Route 53

일시	2023.04.02. (일)	작성자	김민경

1. DNS

1) 개요

-人에게 친숙한 호스트 이름을 서버 ip 주소로 번역 (ex_www.google.com → 172.217.18.36)

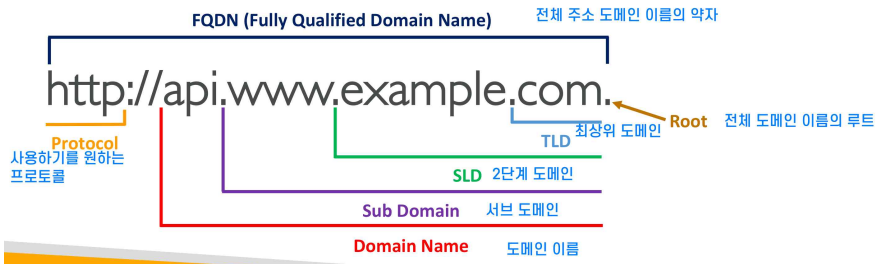
-URL과 호스트 이름을 ip로 번역

-계층적인 이름 구조

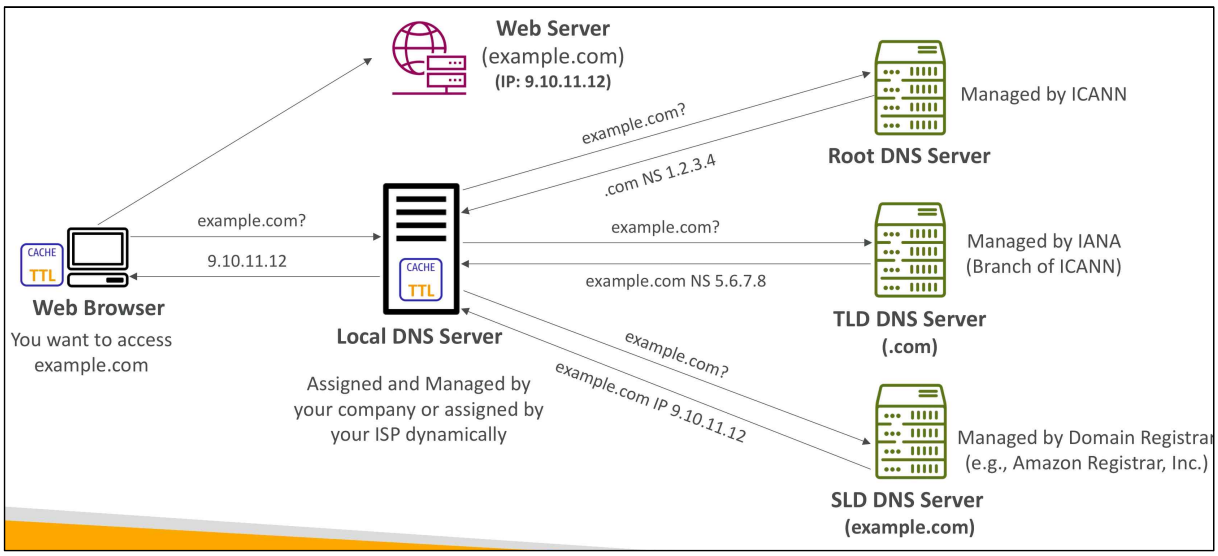
.com
example.com
www.example.com
api.example.com

2) 관련 용어

Domain Registrar (도메인 레지스트라)	우리가 도메인 이름 등록하는 곳 (ex_Route 53, GoDaddy..)
DNS Records	A, AAA, CNAME, NS...
Zone File	모든 DNS 레코드 포함함 (특정 도메인에 대한 모든 리소스 레코드의 필요한 정보가 모두 포함된 간단한 텍스트 파일)
Name Server	DNS 쿼리를 실제로 해결하는 서버
TLD (Top Level Domain)	.com, .us, .in, .gov...
SLD (Second Level Domain)	amazon.com, google,com...(2단어 사이에 . 이 있음)



3) DNS 작동



① Web Browser → Local DNS Server	- example.com 아니? - 몰라
② Local DNS Server → Root DNS Server	- example.com 아니? - 잘 몰라. 그런데 '.com'은 알아. 1.2.3.4.에 가봐
③ Local DNS Server → TLD DNS Server	- example.com 아니? - 잘 몰라. 그런데 5.6.7.8에 가봐.
④ Local DNS Server → SLD DNS Server	- example.com 아니? - 응. IP 9.10.11.12야.
⑤ Local DNS Server → Web Browser	- IP 9.10.11.12이라고 알려줌

• if 다음에도 누군가가 다시 example.com 물어본다면 바로 답변을 줄 수 0

2. Route 53

1) 개요

-HA, 확장성, 완전 관리형, 권한있는 DNS

-Domain Reistrar

-Route 53의 리소스 관련 상태 확인 0

-100% SLA 가용성 제공하는 유일한 AWS 서비스

***권한있다**
:고객인 우리가 DNS 레코드를 업데이트할 수 0
(DNS 완전히 제어 0)

***SLA**
:협약 당사자 간에 특히 서비스 제공자가 가입자에게 합의를 통해 사전에 정의된 수준의 서비스 제공하기로 협약 맺는 것

2) Records

-레코드를 통해 특정 도메인으로 라우팅하는 방법 정의

-각 레코드는 다음과 같은 정보 포함

Domain/subdomain Name	ex_ example.com
Record Type (레코드 종류)	ex_ A or AAAA
Value(레코드의 값)	ex_ 12.34.56.78
Routing Policy (라우팅 정책)	Route 53이 쿼리에 응답하는 방식
TTL	DNS Resolvers에서 레코드가 캐싱되는 시간

-레코드는 A, AAAA, CNAME, NS 등을 지원함

A	호스트 이름과 IPv4를 매핑
AAAA	호스트 이름과 IPv6를 매핑
CNAME	호스트 이름을 다른 호스트 이름과 매핑 상위 노드에 대한 CNAME 생성x
NS	호스팅 존의 이름 서버 트래픽이 도메인으로 라우팅되는 방식 제어

3) Host Zones

-레코드의 컨테이너,
도메인/서브도메인으로 가는 트래픽의 라우팅 방식을 정의

① Public Host Zones	-공개됨(누구든 쿼리 가능) -퍼블릭 도메인 이름을 살 때마다 퍼블릭 호스트 존 만들 수 0
② Private Host Zones	-공개되지 x는 도메인 이름 지원 (비공개)

-월 50센트씩 지불(무료x)

4) Records TTL(Time To Live)

-클라이언트가 재요청 or 같은 호스트 이름으로 접속할 경우 ⇒특정시간 동안 DNS 시스템에 쿼리 보내지x아도 0
(이미 답변을 캐시에 저장했기 때문에)

① High TTL (ex_24h)	-Route 53의 트래픽 현저히 적음 -오래된 레코드 받을 가능성 0
② Low TTL (ex_60sec)	-비용 ⇑ -오래된 레코드의 보관시간 ⇓

-all 레코드에 있어 필수적(Alias records는 제외)

5) CNAME vs Alias

-AWS 리소스(Load Balancer, CloudFront...) 사용하는 경우 ⇒ 호스트 이름 노출,
보유한 도메인에 호스트 이름을 매핑하고자 함

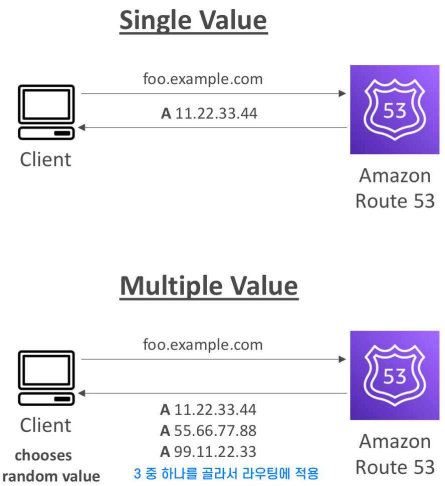
① CNAME	-호스트 이름이 다른 호스트 이름 향하도록 -루트 도메인 이름이 x닌 경우에만 가능 -Zone Apex라는 DNS 네임스페이스의 상위 노드로 사용 x
② Alias	-호스트 이름이 특정 AWS 리소스로 향하도록 -루트 & 비루트 도메인 all에 작동 -무료 -자체적 health check -Zone Apex라는 DNS 네임스페이스의 상위 노드로 사용 0 -AWS 리소스를 위한 Alias 레코드 타입은 A or AAAA, 리소스는 IPv4 or IPv6 中 하나 -TTL 설정x (ROUTE 53에 의해 자동 설정) -EC2 DNS name을 타겟할 수 x

*DNS 네임스페이스
:도메인 주소를 어떻게 관리할지에 대한 방법

6) Routing Policies

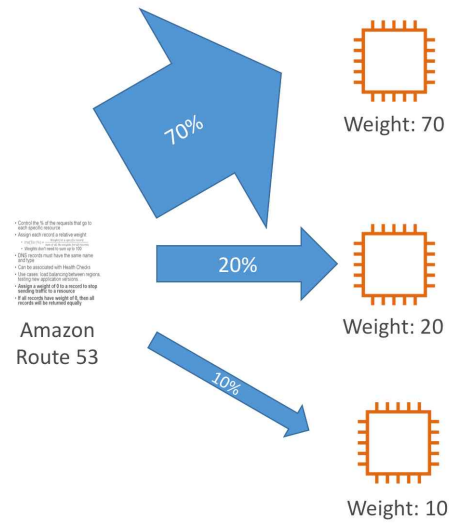
-Route 53이 DNS 쿼리에 응답하는 것 도와줌
-≠트래픽 라우팅 (DNS⇒ 호스트 이름들을 클라이언트가 실제 사용 가능한 엔드포인트로 변환하는 것 도와줌)

(1) Simple (단순 라우팅 정책)



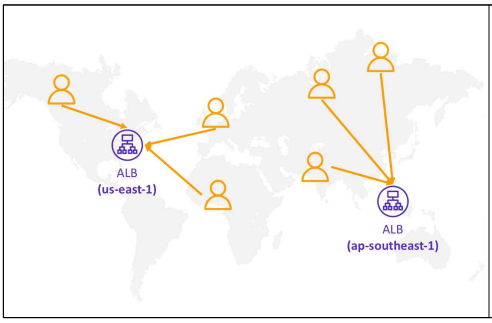
-일반적으로 트래픽을 단일 리소스로 보내는 방식
-동일한 레코드에 여러 개의 값을 지정 0
-DNS에 의해 다중 값을 받을 경우
⇒ 클라이언트 쪽이 무작위로 하나 선택
-단순 라우팅 정책에 Alias 레코드를 함께 사용하면
⇒ 하나의 AWS 리소스만을 대상으로 지정
-Health Checks x

(2) Weighted (가중치 기반 라우팅 정책)



-가중치를 활용해 요청의 일부 비율을 특정 리소스로 보냄
-각 레코드에 상대적으로 가중치 할당
- 트래픽(%) = 해당 레코드의 가중치/전체 가중치
- 합이 100이 x어도 됨
-DNS 레코드들은 동일한 이름 & 유형 가져야 함
-Health Checks과도 관련될 수 0
-Use cases: 서로 다른 지역들에 걸쳐 로드밸런싱 할 때
or 적은 양의 트래픽을 보내 새 app 테스트 할 때
-가중치 0 값을 보내면 ⇒ 특정 리소스에 트래픽 보내기를 중단해
가중치를 바꿀 수 0
-all 리소스 레코드 가중치 값이 0이면
⇒ all 레코드가 다시 동일한 가중치 갖게됨

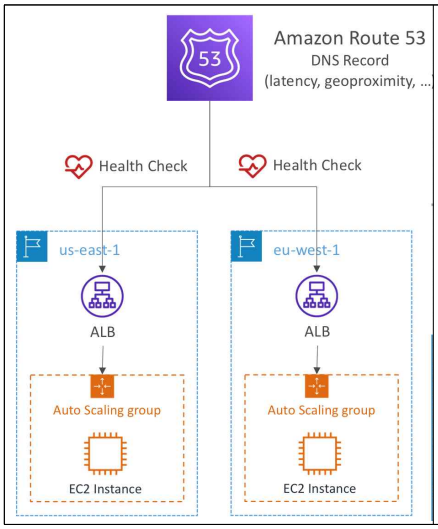
(3) Latency-based (지연 시간 기반 라우팅 정책)



-지연시간이 가장 짧은(가장 가까운) 리소스로 리다이렉팅
-지연시간에 민감한 웹사이트 or app이 있는 경우 유용
-지연시간 ⇒ 유저가 레코드로 가장 가깝다고 식별된 AWS 리전에 연결하기까지 걸리는 시간을 기반으로 측정됨
-Health Checks 연결 0

* **Health Checks**

-오직 public resources에 대한 상태 확인하는 방법



① 공용 엔드포인트 모니터링 (app, 서버 등)
② 다른 Health Checks 모니터링
(Calculated Health Checks: 계산된 상태확인)
③ CloudWatch Alarms 상태 모니터링
(제어가 쉽고, private 리소스에 유용)

-CloudWatch 메트릭에서도 확인 가능

-특정 엔드포인트에 작동하는 방법

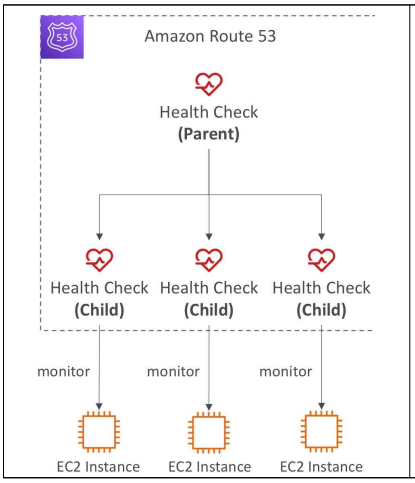
- 전세계에서 온 15개의 health cherks가 엔드포인트 상태를 확인
  -임계값을 정상 or 비정상으로 설정
  -30초마다 정기적 확인(10초마다 할 수도 0 -빠른 상태확인(비용⇑))
  -18% 이상의 health checkers가 엔드포인트를 정상이라 판단 시 ⇒ Route 53도 정상으로 간주
  그렇지 않으면 ⇒ 비정상으로 간주
  -health checks에 사용될 위치 선택0

-로드밸런서로부터 2xx or 3xx의 코드를 받아야 통과됨

-텍스트 기반 응답일 경우 ⇒ 응답의 처음 5120 bytes 확인함 (∵응답자체에 해당 테스트 있는지 보기 위해)

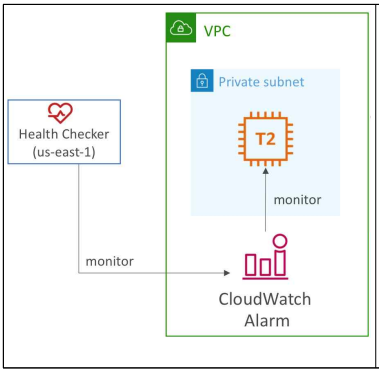
-Route 53의 Health Chckers로 들어오는 모든 요청을 허용해야 함
(∵Health Checks 가능하려면 애플리케이션 밸런서 or 엔드포인트에 접근 가능해야 함)

-Calculated Health Checks (계산된 상태확인)



- 여러개의 Health Check의 결과를 하나로 합쳐주는 기능
- Health Check 결과 합치기 위한 조건 ⇒ OR, AND, NOT 사용
- 하위 상태 확인을 256개까지 모니터링 0
  상위 상태 확인이 통과하기 위해 몇 개의 상태 확인이 통과할지 선택 0
- Usage: health check가 실패하는 일 x이
  상위 상태 확인이 웹사이트를 관리 유지하는 경우

-Private Host Zones (private 리소스 상태확인 하는 방법)



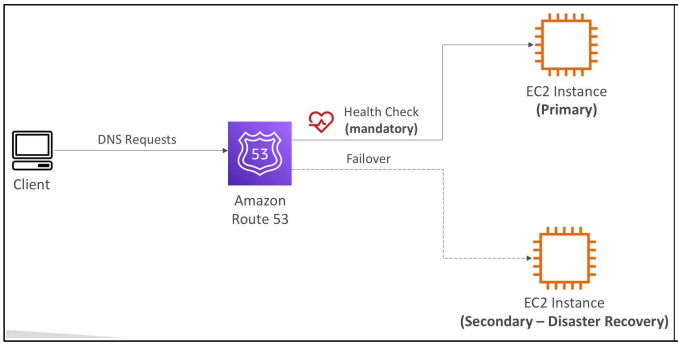
- Route 54 health checkers ⇒ VPC 외부에 있음
- Private 엔드포인트에 액세스x

∴CloudWatch Metric 만들어 CloudWatch Alarm 할당

-CloudWatch Metric 이용해 private subnet 안에 있는 EC2 인스턴스 모니터링

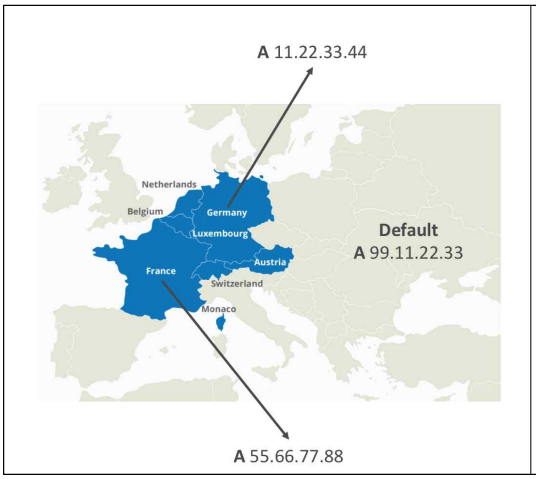
-Metric 침해되면 CloudWatch Alarm → health checks 자동으로 비정상

(4) Failover (Active-Passive, 장애조치)



- Health Check 필수적
- 먼저 기본 레코드 연결
  →비정상일 시 2번째 인스턴스로 자동 장애 조치
- 기본 & 보조 인스턴스 각각 하나씩만 있을 수 0

(5) Geolocation (지리 위치)



- 실제 사용자 위치 기반
- 가장 정확한 위치가 선택되어 그 IP로 라우팅
- 일치하는 위치 x는 경우 ⇒ Default 레코드 생성해야 함
- Use cases: 콘텐츠 분산 제한하고 로드밸런싱 등 실행하는 웹사이트 현지화
- Health Checks와 연결 0

(6) Geoproximity (지리 근접 라우팅)

-사용자 & 리소스의 지리적 위치 기반

-bias(편향값) 사용해 특정 위치에 더 많은 트래픽 이동

(1~99: 편향값 ⇑/ -1~-99: 편향값⇓)

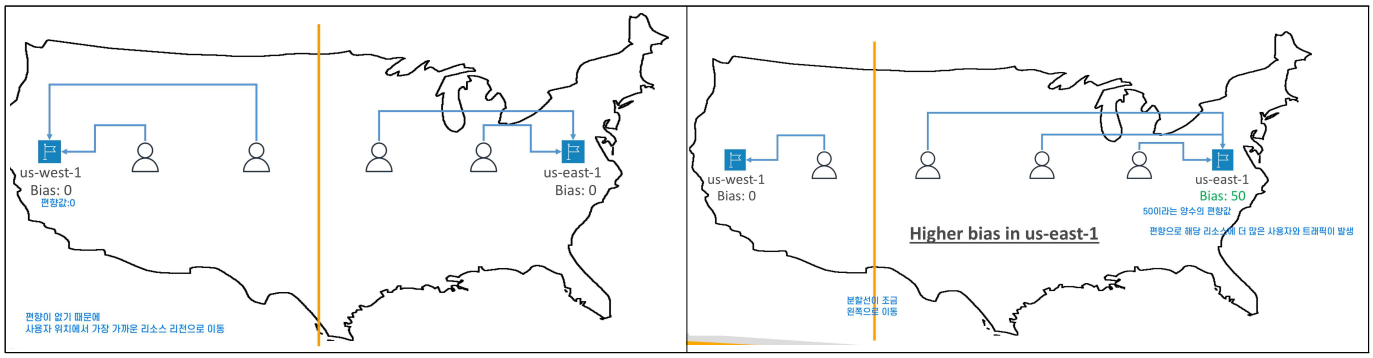
-리소스

AWS 리소스	AWS 리전 지정해 위치 파악
AWS 리소스 x (온프레미스 데이터 센터)	위도, 경도 지정해 위치 파악

-편향 활용 위해 고급 Route 53 Traffic Flow 사용

-편향을 증가시켜 한 리전에서 다른 리전으로 트래픽 보낼 때 유용

***Traffic Flow**
:대규모 및 복잡한 구성에서 레코드 만들고 유지 관리하는 프로세스 간소화



(7) Multi-Value (다중 값 라우팅 정책)

-트래픽을 다중 리소스로 라우팅할 때 사용

-Route 53은 다중값과 리소스를 반환함(요청에 대해 여러 리소스 값 반환 0)

-Health Checks와 연결 시 ⇒ 각 리소스의 상태 확인 가능

∴Route 53은 정상 리소스의 값만 반환

-로드밸런싱을 개선한 방법(ELB를 대체하진 x)

7) Domain Registar vs DNS Service

-Domain Registar 통해 원하는 도메인 이름 구매 0 (매년 비용 지불)

-Domain Registar 통해 도메인 등록 시 ⇒ DNS 레코드 관리 위한 DNS 서비스 제공

-DNS 레코드로 다른 DNS 서비스 사용 0 (ex_GoDaddy...) + DNS 서비스 제공자로 Route 53 사용 0

① Route 53에서 공용 Hosted Zone 생성

② 도메인을 구매한 타사 웹사이트에서 Name Servers 업데이트 → Route 53 이름 서버 가리킴

-도메인 이름 레지스트라는 모두 비슷해보이지만 DNS 서비스가 다름

(but all Domain Registar ⇒ 보통 일부 DNS 기능 제공)