OpenStack으로 AWS VPC를 어떻게 구현할까?

유영환

Speaker’s bio

유영환

kt cloud Core플랫폼팀

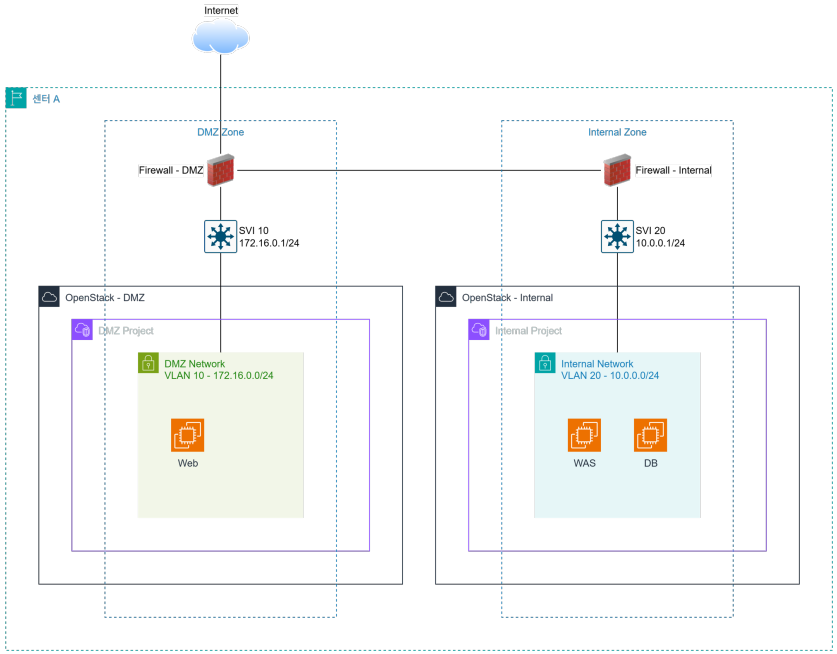
OpenStack 엔지니어

차세대 Block Service Technical Product Owner

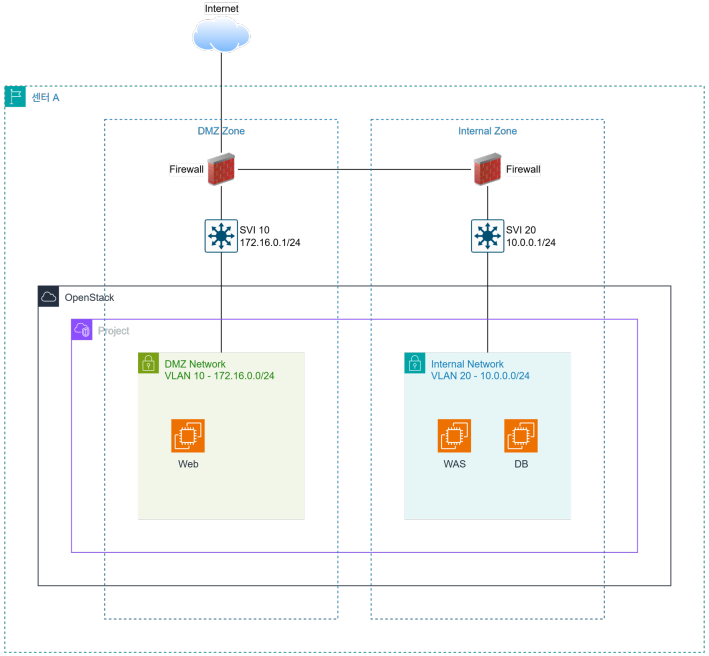
목차

2022년 프로젝트 수행 경험으로 비춰본 Private Cloud 구축 동향 (1/2)

DMZ 존, Internal 존 네트워크 분리 구성



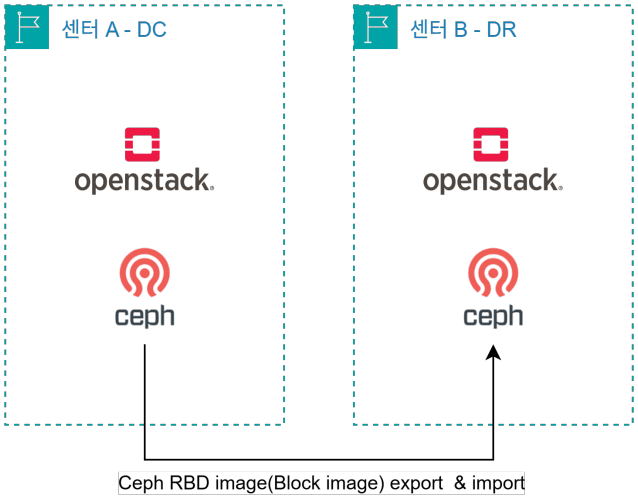
DMZ 존, Internal 존에 각각 OpenStack 설치하여 구성한 사례



OpenStack Nova, Cinder의 AZ를 이용해 1개의 OpenStack으로 구성한 사례

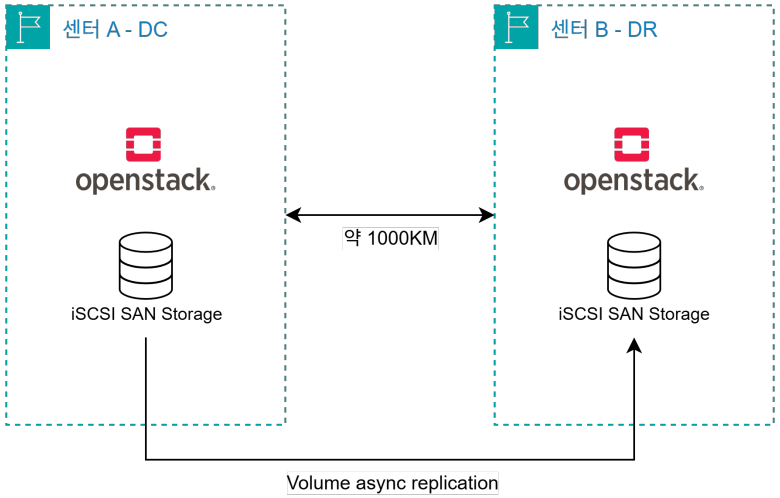
2022년 프로젝트 수행 경험으로 비춰본 Private Cloud 구축 동향 (2/2)

DR 구성 사례



국내 A 기관

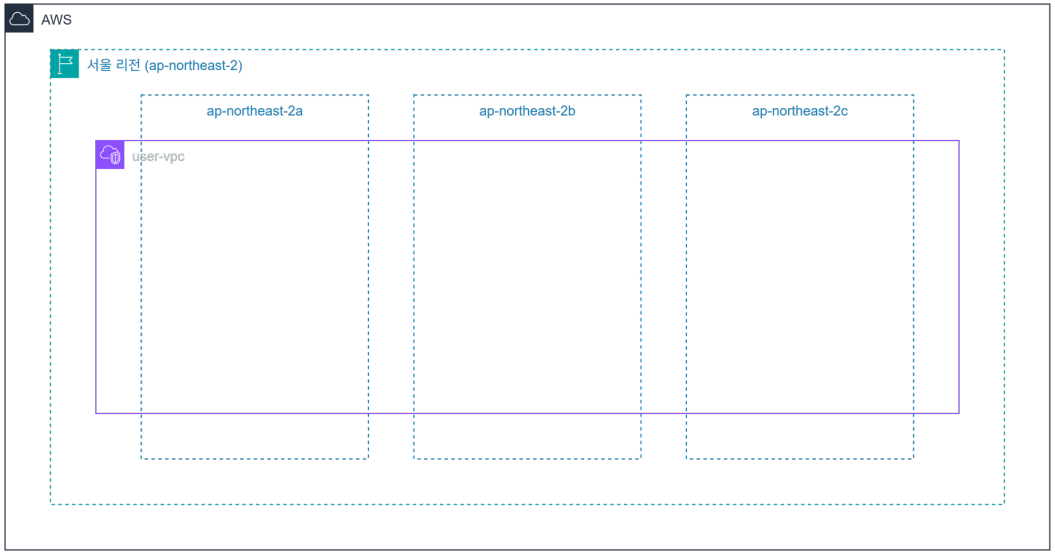
Ceph rbd 이미지를 이용해 데이터 동기화



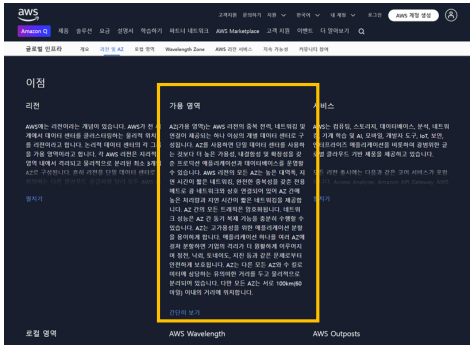
해외 B 기관

각 센터 SAN Storage 볼륨들을 스냅샷 기반 Async로 데이터 복제

AWS VPC에 대한 호기심



AWS VPC 아키텍처 모습

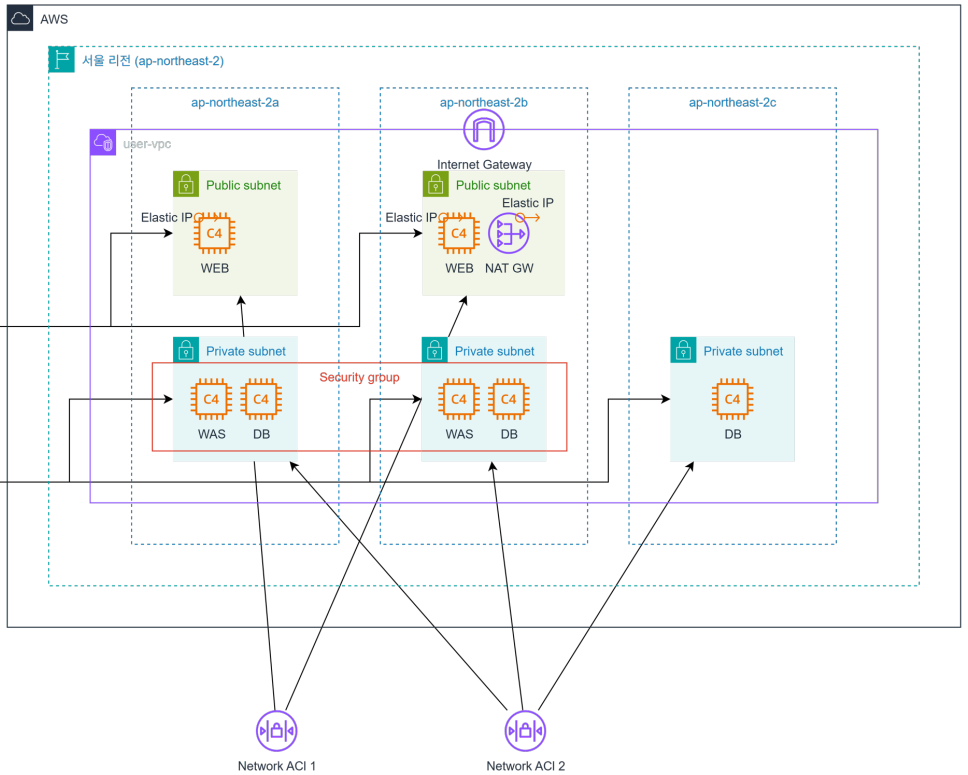


AWS 가용 영역(Availability Zone)

가용 영역

AZ(가용 영역)는 AWS 리전의 중복 전력, 네트워킹 및 연결이 제공되는 하나 이상의 개별 데이터 센터로 구성됩니다. AZ를 사용하면 단일 데이터 센터를 사용하는 것보다 더 높은 가용성, 내결함성 및 확장성을 갖춘 프로덕션 애플리케이션과 데이터베이스를 운영할 수 있습니다. AWS 리전의 모든 AZ는 높은 대역폭, 지연 시간이 짧은 네트워킹, 완전한 중복성을 갖춘 전용 메트로 광 네트워크와 상호 연결되어 있어 AZ 간에 높은 처리량과 지연 시간이 짧은 네트워킹을 제공합니다. AZ 간의 모든 트래픽은 암호화됩니다. 네트워크 성능은 AZ 간 동기 복제 기능을 충분히 수행할 수 있습니다. AZ는 고가용성을 위한 애플리케이션 분할을 용이하게 합니다. 애플리케이션 하나를 여러 AZ에 걸쳐 분할하면 기업의 격리가 더 원활하게 이루어지며 정전, 낙뢰, 토네이도, 지진 등과 같은 문제로부터 안전하게 보호됩니다. AZ는 다른 모든 AZ와 수 킬로미터에 상당하는 유의미한 거리를 두고 물리적으로 분리되어 있습니다. 다만 모든 AZ는 서로 100km(60마일) 이내의 거리에 위치합니다.

AWS VPC 분석



Route table 1	
10.0.0.0/16	local
0.0.0.0/0	igw

Route table 1

Route table 2	
10.0.0.0/16	local

Route table 2

- Region
- Availability Zone
- VPC
- Subnet
- Route table
- Internet Gateway
- Elastic IP
- NAT Gateway
- Security Group
- Network ACL

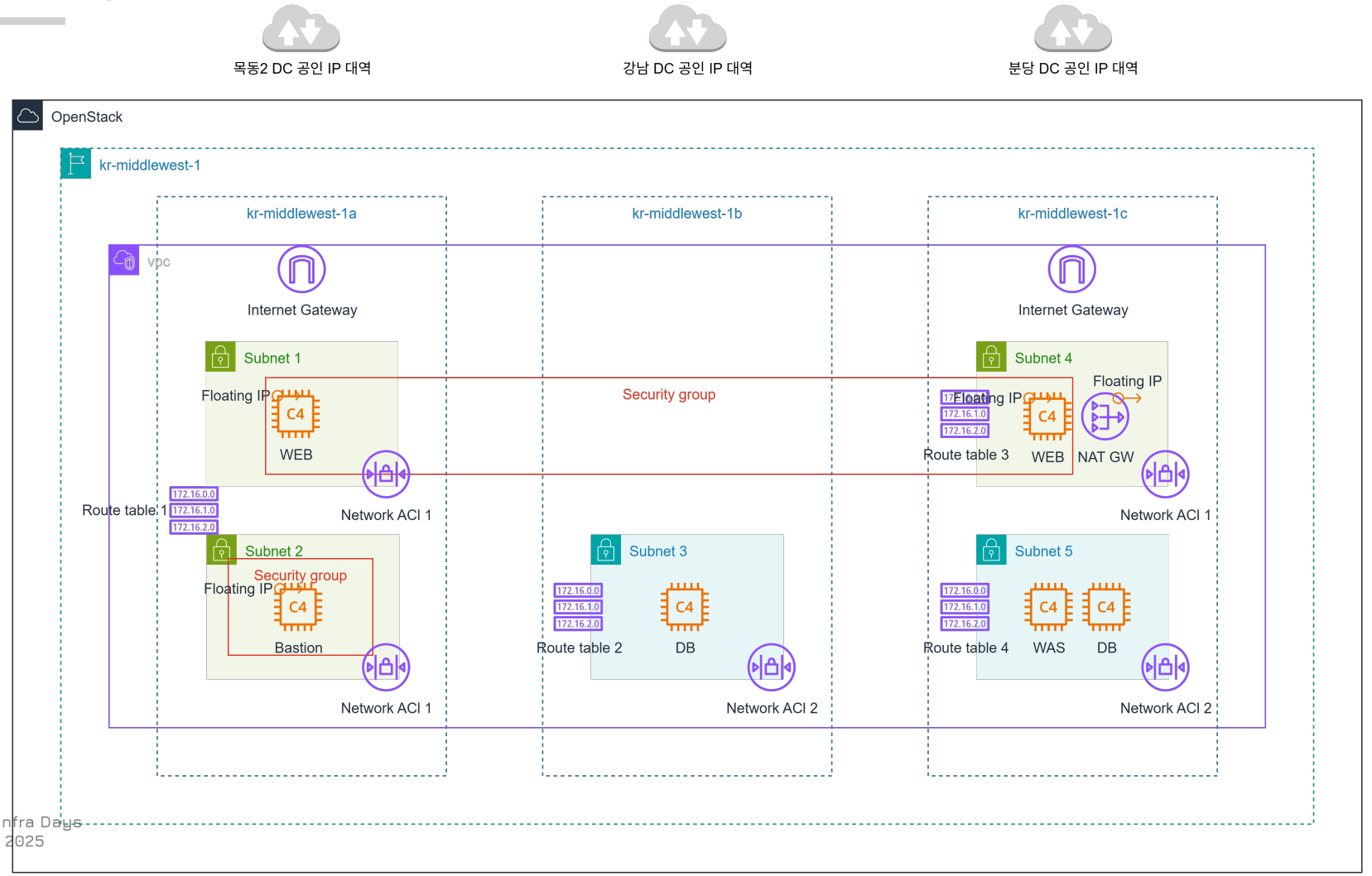
Inbound rule (Source CIDR)					
100	모든 트래픽	모두	모두	0.0.0.0/0	허용
*	모든 트래픽	모두	모두	0.0.0.0/0	거부

Outbound rule (Destination CIDR)					
100	모든 트래픽	모두	모두	0.0.0.0/0	허용
*	모든 트래픽	모두	모두	0.0.0.0/0	거부

Inbound rule (Source CIDR)					
100	모든 트래픽	모두	모두	0.0.0.0/0	허용
*	모든 트래픽	모두	모두	0.0.0.0/0	거부

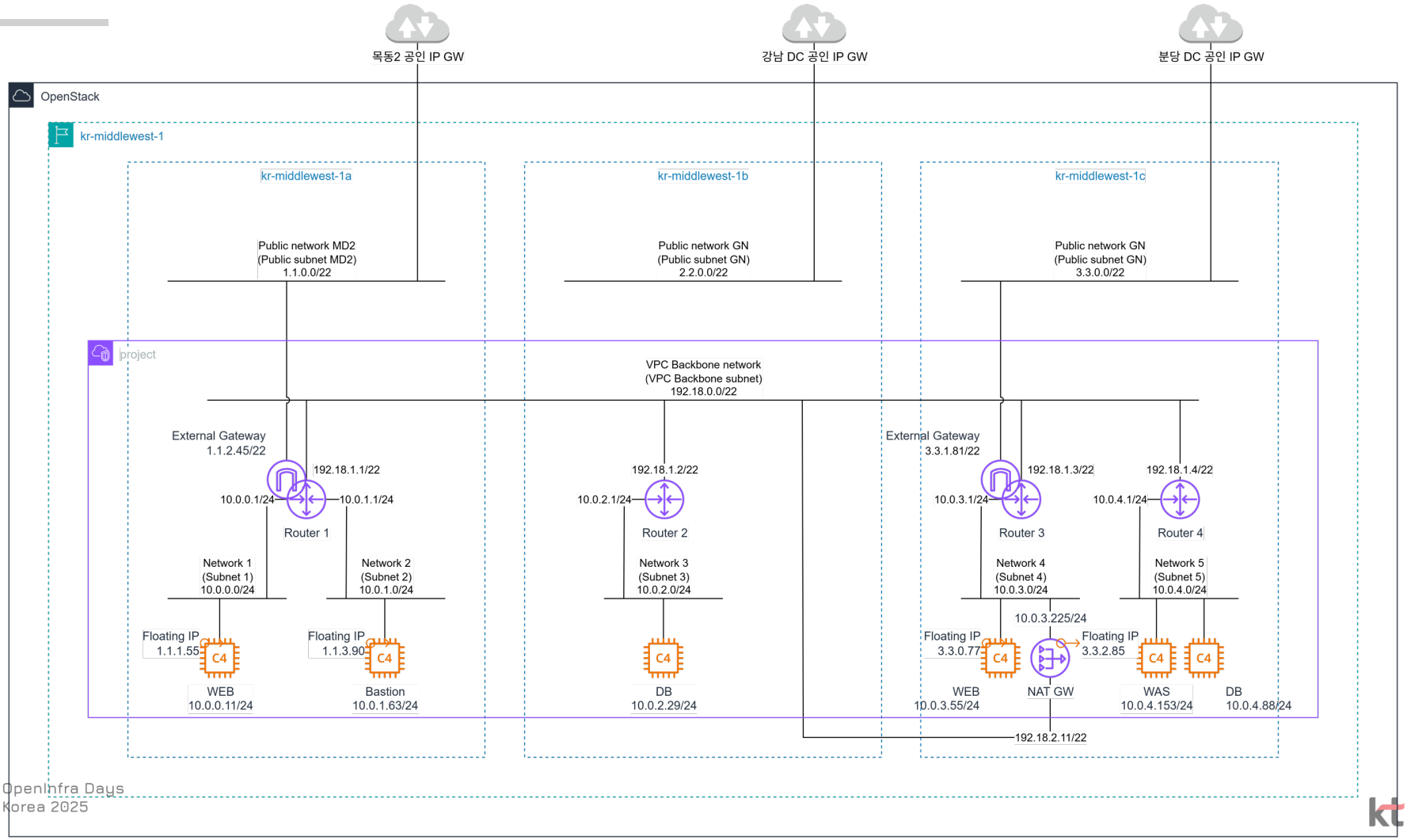
Outbound rule (Destination CIDR)					
100	모든 트래픽	모두	모두	0.0.0.0/0	허용
*	모든 트래픽	모두	모두	0.0.0.0/0	거부

사용자 입장에서 OpenStack VPC를 경험하면?

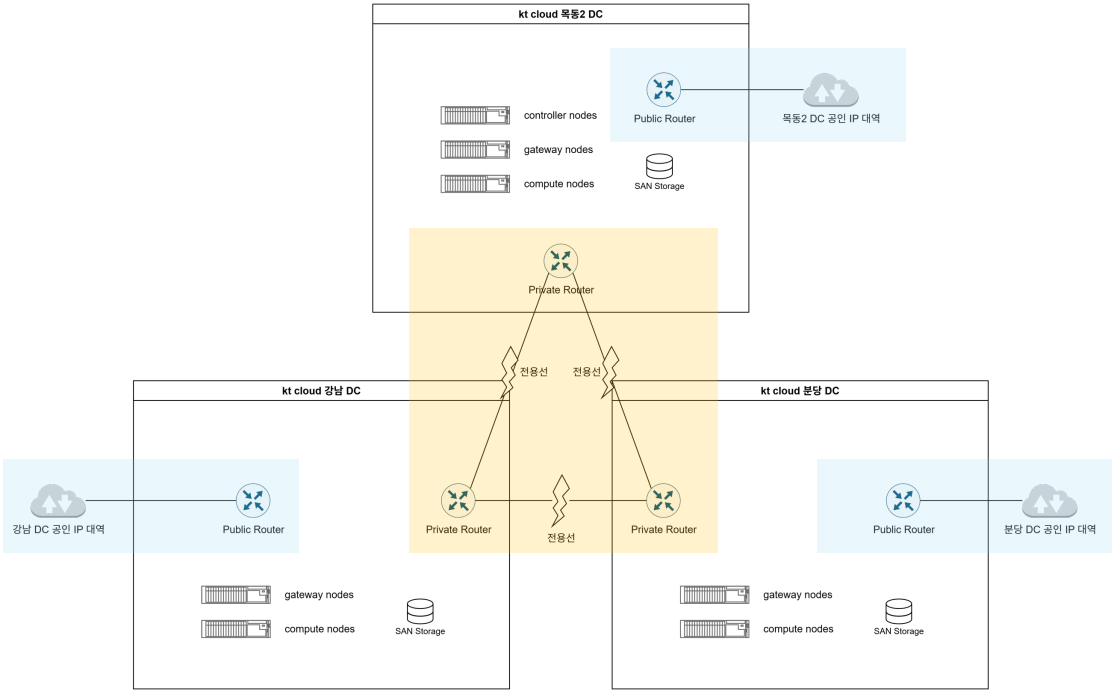


kt cloud

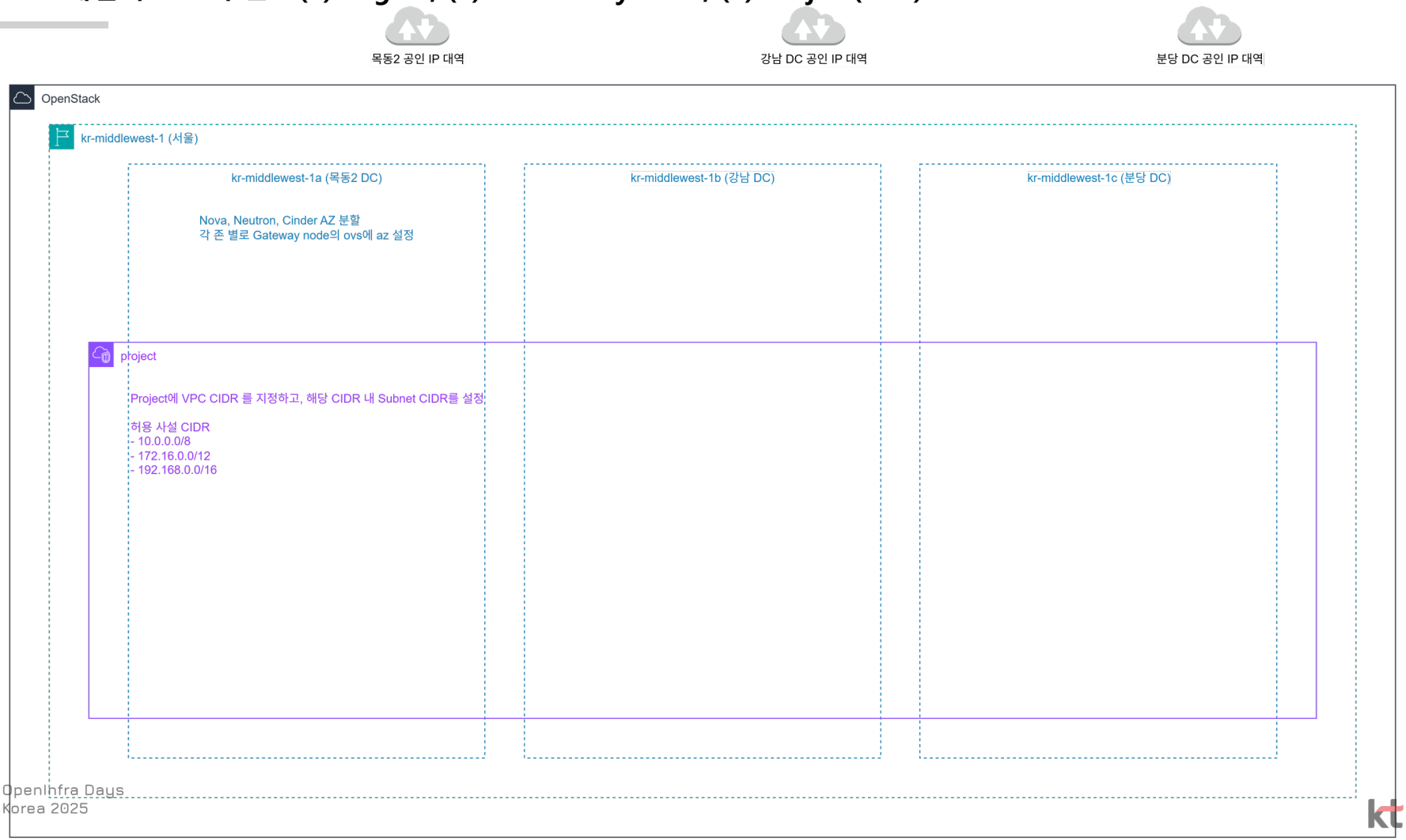
OpenStack 레벨의 VPC 구현



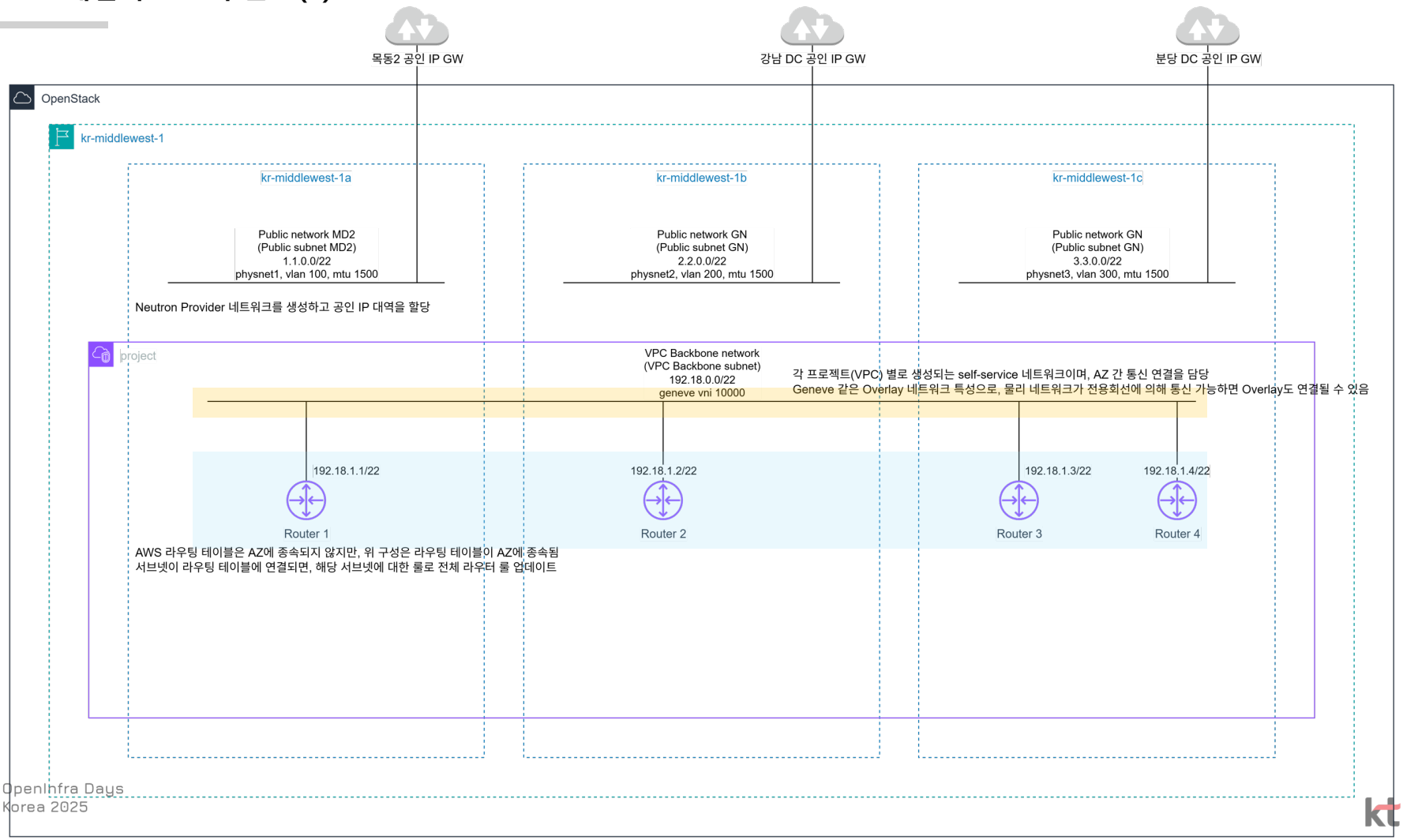
OpenStack 레벨의 VPC 구현 - (1) Region, (2) Availability Zone



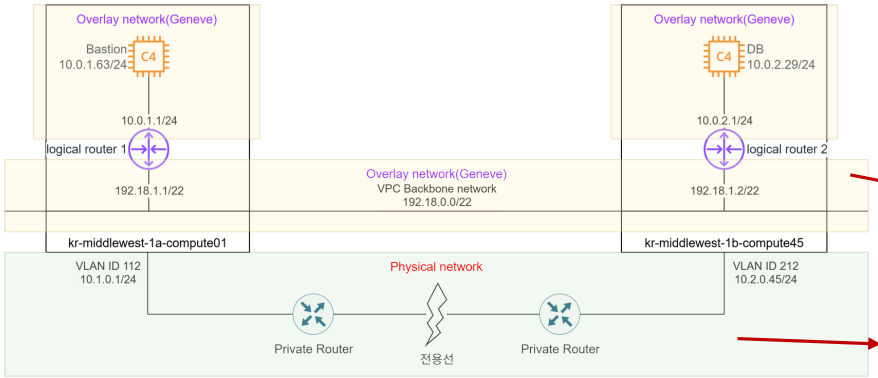
OpenStack 레벨의 VPC 구현 - (1) Region, (2) Availability Zone, (3) Project(VPC)



OpenStack 레벨의 VPC 구현 - (4) Route table



OpenStack 레벨의 VPC 구현 - (4) Route table [VPC 백본 네트워크]



Bastion VM에서 DB VM으로 데이터 전송 시 패킷 구조

Src MAC <홉마다 변경>	Dst MAC <홉마다 변경>	Src IP 10.1.0.1	Dst IP 10.2.0.45	Protocol UDP	Src Port <Src 랜덤>	Dst Port 6081	Geneve VNI 20000	InnerSrc MAC <lr 홉 변경>	InnerDst MAC <lr 홉 변경>	InnerSrc IP 10.0.1.1	InnerDst IP 10.0.2.1	Payload	FCS

물리 서버, 물리 네트워크 장치에 의해 처리

OVN(OVS)에서 처리
*OVN은 여러 OVS를 제어하는 컨트롤러 역할

Bastion VM에서 DB VM으로 데이터 전송 시 패킷 - tcpdump

compute01

```
20:55:33.058302 b6:c2:df:f7:b0:c0 > 06:75:34:7a:8c:e1, ethertype IPv4 (0x0800), length 156: (tos 0x0, ttl 64, id 48840, offset 0, flags [DF], proto UDP (17), length 142)
    10.1.0.1.53799 > 10.2.0.45.6081: [udp sum ok] Geneve, Flags [C], vni 0x3, proto TEB (0x6558), options [class Open Virtual Networking (OVN) (0x102) type 0x80(C) len 8 data 000b0018]
                fa:16:3e:83:21:7c > fa:16:3e:ac:1d:1d, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 62, id 15845, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.1.63 > 10.0.2.29: ICMP echo request, id 10, seq 3, length 64
```

compute45

```
20:55:15.069005 b6:c2:df:f7:b0:c0 > 06:75:34:7a:8c:e1, ethertype IPv4 (0x0800), length 156: (tos 0x0, ttl 64, id 48840, offset 0, flags [DF], proto UDP (17), length 142)
    10.1.0.1.53799 > 10.2.0.45.6081: [bad udp cksum 0x58db -> 0x190c!] Geneve, Flags [C], vni 0x3, proto TEB (0x6558), options [class Open Virtual Networking (OVN) (0x102) type 0x80(C) len 8 data 000b0018]
                fa:16:3e:83:21:7c > fa:16:3e:ac:1d:1d, ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 62, id 15845, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.1.63 > 10.0.2.29: ICMP echo request, id 10, seq 3, length 64
```

OpenStack 레벨의 VPC 구현 - (4) Route table [OpenStack Router]

사용자 라우트 테이블

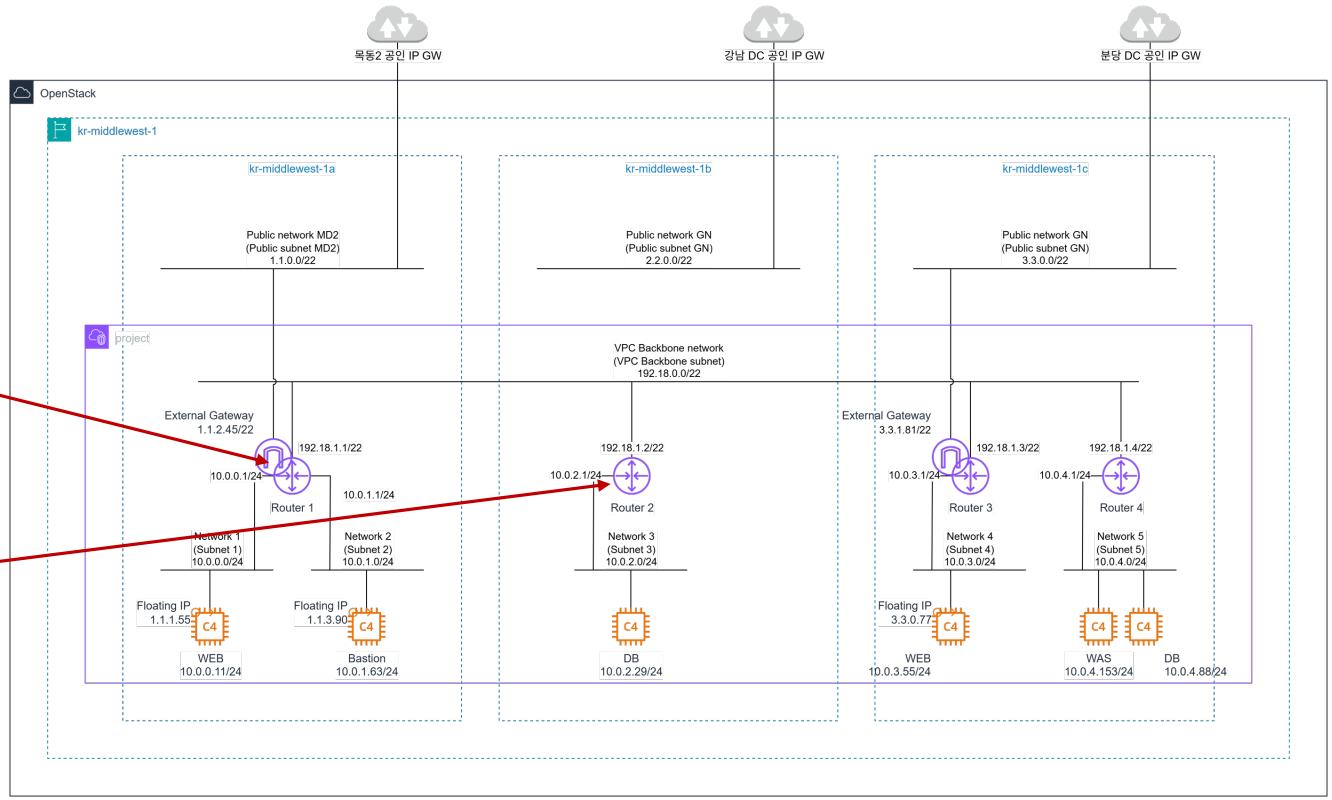
Route table 1	
10.0.0.0/16	local
0.0.0.0/0	igw

Route table 2	
10.0.0.0/16	local

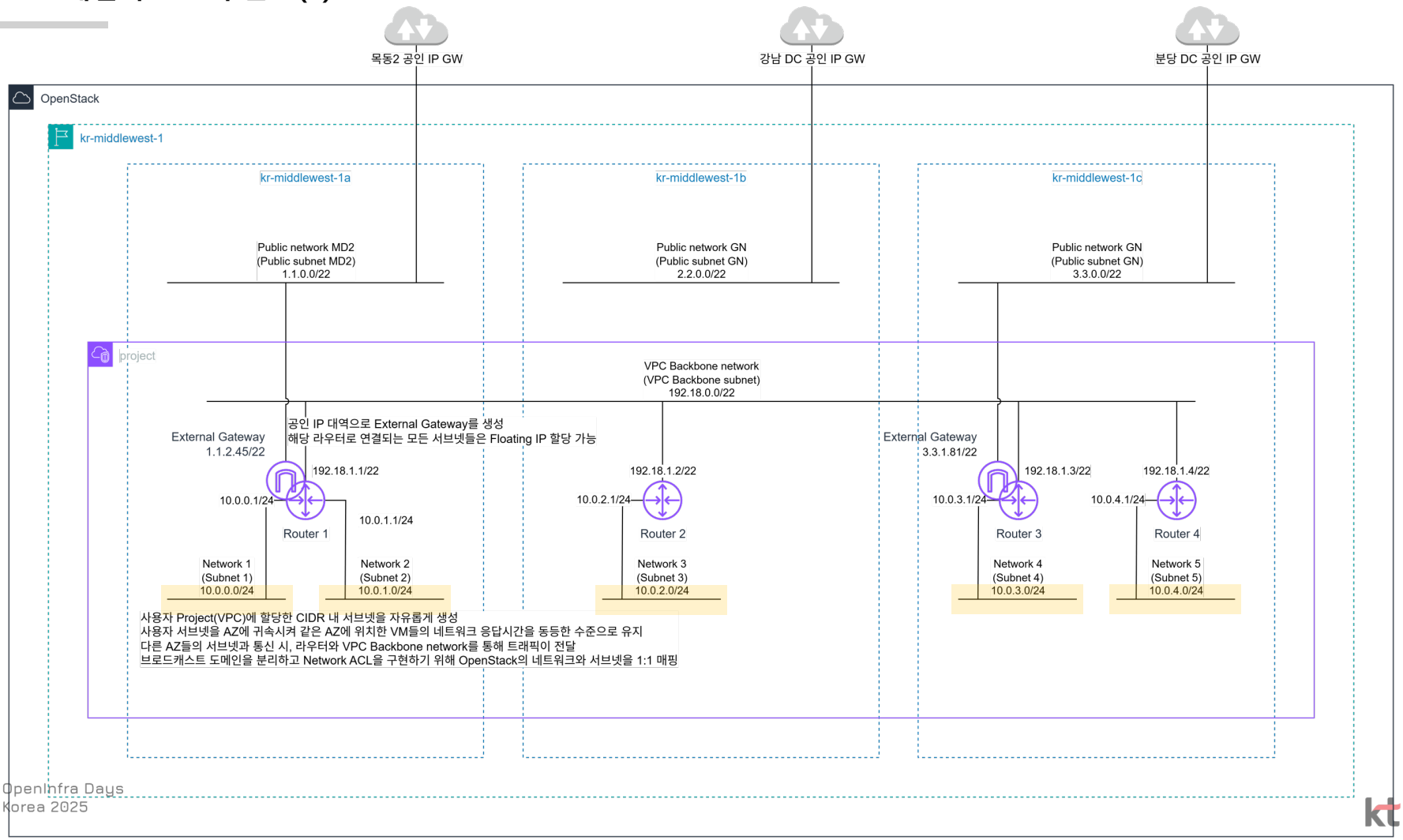
OpenStack 라우터 룰

Router 1	
10.0.2.0/24	192.18.1.2
10.0.3.0/24	192.18.1.3
10.0.4.0/24	192.18.1.4

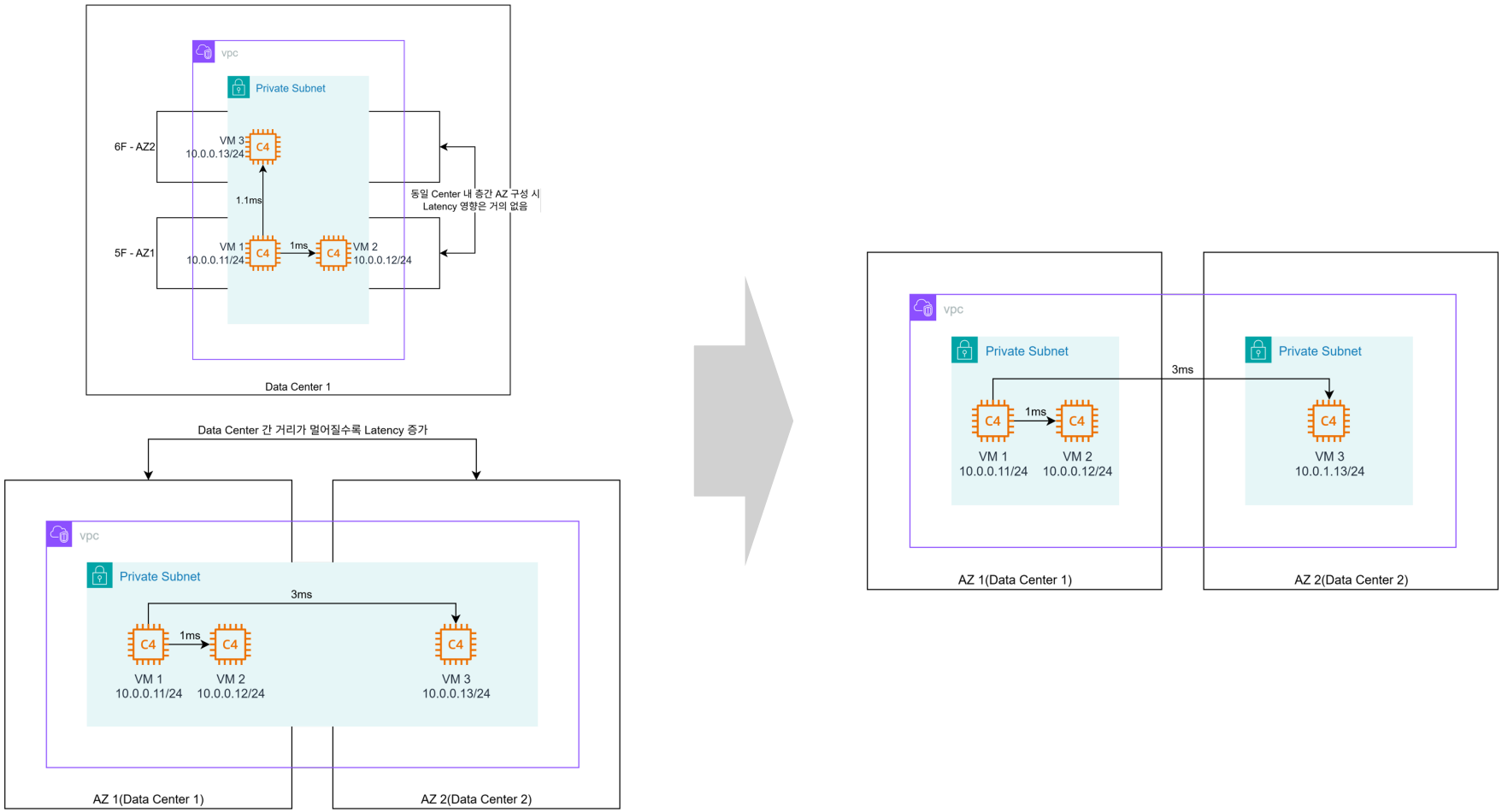
Router 2	
10.0.0.0/24	192.18.1.1
10.0.1.0/24	192.18.1.1
10.0.3.0/24	192.18.1.3
10.0.4.0/24	192.18.1.4



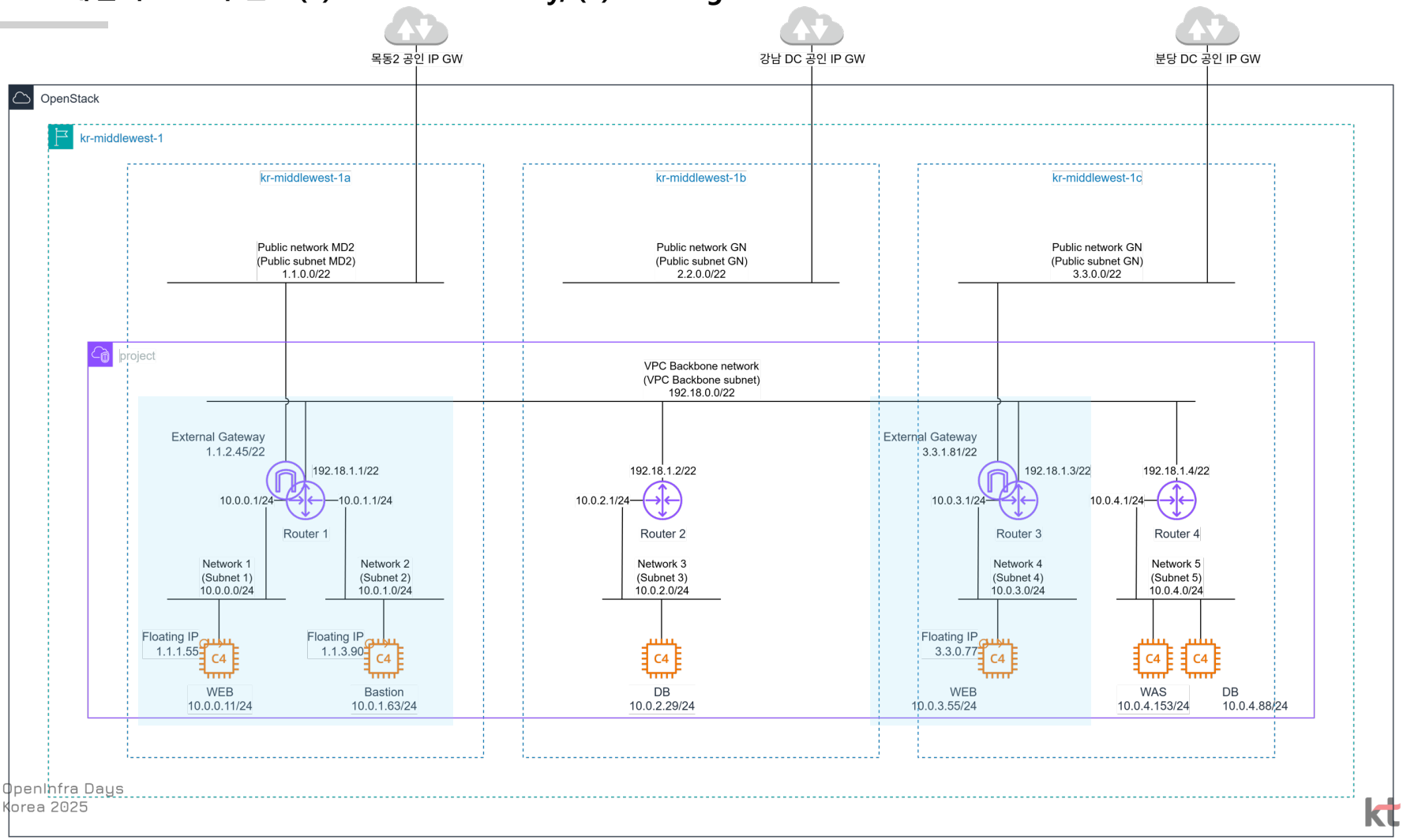
OpenStack 레벨의 VPC 구현 - (5) Subnet



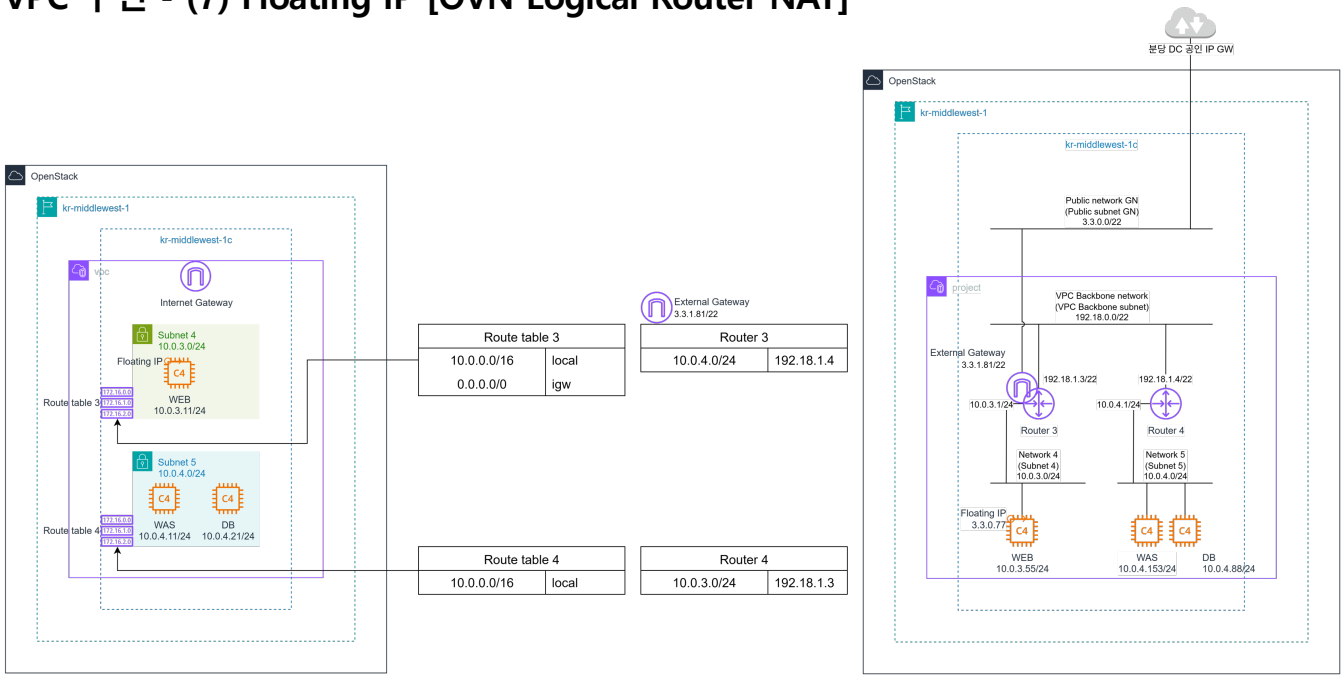
OpenStack 레벨의 VPC 구현 - (5) Subnet [Subnet의 지역성]



OpenStack 레벨의 VPC 구현 - (6) External Gateway, (7) Floating IP



OpenStack 레벨의 VPC 구현 - (7) Floating IP [OVN Logical Router NAT]



① → `[root@controller01 ~]# ovn-nbctl lr-list | grep fb96dd03-48cc-49f0-b76f-52e8311a6082`

```
1a5f0ed9-2425-4f48-ab81-a3b22465ee72 (neutron-fb96dd03-48cc-49f0-b76f-52e8311a6082)
```

② → `[root@controller01 ~]# ovn-nbctl lr-route-list 1a5f0ed9-2425-4f48-ab81-a3b22465ee72`

```
IPv4 Routes
Route Table <main>:
               10.0.4.0/24       192.18.1.4        dst-ip
               0.0.0.0/0         3.3.0.1           dst-ip
```

③ → `[root@controller01 ~]# ovn-nbctl lr-nat-list 1a5f0ed9-2425-4f48-ab81-a3b22465ee72`

```
TYPE             GATEWAY_PORT      EXTERNAL_IP       EXTERNAL_PORT     LOGICAL_IP        EXTERNAL_MAC      LOGICAL_PORT
dnat_and_snat                                        3.3.0.77          10.0.3.55         fa:16:3e:04:16:e5 cd7085a4-95bb-424e-bd63-c03c1c214021
[root@controller01 ~]#
```

`[root@controller01 ~]# ovn-nbctl lr-list | grep 68cc65a0-5b3b-4f63-aa6b-25db43d475d0`

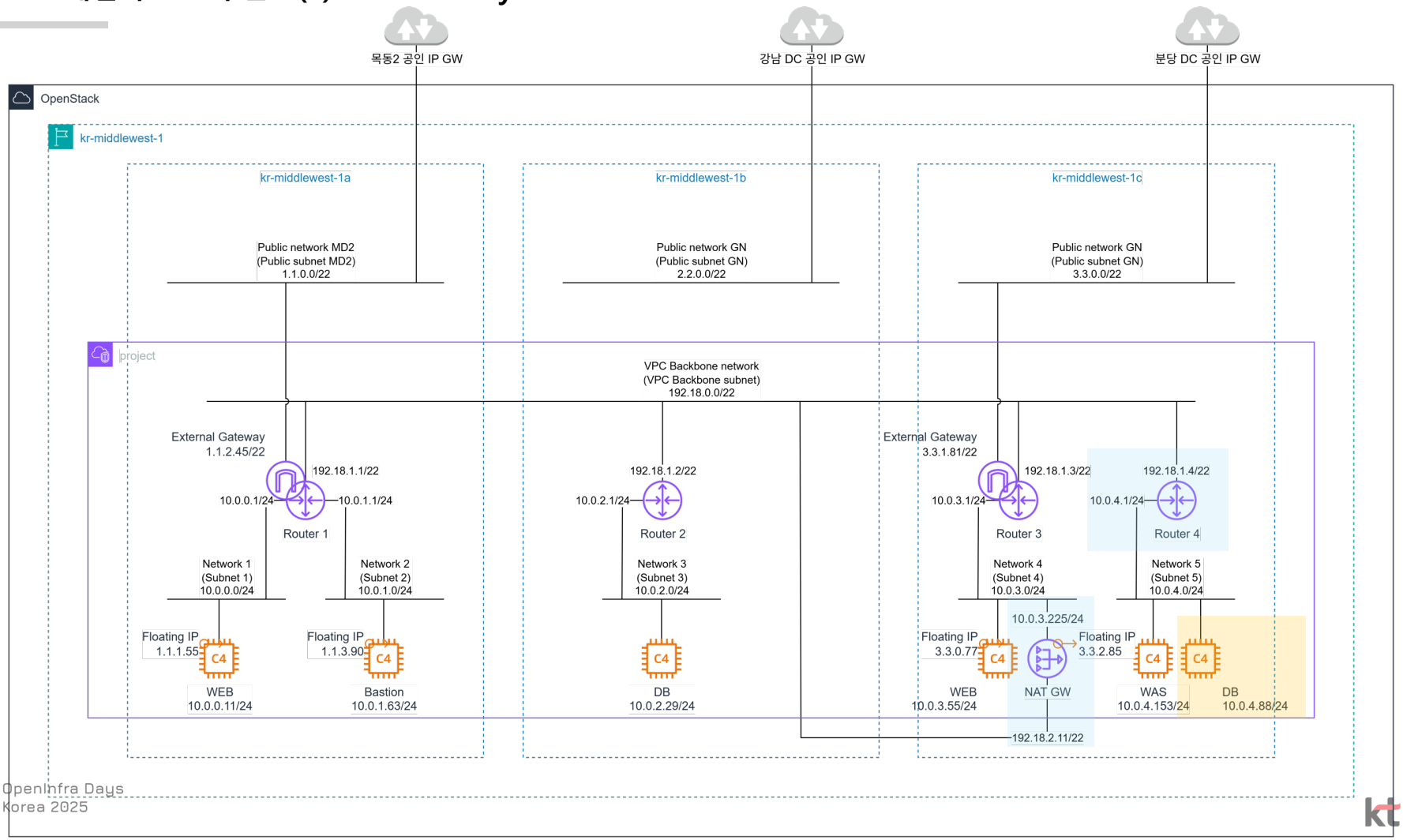
```
180cb399-430f-4633-87cc-9b6a3e0dc4cb (neutron-68cc65a0-5b3b-4f63-aa6b-25db43d475d0)
```

`[root@controller01 ~]# ovn-nbctl lr-route-list 180cb399-430f-4633-87cc-9b6a3e0dc4cb`

```
IPv4 Routes
Route Table <main>:
               10.0.3.0/24       192.18.1.3        dst-ip
```

`[root@controller01 ~]# ovn-nbctl lr-nat-list 180cb399-430f-4633-87cc-9b6a3e0dc4cb`

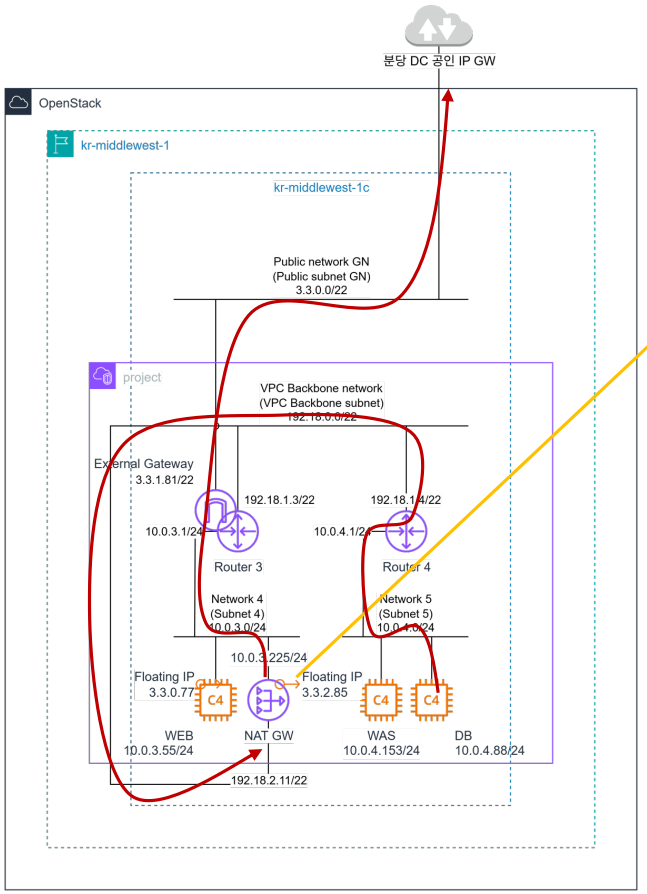
OpenStack 레벨의 VPC 구현 - (8) NAT Gateway

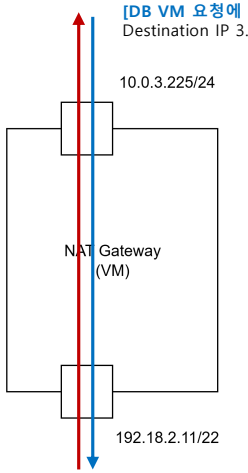


OpenStack 레벨의 VPC 구현 - (8) NAT Gateway [iptables를 이용한 NAT Gateway 구현]

[DB VM 10.0.4.88에서 인터넷으로 나가는 패킷 흐름]
DB VM -> Router 4 -> NAT GW (SNAT, 패킷의 Source IP가 10.0.3.225로 변환) -> Router 3(Source IP가 3.3.2.85로 변환)

[DB VM 요청에 응답하는 패킷 수신 시 흐름]
Destination IP 3.3.2.85 패킷을 Router 3이 수신(Destination IP 10.0.3.225로 변환) -> NAT GW(Destination IP 10.0.4.88로 변환) -> Router 4 -> DB VM



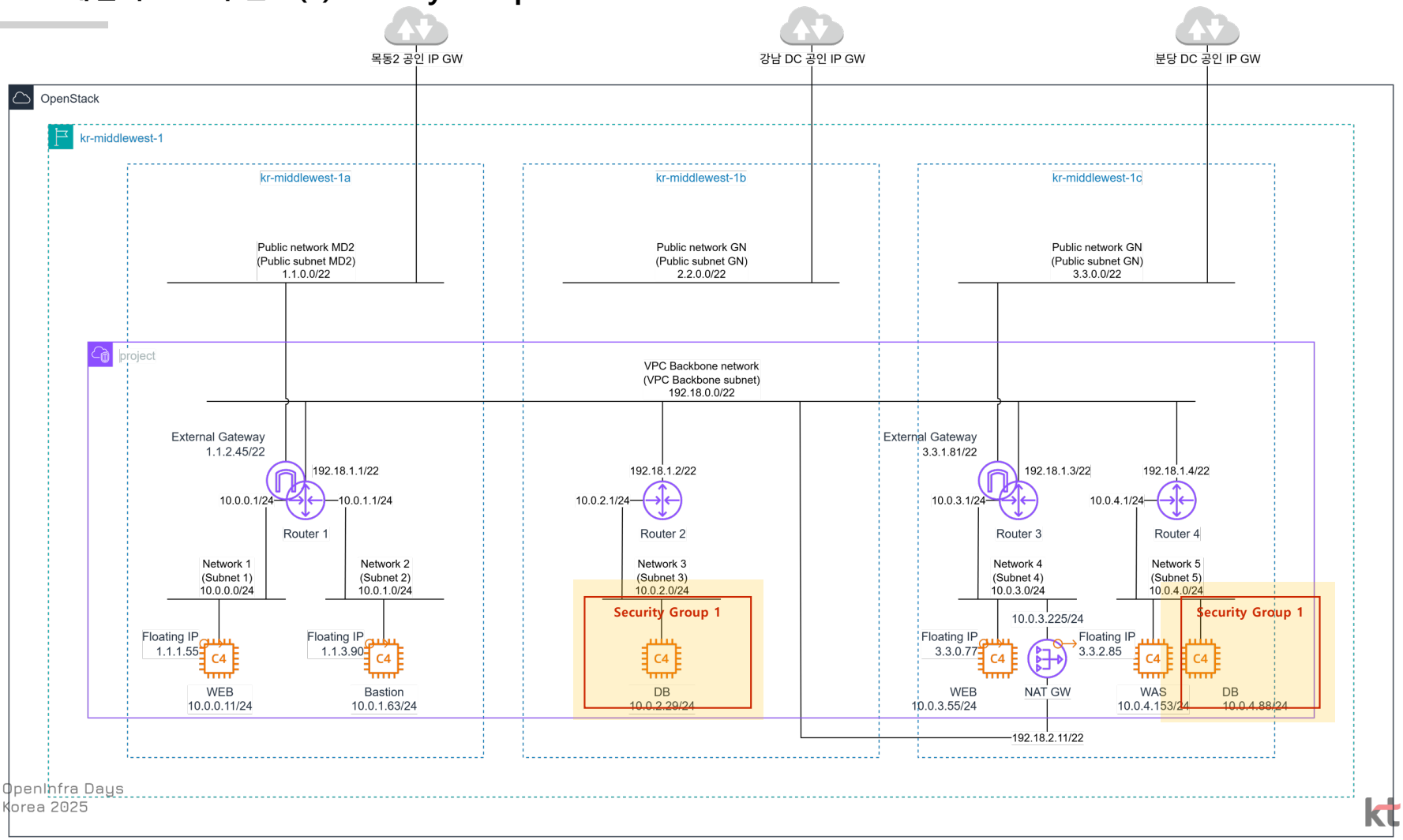


ip route	
0.0.0.0/0	10.0.3.1
10.0.4.0/24	192.18.1.4

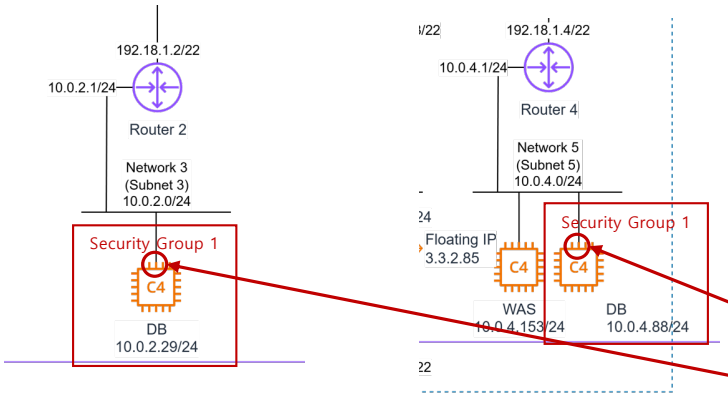
```
[root@controller01 ~]# ovn-nbctl lr-nat-list 1a5f0ed9-2425-4f48-ab81-a3b22465ee72
TYPE             GATEWAY_PORT      EXTERNAL_IP       EXTERNAL_PORT     LOGICAL_IP        EXTERNAL_MAC      LOGICAL_PORT
dnat_and_snat                                        3.3.0.77          10.0.3.55         fa:16:3e:04:16:e5 cd7085a4-95bb-424e-bd63-c03c1c214021
dnat_and_snat                                        3.3.2.85          10.0.3.225        fa:16:3e:5d:da:81 4bad7f26-ab30-4386-aae1-c91943d5bb9e
```

```
root@nat-gateway:~# iptables -t nat -L -vv
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
 33M 1604M  MASQUERADE  all  --  any    enp3s0  anywhere             anywhere
    0     0  MASQUERADE  all  --  any    enp3s0  anywhere             anywhere
```

OpenStack 레벨의 VPC 구현 - (9) Security Group



OpenStack 레벨의 VPC 구현 - (9) Security Group [OpenStack Security Group(OVN Port-group ACL)]



Security Group 1 - 8d20d3b5_3866_49fd_a823_1d2db48ff02f

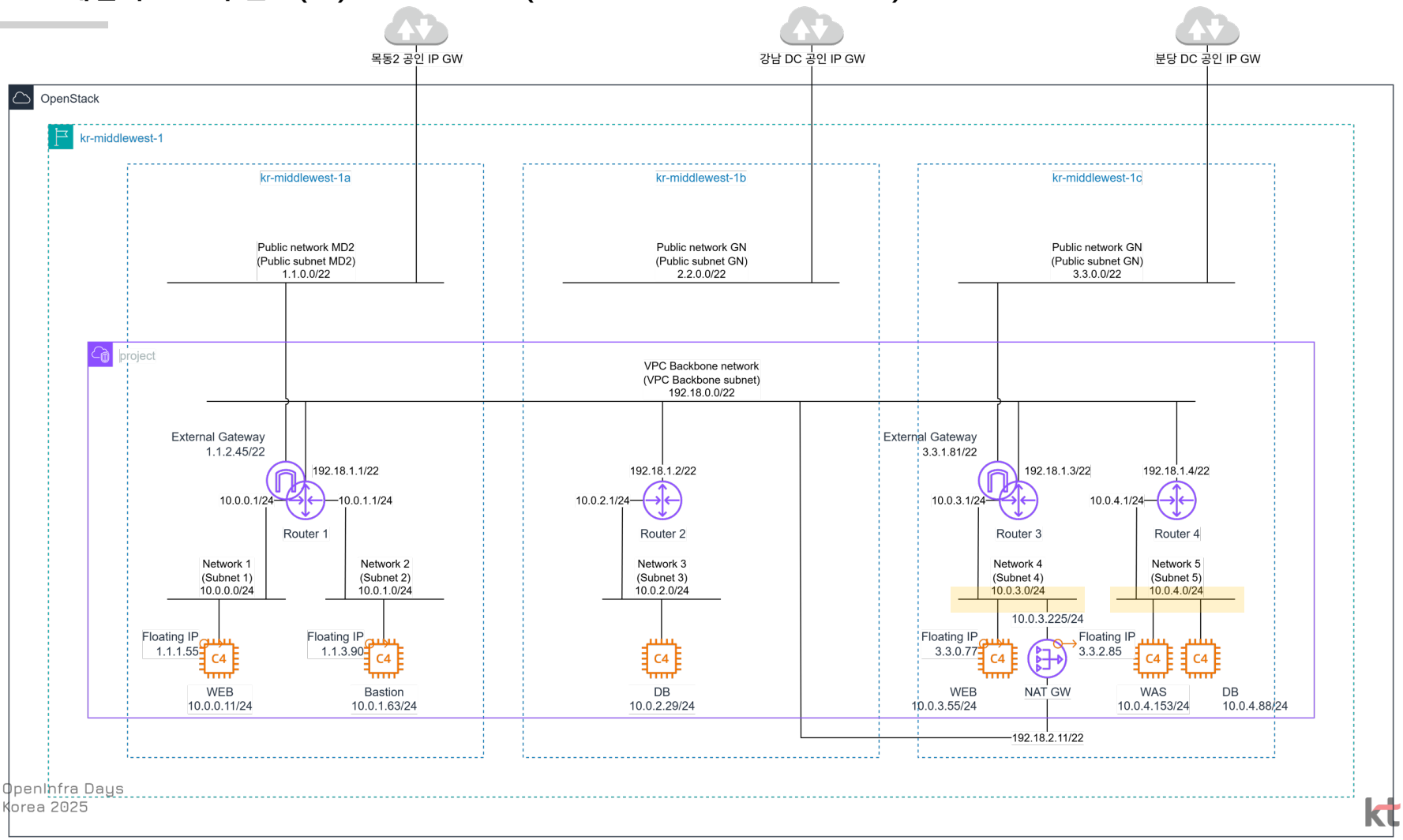
① →

② →

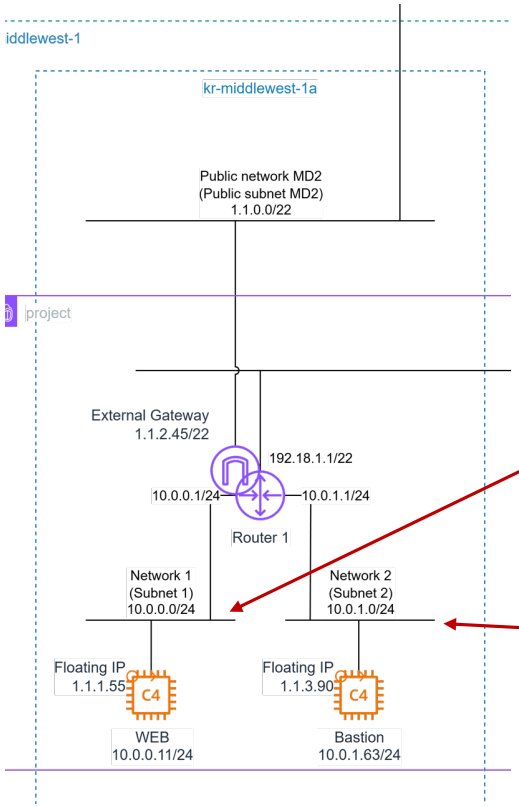
```
_uuid               : f8341320-e69c-46f2-a3cf-167e102fc6db
acls                : [27069796-edf0-440d-bd35-cbc19daa211c, 271f45b6-af7d-4047-9a6a-b4bb5f56a3f5, 2b507b3a-c72c-478e-a620-78b109bdf42a, 38292ff5-
cb4b-4bec-9597-229c3b5e073f, 50fd1eda-2fba-45a8-bc1f-524406756f70, 792bd65c-faa4-4f9d-89fd-725c6d3b43cf, 7cd84b1a-298b-4377-bba5-781d63715f16]
external_ids        : {"neutron:security_group_id"="8d20d3b5-3866-49fd-a823-1d2db48ff02f"}
name                : pg_8d20d3b5_3866_49fd_a823_1d2db48ff02f
ports               : [0142f23d-f375-4eff-86b9-9f2c3060cb22, 02a93c9d-684e-41fe-8d64-7aa05f40fd32]
```

```
(ovn-nb-db)[root@controller01 /]# ovn-nbctl acl-list f8341320-e69c-46f2-a3cf-167e102fc6db
from-lport  1002 (inport == @pg_8d20d3b5_3866_49fd_a823_1d2db48ff02f && ip4) allow-related
from-lport  1002 (inport == @pg_8d20d3b5_3866_49fd_a823_1d2db48ff02f && ip4 && ip4.dst == 0.0.0.0/0 && icmp4) allow-related
from-lport  1002 (inport == @pg_8d20d3b5_3866_49fd_a823_1d2db48ff02f && ip6) allow-related
  to-lport  1002 (outport == @pg_8d20d3b5_3866_49fd_a823_1d2db48ff02f && ip4 && ip4.src == 0.0.0.0/0) allow-related
  to-lport  1002 (outport == @pg_8d20d3b5_3866_49fd_a823_1d2db48ff02f && ip4 && ip4.src == 0.0.0.0/0 && icmp4) allow-related
  to-lport  1002 (outport == @pg_8d20d3b5_3866_49fd_a823_1d2db48ff02f && ip4 && ip4.src == 0.0.0.0/0 && tcp) allow-related
  to-lport  1002 (outport == @pg_8d20d3b5_3866_49fd_a823_1d2db48ff02f && ip4 && ip4.src == 10.0.0.0/16 && tcp && tcp.dst == 22) allow-related
```

OpenStack 레벨의 VPC 구현 - (10) Network ACL(Network Access Control List)



OpenStack 레벨의 VPC 구현 - (10) Network ACL(Network Access Control List) [OVN Logical Switch ACL]



Network ACL 1

Network 1(Subnet 1)

```
(ovn-nb-db)[root@controller01 /]# ovn-nbctl ls-list
73dd3625-dfce-4780-917c-8f783e4c860b (neutron-f8fae423-a562-4b6a-abac-2dfaede55ed1)
```

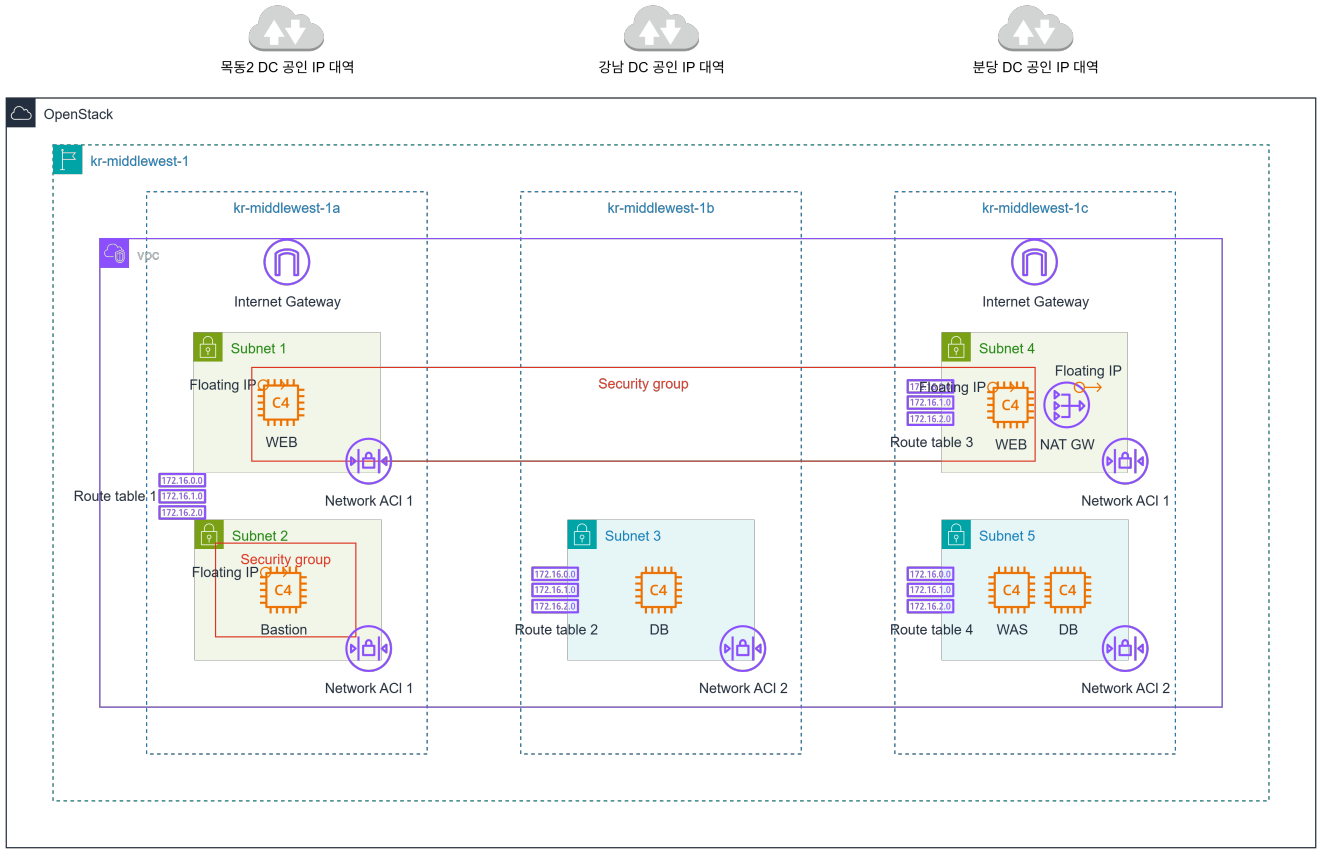
```
(ovn-nb-db)[root@controller01 /]# ovn-nbctl acl-list 73dd3625-dfce-4780-917c-8f783e4c860b
from-lport  2003 (ip4 && ip4.dst == 0.0.0.0/0) allow-stateless
from-lport  2001 (ip4 && ip4.dst == 0.0.0.0/0) drop
  to-lport  2003 (ip4 && ip4.dst == 0.0.0.0/0) allow-stateless
  to-lport  2001 (ip4 && ip4.dst == 0.0.0.0/0) drop
```

Network 2(Subnet 2)

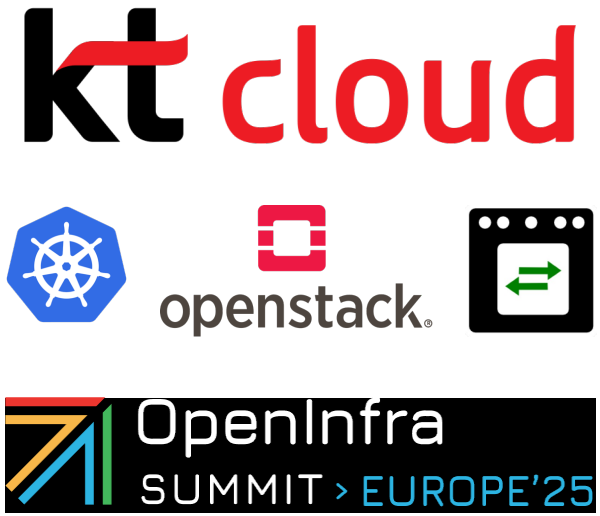
```
(ovn-nb-db)[root@controller01 /]# ovn-nbctl ls-list
73dd3625-dfce-4780-917c-8f783e4c860b (neutron-f8fae423-a562-4b6a-abac-2dfaede55ed1)
```

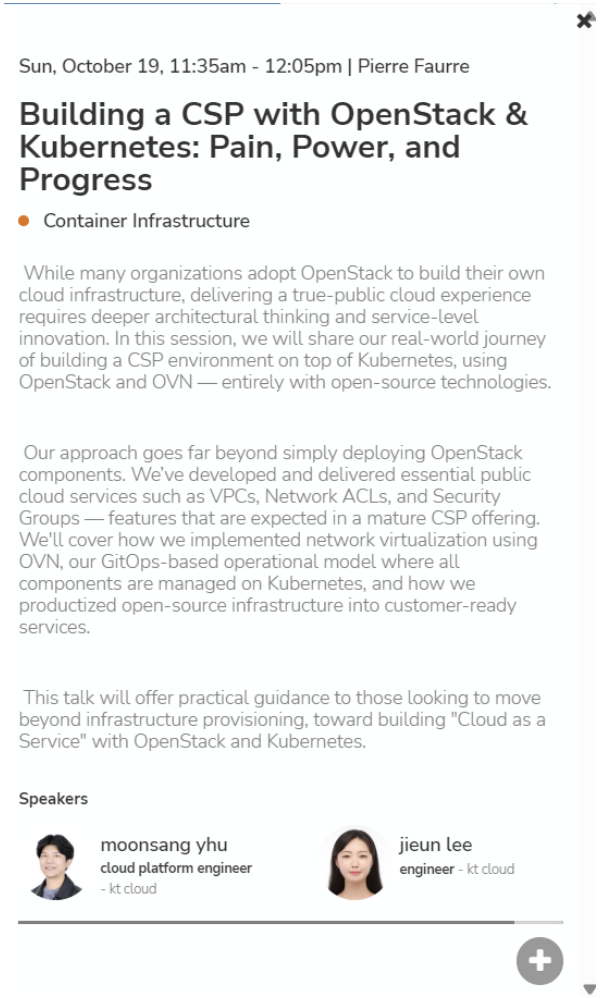
```
(ovn-nb-db)[root@controller01 /]# ovn-nbctl acl-list 73dd3625-dfce-4780-917c-8f783e4c860b
from-lport  2003 (ip4 && ip4.dst == 0.0.0.0/0) allow-stateless
from-lport  2001 (ip4 && ip4.dst == 0.0.0.0/0) drop
  to-lport  2003 (ip4 && ip4.dst == 0.0.0.0/0) allow-stateless
  to-lport  2001 (ip4 && ip4.dst == 0.0.0.0/0) drop
```

마무리: OpenStack Native VPC 그 이후



Ad: OpenInfra Summit Europe 2025 of kt cloud





감사합니다

Q&A