



# On-Premise 구축 프로젝트 제안서

Buddy Project March 2023

Proposed to Company PLAY BUDDY

# 목 차

01	프로젝트 개요	01p
02	토폴로지 구축 설계	16p
03	네트워크 통신 검증	20p
04	웹 페이지 구현	33p
05	보안 점검 Shell Script	45p

# 01 프로젝트 개요

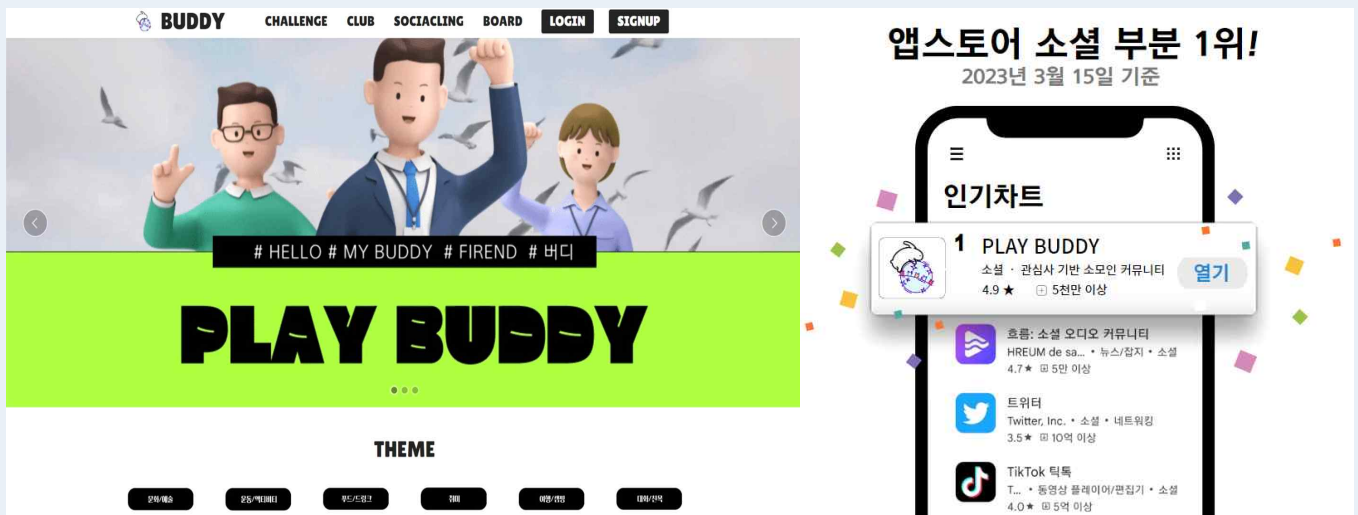
## 사업 명

고객사 "PLAY BUDDY" 인프라 확장 구축 사업

## 사업 배경

'PLAY BUDDY'는 2030 세대들을 타겟으로 한 동네 친구 찾기 및 액티비티 정보 제공 등 부담 없이 소모임부터 대모임까지 나와 취향이 비슷하고 가까운 곳에서 시간을 보낼 수 있는 친구를 만드는 소규모 커뮤니티 플랫폼이다.

### 소셜 커뮤니티 "PLAY BUDDY" 의 플랫폼 확장



최근 사회적 고립에 의해, 1인 가구가 증가하게 되었으며, 현실에서 친구를 사귀는 시간이 없어 현대인들을 위한 2030 세대뿐만 아니라, 전 연령층으로 여가생활을 즐기기 위하여 다양한 욕구와 관심이 있는 만큼 이에 따른 프로그램을 개발·지원을 하였다. 또한, 'PLAY BUDDY' 기업은 1인 가구 지원센터의 다양한 프로그램을 진행하여, 그로 인한 매출은 매년 상승하여 23배 증가되었다.

'PLAY BUDDY' 기업은 다양한 서비스를 확장할 예정이며, 지금보다 더 사용자 증가로 트래픽이 증가하여 서버 과부하 및 불안정한 서버 유지와 같은 네트워크 공급망 전반의 데이터 유형 및 운영 복잡성이 증가 될 것으로 예상된다.

또한, 최근 해킹에 의해 이용자의 개인정보가 유출되는 사건이 증가하는 보안이슈가 증가하는 만큼 고객 개인정보 유출사고 발생으로 인한, 보안 향상할 수 있고, 서버가 더욱 느려질 것으로 예정되어 사용자들에 대한 불만을 줄이기 위하여 기업은 서버 안정화를 위한 인프라 구축 의뢰하다.

## 프로젝트 일정

	WEEK1	WEEK2	WEEK3	WEEK4
시나리오 설계	시나리오 설계			
토폴로지 설계 및 구축		요구사항 분석 · 토폴로지 설계 · 서버 구축 및 운용		
웹 페이지 설계 및 구축		Django 웹 페이지 설계 및 구축		
검증 및 피드백 보완			검증 및 피드백 보완	
기술 문서 및 발표 준비				기술 문서 작성 및 발표 준비

프로젝트 준비 기간은 2023년 2월 20일부터 3월 22일까지 4주로 진행하였다. 팀원 간 여러 번의 회의를 거쳐 네트워크 구축 사업에 대한 시나리오 및 목표 설계 작업을 탄탄히 다졌고, 이를 기반으로 토폴로지를 설계하였다.

연동한 서버 통신 테스트를 위하여 총 4 대의 PC를 사용하였고, 각 PC에 WAS-DB, DNS&WEB, Proxy, Storage Server로 나누어 구축 및 운용하였다. 모든 서버의 정상적인 통신을 확인하고, 임의의 Client에서 접속 시 설계된 네트워크에 따라 통신이 이루어지며 홈페이지를 출력하는 것을 확인하였다. 테스트한 결과를 바탕으로 기술 문서 작성과 PPT 제작을 마치고 중간 발표 후 받은 피드백에 대해 검증 보완 작업을 거쳐 기술 문서와 PPT 제작을 마무리하였다.

## 프로젝트 진행 시 이용한 Tool

### 프로젝트 관리 Tool [ GitHub , Notion ]



#### 1) 형상 관리를 위한 GitHub

여러 사람들이 하나의 저장소를 가지고 보다 쉽게 협업하기 위하여 GitHub를 사용하였다. Git 의 clone 명령어를 이용하여 Django 코드를 로컬 컴퓨터에 내려받기만 하면 수월하게 웹페이지 접속이 가능하다. 또한 코드 수정 작업시 변경 이력 확인을 통한 형상 관리가 가능하다는 점에서 GitHub를 사용하였다.

#### 2) 소통 및 문서 공유를 위한 Notion

문서공유와 팀원한 소통을위하여 Notion을 사용하였다. 여러 명이 한꺼번에 실시간 작성이 가능하여 빠르게 피드백하고 수정 작업하는데 사용하였다.

### 웹 페이지 제작 Tool [ Bootstrap , Python , Django ]



프론트엔드 부분은 Bootstrap을 사용하여 웹 페이지를 디자인하였고, 백엔드 부분은 Django를 이용하여 데이터베이스와 연동하여 HTTP 요청과 응답을 처리할 수 있도록 하였다.


## 네트워크 통신 모니터링 Tool [ Wireshark ]




Wireshark 를 이용하여 네트워크 패킷을 캡처하고 설계 목적에 따라 트래픽이 분산 처리되는지 모니터링 하였다. 또한 패킷 분석을 통하여 네트워크 장애 발생시 트러블 슈팅의 용도로 활용하였다.

## GitHub [ Repository : [https://github.com/happywonhee/buddy\\_project](https://github.com/happywonhee/buddy_project) ]

### 1) WEB 페이지 Code 설계


 [happywonhee / buddy\\_project](https://github.com/happywonhee/buddy_project) Public Pin Unwatch 1

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#)

 django had recent pushes less than a minute ago [Compare & pull request](#)

django 3 branches 0 tags [Go to file](#) [Add file](#) [Code](#)

This branch is 3 commits ahead, 17 commits behind main. [Contribute](#)

 happywonhee Update settings.py ... cb62033 1 minute ago 3 commits

common	First	last week
config	Update settings.py	1 minute ago
media/review/data/2...	First	last week
mysite	First	last week



## 2) 서버 구축 및 연동 / 보안 Shell-Script

happywonhee Add files via upload fb660f6 5 days ago 17 commits		
DB Master-Slave	DB Master-Slave	last week
Security_Shell_Script.txt	Add files via upload	5 days ago
glusterfs.txt	Add files via upload	5 days ago
온프레미스.drawio	온프레미스.drawio 업데이트	last week

## 3) 장비 설정

ShinWonYeol first ad8d7eb last week 1 commit		
ASW	first	last week
CE	first	last week
DSW	first	last week
project	first	last week

 **Notion** [ <https://www.notion.so/Buddy-Project-e9c2ee0e5dd246339ee54bceaeaaab35> ]

나의 팀스페이스

B Buddy Project

Buddy Project

> Buddy Project

> 프로젝트와 작업 보기

팀스페이스 더보기

+ 새 팀스페이스

## 작업 목록 리스트

작업 목록 템플릿을 사용하면 개인 작업을 기록할 수 있습니다.

이 보드에 새 작업을 직접 생성하려면 + 새 작업 을 클릭하세요.

기존 작업을 클릭하면 추가 컨텍스트 또는 하위 작업을 추가할 수 있습니다.

Board View

필터 정렬

새로 만들기

완료 5

DB Master-Slave

네트워크 홈페이지 이미지

홈페이지 제작 (장고)

스토리지서버 구축

- 요구사항 명세서 작성
- 솔루션(기술분석)
- 1) DNS
- 2) 채널본딩(이더채널)
- 3) 서버팜/DMZ분리, VLAN
  - 프로젝트 개발 를 목록 정리 (부트스트

기술문서

+ 새로 만들기

진행 중 3

PPT - Feedback

기술문서 - Feedback

WAS 구축

+ 새로 만들기

할 일 0

+ 새로 만들기

### ■ Git Bash에서 Git 사용자 정보 등록

```
$ git config --global user.name "[이름]"
$ git config --global user.email "[github 계정에 사용한 이메일]"
$ git config --list
```

### ■ 다운 로드 받을 폴더 이동 및 Git 저장소 생성

```
$ git init
$ pwd
/c/buddy_project/buddy_project
```

### ■ 저장소 복제하기

```
$ git clone https://github.com/happywonhee/buddy_project ./
Cloning into '!'...
.....
Resolving deltas: 100% (2161/2161), done.
```

### ■ Branch 전환 ( master → django )

MINGW64 /c/buddy\_project/buddy\_project (main)

```
$ git checkout django
Updating files: 100% (6730/6730), done.
Switched to a new branch 'django'
branch 'django' set up to track 'origin/django'.
```

MINGW64 /c/buddy\_project/buddy\_project (django)

```
$ ll
-rw-r--r-- 1 Kim 197121      7 Mar 13 21:01 README.md
drwxr-xr-x 1 Kim 197121      0 Mar 13 21:01 common/
drwxr-xr-x 1 Kim 197121      0 Mar 13 21:01 config/
-rw-r--r-- 1 Kim 197121 139264 Mar 13 21:01 db.sqlite3
-rwxr-xr-x 1 Kim 197121    684 Mar 13 21:01 manage.py*
drwxr-xr-x 1 Kim 197121      0 Mar 13 21:01 media/
drwxr-xr-x 1 Kim 197121      0 Mar 13 21:01 mysite/
drwxr-xr-x 1 Kim 197121      0 Mar 13 21:01 review/
drwxr-xr-x 1 Kim 197121      0 Mar 13 21:01 static/
drwxr-xr-x 1 Kim 197121      0 Mar 13 21:01 stay/
drwxr-xr-x 1 Kim 197121      0 Mar 13 21:01 templates/
```

### ■ Commit History 조회

```
$ git log --oneline
bb260ae (HEAD -> django, origin/django) First
8f2fe10 Initial commit
$ git log
```

### ■ 원격 저장소 정보 확인

```
$ git remote -v
origin https://github.com/happywonhee/buddy_project (fetch)
origin https://github.com/happywonhee/buddy_project (push)
```

### ■ 원격 저장소로 파일 업로드 하기

```
$ git push -u origin master
# 로컬 워킹트리에만 삭제 정보가 반영되어 원격 저장소의 정보를 변경하고자 할 때에는 -f
# 옵션을 사용하여 push 진행
```

```
$ git merge [브랜치명]      # 지정한 브랜치의 커밋들을 현재 브랜치 및 작업렉터리에 반영
```



네트워크 구축 제안서

담당	팀장	사장
	최종운	

제안 업체	PLAY BUDDY	제안 일자	2023. 02. 20	제 안 인	안 유 진
소 재 지	서울시 동작구 장승배기로 171 2층, 3층				
회사 소개	'PLAY BUDDY' 는 2030 세대를 주 이용 고객으로 둔 커뮤니티 기반의 소셜 네트워크 회사이다. 이용자들이 모임을 주최하고 공통 관심사를 기반으로 공동체를 형성하며 함께 취미 시간을 공유한다. 활발한 커뮤니케이션으로 이용자 사이의 관계망이 확대됨에 따라 서비스 확장을 계획하고 있다.				
구축 사유	<ul style="list-style-type: none"><li>- 이용자 증가에 따른 서버 과부하로 인한 서비스 지연</li><li>- 최근 경쟁 업체의 서버실 화재 이슈로 서버 장애에 대한 문제의 심각성 부각</li><li>- SNS 특성상 고객들의 개인정보 유출에 민감하여 보안성 강화</li></ul>				
요구 사항	<ul style="list-style-type: none"><li>- 서버 과부하 문제를 완화할 수 있는 솔루션 필요</li><li>- 갑작스러운 장애를 대비한 이중화 작업으로 가용성 확보</li><li>- 보안성 강화로 인해 사이트 이용자 및 회사 내부 정보 보호</li></ul>				
비 고	2024년 메타버스로 서비스 확장으로 AWS 마이그레이션 예정 [ BUDDY-VERSE 출시 ]				

## 요구사항 분석



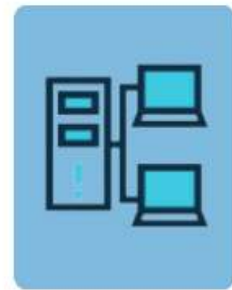
안정적인  
서버 운영관리



DB 안정성 보장



보안 강화



가용성 향상

### 안정적인 서버 운영 관리

고객사는 이용자가 증가하면서 서버 과부하로 인해 웹 사이트 이용 속도가 느려짐에 따라 고객센터에 이용자들의 불만이 폭주하는 문제를 겪고 있다. 갑작스러운 트래픽 증가에도 이용자에게 일관하게 안정적인 속도를 제공하기를 원한다.

### DB 안정성 보장

고객사는 갑작스러운 장애 또는 재해 발생 시 빠르게 복구되길 원한다. 회원들의 개인정보 및 커뮤니티 관련 데이터를 저장하는 DB 백업 솔루션을 통해 안정성 확보를 요구한다.

### 보안 강화

이용자들의 개인정보 및 회사 내부 정보 유출에 대한 우려로 중요한 데이터를 처리하는 서버는 외부와의 직접적인 통신을 차단하기를 원한다.

### 가용성 향상

고객사는 2022년 10월, 카카오 데이터 센터 화재에 대해 이야기하며 서버 가용성 확보를 원한다. 카카오톡의 SK C&C 데이터 센터 화재는 127시간의 복구 시간이 소요되면서 전 국민이 불편함을 겪은 사건이다. 카카오 측은 서비스 복구가 늦어진 가장 큰 이유로 서버 이중화 조치 미흡을 인정하였고, 사회 전반에 서버 이중화에 대한 중요성이 부각 되었다.

고객사 또한 장애 및 재해 발생에 대비하여 모든 서버에 대하여 이중화 구성을 통하여고가용성 솔루션을 요구한다.

## 서버 기술 분석

### DB



- 서비스의 데이터를 저장 & 관리하는 서버

#### 기술 및 구성요소

#### 내 용

#### Replication

- Master 1대, Slave 2대로 Replication을 구성하여 MasterDB에서의 데이터를 2대의 SlaveDB 로 동기화함.

#### [ Replication 구성을 사용한 이유 ]

##### 1. 백업 용이성

장애가 발생한 경우 데이터를 복제받은 Slave DB 서버를 이용하여 빠르게 백업을 수행수 있으므로 백업에 소요되는 시간과 비용을 줄일 수 있다.

##### 2. 읽기와 쓰기 작업 분리

MasterDB는 쓰기작업을 SlaveDB는 읽기작업을 처리는데, 이 때 DB 요청의 60~80% 정도가 대부분 읽기 작업이다. Replication을 통해서 읽기와 쓰기 작업을 처리하는 서버를 분리하여 읽기 성능을 향상시킬 수 있다.

### Storage server



- 대용량의 데이터를 저장하고 관리하는 서버
- 서버와 저장소 장치를 연결하여 대규모의 데이터를 저장하고, 관리 및 보호

#### 기술 및 구성요소

#### 내 용

#### GlusterFS

- 분산 파일 시스템으로서, 다양한 스토리지 서버를 네트워크를 통해 하나의 디스크 풀로 묶어 사용함
- 확장성과 유연성 : 저장소의 크기를 확장할 수 있으며, 서버를 추가하면 스토리지 용량도 확장 가능
- 고가용성 : 데이터를 소프트웨어적으로 복제해 저장할 수 있어

데이터의 안정성과 가용성을 보장하며 디스크 장애 시 자가 복구 기능을 제공함. 또한 스냅샷 기능을 제공하여, 데이터의 백업 및 롤백을 쉽게 수행할 수 있음

- 파일 정보를 각 서버마다 가지고 있으므로 일반적으로 분산 파일 시스템에서 발생하는 중앙 집중식 메타데이터 서버의 부하를 해결할 수 있음

#### [ GlusterFS에서 사용되는 볼륨의 기본 단위 " Brick " ]

- 'Brick' 은 GlusterFS가 제공하는 볼륨의 기본 단위로 데이터의 저장과 관리를 담당함
- 각 브릭은 하나의 서버 노드의 스토리지를 나타내며, 볼륨은 이러한 브릭들을 조합하여 구성됨
- 일반적으로 디렉토리 형태로 구성되어 있으며, GlusterFS 클라이언트에서는 이 디렉토리를 마운트하여 데이터를 읽거나 씸
- [노드IP:마운트포인트] 형태로 표시

#### [ Volume 생성 방법 ]

##### 1. Distributed Volume

- 파일의 데이터가 여러 서버 노드에 분산되어 저장되는 볼륨 유형 (기본)
- 사용 목적 : 쉽고 저렴하게 Scale-Out 가능
- 단점 : Brick 장애 발생시 데이터 유실

##### 2. Replicated Volume

- 파일의 데이터가 여러 서버 노드에 복제되어 저장되는 볼륨 유형
- 사용 목적 : 데이터의 Redundancy를 통한 데이터 안정성 보장
- 단점 : 서버 총 용량의 1/2 이상 사용 불가

##### 3. Stripe Volume

- 파일의 데이터가 여러 서버 노드에 나누어 저장되는 볼륨 유형

\*\* 토폴로지 상에서는 데이터의 안정성을 목적으로 Replication Volume을 사용하여 구성함.

## Proxy

- 클라이언트와 서버 간의 통신을 중계하는 대리 서버



- 클라이언트의 요청을 받아서 서버에 대신 요청을 전달하고, 서버에서 받은 응답을 클라이언트에게 전달

## 기술 및 구성요소

## 내 용

### HAProxy

- 오픈소스 기반의 높은 성능과 로드 밸런싱을 제공하는 TCP / HTTP 기반의 어플리케이션, 역프록시 기능을 제공

### [ HAProxy 의 기능 ]

#### 1) Load Banlancing

- 로드밸런싱 기법은 L4 / L7 로드밸런싱이 존재한다 L4는 차례대로 서버에게 요청을 하며 L7는 7계층 기반의 밸런싱을 진행하여 사용자의 정보(Cookie)에 저장

- 사용자의 요청이 (10.10.20.20 :80) 라고 한다면 프록시 서버가 먼저 해당 트래픽을 받고 적절하게 Was서버에의 요청을 사용자에게 전달시켜줌

- 로드 밸런싱 중 라운드 로빈 방식을 활용하여 특정 웹 서버에게 트래픽이 몰리는걸 방지하여 프로세스의 부하를 방지함

#### 2) VRRP

- 마스터와 슬레이브 구조에서 게이트웨이를 가상의 라우터로 설정하여 한대의 서버가 다운이 되어도 서비스를 유지함

#### 3) 단일 End Point 구성

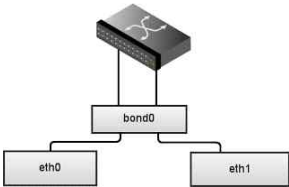
- 단일 EndPoint를 구성하여 내부 네트워크망의 변경된 설정에 대해 웹 프록시 서버에만 등록하면 되므로 관리 용이

#### 4) SSL/TLS 설정

- 사용자의 요청을 먼저 받는다는 특징으로 프록시 서버에 SSL 인증을 받아 웹 서버의 수가 유동적으로 변동이 되어도 사용자는 SSL 인증을 받은 웹서비스를 이용할 수 있어 패킷을 암호화하여 보호받을 수 있음

<b>Cookie/Session</b>	- 사용자의 이전 요청을 저장하여 빠른 서비스를 제공

Channel Bonding



- 여러 개의 NIC 장치를 하나로 묶어 단일 Channel 형식 구성

기술 및 구성요소	내 용
<ul style="list-style-type: none"> <li>- Active - Backup의 Mode 1 로 구성됨</li> <li>- 각 서버와 직접 연결되어 있는 2개의 NIC를 Bond0 으로 묶음</li> <li>- Master Device 에 장애 발생 시 Bond0 내의 대체 NIC 가 통신을 수행</li> <li>- Fault Tolerance 환경 구축과 단일 인터페이스에 대한 Redundancy 구현을 통한 가용성 향상</li> </ul>	



## 장비 기술 분석

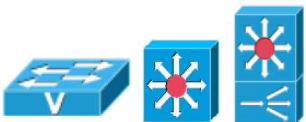
### 네트워크 장비 공통



- Lan, Wan끼리 연결이 가능하게 하여 인터넷을 구성하는 장비

기술 및 구성요소	내 용
보안설정	- 콘솔접속과, 원격 접속, enable 접속시 ID와 Passwd를 통해 인증을 받고 원격 접속가능 인원을 최소화, 세션 연결시간을 지정하여 보안성 향상
Private IP	- 내부 사설망에서만 사용할 수 있는 IP를 사용하여 외부 통신을 차단해 보안성 향상
LACP	- EtherChannel의 프로토콜 중 표준 프로토콜로 여러개의 링크로 구성된 링크들이 rstp로 연결이 끊겨 속도 향상을 목적으로 모든 링크를 하나의 그룹으로 묶어 관리를 하는 프로토콜
dot1q	- vlan으로 나누어진 환경에서 Tag정보를 통해 정보 교환
Duplexing	- 두 개 이상의 링크를 연결하여 하나의 링크가 끊어져도 통신이 가능하게 한다

### 스위치 장비 공통

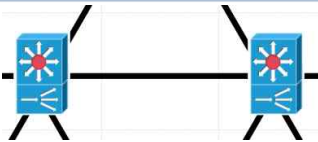


서로 동일한 네트워크 영역에서 프레임을 전달하는 네트워크 장비

RSTP	- 스위치의 장비 특징으로 인하여 발생하는 스토밍현상에 의해 무한 루프에 빠지게 되는 현상을 해결하기 위하여 하나의 링크를 끊어놓는 기술, stp에 비해 복구속도가 빠르다
------	---

<b>VLAN</b>	- 동일 네트워크 영역을 vlan을 통해 서로 다른 네트워크로 분리하여 다이크트 브로드캐스트를 방지하여 보안성을 향상시킬 수 있다
<b>Port Security</b>	- 스위치는 프레임이 들어오게 되면 들어왔던 프레임의 송신지 Mac주소를 학습한다 이때 학습하는 Mac주소의 양을 제한하여 Macof 공격을 통한 프로세스 과부하를 방지한다

## Distribute 스위치



여러 호스트들의 프레임이 통과되는 스위치

<b>Inter-VLAN</b>	- 서로 다른 vlan 끼리 나누어진 장비끼리 통신을 하기 위하여 L3 스위치로 구성
<b>VRRP</b>	- VRRP 표준 프로토콜로 Gateway를 가상으로 두어 이중화를 통한 fault tolerance 환경을 유지

## Server Farm과 DMZ

- Server Farm 과 DMZ 영역을 분리하여 외부에서의 접근 통제

<b>Server Farm 과 DNS 영역 분리</b>	<ul style="list-style-type: none"> <li>- Server Farm : 외부 통신의 직접 접근이 불가능한 영역 WEB, WAS, DB, Storage, 사내 PC</li> <li>- DMZ : 불특정 Client 의 접속이 가능한 영역 Proxy , DNS</li> </ul> <p><b>[ Server Farm 와 DMZ 분리를 통한 기대 효과 ]</b></p> <p><b>1) 보안성 강화</b> Server Farm 에 외부 통신의 직접 접근을 방지하여 내부네트워크 보호</p> <p><b>2) 불필요한 트래픽 감소</b></p>
--------------------------------	--

	<p>Server Farm 은 내부적으로 다양한 서비스가 실행되고 있음.          이때 외부 통신에 대하여 DMZ 에서 필요한 트래픽만 Server Farm 에 전달하여 리소스 관리 가능</p> <p><b>3) 관리 용이성</b>          외부 통신이 가능한 DMZ 에 문제 발생시 Server Farm 으로 전파 되는 것을 방지</p>
--	--

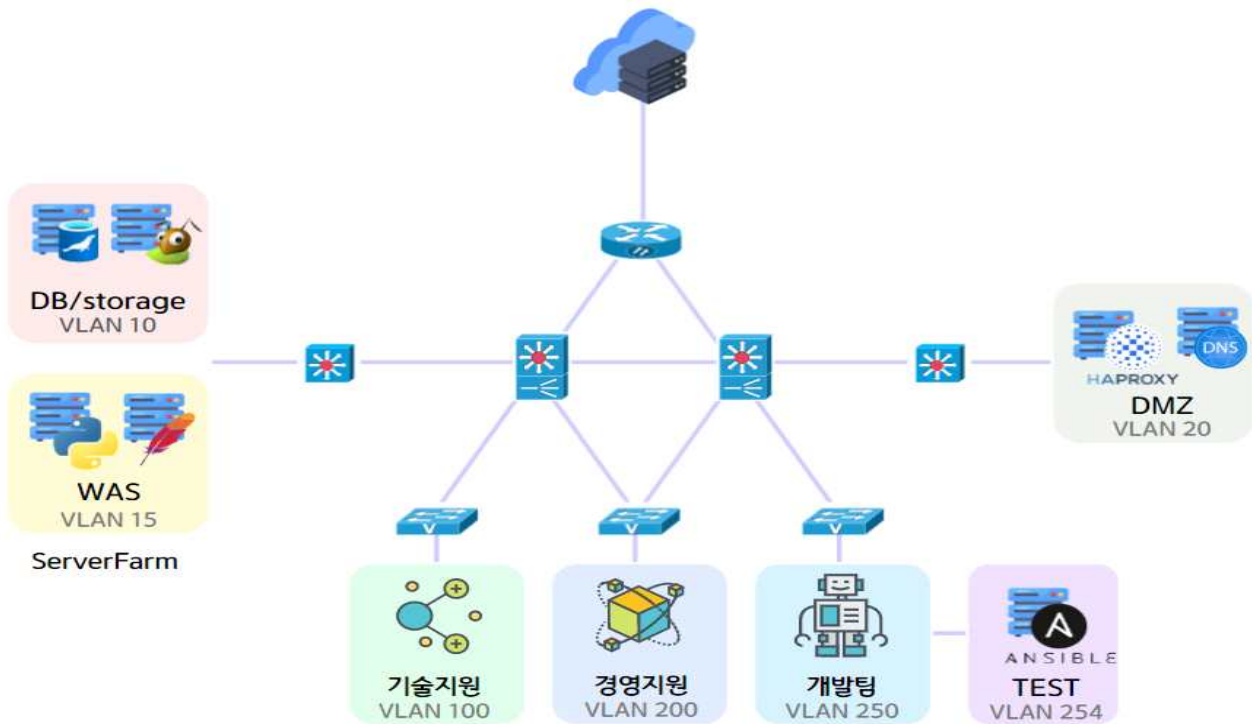
## 라우터



	VLAN 분리를 통한 논리적인 네트워크 대역 분리
Static Routing	- 서로 다른 네트워크 끼리 라우팅 테이블을 통해 패킷을 교환할 수 있도록 설정하고, 관리자가 직접 Static하게 지정
NAT	- 사설 IP를 사용하는 환경에서 외부와의 통신을 위해 사설 IP를 공인 ip로 변경. 이를 통해 내부에서는 인터넷을 사용할 수 있는 환경이 됨
DHCP	<ul style="list-style-type: none"> <li>- IP 주소를 DHCP를 통하여 동적으로 설정하여 자동으로 IP를 받는 환경 으로 구성</li> <li>- 이를 통해 내부 네트워크를 사용하는 사용자에게 동적으로 IP를 할당</li> </ul>

## 02 네트워크 구축 설계

### 온프레미스 토폴로지



#### 온프레미스란 ?

데이터를 회사 전산실 서버에 저장·관리하는 형태로 회사가 자체적으로 보유한 공간에 물리적으로 하드웨어 장비를 가지고 직접 인프라를 구축하는 방식.

특히, 회사가 대규모의 데이터를 처리해야 하는 경우 온프레미스 방식이 효과적임.

#### 왜 온프레미스를 사용하는가?

##### 비용

초기 투자 비용이 높지만, 운영이 장기화됨에 따른 비용상승이 제한적이며 유지보수 비용 이외의 추가비용이 적음.

##### 보안

자체 보안 시스템을 구축하고 운영함으로써 회사의 중요 데이터와 애플리케이션을 외부로부터 보호하는 데에 도움을 줌

##### 커스터마이징

사용자 필요와 요구에 따라 맞춤형 구축이 가능하므로 조직의 업무 흐름과 비즈니스 요구에 더 적합한 애플리케이션을 만들어내는 데에 도움을 줌

##### 직접적인 제어

회사가 자체 데이터 센터나 서버를 보유하고 운영하므로, 데이터와 애플리케이션에 대해 직접적인 제어가 가능함. 보안, 규정 준수, 자원 활용 등의 이점 제공

##### 성능

조직이 자체 하드웨어를 보유하고 운영하므로, 필요에 따라 하드웨어를 업그레이드하거나 최적화가 가능함. 따라서 애플리케이션의 성능과 가용성을 향상시키는 데에 도움을 줌

## 기분 네트워크 정보

### 장비 별 IP 설정 정보

구분	VLAN	장비	IP
DMZ	VLAN 100	PROXY_Master	192.168.100.7
		PROXY_Slave	192.168.100.6
		DNS_Master	192.168.100.5
		DNS_Slave	192.168.100.4
Serverfam	VLAN 110	DB_Master	192.168.110.2
		DB_Slave 1	192.168.110.3
		DB_Slave 2	192.168.110.4
		Storage server 1	192.168.110.5
		Storage server 2	192.168.110.6
WEB Server	VLAN 120	WEB 1	192.168.120.2
		WEB 2	192.168.120.3
		WAS 1	192.168.120.4
		WAS 2	192.168.120.5
개발사업팀	VLAN 10	PC1	192.168.10.2
		PC2	192.168.10.3
기술지원팀	VLAN 20	PC3	192.168.20.2
		PC4	192.168.20.3
경영관리팀	VLAN 30	PC5	192.168.30.2
		PC6	192.168.20.3
TEST	VLAN 15	TEST PC	192.168.15.2

## 인터페이스별 IP 설정 정보

구분	VLAN	장비	IP
DSW1 - CE	Net		192.168.31.0/30
	DSW1	e 0/0	192.168.31.1
	CE	e 0/0	192.168.31.2
DSW2 - CE	Net		192.168.31.4/30
	DSW2	e 1/1	192.168.31.5
	CE	e 1/1	192.168.31.6
DSW1 - DSW2	Net		192.168.31.8/30
	DSW1	e 1/2	192.168.31.9
	DSW2	e 1/2	192.168.31.10

## SVI 및 VLAN Gateway 정보

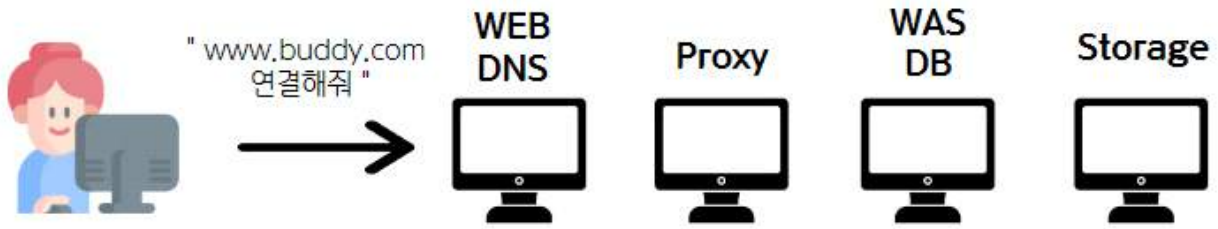
구분	VLAN_GW	구분	VLAN_GW
SVI 10	192.168.10.252/30	SVI 10	192.168.10.253/30
SVI 20	192.168.20.252/30	SVI 20	192.168.20.253/30
SVI 30	192.168.30.252/30	SVI 30	192.168.30.253/30
SVI 40	192.168.40.252/30	SVI 40	192.168.40.253/30
SVI 100	192.168.100.252/30	SVI 100	192.168.100.253/30
SVI 200	192.168.200.252/30	SVI 200	192.168.200.253/30



## 연결 포트정보

장비	포트	장비	포트
ASW1 – DSW1	e2/0 ~ e2/1	ASW1 – DSW2	e3/0 ~ e3/1
ASW2 – DSW1	e3/0 ~ e3/1	ASW2 – DSW2	e2/0 ~ e2/1
ASW3 – DSW1	e2/2 ~ e2/3	ASW3 – DSW2	e3/2 ~ e3/3
ASW4 – DSW1	e1/2 ~ e1/3	ASW4 – DSW2	e0/2 ~ e0/3
ASW5 – DSW1	e0/1 ~ e0/2	ASW5 – DSW2	e0/0 ~ e0/1
CE - DSW1	e0/0	CE - DSW2	e1/1
DSW1 - DSW2	e1/0	CE - ISP	e 1/0 ~ nat0

### 03 네트워크 통신 검증



4대의 PC를 이용하여 통신 테스트를 진행하였고, 각 서버에 WAS&DB, DNS&WEB, Proxy, Storage Server로 나누어 구축하였다. Client가 접속시 모든 서버가 정상적인 통신이 되어 서로 연동됨을 확인하고, Client에서 접속 시 설계된 네트워크에 따라 통신이 이루어지며 홈페이지를 출력하는 것을 확인하였다.

#### 통신 검증용 IP

구분	장비	IP
DMZ	PROXY_Master	192.168.1.140
	PROXY_Slave	192.168.1.141
	PROXY_VIP	192.168.1.150
	DNS_Master	192.168.1.120
	DNS_Slave	192.168.1.121
Serverfam	DB_Master	192.168.1.111
	DB_Slave 1	192.168.1.112
	DB_Slave 2	192.168.1.113
	Storage server 1	192.168.1.130
	Storage server 2	192.168.1.131
WEB Server	WEB 1	192.168.1.101
	WEB 2	192.168.1.102
	WAS 1	192.168.1.114
	WAS 2	192.168.1.115
Client		192.168.1.99

## + 검증 항목

구 분	항 목	검 증 결 과
DB	WAS와 DB의 연동 및 DB Replication 검증	데이터 처리 흐름 및 데이터 백업을 통한 DB 안정성 제공
Storager	GlusterFS의 Replication 동작 검증	데이터의 안정성, 높은 가용성, 데이터 백업
HAProxy	Client 접근시 Proxy Server의 요청 전달 동작 검증	Client 의 접근시 Proxy에서 요청을 전달받고 응답받는 데이터 처리 흐름 파악
	HAProxy 의 고가용성 검증	Master Proxy 장애시 Slave Proxy 에서의 정상 동작을 통한 고가용성 제공
	HAProxy 의 Load Balancing 기능 검증	로드 밸런싱을 통한 안정적인 서버 운영 제공
	HAProxy 의 SSL/TLS 암호화 검증	Client가 HTTP 프로토콜로 접근시 HAProxy에서 HTTPS로 Redirection하여 서버간 통신에서의 보안 강화
DNS	DNS Server 의 이중화 동작 검증	DNS Server 이중화를 통한 가용성 제공

## 검증 결과

### WAS와 DB의 연동 및 DB Replication 검증

게시판에서 게시글을 작성하고 삭제했을 때 DB에서의 확인이 가능한지 검증하고, Master DB(192.168.1.111)와 Slave DB (192.168.1.112)로의 Replication이 정상적으로 동작하는지 검증한다.

write by admin

Created by admin on 2023년 3월 21일 10:07 오전

첨부파일 :

test123

목록

수정

삭제

▶ 게시판에서 write by admin이라는 제목으로 게시글을 작성한다.

```
[root@DB1 ~]# hostnamectl
```

```
Static hostname: DB1
```

```
[root@DB1 ~]# ifconfig
```

```
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
```

```
inet 192.168.1.111 netmask 255.255.255.0 broadcast 192.168.1.255
```

```
MariaDB [django_db]> select * from review_review;
```

id	title	content	file_upload	create_date	modify_date	author_id
1	test1	test		2023-03-06 09:23:12.460930	2023-03-06 09:23:12.460952	2
3	write by admin	test123		2023-03-21 10:07:24.240270	2023-03-21 10:07:24.240292	1

```
2 rows in set (0.000 sec)
```

▶ Master DB (192.168.1.111)에서 방금 생성한 게시판 글 확인을 통하여 WAS와 DB가 연동된 것을 확인하였다.

```

MariaDB [(none)]> show slave status;
+-----+-----+-----+-----+-----+-----+-----+
| Slave_IO_State | Master_Host | Master_User | Master_Port | Connect_Retry | Master_Log_File | Read_Master_Log_Pos |
| Relay_Log_File | Relay_Log_Pos | Relay_Master_Log_File | Slave_IO_Running | Slave_SQL_Running | Replicate_Do_DB |
+-----+-----+-----+-----+-----+-----+-----+
| Waiting for master to send event | 192.168.1.111 | Rep_user | 3306 | 60 | mysql-bin.000011 |
342 | DB2-relay-bin.000032 | 641 | mysql-bin.000011 | Yes | Yes | django_db |
+-----+-----+-----+-----+-----+-----+

```

Slave\_IO\_State : Waiting for master to sen event

Master\_ Host : 192.168.1.111 # Master\_DB 의 IP 주소확인

**Slave\_IO\_Running : YES**

**Slave\_SQL\_Running : YES**

Replicate\_Do\_DB : django\_db

- ▶ Slave DB 로 설정한 DB 에서의 "show slave status" 명령어를 통한 상태 확인을 통하여 Replication 이 정상적으로 작동하는 것을 확인하였다.

```

[root@DB2 ~]# hostnamectl
    Static hostname: DB2

[root@DB2 ~]# ifconfig
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.1.112 netmask 255.255.255.0  broadcast 192.168.1.255

MariaDB [django_db]> select * from review_review;
+----+-----+-----+-----+-----+-----+
| id | title          | content | file_upload | create_date          | modify_date          |
+----+-----+-----+-----+-----+-----+
| 1  | test1          | test    |              | 2023-03-06 09:23:12.460930 | 2023-03-06 09
| 3  | write by admin | test123 |              | 2023-03-21 10:07:24.240270 | 2023-03-21 10
+----+-----+-----+-----+-----+-----+

```

- ▶ Master DB (192.168.1.111) 의 데이터가 Slave DB (192.168.1.112) 로 실시간 백업되는 것을 확인할 수 있다.


CHALLENGE

www.buddy.com 내용:  
정말로 삭제하시겠습니까?

확인
취소

BOARD

REVIEW
LOGIN

write by admin  
Created by admin on 2023년 3월 21일 10:07 오전

첨부파일 :  
test123

목록

수정

삭제

- ▶ 게시판에서 게시글을 삭제한다.

```
[root@DB1 ~]# hostnamectl
Static hostname: DB1

[root@DB1 ~]# ifconfig
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.1.111  netmask 255.255.255.0  broadcast 192.168.1.255

MariaDB [django_db]> select * from review_review;
+----+-----+-----+-----+-----+-----+-----+
| id | title | content | file_upload | create_date           | modify_date           | author_id |
+----+-----+-----+-----+-----+-----+-----+
|  1 | test1 | test    |              | 2023-03-06 09:23:12.460930 | 2023-03-06 09:23:12.460952 |         2 |
+----+-----+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

- ▶ Master DB (192.168.1.111)에서 웹 페이지에서 삭제한 게시글이 삭제한 것을 확인할 수 있다.

```
[root@DB2 ~]# hostnamectl
Static hostname: DB2

[root@DB2 ~]# ifconfig
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.1.112  netmask 255.255.255.0  broadcast 192.168.1.255

MariaDB [django_db]> select * from review_review;
+----+-----+-----+-----+-----+-----+
| id | title | content | file_upload | create_date           | modify_date           |
+----+-----+-----+-----+-----+-----+
|  1 | test1 | test    |              | 2023-03-06 09:23:12.460930 | 2023-03-06 09:23:12.460952 |
+----+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)
```

- ▶ Master DB (192.168.1.111)에서 데이터를 삭제한 후 Slave DB (192.168.1.112) 에서도 게시글이 삭제 된 것을 확인할 수 있다.



Internet Protocol Version 4, Src: 192.168.1.102, Dst: 192.168.1.111

Source Address: 192.168.1.102 **WEB Server 2**

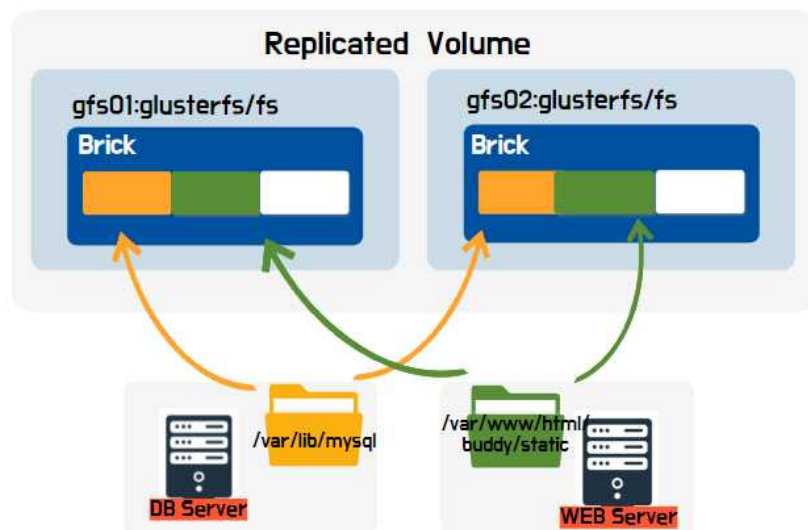
Destination Address: 192.168.1.111 **Master DB**

336	12.986957	192.168.1.111	192.168.1.102	MySQL	702 Response TABULAR Response
337	12.993212	192.168.1.102	192.168.1.111	MySQL	346 Request Query
338	12.993650	192.168.1.111	192.168.1.102	MySQL	639 Response TABULAR Response
339	12.994907	192.168.1.102	192.168.1.111	MySQL	397 Request Query

### ▶ 패킷 트레이서에서의 검증

Client에서 웹 페이지 요청시 WEB Server(192.168.1.102)에서 Master DB (192.168.1.111)로 웹 페이지를 요청하고, Master DB에서 WEB Server로 요청에 응답하는 패킷을 확인할 수 있다.

## GlusterFS의 Replication 동작 검증



Storage Server는 대용량 데이터를 저장하는 역할로 서버와 저장소 장치를 연결하여 대규모의 데이터를 저장하고 관리한다. 시나리오 속 '고가용성', '데이터 안정성'의 고객 요구 사항을 충족시키기 위해 GlusterFS Replication Volume을 사용하였다. Replication Volume은 데이터가 여러 서버에 복제되어 저장되기 때문에 높은 가용성을 지원하여 데이터의 손실 없이 시스템을 작동시킬 수 있다.

이번 검증에서는 MariaDB의 데이터 디렉토리 /var/lib/mysql과 WEB Server의 정적 파일을 저장하는 /static 디렉토리를 GlusterFS Replication Volume에 마운트하여 서버 노드로 분산 저장되는지 확인한다.

```
[root@gluster1 ~]## gluster volume info vol
Volume Name: vol
Type: Replicate
Volume ID: 6f202605-5dee-4bdf-b1b4-ac4a402ad29a
Status: Started
Snapshot Count: 0
Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1: gfs01:/glusterfs/fs
Brick2: gfs02:/glusterfs/fs
Options Reconfigured:
cluster.granular-entry-heal: on
storage.fips-mode-rchecksum: on
transport.address-family: inet
nfs.disable: on
performance.client-io-threads: off
```

- ▶ glusterfs 볼륨 정보에서 Replication 타입의 볼륨이 생성되어 있으며, gfs01:/glusterfs/fs와 gfs02:/glusterfs/fs가 하나의 Volume Pool 로 묶인 것을 확인할 수 있다.

```
[root@web01 ~]# ls -l /var/www/html/buddy/static/
합계 25
drwxr-xr-x 3 root root    82  3월 17 20:00 glusterfs
drwx----- 2 root root     6 10월 11 21:57 httpd24
-rw----- 1 root root  3639  3월 17 19:57 secure
-rw----- 1 root root  3840  3월 17 19:18 tallylog
-rw-r--r-- 1 root root   17  3월 21 10:14 test
-rw----- 1 root root 16195  3월 17 19:28 yum.log
```

- ▶ WEB Server 의 정적 이미지 데이터를 저장하는 /var/www/html/buddy/static 디렉토리를 glusterfs 와 마운트한다.

```
[root@Gluster01 ~]# ls -l /glusterfs/fs/
합계 28
drwxr-xr-x. 3 root root    82  3월 17 20:00 glusterfs
drwx-----, 2 root root     6 10월 11 21:57 httpd24
-rw-----, 2 root root  3639  3월 17 19:57 secure
-rw-----, 2 root root  3840  3월 17 19:18 tallylog
-rw-r--r--, 2 root root   17  3월 21 10:14 test
-rw-----, 2 root root 16195  3월 17 19:28 yum.log
```

```
[root@Gluster02 ~]# ls -l /glusterfs/fs/
합계 28
drwxr-xr-x. 3 root root    82  3월 17 20:00 glusterfs
drwx-----, 2 root root     6 10월 11 21:57 httpd24
-rw-----, 2 root root  3639  3월 17 19:57 secure
-rw-----, 2 root root  3840  3월 17 19:18 tallylog
-rw-r--r--, 2 root root   17  3월 21 10:14 test
-rw-----, 2 root root 16195  3월 17 19:28 yum.log
```

- ▶ 마운트한 디렉토리의 데이터가 2대의 GlusterFS의 서버 노드 Brick 에 각각 분산되어 저장된 것을 확인할 수 있다.

```
[root@mariadb-m ~]# ls -l /var/lib/mysql/django_db/
합계 996
-rw-rw---- 1 mysql mysql 1884 3월 21 10:53 auth_group.frm
-rw-rw---- 1 mysql mysql 81920 3월 21 10:53 auth_group.ibd
-rw-rw---- 1 mysql mysql 1965 3월 21 10:53 auth_group_permissions.frm
-rw-rw---- 1 mysql mysql 98304 3월 21 10:53 auth_group_permissions.ibd
-rw-rw---- 1 mysql mysql 2564 3월 21 10:53 auth_permission.frm
-rw-rw---- 1 mysql mysql 81920 3월 21 10:53 auth_permission.ibd
-rw-rw---- 1 mysql mysql 4206 3월 21 10:53 auth_user.frm
..... 생략 .....
```

- ▶ 이번에는 DB 서버의 데이터 디렉토리를 Glusterfs 와 마운트한다.

```
[root@Gluster01 ~]# ls /glusterfs/fs/django_db/
auth_group.frm          auth_user.frm          db.opt
django_migrations.ibd
auth_group.ibd          auth_user.ibd          django_admin_log.frm    django_session.frm
auth_group_permissions.frm auth_user_groups.frm    django_admin_log.ibd    django_session.ibd
auth_group_permissions.ibd auth_user_groups.ibd    django_content_type.frm review_review.frm
auth_permission.frm     auth_user_user_permissions.frm django_content_type.ibd review_review.ibd
auth_permission.ibd     auth_user_user_permissions.ibd django_migrations.frm
```

```
[root@Gluster02 ~]# ls /glusterfs/fs/django_db/
auth_group.frm          auth_user.frm          db.opt
django_migrations.ibd
auth_group.ibd          auth_user.ibd          django_admin_log.frm    django_session.frm
auth_group_permissions.frm auth_user_groups.frm    django_admin_log.ibd    django_session.ibd
auth_group_permissions.ibd auth_user_groups.ibd    django_content_type.frm review_review.frm
auth_permission.frm     auth_user_user_permissions.frm django_content_type.ibd review_review.ibd
auth_permission.ibd     auth_user_user_permissions.ibd django_migrations.frm
```

- ▶ DB 서버의 데이터 디렉토리 또한 Glusterfs 의 volume 에 분산되어 저장되었다.

## Client 접근시 HAProxy Server의 요청 전달 검증



HAProxy는 클라이언트와 서버 사이에서 동작하며, 클라이언트로부터 요청을 받아서 서버로 전달하고, 서버에서 받은 응답을 클라이언트에게 전달한다. 이를 통해 클라이언트와 서버 사이에 있는 불필요한 연결을 제거하고, 보안성을 높일 수 있다.

와이어 샤크를 이용한 패킷 분석을 이용하여 Client 가 HTTP 요청을 했을 때 HAProxy 가 서버로 요청을 제대로 전달하고 Client 에게 응답할 수 있는지 검증한다.

### [ Client → HAproxy VIP ]

Internet Protocol Version 4, Src: 192.168.1.99, Dst: 192.168.1.150

Source Address: 192.168.1.99

Destination Address: 192.168.1.150

### [ Master HAproxy → WEB\_2 ]

Internet Protocol Version 4, Src: 192.168.1.140, Dst: 192.168.1.102

Transmission Control Protocol, Src Port: 38440, Dst Port: 80, Seq: 0, Len: 0

### [ WEB\_2 → Master HAproxy ]

Internet Protocol Version 4, Src: 192.168.1.102, Dst: 192.168.1.140

Transmission Control Protocol, Src Port: 80, Dst Port: 38440, Seq: 0, Ack: 1, Len: 0

### [ HAproxy VIP → Client ]

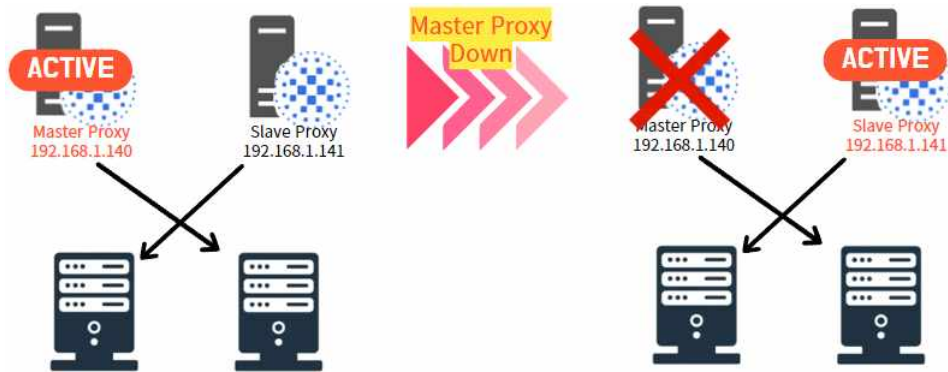
Internet Protocol Version 4, Src: 192.168.1.150, Dst: 192.168.1.99

Source Address: 192.168.1.150

Destination Address: 192.168.1.99

- ▶ Client에서 웹페이지 요청시 Proxy에서 요청을 받아 WEB Server로 요청을 전달한다. 요청을 받은 WEB Server 는 요청에 응답하고 Proxy 가 요청에 대한 응답을 받아 Client 로 전달하면 Client에서 웹페이지로 접속이 가능하다.

## HAProxy 의 고가용성 검증



VRRP(Virtual Router Redundancy Protocol)는 가상 라우터를 사용하여 다수의 라우터들이 물리적으로 한 대의 라우터처럼 동작하도록 하는 프로토콜로 HAProxy는 VRRP를 사용하여 여러 대의 HAProxy 서버가 하나의 가상 IP 주소를 공유하도록 구할 수 있다.

만약 HAProxy 서버 그룹에서 한 대의 서버에 장애가 발생하면, 다른 HAProxy 서버가 이를 감지하고 가상 IP 주소를 취득한다. 이를 통해 장애가 발생한 HAProxy 서버를 대신하여 다른 서버가 클라이언트 요청을 처리하여 서비스의 가용성을 유지하므로 업무 중산 시간을 최소화할 수 있다.

이번 검증은 Master HAProxy의 전원을 차단한 경우 Client가 웹페이지를 요청했을 때 Slave HAProxy 에서 가상 IP를 취득하여 Client 의 요청에 정상적 응답을 하는지를 확인한다.

[ Slave HAProxy → WEB\_1 ]

Internet Protocol Version 4, Src: 192.168.1.141, Dst: 192.168.1.101

Source Address: 192.168.1.141

Destination Address: 192.168.1.101

[ WEB\_1 → Slave HAProxy ]

Internet Protocol Version 4, Src: 192.168.1.101, Dst: 192.168.1.141

Source Address: 192.168.1.101

Destination Address: 192.168.1.141

\* Slave Proxy Server : 192.168.1.141

192.168.1.141	192.168.1.101	HTTP
192.168.1.101	192.168.1.141	TCP

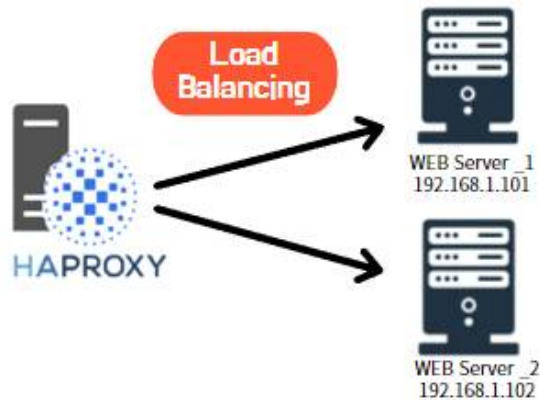
▶ Master Proxy Server를 일시정지 후, Client에서 웹페이지로 접속한다. 이때, Slave Proxy Server 가 요청을 전달하여 서버의 가용성을 제공한다.

192.168.1.141	224.0.0.18	VRRP
192.168.1.141	192.168.1.102	TCP
192.168.1.102	192.168.1.141	TCP
192.168.1.141	192.168.1.102	HTTP

▶ Master Proxy Server 가 정상적으로 동작하지 않아 Slave Proxy Server에서 가상 IP 를 가져와 요청과 응답을 전달하는 것을 확인할 수 있다.



## HAProxy 의 Load Balancing 기능 검증



HAProxy를 단일 엔드 포인트로 구성하여 여러 서버나 서비스로 Load Balancing 이 가능하다. HAProxy 는 Load Balancer로서 Client의 요청을 여러 대의 서버로 분산시킴으로써 서비스의 가용성과 성능을 향상시킨다.

검증을 통하여 Client에서 웹서버에 HTTP 요청시 HAProxy에서 WEB Server\_1(192.168.1.101)과 WEB Server\_2 (192.168.1.102) 로 로드 밸런싱되는 것을 확인한다.

No.	Time	Source	Destination	Protocol	Length	Info
39488	18.545701	192.168.1.141	192.168.1.101	QUIC	75	Protected Payload
39489	18.558503	192.168.1.141	192.168.1.101	TCP	74	49582 → 80 [SYN]
39490	18.558624	192.168.1.101	192.168.1.141	TCP	74	80 → 49582 [SYN,
39491	18.558787	192.168.1.141	192.168.1.101	HTTP	104	HEAD / HTTP/1.0
39492	18.558886	192.168.1.101	192.168.1.141	TCP	66	80 → 49582 [ACK]
39493	18.560817	192.168.1.101	192.168.1.141	HTTP	262	HTTP/1.1 500 Int
39494	18.560870	192.168.1.101	192.168.1.141	TCP	66	80 → 49582 [FIN,
39495	18.561016	192.168.1.141	192.168.1.101	TCP	66	49582 → 80 [FIN,
39496	18.561132	192.168.1.101	192.168.1.141	TCP	66	80 → 49582 [ACK]
39497	18.586232	192.168.1.1	239.255.255.250	SSDP	216	M-SEARCH * HTTP/
39498	18.603854	192.168.1.141	192.168.1.102	QUIC	67	Protected Payload
39499	19.021753	192.168.1.141	192.168.1.102	VRRP	60	Announcement (v2)
39500	19.403327	192.168.1.141	192.168.1.102	TCP	74	39634 → 80 [SYN]
39501	19.403449	192.168.1.102	192.168.1.141	TCP	74	80 → 39634 [SYN,
39502	19.403608	192.168.1.141	192.168.1.102	HTTP	104	HEAD / HTTP/1.0
39503	19.403693	192.168.1.102	192.168.1.141	TCP	66	80 → 39634 [ACK]

\* Slave Proxy Server : 192.168.1.141

\* WEB Server\_1 : 192.168.1.101

192.168.1.141	192.168.1.101	HTTP
192.168.1.101	192.168.1.141	TCP

▶ Client에서요청시 Proxy Server에서 WEB Server\_1 로 요청을 전달하고 응답 받는다.

\* WEB Server\_2 : 192.168.1.102

192.168.1.141	192.168.1.102	TCP
192.168.1.102	192.168.1.141	TCP
192.168.1.141	192.168.1.102	HTTP

▶ Proxy Server에서 트래픽이 Load Balancing 되어 이번에는 WEB Server\_2로 요청을 전달하고 응답받는다.



## HAProxy 의 SSL/TLS 암호화 검증



Client 는 서버와 직접 통신하는 것이 아니라 HAProxy를 통해 요청을 전달하고 응답받는다. HAProxy를 단일 엔드 포인트로 구성하면 외부에서 들어오는 모든 데이터가 HAProxy를 거치게 된다. 따라서 HAProxy 의 SSL/TLS 설정을 통하여 전송되는 모든 데이터를 암호화되어 통신되는 것을 검증한다.

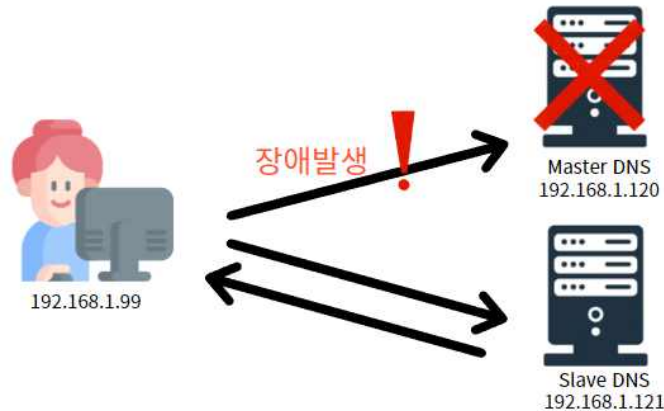
```
<VirtualHost *:80>  
  Redirect "/" https://www.buddy.com"  
</VirtualHost>
```

- ▶ HAProxy의 /etc/httpd/conf/httpd.conf 환경 설정 파일에서 Client가 HTTP 프로토콜로 접근했을 때 HTTPS 로 Redirection 하도록 설정합니다.



- ▶ Client에서 http://www.buddy.com 으로 접근했을 때, HTTPS 로 Redirection 하는 것을 확인할 수 있다. 이때, HTTPS 인증서 발급 기관에서 발급하는 공인 인증서가 아닌 자체적으로 발급한 사설 인증서를 사용하여 신뢰할 수 있는 인증서인지 확인하는 과정을 거치게 된다.  
따라서 Client 는 HTTP 요청시 단일 End Point 로 구성한 HAProxy 로 접근하기 때문에 HTTPS 로 Redirection 되어 서버 간의 통신을 안전하게 보호할 수 있다.

## DNS Server 의 이중화 동작 검증



Client에서 www.buddy.com 으로 접속시 Master DNS에서 Domain Name 과 IP를 Mapping 하여 IP로 변환하고, IP에 따른 WEB Server 로 포워딩한다. 따라서 DNS Server 가 정상적으로 작동하지 않으면 고객들은 Domain Name을 이용하여 WEB Server 에 접속할 수 없게 된다.

따라서 DNS Server 이중화 설정을 통하여 Master DNS에 장애가 발생하면 Slave DNS 가 그 역할을 대신하여고가용성을 제공하는지 검증한다.

### [ Client → DNS Server 1 ]

Internet Protocol Version 4, Src: 192.168.1.99, Dst: 192.168.1.120

User Datagram Protocol, Src Port: 56339, Dst Port: 53

Source Port: 56339

Destination Port: 53

Domain Name System (query)

Queries

www.buddy.com: type A, class IN

Name: www.buddy.com

### [ DNS Server 1 → Client ]

Internet Protocol Version 4, Src: 192.168.1.120, Dst: 192.168.1.99

User Datagram Protocol, Src Port: 53, Dst Port: 56339

Source Port: 53

Destination Port: 56339

Domain Name System (response)

Queries

www.buddy.com: type A, class IN

Name: www.buddy.com

[Name Length: 13]

[Label Count: 3]

Type: A (Host Address) (1)

Class: IN (0x0001)

▶ Client에서 접속 요청이 들어왔을 때, Master DNS Server 가 정상적으로 작동하는 경우, Master에서 Domain Name을 IP 로 변환하여 준다.

```
[root@dns01 ~]# init 0
```

```
Remote side unexpectedly closed network connection
```

```
Session stopped
```

- Press <Return> to exit tab
- Press R to restart session
- Press S to save terminal output to file

▶ Master DNS를 강제로 종료한 후, Client에서 다시 접속 요청을 시도한다.

#### [ Client → DNS Server 2 ]

```
Internet Protocol Version 4, Src: 192.168.1.99, Dst: 192.168.1.121
```

```
User Datagram Protocol, Src Port: 50373, Dst Port: 53
```

```
Source Port: 50373
```

```
Destination Port: 53
```

```
Domain Name System (query)
```

```
Queries
```

```
www.buddy.com: type A, class IN
```

```
Name: www.buddy.com
```

#### [ DNS Server 2 → Client ]

```
Internet Protocol Version 4, Src: 192.168.1.121, Dst: 192.168.1.99
```

```
User Datagram Protocol, Src Port: 53, Dst Port: 50373
```

```
Source Port: 53
```

```
Destination Port: 50373
```

```
Domain Name System (response)
```

```
Queries
```

```
blog.kgitbank.com: type A, class IN
```

```
Name: www.buddy.com
```

```
[Name Length: 13]
```

```
[Label Count: 3]
```

```
Type: A (Host Address) (1)
```

```
Class: IN (0x0001)
```

▶ Master DNS 가 동작하지 않자 Slave DNS에서 Client 의 요청을 처리하여 높은 가용성을 제공하는 것을 확인할 수 있다.

## 04 웹 페이지 구현

### 프로젝트 UI 설계 및 MTV Model

#### 1) 메인 페이지와 Detail 페이지

- stay.html : 웹 사이트의 메인 페이지로 사용자가 최초 접근 시 제공되는 페이지
- stay/detail\_1.html : 상세 커뮤니티 페이지 (베이킹)
- stay/detail\_2.html : 상세 커뮤니티 페이지 (소셜링)

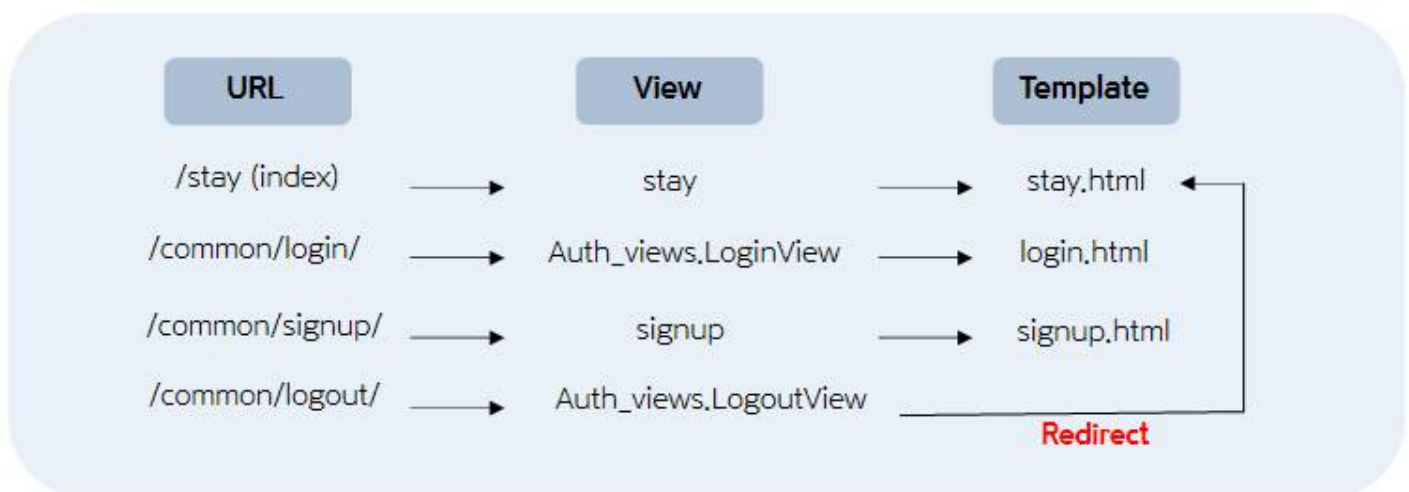
#### 2) 로그인 페이지

- login.html : 회원정보가 등록된 Client 가 로그인하는 페이지
- signup.html : 회원정보가 등록되지 않은 Client 가 회원가입을 수행하는 페이지

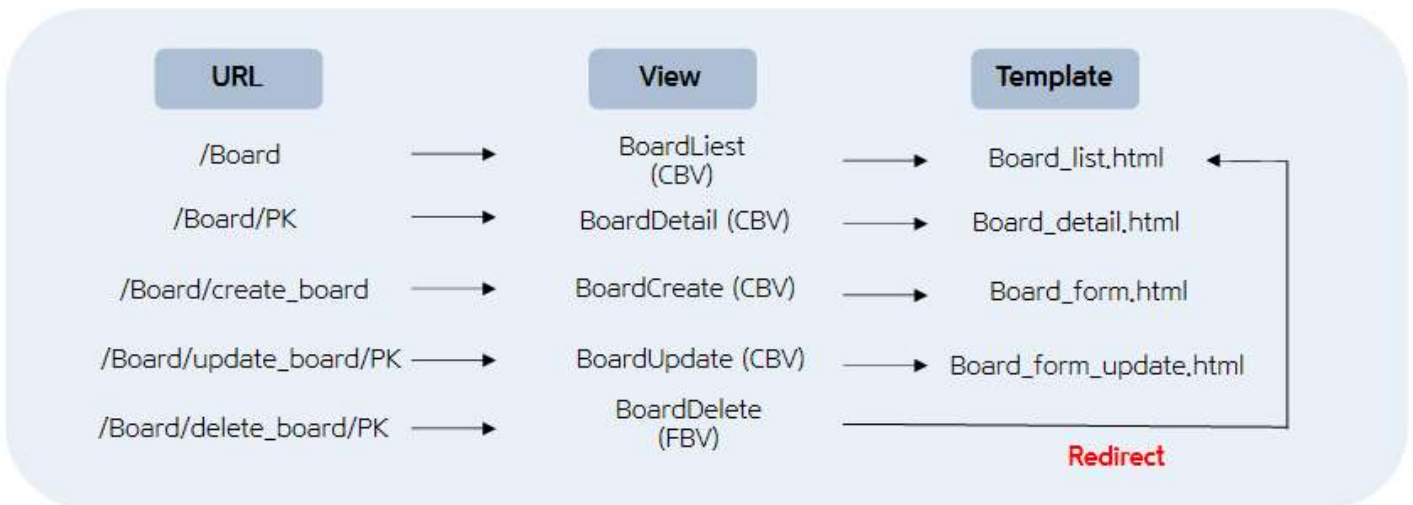
#### 3) 게시판 페이지

- board\_list.html : 전체 게시판 목록을 보여주는 페이지
- board\_detail.html : 게시글의 상세 내용을 출력하는 페이지
- board\_form.html : Client 가 새로운 게시글을 작성하는 페이지
- board\_form.update.html : Client 가 자신이 생성한 게시글을 수정하는 페이지

### [ 로그인 페이지 MTV Model ]



## [ 게시판 페이지 MTV Model ]



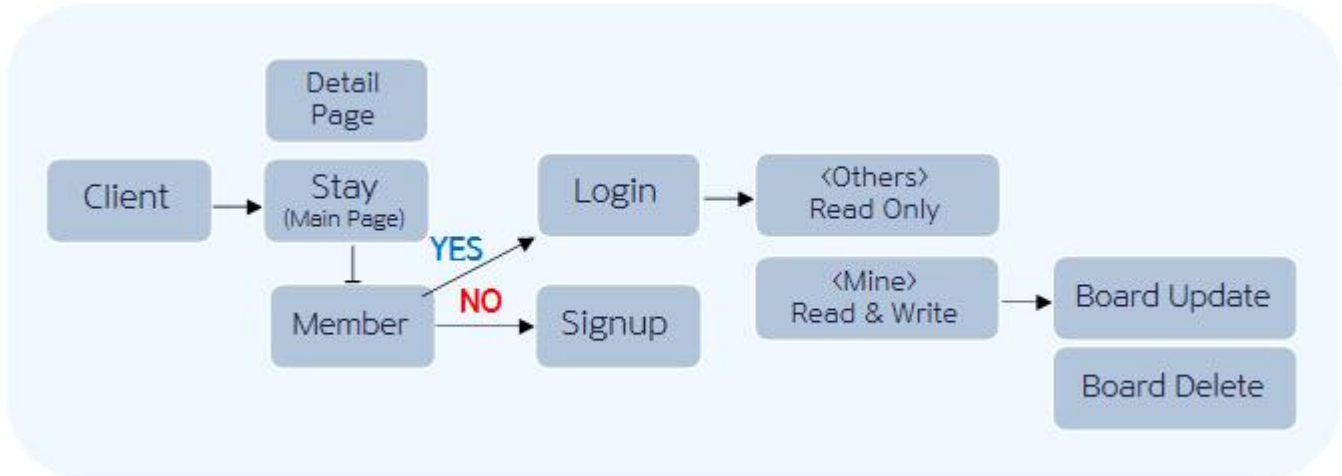
## URL & View & Template

URL	view	template
/stay	Stay Func	stay.html
/stay/detail_1	detail_1 Func	detail_1.html
/stay/detail_2	detail_2 Func	detail_2.html
/board/board_list	board Func	board.html
/board/board_detail	detail Func	detail.html
/board/board_create	create Func	board_form.html
/board/board_update	update Func	board_update.html
/common/login	login Func	login.html
/common/signup	signup Func	signup.html

## 기능별 앱 분리

- 전체 웹 사이트 구조에서 공통적인 기능을 수행하는 웹 사이트 기능의 경우 별도의 APP 을 만들어 기능을 구현한다.
- 현재 프로젝트에서는 메인페이지와 메인 페이지와 연결게임 플랫폼과 관련된 페이지는 "stay app" 으로 분리하고, 회원관리 기능은 "Common app" 으로 기능을 구현하고, 게시판 기능은 "Board app" 으로 기능을 구현한다.

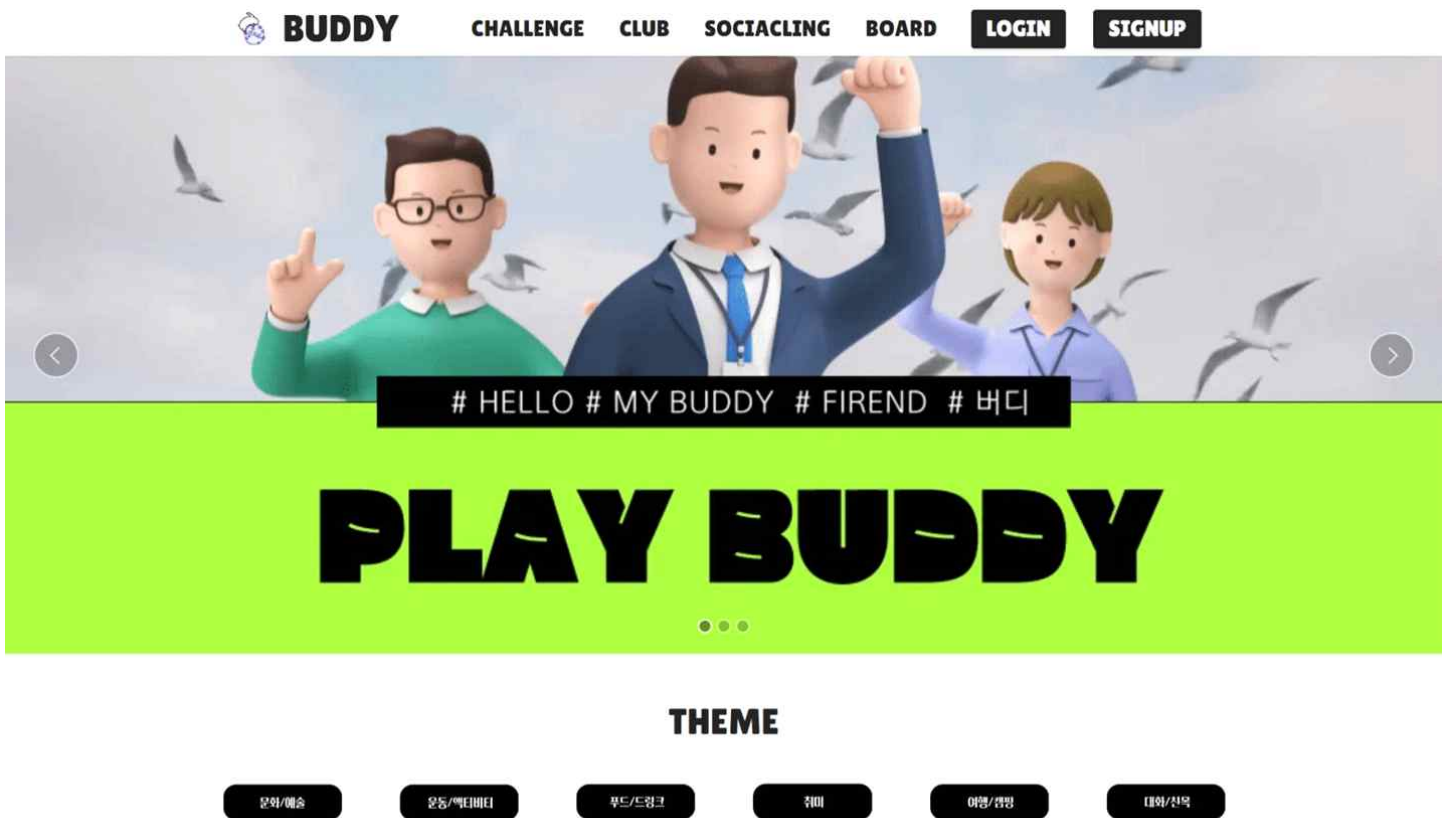
## 웹 사이트 기능의 동작 흐름



모든 사용자는 메인 페이지와 세부 커뮤니티에 대한 페이지에 접근이 가능하다. 로그인 여부에 따라 게시판 기능 사용에 제한을 두어 로그인한 사용자만 게시판 작성 및 상세 게시글 확인이 가능하다. 다른 사용자의 작성 글은 읽기만 가능하며 본인이 작성한 게시글은 수정 수정 및 삭제가 가능하다.


## 웹 페이지 구성

### 메인 페이지





# BIZDANG BAKING




## MUFFIN

원데이 클래스

베이킹 원데이 클래스 공방입니다 :)  
성수역 2번 출구  
자세한 사항은 인스타 그림이나  
블로그로 확인해주세요 ㅇ.<

[SHOW NOW >](#)




## CAKE

주문 제작

모든 재료를 직접 만듭니다  
모든 예약/문의는 '카톡 채널'로 해  
주세요.)

[SHOW NOW >](#)




## COFFEE

체험

예약전화 : XX-XXXX-XXXX.  
생두를 로스팅하고, 로스팅한 원두  
를 그라인딩한 후 핸드드립하여  
직접 만드는 '나만의 커피'

[SHOW NOW >](#)




## SCONE

원데이 클래스

원데이 클래스 공방입니다 :)  
홍대역 2번 출구  
자세한 사항은 인스타 그림이나  
블로그로 확인해주세요 ㅇ.

[SHOW NOW >](#)




## MACARON

원데이 클래스

마카롱 원데이 클래스임  
강남역 8번 출구  
자세한 사항은 인스타 그  
블로그로 확인해주세요

[SHOW NOW >](#)

## HOBBY




Sports

### CLIMBING

실내 클라이밍 모임  
경험이 없는 초보자도 할 수 있으니 부담  
갖지말고 문의해주세요.

[SHOW NOW >](#)




Sports

### RUNNING

런닝 & 마라톤 모임 재력이 없어서 걱정  
이시나요? 걱정하지말고 모임에 참여  
하세요! 같이 운동해요 :)

[SHOW NOW >](#)




Sports

### RUNNING

초보자분들은 힘들 수 있는 모임입니다.  
마라톤이나 런닝 경험이 있으신 분들이  
지원하시면 적합할거 같아요 :)

[SHOW NOW >](#)




Sports

### BILLIARDS


초보자분들도 환영합니다! 부담 갖지 말  
고 오셔서 저희랑 게임해요!

[SHOW NOW >](#)



자연속  
람들의

## MUSIC




## DANCE

참실

.....

[SHOW NOW >](#)




## GUITAR

경원도 순천

.....

[SHOW NOW >](#)




## SONG

대전

.....

[SHOW NOW >](#)




## PIANO

여의도

.....

[SHOW NOW >](#)

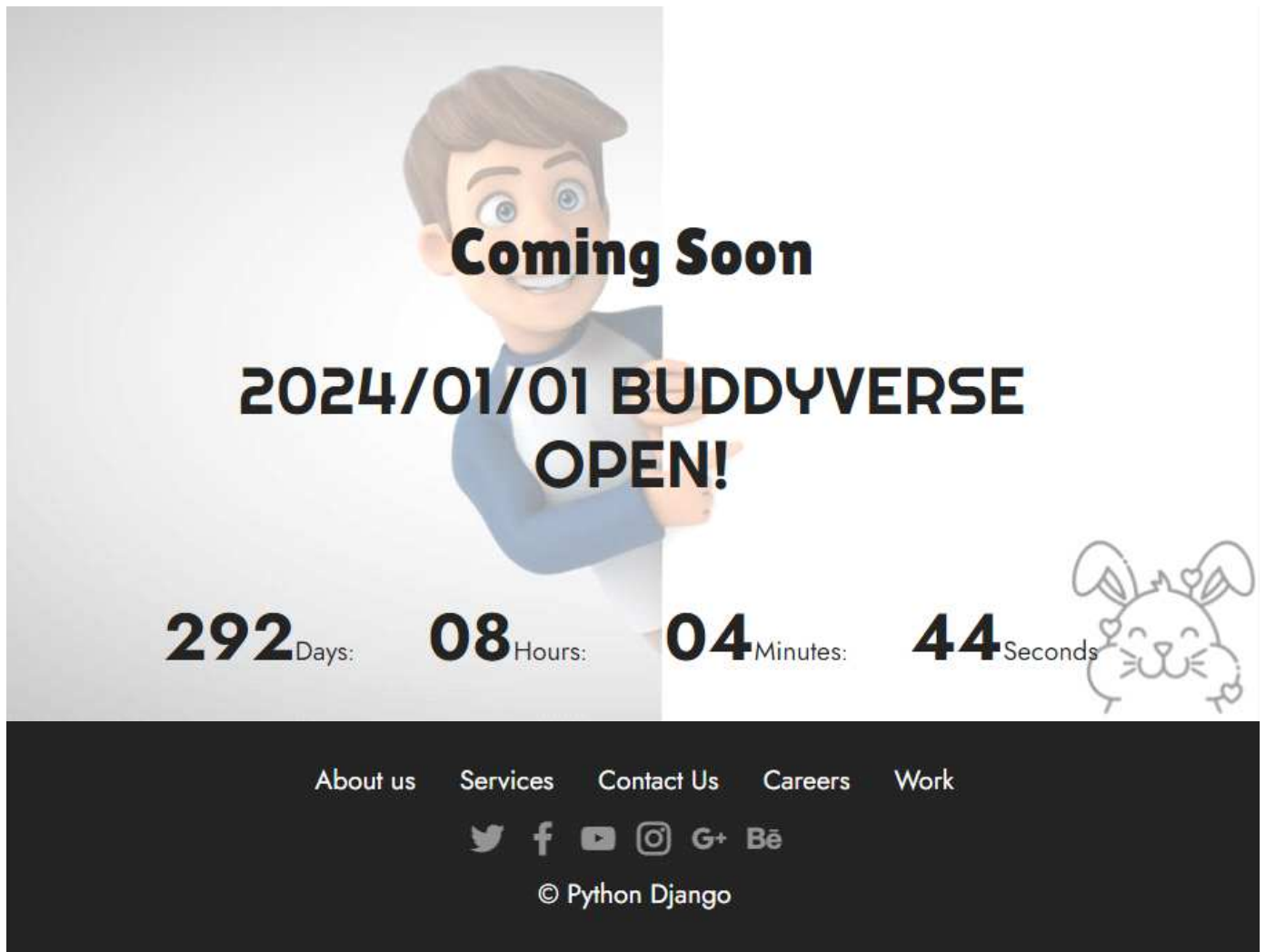


## FESTIVAL

충청도 태안

.....

[SHOW NOW >](#)



**Coming Soon**



**2024/01/01 BUDDYVERSE OPEN!**

**292** Days: **08** Hours: **04** Minutes: **44** Seconds

About us Services Contact Us Careers Work

Twitter Facebook YouTube Instagram G+ Bē

© Python Django

## 💡 서브 페이지 ( 베이킹 )





## 누구나 쉽게 할 수 있는 Baking oneday class

#원데이클래스 #베이킹 #서울



### NOTICE



#### [ 환불 규정 ]

- 클래스기준 3일전 부터 환불 불가
- 예약 후 30분 이내 취소 가능



#### [ 예약변경/품목변경 및 취소 ]

- 4일전 까지 변경 가능
- [ 품목도 변경 가능합니다 :) ]
- 변경은 1번만 가능합니다.



#### [ 무단결석/지각 및 노쇼 ]

- 15분 이상의 지각은 당일취소 됩니다.
- 15분 이상 지각시 환불 x > 레시피만 제공



#### [ 애완동물 동반 ]

- 애완동물 동반은 불가능 합니다.

## REVIEW

사장님이 정말 친절하고 자세히 잘 알려주셨어요! 내부 인테리어도 되게 이쁘고 당연히 맛도 있었습니다! :)



Smith  
Client

사장님이 정말 친절하고 자세히 잘 알려주셨어요! 다음에 또 한번 방문하고 싶어요 :).



Jessica  
Client

## MAP



## 💡 서브 페이지 ( 소셜링 )

# SOCALRING

## 소셜링

### PROFILE

똑같은 일상을 다채롭게  
만들어 줄 원데이 취향모임,  
가볍고 즐거운 모임은 BUDDY

### SPECIAL

문화/예술

푸드/드링크

친목/대화

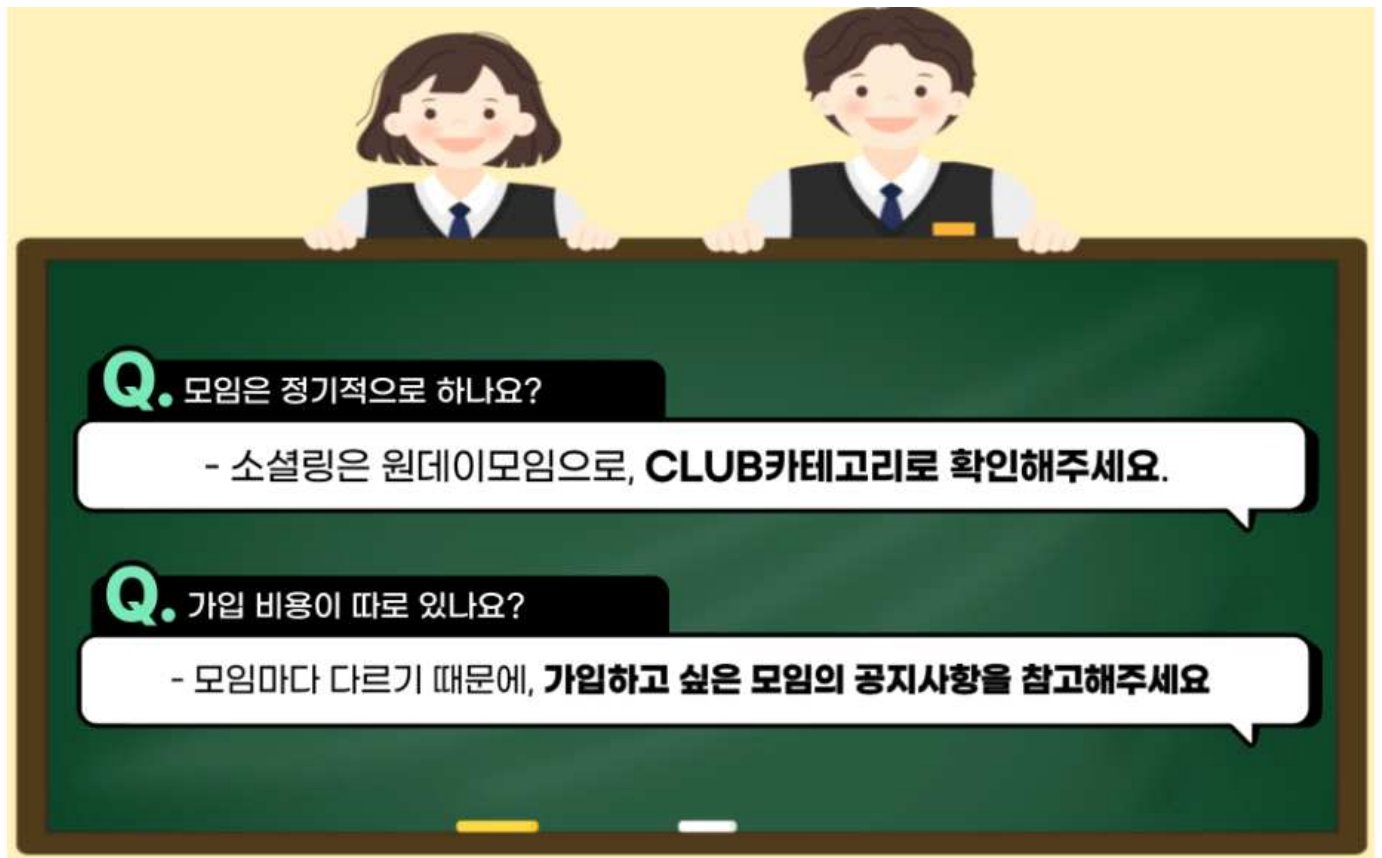
취미

### SKILL

- SPEED
- ATTACK
- SPECIAL
- ACTIVITY
- PLAN

### MBTI





## 로그인 & 회원가입 페이지

[ 로그인 ]

**STAY**

Enter User ID

Enter User Password

LOGIN

[ 회원가입 ]

**SIGN UP**

One of our team will be in contact with you shortly.

ID

Password

Re-Entry Password

E-mail Address

Submit

## Board 목록 페이지

[CHALLENGE](#)[CLUB](#)[SOCIACLING](#)[BOARD](#)[LOGIN](#)[SIGNUP](#)[리뷰 쓰기 가기>](#)

번호	제목	작성자	작성일
4	test3	admin	2023년 2월 27일 2:29 오후
3	test3	test2	2023년 2월 27일 12:20 오후
2	review2	admin	2023년 2월 25일 1:20 오후
1	review1	admin	2023년 2월 25일 1:19 오후

## Board 작성 페이지

제목 입력 test 중입니다

파일 선택 선택된 파일 없음

test 파일입니다

게시글 등록

## Board 세부 페이지

test3

Created by admin on 2023년 2월 27일 2:29 오후

첨부파일 :

test

목록

수정

삭제



## 팝업 페이지

[ 로그인하지 않은 사용자가 Board를 작성할 경우 ]

127.0.0.1:8000 내용:

로그인 후 이용이 가능합니다.

확인

[ 다른 사용자가 작성한 글을 삭제할 경우 ]

127.0.0.1:8000 내용:

자신의 게시글만 수정 및 삭제가 가능합니다.

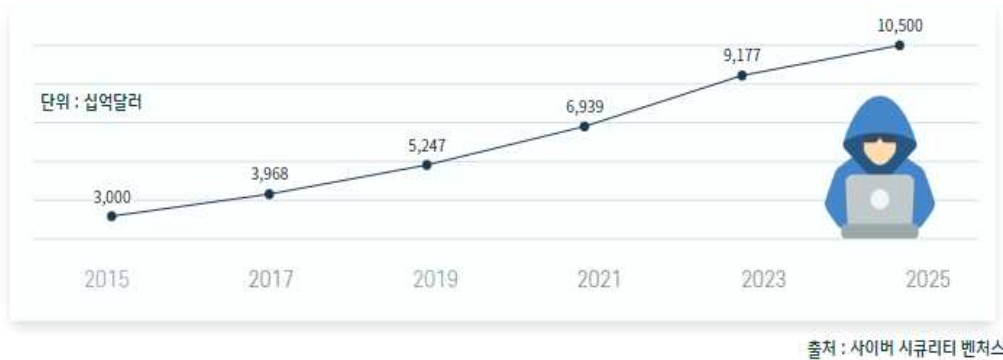
확인

## 05 보안 점검 Shell Script

보안이란 하드웨어 또는 소프트웨어적으로 허가되지 않은 액세스 또는 작업을 방지하여 외부의 위협으로부터 네트워크 데이터와 인프라를 보호하는 것이다. 최근 디지털/비대면 시장의 급격한 성장으로 모든 산업분야에서 정보보호가 내재화되어 정보 보호분야의 중요성이 부각되고 있다.

특히, 서버 보안은 고객 개인정보뿐만 아니라 기업의 주요 정보자산이 파괴되거나 유출하여 치명적인 결과를 초래할 수 있으므로 지속적인 점검을 통하여 사이버 공격으로 부터의 보호가 필요하다. 따라서 서버 보안 솔루션을 통하여 서버 운영체제의 취약점을 보완하고 외부의 악의적인 공격을 사전에 탐지하고 방어하기 위한 기술이 필요하다.

### 글로벌 사이버 범죄 피해 규모



보안은 안전할 때 유지·관리가 필요하다. 지속적인 관리를 통하여 사이버 공격을 당했을 때, 피해 규모를 축소시킬 수 있으며 공격의 위협으로부터 시기를 늦출 수 있다. 그러나 보안 위협에 대응하기 위한 전문 인력의 부족과 비용, 기업 자체적으로 무엇을 어떻게 구체적으로 점검해야 할지, 점검 상태에 따라 어떤 조치를 취해야 할지를 파악하는 것이 어려우므로 기업 자체적으로 사이버 공격에 대처하기에는 한계가 있다.

따라서 네트워크 서버 구축 이후 보안 점검 Shell Script를 배포하고 하위 보안 점검 체크 리스트의 점검 항목에 대하여 각 서버마다 보안 점검을 실시한다. Shell Script 로 작성하여 자동화하여 주기적으로 점검이 가능하다. 점검에 따라 취약 항목의 경우 회사의 보안 지침 상황에 따라 조치를 취하여 보안성을 향상시킨다.

### 보안 스크립트의 필요성



- 정기적인 보안 스크립트를 활용한 지속적인 헬스 체크를 통하여 불법적인 개인 정보 거래 및 기업 내부 정보 거래에 대한 위협을 예방하고 안전한 IT 환경 유지
- 보안 전문 시스템 및 전문 인력 확보에 대한 유지·관리의 부담을 최소화
- 보안 시스템 및 인력 관리에 대한 비용 절감과 보안 위협에 대한 금전적 사후 보상에 대한 문제 예방
- 사용자 정보 유출 방지를 통하여 기업에 대한 신뢰도가 상승

## 보안 점검 체크리스트

No	분류	점검항목	점검결과	조치여부
1	계정 관리	group 파일 권한 설정		
2		root 계정 원격 접속 제한		
3		계정 잠금 임계값 설정		
4		Session Timeout 설정		
5		root 계정 su 제한		
6		패스워드 복잡성 및 최소길이 설정		
7		패스워드 최대 · 최소 사용기간 설정		
8		패스워드 파일 보호		
9		관리자 그룹에 최소한의 계정 포함		
10		사용자 shell 점검		
11		root 이외의 UID가 '0'금지		
12	파일 및 디렉토리 관리	/etc/hosts 파일 소유자 및 권한 설정		
13		/etc/passwd 파일 소유자 및 권한 설정		
14		/etc/shadow 파일 소유자 및 권한 설정		
15		/etc/rsyslog.conf 파일 소유자 및 권한 설정		
16		파일 및 디렉토리 소유자 설정		
17		hosts.equiv 파일 권한 설정		
18		root 홈, 패스 디렉터리 권한 및 패스 설정		
19		UMASK 설정		
20		/etc/(x)inetd.conf 파일 소유자 및 권한 설정		
21	서비스 관리	crond 파일 소유자 및 권한 설정		
22		DNS 보안 버전 패치		
23		NFS 접근통제		
24		NFS 서비스 비활성화		
25	로그 관리	로그 의 정기적 검토 및 보고		
26	패치 관리	최신 보안패치 및 벤더 권고사항 적용		



## Ansible을 이용한 보안 Shell Script 배포

### Ansible이란?



ANSIBLE

Ansible은 여러 개의 서버를 효율적으로 관리할 수 있게 해주는 환경 구성 자동화 툴 중 하나로, 대규모 서버 및 인프라 환경에서 사용되는 배포 및 관리 도구이다.

Ansible은 SSH를 통해 서버에 명령을 전달하고, 코드를 사용하여 인프라를 설치, 구성, 배포, 관리 및 오케스트레이션을 할 수 있다.

### Ansible의 사용 목적

#### 1) 자동화

서버와 네트워크, 보안 등 IT 인프라를 효율적으로 관리할 수 있으며 코드로 인프라를 관리하므로 반복적인 작업이 줄어들고, 인프라의 일관성과 안정성을 높일 수 있다.

#### 2) 일관성

인프라 전체에 일관성 있는 스크립트를 배포하고, 이를 통해 모든 서버에 일관성 있는 보안 정책을 적용할 수 있다.

#### 3) 중앙 집중화

Ansible은 중앙 집중화된 인프라 관리 도구로서, 모든 호스트를 한 번에 관리할 수 있다. 이를 통해 인프라 관리자는 호스트별로 보안 스크립트를 배포하는 복잡한 과정을 간단하게 만들 수 있습니다.

#### 4) 오류 최소화

보안 스크립트를 여러 번 실행하더라도, 결과가 항상 동일하게 유지되므로 보안 스크립트를 반복적으로 배포하는 과정에서 발생할 수 있는 오류를 최소화할 수 있다.

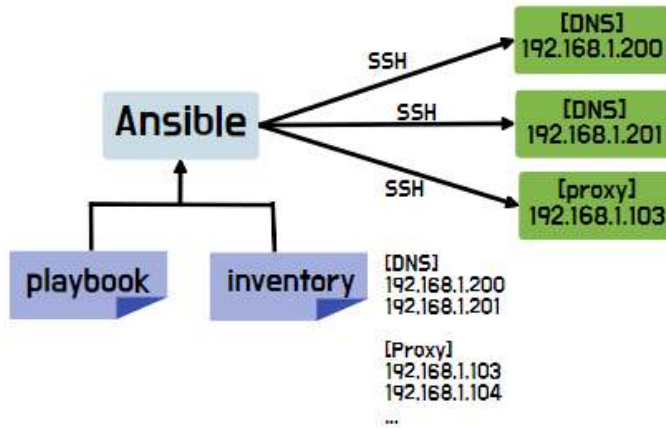
#### 5) 보안성

Ansible은 SSH를 사용하여 인증 및 암호화된 통신을 사용하기 때문에 안전한 방식으로 호스트에 원격 접속할 수 있다.

#### 참고 | Ansible 플레이북이란?

인프라 관리를 위한 Ansible의 핵심 기능 중 하나이다. Ansible 플레이북은 YAML 형식으로 작성되며, Ansible을 사용하여 구성 관리, 배포, 프로비저닝 및 애플리케이션 관리를 자동화하는데 사용된다.

## Ansible를 이용한 보안 Script 배포 검증



```
[root@manage ~]# ansible-playbook -i /etc/ansible/hosts playbook_test1.yml -k
SSH password:
```

```
PLAY [Deploy and run test script] *****
```

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.1.101]
ok: [192.168.1.100]
ok: [127.0.0.1]
ok: [192.168.1.103]
ok: [192.168.1.104]
..... 생략
```

```
TASK [Copy test script to remote host]
```

```
*****
```

```
changed: [192.168.1.101]
changed: [192.168.1.106]
changed: [192.168.1.100]
changed: [192.168.1.105]
ok: [127.0.0.1]
changed: [192.168.1.103]
..... 생략
```

```
PLAY RECAP *****
```

127.0.0.1	: ok=4	changed=1	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.100	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.101	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.103	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.104	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.105	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.106	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.107	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.104	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.105	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.106	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0

192.168.1.107	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.108	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.104	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.105	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.106	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.107	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.108	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.200	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	ignored=0
192.168.1.201	: ok=4	changed=2	unreachable=0	failed=0	skipped=0	rescued=0	"ignored=0

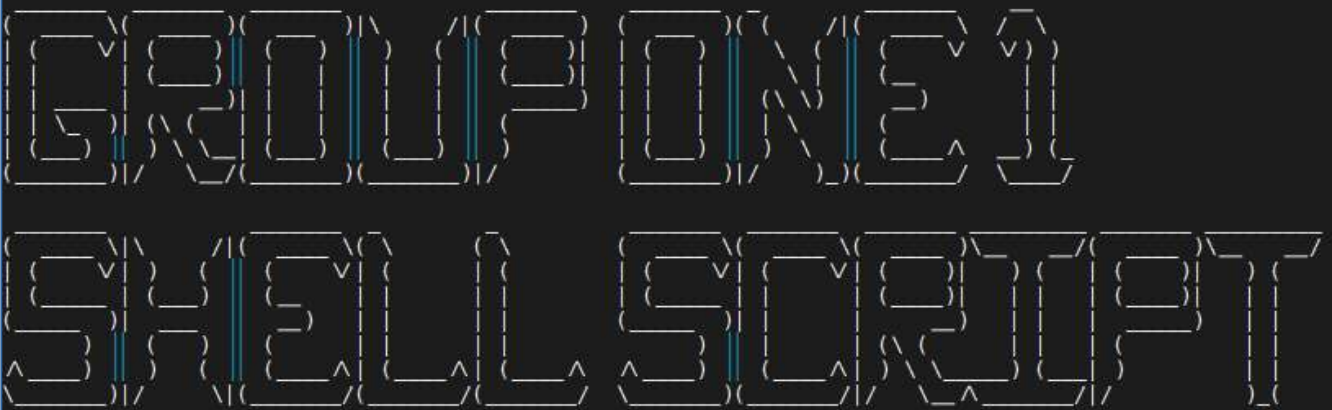
- ▶ Test Server 에 Ansible 을 설치하고, playbook 과 inventory 설정 내용을 바탕으로 각 서버에 보안 스크립트를 배포하여 실행한다.



- ▶ 스크립트를 배포한 각 서버에서 스크립트가 정상적으로 동작함을 확인한다.. 보안 스크립트의 항목 중 log 파일을 백업하는 항목에 대한 스크립트가 동작하여 각 서버에 log 파일이 백업된 것을 확인할 수 있다.



- ▶ 점검 결과를 전달받은 이메일로 각 서버별 점검 내용을 확인할 수 있다. 이메일을 이용한 점검 결과 보고를 통하여 장소에 구애받지 않고 결과 확인이 가능하다.



2023. 02. 16. (목) 15:08:34 KST

# 보안점검보고서

항목에 따라 시간이 다른 항목에 비하여 다소 오래 걸릴수 있습니다  
점검 보고서는security.txt 파일로 저장됩니다  
점검기준은 [기술적 취약점 분석.평가 방법 상세 가이드]입니다.

## 시스템 정보

- ```
▶ IP 주소      : 192.168.1.129
▶ 운영체제     : CentOS Linux release 7.9.2009 (Core)
▶ 커널 버전    : 3.10.0-1160.81.1.el7.x86_64
▶ 사용자 명    : root
```

## 점 검 결 과

## ◆ 계정관리 : group 파일 권한 설정 ◆

- ▶ 양 호 : 파일 소유자 root, 권한이 644 이하
- ▶ 취 약 : 파일 소유자 root, 권한이 644 이하

◆ 계정관리 : root 계정 원격 접속 제한 ◆

- ▶ 양 호 : root계정 SSH 차단
- ▶ 취 약 : root계정 SSH 허용

※ root의 원격접속을 차단해주시오.

◆ 계정관리 : 계정 잠금 임계값 설정 ◆

- ▶ 양 호 : 계정 잠금 임계값이 5회 이하  
▶ 취 약 : 계정 잠금 임계값이 설정되어 있지 않거나, 5회 이하의 값이 아님

※ 계정 잠금 임계값이 설정되어 있지 않습니다.

※ 계정 잠금 임계값을 설정하여 주십시오.

## ◆ 계정관리 : Session Timeout 설정 ◆

- ▶ 양 호 : Session Timeout이 100초 (10분) 이하
- ▶ 취 약 : Session Timeout이 100초 (10분)가 아님

※ export TMOUT 내용이 추가 되어있지않습니다.

※ 아래 내용을 추가해주시오. (단,100초이하)

```
export TMOUT=300
```

/etc/profile 파일 없음

◆ 계정관리 : root 계정 su 제한 ◆

- ▶ 양 호 : su 명령어를 특정 그룹에 속한 사용자만 사용하도록 제한
- ▶ 취 약 : su 명령어를 모든 사용자가 사용하도록 설정

※ 아래의 내용이 추가가 되어있지 않습니다.

※ 아래 내용을 추가해주시오.

|      |          |              |         |
|------|----------|--------------|---------|
| auth | required | pam_wheel.so | use_uid |
|------|----------|--------------|---------|

auth sufficient pam\_wheel.so trust use\_uid

/etc/pam.d/su 파일 없음

◆ 계정관리 : 패스워스 복잡성 설정 및 최소길이 설정 ◆



◆ 계정관리 : 관리자 그룹에 최소한의 계정 포함 ◆

- ▶ 양 호 : 관리자 그룹에 불필요한 계정이 등록되어 있지 않음  
▶ 취 약 : 관리자 그룹에 불필요한 계정이 등록되어 있음

[illegible]

◆ 계정관리 : 사용자 shell 점검 ◆

- ▶ 양 호 : 로그인 가능한 서비스 계정이 존재하지 않음  
▶ 취 약 : 로그인 가능한 서비스 계정이 존재함

[illegible]

◆ 계정관리 : root 이외의 UID가 '0'금지 ◆

- ▶ 양 호 : root 이외의 UID가 0인 사용자 존재하지 않음
- ▶ 취 약 : root 이외의 UID가 0인 사용자 존재함

[illegible]

◆ 파일 및 디렉터리 관리 : /etc/hosts 파일 소유자 및 권한 설정 ◆

- ▶ 양 호 : 파일 권한이 644 및 파일 소유자가 root
- ▶ 추 약 : 파일 권한이 644가 아니거나 소유자가 root가 아님

[illegible]

◆ 파일 및 디렉터리 관리 : /etc/passwd 파일 소유자 및 권한 설정 ◆

- ▶ 양 호 : 파일 권한이 644 및 파일 소유자가 root
- ▶ 취 약 : 파일 권한이 644가 아니거나 소유자가 root가 아님

[illegible]

- ▶ 양 호 : 파일 소유자가 root 및 파일 권한이 400
- ▶ 취 약 : 파일 소유자가 root가 아니거나 파일 권한이 400이 아님

※ 파일 권한을 400으로 설정하십시오.

- ▶ 양 호 : /etc/rsyslog.conf의 파일 권한이 644 및 소유주가 root로 설정
- ▶ 취 약 : /etc/rsyslog.conf의 파일 권한이 644가 아니거나 소유자가 root가 아님

▶ 양 호 : 소유주, 소유권 없는 파일.디렉터리가 없음  
▶ 취 약 : 소유주, 소유권 없는 파일.디렉터리가 존재

- ▶ 양 호 : hosts.equiv 파일이 존재하지 않거나 권한이 400으로 설정
- ▶ 취 약 : hosts.equiv 파일이 존재하며 권한이 400으로 설정되지 않음

- ▶ 양 호 : PATH 환경변수의 중간 혹은 앞에 .가 포함
- ▶ 취 약 : PATH 환경변수의 중간 혹은 앞에 .가 포함되어 있지 않음



- 55 -

[illegible][illegible]

- 56 -

잠시 기다려주세요 ..

◆ 점검 결과 보고서 이메일 전송 ◆

>>>>>>>>>>>>>>>> 메 일 이 전 송 되 었 습 니 다 <<<<<<<<<<<<<<<<<<<

- wjscodus95@nate.com 로 점검 결과 보고서를 전송하였습니다.  
※ 메일 미전송시 스팸메일함을 확인하여 주십시오

## ◆ 점검 결과 EC2 업로드 ◆

>>>>>>>>>>>>>>>> E C 2 업 로 드 완 료 되 었 습 니 다 <<<<<<<<<<<<<<<<<

|                   |      |      |         |       |
|-------------------|------|------|---------|-------|
| Security_List.txt | 100% | 17KB | 1.5MB/s | 00:00 |
|-------------------|------|------|---------|-------|