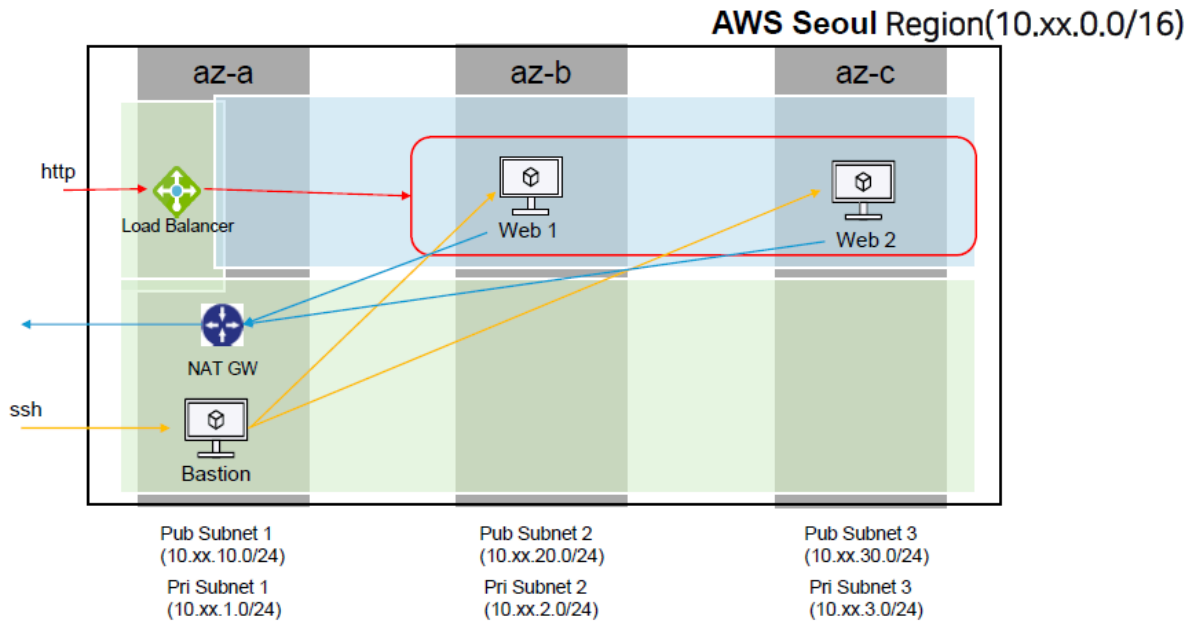


MiniProject : Cloud

아키텍처 구성도



세부 요구조건

- 안정적인 Web 서비스 환경 제공
 - 서울 AZ b, AZ c AZ d 에 분산 배포되는 로드밸런싱 환경 구성 (Web 1,2)
 - Web 1, 2 은 Private Subnet 에 배포 및 NAT 를 통한 외부 통신 가능하도록 구성
 - Web 1, 2 은 Bastion server 를 통해서만 ssh 접속 가능하도록 구성
 - LB 를 통하여 Web 서버 접근 시 서버 별로 다른 웹페이지 출력
- Bastion 서버 환경 구성 (
 - Public subnet 에 Cloud9 인스턴스를 배포
 - AZ a 에 구성

1. VPC 생성

VPC 설정

생성할 리소스 정보
VPC 리소스 또는 VPC 및 기타 네트워킹 리소스를 생성합니다.

☒ VPC만 ☐ VPC 중

이름 태그 - 선택 사항
Name 태그와 (선택)자가 지정하는 값을 포함하는 태그를 생성합니다.

a*****-vpc

IPv4 CIDR 블록 정보
☒ IPv4 CIDR 수동 입력 ☐ IPAM 할당 IPv4 CIDR 블록

IPv4 CIDR
10.0.0.0/16

IPv6 CIDR 블록 정보
☒ IPv6 CIDR 블록 없음 ☐ IPAM 할당 IPv6 CIDR 블록 ☐ Amazon 제공 IPv6 CIDR 블록 ☐ 내가 소유한 IPv6 CIDR

태그 정보
기본값

- AWS Region에 따라 IPv4 CIDR은 10.xx.0.0/16 으로 설정

2. 서브넷 생성

서브넷 설정
서브넷의 CIDR 블록 및 가용 영역을 지정합니다.

1/1개 서브넷

서브넷 이름
a*****-subnet-1

가용 영역 정보
서브넷의 가용 영역을 선택합니다. 선택하고 있으면 Amazon이 자동으로 선택합니다.

아시아 태평양 (서울) / ap-northeast-2a

IPv4 VPC CIDR 블록 정보
서브넷에 대해 VPC의 IPv4 CIDR 블록을 선택합니다. 서브넷의 IPv4 CIDR이 이 블록 내에 있어야 합니다.

10.0.0.0/16

IPv4 서브넷 CIDR 블록
10.0.1.0/24

▼ 태그 - 선택 사항

키
Name

값 - 선택 사항
a*****-subnet-1

새 태그 추가
49글자(문) 태그가 더 추가할 수 있습니다.

제거

새 서브넷 추가

(1) 퍼블릭 서브넷 생성

- 가용영역 a, b, c에 각각 Public Subnet 한 개씩 생성
- 아키텍처 구성도에 따라 a의 퍼블릭 서브넷 주소는 10.xx.10.0/24 , b의 퍼블릭 서브넷 주소는 10.xx.20.0/24 , c의 퍼블릭 서브넷 주소는 10.xx.30.0/24로 지정

(2) 프라이빗 서브넷 생성

- 가용영역 a, b, c에 각각 Private Subnet 한 개씩 생성
- 아키텍처 구성도에 따라 a의 프라이빗 서브넷 주소는 10.xx.1.0/24 , b의 프라이빗 서브넷 주소는 10.xx.2.0/24 , c의 프라이빗 서브넷 주소는 10.xx.3.0/24로 지정

서브넷(6개)

이 VPC 내의 서브넷

ap-northeast-2a

A subnet-pub-1

A subnet-pri-1

ap-northeast-2b

B subnet-pub-2

B subnet-pri-2

ap-northeast-2c

C subnet-pub-3

C subnet-pri-3

3. 인터넷 게이트웨이 생성

인터넷 게이트웨이 설정

이름 태그
Name 태그와 일치하는 값을 포함하는 태그를 생성합니다.

태그 - 선택 사항
태그는 AWS 리소스에 동접하는 레이블입니다. 각 태그는 키와 선택적 값으로 구성됩니다. 태그를 사용하여 리소스를 분류 및 필터링하거나 AWS 비용을 추적할 수 있습니다.

키
Q Name X

값 - 선택 사항
Q a***** X 제거

새 태그 추가
49개(한) 태그가 더 추가될 수 있습니다.

완료 인터넷 게이트웨이 생성

- 인터넷 게이트웨이 생성(Public Subnet 통신용)

4. NAT 게이트웨이 생성

NAT 게이트웨이 설정

이름 - 선택 사항
 'Name' 키와 사용자가 지정하는 값을 포함하는 태그를 생성합니다.
 이름은 최대 256자까지 입력할 수 있습니다.

서브넷
 NAT 게이트웨이를 생성할 서브넷을 선택합니다.

서브넷 선택

(a*****-subnet-1)

1a

- NAT 게이트웨이 생성(Public Subnet 통신용)
- NAT 게이트웨이는 Public 서브넷에 위치해야 인터넷 게이트웨이를 통해 외부와 통신 가능

5. 라우팅 테이블 지정

라우팅 편집

대상
 10.0.0.0/16
 Q 0.0.0.0/0

적용할 추가

대상
 local
 게리더 게이트웨이
 고어 네트워크
 외부 전용 인터넷 게이트웨이
 Gateway Load Balancer 엔드포인트
 인스턴스
 인터넷 게이트웨이
 로컬
 NAT 게이트웨이

인터넷 게이트웨이

Q igw-

- 생성한 인터넷 게이트웨이를 라우팅 테이블에 연결

서브넷 연결 편집

이 라우팅 테이블과 연결된 서브넷을 변경합니다.

이용 가능한 서브넷 (2/2)

Q 서브넷 연결 필터링

<input checked="" type="checkbox"/>	이름	서브넷 ID	IPv4 CIDR	IPv6 CIDR	라우팅 테이블 ID
<input checked="" type="checkbox"/>	a*****-subnet-2	subnet-0d0cb30b9a3a9b2fe	10.0.2.0/24	-	기본 (rtb-05db0ad90a6bfa2b1)
<input checked="" type="checkbox"/>	a*****-subnet-1	subnet-0e07c2667fd9d84db	10.0.1.0/24	-	기본 (rtb-05db0ad90a6bfa2b1)

선택한 서브넷

subnet-0e07c2667fd9d84db / a*****-subnet-1 X subnet-0d0cb30b9a3a9b2fe / a*****-subnet-2 X

취소 **연결 저장**

- 생성한 3개의 퍼블릭 서브넷을 연결

라우팅 테이블 설정

이름 - 선택 사항

Name 키와 사용자가 지정한 값을 포함하는 태그를 생성합니다.

VPC

이 라우팅 테이블과 함께 사용되는 VPC입니다.

태그

태그는 AWS 리소스에 할당하는 레이블입니다. 각 태그는 키와 선택적 값으로 구성됩니다. 태그를 사용하여 리소스를 검색 및 필터링하거나 AWS 비용을 추적할 수 있습니다.

키

Q Name X

값 - 선택 사항

Q a*****-pri-rt X

제거

새 태그 추가

49글자(255) 태그 개 더 추가할 수 있습니다.

취소

라우팅 테이블 생성

라우팅 편집

대상 10.0.0.0/16

Q 0.0.0.0/0 X

라우팅 추가

대상 local

상태 완료

NAT 게이트웨이

캐리어 게이트웨이

코어 네트워크

외부 전용 인터넷 게이트웨이

Gateway Load Balancer 엔드포인트

인스턴스

인터넷 게이트웨이

로컬

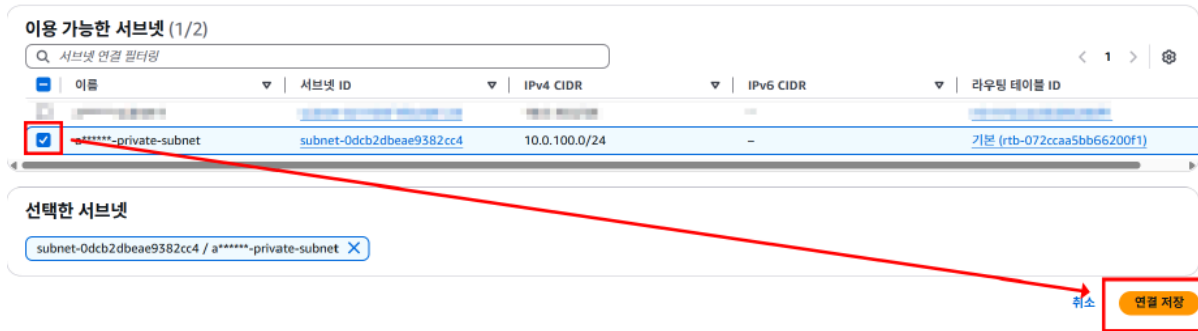
NAT 게이트웨이

네트워크 인터페이스

Outpost 로컬 게이트웨이

피어링 연결

- NAT게이트웨이를 지정할 라우팅 테이블 생성

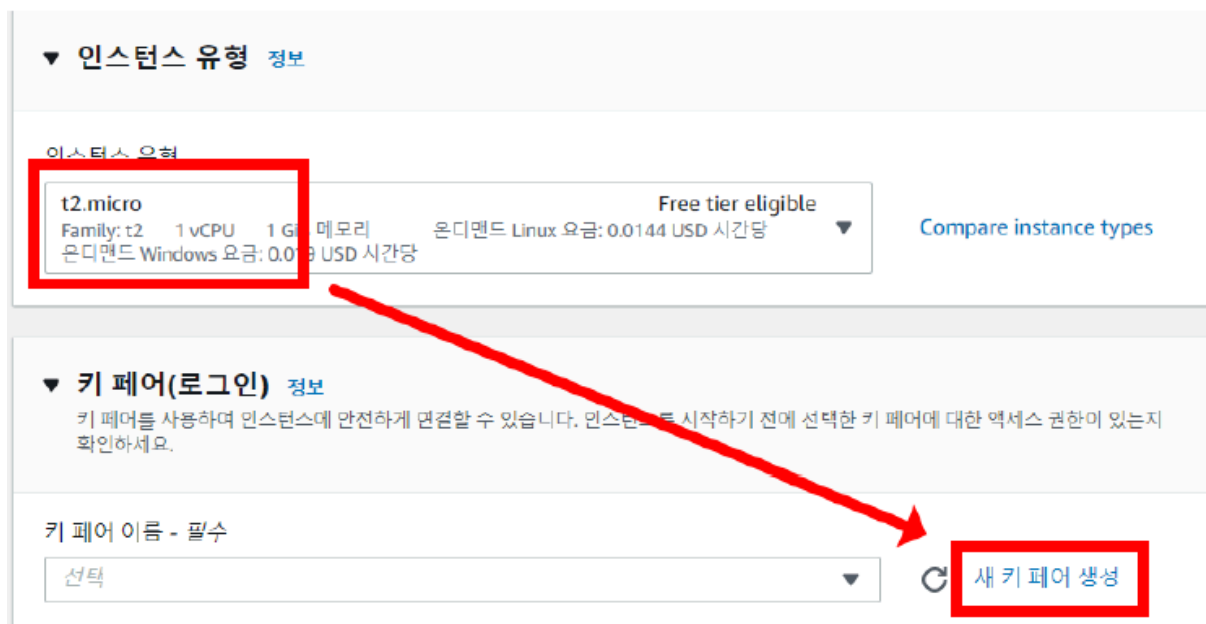


- 생성한 3개의 프라이빗 서브넷을 연결



- 구성 전체 모습

6. EC2 인스턴스 생성 및 Bastion Server 접속



키 페어 유형

☒ RSA
RSA 암호화된 프라이빗 및 퍼블릭 키 페어

☐ ED25519
ED25519 암호화된 프라이빗 및 퍼블릭 키 페어 (Windows 인스턴스에는 지원되지 않음)

프라이빗 키 파일 형식

☒ .pem
OpenSSH와 함께 사용

☐ .ppk
PuTTY와 함께 사용

취소 **키 페어 생성**

(1) Bastion 서버 생성

- 인스턴스 생성에서 인스턴스 유형 선택하고 새 키 페어 생성

VPC - 필수 정보

vpc-06640eee20dcf1787 172.31.0.0/16	(기본값)
닫기	
vpc-06640eee20dcf1787 172.31.0.0/16	(기본값)
vpc-0c2a788baaf4f07f7 (10.0.0.0/16)

- 네트워크 설정에서 내 VPC선택

VPC - 필수 정보

vpc-0c2a788baaf4f07f7 (a*****-vpc)
10.0.0.0/16

서브넷 정보

subnet-0e07c2667fd9d84db a*****-subnet-1
VPC: vpc-0c2a788baaf4f07f7 소유자: 271153858532 가용 영역: ap-northeast-2a
IP 주소 사용 가능: 251 CIDR: 10.0.1.0/24

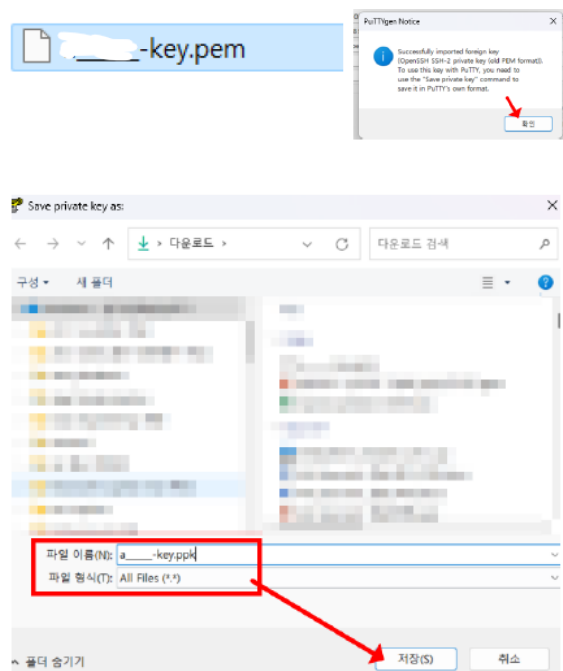
퍼블릭 IP 자동 할당 정보

활성화

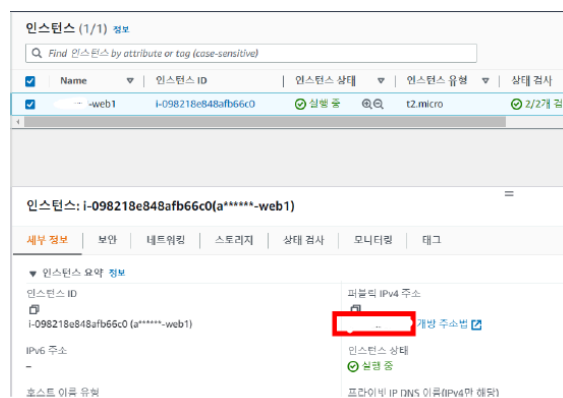
- Bastion 서버이므로 가용영역 A의 퍼블릭 서브넷 선택 및 퍼블릭 IP 자동 할당 활성화

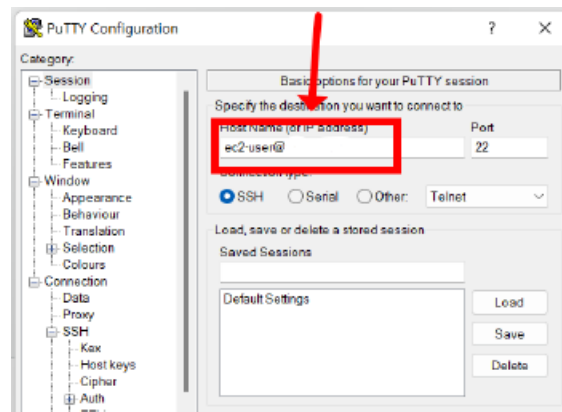


- 보안 규칙은 내 IP로 설정



- PuttyGen에서, 생성한 키 페어 파일 선택 후 ppk로 변경 후 저장





- Putty 실행해서 Bastion server의 퍼블릭 IPv4 주소로 접속

```
ec2-user@ip-10-0-1-234:~
Verifying : mod_httpd-1.15.19-1.amzn2.0.1.x86_64
Verifying : httpd-2.4.54-1.amzn2.x86_64
Verifying : mailcap-2.1.41-2.amzn2.noarch
Verifying : generic-logos-httpd-18.0.0-4.amzn2.noarch
Verifying : httpd-filesystem-2.4.54-1.amzn2.noarch
Verifying : apr-1.7.0-9.amzn2.x86_64

Installed:
  httpd.x86_64 0:2.4.54-1.amzn2

Dependency Installed:
  apr.x86_64 0:1.7.0-9.amzn2
  apr-util.x86_64 0:1.6.1-5.amzn2.0.2
  apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2
  generic-logos-httpd.noarch 0:18.0.0-4.amzn2
  httpd-filesystem.noarch 0:2.4.54-1.amzn2
  httpd-tools.x86_64 0:2.4.54-1.amzn2
  mailcap.noarch 0:2.1.41-2.amzn2
  mod_httpd.x86_64 0:1.15.19-1.amzn2.0.1

Complete!
ec2-user@ip-10-0-1-234 ~]$ sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
ec2-user@ip-10-0-1-234 ~]$
```

- 아래 명령어로 웹서버 구현
 - sudo yum update -y
 - sudo yum install httpd -y
 - sudo service httpd start



It works!

- Bastion 서버의 퍼블릭 IP를 웹브라우저에서 접속

(2) Web Server1, Web Server2 생성

The screenshot shows the AWS VPC console configuration for a new VPC. The 'VPC - 필수 정보' (VPC - Required Information) section shows the VPC ID 'vpc-0c2a788baaf4f07f7' and CIDR '10.0.0.0/16'. The '서브넷 정보' (Subnet Information) section shows the Subnet ID 'subnet-0e07c2667fd9d84db', VPC 'vpc-0c2a788baaf4f07f7', Owner '271153858532', Region 'ap-northeast-2a', and CIDR '10.0.1.0/24'. The '퍼블릭 IP 자동 할당 정보' (Public IP Auto Assignment Information) section shows the '활성화' (Activate) checkbox is checked. A red box highlights the Subnet Information section, and a red '비' (No) is written next to the '활성화' checkbox.

- 서브넷은 각각 가용영역의 프라이빗 서브넷으로 지정하고 퍼블릭 IP 자동 할당은 비활성화로 설정

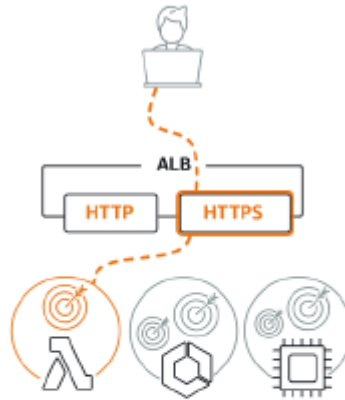
The screenshot shows the AWS Security Groups configuration for SSH access. The '인바운드 보안 그룹 규칙' (Inbound Security Group Rules) section shows a rule for 'ssh' (SSH) with '소스 유형' (Source Type) set to '보안 그룹' (Security Group) and '보안 그룹 ID' (Security Group ID) set to 'sg-0c2a788baaf4f07f7'. The '소스 유형' (Source Type) is set to '보안 그룹' (Security Group) and the '보안 그룹 ID' (Security Group ID) is set to 'sg-0c2a788baaf4f07f7'. The '소스 유형' (Source Type) is set to '보안 그룹' (Security Group) and the '보안 그룹 ID' (Security Group ID) is set to 'sg-0c2a788baaf4f07f7'. The '소스 유형' (Source Type) is set to '보안 그룹' (Security Group) and the '보안 그룹 ID' (Security Group ID) is set to 'sg-0c2a788baaf4f07f7'.

(1) Web Server1, Web Server2는 Bastion 서버를 통해서만 통신 가능하게 하는 것이 과제이므로 ssh의 소스유형은 사용자 지정으로 하고 Bastion 서버의 보안 규칙을 불러옴

(2) 보안그룹 규칙 추가하여 HTTP를(HTTP는 외부 사용자도 접근해야 하므로) 0.0.0.0/0으로 개방

7. 로드밸런서 연결

Application Load Balancer 정보



HTTP 및 HTTPS 트래픽을 사용하는 애플리케이션을 위한 유연한 기능이 필요한 경우 Application Load Balancer를 선택합니다. 요청 수준에 따라 작동하는 Application Load Balancer는 마이크로서비스 및 컨테이너를 비롯한 애플리케이션 아키텍처를 대상으로 하는 고급 라우팅 및 표시 기능을 제공합니다.

생성

- 로드 밸런서 생성에서 ALB(Application Load Balancer) 선택

<input checked="" type="checkbox"/> ap-northeast-2b (apne2-az2)	
서브넷	subnet-0a5733405a9775122
IPv4 주소	
AWS에서 할당	
<input checked="" type="checkbox"/> ap-northeast-2c (apne2-az3)	
서브넷	subnet-00dbdf160d48763b3
IPv4 주소	
AWS에서 할당	

- 매핑에서 Web Server의 프라이빗 서브넷만 선택

보안 그룹 정보

보안 그룹은 로드 밸런서에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 기존 보안 그룹을 선택하거나 새 보안 그룹을 생성할 수 있습니다.

보안 그룹

최대 5개의 보안 그룹 선택

×

- 보안 규칙은 인스턴스 생성시 만들었던 Web Server의 보안규칙 사용

기본 구성

대상 그룹이 생성된 후에는 이 섹션의 설정을 변경할 수 없습니다.

대상 유형 선택

☒ 인스턴스

- 특정 VPC 내의 인스턴스에 대한 로드 밸런싱을 지원합니다.
- Amazon EC2 Auto Scaling 를 사용하여 EC2 용량을 관리하고 크기를 조정할 수 있습니다.

☐ IP 주소

- VPC 및 온프레미스 리소스에 대한 로드 밸런싱을 지원합니다.
- 동일한 인스턴스에 있는 여러 IP 주소 및 네트워크 인터페이스로의 라우팅을 지원합니다.
- 마이크로서비스 기반 아키텍처를 통한 유연성을 제공하여 애플리케이션 간 통신을 간소화합니다.
- IPv6 대상을 지원하여 종단 간 IPv6 통신 및 IPv4에서 IPv6로의 NAT를 활성화합니다.

☐ Lambda 함수

- 단일 Lambda 함수로 라우팅을 지원합니다.
- Application Load Balancer에만 액세스할 수 있습니다.

☐ Application Load Balancer

- Network Load Balancer가 특정 VPC 내에서 TCP 요청을 수락하고 다중화할 수 있는 유연성을 제공합니다.
- Application Load Balancer로 고정 IP 주소 및 PrivateLink를 손쉽게 사용할 수 있습니다.

- 라우팅 대상 유형을 인스턴스로 설정(EC2 인스턴스로 분산시키기 때문에)

대상 등록

이는 대상 그룹을 생성하기 위한 선택적 단계입니다. 그러나 로드 밸런서가 이 대상 그룹으로 트래픽을 라우팅하려면 대상을 등록해야 합니다.

사용 가능한 인스턴스 (2/2)

인스턴스 필터링

<input checked="" type="checkbox"/>	인스턴스 ID	이름	상태	보안 그룹
<input checked="" type="checkbox"/>	i-021adf5969f21fb1b		실행 중	launch-wizard
<input checked="" type="checkbox"/>	i-02d5b0f304f7afee6		실행 중	launch-wizard

2개 선택됨

선택한 인스턴스를 위한 포트
선택한 인스턴스로 트래픽을 라우팅하기 위한 포트입니다.

80

1-65535(임의로 여러 포트 구분)

아래에 보류 중인 것으로 포함

- Web Server1과 Web Server2에 해당하는 인스턴스를 선택한 뒤 '아래에 보류 중인 것으로 포함' 설정 (인스턴스를 즉시 등록하지 않고, 오토 스케일링으로 추가될 인스턴스까지 포함할 수 있도록 준비)

등록된 대상 (2)

Q 속성 또는 값을 기준으로 리소스 필터링

< 1 >

<input type="checkbox"/>	인스턴스 ID	이름	포트	영역	상태 확인	상태 확인 세부 정보
<input type="checkbox"/>	i-0fbf8b614938242ba		80	ap-northeast-2c	unhealthy	Request timed out
<input type="checkbox"/>	i-0ad72e3bbd5b067a2		80	ap-northeast-2c	unhealthy	Request timed out

- 로드밸런서 리스너 및 규칙에서 생성한 대상 그룹에 있는 인스턴스들의 상태를 확인 (Web Server가 구축되지 않으면 unhealthy 상태)

```
Installed:
  httpd.x86_64 0:2.4.58-1.amzn2

Dependency Installed:
  apr.x86_64 0:1.7.2-1.amzn2          apr-util.x86_64 0:1.6.3-1.amzn2.0.1
  apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1  generic-logos-httpd.noarch 0:18.0.0-4.amzn2
  httpd-filesystem.noarch 0:2.4.58-1.amzn2  httpd-tools.x86_64 0:2.4.58-1.amzn2
  mailcap.noarch 0:2.1.41-2.amzn2          mod_http2.x86_64 0:1.15.19-1.amzn2.0.1

Complete!
```

```
[ec2-user@ip-172-31-32-118 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-32-118 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-12-08 04:44:07 UTC; 3min ago
     Docs: man:httpd.service(8)
  Main PID: 1221 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0; Requests served 0/0"
    CGroup: /system.slice/httpd.service
            └─1221 /usr/sbin/httpd -DFOREGROUND
              └─1222 /usr/sbin/httpd -DFOREGROUND
                └─1223 /usr/sbin/httpd -DFOREGROUND
                  └─1224 /usr/sbin/httpd -DFOREGROUND
                    └─1225 /usr/sbin/httpd -DFOREGROUND
                      └─1226 /usr/sbin/httpd -DFOREGROUND

Dec 08 04:44:07 ip-172-31-32-118.ap-northeast-2.compute.internal
Dec 08 04:44:07 ip-172-31-32-118.ap-northeast-2.compute.internal
[ec2-user@ip-172-31-32-118 ~]$
```

- 아래 명령어로 웹서버 설치 및 시작
 - sudo yum install httpd

- sudo systemctl start httpd
- sudo systemctl status httpd

```
[ec2-user@ip-172-31-32-118 ~]$ sudo cp /usr/share/httpd/noindex/index.html /var/www/html/index.html
[ec2-user@ip-172-31-32-118 ~]$
```

- 아래 명령어로 웹서버 페이지 구성 파일을 복사
 - sudo cp /usr/share/httpd/noindex/index.html /var/www/html/index.html

대상 집계	정상	비정상	사용되지 않음	초기	드래이닝
0 이상	1	1	0	0	0

가용 영역별 대상 배포

이래의 등록된 대상 책임자가 적용된 해당 필드를 보려면 이 테이블에서 값을 선택합니다.

대상 모니터링 상태 검사 속성 태그

등록된 대상 (2) 정보

이상 그룹은 지정된 프로파일을 및 모든 번호를 사용하여 등록된 개별 대상으로 요일을 라우팅합니다. 상태 확인은 대상 그룹의 상태 확인 설정에 따라 등록된 모든 대상에 대해 수행됩니다. 이상 탐지는 정상 대상이 3개 이상 있는 HTTP/HTTPS 대상 그룹에 자동으로 적용됩니다.

대상 필터링

인스턴스 ID	이름	포트	영역	상태 확인	상태 확인 세부 정보	이상 탐지
i-02d5b0f04f7afee6		80	ap-northeast-2c	Healthy		Normal
i-021adff5969f21fb1b		80	ap-northeast-2c	Unhealthy	Request timed out	Normal

- 다시 대상 그룹 화면으로 돌아와 인스턴스 상태를 확인

로드 밸런서 (1)

Elastic Load Balancing은 수신 트래픽의 변화에 따라 자동으로 로드 밸런서 용량을 확장합니다.

로드 밸런서 필터링

이름	DNS 이름	상태	VPC ID
		활성	vpc-0dc038dd95b6daf...

- 로드밸런서의 상태가 활성이 될때까지 대기하다가 활성이 되면, 좌측에 DNS 이름 복사해서 웹서버 접속

8. 구성표로 표현

