

클라우드컴퓨팅

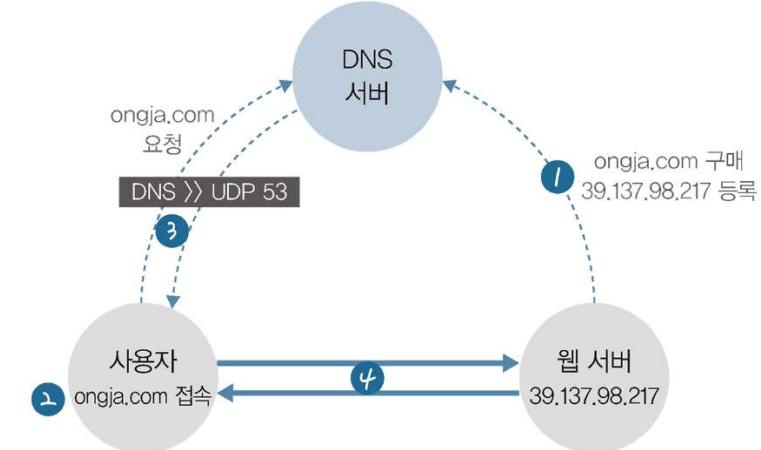


Amazon AWS 활용: 고급네트워킹 서비스

DNS

AWS 고급 네트워킹 서비스

- DNS 란?
 - Domain Name System의 약자
 - 네트워크 통신을 위한 IP 주소와 문자 주소인 도메인 간 매핑하여 연결하는 서비스
 - 도메인 주소의 구조 ;



▲ 그림 7-2 도메인 주소를 이용한 통신 과정

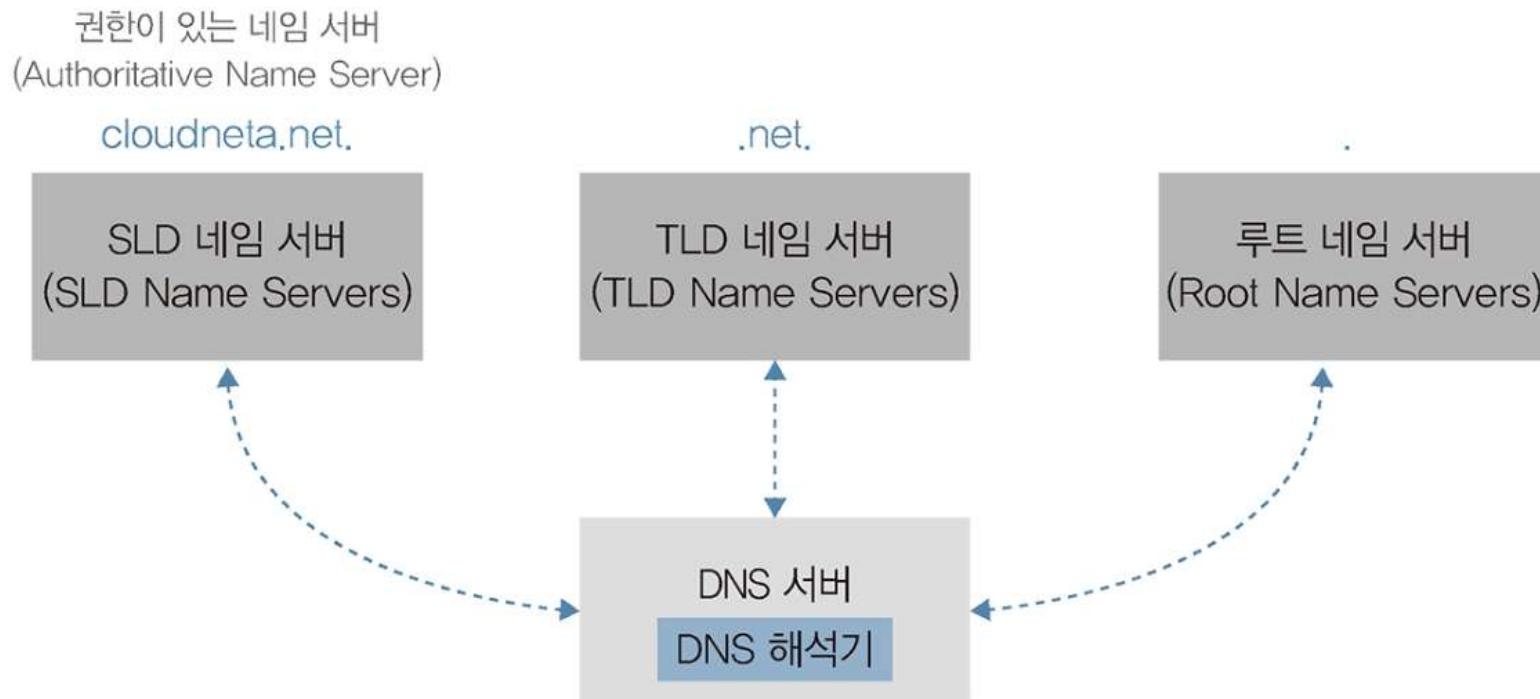
- DNS 서버 종류
 - 도메인 구조를 루트, 탑 레벨, 세컨드 레벨, 서브 도메인으로 구분하는 이유는 영역별 도메인을 관리하는 주체를 분리하기 위함
 - 도메인은 DNS 네임 서버로 관리하는데, 이 때 도메인 영역별로 DNS 네임 서버를 분리해서 관리함

AWS 고급 네트워킹 서비스

- DNS 서버 종류
 - 루트 네임 서버
 - 루트 도메인을 관리하는 DNS 서버
 - DNS 요청에 대해 TLD에 해당하는 네임 서버 정보를 응답
 - 참고: 루트 네임 서버는 전 세계에 13개, A ~ M-root (물리적인 서버의 수는 더 많지만, Anycast 기술을 사용해서 같은 이름을 공유함)
 - TLD 네임 서버
 - 도메인 이름의 최상위 영역인 TLD를 관리하는 DNS 서버
 - TLD 영역에서 식별되는 모든 SLD를 관리하여 DNS 요청에 대해 SLD 네임 서버 정보를 응답
 - 예: .com 이라는 TLD 네임 서버는 .com 내에 있는 google.com 도메인을 관리하는 SLD 네임 서버 정보를 알고 있음. 해당 도메인에 대한 DNS 요청이 있으면 SLD 네임 서버 주소를 알려줌
 - SLD 네임 서버
 - 실질적인 도메인 이름을 관리하는 DNS 서버
 - 실제 도메인의 최종 관리 서버로, '권한이 있는 네임 서버'라고 함
 - 도메인 주소에 대한 IP 주소를 확인하는 가장 마지막 단계

AWS 고급 네트워킹 서비스

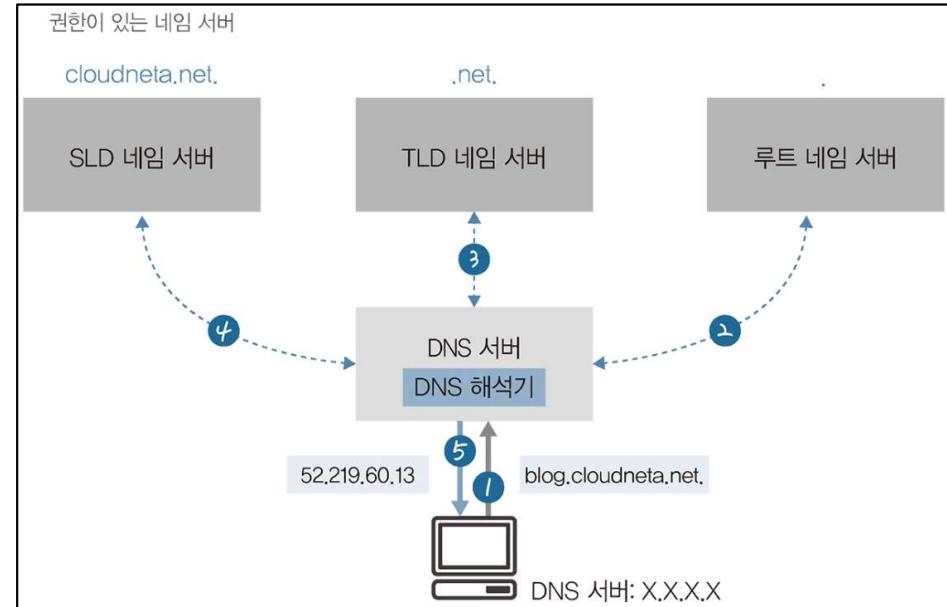
- DNS 해석기
 - 사용자와 네임 서버 사이에서 중계자 역할을 수행
 - 사용자가 DNS 해석기로 DNS 요청을 하면, DNS 해석기가 DNS 네임 서버와 정보를 주고 받아 도메인 주소를 해석하여 최종적으로 IP 주소를 사용자에게 알려줌



AWS 고급 네트워킹 서비스

- DNS 통신의 흐름(사용 순서)

1. 사용자 PC에서 blog.cloudneta.net이라는 도메인 주소의 IP 주소를 알기 위해 DNS 서버에 질의함. 이 때, DNS 서버는 DNS 해석기를 이용하여 다양한 네임 서버와 통신하는 중개자 역할을 수행
2. DNS 해석기는 먼저 루트 네임 서버에 blog.cloudneta.net 도메인 주소의 IP 주소를 질의. 루트 네임 서버는 해당 도메인의 IP 주소는 모르지만, .net의 TLD 네임 서버는 알고 있기 때문에 해당 정보를 DNS 해석기에 전달
3. DNS 해석기는 .net의 TLD에 도메인 주소를 질의함. TLD는 해당 도메인의 주소는 모르지만, cloudneta.net의 SLD는 알고 있으므로 해당 정보를 DNS 해석기에 전달
4. DNS 해석기는 cloudneta.net의 SLD 네임 서버에 blog.cloudneta.net 도메인의 IP 주소를 질의. 해당 SLD는 도메인 주소의 최종 정보가 저장된 '권한이 있는 네임 서버'이며 해당 도메인의 IP 주소를 DNS 해석기에 회신함
5. DNS 해석기는 수신한 IP 주소를 사용자 PC에 전달함



AWS 고급 네트워킹 서비스

- DNS 레코드 유형
 - DNS 레코드는 도메인에 대한 요청 처리 방법을 정의한 것으로, 용도에 따라 DNS 레코드 유형을 분류함
 - A 레코드 유형
 - 도메인 이름을 IPv4 주소로 매플하는 가장 기본적인 DNS 레코드 유형으로, 아래의 형태로 표현
`blog.cloudneta.net A 52.219.60.13`
 - blog.cloudneta.net이라는 도메인 주소로 질의하면 IPv4 주소인 52.219.60.13으로 응답
 - AAAA 레코드 유형
 - A 레코드 유형의 IPv6 버전
`bblog.cloudneta.net AAAA 2001:A10::2001`
 - NS 레코드 유형
 - 도메인 이름을 네임 서버 주소로 매플하는 DNS 레코드 유형으로, 아래의 형태로 표현
`net NS a.gtld-servers.net.`
 - blog.cloudneta.net이라는 도메인 주소로 질의하면 .net의 TLD 네임 서버 주소인 a.gtld-servers.net.이라는 도메인 주소로 응답

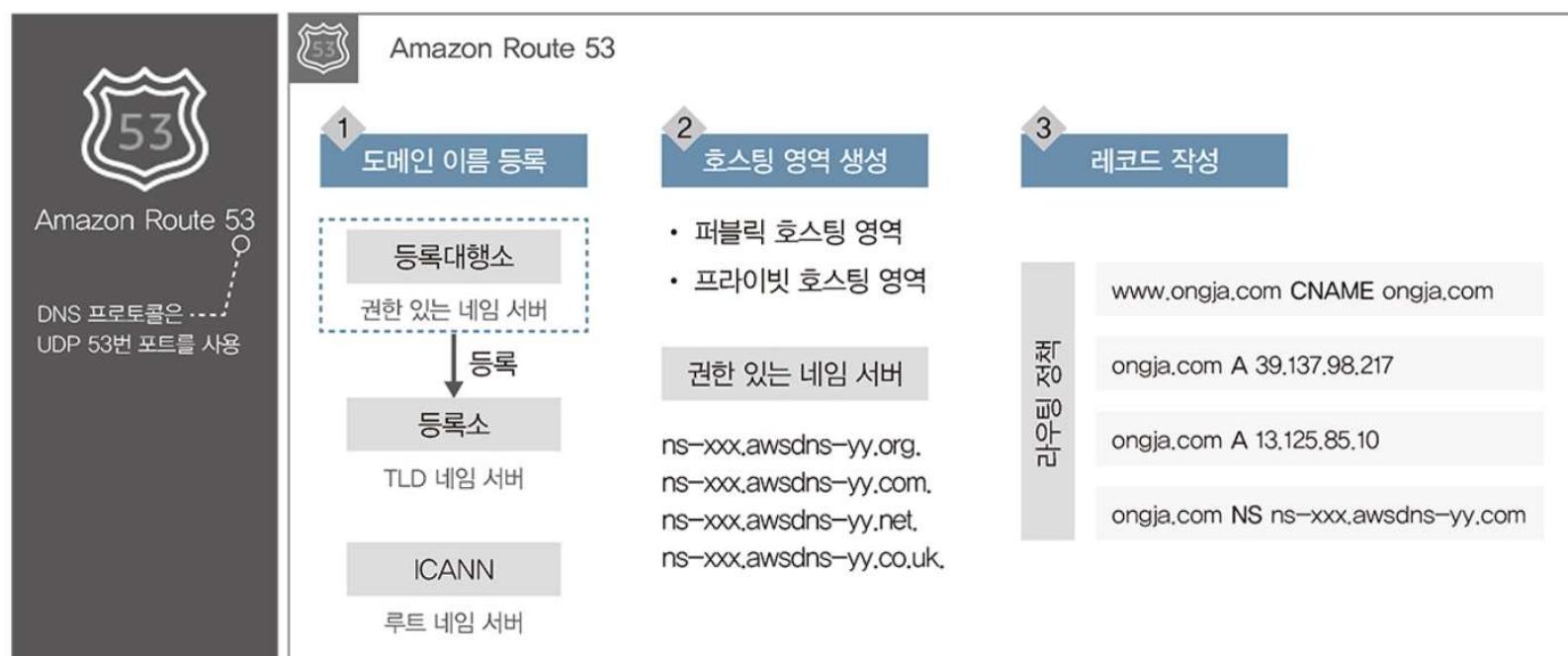
AWS 고급 네트워킹 서비스

- DNS 레코드 유형
 - CNAME 레코드 유형
 - 도메인 이름의 별칭을 지정하는 DNS 레코드 유형
 - 아래의 형태로 표현:

```
www.cloudneta.net CNAME cloudneta.net
```

AWS 고급 네트워킹 서비스

- Amazon Route 53 서비스
 - AWS에서 제공하는 관리형 DNS 서비스
 - DNS 프로토콜은 UDP 53번 포트를 사용하기 때문에, 서비스 이름을 53이라고 부름
 - 주요 기능
 - 도메인 이름 등록
 - 호스팅 영역 생성
 - 레코드 작성 등



AWS 고급 네트워킹 서비스

- Amazon Route 53 서비스
 - 도메인 이름 등록
 - Amazon Route 53을 통해서 도메인 이름 등록을 할 수 있음

The screenshot shows the AWS Route 53 console interface. On the left, a sidebar lists various services: 대시보드, 호스팅 영역, 상태 검사, IP 기반 라우팅, CIDR 모음, 트래픽 흐름, 트래픽 정책, 정책 레코드, 도메인, 등록된 도메인 (highlighted with a blue box and labeled '1 진입'), 대기 중인 요청, 확인자, VPC, 인바운드 엔드포인트, 아웃바운드 엔드포인트, 규칙. The main content area has a header with an information icon and the text: '이제 새로운 Route 53 콘솔 환경을 사용할 수 있습니다. Route 53 콘솔을 더 쉽게 사용할 수 있도록 재설계했습니다. 새 콘솔을 사용해 보십시오. 피드백을 바탕으로 사용자 경험을 계속 개선하고 있습니다. 계속 지켜봐 주세요.' Below this, a large button labeled '2 클릭 도메인' is shown, with the '도메인 등록' tab selected (highlighted with a blue box and labeled '2 클릭'). A search bar contains the placeholder '접두사로 도메인 검색'. A table displays a single registered domain: 'ongja.click' (Domain Name), '연락처 없음' (Contact Information), '2024-03-17' (Last Updated), and a delete icon.

AWS 고급 네트워킹 서비스

- Amazon Route 53 서비스
 - 호스팅 영역 생성
 - 호스팅 영역을 생성하여 네임 서버를 관리할 수 있음
 - 호스팅 영역을 생성해야 Amazon Route 53이 등록된 도메인 이름에 대한 권한 있는 네임 서버이자 SLD 네임 서버의 역할을 수행할 수 있음
 - 호스팅 영역의 네임 서버들은 고가용성을 위해 다수의 서버로 구성
 - 트래픽을 라우팅하는 방식에 대한 정보가 포함
 - 퍼블릭 호스팅 영역 ; 인터넷에서 트래픽을 라우팅하는 방법을 지정하는 레코드를 포함
 - 프라이빗 호스팅 영역 ; Amazon VPC에서 트래픽을 라우팅하는 방법을 지정하는 레코드를 포함

The screenshot shows the AWS Route 53 Hosted Zone Management console for the domain 'cloudneta.click'. The 'Hosted Zone' tab is selected. In the 'Records' section, there are two existing records: one NS record for 'cloudneta.click' and one SOA record for 'cloudneta.click'. A new record is being created, indicated by the 'Record Creation' dialog at the bottom. Step 1, 'Select Type', is completed, showing the 'A' type selected. Step 2, 'Configure Record', is active, with the 'Name' field set to 'cloudneta.click' and the 'Type' dropdown also set to 'A'. The 'Value' field contains the IP address '172.20.10.10'. The 'TTL' dropdown is set to '172800'. The 'Health Check' and 'Forwarding' sections are collapsed. The 'Create Record' button is highlighted in blue.

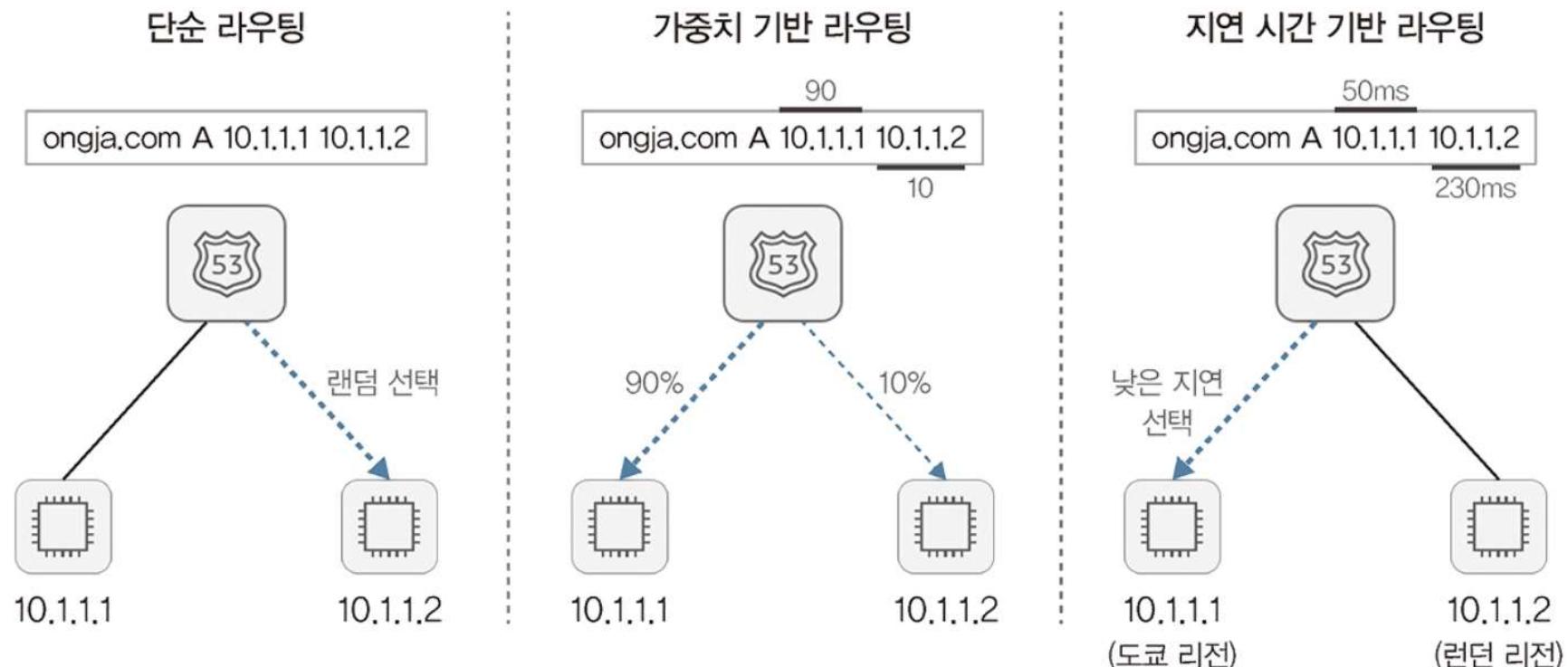
AWS 고급 네트워킹 서비스

- Amazon Route 53 서비스
 - 레코드 작성
 - Amazon Route 53은 DNS 레코드를 정의하여 도메인에 대한 요청 처리 방법을 정의할 수 있음
 - 이런 DNS 레코드는 다양한 형태의 라우팅 정책을 연결하여 도메인 요청에 대한 응답 방식을 정의할 수 있음



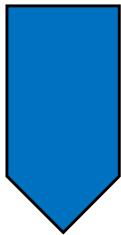
AWS 고급 네트워킹 서비스

- Amazon Route 53 서비스
 - 라우팅 정책



AWS 고급 네트워킹 서비스

- Amazon Route 53 서비스
 - 라우팅 정책 (전체)
 - https://docs.aws.amazon.com/ko_kr/Route53/latest/DeveloperGuide/routing-policy.html
 - 단순 라우팅
 - 장애 조치 라우팅
 - 지리적 라우팅
 - 지리 근접 라우팅
 - 자연 시간 기반 라우팅
 - IP 기반 라우팅
 - 다중값 응답 라우팅
 - 가중치 기반 라우팅

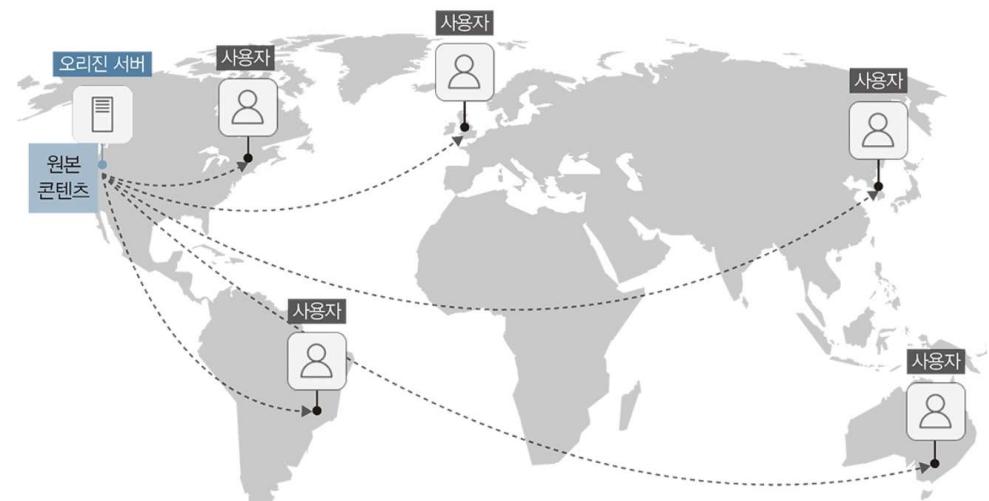


Amazon AWS 활용: 고급네트워킹 서비스

CDN

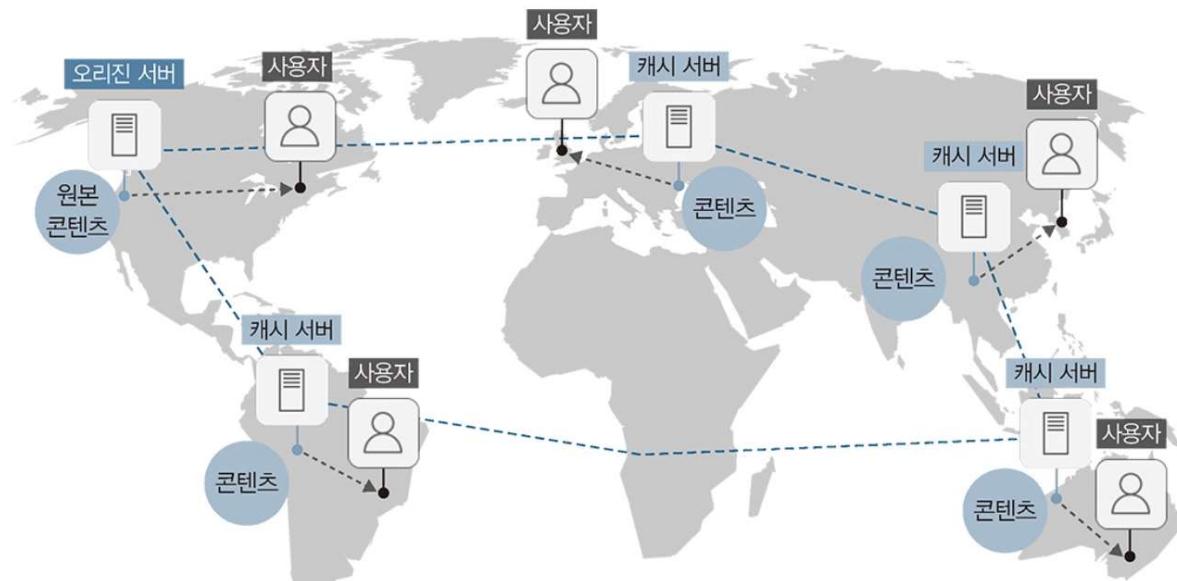
AWS 고급 네트워킹 서비스

- CDN (Contents Delivery Network)
 - 콘텐츠 제공자와 사용자가 지리적으로 멀리 떨어져 있는 환경에서, 캐시 기능을 사용하여 콘텐츠를 빠르게 전달하는 네트워크 기술
 - (참고: CDN 미사용 환경) 일반적인 네트워크 통신 환경에서는 원본 콘텐츠를 가지고 있는 오리진(origin) 서버에서 사용자에게 콘텐츠를 전달함
 - 오리지 서버에 높은 부하가 발생
 - 지리적으로 멀리 떨어진 사용자의 지연 시간이 증가
 - CDN 기술의 핵심은 캐시 서버를 지역적으로 분산하고, 콘텐츠를 동기화하여 분산 처리하는 것



AWS 고급 네트워킹 서비스

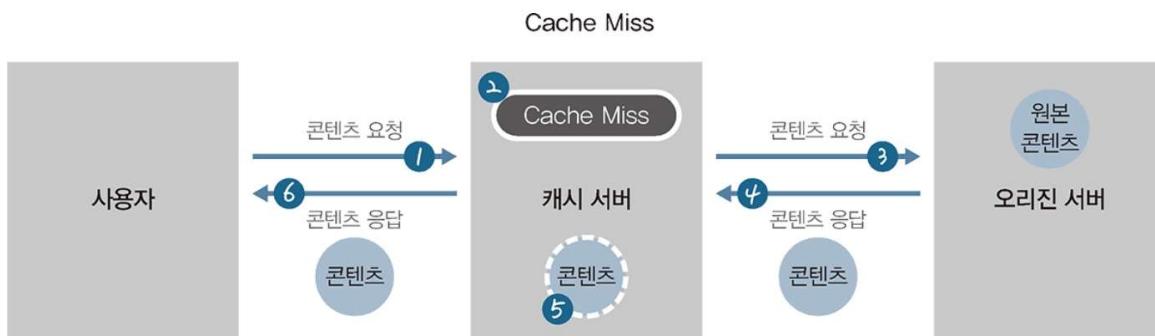
- CDN 환경
 - 오리진 서버에서 지역적으로 분산된 캐시 서버에 콘텐츠를 동기화해서 콘텐츠가 분산된 환경을 구성함
 - 사용자는 지리적으로 인접한 서버로부터 콘텐츠를 전달받아, 빠르고 효율적인 서비스 제공 받음



AWS 고급 네트워킹 서비스

- CDN 캐싱 방식

- 캐싱 ; 오리진 서버의 원본 콘텐츠를 지역적으로 분산된 캐시 서버로 전달하고 콘텐츠를 저장하는 것
 - Cache Miss ; 캐시 서버에 콘텐츠가 없는 경우
 - Cache Hit ; 캐시 서버에 콘텐츠가 있는 경우



캐시 미스 발생 시, 오리진 서버로 부터 콘텐츠 수신
(이 경우, 오리진 서버에 직접 콘텐츠를 요청하는 것 보다 지연 시간이 더 길어질 수 있음)



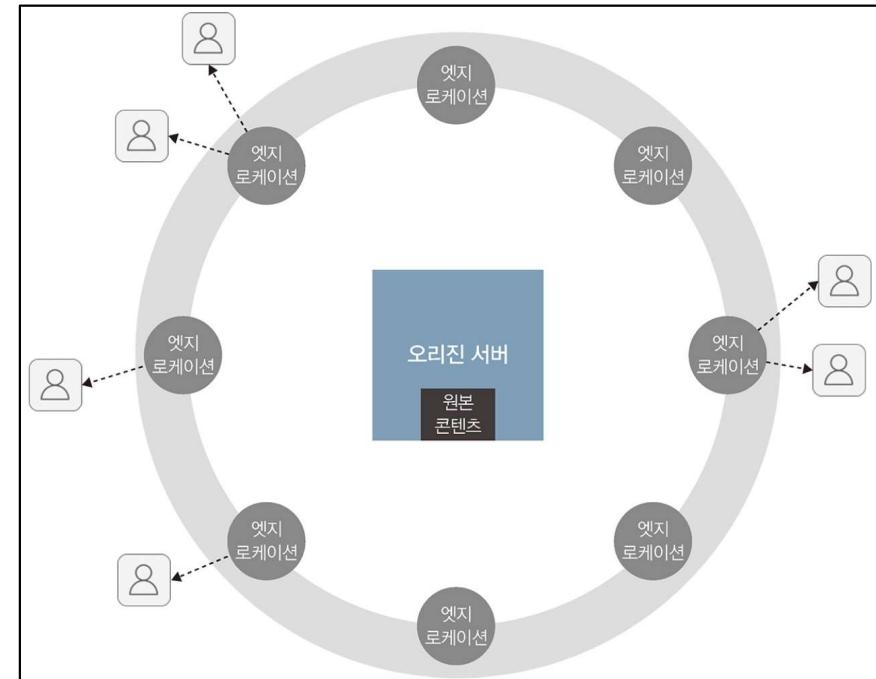
캐시 서버에 저장된 콘텐츠는 TTL로 지정된 시간 동안 저장되며, 시간 경과 시 삭제됨

AWS 고급 네트워킹 서비스

- CDN 캐싱 방식
 - 일반적인 캐싱 방식 두 가지
 - 정적 캐싱 : 정적 콘텐츠를 캐싱하는 것
 - 정적 콘텐츠 ; 변경되거나 수정되지 않는 콘텐츠를 의미
 - 예) 이미지 파일, JavaScript 코드, CSS 코드 등 웹 사이트의 레이아웃을 구성하는 콘텐츠는 대부분 정적 콘텐츠에 해당함
 - 별도의 사용자 요청이 없어도 오리지 서버에서 캐시 서버로 미리 콘텐츠를 복사할 수 있음
 - 동적 캐싱 : 동적 콘텐츠를 캐싱하는 것
 - 동적 콘텐츠 ; 사용자 요청이나 정보에 따라 즉석에서 생성되는 콘텐츠를 의미
 - 예) 사용자 정보를 활용하여, 매번 변경되는 형태의 콘텐츠
 - 동적 콘텐츠는 요청이 발생할 때마다 콘텐츠가 변경되는 특징이 있어서, 캐시 서버에서 콘텐츠를 보관하지 않고 Cache Miss 상태로 동작함
 - 캐시 서버는 콘텐츠를 저장하지 않으므로 매번 요청이 발생할 때마다 오리지 서버로 콘텐츠를 요청함
 - TTL=0(즉, 캐시에 저장되는 기간=0)으로 항상 지정하며, CDN의 이점을 살리기 어려움

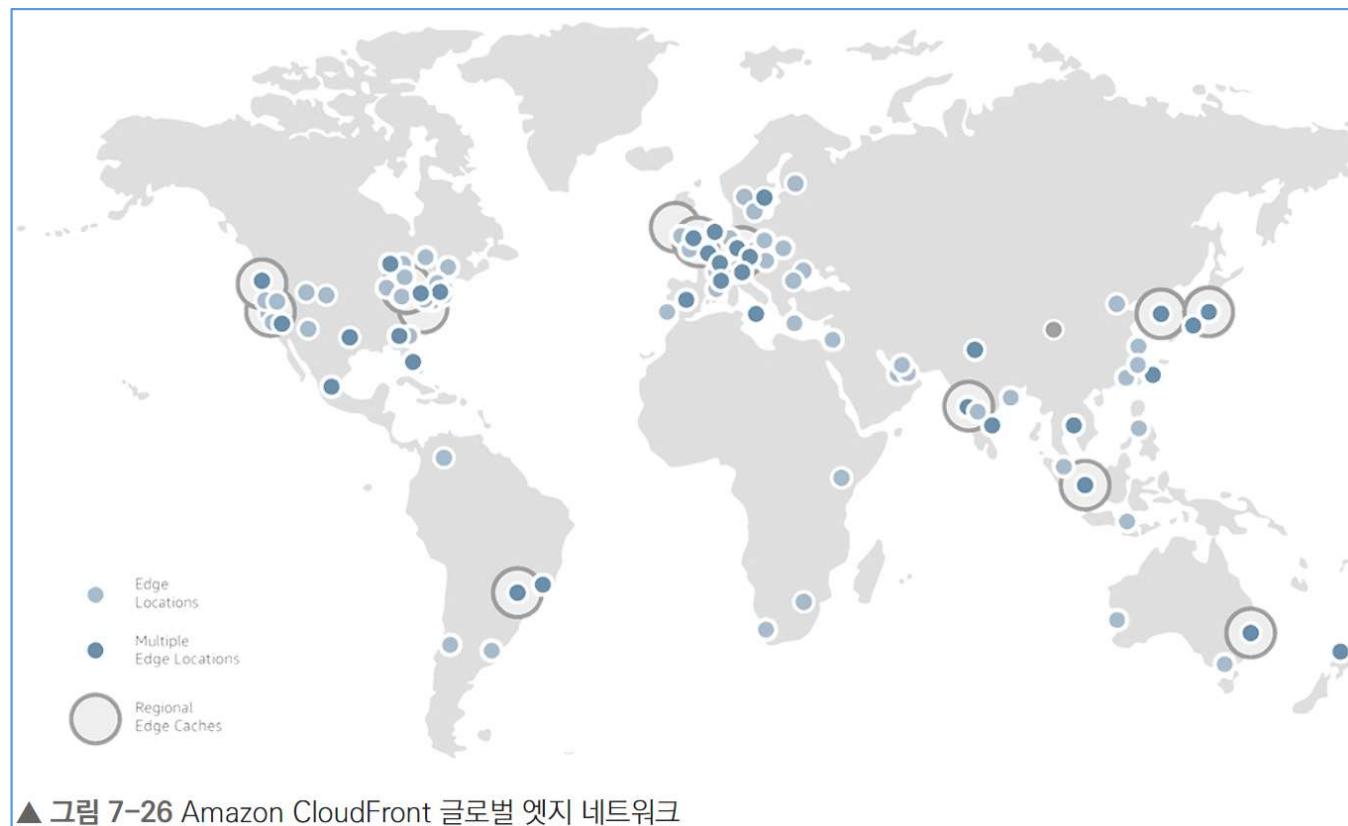
AWS 고급 네트워킹 서비스

- Amazon CloudFront
 - AWS에서 제공하는 CDN 서비스
 - 동적, 정적 콘텐츠를 사용자에게 빠르게 배포하도록 지원하는 서비스
 - 전 세계에 분포된 엣지 로케이션(Edge Location)이라는 곳에 콘텐츠를 캐싱하고 사용자 요청에 따라 가장 지연시간이 낮은 엣지 로케이션이 응답하여 최적의 응답시간 성능을 보장함



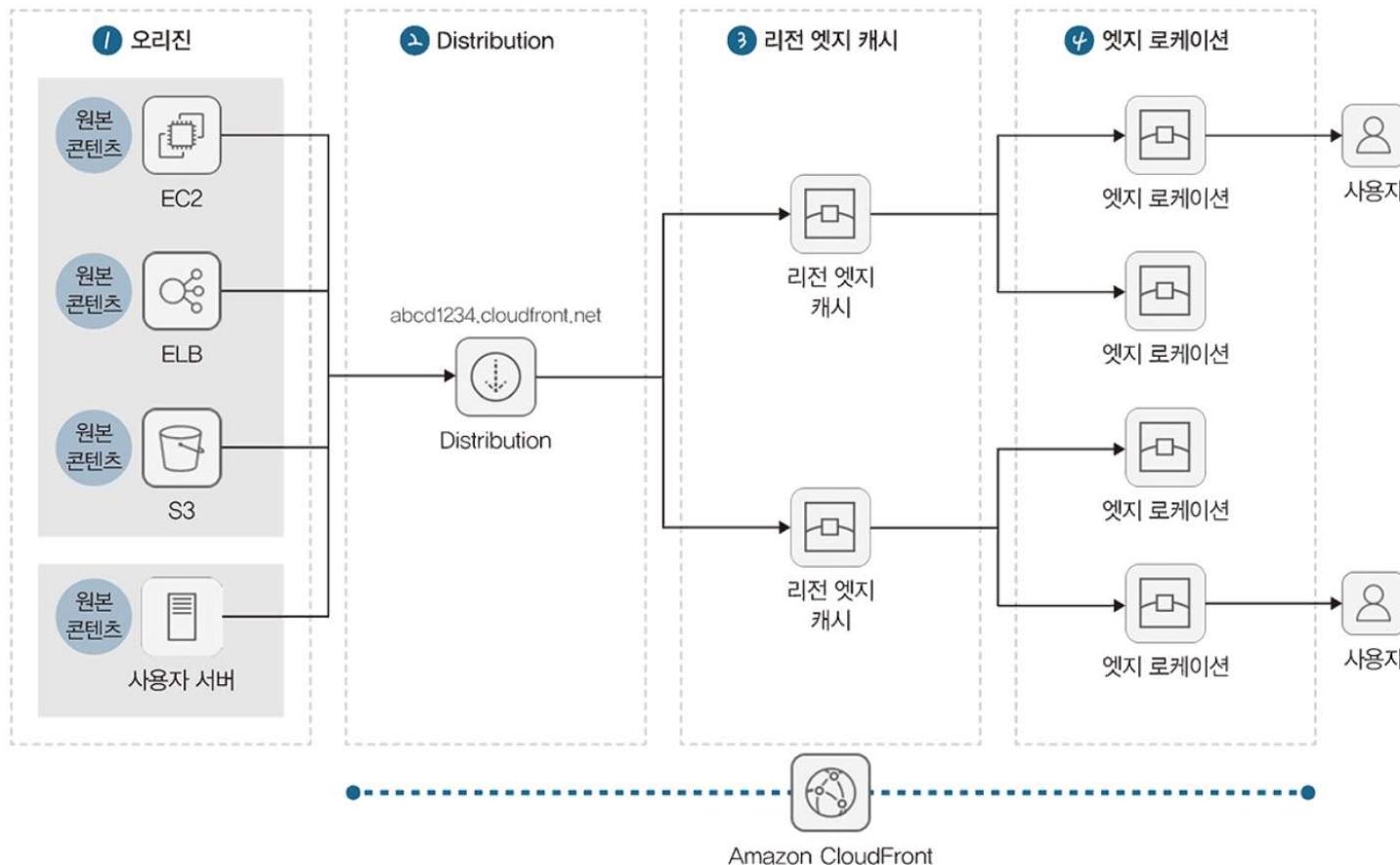
AWS 고급 네트워킹 서비스

- Amazon CloudFront 구성
 - AWS의 글로벌 엣지 네트워크를 이용하여 오리진 서버의 콘텐츠를 전 세계에 위치한 엣지 로케이션과 리전 엣지 캐시에 캐싱하여 CDN 서비스를 제공
 - 참고: Amazon CloudFront는 48개국 90개 이상의 도시에 450개 이상의 엣지 로케이션을 두고 AWS 글로벌 네트워크를 활용하여 서비스 중
 - <https://aws.amazon.com/ko/cloudfront/features/>



AWS 고급 네트워킹 서비스

- Amazon CloudFront 구성
 - 서비스 구성 요소



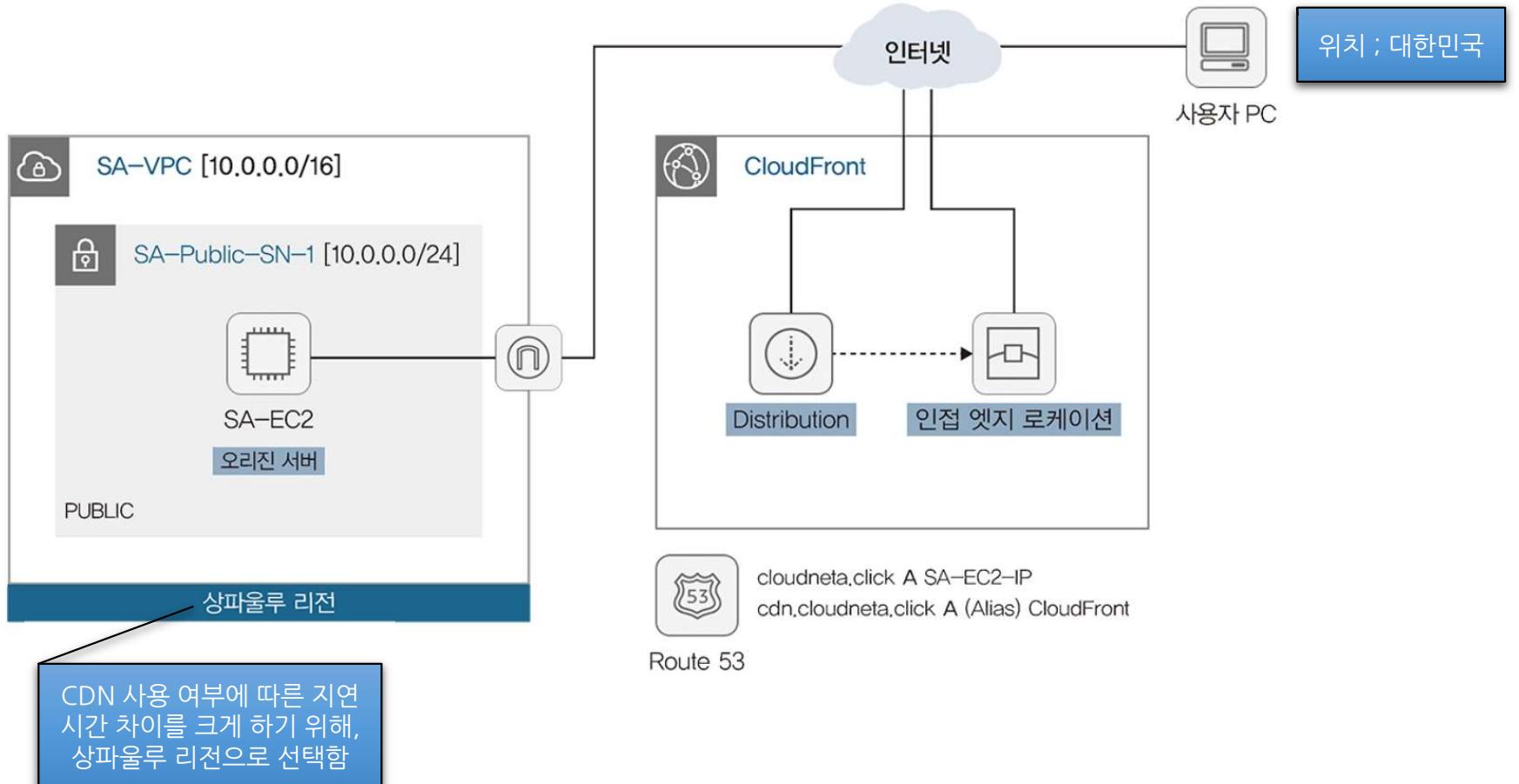
1. 오리진 ; 원본 콘텐츠를 가진 대상. 온 프레미스의 일반 서버나 AWS 서비스의 EC2, ELB, S3 등이 될 수 있음
2. Distribution ; 오리진과 엣지 중간에 서 콘텐츠를 배포하는 역할을 수행하는 CloudFront의 독립적인 단위로, 웹 서비스 전용의 Web Distribution과 스트리밍 전용의 RTMP Distribution으로 분류
3. 리전 엣지 캐시 ; 빈번하게 사용되는 콘텐츠에 대해 캐싱하는 큰 단위의 엣지 영역으로, 오리진과 엣지 로케이션 사이에 위치. 엣지 로케이션에서 오리진으로 콘텐츠를 요청하는 상황을 줄여서 효율적인 CDN 서비스를 제공
4. 엣지 로케이션 ; Distribution으로 배포되는 콘텐츠를 캐싱하는 작은 단위의 엣지 영역으로, 사용자 입장에서 가장 인접한 엣지 로케이션이 콘텐츠를 전달함

AWS 고급 네트워킹 서비스

- Amazon CloudFront 기능
 - 정적 및 동적 콘텐츠 처리
 - 정적 콘텐츠와 동적 콘텐츠에 최적화된 캐싱 동작을 지원
 - HTTPS 지원을 통한 보안 수준 개선 기능
 - 오리진 서버가 HTTPS를 지원하지 않아도 Amazon CloudFront가 알아서 HTTPS 방식으로 통신을 중계함
 - 즉, 사용자와 CloudFront는 HTTPS로 통신하고, CloudFront와 오리진은 HTTP로 통신
 - 다수의 오리진 선택 기능
 - 다수의 오리진을 지정하여 콘텐츠 분산 처리 가능
 - 접근 제어
 - 서명된 URL과 쿠키(cookie)로 사용자 인증을 지원하여 인증된 사용자만 접근할 수 있도록 제어 가능

AWS 고급 네트워킹 서비스

- Amazon CloudFront로 CDN 구성하기
 - 목표 구성도



AWS 고급 네트워킹 서비스

- Amazon CloudFront로 CDN 구성하기
 - 리전 변경 > 남아메리카(상파울루)



AWS 고급 네트워킹 서비스

- Amazon CloudFront로 CDN 구성하기
 - CloudFormation 으로 기본 인프라 배포하기 ; CloudFormation > 스택 > 스택 생성

```
1 Parameters:
2   LatestAmiId:
3     Description: (DO NOT CHANGE)
4     Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
5     Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami
6     AllowedValues:
7       - /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x
8
9 Resources:
10  SaVPC:
11    Type: AWS::EC2::VPC
12    Properties:
13      CidrBlock: 10.0.0.0/16
14      EnableDnsHostnames: true
15      Tags:
16        - Key: Name
17        Value: SA-VPC
18
19  SaIGW:
20    Type: AWS::EC2::InternetGateway
21    Properties:
22      Tags:
23        - Key: Name
24        Value: SA-IGW
25
26  SaIGWAttachment:
27    Type: AWS::EC2::VPCGatewayAttachment
28    Properties:
29      InternetGatewayId: !Ref SaIGW
30      VpcId: !Ref SaVPC
31
32  SaPublicRT:
33    Type: AWS::EC2::RouteTable
34    Properties:
35      VpcId: !Ref SaVPC
36      Tags:
37        - Key: Name
38        Value: SA-Public-RT
39
40  SaDefaultPublicRoute:
41    Type: AWS::EC2::Route
42    DependsOn: SaIGWAttachment
43    Properties:
44      RouteTableId: !Ref SaPublicRT
45      DestinationCidrBlock: 0.0.0.0/0
46      GatewayId: !Ref SaIGW
47
48  SaPublicSN1:
49    Type: AWS::EC2::Subnet
50    Properties:
51      VpcId: !Ref SaVPC
52      AvailabilityZone: !Select [ 0, !GetAZs
53      CidrBlock: 10.0.0.0/24
54      Tags:
55        - Key: Name
56        Value: SA-Public-SN-1
57
58  SaPublicSNRouteTableAssociation:
59    Type: AWS::EC2::SubnetRouteTableAssociation
60    Properties:
61      RouteTableId: !Ref SaPublicRT
62      SubnetId: !Ref SaPublicSN1
63
64  WEBSG:
65    Type: AWS::EC2::SecurityGroup
66    Properties:
67      GroupDescription: Enable HTTP access via port 80
68      VpcId: !Ref SaVPC
69      Tags:
70        - Key: Name
71        Value: WEBSG
72      SecurityGroupIngress:
73        - IpProtocol: tcp
74          FromPort: '80'
75          ToPort: '80'
76          CidrIp: 0.0.0.0/0
77        - IpProtocol: tcp
78          FromPort: '22'
79          ToPort: '22'
80          CidrIp: 0.0.0.0/0
81
82  SaEC2:
83    Type: AWS::EC2::Instance
84    Properties:
85      InstanceType: t2.micro
86      ImageId: !Ref LatestAmiId
87      Tags:
88        - Key: Name
89        Value: SA-EC2
90
91  NetworkInterfaces:
92    - DeviceIndex: 0
93      SubnetId: !Ref SaPublicSN1
94      GroupSet:
95        - !Ref WEBSG
96      AssociatePublicIpAddress: true
97
98  UserData:
99    Fn::Base64:
100      !Sub |
101        #!/bin/bash
102        (
103          echo "qwe123"
104          echo "qwe123"
105          ) | passwd --stdin root
106          sed -i "s/*PasswordAuthentication no/PasswordAuthentication yes/g"
107          sed -i "s/^#PermitRootLogin yes/PermitRootLogin yes/g" /etc/ssh/sshd_config
108          systemctl restart sshd
109          hostnamectl --static set-hostname Saupaulo-EC2
110          yum -y install httpd php tree
111          systemctl start httpd && systemctl enable httpd
112          wget -P /var/www/html/ https://cloudneta.github.io/test.jpg
113          curl -o /var/www/html/index.php
114          curl -o https://s3.ap-northeast-2.amazonaws.com/cloudformation.cloudneta.net/test.jpg
115          sed -i "s/UTC/Asia\Seoul/g" /etc/sysconfig/clock
116          ln -sf /usr/share/zoneinfo/Asia/Seoul /etc/localtime
117
118  Outputs:
119    SaEC2:
120      Value: !GetAtt SaEC2.PublicIp
121
```

대용량(10MB) 이미지 파일이
저장된 웹 서버용 EC2 배포

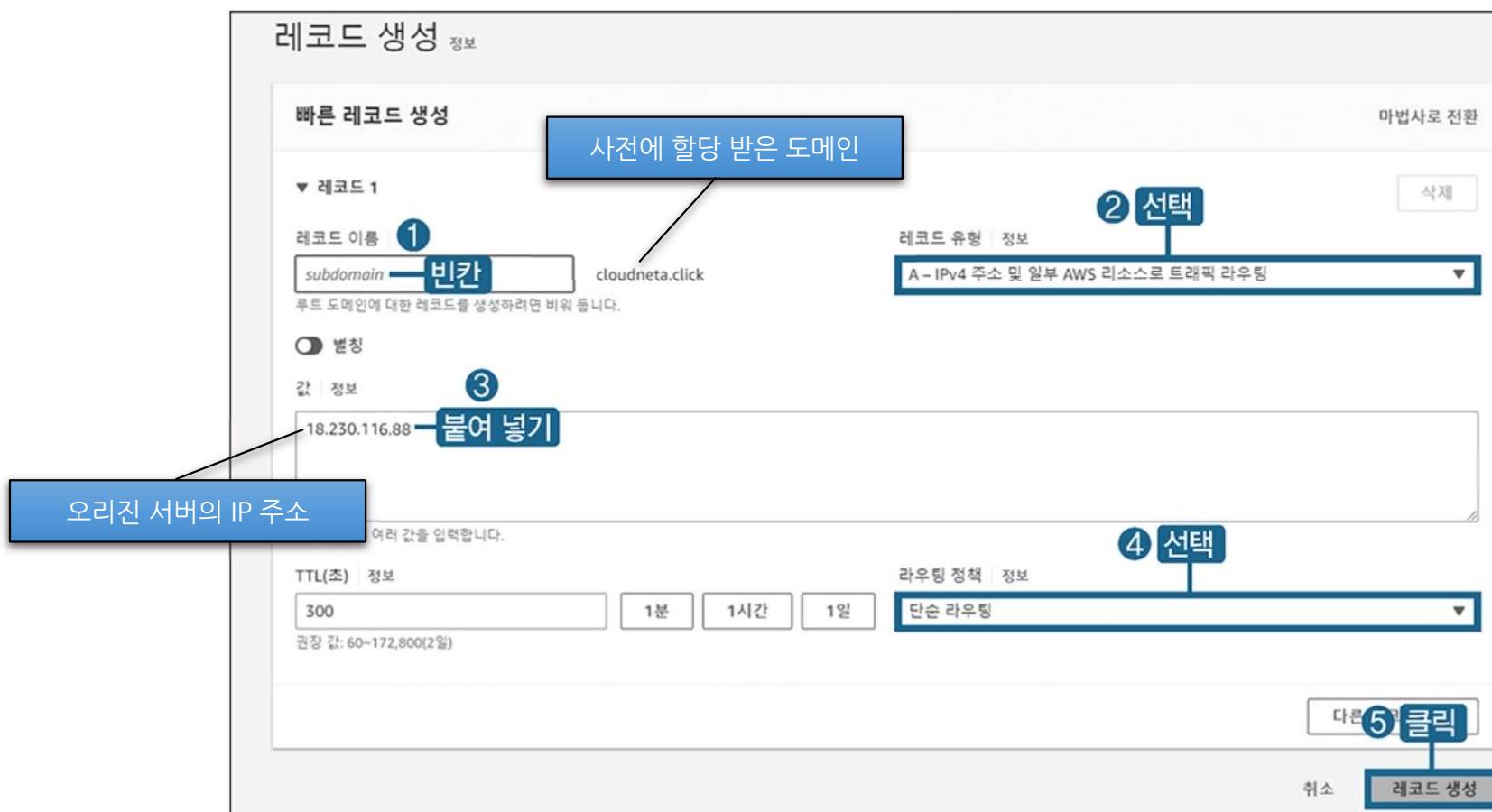
AWS 고급 네트워킹 서비스

- Amazon CloudFront로 CDN 구성하기
 - CloudFormation 으로 기본 인프라 배포하기
 - 생성된 인프라 자원 목록

생성 자원	이름	정보
VPC	SA-VPC	10.0.0.0/16
인터넷 게이트웨이	SA-IGW	SA-VPC에 연결
퍼블릭 라우팅 테이블	SA-Public-RT	0.0.0.0/0 → SA-IGW
퍼블릭 서브넷	SA-Public-SN-1	CH6-PublicRT 연결
보안 그룹	WEBSG	TCP 22/80 허용
EC2 인스턴스	SA-EC2	오리진 서버(웹 서비스)

AWS 고급 네트워킹 서비스

- Amazon CloudFront로 CDN 구성하기
 - Route 53 설정
 - (참고) SaEC2(オリジン 서버)의 IP가 18.230.116.88인 경우를 가정
 - (참고) cloudnera.click 도메인을 사전에 할당 받았다고 가정
 - DNS 레코드 생성

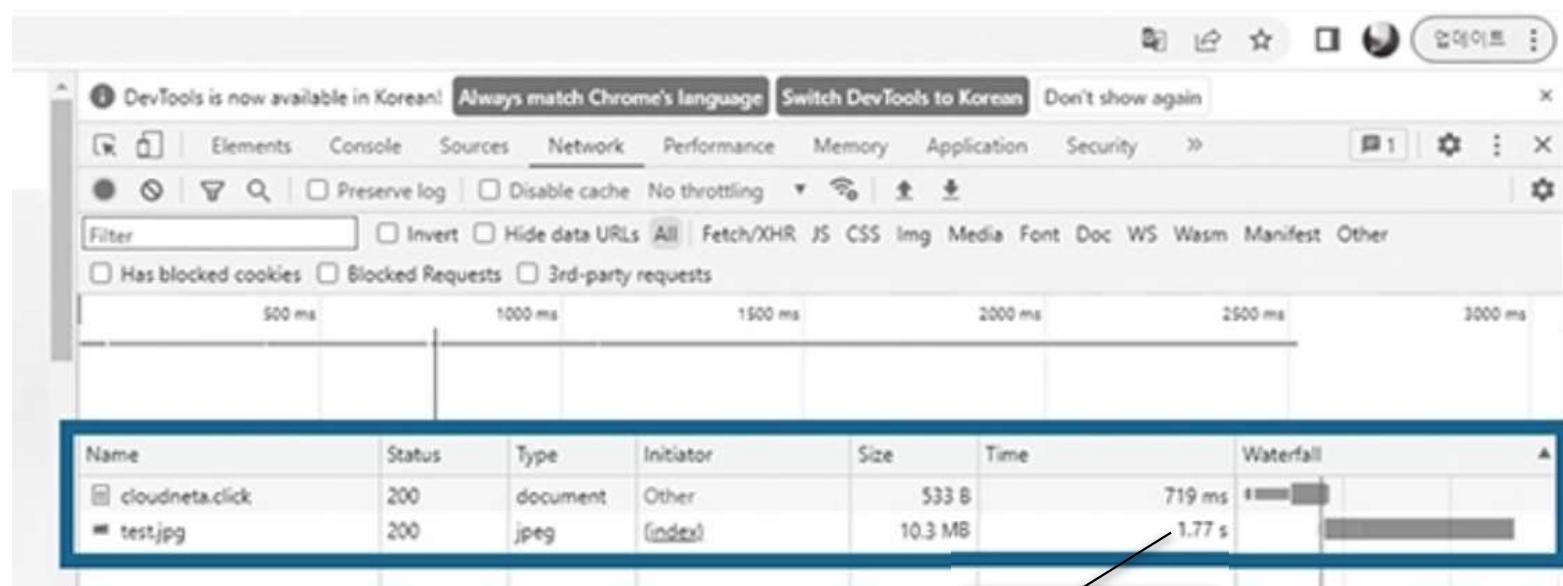


AWS 고급 네트워킹 서비스

- Amazon CloudFront로 CDN 구성하기
 - CDN을 사용하지 않고 오리진 서버에 접근하고 응답시간 확인
 - 등록한 도메인 주소로 웹 서비스 접근

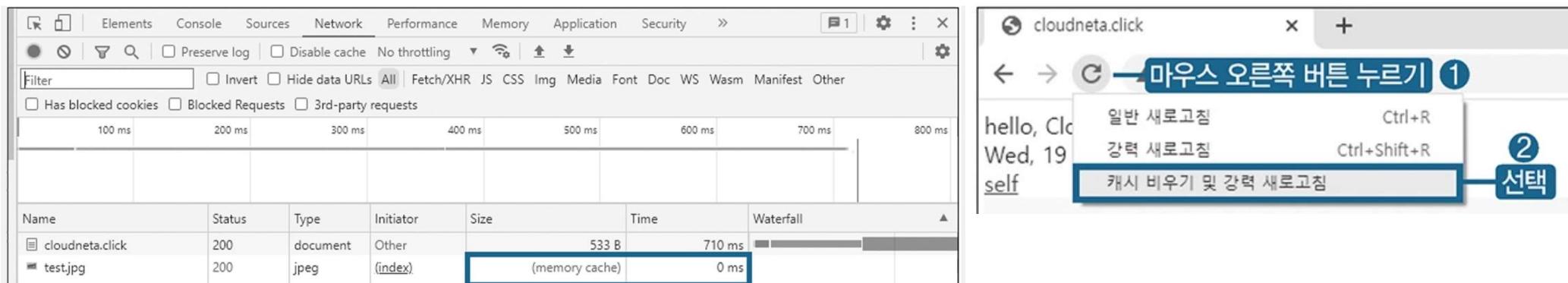


- 크롬 웹 브라우저 '개발자 도구'로 응답시간 확인
 - Network 탭에서 test.jpg 파일(약 10MB)을 수신하는데 걸린 시간 확인 : 1.77초



AWS 고급 네트워킹 서비스

- Amazon CloudFront로 CDN 구성하기
 - CDN을 사용하지 않고 오리진 서버에 접근하고 응답시간 확인
 - 크롬 웹 브라우저 개발자 도구로 응답시간 확인
 - 반복 측정을 통해서 평균 다운로드 시간 확인
 - (주의) 단순히 F5(Refresh)를 하면 로컬 PC에서의 메모리 캐시로 인해 응답시간이 0초로 측정됨(아래 사진). 매번 오리진 서버로부터 다운로드 받기 위해서는 Hard Refresh를 해야함



- 반복 측정한 응답시간 평균 ; 1.78s

접근 방식	1회	2회	3회	4회	5회	평균
인스턴스 직접 접속	1.77초	1.68초	1.82초	1.71초	1.91초	1.78초

AWS 고급 네트워킹 서비스

- Amazon CloudFront로 CDN 구성하기
 - CloudFront Distribution 생성하기



AWS 고급 네트워킹 서비스

- Amazon CloudFront로 CDN 구성하기
 - CloudFront Distribution 생성하기 (계속)

설정

가격 분류 | 정보
지불하려는 최고가와 연관된 가격 분류를 선택합니다.

모든 엣지 로케이션에서 사용(최고의 성능) 선택 7

북미 및 유럽만 사용

북미, 유럽, 아시아, 중동 및 아프리카에서 사용

AWS WAF 웹 ACL - 선택 사항
AWS WAF에서 웹 ACL을 선택하여 이 배포와 연결합니다.

웹 ACL 선택 ▼

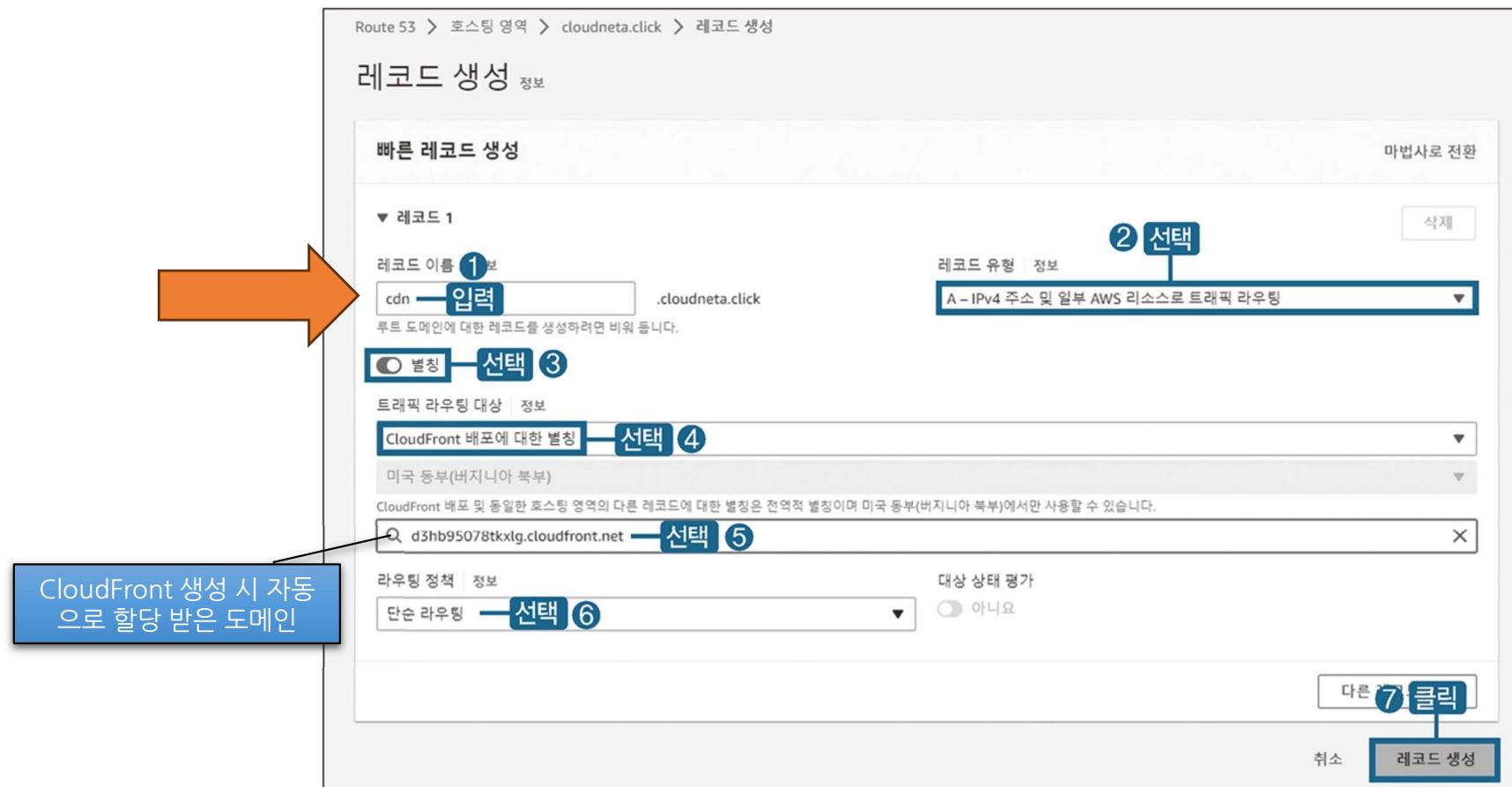
대체 도메인 이름(CNAME) - 선택 사항
이 배포에서 제공하는 파일에 대해 URL에서 사용하는 사용자 정의 도메인 이름을 추가합니다.

cdn.cloudnet.click 입력 8-2

제거

AWS 고급 네트워킹 서비스

- Amazon CloudFront로 CDN 구성하기
 - CloudFront Distribution 생성하기 (계속)
 - 직전에 생성한 대체 도메인 주소가 DNS 서비스에 응답되도록 DNS 레코드 신규 생성하기



AWS 고급 네트워킹 서비스

- Amazon CloudFront로 CDN 구성하기
 - 응답시간 재확인 (최초 접속 시)
 - Route 53 레코드에서 정의한 도메인 주소로 접근 (예: cdn.cloudneta.net)
 - 웹 브라우저로 최초 접근 시 응답시간 ; 4.6s
 - [엣지 로케이션에 요청 > Cache Miss > 오리진 요청] 과정 때문에 응답시간이 더 높게 나타남

The screenshot shows a NetworkMiner tool interface with the following details:

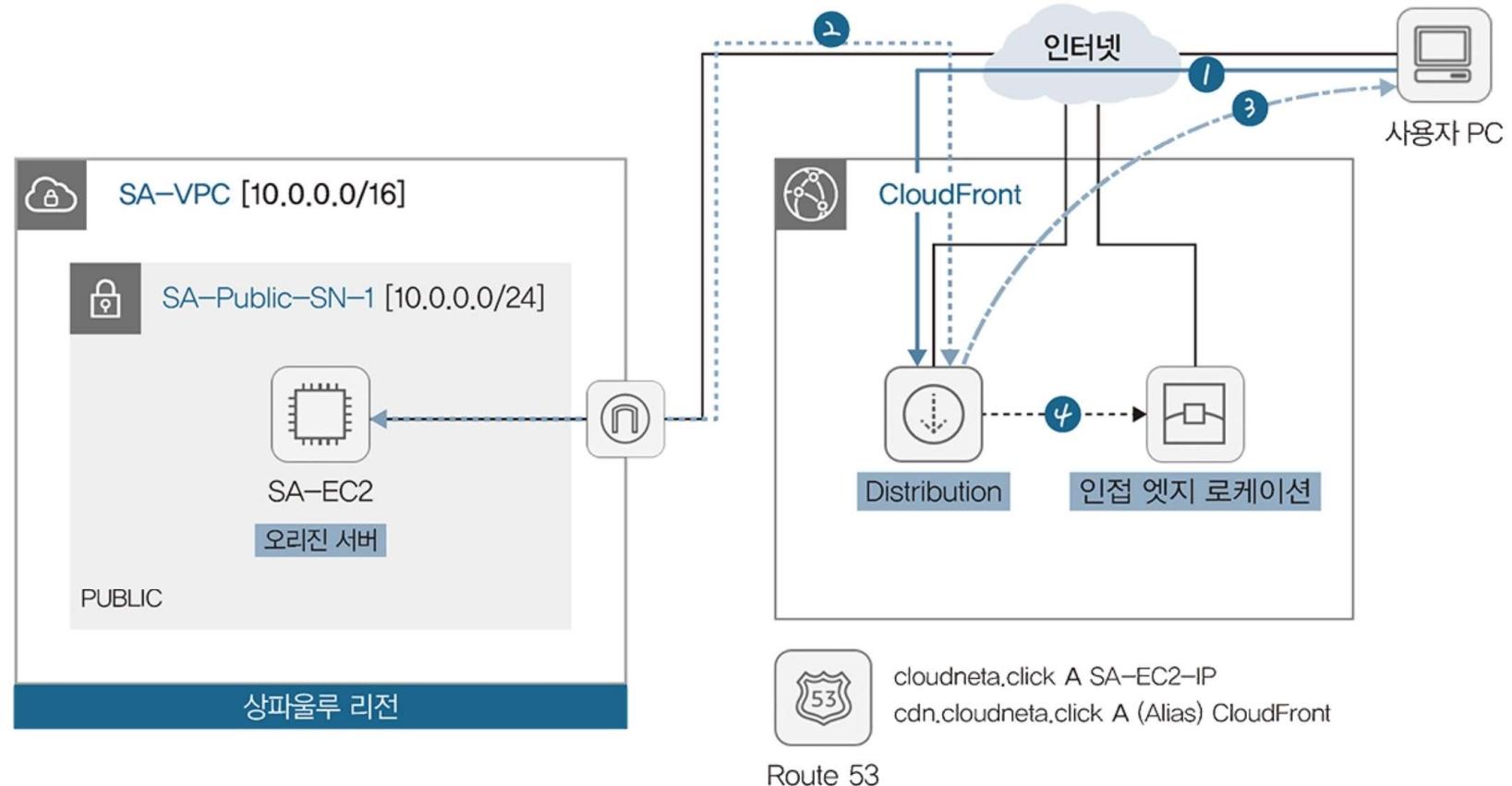
- Name:** cdn.cloudneta.click
- Request URL:** https://cdn.cloudneta.click/test.jpg
- Request Method:** GET
- Status Code:** 200
- Remote Address:** 18.64.8.57:443
- Referrer Policy:** strict-origin-when-cross-origin
- Response Headers:**
 - accept-ranges: bytes
 - content-length: 10297481
 - content-type: image/jpeg
 - date: Wed, 19 Apr 2023 17:55:56 GMT
 - etag: "9d2089-5ac89741b5a80"
 - last-modified: Mon, 10 Aug 2020 17:40:42 GMT
 - server: Apache/2.4.56 () PHP/5.4.16
 - via: 1.1 8cd680512056914f119efae770348786.cloudfront.net (CloudFront)
 - x-amz-cf-id: _RDZ209c8y5dvL3-5ZfuTFF6tpuQa_t7HJn2pVedtIUKWttZK2Nbg==
 - x-amz-cf-pop: ICN57-P2
 - x-cache: Miss from cloudfront

Annotations on the screenshot:

- ① 클릭: A callout points to the 'test.jpg' file entry in the tree view.
- ② 확인: A callout points to the 'x-cache: Miss from cloudfront' header value.
- Cache Miss 발생 확인: A blue box at the bottom left indicates the presence of a Cache Miss.

AWS 고급 네트워킹 서비스

- Amazon CloudFront로 CDN 구성하기
 - 전체 흐름도 (최초 접속 시)



▲ 그림 7-66 Amazon CloudFront Distribution의 최초 접속 통신 흐름

AWS 고급 네트워킹 서비스

- Amazon CloudFront로 CDN 구성하기
 - 응답시간 재확인 (재접속 시) // 캐시 된 정보 사용

The image shows two screenshots illustrating CloudFront metrics and CloudWatch Metrics Explorer.

Top Screenshot (CloudFront Metrics):

Name	Status	Type	Initiator	Size	Time	Waterfall
cdn.cloudneta.click	200	document	Other	546 B	854 ms	[Waterfall Bar]
test.jpg	200	jpeg	(index)	10.3 MB	939 ms	[Waterfall Bar]

An orange arrow points from the "Time" column of the "test.jpg" row towards the bottom screenshot.

Bottom Screenshot (CloudWatch Metrics Explorer):

1. Click on "test.jpg" in the list view.

2. Check the "x-amz-cf-pop" header value in the Response Headers section.

Details shown in the screenshot:

- General:**
 - Request URL: https://cdn.cloudneta.click/test.jpg
 - Request Method: GET
 - Status Code: 200
 - Remote Address: 54.192.175.129:443
 - Referrer Policy: strict-origin-when-cross-origin
- Response Headers:**
 - accept-ranges: bytes
 - age: 1635
 - content-length: 10297481
 - content-type: image/jpeg
 - date: Wed, 19 Apr 2023 17:55:56 GMT
 - etag: "9d2089-Sac89741b5a80"
 - last-modified: Mon, 10 Aug 2020 17:40:42 GMT
 - server: Apache/2.4.56 () PHP/5.4.16
 - via: 1.1 a6ab345505905317042e086e1f18d372.cloudfront.net (CloudFront)
 - x-amz-cf-id: vVrIQ_IN-Se1PMXF4plhK0QF1dYlcqMthv-3gnaaDeQ4fmG729yA==
 - x-amz-cf-pop: ICN55-C1
 - x-cache: Hit from cloudfront

AWS 고급 네트워킹 서비스

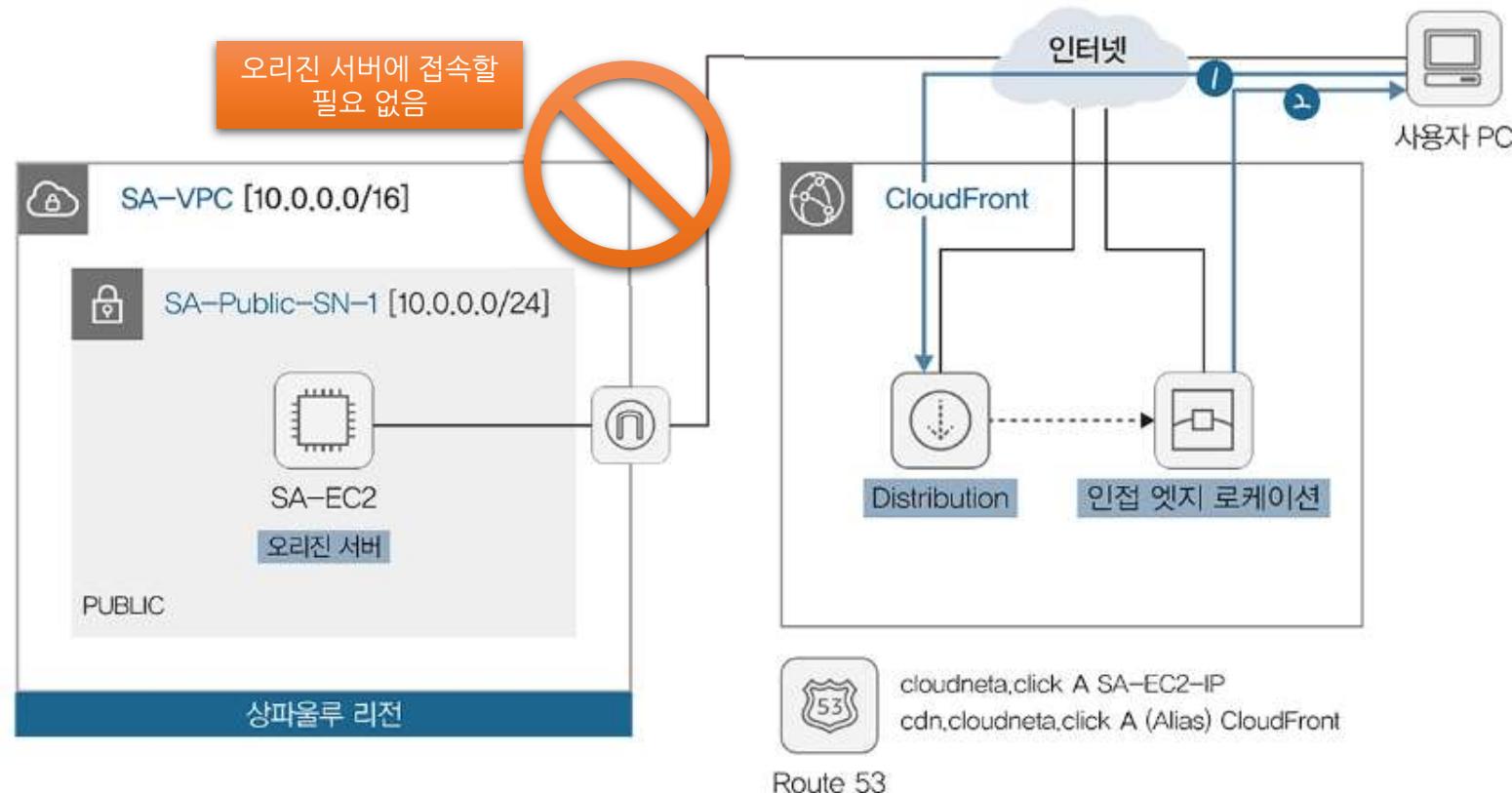
- Amazon CloudFront로 CDN 구성하기
 - 응답시간 재확인 (재접속 시) // 캐시 된 정보 사용
 - CDN 이용 전과 후 시간 비교

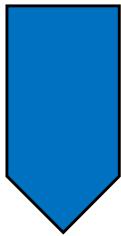
접근 방식	1회	2회	3회	4회	5회	평균
인스턴스 직접 접속	1.77초	1.68초	1.82초	1.71초	1.91초	1.78초
CloudFront로 접속	0.94초	0.89초	0.88초	0.88초	0.87초	0.89초



AWS 고급 네트워킹 서비스

- Amazon CloudFront로 CDN 구성하기
 - 전체 흐름도 (재접속 시) // 캐시 된 정보 사용





Amazon AWS 활용: IAM 서비스

IAM: 배경 소개

- AWS 리소스를 생성하고 관리하는 세 가지 방법
 - AWS 관리 콘솔 (AWS management console)
 - 웹 기반의 GUI 인터페이스를 사용해서 AWS 리소스를 관리하는 방법
 - AWS 관리 콘솔 주소는 <https://aws.amazon.com/ko/console/>



IAM: 배경 소개

- AWS 리소스를 생성하고 관리하는 세 가지 방법
 - AWS CLI 인터페이스
 - AWS CLI는 AWS 서비스를 관리하는 통합 도구로써, 사용자 컴퓨터에 설치한 후 터미널에서 명령어를 사용해서 AWS 리소스를 관리할 수 있음

AWS Command Line Interface

AWS Command Line Interface(AWS CLI)는 AWS 서비스를 관리하는 통합 도구입니다. 하나의 도구만 다운로드하여 구성하면 여러 개의 AWS 서비스를 명령줄에서 제어하고 스크립트를 통해 자동화할 수 있습니다.

AWS CLI v2는 개선된 설치 프로그램, AWS IAM Identity Center(AWS SSO의 후속 서비스)와 같은 새로운 구성 옵션, 다양한 상호작용 기능을 비롯한 여러 가지 새로운 기능을 제공합니다.



Windows

[64비트 Windows 설치](#) 프로그램을 다운로드해서 실행합니다.

MacOS

[MacOS PKG 설치](#) 프로그램을 다운로드해서 실행합니다.

Linux

[Linux 설치](#) 프로그램을 다운로드하고 압축을 해제한 다음 설치합니다.

로컬 PC에서 AWS CLI를 이용해서 클라우드 리소스를 조회/제어하는 예시

```
$ aws ec2 describe-instances
```

```
$ aws ec2 start-instances --instance-ids i-1348636c
```

IAM: 배경 소개

- AWS 리소스를 생성하고 관리하는 세 가지 방법
 - AWS SDK
 - AWS 리소스를 프로그래밍적으로 사용할 수 있도록 제공되는 라이브러리
 - AWS SDK는 Python, Go, Ruby, Java 등 주요 프로그래밍 언어를 지원함
 - Python SDK 예시:

Python용 AWS SDK(Boto3)

Python용 AWS SDK인 [boto3](#)를 사용하여 AWS를 빠르게 시작하십시오.
Boto3를 사용하면 Python 애플리케이션, 라이브러리 또는 스크립트를
Amazon S3, Amazon EC2, Amazon DynamoDB 등 AWS 서비스와 쉽게
통합할 수 있습니다.

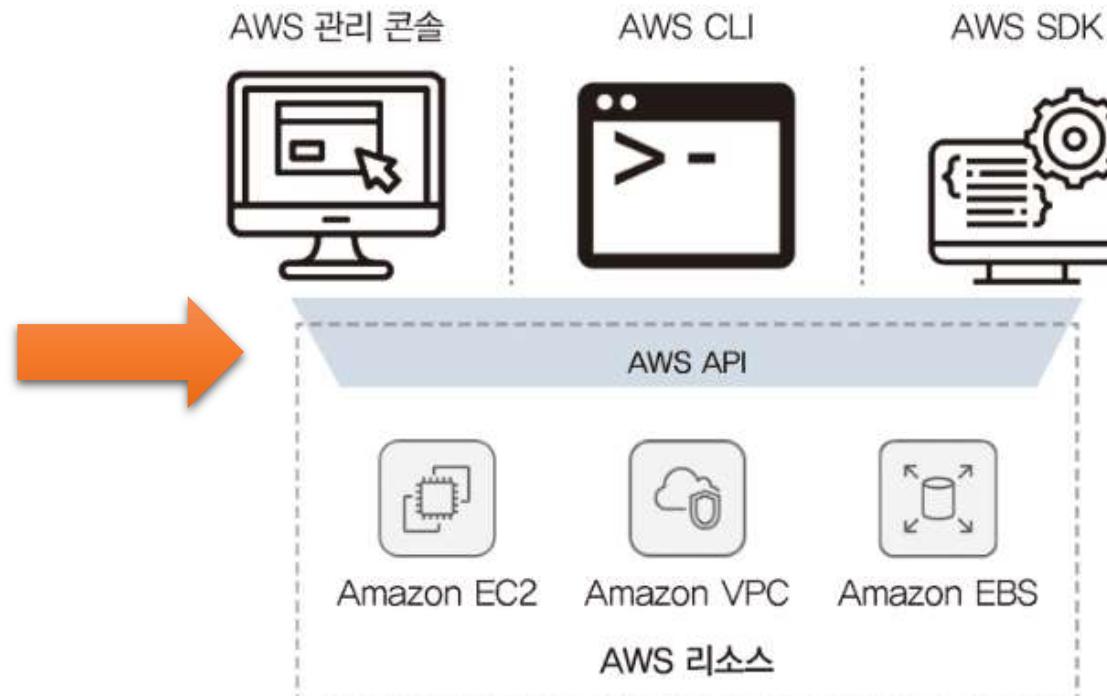
설치

```
pip install boto3
```

```
for i in ec2.instances.all():
    if i.state['Name'] == 'stopped':
        i.start()
```

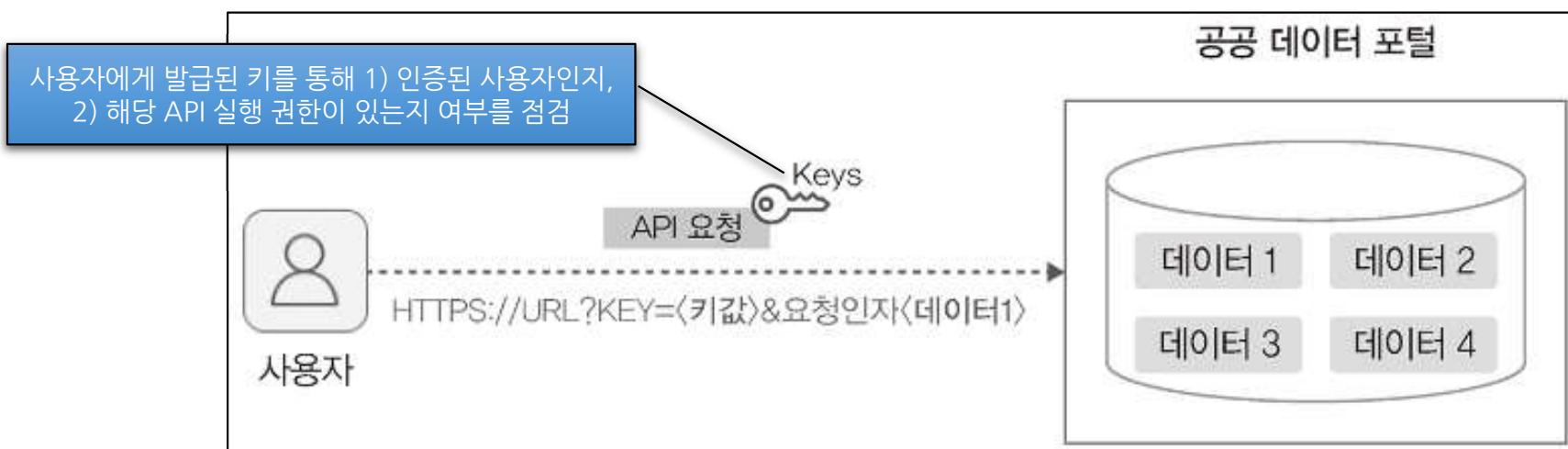
IAM: 배경 소개

- AWS 리소스를 생성하고 관리하는 세 가지 방법
 - 세가지 방법 모두 AWS API를 통해서 요청을 전송하고 응답을 수신함



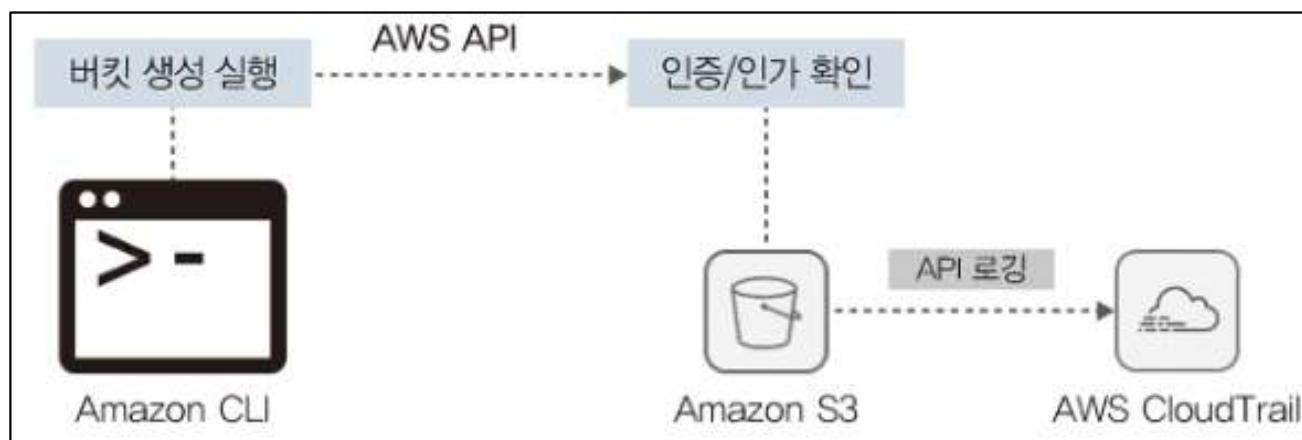
IAM: 배경 소개

- AWS API
 - API (Application Programming Interface)란 두 어플리케이션이 상호 작용할 수 있게 도와주는 매개체
 - AWS API ; 사용자 어플리케이션이 AWS 서비스를 사용할 수 있게 도와주는 매개체
 - 예: 자원 생성 요청 전달, 처리 결과 수신 등
- AWS API를 사용할 때, 인증과 인가 과정이 필수
 - ‘인증’(authentication) ; 인증된 사용자인지 확인
 - ‘인가’(authorization) ; 인증된 사용자의 실행 권한을 확인
- 공공 데이터 포털에 API로 접근하는 예시:

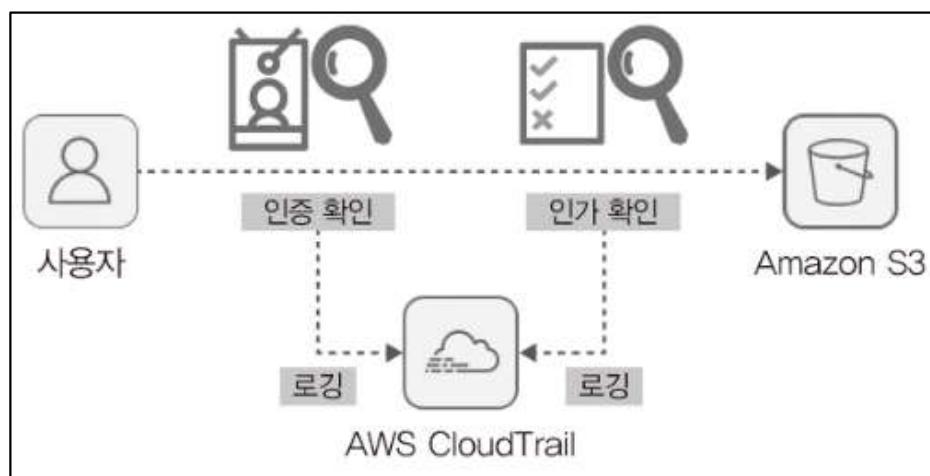


IAM: 배경 소개

- AWS API를 사용할 때, 인증과 인가 과정이 필수
 - 예: AWS CLI를 이용해서 S3 버킷을 생성하는 경우,



- 사용자가 CLI로 버킷 생성 명령을 실행하면, 해당하는 AWS API가 호출되어 서비스 요청을 서버로 전송
- 인증 및 인가 여부를 확인하고, S3 버킷을 생성할 권한이 있는 경우에만 버킷을 생성 함
- 인증/인가가 수행된 경우, 관련된 내용을 AWS CloudTrail 서비스를 이용하여 API 로깅을 남김
- API 로깅 ; AWS의 API 활동 기록을 저장 하는 것



AWS CloudTrail

- AWS 계정의 관리, 규정 준수, 운영 감사, 위험 감사 등을 지원하는 서비스
- AWS 인프라에서 계정 활동과 관련된 작업을 기록하고 지속적으로 모니터링할 수 있음

IAM: 배경 소개

- AWS IAM
 - IAM = Identity & Access Management
 - AWS 서비스와 리소스에 안전하게 접근할 수 있도록 인증과 권한을 관리/통제하는 기능
 - AWS 사용자 및 그룹을 만들고 관리하거나, 권한을 이용하여 AWS 리소스 접근을 허용하거나 거부할 수 있음
- IAM 구성 요소
 - AWS 계정 루트 사용자 ; 가장 처음에 생성한 AWS 계정으로, 해당 계정의 모든 권한을 가짐
 - IAM 사용자(user) ; 한 계정 내에서 권한을 부여 받은 사용자로, 각 사용자마다 서로 다른 권한을 부여할 수 있음
 - IAM 그룹(group) ; IAM 사용자 집합(다수의 IAM 사용자의 권한을 제어하는 용도)
 - IAM 정책(policy) ; 자격 증명이나 리소스와 연결될 때 요청을 허용하거나 거부할 수 있는 권한을 정의하는 AWS 객체
 - IAM 역할(role) ; 특정 권한을 가진 계정에 부여할 수 있는 IAM 자격 증명
 - 보안주체 (principals) ; AWS 계정 루트 사용자, IAM 사용자, IAM 역할을 이용하여 로그인하고 AWS에 서비스를 요청하는 사람 또는 어플리케이션

IAM: 배경 소개

- AWS IAM 사용자
 - '루트 사용자' 와 '일반 IAM 사용자'로 구분
 - 루트 사용자
 - AWS의 모든 리소스에 접근할 수 있는 전체 권한이 있는 사용자
 - AWS 계정 생성/해지 및 IAM 사용자 관리가 가능한 사용자
 - 일반적으로 루트 사용자 계정은 공유하지 않도록 관리하고, 사용자 관리 및 권한 관리만 수행하도록 하는 것이 일반적임(즉, AWS 리소스 관리는 일반 IAM 사용자가 수행하도록 함)
 - 일반 IAM 사용자
 - AWS 리소스 중 일부에 접근(생성/삭제/수정 또는 뷰어) 권한을 부여 받은 일반 사용자
 - AWS 관리 콘솔에 로그인 시, 사용자 종류를 선택할 수 있음



IAM: 배경 소개

- AWS IAM 사용자 사용 예
 - 모든 권한을 가진 admin 사용자와 View 권한만 가진 viewuser 사용자 추가하기
 - 모든 권한을 가진 admin 사용자 추가하기



IAM: 배경 소개

- AWS IAM 사용자 사용 예
 - 모든 권한을 가진 admin 사용자 추가하기

사용자 세부 정보 지정

사용자 세부 정보

사용자 이름 1
admin 입력

사용자 이름은 최대 64자까지 가능합니다. 유효한 문자: A~Z, a~z, 0~9 및 + = . @ _ -(하이픈)

AWS Management Console에 대한 사용자 액세스 권한 제공 – 선택 사항
사용자에게 콘솔 액세스 권한을 제공하는 경우 IAM Identity Center에서 액세스를 관리하는 것은 모범 사례입니다.

① 이 IAM 사용자를 생성한 후 액세스 키 또는 AWS CodeCommit이나 Amazon Keyspaces에 대한 서비스별 보안 인증 정보를 통해 프로그래밍 방식 액세스를 생성할 수 있습니다. 자세히 알아보기 []

2 클릭 취소 다음

IAM: 배경 소개

- AWS IAM 사용자 사용 예
 - 모든 권한을 가진 admin 사용자 추가하기

권한 설정

기존 그룹에 사용자를 추가하거나 새 그룹을 생성합니다. 직무별로 사용자의 권한을 관리하려면 그룹을 사용하는 것이 좋습니다. 자세히 알아보기

권한 옵션

1 선택

그룹에 사용자 추가
기존 그룹에 사용자를 추가하거나 새 그룹을 생성합니다. 그룹을 사용하여 직무별로 사용자 권한을 관리하는 것이 좋습니다.

권한 복사
기존 사용자의 모든 그룹 멤버십, 연결된 관리형 정책 및 인라인 정책을 복사합니다.

직접 정책 연결
관리형 정책을 사용자에게 직접 연결합니다. 사용자에게 연결하는 대신, 정책을 그룹에 연결한 후 사용자를 적절한 그룹에 추가하는 것이 좋습니다.

권한 정책 (1/1051)

새 사용자에 연결할 정책을 하나 이상 선택합니다.

C 정책 생성

2 체크

정책 이름	유형	연결된 엔터티
<input type="checkbox"/> AccessAnalyzerServiceRolePolicy	AWS 관리형	0
<input checked="" type="checkbox"/> AdministratorAccess	AWS 관리형 - 직무	1
<input type="checkbox"/> AdministratorAccess-Amplify	AWS 관리형	0

50

IAM: 배경 소개

- AWS IAM 사용자 사용 예
 - View 권한만 가진 viewuser 사용자 추가하기

사용자 세부 정보 지정

사용자 세부 정보

사용자 이름 **1**
viewuser **입력**

사용자 이름은 최대 64자까지 가능합니다. 유효한 문자: A~Z, a~z, 0~9 및 * = , . @ _ -(하이픈)

AWS Management Console에 대한 사용자 액세스 권한 제공 – 선택 사항
사용자에게 콘솔 액세스 권한을 제공하는 경우 IAM Identity Center에서 액세스를 관리하는 것은 모범 사례입니다.

① 이 IAM 사용자를 생성한 후 액세스 키 또는 AWS CodeCommit이나 Amazon Keyspaces에 대한 서비스별 보안 인증 정보를 통해 프로그래밍 방식 액세스를 생성할 수 있습니다. 자세히 알아보기

2 클릭
취소 **다음**

IAM: 배경 소개

- AWS IAM 사용자 사용 예
 - View 권한만 가진 viewuser 사용자 추가하기

권한 설정

기존 그룹에 사용자를 추가하거나 새 그룹을 생성합니다. 직무별로 사용자의 권한을 관리하려면 그룹을 사용하는 것이 좋습니다. 자세히 알아보기 [\[링크\]](#)

권한 옵션

① 선택

- 그룹에 사용자 추가
기존 그룹에 사용자를 추가하거나 새 그룹을 생성합니다. 그룹을 사용하여 직무별로 사용자 권한을 관리하는 것이 좋습니다.
- 권한 복사
기존 사용자의 모든 그룹 멤버십, 연결된 관리형 정책 및 인라인 정책을 복사합니다.
- 직접 정책 연결
관리형 정책을 사용자에게 직접 연결합니다. 사용자에게 연결하는 대신, 정책을 그룹에 연결한 후 사용자를 적절한 그룹에 추가하는 것이 좋습니다.

권한 정책 (1/1051)

새 사용자에 연결할 정책을 하나 이상 선택합니다.

② 체크

권한 경계 - 선택 사항

권한 경계를 설정하여 이 사용자에 대한 최대 권한을 제어합니다. 권한 관리를 다른 사용자에게 위임하는 데 사용되는 이 고급 기능을 사용합니다. 자세히 알아보기 [\[링크\]](#)

③ 클릭

C 정책 생성 [\[링크\]](#)

1 개 일자 < 1 > ⌂

필터 텍스트 또는 값을 기준으로 배포 필터링 viewonly X 필터 지우기

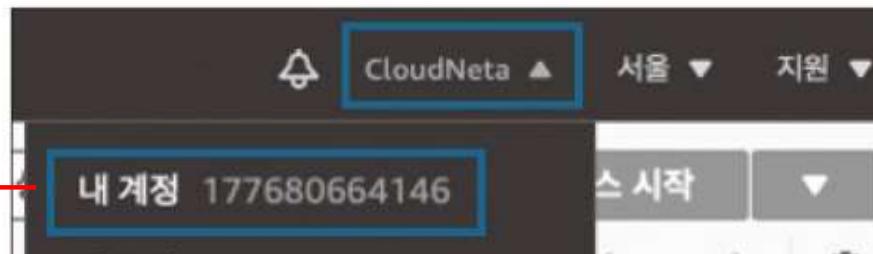
정책 이름	유형	연결된 엔터티
<input checked="" type="checkbox"/> ViewOnlyAccess	AWS 관리형 - 직무	0

취소 이전 다음

IAM: 배경 소개

- AWS IAM 사용자 사용 예

- AWS 계정의 ID 확인



- IAM 사용자 <admin> 로그인하기

로그인

● 루트 사용자
무제한 액세스 권한이 필요하 작업을 수행하는 계정 소유자입니다. 자세히 ① 선택

● IAM 사용자
일일 작업을 수행하는 계정 내 사용자입니다. 자세히 알아보기

계정 ID(12자리) 또는 계정 별칭
177680##### ② 입력

이 계정 기억하기 ③ 클릭

다음

IAM 사용자로 로그인

계정 ID(12자리) 또는 계정 별칭
177680#####

사용자 이름:
admin ④ 입력

암호:
***** ⑤ 클릭

로그인

같은 방식으로 viewuser로 로그인할 수 있음

IAM: 배경 소개

- AWS IAM 사용자 사용 예
 - IAM 사용자의 권한에 따른 동작 제한 확인하기
 - admin 사용자 ; 리소스 생성/삭제/재시작/중지 등 모든 활동에 제약이 없음
 - viewuser 사용자 ; 권한 부족으로 인한 오류 메시지가 생성됨
 - 아래는 EC2 인스턴스 종료 시도 시 실패하는 화면

☒ 인스턴스를 종료하지 못했습니다: You are not authorized to perform this operation.
f0lb8ST2nlNjQWj7TOFeb3m1CpgrouNpHKTj7kli9RpBb9pXxNNtWPO4otneNZIV
DuT248ljW1rV4ZacohwDZPKvi7SL__SKOV1BreoA3Sz755nHPTUey1_i7XeBFeZP

THE END

- 참고

- AWS 교과서, 김원일, 서종호, 김석필 지음, 길벗, 2023.10.20