WEB01 서버

```
# yum -y update
# vi /etc/selinux/config
7 SELINUX=disabled
# yum -y install php php-mysql php-mbstring php-pdo php-gd
# yum -y install httpd-*
# rpm -qa | grep httpd
httpd-manual-2.4.6-97.el7.centos.2.noarch
httpd-devel-2.4.6-97.el7.centos.2.x86_64
httpd-2.4.6-97.el7.centos.2.x86_64
httpd-tools-2.4.6-97.el7.centos.2.x86_64
# systemctl start httpd
# systemctl enable httpd
# firewall-cmd --permanent --add-service=https
# firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address=192.168.1.134 port port="80" protocol="tcp" accept'
# firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address=192.168.1.135 port port="80" protocol="tcp" accept'
# firewall-cmd --reload
# vi /etc/httpd/conf/httpd.conf
66 User nobody
67 Group nobody
119 DocumentRoot "/var/www/html"
120
121 #
122 # Relax access to content within /var/www.
123 #
124 <Directory "/var/www/html">
125     Options None
126     AllowOverride None
127     Require all granted
128 </Directory>
135     DirectoryIndex index.html index.html.var index.php index.php3
167 SetEnvIf Request_Method HEAD Health-Check (원래 있던 167을 168로 내리고)
168 LogFormat "%{x-forwarded-for}i %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
189 CustomLog "logs/access_log" combined env=!Health-Check
257     AddType application/x-httpd-php .php .html .htm .inc
258     AddType application/x-httpd-php-source .phps
# systemctl restart httpd
```

```
# yum -y install git
# git clone https://github.com/KGconcert/WEB.git
# tar -xvf /root/WEB/web.tar -C /var/www/html/
# ls -l /var/www/html/
# php -v
# yum -y install cockpit
# systemctl enable --now cockpit.socket
# firewall-cmd --permanent --add-service=cockpit
# firewall-cmd --reload
# yum -y install wget
# wget https://downloads.cisofy.com/lynis/lynis-2.7.5.tar.gz
# tar -zxvf lynis-2.7.5.tar.gz
```

HAproxy01 서버

[1. HAproxy Install]

```
# yum -y update
# vi /etc/selinux/config
7 SELINUX=disabled
# yum -y install gcc openssl openssl-devel systemd-devel
# yum -y install wget
# mkdir /HAproxy
# cd /HAproxy
# wget http://www.haproxy.org/download/2.3/src/haproxy-2.3.10.tar.gz
# tar xvfz haproxy-2.3.10.tar.gz
# cd haproxy-2.3.10/
# make TARGET=linux-glibc USE_OPENSSL=1 USE_SYSTEMD=1
# make install
# curl
"https://git.haproxy.org/?p=haproxy-2.3.git;a=blob_plain;f=contrib/systemd/haproxy.s
ervice.in;" -o /etc/systemd/system/haproxy.service
# ls -l /etc/systemd/system/haproxy.service
# vi /etc/systemd/system/haproxy.service
 10 ExecStartPre=/usr/local/sbin/haproxy -Ws -f $CONFIG -c -q $EXTRAOPTS
 11 ExecStart=/usr/local/sbin/haproxy -Ws -f $CONFIG -p $PIDFILE $EXTRAOPTS
 12 ExecReload=/usr/local/sbin/haproxy -Ws -f $CONFIG -c -q $EXTRAOPTS
# mkdir /etc/haproxy
# mkdir /etc/haproxy/certs
# mkdir /etc/haproxy/errors
# mkdir /var/log/haproxy
# cd ./examples/errorfiles/
```

```
# cp ./*.http /etc/haproxy/errors/
# ls -l /etc/haproxy/errors/
# cd ~
# useradd -c "HAproxy Daemon User" -s /sbin/nologin haproxy
# tail -1 /etc/passwd
# vi /etc/rsyslog.d/haproxy.conf
   $ModLoad imudp
   $UDPServerAddress 127.0.0.1
   $UDPServerRun 514
   local0.* /var/log/haproxy/haproxy-traffic.log
# firewall-cmd --permanent --add-port=514/udp
# firewall-cmd --reload
# vi /etc/logrotate.d/haproxy
   /var/log/haproxy/*.log {
        daily
        rotate 30
        create 0600 root root
        compress
        notifempty
        missingok
        sharedscripts
        postrotate
        /bin/systemctl restart rsyslog.service > /dev/null 2>/dev/null || true
        endscript
   }
```

[2. "/etc/haproxy/haproxy.cfg" Main 설정파일 편집]

```
# vi /etc/haproxy/haproxy.cfg
global
        daemon
        maxconn 4000
        user haproxy
        group haproxy
        log 127.0.0.1:514 local0

defaults
        mode http
        option redispatch
        retries 3
        log global
```

```
        option httplog
        option dontlognull
        option dontlog-normal
        option http-server-close
        option forwardfor
        maxconn 2000
        timeout connect 10s
        timeout http-request 10s
        timeout http-keep-alive 10s
        timeout client 1m
        timeout server 1m
        timeout queue 1m
        errorfile 400 /etc/haproxy/errors/400.http
        errorfile 403 /etc/haproxy/errors/403.http
        errorfile 408 /etc/haproxy/errors/408.http
        errorfile 500 /etc/haproxy/errors/500.http
        errorfile 502 /etc/haproxy/errors/502.http
        errorfile 503 /etc/haproxy/errors/503.http
        errorfile 504 /etc/haproxy/errors/504.http

listen stats
        bind *:9000
        stats enable
        stats realm Haproxy Stats Page
        stats uri /
        stats auth admin:haproxy1

frontend proxy
        bind *:80
        default_backend WEB_SRV_list

backend WEB_SRV_list
        balance roundrobin
        option httpchk HEAD /
        http-request set-header X-Forwarded-Port %[dst_port]
        cookie SRVID insert indirect nocache maxlife 10m
        server WEB_01 192.168.1.128:80 maxconn 1000 cookie WEB_01 check inter
3000 fall 5 rise 3
        server WEB_02 192.168.1.129:80 maxconn 1000 cookie WEB_02 check inter
3000 fall 5 rise 3
        server WEB_03 192.168.1.130:80 maxconn 1000 cookie WEB_02 check inter
```

3000 fall 5 rise 3

```
# haproxy -f /etc/haproxy/haproxy.cfg -c
# systemctl start haproxy
# systemctl stop haproxy
# systemctl enable haproxy
# firewall-cmd --permanent --add-service=http
# firewall-cmd --permanent --add-port=9000/tcp
# firewall-cmd --reload
```

[3. SSL/TLS 적용]

```
# rpm -qa | grep openssl
# openssl genrsa -out /etc/haproxy/certs/ha01.key 2048
# openssl req -new -key /etc/haproxy/certs/ha01.key -out
/etc/haproxy/certs/ha01.csr
    Country Name (2 letter code) [XX]:KR
    State or Province Name (full name) []:Seoul
    Locality Name (eg, city) [Default City]:Gangnam
    Organization Name (eg, company) [Default Company Ltd]:KGITBANK
    Organizational Unit Name (eg, section) []:CloudTeam
    Common Name (eg, your name or your server's hostname) []:www.kgmusic.com
    Email Address []:kgmusic.com
    A challenge password []: (Enter)
    An optional company name []: (Enter)
# openssl x509 -req -days 365 -in /etc/haproxy/certs/ha01.csr -signkey
/etc/haproxy/certs/ha01.key -out /etc/haproxy/certs/ha01.crt
# cd /etc/haproxy/certs
# cat ha01.crt ha01.key > ha01_ssl.crt
# cd ~
# vi /etc/haproxy/haproxy.cfg
 [ Global 영역에 추가작성 ]
 ssl-default-bind-ciphers
ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECD
SA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128
-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:EC
DHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA
256
 ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets
```

```
 [ Frontend 영역에 추가작성 ]
 bind *:443 ssl crt /etc/haproxy/certs/ha01_ssl.crt
 http-request redirect scheme https code 308 unless { ssl_fc }

# haproxy -f /etc/haproxy/haproxy.cfg -c
# firewall-cmd --permanent --add-service=https
# firewall-cmd --reload
# systemctl restart haproxy
```

[4. HAproxy 고가용성 구축]

```
# echo net.ipv4.ip_nonlocal_bind=1 >> /etc/sysctl.conf
# sysctl -p
    net.ipv4.ip_nonlocal_bind = 1
# yum -y install keepalived-*
# vi /etc/keepalived/keepalived.conf
 ※ 내용 싹 다 지우고 아래꺼 복사
global_defs {
   router_id HA_01
}

vrrp_script HA_Check {
        script "killall -0 haproxy"
        interval 1
        rise 3
        fall 3
        weight 2
}

vrrp_instance HAGroup_1 {
    state MASTER
    interface ens32
    garp_master_delay 5
    virtual_router_id 51
    priority 110
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass test123
    }
    virtual_ipaddress {
```

```
        192.168.1.150
    }
    track_script {
        HA_Check
    }
}
```

```
# firewall-cmd --direct --add-rule ipv4 filter INPUT 1 -i ens32 -d 224.0.0.18 -p vrrp -j ACCEPT
# firewall-cmd --direct --add-rule ipv4 filter OUTPUT 1 -o ens32 -d 224.0.0.18 -p vrrp -j ACCEPT
# firewall-cmd --runtime-to-permanent
# firewall-cmd --direct --get-all-rules
# systemctl start keepalived
# systemctl enable keepalived
# yum -y install cockpit
# systemctl enable --now cockpit.socket
# firewall-cmd --permanent --add-service=cockpit
# firewall-cmd --reload
# wget https://downloads.cisofy.com/lynis/lynis-2.7.5.tar.gz
# tar -zxvf lynis-2.7.5.tar.gz
```

DB01 서버

[DB01 기본설치]

```
# yum -y update
# vi /etc/selinux/config
7 SELINUX=disabled
# yum -y install mariadb-*
# vi /etc/my.cnf
※[mysqld] 밑에 새로 추가
2 character-set-server=utf8
# systemctl start mariadb
# systemctl enable mariadb
# mysql_secure_installation
Enter current password for root (enter for none): 엔터 누르기
Set root password? [Y/n] y
New password:itbank
Re-enter new password:itbank
Remove anonymous users? [Y/n] y
```

Disallow root login remotely? [Y/n] y
Remove test database and access to it? [Y/n] y
Reload privilege tables now? [Y/n] y
firewall-cmd --permanent --add-service=mysql
firewall-cmd --reload
yum -y install cockpit
systemctl enable --now cockpit.socket
firewall-cmd --permanent --add-service=cockpit
firewall-cmd --reload
yum -y install wget
wget https://downloads.cisofy.com/lynis/lynis-2.7.5.tar.gz
tar -zxvf lynis-2.7.5.tar.gz

Ansible

[메인서버]

yum -y update
vi /etc/selinux/config
7 SELINUX=disabled
yum -y install centos-release-ansible-29
yum -y install ansible
vi /etc/ansible/hosts
※내용 다 지운 후 진행
[WEB]
WEB2 ansible_host=192.168.1.129
WEB3 ansible_host=192.168.1.130
[DB]
DB2 ansible_host=192.168.1.133

[Proxy]
Proxy2 ansible_host=192.168.1.135

[WEB:vars]
ansible_connection=ssh
ansible_user=root

[DB:vars]
ansible_connection=ssh
ansible_user=root

[Proxy:vars]

ansible_connection=ssh
ansible_user=root

yum -y install cockpit
yum -y install cockpit-dashboard
systemctl enable --now cockpit.socket
firewall-cmd --permanent --add-service=cockpit
firewall-cmd --permanent --add-service=http
firewall-cmd --reload
wget https://downloads.cisofy.com/lynis/lynis-2.7.5.tar.gz
tar -zxvf lynis-2.7.5.tar.gz

[관리대상 서버들 초기 환경 구성]

※ WEB02, WEB03, Proxy02, DB02 서버들 에서 아래 내용 똑같이 진행

WEB02

yum -y update
mkdir /root/.ssh
vi /etc/selinux/config
7 SELINUX=disabled
ssh-keygen -t rsa
Enter file in which to save the key (/root/.ssh/id_rsa):"enter"
Enter passphrase (empty for no passphrase):"enter"
Enter same passphrase again:"enter"
chmod 600 ~/.ssh/*
scp ~/.ssh/id_rsa.pub root@192.168.1.170:~/.ssh/WEB02_authorized_keys

WEB03

yum -y update
mkdir /root/.ssh
vi /etc/selinux/config
7 SELINUX=disabled
ssh-keygen -t rsa
Enter file in which to save the key (/root/.ssh/id_rsa):"enter"
Enter passphrase (empty for no passphrase):"enter"
Enter same passphrase again:"enter"
chmod 600 ~/.ssh/*
scp ~/.ssh/id_rsa.pub root@192.168.1.170:~/.ssh/WEB03_authorized_keys

DB02

```
# yum -y update
# mkdir /root/.ssh
# vi /etc/selinux/config
7 SELINUX=disabled
# ssh-keygen -t rsa
Enter file in which to save the key (/root/.ssh/id_rsa):"enter"
Enter passphrase (empty for no passphrase):"enter"
Enter same passphrase again:"enter"
# chmod 600 ~/.ssh/*
# scp ~/.ssh/id_rsa.pub root@192.168.1.170:~/.ssh/DB02_authorized_keys
```

Proxy02

```
# yum -y update
# mkdir /root/.ssh
# vi /etc/selinux/config
7 SELINUX=disabled
# ssh-keygen -t rsa
Enter file in which to save the key (/root/.ssh/id_rsa):"enter"
Enter passphrase (empty for no passphrase):"enter"
Enter same passphrase again:"enter"
# chmod 600 ~/.ssh/*
# scp ~/.ssh/id_rsa.pub root@192.168.1.170:~/.ssh/Proxy02_authorized_keys
```

[02 서버들 초기 환경 구성 후 Ansible서버에서 진행]

```
# yum -y install git
# git clone https://github.com/KGconcert/Ansible.git
# vi /etc/ansible/ansible.cfg
70 # uncomment this to disable SSH key host checking
71 host_key_checking = False

# ssh-keygen -t rsa
Enter file in which to save the key (/root/.ssh/id_rsa):"enter"
Enter passphrase (empty for no passphrase):"enter"
Enter same passphrase again:"enter"
# chmod 600 ~/.ssh/*
# scp ~/.ssh/id_rsa.pub root@192.168.1.129:~/.ssh/authorized_keys
Are you sure you want to continue connecting (yes/no)? yes
# scp ~/.ssh/id_rsa.pub root@192.168.1.130:~/.ssh/authorized_keys
```

Are you sure you want to continue connecting (yes/no)? yes
scp ~/.ssh/id_rsa.pub root@192.168.1.133:~/.ssh/authorized_keys
Are you sure you want to continue connecting (yes/no)? yes
scp ~/.ssh/id_rsa.pub root@192.168.1.135:~/.ssh/authorized_keys
Are you sure you want to continue connecting (yes/no)? yes
ansible all -m ping -k
tar -xvf /root/Ansible/backup.tar -C /root/
tar -xvf /root/Ansible/yml.tar -C /root/
rm -rf /root/backup.tar
rm -rf /root/yml.tar
ansible-galaxy collection install ansible.posix
ansible-playbook /root/yml/web.yml
ansible-playbook /root/yml/proxy.yml
ansible-playbook /root/yml/db.yml

DB01 서버

[DB01 Replication 작업]

mysql -u root -pitbank mysql
MariaDB [mysql]> grant replication slave on *.* to Rep_user@'%' identified by 'itbank';
MariaDB [mysql]> flush privileges;
MariaDB [mysql]> quit
vi /etc/my.cnf
log-bin=mysql-bin
server-id=1
replicate-do-db='webdb'
mysql -u root -pitbank mysql
MariaDB [mysql]> change master to master_host='192.168.1.133',
-> master_user='Rep_user',
-> master_password='itbank',
-> master_log_file='mysql-bin.000001',
-> master_log_pos=245;
MariaDB [mysql]> quit
systemctl restart mariadb

DB02 서버

[DB02 Replication 작업]

※DB01 Replication 작업 끝난 후 진행

```
# vi /etc/my.cnf
server-id=2
replicate-do-db='webdb'
log-bin=mysql-bin
# mysql -u root -pitbank mysql
MariaDB [mysql]> change master to master_host='192.168.1.132',
-> master_user='Rep_user',
-> master_password='itbank',
-> master_log_file='mysql-bin.000001',
-> master_log_pos=245;
MariaDB [mysql]> grant replication slave on *.* to Rep_user@'%' identified by 'itbank';
MariaDB [mysql]> flush privileges;
MariaDB [mysql]> quit
# systemctl restart mariadb
# mysql -u root -pitbank mysql
MariaDB [mysql]> create user itbank@'192.168.1.%' identified by 'itbank';
MariaDB [mysql]> create user 'itbank'@'localhost' identified by 'itbank';
MariaDB [mysql]> grant all privileges on webdb.* to itbank@'192.168.1.%' identified by 'itbank';
MariaDB [mysql]> grant all privileges on webdb.* to 'itbank'@'localhost';
MariaDB [mysql]> flush privileges;
MariaDB [mysql]> quit
# systemctl restart mariadb
```

DB01 서버

[전체 Replication 작업 끝낸 후 webdb 데이터베이스 적용]

```
# yum -y install git
# git clone https://github.com/KGconcert/DB.git
# chmod 700 ./DB/*
# mysql -u root -pitbank mysql
MariaDB [mysql]> create database webdb;
MariaDB [mysql]> create user itbank@'192.168.1.%' identified by 'itbank';
MariaDB [mysql]> create user 'itbank'@'localhost' identified by 'itbank';
MariaDB [mysql]> grant all privileges on webdb.* to itbank@'192.168.1.%' identified by 'itbank';
MariaDB [mysql]> grant all privileges on webdb.* to 'itbank'@'localhost';
MariaDB [mysql]> flush privileges;
MariaDB [mysql]> exit
# mysql -u root -pitbank webdb< /root/DB/kgcondb.sql
```

```
# systemctl restart mariadb
# firewall-cmd --reload
```