

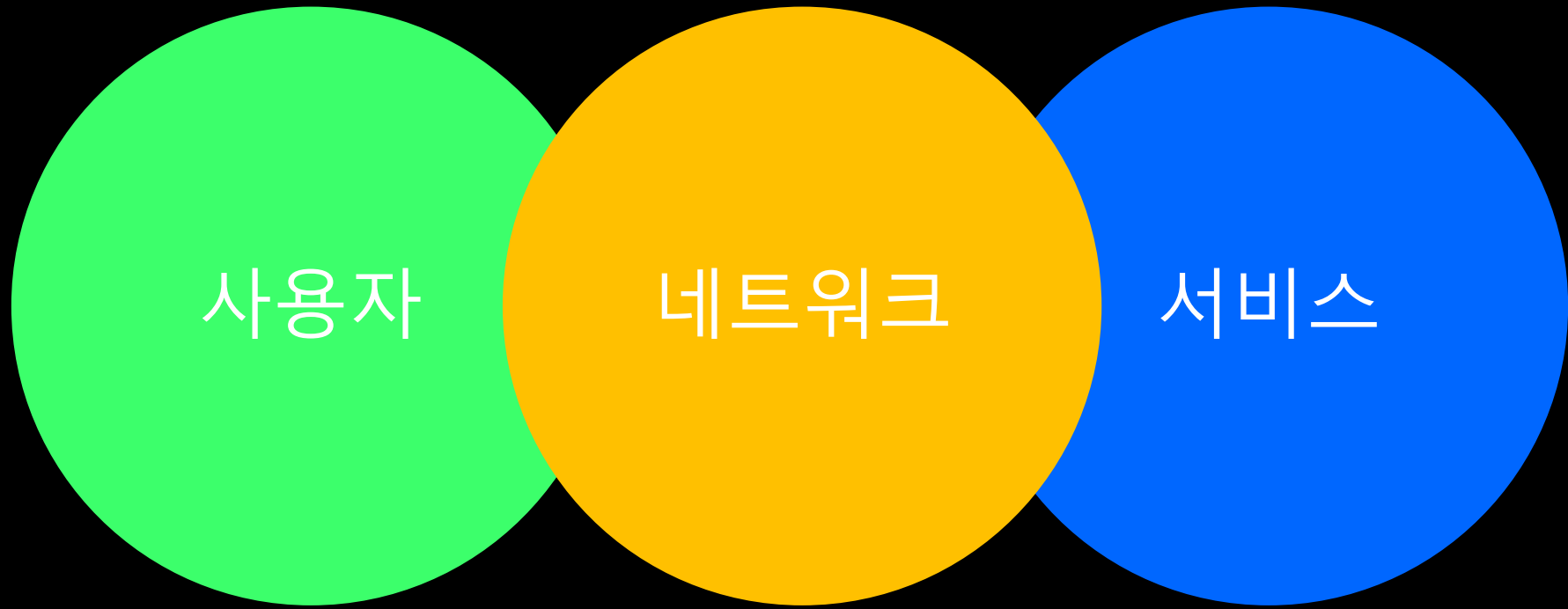
# 클라우드 Network 200% 활용하기

발표자 Master Seo  
topasvga@naver.com

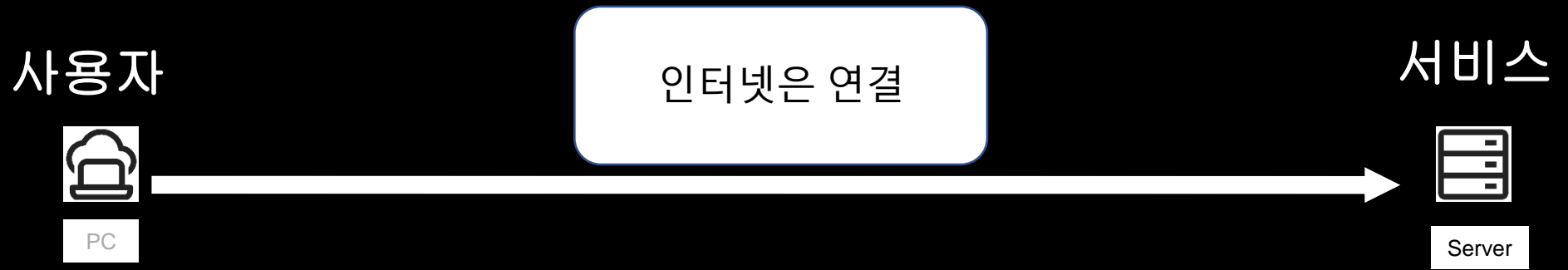
1. 네이버 클라우드 Network 알아보기
2. Global DNS 사용해보자
3. 쿠버네티스 사용해보자
4. 테라폼 사용해보자

# 1. 네이버 클라우드 Network 알아보기

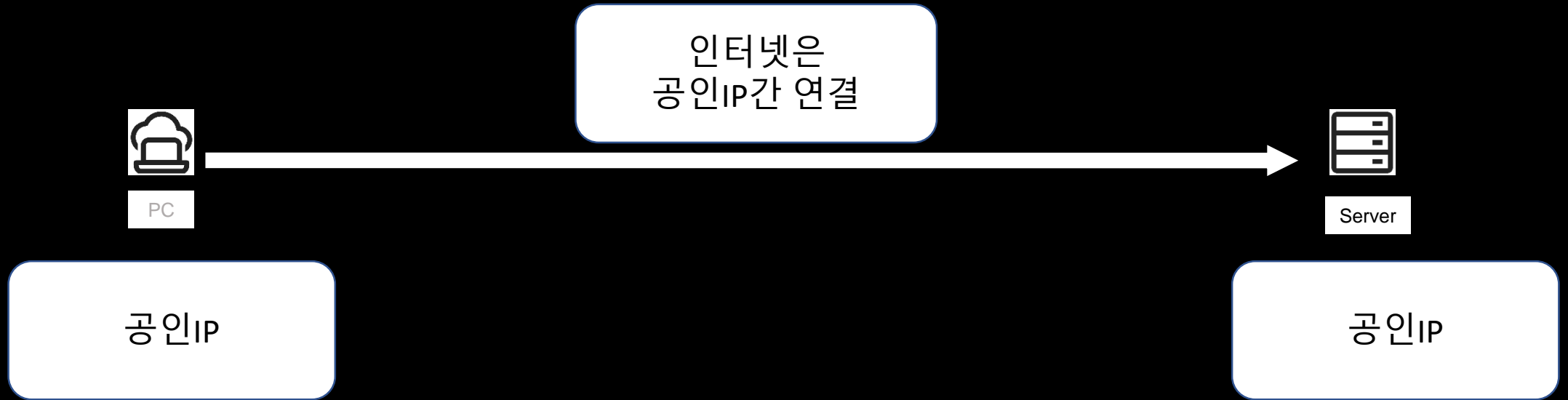
Network = 연결1



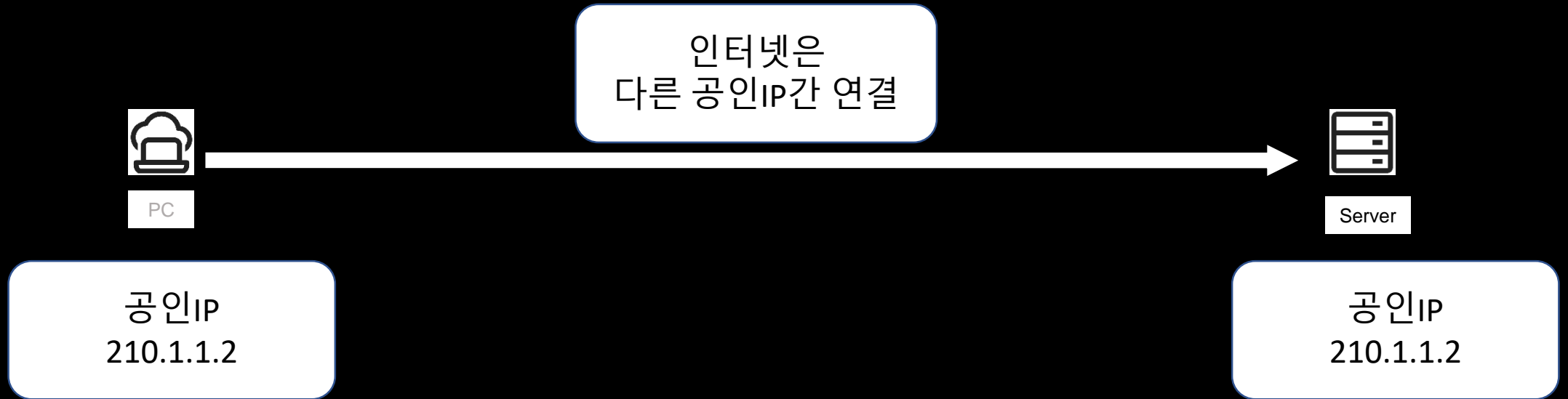
# Network = 연결2



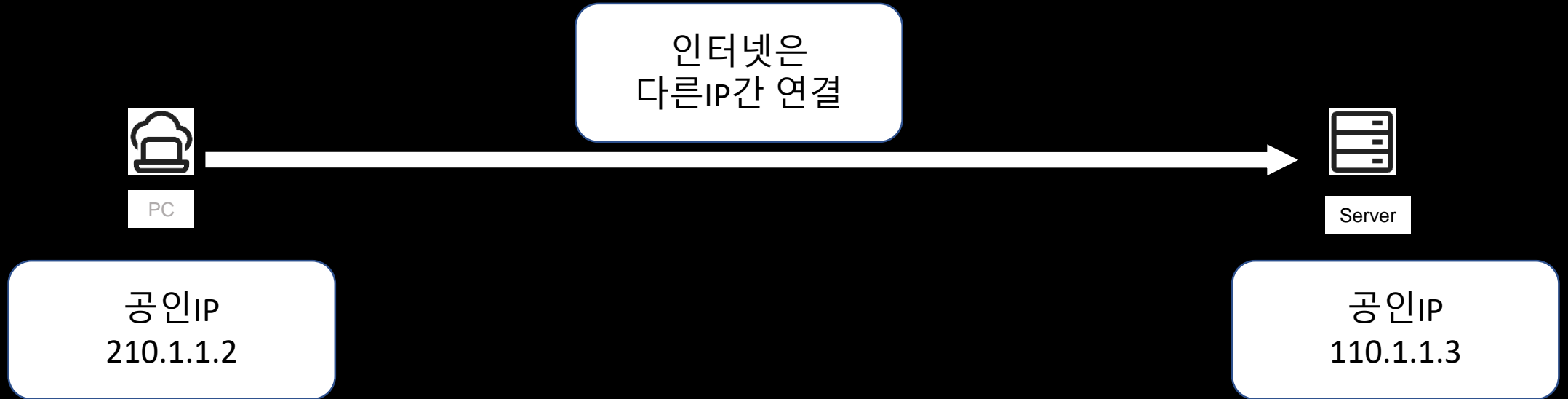
# Network = 연결3



# Network = 연결4

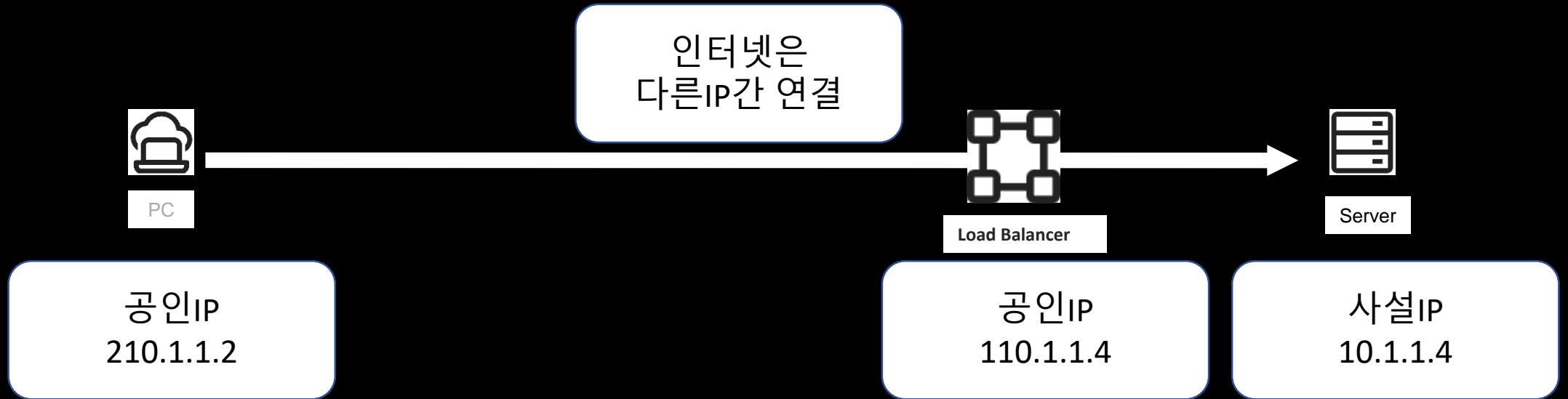


# Network = 연결5

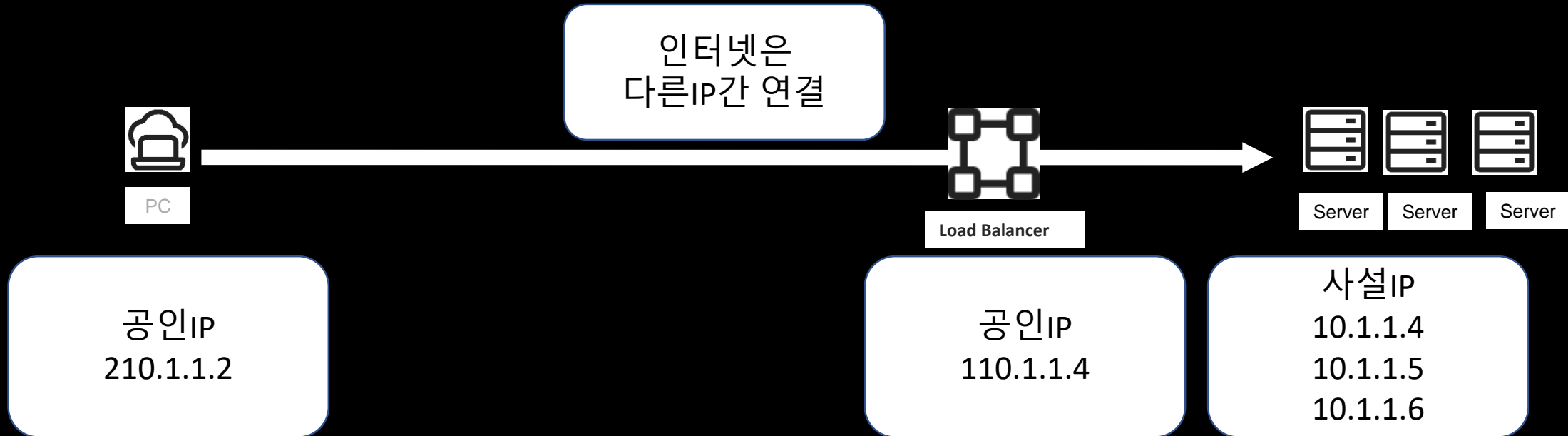




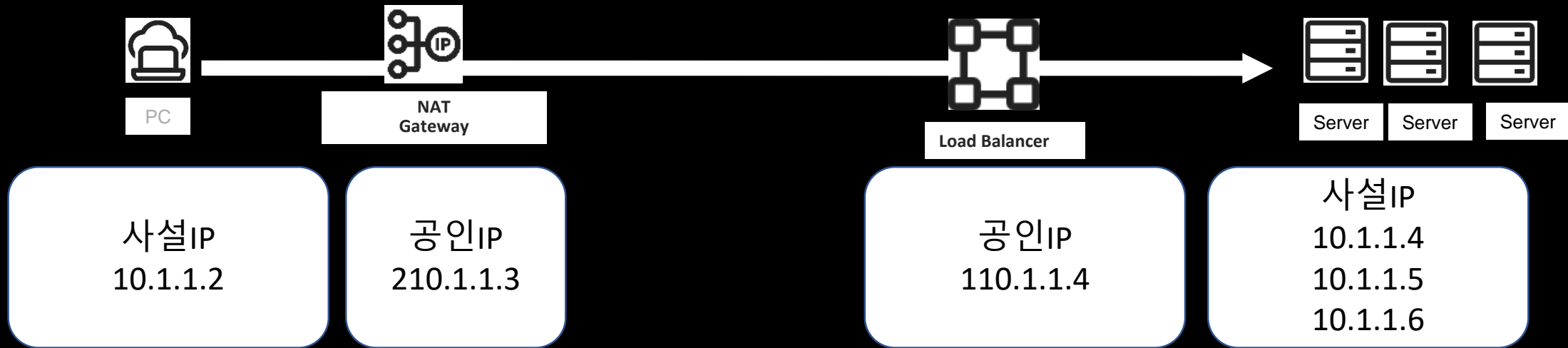
# Network = 연결6



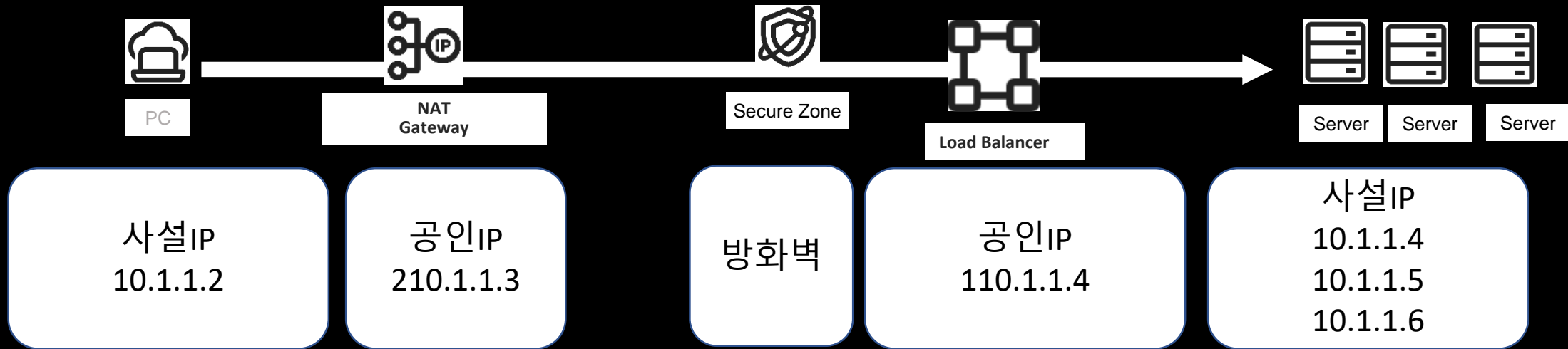
# Network = 연결7



# Network = 연결8



# Network = 연결9



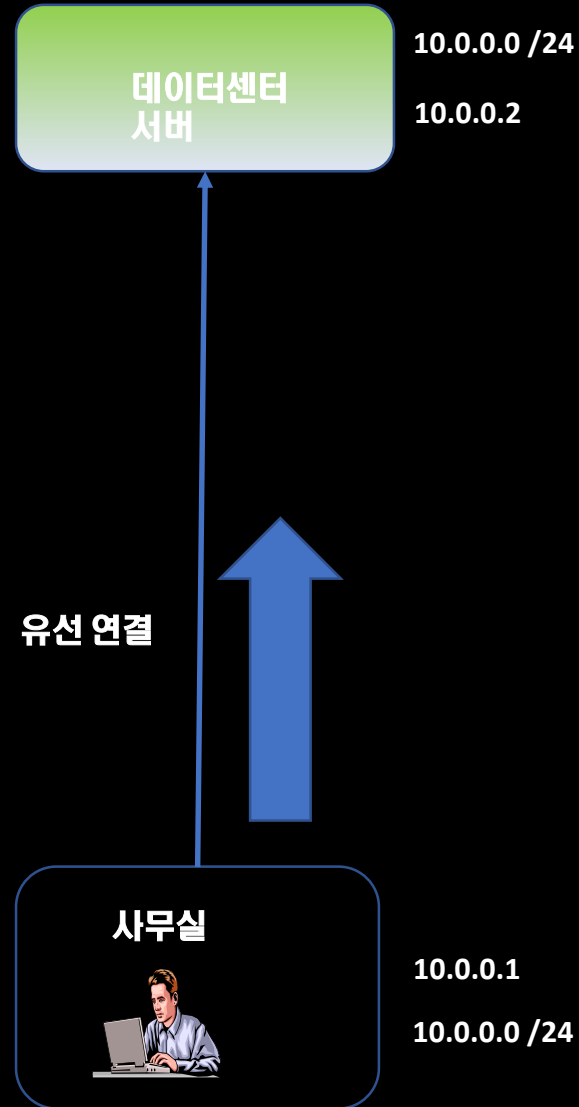
# 사무실에서 데이터 센터까지 네트워크 연결10

## 3. Private Address Space

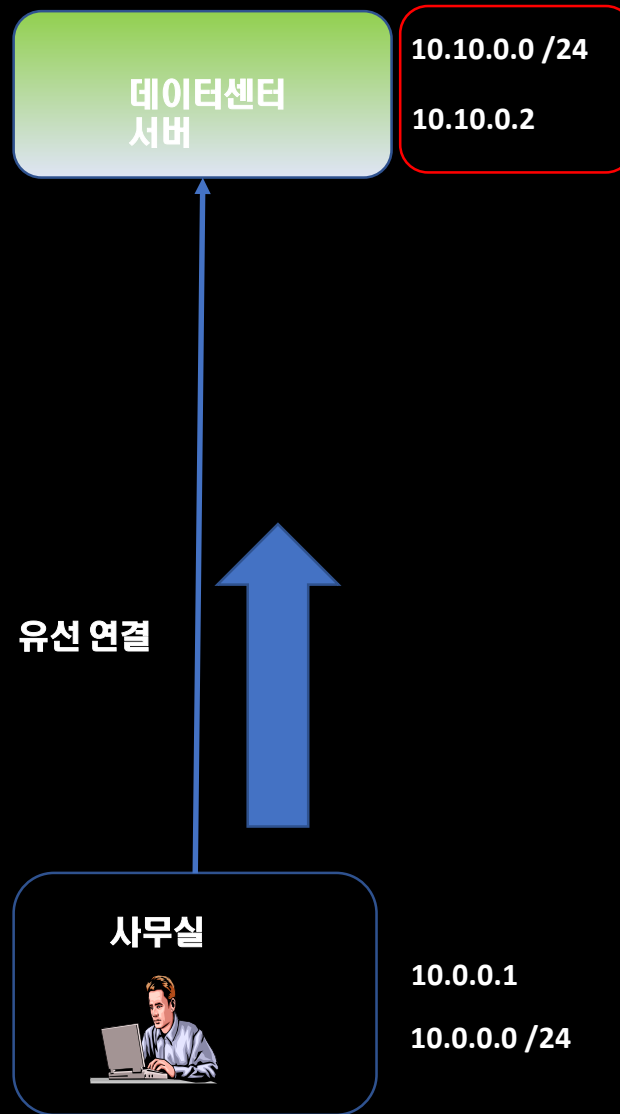
The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private internets:

10.0.0.0	-	10.255.255.255	(10/8 prefix)
172.16.0.0	-	172.31.255.255	(172.16/12 prefix)
192.168.0.0	-	192.168.255.255	(192.168/16 prefix)

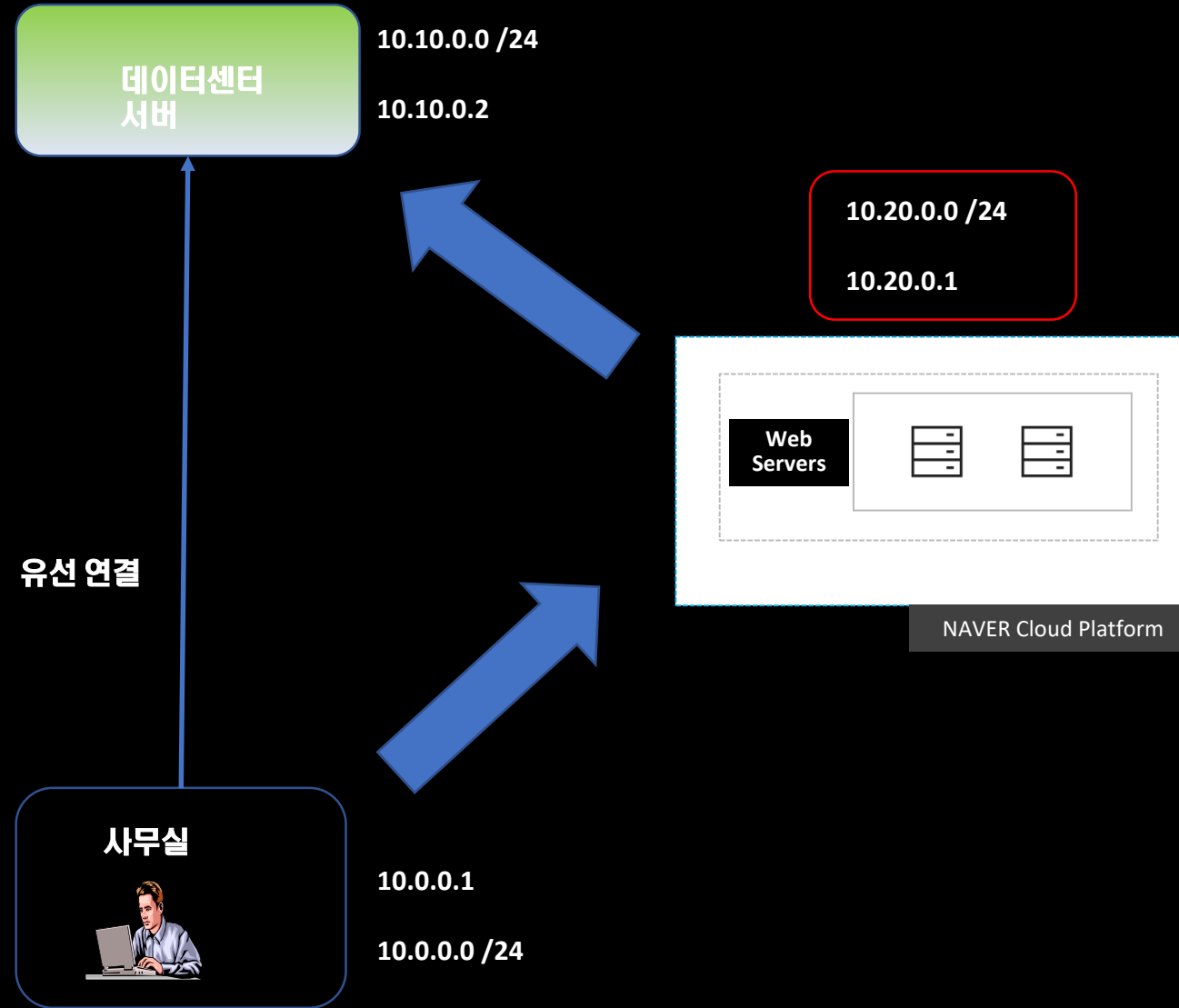
# 사무실에서 데이터 센터까지 네트워크 연결 11 - 10.0.0.0/24



# 사무실에서 데이터 센터까지 네트워크 연결 12 - 10.10.0.0/24

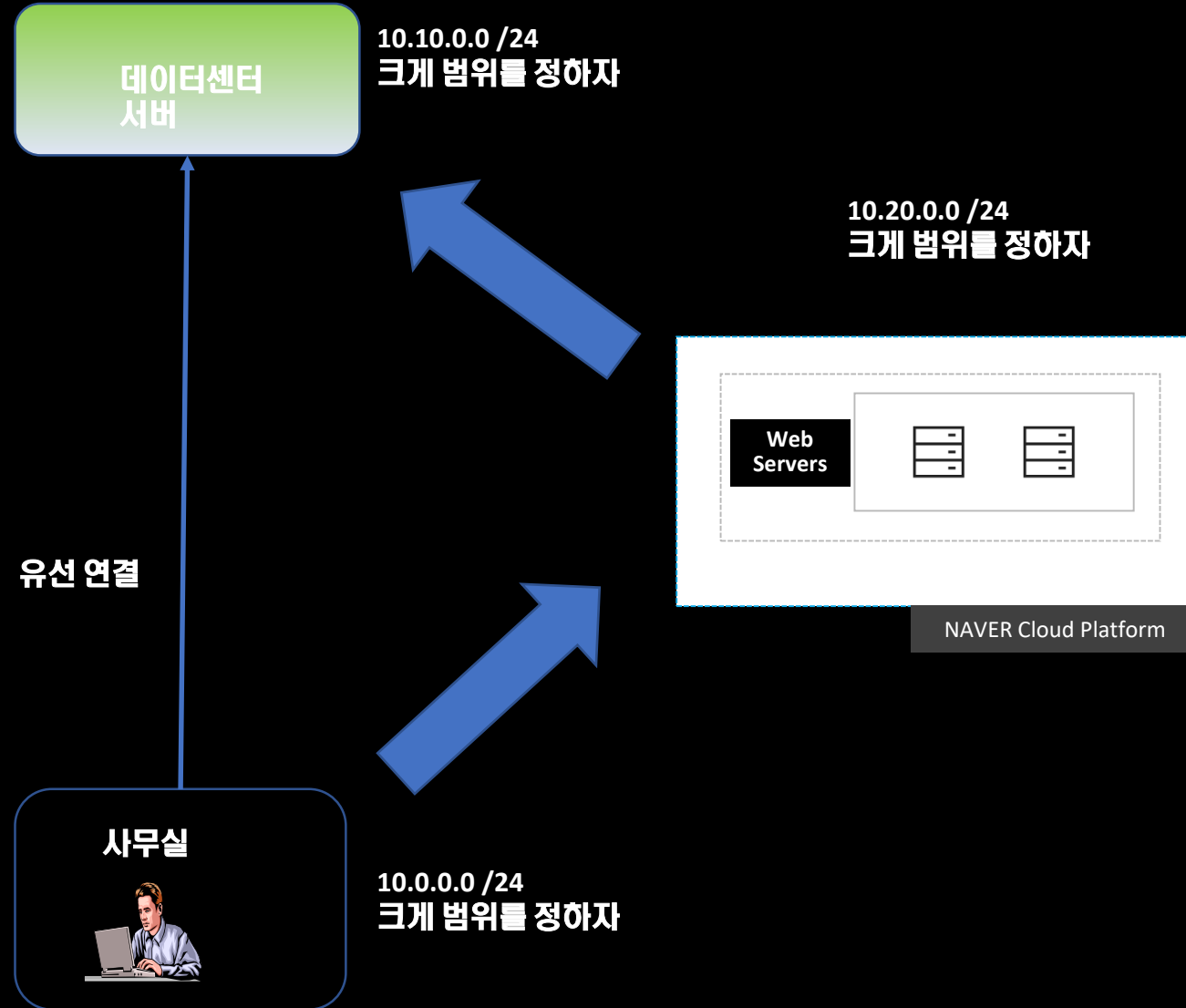


# 사무실에서 데이터 센터까지 네트워크 연결 13 - 10.20.0.0/24

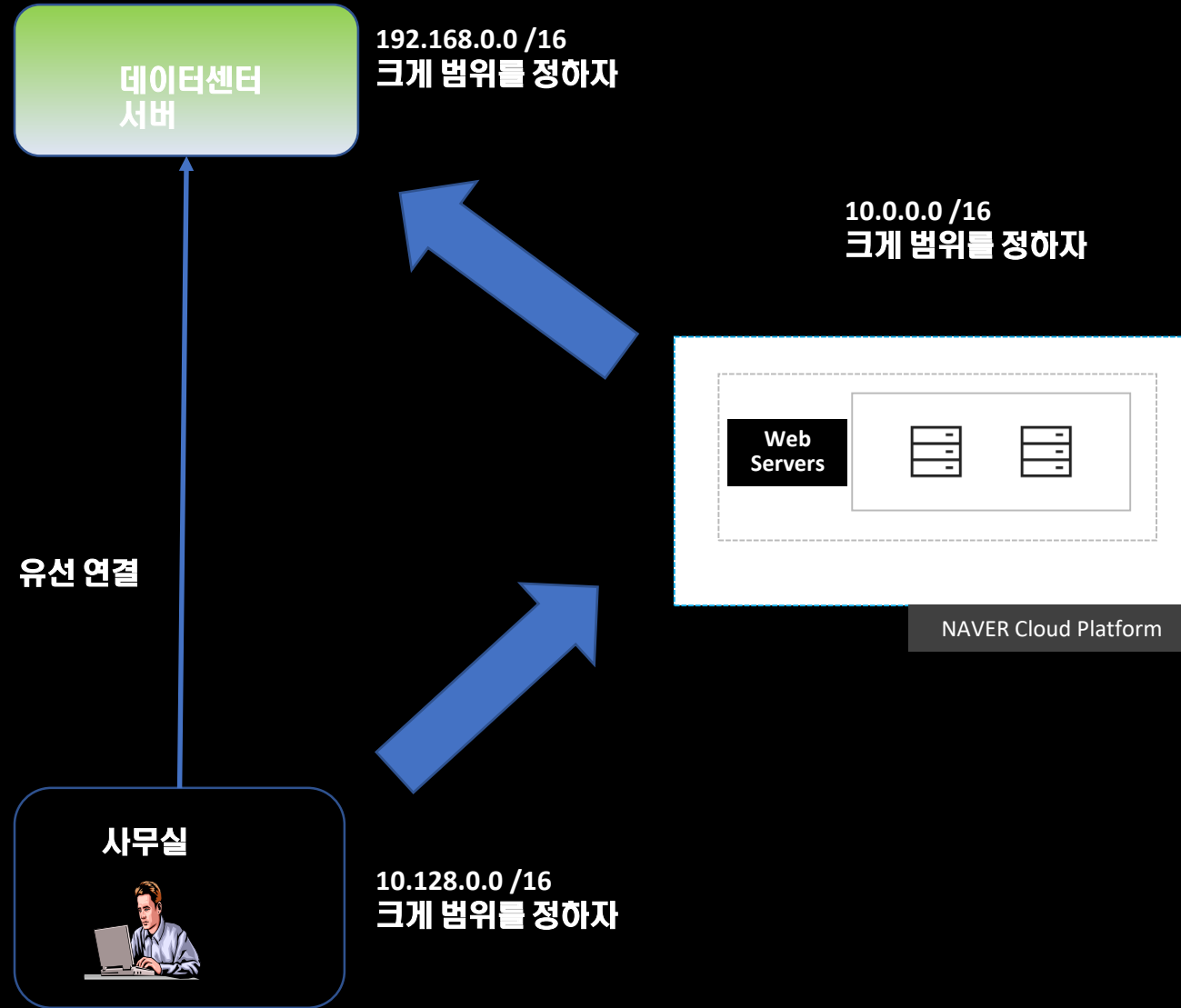




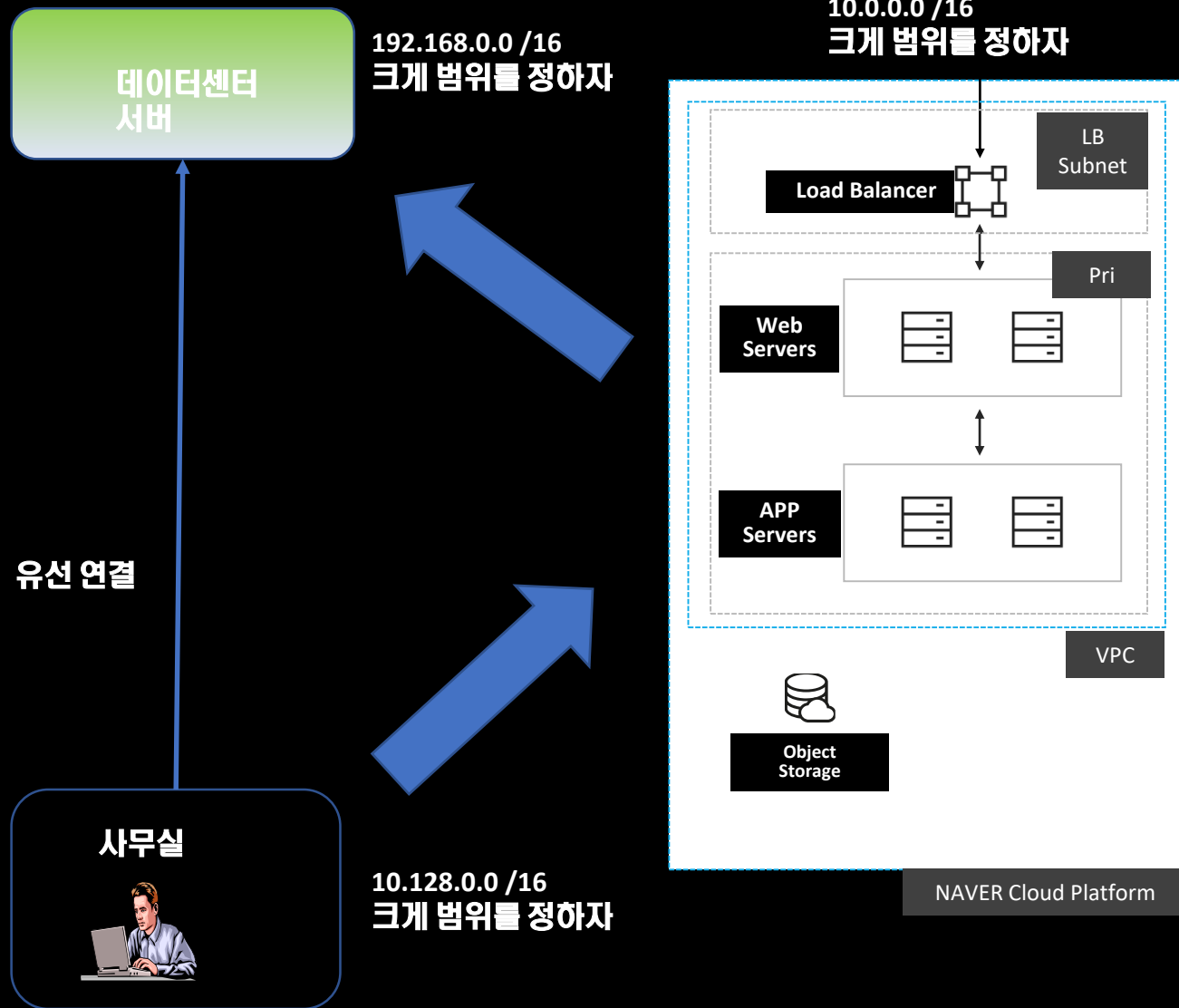
# 사무실에서 데이터 센터까지 네트워크 연결14



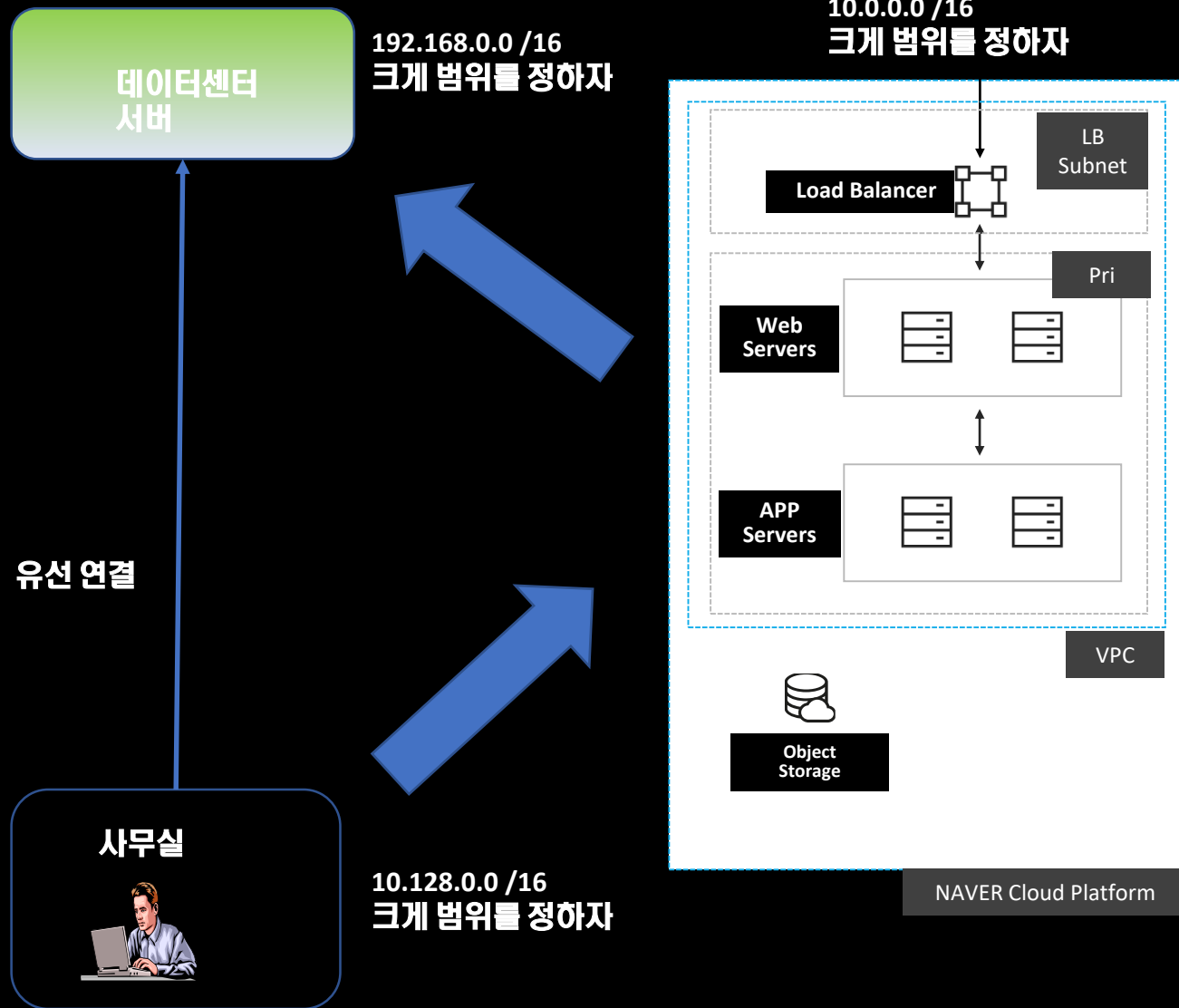
# 사무실에서 데이터 센터까지 네트워크 연결15



# 사무실에서 데이터 센터까지 네트워크 연결 16-법적 요건-서브네팅



# 사무실에서 데이터 센터까지 네트워크 연결 17 - 예제



# 네이버 클라우드 네트워크18 – VPC

VPC = 네트워크

네이버 클라우드 VPC 는 리전당 최대 3개의 VPC 생성가능

VPC는 1개가 기본으로 생성가능하며, 3개 사용하려면 고객센터에 추가 요청 해야 한다.

가능한 IP 주소 범위 - 사설 IP는 모두 사용가능하다.

10.0.0.0/8

172.16.0.0/12

192.168.0.0/16

그러나?

할당은 /16 범위만 가능하다.

10.0.0.0 /8 할당은? 안된다.

그럼?

10.0.0.0 /16 가능

10.128.0.0 /16 가능

최소는?

최소 /28이다. = 16개 IP

# 네이버 클라우드 네트워크19 - 서브네팅

10.0.0.0/12 안됨 = 최대가 16이니까 12는 안된다.  
192.168.0.0/12 도 안된다 = 최대가 16이니까 12는 안된다.

10.0.0.0/16 된다.  
10.0.0.0/24는 된다.  
172.168.0.0/16 = 된다  
192.168.0.0/16 = 된다.

5.0.0.0/16 - 안된다. 범위가 아니다.  
Pv4 CIDR block은 private 대역(10.0.0.0/8,172.16.0.0/12,192.168.0.0/16)  
내에서 /16~/28 범위여야 합니다.

# 네이버 클라우드 네트워크20 - 서브네팅

10.0.0.0 /20 = C-Class16개

10.0.0.0 /24

10.0.1.0 /24

10.0.2.0 /24

10.0.3.0 /24

10.0.4.0 /24

10.0.5.0 /24

10.0.6.0 /24

10.0.7.0 /24

10.0.8.0 /24

10.0.9.0 /24

10.0.10.0 /24

10.0.11.0 /24

10.0.12.0 /24

10.0.13.0 /24

10.0.14.0 /24

10.0.15.0 /24

10.0.16.0 /20 = C-Class16개

10.0.16.0 /24

10.0.17.0 /24

10.0.18.0 /24

10.0.19.0 /24

10.0.20.0 /24

10.0.21.0 /24

10.0.22.0 /24

10.0.23.0 /24

10.0.24.0 /24

10.0.25.0 /24

10.0.26.0 /24

10.0.27.0 /24

10.0.28.0 /24

10.0.29.0 /24

10.0.30.0 /24

10.0.31.0 /24

# 네이버 클라우드 네트워크21 - LB서브넷, NAT 서브넷

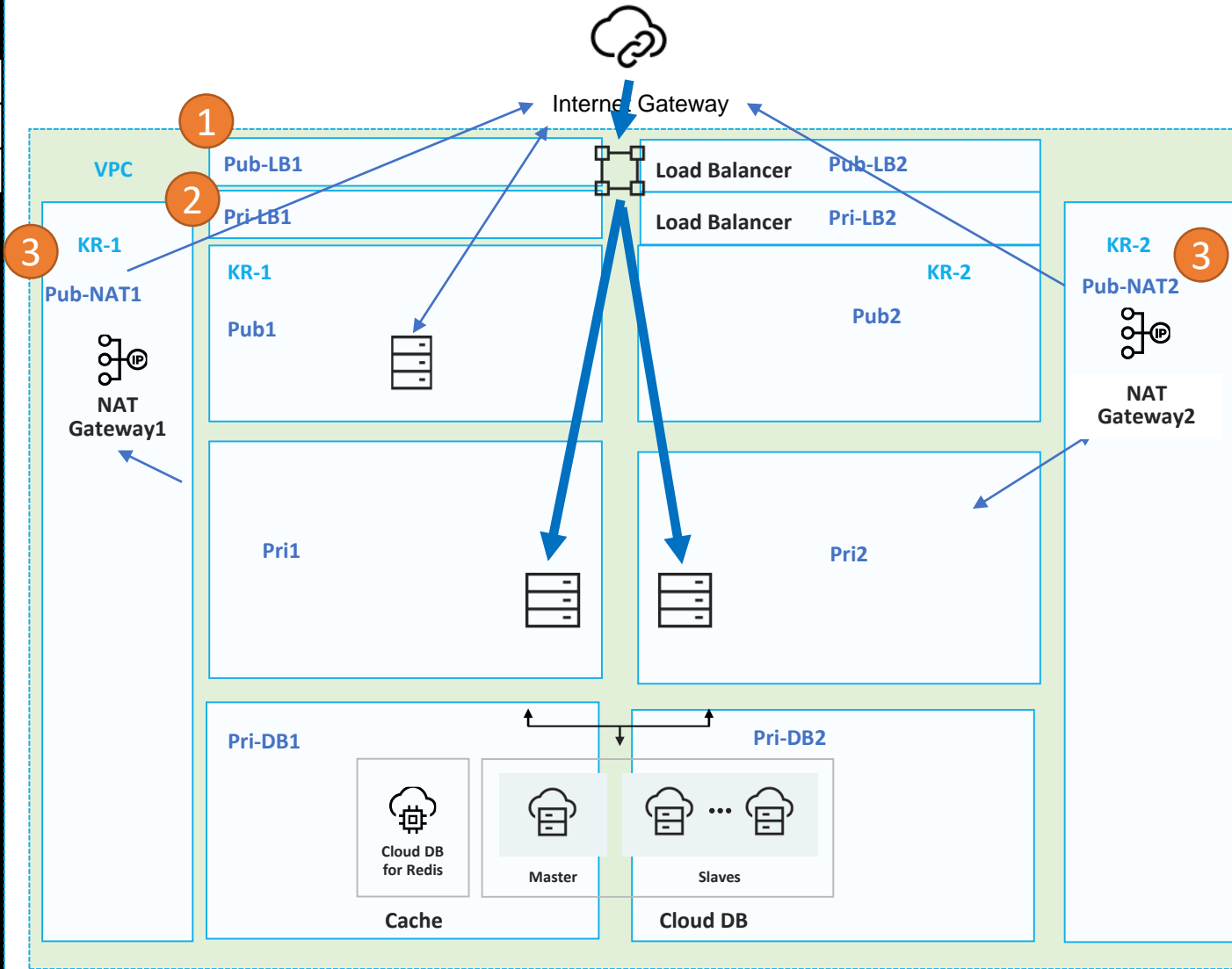
네이버 클라우드는

LB 서브넷을 별도로 구축한다.  
NAT 서브넷을 별도로 구축한다.



# 네이버 클라우드 네트워크22 (현재 네트워크)

리전



목적지	Target유형
10.0.0.0/20	LOCAL
0.0.0.0/0	IGW

목적지	Target유형
10.0.0.0/20	LOCAL
0.0.0.0/0	NATGW1

목적지	Target유형
10.0.0.0/20	LOCAL

목적지	Target유형
10.0.0.0/20	LOCAL
0.0.0.0/0	NATGW2

## 네이버 클라우드 네트워크23 (현재 네트워크)

VPC	할당 VPC	C	서브넷 IP	서브넷 네임	Zone	Public/Pri
bgame-naver-dev	10.0.0.0/20	1	10.0.0.0/24	Pub1	KR-1	Pub
		2	10.0.1.0/24	예비	-	-
		3-4	10.0.2.0/23	Pri1	KR-1	Pri
		5	10.0.4.0/24	Pub-Nat1	KR-1	Pub-Nat1
		6	10.0.5.0/24	Pub-LB1	KR-1	Pub-LB1
		7	10.0.6.0/24	Pri-DB1	KR-1	Pri-DB1
		8	10.0.7.0/24	Pri-LB1	KR-1	Pri-LB1
		9	10.0.8.0/24	Pub2	KR-2	Pub
		10	10.0.9.0/24	예비	-	-
		11-12	10.0.10.0/23	Pri2	KR-2	Pri
		13	10.0.12.0/24	Pub-Nat2	KR-2	Pub-Nat2
		14	10.0.13.0/24	Pub-LB2	KR-2	Pub-LB2
		15	10.0.14.0/24	Pri-DB2	KR-2	Pri-DB2
		16	10.0.15.0/24	Pri-LB2	KR-2	Pri-LB2

# 네이버 클라우드 네트워크24 (구 네트워크)

리전

1

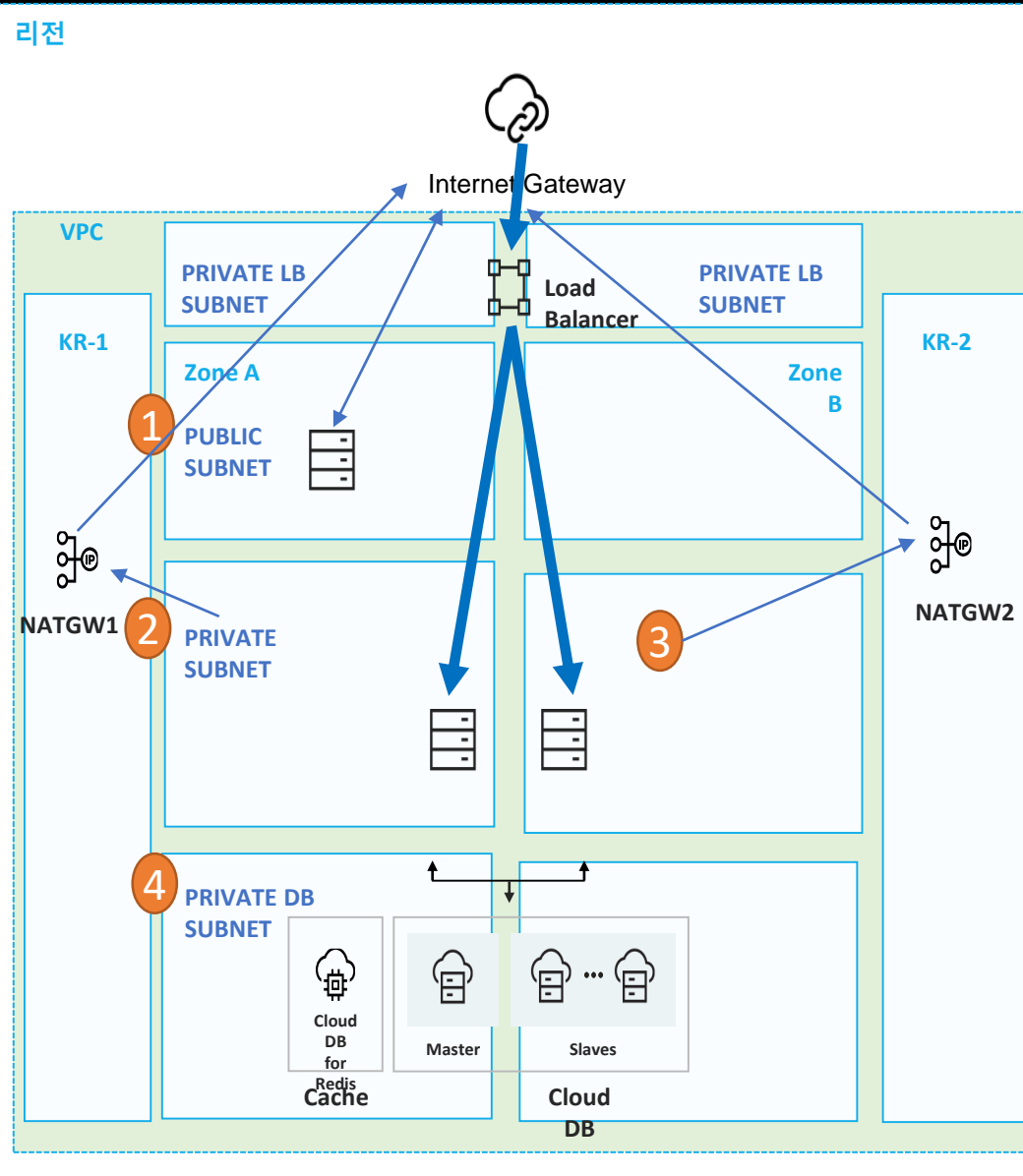
목적지	Target유형
10.0.0.0/20	LOCAL
0.0.0.0/0	IGW

2

목적지	Target유형
10.0.0.0/20	LOCAL
0.0.0.0/0	NATGW1

4

목적지	Target유형
10.0.0.0/20	LOCAL



3

목적지	Target유형
10.0.0.0/20	LOCAL
0.0.0.0/0	NATGW2

# 네이버 클라우드 네트워크25 - 서버 생성

Pub1

10.0.0.0/24

서버 생성 -로키 리눅스 - Pub1에 하나  
외부에서 접속하게 하려면 생성시 공인 IP 할당

Pri1

10.0.2.0 /23

서버 생성 -로키 리눅스 - Pri1에 하나  
외부에서 접속할 수 없는 환경  
외부에서 접속하려면 로드 밸런서가 필요하다. = 로드밸런서 전용 서브넷 생성 - 로드 밸런서 생성. (네이버)

# 네이버 클라우드 네트워크26 - 로드 밸런서 생성

agame-pub1-lb1 = 로드밸런서 생성

Agame-pub1-lb1-tg = 티켓 그룹 생성

pub1 서버에 웹서비스 올리기  
80보안 그룹 허용

pub1 웹서버로 접속해 보기

root

암호

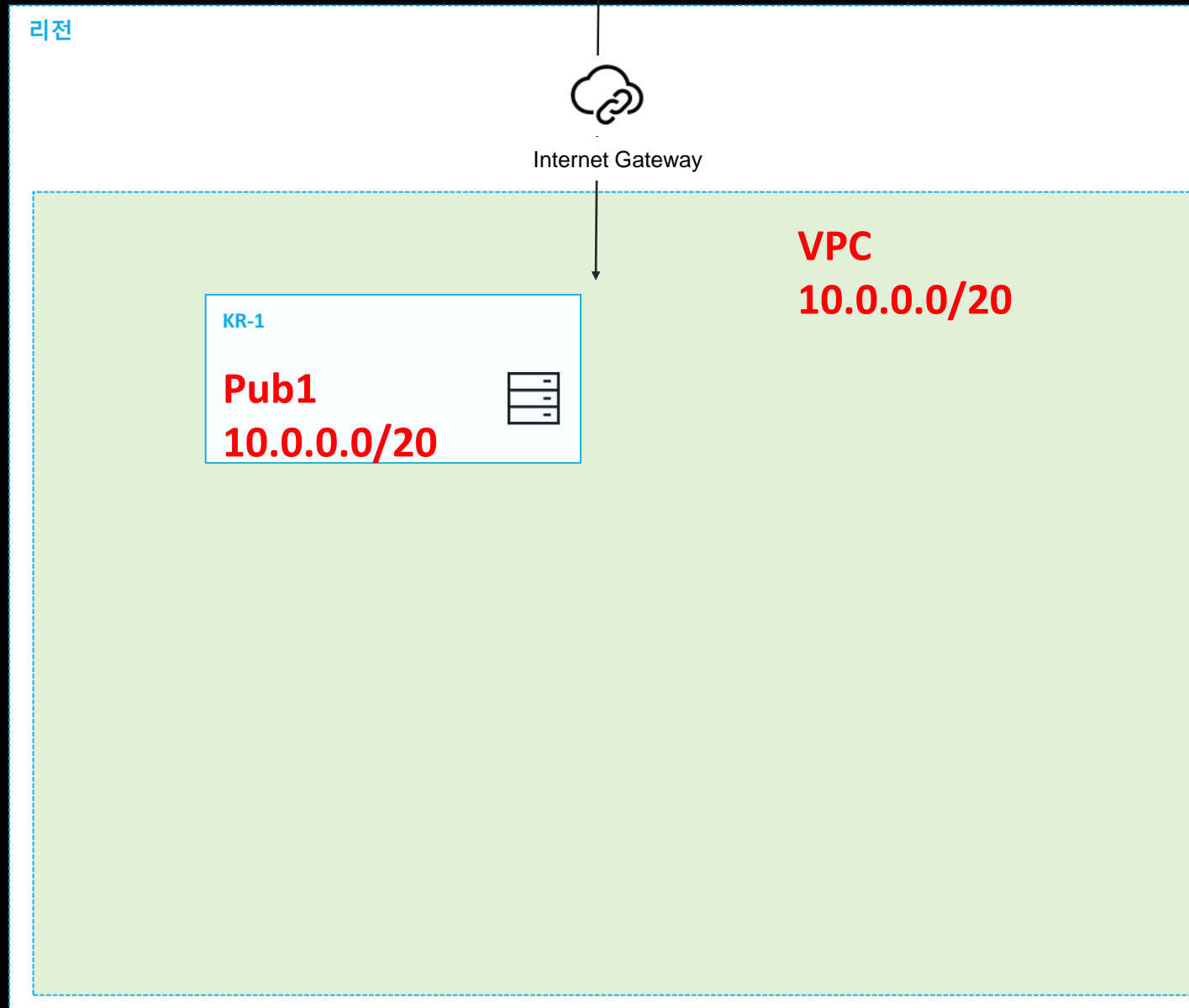
passwd

# 네이버 클라우드 네트워크27 - 로드 밸런서 생성

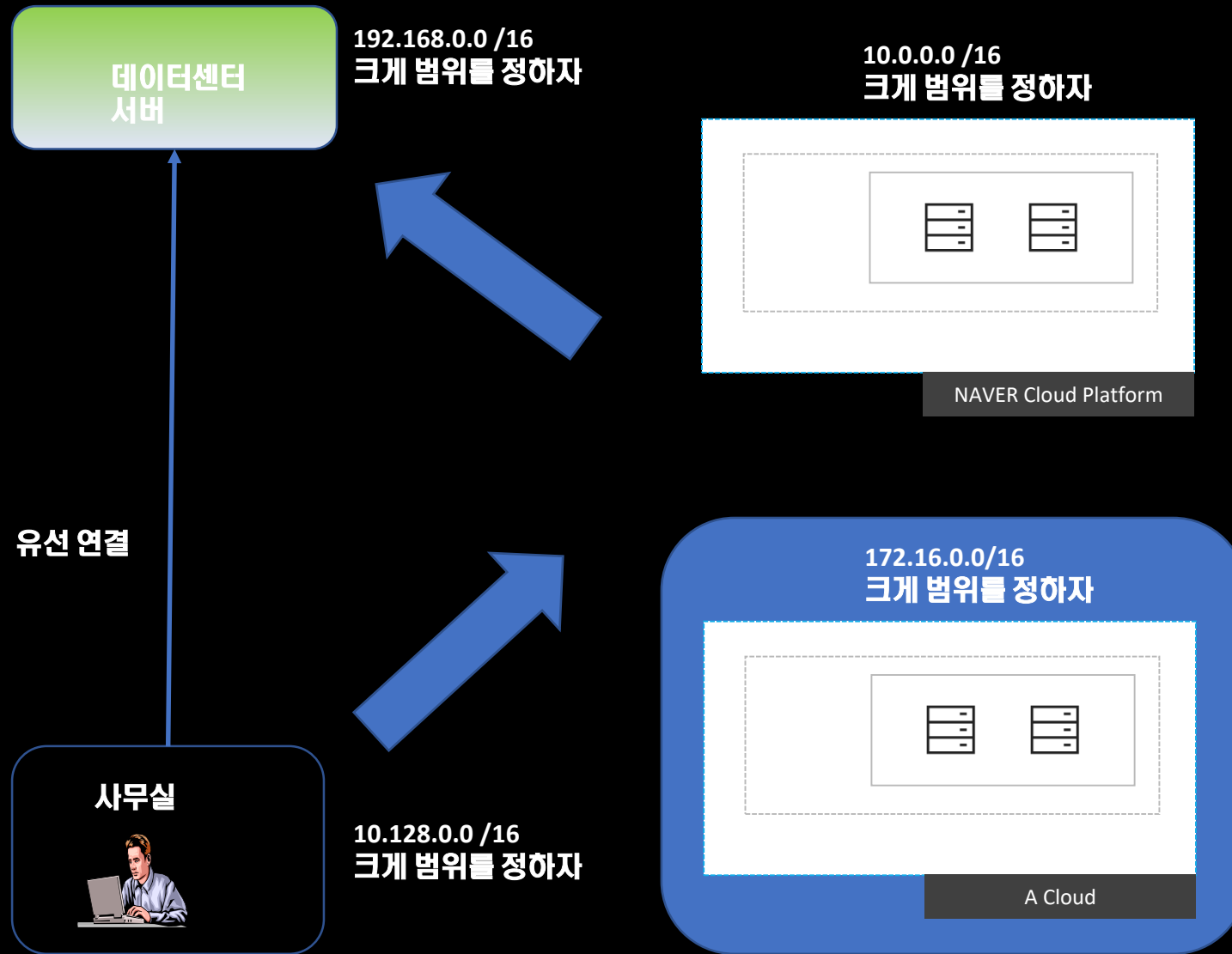
서버 생성시 웹서비스 스크립트

```
#!/bin/bash
yum -y install httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h1>Hello From Your Web Server!</h1></html>' > /var/www/html/index.html
```

# 네이버 클라우드 네트워크28 - 자주 묻는 질문1 - 서브네팅 오류

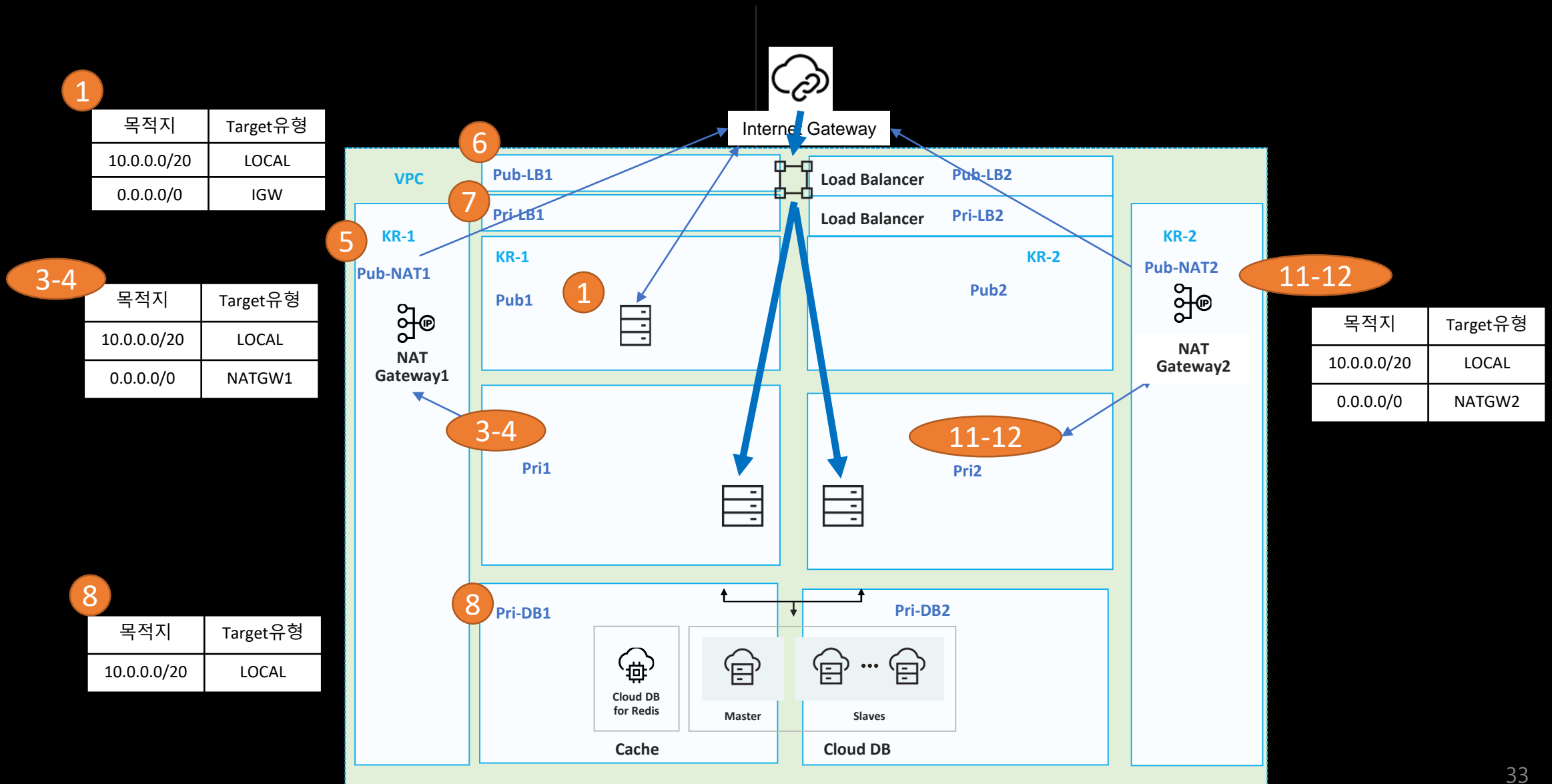


# 네이버 클라우드 네트워크29 - 자주 묻는 질문2 - CSP추가





# 네이버 클라우드 네트워크30 - 자주 묻는 질문3 - 안정적인 네트워크 구축-ISMS

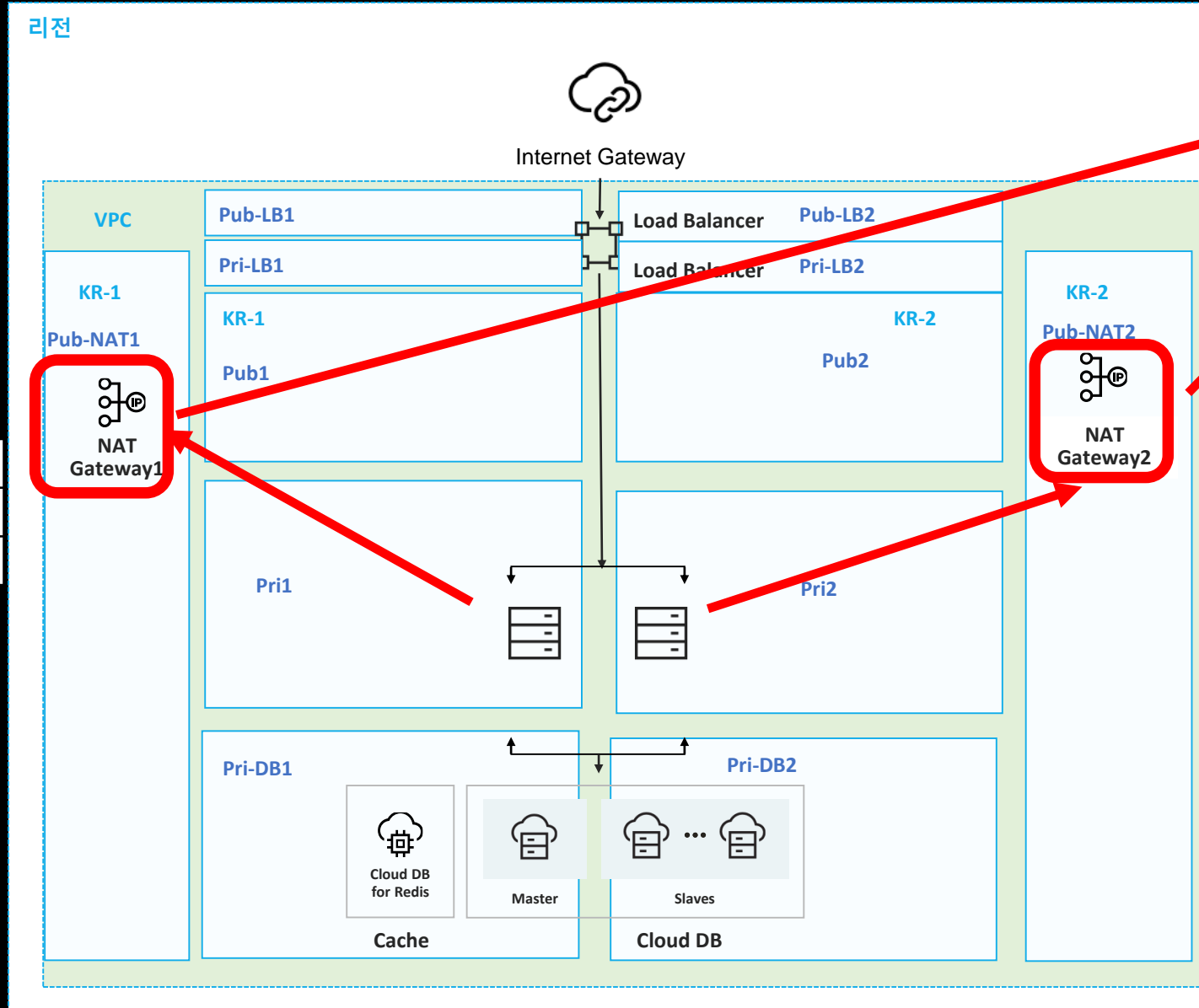


# 네이버 클라우드 네트워크31 - 자주 묻는 질문3 - 안정적인 네트워크 구축-ISMS

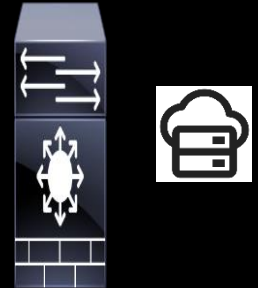
VPC	할당 VPC	C	서브넷 IP	서브넷 네임	Zone	Public/Pri
bgame-naver-dev	10.0.0.0/20	1	10.0.0.0/24	Pub1	KR-1	Pub
		2	10.0.1.0/24	예비	-	-
		3-4	10.0.2.0/23	Pri1	KR-1	Pri
		5	10.0.4.0/24	Pub-Nat1	KR-1	Pub-Nat1
		6	10.0.5.0/24	Pub-LB1	KR-1	Pub-LB1
		7	10.0.6.0/24	Pri-DB1	KR-1	Pri-DB1
		8	10.0.7.0/24	Pri-LB1	KR-1	Pri-LB1
		9	10.0.8.0/24	Pub2	KR-2	Pub
		10	10.0.9.0/24	예비	-	-
		11-12	10.0.10.0/23	Pri2	KR-2	Pri
		13	10.0.12.0/24	Pub-Nat2	KR-2	Pub-Nat2
		14	10.0.13.0/24	Pub-LB2	KR-2	Pub-LB2
		15	10.0.14.0/24	Pri-DB2	KR-2	Pri-DB2
		16	10.0.15.0/24	Pri-LB2	KR-2	Pri-LB2

# 네이버 클라우드 네트워크32 - 자주 묻는 질문4 - 외부 연동

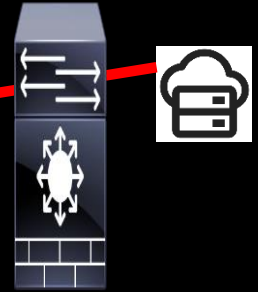
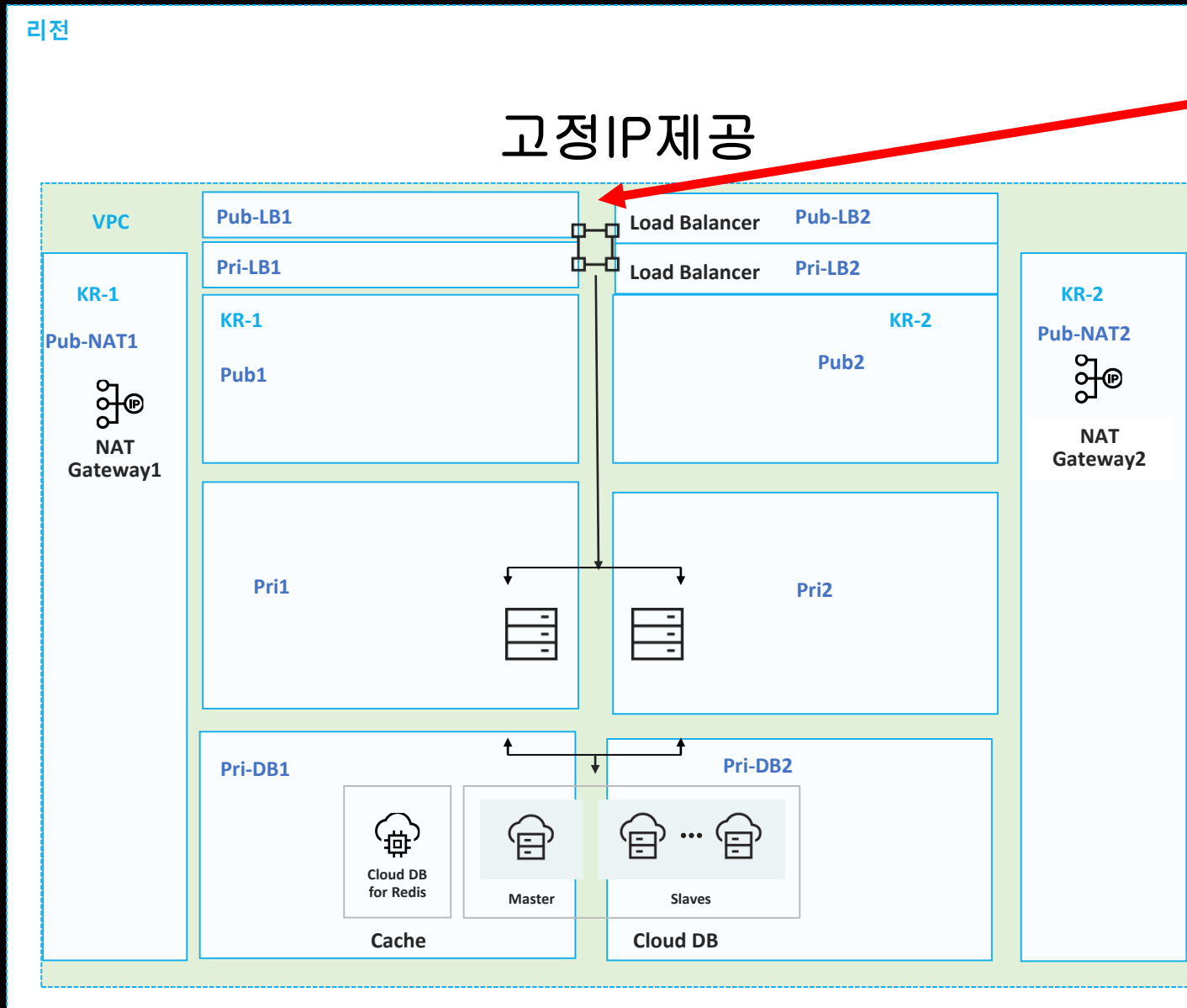
목적지	Target유형
10.0.0.0/20	LOCAL
0.0.0.0/0	NATGW1



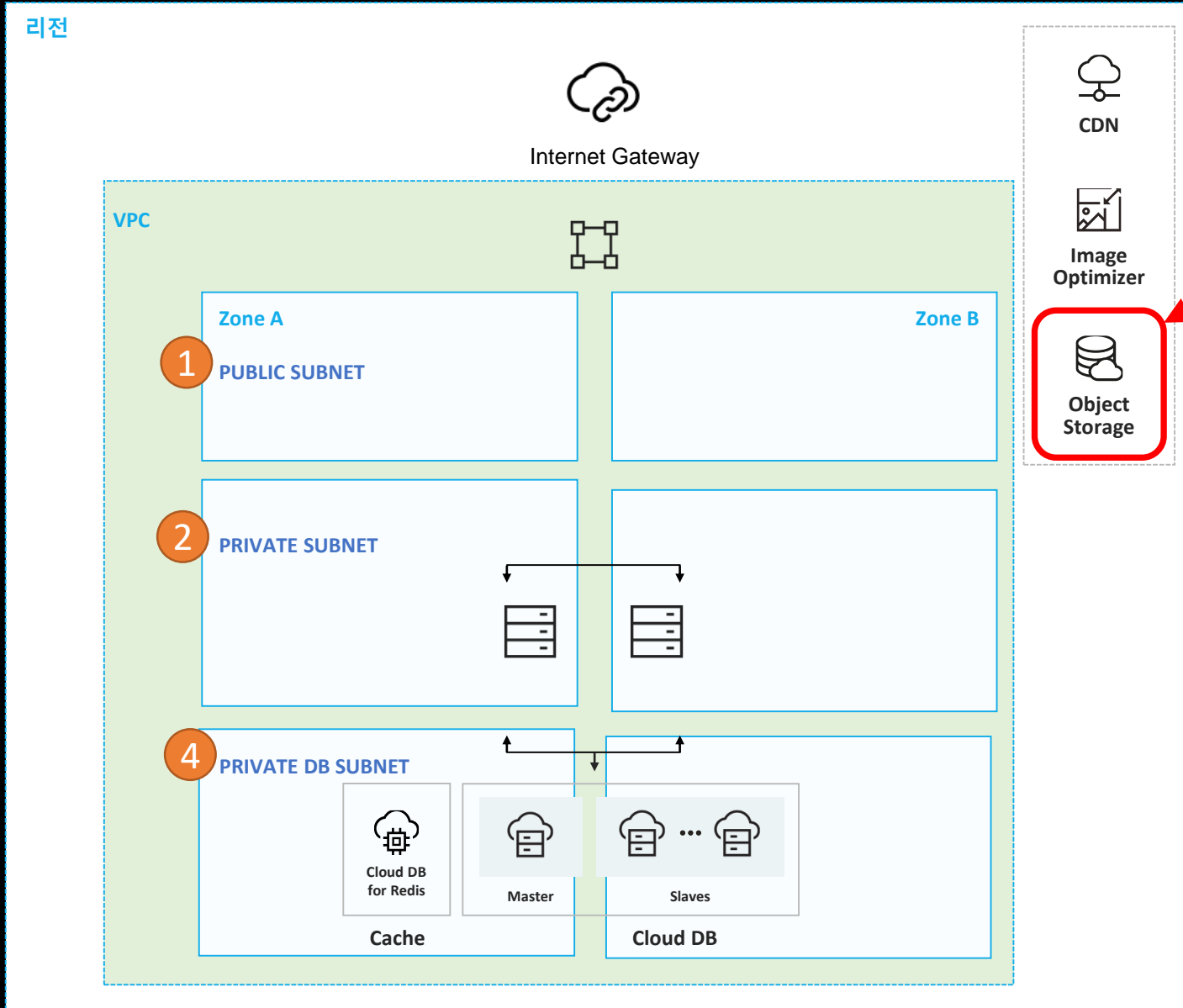
목적지	Target유형
10.0.0.0/20	LOCAL
0.0.0.0/0	NATGW2



# 네이버 클라우드 네트워크33 - 자주 묻는 질문5 - 로드밸런서 고정IP제공



# 네이버 클라우드 네트워크34 - 자주 묻는 질문6 - 오브젝트 스토리지 위치



## 2. Global DNS 사용해보자

# 네이버 클라우드 네트워크 - Global DNS등록하기 = DNS서비스1

1

DNS(Deprecated) - 종료 서비스

2

**Global DNS** 클릭해 사용

ns1-1.ns-ncloud.com

ns1-2.ns-ncloud.com

3

배포하기 (필수)

참고 자료

<https://brunch.co.kr/@topasvga/3725>

# 네이버 클라우드 네트워크 - Global DNS등록하기 = DNS서비스2

도메인

- serverup11.co.kr
- taeho11.co.kr**

레코드 추가    레코드 수정    레코드 삭제    레코드 타입 전체

등록기관 설정 정보    **설정 적용 (0)**

도메인	TTL	Master NS	Refresh	Retry	Expire	Serial
taeho11.co.kr	300	ns1-1.ns-ncloud.com. ns1-2.ns-ncloud.com.	21600	1800	1209600	11

<input type="checkbox"/>	호스트	도메인	레코드 타입	레코드 값	TTL	신청일시 (UTC+09:00)	최종 변경 일시 (UTC+09:00)
<input type="checkbox"/>	@	taeho11.co.kr	SOA	ns1-1.ns-ncloud.com. ns1-2.ns-ncloud.com. 11 21600 1800 1209600 300	300	2024-03-13 18:29:31	2024-05-20 18:14:43
<input type="checkbox"/>	@	taeho11.co.kr	NS	ns1-1.ns-ncloud.com ns1-2.ns-ncloud.com	86400	2024-03-13 18:29:31	2024-03-13 18:29:31
<input type="checkbox"/>	www	taeho11.co.kr	A	101.79.11.13 101.79.10.170	60	2024-05-20 18:12:00	2024-05-20 18:14:43
<input type="checkbox"/>	test	taeho11.co.kr	A	1.1.1.1	300	2024-04-13 23:26:15	2024-04-13 23:26:20

1

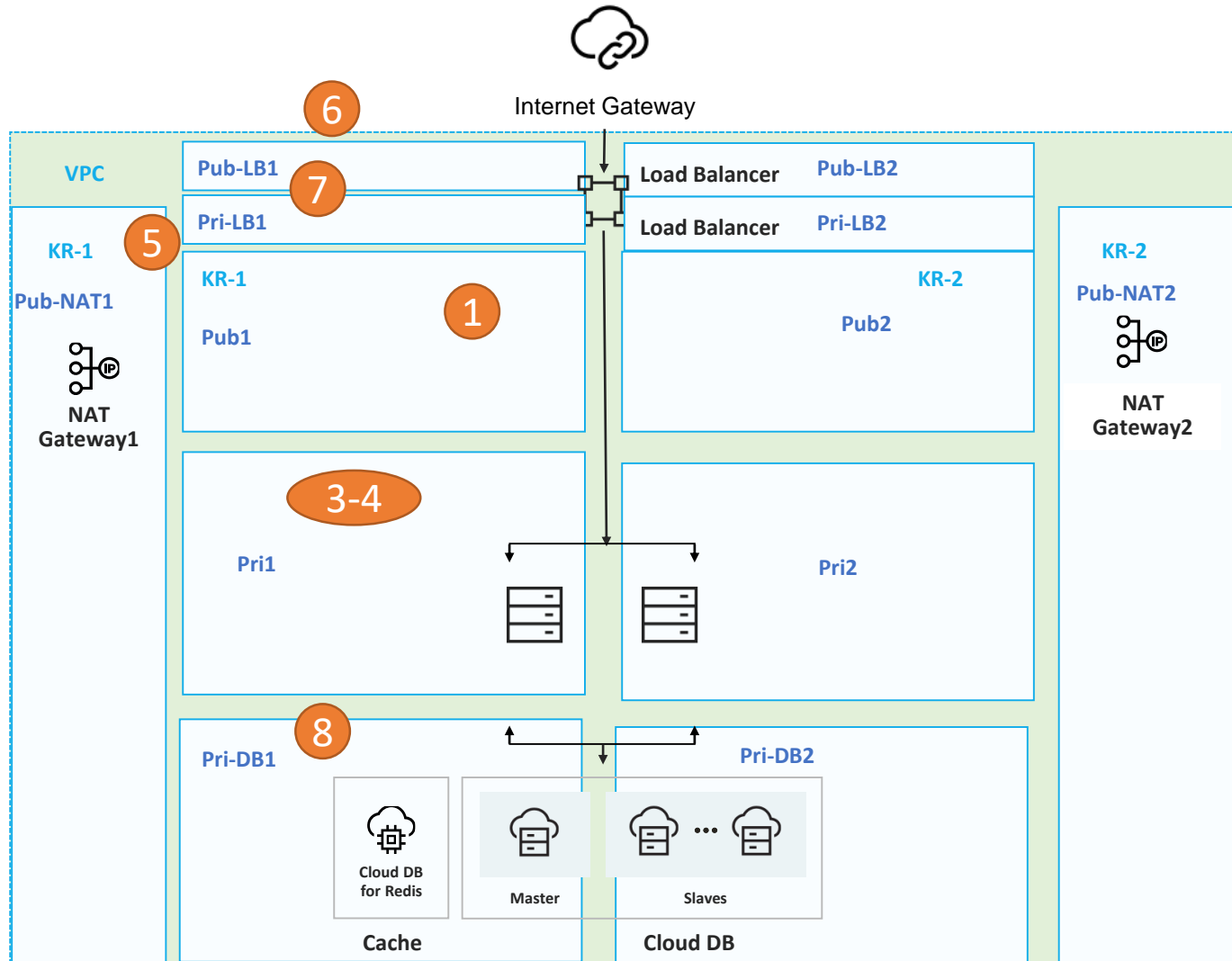
40



### 3. 쿠버네티스 사용해보자

# 네이버 클라우드 네트워크 - 쿠버네티스1

리전



## 네이버 클라우드 네트워크 - 쿠버네티스2

VPC	할당 VPC	C	서브넷 IP	서브넷 네임	Zone	Public/Pri
bgame-naver-dev	10.0.0.0/20	1	10.0.0.0/24	Pub1	KR-1	Pub
		2	10.0.1.0/24	예비	-	-
		3-4	10.0.2.0/23	Pri1	KR-1	Pri
		5	10.0.4.0/24	Pub-Nat1	KR-1	Pub-Nat1
		6	10.0.5.0/24	Pub-LB1	KR-1	Pub-LB1
		7	10.0.6.0/24	Pri-DB1	KR-1	Pri-DB1
		8	10.0.7.0/24	Pri-LB1	KR-1	Pri-LB1
		9	10.0.8.0/24	Pub2	KR-2	Pub
		10	10.0.9.0/24	예비	-	-
		11-12	10.0.10.0/23	Pri2	KR-2	Pri
		13	10.0.12.0/24	Pub-Nat2	KR-2	Pub-Nat2
		14	10.0.13.0/24	Pub-LB2	KR-2	Pub-LB2
		15	10.0.14.0/24	Pri-DB2	KR-2	Pri-DB2
		16	10.0.15.0/24	Pri-LB2	KR-2	Pri-LB2

## 네이버 클라우드 네트워크 - 쿠버네티스3

<1> 명령서버를 위한 192 네트워크 생성과 명령서버 1대 생성

vpc 192.168.0.0/16 , pub1 192.168.0.0/24로 네트워크 만든다.  
로키 리눅스로 명령서버 1대 생성한다.

# 네이버 클라우드 네트워크 - 쿠버네티스4

## <2> 쿠버네티스 용 네트워크와 쿠버네티스 생성

1

vpc생성

기본 네트워크로 pub2, pri1, pub-nat1, pub-lb1, pri-lb1 서브넷 5개 만들어보자.

## 네이버 클라우드 네트워크 - 쿠버네티스5

가용 Zone	KR-1	▼
네트워크 타입	Private	▼
Subnet	bgame-dev-pri1   KR-1   10.0.2.0/23   Private X	▼
LB Private 서브넷	bgame-dev-pri-lb1   KR-1   10.0.7.0/24   Private	▼
LB Public 서브넷	bgame-dev-pub-lb1   KR-1   10.0.5.0/24   Public	▼

LB Private 서브넷 , LB Public 서브넷 이 별도로 필수이다. (네이버 클라우드)

# 네이버 클라우드 네트워크 - 쿠버네티스6

## <3> 명령서버에 툴설치와 권한 주기

1

이용관리 > 마이 페이지 > 계정 관리

ncloud 환경 설정 하기

access-key , secret-key를 알아야 한다.

my page > 인증키 관리에서 확인할 수 있다.

2

ncloud CLI 다운로드

wget [https://www.ncloud.com/api/support/download/files/cli/CLI\\_1.1.19\\_20240321.zip](https://www.ncloud.com/api/support/download/files/cli/CLI_1.1.19_20240321.zip)

```
unzip CLI_1.1.19_20240321.zip
```

```
cd CLI_1.1.19_20240321/
```

```
cd cli_linux/
```

```
cp ncloud /usr/bin
```

```
ncloud help
```

최신버전

<https://cli.ncloud-docs.com/docs/guide-clichange>

## 네이버 클라우드 네트워크 - 쿠버네티스7

3

```
ncloud configure
```

```
ncloud server getRegionList
```

4

```
sudo yum install -y yum-utils shadow-utils
```

```
sudo yum-config-manager --add-repo
```

```
https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
```

```
sudo yum -y install terraform
```

```
terraform -version
```



# 네이버 클라우드 네트워크 – 쿠버네티스8

5

사용을 위한 인증

ncp-iam 설치

```
curl -o ncp-iam-authenticator -L https://github.com/NaverCloudPlatform/ncp-iam-authenticator/releases/latest/download/ncp-iam-authenticator_linux_amd64
```

```
chmod +x ./ncp-iam-authenticator
```

```
mkdir -p $HOME/bin && cp ./ncp-iam-authenticator $HOME/bin/ncp-iam-authenticator &&  
export PATH=$PATH:$HOME/bin
```

```
echo 'export PATH=$PATH:$HOME/bin' >> ~/.bash_profile  
ncp-iam-authenticator help
```

메뉴얼

<https://guide.ncloud-docs.com/docs/k8s-iam-auth-ncp-iam-authenticator>

## 네이버 클라우드 네트워크 - 쿠버네티스9

6

cd

```
ncp-iam-authenticator create-kubeconfig --region KR --clusterUuid b01xxxxxxxxxx --output  
kubeconfig.yaml
```

7

kw

# 네이버 클라우드 네트워크 - 쿠버네티스10

## <4> 쿠버네티스에 서비스 올리기

1

```
kubectl create deployment nginx-project --image=nginx --dry-run=client -o yaml --port=80 >  
nginx-deploy.yaml --kubeconfig kubeconfig.yaml
```

```
kubectl apply -f nginx-deploy.yaml --kubeconfig kubeconfig.yaml
```

```
kubectl get pods --kubeconfig kubeconfig.yaml
```

# 네이버 클라우드 네트워크 - 쿠버네티스11

2

```
cat <<EOF | k create -f -
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-2048
spec:
  selector:
    matchLabels:
      app.kubernetes.io/name: app-2048
  replicas: 2
  template:
    metadata:
      labels:
        app.kubernetes.io/name: app-2048
    spec:
      containers:
        - image: alexwhen/docker-2048
          name: app-2048
          ports:
            - containerPort: 80
EOF

k expose deployment deployment-2048 --port=80 --type=LoadBalancer
```

## 네이버 클라우드 네트워크 - 쿠버네티스12

3

```
vi ~/.bash_profile
```

맨 아래줄에 아래 내용 추가

```
alias k='kubectl --kubeconfig="/root/kubeconfig.yaml"'
```

```
alias kw='watch -d kubectl get deploy,svc,pods --kubeconfig="/root/kubeconfig.yaml"'
```

```
source ~/.bash_profile
```

```
k get nodes
```

4

콘솔에서 사설 라우팅 테이블에 0.0.0.0/0 NAT1 설정을 하면 pod들이 서비스 가능해진다.

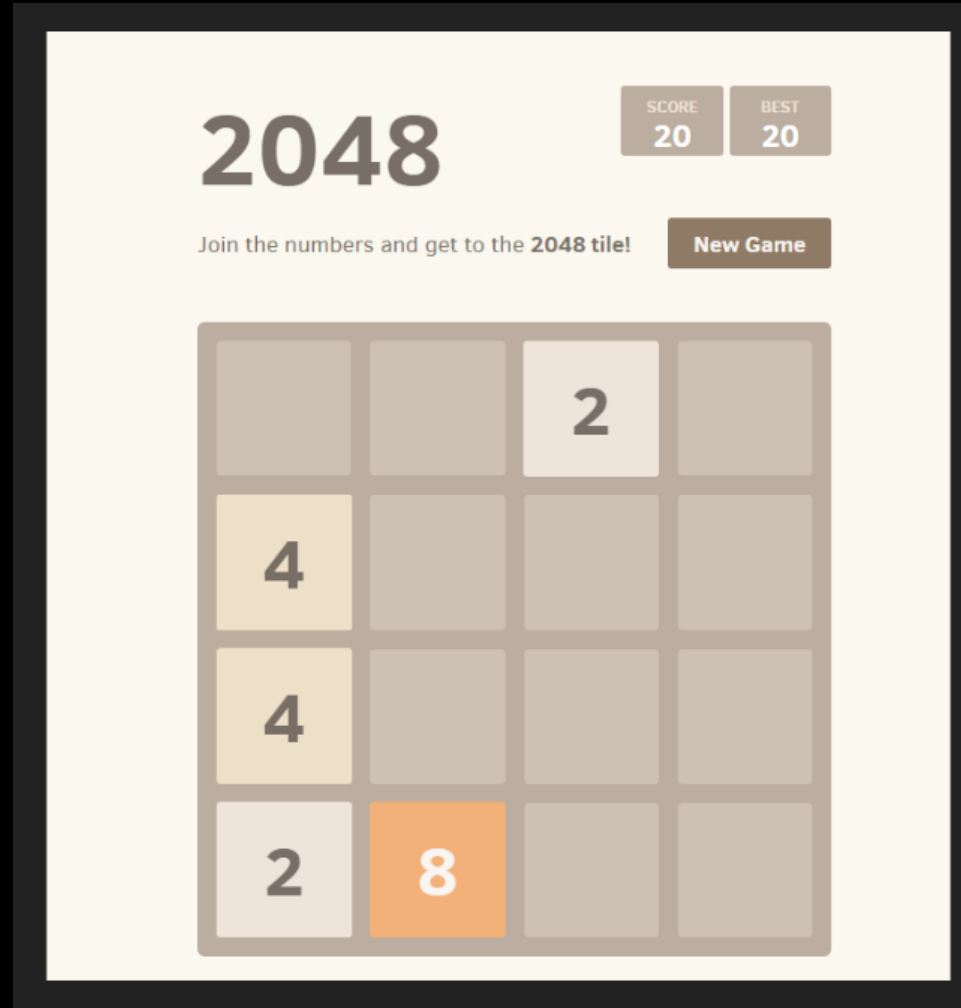
```
kw
```

모니터링 상태

# 네이버 클라우드 네트워크 - 쿠버네티스13

5

서비스 동작확인 = 로드 밸런서로 접속하기



## 네이버 클라우드 네트워크 - 리소스 삭제14

<6> 리소스 삭제 - 라우팅 테이블 삭제, 쿠버네티스 삭제, LB 삭제

1

라우팅 테이블에서 NAT 설정 삭제, 프라이빗에 0.0.0.0/0 NAT1 라우팅 테이블 삭제

2

NAT1 삭제

3

쿠버네티스 삭제 (5분)

쿠버네티스는 Server에서 반납이 안됨. Ncloud Kubernetes Service에서 삭제해야 함.

4

LB 삭제 = 네트워크 삭제 - VPC 삭제 전 서브넷부터 삭제하자.

5

서브넷 삭제

6

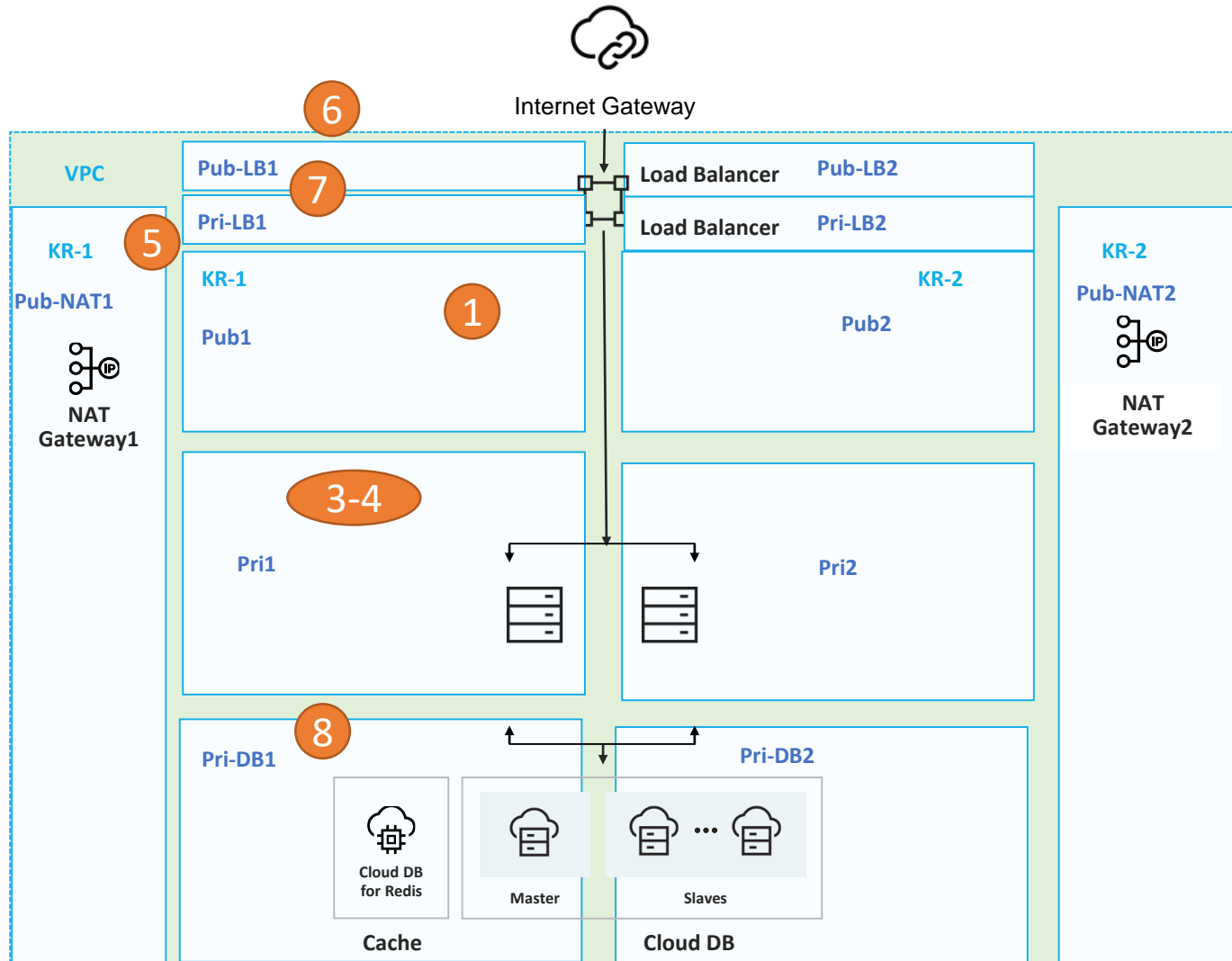
VPC 삭제

## 4. 테라폼 사용해보자



# 네이버 클라우드 네트워크 - 테라폼1

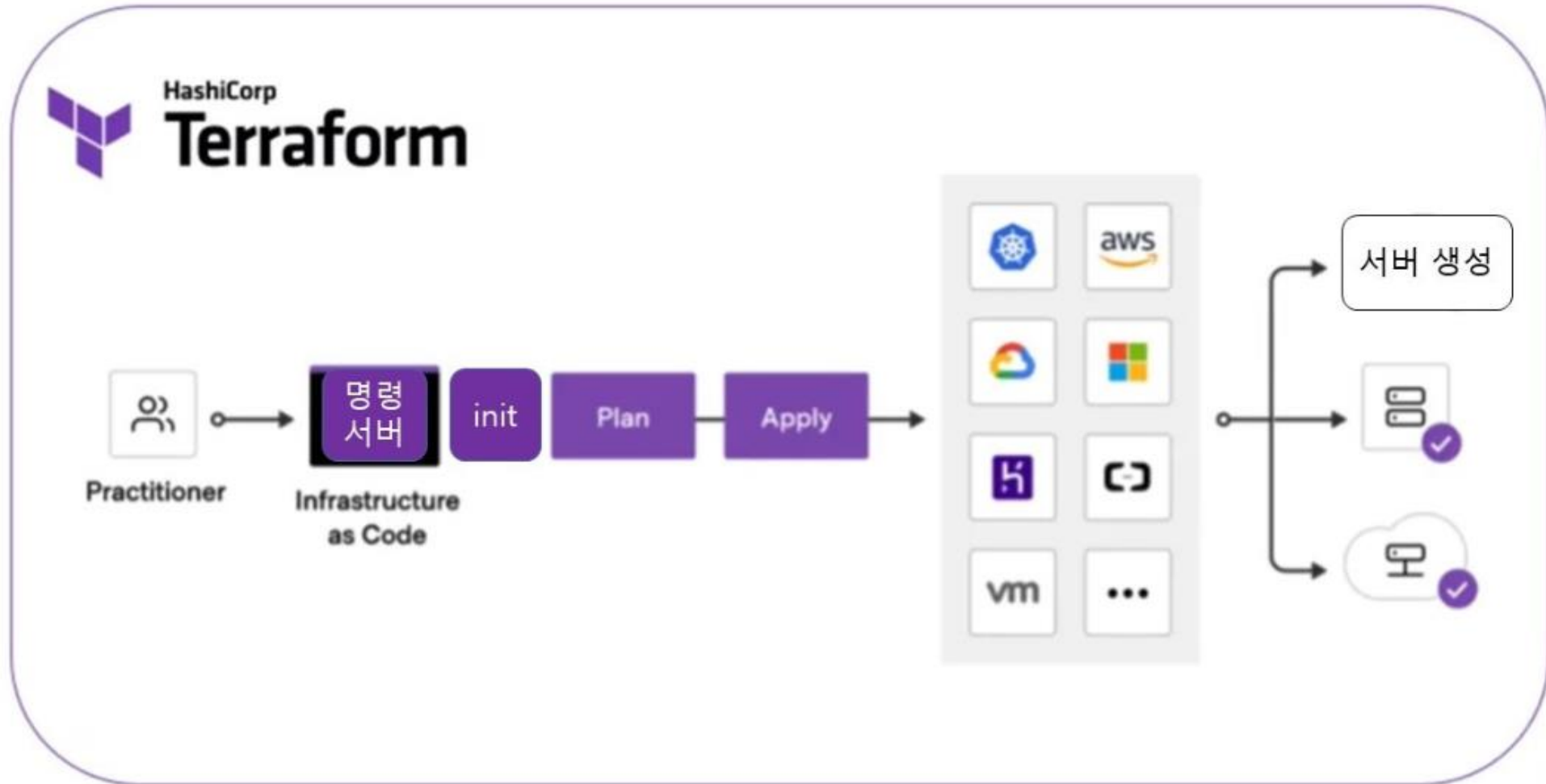
리전



## 네이버 클라우드 네트워크 - 테라폼2

VPC	할당 VPC	C	서브넷 IP	서브넷 네임	Zone	Public/Pri
bgame-naver-dev	10.0.0.0/20	1	10.0.0.0/24	Pub1	KR-1	Pub
		2	10.0.1.0/24	예비	-	-
		3-4	10.0.2.0/23	Pri1	KR-1	Pri
		5	10.0.4.0/24	Pub-Nat1	KR-1	Pub-Nat1
		6	10.0.5.0/24	Pub-LB1	KR-1	Pub-LB1
		7	10.0.6.0/24	Pri-DB1	KR-1	Pri-DB1
		8	10.0.7.0/24	Pri-LB1	KR-1	Pri-LB1
		9	10.0.8.0/24	Pub2	KR-2	Pub
		10	10.0.9.0/24	예비	-	-
		11-12	10.0.10.0/23	Pri2	KR-2	Pri
		13	10.0.12.0/24	Pub-Nat2	KR-2	Pub-Nat2
		14	10.0.13.0/24	Pub-LB2	KR-2	Pub-LB2
		15	10.0.14.0/24	Pri-DB2	KR-2	Pri-DB2
		16	10.0.15.0/24	Pri-LB2	KR-2	Pri-LB2

# 네이버 클라우드 네트워크 - 테라폼3



## 네이버 클라우드 네트워크 - 테라폼4

여기서는 명령서버 만들자~

<1> 192.168 네트워크에, 콘솔로 테라폼 설치된 명령서버 1대 생성

1

Pub1 서브넷에 로키 리눅스나 우분트로 서버를 생성한다.

로키 리눅스에서는 테라폼 Amazon Linux 버전으로 설치해야 동작한다.

passwd

2

access-key , secret-key 생성해 준비한다.

콘솔 로그인 > 계정 관리 > 인증키 관리에서 access-key , secret-key 생성해 메모장에 복사해둔다.

# 네이버 클라우드 네트워크 - 테라폼5

3

서버에 로그인 - passwd

4

ncloud CLI 다운로드

wget [https://www.ncloud.com/api/support/download/files/cli/CLI\\_1.1.19\\_20240321.zip](https://www.ncloud.com/api/support/download/files/cli/CLI_1.1.19_20240321.zip)

unzip CLI\_1.1.19\_20240321.zip

cd CLI\_1.1.19\_20240321/

cd cli\_linux/

cp ncloud /usr/bin

ncloud help

최신 버전

<https://cli.ncloud-docs.com/docs/guide-clichange>

## 네이버 클라우드 네트워크 - 테라폼6

5

네이버 클라우드 리소스 만들수 있도록 권한 받기

Access키, Secret키로 권한 받기

ncloud configure

6

권한 부여 확인

ncloud server getRegionList

# 네이버 클라우드 네트워크 - 테라폼7

7  
테라폼 설치 ?  
로키 리눅스의 경우 - 아마존 리눅스 버전 설치하면 동작 한다.  
Linux - Amazon Linux

```
sudo yum install -y yum-utils shadow-utils  
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo  
sudo yum -y install terraform  
terraform -version
```

8  
쿠버네티스 API에 명령을 내리는 kubectl 명령어 설치

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"  
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl  
kubectl version
```

## 네이버 클라우드 네트워크 - 테라폼8

9

TF 소스 받아 쿠버네티스 클러스터 생성

테라폼 쿠버네티스 소스

cd

wget <https://github.com/NaverCloudPlatform/terraform-provider-ncloud/archive/refs/heads/master.zip>

unzip [master.zip](#)

cd /root/terraform-provider-ncloud-main/examples/nks



# 네이버 클라우드 네트워크 - 테라폼9

10

네이버 쿠버네티스 서비스의 권한을 받기 위한 툴 설치

ncp-iam-authenticator 설치

```
curl -o ncp-iam-authenticator -L https://github.com/NaverCloudPlatform/ncp-iam-authenticator/releases/latest/download/ncp-iam-authenticator_linux_amd64
```

```
chmod +x ./ncp-iam-authenticator
```

```
mkdir -p $HOME/bin && cp ./ncp-iam-authenticator $HOME/bin/ncp-iam-authenticator &&  
export PATH=$PATH:$HOME/bin
```

```
echo 'export PATH=$PATH:$HOME/bin' >> ~/.bash_profile  
ncp-iam-authenticator help
```

메뉴얼

<https://guide.ncloud-docs.com/docs/k8s-iam-auth-ncp-iam-authenticator>

# 네이버 클라우드 네트워크 - 테라폼10

```
//  
인증  
cd  
ncp-iam-authenticator create-kubeconfig --region KR --clusterUuid a2143bde-4e6f-44d8-bfaf-075ede2a452f --  
output kubeconfig.yaml
```

11  
테라폼으로 쿠버네티스 생성해 보자

아래 3과정을 거친다.

```
terraform init  
terraform plan  
terraform apply -auto-approve
```

# 네이버 클라우드 네트워크 - 테라폼11

12

변수 파일~~~~  
값을 넣어야 한다!!!!

vi variables.tf

```
[root@command1 nks]# more variables.tf
```

```
variable name {  
  default = "tf-nks"  
}  
variable nks_version {  
  default = "1.27"  
}  
variable client_ip {  
  default = "YOUR_CLIENT_IP"  
}  
variable access_key {  
  default = "YOUR_ACCESS_KEY"  
}  
variable secret_key {  
  default = "YOUR_SECRET_KEY"  
}  
variable login_key {  
  default = "YOUR_LOGIN_KEY"  
}  
[root@command1 nks]#
```

## 네이버 클라우드 네트워크 - 테라폼12

13

```
[root@command1 nks]# terraform plan
```

```
Planning failed. Terraform encountered an error while generating this plan.
```

```
|  
| Error: Status: 401 Unauthorized, Body: {"error":{"errorCode":"200","message":"Authentication  
Failed","details":"This account is not allowed."}}
```

```
|  
| with provider["registry.terraform.io/navercloudplatform/ncloud"],  
| on main.tf line 4, in provider "ncloud":  
| 4: provider "ncloud" {
```

```
[root@command1 nks]#
```

테라폼으로 쿠버네티스를 생성하려니 권한이 없다는 것이다.

테라폼 변수 파일에 Access-key나 Secret-key를 입력해야 권한이 생긴다.

# 네이버 클라우드 네트워크 - 테라폼13

14

terraform plan

ncloud\_nks\_cluster.cluster: Creating...

```
|  
| Error: Status: 400 Bad Request, Body: {"error":{"errorCode":400,"message":"Bad Request","details":"Require  
lbPublicSubnetNo"},"timestamp":"2024-06-01T11:10:53.147Z"}
```

```
|  
| with ncloud_nks_cluster.cluster,  
| on main.tf line 48, in resource "ncloud_nks_cluster" "cluster":  
| 48: resource "ncloud_nks_cluster" "cluster" {  
|  
|
```

[root@command1 nks]#

## 네이버 클라우드 네트워크 - 테라폼14

15

public lb 추가 필요

vi main.tf

```
resource "ncloud_subnet" "lb_subnet2" {  
  vpc_no      = ncloud_vpc.vpc.id  
  subnet      = "10.0.102.0/24"  
  zone        = "KR-1"  
  network_acl_no = ncloud_vpc.vpc.default_network_acl_no  
  subnet_type  = "PUBLIC"  
  name         = "lb-subnet2"  
  usage_type   = "LOADB"  
}
```

## 네이버 클라우드 네트워크 - 테라폼15

```
resource "ncloud_nks_cluster" "cluster" {  
  cluster_type      = "SVR.VNKS.STAND.C002.M008.NET.SSD.B050.G002"  
  k8s_version      = data.ncloud_nks_versions.version.versions.0.value  
  login_key_name   = ncloud_login_key.loginkey.key_name  
  name             = "sample-cluster"  
  lb_private_subnet_no = ncloud_subnet.lb_subnet.id  
  lb_public_subnet_no = ncloud_subnet.lb_subnet2.id
```

## 네이버 클라우드 네트워크 - 테라폼16

16

taint 부분 삭제~

```
label {  
  key = "foo"  
  value = "bar"  
}  
taint {  
  key = "foo"  
  value = "bar"  
  effect = "NoExecute"  
}
```

<https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/>



# 네이버 클라우드 네트워크 - 테라폼17

```
terraform apply -auto-approve
```

35분 후

## 네이버 클라우드 네트워크 - 테라폼18

17

생성된 쿠버네티스 권한 등록

cd

```
ncp-iam-authenticator create-kubeconfig --region KR --clusterUuid b01xxxxxxxxxx --output kubeconfig.yaml
```

```
ncp-iam-authenticator create-kubeconfig --region <region-code> --clusterUuid <cluster-uuid> --output  
kubeconfig.yaml
```

콘솔에서 확인

# 네이버 클라우드 네트워크 - 테라폼19

18

```
vi ~/.bash_profile
```

맨 아래줄에 아래 내용 추가

```
alias k='kubectl --kubeconfig="/root/kubeconfig.yaml"'
```

```
alias kw='watch -d kubectl get deploy,svc,pods --kubeconfig="/root/kubeconfig.yaml"'
```

```
source ~/.bash_profile
```

```
k get nodes
```

참고사이트

<https://medium.com/naver-cloud-platform/cbe27e197a97>

# 네이버 클라우드 네트워크 - 테라폼20

19

```
cat <<EOF | k create -f -
```

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  name: deployment-2048
```

```
spec:
```

```
  selector:
```

```
    matchLabels:
```

```
      app.kubernetes.io/name: app-2048
```

```
  replicas: 2
```

```
  template:
```

```
    metadata:
```

```
      labels:
```

```
        app.kubernetes.io/name: app-2048
```

```
    spec:
```

```
      containers:
```

```
        - image: alexwhen/docker-2048
```

```
          name: app-2048
```

```
          ports:
```

```
            - containerPort: 80
```

```
EOF
```

```
k expose deployment deployment-2048 --port=80 --type=LoadBalancer
```

20

KW

# 네이버 클라우드 네트워크 - 테라폼21

21

```
cd /root/terraform-provider-ncloud-main/examples/nks
```

```
pub1.tf
```

```
.....
```

```
resource "ncloud_subnet" "pub1_subnet" {  
  vpc_no      = ncloud_vpc.vpc.id  
  subnet      = "10.0.0.0/24"  
  zone        = "KR-1"  
  network_acl_no = ncloud_vpc.vpc.default_network_acl_no  
  subnet_type  = "PUBLIC"  
  name         = "pub1-subnet"  
  usage_type   = "GEN"  
}
```

```
terraform plan
```

```
terraform apply -auto-approve
```

## 네이버 클라우드 네트워크 - 테라폼22

22

명령서버를 Server Image 로 만들자.

Server Image를 이용해 pub1에 명령서버를 만들자~

## 네이버 클라우드 네트워크 - 테라폼23

- <1> 주석 알아보기
- <2> ncloud CLI
- <3> ncloud 소스
- <4> 테라폼 네트워크에 NAT 추가
- <5> 테라폼으로 네트워크 + 쿠버네티스 구축
- <6> 쿠버네티스 사용하기
- <7> 테라폼으로 VPC, 쿠버네티스 생성하기 테라폼 파일
- <8> NATGW 라우팅 테이블 별도로 만들기 (선택)
- <9> 삭제

# 네이버 클라우드 네트워크 - 테라폼24

<1> 주식 알아보자

#

//

/\*

\*/



# 네이버 클라우드 네트워크 - 테라폼25

```
cd /root/terraform-provider-ncloud-main/examples/nks

1
vi nat.tf

resource "ncloud_subnet" "subnet_scn_02_public_natgw" {
# vpc_no      = ncloud_vpc.vpc_scn_02.id
  vpc_no      = ncloud_vpc.vpc.id
# subnet      = cidrsubnet(ncloud_vpc.vpc_scn_02.ipv4_cidr_block, 8, 2)
  subnet      = "10.0.2.0/24"
  zone        = "KR-1"
  network_acl_no = ncloud_vpc.vpc.default_network_acl_no
# network_acl_no = ncloud_network_acl.network_acl_02_public.id
  subnet_type = "PUBLIC"
  usage_type  = "NATGW"
}
# NAT Gateway
resource "ncloud_nat_gateway" "nat_gateway_scn_02" {
# vpc_no      = ncloud_vpc.vpc_scn_02.id
  vpc_no      = ncloud_vpc.vpc.id
  subnet_no   = ncloud_subnet.subnet_scn_02_public_natgw.id
  zone        = "KR-1"
  name        = var.name_scn02
#name        = "nat_gateway_scn_02"
}
# Route Table
resource "ncloud_route" "route_scn_02_nat" {
  route_table_no      = ncloud_vpc.vpc.default_private_route_table_no
  destination_cidr_block = "0.0.0.0/0"
  target_type          = "NATGW"
  // NATGW (NAT Gateway) | VPCPEERING (VPC Peering) | VGW (Virtual Private Gateway).
  target_name          = ncloud_nat_gateway.nat_gateway_scn_02.name
  target_no             = ncloud_nat_gateway.nat_gateway_scn_02.id
}
```

## 네이버 클라우드 네트워크 - 테라폼26

2

```
[root@command1 nks]# terraform plan
```

```
|  
| Error: Reference to undeclared input variable  
|  
|   on nat.tf line 37, in resource "ncloud_nat_gateway" "nat_gateway_scn_02":  
|   37:   name    = var.name_scn02  
|  
| An input variable with the name "name_scn02" has not been declared. This variable can be declared with a variable "name_scn02" {}  
| block.  
|
```

```
[root@command1 nks]#
```

```
vi variables.tf
```

```
variable name_scn02 {  
  default = "tf-scn02"  
}
```

3

```
terraform apply -auto-approve
```

## 네이버 클라우드 네트워크 - 테라폼27

<2> ncloud CLI

1

ncloud CLI 다운로드

wget [https://www.ncloud.com/api/support/download/files/cli/CLI\\_1.1.19\\_20240321.zip](https://www.ncloud.com/api/support/download/files/cli/CLI_1.1.19_20240321.zip)

```
unzip CLI_1.1.19_20240321.zip
```

```
cd CLI_1.1.19_20240321/
```

```
cd cli_linux/
```

```
cp ncloud /usr/bin
```

```
ncloud help
```

최신본 다운로드

<https://cli.ncloud-docs.com/docs/guide-clicchange>

## 네이버 클라우드 네트워크 - 테라폼28

2

ncloud configure

set [DEFAULT]'s configuration.

Ncloud Access Key ID []: EQCygJ E E E E E E E E E E E E E E E E

Ncloud Secret Access Key []: 3z0xY E E E E E E E E E E E E E E E E

Ncloud API URL (default:https://ncloud.apigw.ntruss.com) []:

3

ncloud server getRegionList

## 네이버 클라우드 네트워크 - 테라폼29

<3> ncloud 소스

1

cd

wget <https://github.com/NaverCloudPlatform/terraform-provider-ncloud/archive/refs/heads/master.zip>

unzip [master.zip](#)

cd /root/terraform-provider-ncloud-main/examples/nks

## 네이버 클라우드 네트워크 - 테라폼30

```
2
vi variables.tf
[root@cl4-na-dev-command1 nks]# vi variables.tf
variable name {
  default = "tf-nks"
}
variable nks_version {
  default = "1.27"
}
variable client_ip {
  default = "2.2.2.2"
}
variable access_key {
  default = "B6wVLur"
}
variable secret_key {
  default = "CvT89R"
}
variable login_key {
  default = "kk1"
}
}
```

## 네이버 클라우드 네트워크 - 테라폼31

3

로키 리눅스에 테라폼 설치

```
sudo yum install -y yum-utils shadow-utils
```

```
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
```

```
sudo yum -y install terraform
```

```
terraform -version
```

```
terraform init
```

```
terraform plan
```

# 네이버 클라우드 네트워크 - 테라폼32

```
terraform apply -auto-approve
```

```
| Error: Status: 400 Bad Request, Body: {"error":{"errorCode":400,"message":"Bad Request","details":"Require  
lbPublicSubnetNo"},"timestamp":"2024-05-30T20:22:08.172Z"}
```

```
|  
| with ncloud_nks_cluster.cluster,  
| on main.tf line 48, in resource "ncloud_nks_cluster" "cluster":  
| 48: resource "ncloud_nks_cluster" "cluster" {
```

```
[root@new2222222 nks]# more lb-pub2.tf  
resource "ncloud_subnet" "lb_subnet2" {  
  vpc_no      = ncloud_vpc.vpc.id  
  subnet      = "10.0.102.0/24"  
  zone        = "KR-1"  
  network_acl_no = ncloud_vpc.vpc.default_network_acl_no  
  subnet_type  = "PUBLIC"  
  name         = "lb-subnet2"  
  usage_type   = "LOADB"  
}
```

```
vi main.tf
```

추가

```
lb_public_subnet_no = ncloud_subnet.lb_subnet2.id
```



## 네이버 클라우드 네트워크 - 테라폼33

```
4
| Error: Create Vpc Instance, err params={0xc000654618 0xc0001a7838 0xc0001a7868}
|
| with ncloud_vpc.vpc,
| on main.tf line 11, in resource "ncloud_vpc" "vpc":
| 11: resource "ncloud_vpc" "vpc" {
|
| Status: 400 Bad Request, Body: {
|   "responseError": {
|     "returnCode": "1000037",
|     "returnMessage": "Cannot create with duplicate VPC name."
|   }
| }
```

콘솔에서 기존 VPC 삭제

## 네이버 클라우드 네트워크 - 테라폼34

```
5
| Error: Error Creating LoginKey
|
| with ncloud_login_key.loginkey,
| on main.tf line 44, in resource "ncloud_login_key" "loginkey":
| 44: resource "ncloud_login_key" "loginkey" {
|
| Status: 500 Internal Server Error, Body: {
|   "responseError": {
|     "returnCode": "140002",
|     "returnMessage": "LOGIN KEY with the same name already exists."
|   }
| }
|
```

변수 파일에서 login key 변경

```
[root@new222222 nks]# vi variables.tf
```

**terraform apply -auto-approve**

NKS 생성까지 35분

## 네이버 클라우드 네트워크 - 테라폼35

<4> 테라폼 네트워크에 NAT 추가

아래 네트워크를 구축해보자.

NAT 추가!!!

```
root@command2:~/terraform-provider-ncloud-main/examples/nks# ls  
backup main.tf nat.tf pub1.tf terraform.tfstate terraform.tfstate.backup variables.tf versions.tf
```

참고 사이트

<https://brunch.co.kr/@topasvga/3828>

# 네이버 클라우드 네트워크 - 테라폼36

<5> 테라폼으로 네트워크 + 쿠버네티스 구축

1

우선 동작만 하게 코드를 편집했습니다.  
코드는 변수처리로 하거나 좀 정리해야 합니다~

```
terraform init
terraform plan
terraform apply -auto-approve
```

약 35분 소요 됨.

클러스터 16분  
node 19분

참고 사이트

<https://brunch.co.kr/@topasvga/3828>

# 네이버 클라우드 네트워크 - 테라폼37

## <6> 쿠버네티스 사용하기

1

사용을 위한 인증

ncp-iam 설치

```
curl -o ncp-iam-authenticator -L https://github.com/NaverCloudPlatform/ncp-iam-authenticator/releases/latest/download/ncp-iam-authenticator_linux_amd64
```

```
chmod +x ./ncp-iam-authenticator
```

```
mkdir -p $HOME/bin && cp ./ncp-iam-authenticator $HOME/bin/ncp-iam-authenticator &&  
export PATH=$PATH:$HOME/bin
```

```
echo 'export PATH=$PATH:$HOME/bin' >> ~/.bash_profile  
ncp-iam-authenticator help
```

메뉴얼

<https://guide.ncloud-docs.com/docs/k8s-iam-auth-ncp-iam-authenticator>

## 네이버 클라우드 네트워크 - 테라폼38

등록

```
ncp-iam-authenticator create-kubeconfig --region KR --clusterUuid b01xxxxxxxxxx --output kubeconfig.yaml
```

```
ncp-iam-authenticator create-kubeconfig --region <region-code> --clusterUuid <cluster-uuid> --output  
kubeconfig.yaml
```

## 네이버 클라우드 네트워크 - 테라폼39

2

kubectl 명령어 설치

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"  
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

3

vi ~/.bash\_profile

맨 아래줄에 아래 내용 추가

```
alias k='kubectl --kubeconfig="/root/kubeconfig.yaml"'
```

```
alias kw='watch -d kubectl get deploy,svc,pods --kubeconfig="/root/kubeconfig.yaml"'
```

```
source ~/.bash_profile
```

```
k get nodes
```

# 네이버 클라우드 네트워크 - 테라폼40

```
4
cat <<EOF | kubectl create -f -

cat <<EOF | k create -f -
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deployment-2048
spec:
  selector:
    matchLabels:
      app.kubernetes.io/name: app-2048
  replicas: 2
  template:
    metadata:
      labels:
        app.kubernetes.io/name: app-2048
    spec:
      containers:
        - image: alexwhen/docker-2048
          name: app-2048
          ports:
            - containerPort: 80
EOF
```

클래식 로드 밸런서 연결 = 외부에서 접속

```
k expose deployment deployment-2048 --port=80 --type=LoadBalancer
```



## 네이버 클라우드 네트워크 - 테라폼41

<7> 테라폼으로 VPC, 쿠버네티스 생성하기 테라폼 파일

1

```
cd /root/terraform-provider-ncloud-main/examples/nks
```

2

```
vi main.tf
```

소스에서 taints 부분 삭제 또는 주석 처리

## 네이버 클라우드 네트워크 - 테라폼42

3

terraform plan

terraform apply -auto-approve

참고 자료

<https://brunch.co.kr/@topasvga/3828>

# 감사합니다.

발표자 Master Seo  
topasvga@naver.com

참고자료

<https://brunch.co.kr/@topasvga/3806>

**NAVER CLOUD**

**MASTER**

**PLATFORM**