

하이브리드 클라우드 보안구축

Microsoft Cybersecurity School

Team3

기준서 이승민 이현범 임창현

2025.03.28

목차

I. 프로젝트 개요	2
1. 선정배경	
2. 프로젝트 요구사항	
3. 기술배경	
4. 개발일정	
5. 기술스택	
6. 팀원구성	
II. 프로젝트 설계	6
III. DB 서버 구축	7
1. Docker 이중화 구축	
2. DB 구축 스크립트	
3. cAdvisor 모니터링	
IV. BLUEMAX NGF 100 방화벽 구축.....	12
1. 인터페이스 설정	
2. NAT 및 방화벽정책 설정	
3. VPN 설정	
V. Azure Cloud 구축	15
1. Vnet 구성	
2. Subnet 구성	
3. 공인 IP 할당	
4. 네트워크 보안그룹 적용	
5. NAT 게이트웨이 적용	
6. SSH 키 및 워드프레스 생성 시작스크립트	
7. 가상머신 생성	
8. 어플리케이션 게이트웨이 적용	
9. 웹서비스에 DNS 적용	
10. VPN 연결	
11. ELK Stack	
12. 테라폼 자동화코드	
VI. 향후 개선방향	36
1. 방화벽 정책설정	
2. ELK 구현 개선	
3. VMSS 적용 검토	
VII.참고문헌	37

I. 프로젝트 개요

1. 선정배경

어플리케이션을 배포할 회사가 비용 절감과 성능 최적화를 위해 하이브리드 클라우드를 도입할 수 있습니다. 온프레미스 장비와 퍼블릭 클라우드를 모두 활용하는 하이브리드 클라우드는 다음과 같은 장점이 있습니다.

- ① 개인정보, 금융 데이터와 같은 민감정보는 법적/규제적 이유로 온프레미스에 보관해야 하는데, 하이브리드 환경을 통해 클라우드의 확장성을 활용하면서도 이를 보호할 수 있습니다.
- ② 데이터베이스를 온프레미스에 두면 레거시 자원 낭비를 줄이고, 트래픽 비용 절감과 내부 네트워크를 통한 빠른 접근으로 성능을 높일 수 있습니다.
- ③ 웹 서버를 Azure 에 두고 Azure Front Door, CDN, Application Gateway 를 활용하면 글로벌 사용자에게 빠르고 안정적인 서비스를 제공할 수 있습니다.
- ④ 클라우드와 온프레미스를 연계하여 재해복구 환경을 구축할 수 있으며, 온프레미스 DB 를 Azure Storage 에 백업하고 클라우드 DB 를 대기 서버로 활용할 수 있습니다.

2. 프로젝트 요구사항

- ① 온프레미스에 DB 서버를, Azure 에 WEB 서버를 구축하여 VPN 으로 연결합니다.
- ② 온프레미스 방화벽으로 NAT 를 설정합니다.
- ③ 보안시스템(방화벽)의 관리자 접속(web console)은 내부망에서만 접속 가능합니다.
- ④ DB 는 Active-Standby 이중화 설정합니다.
- ⑤ (SNAT) 내부 직원 end-point PC 의 다수 사설 IP 는 하나의 공인 IP 를 통해 외부로 접속합니다.
- ⑥ (DNAT) 클라우드 web 서버는 하나의 공인 ip 를 통해 어디에서든 접속 가능하도록 합니다.
- ⑦ 온프레미스 방화벽으로 Packet Filtering 정책을 설정합니다.
- ⑧ Docker 는 cAdvisor 로 모니터링하고, DB 는 Elastic Search 로 모니터링합니다.

3. 기술배경

① Docker

Docker 는 2013 년 Solomon Hykes 가 개발하여 오픈소스로 공개한 컨테이너 기술로, 어플리케이션을 더 쉽게 배포하고 실행할 수 있도록 설계되었습니다. Docker Inc.가 관리하며 현재는 CNCF(Cloud Native Computing Foundation) 생태계의 핵심 기술 중 하나입니다.

기존 가상머신보다 가볍고 빠르며, 리소스 효율성이 높은 컨테이너 기반 가상화 기술을 제공합니다. 이미지 기반 배포, 네트워크 격리, 볼륨 관리, 보안 기능을 통해 개발부터 운영까지 일관된 환경을 유지할 수 있습니다.

DevOps, CI/CD 파이프라인, 마이크로서비스 아키텍처, 클라우드 네이티브 어플리케이션 개발에 활발히 사용되며 Windows, macOS, Linux 등 다양한 OS 에 독립적으로 실행 가능합니다. 또한 AWS, Azure, GCP 등의 멀티클라우드 환경에서도 의존성 충돌없이 배포할 수 있도록 지원합니다. 그리고 Kubernetes 와 함께 컨테이너 오케스트레이션을 활용하여 자동 복구, 로드밸런싱, 확장성을 높일 수 있습니다.

2025 년 기준 1 억개 이상의 Docker 이미지가 Docker Hub 에서 제공되고 있습니다. 글로벌 IT 기업과 스타트업은 Docker 를 DevOps 와 클라우드 네이티브 환경에서 필수로 활용하고 있습니다.

② 방화벽

방화벽은 정보통신망을 이용하여 허가되지 않은 접근을 차단하고, 기업의 중요 정보가 인터넷 외부로 유출되는 것을 막는 네트워크 보안 시스템입니다. 신뢰할 수 없는 외부 네트워크와 내부 네트워크 간의 관문에 역할을 하며, 모든 네트워크 트래픽을 감시하고, 특정 규칙에 따라 차단 또는 허용합니다.

이를 위해 패킷 필터링을 통해 IP 주소와 포트를 기반으로 접근을 통제하거나, 도메인 및 웹서비스별로 접근 규칙을 정의할 수 있습니다. 또한 NAT 변환을 활용하여 내부 네트워크가 직접 노출되지 않도록 보호합니다.

방화벽은 하드웨어 또는 소프트웨어 형태로 구현될 수 있으며, 일반적으로 네트워크 방화벽(전용 장비)와 소프트웨어 방화벽(OS 내장)이 함께 사용됩니다. 최신 방화벽 기술은 차세대 방화벽(NGFW)으로 발전하여 IDS, IPS 기능을 포함하고 있으며 클라우드에서도 활용되고 있습니다.

최근에는 제로 트러스트 보안 모델이 도입되면서 방화벽은 단순한 네트워크 경계 보호를 넘어 사용자 및 디바이스 인증과 세분화된 접근 제어 기능을 수행하고 있습니다. 기업 환경에서는 SD-WAN(Security-Defined WAN)과 결합하여 지능형 위협 방어 및 원격 근무 보안을 강화하는 방향으로 발전하고 있습니다.

③ VPN

하이브리드 클라우드는 온프레미스 데이터센터와 퍼블릭 클라우드를 함께 운영하는 환경으로 두 환경 간 안전한 데이터 전송이 필수적입니다. 이를 위해 VPN(Virtual Private Network)를 활용하면 인터넷을 통한 데이터 전송 시 보안을 강화하고 안정적인 연결을 유지할 수 있습니다.

VPN은 공용 네트워크를 사용하지만, 사설 네트워크처럼 암호화된 통신 터널을 형성하여 데이터를 안전하게 전송하는 기술입니다. 일반적으로 기업 환경에서는 Site-to-Site VPN 방식이 사용되며, 온프레미스 방화벽과 클라우드 간에 보안 터널을 형성하여 내부 네트워크처럼 동작하도록 합니다.

VPN의 핵심 보안 기술 중 하나가 IPsec(IP security)입니다. 네트워크 Layer3에서 데이터 패킷을 암호화하고 인증하여 보안성을 강화하는 프로토콜 모음으로, 데이터 무결성과 기밀성을 보장합니다. IPsec은 ESP(Encapsulating Security Payload)와 AH(Authentication Header) 프로토콜을 활용하여 암호화 및 인증을 수행하며, 네트워크 간 VPN 터널을 안전하게 유지합니다.

VPN을 구축할 때는 암호화 방식, 인증 방식, DH(Diffie-Hellman) 그룹을 일치시키는 것이 필수적입니다. 암호화 방식으로는 AES-256이 널리 사용되며, 인증 방식으로는 Pre-Shared Key(PSK) 또는 디지털 인증서 기반 인증이 적용된다. 또한 VPN 키 교환을 위해 DH 그룹이 사용되는데 이는 보안 강도를 결정하는 요소이며 Azure VPN과 시큐아이 방화벽에서 동일한 DH 그룹으로 설정해야 정상적으로 터널이 형성된다.

최근에는 차세대 방화벽(NGFW)과 함께 활용하여 IDS, IPS와 통합 운영하는 방식으로 발전하고 있으며, 클라우드 기반 보안 모델인 SASE(Secure Access Service Edge)와 연계하여 원격 근무 환경에서도 안전한 접속을 지원하는 방향으로 확장되고 있다.














VPN 방식	설명
Site-to-Site VPN	온프레미스 방화벽과 Azure VPN Gateway 간 IPsec 터널을 형성하여 고정된 네트워크 연결을 제공
Point-to-Site VPN	개별 PC나 모바일 기기에서 Azure VNet으로 직접 연결하는 방식
ExpressRoute	VPN보다 더 빠르고 안정적인 전용회선을 사용하나 비용이 높음

IPsec 기능	ESP	AH
데이터 암호화	지원 (Payload 암호화)	미지원 (무결성만 보장)
무결성 검사	지원 (HMAC 등 사용)	지원
NAT 호환성	호환 가능	호환 불가
지원 모드	전송 모드, 터널 모드	전송 모드, 터널 모드
VPN에서의 활용	기본적으로 사용	잘 사용되지 않음

4. 개발일정

일자	2025-03-24	2025-03-25	2025-03-26	2025-03-27	2025-03-28
요구사항 정의	○				
아키텍처 설계	○	○			
Docker DB 구축		○	○	○	
방화벽 구성		○	○	○	
Azure 구축				○	○
문서화 및 테스트				○	○

5. 기술스택

구분	Tools
Azure Services	 VPN  NAT  APP-GW  DNS Zones
Dev tool	 Terraform 1.11.1  draw.io  Xshell 8
WEB & WAS	 Apache 2.4.63  Wordpress 6.7.2
Database	 MySQL 8.0  Docker
Virtual OS	 RockyLinux 9.3.20231113
Code Editor	 Visual Studio Code 1.98.2

6. 팀원구성

이름	기준서	이승민	이헌범	임창현
역할	하이브리드 설계 구성도 작성	요구사항 분석 방화벽 구축	Docker DB 구축 PPT 작성	Azure 테스트 보고서 작성

II. 프로젝트 설계

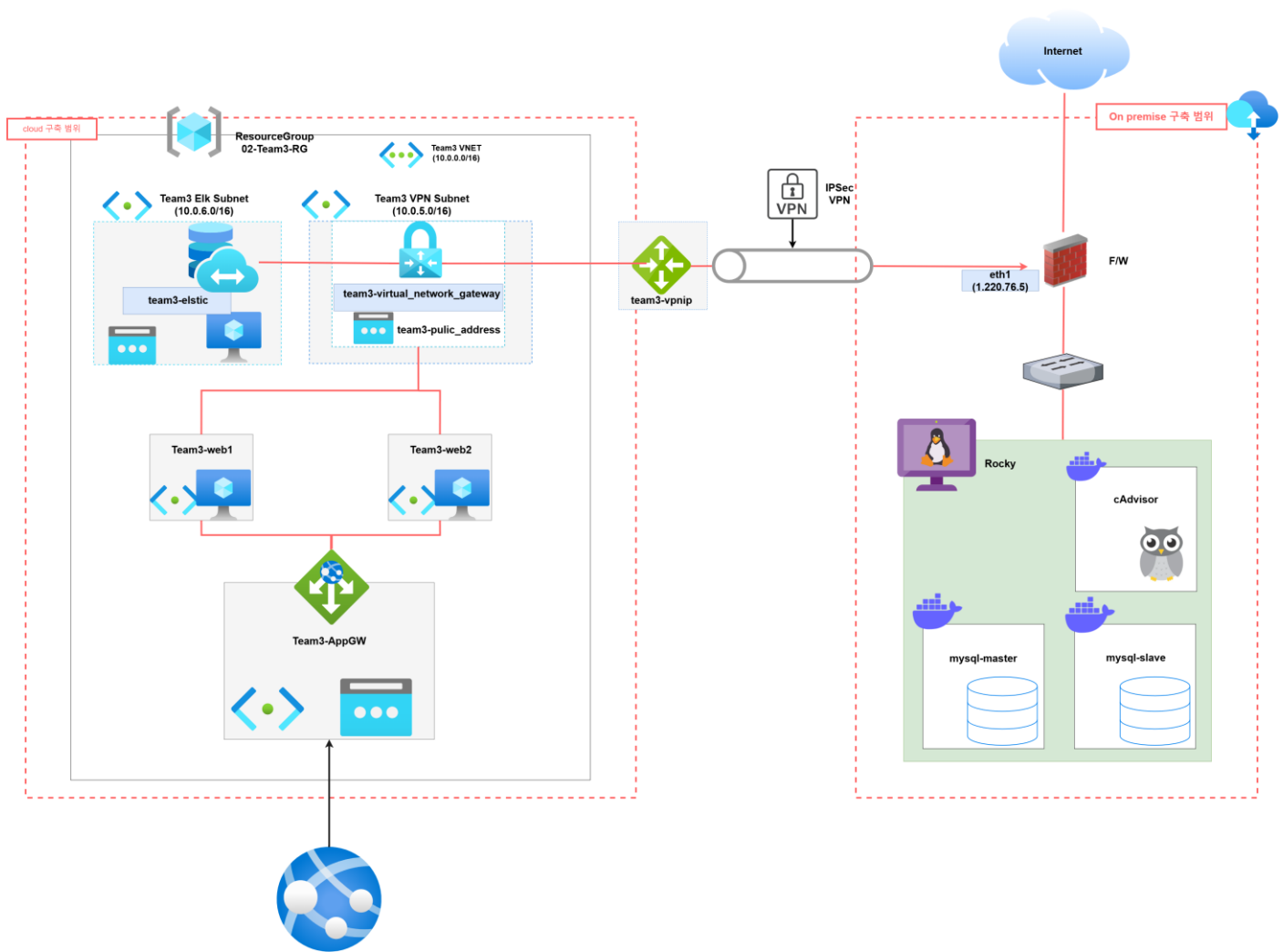


그림 1 하이브리드 클라우드 아키텍처

이 프로젝트는 Azure 클라우드와 온프레미스 환경을 VPN 을 통해 연결하는 하이브리드 클라우드 인프라 구축을 목표로 합니다.

Azure 에는 Team3 가상 네트워크(VNet)가 설정되었으며, 내부에는 Team3 VPN 서브넷과 Team3 ELK 서브넷이 포함됩니다. VPN 연결을 위해 Team3 Virtual Network Gateway 가 생성되었고, 온프레미스 방화벽과의 IPsec VPN 터널을 통해 연결됩니다.

Azure 내부에서는 AppGW(Application Gateway)가 배포되어 트래픽을 관리하며, 내부 애플리케이션 서버(Team3-web2)와 ELK 스택이 운영됩니다.

온프레미스 환경은 Rocky Linux 기반 서버로 구성되며, 해당 서버에는 cAdvisor(컨테이너 모니터링 도구)와 MySQL Master-Slave 데이터베이스가 운영됩니다.

보안 강화를 위해 방화벽(FW)이 VPN 터널을 통해 들어오는 트래픽을 제어하며, VPN 연결 시 암호화 및 인증을 위한 IPsec 프로토콜이 적용되었습니다.

III. DB 서버 구축

1. Docker 이중화 구축

DB 서버를 Docker 로 이중화하면 아래와 같은 장점이 있습니다.

이 시스템은 Docker 기반으로 배포되므로, 컨테이너를 활용해 데이터베이스를 실행하면 빠른 배포와 환경 복원이 가능합니다. 또한, 필요에 따라 추가적인 Sub DB 컨테이너를 배포하여 읽기 부하를 분산시킴으로써 스케일 아웃이 가능합니다.

비용 절감 측면에서도 효과적입니다. 클라우드 DB 서비스를 이용하는 것보다 온프레미스 환경을 활용하면 운영 비용을 줄일 수 있으며, 기존 Host PC 의 자원을 재활용하여 추가적인 인프라 비용 없이 데이터베이스를 운영할 수 있습니다.

관리 편의성 또한 높습니다. 컨테이너 이미지를 관리함으로써 데이터베이스 버전 롤백, 복제, 배포가 용이하며, 컨테이너 기반으로 데이터 마이그레이션이 쉽기 때문에 백업과 복구 작업이 간편합니다.

마지막으로, 네트워크 및 보안 제어가 용이합니다. 민감한 데이터가 외부 클라우드에 저장되지 않아 온프레미스 환경에서 보안이 강화되며, 방화벽과 VPN 설정을 통해 Host PC 에서 직접 트래픽을 제어할 수 있어 보다 안전한 데이터 운영이 가능합니다.

2. DB 구축 스크립트

아래 Bash 스크립트는 MySQL 마스터-슬레이브 복제를 설정하는 과정입니다. 먼저, SELinux 를 비활성화하여 보안 정책이 Docker 와 MySQL 실행에 영향을 주지 않도록 설정합니다. 이후, Docker 저장소를 추가하고 Docker 및 관련 패키지를 설치한 후, 서비스를 활성화합니다.

다음으로, MySQL 8.0 컨테이너를 실행하기 위해 마스터와 슬레이브 설정 파일을 각각 생성하고, 이를 적용한 MySQL 컨테이너를 실행합니다. 이후, 마스터 서버에서 team3 사용자를 생성하고 wordpress 데이터베이스에 대한 모든 권한을 부여합니다. 초기 데이터 동기화를 위해 마스터에서 mysqldump 를 사용하여 데이터를 백업한 후, 이를 슬레이브 컨테이너로 복사하여 복원합니다.

마스터의 바이너리 로그 파일과 위치 정보를 확인한 후, 슬레이브 서버에서 마스터 정보를 설정합니다. 이후, 슬레이브 복제를 시작하고 복제가 정상적으로 작동하는지 확인합니다.

마지막으로, 컨테이너 모니터링을 위해 cAdvisor 를 실행하여 58080 포트에서 접근할 수 있도록 설정합니다.

다음 페이지에 스크립트를 첨부하였습니다.


```
#!/bin/bash
setenforce 0
grubby --update-kernel ALL --args selinux=0
systemctl restart chronyd
sleep 10

dnf install -y dnf-plugins-core
dnf config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
dnf install -y docker-ce docker-ce-cli containerd.io
systemctl enable --now docker
docker pull mysql:8.0

mkdir /master
mkdir /slave
echo -e "[mysqld]\n\
log-bin=mysql-bin\n\
server-id=1" >> /master/master.conf
echo -e "[mysqld]\n\
log-bin=mysql-bin\n\
server-id=2" >> /slave/slave.conf

docker run -d --name mysql-m -p 3306:3306 \
-e MYSQL_ROOT_PASSWORD=It12345! \
-e MYSQL_DATABASE=wordpress \
-v /master/master.conf:/etc/mysql/conf.d/master.conf \
-d mysql:8.0

docker run -d --name mysql-s -p 63306:3306 \
-e MYSQL_ROOT_PASSWORD=It12345! \
-e MYSQL_DATABASE=wordpress \
-v /slave/slave.conf:/etc/mysql/conf.d/slave.conf \
-d mysql:8.0

sleep 20
set +H
docker exec -it mysql-m mysql -uroot -pIt12345! \
-e "CREATE USER 'team3'@'%' IDENTIFIED BY 'It12345!'; GRANT ALL PRIVILEGES ON wordpress.*
TO 'team3'@'%; FLUSH PRIVILEGES;"
docker exec -it mysql-m mysql -uroot -pIt12345! \
-e "alter user 'team3'@'%' identified with mysql_native_password by 'It12345!'; FLUSH
PRIVILEGES;"
docker exec -it mysql-m bash -c \
"echo -e '[mysqld]\nbind-address = 0.0.0.0\n[client]\nuser=root\npassword=It12345!\n' >
/etc/mysql/conf.d/rootaccess.cnf"
docker exec -it mysql-m mysql -uroot -pIt12345! \
-e "USE wordpress; CREATE TABLE wdtb(no int(10) not null, name varchar(100) not null);
DESC wdtb;"

docker exec -it mysql-m bash -c "mysqldump -uroot -pIt12345! wordpress > word.sql"
docker cp mysql-m:word.sql .
```

```

docker cp word.sql mysql-s:.
docker exec -it mysql-s bash -c "mysql -uroot -pIt12345! wordpress < word.sql"

MASTER_CONTAINER="mysql-m"
SLAVE_CONTAINER="mysql-s"

MYSQL_USER="root"
MYSQL_PASSWORD="It12345!"

REPL_USER="team3"
REPL_PASSWORD="It12345!"

read BINLOG_FILE BINLOG_POS <<< $(docker exec -i $MASTER_CONTAINER \
mysql -u$MYSQL_USER -p$MYSQL_PASSWORD -N -e "SHOW MASTER STATUS;" | awk '{print $1,
$2}')

echo "Master binlog file: $BINLOG_FILE"
echo "Master binlog position: $BINLOG_POS"

docker exec -i $SLAVE_CONTAINER \
mysql -u$MYSQL_USER -p$MYSQL_PASSWORD \
-e "CHANGE MASTER TO \
MASTER_HOST='10.10.5.11', \
MASTER_USER='$REPL_USER', \
MASTER_PASSWORD='$REPL_PASSWORD', \
MASTER_LOG_FILE='$BINLOG_FILE', \
MASTER_LOG_POS=$BINLOG_POS;"

docker exec -i $SLAVE_CONTAINER \
mysql -u$MYSQL_USER -p$MYSQL_PASSWORD \
-e "START SLAVE; SHOW SLAVE STATUS\G"

docker run --volume=/:/rootfs:ro --volume=/var/run:/var/run:rw --volume=/sys:/sys:ro --
volume=/var/lib/docker:/var/lib/docker:ro --publish=58080:8080 --detach=true --name=cad
gcr.io/cadvisor/cadvisor

```

```

mysql> START SLAVE;
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> SHOW SLAVE STATUS\G
1. row *****
      Slave_IO_State: Waiting for source to send event
      Master_Host: 10.10.5.11
      Master_User: root
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: binlog.000002
      Read_Master_Log_Pos: 1898
      Relay_Log_File: 15f6ffa2f3c8-relay-bin.000004
      Relay_Log_Pos: 323
      Relay_Master_Log_File: binlog.000002
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
      Replicate_Do_DB:

```

그림 2 Master-Slave 연결 확인

3. cAdvisor 모니터링

cAdvisor(Container Advisor)는 컨테이너의 리소스 사용량을 모니터링하는 도구로, CPU, 메모리, 디스크 I/O, 네트워크 사용량 등의 성능 데이터를 실시간으로 수집하고 시각화합니다. 이를 통해 컨테이너의 상태를 파악하고 최적화할 수 있으며 Docker 환경에서 효율적인 리소스 관리가 가능합니다.

설치는 Docker 컨테이너로 실행할 수 있습니다. 먼저, docker run 명령어를 사용하여 cAdvisor 컨테이너를 배포하며, --volume 옵션을 활용해 호스트 시스템의 /var/lib/docker/와 /sys/ 디렉터리를 마운트하면 컨테이너 정보를 수집할 수 있습니다. 실행 후 웹 브라우저에서 http://<서버 IP>:58080 에 접속하면 실시간 모니터링이 가능합니다

cAdvisor 는 개별 컨테이너뿐만 아니라 전체 시스템 리소스 사용량을 분석하는 기능도 제공하며, Prometheus, InfluxDB 등과 연동해 장기적인 성능 데이터를 저장하고 분석할 수도 있습니다. 이를 활용하면 컨테이너 환경에서 성능 병목을 사전에 감지하고, 적절한 리소스 할당 및 조정을 통해 안정적인 운영을 유지할 수 있습니다.

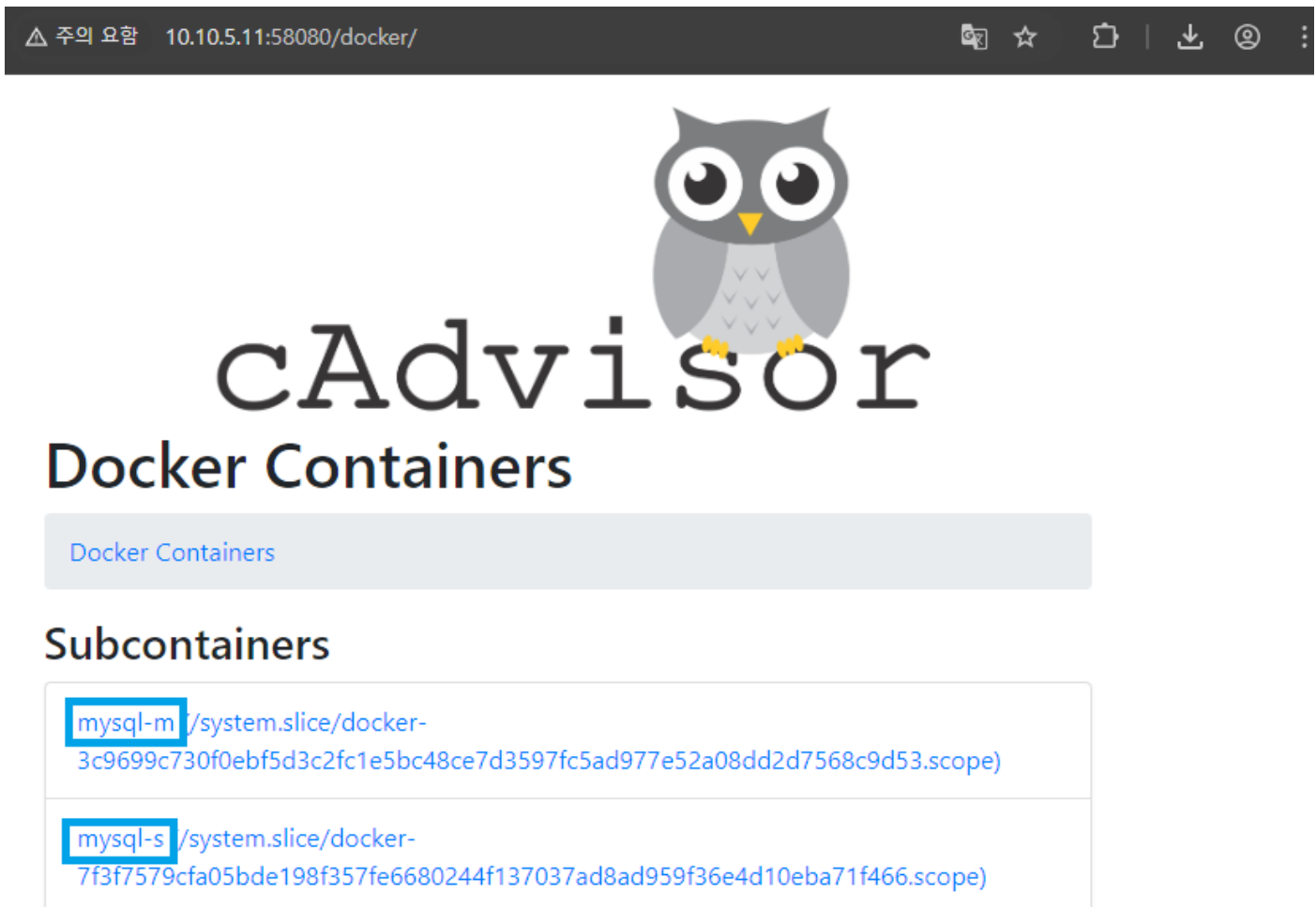


그림 3 cAdvisor 웹브라우저 접속화면

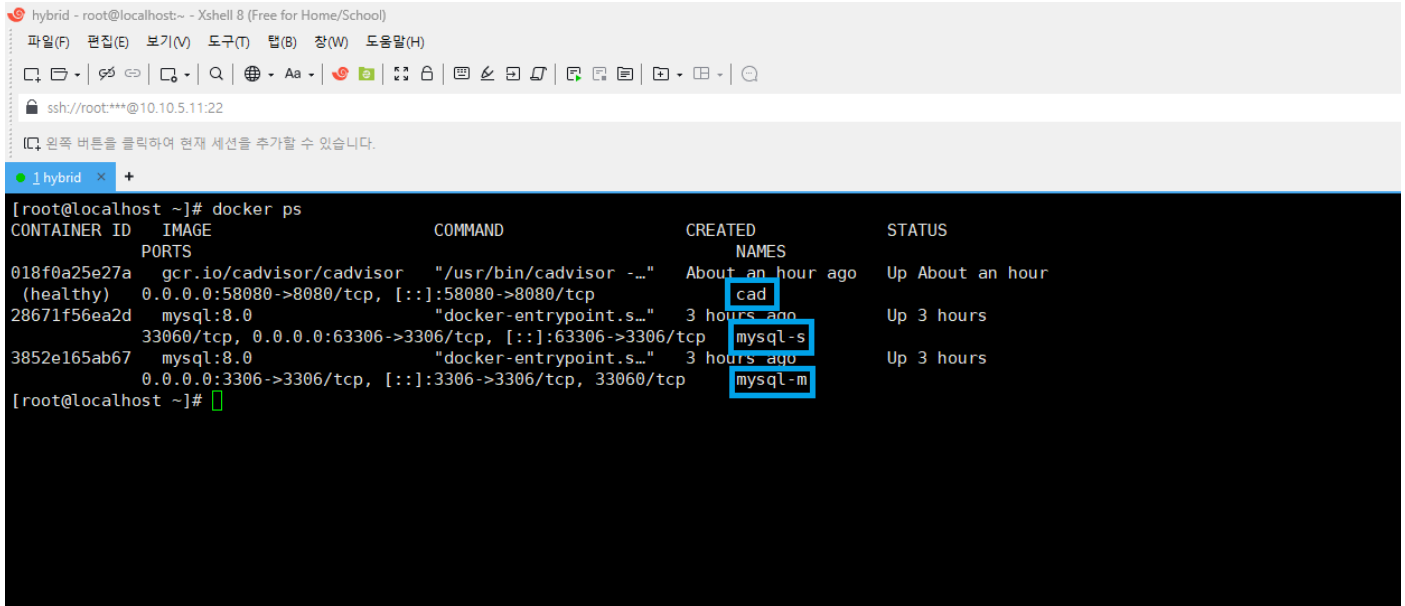


그림 4 cAdvisor 와 Docker DB 의 연결설정 확인

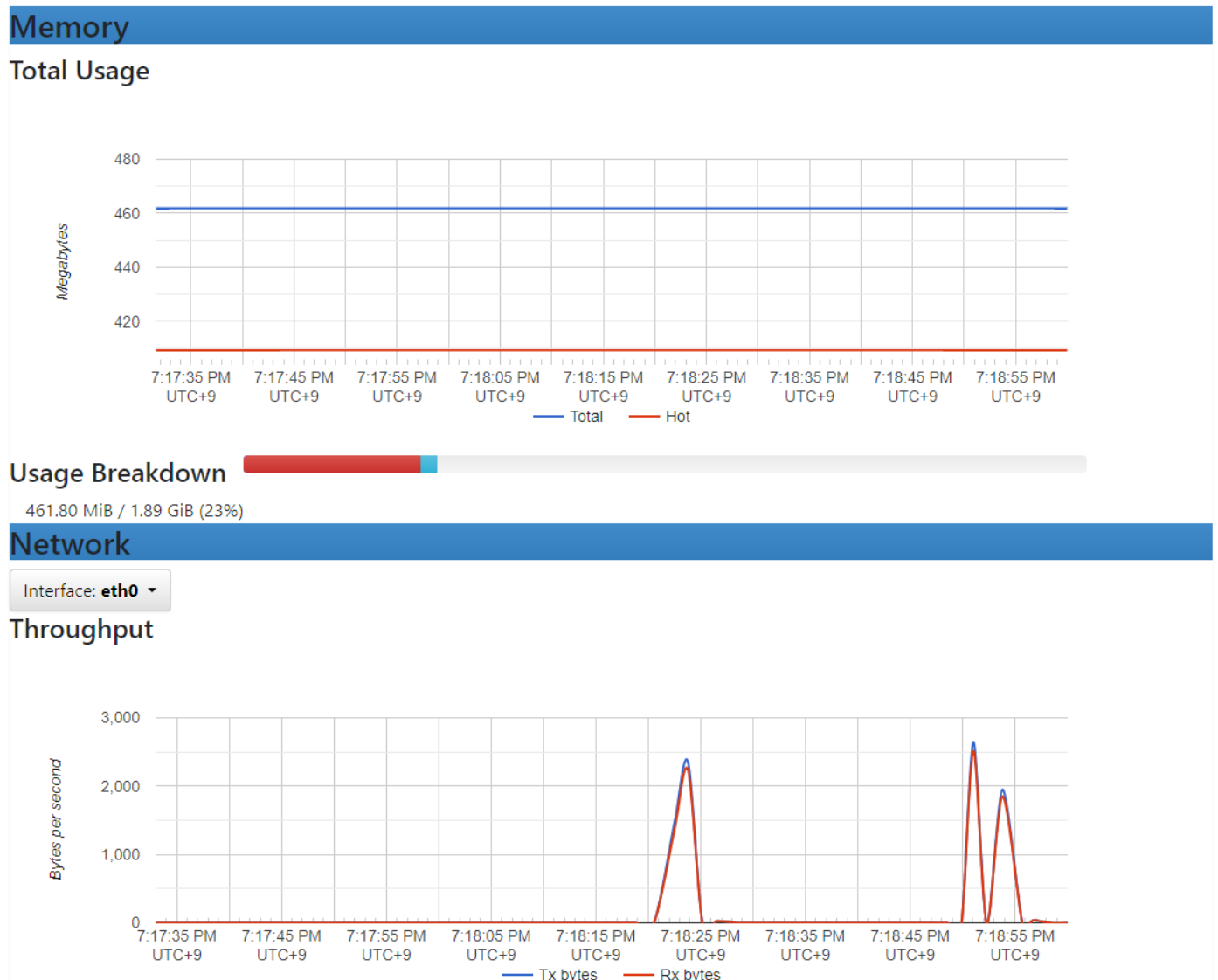


그림 5 cAdvisor Memory Network 모니터링

IV. BLUEMAX NGF 100 방화벽 구축

1. 인터페이스 설정

방화벽 eth1 에 외부 공인 IP 를, 스위치포트인 eth4 에 내부 사설 IP 를 부여합니다.

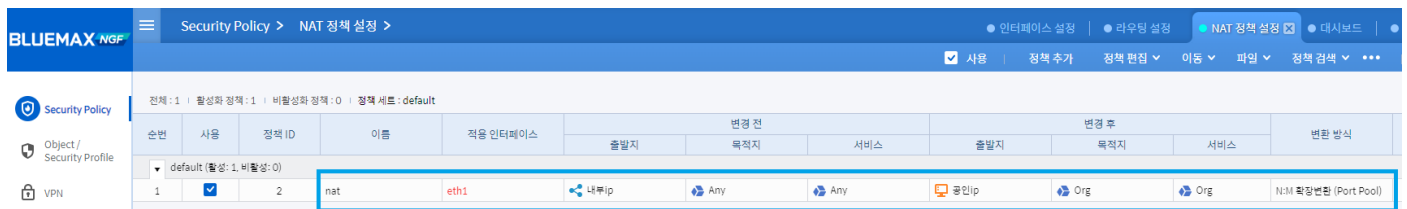


인터페이스 이름	모드	IP 주소	MTU	속도	Duplex	Auto-Negotiation	Up/Down	링크 연결	Zone
eth0	L3	192.168.10.10/24	1500			on	UP	no	내부망
eth1	L3	1.220.76.5/29	1500	100M	Full Duplex	on	UP	yes	외부망
eth2	L3		1500			on	UP	no	
eth3	L3		1500			on	UP	no	
eth4	L3	10.10.5.254/24	1500	1G	Full Duplex	on	UP	yes	내부망

그림 6 방화벽 NIC 에 IP 부여

2. NAT 및 방화벽 정책 설정

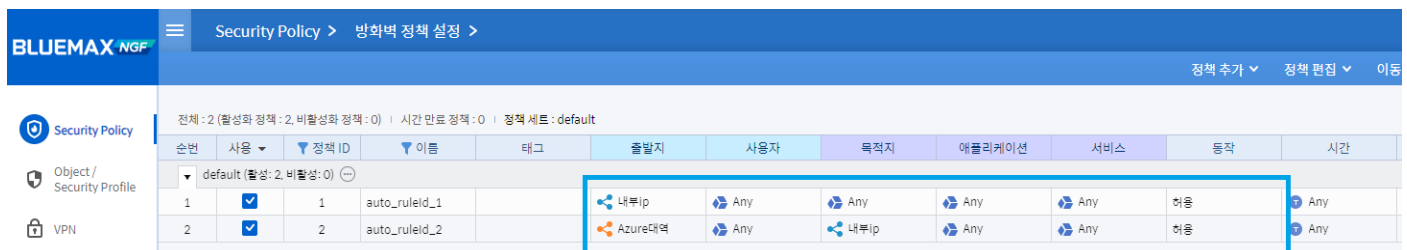
내부 사설 IP 가 외부 공인 IP 를 통해 외부로 향하는 NAT 정책을 설정합니다.



순번	사용	정책 ID	이름	적용 인터페이스	출발지	변경 전 목적지	서비스	출발지	변경 후 목적지	서비스	변환 방식
1	<input checked="" type="checkbox"/>	2	nat	eth1	내부ip	Any	Any	공인ip	Org	Org	N:M 확장변환 (Port Pool)

그림 7 NAT 정책 설정

내부 사설 IP 는 외부로, Azure IP 는 내부로 향하는 방화벽 정책을 설정합니다.



순번	사용	정책 ID	이름	태그	출발지	사용자	목적지	애플리케이션	서비스	동작	시간
1	<input checked="" type="checkbox"/>	1	auto_ruleid_1		내부ip	Any	Any	Any	Any	허용	Any
2	<input checked="" type="checkbox"/>	2	auto_ruleid_2		Azure대역	Any	내부ip	Any	Any	허용	Any

그림 8 방화벽 정책 설정

3. VPN 설정

① 방화벽 지점 연결 기본 설정

Azure Virtual Network Gateway 의 공인 IP 를 입력합니다.

지점 연결 설정 편집

기본 설정 | 고급 설정

사용 * ☒ ON

연결 이름 * team3-vpn-test

센터 장비 IP 주소 * 20.196.89.103

인증 방법 * ☒ 사전 공유키 방식 ☐ RSA 인증서 방식

지점 ID * vpn-test

인증 비밀번호 / 확인 * /

그림 9 방화벽 지점 연결 기본 설정

② 방화벽 지점 연결 고급 설정

지점 연결 설정 편집

기본 설정 | 고급 설정

연결 모드 ☒ IKEv2 ☐ Aggressive ☐ Main

로컬 IP 주소 ☒ ANY ☐ IP 주소

보안 정책 이름 AES-256-SHA-256-Group2

동작 모드 ESP-Tunnel

IKE SA 생명 주기(초)* 160000

IPSec SA 생명 주기(초)* 80000

센터 IKE 포트 500

UDP Encapsulation 강제 적용 * OFF

표준 IPSec ☒ ON

확장 모드 ☒ ON

센터 ☒ 외부망

그림 10 방화벽 지점 연결 고급 설정

③ 방화벽 지점 연결 외부 네트워크 추가

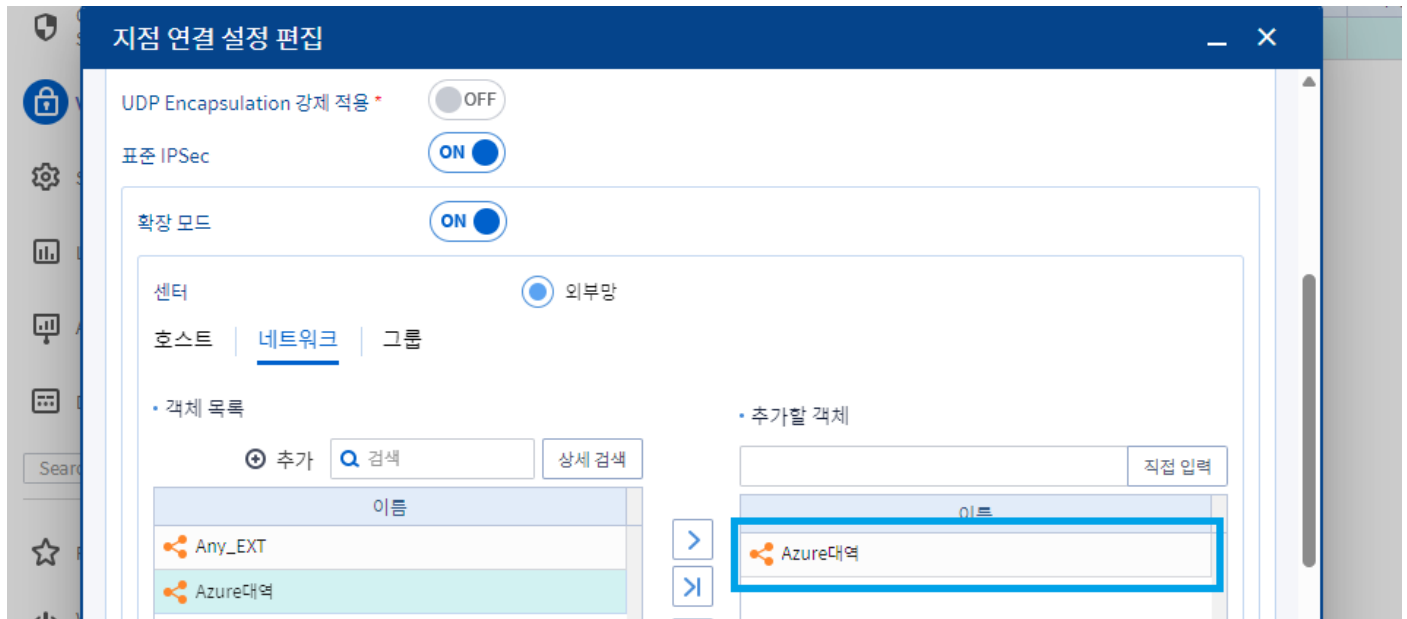


그림 11 방화벽 지점 연결 외부 네트워크 추가

④ 방화벽 지점 연결 내부 네트워크 추가

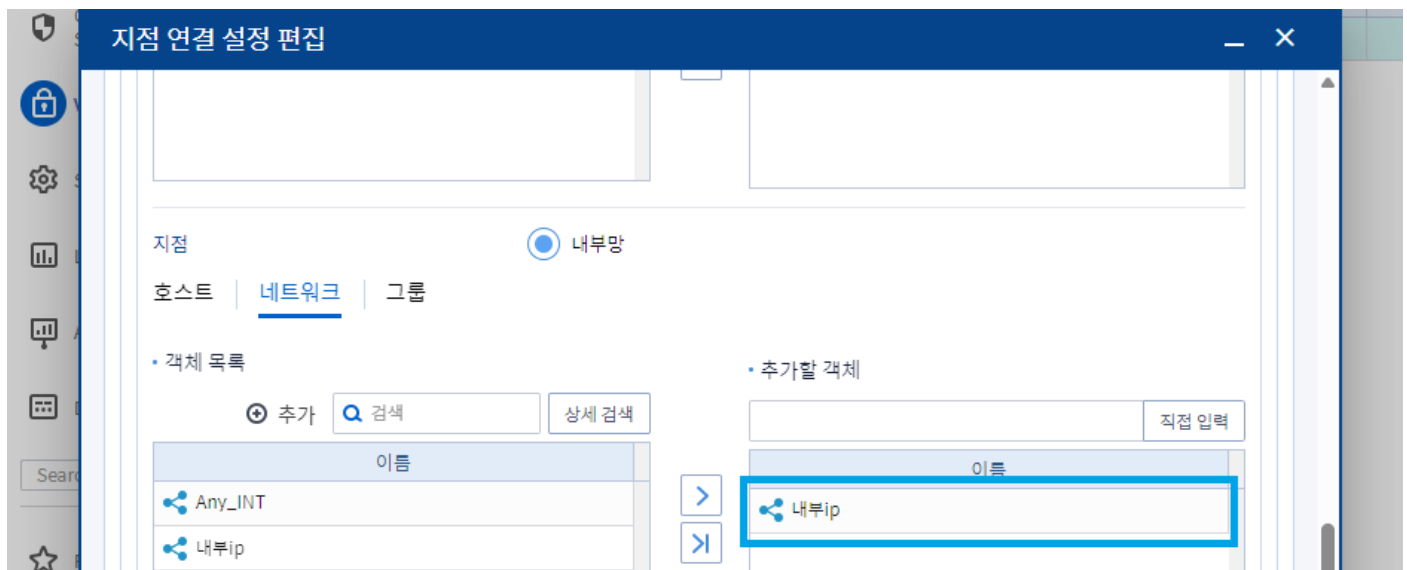


그림 12 방화벽 지점 연결 내부 네트워크 추가

⑤ 방화벽 보안 정책 설정

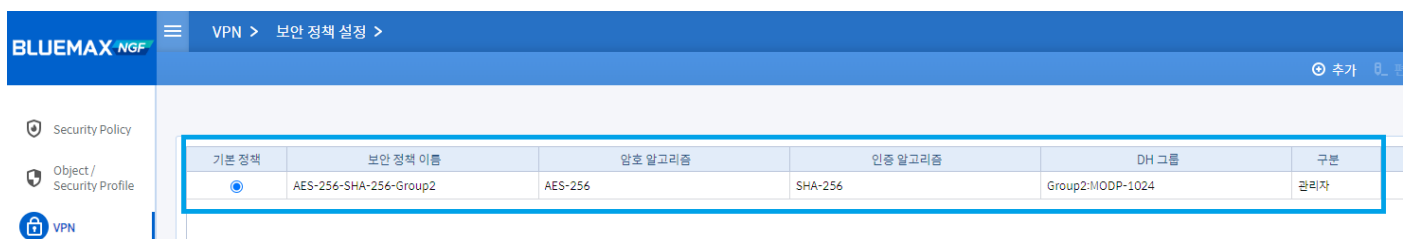


그림 13 방화벽 보안 정책 설정

V. Azure Cloud 구축

1. Vnet 구성

Vnet	IPv4 CIDR	역할
team3_vnet	10.0.0.0/16	Korea Central 에 위치한 Vnet

2. Subnet 구성

Subnet	IPv4 CIDR	역할
team3_bastion	10.0.0.0/24	보안관리자만 접속가능하다
team3_load	10.0.1.0/24	외부에서 오는 HTTP 트래픽을 로드밸런싱한다
team3_nat	10.0.2.0/24	내부 VM 이 공인 IP 을 통해 외부로 향한다
team3_web1	10.0.3.0/24	WEB1 서버가 위치한 곳이다
team3_web2	10.0.4.0/24	WEB2 서버가 위치한 곳이다
team3_vnetgw	10.0.5.0/24	DB 서버가 위치한 곳이다
team3_elk	10.0.6.0/24	Elastic 모니터링 서버가 위치 곳이다

3. 공인 IP 할당

IP.id	IPv4	역할
team3_bastion_ip	생성 시 할당	Bastion 의 공인 IP
team3_nat_ip	생성 시 할당	NAT 의 공인 IP
team3_appgwip	생성 시 할당	Application Gateway 의 공인 IP
team3_vpnip	생성 시 할당	VPN 의 공인 IP

4. 네트워크 보안그룹 적용

① team3_bastion_nsg

순위	프로토콜	접근	출발주소	출발포트	도착주소	도착포트	역할
200	TCP	Inbound	1.220.76.5	Any	10.0.0.4	22	원격접속

② team3_web_nsg

순위	프로토콜	접근	출발주소	출발포트	도착주소	도착포트	역할
200	TCP	Inbound	10.0.0.4	Any	Any	22	원격접속
210	TCP	Inbound	Any	Any	Any	80	웹서비스
220	TCP	Inbound	Any	Any	10.10.5.11	3306	MySQL

③ team3_elk_nsg

순위	프로토콜	접근	출발주소	출발포트	도착주소	도착포트	역할
200	TCP	Inbound	Any	Any	Any	22	원격접속
220	TCP	Inbound	Any	Any	Any	9200	Elastic
230	TCP	Inbound	Any	Any	Any	5601	Kibana

5. NAT 게이트웨이 적용

① team3_natgw

내부 서브넷의 VM 이 인터넷에 직접 노출되지 않도록, NAT IP 를 이용하여 외부로 나가는 패킷을 변환합니다. 내부 서브넷 web1, web2, auto 에 적용하여 보안을 향상시킵니다.

6. SSH 키 및 워드프레스 생성 시작스크립트

① local.id_rsa

SSH 개인키를 Bastion VM 에 배치하여 WEB 서버에 SSH 접속이 가능하게 하는 스크립트입니다.

② local.wd

WEB1 및 WEB2 VM 에서 워드프레스를 설치하고 DB 와 연동하게 하는 스크립트입니다.

7. 가상머신 생성

VM.id	IPv4	Storage	OS	역할
team3_bastion	10.0.0.4	StandardSSD	Rocky 9.3.20231113	보안관리자 PC
team3_web1	10.0.3.4	StandardSSD	Rocky 9.3.20231113	웹서버 1
team3_web2	10.0.4.4	StandardSSD	Rocky 9.3.20231113	웹서버 2
team3_elk	10.0.6.4	StandardSSD	Rocky 9.3.20231113	Elastic Search 모니터링

8. 어플리케이션 게이트웨이 적용

① team3_appgw

어플리케이션 게이트웨이는 7 계층 로드밸런서 역할을 수행합니다. 웹서비스 HTTP 에 대한 트래픽을 관리하고 특정 백엔드풀로 라우팅할 수 있도록 지정했습니다.

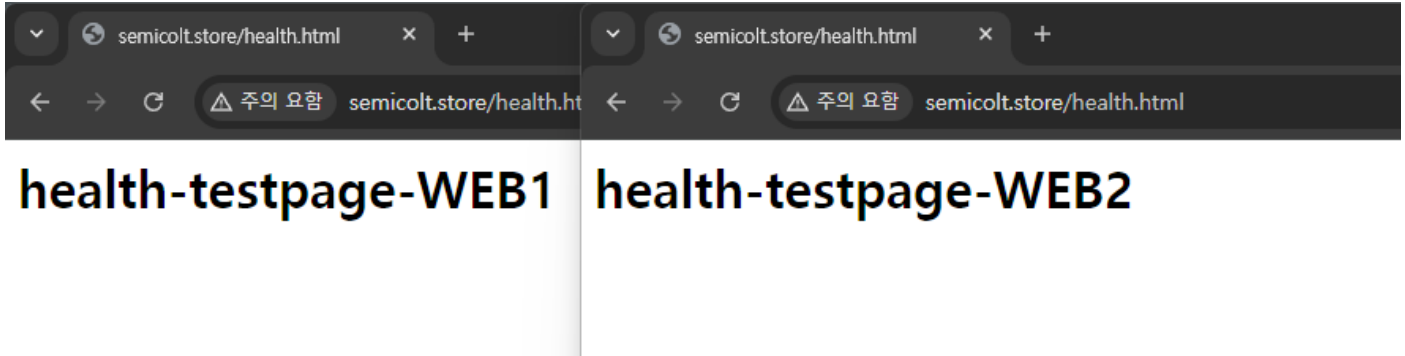


그림 14 어플리케이션 게이트웨이에 의해 부하 분산된 모습

9. 웹서비스에 DNS 적용

① team3_dns

DNS 레코드를 관리하기 위한 기본 영역을 설정합니다.

② team3_root_record

루트 도메인(@)에 대해 A 레코드를 생성하여, 지정된 퍼블릭 IP 에 연결합니다.

③ team3_root_cname

www 서브도메인에 대한 A 레코드를 생성하고, 지정된 퍼블릭 IP 를 타겟으로 연결합니다.

④ team3_ns

"team3-ns"라는 네임서버(NS) 레코드를 생성하여, 도메인의 네임서버를 설정합니다.

⑤ team3_ptr

IP 주소에서 도메인 이름으로의 역방향 검색을 가능하게 합니다.



그림 15 가비아 (https://domain.gabia.com)에서 Azure 네임서버를 등록

10. VPN 연결

① team3_vpn

VPN Gateway 를 생성합니다.

② team3-home

Local Network Gateway 생성합니다.

③ team3_vpn_connection

방화벽과 패킷설정을 동일하게 하여 VPN Gateway 와 Local Network Gateway 를 연결합니다.

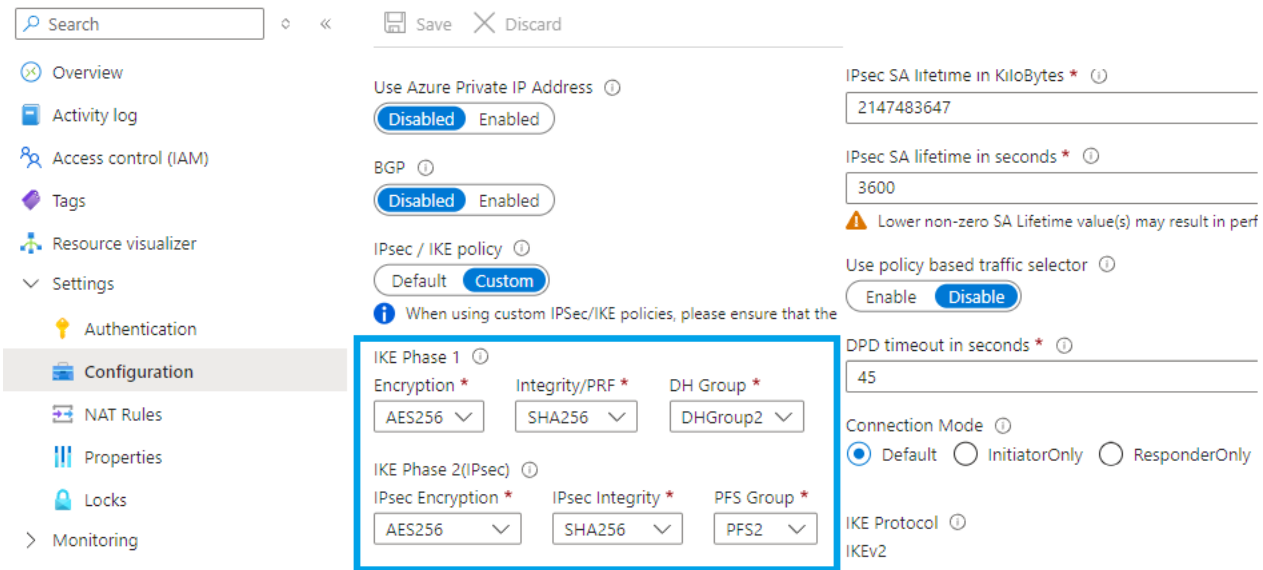


그림 16 Azure VPN connection 설정

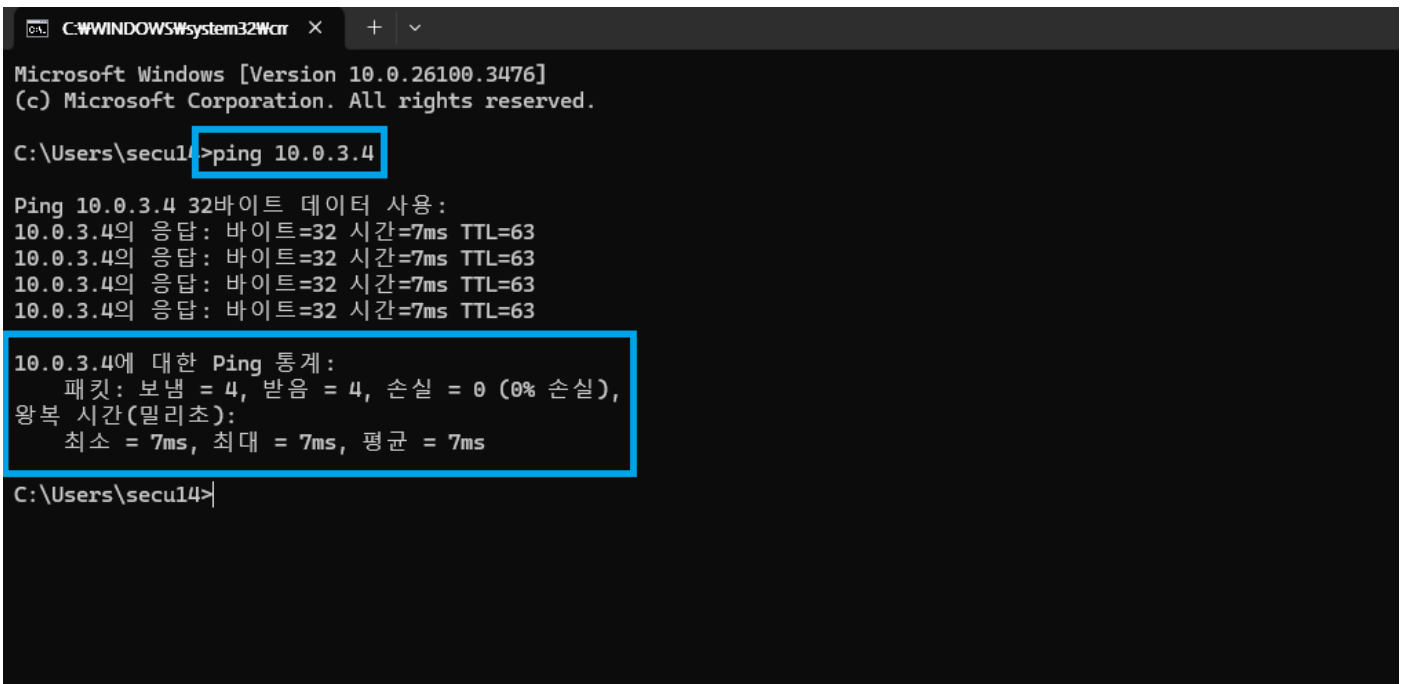


그림 17 VPN 설정 후 Rocky 에서 Azure WEB1 서버에 접속한 모습

11. ELK Stack

① ELK Stack

Elastic Search, Logstash, Kibana 의 조합으로, 로그 데이터를 수집하고 분석하는 오픈소스 솔루션입니다.

Elasticsearch 는 분산형 검색 및 분석 엔진으로, 수집된 로그 데이터를 저장하고 빠르게 검색할 수 있도록 지원합니다. 고성능의 색인 및 검색 기능을 제공하며, 대량의 데이터를 효율적으로 처리할 수 있습니다.

Elasticsearch 에 저장된 데이터를 검색하고, 그래프나 차트 형태로 시각화하여 실시간 모니터링과 트렌드 분석이 가능하도록 지원합니다.

Logstash 는 다양한 소스에서 로그를 수집하고, 데이터를 변환 및 필터링한 후 Elasticsearch 로 전송하는 데이터 처리 파이프라인 역할을 합니다. JSON 변환, 필터링, 파싱 등의 기능을 통해 데이터를 정제할 수 있습니다.

Kibana 는 수집된 데이터를 시각적으로 분석하는 대시보드 및 인터페이스를 제공합니다.

② DB 서버 모니터링 구축

먼저 DB 서버에 Filebeat 와 Logstash 를 설치하고 설정합니다. Filebeat 는 DB 접근 로그, Slow Query Log, Error log 등 다양한 로그를 수집합니다. Logstash 는 다양한 형식의 로그를 전처리한 후 Elasticsearch 로 전송합니다.

데이터 저장 및 분석을 위해 Azure ELK VM 에 Elasticsearch 를 설치하고 데이터 저장 구조를 만듭니다. 그리고 Kibana 를 설치하여 대시보드를 설정하고, 수집된 로그를 필터링 및 검색할 수 있도록 구성합니다. 이를 통해 DB 접근 패턴을 시각화합니다.

이를 통해 DB 서버의 보안 및 성능을 실시간으로 모니터링할 수 있으며, 로그인 시도 탐지 또는 느린 쿼리를 등의 이상 징후 발생 시 신속한 대응이 가능합니다. 또한 수집된 기반으로 장기적인 트렌드 분석 및 성능 최적화 전략을 수립할 수 있습니다.

서버	로그 종류	수집 도구	처리 방식	표현 도구
Host PC (DB)	DB 접근 로그	Filebeat	Logstash → Elasticsearch	Kibana

③ DB 서버 설정화면

```
# ===== Filebeat inputs =====  
  
filebeat.inputs:  
- type: log  
  id: team3  
  enabled: true  
  paths:  
    - /var/lib/docker/containers/*/*-json.log  
    #- c:\programdata\elasticsearch\logs\  
  json.keys_under_root: true  
  json.add_error_key: true
```

그림 18 Filebeat inputs 설정

```
# ===== Outputs =====  
  
# Configure what output to use when sending the data collected by the beat.  
  
# ----- Elasticsearch Output -----  
output.elasticsearch:  
  # Array of hosts to connect to  
  hosts: ["http://10.0.6.4:9200"]  
  
  # Performance preset - one of "balanced", "throughput", "scale",  
  # "latency", or "custom".  
  preset: balanced  
  
  # Protocol - either `http` (default) or `https`.  
  #protocol: "https"  
  
  # Authentication credentials - either API key or username/password.  
  #api_key: "id:api_key"  
  #username: "elastic"  
  #password: "changeme"
```

그림 19 Filebeat outputs 설정

```
team3@team3-elk:~$ curl localhost:9200  
{  
  "name" : "node1",  
  "cluster_name" : "monitor",  
  "cluster_uuid" : "RWz_ndSFQmCg2zwlNpJveA",  
  "version" : {  
    "number" : "8.17.4",  
    "build_flavor" : "default",  
    "build_type" : "deb",  
    "build_hash" : "c63c7f5f8ce7d2e4805b7b3d842e7e792d84dda1",  
    "build_date" : "2025-03-20T15:39:59.811110136Z",  
    "build_snapshot" : false,  
    "lucene_version" : "9.12.0",  
    "minimum_wire_compatibility_version" : "7.17.0",  
    "minimum_index_compatibility_version" : "7.0.0"  
  },  
  "tagline" : "You Know, for Search"  
}
```

그림 20 Elastic Search 설치 확인

④ ELK VM 에서 설치 확인

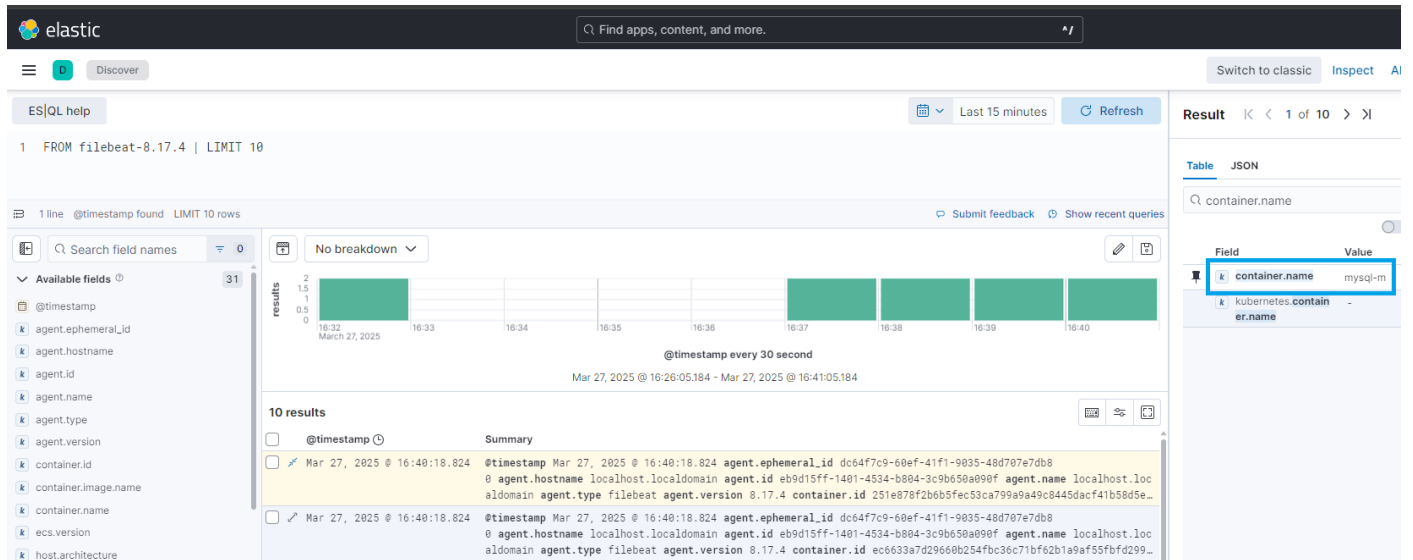


그림 21 메인 DB 로그

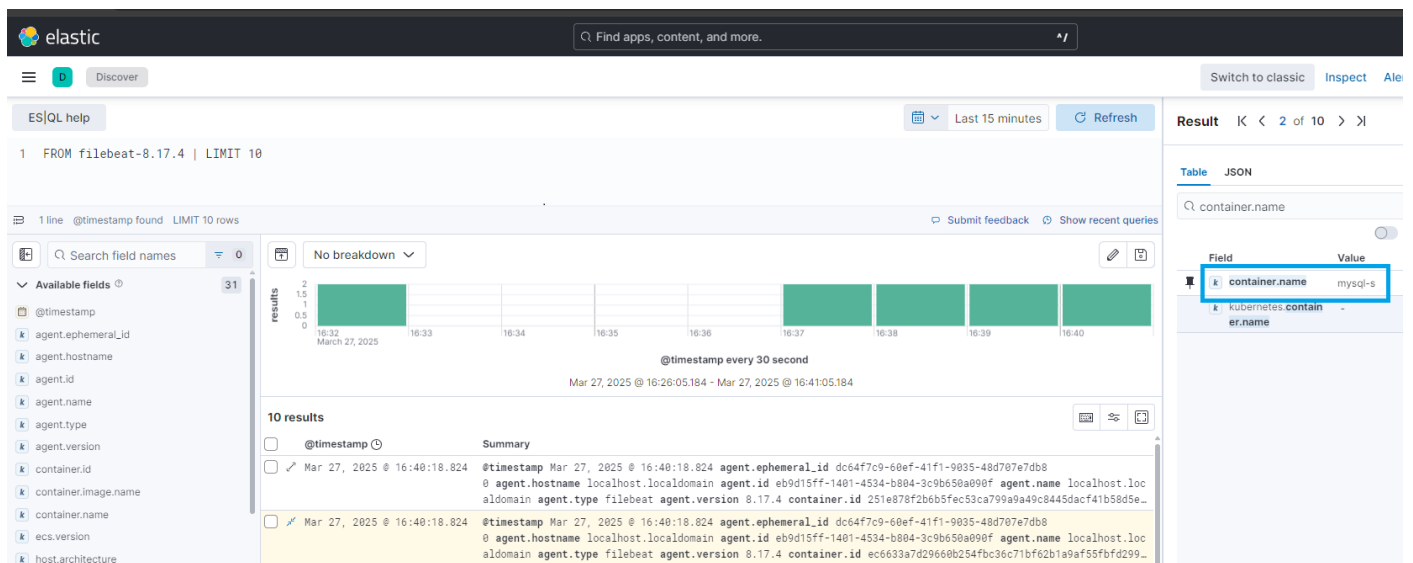


그림 22 서버 DB 로그

12. 테라폼 자동화코드

00_init.tf

```
terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "~> 4.23"
    }
  }
}

provider "azurerm" {
  subscription_id = var.subid
  features {}
}
```

01_rg.tf

```
# Create Resource Group
resource "azurerm_resource_group" "team3_rg" {
  name      = "02-${var.name}-rg"
  location = var.location
}
```

02_vnet.tf

```
# Create Virtual Network
resource "azurerm_virtual_network" "team3_vnet" {
  name                = "${var.name}-vnet"
  resource_group_name = azurerm_resource_group.team3_rg.name
  location            = azurerm_resource_group.team3_rg.location
  address_space       = ["10.0.0.0/16"]
}

# Create Public Subnet for Bastion
resource "azurerm_subnet" "team3_bastion" {
  name                = "${var.name}-bastion"
  resource_group_name = azurerm_resource_group.team3_rg.name
  virtual_network_name = azurerm_virtual_network.team3_vnet.name
  address_prefixes    = ["10.0.0.0/24"]
}

# Create Load Subnet
resource "azurerm_subnet" "team3_load" {
  name                = "${var.name}-load"
  resource_group_name = azurerm_resource_group.team3_rg.name
  virtual_network_name = azurerm_virtual_network.team3_vnet.name
}
```

```

    address_prefixes      = ["10.0.1.0/24"]
}

# Create NAT Subnet
resource "azurerm_subnet" "team3_nat" {
    name                  = "${var.name}-nat"
    resource_group_name  = azurerm_resource_group.team3_rg.name
    virtual_network_name = azurerm_virtual_network.team3_vnet.name
    address_prefixes     = ["10.0.2.0/24"]
}

# Create Web1 Subnet
resource "azurerm_subnet" "team3_web1" {
    name                  = "${var.name}-web1"
    resource_group_name  = azurerm_resource_group.team3_rg.name
    virtual_network_name = azurerm_virtual_network.team3_vnet.name
    address_prefixes     = ["10.0.3.0/24"]
}

# Create Web2 Subnet
resource "azurerm_subnet" "team3_web2" {
    name                  = "${var.name}-web2"
    resource_group_name  = azurerm_resource_group.team3_rg.name
    virtual_network_name = azurerm_virtual_network.team3_vnet.name
    address_prefixes     = ["10.0.4.0/24"]
}

# Create VPN Subnet
resource "azurerm_subnet" "team3_vnetgw" {
    name                  = "GatewaySubnet"
    resource_group_name  = azurerm_resource_group.team3_rg.name
    virtual_network_name = azurerm_virtual_network.team3_vnet.name
    address_prefixes     = ["10.0.5.0/24"]
}

```

03_publicip.tf

```

# Public IP for Bastion VM
resource "azurerm_public_ip" "team3_bastion_ip" {
    name                  = "${var.name}-bastion-ip"
    resource_group_name  = azurerm_resource_group.team3_rg.name
    location              = azurerm_resource_group.team3_rg.location
    allocation_method     = "Static"
    sku                  = "Standard"
}

# Public IP for NAT Gateway
resource "azurerm_public_ip" "team3_nat_ip" {
    name                  = "${var.name}-nat-ip"
    resource_group_name  = azurerm_resource_group.team3_rg.name
    location              = azurerm_resource_group.team3_rg.location
    allocation_method     = "Static"
}

```



```

    sku                = "Standard"
}

# Public IP for Load Balancer
resource "azurerm_public_ip" "team3_appgwip" {
  name                = "${var.name}-lb-ip"
  resource_group_name = azurerm_resource_group.team3_rg.name
  location            = azurerm_resource_group.team3_rg.location
  allocation_method   = "Static"
  sku                = "Standard"
}

# Public IP for VNetGW
resource "azurerm_public_ip" "team3_vpnip" {
  name                = "${var.name}-vpn-ip"
  resource_group_name = azurerm_resource_group.team3_rg.name
  location            = azurerm_resource_group.team3_rg.location
  allocation_method   = "Static"
  sku                = "Standard"
}

```

04_nsg.tf

```

# Network Security Group for Bastion
resource "azurerm_network_security_group" "team3_bastion_nsg" {
  name                = "${var.name}-bat-nsg"
  location            = azurerm_resource_group.team3_rg.location
  resource_group_name = azurerm_resource_group.team3_rg.name
  # Bastion SSH Allow Rule
  security_rule {
    name                = "Allow-SSH-From-Host"
    priority            = 200
    direction           = "Inbound"
    access              = "Allow"
    protocol            = "Tcp"
    source_port_range   = "*"
    destination_port_range = "22"
    source_address_prefix = var.local_public_ip
    destination_address_prefix = var.bastion_ip
  }
  security_rule {
    name                = "Allow-Ping"
    priority            = 205
    direction           = "Inbound"
    access              = "Allow"
    protocol            = "Icmp"
    source_port_range   = "*"
    destination_port_range = "*"
    source_address_prefix = "*"
    destination_address_prefix = "*"
  }
}

```

```

}

# Network Security Group for Internal Network
resource "azurerm_network_security_group" "team3_web_nsg" {
  name                = "${var.name}-web-nsg"
  location            = azurerm_resource_group.team3_rg.location
  resource_group_name = azurerm_resource_group.team3_rg.name
  # Web SSH Allow Rule
  security_rule {
    name                = "Allow-SSH-From-Bastion"
    priority            = 200
    direction          = "Inbound"
    access             = "Allow"
    protocol            = "Tcp"
    source_port_range   = "*"
    destination_port_range = "22"
    source_address_prefix = var.bastion_ip
    destination_address_prefix = "*"
  }
  # Ping Allow Rule
  security_rule {
    name                = "Allow-Ping"
    priority            = 205
    direction          = "Inbound"
    access             = "Allow"
    protocol            = "Icmp"
    source_port_range   = "*"
    destination_port_range = "*"
    source_address_prefix = "*"
    destination_address_prefix = "*"
  }
  # Web Http Allow Rule
  security_rule {
    name                = "Allow-HTTP-From-All"
    priority            = 210
    direction          = "Inbound"
    access             = "Allow"
    protocol            = "Tcp"
    source_port_range   = "*"
    destination_port_range = "80"
    source_address_prefix = "*"
    destination_address_prefix = "*"
  }
  # DB Allow Rule
  security_rule {
    name                = "Allow-MySQL-To-DB"
    priority            = 220
    direction          = "Inbound"
    access             = "Allow"
    protocol            = "Tcp"
    source_port_range   = "*"
    destination_port_range = "3306"
  }

```

```

    source_address_prefix      = "*"
    destination_address_prefix = var.DB_ip
  }
}

```

05_natgw.tf

```

# Create NAT Gateway
resource "azurerm_nat_gateway" "team3_natgw" {
  name                = "${var.name}-natgw"
  location            = azurerm_resource_group.team3_rg.location
  resource_group_name = azurerm_resource_group.team3_rg.name
}

# NAT Gateway Association

resource "azurerm_subnet_nat_gateway_association" "team3_web1_nat" {
  subnet_id          = azurerm_subnet.team3_web1.id
  nat_gateway_id     = azurerm_nat_gateway.team3_natgw.id
}

resource "azurerm_subnet_nat_gateway_association" "team3_web2_nat" {
  subnet_id          = azurerm_subnet.team3_web2.id
  nat_gateway_id     = azurerm_nat_gateway.team3_natgw.id
}

# Attachment Public IP -> NAT Gateway
resource "azurerm_nat_gateway_public_ip_association" "team3_natgwp_pubip" {
  nat_gateway_id      = azurerm_nat_gateway.team3_natgw.id
  public_ip_address_id = azurerm_public_ip.team3_nat_ip.id
}

```

06_nic.tf

```

# Network Interface Card for Bastion VM
resource "azurerm_network_interface" "team3_bat_nic" {
  name                = "${var.name}-bat-nic"
  location            = azurerm_resource_group.team3_rg.location
  resource_group_name = azurerm_resource_group.team3_rg.name

  ip_configuration {
    name                        = "${var.name}-bat-ip"
    subnet_id                  = azurerm_subnet.team3_bastion.id
    private_ip_address_allocation = "Static"
    private_ip_address          = var.bastion_ip
    public_ip_address_id         = azurerm_public_ip.team3_bastion_ip.id
  }
}

# Network Interface Card for Web1
resource "azurerm_network_interface" "team3_web1_nic" {

```

```

name                = "${var.name}-web1-nic"
location            = azurerm_resource_group.team3_rg.location
resource_group_name = azurerm_resource_group.team3_rg.name

ip_configuration {
  name                = "${var.name}-web1-ip"
  subnet_id          = azurerm_subnet.team3_web1.id
  private_ip_address_allocation = "Static"
  private_ip_address  = "10.0.3.4"
}
}

# Network Interface Card for Web2
resource "azurerm_network_interface" "team3_web2_nic" {
  name                = "${var.name}-web2-nic"
  location            = azurerm_resource_group.team3_rg.location
  resource_group_name = azurerm_resource_group.team3_rg.name

  ip_configuration {
    name                = "${var.name}-web2-ip"
    subnet_id          = azurerm_subnet.team3_web2.id
    private_ip_address_allocation = "Static"
    private_ip_address  = "10.0.4.4"
  }
}

```

07_nsg-nic.tf

```

# Bastion NIC <-> NSG
resource "azurerm_network_interface_security_group_association" "team3_bat_nic_nsgasso" {
  network_interface_id      = azurerm_network_interface.team3_bat_nic.id
  network_security_group_id = azurerm_network_security_group.team3_bastion_nsg.id
}

# Web NIC <-> NSG
resource "azurerm_network_interface_security_group_association" "team3_web1_nic_nsgasso" {
  network_interface_id      = azurerm_network_interface.team3_web1_nic.id
  network_security_group_id = azurerm_network_security_group.team3_web_nsg.id
}

resource "azurerm_network_interface_security_group_association" "team3_web2_nic_nsgasso" {
  network_interface_id      = azurerm_network_interface.team3_web2_nic.id
  network_security_group_id = azurerm_network_security_group.team3_web_nsg.id
}

```

08_bastion_vm.tf

```

# Create Bastion VM
resource "azurerm_linux_virtual_machine" "team3_bastion" {
  name = "${var.name}-bastion"

```

```

resource_group_name = azurerm_resource_group.team3_rg.name
location            = azurerm_resource_group.team3_rg.location
size                = "Standard_F1s"
admin_username      = var.name
network_interface_ids = [azurerm_network_interface.team3_bat_nic.id]
user_data           = base64encode(local.id_rsa)

admin_ssh_key {
  username   = var.name
  public_key = file("id_rsa.pub")
}

os_disk {
  caching              = "ReadWrite"
  storage_account_type = "StandardSSD_LRS"
}

source_image_reference {
  publisher = "resf"
  offer     = "rockylinux-x86_64"
  sku      = "9-lvm"
  version  = "9.3.20231113"
}

plan {
  publisher = "resf"
  product   = "rockylinux-x86_64"
  name      = "9-lvm"
}
}

```

09_web_vm.tf

```

# Create VM Web1
resource "azurerm_linux_virtual_machine" "team3_web1" {
  name                = "${var.name}-web1"
  resource_group_name = azurerm_resource_group.team3_rg.name
  location            = azurerm_resource_group.team3_rg.location
  size                = "Standard_F1s"
  admin_username      = var.name
  network_interface_ids = [azurerm_network_interface.team3_web1_nic.id]
  user_data           = base64encode(local.wd)

  admin_ssh_key {
    username   = var.name
    public_key = file("id_rsa.pub")
  }

  os_disk {
    caching              = "ReadWrite"
    storage_account_type = "StandardSSD_LRS"
  }
}

```

```

}

source_image_reference {
  publisher = "resf"
  offer     = "rockylinux-x86_64"
  sku       = "9-lvm"
  version   = "9.3.20231113"
}

plan {
  publisher = "resf"
  product   = "rockylinux-x86_64"
  name      = "9-lvm"
}
}

# Create VM Web2
resource "azurerm_linux_virtual_machine" "team3_web2" {
  name                        = "${var.name}-web2"
  resource_group_name        = azurerm_resource_group.team3_rg.name
  location                   = azurerm_resource_group.team3_rg.location
  size                       = "Standard_F1s"
  admin_username             = var.name
  network_interface_ids     = [azurerm_network_interface.team3_web2_nic.id]
  user_data                  = base64encode(local.wd)

  admin_ssh_key {
    username   = var.name
    public_key = file("id_rsa.pub")
  }

  os_disk {
    caching              = "ReadWrite"
    storage_account_type = "StandardSSD_LRS"
  }

  source_image_reference {
    publisher = "resf"
    offer     = "rockylinux-x86_64"
    sku       = "9-lvm"
    version   = "9.3.20231113"
  }

  plan {
    publisher = "resf"
    product   = "rockylinux-x86_64"
    name      = "9-lvm"
  }
}

```

10_appgw.tf

```
# Create Application Gateway
resource "azurerm_application_gateway" "team3_appgw" {
  name                = "${var.name}-appgw"
  resource_group_name = azurerm_resource_group.team3_rg.name
  location            = azurerm_resource_group.team3_rg.location

  sku {
    name     = "Basic"
    tier      = "Basic"
    capacity = 2
  }

  gateway_ip_configuration {
    name          = "${var.name}-gateway-ip-configuration"
    subnet_id     = azurerm_subnet.team3_load.id
  }
  # For HTTP Service
  frontend_port {
    name = var.frontend_port_name
    port = 80
  }
  # Attachment Public IP
  frontend_ip_configuration {
    name          = var.frontend_ip_configuration_name
    public_ip_address_id = azurerm_public_ip.team3_appgwip.id
  }
  # Backend IP Setting
  backend_address_pool {
    name          = var.backend_address_pool_name
    ip_addresses = ["10.0.3.4", "10.0.4.4"]
  }

  backend_http_settings {
    name          = var.http_setting_name
    cookie_based_affinity = "Disabled"
    path          = "/"
    port          = 80
    protocol      = "Http"
    request_timeout = 60
  }

  http_listener {
    name          = var.listener_name
    frontend_ip_configuration_name = var.frontend_ip_configuration_name
    frontend_port_name            = var.frontend_port_name
    protocol                      = "Http"
  }

  request_routing_rule {
    name = var.request_routing_rule_name
```

```

    rule_type          = "Basic"
    http_listener_name  = var.listener_name
    backend_address_pool_name = var.backend_address_pool_name
    backend_http_settings_name = var.http_setting_name
    priority            = 100
  }
}

```

11_dns.tf

```

resource "azurerm_dns_zone" "team3_dns" {
  name                = "semicolts.store"
  resource_group_name = azurerm_resource_group.team3_rg.name
}

resource "azurerm_dns_a_record" "team3_root_record" {
  name                = "@"
  resource_group_name = azurerm_resource_group.team3_rg.name
  zone_name           = azurerm_dns_zone.team3_dns.name
  ttl                 = 300
  target_resource_id  = azurerm_public_ip.team3_appgwip.id
}

resource "azurerm_dns_a_record" "team3_root_cname" {
  name                = "www"
  resource_group_name = azurerm_resource_group.team3_rg.name
  zone_name           = azurerm_dns_zone.team3_dns.name
  ttl                 = 300
  target_resource_id  = azurerm_public_ip.team3_appgwip.id
}

resource "azurerm_dns_ns_record" "team3_ns" {
  name                = "team3-ns"
  zone_name           = azurerm_dns_zone.team3_dns.name
  resource_group_name = azurerm_resource_group.team3_rg.name
  ttl                 = 300
  records              = ["ns1.semicolts.store"]
}

resource "azurerm_dns_ptr_record" "team3_ptr" {
  name                = "team3-ptr"
  zone_name           = azurerm_dns_zone.team3_dns.name
  resource_group_name = azurerm_resource_group.team3_rg.name
  ttl                 = 300
  records              = ["semicolts.store"]
}

```


12_vngw.tf

```
resource "azurerm_virtual_network_gateway" "team3_vpn" {
  name                = "${var.name}-vpn"
  location            = azurerm_resource_group.team3_rg.location
  resource_group_name = azurerm_resource_group.team3_rg.name

  type      = "Vpn"
  vpn_type = "RouteBased"

  active_active = false
  enable_bgp    = false
  sku           = "VpnGw1"
  generation    = "Generation1"

  ip_configuration {
    name                = "${var.name}-vnetGatewayConfig"
    public_ip_address_id = azurerm_public_ip.team3_vpnip.id
    private_ip_address_allocation = "Dynamic"
    subnet_id           = azurerm_subnet.team3_vnetgw.id
  }
}
```

13_localnwgw.tf

```
resource "azurerm_local_network_gateway" "team3_home" {
  name                = "${var.name}-backhome"
  resource_group_name = azurerm_resource_group.team3_rg.name
  location            = azurerm_resource_group.team3_rg.location
  gateway_address     = "1.220.76.5"
  address_space       = ["10.10.5.0/24"]
}
```

14_vpncon.tf

```
resource "azurerm_virtual_network_gateway_connection" "team3_vpn_connection" {
  name                = "${var.name}-vpn-connection"
  location            = azurerm_resource_group.team3_rg.location
  resource_group_name = azurerm_resource_group.team3_rg.name

  type = "IPsec"

  virtual_network_gateway_id = azurerm_virtual_network_gateway.team3_vpn.id
  local_network_gateway_id   = azurerm_local_network_gateway.team3_home.id

  shared_key = "It12345!"

  ipsec_policy {
    dh_group      = "DHGroup2"
    ike_encryption = "AES256"
    ike_integrity  = "SHA256"
  }
}
```

```

    ipsec_encryption = "AES256"
    ipsec_integrity  = "SHA256"
    pfs_group        = "PFS2"
    sa_lifetime      = 3600
    sa_datasize       = 2147483647
  }
}

```

100_var.tf

```

variable "location" {
  type    = string
  default = "Korea Central"
}

variable "name" {
  type    = string
  default = "team3"
}

variable "bastion_ip" {
  type    = string
  default = "10.0.0.4"
}

variable "local_public_ip" {
  type    = string
  default = "1.220.76.5"
}

variable "DB_ip" {
  type    = string
  default = "10.10.5.11"
}

variable "password" {
  type    = string
  default = "It12345!"
}

variable "id_rsa" {
  type    = string
  default = "-----BEGIN RSA PRIVATE KEY-----
\nMIIEEgIBAAKCAQEAK8ugubI2AOE5utL6qcZ68khg5KsU6BnNRzQF40MasdupYIRh\nnwNLKT+kT/TJhdOsredmbaP
PyozKACx90K2aqxkPlIcq6UwymXDRQatRY4RLbj/qc\nCJMhLGQfBSa00RgWDFvxTsmC5cQdr2MFcqH8AQ6ttQIR9S
CdbXwJuRfiroa8GFI1\nXiyxLUPzXLOCf5vcgB2GAZMr1Gs2bbgkzgu7ZnXngh7kpAjZkgZ1bYsiAm2CyZNw\nnH2z1
3i+78gGYGSTxR5aT08NaPW0RaJM1KVG5YhGU0xuLAMzRmjKnsZPL8zXhPBb9\nnnBpIeE31zuX5GBDxQaut/GzowIJw
qPz70uPqaQIDAQABAOIBACAgdXGoQYDzdH8d\nn+YsKMBOWlClg3wyM/0hpEJYQB+9911X6sN0fdQb13Yh+eRzXOf4k
xVmNHcNH6PmB\nn/tioWpr6hbYQMwx4JXmJjamLOz7BwKSNj3l1shJaOTv/Y7AkEjbjhJZTYzPFxLhq/\n5aFDgOoh7L
SaHJkyW8Y+3kYz4PyazJsPCvBgV9db26d10gszZc7z6Ex4tltXLGOH\nnmcfrEyiOXurgxtT8PVWlG5NqV9280Q/Lhq
7Ug7f02yqYwuYo6wAKnf00cqM0c2bN\nnD4JpAh9ilAs+xOygrzrwz7IXzwh8osE2f1XCCAAIVmRo03AMc/+1BB7g+d

```

```
tkb0aB\n0Z7gkBkCgYEAw+koBhEuMziDWrRHL+HP+MxmdBi3YzF7q0JeSf/uqUDqnbKEJx91\nRPKIVyYQcR3rabZa
byo6Jo+JtzhnBY5etcOp+I6CyyNzte3QMh0MN8zt6xt7PmHa\nfZqAKEKJg57AG538VG3K2qVowiv8j3+6rg9sxQNp
P1zFwHbhMT1xVaMCgYEAWSB4\nv9yND8466uaSm45aMYFwtoslH0lhxlpLD1LlaxrFRNQsoCjERf0SbgOEsfqRov
\nODDoulABADTX0rbXCBFtM1lcT0WesBPbhHDuDgdYAHGHwpzC49ZRL+5ZJoFRqnxV\nnid9qY2ZsLzPPbkMgGaWYOU
zg286CYbjT349oCIMCgYA6mXwqTgtIUOghcnk0n1PJ\n9MHUiiwYlI2Ll6JiehZWNB+5adNTrHo0VGNpxe33F975sN
zeEYBxVk/4KPvP0qUL\ns2oc+euvfXw3991llAC6lCa6Q28a2sQy+8rHwBdH8m0+FYSLcIfT3NQ3+FZESg15\nn0Q1U
5M9Pb52LE0QKqXlEBwKBgAK3jDtWxP1F73pCiCl8FTTBf27aAkmwZJm/x3KN\nvngCFveS7/2y5Zh199/ZoQkOmtCIY
zjVCKuQEIB96ntauD9Rj0mAM9cl/tz306bWx\nnSQgVf11z025A6lT+P00gTMzH4Pg/C40HAZ4oYL1BR1fLxfXWaI/V
zwwImbj6OrDY\nneaQ1AoGAZxMCQ3xuYg63bpusXA+kMxRXgCJ7hkj3ME7e3fswitN0yhNu5Nquw6+p\nnMD/Km5s+1F
uNw5KNCGLWPe99RAP9AR6PYoXCssR2Pn2I90WGbOyad6LWVu10BaK\nn1mk1KXlzo4xCuv6YwsX0orQlgrp753irY0
WNq7t8BX8AY/mJDWQ=\n-----END RSA PRIVATE KEY-----"
}
```

```
variable "subid" {
  type    = string
  default = "99b79efe-ebd6-468c-b39f-5669acb259e1"
}
```

```
variable "backend_address_pool_name" {
  default = "team3-Backaddrpool"
}
```

```
variable "frontend_port_name" {
  default = "team3-FrontendPort"
}
```

```
variable "frontend_ip_configuration_name" {
  default = "team3-AGIPConfig"
}
```

```
variable "http_setting_name" {
  default = "team3-HTTPsetting"
}
```

```
variable "listener_name" {
  default = "team3-Listener"
}
```

```
variable "request_routing_rule_name" {
  default = "team3-RoutingRule"
}
```

101_local.tf

```
# USER DATA File for Bastion VM
locals {
  id_rsa = <<USER_DATA
#!/bin/bash
mkdir /home/${var.name}/.ssh
echo -e "${var.id_rsa}" > /home/${var.name}/.ssh/id_rsa
chmod 600 /home/${var.name}/.ssh/id_rsa
```

```

chown ${var.name}.${var.name} /home/${var.name}/.ssh/id_rsa
USER_DATA
}

# USER DATA File for Web Service VM
locals {
  wd = <<USER_DATA
#!/bin/bash
setenforce 0
grubby --update-kernel ALL --args selinux=0
yum install -y httpd wget tar php php-cli php-pdo php-fpm php-json php-mysqlnd
wget https://ko.wordpress.org/wordpress-6.7.2-ko_KR.tar.gz
tar xvfz wordpress-6.7.2-ko_KR.tar.gz
cp -ar wordpress/* /var/www/html/
sed -i "s/DirectoryIndex index.html/DirectoryIndex index.php/g" /etc/httpd/conf/httpd.conf
cp /var/www/html/{wp-config-sample.php,wp-config.php}
sed -i "s/database_name_here/wordpress/g" /var/www/html/wp-config.php
sed -i "s/username_here/${var.name}/g" /var/www/html/wp-config.php
sed -i "s/password_here/It12345!/g" /var/www/html/wp-config.php
sed -i "s/localhost/10.10.5.11/g" /var/www/html/wp-config.php
cat > /var/www/html/health.html << eof
<html><body><h1>health-testpage</h1></body></html>
eof
chown -R apache.apache /var/www
systemctl enable --now httpd
USER_DATA
}

```

102_output.tf

```

# Check Public IP
output "Bastion_Public_IP" {
  value = azurerm_public_ip.team3_bastion_ip.ip_address
}

output "LB_Public_IP" {
  value = azurerm_public_ip.team3_appgwip.ip_address
}

output "VPN_Public_IP" {
  value = azurerm_public_ip.team3_vpnip.ip_address
}

```

VI. 향후 개선방향

1. 방화벽 정책설정

현재 일부 방화벽 정책이 화이트리스트 방식을 사용하면서도 Any 설정이 되어 있어, 보안성이 다소 미흡한 상태입니다. 이를 개선하기 위해 최소 권한 원칙 적용으로 특정 IP 대역과 포트만 허용하여 불필요한 접근을 차단할 수 있습니다.

순번	사용	정책 ID	이름	태그	출발지	사용자	목적지	애플리케이션	서비스	동작
1	<input checked="" type="checkbox"/>	1	auto_ruleid_1		내부ip	Any	Any	Any	Any	허용
2	<input checked="" type="checkbox"/>	2	auto_ruleid_2		Azure대역	Any	내부ip	Any	Any	허용

그림 23 현재 방화벽 정책설정

2. ELK 구현 개선

현재 ELK Stack 을 활용하여 로그 데이터를 수집하고 분석하는 구조를 구축하는 과정을 자동화하지 않았으나, 운영의 효율성을 높이기 위해 자동화 및 최적화를 고려할 수 있습니다.

ELK 설치 및 설정을 자동화하는 스트립트를 작성하면, 여러 서버에 동일한 환경을 빠르게 구축할 수 있습니다. 또한 테라폼을 활용하여 자동 배포 및 설정 관리가 가능하도록 개선할 수 있습니다.

그리고 현재 웹 트래픽 처리를 고려하여 DB 서버 모니터링용 VM 을 Azure 에 두었는데 이를 HostPC 에 두어서 리소스를 절약하고 모니터링 성능을 향상할 수 있습니다.

3. VMSS 적용 검토

퍼블릭 클라우드 구축 프로젝트와 달리, 이번 프로젝트에서는 리소스 최적화를 위해 VMSS 를 구성하지 않았습니다. 그렇지만 실제 운영 환경에서는 고가용성을 보장하기 위해 VMSS 를 적용하는 것이 필요합니다.

VII. 참고문헌

1. Microsoft. (n.d.). *Microsoft Azure Portal*. Retrieved March 28, 2025

<https://portal.azure.com>

2. Microsoft. (n.d.). *Design and implement Microsoft Azure networking solutions (AZ-700)*. Microsoft Learn. Retrieved March 28, 2025

<https://learn.microsoft.com/ko-kr/training/paths/design-implement-microsoft-azure-networking-solutions-az-700/>

3. HashiCorp. (n.d.). *Azure Resource Manager (azurerm) provider*. Terraform Registry. Retrieved March 28, 2025

<https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs>

4. Microsoft. (n.d.). *Install the Azure CLI on Windows using winget*. Microsoft Learn. Retrieved March 28, 2025

<https://learn.microsoft.com/en-us/cli/azure/install-azure-cli-windows?pivot=winget>

5. Google. (n.d.). *cAdvisor storage documentation*. Retrieved March 28, 2025

<https://github.com/google/cadvisor/tree/master/docs/storage>

6. Elastic. (n.d.). *Elasticsearch: The official distributed search & analytics engine*. Retrieved March 28, 2025

<https://www.elastic.co/kr/elasticsearch>