



Architecting with AWS (실습편)

GSNeotek

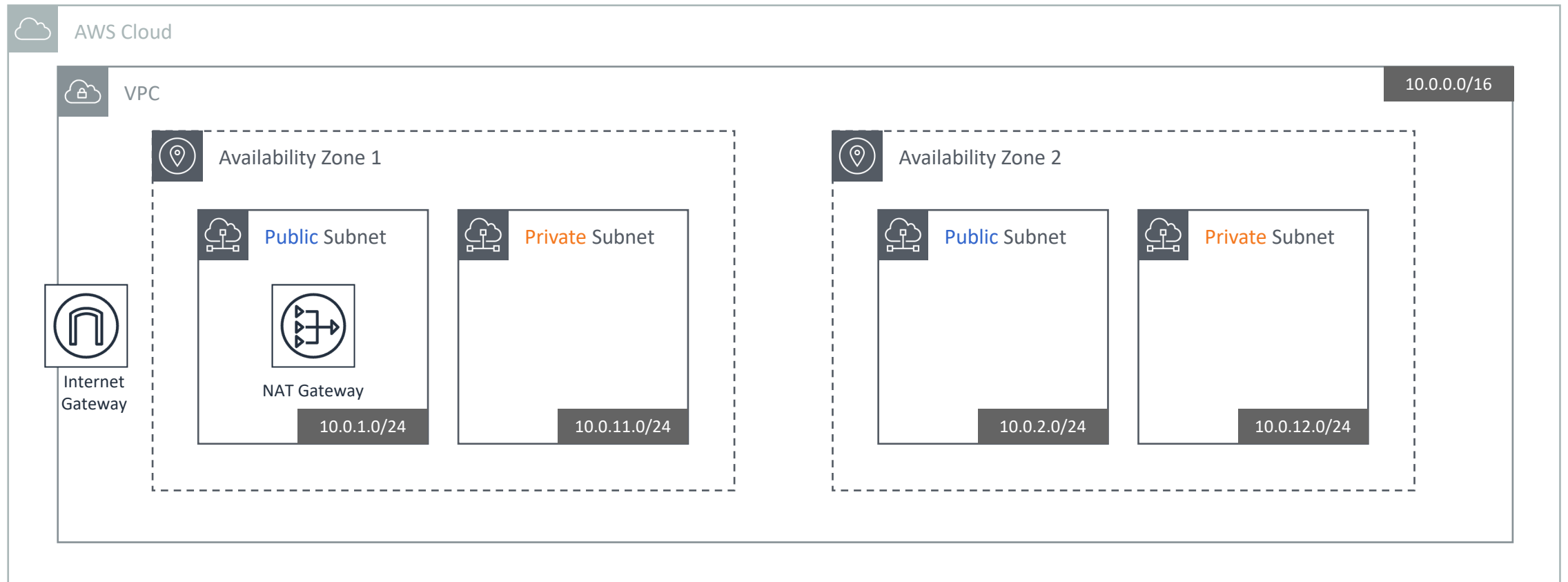


01

기본 네트워크 구성



STEP 1. 기본 네트워크 구성



Public Subnet's
Route table

172.16.0.0
172.16.1.0
172.16.2.0

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-xxxxxxx

Private Subnet's
Route table

172.16.0.0
172.16.1.0
172.16.2.0

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	nat-xxxxxxx

STEP 1. 기본 네트워크 구성

1-1. VPC 생성

1-2. VPC Subnet 생성 x 4EA

1-3. Route Table 생성 x 2EA

1-4. Internet G/W 생성 + Attach to VPC

1-5. NAT G/W 생성

1-6. Route Table 규칙 수정 x 2EA

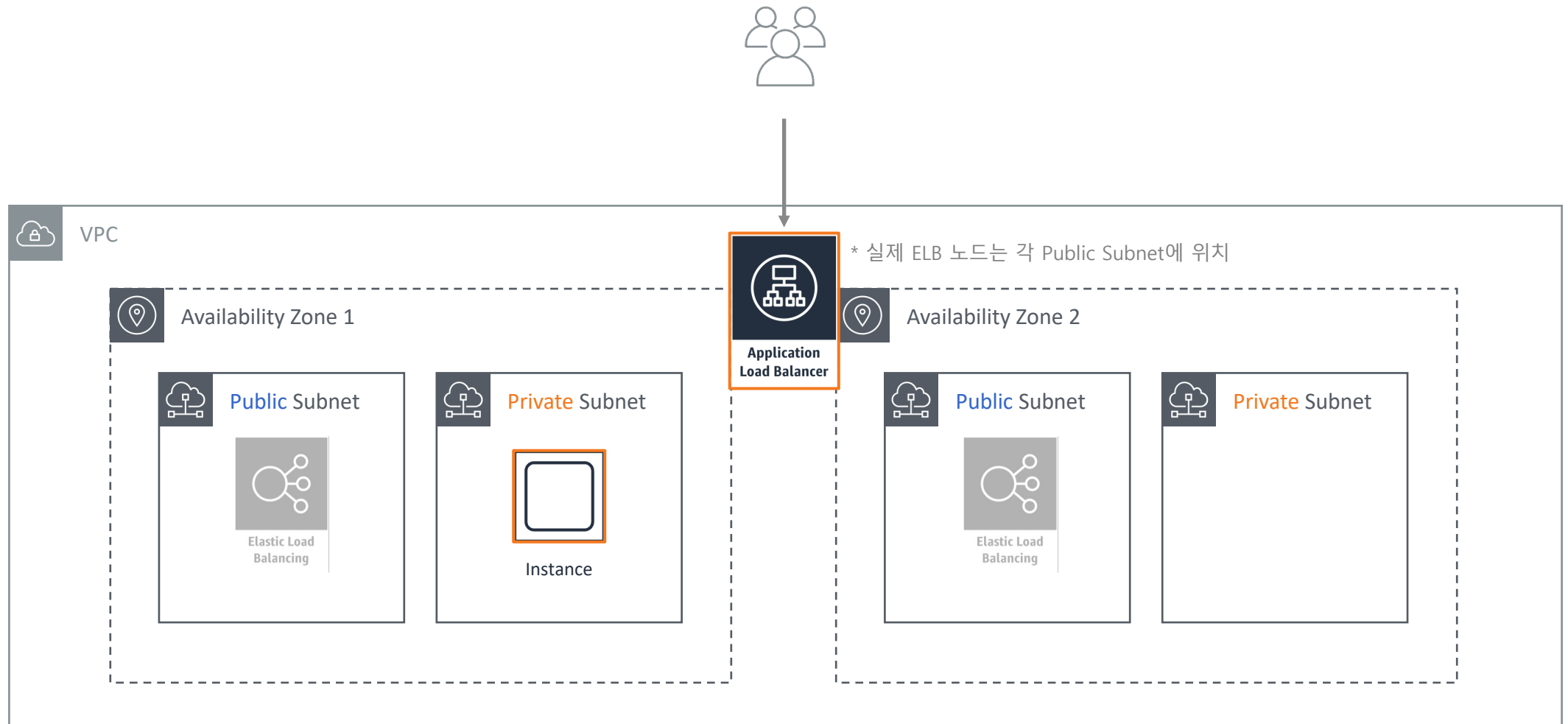
1-7. VPC Subnet에 Route Table 연결



02

단일 EC2 인스턴스 + LB 구성

STEP 2. 단일 EC2 인스턴스 + Load Balancer 구성



STEP 2. 단일 EC2 인스턴스 + Load Balancer 구성

2-1. EC2 인스턴스 생성

2-2. EC2 인스턴스 상태 확인

2-3. ELB(Application LB) 생성

2-4. Target Group 생성 + EC2 인스턴스 등록

2-5. Target Group에서 EC2 Status 확인

2-6. 웹브라우저에서 ELB 주소로 요청/테스트

2-7. (페이지가 뜨지 않을 경우) 원인 찾기

STEP 2. 단일 EC2 인스턴스 + Load Balancer 구성

하단 예시중에 1개 선택 후 진행

User Data 예시-1

```
#!/bin/sh
yum -y install httpd php
chkconfig httpd on
systemctl start httpd
cd /var/www/html
wget dbxh6vvykosu3.cloudfront.net/web-php-v1.tar.gz
tar xvfz web-php-v1.tar.gz
yum -y update
amazon-linux-extras install -y epel
yum -y install stress
```

User Data 예시-2

```
# 아래링크에서 스크립트 복사 후 붙여넣기
# https://github.com/jogilsang/user-data
```

부하테스트(CPU) 시나리오

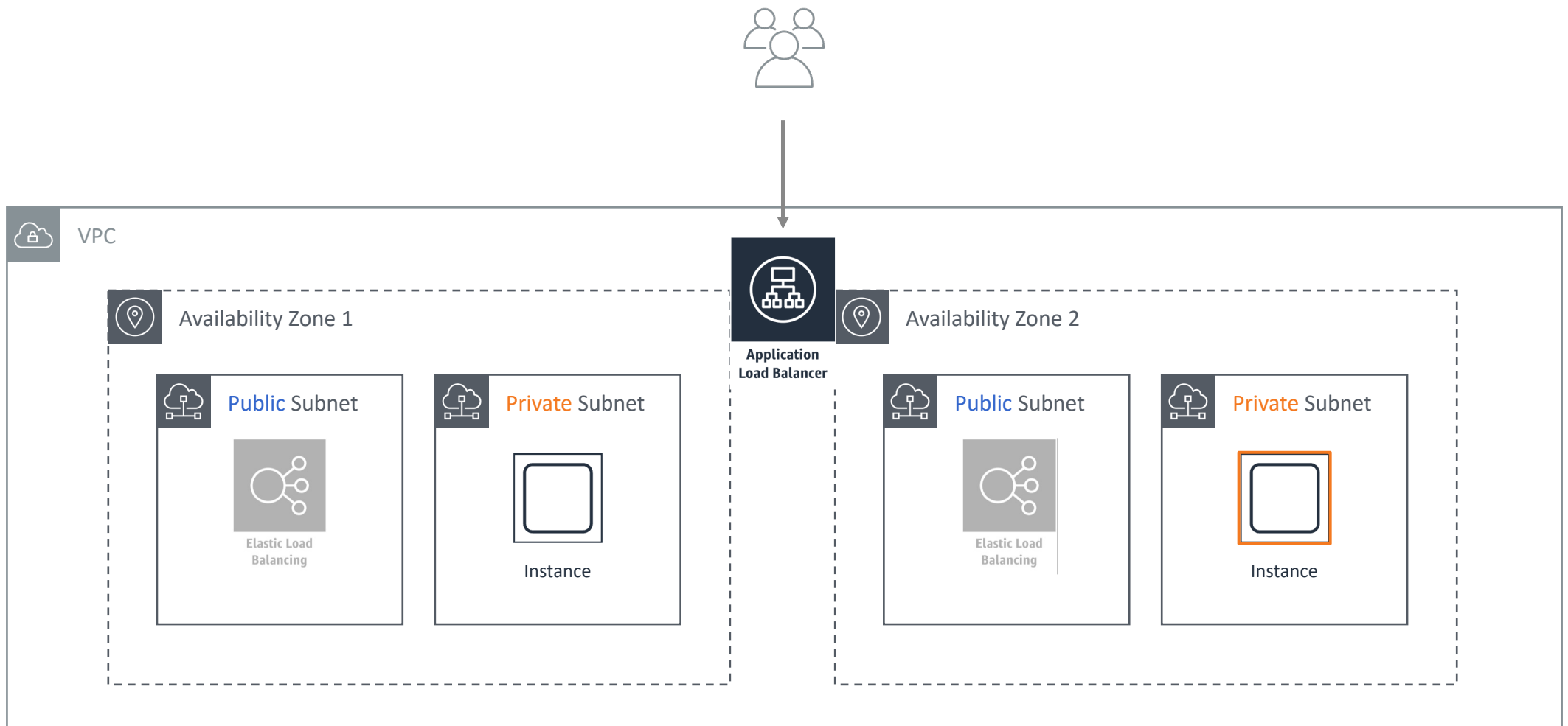
```
stress --cpu 1 --timeout 300s
```




03

고가용성 구성

STEP 3.고가용성 구성



STEP 3. 고가용성 구성

3-1. AMI(Amazon Machine Image) 생성

3-2. 추가 EC2 인스턴스 생성

3-3. 추가 EC2 인스턴스를 Target Group에 추가

3-4. Target Group에 추가 인스턴스 Status 확인

3-5. 웹브라우저에서 ELB 주소로 요청/테스트

3-6. 특정 EC2 인스턴스 중지 (시뮬레이션)

3-7. 웹브라우저에서 ELB 주소로 요청/테스트

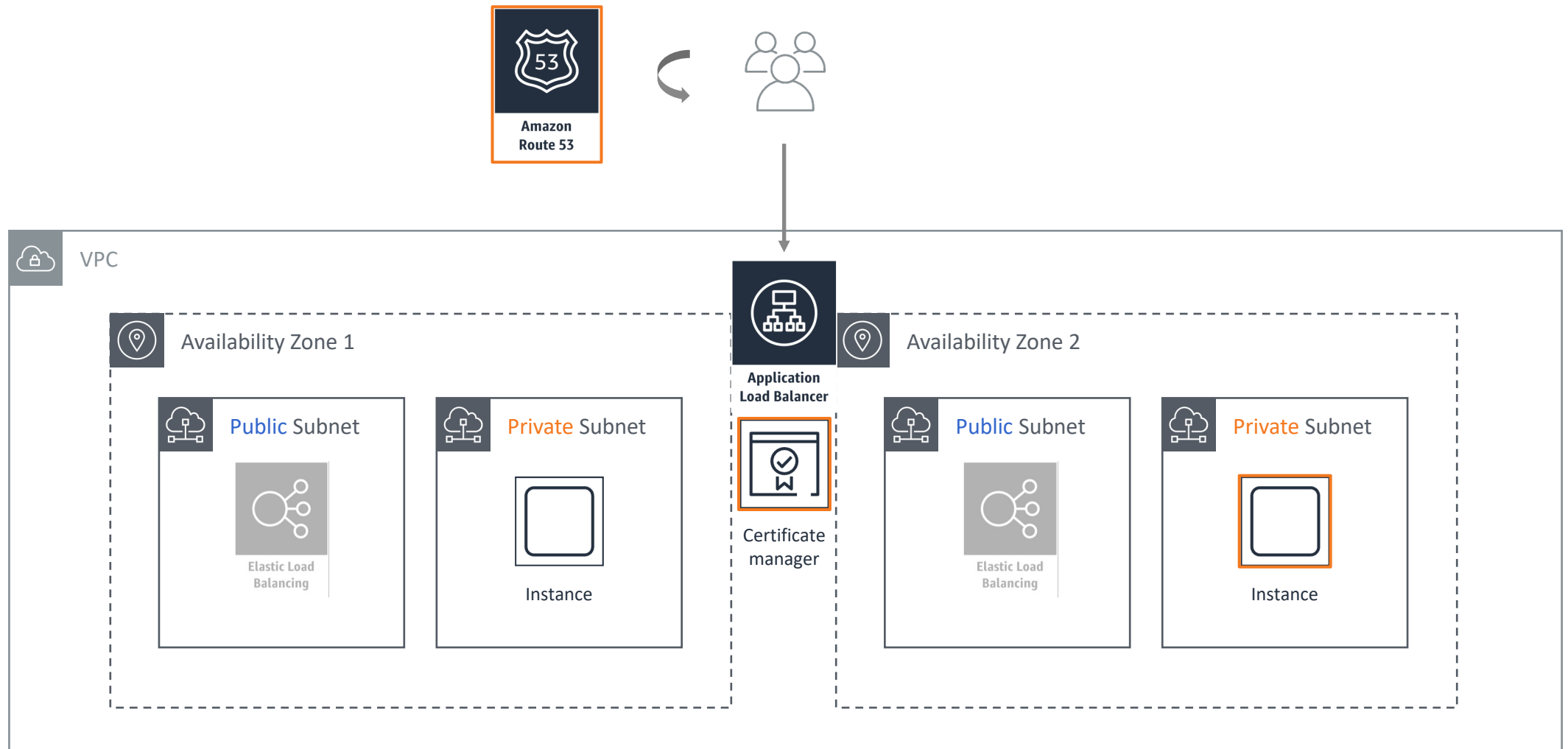


04

HTTPS 서비스 구성



STEP 4. HTTPS 서비스 구성

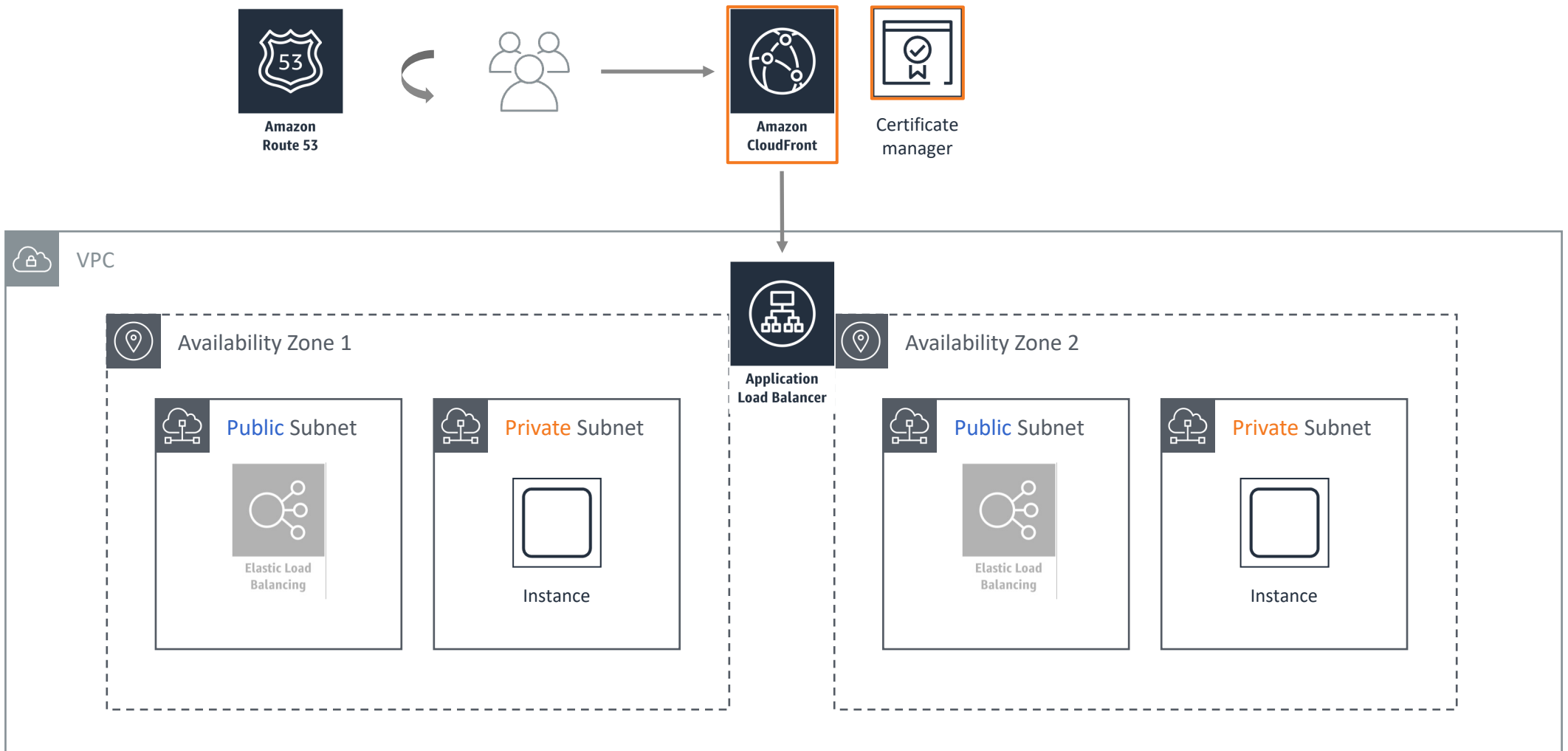




05

CDN 계층 구성

STEP 5. CDN 계층 구성



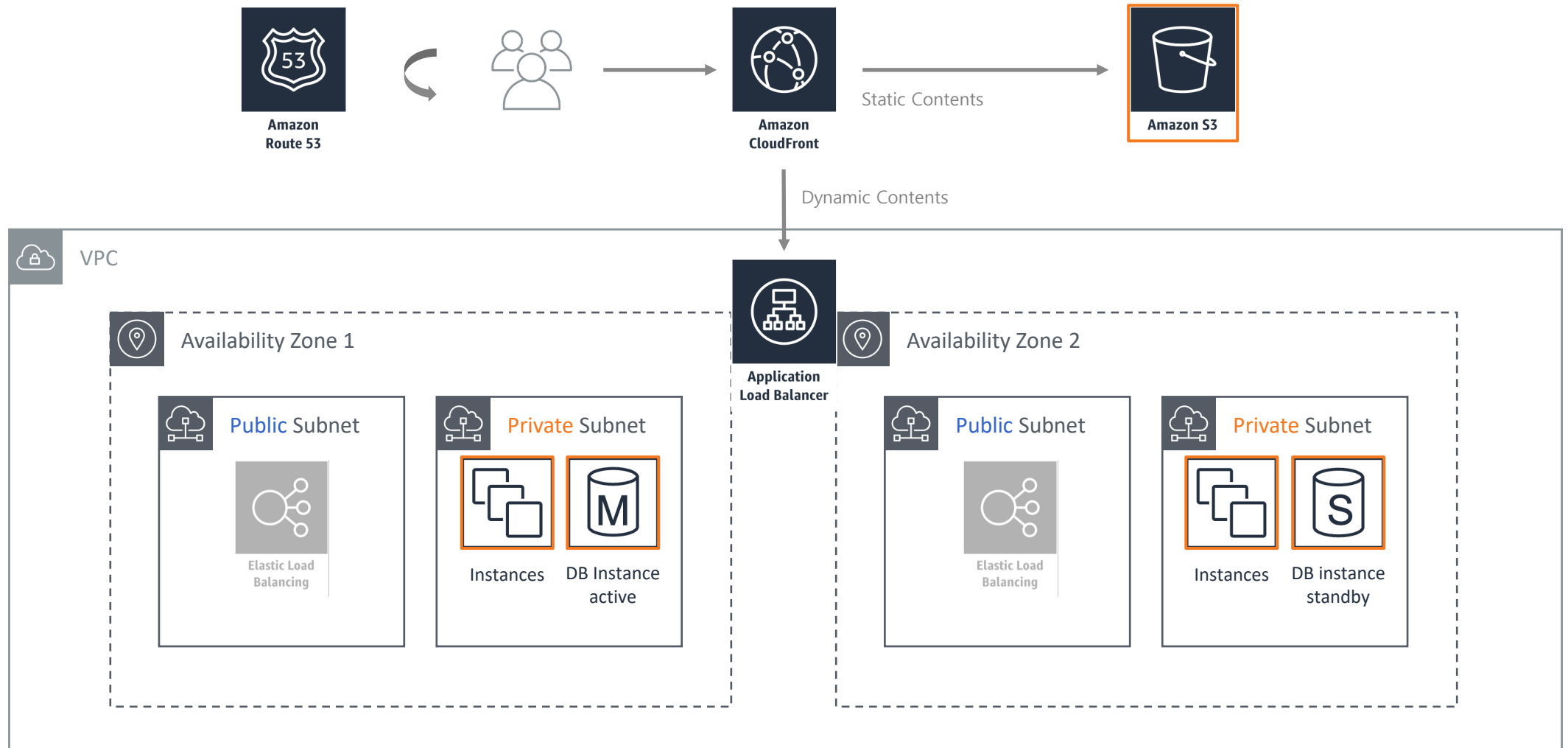


06

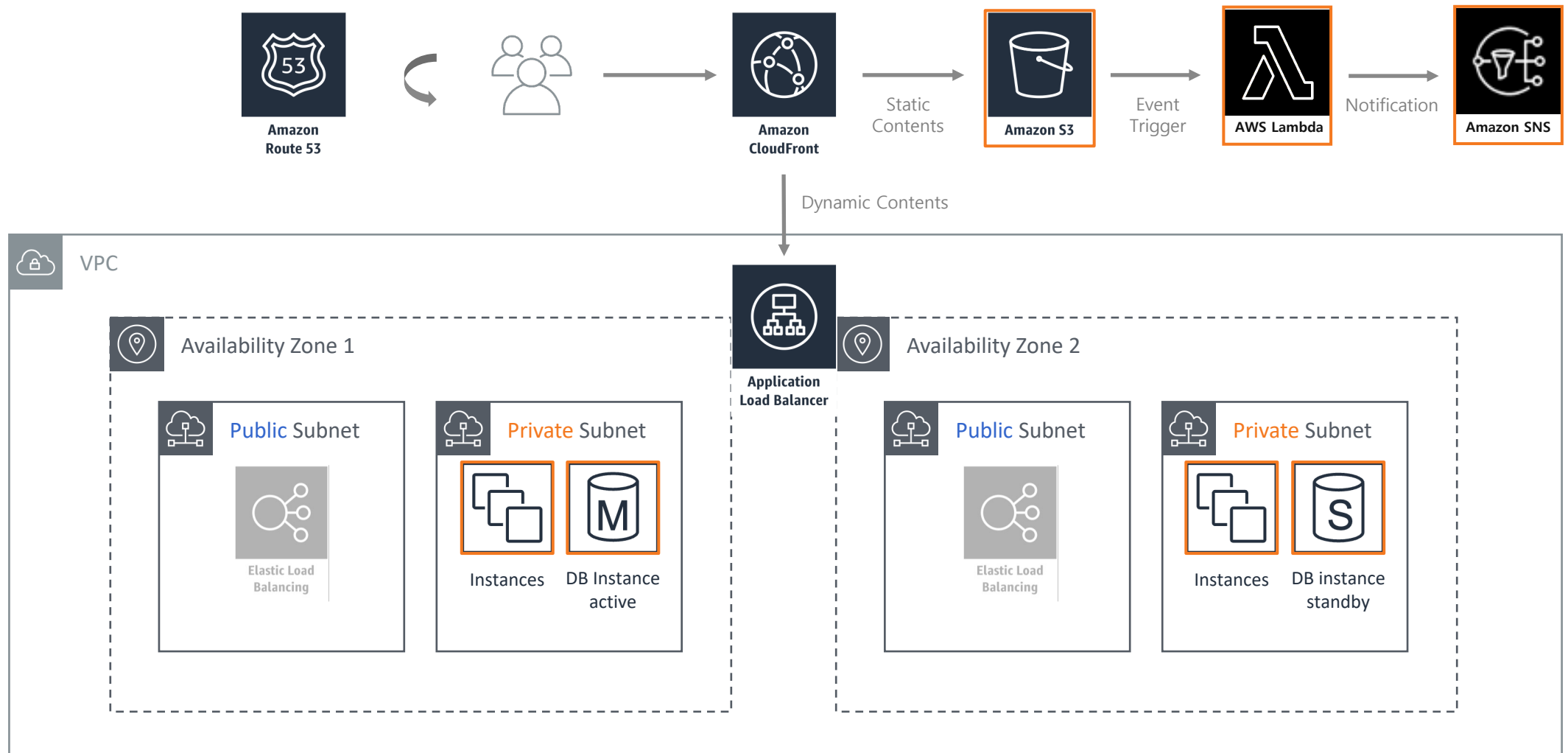
최종 표준 구성



STEP 6. 최종 표준 구성



(별첨). 이벤트 트리거 및 알람추가



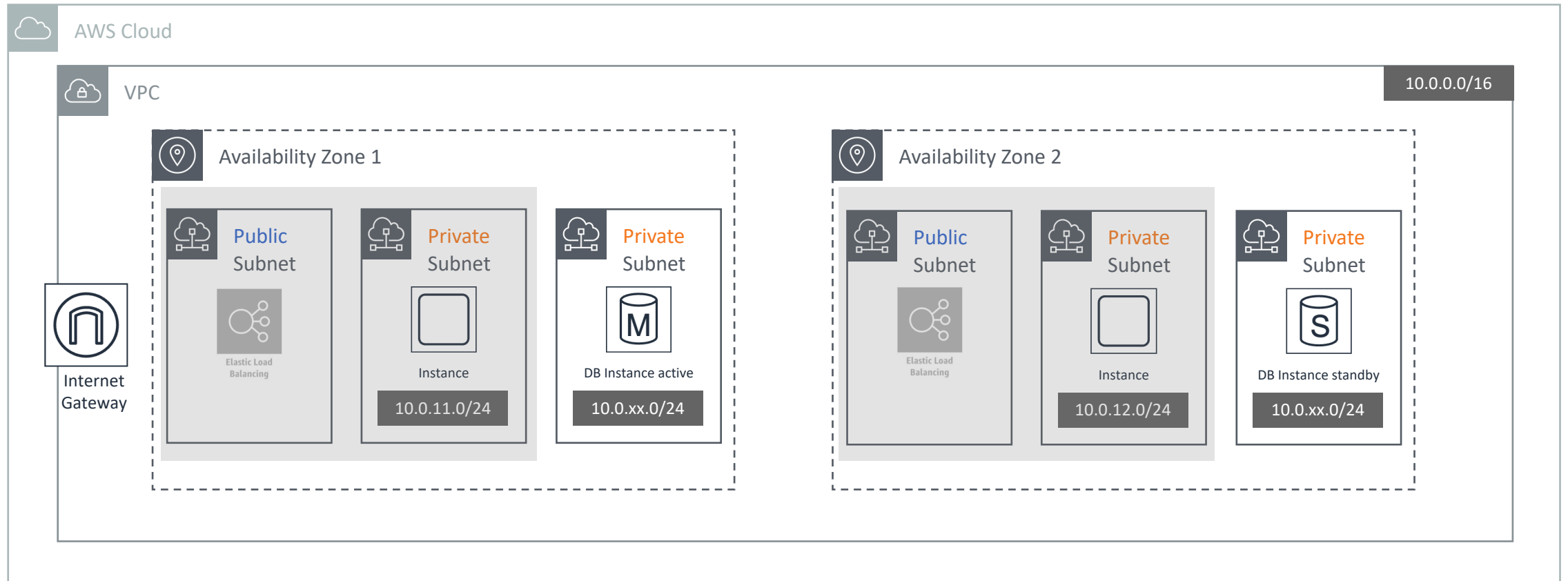


07

(별첨) DB 인스턴스 구성



STEP 7. DB인스턴스 구성



Instance
Security Group



Protocol	Port-Range	Source
TCP	80	LB-SG
TCP	443	LB-SG

RDS
Security Group



Protocol	Port-Range	Source
TCP	3306	Instance-SG

STEP 7. DB 인스턴스 구성

7-1. AWS RDS를 통해 DB인스턴스(MySQL) 생성

7-2. DB인스턴스에 대한 Security Group 수정

7-3. CLI를 이용한 DB Connection 확인

```
mysql -u admin -P 3306 -h [endpoint주소] -p [패스워드]
```

7-4. CLI를 이용한 TABLE 생성 및 SELECT 쿼리

```
-- DML  
use mysql;  
select user from user;  
  
-- DDL  
CREATE DATABASE test;
```

7-5. SNS 생성 및 DB인스턴스 FAIL OVER 이벤트 구독

7-6. DB인스턴스 FAILOVER 알람 확인

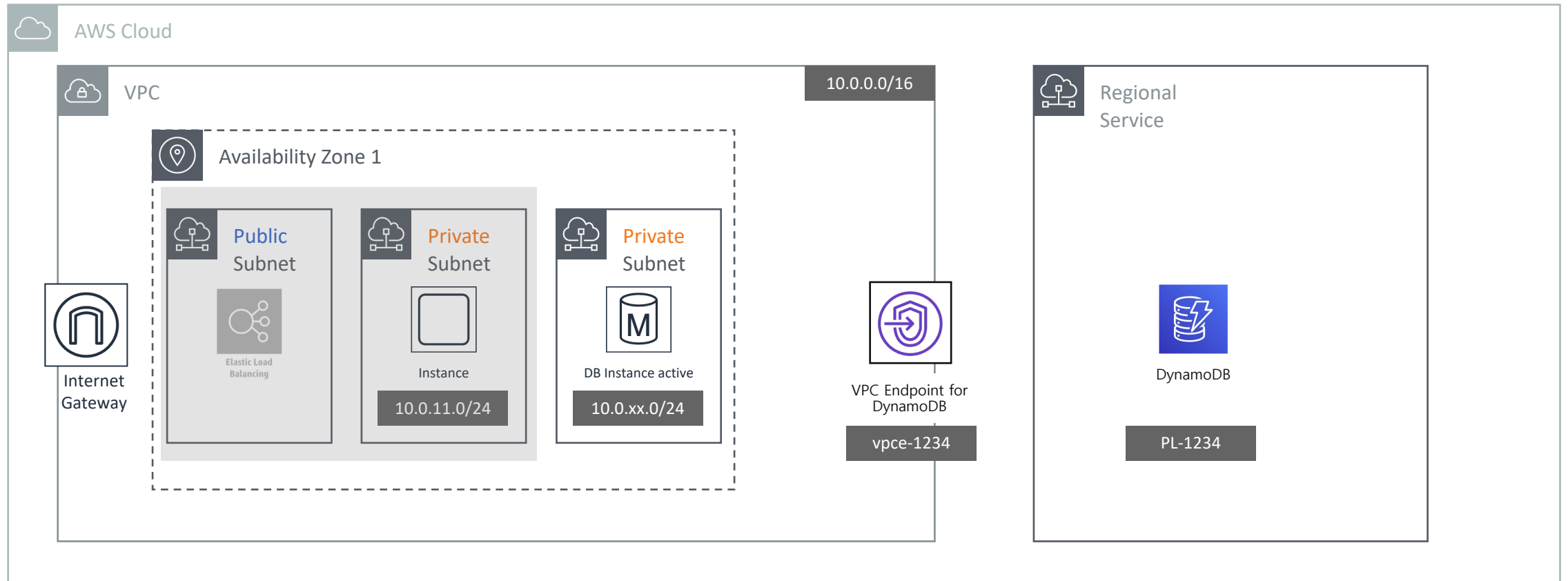


08

(별첨) DynamoDB 구성



STEP 8. DynamoDB 구성



Private Subnet's
Route table

172.16.0.0
172.16.1.0
172.16.2.0

Destination	Target
PL-1234	vpce-1234
0.0.0.0/0	nat-xxxxxxx
10.0.0.0/16	local



STEP 8. DynamoDB 구성

8-1. AWS DynamoDB 콘솔화면 둘러보기

8-2. 테이블 생성하기

8-3. 데이터 WRITE, READ 및 쿼리 진행

8-4. Global Secondary INDEX 생성 및 쿼리

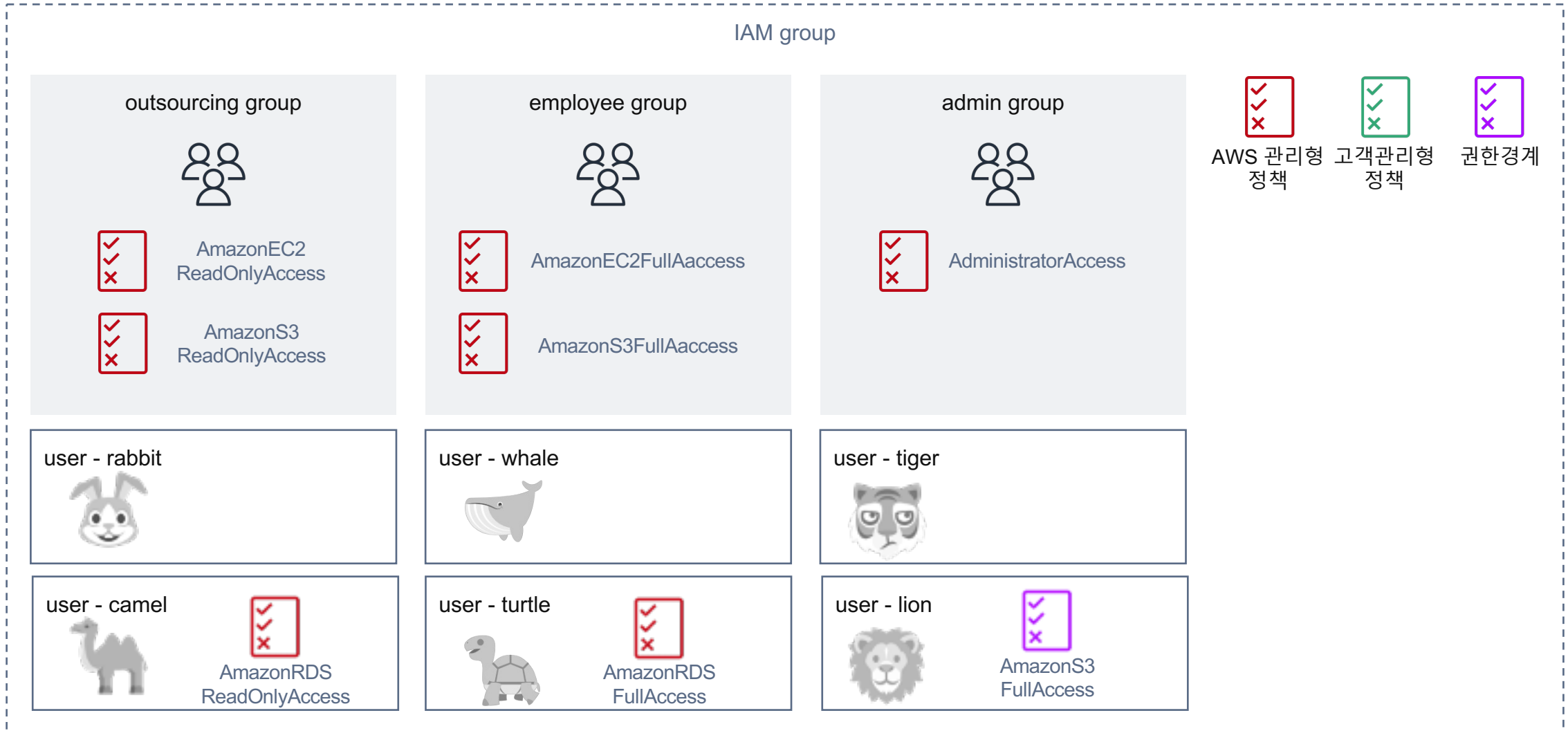
8-5. VPC Endpoint Gateway 생성 및 확인



09

(별첨) IAM 구성

STEP 9. IAM 구성



STEP 9. 단일 EC2 인스턴스 + Load Balancer 구성

하단 예시중에 1개 선택 후 진행

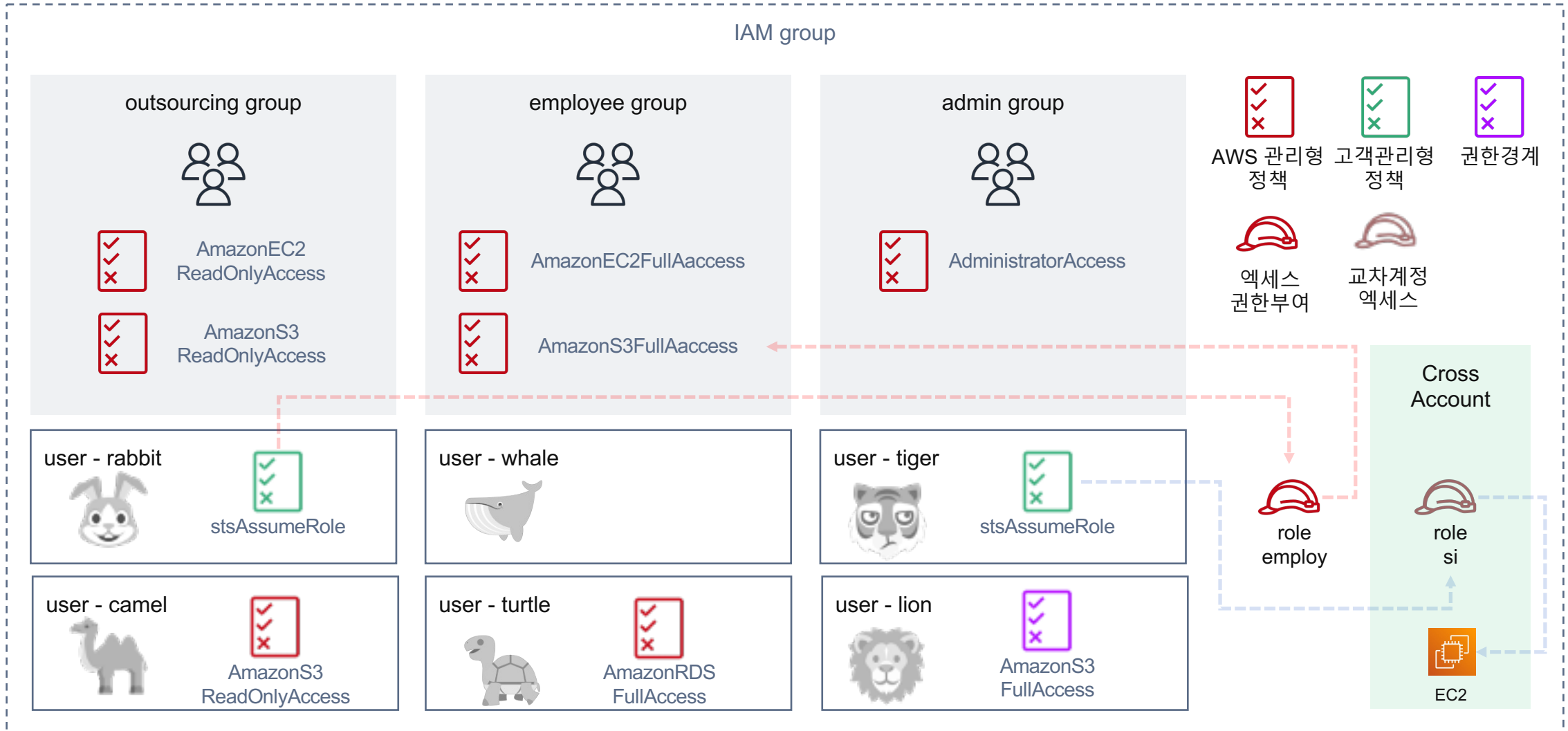
Role – 신뢰정책(Trust Entity)

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::[account-id]:root"  
      },  
      "Action": "sts:AssumeRole",  
      "Condition": {}  
    }  
  ]  
}
```

Role – 정책부여(Policy)

```
SupportUser  
ReadOnlyAccess  
AWSSupportAccess
```

STEP 9. IAM 구성



STEP 9. IAM구성

9-1. 사용자(User) 생성

9-2. 그룹(Group) 생성

9-3. 그룹에 정책(Policy) 추가

9-4. 그룹에 사용자 추가

9-5. 사용자 별 로그인 후 권한확인

9-6. 역할전환(Role Switch) 이해

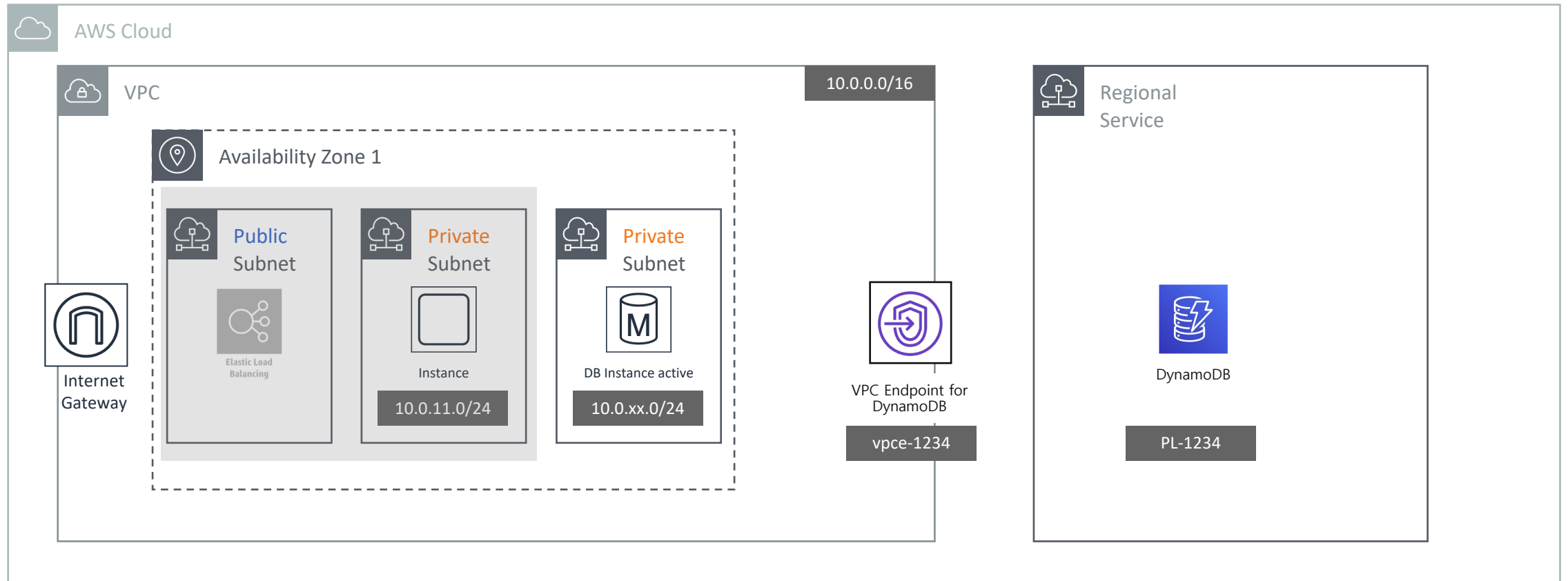


09

(별첨) Auto Scaling Group 구성



STEP 8. DynamoDB 구성



Private Subnet's
Route table

172.16.0.0
172.16.1.0
172.16.2.0

Destination	Target
PL-1234	vpce-1234
0.0.0.0/0	nat-xxxxxxx
10.0.0.0/16	local



STEP 8. DynamoDB 구성

8-1. AWS DynamoDB 콘솔화면 둘러보기

8-2. 테이블 생성하기

8-3. 데이터 WRITE, READ 및 쿼리 진행

8-4. Global Secondary INDEX 생성 및 쿼리

8-5. VPC Endpoint Gateway 생성 및 확인



Thank you



www.gsneotek.co.kr



www.wisen.co.kr



www.wisen.co.kr/pages/blog/blog.html



facebook.com/gswisen

