Cloudformation 과제계획서

인턴 주진우

구성도

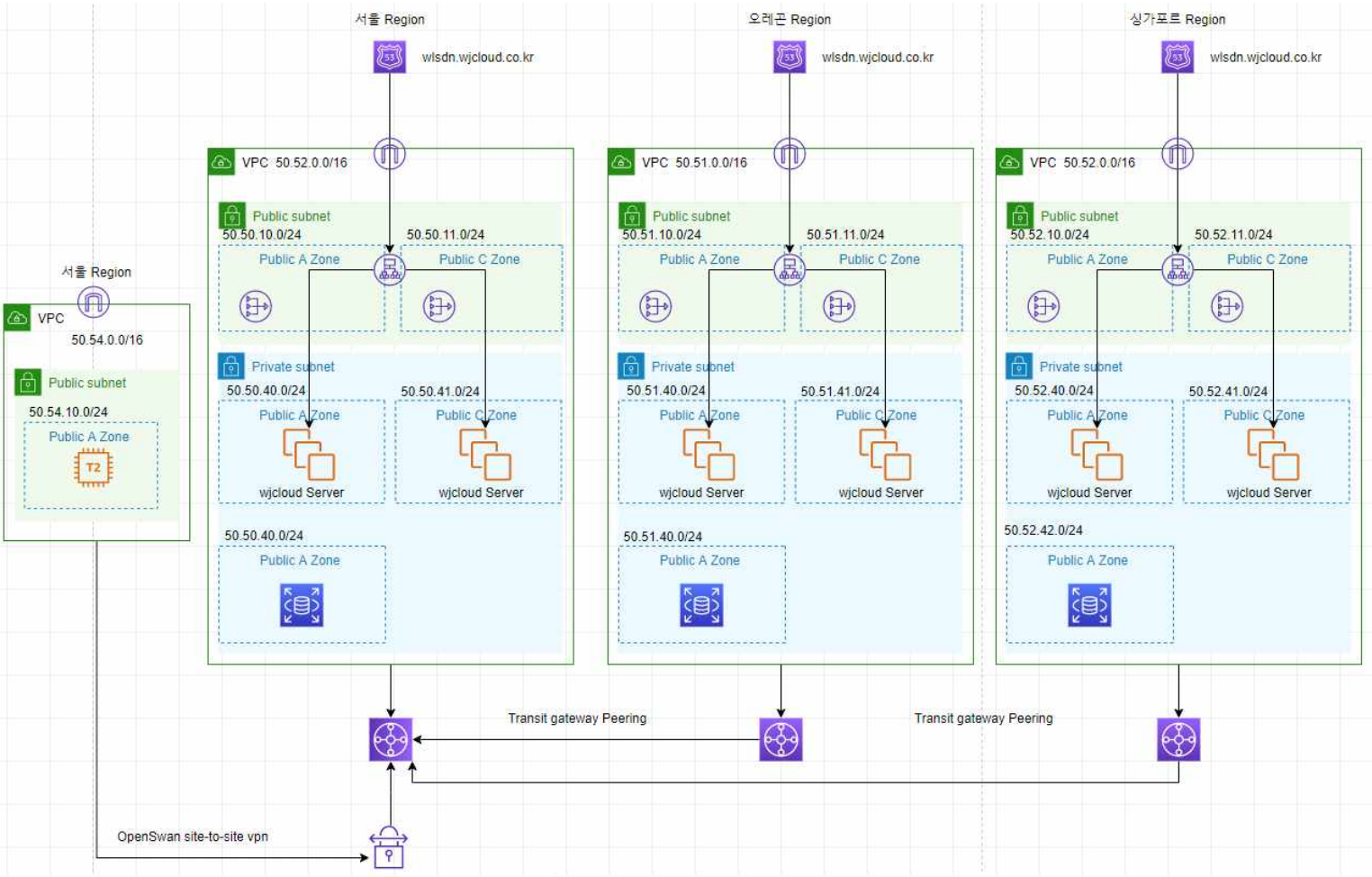
서울 Region
wlsdn.wjcloud.co.kr
VPC 50.52.0.0/16
Public subnet
50.50.10.0/24
50.50.11.0/24
Public A Zone
Public C Zone
Private subnet
50.50.40.0/24
50.50.41.0/24
wjcloud Server
50.50.40.0/24
Public A Zone
서울 Region
VPC
50.54.0.0/16
Public subnet
50.54.10.0/24
Public A Zone
T2
오레곤 Region
wlsdn.wjcloud.co.kr
VPC 50.51.0.0/16
Public subnet
50.51.10.0/24
50.51.11.0/24
Public A Zone
Public C Zone
Private subnet
50.51.40.0/24
50.51.41.0/24
wjcloud Server
50.51.40.0/24
Public A Zone
싱가포르 Region
wlsdn.wjcloud.co.kr
VPC 50.52.0.0/16
Public subnet
50.52.10.0/24
50.52.11.0/24
Public A Zone
Public C Zone
Private subnet
50.52.40.0/24
50.52.41.0/24
wjcloud Server
50.52.42.0/24
Public A Zone
Transit gateway Peering
Transit gateway Peering
OpenSwan site-to-site vpn

그림1. 구성도

I. 목표

1. cloudformation을 통한 리전간 인프라 구축 자동화

II. 개요

2-1) 모 기업은 현재 한국, 미국, 싱가폴에서 동일한 웹 서비스를 제공하고 있다고 가정한다.

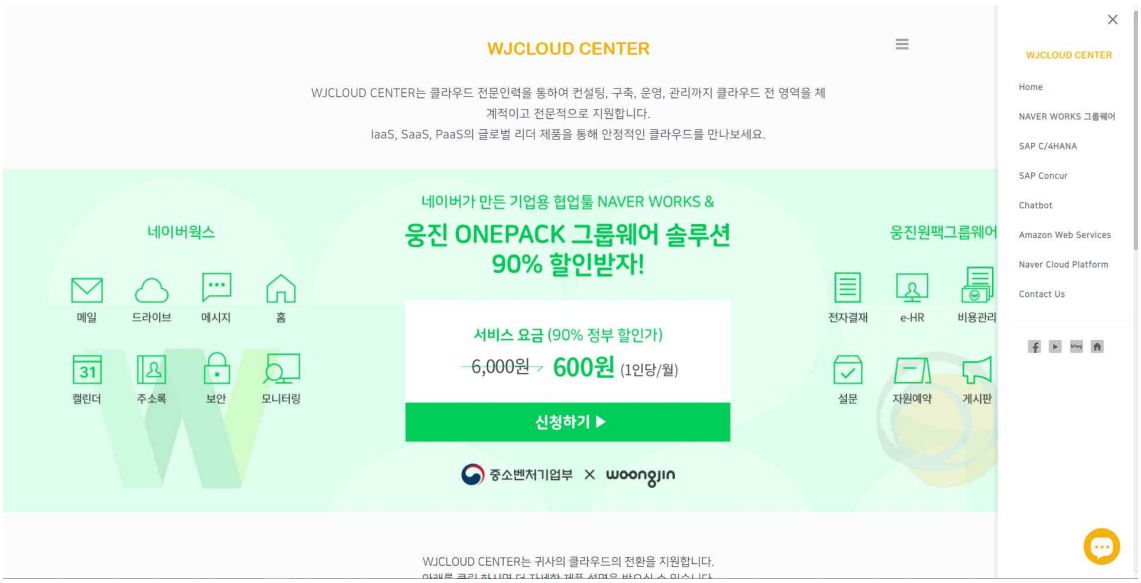

그림2. wjcloud.co.kr

2-2)

VPC를 생성하고 Subnet, igw, ngw, security group, Route를 생성한다.
center database가 생성되어있는 Mysql5.7 버전 RDS 스냅샷을 리전별로 공유하고, RDS를 생성한다.
Tomcat 설치가 끝난 AMI를 리전별로 공유하고 인스턴스를 생성한다.
Load balancer를 생성하고 TargetGroup과 Listener를 생성한다.
Route53 Record를 생성, 지연 시간 기반 라우팅을 사용하여 지연 시간에 대한 데이터를 참조하여 서울리전의 Load balancer 또는 오레곤리전의 Load balancer, 싱가포르 리전의 Load balancer중 지연시간이 짧은 Load balancer로 라우팅한다.

IDC 환경과 Transit gateway 간의 site-to-site VPN을 구현하기 위해서 50.54.10.0/16 VPC의 퍼블릭 인스턴스에 OpenSwan을 설치하여 가상의 On-premise 환경을 구성한다.
TransitGateway를 리전마다 생성하고, PeeringConnection을 사용하여 서로 다른 리전과 연결한다.

2-3) 일련의 과정들을 4개의 Cloudformation yaml 파일로 작성한다.

- Cloudformation_Seoul.yaml
- Cloudformation_Oregon.yaml
- Cloudformation_Singapore.yaml
- Cloudformation_OpenSwan.yaml

III. 이점

3-1) Cloudformation을 사용함으로써 AWS 리소스를 간편하게 프로비저닝하고, 여러개의 스택으로 구성하고 관리 할 수 있다. 또한 신속하게 인프라 복제가 가능하다.

3-2) 지연 시간 기반 라우팅을 통해 최종 사용자에게 최저 지연 시간을 제공하는 엔드포인트로 라우팅(Routing)을 제공하여 리전 사용자에게 제공하는 서비스 품질을 높일 수 있다.

3-3) TransitGateway를 사용함으로써 여러 AWS 리전에서 VPC가 점차 늘어남에 따라 쉽게 연결하고 관리할 수 있다. 또한 Transit Gateway 리전 간 피어링을 사용함으로써 장애 발생 지점이나 대역폭 병목 없이 리전 간 트래픽을 암호화한다. 또한 온프레미스 환경과 TransitGateway VPN 연결로 여러 리전간 VPC 관리가 쉬워진다.

IV. 진행상황

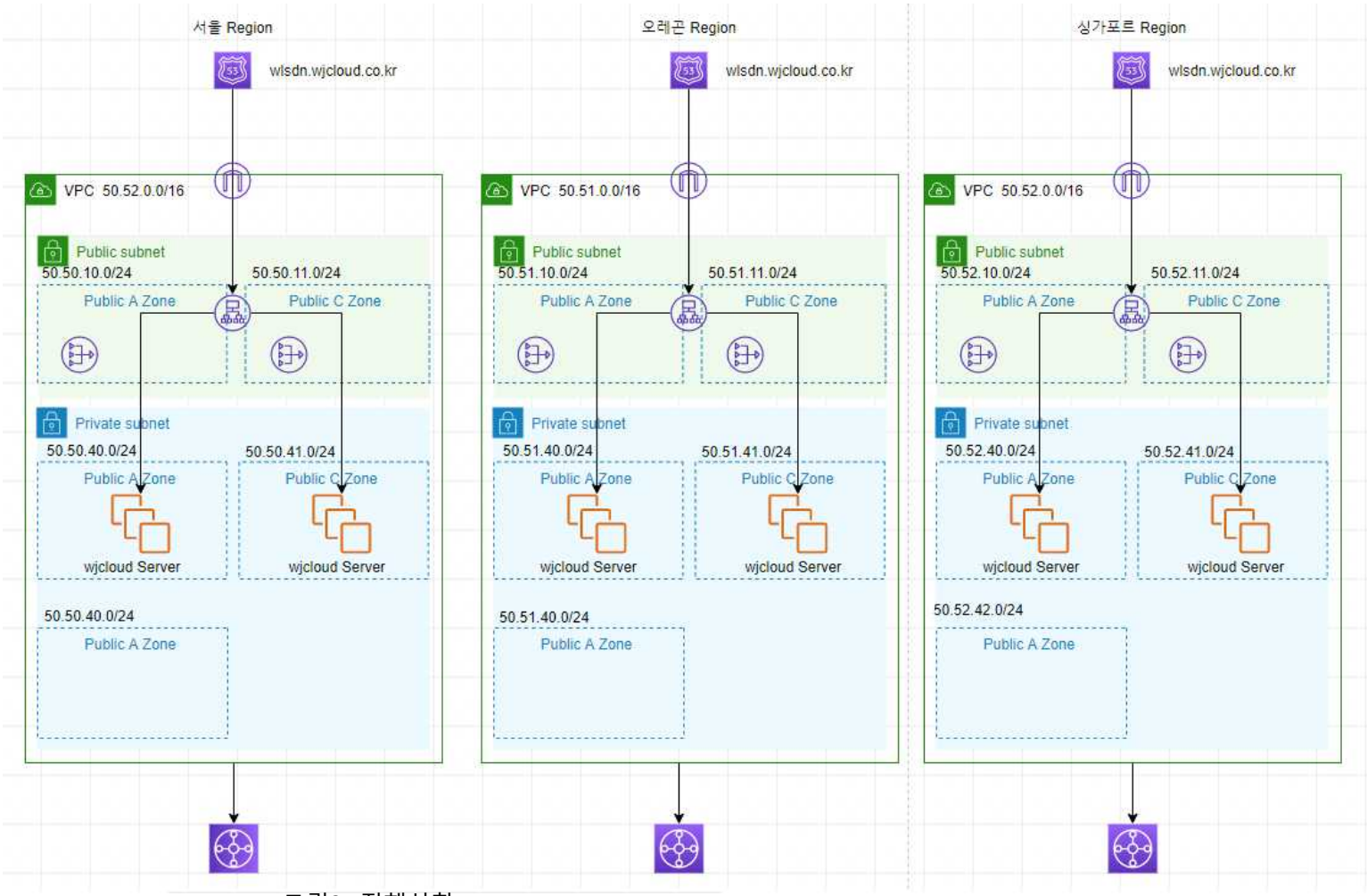

그림3. 진행상황

1. 그림2 인프라의 Cloudformation Yaml 파일 작성 완료.

1-1) Cloudformation_Seoul.yaml : 지연 시간 기반 Record 생성 필요, RDS 생성 필요, Tomcat과 데이터베이스 연동 자동화 필요

1-2) Cloudformation_Oregon.yaml : 지연 시간 기반 Record 생성 필요, RDS 생성 필요, Tomcat과 데이터베이스 연동 자동화 필요

1-3) Cloudformation_Singapore.yaml : 지연 시간 기반 Record 생성 필요, RDS 생성 필요, Tomcat과 데이터베이스 연동 자동화 필요

1-4) Cloudformation_OpenSwan.yaml : 작성 필요

V. 개선사항

1. Route53 지연시간 기반 Properties를 추가하여 지연시간 기반 라우팅 Record생성, 그 후 가중치 기반 레코드를 생성하여 지정한 가중치에 따라 리전 내 인스턴스로 트래픽을 라우팅하는 방법 고려.

2. OpenSwan yaml 파일 작성

2-1) OpenSwan site-to-site vpn 구현

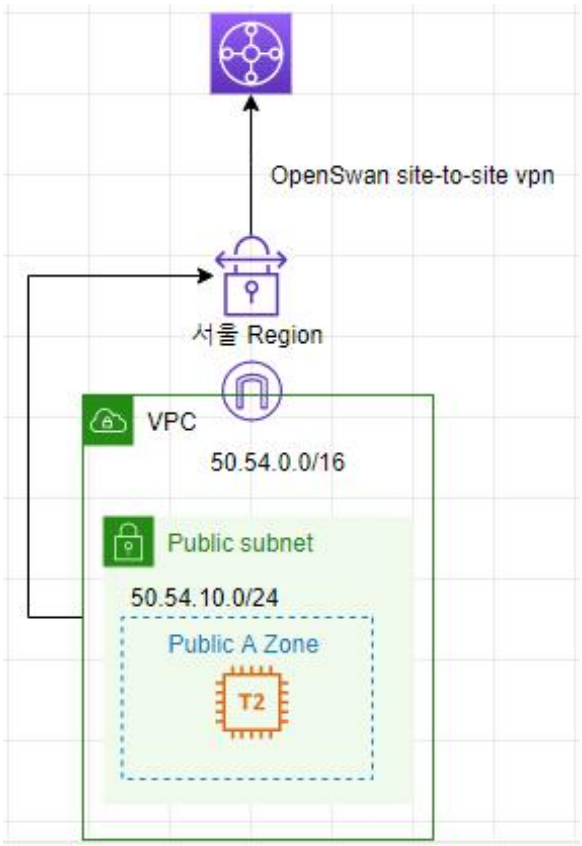

그림4. OpenSwan

3. Tomcat과 데이터베이스 연동 자동화

4. TransitGateway Peering 자동화

VI. 개선사항을 위한 자료수집

1. Route53 지연시간 기반 라우팅 Parameter 추가
   -> AWS::Route53::RecordSet Region Properties 추가

```
Region: String
```

```
JinwooDNSRecord:
  DependsOn: JJWALB
  Type: AWS::Route53::RecordSet
  Properties:
    Region: ap-northeast-2
    HostedZoneName: !Join ['', [!Ref HostedZone, "."]] #
    Name: !Join ['',[wlsdn, ".", !Ref HostedZone, "."]]
    Type: CNAME
    TTL: '900'
    ResourceRecords: [!GetAtt JJWALB.DNSName]   # CNAME은
```

그림5. Route53.RecordSet

https://docs.aws.amazon.com/ko_kr/Route53/latest/DeveloperGuide/TutorialTransitionToLBR.html -> Amazon Route 53에서 지연 시간 기반 라우팅으로 전환 공부

https://docs.aws.amazon.com/ko_kr/Route53/latest/DeveloperGuide/TutorialLBRMultipleEC2InRegion.html -> Amazon Route 53의 지연 시간 및 가중치 기반 레코드를 사용하여 한 리전의 여러 Amazon EC2 인스턴스로 트래픽 라우팅 공부

-> 스택 생성 오류 발생하여 문법 검색중.

2. OpenSwan yaml 파일 작성

2-1) OpenSwan site-to-site vpn 구현

-> 콘솔로 직접 생성하여 TEST

```
vi /etc/ipsec.d/aws.conf

conn Tunnel1
        authby=secret
        auto=start
        left=%defaultroute
        leftid= # IDC(openswan) 공인 ip -> EC2의 Public ip 입력
        right= # VPN Tunnel의 외부 ip (터널 하나)
        type=tunnel
        ikelifetime=8h
        keylife=1h
        phase2alg=aes128-sha1;modp1024
        ike=aes128-sha1;modp1024
        keyingtries=%forever
        keyexchange=ike
        leftsubnet= # IDC의 사설 IP 대역 EC2의 Private ip 입력
        rightsubnet= # AWS의 사설 IP 대역(Seoul)
        dpddelay=10
        dpdtimeout=30
        dpdaction=restart_by_peer
```

그림6. OpenSwan설정

->TransitGateway의 참조값 필요하여 검색중

3. Tomcat과 데이터베이스 연동 자동화

3-1) EC2 AMI, UserData를 사용하여 RDS의 endpoint, 포트 번호 변경, 서버 시작 명령줄 추가

```
UserData: # 시작시 Linux 인스턴스에서 명령 실행
  Fn::Base64:
    !Sub |
      sed -i 's/prd-wjcc-db.cqkl80nec9ru/RDS엔드포인트' /server/was/instances/center/conf/server.xml
      sed -i 's/8080/8081' /server/was/instances/center/conf/server.xml
      /server/was/launcher/startup_center.sh
```

그림7. 명령줄 추가

4. 리전간 TransitGatewayPeering은 TransitGatewayAttachment에서 관리한다.

```
Type: AWS::EC2::TransitGatewayAttachment
Properties:
  SubnetIds:
    - String
  Tags:
    - Tag
  TransitGatewayId: String
  VpcId: String
```

리전간 TransitGatewayPeering은 Properties 옵션에 없어 수동으로 Peering이 필요하다.