

결 재	담당	원장

클라우드데브옵스(DevOps) 엔지니어및관리자 양성과정(8기)

## 2차 프로젝트 완료 보고서

- VPN을 활용한 하이브리드 클라우드 구축 -

2023.10.04

구성원 : 김경태, 김학남, 서희경



고용노동부



한국정보교육원  
구.경원직업전문학교

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

문서 개정 이력

개정번호	개정일자	시행일자	개정내용	담당자
1.0	2023.10.04.		최초 작성	김경태

교 육 기 관 : 한 국 정 보 교 육 원  
 팀 명 :  
 팀 장 :  
 팀 원 :

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

# 

1. 프로젝트 개요	
● 프로젝트 명 .....	4
● 프로젝트 기간 .....	4
● 프로젝트 목표 .....	4
● 프로젝트 시나리오 .....	4
● 프로젝트 수행 요건 .....	4
2. 프로젝트 추진 체계	
● 프로젝트 참여 인력 총괄표 .....	5
● 참여 인력 업무 분장 .....	5
3. 세부 프로젝트 내용	
● 전체 구성도 .....	6
● 네트워크 구성도 .....	6
● 서버(물리/가상) 구성 현황 .....	7
● 네트워크 구성 현황 .....	9
● 상세 구축 및 구성 내용 .....	14
● 구축 결과 .....	20
4. 프로젝트 일정 .....	24
5. 피드백	
● 미달성 목표 .....	21
● 향후 계획 .....	21

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

## 1. 프로젝트 개요

- 프로젝트 명

VPN을 활용한 하이브리드 클라우드 구축

- 프로젝트 기간

2023.08.21. ~ 2023.10.4.(총 45일)

- 프로젝트 목표

가) **AWS**를 이용한 클라우드 컴퓨팅 서비스 구축

나) 가용영역 이중화 및 **Public/Private Subnet** 구분으로 가용성과 보안성 확보

다) Service **Load Balancing, Auto Scaling** 서비스 구축

라) **Kubernetes**를 활용하여 컨테이너 자동 배포 시스템 구축 -> **PaaS** 제공

마) 퍼블릭 클라우드와 온프레미스를 **VPN**을 활용하여 **하이브리드 클라우드** 구성

- 프로젝트 시나리오

가) 대상

- 00컴퍼니 (가상)

나) 가정

- 유튜브 동영상 시청이 가능한 학원 커뮤니티가 필요해짐.

1. 보안 및 규정 준수 문제:

고객사는 민감한 고객 데이터를 다루고 있으며, 관련 규정 및 규정 준수 요구 사항을 준수해야 합니다. 클라우드 서비스를 도입하지 않으면 보안 및 규정 준수 요구 사항을 충족하기 어렵습니다.

2. 확장성 및 리소스 최적화 문제

수시로 변하는 요구 사항에 대응하기 위한 유연한 IT 인프라가 필요합니다.

현재의 온프레미스 환경은 확장이 제한되며, 리소스를 최적화할 수 없어서 비용 효율적인 운영이 어렵습니다.

3. 재해 복구 및 비상 대응 미비성

현재의 온프레미스 인프라는 단일 지역에 집중되어 있어 자연 재해 또는 기타 재난 시에 업무 중단 위험이 큼. 클라우드 기술을 활용하여 다중 지역에 데이터 백업 및 복구 옵션을 구축하려고 합니다.

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

다) 고객사의 요구사항

1. IAM 생성

- 관리자 권한 IAM 생성
- 해당 IAM으로 AWS 관리

2. AZ 이중화

- 가용성 확보
- 서브넷 분리 -> 사설 네트워크 보호

3. Bastion Host

- 사설 네트워크의 인스턴스 -> 외부 인터넷 통신

4. Auto Scaling

- 리소스 사용량에 따른 인스턴스, Pod 조절

5. Load Balancer

- 가용성 증가 및 부하 분산
- ALB, NLB
- (K8S) LoadBalancer Service

6. Template

- 시작 템플릿
- 도커 이미지 빌드

● 프로젝트 수행요건

가) 설계 및 개발 요건

- 시스템의 물리적/논리적 Scale out/up에 대비하여 설계되어야 한다.
- WEB, WAS, DB 각각 서비스를 이중화 해 서비스의 가용성 확보를 최우선으로 한다.
- 라우터의 NAT기능을 통해 내/외부 간 통신은 가능케 하되, 직접적인 연결은 차단한다.
- NFS의 운영 ↔ DR 간 데이터 동기화를 통해 DR로 전환 시, 최대한의 가용성을 확보한다.

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

## 2. 프로젝트 추진 체계

- 프로젝트 참여 인력 총괄표

성명	소속	역할	담당업무
김학남	한국정보교육원	Project Leader	PM, VPN 구성
김경태	한국정보교육원	Project Assistant	AWS 구축
사희경	한국정보교육원	Project Assistant	On premise 구축

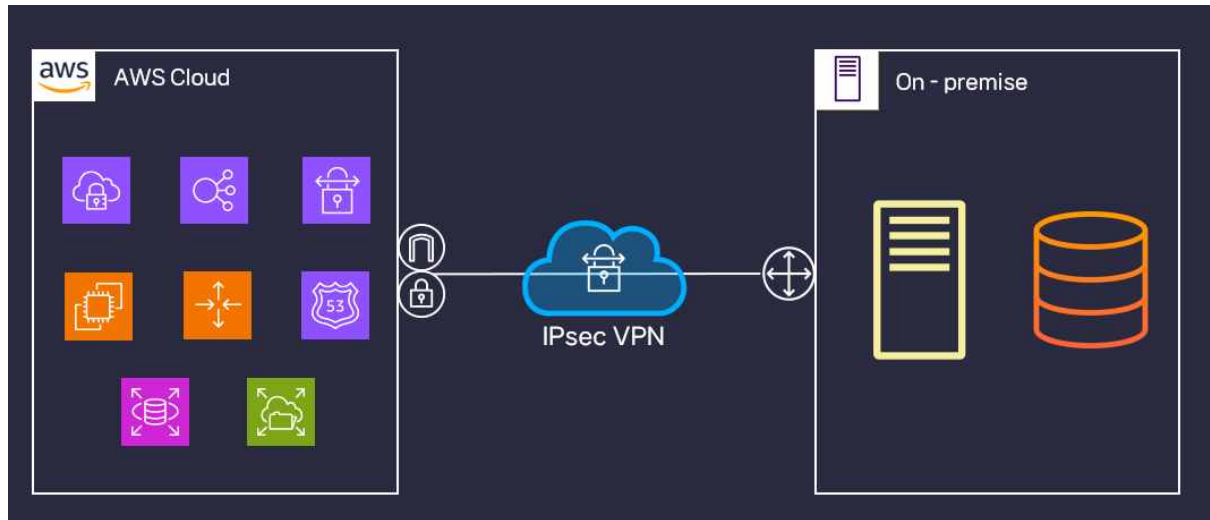
- 참여 인력 업무 분장

업무명	업무내용
PM	<ul style="list-style-type: none"> <li>- 프로젝트 수행 관리 및 책임</li> <li>- 프로젝트 범위, 인원, 일정, 결과 보고</li> <li>- 프로젝트 진행 상황에 따른 계획 조정</li> <li>- 기타 서류, 보고서 작성 및 발표</li> </ul>
AWS 구축	<ul style="list-style-type: none"> <li>- 네트워크 토폴로지 구성</li> <li>- IAM 계정 생성</li> <li>- VPC 생성 및 네트워크 구축</li> <li>- 3 Tier Architecture 구축 (EC2 Instance / EFS 설정)</li> <li>- Auto Scaling / Load Balancer 구축 (health check 설정)</li> </ul>
On premise 구축	<ul style="list-style-type: none"> <li>- 네트워크 토폴로지 구성</li> <li>- Kubernetes 설치</li> <li>- 3 Tier Architecture 구축</li> <li>- Deployment / Service를 위한 Yaml 파일 작성</li> </ul>
VPN을 통한 연동	<ul style="list-style-type: none"> <li>- 네트워크 토폴로지 구성</li> <li>- FRR 설치</li> <li>- ipsec 설치 및 환경설정(libreswan) -&gt; BGP 연결</li> <li>- VPN 구성 및 연결</li> </ul>

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

### 3. 세부 프로젝트 내용

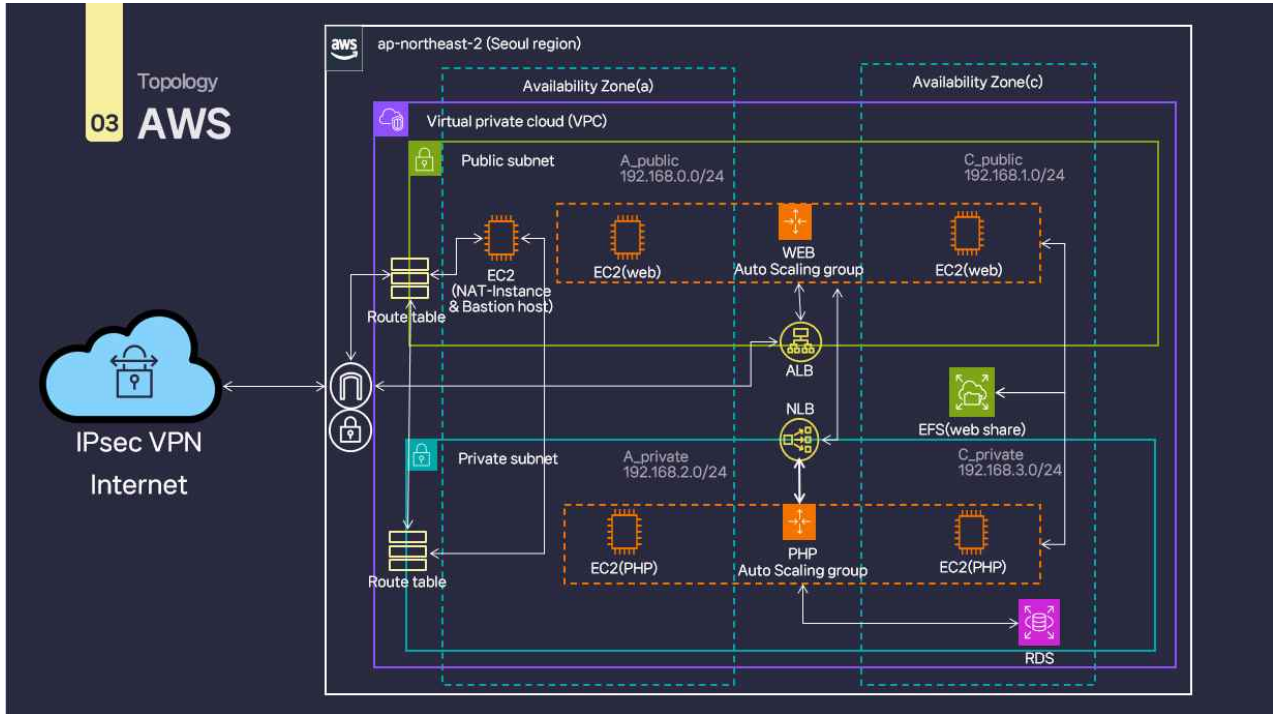
- 전체 구성도



- 가) vlan, inter-vlan 설정으로 운영 VMWARE 두 대의 IP를 다르게 구성(상호 통신 가능)
- 나) Prod#1, #2 중 한 Node에 장애가 발생해도 다른 한 Node에서 처리 가능하도록 구성
- 다) NAT를 통해 한국정보교육원에서 WEB 접속이 가능하도록 설정
- 라) DR용 전용회선을 사용해 운영서버 All Fault 시, DR Node를 통해 접속 가능하도록 처리

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

● AWS 구성도



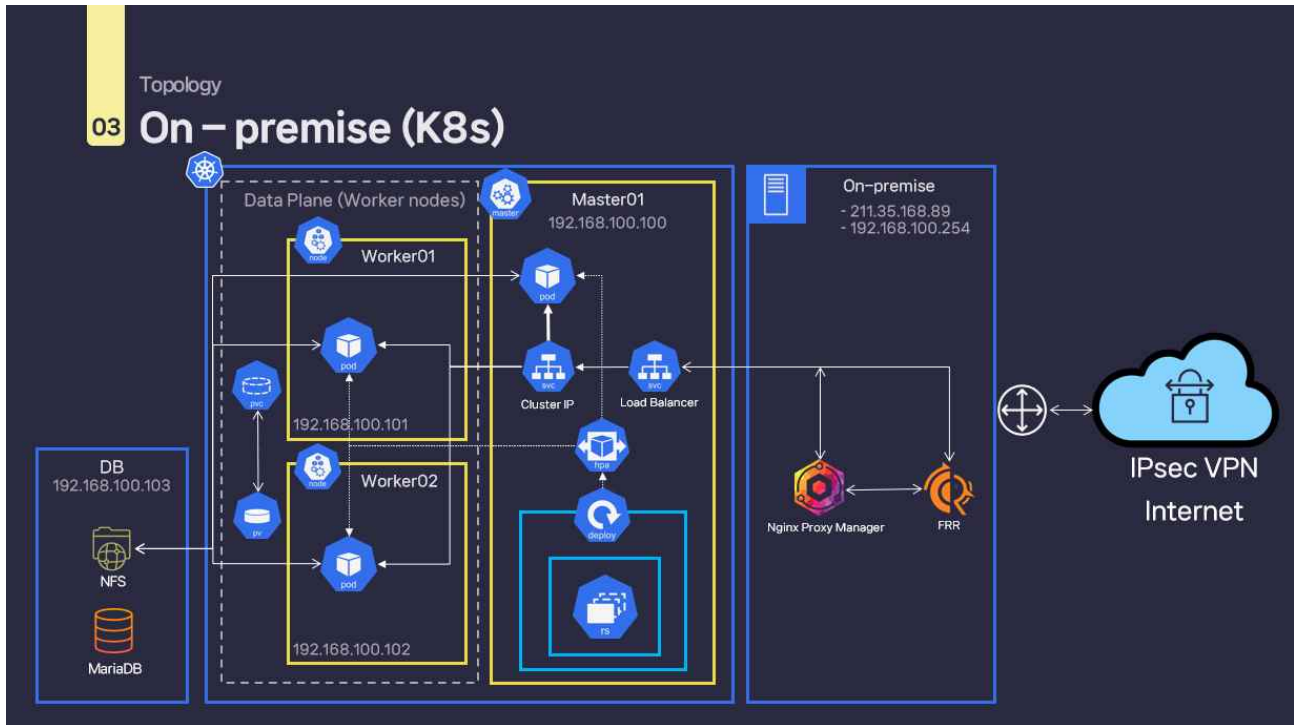
AWS 상세구축

1. 서울 리전에서 2개의 a와 c 프리티어 가용영역(지속적 서비스 운영을 위한 가용영역 분리)과 4개의 서브넷구성으로 퍼블릭과 프라이빗으로 구성
2. 퍼블릭에는 인터넷게이트웨이 생성으로 인터넷 연결 및 프라이빗에는 NAT 인스턴스를 생성하여 베스천호스트로 사용하여 인터넷 연결
3. 오토 스케일링을 통해 인스턴스 자동 생성 및 로드밸런싱을 통한 부하분산
4. RDS를 통해 데이터베이스를 구축
5. EFS를 통해 파일시스템 공유하여 서비스를 구축



프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

- ON-Premise (K8s)  
가) 물리서버 구성 현황



온프레미스 상세구축

1. 마스터노드와 워커노드 두 대를 두어 파드 생성
2. 각 파드를 NFS로 분리된 DB서버와 연동시켜 서비스 구축 및 FRR를 통해 라우팅
3. 각 서비스들은 NGINX PROXY 매니저를 통해 외부에서 접근 가능

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

사용장비 및 소프트웨어

※ 설치정보

서버구축	Hypervisor OS	VMware v17.0.1
	가상 라우터	FRR v7.5.1
	Server OS	Rocky Linux 8.8
WEB	NGINX Proxy Manager	NGINX Proxy Manager v.2.10.4
	NGINX	NGINX v.1.25.2
	PHP	PHP v8.2.0
Service	DB	MariaDB v10.6.15
		Nextcloud
		Gnuboard
		Wordpress
	DNS	Gabia
		CloudFlare

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

#### 사용장비 및 소프트웨어

##### ※ 설치정보

컨테이너	Docker v24.0.6
	Docker compose v2.21.0
	Kubernetes v1.28.2
AWS	IAM
	VPC      VPN
	EC2
	Route53
	Auto Scaling Group    EC2
	RDS
	EFS
	Application Load Balancer EC2
	Network Load Balancer   EC2E
	Cloudwatch
	VPN

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS	IAM
-----	-----

The screenshot displays the AWS IAM console interface. At the top, the breadcrumb navigation shows 'IAM > 사용자 그룹'. The main heading is '사용자 그룹 (1) 정보' (User Group (1) Information). Below this, there's a search bar and a table listing the group 'SK2\_User' with a status of '생성됨' (Created) and a creation time of '17분 전' (17 minutes ago). The second part of the screenshot shows the 'SK2\_User' group details page. It includes a '요약' (Summary) section with the group name 'SK2\_User', creation time 'September 15, 2023, 16:39 (UTC+09:00)', and ARN 'arn:aws:iam::806200757969:group/SK2\_User'. The '권한' (Permissions) tab is selected, showing a list of permissions. A red box highlights the first three permissions: 'AdministratorAccess', 'AmazonEC2FullAccess', and 'IAMFullAccess'. The table also shows the type of each permission (e.g., 'AWS 관리형 - 직무' for AdministratorAccess) and a brief description.

- IAM 사용자 그룹을 생성하여 관리자 권한을 추가
- 그룹 생성 및 권한 추가
  - 시스템 관리 및 생성을 하기 위해 관리자 권한 부여

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS	IAM
-----	-----

이 그룹의 사용자 (2)

사용자 이름	그룹	마지막 활동	생성 시간
skangjae		1	5분 전
nahnami		1	8분 전

- 해당 그룹에 사용자 계정 두 개를 추가하여 작업
- 2명의 사용자 계정 생성 후 그룹 추가
  - 권한이 추가 된 그룹에 사용자 추가
  - 각각의 사용자 계정에 MFA 설정을 통한 보안 강화

AWS	VPC
-----	-----

Name	VPC ID	상태	IPv4 CIDR
sk-vpc	vpc-083d35d8053b68a4a	Available	192.168.0.0/22

- 네트워크 격리 및 보안을 위해 VPC 생성
- VPC 생성
  - 네트워크 격리 및 보안
  - 고가용성 및 확장성

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS	VPC
-----	-----



서브넷 (4) [정보](#)

Find resources by attribute or tag

<input type="checkbox"/>	Name	서브넷 ID	상태	VPC	IPv4 CIDR	가용 영역	퍼블릭 IPv4 주소 자동 할당
<input type="checkbox"/>	a_public	subnet-05a7b43ca72b43609	Available	vpc-083c15c8053968a4a   sk-vpc	192.168.0.0/24	ap-northeast-2a	예
<input type="checkbox"/>	c_public	subnet-0f74016bc3f4f1a0d	Available	vpc-083c15c8053968a4a   sk-vpc	192.168.1.0/24	ap-northeast-2c	예
<input type="checkbox"/>	a_private	subnet-0688319c97f3af212	Available	vpc-083c15c8053968a4a   sk-vpc	192.168.2.0/24	ap-northeast-2a	아니요
<input type="checkbox"/>	c_private	subnet-0f6dc1dc01b0924e1	Available	vpc-083c15c8053968a4a   sk-vpc	192.168.3.0/24	ap-northeast-2c	아니요

가용영역을 퍼블릭 및 프라이빗으로 이중화하여 가용성을 확보

- 서브넷을 4개로 분리하여 사설 네트워크 보호

#### Availability Zone


- Public / Private 각각의 2개의 가용영역을 사용한 가용영역 이중화로 가용성 확보

#### Subnet

- Public / Private 서브넷을 분리하여 사설 네트워크 보호

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

- 네트워크 구성 현황  
가) Router 구성

AWS	VPC
	
<p>퍼블릭 가용 영역에 공인 IP 할당 - 인터넷 액세스 및 외부 서비스 통합</p>	

AWS	인터넷 게이트웨이
	
<p>인터넷 게이트웨이 생성 - 인터넷 디폴트 게이트웨이 역할 - 인터넷 액세스 제공 - 퍼블릭 서비스 제공</p>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

Router

라우팅 테이블 (2) 정보

Find resources by attribute or tag

<input type="checkbox"/>	Name	라우팅 테이블 ID	명시적 서브넷 연결
<input type="checkbox"/>	private	rtb-0789a499339ce4c3a	2 서브넷
<input type="checkbox"/>	public	rtb-0cb10c7e6990c9bfc	2 서브넷

VPC > 라우팅 테이블 > rtb-0cb10c7e6990c9bfc  
**rtb-0cb10c7e6990c9bfc / public**

세부 정보

라우팅 테이블 ID

rtb-0cb10c7e6990c9bfc

기본

예

명시적 서브넷 연결

2 서브넷

VPC

vpc-083d35d8053b68a4a | sk-vpc

소유자 ID

456050757969

라우팅 | 서브넷 연결 | 엣지 연결 | 라우팅 전파 | 태그

라우팅 (2)
 라우팅 필터

모두

대상	대상	상태
0.0.0.0/0	igw-0a4fb90456022758e	활성
192.168.0.0/22	local	활성

Internet - gateway

퍼블릭 라우팅 테이블에 등록하여 인터넷과 연결되도록 설정

- Public 서브넷들은 인터넷게이트웨이를 통해 외부와 연결



프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS

보안 그룹

보안 그룹 (9) 정보

🔄

작업 ▼

보안 그룹을 CSV로 내보내기 ▼

보안 그룹 생성

🔍 보안 그룹 필터링

< 1 > ⌂

<input type="checkbox"/>	Name	🔍	보안 그룹 ID	🔍	보안 그룹 이름	🔍	VPC ID	🔍	설명	🔍	소유자	🔍	전세로드 규칙 수	🔍	액세스포인트
<input type="checkbox"/>	-		sg-0f874ff95ea96d07b9		HTTP		vpc-083d35d8053b68a4a ...		permit-http		456050757969		1 권한 할책		1 권한 할책
<input type="checkbox"/>	-		sg-006b049f18dfef308		DB		vpc-083d35d8053b68a4a ...		permit-db		456050757969		1 권한 할책		1 권한 할책
<input type="checkbox"/>	-		sg-07ca8c9eb68a9f12c		SSH		vpc-083d35d8053b68a4a ...		permit-ssh		456050757969		1 권한 할책		1 권한 할책
<input type="checkbox"/>	-		sg-0c3cc9cda3cd02d39		PHP		vpc-083d35d8053b68a4a ...		permit-php		456050757969		1 권한 할책		1 권한 할책
<input type="checkbox"/>	-		sg-0d2a8b7198e5a0895		ICMP		vpc-083d35d8053b68a4a ...		permit-icmp		456050757969		1 권한 할책		1 권한 할책
<input type="checkbox"/>	-		sg-0c8ac62108880a655		default		vpc-083d35d8053b68a4a ...		default VPC security gr...		456050757969		1 권한 할책		1 권한 할책
<input type="checkbox"/>	-		sg-052e3d504b1b93c58		NAT		vpc-083d35d8053b68a4a ...		permit-nat		456050757969		1 권한 할책		1 권한 할책
<input type="checkbox"/>	-		sg-01a1b77b2bbf7dae7		NFS		vpc-083d35d8053b68a4a ...		permit-nfs		456050757969		1 권한 할책		1 권한 할책
<input type="checkbox"/>	-		sg-09466ba5a351a8fbf		HTTPS		vpc-083d35d8053b68a4a ...		permit-https		456050757969		1 권한 할책		1 권한 할책

키 페어 (1) 정보

🔍 검색

☐

이름

▼

☐

유형

▼

☐

생성 원료

▼

☐

지문

▼

☐

ID

<input type="checkbox"/>	4_Group_key	rsa	2023/09/15 17:20 GMT+9	f8:ce:09:25:47:43:fc:a8:58:39:d0:5f:29:2...	key-0efebb4bfd5558aa8
--------------------------	-------------	-----	------------------------	---	-----------------------

인스턴스 생성에 필요한 보안그룹을 등록 , 키페어 생성

보안그룹 생성


- 네트워크 보안 제어
- 방화벽 역할
- 로그 및 감사

키페어 생성

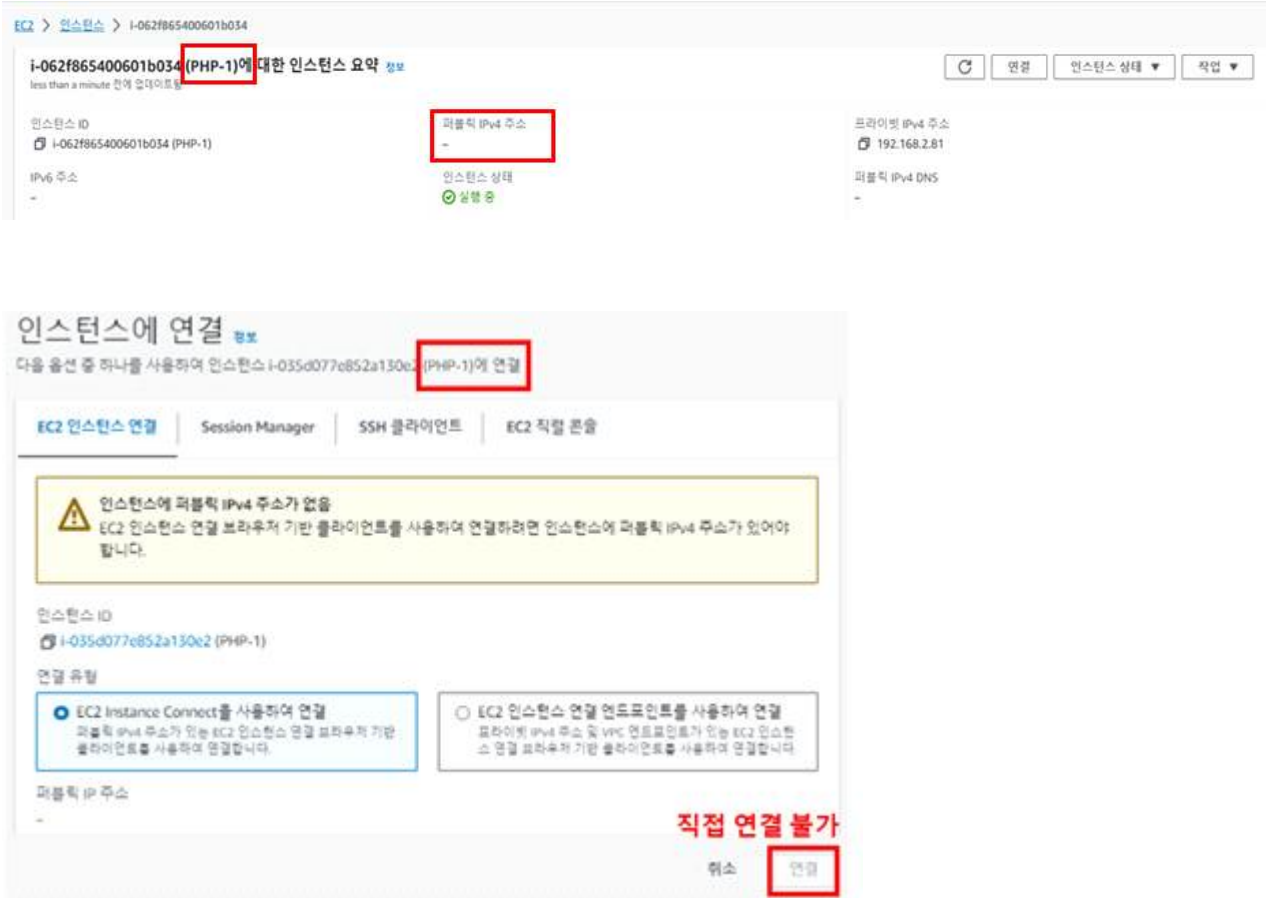
- 인스턴스 액세스 제어
- SSH & RDP 액세스
- 암호화 키 관리

페이지 17 / 78

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS	EC2
 <p>The screenshot displays the AWS Management Console for a NAT instance (i-07ffd9f7f64393d2b). A red box highlights the 'Public IPv4 주소' (Public IPv4 address) field, which shows '52.78.236.220'. Below this, a terminal window shows the output of a 'ping 8.8.8.8' command, indicating successful connectivity with response times around 27ms. The terminal output is as follows:</p> <pre>NAT ~ \$ ping 8.8.8.8 PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data: 64 bytes from 8.8.8.8: icmp_seq=1 ttl=104 time=27.6 ms 64 bytes from 8.8.8.8: icmp_seq=2 ttl=104 time=27.7 ms 64 bytes from 8.8.8.8: icmp_seq=3 ttl=104 time=27.6 ms 64 bytes from 8.8.8.8: icmp_seq=4 ttl=104 time=27.6 ms 64 bytes from 8.8.8.8: icmp_seq=5 ttl=104 time=27.7 ms</pre> <p>Below the terminal window, the NAT instance details are shown:</p> <pre>i-07ffd9f7f64393d2b (NAT-instance) PublicIPs: 52.78.236.220 PrivateIPs: 192.168.0.168</pre>	
Public 인스턴스는 인터넷 게이트웨이 라우팅으로 인해 외부와 연결이 됨	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS	EC2
 <p>인스턴스에 연결</p> <p>다음 옵션 중 하나를 사용하여 인스턴스 i-035d077e852a130e2 (PHP-1)에 연결</p> <p>인스턴스 ID: i-035d077e852a130e2 (PHP-1)</p> <p>연결 유형</p> <p>EC2 Instance Connect를 사용하여 연결 퍼블릭 IPv4 주소가 있는 EC2 인스턴스 연결 브라우저 기반 클라이언트를 사용하여 연결합니다.</p> <p>EC2 인스턴스 연결 엔드포인트를 사용하여 연결 프라이빗 IPv4 주소 및 VPC 엔드포인트가 있는 EC2 인스턴스 연결 브라우저를 통한 클라이언트를 사용하여 연결합니다.</p> <p>직접 연결 불가</p> <p>취소 연결</p>	
<p>Private 인스턴스는 퍼블릭 IP를 할당받지 못 했기 때문에 직접 접속이 되지 않고 인터넷 연결 또한 되지 못 함</p>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS	EC2 – NAT Instance
-----	--------------------

소스/대상 확인 변경

소스/대상 확인은 인스턴스가 송수신되는 모든 트래픽의 소스 또는 대상인지 확인합니다. 각 EC2 인스턴스는 기본적으로 소스 및 대상 확인을 수행합니다. [자세히 알아보기](#)

인스턴스 ID  
i-0f5a65a2cfa771de1 (NAT-instance)

네트워크 인터페이스  
eni-0613b2d14d89d0df9

소스/대상 확인  
소스 또는 대상이 그 자체가 아닐 때 인스턴스가 트래픽을 송수신할 수 있도록 하려면 중지합니다.

☒ 중지

취소 저장

```
NAT ~ $ sudo iptables -nL POSTROUTING -t nat -v
Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source         destination
  168 12344 MASQUERADE all  --  *      eth0    0.0.0.0/0      0.0.0.0/0
    0    0 MASQUERADE all  --  *      eth0    0.0.0.0/0      0.0.0.0/0
NAT ~ $ sudo cat /proc/sys/net/ipv4/ip_forward
1
NAT ~ $
```

i-07ffd9f7f64393d2b (NAT-instance)

PublicIPs: 52.78.236.220 PrivateIPs: 192.168.0.168

NAT 인스턴스를 생성하여 Private 인스턴스에서도 인터넷 연결 뿐 만 아니라 Bastion host로 접속이 가능하도록 설정

NAT instance

- NAT instance를 Bastion host로 사용
- Private cloud의 인터넷 연결 가능
- 사설 네트워크 보안 강화

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS
라우팅 테이블

[VPC](#) > [라우팅 테이블](#) > rtb-0789a499339ce4c3a  
rtb-0789a499339ce4c3a / private

---

**세부 정보** 정보

라우팅 테이블 ID  
rtb-0789a499339ce4c3a

VPC  
vpc-081d35d80531b68a4a | [sk-vpc](#)

기본  
아니요

소유자 ID  
456050757969

공식적 서브넷 관련  
[공식문서](#)

---

[라우팅](#)
[서브넷 연결](#)
[넷지 연결](#)
[라우팅 통과](#)
[태그](#)

**라우팅 (2)**

모두 ▼

대상	대상	상태
0.0.0.0/0	<div style="border: 2px solid red; padding: 2px; display: inline-block;">eni-0a775d421e4c6d8e7</div> <span style="color: red; font-weight: bold; font-size: 1.2em;">NAT- instance</span>	<div style="color: green; font-size: 1.2em;">↻</div> 활성
192.168.0.0/22	local	<div style="color: green; font-size: 1.2em;">↻</div> 활성

```

NAT - $
NAT - $
NAT - $ ssh -i 4_Group_key.pem ec2-user@192.168.2.202
Last login: Fri Sep 22 01:33:21 2023 from ip-192-168-0-168.ap-northeast-2.compute.internal

  _  _  _  _  _
 _/  _/  _/  _/  _/  Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
6 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-2-202 ~]$ PS1="PHP-1 \W $ "
PHP-1 ~ $ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_seq=1 ttl=103 time=28.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=103 time=28.1 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=103 time=28.0 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=103 time=28.1 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=103 time=28.0 ms

```

인터넷 연결

i-07ffd9f7f64393d2b (NAT-instance)  
PublicIPs: 52.78.236.220 PrivateIPs: 192.168.0.168

NAT 인스턴스를 Private Routing Table에 등록하여 인터넷 연결

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS

로드 밸런서

EC2 > 로드 밸런서

로드 밸런서 (1/2)



Elastic Load Balancing은 수신 트래픽의 분파에 따라 자동으로 로드 밸런서 용량을 확장합니다.

🔍 검색 또는 값을 기준으로 필터링

	이름	DNS 이름	상태	VPC ID	가용 영역	유형
<input checked="" type="checkbox"/>	<a href="#">PHP-NLB</a>	PHP-NLB-6b216577994fe...	🟢 활성	vpc-083d35d8053b68a4a	2. 가용 영역	network
<input type="checkbox"/>	<a href="#">WEB-ALB</a>	WEB-ALB-1986280372.ap...	🟢 활성	vpc-083d35d8053b68a4a	2. 가용 영역	application

웹은 퍼블릭 가용영역에서 어플리케이션 로드밸런서로 생성하여 트래픽 분산,  
PHP는 프라이빗 가용영역에서 네트워크 로드밸런서를 생성하여 트래픽을 분산하여 가용성 증가

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS	대상 그룹, 상태 검사
<div> <div>  </div> <div>  </div> </div> <div> <ul style="list-style-type: none"> <li>- 대상 그룹을 사용하여 애플리케이션의 신뢰성과 가용성 향상</li> <li>- 트래픽을 효과적으로 관리</li> <li>- 상태 검사 경로 : /health.html</li> </ul> </div>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

## 대상 그룹, 상태 검사

**대상 그룹**

대상 그룹 (1/2) 정보

Q 대상 그룹 검색 또는 필터링

	이름	ARN	포트	프로토콜	대상 유형	로드 밸런서	VPC ID
<input type="checkbox"/>	PHP-ALB-GROUP	arn:aws:elasticloadbalancing::ap-northeast-2:aws-elasticloadbalancing:targetgroup/php-alb-group/8c0e3d35-d805-5b68-a4a	9000	TCP	인스턴스	PHP-ALB	vpc-083d35d8055b68a4a
<input checked="" type="checkbox"/>	WEB-ALB-GROUP	arn:aws:elasticloadbalancing::ap-northeast-2:aws-elasticloadbalancing:targetgroup/web-alb-group/8c0e3d35-d805-5b68-a4a	80	HTTP	인스턴스	WEB-ALB	vpc-083d35d8055b68a4a

**대상 그룹: WEB-ALB-GROUP**

세부 정보 | **대상** | 모니터링 | 상태 검사 | 속성 | 태그

등록된 대상 (2)

Q 대상 또는 인스턴스를 기준으로 리소스 필터링

	인스턴스 ID	이름	포트	영역	상태 확인
<input type="checkbox"/>	i-0c6325f9ce2769720	web-Auto2	80	ap-northeast-2a	healthy
<input type="checkbox"/>	i-09874c5573c1e090d	web-Auto1	80	ap-northeast-2c	healthy

health check를 통해서 정상 작동되지 않는 인스턴스로는 트래픽을 보내지 않도록 설정
WEB-ALB-GROUP 성공 코드 - 대상으로부터 응답 성공 확인 코드 - 200-399번 코드 지정 -> healthy

WEB-ALB-GROUP 성공 코드  
- 대상으로부터 응답 성공 확인 코드  
- 200-399번 코드 지정 -> healthy



프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

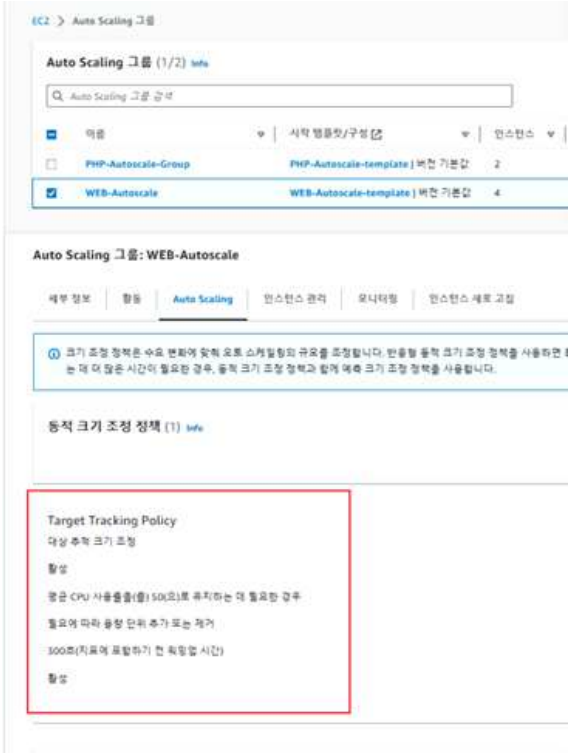

AWS

AMI

The screenshot shows the AWS Management Console. The top panel displays the 'Amazon Machine Images(AMI)' list with two entries: 'ami-0705898397baec048' (WEB) and 'ami-04f9ba2617885fe3' (PHP). The bottom panel shows the 'Launch Template' configuration for 'PHP-AutoScale-template', where the 'AMI' is set to 'ami-04f9ba2617885fe3'. A third panel shows the 'Auto Scaling Group' configuration for 'PHP-AutoScale-Group', where the 'Launch Template' is set to 'PHP-AutoScale-template'. Red arrows indicate the flow from the AMI selection to the Launch Template configuration and then to the Auto Scaling Group configuration.

오토 스케일링 생성으로 이미지를 생성하고 시작 템플릿을 생성함으로써 이루어 짐

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS	Auto Scaling 그룹
	
	
<ul style="list-style-type: none"> <li>- 오토 스케일링 조절 정책으로 CPU 사용률에 따라 인스턴스 자동 추가 및 제거가 되도록 설정</li> <li>- 조절 정책으로 평균 CPU 사용률을 50% 이상 초과 할 시 2개에서 최대 4개까지 늘어나도록 설정</li> </ul>	





프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS

AMI version

Amazon Machine Images(AMI) (3) 정보

Name	AMI ID	AMI 이름	플랫폼	버전	소유자	표시 여부	상태
	ami-0703898397baec048	WEB	456050757969/WEB	456050757969	프라이빗	사용 가능	
	ami-06951957ad5185d4a	PHP-DB	456050757969/PHP-DB	456050757969	프라이빗	사용 가능	
	ami-04f9ba26bf7885fe3	PHP	456050757969/PHP	456050757969	프라이빗	사용 가능	

Autoscaling 버전 생성

기본 버전 설정

Which template version would you like to make the default version?

Template: PHP-Autoscale-template (H-Dec2dad6b3a5e6958)

Template version: 2

Set as default version

편집 PHP-Autoscale-Group

시작 템플릿

버전: Default (2)

성능 개선 및 최적화를 시키기 위해 버전을 변경하여 오토 스케일링에 적용

Auto scaling 버전 변경

- 새로운 기능과 성능 개선 및 최적화
- 버그 수정과 안정성 향상
- 호환성 유지

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS	File System
<div> <div> <div>Amazon EFS &gt; 파일 시스템 &gt; fs-0ae6e3dd11a633f6b</div> <div>web-share (fs-0ae6e3dd11a633f6b)</div> </div> <div> <div>일반</div> <div> <div> <div>상태 코드</div> <div>정상</div> </div> <div> <div>작동 방식</div> <div>활성화됨</div> </div> <div> <div>접근 권한</div> <div>5a42f730-0b9d-4f02-86f3-35864c4da4bb (aws/elasticfilesystem)</div> </div> <div> <div>파일 시스템 상태</div> <div>사용 가능</div> </div> <div> <div>DNS 이름</div> <div>fs-0ae6e3dd11a633f6b.efs.ap-northeast-2.amazonaws.com</div> </div> </div> </div> </div> <div> <pre> WEB1 ~ \$ df -Th Filesystem      Type      Size  Used Avail Use% Mounted on devtmpfs        devtmpfs  468M   0    468M   0% /dev tmpfs           tmpfs     477M   0    477M   0% /dev/shm tmpfs           tmpfs     477M 732K  476M   1% /run tmpfs           tmpfs     477M   0    477M   0% /sys/fs/cgroup /dev/xvda1      xfs       8.0G  1.9G  6.2G  24% / 127.0.0.1:/     nfs4      8.0E  106M  8.0E   1% /usr/share/nginx/html tmpfs           tmpfs     96M   0    96M   0% /run/user/0 tmpfs           tmpfs     96M   0    96M   0% /run/user/1000  PHP-1 ~ \$ df -Th Filesystem      Type      Size  Used Avail Use% Mounted on devtmpfs        devtmpfs  468M   0    468M   0% /dev tmpfs           tmpfs     477M   0    477M   0% /dev/shm tmpfs           tmpfs     477M 612K  476M   1% /run tmpfs           tmpfs     477M   0    477M   0% /sys/fs/cgroup /dev/xvda1      xfs       8.0G  2.0G  6.1G  24% / 127.0.0.1:/     nfs4      8.0E  106M  8.0E   1% /usr/share/nginx/html tmpfs           tmpfs     96M   0    96M   0% /run/user/1000 </pre> </div>	
<p>파일시스템 공유를 위해 EFS를 생성 및 연동</p> <p>EFS</p> <ul style="list-style-type: none"> <li>- EFS를 이용해 NFS 사용</li> <li>- 추후 로드밸런싱을 이용한 웹서버를 구축하기 위해 NGINX의 루트 디렉터리 /usr/share/nginx/html을 NFS mount</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS

RDS, 파라미터 그룹

데이터베이스 생성

데이터베이스 생성 방식 선택

표준 방식

가용성, 보안, 백업 등 복제 및 장애에 대한 용량을 제공하여 기존 DB 용량을 확장합니다

분리된 방식

유량 요청 시에 수평을 늘릴 수 있는 용량 증가를 위한 리스본스 생성 및 운영을 지원합니다

엔진 옵션

엔진 유형 선택

Aurora MySQL Compatible

Aurora PostgreSQL Compatible

MySQL

MariaDB

PostgreSQL

Oracle

Microsoft SQL Server

추가 옵션

Aurora RDS 호환화된 스키를 지원하는 보안 프로시저

Aurora RDS 호환화된 스키를 지원하는 보안 프로시저

엔진 버전

MariaDB 10.6.14

RDS > 데이터베이스

데이터베이스 (1)

데이터베이스를(들) 기준으로 필터링

DB 식별자

상태

역할

엔진

리전 및 AZ

RDS > 파라미터 그룹 > 파라미터 그룹 생성

파라미터 그룹 세부 정보

파라미터 그룹 패밀리

이 DB 파라미터 그룹을 적용할 DB 패밀리

mariadb10.6

그룹 이름

DB 파라미터 그룹의 식별자

db\_parameter

설명

DB 파라미터 그룹에 대한 설명

parameter\_group

고가용성 및 자동 백업을 위해 RDS 생성

RDS 생성

- 고가용성 및 자동 백업을 위해 RDS 사용

- 분리된 DB 서버 구축으로 DB 안정성 및 보안 강화

페이지 31 / 78



프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS	MariaDB
<pre> 52 tomcat9          available [ =stable ] 53 unbound1.13      available [ =stable ] 54 †mariadb10.5=latest enabled    [ =stable ] </pre> <pre> [ec2-user@ip-192-168-3-178 ~]\$ mysql -u admin -p -h db.meongchung.shop Enter password: Welcome to the MariaDB monitor.  Commands end with ; or \g. Your MariaDB connection id is 206 Server version: 10.6.14-MariaDB managed by https://aws.amazon.com/rds/  Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  Type 'help;' or '\h' for help. Type '\c' to clear the current input statement. </pre> <pre> MariaDB [(none)]&gt; show databases; +-----+   Database   +-----+   gnuboard     information_schema     innodb     mysql     performance_schema     sys     wordpress   +-----+ 7 rows in set (0.001 sec)  MariaDB [(none)]&gt; select user,host,password from mysql.user; +-----+-----+-----+   User   Host   Password   +-----+-----+-----+   mariadb.sys   localhost   *A6258D0A818748EED2F480CF4F45971C24A13E40     rdsadmin   localhost   *EA8D87B5453F22FC060E38F7F759A2A37DC325E1     admin   %   *A4B6157319038724E3560894F7F932C8886EBFCF     wordpress   %   *A4B6157319038724E3560894F7F932C8886EBFCF     gnuuser   localhost   *A4B6157319038724E3560894F7F932C8886EBFCF     gnuuser   %   *A4B6157319038724E3560894F7F932C8886EBFCF   +-----+-----+-----+ 6 rows in set (0.001 sec) </pre>	
<p>- Private 가용영역에 DB를 설치해서 계정 생성 및 권한 설정을 작업을 할 수 있게 함</p>	



프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS

Route53

Route 53 > 호스팅 영역 > meongchung.shop

meongchung.shop

영역 삭제

레코드 테스트

원래 로컬 구성

호스팅 영역 세부 정보

호스팅 영역 편집

레코드 (5)

DNSSEC 시험

호스팅 영역 태그 (0)

레코드 (5) 정보

Automatic 레코드는 최상의 결과에 최적화된 현재 값에 동의합니다. 모든 변경 사항은 로컬 (awscli)로 적용합니다.

레코드를 새로고침

레코드 삭제

영역 파일 가져오기

레코드 생성

속성 또는 값을 기준으로 레코드를 필터링

필터

라우팅 정책

분할

레코드 이름

유형

라우팅 ...

자본...

별칭

값/브레이크 라우팅 대상

TTL(초)

현재 값...

대상 ...

계...

meongchung.shop

NS

단순

-

아니요

ns-540.awsdns-03.net, ns-1356.awsdns-41.org, ns-1356.awsdns-23.com, ns-2036.awsdns-62.co.uk

172800

-

-

-

meongchung.shop

SOA

단순

-

아니요

ns-540.awsdns-03.net, awsd...

900

-

-

-

gabia, 도메인 관리

전체 도메인 01개

도메인 정보 변경

내임서버 설정

도메인 정보 변경

meongchung.shop

도메인 보안

예약 도메인 관리

관심 도메인

상표 보호

구분

호스팅

구분

호스팅

1차

ns-540.awsdns-03.net

2차

ns-1356.awsdns-41.org

- 가비아에서 구매한 도메인을 Route53에서 관리하기 위해 네임서버로 설정

페이지
 33
 /
 78

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS

RDS 엔드포인트

RDS > 데이터베이스 > db-1

db-1

요약

DB 식별자

db-1

CPU

2.02%

상태

사용 가능

역할

인스턴스

현재 활동

0 연결

엔진

MariaDB

연결 및 보안

모니터링

로그 및 이벤트

구성

유지 관리 및 백업

태그

연결 및 보안

엔드포인트 및 포트

엔드포인트

db-1.cf9wohmgsmp6.ap-northeast-2.rds.amazonaws.com

네트워킹

가용 영역

ap-northeast-2c

보안

VPC 보안 그룹

default [sg-Oc8ac62108880a655]

활성

Route 53 > 호스팅 영역 > meongchung.shop

리소스

meongchung.shop

정보

호스팅 영역 세부 정보

레코드(5)

DNSSEC 서명

호스팅 영역 태그(0)

레코드 (5) 정보

Automatic 모드는 최상의 결과에 최적화된 현재 값에 동작합니다. 모드를 변경하려면 설정(settings)으로 이동합니다.

검색 또는 값을 기준으로 레코드 필터링

1개 일치

CNAME

라우팅 정책

표시

유형 = CNAME

필터 지우기

레코드 이름

유형

라우팅 ...

차별...

병합

값/트래픽 라우팅 대상

db.meongchung.shop

CNAME

단순

-

아니요

db-1.cf9wohmgsmp6.ap-nor...

- RDS 엔드포인트를 Route53 가상호스트에 등록하여 사용

페이지 34 / 78

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS

로드 밸런서 (ALB, NLB)

로드 밸런서 (2)

Elastic Load Balancing은 수신 트래픽의 분포에 따라 자동으로 로드 밸런서 용량을 확장합니다.

Q 속성 또는 값을 기준으로 필터링

<input type="checkbox"/>	이름	DNS 이름	상태	VPC ID	가용 영역	유형
<input type="checkbox"/>	<a href="#">WEB-ALB</a>	WEB-ALB-1986280372.ap...	🟢 활성	vpc-083d35d8053b68a4a	2 가용 영역	application
<input type="checkbox"/>	<a href="#">PHP-NLB</a>	PHP-NLB-6b216577994fe...	🟢 활성	vpc-083d35d8053b68a4a	2 가용 영역	network

Route 53 > 호스팅 영역 > meongchung.shop

meongchung.shop

영역 삭제 레코드 테스트 권역 로깅 구성

호스팅 영역 세부 정보

호스팅 영역 편집

레코드 (5)

DNSSEC 시험 호스팅 영역 태그 (0)

레코드 (5) 정보

Automatic 모드는 최상의 필터 결과에 최적화된 현재 값만 동작합니다. 모드를 변경하려면 설정(setting)으로 이동합니다.

Q 속성 또는 값을 기준으로 레코드 필터링

2개 일치 A 라우팅 정책 분할 2개 일치

유형 A

필터 지우기

<input type="checkbox"/>	레코드 이름	유형	라우팅 ...	차별...	병합	입/트래픽 라우팅 대상	TTL(초)	상태 확...	대상 ...	레...
<input type="checkbox"/>	nlb.meongchung.shop	A	단순	-	예	php-nlb-6b216577994fe7c6...	-	-	예	-
<input type="checkbox"/>	web.meongchung.shop	A	단순	-	예	dualstack.web-alb-19862803...	-	-	예	-

- 각 로드밸런서 DNS를 Route53 가상호스트에 등록하여 사용

Load Balancer : 로드밸런싱을 통해 가용성 증가 및 부하 분산

Application LB를 이용해

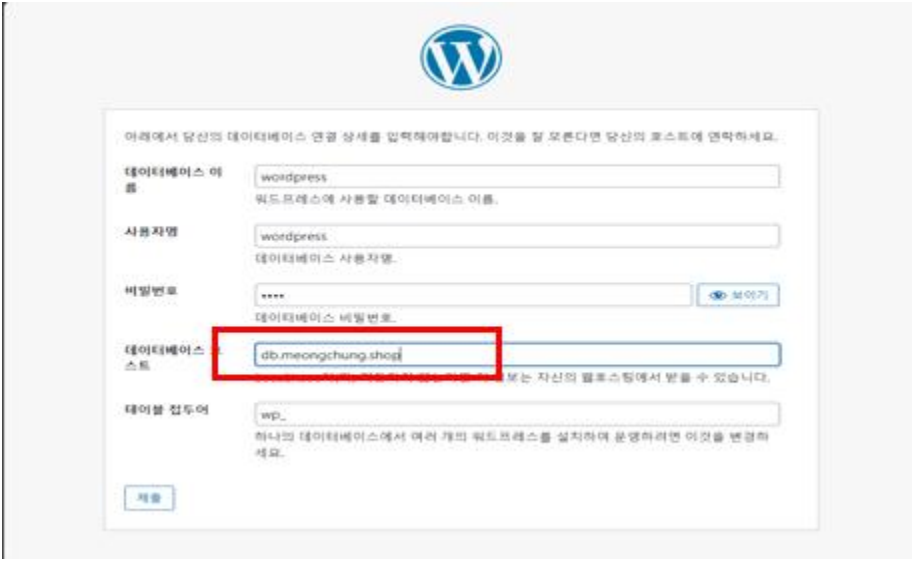
-> HTTP 로드밸런싱

Network LB를 이용해

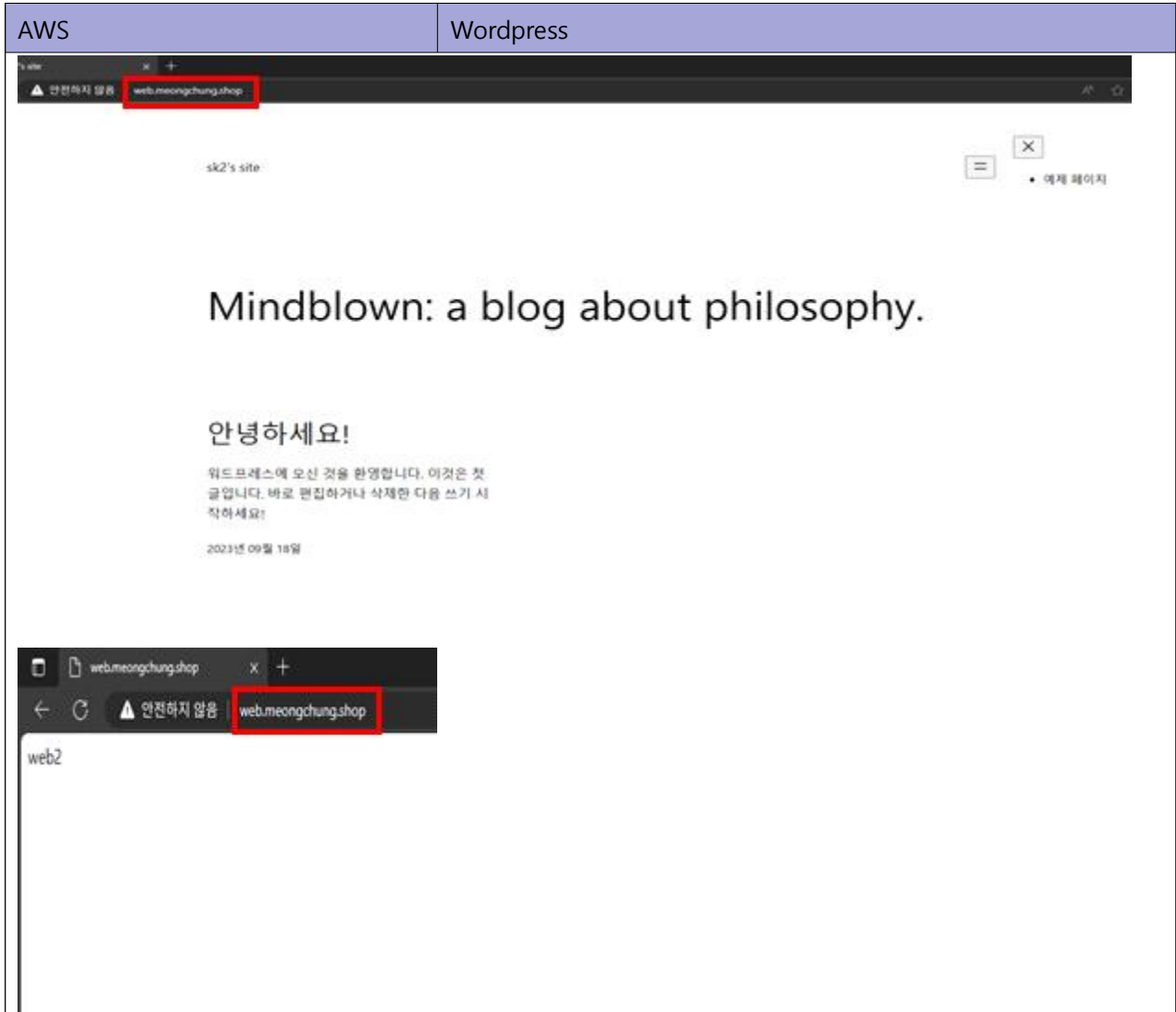
-> PHP용 로드밸런싱

페이지 35 / 78

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

AWS	Nginx - PHP 연동
<pre>location ~ [^/].php(/ \$) {     fastcgi_split_path_info ^(.+?\.php)(/.*)\$;     set \$path_info \$fastcgi_path_info;     fastcgi_index index.php;     include fastcgi_params;     fastcgi_pass nlb.meongchung.shop:9000;     fastcgi_param SCRIPT_FILENAME \$document_root\$fastcgi_script_name; }</pre>	
 <p>The image shows the WordPress installation database configuration screen. The 'Database Name' field is filled with 'wordpress'. The 'Database Username' field is filled with 'wordpress'. The 'Database Password' field is filled with '****'. The 'Database Hostname' field is filled with 'db.meongchung.shop', which is highlighted with a red box. The 'Database Prefix' field is filled with 'wp_'. There is a 'Save' button at the bottom left.</p>	
<ul style="list-style-type: none"> <li>- NLB 가상호스트를 적용하여 NGINX와 PHP를 연동</li> <li>- 워드프레스 설치 시 데이터 호스트를 RDS 가상호스트에 적용</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04



- ALB 가상호스트로 로드밸런싱하여 웹 접속

ALB 접속하는 Domain name

- Route53에 web.meongchung.shop 생성
- 해당 도메인으로 ALB 접근(WEB)
- NGINX Load Balancing
- 두 인스턴스로 부하 분산

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

● On Premise 상세 기술

ON-PREMISE	VM Partitioning, Resource
<div data-bbox="172 427 1423 1240"> </div>	
<div> <div>Partitioning</div> <div>20GB 용량기준</div> <ul style="list-style-type: none"> <li>- /root: 17GB</li> <li>- /boot: 1GB</li> <li>- /swap: 2GB</li> </ul> </div> <div> <div>Resource</div> <ul style="list-style-type: none"> <li>- CPU core: 2</li> <li>- Memory: 4096MB</li> <li>- Disk (SATA): 20GB</li> </ul> </div>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	VM 생성
<pre>[root@onpremise ~]# ip addr 1: lo: &lt;LOOPBACK,UP,LOWER_UP&gt; mtu 65536 qdisc noqueue state UNKN link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 inet 127.0.0.1/8 scope host lo valid_lft forever preferred_lft forever inet6 ::1/128 scope host valid_lft forever preferred_lft forever 2: ens160: &lt;BROADCAST,MULTICAST,UP,LOWER_UP&gt; mtu 1500 qdisc mq s link/ether 00:0c:29:e0:ab:a6 brd ff:ff:ff:ff:ff:ff altname enp3s0 inet 211.35.168.89/24 brd 211.35.168.255 scope global dynami valid_lft 6967sec preferred_lft 6967sec inet6 fe80::20c:29ff:fea0:aba6/64 scope link noprefixroute valid_lft forever preferred_lft forever 3: ens192: &lt;BROADCAST,MULTICAST,UP,LOWER_UP&gt; mtu 1500 qdisc mq s link/ether 00:0c:29:e0:ab:b0 brd ff:ff:ff:ff:ff:ff altname enp11s0 inet 192.168.100.254/24 brd 192.168.100.255 scope global nop valid_lft forever preferred_lft forever inet6 fe80::5e14:1a20:a977:64c6/64 scope link noprefixroute valid_lft forever preferred_lft forever</pre> <pre>[root@worker01 ~]# ip addr 1: lo: &lt;LOOPBACK,UP,LOWER_UP&gt; mtu 65536 qdisc no link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 inet 127.0.0.1/8 scope host lo valid_lft forever preferred_lft forever inet6 ::1/128 scope host valid_lft forever preferred_lft forever 2: ens160: &lt;BROADCAST,MULTICAST,UP,LOWER_UP&gt; mtu link/ether 00:0c:29:a0:b9:bd brd ff:ff:ff:ff:ff:ff altname enp3s0 inet 192.168.100.101/24 brd 192.168.100.255 valid_lft forever preferred_lft forever inet6 fe80::20c:29ff:fea0:b9bd/64 scope link valid_lft forever preferred_lft forever</pre> <pre>[root@0B ~]# ip addr 1: lo: &lt;LOOPBACK,UP,LOWER_UP&gt; mtu 65536 qdisc no link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 inet 127.0.0.1/8 scope host lo valid_lft forever preferred_lft forever inet6 ::1/128 scope host valid_lft forever preferred_lft forever 2: ens160: &lt;BROADCAST,MULTICAST,UP,LOWER_UP&gt; mtu link/ether 00:0c:29:ed:cc:b8 brd ff:ff:ff:ff:ff:ff altname enp3s0 inet 192.168.100.103/24 brd 192.168.100.255 valid_lft forever preferred_lft forever inet6 fe80::20c:29ff:feed:ccb8/64 scope link valid_lft forever preferred_lft forever</pre>	<pre>[root@master01 ~]# ip addr 1: lo: &lt;LOOPBACK,UP,LOWER_UP&gt; mtu 65536 qdisc noqu link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 inet 127.0.0.1/8 scope host lo valid_lft forever preferred_lft forever inet6 ::1/128 scope host valid_lft forever preferred_lft forever 2: ens160: &lt;BROADCAST,MULTICAST,UP,LOWER_UP&gt; mtu 1 link/ether 00:0c:29:52:85:74 brd ff:ff:ff:ff:ff:ff altname enp3s0 inet 192.168.100.100/24 brd 192.168.100.255 sc valid_lft forever preferred_lft forever inet6 fe80::20c:29ff:fe52:8574/64 scope link r valid_lft forever preferred_lft forever</pre> <pre>[root@worker02 ~]# ip addr 1: lo: &lt;LOOPBACK,UP,LOWER_UP&gt; mtu 65536 qdisc no link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 inet 127.0.0.1/8 scope host lo valid_lft forever preferred_lft forever inet6 ::1/128 scope host valid_lft forever preferred_lft forever 2: ens160: &lt;BROADCAST,MULTICAST,UP,LOWER_UP&gt; mtu link/ether 00:0c:29:ab:6c:5f brd ff:ff:ff:ff:ff:ff altname enp3s0 inet 192.168.100.102/24 brd 192.168.100.255 valid_lft forever preferred_lft forever inet6 fe80::20c:29ff:feab:6c5f/64 scope link valid_lft forever preferred_lft forever</pre>
<p>VM 생성</p> <ul style="list-style-type: none"> <li>- 기존 Linux 템플릿을 Cloning하여 5대의 리눅스 가상머신을 생성</li> <li>- ON-PREMISE 가상머신은 Bridge, Host-only 2개의 네트워크 인터페이스를 생성</li> <li>- 그 외 나머지 VM은 host-only VMnet1 네트워크만 사용</li> </ul> <p>VM IP고정</p> <ul style="list-style-type: none"> <li>- ON-PREMISE VM : Bridged, host-only(VMnet1)</li> <li>- host-only(VMnet1): Static IP 설정</li> </ul>	



프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	Docker, Docker-compose 설치
<pre>[root@Onpremise ~]# docker compose version Docker Compose version v2.21.0</pre> <pre>[root@Onpremise ~]# docker version Client: Docker Engine - Community Version: 24.0.6 API version: 1.43 Go version: go1.20.7 Git commit: ed223bc Built: Mon Sep  4 12:33:07 2023 OS/Arch: linux/amd64 Context: default  Server: Docker Engine - Community Engine: Version: 24.0.6 API version: 1.43 (minimum version 1.12) Go version: go1.20.7 Git commit: 1a79695 Built: Mon Sep  4 12:32:10 2023 OS/Arch: linux/amd64 Experimental: false containerd: Version: 1.6.24 GitCommit: 61f9fd88f79f081d64d6fa3bb1a0dc71ec870523 runc: Version: 1.1.9 GitCommit: v1.1.9-0-gccaecfc docker-init: Version: 0.19.0 GitCommit: de40ad0</pre>	
<p>- ON-PREMISE 가상머신에 도커 v24.0.6와 도커컴포즈 v2.21.0 설치</p> <p>주요 목적</p> <ul style="list-style-type: none"> <li>- Docker Build를 통해 서비스 이미지 생성</li> <li>- NGINX Proxy Manager 설치</li> </ul>	



프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	FRRouting
<pre>[root@onpremise ~]# systemctl status frr ● frr.service - FRRouting    Loaded: loaded (/usr/lib/systemd/system/frr.service; enabled; vendor pr    Active: active (running) since Mon 2023-09-18 11:13:30 KST; 4s ago      Docs: <a href="https://frrouting.readthedocs.io/en/latest/setup.html">https://frrouting.readthedocs.io/en/latest/setup.html</a>   Process: 5322 ExecStart=/usr/libexec/frr/frrinit.sh start (code=exited,    Status: "FRR Operational"     Tasks: 15 (limit: 24660)    Memory: 20.0M     CGroup: /system.slice/frr.service             └─5339 /usr/libexec/frr/watchfrr -d -F traditional zebra bgpd o               └─5357 /usr/libexec/frr/zebra -d -F traditional -A 127.0.0.1 -s                 └─5362 /usr/libexec/frr/bgpd -d -F traditional -A 127.0.0.1                   └─5369 /usr/libexec/frr/ospfd -d -F traditional -A 127.0.0.1                     └─5372 /usr/libexec/frr/staticd -d -F traditional -A 127.0.0.1</pre> <pre>[root@onpremise ~]# vtysh  Hello, this is FRRouting (version 7.5.1). Copyright 1996-2005 Kunihiro Ishiguro, et al.  frr#</pre>	
<ul style="list-style-type: none"> <li>- ON-PREMISE 가상머신의 서로 다른 네트워크 대역인 Bridged와 VMnet1을 통신하기 위해, 오픈소스 네트워크 라우팅 소프트웨어인 FRR 설치</li> <li>- 네트워크의 라우팅 및 경로 선택을 관리</li> <li>- Bridged와 Host-only (VMnet1) 네트워크 통신 가능</li> <li>- vtyos 명령어로 FRR 접근</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	FRRouting table
<div>Public IP (Bridged)</div> <div>Private IP (VMnet1)</div>	<pre> Frr# sh ip route Codes: K - kernel route, C - connected, S - static, R - RIP,        O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,        T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,        F - PBR, f - OpenFabric,        &gt; - selected route, * - FIB route, q - queued, r - rejected, b - backup  K&gt;* 0.0.0.0/0 [0/100] via 211.35.168.254, ens160, src 211.35.168.89, 1d07h19m C&gt;* 172.18.0.0/16 is directly connected, br-1040b26c52e8, 1d07h19m C&gt;* 192.168.100.0/24 is directly connected, ens192, 1d07h19m C&gt;* 211.35.168.0/24 is directly connected, ens160, 1d07h19m  [root@onpremise ~]# ip addr 1: lo: &lt;LOOPBACK,UP,LOWER_UP&gt; mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000     link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00     inet 127.0.0.1/8 scope host lo         valid_lft forever preferred_lft forever     inet6 ::1/128 scope host         valid_lft forever preferred_lft forever 2: ens160: &lt;BROADCAST,MULTICAST,UP,LOWER_UP&gt; mtu 1500 qdisc mq state UP group default qlen 1000     link/ether 00:0c:29:e0:ab:a6 brd ff:ff:ff:ff:ff:ff     altname enp3s0     inet 211.35.168.89/24 brd 211.35.168.255 scope global dynamic noprefixroute ens160         valid_lft 89s98sec preferred_lft 5956sec     inet6 fe80::20c:29ff:fe0:aba6/64 scope link noprefixroute         valid_lft forever preferred_lft forever 3: ens192: &lt;BROADCAST,MULTICAST,UP,LOWER_UP&gt; mtu 1500 qdisc mq state UP group default qlen 1000     link/ether 00:0c:29:e0:ab:b0 brd ff:ff:ff:ff:ff:ff     altname enp11s0     inet 192.168.100.254/24 brd 192.168.100.255 scope global noprefixroute ens192         valid_lft forever preferred_lft forever     inet6 fe80::5e14:1a20:a977:64c6/64 scope link noprefixroute         valid_lft forever preferred_lft forever 4: br-1040b26c52e8: &lt;BROADCAST,MULTICAST,UP,LOWER_UP&gt; mtu 1500 qdisc noqueue state UP group default     link/ether 02:42:b0:98:00:10 brd ff:ff:ff:ff:ff:ff     inet 172.18.0.1/16 brd 172.18.255.255 scope global br-1040b26c52e8         valid_lft forever preferred_lft forever     inet6 fe80::42:b0ff:fe98:10/64 scope link         valid_lft forever preferred_lft forever </pre>
<ul style="list-style-type: none"> <li>- FRR 라우팅 테이블에 두 네트워크가 connected로 통신이 가능한 것 확인 가능</li> <li>- Bridged Network : 211.35.168.89/24</li> <li>- VMnet1 :192.168.100.254/24</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	DDNS, Crontab
<pre>[root@onpremise ~]# cat ddns.sh #!/bin/bash  ### dnf install bind-utils -y 를 해서 host command 설치 되어야 함 ### cloudflare information  dns_record="kkangtae.store" zoneid="c60e10dd3670eab98219707b69c16ff4" cloudflare_zone_api_token="zuZ5npTACE08w39czf9X3hvNHdat0kPLM-8TYZ_T" proxied="false" ttl=120 # internal로 변경 시 서버의 interface에 있는 IP가 입력됨 what_ip=external  ### Check validity of "ttl" parameter if [ "\${ttl}" -lt 120 ]    [ "\${ttl}" -gt 7200 ] &amp;&amp; [ "\${ttl}" -ne 1   echo "Error! ttl out of range (120-7200) or not set to 1"  [root@onpremise ~]# crontab -l #cloudfalre-ddns */5 * * * * ~/ddns.sh @reboot ~/ddns.sh</pre>	
<ul style="list-style-type: none"> <li>- Bridged 네트워크의 동적 IP 주소가 바뀌어도 도메인 이름으로 웹 서비스에 접근이 가능하도록 DDNS 쉘 스크립트를 만들어 실행</li> <li>- 위 작업을 재부팅, 그리고 5분마다 자동 실행을 위한 Crontab 설정</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	K8s 설치 및 연결
<pre>[root@master1 ~]# kubectl version --client Client Version: v1.28.2 Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3</pre>	
<pre>[root@worker1 ~]# kubectl version --client Client Version: v1.28.2 Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3</pre>	
<pre>[root@worker2 ~]# kubectl version --client Client Version: v1.28.2 Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3</pre>	
<pre>[root@master01 ~]# k get node -o wide NAME          STATUS    ROLES    KERNEL-VERSION   AGE    VERSION    INTERNAL-IP    EXTERNAL-IP    OS-IMAGE master01      Ready    control-plane  4.18.0-477.21.1.el8_8.x86_64  3d    v1.28.2    192.168.100.100 &lt;none&gt;         Rocky Linux 8. 8 (Green Obsidian) worker01      Ready    &lt;none&gt;    4.18.0-477.21.1.el8_8.x86_64  3d    v1.28.2    192.168.100.101 &lt;none&gt;         Rocky Linux 8. 8 (Green Obsidian) worker02      Ready    &lt;none&gt;    4.18.0-477.21.1.el8_8.x86_64  3d    v1.28.2    192.168.100.102 &lt;none&gt;         Rocky Linux 8. 8 (Green Obsidian)</pre>	
<ul style="list-style-type: none"> <li>- 컨테이너 어플리케이션 배포를 위한 Kubernetes v1.28.2 설치</li> <li>- Master, worker node로 사용 할 가상머신에 Kubernetes 설치 및 클러스터 구성</li> <li>- master01, worker01, worker02 node에 K8s 설치</li> <li>- Master node 및 Worker node 지정</li> <li>- Master, Worker node 연동하여 Cluster 구축</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	Taint 설정
<pre>[root@master1 ~]# kubectl taint nodes master1 node-role.kubernetes.io/control-plane:NoSchedule- node/master1 untainted</pre> <pre>[root@master1 ~]# kubectl describe node master1   grep -i taint Taints:             &lt;none&gt;</pre>	
<ul style="list-style-type: none"> <li>- 클러스터 내 자원이 부족 할 것을 대비하여 Master node에 taint (테인트) 옵션을 제거</li> <li>- untainted 상태의 Master node에도 Worker node처럼 Pod을 스케줄링 가능</li> <li>- default : taint 상태</li> </ul>	



프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	Calico 배포
------------	-----------

```
[root@master01 ~]# cat calico.yaml
```

```
    name: calico-config
    key: veth_mtu
    # The default IPv4 pool to create on startup if none exists. Pod IPs will be
    # chosen from this range. Changing this value after installation will have
    # no effect. This should fall within '--cluster-cidr'.
    - name: CALICO_IPV4POOL_CIDR
      value: "192.168.101.0/24"
    # Disable file logging so 'kubectl logs' works.
    - name: CALICO_DISABLE_FILE_LOGGING
      value: "true"
    # Set Felix endpoint to host default action to ACCEPT
```

```
[root@master1 ~]# kubectl get pods -n kube-system
```

NAME	READY	STATUS	RESTARTS	AGE
calico-kube-controllers-7ddc4f45bc-d27wr	1/1	Running	0	43s
calico-node-44mfc	1/1	Running	0	43s
calico-node-k8g2f	1/1	Running	0	43s
calico-node-x8w9f	1/1	Running	0	43s
coredns-5dd5756b68-68s25	1/1	Running	0	22m
coredns-5dd5756b68-jn9b4	1/1	Running	0	22m
etcd-master1	1/1	Running	0	23m
kube-apiserver-master1	1/1	Running	0	23m
kube-controller-manager-master1	1/1	Running	0	23m
kube-proxy-52fq8	1/1	Running	0	21m
kube-proxy-fvcwx	1/1	Running	0	22m
kube-proxy-m8gt7	1/1	Running	0	21m
kube-scheduler-master1	1/1	Running	0	23m

```
[root@master1 ~]#
```

```
[root@master1 ~]# k get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
master1	Ready	control-plane	23m	v1.28.2
worker1	Ready	<none>	21m	v1.28.2
worker2	Ready	<none>	21m	v1.28.2

- 컨테이너를 위한 네트워킹, IP 관리, 접근 제어 등의 기능 제공을 위해 Calico CNI 설치
- 각 Node에 설치하여 각 Pod 간 네트워크 통신 가능
- Calico 설치를 위한 yaml 파일을 생성 및 실행
- 이때, 새로 생성될 Pod의 IPv4 Pool : 192.168.101.0/24
- Calico Pod가 생성 및 실행되는 것을 확인

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	MetalLB 설치
<pre>[root@master01 ~]# cat metallb-ip.yaml # Loadbalancer가 사용할 IP 주소 대역 환경 설정 apiVersion: metallb.io/v1beta1 kind: IPAddressPool metadata:   name: first-pool   namespace: metallb-system spec:   addresses:     - 192.168.100.200-192.168.100.250 #k8s가 사용하는 네트워크 대역 중 서비스에 줄 IP 대역 --- # IP가 L2 계층에서 ARP protocol에 반응할 수 있는 환경 설정 apiVersion: metallb.io/v1beta1 kind: L2Advertisement metadata:   name: default   namespace: metallb-system spec:   ipAddressPools:     - first-pool  [root@master1 ~]# kubectl get all -n metallb-system NAME                                READY   STATUS    RESTARTS   AGE pod/controller-5c6b6c8447-p6gft     1/1     Running   0           94s pod/speaker-56rdj                   1/1     Running   0           94s pod/speaker-ntjrk                   1/1     Running   0           94s pod/speaker-zp4qc                   1/1     Running   0           94s  NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP   PORT(S) service/webhook-service             ClusterIP     10.106.13.77  &lt;none&gt;        443/TCP  NAME                                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE daemonset.apps/speaker              3         3         3       3             3  NAME                                READY   UP-TO-DATE   AVAILABLE   AGE deployment.apps/controller          1/1     1             1           94s  NAME                                DESIRED   CURRENT   READY   AGE replicaset.apps/controller-5c6b6c8447 1         1         1       94s</pre>	
<ul style="list-style-type: none"> <li>- Kubernetes Cluster에서 사용할 수 있는 오픈소스 로드밸런서</li> <li>- MetalLB 설치를 위한 yaml 작성 및 실행</li> <li>- 이때, 로드밸런서 Service의 EXTERNAL-IP 대역을 192.168.100.200 ~ 192.168.100.250/24 로 지정</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	Metric-Server																				
<pre>[root@master01 ~]# kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml</pre>																					
<pre>[root@master01 ~]# kubectl edit deploy metrics-server -n kube-system</pre>																					
<pre>39     - args: 40       - --cert-dir=/tmp 41       - --secure-port=4443 42       - --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname 43       - --kubelet-use-node-status-port 44       - <del>metric-resolution=15s</del> 45       - --kubelet-insecure-tls</pre>																					
<pre>33     volumeMounts: 34       - mountPath: /tmp 35         name: tmp-dir 36     dnsPolicy: ClusterFirst 37     hostNetwork: true 38     nodeSelector: 39       kubernetes.io/os: linux</pre>																					
<pre>Every 1.0s: kubectl top node</pre> <table><thead><tr><th>NAME</th><th>CPU(cores)</th><th>CPU%</th><th>MEMORY(bytes)</th><th>MEMORY%</th></tr></thead><tbody><tr><td>master1</td><td>210m</td><td>10%</td><td>1760Mi</td><td>46%</td></tr><tr><td>worker1</td><td>44m</td><td>2%</td><td>1147Mi</td><td>62%</td></tr><tr><td>worker2</td><td>46m</td><td>2%</td><td>1160Mi</td><td>63%</td></tr></tbody></table>		NAME	CPU(cores)	CPU%	MEMORY(bytes)	MEMORY%	master1	210m	10%	1760Mi	46%	worker1	44m	2%	1147Mi	62%	worker2	46m	2%	1160Mi	63%
NAME	CPU(cores)	CPU%	MEMORY(bytes)	MEMORY%																	
master1	210m	10%	1760Mi	46%																	
worker1	44m	2%	1147Mi	62%																	
worker2	46m	2%	1160Mi	63%																	
<ul style="list-style-type: none"><li>- Kubernetes Cluster 내에서 리소스 사용량 체크 및 모니터링을 위해 모든 Node에 Metric-Server 설치</li><li>- top 명령어를 통해 CPU, 메모리 사용량 등 리소스 사용량 확인 가능</li></ul>																					



프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	MariaDB
<pre>[root@DB ~]# systemctl status mariadb ● mariadb.service - MariaDB 10.6.15 database server    Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)    Drop-In: /etc/systemd/system/mariadb.service.d             └─migrated-from-my.cnf-settings.conf    Active: active (running) since Wed 2023-09-20 18:41:06 KST; 1 day 22h ago      Docs: man:mariadb(8)            https://mariadb.com/kb/en/library/systemd/</pre>	
<pre>MariaDB [(none)]&gt; show databases; +-----+   Database   +-----+   gnuboard     information_schema     mysql        nextcloud     performance_schema     sys          xeboard    +-----+ 7 rows in set (0.001 sec)</pre>	
<pre>MariaDB [(none)]&gt; select user from mysql.user; +-----+   User   +-----+   gnuuser     nextuser     root     xeuser     gnuuser     mariadb.sys     mysql     nextuser     root     xeuser   +-----+ 10 rows in set (0.005 sec)</pre>	
<ul style="list-style-type: none"> <li>- 웹서비스 Gnuboard와 Nextcloud의 DB를 별도의 VM을 생성하여 지정</li> <li>- MariaDB v10.6.15 설치</li> <li>- 데이터베이스 생성 : 'gnuboard', 'nextcloud'</li> <li>- 사용자 생성 : 'gnuuser', 'nextuser'</li> <li>- 사용자 권한 부여</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	NFS
<pre>[root@DB ~]# systemctl status nfs-utils ● nfs-utils.service - NFS server and client services    Loaded: loaded (/usr/lib/systemd/system/nfs-utils.service; static; vendor preset: disabled)    Active: active (exited) since Fri 2023-09-22 17:38:07 KST; 4s ago    Process: 102088 ExecStart=/bin/true (code=exited, status=0/SUCCESS)    Main PID: 102088 (code=exited, status=0/SUCCESS)      Tasks: 0 (limit: 24660)     Memory: 0B    CGroup: /system.slice/nfs-utils.service</pre>	
<pre>[root@DB ~]# showmount -e Export list for DB: /web *</pre>	
<pre>[root@DB ~]# ls -l /web total 12 drwxr-xr-x. 16  33 tape 4096 Sep 21 09:36 gnu -rw-r--r--.  1 root root   5 Sep 21 11:06 index.html -rw-r--r--.  1 root root  20 Sep 20 12:34 info.php drwxr-xr-x.  2  33 tape   6 Sep 21 11:09 nextcloud</pre>	
<ul style="list-style-type: none"> <li>- DB VM에 NFS 서버를 설치</li> <li>- /web 공유디렉토리 지정</li> <li>- DB VM의 IP : 192.168.100.103/24</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	PV, PVC
<pre>[root@master01 ~]# cat &lt;&lt; 'EOF' &gt; nfs-pv.yaml &gt; apiVersion: v1 &gt; kind: PersistentVolume &gt; metadata: &gt;   name: nfs-pv &gt; spec: &gt;   capacity: &gt;     storage: 10Gi &gt;   volumeMode: Filesystem &gt;   accessModes: &gt;     - ReadWriteMany &gt;   mountOptions: &gt;     - nfsvers=4.2 &gt;   nfs: &gt;     path: /web &gt;     server: 192.168.100.103 &gt; EOF [root@master01 ~]# ls -l total 256 -rw-----. 1 root root 1364 Sep 18 10:04 anaconda-ks.cfg -rw-r--r--. 1 root root 244732 Sep 19 15:39 calico.yaml -rwxr-xr-x. 1 root root 1781 Sep 19 14:57 default_config.sh -rw-r--r--. 1 root root 519 Sep 19 16:11 metallb-ip.yaml -rw-r--r--. 1 root root 404 Sep 19 17:45 nfs-pv.yaml drwxr-xr-x. 2 root root 81 Sep 19 16:32 yaml [root@master01 ~]# k apply -f nfs-pv.yaml persistentvolume/nfs-pv created persistentvolumeclaim/nfs-pvc created  [root@master01 ~]# k get pv,pvc NAME                                CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM N  AGE persistentvolume/nfs-pv            10Gi      Rwx           Retain          Bound   default/nfs-pvc 24s  NAME                                STATUS  VOLUME  CAPACITY  ACCESS MODES  STORAGECLASS  AGE persistentvolumeclaim/nfs-pvc      Bound   nfs-pv  10Gi      Rwx           nfs-pv         24s</pre>	
<ul style="list-style-type: none"> <li>- PV, PVC 생성을 위한 yaml 작성 및 실행</li> <li>- 이때, nfs의 path와 NFS 서버의 IP를 지정 (DB VM)</li> <li>- PVC Bound 상태 확인</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	Nginx Proxy Manager
------------	---------------------

```
[root@Onpremise ~]# cat docker-compose.yaml
version: '3.8'
services:
  app:
    image: 'jc21/nginx-proxy-manager:latest'
    restart: unless-stopped
    ports:
      - '80:80'
      - '59081:81'
      - '443:443'
    volumes:
      - './data:/data'
      - './letsencrypt:/etc/letsencrypt'
```

▲ 안전하지 않음 211.35.168.89:59081 login



#### Nginx Proxy Manager 설치

- 웹 기반 관리
- 도메인 및 서브도메인 관리
- SSL/TLS 인증서 관리
- 웹서버 및 프록시 서버 설정을 더욱 쉽게 관리하고 보안 강화
- docker-compose.yaml 작성 및 실행하여 설치
- 관리모드 접속포트는 59081/tcp로 지정

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	Nextcloud - yaml
<pre>[root@master01 ~]# ls -l nextcloud-yaml/ total 12 -rw-r--r--. 1 root root 382 Sep 20 16:10 hpa.yaml -rw-r--r--. 1 root root 1700 Sep 20 15:47 nextcloud.yaml -rw-r--r--. 1 root root 645 Sep 20 14:07 pv-pvc.yaml  [root@master01 nextcloud-yaml]# cat pv-pvc.yaml apiVersion: v1 kind: PersistentVolume metadata:   name: pv-nextcloud   namespace: ns-nextcloud spec:   capacity:     storage: 30Gi   volumeMode: Filesystem   accessModes:     - ReadWriteOnce     - ReadWriteMany   persistentVolumeReclaimPolicy: Recycle   storageClassName: nextcloud   mountOptions:     - nfsvers=4.2   nfs:     path: /web/nextcloud     server: 192.168.100.103 --- kind: PersistentVolumeClaim apiVersion: v1 metadata:   name: pvc-nextcloud   namespace: ns-nextcloud spec:   accessModes:     - ReadWriteOnce     - ReadWriteMany   volumeMode: Filesystem   resources:     requests:       storage: 30Gi   storageClassName: nextcloud</pre> <pre>[root@master01 nextcloud-yaml]# cat nextcloud.yaml apiVersion: apps/v1 kind: Deployment metadata:   name: nextcloud-deployment   namespace: ns-nextcloud spec:   replicas: 1   selector:     matchLabels:       app: nextcloud   template:     metadata:       labels:         app: nextcloud     spec:       containers:         - name: APC_SHM_SIZE           value: 256M         - name: CRON_PERIOD           value: 10m         - name: NEXTCLOUD_MAX_TIME           value: "3600"         - name: NEXTCLOUD_MEMORY_LIMIT           value: 1024M         - name: NEXTCLOUD_TRUSTED_DOMAINS           value: cloud.kkangtae.store         - name: OVERWRITEHOST           value: cloud.ddomung.site         - name: OVERWRITEPROTOCOL           value: https         - name: PHP_MEMORY_LIMIT           value: 1024M         - name: PHP_UPLOAD_LIMIT           value: 10G         - name: TZ           value: Asia/Seoul         - name: UPLOAD_MAX_SIZE           value: 10G       name: nextcloud       image: nextcloud       ports:         - containerPort: 80       volumeMounts:         - name: nextcloud-storage-volume           mountPath: /var/www/html       resources:         limits:           cpu: 1000m           requests:             cpu: 500m       volumes:         - name: nextcloud-storage-volume           persistentVolumeClaim:             claimName: pvc-nextcloud</pre> <pre>[root@master01 nextcloud-yaml]# cat hpa.yaml apiVersion: autoscaling/v1 kind: HorizontalPodAutoscaler metadata:   name: autoscale-nextcloud   namespace: ns-nextcloud spec:   maxReplicas: 10   minReplicas: 1   scaleTargetRef:     apiVersion: apps/v1     kind: Deployment     name: nextcloud-deployment     targetCPUUtilizationPercentage: 50</pre> <pre>apiVersion: v1 kind: Service metadata:   name: nextcloud-service   namespace: ns-nextcloud spec:   type: LoadBalancer   selector:     app: nextcloud   ports:     - name: nextcloud-service-port       protocol: TCP       port: 8000       targetPort: 80</pre>	
<ul style="list-style-type: none"> <li>- Nextcloud 웹서비스를 위한 3개의 yaml 생성</li> <li>- hpa.yaml : 리소스 사용량에 따라 최소 1개 ~ 최대 10개의 Pod 생성 설정</li> <li>- nextcloud.yaml : Deployment, LoadBalancer Service 등 지정</li> <li>- pv-pvc.yaml : 용량 및 NFS mount 지정(DB)</li> <li>- Namespace : ns-nextcloud</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE

Nextcloud – Proxy Hosts

```

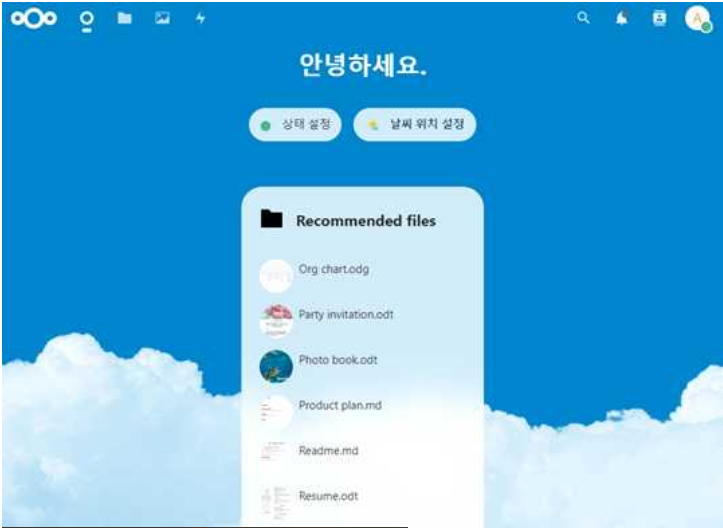

Every 1.0s: kubectl get po,rs,deploy,svc,pv,pvc -n ns-nextcloud -o wide
NAME                                READY    STATUS    RESTARTS   AGE    IP             NODE    NOMINATED NODE    READINESS GATES
pod/nextcloud-deployment-77d5954b4d-l2vsx  1/1      Running   0           42m    192.168.101.208 worker01 <none> <none>
NAME                                DESIRED   CURRENT   READY    AGE    CONTAINERS    IMAGES    SELECTOR
replicaset.apps/nextcloud-deployment-77d5954b4d  1         1         1        42m    nextcloud     nextcloud  app=nextcloud,pod-template-hash=77d5954b4d
NAME                                READY    UP-TO-DATE   AVAILABLE   AGE    CONTAINERS    IMAGES    SELECTOR
deployment.apps/nextcloud-deployment  1/1      1            1           42m    nextcloud     nextcloud  app=nextcloud
NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE    SELECTOR
service/nextcloud-service  LoadBalancer  10.110.249.243  192.168.100.201  8000:30111/TCP   42m    app=nextcloud
NAME                                CAPACITY  ACCESS MODES  RECLAIM POLICY   STATUS   CLAIM                STORAGECLASS  REASON   AGE    VOLUME MODE
persistentvolume/pv-nextcloud  30Gi      RWO,RWX      Recycle          Bound    ns-nextcloud/pvc-nextcloud  nextcloud    <none>   42m    Filesystem
NAME                                STATUS    VOLUME      CAPACITY  ACCESS MODES  STORAGECLASS  AGE    VOLUME MODE
persistentvolumeclaim/pvc-nextcloud  Bound    pv-nextcloud  30Gi      RWO,RWX      nextcloud     42m    Filesystem

```

- Nginx Proxy Manager에 접속하여 Nextcloud 로드밸런서 서비스의 외부 IP를 목적지로 지정한 Proxy Host를 생성 (cloud.kkangtae.store)
- Service port인 8000/tcp를 사용하여 Nextcloud 접속



프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	Nextcloud – 접속 및 구축
 	
<ul style="list-style-type: none"> <li>- 데이터베이스 호스트 : DB VM의 IP</li> <li>- DB VM에서 생성한 데이터베이스(nextcloud)와 사용자(nextuser) 지정</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE

Nextcloud – HTTPS 접속

- Let's Encrypt를 이용하여 와일드카드 인증서 발급
- 이때, Cloudflare에 등록된 도메인(kkangtae.store)의 API token을 입력
- HTTPS 보안 웹접속 가능



## 프로젝트 완료 보고서

프로젝트 주제

VPN을 활용한 하이브리드 클라우드 구축

단계 : 프로젝트 완료

작성자 : 김경태

작성일 : 2023.10.04

ON-PREMISE

Nextcloud – Auto Scaling

```

root@nextcloud-deployment-77d5954b4d-l2vsx:/var/www/html# stress --vm 2 -t 60s
stress: info: [214] dispatching hogs: 0 cpu, 0 io, 2 vm, 0 hdd
stress: info: [214] successful run completed in 60s
Every 1.0s: kubectl get po,rs,deploy,svc,pv,pvc -n ns-nextcloud -o wide
master01: Wed Sep 20 10:10:10 2023

NAME                                READY    STATUS    RESTARTS   AGE    IP              NODE    NOMINATED NODE    READINESS GATES
pod/nextcloud-deployment-77d5954b4d-28zt7    1/1      Running   0           31s    192.168.101.90  worker02    <none>             <none>
pod/nextcloud-deployment-77d5954b4d-5g96f    1/1      Running   0           16s    192.168.101.209 worker01    <none>             <none>
pod/nextcloud-deployment-77d5954b4d-6xhnm    1/1      Running   0           31s    192.168.101.134 master01    <none>             <none>
pod/nextcloud-deployment-77d5954b4d-l2vsx    1/1      Running   0           96m    192.168.101.208 worker01    <none>             <none>

NAME                                DESIRED   CURRENT   READY    AGE    CONTAINERS    IMAGES    SELECTOR
replicaset.apps/nextcloud-deployment-77d5954b4d    4          4          4        96m    nextcloud     nextcloud  app=nextcloud,pod-template-hash=77d5954b4d

NAME                                READY    UP-TO-DATE   AVAILABLE   AGE    CONTAINERS    IMAGES    SELECTOR
deployment.apps/nextcloud-deployment    4/4      4             4           96m    nextcloud     nextcloud  app=nextcloud

NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE    SELECTOR
service/nextcloud-service    LoadBalancer  10.110.249.243  192.168.100.201  8000:30131/TCP   96m    app=nextcloud

NAME                                CAPACITY    ACCESS MODES    RECLAIM POLICY    STATUS    CLAIM    STORAGECLASS    REASON    AGE    VOLUME MODE
persistentvolume/pv-nextcloud    30Gi        RWO,RWX         Recycle           Bound     ns-nextcloud/pvc-nextcloud    nextcloud    <none>    96m    Filesystem

NAME                                STATUS    VOLUME    CAPACITY    ACCESS MODES    STORAGECLASS    AGE    VOLUME MODE
persistentvolumeclaim/pvc-nextcloud    Bound     pv-nextcloud    30Gi        RWO,RWX         nextcloud       96m    Filesystem

Every 1.0s: kubectl get po,rs,deploy,svc,pv,pvc -n ns-nextcloud -o wide
master01: Wed Sep 20 10:10:11 2023

NAME                                READY    STATUS    RESTARTS   AGE    IP              NODE    NOMINATED NODE    READINESS GATES
pod/nextcloud-deployment-77d5954b4d-6xhnm    1/1      Running   0           8m16s    192.168.101.134 master01    <none>             <none>

NAME                                DESIRED   CURRENT   READY    AGE    CONTAINERS    IMAGES    SELECTOR
replicaset.apps/nextcloud-deployment-77d5954b4d    1          1          1        104m    nextcloud     nextcloud  app=nextcloud,pod-template-hash=77d5954b4d

NAME                                READY    UP-TO-DATE   AVAILABLE   AGE    CONTAINERS    IMAGES    SELECTOR
deployment.apps/nextcloud-deployment    1/1      1             1           104m    nextcloud     nextcloud  app=nextcloud

NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE    SELECTOR
service/nextcloud-service    LoadBalancer  10.110.249.243  192.168.100.201  8000:30131/TCP   104m    app=nextcloud

NAME                                CAPACITY    ACCESS MODES    RECLAIM POLICY    STATUS    CLAIM    STORAGECLASS    REASON    AGE    VOLUME MODE
persistentvolume/pv-nextcloud    30Gi        RWO,RWX         Recycle           Bound     ns-nextcloud/pvc-nextcloud    nextcloud    <none>    104m    Filesystem

NAME                                STATUS    VOLUME    CAPACITY    ACCESS MODES    STORAGECLASS    AGE    VOLUME MODE
persistentvolumeclaim/pvc-nextcloud    Bound     pv-nextcloud    30Gi        RWO,RWX         nextcloud       104m    Filesystem

```

- 임의의 Pod에서 CPU 부하테스트 진행 (2개 Process, 60초간)
- 부하 증가 : Pod 1개에서 4개로 증가
- 부하테스트 종료 : 기존 Pod 1개로 복구

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	Docker Build GNU – PHP config
<pre> [root@DB ~]# cat web/conf/php/www.conf [www] user = www-data group = www-data  listen = 9000 listen.owner = www-data listen.group = www-data  pm = dynamic pm.max_children = 30 pm.start_servers = 5 pm.min_spare_servers = 5 pm.max_spare_servers = 10 pm.max_requests = 500  [root@DB ~]# cat web/conf/php/php.ini [PHP] engine = On short_open_tag = On precision = 14 output_buffering = 4096 zlib.output_compression = off implicit_flush = off unserialize_callback_func = serialize_precision = -1 disable_functions =  [root@DB ~]# tree web web ├── conf │   ├── nginx │   │   ├── conf.d │   │   │   └── default.conf │   │   └── nginx.conf │   └── php │       ├── php.ini │       └── www.conf </pre>	
- DB VM의 공유디렉토리 '/web'에 PHP conf 파일 생성	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	Docker Build GNU – PHP Dockerfile
------------	-----------------------------------

```
[root@DB ~]# cat Dockerfile
# FROM image이름:tag 해당 이미지를 사용
FROM php:8.2-fpm
# LABEL은 docker inspect 또는 docker info를 통해 해당이미지의 간략한 정보 작성
LABEL maintainer="meongchung@meonchung.shop"
LABEL issue_date=""
LABEL description="php-fpm module add"
# RUN은 이미지에서 특정 명령을 수행하는 것으로 가능한 하나로 합쳐서 사용하는 것이 좋음
# RUN은 Layer가 되어서 여러 명령을 RUN으로 구분하면 Layer가 너무 많아지는 문제 발생
RUN apt update \
    && apt install -y \
    libmagickwand-dev --no-install-recommends \
    libicu-dev \
    libonig-dev \
    && printf "\n" | pecl install imagick \
    && docker-php-ext-enable imagick \
    && docker-php-ext-install mysqli \
    && docker-php-ext-install pdo_mysql \
    && docker-php-ext-install iconv \
    && docker-php-ext-install intl \
    && docker-php-ext-install opcache \
    && docker-php-ext-install mbstring \
    && apt update -y

# COPY는 php.ini와 www.conf 같은 환경설정 파일을 host에 작성한 것을 build 시 추가
COPY ./web/conf/php/php.ini /usr/local/etc/php/php.ini
COPY ./web/conf/php/www.conf /usr/local/etc/php-fpm.d/www.conf
# EXPOSE 명령은 컨테이너 port를 open
EXPOSE 9000
# WORKDIR은 docker 접속 시 해당 경로로 바로 접근하고 컨테이너에서 명령어가 실행될 때의 default 위치
WORKDIR /root
# 컨테이너 실행 시 실행될 명령어
CMD ["php-fpm"]
```

- PHP 이미지를 실행하는 Dockerfile 생성
- 이때, PHP conf 파일을 복사하는 스크립트 내용 추가

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	Docker Build GNU – PHP 이미지 Build & Push
<pre> [root@DB ~]# docker images REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE php-8/php            8.2-fpm            fe87a163fd38       15 minutes ago     752MB meongchung/php-8.2  8.2-fpm            fe87a163fd38       15 minutes ago     752MB  [root@DB ~]# docker push meongchung/php-8.2:8.2-fpm The push refers to repository [docker.io/meongchung/php-8.2] 5f70bf18a086: Mounted from meongchung/php-8 a509caf6cfef: Pushed 4b5b0de77fe6: Pushed 100ff102f9d8: Pushed c925048cfc5b: Mounted from meongchung/php-8 9d2c2f7ef22b: Mounted from meongchung/php-8 ffb257a6a684: Mounted from meongchung/php-8 0fe0089ee16d: Mounted from meongchung/php-8 1775f7f20de4: Mounted from meongchung/php-8 1283d7cc1b75: Mounted from meongchung/php-8 e1b5a500198: Mounted from meongchung/php-8 e2de06477579: Mounted from meongchung/php-8 1ff185c2e955: Mounted from meongchung/php-8 a2d7501dfb35: Mounted from meongchung/php-8 8.2-fpm: digest: sha256:7a3b4458c67a34abb0c36e343109bb170c52c6c222eafded639373c4445b46fe size: 3244 </pre>	
- 생성된 PHP 이미지를 Docker hub의 개인 Repository로 Push (meongchung)	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	Docker Build GNU – Nginx config 생성
<pre>[root@DB ~]# cat web/conf/nginx/nginx.conf user www-data; worker_processes auto; error_log /var/log/nginx/error.log notice; pid /run/nginx.pid;  # Load dynamic modules. See /usr/share/doc/nginx/README.dynamic. include /usr/share/nginx/modules/*.conf;  events {     worker_connections 1024; }  http {     log_format main '\$remote_addr - \$remote_user [\$time_local] "\$request" '         '\$status \$body_bytes_sent "\$http_referer" '         '"\$http_user_agent" "\$http_x_forwarded_for"';      #access_log /var/log/nginx/access.log main;      sendfile            on;     tcp_nopush          on;     keepalive_timeout   65;     types_hash_max_size 4096;      include              /etc/nginx/mime.types;     default_type         application/octet-stream;      # Load modular configuration files from the /etc/nginx/conf.d directory.     # See http://nginx.org/en/docs/nginx_core_module.html#include     # for more information.     include /etc/nginx/conf.d/*.conf; }  [root@DB ~]# cat web/conf/nginx/conf.d/default.conf server {     listen      80;     listen      [::]:80;     server_name _;     root        /usr/share/nginx/html;      location / {         index index.php index.html index.htm;         try_files \$uri \$uri/ /index.php\$request_uri;     }      # Load configuration files for the default server block.     include /etc/nginx/default.d/*.conf;     error_page 404 /404.html;     location = /404.html {     }      error_page 500 502 503 504 /50x.html;     location = /50x.html {     }      location ~ [^/].php(/ \$) {         fastcgi_split_path_info ^(.+?\.php)(/.*)\$;         set \$path_info \$fastcgi_path_info;         fastcgi_index index.php;         include fastcgi_params;         fastcgi_pass 10.97.112.94:9000;         fastcgi_param SCRIPT_FILENAME \$document_root\$fastcgi_script_name;     }      # Deny access to .htaccess files     location ~ /\.ht {         deny all;     } }</pre>	
<ul style="list-style-type: none"> <li>- DB VM의 '/web' 공유디렉토리에 Nginx conf 파일 생성</li> <li>- 이때, default.conf에서 연동할 PHP cluster 고정 IP와 9000 포트 지정</li> </ul>	



프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	Docker Build GNU – Nginx Dockerfile
<pre>[root@DB ~]# cat Dockerfile FROM nginx:latest LABEL maintainer="meongchung@meonchung.shop" LABEL description="nginx with config" RUN apt update \     &amp;&amp; apt install vim -y COPY ./web/conf/nginx/nginx.conf /etc/nginx/nginx.conf COPY ./web/conf/nginx/conf.d/default.conf /etc/nginx/conf.d/default.conf VOLUME [/etc/nginx/conf.d] EXPOSE 80 CMD ["nginx", "-g", "daemon off;"]</pre>	
<ul style="list-style-type: none"> <li>- Nginx 이미지를 실행하는 Dockerfile 생성</li> <li>- PHP와 연동하는 conf 적용</li> <li>- 이때, Nginx conf 파일을 복사하는 스크립트 내용 추가 --&gt; 해당 이미지 안에 conf 파일 포함하려는 목적</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE

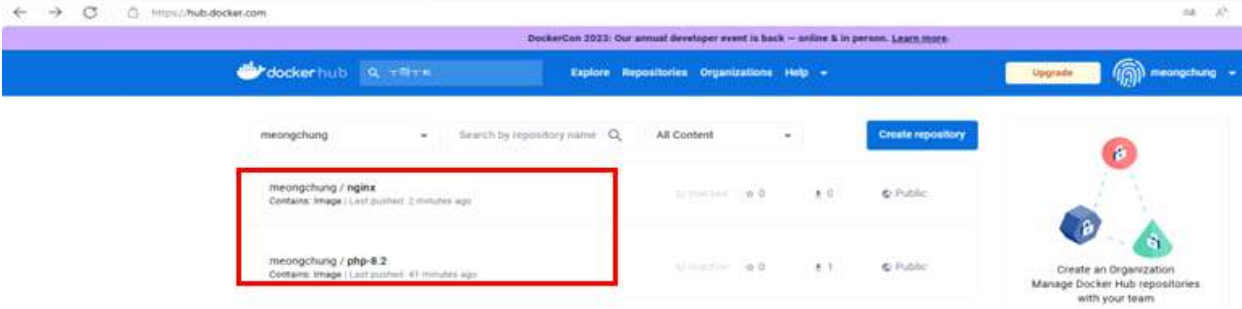
Docker Build GNU – 이미지 Build & Push

```

[root@DB ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
meongchung/nginx    latest             2ecb26d578b4       39 seconds ago     248MB

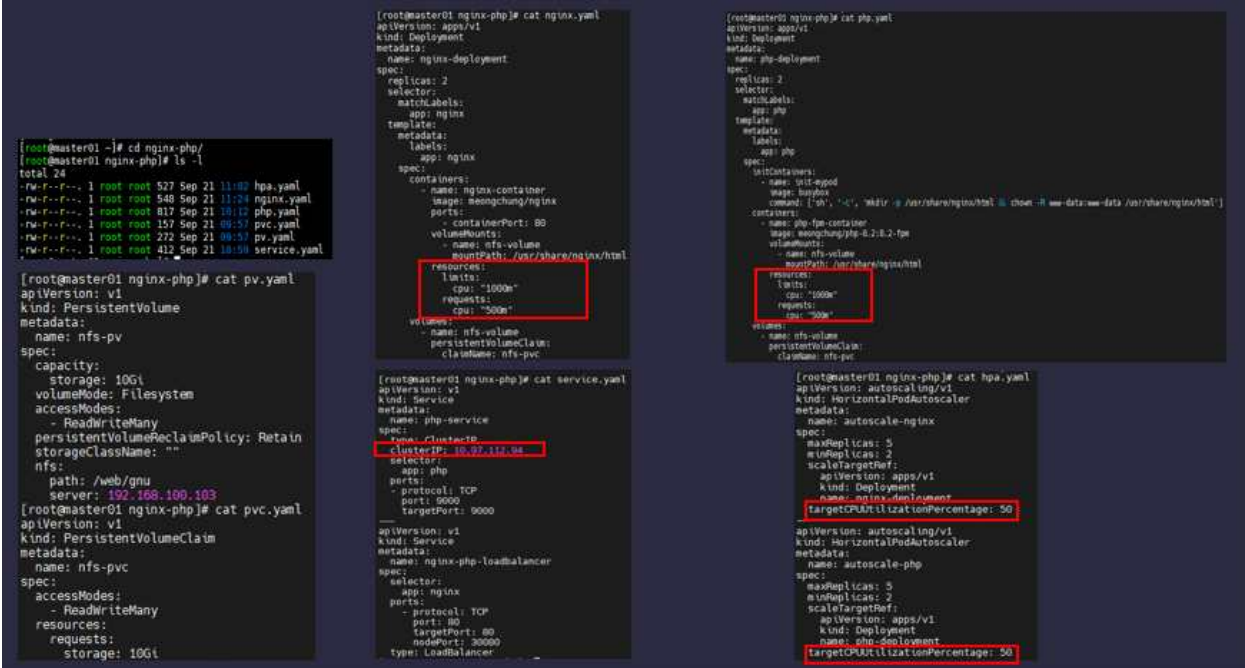
[root@DB ~]# docker push meongchung/nginx:latest
The push refers to repository [docker.io/meongchung/nginx]
dc14c0659a2d: Pushed
39a447d40cff: Pushed
54af2b7d18bc: Pushed
aae231785348: Mounted from library/nginx
e48f2ce44b27: Mounted from library/nginx
3bfd54ea739a: Mounted from library/nginx
bf4045499bea: Mounted from library/nginx
1b34d645672f: Mounted from library/nginx
c74e4ebd2844: Mounted from library/nginx
a2d7501dfb35: Mounted from meongchung/php-8.2
latest: digest: sha256:ad5c2efda8ff2783be354d4b2ca00e6cc6d7a7dd81ae18197f53730b557dd6b7 size: 2404

```



- 생성된 Nginx 이미지를 Docker hub의 개인 Repository로 Push

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	Gnuboard - yaml
	
<ul style="list-style-type: none"> <li>- Kubernetes로 Nginx와 PHP Cluster를 활용한 Gnuboard 구축</li> <li>- 6개의 yaml 생성</li> <li>- PHP Cluster IP 고정, 리소스 제한, AutoScaling 기능 등 추가 및 구현</li> <li>- 이때, CPU 사용률이 50% 이상이 되면 Pod는 최대 5개까지 증가할 수 있도록 지정 (부하분산)</li> </ul>	



## 프로젝트 완료 보고서

프로젝트 주제

VPN을 활용한 하이브리드 클라우드 구축

단계 : 프로젝트 완료

작성자 : 김경태

작성일 : 2023.10.04

### ON-PREMISE

### Gnuboard – Proxy Hosts

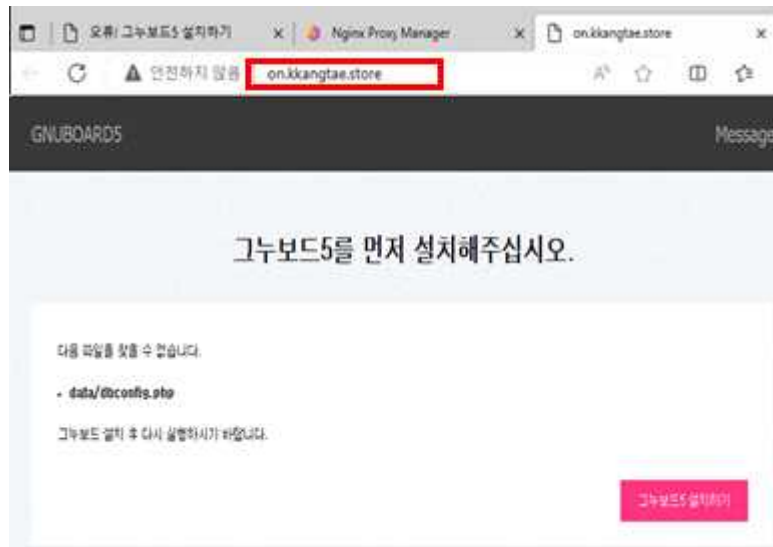
```
[root@master01 nginx-php]# kubectl get po,rs,deploy,svc,pv,pvc -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
pod/nginx-deployment-74668f8c88-4f7mw 1/1 Running 0 9m49s 192.168.101.218 worker01 <none> <none>
pod/nginx-deployment-74668f8c88-ftkcg 1/1 Running 0 9m34s 192.168.101.137 master01 <none> <none>
pod/php-deployment-64dd6c496f-9nhsc 1/1 Running 0 9m49s 192.168.101.88 worker02 <none> <none>
pod/php-deployment-64dd6c496f-r86sk 1/1 Running 0 9m34s 192.168.101.219 worker01 <none> <none>

NAME DESIRED CURRENT READY AGE CONTAINERS IMAGES SELECTOR
replicaset.apps/nginx-deployment-74668f8c88 2 2 2 9m49s nginx-container meongchung/nginx
replicaset.apps/php-deployment-64dd6c496f 2 2 2 9m49s php-fpm-container meongchung/php-8.2.0-2-fpm

NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR
deployment.apps/nginx-deployment 2/2 2 2 9m49s nginx-container meongchung/nginx app=nginx
deployment.apps/php-deployment 2/2 2 2 9m49s php-fpm-container meongchung/php-8.2.0-2-fpm app=php

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE SELECTOR
service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 44h <none>
service/nginx-php-loadbalancer LoadBalancer 10.106.96.30 192.168.100.200 80/TCP,9000/TCP 9m49s app=nginx
service/php-service ClusterIP 10.97.112.94 <none> 9000/TCP 9m49s app=php

NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE VOLUMEMODE
persistentvolume/nfs-pv 10Gi RWX Retain Bound default/nfs-pvc STORAGECLASS REASON AGE VOLUMEMODE
persistentvolumeclaim/nfs-pvc Bound nfs-pv 10Gi RWX 10Gi 9m49s Filesystem
```



- PHP와 연동된 Nginx 웹서비스 실행
- 이때, Nginx Proxy Manager에 접속하여 로드밸런서 외부 IP 192.168.100.200으로 Proxy Host 추가
- on.kkangtae.store 라는 도메인 이름으로 Gnuboard 접속 확인

# 프로젝트 완료 보고서

프로젝트 주제

VPN을 활용한 하이브리드 클라우드 구축

단계 : 프로젝트 완료

작성자 : 김경태

작성일 : 2023.10.04

ON-PREMISE

Gnuboard – AutoScaling

```
root@nginx-deployment-76db77d77f-9w6zm:/# stress --vm 1 -t 1800s
stress: info: [63] dispatching hogs: 0 cpu, 0 io, 1 vm, 0 hdd
```

Every 1.0s: kubectl top node

NAME	CPU(cores)	CPU%	MEMORY(bytes)	MEMORY%
master01	374m	18%	2410Mi	63%
worker01	1039m	51%	2118Mi	55%
worker02	43m	2%	1819Mi	47%

Every 1.0s: kubectl get po,svc,rs,deploy,pv,pvc -o wide

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
pod/nginx-deployment-76db77d77f-9w6zm	1/1	Running	0	3m54s	192.168.101.129	worker01	<none>	<none>
pod/nginx-deployment-76db77d77f-j4kmp	1/1	Running	0	3m58s	192.168.101.66	worker02	<none>	<none>
pod/nginx-deployment-76db77d77f-td8lw	1/1	Running	0	36s	192.168.101.130	worker01	<none>	<none>
pod/nginx-deployment-76db77d77f-v2tcr	1/1	Running	0	36s	192.168.101.193	master01	<none>	<none>
pod/php-deployment-64dd6c496f-8b6sk	1/1	Running	0	3h19m	192.168.101.90	worker02	<none>	<none>
pod/php-deployment-64dd6c496f-r865k	1/1	Running	0	3h19m	192.168.101.219	worker01	<none>	<none>

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	SELECTOR
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	47h	<none>
service/nginx-php-loadbalancer	LoadBalancer	10.106.96.30	192.168.100.200	80:30080/TCP	3h19m	app=nginx
service/php-service	ClusterIP	10.97.112.94	<none>	9000/TCP	3h19m	app=php

NAME	DESIRED	CURRENT	READY	AGE	CONTAINERS	IMAGES	SELECTOR
replicaset.apps/nginx-deployment-76db77d77f	0	0	0	3h19m	nginx-container	meongchung/nginx	app=nginx
replicaset.apps/nginx-deployment-76db77d77f	4	4	4	3m58s	nginx-container	meongchung/nginx	app=nginx
replicaset.apps/php-deployment-64dd6c496f	2	2	2	3h19m	php-fpm-container	meongchung/php-8.2:8.2-fpm	app=php

NAME	READY	UP-TO-DATE	AVAILABLE	AGE	CONTAINERS	IMAGES	SELECTOR
deployment.apps/nginx-deployment	4/4	4	4	3h19m	nginx-container	meongchung/nginx	app=nginx
deployment.apps/php-deployment	2/2	2	2	3h19m	php-fpm-container	meongchung/php-8.2:8.2-fpm	app=php

NAME	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM	STORAGECLASS	REASON	AGE	VOLUME MODE
persistentvolume/nfs-pv	10Gi	RWX	Retain	Bound	default/nfs-pvc			3h19m	Filesystem

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE	VOLUME MODE
persistentvolumeclaim/nfs-pvc	Bound	nfs-pv	10Gi	RWX		3h19m	Filesystem

Every 1.0s: kubectl get po,svc,rs,deploy,pv,pvc -o wide

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
pod/nginx-deployment-76db77d77f-j4kmp	1/1	Running	0	11m	192.168.101.66	worker02	<none>	<none>
pod/nginx-deployment-76db77d77f-v2tcr	1/1	Running	0	7m52s	192.168.101.193	master01	<none>	<none>
pod/nginx-deployment-76db77d77f-td8lw	1/1	Running	0	3h26m	192.168.101.66	worker02	<none>	<none>
pod/php-deployment-64dd6c496f-r865k	1/1	Running	0	3h26m	192.168.101.219	worker01	<none>	<none>

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	SELECTOR
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	47h	<none>
service/nginx-php-loadbalancer	LoadBalancer	10.106.96.30	192.168.100.200	80:30080/TCP	3h26m	app=nginx
service/php-service	ClusterIP	10.97.112.94	<none>	9000/TCP	3h26m	app=php

NAME	DESIRED	CURRENT	READY	AGE	CONTAINERS	IMAGES	SELECTOR
replicaset.apps/nginx-deployment-76db77d77f	0	0	0	3h26m	nginx-container	meongchung/nginx	app=nginx
replicaset.apps/nginx-deployment-76db77d77f	2	2	2	11m	nginx-container	meongchung/nginx	app=nginx
replicaset.apps/php-deployment-64dd6c496f	2	2	2	3h26m	php-fpm-container	meongchung/php-8.2:8.2-fpm	app=php

NAME	READY	UP-TO-DATE	AVAILABLE	AGE	CONTAINERS	IMAGES	SELECTOR
deployment.apps/nginx-deployment	2/2	2	2	3h26m	nginx-container	meongchung/nginx	app=nginx
deployment.apps/php-deployment	2/2	2	2	3h26m	php-fpm-container	meongchung/php-8.2:8.2-fpm	app=php

NAME	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM	STORAGECLASS	REASON	AGE	VOLUME MODE
persistentvolume/nfs-pv	10Gi	RWX	Retain	Bound	default/nfs-pvc			3h26m	Filesystem

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE	VOLUME MODE
persistentvolumeclaim/nfs-pvc	Bound	nfs-pv	10Gi	RWX		3h26m	Filesystem

- 임의의 Nginx Pod에 Stress 명령어를 이용하여 CPU 부하 테스트를 진행
- 부하가 증가함에 따라 Pod가 4개까지 증가함
- 부하테스트 종료 후 기존의 Pod 2개로 복구됨

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE

Gnuboard – Rolling Update

```
[root@master01 nginx-php]# kubectl exec -it pod/nginx-deployment-76db77d77f-j4kmp -- nginx -v
nginx version: nginx/1.25.2
[root@master01 nginx-php]#
[root@master01 nginx-php]# kubectl exec -it pod/nginx-deployment-76db77d77f-v2tcr -- nginx -v
nginx version: nginx/1.25.2
```

```
[root@master01 nginx-php]# kubectl set image deploy nginx-deployment nginx-container=nginx:1.16
deployment.apps/nginx-deployment image updated
```

```
[root@master01 nginx-php]# kubectl get po,rs,deploy,svc --show-labels -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE
pod/nginx-deployment-5b8b95488c-tbpdh	1/1	Running	0	9s	192.168.101.67	worker02	<none>
pod/nginx-deployment-5b8b95488c-xndfh	1/1	Running	0	16s	192.168.101.131	worker01	<none>
pod/php-deployment-64dd6c496f-9nnsd	1/1	Running	0	5h41m	192.168.101.98	worker02	<none>
pod/php-deployment-64dd6c496f-r865k	1/1	Running	0	5h41m	192.168.101.219	worker01	<none>

```
[root@master01 nginx-php]# kubectl get rs --show-labels -o wide
```

NAME	DESIRED	CURRENT	READY	AGE	CONTAINERS	IMAGES
replicaset.apps/nginx-deployment-5b8b95488c -hash=5b8b95488c	2	2	2	16s	nginx-container	nginx:1.16
replicaset.apps/nginx-deployment-74668f8c88 -hash=74668f8c88	0	0	0	5h41m	nginx-container	meongchung/nginx
replicaset.apps/nginx-deployment-76db77d77f -hash=76db77d77f	0	0	0	146m	nginx-container	meongchung/nginx
replicaset.apps/php-deployment-64dd6c496f -hash=64dd6c496f	2	2	2	5h41m	php-fpm-container	meongchung/php-8.1

- 기존의 Nginx 버전은 v1.25.2

- Nginx v1.16으로 Rolling Update를 진행하여 Pod 하나씩 순차적으로 버전 Update 진행

ON-PREMISE	Gnuboard – RollBack
------------	---------------------

```
[root@master01 nginx-php]# kubectl rollout undo deploy nginx-deployment
deployment.apps/nginx-deployment rolled back
```

```
[root@master01 nginx-php]# kubectl get po,rs,deploy,svc --show-labels -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	RE
pod/nginx-deployment-5b8b95488c-xndfh	1/1	Terminating	0	3m15s	192.168.101.131	worker01	<none>	<n
pod/nginx-deployment-76db77d77f-2lqpq	1/1	Running	0	4s	192.168.101.132	worker01	<none>	<n
pod/nginx-deployment-76db77d77f-vm1c7	1/1	Running	0	7s	192.168.101.194	master01	<none>	<n
pod/php-deployment-64dd6c496f-9nnsd	1/1	Running	0	5h44m	192.168.101.98	worker02	<none>	<n
pod/php-deployment-64dd6c496f-r865k	1/1	Running	0	5h44m	192.168.101.219	worker01	<none>	<n

```
[root@master01 nginx-php]# kubectl exec -it pod/nginx-deployment-76db77d77f-2lqpq -- nginx -v
nginx version: nginx/1.25.2
[root@master01 nginx-php]#
[root@master01 nginx-php]# kubectl exec -it pod/nginx-deployment-76db77d77f-vm1c7 -- nginx -v
nginx version: nginx/1.25.2
```

- 반대로 원래 버전(v1.25.2)로 복구하기 위한 RollBack 진행
- Pod 하나씩 순차적으로 Rollback 진행
- 모든 Nginx Pod의 버전이 다시 v1.25.2로 복구된 것을 확인



프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

## ● Site-to-Site VPN

ON-PREMISE	Libreswan
<pre>[root@onpremise ~]# ipsec verify Verifying installed system and configuration files  Version check and ipsec on-path [OK] Libreswan 4.9 Checking for IPsec support in kernel [OK] NETKEY: Testing XFRM related proc values [OK] ICMP default/send_redirects [OK] ICMP default/accept_redirects [OK] XFRM larval drop [OK] Pluto ipsec.conf syntax [OK] Checking rp_filter [OK] Checking that pluto is running [OK] Pluto listening for IKE on udp 500 [OK] Pluto listening for IKE/NAT-T on udp 4500 [OK] Pluto ipsec.secret syntax [OK] Checking 'ip' command [OK] Checking 'iptables' command [OK] Checking 'prelink' command does not interfere with FIPS [OK] Checking for obsolete ipsec.conf options [OK]  [root@onpremise ~]# ip addr 1: lo: &lt;LOOPBACK,UP,LOWER_UP&gt; mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000     link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00     inet 127.0.0.1/8 scope host lo         valid_lft forever preferred_lft forever     inet6 ::1/128 scope host         valid_lft forever preferred_lft forever 2: ens160: &lt;BROADCAST,MULTICAST,UP,LOWER_UP&gt; mtu 1500 qdisc mq state UP group default qlen 1000     link/ether 00:0c:29:e0:ab:a6 brd ff:ff:ff:ff:ff:ff     altname enp3s0     inet 211.35.168.89/24 brd 211.35.168.255 scope global dynamic noprefixroute ens160         valid_lft 6030sec preferred_lft 6030sec     inet6 fe80::20c:29ff:fe0:aba6/64 scope link noprefixroute         valid_lft forever preferred_lft forever 3: ens192: &lt;BROADCAST,MULTICAST,UP,LOWER_UP&gt; mtu 1500 qdisc mq state UP group default qlen 1000     link/ether 00:0c:29:e0:ab:b0 brd ff:ff:ff:ff:ff:ff     altname enp1s0     inet 192.168.100.254/24 brd 192.168.100.255 scope global noprefixroute ens192         valid_lft forever preferred_lft forever     inet6 fe80::5e14:1a20:a977:64c6/64 scope link noprefixroute         valid_lft forever preferred_lft forever  000 "aws-tunnel-1": 0.0.0.0/0==61.82.182.200--61.82.182.254...15.164.105.233&lt;15.164.105.233&gt;==0.0.0.0/0; erouted; 000 "aws-tunnel-1": oriented; my_ip=unset; their_ip=unset; my_updown=ipsec_updown; 000 "aws-tunnel-1": xauth us:none, xauth them:none, my_username=[any]; their_username=[any] 000 "aws-tunnel-1": our_auth=secret, their_auth=secret 000 "aws-tunnel-1": modecfg info: us:none, them:none; modecfg policy:push, dns:unset, domains:unset, cat:unset; 000 "aws-tunnel-1": sec_label=unset; 000 "aws-tunnel-1": ike life: 28800s; ipsec life: 3600s; replay window: 32; rekey margin: 540s; rekey fuzz: 100%; 000 "aws-tunnel-1": retransmit interval: 500ms; retransmit timeout: 60s; iketcp: no; iketcp port: 4500; 000 "aws-tunnel-1": initial contact: no; cisco-unix: no; fake-strongswan: no; send vendorid: no; send no-esp-tfc: no; 000 "aws-tunnel-1": policy: IKEv2+PSK+ENCRYPT+TUNNEL+PFS+UP+IKE_FRAG_ALLOW+ESN_NO; 000 "aws-tunnel-1": v2-auth-hash-policy: none; 000 "aws-tunnel-1": conn_prio: 0.0; interface: ens160; metric: 0; mtu: unset; sa prio:auto; sa tfc:none; 000 "aws-tunnel-1": nftlog-group: unset; mark: 5/0xffffffff, 5/0xffffffff; vti-iface: vti1; vti-routing: no; vti-share: 000 "aws-tunnel-1": our idtype: ID_IPV4_ADDR; our id=61.82.182.200; their idtype: ID_IPV4_ADDR; their id=15.164.10 000 "aws-tunnel-1": ddp: action:restart; delay:10; timeout:30; nat-t: encaps:yes; nat_keepalive:yes; ikev1_natt:bo 000 "aws-tunnel-1": newest ISAKMP SA: #1; newest IPsec SA: #2; conn serial: \$1; 000 "aws-tunnel-1": IKE algorithms: AES_CBC_256-HMAC_SHA2_256-MOOP2048 000 "aws-tunnel-1": IKEv2 algorithm newest: AES_CBC_256-HMAC_SHA2_256-MOOP2048 000 "aws-tunnel-1": ESP algorithms: AES_GCM_16-NONE 000 "aws-tunnel-1": ESP algorithm newest: AES_GCM_16_128-NONE; pfsgroup=&lt;Phase1&gt; 000 000 Total IPsec connections: loaded 1, active 1 000 000 State Information: DoS cookies not required. Accepting new IKE connections 000 IKE SAs: total(1), half-open(0), open(0), authenticated(1), anonymous(0) 000 IPsec SAs: total(1), authenticated(1), anonymous(0) 000 000 #1: "aws-tunnel-1":4500 STATE V2 ESTABLISHED IKE SA (established IKE SA); REKEY in 28042s; newest ISAKMP: idle; 000 #2: "aws-tunnel-1":4500 STATE V2 ESTABLISHED CHILD SA (established Child SA); REKEY in 2601s; newest IPsec: erro 000 #2: "aws-tunnel-1" esp.c3947aa1@15.164.105.233 esp.7d3680ba@61.82.182.200 tun.0015.164.105.233 tun.0061.82.182.2 000 000 Bare Shunt list: 000 [root@localhost ~]#</pre>	
<ul style="list-style-type: none"> <li>- IPsec 및 가상 사설 네트워크 VPN 연결을 위해 Libreswan 설정</li> <li>- 'Onpremise' VM에서 진행</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	AWS - CGW
<div>VPC &gt; 고객 게이트웨이 &gt; 고객 게이트웨이 생성</div> <div>고객 게이트웨이 생성 <small>정보</small></div> <div>고객 게이트웨이는 AWS에서 생성하는 리소스로, 온프레미스 네트워크의 고객 게이트웨이 디바이스를 나타냅니다.</div> <div><div>세부 정보</div><div><div>이름 태그 - 선택 사항</div><div>'이름'인 키와 사용자가 지정하는 값을 사용하여 태그를 생성합니다.</div><div>cgw-project</div><div>값의 길이가 256자 이하여야 합니다.</div><div>BGP ASN 정보</div><div>고객 게이트웨이 디바이스의 ASN입니다.</div><div>65000</div><div>값은 1~2147483647의 범위여야 합니다.</div><div>IP 주소 정보</div><div>고객 게이트웨이 디바이스의 외부 인터페이스에 대한 IP 주소를 지정합니다.</div><div>211.35.168.89</div><div>인증서 ARN</div><div>AWS Certificate Manager(ACM)에 프로비저닝된 프라이빗 인증서의 ARN입니다.</div><div>인증서 ARN 선택</div><div>디바이스 - 선택 사항</div><div>고객 게이트웨이 디바이스의 이름을 입력합니다.</div><div>디바이스 이름 입력</div></div></div>	
<div>- AWS에서 고객 게이트웨이 생성</div> <div>- 이때 IP주소는 'Onpremise' VM의 Public IP를 지정</div>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE

AWS - VPG

VPC > 가상 프라이빗 게이트웨이 > 가상 프라이빗 게이트웨이 생성

가상 프라이빗 게이트웨이 생성

가상 프라이빗 게이트웨이는 Site-to-Site VPN 연결의 Amazon 측에 있는 VPN 접선기입니다.

세부 정보

이름 태그 - 선택 사항

이름은 키와 사용자가 지정한 값을 사용하여 태그를 생성합니다.

vpg-project

값의 길이가 256자 이하여야 합니다

자물 시스템 번호(ASN)

Amazon 기본 ASN

사용자 지정 ASN

태그

태그는 사용자가 AWS 리소스에 할당하는 레이블입니다. 각 태그는 키와 선택적 값으로 구성됩니다. 태그를 사용하여 리소스를 분류 및 필터링하거나 AWS 비용을 추적할 수 있습니다. 이름 태그는 리소스를 일괄 추적하도록 프로젝트로 추가하는 것이 좋습니다.

키

값 - 선택 사항

Q Name

X

Q vpg-project

X

제거

새로운 태그 추가

태그 이름(필수)에 대해 추가할 수 있음

취소

가상 프라이빗 게이트웨이 생성

가상 프라이빗 게이트웨이 (1/1) 정보

가상 프라이빗 게이트웨이 목록

가상 프라이빗 게이트웨이 ID: vgw-046c0430c7804f40a X

필터 지우기

Name	가상 프라이빗 게이트웨이 ID	상태	유형
vpg-project	vgw-046c0430c7804f40a	Attached	ipsec.1

- 가상 사설 게이트웨이 생성

- AWS의 VPC와 연결하는 것이 목적

페이지 70 / 78

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	AWS – VPN 연결
<div><div><div><div><div><div><span>VPC &gt; VPN 연결 &gt; VPN 연결 생성</span></div></div></div><div><div><div><span>VPN 연결 생성</span></div><div><span>정보</span></div></div><div>site-to-site VPN 연결에 사용하려는 리소스 및 추가 구성 옵션을 선택합니다.</div></div></div><div><div><div><div>세부 정보</div></div><div><div>이름 태그 - 선택 사항 이름인 키와 사용자가 지정하는 값을 사용하여 태그를 생성합니다. 값의 길이가 256자 이하여야 합니다.</div><div>VPN-project</div></div><div><div>대상 게이트웨이 유형 정보</div><div><div><input checked="" type="radio"/> 가상 프라이빗 게이트웨이</div><div><input type="radio"/> Transit Gateway</div><div><input type="radio"/> 연결되지 않음</div></div><div>가상 프라이빗 게이트웨이</div><div>vgw-046c0430c7804f40a / vpg-project</div></div><div><div>고객 게이트웨이 정보</div><div><div><input checked="" type="radio"/> 기존</div><div><input type="radio"/> 신규</div></div><div>고객 게이트웨이 ID</div><div>cgw-06403d3d9b3e99b8d / cgw-project</div></div><div><div>라우팅 옵션 정보</div><div><div><input checked="" type="radio"/> 동적(BGP 필요)</div><div><input type="radio"/> 정적</div></div></div></div></div><div><div><div>터널 1 옵션 - 선택 사항 정보</div></div><div><div>터널 1의 내부 IPv4 CIDR Amazon에서 선택 10.0.254.0/24 범위의 256개 192.168.0.0/24 IPv4 CIDR 블록</div><div>터널 1의 사설 공유 키 가상 프라이빗 게이트웨이와 고객 게이트웨이의 간에 초기 인증을 설정하기 위한 사설 공유 키(PSK)입니다. Amazon에서 선택 사설 공유 키를 64자 이하로 만듭니다. 최소한 문자는 하나, a-z, 0-9, _를 포함해야 합니다. 처음 30자를 사용할 수 있습니다.</div><div>터널 1의 고급 옵션 <div><div><input checked="" type="radio"/> 기본 옵션 사용</div><div><input type="radio"/> 터널 1 옵션 편집</div></div><div>VPN 로깅 정보 터널 활동 로그 터널 활동 로그는 VPN 활동 및 IPsec 프로토콜 메시지에 대한 로그 메시지를 포함합니다. <input type="checkbox"/> 활성화</div><div>터널 유지 관리 터널 엔드포인트 수명 주기 제어 정보 터널 엔드포인트 수명 주기 제어는 엔드포인트로 교체될 때까지 유지합니다. <input type="checkbox"/> 제거</div></div></div><div><div><div>터널 2 옵션 - 선택 사항 정보</div></div><div><div>태그 태그를 사용하여 AWS 리소스에 설명하는 정보를 만듭니다. 각 태그는 키-값 쌍으로 구성됩니다. 태그를 사용하여 리소스를 쉽게 찾고 분류하거나 AWS 비용을 추적할 수 있습니다. 이름 태그는 리소스를 쉽게 추적하도록 도와줍니다. 추가하는 것이 좋습니다.</div><div><div>키</div><div>값 - 선택 사항</div><div><div>Q Name X</div><div>Q VPN-project X</div><div>제거</div></div><div>새로운 태그 추가</div><div>태그 이름: 40자 이하의 문자열</div></div></div><div><div>취소</div><div>VPN 연결 생성</div></div></div></div></div></div>	
<div><div><div><div>- 온프레미스 네트워크와 AWS VPC 간의 안전한 통신을 위한 VPN 연결 생성</div><div>- 생성했던 VPG와 CGW를 지정</div><div>- 라우팅 옵션 : 동적(BGP)</div></div></div></div>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE

AWS – 라우팅 전파

[VPC](#) > [라우팅 테이블](#) > rtb-0cb10c7e6990c9bfc

라우팅 테이블 ID

기본

명시적 서브넷 연결

엣지 연결

VPC

소유자 ID

2 서브넷

-

rtb-0cb10c7e6990c9bfc

예

vpc-083d35d8053b68a4a | sk-vpc

456050757969

라우팅

서브넷 연결

엣지 연결

라우팅 전파

태그

라우팅 전파 (1)

라우팅 전파 편집

가상 프라이빗 게이트웨이 찾기

< 1 > ⚙

가상 프라이빗 게이트웨이

전파

vgw-046c0430c7804f40a / vpg-project

예

- 해당 라우팅 테이블에서 라우팅 전파 기능 활성화
- Public Cloud AWS와 온프레미스 네트워크 간의 원활한 통신
- VPG가 자동으로 라우팅 테이블에게 라우팅 정보를 전파
- 즉, 라우팅 테이블에 수동으로 VPN 라우팅 정보를 입력할 필요가 없음



프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	BGP
<pre>[root@onpremise ~]# vtysh Hello, this is FRRouting (version 7.5.1). Copyright 1996-2005 Kunihiro Ishiguro, et al.  Frr# show ip route Codes: K - kernel route, C - connected, S - static, R - RIP,        O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,        T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,        F - PBR, f - OpenFabric,        &gt; - selected route, * - FIB route, q - queued, r - rejected, b - backup  K&gt;* 0.0.0.0/0 [0/100] via 211.35.168.254, ens160, src 211.35.168.89, 02:40:36 C&gt;* 169.254.210.84/30 is directly connected, vtil, 02:05:32 C&gt;* 172.18.0.0/16 is directly connected, br-1040b26c52e8, 02:40:34 K&gt;* 192.168.0.0/22 [0/0] via 169.254.210.85, vtil, 01:52:35 B 192.168.0.0/22 [20/100] via 169.254.210.85, vtil, weight 1, 02:00:32 C&gt;* 192.168.100.0/24 is directly connected, ens192, 02:40:36 C&gt;* 211.35.168.0/24 is directly connected, ens160, 02:40:36 Frr# exit</pre>	
<ul style="list-style-type: none"> <li>- 'Onpremise' VM에서 FRR로 접근 (vtyos)</li> <li>- BGP 라우팅 프로토콜 선언</li> <li>- AWS Public IP 대역(169.254.210.84/30)과 Connected 연결로 통신 가능</li> <li>- AWS Private IP 대역(192.168.0.0/22)과 BGP로 통신 가능</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	Test 1
<pre>[root@Onpremise ~]# ping 192.168.1.97 PING 192.168.1.97 (192.168.1.97) 56(84) bytes of data. 64 bytes from 192.168.1.97: icmp_seq=1 ttl=254 time=6.34 ms 64 bytes from 192.168.1.97: icmp_seq=2 ttl=254 time=6.00 ms 64 bytes from 192.168.1.97: icmp_seq=3 ttl=254 time=6.64 ms 64 bytes from 192.168.1.97: icmp_seq=4 ttl=254 time=6.17 ms</pre> <pre>[root@Onpremise ~]# ping 192.168.2.242 PING 192.168.2.242 (192.168.2.242) 56(84) bytes of data. 64 bytes from 192.168.2.242: icmp_seq=1 ttl=254 time=6.100 ms 64 bytes from 192.168.2.242: icmp_seq=2 ttl=254 time=6.47 ms 64 bytes from 192.168.2.242: icmp_seq=3 ttl=254 time=10.1 ms 64 bytes from 192.168.2.242: icmp_seq=4 ttl=254 time=8.77 ms 64 bytes from 192.168.2.242: icmp_seq=5 ttl=254 time=7.08 ms</pre> <pre>[root@DB .ssh]# ls -l total 8 -rw-----. 1 root root 1674 Sep 21 10:42 4_Group_key.pem -rw-r--r--. 1 root root 175 Sep 21 10:45 known_hosts [root@DB .ssh]# ssh -i 4_Group_key.pem ec2-user@192.168.0.190 Last login: Thu Sep 21 01:52:01 2023 from 169.254.210.86   _ _   ( _ _ )  _   ( _ _ /   Amazon Linux 2 AMI  _   \ _ _   _    https://aws.amazon.com/amazon-linux-2/ No packages needed for security; 2 packages available Run "sudo yum update" to apply all updates. [ec2-user@ip-192-168-0-190 ~]\$</pre>	
<ul style="list-style-type: none"> <li>- ICMP ping test 진행</li> <li>- 'Onpremise' VM에서 AWS 인스턴스의 Private IP로 통신 가능</li> <li>- 동일한 key pair를 가지고 있는 상태에서 DB(VMnet1) VM에서 AWS 인스턴스(web-1)의 Private IP로 SSH 접속 test 진행 및 성공</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE

Test 2

인스턴스 (1/5) 정보

인스턴스 상태: running

필터 지우기

Name	인스턴스 ID	인스턴스 상태	인스턴스 유형	상태 검사	경보 상태	가용 영역	퍼블릭 IPv4 DNS
NAT-instance	i-07ff9f7f64595d2b	실행 중	t2.micro	2/2개 검사 통과...	경보 없음	ap-northeast-2a	ec2-52-78-236-220.ap-...
PHP-1	i-035d077e852a130e2	실행 중	t2.micro	2/2개 검사 통과...	경보 없음	ap-northeast-2a	-
web-1	i-0cea83c5fa1444c9d	실행 중	t2.micro	2/2개 검사 통과...	경보 없음	ap-northeast-2a	ec2-13-124-151-32.ap-...

인스턴스: i-0cea83c5fa1444c9d(web-1)

인스턴스 요약 정보

인스턴스 ID

i-0cea83c5fa1444c9d (web-1)

퍼블릭 IPv4 주소

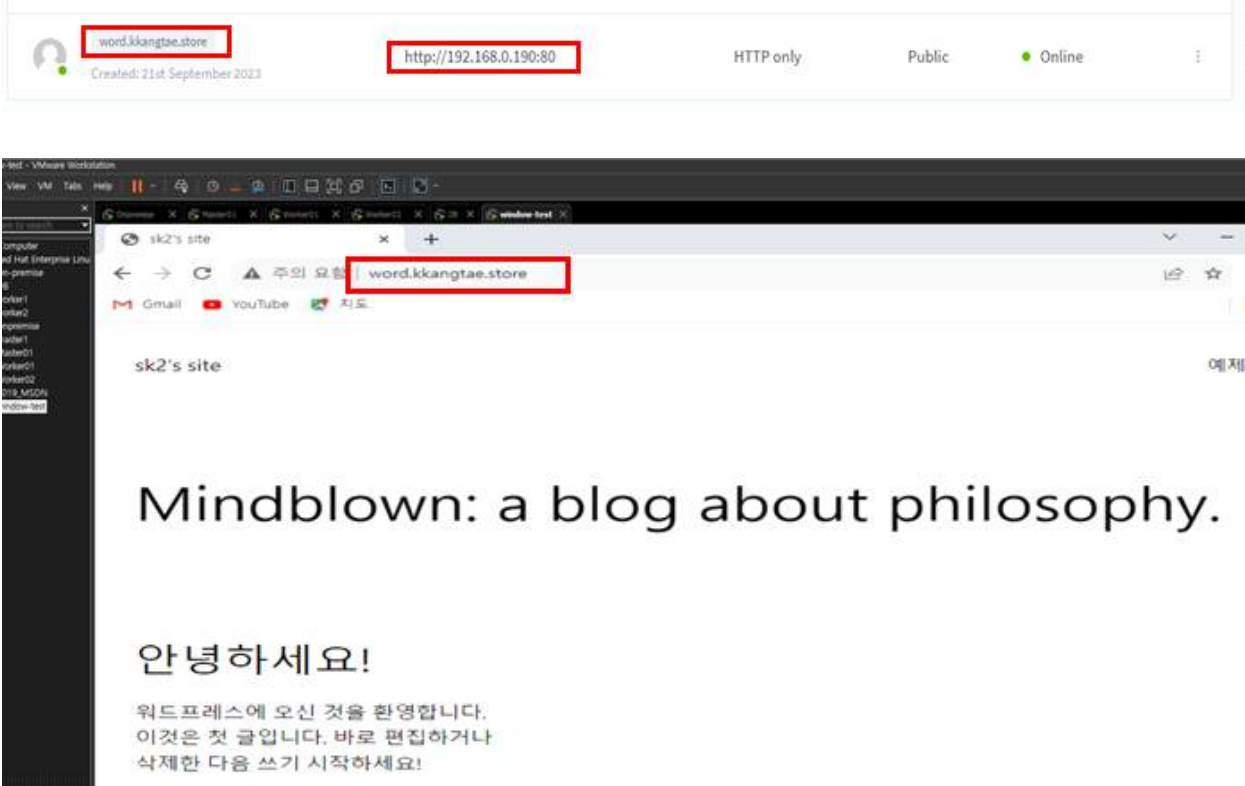
13.124.151.32 [개방 주소법](#)

프라이빗 IPv4 주소

192.168.0.190

- 'web-1' AWS 인스턴스의 Private IP로 웹접속 테스트를 위해 온프레미스 환경의 VMnet1 네트워크만을 가진 임의의 Windows VM 생성 후 Test
- 정상적으로 Wordpress에 접속

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

ON-PREMISE	Test 3
	
<ul style="list-style-type: none"> <li>- Nginx Proxy Manager에 해당 Private IP로 Proxy Host를 생성 (word.kkangtae.store)</li> <li>- 해당 도메인 이름으로 Wordpress 접속 가능</li> </ul>	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

#### 4. 프로젝트 일정

항목	세부 내용	어플리케이션	일정	담당	비고
요구 분석	요구사항 분석		2023-08-21 ~ 2023-08-22	서희경	
	프로젝트 분석 및 설계	Excel	2023-08-21 ~ 2023-08-22	김학남	
설계	AWS 구성	Powerpoint	2023-08-22 ~ 2023-08-23	김경태	
	On-premise 구성	Powerpoint	2023-08-24 ~ 2023-08-25	서희경	
하이브리드 클라우드 구축	AWS Network 구축	VPC, EC2, RDS, Route53, EFS 등	2023-08-26 ~ 2023-09-06	김경태	
	On-premise 환경 구축	K8s, Docker, MariaDB, NFS, VMware, Linux 등	2023-09-07 ~ 2023-09-26	서희경	
	VPN 구성 및 연결	Site-to-Site VPN, FRR, BGP, IPsec, VMware, Linux 등	2023-09-27 ~ 2023-10-01	김학남	
테스트	웹서비스 접속 및 디버깅		2023-10-01 ~ 2023-10-02	김학남	
보고서 작성, 발표		Powerpoint, Excel, hwp	2023-10-03 ~ 2023-10-04	김경태	

프로젝트 완료 보고서		
프로젝트 주제	VPN을 활용한 하이브리드 클라우드 구축	
단계 : 프로젝트 완료	작성자 : 김경태	작성일 : 2023.10.04

## 5. 피드백

### ● 미달성 목표

- 가) HAproxy를 통한 WEB 이중화 구성 실패
  - HAproxy 설치 및 테스트는 VMWARE Workstation Local Network에서 완료
  - 이후, 인프라에 적용 시 134.100.X.X 대역과 연동 실패(원인 분석 중)
- 나) Youtube OPEN API를 활용한 Youtube 실시간 인기동영상 크롤링 기능 개발 실패
  - 개발 스킬 부족으로 인한 기능 구현 실패
- 다) 커뮤니티 기능 중 게시판 기능 구현 실패
  - 개발 스킬 및 백엔드 구성에 대한 이해 부족으로 기능 구현 실패
- 라) 운영 ↔ DR간 데이터 실시간 데이터 동기화 구현 실패
  - Third Party App 없이 스케줄링으로 구현하기는 한계가 있어 대책 강구 필요.

### ● 향후 계획

- 가) Trouble Shooting을 통해 HAproxy Fault에 대한 원인 분석 완료 후,  
재구성 하여 WEB 서버 이중화 구현(WEB 이중화 구현 시, 서비스 단 이중화 구성 완료)
- 나, 다) 개발에 대한 기술적인 보완 및 백엔드 인프라에 대한 전반적인 이해 필요.  
이후, 개발을 통한 초기 기획 서비스 구축 완료
- 라) CDC나 Veritas 같은 상용 솔루션 외에 방식