

구름 쿠버네티스 전문가 양성과정 세미프로젝트

뭉게구름조(5조)

웹 인프라 이중화 프로젝트



목차

1. 프로젝트 개요
2. 팀 소개 및 프로젝트 환경
3. 일정 및 문제 정의
4. 프로젝트 구성 및 구현
 - 4.1. DNS
 - 4.2. Load balancing
 - 4.3. Web
 - 4.4. DB
 - 4.5. etc
5. 결론

1. 프로젝트 개요



프로젝트 컨셉 시나리오 (프로젝트 목적)

구름EDU의 수업 후기를 적는 서비스를 담당하는 구름의 뭉게구름팀

구름EDU의 쿠버네티스 양성과정은 8기까지 진행되었으며 향후 계속해서 진행될 예정이다. 이 과정속에서 교육을 듣는 수강생들의 수많은 스터디 기록들을 더욱 안전하고 좋은 서비스로 제공하는것을 목표로 서비스를 하는 중카카오의 데이터서버 화재의 이슈를 계기로 뭉게구름팀이 맡고 있는 구름의 수업후기를 적는 서비스의 가용성을높이기 위하여 이중화 프로젝트를 실시하기로 하였다.

프로젝트 이름은 **뭉게뭉게구름프로젝트**로 정했다.

프로젝트 목표

몽게몽게구름프로젝트는 주어진 일주일 이라는 시간을 가지고 현재 서비스중인 서버를 이중화를 작업하는것을우선으로 하고 **HA PROXY**를 이용한 로드밸런서 구축 및 이중화 스토리지 구축 등을 더해서 더 나은 서비스를 제공하는것을 목표로 한다.

2. 팀 소개 및 프로젝트 환경



뭉게구름 팀 구성

팀 구성

팀장: 정한교

팀원: 김지원, 배수빈, 이승수

구성원 역할

정한교: 네트워크 구성, 자료 정리

김지원: 웹 서버 구축, 자료 정리

배수빈: DNS 서버 구축, 프론트엔드

이승수: DB 서버 구축, 프론트엔드

프로젝트 활용 핵심 툴, 어플리케이션, 운영체제

2. 팀 소개 및 프로젝트 환경

FreeBSD 기반 방화벽 및
라우팅 오픈소스 소프트웨어



오픈소스 베어메탈
하이퍼바이저



동시접속 처리에 특화된 웹
서버 프로그램



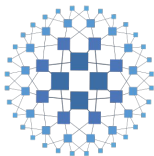
PHP기반 MySQL과
MariaDB 관리 도구



텍스트, 이미지, 비디오,
음성 커뮤니케이션에
특화된 어플리케이션



HAPROXY
하드웨어 L4, L7 스위치
대안 오픈소스 로드밸런서



PHP기반 웹 CMS

MariaDB
Foundation
MySQL 기반 오픈소스
RDBMS



Notion

프로젝트 관리 및 기록
소프트웨어

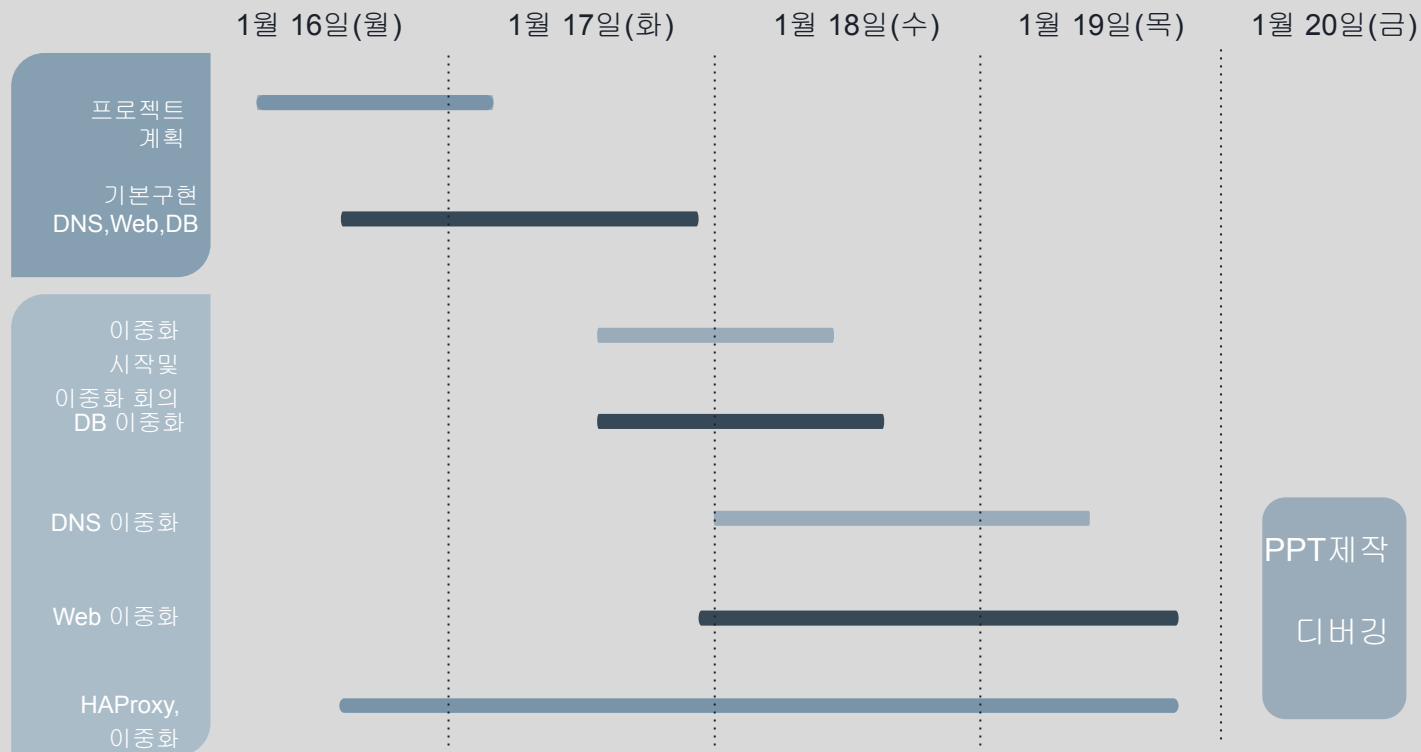


레드햇 엔터프라이즈 리눅스(RHEL)
에서 파생된 리눅스계 OS

3. 일정 및 문제정의



프로젝트 일정



AS - IS

현재 뭉게구름조의 서비스는 단일 서버체제의 운용을 하고 있고 점점 이용자들이 늘어나고 있다

문제점을 나열해 보면

1. 서비스 장애 발생 가능성이 높음
2. 갈수록 많아지는 트래픽
3. 데이터베이스의 손실 및 훼손시 백업 부재
4. 공유 서버의 부재 및 스토리지 백업부재
5. http 사용으로 보안성이 낮음

등등의 문제를 가지고 있다.

TO - BE

이중화 혹은 그 이상의 중첩된 서버를 이용해서고가용성 서비스를 지향하면서 웹 서버의 단순히 수용 트래픽을 늘리는 것이 아닌 **HAProxy** 로드밸런싱을 통해 트래픽을 분산시켜서 트래픽을 감당하고, 저장 자료의 백업을 주기적으로 만들어서 자료 훼손 및 손실을 방지하고 **https** 인증서를 적용 및 서버의 방화벽, 권한설정을 검토해서 보안성을 높이도록 한다.

현재 서비스는 연간 7.31의 다운타임을 보였지만 뭉게뭉게구름 프로젝트를 진행하고 난 후의 가용성은 연간 다운타임이 3.65인 99%의 가용성을 가질것으로 기대효과를 가지고 있다.

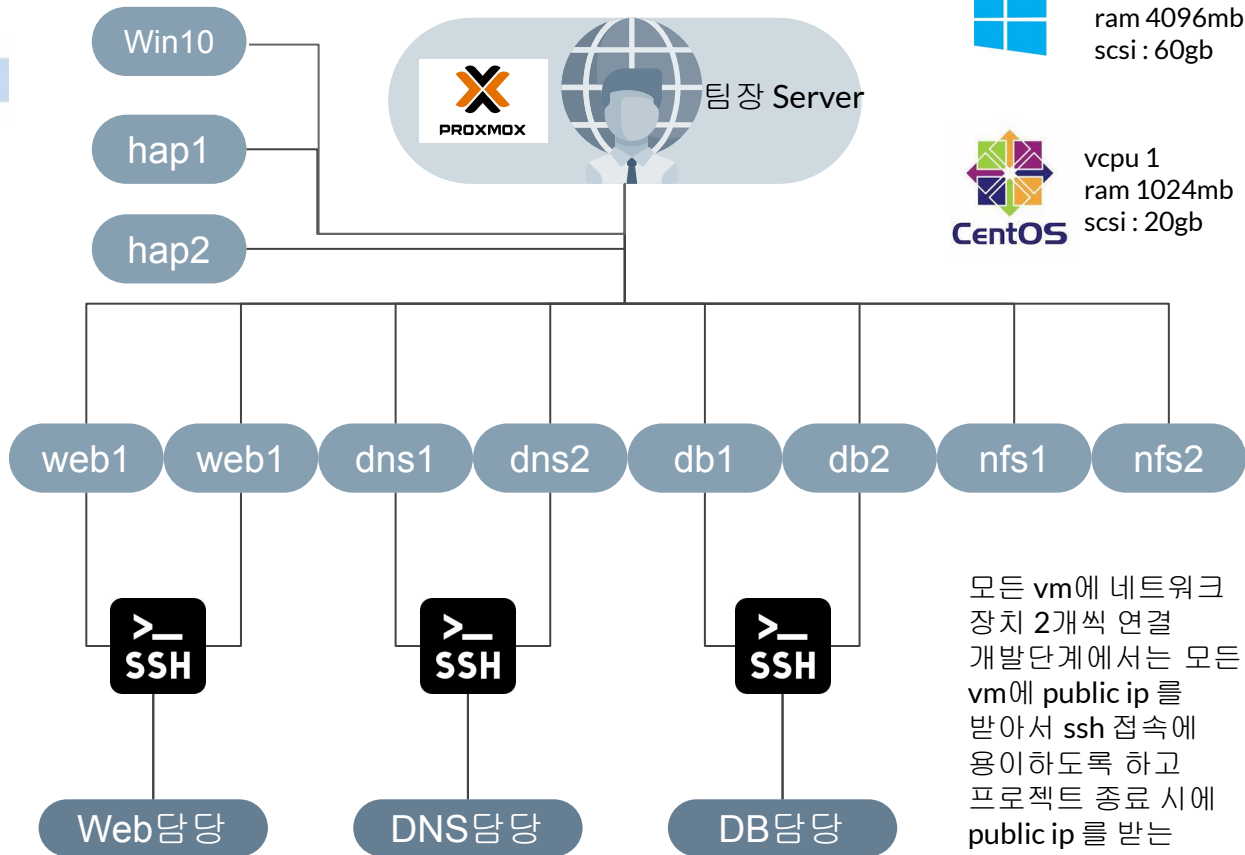
4. 프로젝트 구성과정 및 구현



운영 시스템 구성

데이터센터

- 노드
- 가상머신
 - 100 (windows10-test)
 - 109 (Opsense)
 - 400 (centos7-web1-pj)
 - 402 (centos7-dns1-pj)
 - 702 (centos7-web2-pj)
 - 801 (centos7-dns1-pj)
 - 802 (centos7-dns2-pj)
 - 901 (centos7-db1-pj)
 - 902 (centos7-db2-pj)
 - 1001 (centos7-hap1-pj)
 - 1002 (centos7-hap2-pj)
 - 1101 (centos7-nfs-pj)
 - 1102 (centos7-nfs2-pj)
 - 1202 (openvpn)
 - 903 (centos7-clean)
- 스토리지



모든 vm에 네트워크 장치 2개씩 연결
개발단계에서는 모든 vm에 public ip 를 받아서 ssh 접속에 용이하도록 하고
프로젝트 종료 시에 public ip 를 받는 네트워크 장치를 삭제하고 private ip만 받게 하였다.

운영 시스템 구성



정한교 2023.01.16. 오후 2:59
webserver VM Primary : ssh root@118.46.50.135
내부 : 192.168.1.110
webserver VM Secondary : ssh root@118.46.49.45
내부 : 192.168.1.111

database VM Primary : ssh root@118.46.49.13
내부 : 192.168.1.103
database VM Secondary : ssh root@118.46.49.40
내부 : 192.168.1.104

DNS VM Primary : ssh root@121.153.139.131
ssh-root@118-46-50-36
내부 : 192.168.1.108
DNS VM Secondary : ssh root@118.46.49.21
내부 : 192.168.1.109

HAProxy VM Primary : 118.46.49.138
내부 : 192.168.1.106
HAProxy VM Secondary : 118.46.49.56
내부 : 192.168.1.116

VIP : 192.168.1.200

nfs-storage + FTP vm : ssh root@118.46.49.27
내부 : 192.168.1.105
nfs-srorage-backup vm
내부 : 192.168.1.122 (수정됨)



배수빈 2023.01.18. 오후 4:26
흠 그렇군요
죄송한데 118.46.50.30 vm한번만 더 확인해주실수 있나요? 계속 connention failed가 뜨네요..ㅠㅠ



정한교 2023.01.18. 오후 4:32
ssh 접속이안되네요
잠시만요



배수빈 2023.01.18. 오후 4:32
118.46.49.21도 같이 부탁드려요... 애도 안되네요..



정한교 2023.01.18. 오후 4:32
네



배수빈 2023.01.18. 오후 4:32
넵,,!



이승수 2023.01.18. 오후 4:32
저도 db slave만 들어가지고 master가 안되네요...



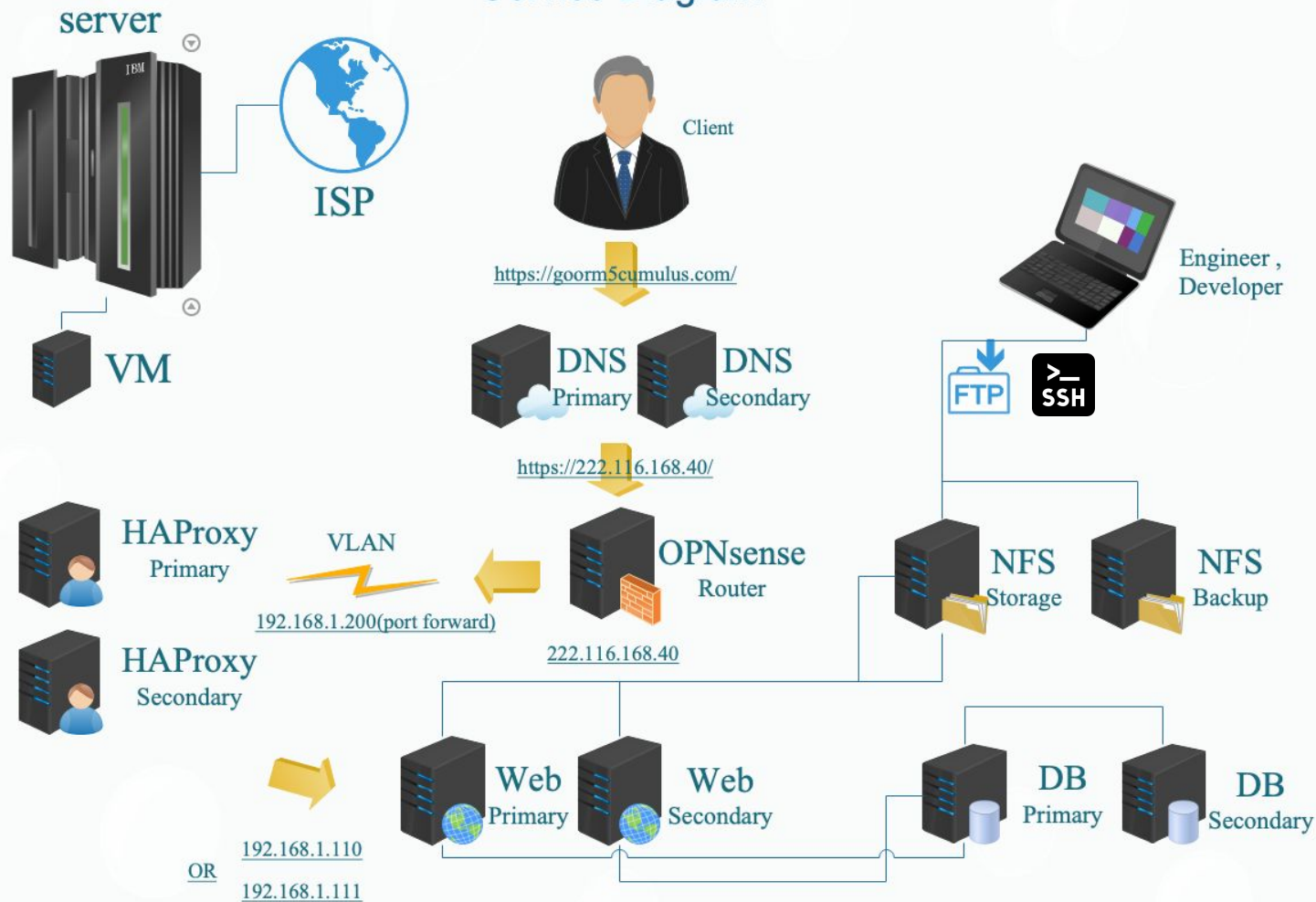
정한교 2023.01.18. 오후 4:37
118.46.50.30 은 재부팅하니까 되네요
db도 이제 두개다 되고요

프로젝트의 협업툴로 지정한 디스코드와 노션을 사용해서 잊어버리기 쉬운 서버의 ip를 저장하고 팀원간의 소통으로 시스템을 지속적으로 관리한다



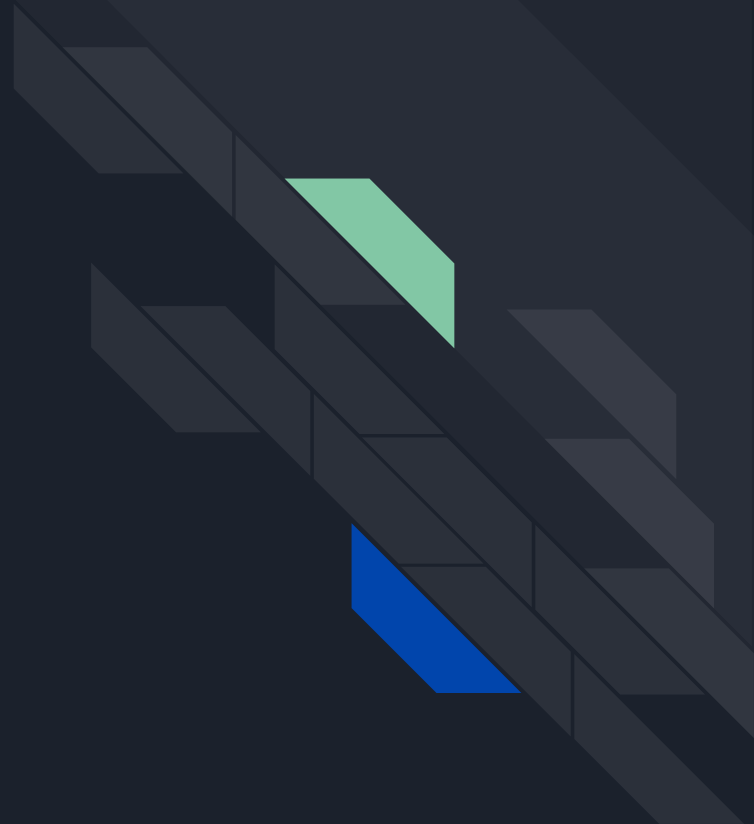
VM (Virtual Machine) ...							
Aa 이름	os	ip	설치된 패키지	cpu	disk	ram	날짜
webserver1	centos7	118.46.50.135 / 192.168.1.110	wordpress nginx	1 core	20gb scsi	2048mb	2023년 1월 16일
database1	centos7	125.140.173.233 / 192.168.1.103	mariadb	1 core	20gb scsi	1024 mb	2023년 1월 16일
dns1	centos7	121.153.139.131 / 192.168.1.108	bind	1 core	20gb scsi	1024 mb	2023년 1월 16일
HAProxy1	centos7	118.46.49.138 / 192.168.1.106	haproxy keepalive	1 core	20gb scsi	1024 mb	2023년 1월 17일

Service Diagram



4-1 DNS

DNS 서버 구현 및 이중화



DNS 구현 진행방향

DNS서버 이중화

Master DNS서버 장애 발생시 Slave DNS 서버가 대신 서비스 제공

goorm5cumulus.com → 222.116.168.40

Primary DNS 서버

외부IP : 121.153.139.131

내부IP : 192.168.1.108

Secondary DNS 서버

외부IP : 118.46.49.21

내부IP : 192.168.1.109

HAProxy 서버

외부IP : 118.46.49.56

내부IP : 192.168.1.116

DNS 구현 진행순서

Primary DNS 서버구축

- 기본설정
- 도메인 zone 설정
- 도메인 네임서버 지정

Secondary DNS 서버구축

- 기본설정
- 도메인 zone 설정

Primary DNS 서버 구축

기본 설정

1. Bind 패키지 설치

```
# yum -y install bind bind-chroot bind-utils
```

2. 방화벽 설정

```
[root@localhost ~]# firewall-cmd --permanent --add-service=dns
success
[root@localhost ~]# firewall-cmd --reload
success
[root@localhost ~]# firewall-cmd --info-service=dns
dns
  ports: 53/tcp 53/udp
  protocols:
  source-ports:
  modules:
  destination:
```

Primary DNS 서버 구축

도메인 zone 설정

1. named.conf 설정

경로 : /etc/named.conf

- option 설정 변경
- Domain zone 정보 추가

2. Zone 파일 생성

경로 : /var/named/goorm5cumulus.com

- Zone 파일 레코드 수정

Primary DNS 서버 구축

도메인 zone 설정

1. named.conf 설정

- option 설정 변경

```
12 options {  
13     listen-on port 53 { any; };           // ipv4, 53번 포트 외부 접근 허용  
14     listen-on-v6 port 53 { none; };       // ipv6, 외부접근 차단  
15     directory      "/var/named";  
16     dump-file       "/var/named/data/cache_dump.db";  
17     statistics-file "/var/named/data/named_stats.txt";  
18     memstatistics-file "/var/named/data/named_mem_stats.txt";  
19     recursing-file  "/var/named/data/named.recursing";  
20     secroots-file   "/var/named/data/named.secroots";  
21     allow-query     { any; };             // 외부의 DNS쿼리 요청 허용  
22     masterfile-format text;               // slave에서 zone파일 동기화시 text변환
```

Primary DNS 서버 구축

도메인 zone 설정

1. named.conf 설정

- Domain zone 정보 추가

```
zone "goorm5cumulus.com" IN {  
    type master;  
    file "goorm5cumulus.com.zone";  
    also-notify { 118.46.49.21; };  
};
```

Primary DNS 서버 구축

도메인 네임서버 지정

DNS NameServer 등록

`/etc/resolv.conf`

서버가 등록된 DNS를 찾을 수 있도록 DNS서버 IP 추가

*DNS서버를 비롯한 다른 VM에도 설정

```
# Generated by NetworkManager
search goorm5cumulus.com
nameserver 121.153.139.131
nameserver 168.126.63.1
nameserver 168.126.63.2
```

Secondary DNS 서버 구축

도메인 zone 설정

named.conf 설정

1. options 설정 변경

Master DNS 서버와 동일하게 수정

2. Zone 설정

```
zone "goorm5cumulus.com" IN {  
    type slave;  
    masters { 121.153.139.131; };  
    file "/var/named/slaves/goorm5cumulus.com.zone";  
};
```


DNS 설정확인

- Primary, Secondary 동기화 확인

```
[root@localhost ~]# ls -al /var/named/slaves/
total 8
drwxrwx---. 2 named named 61 Jan 17 17:39 .
drwxrwx--T. 6 root  named 141 Jan 19 10:56 ..
-rw-r--r--. 1 named named 312 Jan 17 17:27 goorm5cumulus.com
-rw-r--r--. 1 named named 355 Jan 18 17:05 goorm5cumulus.com.zone
```

- nslookup으로 도메인 조회

```
[root@server_one ~]# nslookup
> goorm5cumulus.com
Server:          121.153.139.131
Address:         121.153.139.131#53

Name:   goorm5cumulus.com
Address: 118.46.49.56
```

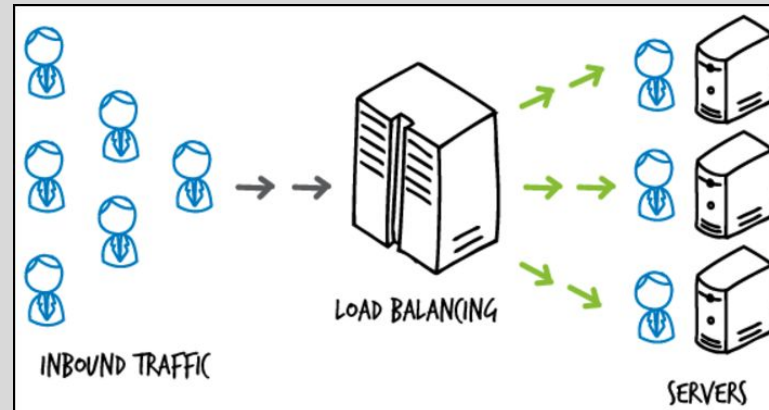
4-2 Road Balance



Load Balancing

로드밸런싱이란

서버가 처리해야 할 업무 혹은 요청(Load)을 여러 대의 서버로 나누어(Balancing) 처리하는 것

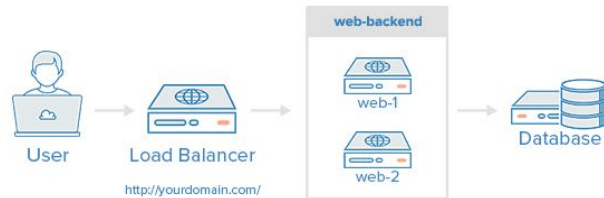


HAProxy

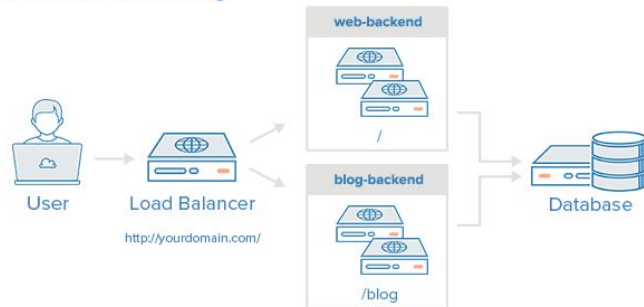
HAProxy란

L4, L7와 같은 Hardware LoadBalancer를 대체하기 위한 Open Source로 Reverse Proxy를 기반으로 한 L4, L7 Software LoadBalancer이다.

Layer 4 Load Balancing



Layer 7 Load Balancing

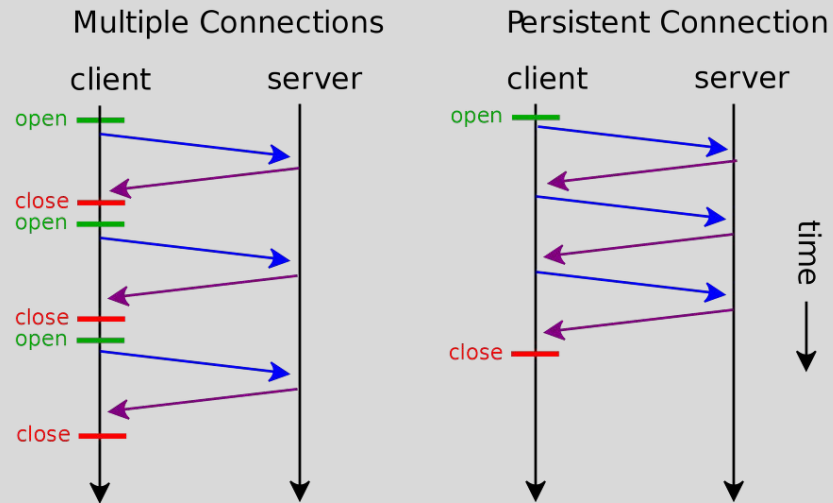


KeepAlive

KeepAlive란

여러 클라이언트의 접속을 원활히 하기
위해 사용하는 한번 맺은 세션을 요청이
끝나더라도 유지해주는 기능

TCP, HTTP KeepAlive가 있다



로드밸런싱 서버 구성

HAProxy : 2.5.5

Keepalived : 2.0.7

openssl : 1.0.2k-fips



HAProxy VM Primary :
192.168.1.106

HAProxy VM Secondary :
192.168.1.116

VIP : 192.168.1.200

```

log      127.0.0.1:514 local1
chroot   /
external-check
insecure-fork-wanted
stats socket /var/run/haproxy/stats.sock mode 660 group haproxy level admin expose-
stats timeout 30s
pidfile   /var/run/haproxy/haproxy.pid
ulimit-n  655350
maxconn   100000
user      haproxy
group     haproxy
daemon
nbthread  4

```

```

defaults
mode      http
log        global
option     tcplog
option     dontlognull
option     tcpka
timeout queue 1m
timeout connect 5s
timeout client 400m
timeout server 400m
timeout check 5s

```

```

listen stats
bind *:9001
mode http
option dontlog-normal
stats enable
stats realm Haproxy Statistics
stats uri /haproxy
http-request use-service prometheus-exporter if { path /metrics }

```

```

frontend http-front
mode http
bind *:80
bind *:443 ssl crt /etc/haproxy/certs/test.pem alpn h2,http/1.1
redirect scheme https code 301 if !{ ssl_fc }
default_backend http-backend

```

```

backend http-backend
balance roundrobin
server Server-01 192.168.1.110:80 check port 80
server Server-02 192.168.1.111:80 check port 80

```

로드밸런싱할 서버 목록 내부 web서버 1,2 80포트를 연결

"haproxy.cfg" 46L, 1194C

4-2. Load Balancing

HAProxy 세팅 (로드밸런싱)

HAProxy Primary, Secondary 세팅

/etc/haproxy/haproxy.cfg

listem stats 빼고 같음

로드밸런싱 알고리즘

Round Robin: 가장 기본적인 알고리즘입니다. 이 알고리즘은 서버의 가중치에 따라 서버들에게 차례로 요청을 넘겨줍니다. 서버의 처리 시간이 가장 균등하게 분배되기에 가장 부드럽고 공정한 알고리즘입니다.

source: 이 알고리즘은 서버가 다운되거나 중지되지 않는한, 동일한 클라이언트 IP는 항상 동일한 서버에게 연결됩니다.

등등 여러가지가 있습니다.

```
global_defs {
```

```
    router_id HAproxy      HAproxy = Primary  
                           HAproxyB = Secondary
```

```
# Define the script used to check if haproxy is still working
```

```
vrrp_script chk_haproxy {
```

```
    script "killall -0 haproxy"
```

```
    interval 2
```

```
    weight 2
```

```
}
```

```
Configuration for the virtual interface
```

```
vrrp_instance VIS_1 {
```

```
    interface      eth1
```

```
    state          MASTER
```

```
    priority       101      우선순위 높은순서  
                           Primary = 101 / Secondary = 100
```

```
    virtual_router_id 51
```

```
    advert_int     1
```

```
# The virtual ip address shared between the two loadbalancers
```

```
virtual_ipaddress {
```

```
    192.168.1.200
```

```
}
```

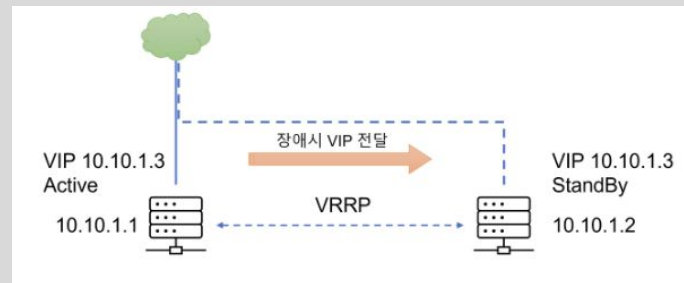
```
"/etc/keepalived/keepalived.conf" 59L, 743C
```

4-2. Load Balancing

Keepalived 세팅 (vip공유 서버세팅)

Keepalived Primary , Secondary 세팅

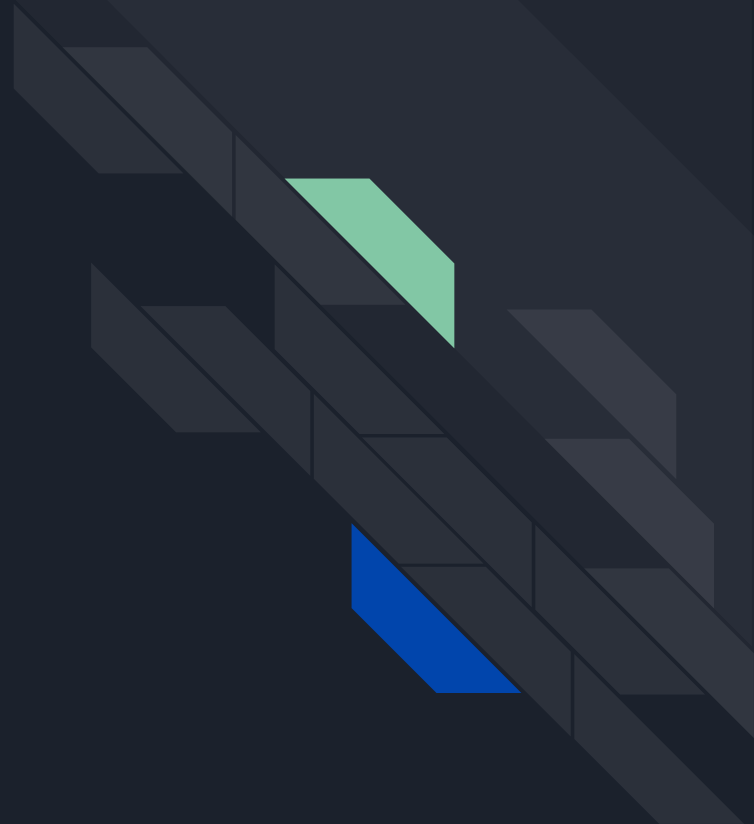
/etc/keepalived/keepalived.conf



4-3 Web

웹 서버 구축 및 이중화

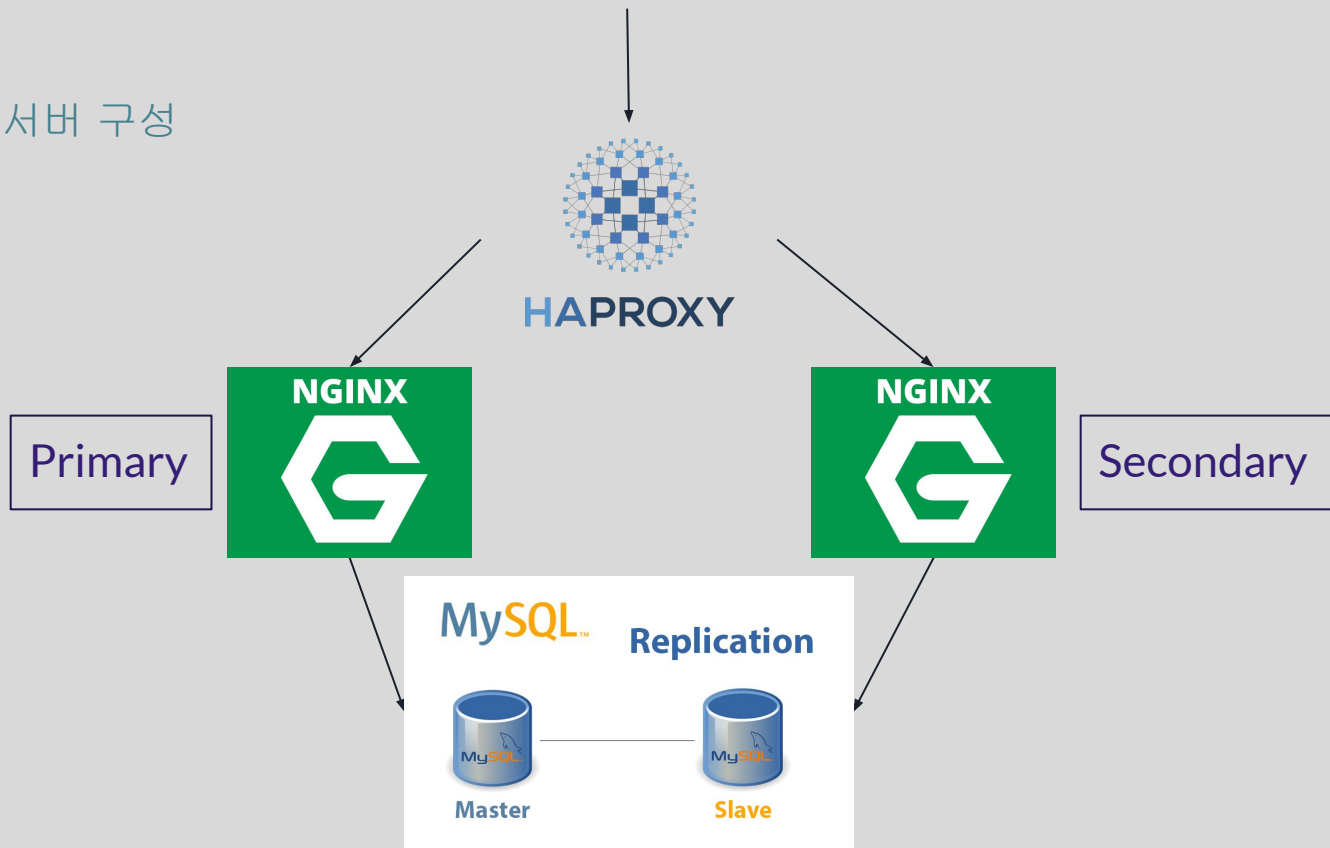
wordpress



웹서버 설계

1. LAMP -> LEMP
 - Apache -> Nginx
2. LEMP
 - Linux -> centos7 3.10.0
 - Nginx -> 1.221
 - MariaDB -> 10.8.6
 - PHP -> 8.1.14
3. WordPress
 - CMS(content management system)
 - 6.11 version
4. Address
 - 118.46.50.135(Primary)
 - 118.46.49.45(Secondary)

웹서버 구성



사전 준비

1. yum 업데이트
 - yum update
2. yum-utils 설치
 - yum install yum-utils
 - yum-config-manager를 사용하기 위해 설치
3. Net-tools 설치
 - yum install net-tools
 - 서버의 ip를 확인할 때 사용
4. Rdate 설치
 - yum install rdate
 - 서버의 시간을 동기화하기 위해 설치
5. 리눅스 시간 동기화
 - rdate -s time.bora.net
 - Lg u+에서 제공하는 ntp서버의 시간과 동기화

Nginx 설치

1. Repo 파일 작성으로 yum의 경로 지정
 - `vi /etc/yum.repos.d/nginx.repo`

```
[nginx-stable]
name=nginx stable repo
baseurl=http://nginx.org/packages/centos/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key
module_hotfixes=true
```

2. Nginx에 관련된 패키지들 설치
 - `yum -y install pcre-devel libxml2-devel zlib zlib-devel openssl openssl-devel gcc g++ cpp gcc-c++ libxslt libxslt-devel libgd-devgd gd-devel perl perl-ExtUtils-Embed geoip-devel`



Nginx 설치

3. Nginx 설치

- `yum -y install nginx`

4. Nginx 버전 확인

- `nginx -v`

5. Nginx 방화벽 해제

- `firewall-cmd --permanent --add-service=http`
- `firewall-cmd --permanent --add-service=https`
- `firewall-cmd --reload`
- 80번 포트(http)와 443번 포트(https) 개방

6. Nginx 서비스 시작과 자동 시작

- `systemctl start nginx`
- `systemctl enable nginx`

PHP 설치

1. EPEL(Extra Packages for Enterprise Linux)설치

- `yum -y install epel-release`

2. Remirepo 설치

- `yum -y install http://rpms.remirepo.net/enterprise/remi-release-7.rpm`

3. 현재 yum에 등록된 PHP 버전 확인

- `yum info php -> 현재 5.4`

4. yum에 등록된 php 버전 변경

- `yum-config-manager --disable remi-php54`
- `yum-config-manager --enable remi-php81`
- `yum info php -> 8.1.14 변경`

PHP 설치

5. PHP와 관련 패키지들 설치

- `yum install php php-cli php-fpm php-mysqlnd php-mysql php-mysqli php-devel php-zip php-gd php-curl php-xml php-json php-intl php-mbstring php-mcrypt php-posix php-shmop php-soap php-sysvmsg php-sysvsem php-sysvshm php-xmlrpc php-opcache`
- PHP 버전을 8.1로 등록해놔서 관련 패키지들도 8.1버전에 맞게 설치됨

6. php-fpm 시작과 자동 시작

- `systemctl start php-fpm`
- `systemctl enable php-fpm`

Nginx와 PHP 설정

1. php-fpm 설정 파일 수정

- `vi /etc/php-fpm.d/www.conf`
- `user = apache -> nginx`
- `group = apache -> nginx`
- `listen = 127.0.0.1:9000 -> /var/run/php-fpm/php-fpm.sock`
- `;listen.owner = nobody -> nginx 변경 후 주석(;) 제거`
- `;listen.group = nobody -> nginx 변경 후 주석(;) 제거`

2. 웹서버의 루트 디렉토리 생성

- `cp -r /usr/share/nginx/html/ /루트 디렉토리 위치와 이름`
- `chmod -R 755 /루트 디렉토리 위치와 이름`
- `chown -R nginx:nginx / 루트 디렉토리 위치와 이름`
- 여기서는 서버 이중화를 위해 공유된 디스크로 지정

Nginx와 PHP 설정

3. 웹서버의 루트 디렉토리 변경과 소켓 변경

- vi /etc/nginx/conf.d/default.conf

4. PHP에서 JIT 사용 설정

- vi /etc/php-zts.d/10-opcache.ini
- opcache.enable=1 밑에 추가
- opcache.jit_buffer_size=100M
- opcache.jit=tracing

```
server {
    listen      80 default;
    server_name _;
    add_header X-Frame-Options "SAMEORIGIN";
    charset utf-8;

    location / {
        root    /nfs-storage/localhost;
        index   index.html index.htm index.php;
    }
    #error_page 404              /404.html;
    error_page 500 502 503 504  /50x.html;
    location ~ /50x.html {
        root    /usr/share/nginx/html;
    }
    location ~ /\.php$ {
        fastcgi_pass    unix:/var/run/php-fpm/php-fpm.sock;
        fastcgi_index   index.php;
        fastcgi_param    SCRIPT_FILENAME    /nfs-storage/localhost$fastcgi_script_name;
        include          fastcgi_params;

        fastcgi_buffers 8 512k;
        fastcgi_buffer_size 256k;
        fastcgi_send_timeout 5m;
        fastcgi_read_timeout 5m;
        fastcgi_connect_timeout 5m;
    }
    location ~ /\.ht {
        deny all;
    }
}
```

Nginx와 PHP 설정

5. php-fpm와 Nginx 재시작

- systemctl restart php-fpm
- systemctl restart nginx

6. PHP 테스트

- echo "<?php phpinfo(); ?>" >
- /웹서버 루트 디렉토리/phpinfo.php
- 정상적으로 표시되는지 확인 후 삭제

PHP Version 8.1.14	
	
System	Linux server.ans 3.10.0-1160.42.2.el7.x86_64 #1 SMP Thu Sep 7 14:49:57 UTC 2021 x86_64
Build Date	Jan 4 2023 06:45:14
Build System	Red Hat Enterprise Linux Server release 7.9 (Maipo)
Build Provider	Ram's RPM repository <https://rpm.ramrepo.net/> #StandWithUkraine
Compiler	gcc (GCC) 10.2.1 20210130 (Red Hat 10.2.1-11)
Architecture	x86_64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/10-opcache.ini, /etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-fpm.ini, /etc/php.d/20-gd.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-intl.ini, /etc/php.d/20-mbstring.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-posix.ini, /etc/php.d/20-shmop.ini, /etc/php.d/20-simplexml.ini, /etc/php.d/20-soap.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sodium.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-system.ini, /etc/php.d/20-systemd.ini, /etc/php.d/20-sysvshm.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-xml.ini, /etc/php.d/20-xmlreader.ini, /etc/php.d/20-xmlwriter.ini, /etc/php.d/20-zip.ini, /etc/php.d/20-zlib.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini, /etc/php.d/30-readline.ini, /etc/php.d/30-xmlrpc.ini, /etc/php.d/30-zip.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API20210902/NTS
PHP Extension Build	API20210902/NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
OTrace Support	available, disabled
Registered PHP Streams	ftp, ftps, compress.zlib, php, file, glob, data, http, https, compress.bzip2, phar.zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv2, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, bzip2.*, convert.iconv.*, mcrypt.*, mdecrypt.*

Wordpress 설치

1. 설치 파일 다운로드

- `wget http://wordpress.org/latest.tar.gz`

2. 웹서버 루트 디렉토리에 압축 해제 후 권한,소유자 설정

- `tar xvfz latest.tar.gz -C /웹서버 루트 디렉토리`
- `chown -R nginx:nginx /웹서버 루트 디렉토리`
- `chmod -R 755 /웹서버 루트 디렉토리`

3. Wordpress 설치

- 웹 브라우저에서 서버주소/wordpress로 이동



The image shows the WordPress installation configuration screen. At the top is the WordPress logo. Below it is a text box with the instruction: "아래에서 당신의 데이터베이스 연결 상세를 입력해야 합니다. 이것을 잘 모른다면 당신의 호스트에 연락하세요." (Enter your database connection details below. If you don't know this, contact your host.)

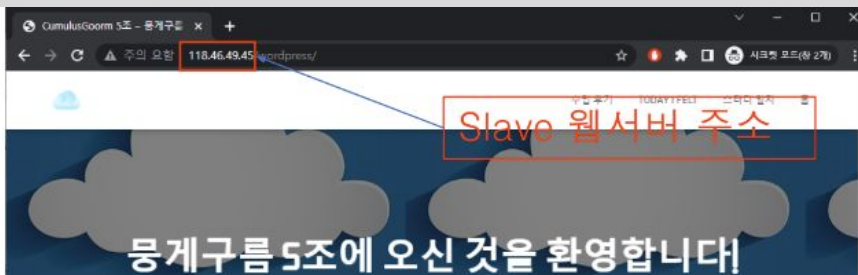
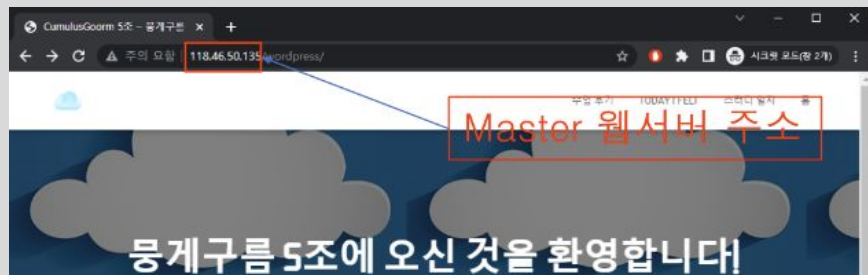
The form contains the following fields and labels:

- 데이터베이스 이름** (Database Name): Input field containing "db서버의 wp용 db이름". To the right, it says "워드프레스에 사용할 데이터베이스 이름." (Database name to use for WordPress.)
- 사용자명** (Username): Input field containing "사용자명". To the right, it says "데이터베이스 사용자명." (Database username.)
- 비밀번호** (Password): Input field containing "비밀번호". To the right, it says "데이터베이스 비밀번호." (Database password.)
- 데이터베이스 호스트** (Database Host): Input field containing "db서버의 ipv4주소". To the right, it says "localhost이(가) 작동하지 않는다면 이 정보는 자신의 웹호스팅에서 받을 수 있습니다." (If localhost doesn't work, this information can be obtained from your own web hosting.)
- 테이블 접두어** (Table Prefix): Input field containing "wp_". To the right, it says "하나의 데이터베이스에서 여러 개의 워드프레스를 설치하여 운영하려면 이것을 변경하세요." (To operate multiple WordPress installations on one database, change this.)

At the bottom left is a button labeled "계속" (Continue).

웹서버 이중화

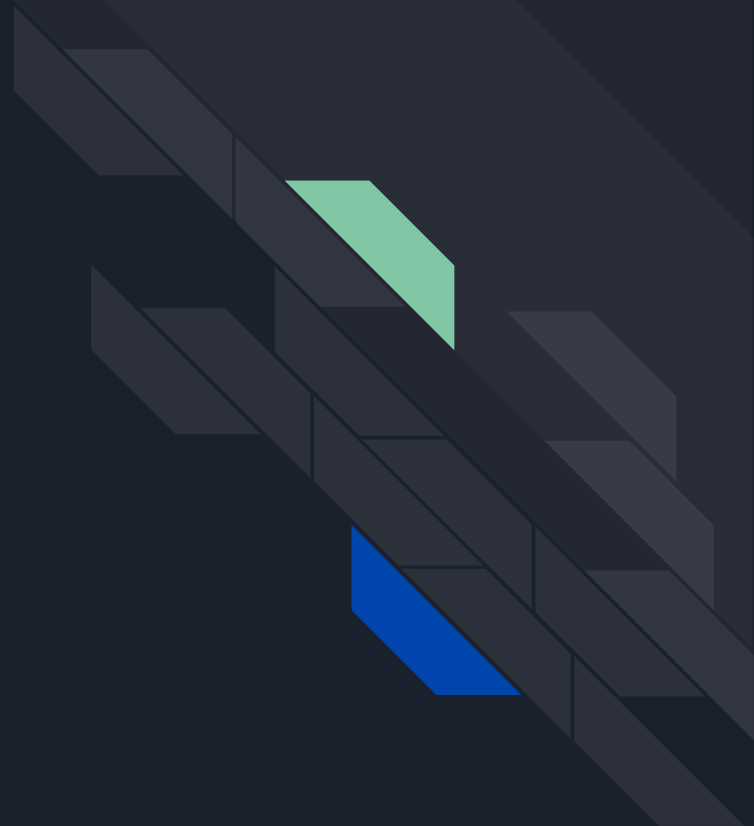
- Wordpress는 공유 스토리지에 설치했기 때문에 master에서만 설치해도 slave에서도 운용 가능
- 다만, slave 서버에서 페이지 이동시 주소가 master의 주소로 변경되는 문제가 존재
- Wordpress 설정 파일을 수정해서 해당 문제를 해결
 - `vi /nfs-storage/localhost/wordpress/wp-config.php` 파일에 아래 두 줄을 추가
 - `define('WP_HOME','http://'.$_SERVER['HTTP_HOST'].'/wordpress');`
 - `define('WP_SITEURL','http://'.$_SERVER['HTTP_HOST'].'/wordpress');`



4-4 DB

DB(DataBase) 구축 및 이중화

MySQL, MariaDB



[DB 구축 진행 순서]

MariaDB, MySQL 설치 → 방화벽 설정 → 이중화

```
[root@localhost ~]# vi /etc/yum.repos.d/MariaDB.repo

# MariaDB 10.8 CentOS repository list - created 2023-01-16 07:20 UTC
# https://mariadb.org/download/
[mariadb]
name = MariaDB
baseurl = https://tw1.mirror.blendbyte.net/mariadb/yum/10.8/centos7-amd64
gpgkey=https://tw1.mirror.blendbyte.net/mariadb/yum/RPM-GPG-KEY-MariaDB
gpgcheck=1

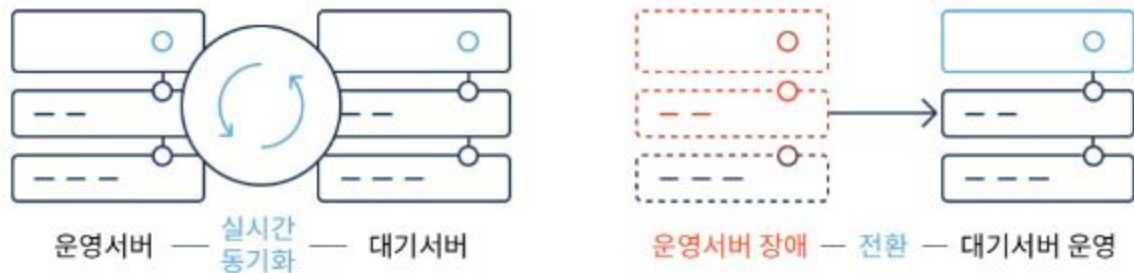
[root@localhost ~]# systemctl start mysqld
[root@localhost ~]# systemctl enable mysqld
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to /usr/l

[root@localhost ~]# systemctl start mariadb
[root@localhost ~]# systemctl enable mariadb

[root@localhost ~]# firewall-cmd --add-service=mysql --permanent
success
[root@localhost ~]# firewall-cmd --reload
success
```

DB 서버 이중화 [Replication]

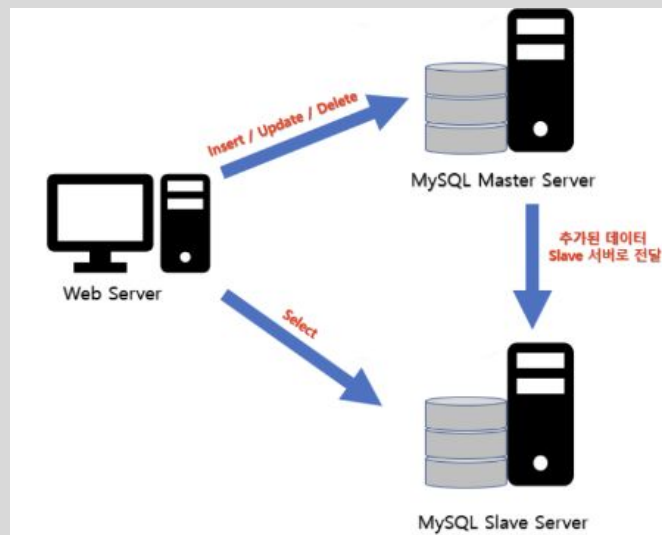
DB Primary서버와 Secondary 서버로 이중화



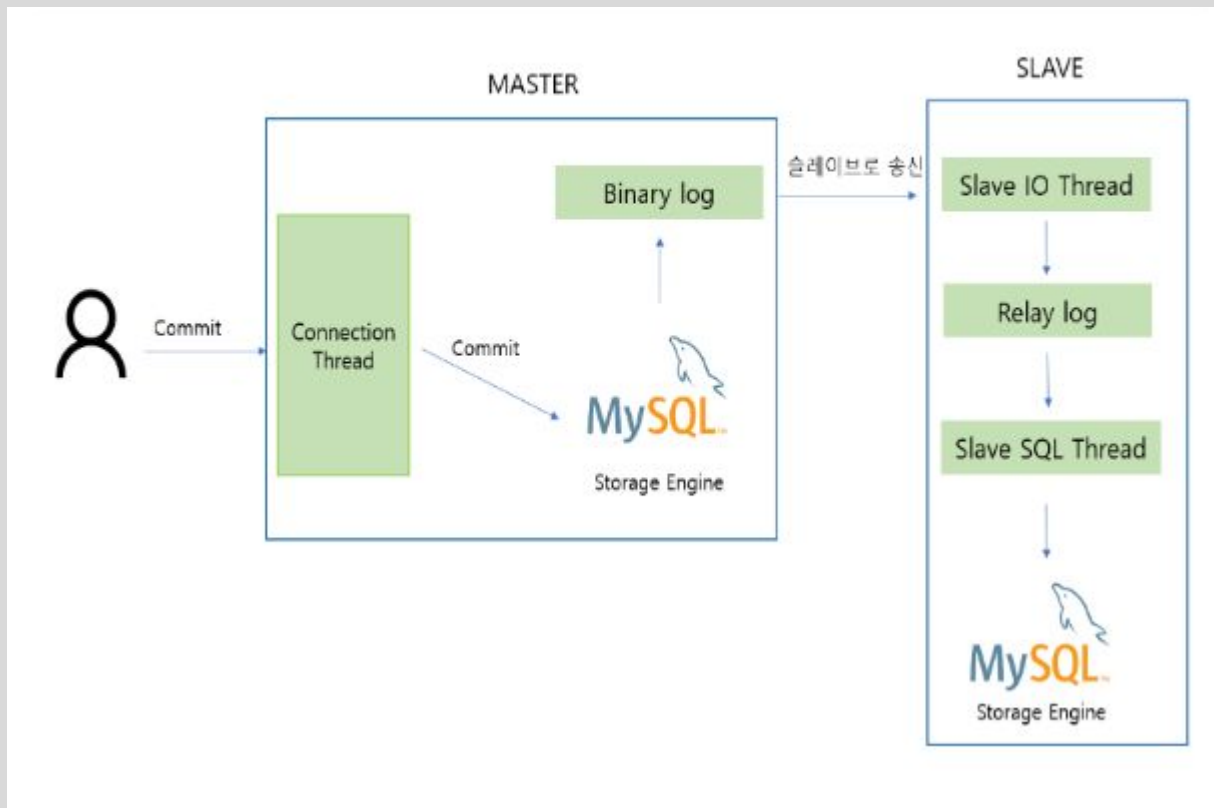
- Secondary 서버는 Primary서버의 로그기록을 읽고 데이터를 업데이트 하는 방식

- Primary 서버는 데이터 베이스 복사를 하는 메인 DB

- Secondary 서버는 읽기만 가능하고 쓰기는 Primary 서버에서만 수행



Replication 동작 원리



[MySQL Replication 설정]

- MySQL Replication을 사용하기 위해 먼저 DB, 계정, 리플리케이션 계정을 생성해야한다.

- Primary Server

IP : 118.46.49.13

DataBase : repl_db

ID : test1

PW : test123

- Replication 계정

IP : 118.46.49.13

ID : repl_user

PW : test456

[MySQL Replication 설정] - Primary Server

1. DB생성

```
mysql> create database repl_db default character set utf8;
```

2. 계정 생성 및 권한 부여

```
mysql> create user test1@'%' identified by 'test123';
```

```
mysql> grant all privileges on repl_db.* to test1@'%' identified by 'test123';
```

3. Replication 계정 생성

```
mysql> create user test1@'%' identified by 'test123';
```

```
mysql> grant all privileges on repl_db.* to repl_user@'%' identified by 'test456';
```

[MySQL Replication 설정] - Primary Server

4. SQL 재시작

`systemctl restart mysqld`

5. Primary Server 정보 확인

```
MariaDB [mysql]> show master status;    ## 파일명과 포지션 위치 기억하기
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB
mysql-bin.000001	328		

```
1 row in set (0.014 sec)
```

File : MySQL 로그파일
Position : 로그 파일내 읽을 위치
Binlog_Do_DB : 바이너리 로그파일 = 변경된 이벤트가 쌓이는 파일
Binlog_Ignore_DB : 복제 제외 정보

[MySQL Replication 설정] - Secondary Server

1. Primary Server와 같이 계정, DB 생성

```
mysql> create database repl_db default character set utf8;
```

2. 계정 생성 및 권한 부여

```
mysql> create user test1@'%' identified by 'test123';
```

```
mysql> grant all privileges on repl_db.* to test1@'%' identified by 'test123';
```

3. Secondary Server 설정

1) mysql에 접속하여 설정 [택]

2) 설정파일(my.cnf)에 직접 설정

[MySQL Replication 설정] - Secondary Server

4. 파일 수정

```
MariaDB [(none)]> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [mysql]> change master to
  -> master_host='118.46.49.13',
  -> master_user='repl_user',
  -> master_password='test456',
  -> master_log_file='mysql-bin.000001',
  -> master_log_pos=328;
Query OK, 0 rows affected (0.074 sec)
```

Primary Server 정보와 ip를 넣어준다.

Master_host : master 서버 ip

Master_user : replication id

Master_passwd : replication pw

Master_log_file : master status log

Master_log_pos : position 값

[Primary, Secondary Server 상태 확인]

Primary Server에서 Relication 상태 확인

```
MariaDB [mysql]> show master status\G
***** 1. row *****
      File: mysql-bin.000002
      Position: 2300594
      Binlog_Do_DB:
      Binlog_Ignore_DB:
1 row in set (0.000 sec)
```

```
MariaDB [mysql]> show processlist\G
***** 2. row *****
      Id: 50
      User: repl_user
      Host: 118.46.49.40:57768          ##### slave 서버에 연결 확인
      db: NULL
      Command: Binlog Dump
      Time: 51
      State: Master has sent all binlog to slave; waiting for more updates
      Info: NULL
      Progress: 0.000
```

[Primary, Secondary Server 상태 확인]

Secondary Server 상태 확인

1번, 2번, 3번 확인 되면 정상적 이중화가 완료된 것.

```
MariaDB [mysql]> show slave status\G
***** 1. row *****
Slave_IO_State: Waiting for master to send event          ##### 1번
Master_Host: 118.46.49.13                                ##### 마스터 서버 연결확인
Master_User: repl_user                                    ##### 마스터 사용자 확인
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000002
Read_Master_Log_Pos: 2300594
Relay_Log_File: localhost-relay-bin.000004
Relay_Log_Pos: 1325607
Relay_Master_Log_File: mysql-bin.000002
Slave_IO_Running: Yes                                     ##### 2번
Slave_SQL_Running: Yes                                    ##### 3번
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
```


[MySQL Replication 작동 확인]

Primary Server에서 테이블 만든 후, Secondary 에서 확인

```
MariaDB [(none)]> use repl_db
Database changed
MariaDB [repl_db]> CREATE TABLE TEST (
  -> num int(10) not null,
  -> name varchar(10) not null);
Query OK, 0 rows affected (0.022 sec)
```

테이블 만들어보기

```
MariaDB [repl_db]> show tables;
+-----+
| Tables_in_repl_db |
+-----+
| TEST               |
+-----+
1 row in set (0.000 sec)
```

```
MariaDB [(none)]> use repl_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
MariaDB [repl_db]> show tables;
```

테이블이 만들어진 것을 확인가능

```
+-----+
| Tables_in_repl_db |
+-----+
| TEST               |
+-----+
1 row in set (0.000 sec)
```

4-5 ETC

NFS 공유 Storage 및 백업
FTP
포트포워딩



NFS 공유 스토리지

rpm -qa nfs-utils

vi /etc/exports

```
/root/nfs-storage *(rw,sync)
```

rw : read, write 권한 설정

sync : 실시간 동기화 설정

공유 NFS 스토리지 확인

showmount -e

```
root@localhost localhostl# showmount -e
Export list for localhost.localdomain:
root/nfs-storage *
root@localhost localhostl#
```

exportfs -v

```
root@localhost localhostl# exportfs -v
root/nfs-storage
<world>(sync,wdelay,hide,no_
root@localhost localhostl#
```

NFS 공유 스토리지 백업 저장소

auto_backup_shell.sh 생성

```
#!/bin/bash

/bin/mount -t nfs 192.168.1.105:/root/nfs-storage/ /nfs-storage/

rsync -av --delete /nfs-storage/ /nfs-backup/

/bin/umount /nfs-storage/
```

crontab 등록 (10분마다 쉘 실행)

```
*/10 * * * * /root/auto_backup_shell
```

192.168.1.105의
/root/nfs-storage/를

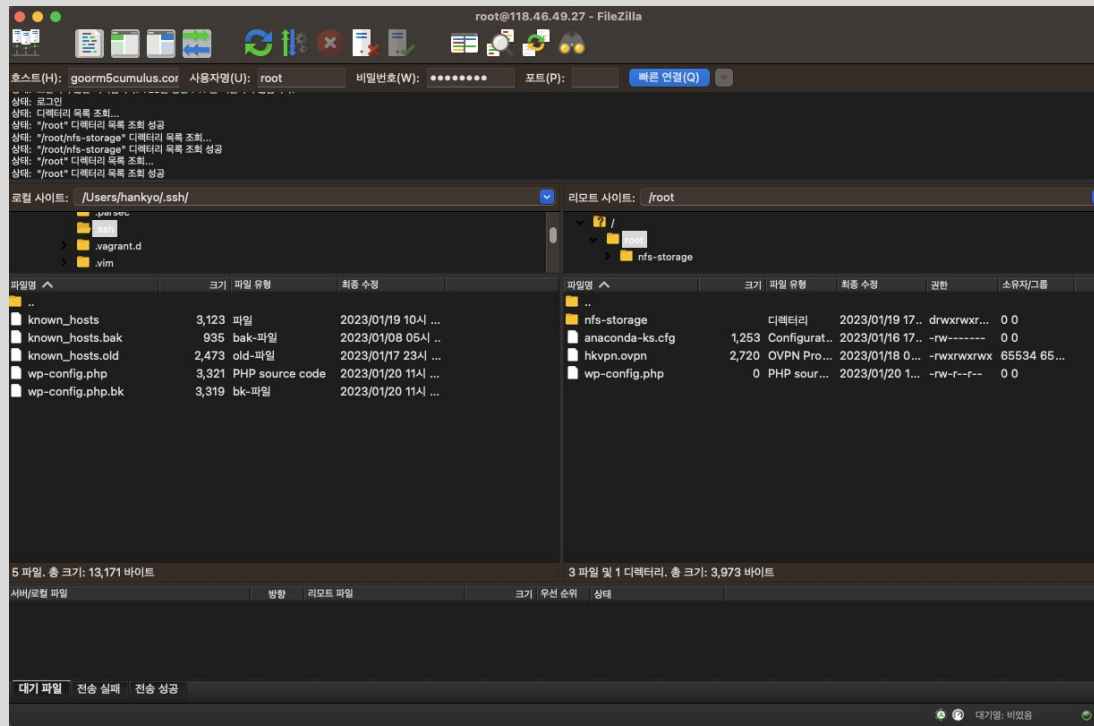
/nfs-storage에 마운트 후

rsync(동기화시켜주는 패키지) 사용

/nfs-backup 에 동기화 후

마운트 해제

FTP



FileZilla를 사용한 FTP

NFS 공유 서버에
FTP서버를 같이 올림

OPNsense 포트포워딩 설정

OPNsense (c) 2014-2022 Deciso B.V.

root@OPNsense.localdomain

Firewall: NAT: Port Forward

Select category

	Source		Destination		NAT		
	Interface	Proto	Address	Ports	Address	Ports	Description
<input type="checkbox"/>	LAN	TCP	*	*	LAN address	443	Anti-Lockout Rule
<input type="checkbox"/>	WAN	TCP	*	*	WAN address	80 (HTTP)	192.168.1.200 : VIP (80,443 포트)
<input type="checkbox"/>	WAN	TCP	*	*	WAN address	443 (HTTPS)	& VIP HAProxy(9001,9002 포트)
<input type="checkbox"/>	WAN	TCP	*	*	WAN address	9000	192.168.1.105 : NFS서버(FTP 포트)
<input type="checkbox"/>	WAN	TCP	*	*	WAN address	21 (FTP)	
<input type="checkbox"/>	WAN	TCP	*	*	WAN address	9001	
<input type="checkbox"/>	WAN	TCP	*	*	WAN address	9002	
<input checked="" type="checkbox"/>	Enabled rule		No redirect		Linked rule		
<input type="checkbox"/>	Disabled rule		Disabled no redirect		Disabled linked rule		

Alias (click to view/edit)

5. 결론 및 후기



결론 및 후기

많은 우여곡절속에서 **몽게몽게구름 프로젝트**는 어느정도 성공적으로 마무리 되었다고 생각된다.

아쉬운 점도 많지만 시간대비 많은것을 최선을 다했다고 생각하고 프로젝트중에 목표했던 서비스의 고가용성을위해 공부하면서 팀원들과 같이 많은 성장을 했다는 점이 가장 큰 이익이라고 생각된다.

그래도 아쉬운점과 개선할 점을 꼽아보자면

- 데이터 저장장치가 물리적인 데미지가 스토리지에 받는다면 서비스에 막대한 피해를 입는다.

- 해결방안은 **RAID** 구성으로 해결할 수 있을것 같다.

- 위와 마찬가지로 **vm**은 여러개여도 물리적인 온프레미스 서버의 대수는 한대이므로 서버 컴퓨터의 하드웨어적인 문제가 생길경우 서비스를 지속하기 어렵게 된다.

- 해결방안은 온프레미스의 물리적인 서버의 대수를 늘려서 **2중** 이상의 중첩된 서비스를 한다.
이때 물리적인 서버들의 위치는 인터넷의 문제와 정전의 문제가 있을경우를 대비해 다른 지역에 배치하는것이 좋을것이라 생각한다.



Thank you!

QNA

웅게웅게구름
구름 쿠버네티스 11회차 웅게구름조
세미프로젝트

goorm