



laas Infra 구축

조정환 zozh8835@gmail.com

010-8007-3329



목차

01

프로젝트 개요

02

토폴로지 구성

03

테스트 결과

04

수행절차 및 방법

05

한계점 및 개선 방안

07

느낀점

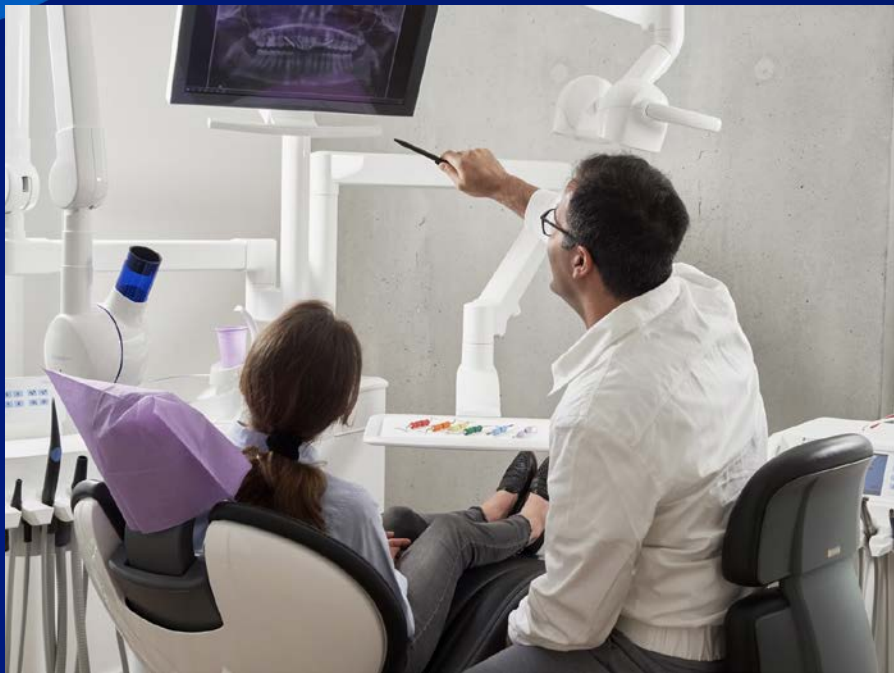




01

프로젝트 개요

1. 고객 소개



소개

- 이름: 홍길동
- 나이: 42세
- 직업: 치과 의사
- 관심사: 최신 치과 기술 및 트렌드, 대형 치과 프렌차이즈 운영

배경

치과를 운영하고 있는 김승수씨는 높은 품질의 의료 서비스를 제공하여 지난 5년간 꾸준히 전국적으로 병원을 확장해왔다. 이로 인해 병원을 찾는 환자들의 수가 증가하였다. 이에 따라, 환자들과의 소통을 강화하고 정보 전달을 더욱 효율적으로 하기 위하여 클라우드에 웹사이트를 만들어 클라우드에 배포하고자 요청서를 넣게 되었다.



2. 요구 사항 분석



01

웹사이트는 매일 약 10,000명의 사용자가 방문할 것으로 예상되는 높은 트래픽을 처리해야 합니다

02

제한된 숫자의 도메인 주소를 사용하여 여러 영강치과 웹사이트에 접속할 수 있어야 합니다

03

HIWARE Solution을 통해 보안접근을 할 수 있어야 합니다



3. 해결 방안



Load Balancing

로드 밸런서 서비스를 사용하여 서버 간 트래픽을 분산함으로써 높은 트래픽으로 인해 발생하는 문제를 방지했습니다

Multi-Cloud & Subdomain

멀티 클라우드를 사용하여 각 웹사이트를 개별적으로 배포하였으며 서브도메인을 사용해 1개의 도메인으로 여러 웹사이트에 접속할 수 있도록 구축했습니다

Security Control

NET엔드의 HIWARE 솔루션을 이용하여 보안접근이 가능하도록 하였으며 모든 작업은 Bastion을 통해서만 작업이 가능하도록 구축했습니다



4. 프로젝트 일정



Sprint 1 (03.12 - 03.13)

- 사용자 스토리 및 프로젝트 요구 사항 이해 및 분석
- 프로젝트 범위 정리
- Sprint 1 준비
- 각 클라우드 서비스 자료조사



Sprint 1 (03.14 - 03.24)

- 클라우드 아키텍처 설계
- 인프라 구축 및 테스트
- 웹사이트 디자인 구성



4. 프로젝트 일정



Sprint 4 (03.25 - 03.27)

- 웹사이트 디자인 마무리
- 네트워크 토폴로지 마무리
- 클라우드 아키텍처 수정 및 마무리
- 최종 테스트 및 전체 아키텍처 검토

마무리 (03.28 - 03.29)

- 프로젝트 문서화
- 프로젝트 프레젠테이션 준비
- 프로젝트 데모 녹화





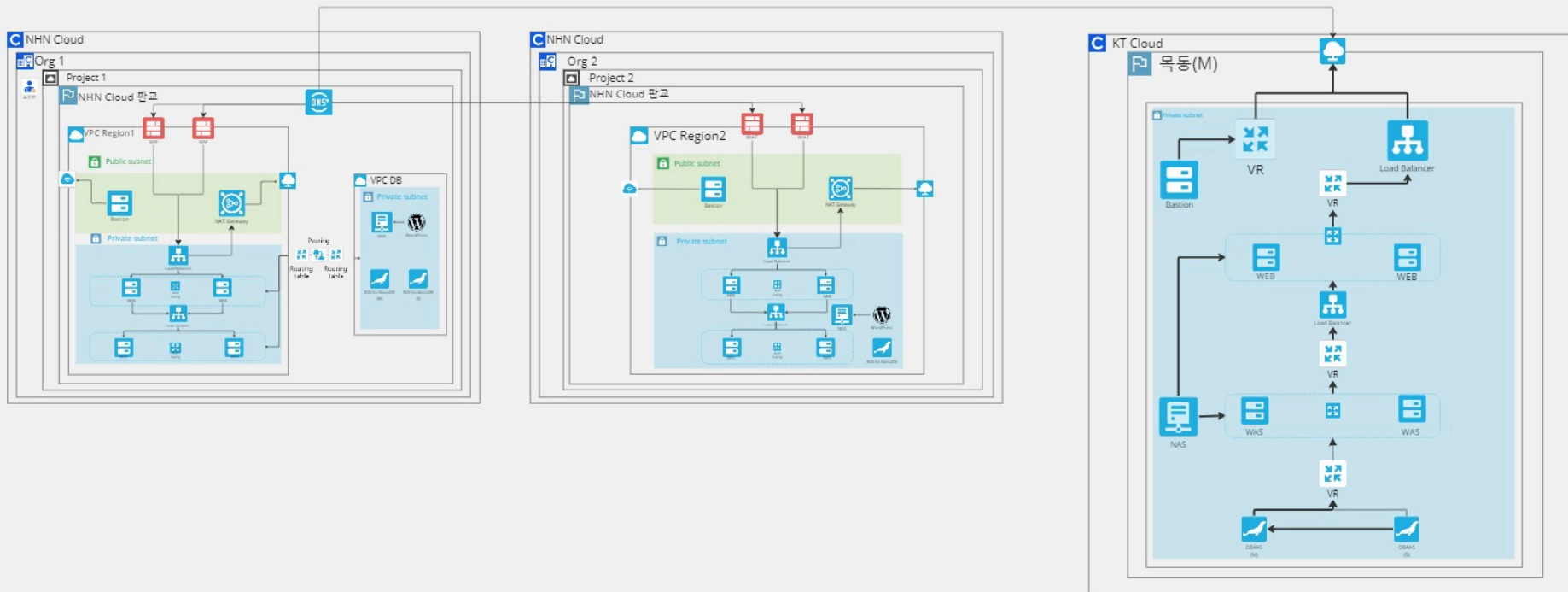
02

Topology

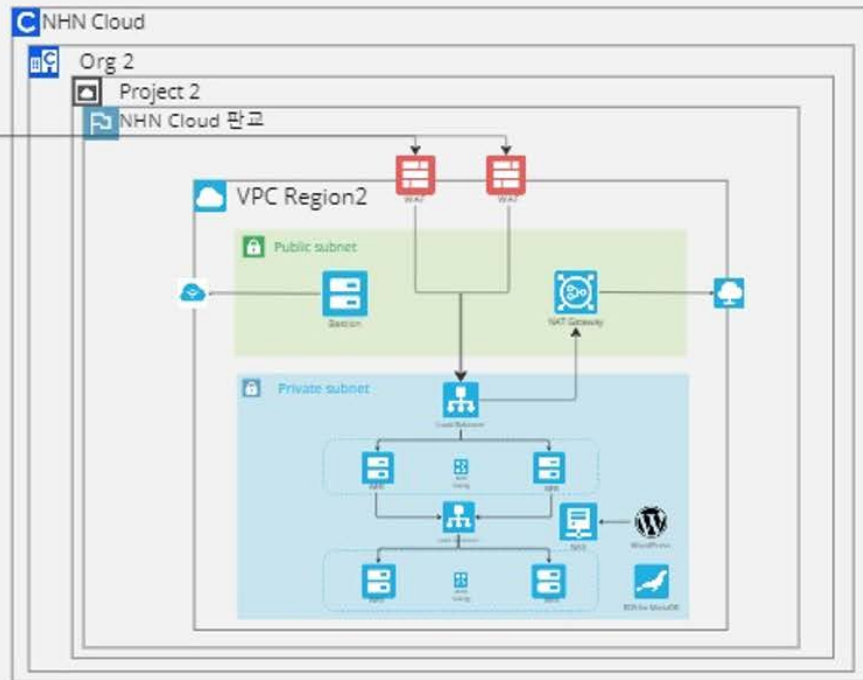
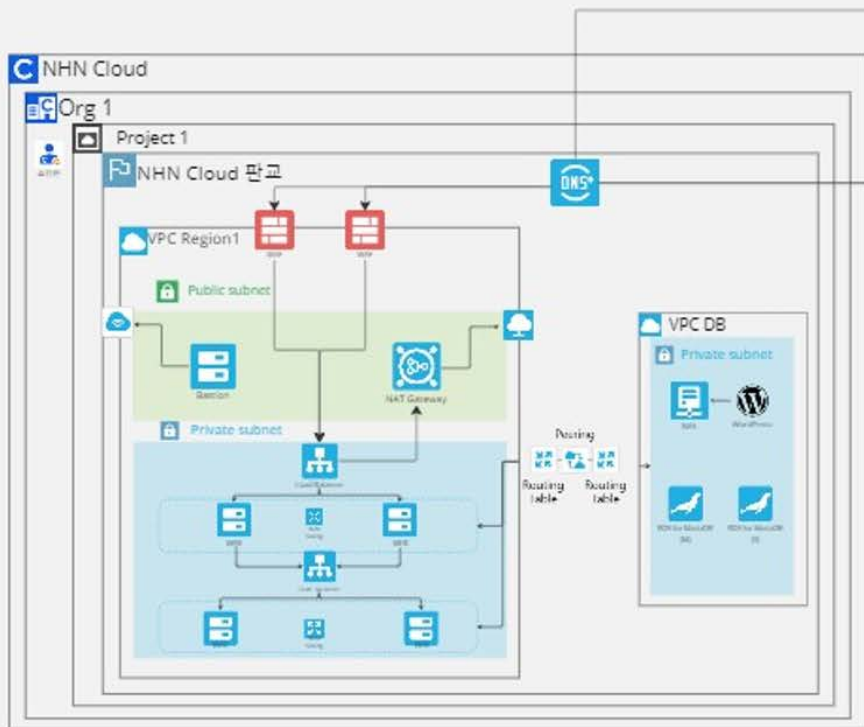


2-1. Topology

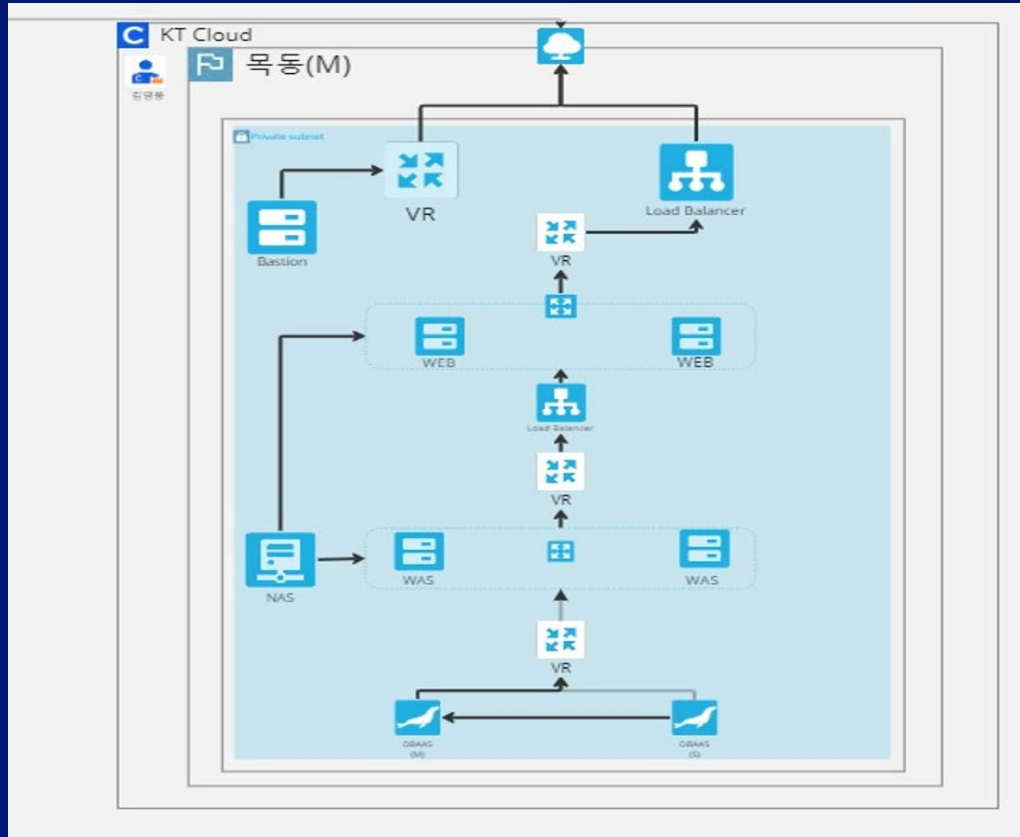
ND



2-2. NHN Topology



2-3. KT Topology



2-4. Network

NHN Region 판교				KT 서울 M
Project 1			Project 2	공인IP (VPC) 211.253.29.222
	VPC - Region1 172.16.0.0/16	VPC - DB 172.17.0.0/16	VPC - Region2 172.18.0.0/16	
Public	172.16.10.0/24	-	172.18.10.0/24	172.23.0.0/16
Private	WEB - 172.16.20.0/24 WAS - 172.16.30.0/24	172.17.10.0/24	WEB - 172.18.20.0/24 WAS - 172.18.30.0/24 NFS - 172.18.40.0/24	10.28.3.160/27
Bastion	172.16.10.18	-	172.18.10.97	172.27.0.123
WAF 1	172.16.10.82	-	172.18.10.27	-
WAF 2	172.16.10.46	-	172.18.10.18	-
WEB LB	172.16.20.101	-	172.18.20.98	211.253.14.77
WAS LB	172.16.30.81	-	172.18.30.91	
NAS	-	172.17.10.80	172.18.40.68	10.28.100.6
DB	-	RDS	RDS	M - 10.28.3.173 S - 10.28.3.168

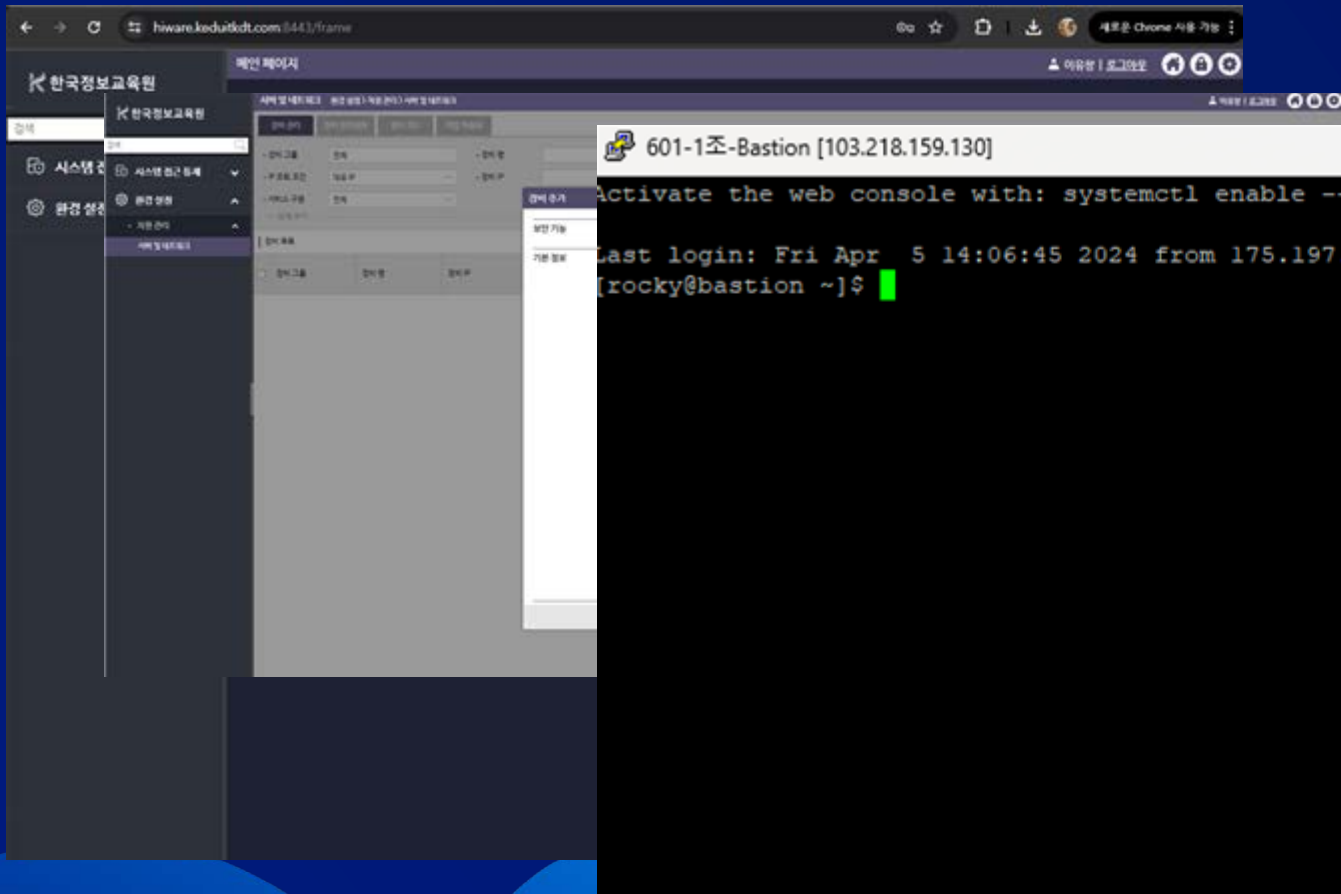




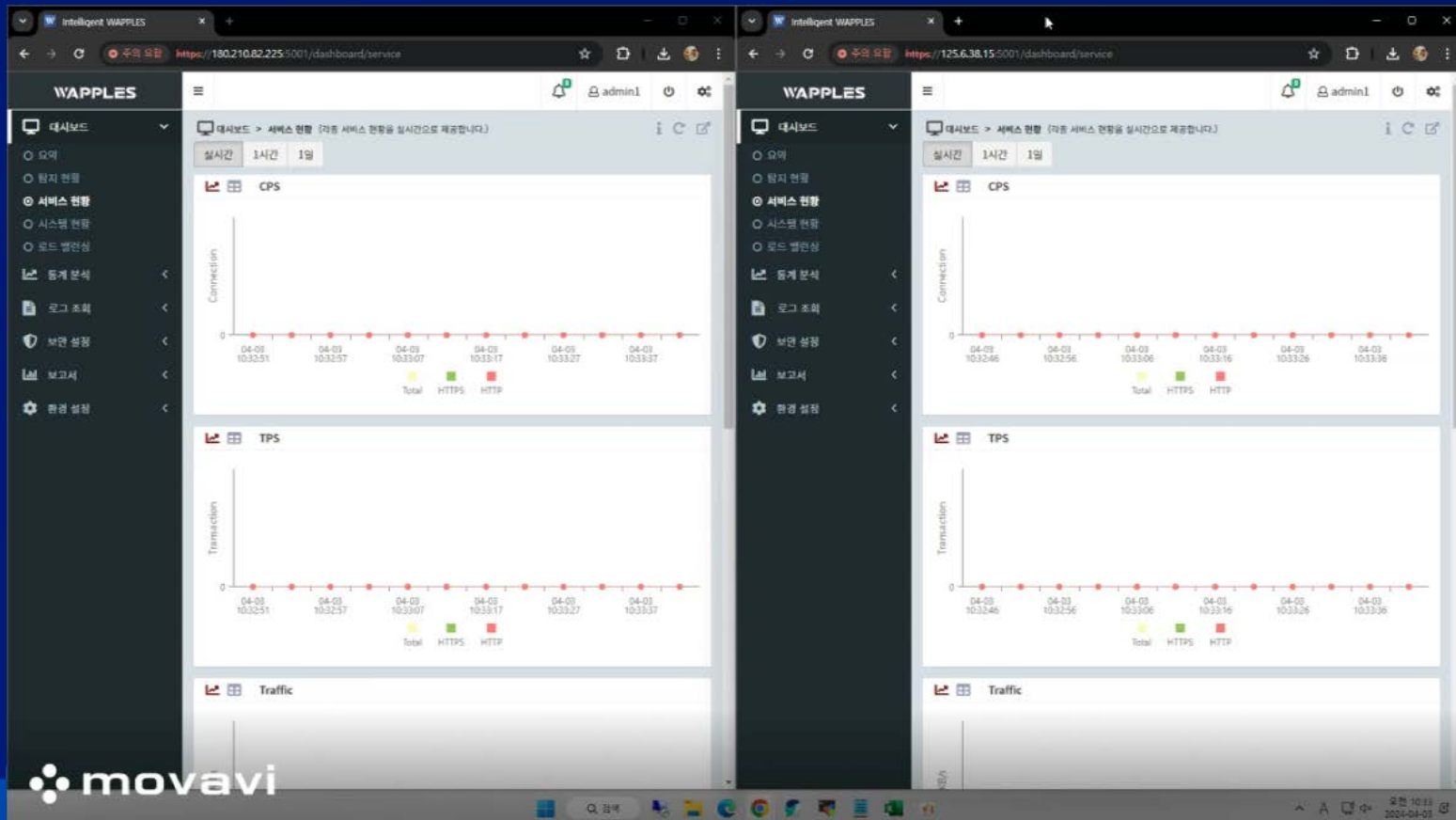
03

테스트 결과

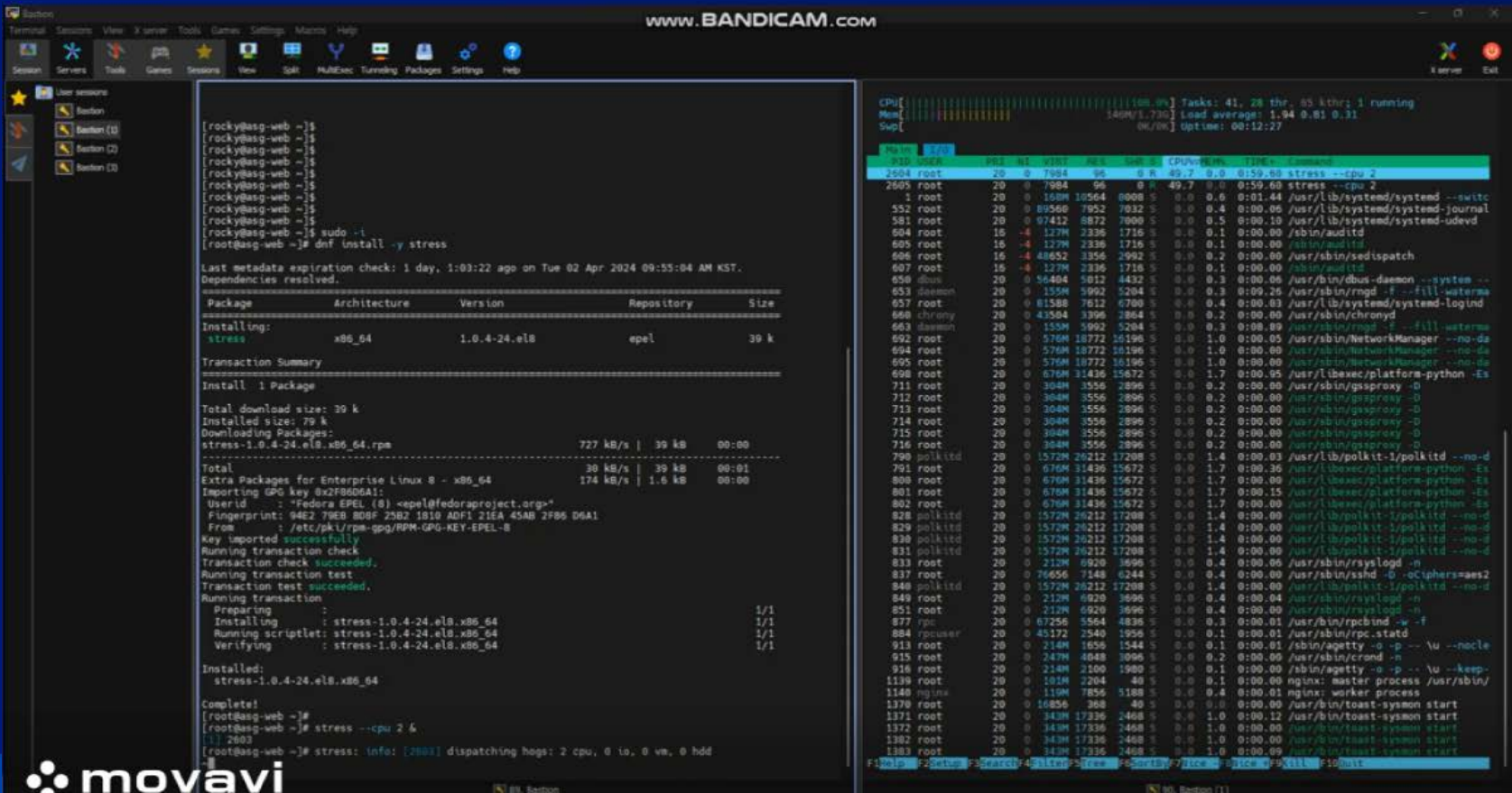
3-1. HIWARE 솔루션 접근



3-2. WAF 테스트 & WEB Auto-Scaling



3-3. WAS Auto-Scaling 테스트



3-4. KT WEB, WAS Auto-Scaling 테스트

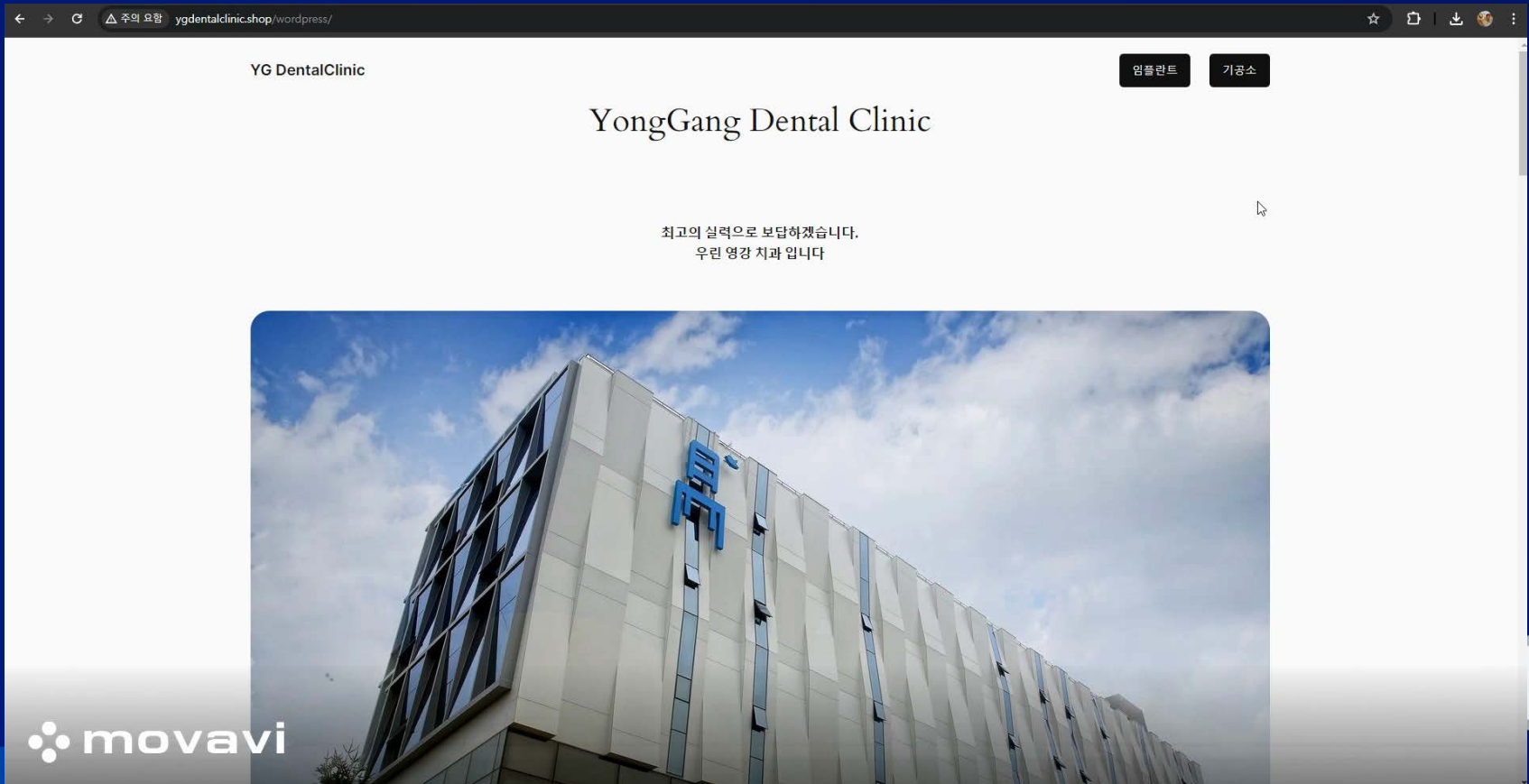
```
top - 11:35:57 up 1 day, 1:49, 2 users, load average: 2.44, 1.78, 2.53
Tasks: 125 total, 3 running, 122 sleeping, 0 stopped, 0 zombie
%Cpu(s): 98.7 us, 0.0 sy, 0.0 ni, 0.0 id, 0.0 wa, 1.0 hi, 0.0 si, 0.3 st
MiB Mem : 751.1 total, 69.1 free, 388.5 used, 293.4 buff/cache
MiB Swap: 2048.0 total, 1663.7 free, 384.2 used, 233.5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
75176	root	20	0	7984	96	0	R	33.2	0.0	0:20.88	stress
75175	root	20	0	7984	96	0	R	32.9	0.0	0:20.87	stress
69332	nginx	20	0	717900	264960	0	S	32.6	34.5	101:32.33	kdevtmpfsi
986	root	20	0	482492	7840	1668	S	0.3	1.0	2:15.05	tuned
1	root	20	0	238500	5456	3296	S	0.0	0.7	0:09.16	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.02	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
7	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H+
9	root	20	0	0	0	0	I	0.0	0.0	0:00.46	kworker/u30:0+
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_rud+
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_tra+
13	root	20	0	0	0	0	S	0.0	0.0	0:00.31	ksoftirqd/0
14	root	20	0	0	0	0	I	0.0	0.0	0:01.16	rcu_sched
15	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
16	root	rt	0	0	0	0	S	0.0	0.0	0:00.14	watchdog/0
17	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
20	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
21	root	20	0	0	0	0	S	0.0	0.0	0:00.02	kauditd
22	root	20	0	0	0	0	S	0.0	0.0	0:00.28	xenbus
23	root	20	0	0	0	0	S	0.0	0.0	0:00.00	xenwatch
24	root	20	0	0	0	0	S	0.0	0.0	0:00.01	khungtaskd
25	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper
26	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	writeback
27	root	20	0	0	0	0	S	0.0	0.0	0:00.02	kcompactd0
28	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
29	root	39	19	0	0	0	S	0.0	0.0	0:01.52	khugepaged
30	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	crypto
31	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kintegrityd
32	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kblockd
33	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	blkcg_punt_bio
34	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	tpm_dev_wq
35	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	md
36	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	edac-poller

```
top - 11:35:59 up 1 day, 1:51, 2 users, load average: 1.71, 0.55, 0.38
Tasks: 120 total, 3 running, 117 sleeping, 0 stopped, 0 zombie
%Cpu(s): 99.0 us, 0.3 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.7 hi, 0.0 si, 0.0 st
MiB Mem : 751.1 total, 554.6 free, 95.2 used, 101.2 buff/cache
MiB Swap: 2048.0 total, 1958.1 free, 89.9 used, 542.3 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
184615	root	20	0	7984	96	0	R	48.7	0.0	0:34.27	stress
184616	root	20	0	7984	96	0	R	48.7	0.0	0:34.27	stress
655	dbus	20	0	56748	512	0	S	0.3	0.1	1:56.27	dbus-daemon
184721	root	20	0	136828	9364	8132	S	0.3	1.2	0:00.01	sshd
1	root	20	0	238472	4008	3188	S	0.0	0.5	0:23.77	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
7	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H+
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_ru+
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_tr+
13	root	20	0	0	0	0	S	0.0	0.0	0:03.27	ksoftirqd/0
14	root	20	0	0	0	0	I	0.0	0.0	0:04.09	rcu_sched
15	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
16	root	rt	0	0	0	0	S	0.0	0.0	0:00.17	watchdog/0
17	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
20	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
21	root	20	0	0	0	0	S	0.0	0.0	0:02.75	kauditd
22	root	20	0	0	0	0	S	0.0	0.0	0:00.26	xenbus
23	root	20	0	0	0	0	S	0.0	0.0	0:00.00	xenwatch
24	root	20	0	0	0	0	S	0.0	0.0	0:00.01	khungtaskd
25	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper
26	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	writeback
27	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kcompactd0
28	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
29	root	39	19	0	0	0	S	0.0	0.0	0:00.42	khugepaged
30	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	crypto
31	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kintegrityd
32	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kblockd
33	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	blkcg_punt_b+
34	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	tpm_dev_wq
35	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	md
36	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	edac-poller
37	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog

3-5. DNS Plus Domain 테스트

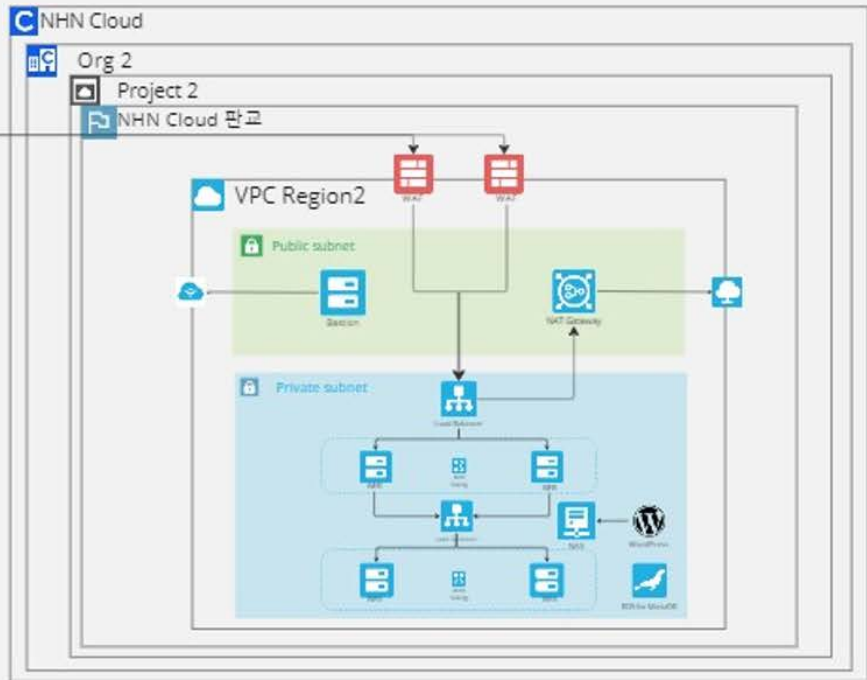
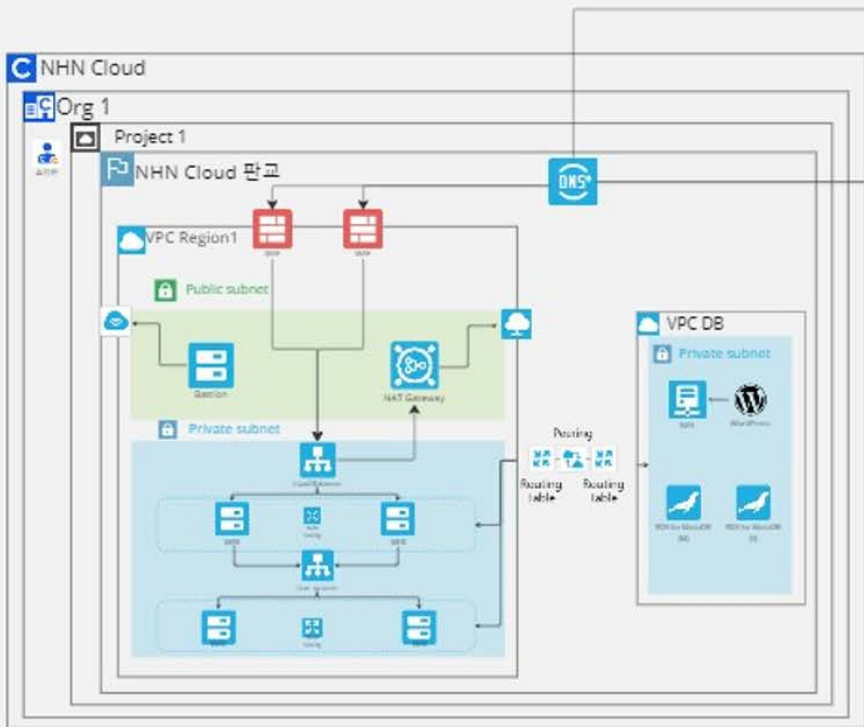




04

수행절차 및 방법

NHN Cloud



NHN DNS Plus & Gabia

네임서버 설정

☐ 전체 가비아 네임서버 사용

ygdentalclinic.shop

네임서버 목록

구분	호스트명	구분	호스트명
1차	ns.toastdns-chichi.com	2차	ns.toastdns-chichi.net

- 네임서버는 IP(숫자)를 제외한 호스트명만 입력합니다. (예. ns.gabia.co.kr)
- 네임서버 값을 복사해서 입력하는 경우, 공란이 포함되지 않도록 주의하시기 바랍니다.
- 각 도메인마다 네임서버 최소/최대 값이 다릅니다.
- 따라서 여러 개의 네임 서버를 입력하여도 도메인의 허용된 개수에 따라 적용됩니다.
- 가비아 네임서버는 DNSSEC를 지원하지 않습니다. DNSSEC이 설정된 도메인을 가비아 네임서버로 변경하시는 경우, 웹사이트 접속(리플링)이 제한됩니다.

+ 추가

설정 방법

1. Gabia 도메인 구매
2. NHN에서 네임 서버를 Gabia에 입력
3. NHN에서 레코드 세트 생성
4. WAF로 IP 주소로 경로 설정

ygdentalclinic.shop.

레코드 세트

+ 레코드 세트 생성 + 레코드 세트 대량 생성 레코드 세트 삭제 레코드 세트 수정 레코드 세트 통계

타입 전체

이름을 입력해 주세요.

검색

새로 고침

<input type="checkbox"/>	레코드 세트 이름	레코드 세트 타입	레코드값	TTL(초)
<input type="checkbox"/>	hong.ygdentalclinic.shop.	A	180.210.82.225 125.6.38.15	60
<input type="checkbox"/>	kt.ygdentalclinic.shop.	A	211.253.14.77	60
<input type="checkbox"/>	www.ygdentalclinic.shop.	A	180.210.82.225 125.6.38.15	60
<input type="checkbox"/>	ygdentalclinic.shop.	NS	ns.toastdns-chichi.com. ns.toastdns-chichi.net.	21600

NHN Peering

Peering Gateway >

피어링 리전 피어링

+ 피어링 생성 피어링



A-B

A-B

기본 정보

피어링

라우트 생성

IP 버전

IPv4

대상 CIDR

! CIDR 형식의 네트워크 주소 (예: 192.168.0.0/16)

게이트웨이

게이트웨이 선택

PEERING: A-B (172.17.10.19)

INSTANCE: ASG-WEB (172.17.10.64)

INSTANCE: ASG-WAS (172.17.10.52)

INSTANCE: ASG-WAS (172.17.10.33)

INSTANCE: ASG-WEB (172.17.10.97)

- 대상 CIDR은 게이트웨이로
- 라우터는 라우팅 테이블당

Appkey

사용자 가이드

새로 고침

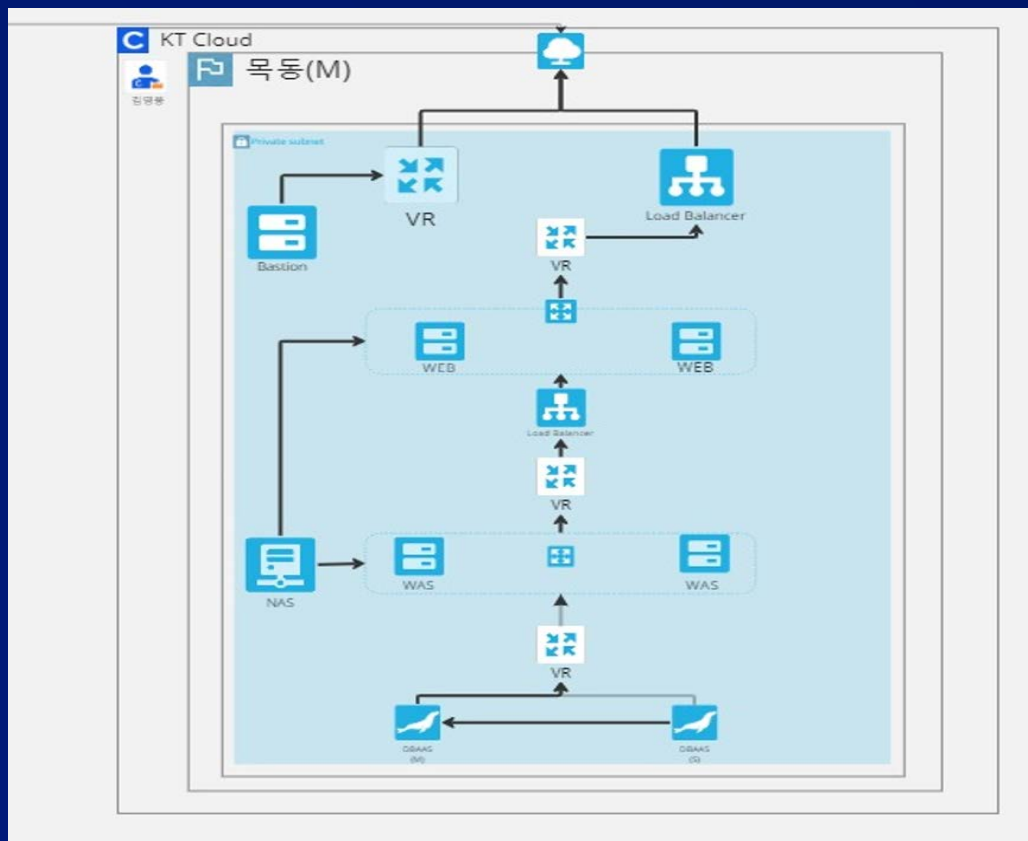
(16)

×

복사

복사

KT Cloud

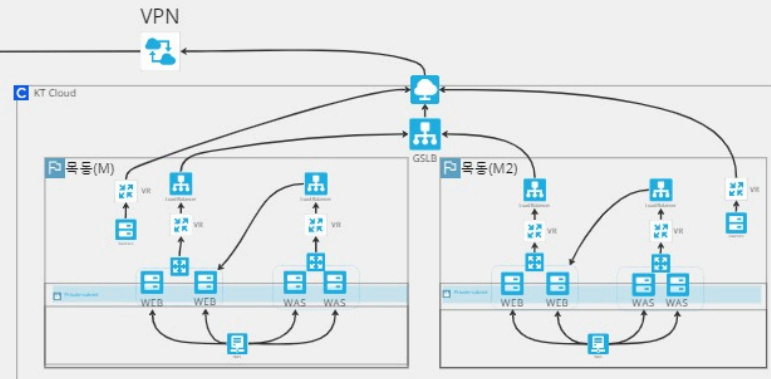
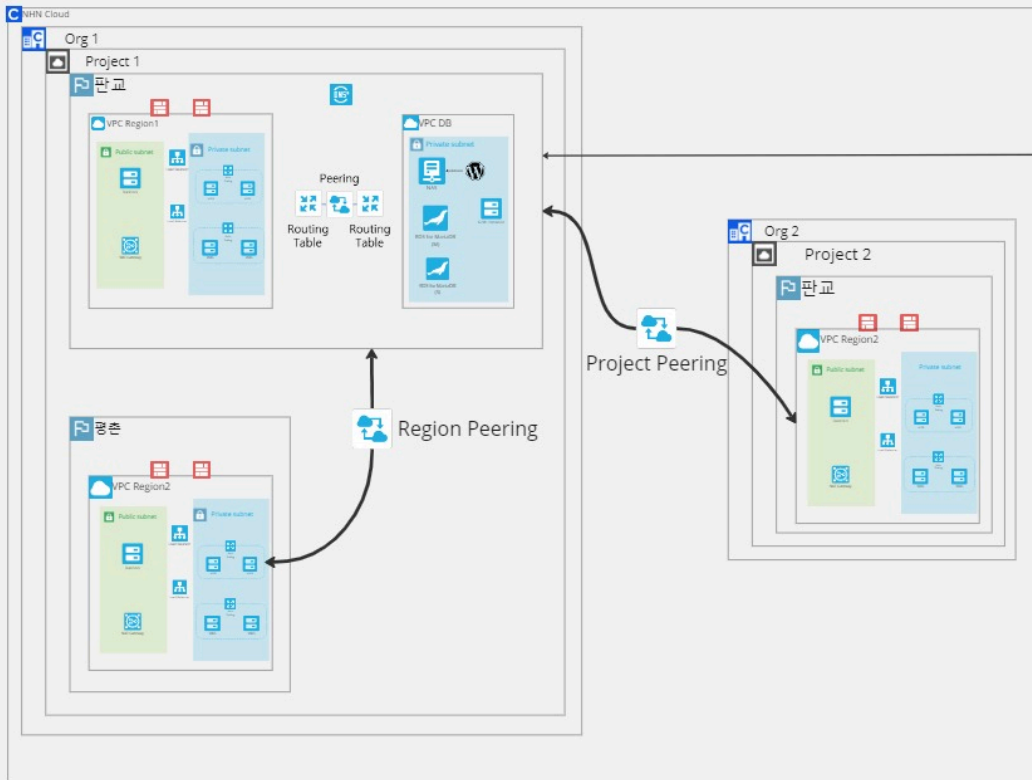




05

한계점 및
개선 방안

5-1. 최초 Topology



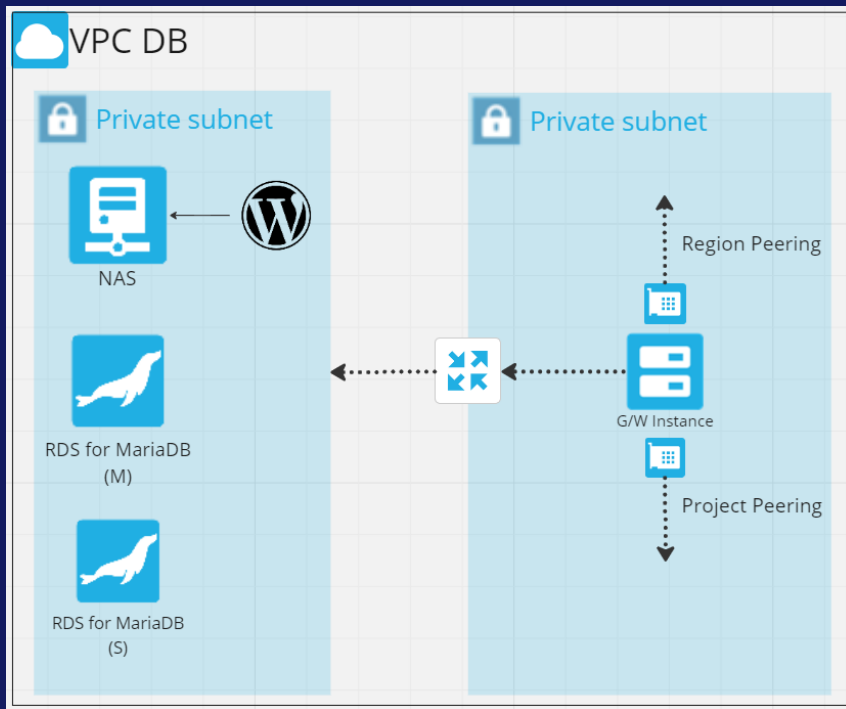
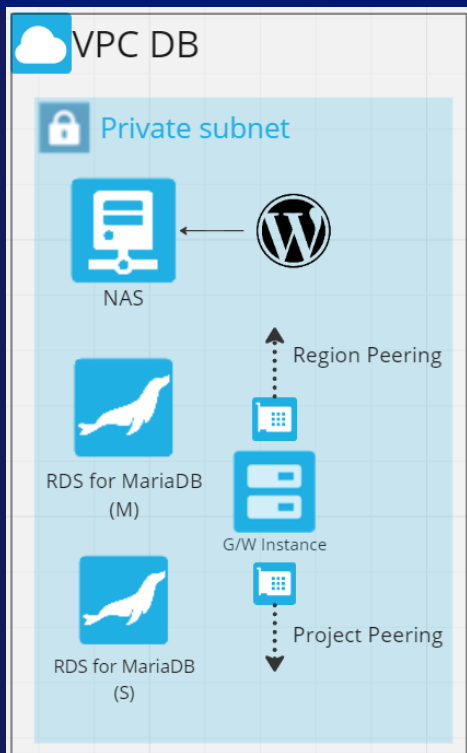
5-2. Peering 실패, Software VPN 실패

한계점

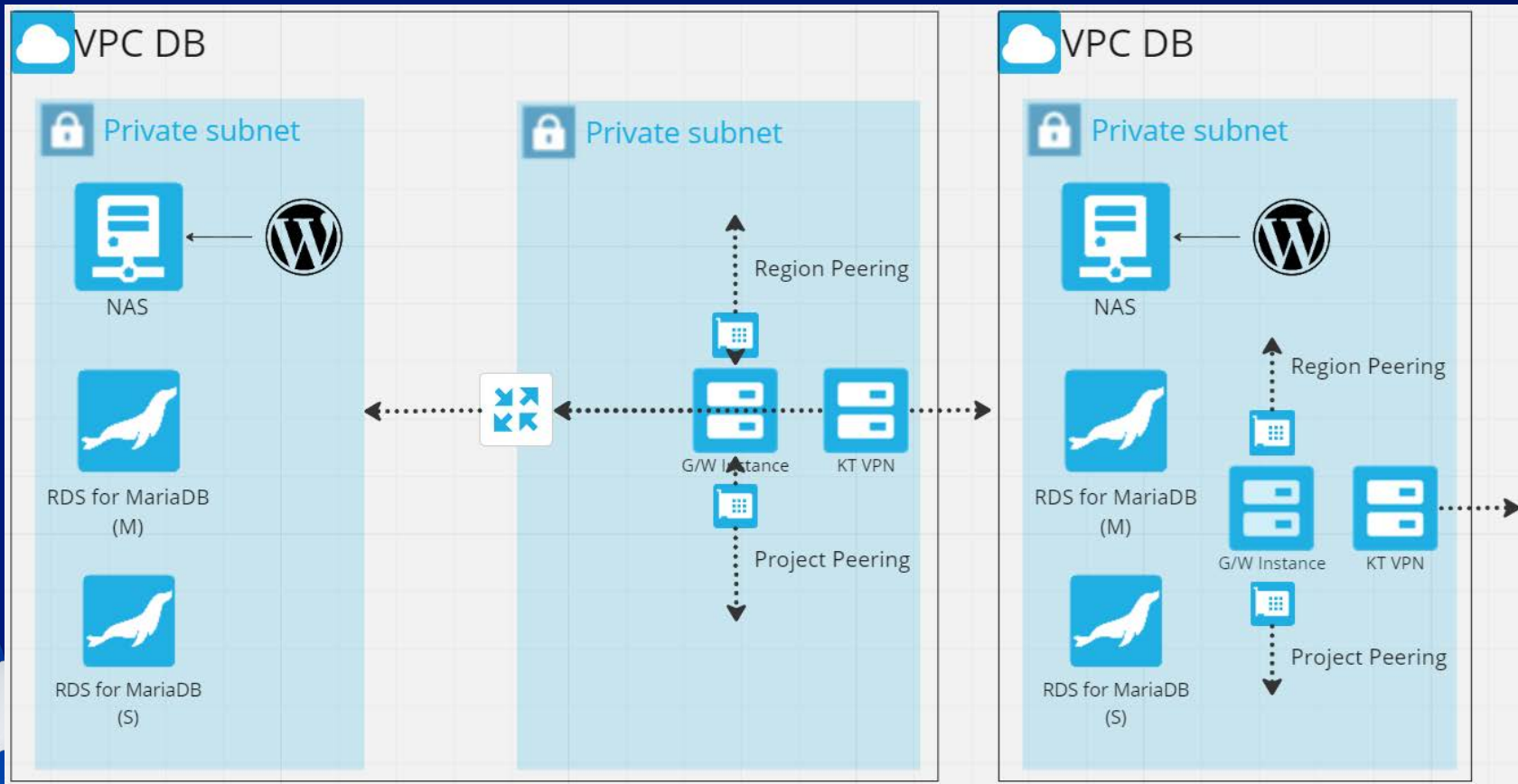
1. Region Peering과 Project Peering의 구조 이해가 부족
2. NHN 네트워크 구조에 대한 이해 부족
3. Routing Table 설정 이해 부족
4. VPN 설치 실패



5-3. Peering 실패



5-4. VPN 실패



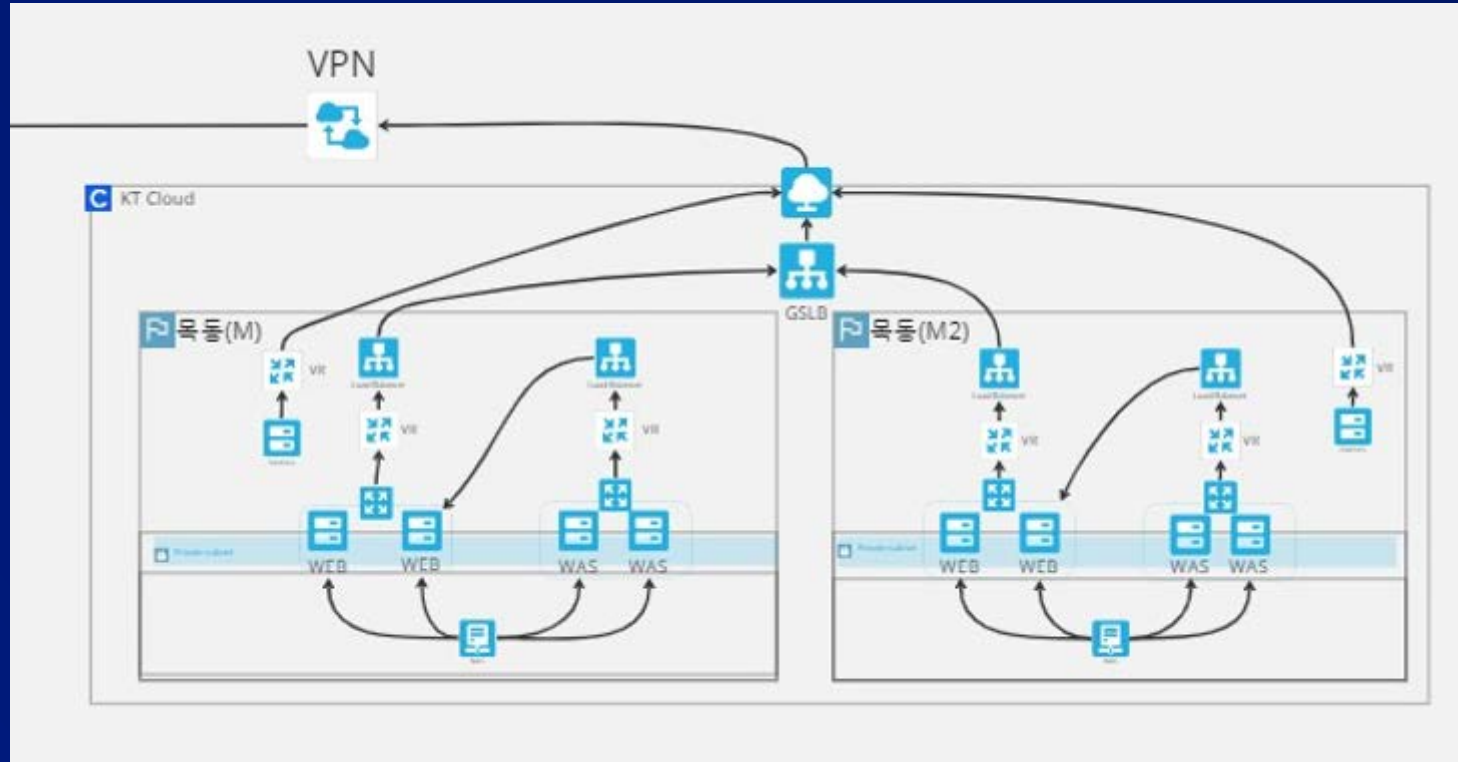
5-5. 개선 방안

개선 방안

1. 물리적 장비를 이용해서 VPN을 구성하여 멀티 클라우드 구현
2. 외부 엔지니어 분들과 NHN 고객센터와의 지속적인 연락으로 3차 프로젝트때 구현



5-6최초 KT Topology



5-7 KT Cloud 한계점

한계점

- 1. VPN 구축 실패로 DB 공유 실패
- 1. GSLB 서비스 문제로 인한 DR 구조 구현 실패
- 1. WAF 서비스 문제로 인한 WEB Firewall 구현 실패



5-8개선 방안

개선 방안

1. 추후 서비스가 정상적으로 작동하면 구현하여 기존에 하지 못했던 작업을 마무리 하겠습니다
2. 외부 엔지니어 분들과 KT 고객센터와의 지속적인 연락으로 3차 프로젝트때 구현



That's all folks!
Thank you

