

로드 밸런싱(Load Balancing)

로드밸런싱이란

웹 트래픽 증가에 대한 처리방식

Scale-Up

Scale-Out

로드 밸런싱의 방식

Round Robin

Hash

Least Connection

Response Time

Sticky Session

Amazon Elastic Load Balancing 의 종류 및 유형

인터넷에 연결할 것인지 아닌지에 따라 두 가지로 나뉨

ELB의 주요 특징

상태 확인 서비스(Health Check)

Sticky Session

고가용성 구성

실습

EC2 2개 생성 : AWS 콘솔로 진행

SSH 접속 후 설정

usermod 사용법

usermod 옵션

다른 EC2에 동일한 작업 수행

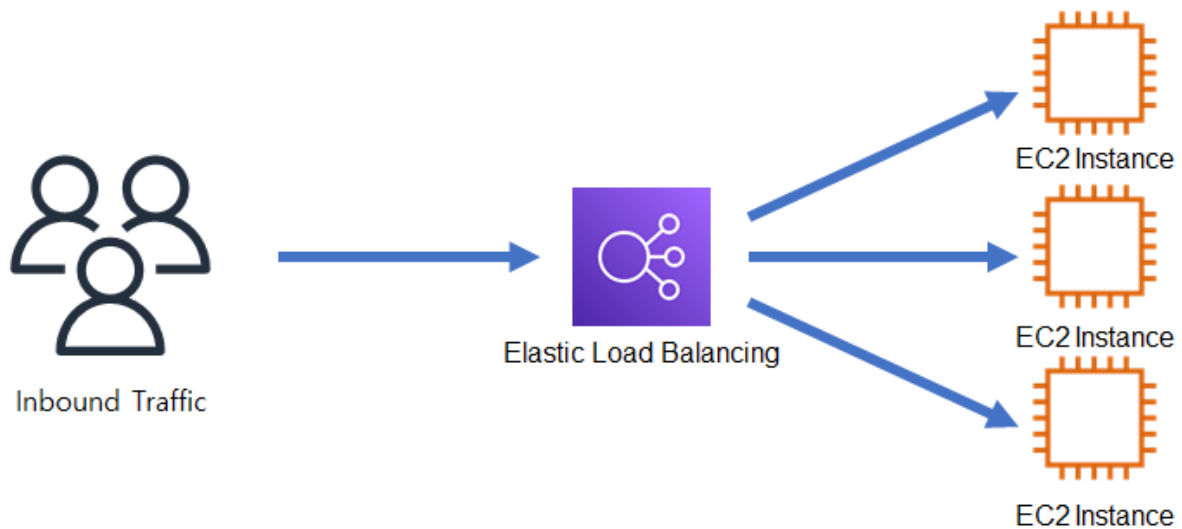
실습 : ELB 구성하기

AWS 콘솔에서 수행

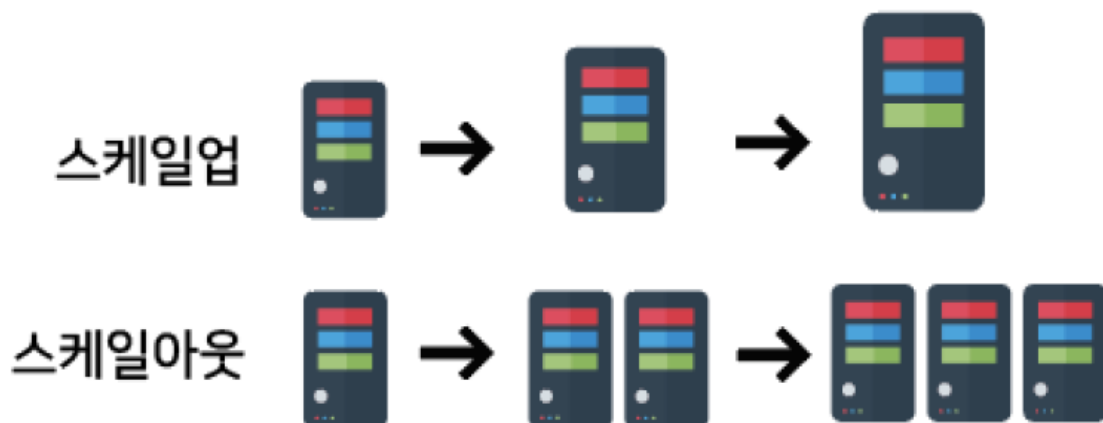
로드밸런싱이란

네트워크 기술의 일종으로 네트워크 트래픽을 하나 이상의 서버나 장비로 분산하기 위해 사용되는 기술로, 로드 밸런싱을 수행하는 소프트웨어나 하드웨어를 로드 밸런서라고 한다.

로드 밸런싱 서비스를 통해 외부에서 발생하는 많은 인터넷 트래픽을 여러 웹 서버나 장비로 부하를 분산하여 처리할 수 있다.



웹 트래픽 증가에 대한 처리방식



Scale-Up

- 기존보다 높은 성능을 보유한 웹서버로 시스템을 업그레이드 함으로써 문제를 해결하는 방식
- 필요로 하는 성능이 높아질수록 비용이 기하급수적으로 늘어나는 단점 발생
- 하나의 서버에서 웹 서비스를 제공하여 서버 중지 및 장애로 인해 서비스 가용성에 문제가 발생할 수 있다.

Scale-Out

- Cluster로 구성하는 경우 Cluster 내 하나의 노드에 문제가 발생하여도 웹 서비스가 중단되지 않으므로 가용성이 높은 웹 서비스를 구성할 수 있다.
 - 로드 밸런싱은 Scale-Out 방식의 서비스 구성에 주로 사용되며, 네트워크 트래픽을 웹 서비스의 Port 단위로 제어하고, 트래픽을 분산 처리함으로써 높은 가용성과 부하 분산을 통한 고효율 웹 서비스를 제공한다.
-

로드 밸런싱의 방식

Round Robin

- Real 서버로의 Session 연결을 순차적으로 맺어주는 방식
- 연결되어 있는 Session 수에 상관없이 순차적으로 연결시키는 방식으로 Session에 대한 보장을 제공하지 않는다.

Hash

- Hash 알고리즘을 이용한 로드 밸런싱 방식
- Client와 Server 간에 연결된 Session을 계속 유지해 주는 방식으로 Client가 특정 Server로 연결된 이후 동일 서버로만 연결되는 구조로 Session에 대한 보장을 제공한다.

Least Connection

- Session 수를 고려하여 가장 작은 Session을 보유한 서버로 Session을 맺어주는 연결 방식
- Session에 대한 보장을 제공하지 않는다.

Response Time

- 서버 간의 Resource와 Connection의 차이가 있는 환경에서 사용되는 방식으로 응답시간을 고려하여 빠른 응답시간을 제공하는 서버로 Session을 맺어주는 방식이며,

Session에 대한 보장을 제공하지 않는다.

Sticky Session

- 세션 고정 기능
- 하나의 요청이 특정 EC2로 들어와 세션이 생성되었다가 종료되고, 들어왔던 요청이 다시 들어올 경우 이미 연결되었던 특정 EC2로 전달하는 기능
- ALB와 CLB는 쿠키(Cookie)를 활용하며, NLB는 Source IP를 기억하는 방법을 사용

Amazon Elastic Load Balancing 의 종류 및 유형

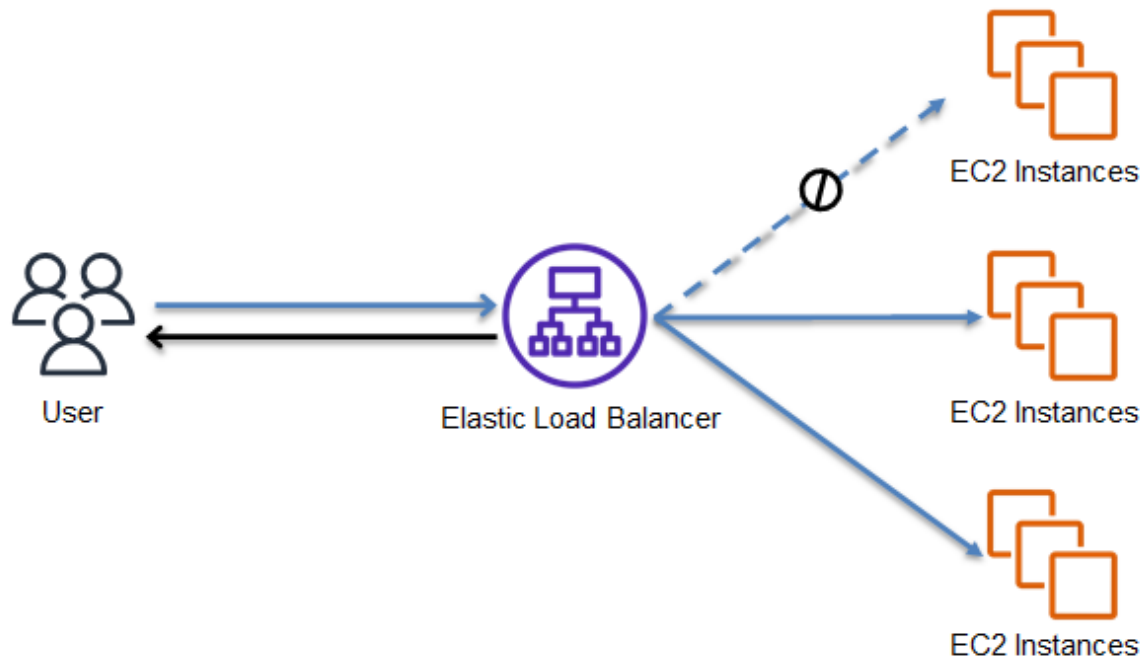
구분	내용	적합 서비스
Application Load Balancer	- OSI 모델 7계층에서 작동하며, HTTP, HTTPS와 같은 고급 로드 밸런싱 서비스에 적합 - 마이크로 서비스 및 컨테이너 기반 애플리케이션, 최신 애플리케이션 서비스에 최적화된 로드 밸런싱 제공 - SSL/TLS 암호화 및 프로토콜을 사용하여 보안성 보장	HTTP/HTTPS 서비스
Network Load Balancer	- OSI 모델 4계층에서 작동하며, TCP 트래픽의 로드 밸런싱 서비스에 적합 - 짧은 지연 시간과 초당 수백만 개의 요청 처리가 가능하며, 가용 영역당 1개의 정적 주소를 사용하면서 트래픽의 변동이 심한 서비스에 최적화	TCP 트래픽 로드 밸런싱 및 짧은 지연 시간
Classic Load Balancer	- OSI 모델 4계층, 3계층에서 작동 - EC2-Classic 네트워크 내에 구축된 애플리케이션을 대상으로 제공	EC2-Classic 네트워크 구축된 애플리케이션

인터넷에 연결할 것인지 아닌지에 따라 두 가지로 나뉨

항목	External Elastic Load Balance	Internal Elastic Load Balance
인터넷 연결	연결 가능	연결 불가
사용 가능 IP	Public IP, Private IP	Private IP
접속 가능 영역	인터넷, VPC 내부	VPC 내부

ELB의 주요 특징

상태 확인 서비스(Health Check)



Elastic Load Balancing은 ELB와 연결된 인스턴스의 연결 상태를 수시로 체크하여 인스턴스의 OS나 애플리케이션의 문제로 인해 연결 장애나 서비스 가능 여부에 대한 Health Check를 지속적으로 수행한다.

이러한 Health Check가 실패하는 경우 해당 인스턴스로 트래픽을 전달하지 않는다.

이를 위해 HTTP나 HTTPS 상태 확인 빈도, 실패 임계치, 성공 시 응답 코드를 임의 설정 가능하며, 자세한 상태 확인과 실패 원인은 API를 통해 확인은 물론 AWS 콘솔에도 표시된다.

TCP방식으로 Health Check를 수행하는 경우 Port나 오픈 여부 및 연결 가능 여부를 확인하며, HTTP나 HTTPS 방식은 특정 웹 페이지의 접속 시도에 따른 응답 코드(200)가 정상 반환 여부를 확인해서 Health Check 성공/실패 여부를 판단한다.

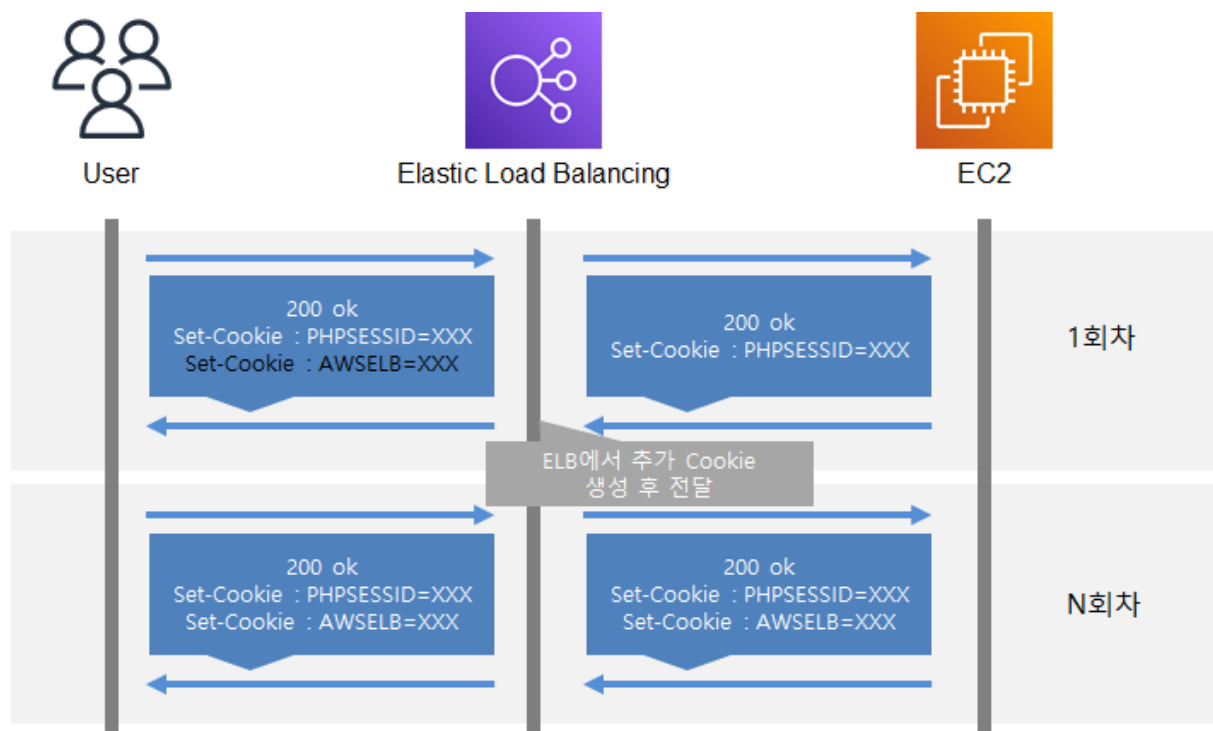
Sticky Session

Elastic Load Balancing을 통해 트래픽을 부하 분산하는 경우 기본적으로 Round Robin 방식으로 트래픽을 분산한다. 이 경우 한 번 연결된 Session은 다음 연결 시 그대로 연결되지

않으며, 다음 번 연결 시 다른 인스턴스로 연결될 수 있어서 애플리케이션의 Session을 유지할 수 없게 된다.

특히 세션 유지가 필요한 백오피스 웹사이트(그룹웨어 등)의 경우 연결이 끊어지거나 웹사이트의 로그인 및 인증 정보를 유지할 수 없게 된다.

Sticky Session을 사용하면 처음 연결된 Client에 별도의 Http 기반의 쿠키 값을 생성하여 다음 번 연결 요청에 대해 처음 접속했던 서버로 계속 연결하도록 트래픽을 처리하게 된다.



고가용성 구성

Elastic Load Balancing은 단일 가용 영역 또는 여러 가용 영역에 있는 여러 대상에 걸쳐 트래픽을 자동으로 분산할 수 있다. 특히 고가용성 구성을 위해 Router53와 같은 Amazon Web Service의 다른 서비스와의 연계를 통해 가용성 서비스를 제공할 수 있다.

실습

EC2 2개 생성 : AWS 콘솔로 진행

SSH 접속 후 설정

```
[ec2-user@ip-172-1-1-1 ~]$ sudo yum install httpd # 웹서버 설치
[ec2-user@ip-172-1-1-1 ~]$ sudo service httpd start # 웹서버 실행
[ec2-user@ip-172-1-1-1 ~]$ sudo chkconfig httpd on # 서버 부팅 시 웹서버 자동 실행

# ec2-user로 웹서버의 기본 루트페이지의 파일 수정 가능하도록 권한 설정
[ec2-user@ip-172-1-1-1 ~]$ sudo groupadd www # www 그룹 생성
[ec2-user@ip-172-1-1-1 ~]$ sudo usermod -a -G www ec2-user # ec2-user을 www 그룹에 포함
[ec2-user@ip-172-1-1-1 ~]$ exit
```

usermod 사용법

```
# usermod [option] 사용자 계정
# 홈 디렉토리 관련없이 계정이름만 변경
$ usermod -l <새로운계정> <기존계정>

# 홈 디렉토리 변경
# 해당 디렉토리를 미리 만들어 두어야 한다.
$ usermod -d /<홈디렉토리>/<새로운계정> <기존계정>
```

usermod 옵션

옵션	의미
-d (--home)	사용자의 홈 디렉터리를 변경한다. 변경할 디렉터리는 미리 생성되어야 한다. -m과 같이 사용하면 사용하던 홈 디렉터리 파일과 디렉터리도 같이 옮겨준다. The user's new login directory. (AIX UNIX, LINUX 공통)
-m	사용자의 홈 디렉터리 변경 시 기존에 사용하던 파일 및 디렉터리를 옮겨주는 옵션으로 -d와 함께 쓰인다. Moves the contents of the user's current home directory to the new home directory. Only used with the -d flag (AIX UNIX, LINUX 공통)
-g	사용자의 그룹을 변경한다. (AIX UNIX, LINUX 공통)
-s	사용자의 셸을 변경한다. (AIX UNIX, LINUX 공통)
-u	사용자의 UID 값을 변경한다. (AIX UNIX, LINUX 공통)
-e	계정만기일을 변경한다. YYYY-MM-DD 또는 MM/DD/YY 형태로 지정한다. (AIX UNIX, LINUX 공통)
-f	패스워드 만기일이 지난 후 패스워드에 Lock을 설정할 유예기간을 지정한다.
-c	사용자의 간단한 정보를 입력하거나 변경한다. (AIX UNIX, LINUX 공통) The new value of the user's password file comment field. It is normally modified using the chfn(1) utility.
-G	추가로 다른 그룹에 속하게 할 때 쓰인다. (AIX UNIX, LINUX 공통)
-a (--append)	-G옵션과 같이 사용하는 옵션으로 기존의 2차 그룹 이외에 추가로 2차 그룹을 지정할 때 사용한다. Add the user to the supplementary group(s). Use only with -G option.
-p	/etc/shadow의 2번째 필드인 암호화된 패스워드 값을 변경할 때 쓴다. 이 옵션 사용할 때는 암호화된 값으로 값을 지정해야 한다.
-l	사용자 아이디를 변경한다 (--login) (AIX UNIX, LINUX 공통)
-L	사용자의 패스워드에 LOCK을 걸어 로그인을 막는다. (--lock)
-U	사용자의 패스워드에 걸린 LOCK을 푼다. (--unlock)

다시 SSH 접속

```
[ec2-user@ip-172-31172-1-1-1 ~]$ cd /var/www
[ec2-user@ip-172-1-1-1 www]$ ls -l
total 0
drwxr-xr-x 2 root root 6 Dec 30 21:40 cgi-bin
drwxr-xr-x 2 root root 6 Dec 30 21:40 html

# 서버의 접속 권한 수정
[ec2-user@ip-172-1-1-1 www]$ sudo chown -R root:www /var/www

[ec2-user@ip-172-1-1-1 www]$ ls -l
total 0
drwxr-xr-x 2 root www 6 Dec 30 21:40 cgi-bin
drwxr-xr-x 2 root www 6 Dec 30 21:40 html

# 2775 : 그룹에 속한 사용자가 실행 시 소유자의 권한으로 실행
[ec2-user@ip-172-1-1-1 www]$ sudo chmod 2775 /var/www/html
[ec2-user@ip-172-1-1-1 www]$ ls -l
total 0
```



```
drwxr-xr-x 2 root www 6 Dec 30 21:40 cgi-bin
drwxrwsr-x 2 root www 6 Dec 30 21:40 html

[ec2-user@ip-172-1-1-1 www]$ cd html
[ec2-user@ip-172-1-1-1 html]$ cat > index.html
Test Website - EC2 Instance 1
^C
```

다른 EC2에 동일한 작업 수행

실습 : ELB 구성하기

AWS 콘솔에서 수행

