

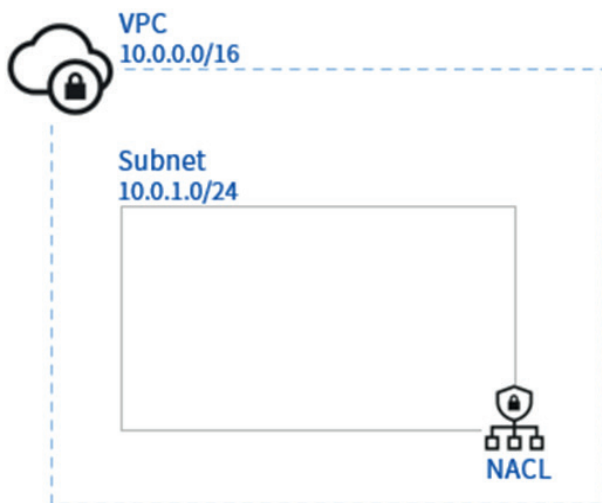
NCP 를 활용한 클라우드 사용 기초 튜토리얼

Hands-on Lab

NCP 환경에서의 클라우드 기초 이론 지식을 습득해 보고,
IT 환경에서의 클라우드 접목 방법을 배워 보면서,
네이버 클라우드 플랫폼을 이용하여고가용성을 보장하는
웹서버를 만들어 보는 핸즈온 세션입니다.

1.1 웹 서버를 위한 네트워크 환경 구성하기

VPC(Virtual Private Cloud) 는 퍼블릭 클라우드 상에서 제공되는 고객 전용 사설 네트워크를 의미 합니다.
Subnet(Subnet)은 VPC 네트워크 공간을 세분화하여 사용할 수 있는 기능입니다.
네트워크 환경을 구성하고 네트워크 환경에 보안성을 높이는 NACL 개념을 습득해봅니다.



Spec

VPC

- IP 주소 범위: 10.0.0.0/16

Subnet

- VPC: 생성한 VPC 선택
- IP 주소 범위 : 10.0.1.0/24
- 가용 Zone: KR-2
- Internet Gateway 전용 여부: Y
- 용도: 일반

Task

1. VPC 를 생성합니다.
2. Default 로 생성된 ACL 을 확인해봅니다.
3. Subnet 을 생성합니다.

Hints

VPC 및 Subnet <https://guide-fin.ncloud-docs.com/docs/networking-vpc-vpcdetailedsubnet>

ACL <https://guide-fin.ncloud-docs.com/docs/networking-vpc-vpcsecurity>

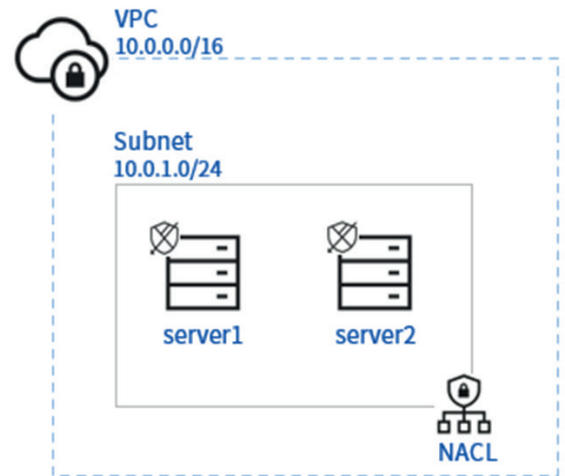
1.2 웹 서버 구성하기

Server는 물리적인 서버 자원을 구매하지 않고 클라우드 환경에서 서버를 생성하고 사용할 수 있게 해주는 서비스입니다.

ACG(Access Control Group) 서버 간 네트워크 접근 제어 및 관리를 할 수 있는 IP/Port 기반 필터링 방화벽 서비스입니다.

Server Image 를 사용시 현재 사용중인 서버의 이미지를 생성해 서버 현재의 상태를 저장하고, 저장한 내 서버 이미지를 이용하여 쉽게 생성할 수 있습니다.

Server1 생성 후 이미지를 통해 Server2 를 생성해보고, ACG 를 통해 생성 된 서버에 붙여 보안성을 향상시켜봅니다.



Spec

ACG

- 기존 VPC 사용

ACG 설정

- Inbound 규칙설정
 - 프로토콜 : ICMP, 접근 소스 : 0.0.0.0/0 입력 후, "+추가"
 - 프로토콜 : TCP, 접근 소스 : 0.0.0.0/0 허용 포트 (서비스) : 80 입력 후 "+추가"
 - 프로토콜 : TCP, 접근 소스 : mylp 허용 포트 : 22 선택 후, "+추가"
- Outbound 규칙설정
 - 프로토콜 : ICMP, 목적지 : 0.0.0.0/0 입력 후, "+추가"
 - 프로토콜 : TCP, 목적지 : 0.0.0.0/0 허용 포트 : 1-65535 입력 후, "+추가"
 - 프로토콜 : UDP, 목적지 : 0.0.0.0/0 허용 포트 : 1-65535 입력 후, "+추가"
- Init Script
 - OS 타입 : Linux
 - Script :

```
#!/bin/bash
yum -y remove mariadb-libs
yum -y install httpd php mysql php-mysql
systemctl enable httpd
cd /var/www/html
wget http://211.249.50.207/lab/lab.tgz
tar xvfz lab.tgz
cat phpadd >> /etc/httpd/conf/httpd.conf
systemctl start httpd
```

- Server1
 - 서버 이미지 이름: Centos-7.8-64
 - 서버 타입: Standard vCPU 2EA, Memory 8GB, Disk 50GB
 - VPC, Subnet: 기존 생성한 리소스 선택
 - Network Interface: IP 10.0.1.101
 - Scrip: 생성한 Init Scrip 선택
- 이미지를 통한 서버 Server2
 - Network Interface IP: 10.0.1.102
 - Script 선택: 선택 안함
 - 나머지 기존 서버1 스펙과 동일하게 설정

Task

1. 웹 서버에 적용된 ACG의 Traffic 규칙을 설정합니다.
2. 웹 서버를 만들기 위한 Init Script 생성합니다.
3. Init Script를 사용하여 Spec에 맞춰 웹서버1을 생성합니다.
4. 공인 IP를 생성 후 만든 서버1 에 적용시켜봅니다.
5. 공인 IP 로 웹 브라우저를 이용해 접근해봅니다.
6. 서버1로 이미지를 만들어봅니다.
7. 만든 이미지로 서버2를 생성해봅니다.

Hints

Server

<https://guide-fin.ncloud-docs.com/docs/compute-server-serveroverview>

ACG

<https://guide-fin.ncloud-docs.com/docs/security-acg-acgoverview>

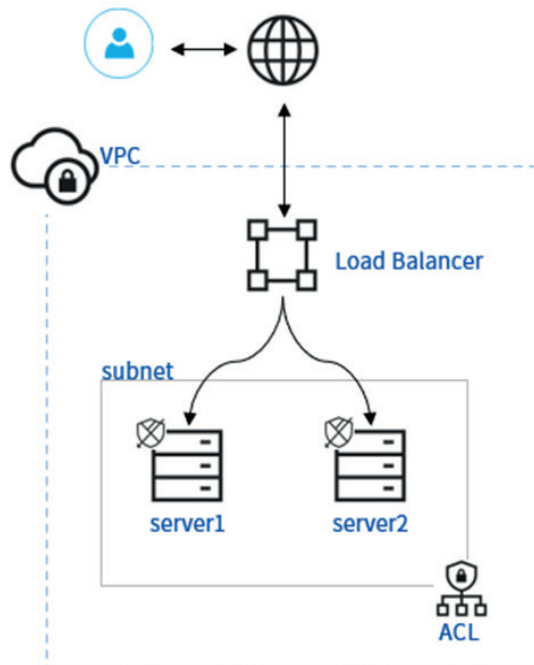
Server Image

<https://guide-fin.ncloud-docs.com/docs/compute-compute-5-1-v2>

1.3 Load Balancer 구성해보기

로드 밸런서는 서버 성능과 부하량을 고려하여 네트워크 트래픽을 다수의 서버로 분산해주는 서비스입니다. 연결된 서버 중 일부 서버에 장애가 발생하면 자동으로 다른 정상 서버로 부하를 배분해 무정지 서비스가 가능하게 해 서비스의 높은 안정성을 보장합니다.

로드 밸런서 리소스를 구성하는 방법을 알아보고, 로드 밸런싱을 확인해봅니다.



Spec

LB 를 위한 Subnet

- VPC: 기존 생성한 VPC
- IP 주소 범위: 10.0.2.0/24
- 가용 Zone: KR-2
- Network ACL: 기존 생성한 ACL
- Internet Gateway 전용 여부: N(private)
- 용도: LoadBalancer

Load Balancer

- 애플리케이션 로드밸런서
- Network: Public IP
- 부하 처리 성능: Small
- 대상 VPC: 기존 생성한 VPC
- 서브넷 선택: KR-2 / LB 용 Subnet
- 리스너: HTTP 80

Target Group

- Target 유형: VPC Server
- VPC: 생성한 VPC
- 프로토콜: HTTP
- 포트: 80
- Health Check 설정: HTTP Method HEAD 외 기본 설정
- Target 추가: 대상 서버 추가

Task

1. 로드 밸런서를 위한 Subnet 을 생성합니다.
2. Target Group 을 생성합니다.
3. 로드 밸런서를 생성합니다.
4. 로드 밸런서의 접속 정보 IP 를 통하여 접속하여 부하 분산 되는 것을 확인해봅니다.

Hints

Application Load Balancer

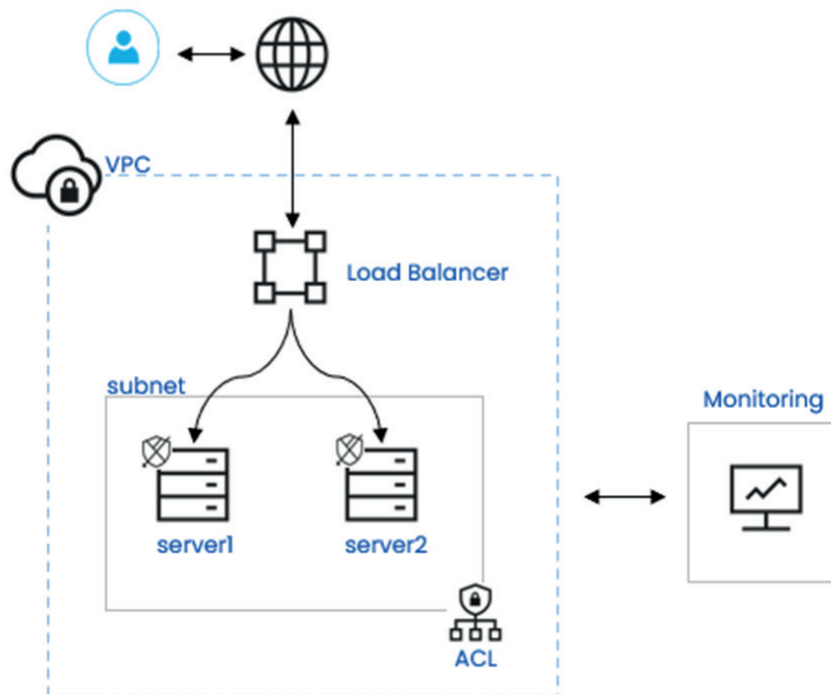
<https://guide-fin.ncloud-docs.com/docs/networking-loadbalancer-applicationlbconsole>

Target Group

<https://guide-fin.ncloud-docs.com/docs/networking-loadbalancer-targetgroupconsole>

1.4 모니터링 구성

대시보드를 통한 기본 Metric을 한눈에 볼 수 있으며, 이벤트 발생 및 통보의 기능을 제공하고 있습니다.
모니터링을 통한 서버 안정성을 확보할 수 있습니다.
대시보드를 통한 서버 메트릭 확인 및 알람 설정을 진행해봅니다.



Task

1. 콘솔에서 서버를 선택 후 메트릭을 확인해 봅니다.
2. Cloud Insight를 이용하여 Dashboard 에서 메트릭을 확인해봅니다.
3. 알람 설정을 진행해봅니다.
 - ① 알림 받을 대상을 추가해봅니다. Notification Recipient
 - ② Server(VPC) 에 대한 Event Rules 를 생성해봅니다. (CPU / MEM 등 자유로이 설정)

Hints

Cloud Insight

<https://guide-fin.ncloud-docs.com/docs/cloudinsight-cloudinsightoverview>

Event Rule

<https://guide-fin.ncloud-docs.com/docs/cloudinsight-use-eventrule>