

# 보건 제공 제단

## 단골집



**팀장 : 이태호**

**부팀장 : 김태현**

**팀원 : 김용문, 황승우, 이서진**

# 목차

## 1. 프로젝트 개요

## 2. 팀 소개

## 3. 시스템 구축 과정

- 1) 리눅스 환경 세팅
- 2) 네트워크 구축
- 3) 서비스 설정
- 4) 보안 설정
- 5) 전체 네트워크 인프라 구축

## 4. 문제점 및 문제 해결

## 5. 결론 및 느낀점

# 프로젝트 개요

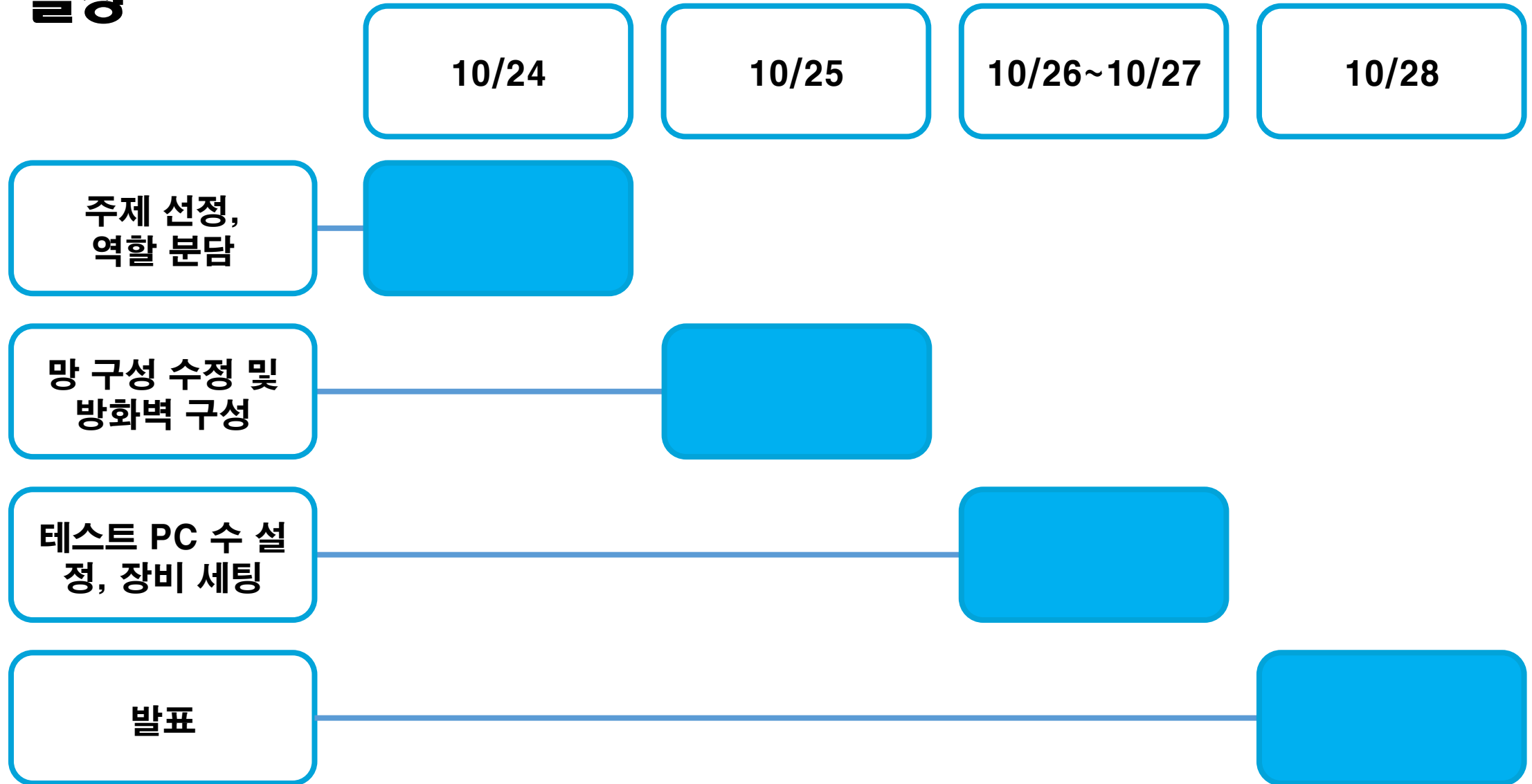
## 프로젝트 목적 : 병원 내 네트워크 및 인프라 구축

### 프로젝트 진행을 통해

- 안정적이고 보안성 높은 네트워크 인프라 구축
- 부서 간 효율적 통신
- 트래픽 관리 효율화와 시스템 확장성 확보
- 실무 역량 강화



## 일정



## 기대 효과

1. 안정적인 진료 서비스 제공
2. 보안성 강화 및 환자 정보 보호
3. 부서 간 효율적 자원 관리
4. 확장성과 유지 보수 성 확보
5. 실무 역량 파악

## 사용 장비

### 가상 머신



### 네트워크 에뮬레이터 도구



Cisco Packet Tracer



### 테스트 장비



wireshark



## 프로젝트 진행 흐름

Rocky Linux 설치,  
초기 설정  
사용자 계정/권한 관리

Packet Tracer, GNS3  
이용한 네트워크 설계,  
라우팅 구성

TFTP, DNS, HTTP  
서비스 제공 위한  
서버 구축

SELinux, 방화벽,  
접근제어 설정 및 점검

GNS3 연결 통해  
최종 인프라 구축



## 네트워크 구축 방향

재무인사, 진료과,  
병동, 관리/전산으로  
내부망 구축

외부망에  
HTTP, DNS 서버 구축

DMZ존 이용하여  
내부망과 외부망  
통신 관리

## 팀 소개



**이태호**  
(총괄 책임자)

전체 네트워크  
인프라 구축

**김태현**  
(서비스 운영자)

FTP/HTTP/DNS/  
TFTP 서버 구축 및  
권한 설정

**황승우**  
(시스템 관리자)

Rocky Linux  
설치 및 초기 설정  
사용자 계정/권한 관리

**김용문**  
(네트워크 엔지니어)

GNS3, Packet Tracer  
네트워크 설계 및  
라우팅 구성

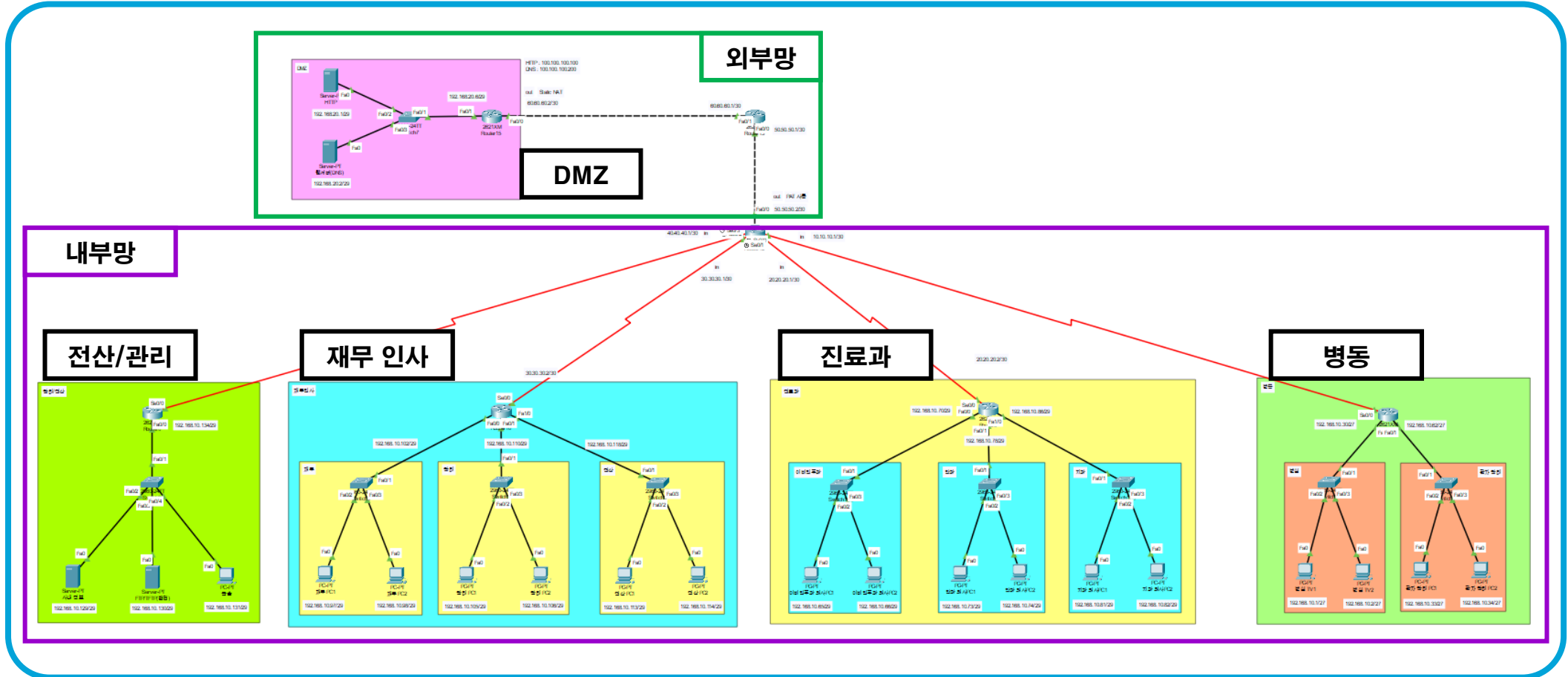
**이서진**  
(보안 담당자)

방화벽  
접근제어  
설정 및 점검

## 시스템 구축 과정

# 네트워크 구축

김용문



## • 전체 네트워크 구성

	호스트수	네트워크	브로드캐스트	서브넷마스크	사용가능IP
병동	64	192.168.10.0/26	192.168.10.63	255.255.255.192	192.168.10.1 ~ 192.168.10.62
진료과	32	192.168.10.64/27	192.168.10.95	255.255.255.224	192.168.10.65 ~ 192.168.10.94
재무인사	32	192.168.10.96/27	192.168.10.127	255.255.255.224	192.168.10.97 ~ 192.168.10.126
관리/전산	8	192.168.10.128/29	192.168.10.135	255.255.255.248	192.168.10.129 ~ 192.168.10.134
DMZ	8	192.168.20.0/29	192.168.20.7	255.255.255.248	192.168.20.1 ~ 192.168.20.6
병동	64	192.168.10.0/26	192.168.10.63	255.255.255.192	192.168.10.1 ~ 192.168.10.62

## • 병동/진료과 네트워크 구성

병동	호스트수	네트워크	브로드캐스트	서브넷마스크	사용가능IP
병실TV	32(실제 25)	192.168.10.0/27	192.168.10.31	255.255.255.224	192.168.10.1 ~ 192.168.10.30
간호사PC	32(실제 14)	192.168.10.32/27	192.168.10.63	255.255.255.224	192.168.10.33 ~ 192.168.10.62

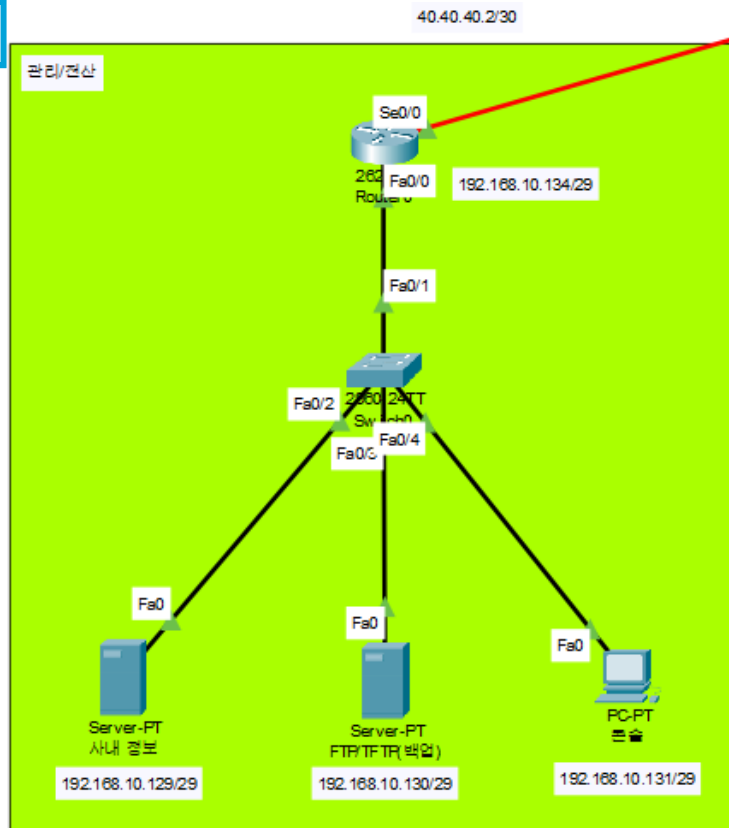
진료과	호스트수	네트워크	브로드캐스트	서브넷마스크	사용가능IP
이비인후과	8(실제 5명)	192.168.10.64/29	192.168.10.71	255.255.255.248	192.168.10.65 ~ 192.168.10.70
치과	8(실제 5명)	192.168.10.72/29	192.168.10.79	255.255.255.248	192.168.10.73 ~ 192.168.10.78
안과	8(실제 5명)	192.168.10.80/29	192.168.10.87	255.255.255.248	192.168.10.81 ~ 192.168.10.86



## • 재무인사 네트워크 구성

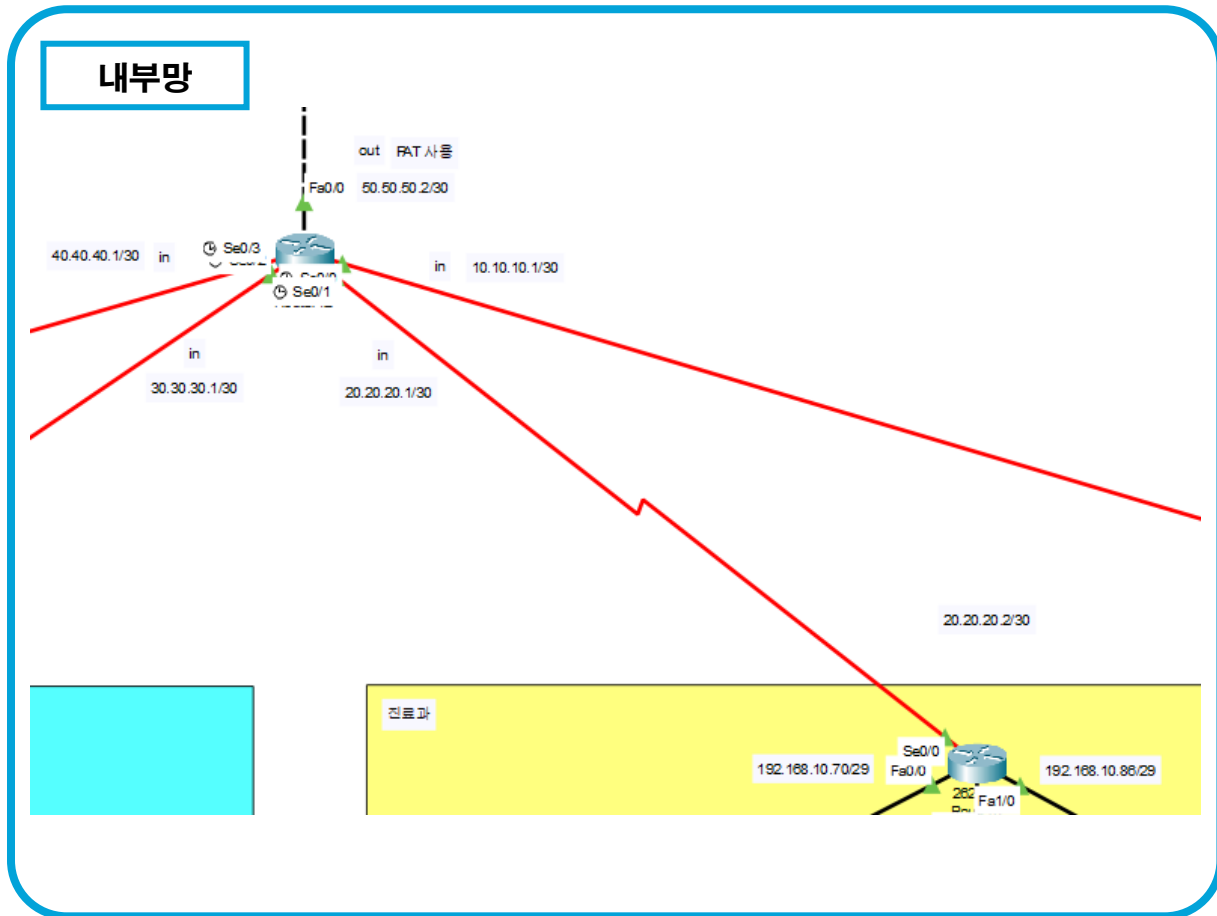
재무인사	호스트수	네트워크	브로드캐스트	서브넷마스크	사용가능IP
재무	8(실제 5명)	192.168.10.96/29	192.168.10.103	255.255.255.248	192.168.10.97 ~ 192.168.10.102
관리	8(실제 5명)	192.168.10.104/29	192.168.10.111	255.255.255.248	192.168.10.105 ~ 192.168.10.110
전산	8(실제 5명)	192.168.10.112/29	192.168.10.119	255.255.255.248	192.168.10.113 ~ 192.168.10.118

## 전산/관리



하나의 네트워크를  
VLSM을 이용하여  
IP가 낭비되는것을 방지

사내정보와 백업 서버를  
하나의 영역에 분리하여  
보안성을 강화



PAT방식을 사용하여  
여러 사용자가 있는  
내부망을  
하나의 공인 IP 사용해  
공인 IP를 절약

외부에서는  
내부의 사설 IP를  
알 수 없음을 이용하여  
보안성을 강화

**각각의 인터페이스의  
Inside와 outside 설정**

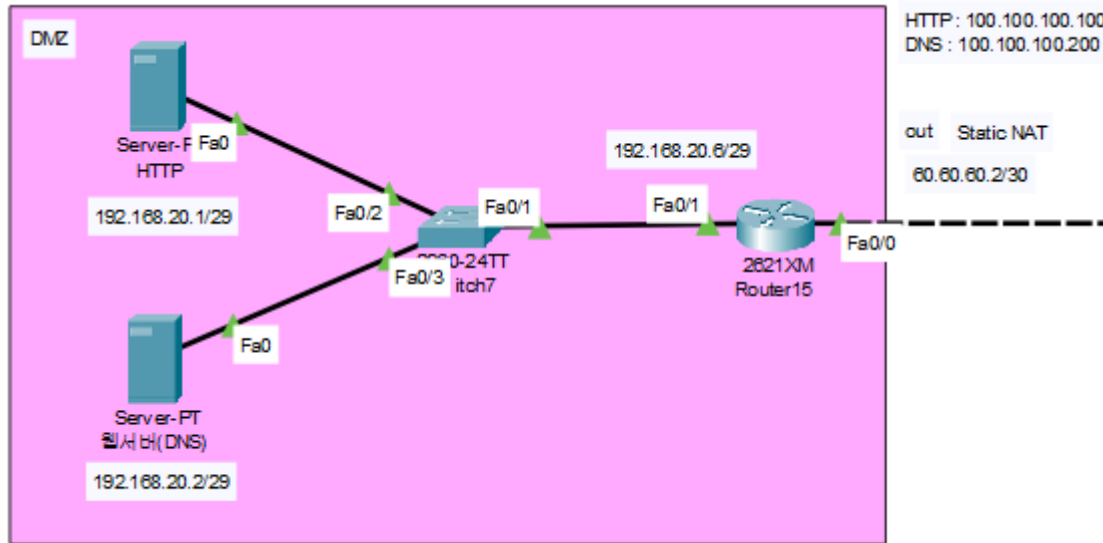
**공인IP와 매칭시킬  
사설IP 네트워크 대역대  
지정 후**

```
Router(config)#int f0/0
Router(config-if)#ip nat out
Router(config-if)#int s0/0
Router(config-if)#ip nat in
Router(config-if)#int s0/1
Router(config-if)#ip nat in
Router(config-if)#int s0/2
Router(config-if)#ip nat in
Router(config-if)#int s0/3
Router(config-if)#ip nat in
```

```
Router(config)#access-list 1 permit 192.168.10.0 0.0.0.63
Router(config)#access-list 1 permit 192.168.10.64 0.0.0.31
Router(config)#access-list 1 permit 192.168.10.96 0.0.0.31
Router(config)#access-list 1 permit 192.168.10.128 0.0.0.7
```

```
Router(config)#ip nat inside source list 1 int f0/0 overload
```

## DMZ



내부망과 분리된  
외부 공간에  
DMZ를 설정하여  
보안성을 강화

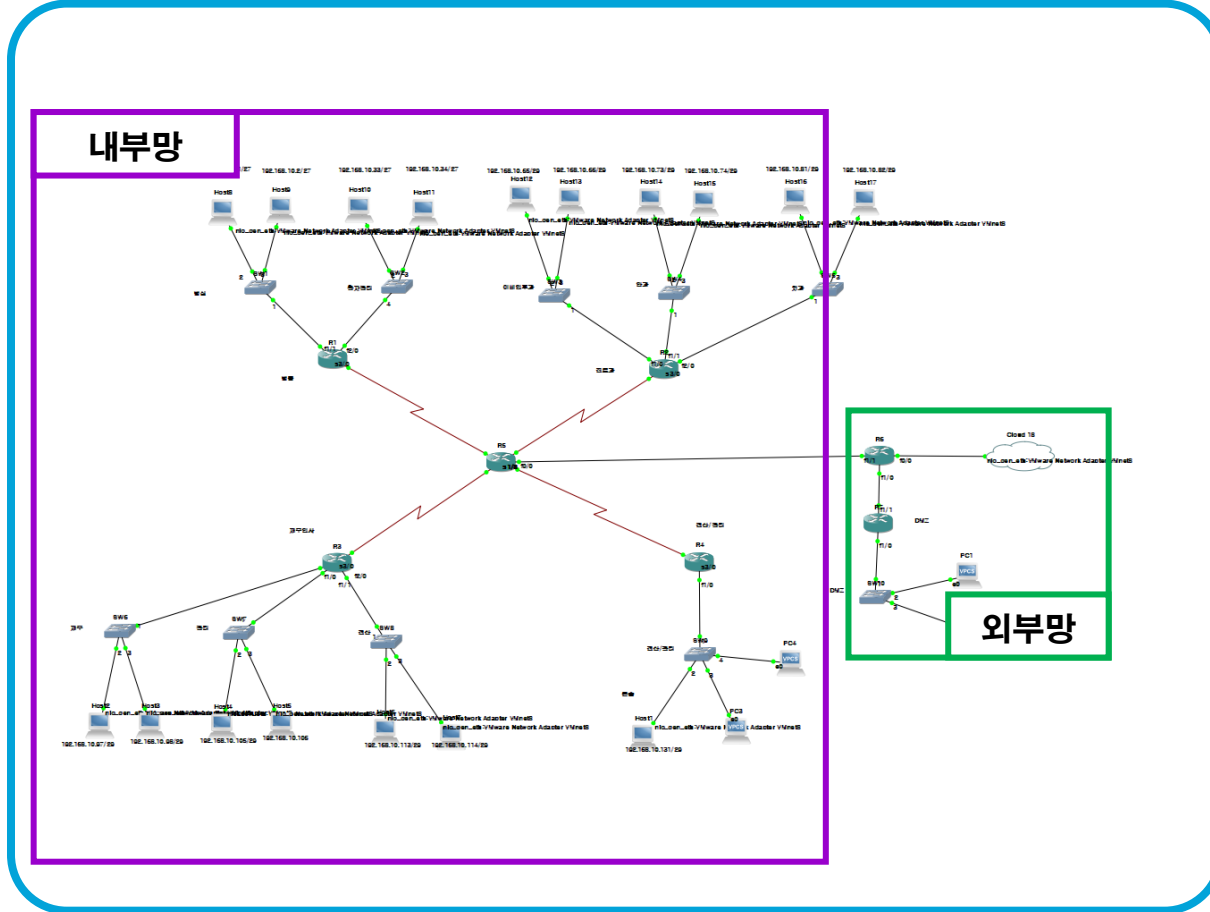
STATIC NAT를 설정해  
DNS와 WEB 서버 각각  
공인 IP와 사설 IP를  
1:1 매칭

**각각의 인터페이스의  
Inside와 outside 설정**

**공인IP와 매칭시킬  
사설IP를 각각  
1:1 매칭**

```
Router(config)#int f0/0  
Router(config-if)#ip nat out  
Router(config-if)#int f0/1  
Router(config-if)#ip nat in
```

```
Router(config)#ip nat inside source static 192.168.20.1 100.100.100.100  
Router(config)#ip nat inside source static 192.168.20.2 100.100.100.200
```



서비스 들을 취합하여  
전체 네트워크 인프라  
구축

# 문제점 및 문제 해결





**TCP 접근 거부 실패**

```
[root@localhost portentry-1.0]# vi /etc/hosts.deny  
[root@localhost portentry-1.0]# cat /etc/hosts.deny  
ALL: ALL  
[root@localhost portentry-1.0]#
```

```
root@kali: ~  
File Actions Edit View Help  
(root@kali)~  
# nmap -sT 192.168.10.131  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-10-27 20:43 KST  
Nmap scan report for 192.168.10.131  
Host is up (0.28s latency).  
Not shown: 982 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
1/tcp     open  tcpmux  
21/tcp    open  ftp  
22/tcp    open  ssh  
79/tcp    open  finger  
80/tcp    open  http  
111/tcp   open  rpcbind  
119/tcp   open  nntp  
143/tcp   open  imap  
1080/tcp  open  socks  
1524/tcp  open  ingreslock  
2000/tcp  open  cisco-sccp  
6667/tcp  open  irc  
12345/tcp open  netbus  
31337/tcp open  Elite  
32771/tcp open  sometimes-rpc  
32772/tcp open  sometimes-rpc  
32773/tcp open  sometimes-rpc  
32774/tcp open  sometimes-rpc  
Nmap done: 1 IP address (1 host up) scanned in 21.77 seconds
```

hosts.deny는  
어플리케이션 레벨  
접근 제어 일 뿐  
네트워크 레벨에서의  
스캔은 차단불가

록키리눅스에서  
테스트를 위해 꺼둔  
방화벽을 활성화하여 해결

```

root@kali: ~
File Actions Edit View Help
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost ~]# 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=125 time=109 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=125 time=123 ms
^C
— 8.8.8.8 ping statistics —
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 109.454/116.099/122.745/6.654 ms
[root@localhost ~]# exit
logout
Connection to 192.168.10.131 closed.

(root@kali)-[~]
# nmap -sT 192.168.10.131
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-10-27 20:33 KST
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0

(root@kali)-[~]
#

```

**hosts.deny는  
어플리케이션 레벨  
접근 제어 일 뿐  
네트워크 레벨에서의  
스캔은 차단불가**

**록키리눅스에서  
테스트를 위해 꺼둔  
방화벽을 활성화하여 해결**

## 결론 및 느낀점

**김용문 : 네트워크를 구성하면서 내부 망과 DMZ 구역에 각각 STATIC NAT와 PAT를 적용하였는데 각각의 방식이 사용되는 경우를 더 자세히 알아볼 수 있었다.**

**감사합니다.**