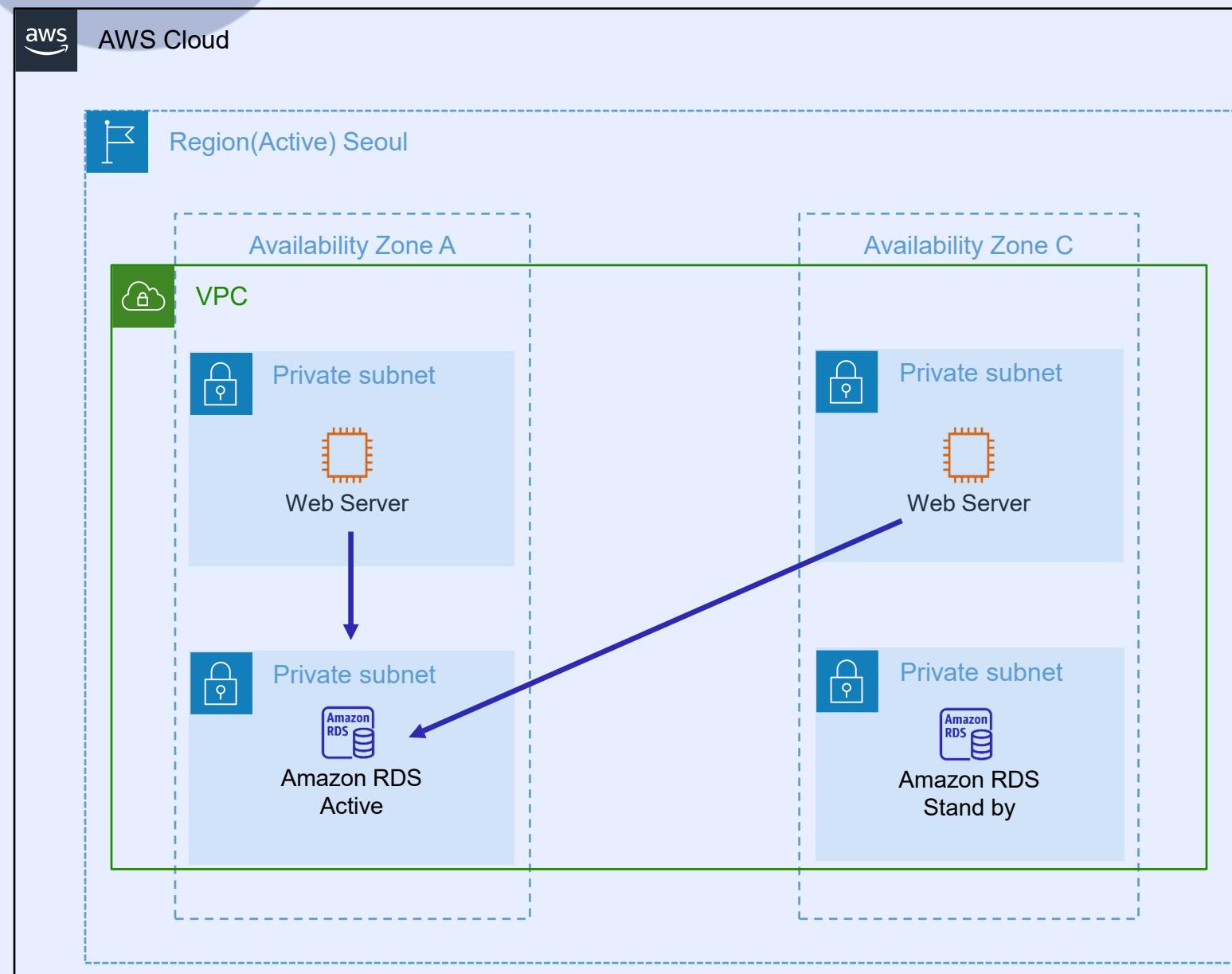
4. 상적으로 생성이 되면
Endpoint 주소를 확인 해주면 된다.

MySQL Community ap-northeast-2c db.t2.micro



서버

Wordpress 구축 – EC2 : Apache, MySQL



1. EC2 Instance에 Apache Web Server 를 설치해주고 http service Start, Enable
2. DB 연동을 위해 MySQL 설치 후 생성된 DB의 Endpoint 주소를 통해 접속

```
[root@ip-10-1-10-138 ~]# export MYSQL_HOST=wordpress.cmw6a4rr8t81.ap-northeast-2.rds.amazonaws.com
[root@ip-10-1-10-138 ~]# mysql --user=cbtadmin --password=asdqwe123 wordpress
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [wordpress]>
```

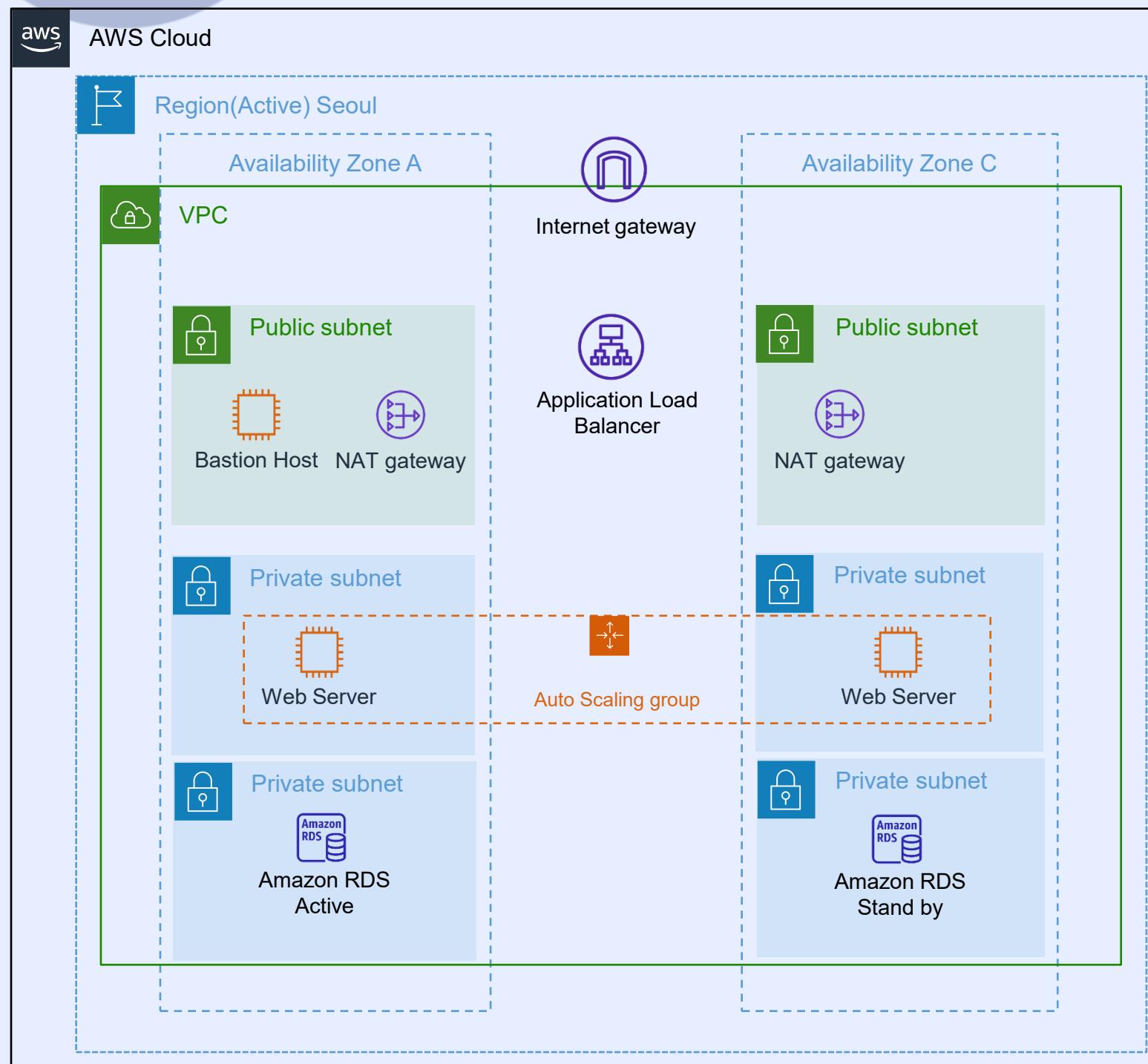
WordPress 데이터베이스 사용자를 만듭니다.
CREATE USER 'wordpress' IDENTIFIED BY 'asdqwe123';

WordPress 사용자에게 WordPress 데이터베이스에 대한
전체 액세스 권한을 부여합니다.
GRANT ALL PRIVILEGES ON wordpress.* TO wordpress;

MySQL을 새로고침하고 종료합니다.
FLUSH PRIVILEGES;
Exit

서버

Wordpress 구축 – EC2 : Wordpress



1. Wget 으로 Wordpress 를 다운로드, 압축해제
2. cd wordpress
3. wp-config-sample.php → wp-config.php
4. vi wp-config.php

```
define( 'DB_NAME', 'wordpress' );
/** Database username */
define( 'DB_USER', 'cbtadmin' );

/** Database password */
define( 'DB_PASSWORD', 'asdqwe123' );

/** Database hostname */
define( 'DB_HOST', 'wordpress.cmw6a4rr8t81.ap-northeast-2.rds.amazonaws.com' );

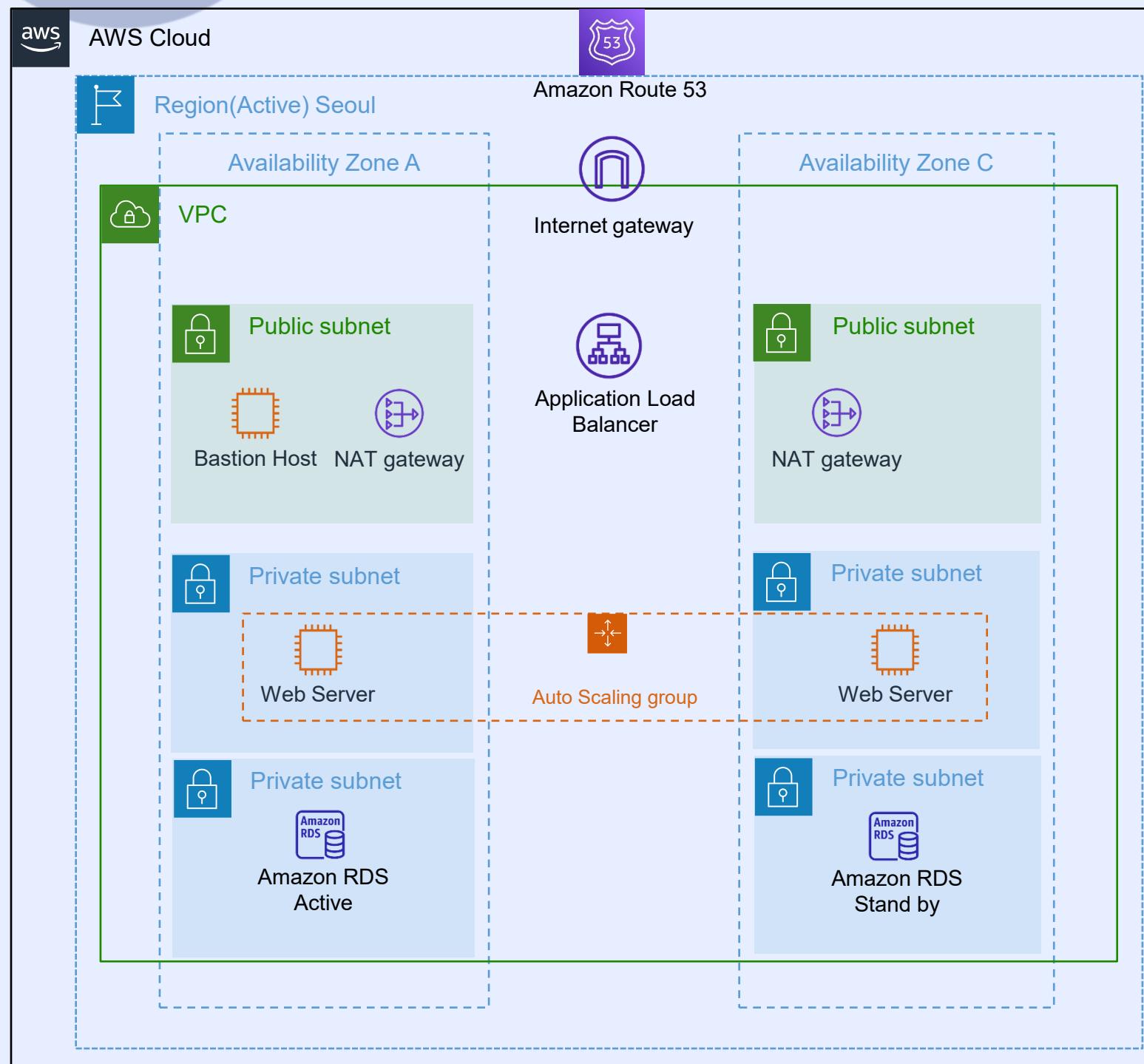
/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );
```

5. Apache 권한 주고 재 시작



서버

Wordpress 구축 – EC2 : Wordpress



이전에 ALB를 생성 했기에 Target Group에 Web Instance가
잡히는 것을 확인 한 후에 접속해준다.

DNS 이름 ALB-Seoul-600096074.ap-northeast-2.elb.amazonaws.com (A 레코드)

cbtcompany.xyz

Username or Email Address: cbtadmin

Password: [REDACTED]

Remember Me

Log In

워드프레스에 오신 것을

5.9.3 버전에 대해 더 알아보기.

알림판

블록과 패턴으로 풍부한 콘텐츠를 만드세요

블록 패턴은 사전 설정된 블록 레이아웃입니다. 영감을 얻기 위해 사용하거나 새 페이지를 바로 만드세요.

서버

Wordpress 구축 – 웹 페이지 세팅

CBT Global korea

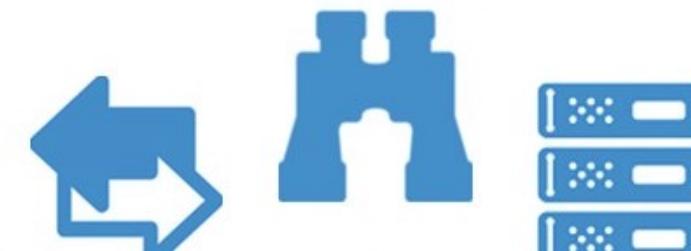
service solution partener client about



Network infra Service

24x7 Monitoring

Standardization

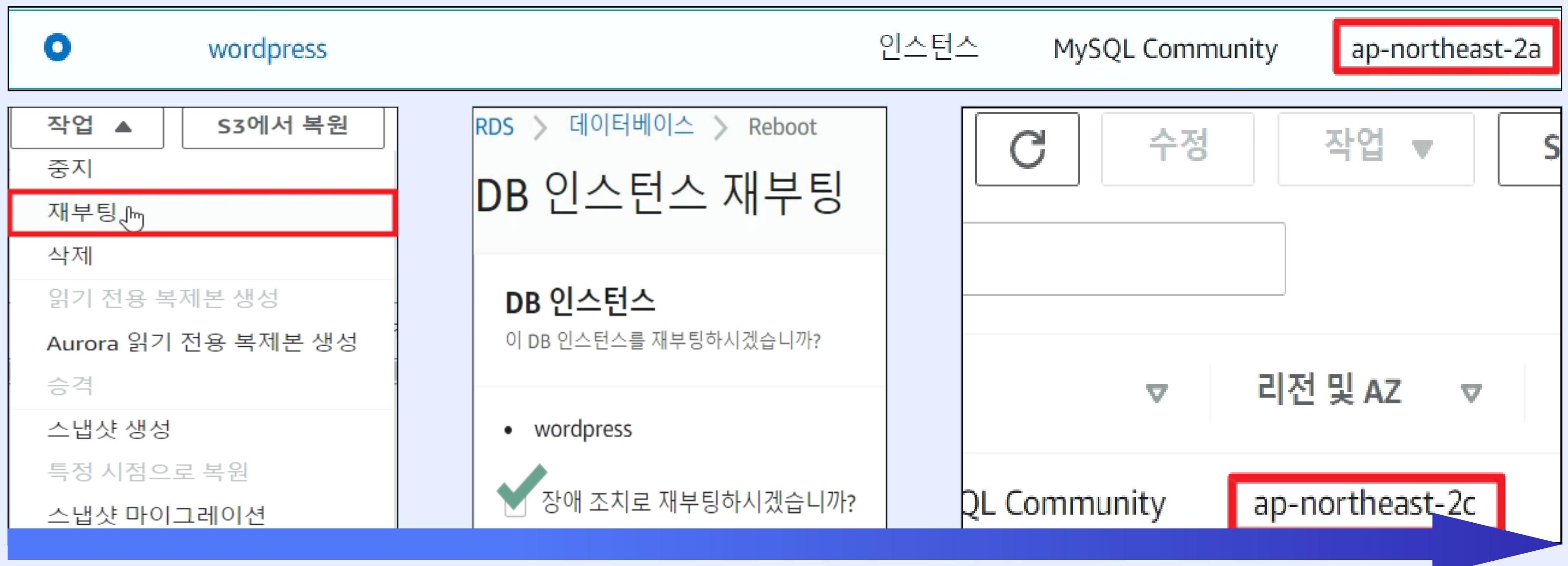


Backup & Disaster Recovery

서버

Multi-AZ의 Failover 매커니즘

장애, Backup 등의 이벤트가 발생하면 RDS의 Failover 매커니즘에 의해 Standby가 Active로 올라간다.



서버

RDS S3 snapshot export



고객 관리형 키 (1)	
<input type="text"/> 속성 또는 태그로 키 필터링	
별칭	키 ID
<input type="checkbox"/> s3bucket-kms	mrk- 90ad9f6fcf404abeae21d133bcf 91f2d

Amazon RDS (Relational Database Service) S3 snapshot export

- 2020년도에 출시된 새로운 RDS의 기능
- 백그라운드로 동작하기 때문에 DB클러스터에 부하를 주지 않음
- S3 버킷에 저장함으로써 백업과 복구에 편리
- Amazon S3 트리거를 사용하여 Lambda 함수 호출 할 수 있음
- 내보내기 전에 KMS 암호화 키가 필요하다.

서버

RDS S3 snapshot export

내보내기 선택



옵션 선택

- 1 내보내기 식별자**
내보내기를 식별할 이름을 입력합니다. 이름은 현재 AWS 지역에서 해당 AWS 계정이 소유한 모든 스냅샷입니다.
- 2 S3 버킷**
cbt-seoul-db-bucket
- 3 IAM 역할**
S3버킷에 쓰기 액세스 권한을 부여할 IAM 역할을 선택하거나 생성합니다.

IAM 역할 이름
- 4 AWS KMS 키 정보**

용량에 따라 내보내는 시간이 다르다.

Amazon S3의 내보내기 (1)	
<input type="button" value="Amazon S3의 내보내기 필터링"/> ▼ 상태	
<input type="checkbox"/>	이름
<input type="checkbox"/>	wordpress-0511 ... 시작 중

02

네트워크

다중 지역(국가) 간 네트워크 연결과 호스팅 서비스

- 김지환



TEAM CBT

재해 대비 멀티 리전 구성

네트워크

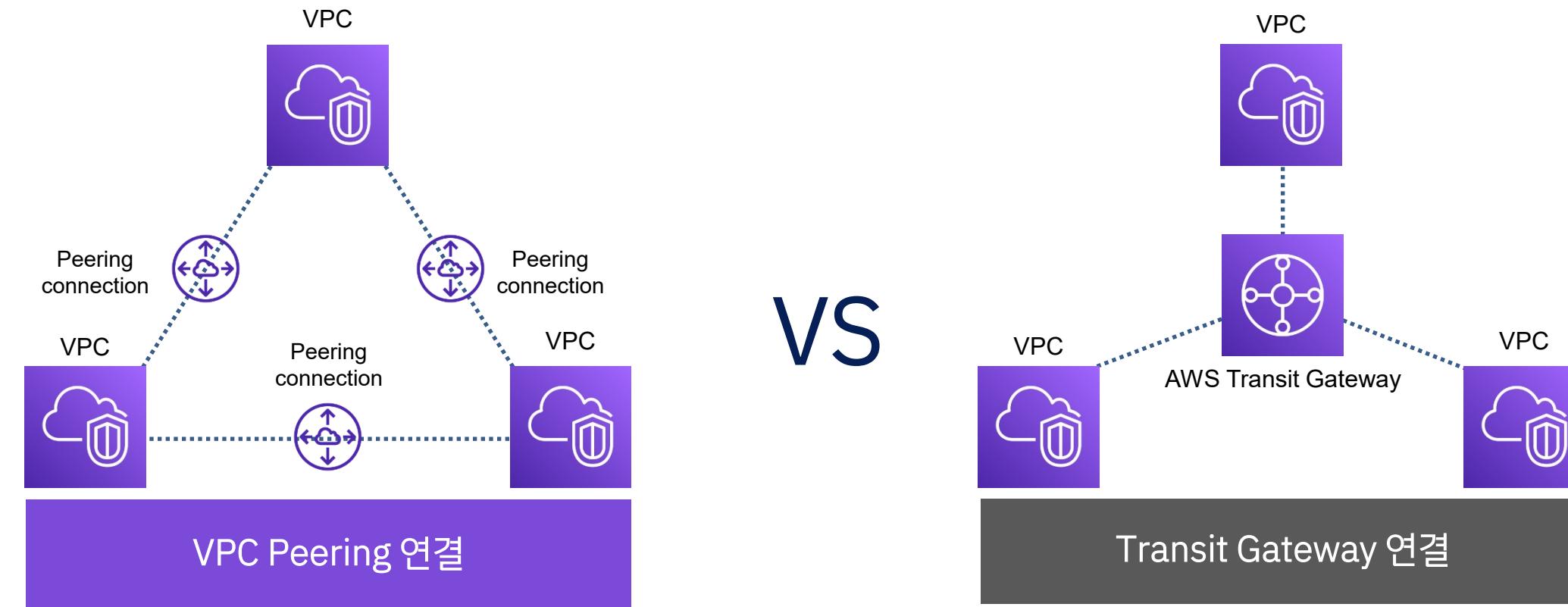
김지환

VPC Peering
Global Accelerator
Route53

리전 간 프라이빗한 통신을 원한다면,

VPC Peering 을 사용합니다.

VPC Peering VS Transit Gateway



- 비공개로 다른 VPC와 통신을 할 수 없을까?
- VPC Peering과 Transit Gateway는 VPC를 연결시 구조적 차이

VPC Peering VS Transit Gateway

VPC Peering

대역폭 : 제한 없음

최대 구성 가능 : VPC 당 125개

Transit Gateway

50 Gbps

VS

최대 구성 가능 : Transit Gateway 당 VPC 연결 5000개

네트워크

VPC Peering VS Transit Gateway

VPC Peering

VPC 연결비용

제한 없음

VPC 데이터 전송 비용

GB per \$0.02
(송신:\$0.01 / 수신:\$ 0.01)

예상 금액

총 \$20

Transit Gateway

Hour per \$0.07(서울리전)

GB per \$0.02

총 \$171.2

VS

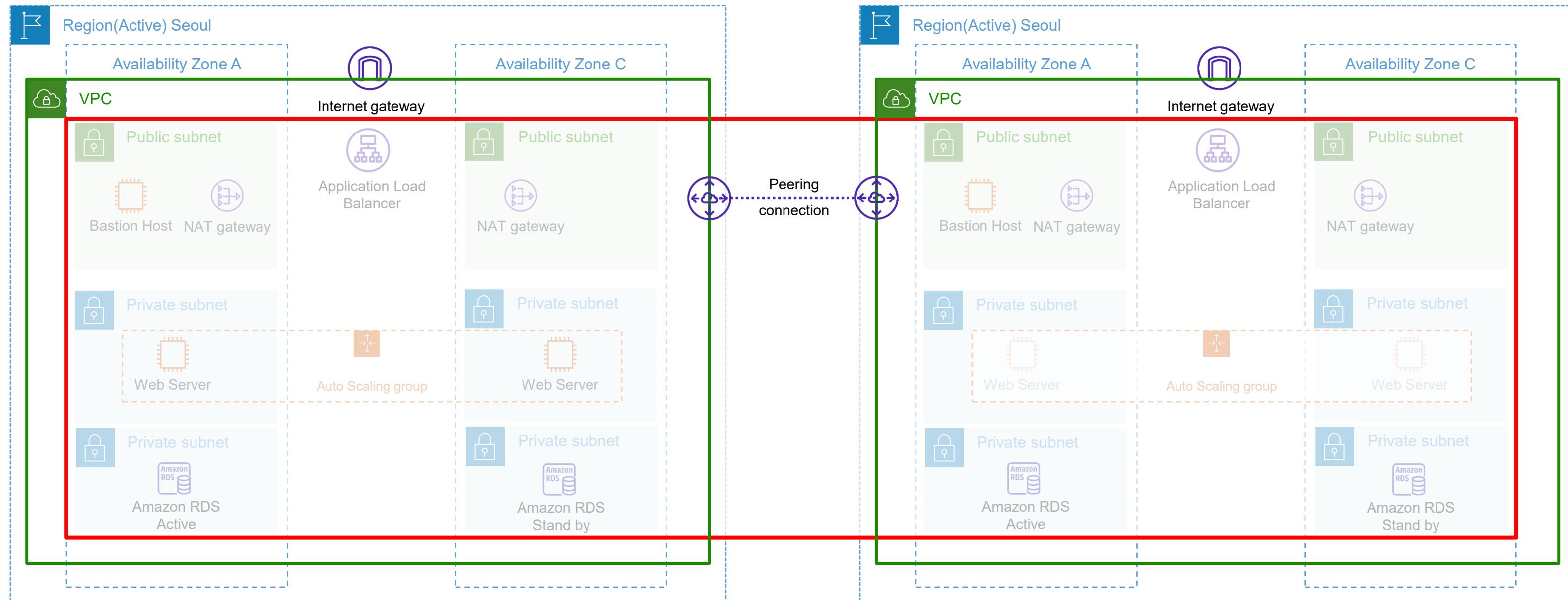
한 달의 사용 기간 동안 VPC 3개를 연결하여 VPC 1TB (1000GB) 의 데이터를 전송 할 때 총 예상 비용

네트워크

VPC Peering

VPC Peering이란?

서로 다른 VPC간 비공개로 트래픽을 라우팅 할 수 있도록 하는 네트워크 연결

VPC Peering 왜 필요할까요? 서비스 구분이나 관리 목적에 따라 VPC를 구분하지만 부득이하게 VPC간에 통신이 필요할 때 사용

네트워크

VPC Peering

피어링 연결 설정

이름 - 선택 사항
'Name' 키와 사용자가 지정하는 값을 포함하는 태그를 생성합니다.

Seoul-Tokyo Peering

피어링할 로컬 VPC 선택
VPC ID(요청자)

vpc-08b90c82777380b09 (SeoulCBT-vpc)

vpc-08b90c82777380b09 (SeoulCBT-vpc)용 VPC CIDR

CIDR	상태	상태 사유
10.1.0.0/16	Associated	-

피어링할 다른 VPC 선택
계정

내 계정
 다른 계정

리전

현재 리전(ap-northeast-2)
 다른 리전

아시아 태평양 (도쿄) (ap-northeast-1)

VPC ID(수락자)

vpc-09d1b93d4174cc563



리전

현재 리전(ap-northeast-2)
 다른 리전

아시아 태평양 (도쿄) (ap-northeast-1)

VPC ID(수락자)

vpc-09d1b93d4174cc563

1. VPC Peering 연결 요청

“가상 프라이빗 클라우드 > 피어링 연결”

2. VPC Peering 상대 리전에 대한 VPC ID 값 알아두기

네트워크

VPC Peering

VPC Peering 연결 수락 "가상 프라이빗 클라우드 → 피어링 연결 → 요청 수락" VPC Peering 연결이 활성화 !!!

The screenshot displays the AWS VPC Peering Connections interface. It features two main sections: 'Peering Requests' and 'Peering Connections'.

Peering Requests:

- Header: C 작업 ▲ 피어링 연결 생성
- Table columns: 상태, 요청자 VPC, 수락자 VPC, 요청
- Data row: 수락 대기 중, vpc-08b90c82777380b09, vpc-09d1b93d4174cc563 / CB..., 10.1
- Action buttons: 세부 정보 보기, 요청 수락 (highlighted with a red box), 요청 거부, DNS 설정 편집, ClassicLink 설정 편집

Peering Connections:

- Header: C 작업 ▼ 피어링 연결 생성
- Table columns: Name, 피어링 연결 ID, 상태
- Data row: Seoul-Tokyo P..., pcx-07d049395f4631e1c, 활성 (highlighted with a red box)
- Action buttons: C, 작업 ▼, 피어링 연결 생성

네트워크

VPC Peering

VPC Peering 연결 수락라우팅 테이블 설정

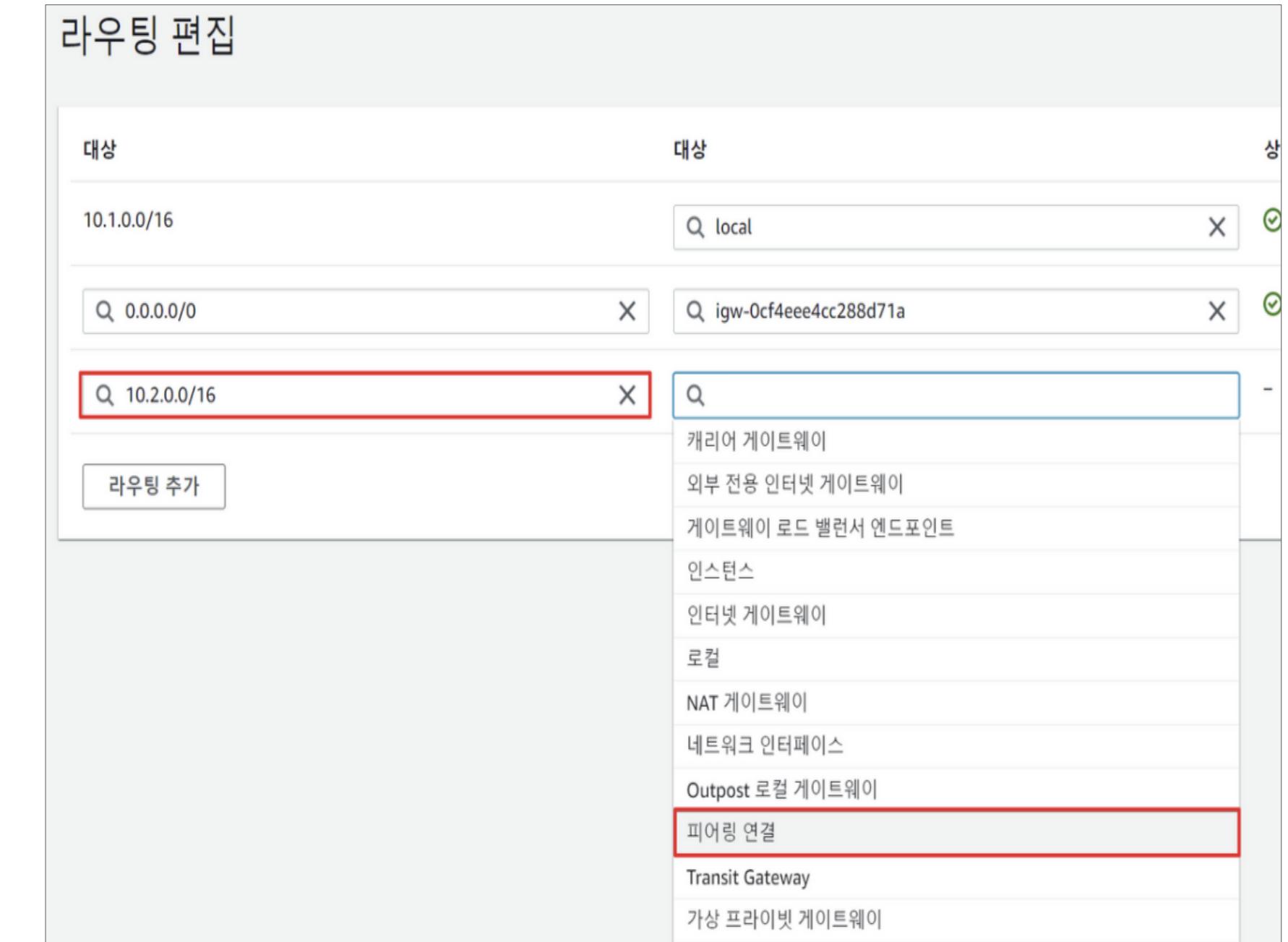
→ VPC 네트워크를 피어링 연결로 라우팅 하도록 설정

라우팅 테이블 설정시 고려사항

VPC Peering을 통하여 통신 :

프라이빗 서브넷끼리만?

퍼블릭, 프라이빗 서브넷 모두?



네트워크

VPC Peering

서울 리전 -> 도쿄 리전 VPC Peering 통신

```
[ec2-user@ip-10-1-1-101 ~]$ ping -c 5 10.2.1.219
PING 10.2.1.219 (10.2.1.219) 56(84) bytes of data.
64 bytes from 10.2.1.219: icmp_seq=1 ttl=64 time=33.5 ms
64 bytes from 10.2.1.219: icmp_seq=2 ttl=64 time=33.6 ms
64 bytes from 10.2.1.219: icmp_seq=3 ttl=64 time=33.4 ms
64 bytes from 10.2.1.219: icmp_seq=4 ttl=64 time=33.5 ms
64 bytes from 10.2.1.219: icmp_seq=5 ttl=64 time=33.5 ms
```

도쿄 리전 -> 서울 리전 VPC Peering 통신

```
[root@ip-10-2-4-168 ~]# ping -c 5 10.1.4.156
PING 10.1.4.156 (10.1.4.156) 56(84) bytes of data.
64 bytes from 10.1.4.156: icmp_seq=1 ttl=255 time=34.5 ms
64 bytes from 10.1.4.156: icmp_seq=2 ttl=255 time=34.5 ms
64 bytes from 10.1.4.156: icmp_seq=3 ttl=255 time=34.5 ms
64 bytes from 10.1.4.156: icmp_seq=4 ttl=255 time=34.6 ms
64 bytes from 10.1.4.156: icmp_seq=5 ttl=255 time=34.6 ms
```

“활성화된 VPC Peering 연결이 정상적으로 통신이 되는지 확인하고자 Ping TEST를 통해서 통신을 해보았더니,

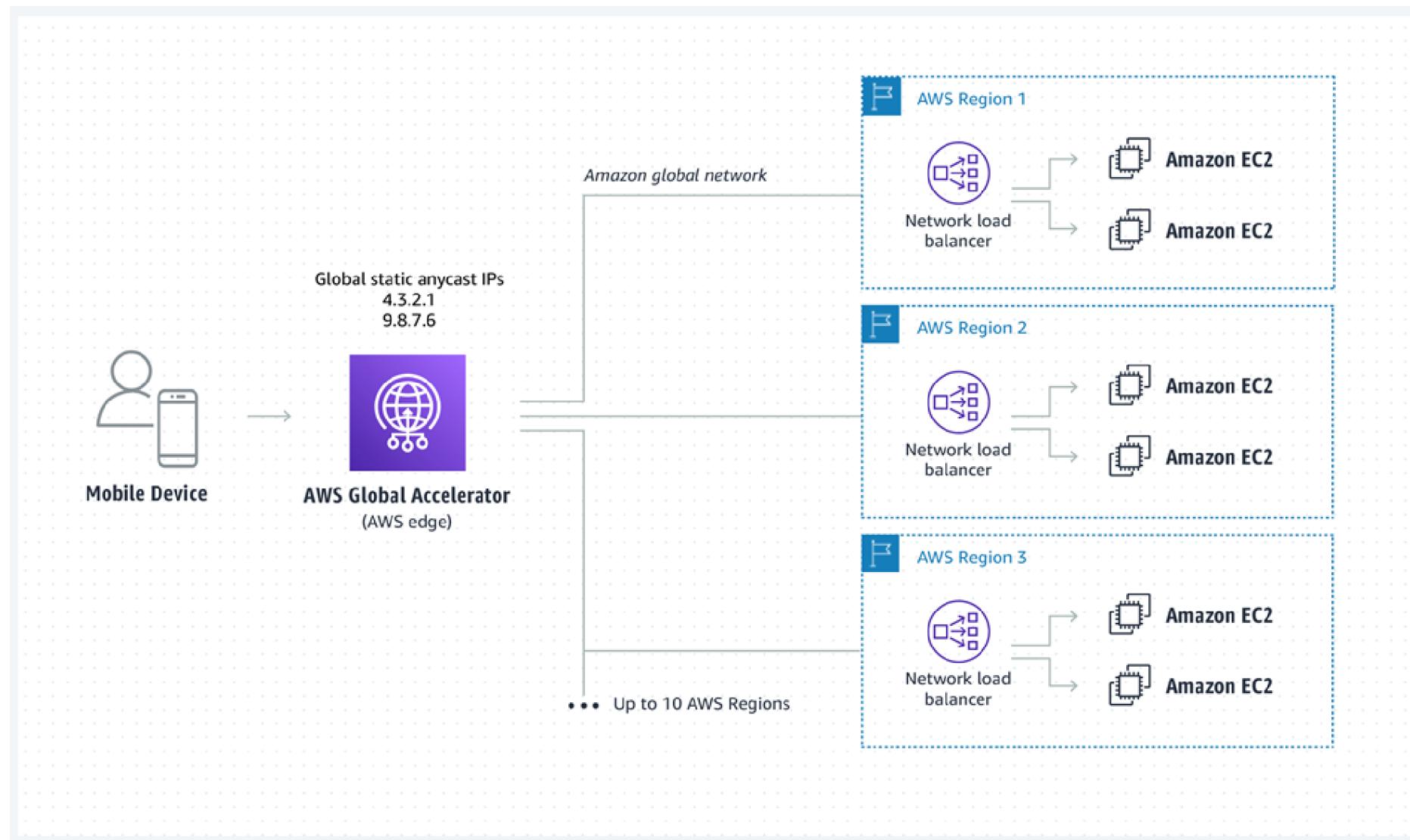
정상적으로 통신!”

좀 더 빨리.. 좀 더 높은 가용성을 원한다면 ,

Global Accelerator 를 사용합니다.

네트워크

Global Accelerator



Global Accelerator란?

AWS의 글로벌 네트워크 인프라를 사용
사용자 트래픽의 성능을 최대 60%
개선하는 네트워크 서비스

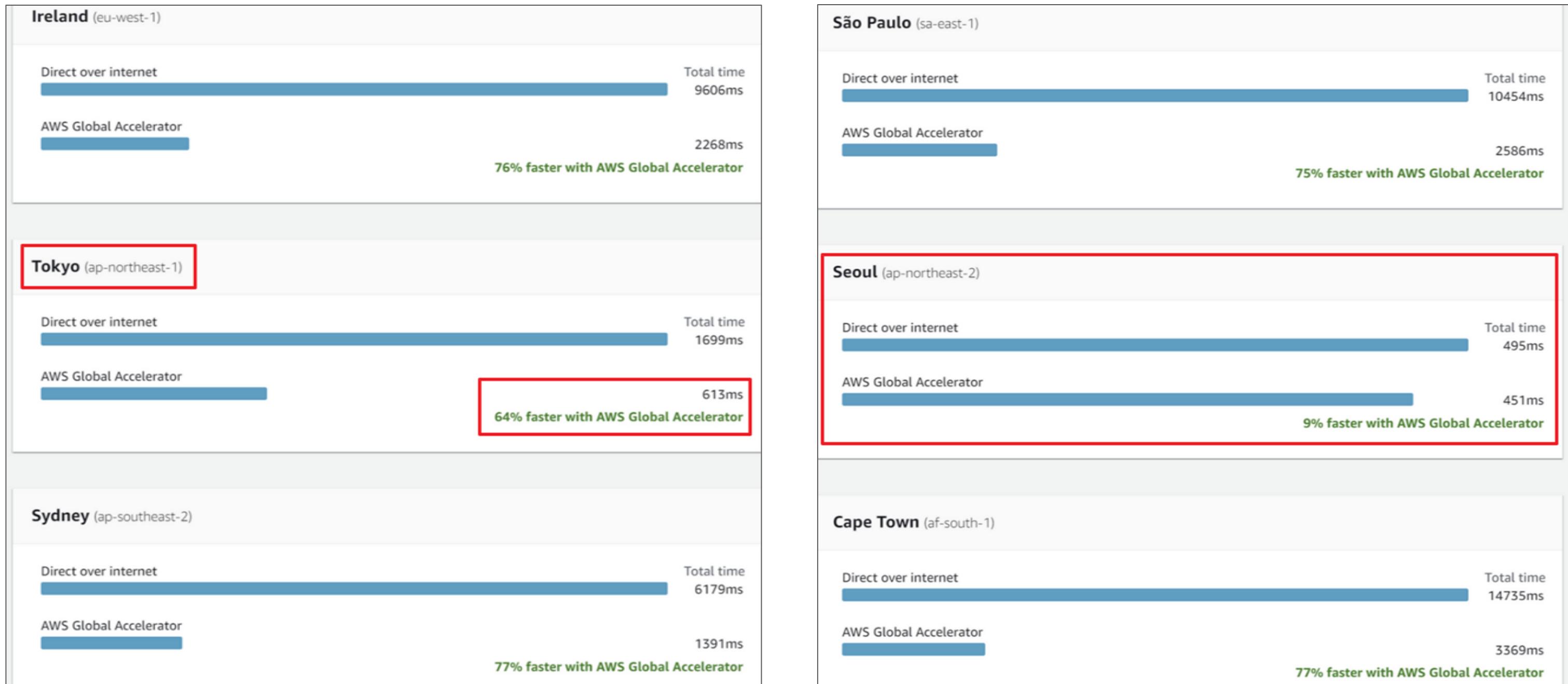
Global Accelerator 왜 필요할까요?

인터넷이 혼잡한 경우
AWS Global Accelerator는 경로를
최적화하여, 글로벌 애플리케이션의
가용성 및 성능을 개선

네트워크

Global Accelerator

Global Accelerator, 공용 인터넷 다운로드 속도 비교



네트워크

Global Accelerator

AWS Global Accelerator

Networking & Content Delivery

Improve availability and performance for your applications

AWS Global Accelerator is a service that improves the availability and performance of your applications with local or global users. It provides static IP addresses that act as a fixed entry point to your application endpoints in a single or multiple AWS Regions, and uses the AWS global network to optimize the path from your users to your applications.

Create accelerator

▼ Endpoint group: ap-northeast-2

Traffic dial: 100%

Endpoint type Info	Endpoint Info	Weight Info
Application Load Balancer	arn:aws:elasticloadbalancing:ap-nor...	128

A number from 0 to 255.

Preserve client IP address [Info](#)

Global Accelerator preserves the client IP address for internet-facing Application Load Balancers unless you clear the check box to disable the feature. All internal Application Load Balancers and EC2 instances automatically preserve the client IP address. Make sure that your endpoints are configured to accept traffic from the preserved client IP addresses.

Preserve client IP address

Add endpoints

Now that you've added one or more endpoint groups, you can add endpoints to each one. If you don't have any endpoints yet, create one on the Elastic Load Balancing (ELB) console or create an EC2 instance. Endpoints must be in the same Region as the endpoint group. You must have the required permissions to view available endpoint resources.

Listener: 80 TCP

Global Accelerator routes traffic that arrives on these ports to endpoints in regional endpoint groups. All endpoints for an endpoint group must be in the same Region.

▼ Endpoint group: ap-northeast-2

Traffic dial: 100%

Endpoint type Info	Endpoint Info	Weight Info
Application Load Balancer	arn:aws:elasticloadbalancing:ap-nor...	128

A number from 0 to 255.

Preserve client IP address

Add endpoint

▼ Endpoint group: ap-northeast-1

Traffic dial: 100%

Endpoint type Info	Endpoint Info	Weight Info
Application Load Balancer	arn:aws:elasticloadbalancing:ap-nor...	128

A number from 0 to 255.

Preserve client IP address [Info](#)

Global Accelerator preserves the client IP address for internet-facing Application Load Balancers unless you clear the check box to disable the feature. All internal Application Load Balancers and EC2 instances automatically preserve the client IP address. Make sure that your endpoints are configured to accept traffic from the preserved client IP addresses.

Preserve client IP address

Add endpoint

네트워크

Global Accelerator

CBTGlobal

You can preserve the client IP address for an Application Load Balancer or EC2 instance
Client IP address preservation allows your applications to use logic that is specific to a user's IP address. To use this feature, add a new Application Load Balancer or EC2 instance endpoint, or select the option for an existing Application Load Balancer endpoint. When you enable the feature for existing endpoints, we recommend that you gradually transition them into service. Some Regions don't support this feature. [Learn more](#)

Global Accelerator 생성!

CBTGlobal configuration

Name CBTGlobal	Name CBTGlobal
Static IP address set	Static IP address set
IP address 13.248.239.38 76.223.112.188	IP address 13.248.239.38 76.223.112.188

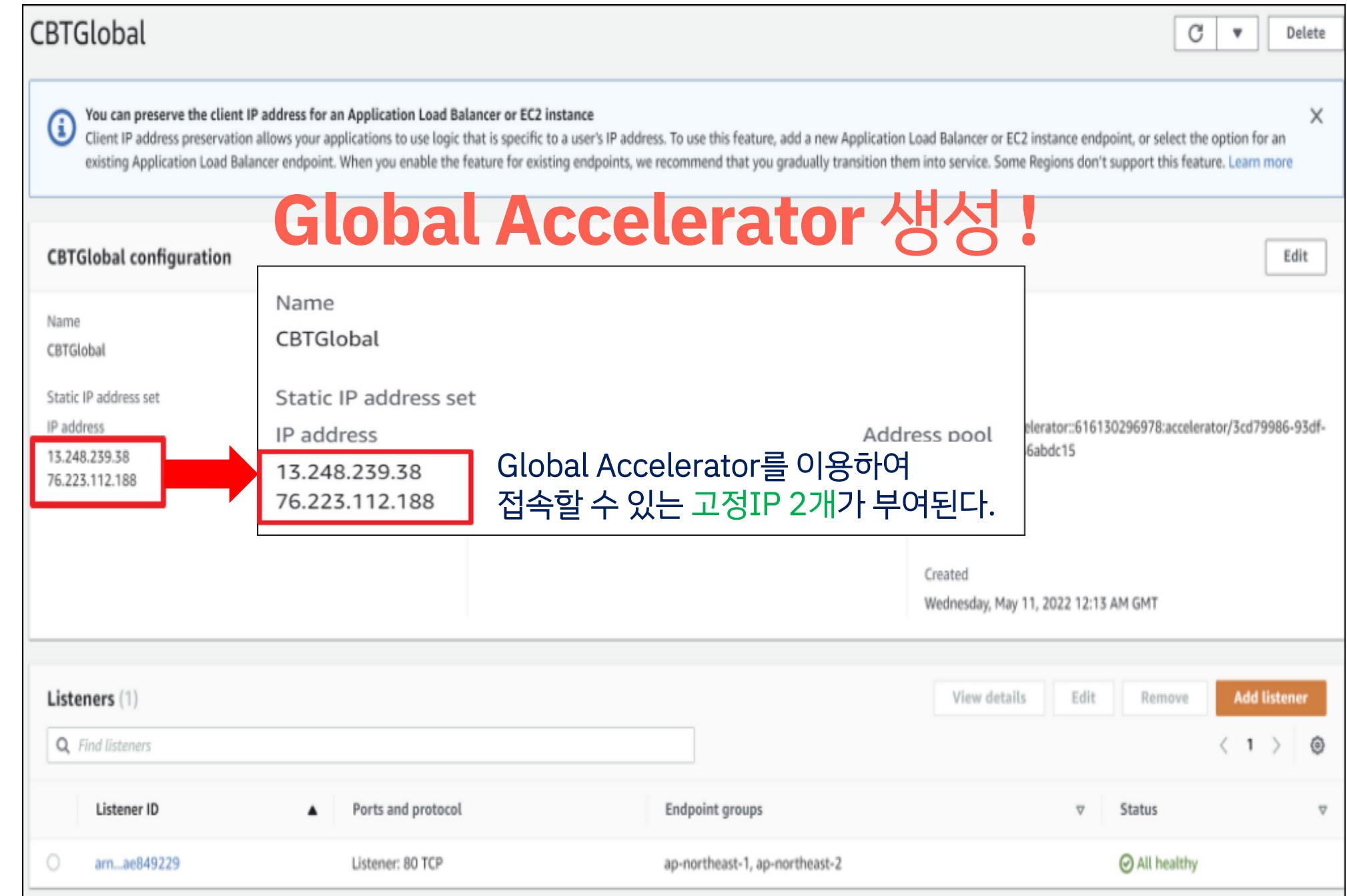
Address pool
elerator::616130296978:accelerator/3cd79986-93df-6abdc15

Created
Wednesday, May 11, 2022 12:13 AM GMT

Listeners (1)

Listener ID	Ports and protocol	Endpoint groups	Status
arn:aws:elasticloadbalancing:ap-northeast-1:849229	Listener: 80 TCP	ap-northeast-1, ap-northeast-2	All healthy

Global Accelerator를 이용하여 접속할 수 있는 고정IP 2개가 부여된다.



네트워크

Global Accelerator

The screenshot shows a web browser window with the URL 13.248.239.38 highlighted in red. The page content includes:

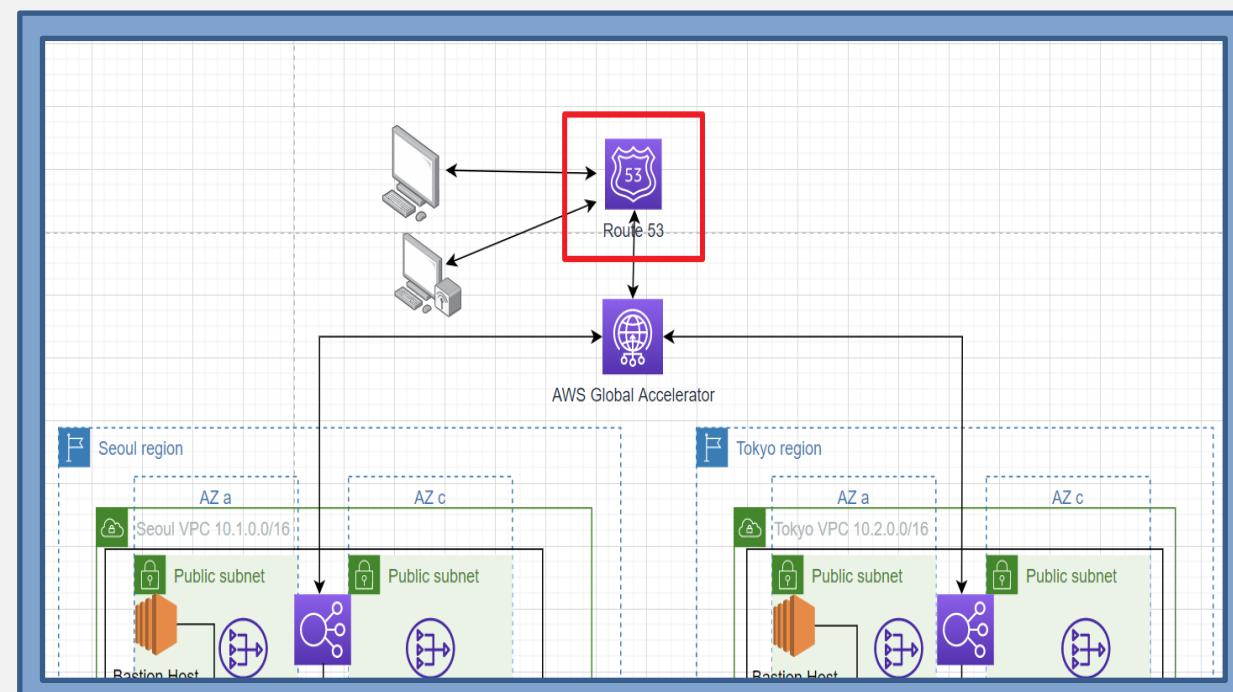
- CBT Global korea** (navigation menu: service, solution, partner, client, about)
- Team CBT 웹 페이지를 Global Accelerator를 통해 접속 !!** (Promotional message)
- 글로벌 네트워크를 선도하는 기업** (Text overlay on a world map background)
- 지속가능한 네트워크 인프라를 구축하며 서울 – 도쿄간 안정적인 네트워크 솔루션을 제공합니다.** (Text overlay on a world map background)
- Network infra Service** (Section title)
- 24x7 Monitoring** (Icon: binoculars)
- Standardization** (Icon: circular arrow)
- Remote Help Desk** (Icon: headset)
- Backup & Disaster Recovery** (Icon: server)
- Onsite Support** (Icon: person)

가용성과 확장성이 뛰어난 DNS 웹서비스,

Route53 을 사용합니다.

네트워크

Route53

**Route 53**

Route53 이란?

가용성과 확장성이 뛰어난 AWS의 DNS
웹 서비스

Route53 기능

도메인 등록
DNS 라우팅
상태 확인

네트워크

Route53

cbtcompany.xyz 정보

▶ 호스팅 영역 세부 정보

호스팅 영역 편집

레코드(2) DNSSEC 서명 호스팅 영역 태그(1)

레코드 (2) 정보

Automatic 모드는 최상의 필터 결과에 최적화된 현재 검색 동작입니다. 모드를 변경하려면 설정(settings)으로 이동합니다.

속성 또는 값을 기준으로 레코드 필터링

유형 라우팅 정책 별칭

레코드 이름 유형 라우팅 정책 차별화된 속성 값/트래픽 라우팅 대상

레코드 이름	유형	라우팅 정책	차별화된 속성	값/트래픽 라우팅 대상
cbtcompany... NS	NS	단순	-	ns-3.awsdns-00.com. ns-1162.awsdns-17.org. ns-1709.awsdns-21.co.uk. ns-591.awsdns-09.net.
cbtcompany... SOA	SOA	단순	-	ns-3.awsdns-00.com. awsdns-hostmaster.amazon.com. 17200 900 1209600 86400

호스팅 업체를 통해 해당 도메인을 추적
Route53에 등록

cbtcompany.xyz

▶ 네임서버명

ns-3.awsdns-00.com	IP: 205.251.192.3
ns-1162.awsdns-17.org	IP: 205.251.196.138
ns-1709.awsdns-21.co.uk	IP: 205.251.198.173
ns-591.awsdns-09.net	IP: 205.251.194.79

네임서버명 Ex) ns1.hosting.co.kr

호스팅 업체의 네임서버를
AWS에서 제공하는 네임서버로 변경

네트워크

Route53

라우팅 정책 선택 정보

라우팅 정책은 Amazon Route 53가 쿼리에 응답하는 방식을 결정합니다.

라우팅 정책

- 단순 라우팅** (선택됨): 모든 클라이언트가 동일한 응답을 수신하도록 하려면 사용합니다.
- 가중치 기반**: 동일한 작업을 수행하는 리소스가 있고 각 리소스로 향한 트래픽이 비율을 지정하려는 경우 사용합니다. 예: EC2 인스턴스 이상.
- 지연 시간**: 여러 AWS 리전에 리소스가 있고 가장 짧은 지연 시간을 제공하는 리전으로 트래픽을 라우팅하려는 경우 사용합니다.
- 장애 조치**: 리소스가 정상일 때 해당 리전의 트래픽을 라우팅하거나 첫 번째 리소스가 비정상일 때 다른 리전으로 트래픽을 라우팅하려는 경우입니다.

빠른 레코드 생성 정보

레코드 1

레코드 이름: co

레코드 유형: A - IPv4 주소 및 일부 AWS 리소스로 트래픽 라우팅

루트 도메인에 대한 레코드를 생성하려면 비워 둡니다.

트래픽 라우팅 대상 정보

별칭: Global Accelerator에 대한 별칭

미국 서부(오리온): ad82122c77728891.awsglobalaccelerator.com

라우팅 정책 정보

대상 상태 평가: 단순 라우팅

단순 라우팅으로 도메인 등록

CBT Global Korea

service solution partner client about

글로벌 네트워크를 선도하는 기업

지속 가능한 네트워크 인프라를 구축하며 서울 - 도쿄간 안정적인 네트워크 솔루션을 제공합니다.

Network infra Service

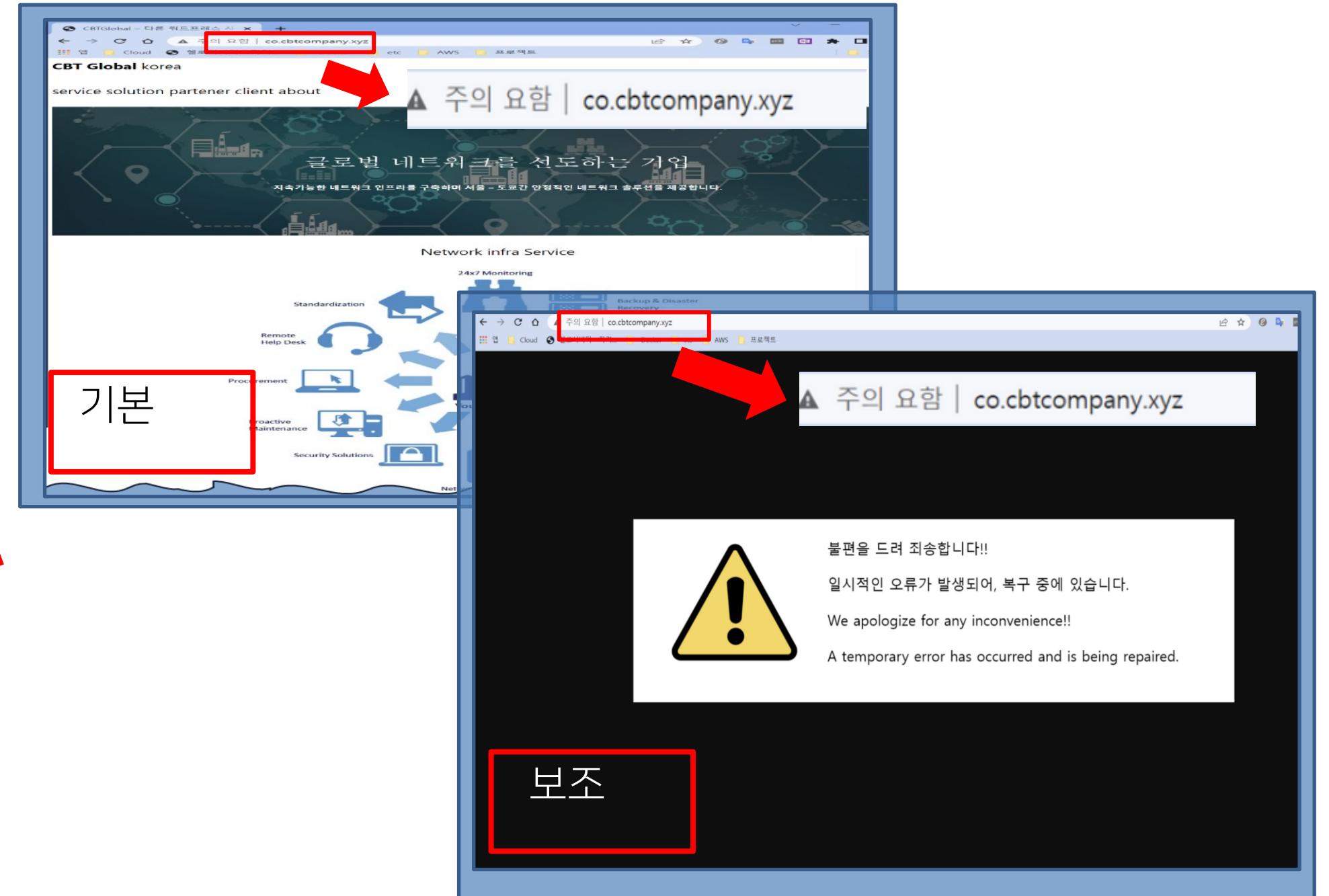
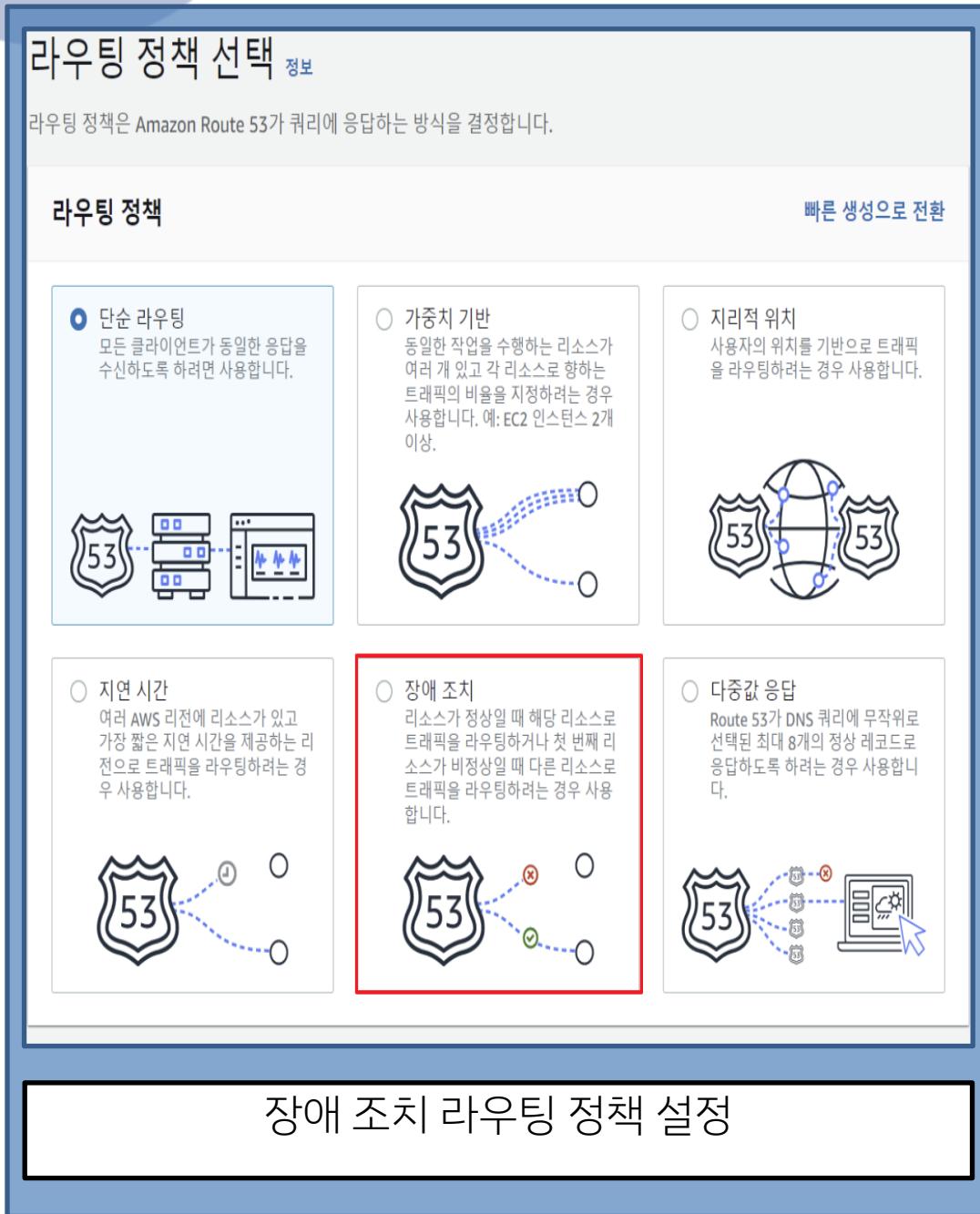
24x7 Monitoring, Standardization, Remote Help Desk, Procurement, Proactive Maintenance, Security Solutions, Network Optimization, Backup & Disaster Recovery, Onsite Support, Cloud Services, Vendor Management, Mobile Device Support.

Your Business

등록한 레코드 주소로 정상적으로 접속

네트워크

Route53



03

보안, 운영

웹서버의 취약점에 대비하고 증적 보관

- 안길환



TEAM CBT

재해 대비 멀티 리전 구성

보안, 운영

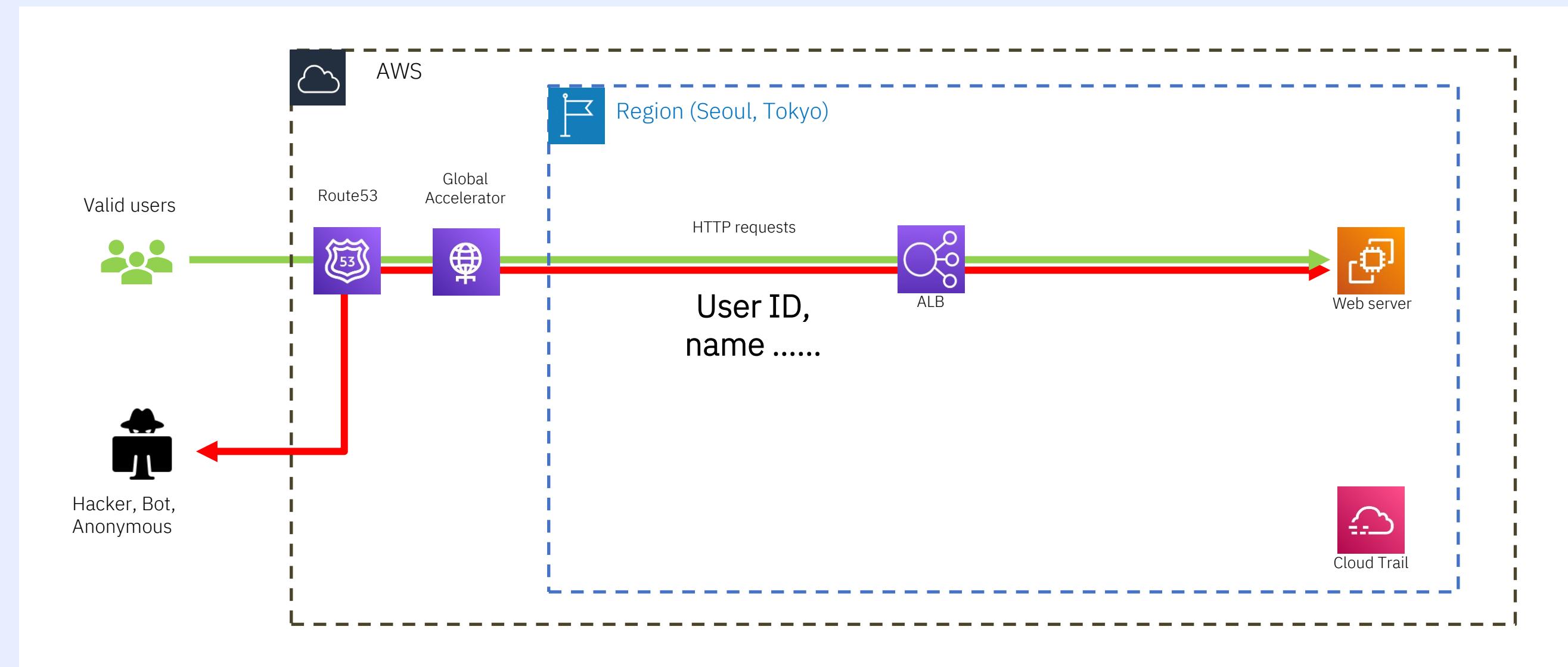
안길환

WAF, Cloudwatch, Kinesis Firehose

보안, 운영

웹 서비스 문제

모든 서비스가 웹과 앱으로 개발, 실수와 보안 문제



보안, 운영

웹 서비스 문제

취약점, 공격, 탈취, 그 다음은?

DVWA

Vulnerability: SQL Injection

User ID: ' OR 1=1 #

Vulnerability: SQL Injection

User ID:

ID: ' OR 1=1 #

First name: admin
Surname: admin

ID: ' OR 1=1 #
First name: Gordon
Surname: Brown

ID: ' OR 1=1 #
First name: Hack
Surname: Me

ID: ' OR 1=1 #
First name: Pablo
Surname: Picasso

ID: ' OR 1=1 #
First name: Bob
Surname: Smith

DVWA

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name? <script>alert(document.cookie)

<script>alert(document.cookie)</script>

More Information...

...ul-600096074.ap-northeast-2.elb.amazonaws.com 내용:

PHPSESSID=acluc73avjerrdrv3od1ld9i6; SLG_G_WPT_TO=ko;
SLG_GWPT_Show_Hide_tmp=1; SLG_wptGlobTipTmp=1; security=low;
AWSALB=KJM/s6BTBoL0yaX6KDbODDn0u5uYiOS00eiNwWcHaBD31nN/
Aa1GoBDIMOdK1l0Xlyp9Duxm/
qqiHAZbzK1t3DMY2t8ahEPuGYDzcQDdYRLIMNBNpmwxcztTShRW

Scripting (XSS)

확인

Hello

보안, 운영

웹 서비스 문제

무엇으로 막을 수 있을까?



Shield



WAF

Security, Identity, and Compliance

AWS WAF

Protect your web applications from common web exploits

AWS WAF is a web application firewall service that lets you monitor web requests that are forwarded to an Amazon API Gateway API, an Amazon CloudFront distribution, or an Application Load Balancer. You can protect those resources based on conditions that you specify, such as the IP addresses that the requests originate from.

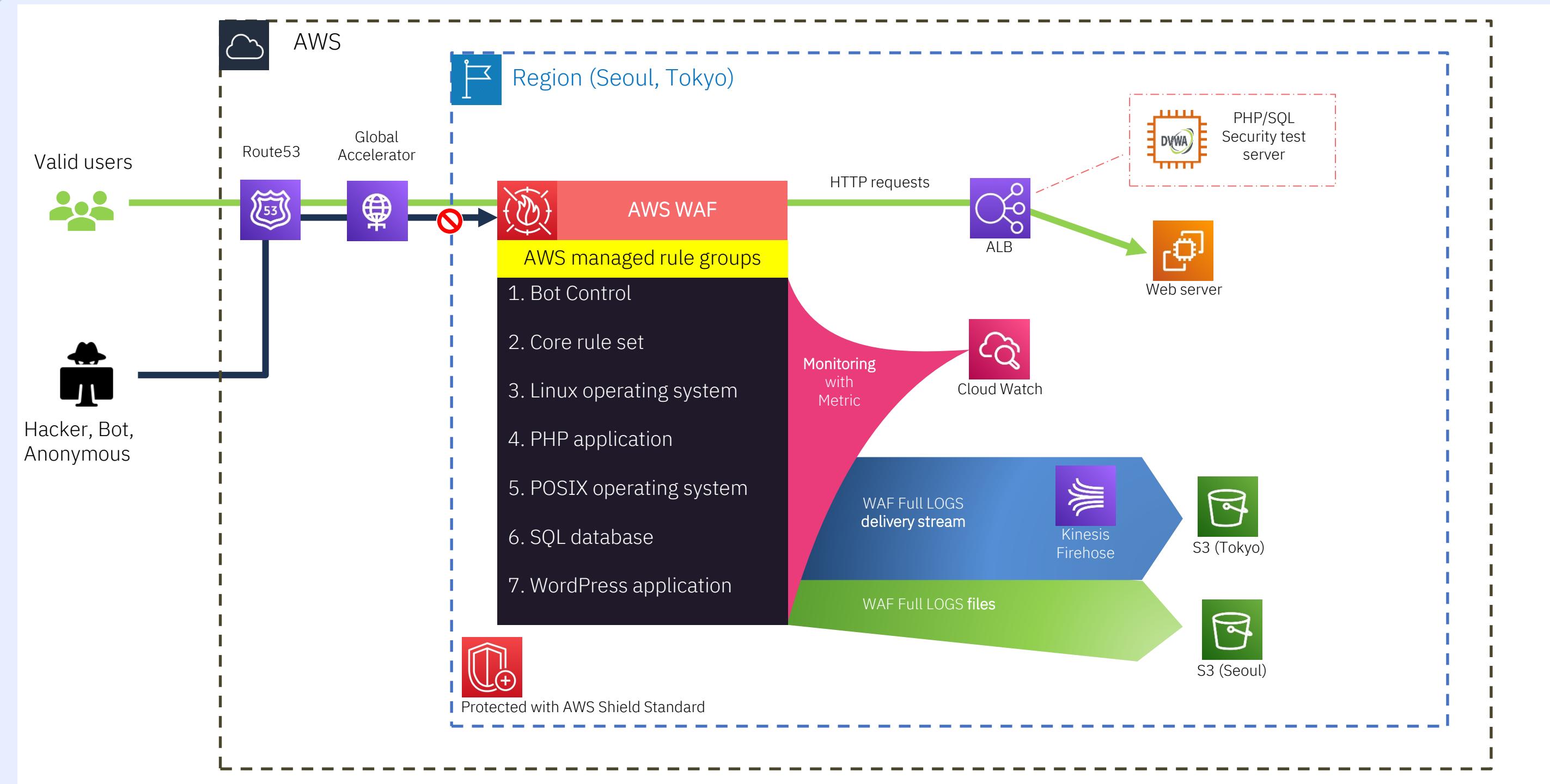
What's new

Feature	New AWS WAF	AWS WAF Classic
AWS managed rule groups	<input checked="" type="checkbox"/>	-
AWS Marketplace seller managed rule groups	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Number of rules per web ACL	Up to the web ACL capacity limit	10
Number of rule groups per web ACL	Up to the web ACL capacity limit	2

A web ACL has a capacity of 1,500. You can add hundreds of rules and rule groups to a web ACL. The total number that you can add is based on the complexity and capacity of each rule.

보안, 운영

보안 인프라 구성



보안, 운영

기대효과

방화벽 규칙

Add managed rule groups

Managed rule groups are created and maintained for you by AWS and AWS Marketplace sellers.

AWS managed rule groups

Paid rule groups

Name	Capacity	Action
Bot Control	50	<input checked="" type="radio"/> Add to web ACL Edit

Free rule groups

Name	Capacity	Action
Admin protection	100	<input type="radio"/> Add to web ACL
Amazon IP reputation list	25	<input type="radio"/> Add to web ACL
Anonymous IP list	50	<input type="radio"/> Add to web ACL
Core rule set	700	<input checked="" type="radio"/> Add to web ACL Edit
Known bad inputs	200	<input type="radio"/> Add to web ACL
Linux operating system	200	<input type="radio"/> Add to web ACL Edit
PHP application	100	<input type="radio"/> Add to web ACL Edit
POSIX operating system	100	<input type="radio"/> Add to web ACL Edit
SQL database	200	<input checked="" type="radio"/> Add to web ACL Edit
Windows operating system	200	<input type="radio"/> Add to web ACL
WordPress application	100	<input type="radio"/> Add to web ACL Edit

DVWA

Vulnerability: SQL Injection

User ID: ' OR 1=1 #

More Information

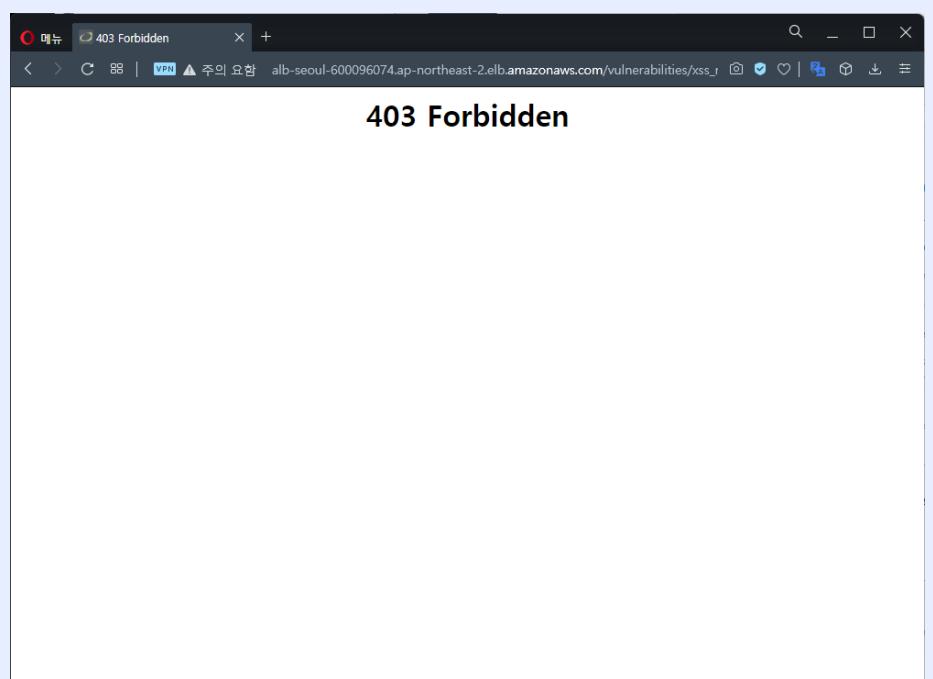
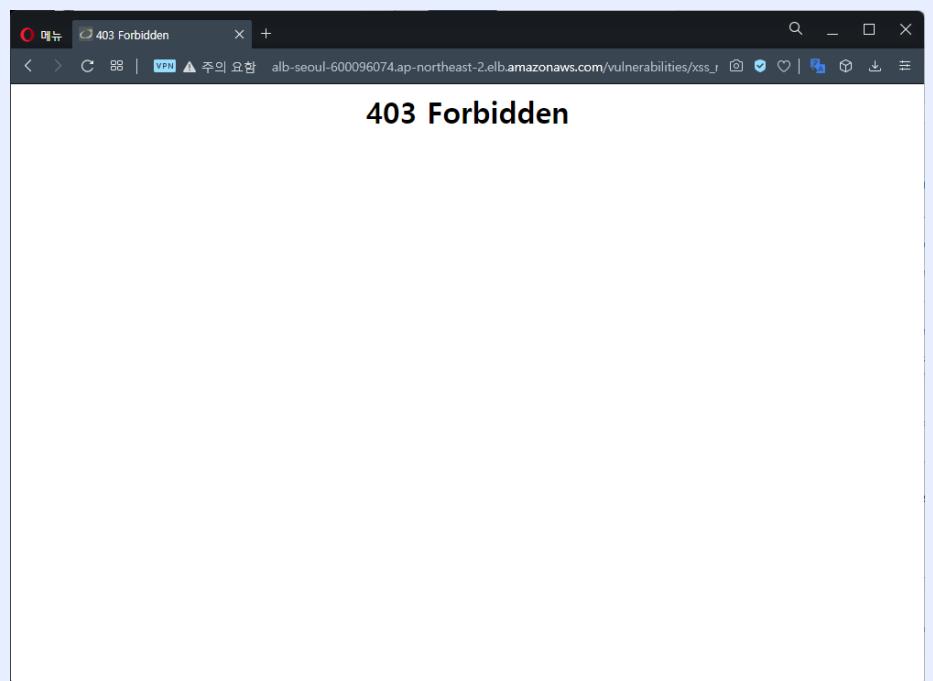
- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://terruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_Injection
- <http://bobby-tables.com/>

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name? <script>alert(document.cookie)</script>

More Information

- [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- <http://www.cgisecurity.com/xss-faq.html>
- <http://www.scriptalert1.com/>



보안, 운영

기대효과

WAF 탐지 결과

Sampled requests
Samples of requests from the past 3 hours.

Metric name	Source IP	URI	Rule inside rule group	Action
AWS-AWSManagedRulesBotControlRuleSet	184.105.139.68 (US)	/	AWS#AWSManagedRulesBotControlRuleSet#SignalNonBrowserUserAgent	BLOCK
AWS-AWSManagedRulesBotControlRuleSet	216.218.206.66 (US)	/	AWS#AWSManagedRulesBotControlRuleSet#SignalNonBrowserUserAgent	BLOCK
AWS-AWSManagedRulesBotControlRuleSet	135.125.246.110 (FR)	/env	AWS#AWSManagedRulesBotControlRuleSet#SignalKnownBotDataCenter	BLOCK
AWS-AWSManagedRulesBotControlRuleSet	134.122.118.31 (US)	/	AWS#AWSManagedRulesBotControlRuleSet#SignalKnownBotDataCenter	BLOCK
AWS-AWSManagedRulesBotControlRuleSet	135.125.246.110 (FR)	/	AWS#AWSManagedRulesBotControlRuleSet#SignalKnownBotDataCenter	BLOCK
AWS-AWSManagedRulesBotControlRuleSet	134.122.118.31 (US)	/	AWS#AWSManagedRulesBotControlRuleSet#SignalKnownBotDataCenter	BLOCK
AWS-AWSManagedRulesBotControlRuleSet	130.211.54.158 (BE)	/	AWS#AWSManagedRulesBotControlRuleSet#CategoryHttpLibrary	BLOCK
AWS-AWSManagedRulesBotControlRuleSet	104.152.52.110 (US)	/	AWS#AWSManagedRulesBotControlRuleSet#SignalNonBrowserUserAgent	BLOCK
AWS-AWSManagedRulesCommonRuleSet	77.111.247.99 (-)	/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28document.cookie%29%3C.../	AWS#AWSManagedRulesCommonRuleSet#CrossSiteScripting_QUERYARGUMENTS	BLOCK
AWS-AWSManagedRulesSQLIRuleSet	77.111.247.99 (-)	/vulnerabilities/sqli/?id=%27+OR+1%3D1+%23&Submit=Submit	AWS#AWSManagedRulesSQLIRuleSet#SQLI_QUERYARGUMENTS	BLOCK

보안, 운영

기대효과

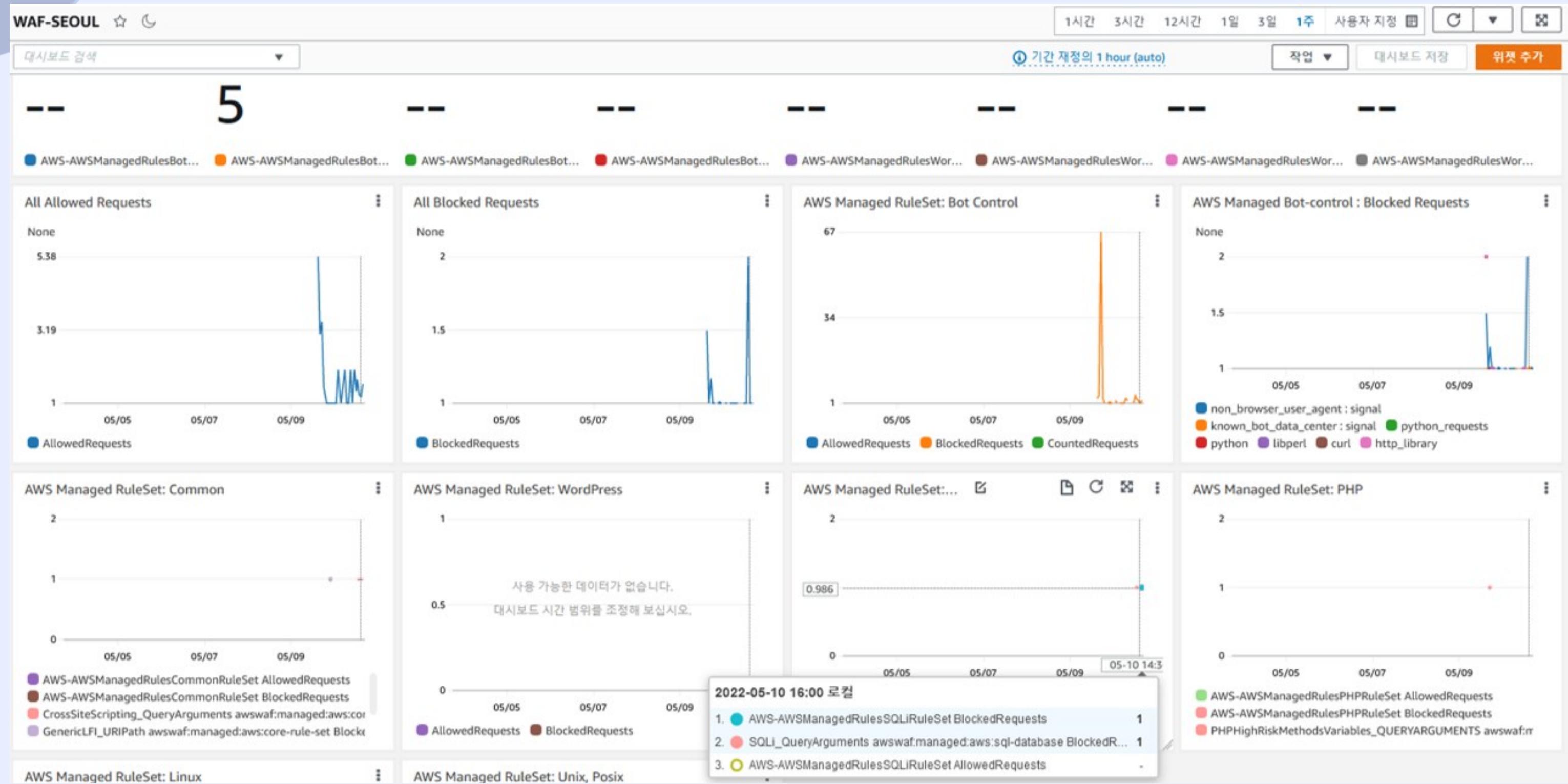
WAF 탐지 결과

Sampled request for metric AWS-AWSManagedRulesSQLiRuleSet				
Source IP	Rule inside rule group	Action	Time	
77.111.247.99	AWS#AWSManagedRulesSQLiRuleSet#SQLi_QUERYARGUMENTS	BLOCK	Tue May 10 2022 16:40:47 GMT+0900 (대한민국 표준시)	
Country	URI			
-	/vulnerabilities /sqli/?id=%27+OR+1%3D1+ %23&Submit=Submit			
Request				
<pre>GET /vulnerabilities/sqli/?id=%27+OR+1%3D1+&Submit=Submit Host: alb-seoul-600096074.ap-northeast-2.elb.amazonaws.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.84 Safari/537.36 OPR/85.0.4341.75 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9 Accept-Encoding: gzip, deflate Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7 Cookie: PHPSESSID=acluc73avjerrtdrv3od1ld9i6; SLG_G_WPT_TO=ko; SLG_GWPT_Show_Hide_tmp=1; SLG_wptGlobTipTmp=1; security=low; AWSALB=K6FSo371jrRpJ12XAsRBbcuV4CKwTxlscQr/eBG8V0BdY13txkSI2NJ5QEexWCxkR4 /ie5nivqqG4gwOzfzc5BvkhhNQmAgVfVb1jwoNj9mLyIWY2CQdhwPjtPsn Referer: http://alb-seoul-600096074.ap-northeast-2.elb.amazonaws.com/vulnerabilities/sqli/ Upgrade-Insecure-Requests: 1</pre>				

Sampled request for metric AWS-AWSManagedRulesCommonRuleSet				
Source IP	Rule inside rule group	Action	Time	
77.111.247.99	AWS#AWSManagedRulesCommonRuleSet#CrossSiteScripting_QUERYARGUMENTS	BLOCK	Tue May 10 2022 16:39:43 GMT+0900 (대한민국 표준시)	
Country	URI			
-	/vulnerabilities/xss_r /?name=%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E			
Request				
<pre>GET /vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E Host: alb-seoul-600096074.ap-northeast-2.elb.amazonaws.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.84 Safari/537.36 OPR/85.0.4341.75 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9 Accept-Encoding: gzip, deflate Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7 Cookie: PHPSESSID=acluc73avjerrtdrv3od1ld9i6; SLG_G_WPT_TO=ko; SLG_GWPT_Show_Hide_tmp=1; SLG_wptGlobTipTmp=1; security=low; AWSALB=QSFTP4Nb3T4KdbNxM6XleZb4pNHudouLhDeObfMTusshdstkXh2+r1QcMC8Tfv3Dv/mmhv+K2nHQpA /srP8JTkl8i735q/LRRoQfjOWQudrmwdqM7DAxmUj6Hc2 Referer: http://alb-seoul-600096074.ap-northeast-2.elb.amazonaws.com/vulnerabilities/xss_r/ Upgrade-Insecure-Requests: 1</pre>				

보안, 운영

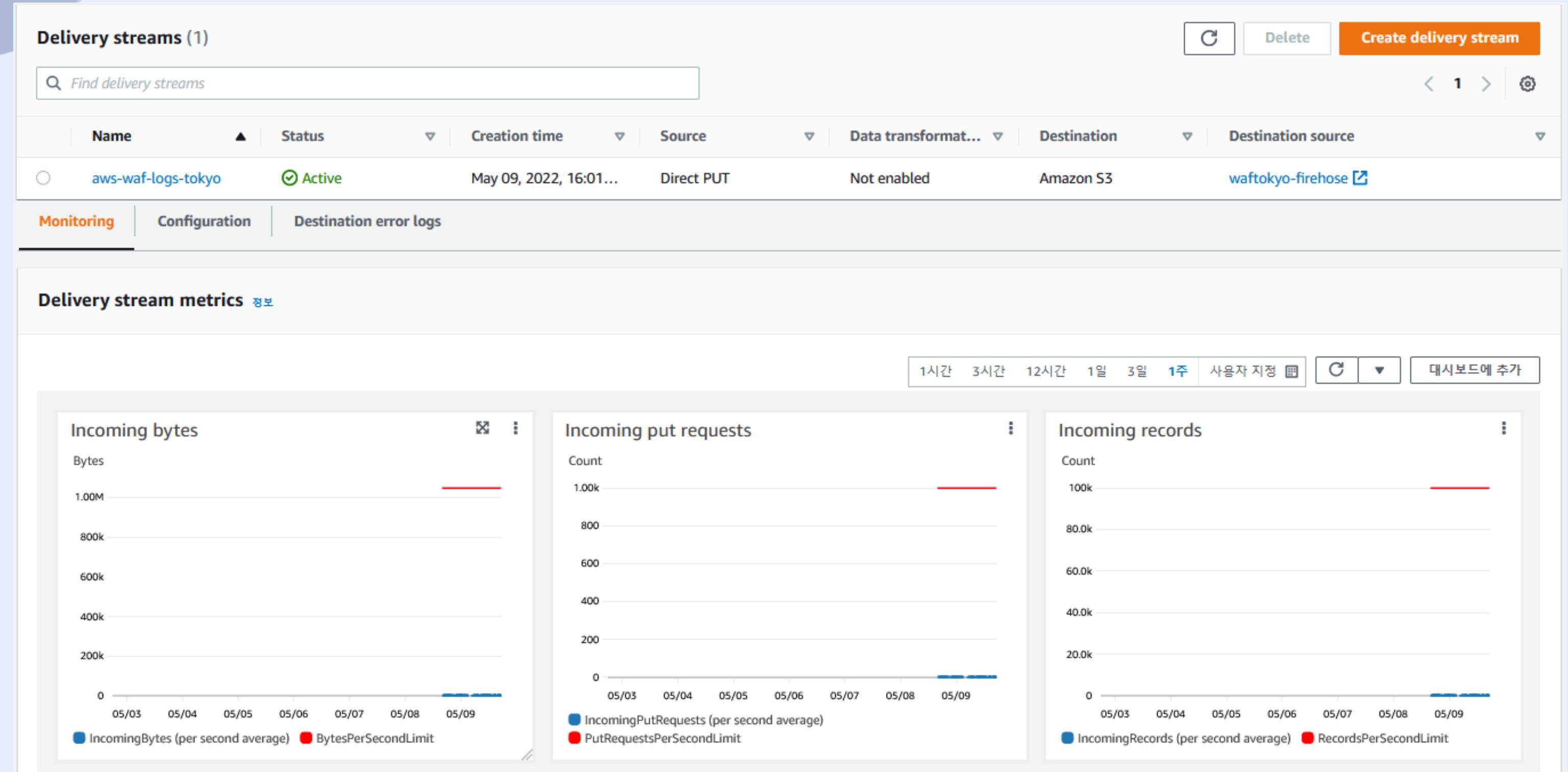
기대효과 모니터링 – cloudwatch Dashboard 연동



보안, 운영

기대효과

모니터링 – Kinesis Firehose 실시간 전송 중인 로그 백업 상태



보안, 운영

기대효과

로그의 저장 - 분석과 증적

버킷 (2) 정보

C ARN 복사 비어 있음 삭제 버킷 만들기

버킷은 S3에 저장되는 데이터의 컨테이너입니다. 자세히 알아보기

이름으로 버킷 찾기 < 1 >

이름	AWS 리전	액세스	생성 날짜
aws-waf-logs-seoulwaf	아시아 태평양(서울) ap-northeast-2	버킷 및 객체가 퍼블릭이 아님	2022. 5. 9. pm 3:31:48 PM KST
waftokyo-firehose	아시아 태평양(도쿄) ap-northeast-1	버킷 및 객체가 퍼블릭이 아님	2022. 5. 9. pm 3:40:41 PM KST

보안, 운영

기대효과

로그의 저장 - 분석과 증적

버킷 (2) 정보

버킷은 S3에 저장되는 데이터의 컨테이너입니다. 자세히 알아보기 [\[\]](#)

이름	AWS 리전	액세스	생성 날짜
aws-waf-logs-seoulwaf	아시아 태평양(서울) ap-northeast-2	버킷 및 객체가 퍼블릭이 아님	2022. 5. 9. pm 3:31:48 PM KST
waftokyo-firehose	아시아 태평양(도쿄) ap-northeast-1	버킷 및 객체가 퍼블릭이 아님	2022. 5. 9. pm 3:40:41 PM KST

이름

[\[\]](#) 035671322814_waflogs_ap-northeast-2_waf-alb-seoul_20220510T0805Z_accb0635.log.gz **gz**

이름

[\[\]](#) aws-waf-logs-tokyo-1-2022-05-10-08-00-22-ca9324e9-5e5a-4c41-9fa8-f146e7e288cc

[\[\]](#) aws-waf-logs-tokyo-1-2022-05-10-08-11-37-0db856a5-8d7e-498e-9fe5-da5b3676d0f5

보안, 운영

기대효과

로그의 저장 – Json 형식

```
D: > {} aws-waf-logs-tokyo-1-2022-05-10-08-11-37-0db856a5-8d7e-498e-9fe5-da5b3676d0f5 > ...
1  {
2    "timestamp": 1652170290960,
3    "formatVersion": 1,
4    "webaclId": "arn:aws:wafv2:ap-northeast-1:035671322814:regional/webacl/CBT-WAF-tokyo/dcf2a77b-1649-4a36-9643-73c18fb278ce",
5    "terminatingRuleId": "AWS-AWSManagedRulesBotControlRuleSet",
6    "terminatingRuleType": "MANAGED_RULE_GROUP",
7    "action": "BLOCK",
8    "terminatingRuleMatchDetails": [],
9    "httpSourceName": "ALB",
10   "httpSourceId": "035671322814-app/CBT-ALB-tokyo/348f8d560838d16b",
11   "ruleGroupList": [
12     {
13       "ruleGroupId": "AWS#AWSManagedRulesBotControlRuleSet",
14       "terminatingRule": {
15         "ruleId": "SignalNonBrowserUserAgent",
16         "action": "BLOCK",
17         "ruleMatchDetails": null
18       },
19       "nonTerminatingMatchingRules": [],
20       "excludedRules": null
21     }
22   ],
23   "rateBasedRuleList": [],
24   "nonTerminatingMatchingRules": [],
25   "requestHeadersInserted": null,
26   "responseCodeSent": null,
27   "httpRequest": {
28     "clientIp": "216.218.206.66",
29     "country": "US",
30     "headers": [
31       {
32         "name": "Host",
33         "value": "18.181.132.14"
34       }
35     ],
36     "uri": "/",
37     "args": "",
38     "httpVersion": "HTTP/1.1",
39     "httpMethod": "GET",
40     "requestId": "1-627a1e32-1298e27f6a2063515e3204e3"
41   },
42   "labels": [
43     {
44       "name": "awswaf:managed:aws:bot-control:signal:non_browser_user_agent"
45     }
46   ]
47 }
```

04

백업

재해대비 멀티 리전 **AWS** 백업

- 고찬희



TEAM CBT

재해 대비 멀티 리전 구성

백업

고찬희

백업 계획, 규칙, 볼트, 백업 복원

백업

AWS Backup 소개

TEAM CBT

재해 대비 멀티 리전 구성



AWS Backup

- 2019년 7월에 시작한 서비스
- 기존의 ec2 스냅샷, RDS 자동복제 같은 수동 백업을 확장한 중앙집중화
- 완전 자동화 서비스
- 최소비용은 없으며 백업스토리지 양과 복원한 데이터의 양에 대해서 비용청구

백업

AWS Backup 특성



AWS Backup

- Ec2, DB인스턴스, S3 단위의 백업구성
- 최초의 백업은 전체백업
이후의 백업은 추가된 데이터 만큼의 증분백업
- 백업할 리소스를 개별 혹은 단위로 묶어서 플랜 생성
- Storage Gateway를 통한 로컬 스토리지 하이브리드 백업가능
- 콜드 스토리지로 전환 후 수명주기 정책으로 90일 후 자동삭제가능
- 다른 리전, 계정간 교차백업
- VPC와 서브넷에 관계없이 해당리전의 구성요소를 모두 지정가능

백업

AWS Backup 특성

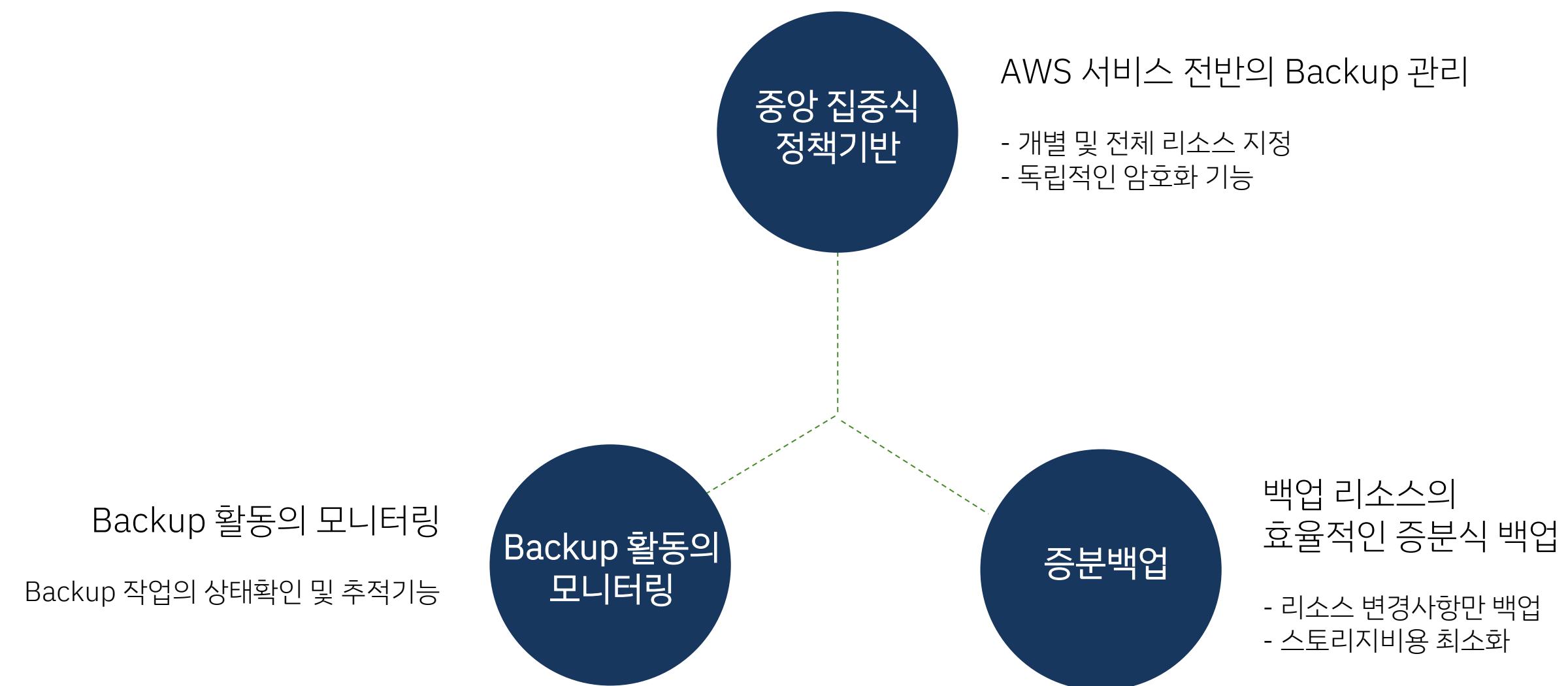


AWS Backup

- Ec2, DB인스턴스, S3 단위의 백업구성
- 최초의 백업은 전체백업
이후의 백업은 추가된 데이터 만큼의 증분백업
- 백업할 리소스를 개별 혹은 단위로 묶어서 플랜 생성
- Storage Gateway를 통한 로컬 스토리지 하이브리드 백업가능
- 콜드 스토리지로 전환 후 수명주기 정책으로 90일 후 자동삭제가능
- 다른 리전, 계정간 교차백업
- VPC와 서브넷에 관계없이 해당리전의 구성요소를 모두 지정가능

백업

AWS Backup 을 사용해야 하는 이유



AWS Backup 을 사용해야 하는 이유



Snapshot

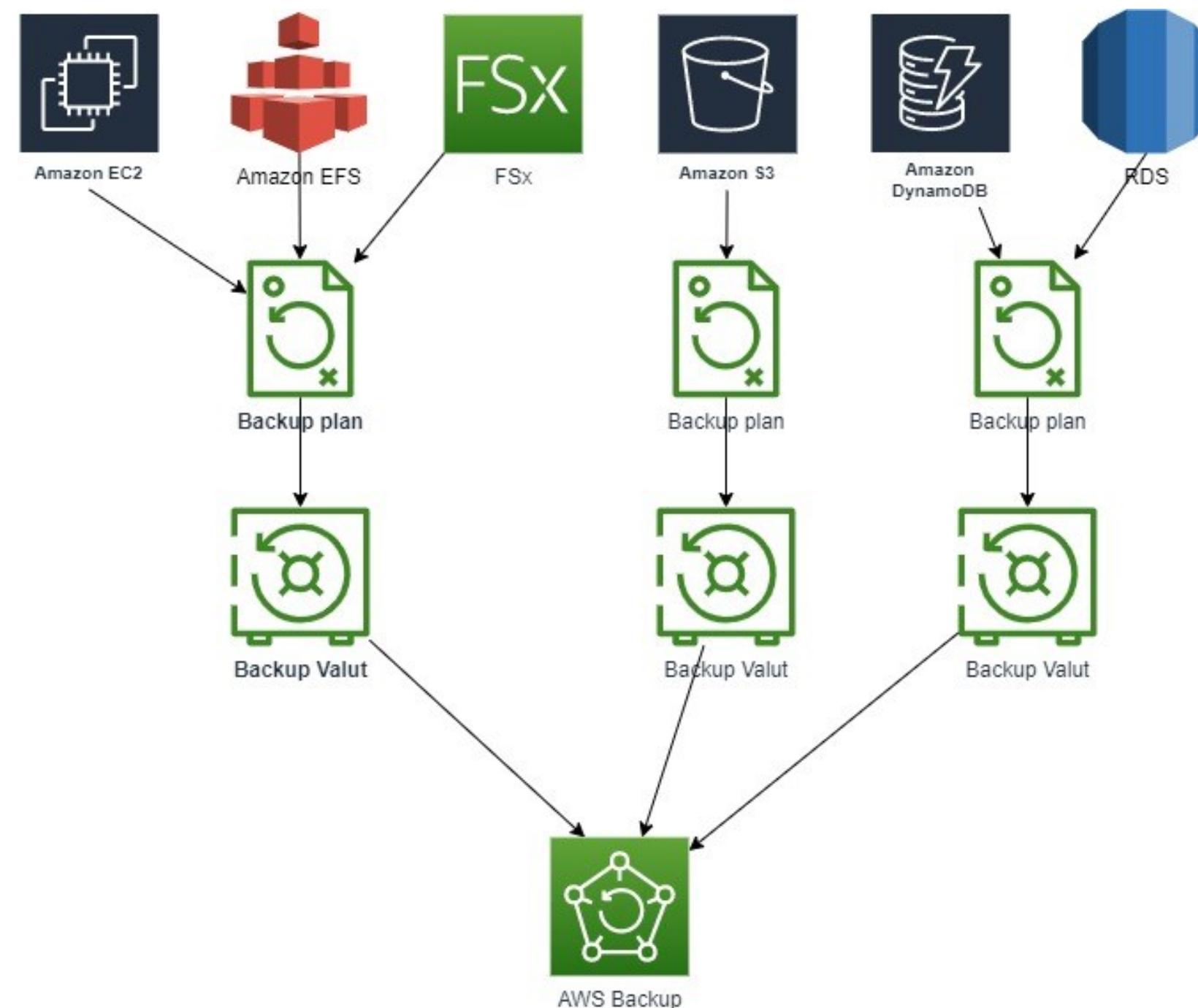


AWS Backup

- 디스크 오류시 디스크
- 오류지점까지 그대로 복제됨
- VPC 외 개별 데이터 독립 복사본을 생성하여 디스크 오류지점에 대한 독립성을 보장

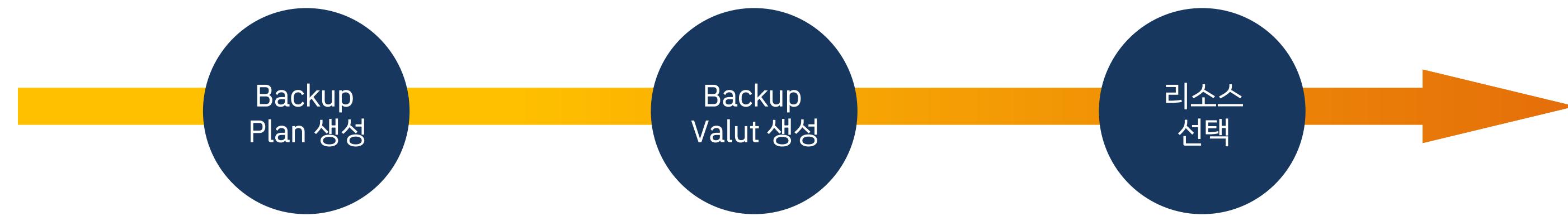
백업

AWS Backup 기본 구성도



백업

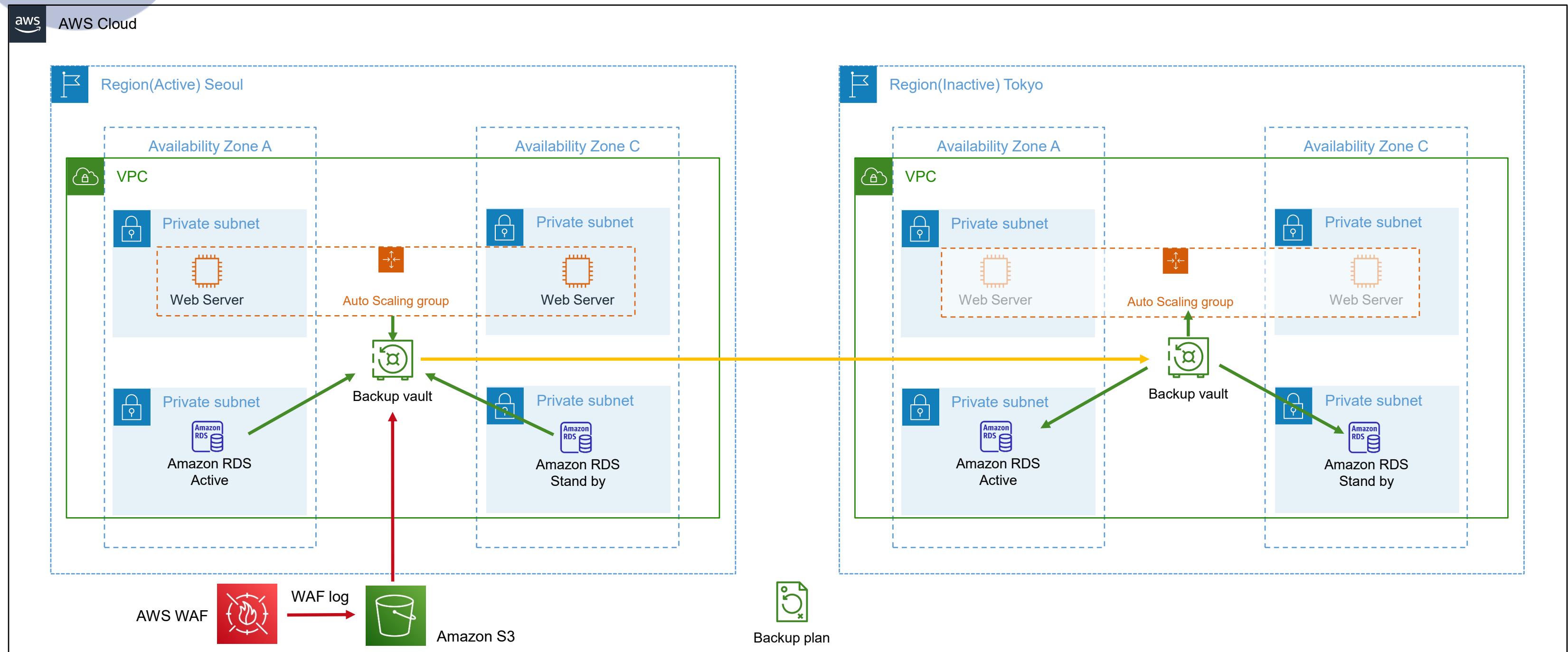
AWS Backup 순서도



백업

재해대비 멀티리전 Backup 구성도

서울리전의 Web server, RDS, S3를 백업 후
도쿄 리전의 백업볼트에 교차백업플랜 구성



백업

Backup Plan 생성

생성 후 새 계획 수립, 백업 계획 이름 지정

AWS Backup > 백업 계획 > 백업 계획 생성

백업 계획 생성 정보

시작 옵션

시작하는 방법을 선택합니다. 정보

- 템플릿으로 시작
AWS Backup에서 제공하는 템플릿을 기반으로 백업 계획을 생성합니다.
- 새 계획 수립
처음부터 새 백업 계획을 구성합니다.
- JSON을 사용하여 계획 정의
기존 백업 계획의 JSON 표현식을 수정하거나 새 표현식을 생성합니다.

백업 계획 이름
백업 계획 이름 지정

dailybackupplan

백업 계획 이름은 대/소문자를 구분합니다. 1~50자의 영숫자 혹은 '-' 문자를 포함해야 합니다.

▼ 백업 계획에 추가된 태그

태그가 없습니다.

새 태그 추가

최대 50개의 태그를 더 추가할 수 있습니다.

백업

Backup Plan 생성

백업 규칙 구성 정보
백업 일정, 백업 기간 및 수명 주기 규칙을 정의하여 백업 규칙을 추가합니다. 나중에 이 백업 계획에 다른 백업 규칙을 추가할 수 있습니다. 백업 비용은 백업 구성에 따라 다르게 발생합니다.

백업 규칙 이름
dailybackupplan
백업 규칙 이름은 대/소문자를 구분합니다. 1~50자의 영숫자 혹은 '-' 문자를 포함해야 합니다.

백업 볼트 정보
Default 새 백업 볼트 생성

백업 빈도 정보
매일

특정 시점으로 복구(PITR)에 대한 지속적 백업 활성화 정보
RDS 및 S3 리소스에 사용할 수 있습니다.

백업 기간
 백업 기간 기본값 사용 - 권장 정보
8시간 이내에 시작되는 오전 5시(UTC)입니다.
 백업 기간 사용자 지정

백업 기간 시작 시간
05 : 00 PM UTC 시간

다음 시간 내에 시작 정보
1시간

다음 시간 내에 완료 정보
6시간

백업 규칙 생성

- 백업빈도
- 시작시간(사벽 2시)
- 백업대상볼트 지정

※ 백업시간은 UTC기준 / 한국의 경우 UTC시간-9시간

백업

Backup Vault 생성

백업 볼트 생성

일반

백업 볼트 이름
 백업 볼트 이름은 대/소문자를 구분합니다. 2~50자의 영숫자 또는 '-' 문자를 포함해야 합니다.

암호화 키 정보

설명	계정	키 ID	상태
-	이 계정(035671322814)	f566bc8d-ae3e-4268-9dbf-fa4037593281	<input checked="" type="checkbox"/> 활성화됨

백업 볼트 태그 - 선택 사항
여기에 지정된 태그는 백업 볼트를 구성하고 추적하는 데 유용합니다.

키	값 - 선택 사항
<input type="text" value="Name"/> X	<input type="text" value="SeoulVault"/> X

새 태그 추가

최대 49개의 태그를 더 추가할 수 있습니다.

취소 **백업 볼트 생성**

서울리전의 백업볼트 생성

백업

Backup Plan 생성

콜드 스토리지로 전환 정보

일 30

보존 기간 정보

일 180

대상으로 복사 정보

아시아 태평양 (도쿄) 제거

다른 계정의 볼트로 복사

대상 백업 볼트
백업 사본이 생성될 볼트입니다.

Default 새 백업 볼트 생성

▶ 고급 설정

복사 추가

▶ 복구 시점에 추가되는 태그
AWS Backup은(는) 생성 시 보호된 리소스의 태그를 복구 시점으로 복사합니다. 복구 시점에 추가할 추가 태그를 지정할 수 있습니다.

백업 규칙 생성

- 콜드 스토리지로 전환
- 보존기간 지정
- 교차리전의 백업 대상 볼트 지정

백업

Backup Plan 생성

백업 볼트 생성

일반

백업 볼트 이름

TokyoVault

백업 볼트 이름은 대/소문자를 구분합니다. 2~50자의 영숫자 또는 '-' 문자를 포함해야 합니다.

암호화 키 정보

(기본값) aws/backup

설명	계정	키 ID	상태
Default key that protects my Backup data when no other key is defined	이 계정(035671322814)	7b2f3ccd-46a9-48d1-b828-7e0f770dcc15	활성화됨

백업 볼트 태그 - 선택 사항

여기에 지정된 태그는 백업 볼트를 구성하고 추적하는데 유용합니다.

키	값 - 선택 사항
<input type="text" value="Name"/> X	<input type="text" value="tokyoVault"/> X

새 태그 추가

최대 49개의 태그를 더 추가할 수 있습니다.

취소

백업 볼트 생성

도쿄리전의 백업볼트 생성

백업

Backup Vault 생성

dailybackupplan

삭제 JSON 보기

요약

백업 계획 이름 dailybackupplan	버전 ID ZGMyZWQ4ZjktZWFiNS00MjllLTlMTgtMDc2N2FlNmNiZjUy	마지막으로 수정한 날짜 May 11th, 2022, 1:36 PM (UTC+09:00)	마지막 실행 시간 -
백업 계획 ID 85d156cd-6810-4477-8cb3-3dc8817284c5			

백업 규칙 (1)

백업 규칙은 백업 일정, 백업 기간 및 수명 주기 규칙을 지정합니다.

편집 삭제 백업 규칙 추가

이름	백업 볼트	대상 백업 볼트
<input type="radio"/> dailybackupplan	SeoulVault	TokyoVault

리소스 할당 (0)

리소스 할당은 이 백업 계획에 따라 백업되는 리소스를 지정합니다.

삭제 리소스 할당

이름	IAM 역할 ARN	생성 시간

백업

리소스 할당

리소스 할당 (0)

리소스 할당은 이 백업 계획에 따라 백업되는 리소스를 지정합니다.

삭제 리소스 할당

< 1 ... > | 설정

이름	IAM 역할 ARN	생성 시간
빈 리소스		
리소스 할당이 없습니다.		
리소스 할당		

백업

리소스 할당

리소스 할당 정보
태그 및 리소스 ID를 사용하여 이 백업 계획에 리소스를 할당합니다.

1. 리소스 선택 정의 정보
모든 리소스를 보호하거나 유형별 또는 ID별로 리소스를 지정합니다.

모든 리소스 유형 포함
계정에서 활성화된 모든 리소스 유형을 보호합니다.

특정 리소스 유형 포함
유형별로 리소스를 선택하거나 ID별로 개별 리소스를 지정합니다.

2. 특정 리소스 유형 선택 정보
이 백업 계획으로 보호할 특정 리소스 유형을 선택합니다. 선택 항목에서 특정 리소스 ID를 제외할 수도 있습니다.

리소스 유형 선택 ▾

리소스 유형	인스턴스 ID
EC2	리소스 선택 ▾
	제거
i-0efc2ac49c21f2ab2 X	
i-0863c770c813acac1 X	

리소스 유형	데이터베이스 이름
RDS	리소스 선택 ▾
	제거
모든 데이터베이스 X	

리소스 유형	버킷 이름
S3	리소스 선택 ▾
	제거
S3 버킷의 버전 관리가 활성화되어 있어야 합니다. 자세히 알아보기 ⓘ	
모든 버킷 X	

백업할 리소스 지정

특정 리소스를 선택하거나 해당 리전의 모든 리소스를 선택 가능

백업

리소스 할당

리소스 할당 (1)				
리소스 할당은 이 백업 계획에 따라 백업되는 리소스를 지정합니다.				
이름	IAM 역할 ARN	생성 시간		
<input type="radio"/> basicresource	arn:aws:iam::035671322814:role/service-role/AWSBackupDefaultServiceRole	May 11th, 2022, 1:42 PM (UTC+09:00)	<	1 ... >

백업

Backup 작업의 확인

서울리전의 백업작업확인

AWS Backup > 백업 블트 > backupvalut-seoul

backupvalut-seoul

[삭제](#) [액세스 관리](#)

요약

백업 블트 이름	backupvalut-seoul	생성 날짜	May 9th, 2022, 4:21 PM (UTC+09:00)	KMS 암호화 키 ID	f566bc8d-ae3e-4268-9dbf-fa4037593281
백업 블트 ARN	arn:aws:backup:ap-northeast-2:035671322814:backup-vault:backupvalut-seoul				

백업 (4)

[리소스 유형, 복구 시점 ID 또는 소스 계정 ID로 필터링](#)

<input type="checkbox"/> 복구 시점 ID	상태	리소스 ID	리소스 유형	백업 유형	생성 시간
aws-waf-logs-seoulwaf-20220509131945-d47e7158	완료됨	aws-waf-logs-seoulwaf	S3	스냅샷	May 9th, 2022, 10:00 PM
image/ami-0d35ec7699064caa2	완료됨	instance/i-0ed81535c9a4c47b8	EC2	이미지	May 9th, 2022, 6:00 PM
image/ami-0069e7f139abb08e9	완료됨	instance/i-0863c770c813acac1	EC2	이미지	May 9th, 2022, 6:00 PM
image/ami-0bed11ae4633d74af	완료됨	instance/i-0efc2ac49c21f2ab2	EC2	이미지	May 9th, 2022, 6:00 PM

백업

Backup 작업의 확인

도쿄리전의 교차백업 작업 확인

AWS Backup > 백업 블트 > backupvault-tokyo

backupvault-tokyo

[삭제](#) [액세스 관리](#)

요약					
백업 블트 이름 backupvault-tokyo	생성 날짜 May 9th, 2022, 4:22 PM (UTC+09:00)	KMS 암호화 키 ID 7b2f3ccd-46a9-48d1-b828-7e0f770dcc15			
백업 블트 ARN arn:aws:backup:ap-northeast-1:035671322814:backup-vault:backupvault-tokyo					
백업 (4)					
<input type="text"/> 리소스 유형, 복구 시점 ID 또는 소스 계정 ID로 필터링					
복구 시점 ID	상태	리소스 ID	리소스 유형	백업 유형	생성 시간
aws-waf-logs-seoulwaf-20220509131945-d47e7158	완료됨	aws-waf-logs-seoulwaf	S3	스냅샷	May 9th, 2022, 10:00 PM
image/ami-0d35ec7699064caa2	완료됨	instance/i-0ed81535c9a4c47b8	EC2	이미지	May 9th, 2022, 6:00 PM
image/ami-0069e7f139abb08e9	완료됨	instance/i-0863c770c813acac1	EC2	이미지	May 9th, 2022, 6:00 PM
image/ami-0bed11ae4633d74af	완료됨	instance/i-0efc2ac49c21f2ab2	EC2	이미지	May 9th, 2022, 6:00 PM

백업

Backup 된 이미지 복구

복원할 리소스와 인스턴스의 유형, VPC와 서브넷을 지정 가능

백업 복원

예약된 백업, 수명 주기 관리 및 빠른 복원과 같은 주요 기능을 사용하면서 다른 리소스의 백업을 중앙에서 관리할 수 있도록 EC2 인스턴스를 복원합니다. 전체 인스턴스 복원 기능에 액세스하려면 다음을 참조하십시오. [인스턴스 시작 마법사](#)

네트워크 설정

인스턴스 유형 정보
인스턴스의 컴퓨팅 및 메모리 용량을 정의합니다.

t2.micro - 1 vCPU, 1 GiB RAM

Virtual Private Cloud(VPC)
VPC를 선택하여 가상 네트워킹 환경을 정의합니다.

기본 VPC(vpc-052774b33536a14fa)

서브넷 정보
서로 다른 EC2 리소스를 서로 간에 또는 인터넷으로부터 격리하는 데 사용할 수 있는 VPC의 IP 주소 범위를 지정합니다. 각 서브넷은 하나의 가용 영역에 상주합니다.

subnet-052c628064e99099f | default

보안 그룹 정보
보안 그룹을 지정하여 인스턴스의 트래픽을 제어하는 방화벽 규칙 세트를 결정합니다.

보안 그룹 추가

launch-wizard-1 X

인스턴스 IAM 역할 정보
EC2 인스턴스에 AWS 자격 증명을 자동으로 배포할 IAM 역할을 지정합니다.

원래 IAM 역할로 복원

백업

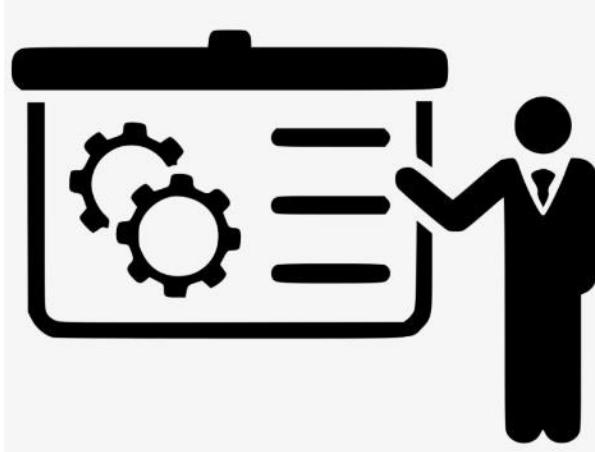
Backup 작업의 확인

EC2 인스턴스 및 S3의 복원 확인

인스턴스 (3) 정보								
		C		연결	인스턴스 상태 ▾	작업 ▾	인스턴스 시작	▼
<input type="text"/> 검색					< 1 >	<input type="button"/>		
인스턴스 상태 = running		X	필터 지우기					
□	Name	▼	인스턴스 ID	인스턴스 상태	▼	인스턴스 유형	▼	상태 검사
<input type="checkbox"/>	WebApp1		i-024ebe12190a4d2ac	<input checked="" type="radio"/> 실행 중	<input type="radio"/> Q	t2.micro		<input checked="" type="radio"/> 2/2개 검사 통과
<input type="checkbox"/>	WebApp2		i-0b62e7b4e1ffcec14	<input checked="" type="radio"/> 실행 중	<input type="radio"/> Q	t2.micro		<input checked="" type="radio"/> 2/2개 검사 통과
<input type="checkbox"/>	-		i-00898ad733b0b7e3a	<input checked="" type="radio"/> 실행 중	<input type="radio"/> Q	t2.micro		<input checked="" type="radio"/> 2/2개 검사 통과

백업

실제 업무 적용 시 보완점



IAM 유저권한 설정으로 Backup plan에 접근 가능한
유저생성 및 권한설정 필요

Backup Vault에 kms 암호화키를 적용하여 암호화된 백업

하이브리드 클라우드의 경우 Storage Gateway 구성 후
온프레미스 스토리지의 통합 백업

05

총평

프로젝트 평가

- 홍서의

종평

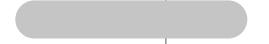
홍세의

현재가치, 향후방향, 프로젝트 관리
질의응답

종평

프로젝트 관리

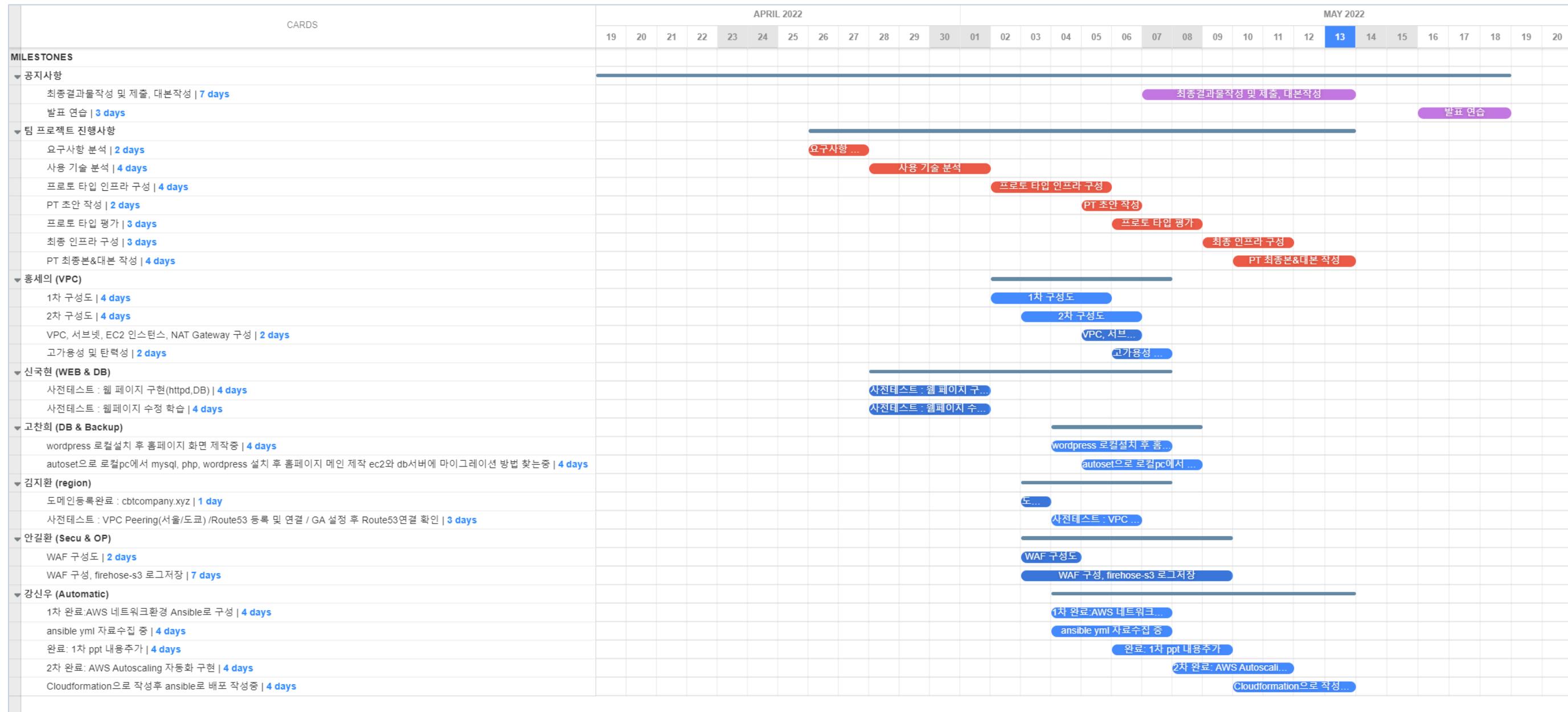
프로젝트 기간: 5주

	4월		5월		
	3째주	4째주	1째주	2째주	3째주
아이디어 접근					
관련자료 조사					
사용기술 분석					
사전 테스트, 프로토타입					
최종 인프라 구현					
프로젝트 발표/시연					

종평

프로젝트 관리

프로젝트 관리 : 세부 일정 및 WBS



종평

프로젝트 관리

프로젝트 관리 : 협업툴 사용 / 이슈사항, 일정 등에 활용

트렐로 (Elegantt | The leading Gantt Chart for Trello)

The screenshot shows a Trello board titled "AWS" with a Gantt chart view. The Gantt chart spans from April 19 to May 25, 2022. Key tasks shown include "최종결과물작성 및 제출, 대본작성" (7 days), "발표 연습" (3 days), " 요구사항 분석" (2 days), "사용 기술 분석" (4 days), "프로토 타입 인프라 구성" (4 days), "PT 조안 작성" (2 days), and "프로토 타입 평가" (3 days). Below the Gantt chart, there are several columns of cards representing different project components:

- 팀 프로젝트 진행사항:** Includes cards for "요구사항 분석", "사용 기술 분석", "프로토 타입 인프라 구성", "PT 조안 작성", and "프로토 타입 평가".
- 홍세의 (VPC):** Shows a diagram of a VPC network with multiple subnets and gateways.
- 신국현 (WEB & DB):** Shows a diagram of a web and database architecture with multiple services connected.
- 고진희 (DB & Backup):** Shows a diagram of a database and backup architecture.
- 김지환 (region):** Shows a diagram of AWS regions and their connectivity.
- 안길환 (Secu & OP):** Shows a diagram of security and operations architecture.
- 강신우 (Automatic):** Shows a diagram of an automatic deployment architecture using Ansible.
- 구성도 참고:** Includes diagrams for "Multi-site active/active DR architecture", "AWS High Availability & Fault Tolerance Architecture", and "AWS Well-Architected Labs".
- 참고 기사:** Includes articles such as "2022.05.02 월서버 RDS연동 Wordpress 구현 실습내용.pdf", "Amazon RDS에서 Wordpress 배포", and "2022.05.02 월서버 SSH 보안 기법 [SSH 터널링]".

종평

프로젝트 관리

프로젝트 관리 : 협업툴 사용 / 노트 작업, 회의, 기술자료 공유

노션 ([Notion \(노션\)](#) – 모든 팀을 위한 하나의 워크스페이스)

팀 프로젝트

1조 : CBT (Close Beta Test)

- ▶ 이전 기수 프로젝트 관련자료 및 발표자료, 피드백
- ▶ 솔루션 아키텍처 덤프 오답 정오표

구글 미트 주소 → <https://meet.google.com/uqm-ppes-vvi>

Team Project 산출물

- AWS Icon 참고용.pptx 25108.5KB
- Draw.io AWS 구성 도 원본
- 원본 CBT_TeamProject_template.pptx 3160.3KB
- CBT_TeamProject2.pptx 16.87 MB

반영 완료

CBT_TeamProject_김지환_mutireigon_network(ver.1.1).pptx 8343.8KB

CBT Team project ver2.pptx 15720.2KB

역할 분담

Pilot Light 설명도

- DR 및 Pilot Light 설명은 아래 링크에 한글로 잘 정리되어 있어 복마킹 합니다.

AWS Disaster Recovery Whitepaper
AWS Disaster Recovery Whitepaper
<https://xjayleex.github.io/posts/aws/awssap-dr-whitepaper.html>

Backup & Restore
Pilot Light
Warm Standby
Multi Site

- Lower Priority Services
 - Lower Priority Services
 - Solutions Cloud Monitoring Services
 - Cloud Iot
- Lower Priority Services
 - Lower Priority Services
 - Solutions Cloud Monitoring Services
 - Cloud Iot
- Cloud Amplification
 - Cloud Amplification
 - Solutions Cloud Monitoring Services
 - Cloud Iot
- Major Critical Services
 - Major Critical Services
 - Cloud Amplification
 - Solutions Cloud Monitoring Services
 - Cloud Iot

[AWS] 재해복구(disaster recovery) (2) 파일럿 라이트 복구 시나리오 해설

종평

프로젝트 관리

사용된 기술

TEAM CBT

재해 대비 멀티 리전 구성

자동화	OS	ubuntu Linux LTS 20.04
	파이썬	
Web Server	워드프레스	Ver.
	웹 서비스	Apache, PHP
	데이터베이스	MySQL
코드 편집툴	VSCode	
Domain Service	freenom.com	
Cloud Service	AWS	
프로젝트 관리	trello(Elegantt), Notion	

종평

현재 가치

프로젝트의 완성도

TEAM CBT

재해 대비 멀티 리전 구성

AWS Elastic Disaster Recovery 서비스 의 탄력성, 연속성에 준하는 모델

민첩성, 탄력성, 복구에 있어서 충분한 테스트가 필요하다.

Amazon 자체의 복구 서비스가 존재하지만 그 표준을 최대한 따를 수 있게 직접 만들어 보며 추후 고객들에게 서비스에 대해 설명하고 이해하는데 도움이 될 것이라 생각했다.

실시간 과금인 AWS 자체의 복구 서비스의 경우 복구 서비스의 원리를 설명해야 과금대비 필요한 서비스인지 설명할 수 있다.

종평

향후 방향

프로젝트의 발전 가능성

- ### 멀티 클라우드와의 접목

멀티 클라우드가 대세가 되어가고 있다.

추후에는 AWS 뿐만 아니라 NCP, GCP 등의 타 플랫폼과 멀티 클라우드 구현을 목표로 하고자 한다.

- ### 보안의 강화

AWS 접속자의 그룹과 구조 IAM 역할을 구체적으로 정의 할 필요 있으며 WAF 의 효율적 운용을 위해 AWS Firewall Manager 사용과 실시간 로그 수집과 보관을 위한 Fierhose, S3, Lambda function 을 도입하여 증적감사와 분석에 활용하는 한편 어플리케이션 취약성 관리를 위해 amazon inspector 도입도 검토해야 한다. 또한, AWS Config 와 Cloud Trail 을 활용한 관리적 보안으로도 발전방향을 이어나갈 필요있다.

- ### 자동화 툴의 숙련

Ansible을 포함한 Cloudformation, Terraform 으로

구축한 네트워크 환경 및 서비스 중 AWS CLI로 구현할 수 있는 것들은 추후 대부분 자동화해보고자 한다.

질의 응답

01

서버

높은 가용성과 탄력성을 가지는
웹 서버의 구축

- 강신우 : EC2 부하분산, 자동확장
Ansible을 활용한 네트워크 구성 자동화
- 신국현 : Wordpress, RDS DB 구성과 백업

02

네트워크

다중 지역(국가) 간 네트워크 연결과
호스팅 서비스

- 김지환 : VPC Peering, Route53
Global Accelerator

03

보안, 운영

웹서버의 취약점에 대비하고
증적에 대한 보관

- 안길환 : WAF, Cloudwatch,
Kinesis Firehose

04

백업

재해대비 멀티 리전
AWS 백업

- 고찬희 : 백업 계획, 규칙, 볼트
백업 복원

05

총평

프로젝트 평가

- 홍세의 : 현재가치, 향후방향,
프로젝트 진행 사항
질의응답

2022 TEAM PROJECT

Thank you

재해 대비 멀티 리전 구성

2022. 5. 20