**NAT창선 NETWORKS   
네트워크 설계 서버 구축 가이드**

네트워크 설계 및 사용자 설명서

# 개요

더본코리아의 나마을식당 네트워크 서버 설계 프로젝트에서는 웹(WEB), 도메인 이름 시스템(DNS), 동적 호스트 설정 프로토콜(DHCP), 로그(Log), 파일 전송 프로토콜(FTP), 인증 기관(CA), 데이터베이스(DB)의 총 7개 서버를 효과적으로 구축하고 운영하기 위한 가이드를 제공합니다.

## 설계 서버 목적

더본코리아/나마을식당 인프라 구축 시 설계될 네트워크 서버에 대한 사용 목적입니다.

* WEB 서버: 웹 서버는 HTTP 프로토콜을 통해 클라이언트의 요청에 응답하며, 네트워크의 인터페이스를 제공합니다.
* DNS 서버: DNS 서버는 도메인 이름을 IP 주소로 변환하는 역할을 합니다. 이는 네트워크 내의 모든 서버와 클라이언트의 통신을 용이하게 합니다.
* DHCP 서버: DHCP 서버는 네트워크에 연결된 장치들에게 동적으로 IP 주소를 할당하는 역할을 합니다.
* Log 서버: Log 서버는 네트워크의 모든 이벤트와 활동을 기록합니다. 이는 네트워크의 안정성과 보안을 유지하는 데 중요하며, EEM 기술을 활용하여 이벤트 처리를 자동화할 수 있습니다.
* FTP 서버: FTP 서버는 네트워크를 통해 파일을 안전하게 전송하는 역할을 합니다.
* CA 서버: CA 서버는 네트워크 보안을 담당하며, SSL/TLS 인증서를 발급하고 관리합니다.

## 솔루션

**1. WEB Server**

- 설계 목적 : 나마을식당의 내부 업무용 서버와 외부 고객용 서버를 분리하여, 각각 업무 효율성 향상과 고객 서비스 개선에 중점을 두는 것입니다.

- 사용 소프트웨어 : Apache2

* **/etc/apache2/**
  + apache2.conf: Apache2의 메인 설정 파일로서, 전체적인 서버 설정을 담고 있습니다.
    - ports.conf: 웹 서버가 어떤 포트를 통해 들어오는 요청을 처리할 것인지 정의하는 파일입니다.
  + mods-enabled: Apache2에서 사용되는 모듈들의 설정 파일들을 저장하는 디렉토리입니다.
    - \*.load: 각 모듈을 로드하는 데 필요한 설정 파일입니다.
    - \*.conf: 모듈별로 추가 설정을 담고 있는 파일입니다.
  + conf-enabled: 서버 설정에 추가적으로 필요한 설정 파일들을 저장하는 디렉토리입니다.
    - \*.conf: 추가적인 설정 파일들입니다.
  + sites-enabled: 웹 서버에서 호스팅하는 웹사이트의 설정 파일들을 저장하는 디렉토리입니다.
    - \*.conf: 각 웹사이트별로 필요한 설정을 담고 있는 파일입니다.
  1. **Apache2 설정**

- 경로 : /etc/apache2/sites-available/default-ssl.conf

- Web Server에 인증서 지정 및 Server name 지정

|  |
| --- |
| IfModule mod\_ssl.c>  <VirtualHost \*:80>  ServerName namaul.thebone.com  RewriteEngine On  RewriteRule ^(.\*)$ https://%{HTTP\_HOST}$1 [R=301,L]  </VirtualHost>  <VirtualHost \_default\_:443>  ServerAdmin ncs@thebone.co.kr  ServerName namaul.thebone.com  SSLEngine on  SSLCertificateFile /etc/apache2/ssl/webserver.crt  SSLCertificateKeyFile /etc/apache2/ssl/webserver.key  DocumentRoot /var/www/html  ErrorLog ${APACHE\_LOG\_DIR}/error.log  CustomLog ${APACHE\_LOG\_DIR}/access.log combined\  <FilesMatch "\.(cgi|shtml|phtml|php)$">  SSLOptions +StdEnvVars  </FilesMatch>  <Directory /usr/lib/cgi-bin>  SSLOptions +StdEnvVars  </Directory>  </VirtualHost>  </IfModule> </VirtualHost>  </IfModule> |

* 1. **Default-SSL 설정**
* 경로 : /etc/apache2/apache2.conf
* Server 기본 설정 및 ServerName 추가

|  |
| --- |
| # configuration directives that give the server its instructions.  # See http://httpd.apache.org/docs/2.4/ for detailed information about  # the directives and /usr/share/doc/apache2/README.Debian about Debian specific  # hints.  #  #  # Summary of how the Apache 2 configuration works in Debian:  # The Apache 2 web server configuration in Debian is quite different to  # upstream's suggested way to configure the web server. This is because Debian's  # default Apache2 installation attempts to make adding and removing modules,  # virtual hosts, and extra configuration directives as flexible as possible, in  # order to make automating the changes and administering the server as easy as  # possible.  # It is split into several files forming the configuration hierarchy outlined  # below, all located in the /etc/apache2/ directory:  #  # /etc/apache2/  # |-- apache2.conf  # | `-- ports.conf  # |-- mods-enabled  # | |-- \*.load  # | `-- \*.conf  # |-- conf-enabled  # | `-- \*.conf  # `-- sites-enabled  # `-- \*.conf  #  #  # \* apache2.conf is the main configuration file (this file). It puts the pieces  # together by including all remaining configuration files when starting up the  # web server.  #  # \* ports.conf is always included from the main configuration file. It is  # supposed to determine listening ports for incoming connections which can be  # customized anytime.  #  # \* Configuration files in the mods-enabled/, conf-enabled/ and sites-enabled/  # directories contain particular configuration snippets which manage modules,  # global configuration fragments, or virtual host configurations,  # respectively.  #  # They are activated by symlinking available configuration files from their  # respective \*-available/ counterparts. These should be managed by using our  # helpers a2enmod/a2dismod, a2ensite/a2dissite and a2enconf/a2disconf. See  # their respective man pages for detailed information.  #  # \* The binary is called apache2. Due to the use of environment variables, in  # the default configuration, apache2 needs to be started/stopped with  # /etc/init.d/apache2 or apache2ctl. Calling /usr/bin/apache2 directly will not  # work with the default configuration.  # Global configuration  #  #  # ServerRoot: The top of the directory tree under which the server's  # configuration, error, and log files are kept.  #  # NOTE! If you intend to place this on an NFS (or otherwise network)  # mounted filesystem then please read the Mutex documentation (available  # at <URL:http://httpd.apache.org/docs/2.4/mod/core.html#mutex>);  # you will save yourself a lot of trouble.  #  # Do NOT add a slash at the end of the directory path.  #  #ServerRoot "/etc/apache2"  # The accept serialization lock file MUST BE STORED ON A LOCAL DISK.  #  #Mutex file:${APACHE\_LOCK\_DIR} default  # The directory where shm and other runtime files will be stored.  #  DefaultRuntimeDir ${APACHE\_RUN\_DIR}  #  # PidFile: The file in which the server should record its process  # identification number when it starts.  # This needs to be set in /etc/apache2/envvars  #  PidFile ${APACHE\_PID\_FILE}  # Timeout: The number of seconds before receives and sends time out.  #  Timeout 300  # KeepAlive: Whether or not to allow persistent connections (more than  # one request per connection). Set to "Off" to deactivate.  #  KeepAlive On  #  # MaxKeepAliveRequests: The maximum number of requests to allow  # during a persistent connection. Set to 0 to allow an unlimited amount.  # We recommend you leave this number high, for maximum performance.  #  MaxKeepAliveRequests 100  #  # KeepAliveTimeout: Number of seconds to wait for the next request from the  # same client on the same connection.  #  KeepAliveTimeout 5  # These need to be set in /etc/apache2/envvars  User ${APACHE\_RUN\_USER}  Group ${APACHE\_RUN\_GROUP}  # HostnameLookups: Log the names of clients or just their IP addresses  # e.g., www.apache.org (on) or 204.62.129.132 (off).  # The default is off because it'd be overall better for the net if people  # had to knowingly turn this feature on, since enabling it means that  # each client request will result in AT LEAST one lookup request to the  # nameserver.  #  HostnameLookups Off  # ErrorLog: The location of the error log file.  # If you do not specify an ErrorLog directive within a <VirtualHost>  # container, error messages relating to that virtual host will be  # logged here. If you \*do\* define an error logfile for a <VirtualHost>  # container, that host's errors will be logged there and not here.  #  ErrorLog ${APACHE\_LOG\_DIR}/error.log  #  # LogLevel: Control the severity of messages logged to the error\_log.  # Available values: trace8, ..., trace1, debug, info, notice, warn,  # error, crit, alert, emerg.  # It is also possible to configure the log level for particular modules, e.g.  # "LogLevel info ssl:warn"  #  LogLevel warn  # Include module configuration:  IncludeOptional mods-enabled/\*.load  IncludeOptional mods-enabled/\*.conf  # Include list of ports to listen on  Include ports.conf  # Sets the default security model of the Apache2 HTTPD server. It does  # not allow access to the root filesystem outside of /usr/share and /var/www.  # The former is used by web applications packaged in Debian,  # the latter may be used for local directories served by the web server. If  # your system is serving content from a sub-directory in /srv you must allow  # access here, or in any related virtual host.  <Directory />  Options FollowSymLinks  AllowOverride None  Require all denied  </Directory>  <Directory /usr/share>  AllowOverride None  Require all granted  </Directory>  <Directory /var/www/>  Options Indexes FollowSymLinks  AllowOverride None  Require all granted  </Directory>  #<Directory /srv/>  # Options Indexes FollowSymLinks  # AllowOverride None  # Require all granted  #</Directory>  # AccessFileName: The name of the file to look for in each directory  # for additional configuration directives. See also the AllowOverride  # directive.  #  AccessFileName .htaccess  #  # The following lines prevent .htaccess and .htpasswd files from being  # viewed by Web clients.  #  <FilesMatch "^\.ht">  Require all denied  </FilesMatch>  # The following directives define some format nicknames for use with  # a CustomLog directive.  #  # These deviate from the Common Log Format definitions in that they use %O  # (the actual bytes sent including headers) instead of %b (the size of the  # requested file), because the latter makes it impossible to detect partial  # requests.  #  # Note that the use of %{X-Forwarded-For}i instead of %h is not recommended.  # Use mod\_remoteip instead.  LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost\_combined  LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined  LogFormat "%h %l %u %t \"%r\" %>s %O" common  LogFormat "%{Referer}i -> %U" referer  LogFormat "%{User-agent}i" agent  # Include of directories ignores editors' and dpkg's backup files,  # see README.Debian for details.  # Include generic snippets of statements  IncludeOptional conf-enabled/\*.conf  # Include the virtual host configurations:  IncludeOptional sites-enabled/\*.conf  # vim: syntax=apache ts=4 sw=4 sts=4 sr noet  ServerName namaul.thebone.comshow |

* ServerRoot: 서버의 설정, 오류, 로그 파일이 저장되는 디렉토리 트리의 최상위 경로를 설정합니다.
* Mutex: 서버 내 다중 스레드를 관리하기 위한 락 파일의 위치와 타입을 설정합니다.
* DefaultRuntimeDir: 실행 시에 필요한 파일들이 저장되는 디렉토리를 설정합니다.
* PidFile: 서버가 시작될 때 생성되는 프로세스 ID가 저장되는 파일의 경로를 설정합니다.
* Timeout: 요청을 수신하고 응답하기까지의 최대 대기 시간을 설정합니다.
* KeepAlive: 연결 유지(KeepAlive) 기능을 켜거나 끕니다. 이 기능이 활성화되면 한 클라이언트가 여러 요청을 하나의 연결을 통해 할 수 있습니다.
* MaxKeepAliveRequests: 연결 유지 상태에서 한 번에 처리 가능한 최대 요청 수를 설정합니다.
* KeepAliveTimeout: 연결 유지 상태에서 다음 요청을 대기하는 시간을 설정합니다.
* User, Group: Apache 서버가 실행될 때 사용되는 사용자와 그룹을 설정합니다.
* HostnameLookups: 클라이언트의 IP 주소 또는 호스트명을 로그에 남길지 설정합니다.
* ErrorLog: 에러 로그 파일의 위치를 설정합니다.
* LogLevel: 로그 파일에 남길 메시지의 수준을 설정합니다.
* IncludeOptional: 특정 디렉토리에 있는 설정 파일들을 포함시킵니다.
* <Directory>: 특정 디렉토리에 대한 접근 권한과 옵션을 설정합니다.
* AccessFileName: 디렉토리마다 추가적인 설정을 담은 파일의 이름을 설정합니다. 기본값은 .htaccess입니다.
* <FilesMatch "^.ht">: .ht로 시작하는 파일에 대한 접근 권한을 설정합니다.
* LogFormat: 로그 파일에 남길 로그의 형식을 설정합니다.
* ServerName: 서버의 이름을 설정합니다.

각 설정 항목은 필요에 따라 수정하여 사용할 수 있습니다. 참조 : < <https://httpd.apache.org/docs/current/> >

**2. DNS Server**

- 설계 목적 : 나마을식당의 본사 및 지사에서 DNS 기능을 제공하고, 자체 서버들에 대한 도메인을 제공하는 것을 목표로 합니다. 또한, 특정 사이트 도메인을 차단하는 데에도 사용됩니다.

- 사용 소프트웨어 : Bind9

**2.1 DNS 기본 설정**

|  |
| --- |
| options {  directory "/var/cache/bind";  forwarders {8.8.8.8; 8.8.4.4;};  listen-on { any; };  allow-query { any; };  };  zone "restaurant.com" {  type master;  file "/var/lib/bind/restaurant.com.zone";  };  zone "thebone.com" {  type master;  file "/var/lib/bind/thebone.com.zone";  }; |

* 기본 옵션 설정(DNS 서버는 몇 가지 핵심 옵션을 설정):
  + 디렉토리: DNS 서버는 "/var/cache/bind" 디렉토리에서 존 파일을 찾도록 설정되어야 합니다.
  + 포워더: DNS 서버는 미리 정의된 외부 DNS 서버(8.8.8.8과 8.8.4.4)로 요청을 전달하도록 설정되어야 합니다.
  + 청취: DNS 서버는 모든 네트워크 인터페이스에서 DNS 쿼리를 수신하도록 설정되어야 합니다.
  + 쿼리 허용: DNS 서버는 모든 출처에서 오는 쿼리를 허용하도록 설정되어야 합니다.
* 존 설정:
  + "restaurant.com": 이 존의 유형은 'master'로 설정되며, 해당 존 파일은 "/var/lib/bind/restaurant.com.zone"에 위치하고 있습니다.
  + "thebone.com": 이 존의 유형 또한 'master'로 설정되며, 해당 존 파일은 "/var/lib/bind/thebone.com.zone"에 위치하고 있습니다.

**2.2 Main : thebone.com.zone**

|  |
| --- |
| $TTL 604800  @ IN SOA thebone.com. root.thebone.com. (  2023072601 ; Serial  604800 ; Refresh  86400 ; Retry  2419200 ; Expire  604800 ) ; Negative Cache TTL  ;  @ IN NS thebone.com.  @ IN A 172.20.1.2  namaul IN A 172.20.1.2 |

**2.3 Main : restaurant.com.zone**

|  |
| --- |
| $TTL 604800  @ IN SOA restaurant.com. root.restaurant.com. (  2023072601 ; Serial  604800 ; Refresh  86400 ; Retry  2419200 ; Expire  604800 ) ; Negative Cache TTL  ;  @ IN NS restaurant.com.  @ IN A 172.20.1.2  hq IN A 172.20.1.2  ftp IN A 172.20.1.3  dns IN A 172.20.1.4  dhcp IN A 172.20.1.5  syslog IN A 172.20.1.6 |

**2.4 Backup : thebone.com.zone**

|  |
| --- |
| $TTL 604800  @ IN SOA thebone.com. root.thebone.com. (  2023072601 ; Serial  604800 ; Refresh  86400 ; Retry  2419200 ; Expire  604800 ) ; Negative Cache TTL  ;  @ IN NS thebone.com.  @ IN A 172.20.2.2  namaul IN A 172.20.2.2 |

**2.5 Backup : restaurant.com.zone**

|  |
| --- |
| $TTL 604800  @ IN SOA restaurant.com. root.restaurant.com. (  2023072601 ; Serial  604800 ; Refresh  86400 ; Retry  2419200 ; Expire  604800 ) ; Negative Cache TTL  ;  @ IN NS restaurant.com.  @ IN A 172.20.1.2  hq IN A 172.20.2.2  ftp IN A 172.20.2.3  dns IN A 172.20.2.4  dhcp IN A 172.20.2.5  syslog IN A 172.20.2.6 |

* TTL(Time to Live): 이 값은 DNS 정보의 캐시 가능 기간을 설정합니다($TTL 604800). 여기서는 604800초(1주)로 설정되어 있습니다.
* SOA(Start of Authority) 레코드: 이 레코드는 이 zone에 대한 권한이 있는 서버와 관련 정보를 포함하며, zone의 주요 설정을 담고 있습니다.
  + MNAME: 이 도메인의 권한을 가진 서버의 이름(restaurant.com.).
  + RNAME: 관리자의 이메일 주소(root.restaurant.com.), 이메일에서 '@' 대신 '.'을 사용합니다.
  + SERIAL: Zone 파일의 버전을 표시하며, 이 값을 증가시키면 slave DNS 서버들이 새로운 정보로 업데이트하게 됩니다.
  + REFRESH, RETRY, EXPIRE, MINIMUM: 각각 새로고침 주기, 재시도 주기, 만료 시간, 캐시의 최소 TTL을 의미합니다.
* NS(Name Server) 레코드: 도메인에 대한 DNS 서버를 지정합니다(@ IN NS restaurant.com.). '@'는 해당 도메인 이름(여기서는 'restaurant.com.')을 나타냅니다.
* A(Address) 레코드: 도메인 이름을 해당하는 IP 주소로 변환하는 데 사용됩니다. 여기서는 도메인 restaurant.com(@ IN A 172.20.1.2)과 서브도메인들(hq, ftp, dns, dhcp, syslog)의 IP 주소를 지정합니다.

**3. DHCP Server**

- 설계 목적 : 나마을식당의 본사에서 IP 주소를 효율적으로 관리하는 것을 목표로 합니다.

- 사용 소프트웨어 : dhcpd

**3.1 DHCP 설정**

- 경로 : /etc/dhcpd/dhcpd.conf

|  |
| --- |
| # dhcpd.conf  allowinterfaces eth\*  # option definitions common to all supported networks...  option domain-name "dhcp.restaurant.com. ";  option domain-name- servers 172.20.1.2, 172.20.2.2;  default-lease-time 600;  max-lease-time 7200;  ddns-update-style none;  subnet 172.20.0.0 netmask 255.255.255.0 {  range 172.20.0.11 172.20.0.250;  option subnet-mask 255.255.255.0;  option domain-name-servers 172.20.1.2, 172.20.2.2;  host client1 {  hardware ethernet 00:1A:2B:3C:4D:5E;  fixed-address 172.20.0.1;  }  }  subnet 172.20.1.0 netmask 255.255.255.0 {  range 172.20.1.11 172.20.1.250;  option subnet-mask 255.255.255.0;  option routers 172.20.11.254;  option domain-name-servers 172.20.1.2, 172.20.2.2;  }  subnet 172.20.2.0 netmask 255.255.255.0 {  range 172.20.2.11 172.20.2.250;  option subnet-mask 255.255.255.0;  option routers 172.20.2.254;  option domain-name-servers 172.20.2.2, 172.20.1.2;  }  subnet 172.20.10.0 netmask 255.255.255.0 {  range 172.20.10.11 172.20.10.250;  option subnet-mask 255.255.255.0;  option routers 172.20.10.254;  option domain-name-servers 172.20.1.2, 172.20.2.2;  }  subnet 172.20.20.0 netmask 255.255.255.0 {  range 172.20.20.11 172.20.20.250;  option subnet-mask 255.255.255.0;  option routers 172.20.20.254;  option domain-name-servers 172.20.1.2, 172.20.2.2;  }  subnet 172.20.30.0 netmask 255.255.255.0 {  range 172.20.30.11 172.20.30.250;  option subnet-mask 255.255.255.0;  option routers 172.20.30.254;  option domain-name-servers 172.20.1.2, 172.20.2.2;  }  subnet 172.20.40.0 netmask 255.255.255.0 {  range 172.20.40.11 172.20.40.250;  option subnet-mask 255.255.255.0;  option routers 172.20.40.254;  option domain-name-servers 172.20.1.2, 172.20.2.2;  }  subnet 172.20.50.0 netmask 255.255.255.0 {  range 172.20.50.11 172.20.50.250;  option subnet-mask 255.255.255.0;  option routers 172.20.50.254;  option domain-name-servers 172.20.1.2, 172.20.2.2;  }  subnet 172.20.60.0 netmask 255.255.255.0 {  range 172.20.60.11 172.20.60.250;  option subnet-mask 255.255.255.0;  option routers 172.20.60.254;  option domain-name-servers 172.20.1.2, 172.20.2.2;  }  subnet 172.20.70.0 netmask 255.255.255.0 {  range 172.20.70.11 172.20.70.250;  option subnet-mask 255.255.255.0;  option routers 172.20.70.254;  option domain-name-servers 172.20.1.2, 172.20.2.2;  }  subnet 172.20.80.0 netmask 255.255.255.0 {  range 172.20.80.11 172.20.80.250;  option subnet-mask 255.255.255.0;  option routers 172.20.80.254;  option domain-name-servers 172.20.1.2, 172.20.2.2;  }  subnet 172.20.90.0 netmask 255.255.255.0 {  range 172.20.90.11 172.20.90.250;  option subnet-mask 255.255.255.0;  option routers 172.20.90.254;  option domain-name-servers 172.20.1.2, 172.20.2.2;  }  subnet 172.20.100.0 netmask 255.255.255.0 {  range 172.20.100.11 172.20.100.250;  option subnet-mask 255.255.255.0;  option routers 172.20.100.254;  option domain-name-servers 172.20.1.2, 172.20.2.2;  } |

* 인터페이스 설정:
  + DHCP 서버는 'eth'로 시작하는 모든 인터페이스에서 DHCP 요청을 수락하도록 설정되어야 합니다. 이는 각 인터페이스에서 트래픽을 수용하고, 서버의 전체 이더넷 포트에서 트래픽을 처리할 수 있도록 합니다.
* 공통 옵션 설정:
  + 도메인 이름: 모든 네트워크에 대해 도메인 이름을 " dhcp.restaurant.com"로 설정합니다.
  + 도메인 이름 서버: 모든 네트워크에 대해 도메인 이름 서버를 172.20.1.2와 172.20.2.2로 설정합니다.
  + 임대 시간: DHCP 임대의 기본 및 최대 시간을 각각 600초와 7200초로 설정합니다.
  + 동적 DNS 업데이트: 동적 DNS 업데이트는 사용되지 않도록 설정합니다.
* 서브넷 설정:
  + 각 서브넷마다 고유한 설정을 정의합니다. 이 설정에는 IP 주소 범위, 서브넷 마스크, 라우터, 도메인 이름 서버가 포함됩니다. 예를 들어, 172.20.0.0 서브넷의 IP 주소 범위는 172.20.0.11에서 172.20.0.250까지입니다.
* 고정 주소 할당:
  + 특정 호스트, 예를 들어 "client1"에 대해 고정 IP 주소를 설정합니다. 이 호스트의 MAC 주소는 00:1A:2B:3C:4D:5E이며, 고정 IP 주소는 172.20.0.1입니다.
* 서브넷 구성:
  + 총 12개의 서브넷이 있습니다. 이들은 172.20.0.0부터 172.20.100.0까지의 범위를 포함하며, 각 서브넷은 고유한 IP 주소 범위, 서브넷 마스크, 기본 라우터, 도메인 이름 서버를 갖습니다.

**4. FTP Server**

- 설계 목적 : 나마을식당의 내부 사용자가 필요한 파일을 안전하게 전송하고, CA 서버를 통한 인증으로 보안을 강화하는 것을 목표로 합니다.

- 사용 소프트웨어 : vsftpd

**4.1 FTP 설정**

- 경로 : /etc/vsftpd.conf

|  |
| --- |
| anonymous\_enable=NO  local\_enable=YES  write\_enable=YES  local\_umask=022  dirmessage\_enable=YES  xferlog\_enable=YES  xferlog\_file=/var/log/xferlog  xferlog\_std\_format=YES  banner\_file=/test/ftp.txt  chroot\_list\_enable=YES  chroot\_list\_file=/etc/vsftpd/chroot\_list  allow\_writeable\_chroot=YES  listen=YE  pam\_service\_name=vsftpd  userlist\_enable=YES  tcp\_wrappers=YES  pasv\_enable=YES  pasv\_min\_port=50000  pasv\_max\_port=50010  rsa\_cert\_file=/ssl/server.crt  rsa\_private\_key\_file=/ssl/server.key  ssl\_enable=YES  force\_local\_data\_ssl=YES  force\_local\_logins\_ssl=YES    ssl\_tlsv1=YES  ssl\_ciphers=HIGH |

* 기본 설정:
  + 익명 사용자: FTP 서버는 익명 사용자를 허용하지 않습니다(anonymous\_enable=NO).
  + 로컬 사용자: 로컬 사용자의 로그인은 허용됩니다(local\_enable=YES).
  + 쓰기 허용: 로컬 사용자는 파일 쓰기 권한을 가지게 됩니다(write\_enable=YES).
  + Umask: 로컬 사용자가 생성하는 파일의 기본 권한은 755로 설정됩니다(local\_umask=022).
  + 디렉토리 메시지: 디렉토리 메시지는 사용자가 새로운 디렉토리로 이동할 때 표시됩니다(dirmessage\_enable=YES)

.

* 로그 설정:
  + 전송 로그: 전송 로그는 활성화 되어 있으며, 모든 파일 전송은 /var/log/xferlog 파일에 기록됩니다(xferlog\_enable=YES).
* 보안 설정:
  + chroot: 사용자는 각자의 홈 디렉토리로 제한(chroot) 됩니다(chroot\_list\_enable=YES).
  + 쓰기 가능한 chroot: 쓰기 가능한 디렉토리로의 chroot가 허용됩니다(allow\_writeable\_chroot=YES).
  + PAM: vsftpd는 사용자 인증에 PAM을 사용합니다(pam\_service\_name=vsftpd).
  + 사용자 목록: 특정 사용자들에게만 접속을 허용할 수 있습니다(userlist\_enable=YES).
* 패시브 모드 설정:
  + 패시브 모드: 패시브 모드를 지원합니다(pasv\_enable=YES).
  + 패시브 포트 범위: 패시브 모드에서 사용하는 포트 범위는 50000에서 50010까지입니다(pasv\_min\_port=50000, pasv\_max\_port=50010).
* SSL/TLS 설정:
  + SSL 활성화: SSL이 활성화되어 있습니다(ssl\_enable=YES).
  + 강제 SSL: 모든 데이터 전송과 로그인은 SSL을 통해 이루어집니다(force\_local\_data\_ssl=YES, force\_local\_logins\_ssl=YES).
  + TLSv1: TLSv1이 지원됩니다(ssl\_tlsv1=YES).
  + 암호화: 높은 보안 수준의 암호화를 사용합니다(ssl\_ciphers=HIGH).
  + 인증서: RSA 인증서 파일은 /ssl/server.crt에, 개인 키 파일은 /ssl/server.key에 위치하고 있습니다(rsa\_cert\_file=/ssl/server.crt, rsa\_private\_key\_file=/ssl/server.key).

**5. Log Server**

- 설계 목적 : 나마을식당의 내부 네트워크 장비에 대한 로그 정보를 중앙화하여 관리하고 모니터링하는 것입니다.

- 사용 소프트웨어 : rsyslog

**5.1 Rsyslog 설정**

- 경로 : /etc/rsyslog.conf

|  |
| --- |
| # /etc/rsyslog.conf configuration file for rsyslog  #  # For more information install rsyslog-doc and see  # /usr/share/doc/rsyslog-doc/html/configuration/index.html  #  # Default logging rules can be found in /etc/rsyslog.d/50-default.conf  #################  #### MODULES ####  #################  module(load="imuxsock") # provides support for local system logging  #module(load="immark") # provides --MARK-- message capability  # provides UDP syslog reception  module(load="imudp")  input(type="imudp" port="514")  # provides TCP syslog reception  #module(load="imtcp")  #input(type="imtcp" port="514")  # provides kernel logging support and enable non-kernel klog messages  module(load="imklog" permitnonkernelfacility="on")  ###########################  #### GLOBAL DIRECTIVES ####  ###########################  #  # Use traditional timestamp format.  # To enable high precision timestamps, comment out the following line.  #  $ActionFileDefaultTemplate RSYSLOG\_TraditionalFileFormat  # Filter duplicated messages  $RepeatedMsgReduction on  #  # Set the default permissions for all log files.  #  $FileOwner syslog  $FileGroup adm  $FileCreateMode 0640  $DirCreateMode 0755  $Umask 0022  $PrivDropToUser syslog  $PrivDropToGroup syslog  #  # Where to place spool and state files  #  $WorkDirectory /var/spool/rsyslog  # Include all config files in /etc/rsyslog.d/  $IncludeConfig /etc/rsyslog.d/\*.conf |

* 모듈 설정:
  + 로컬 시스템 로깅: 로컬 시스템 로깅을 지원하는 모듈인 'imuxsock'이 로드되었습니다(module(load="imuxsock")).
  + UDP syslog 수신: UDP syslog 메시지를 수신하기 위해 'imudp' 모듈이 로드되었습니다. 해당 모듈은 UDP 포트 514에서 메시지를 수신합니다(module(load="imudp"), input(type="imudp" port="514")).
  + 커널 로깅: 커널 로깅 지원과 비커널 klog 메시지를 활성화하는 'imklog' 모듈이 로드되었습니다(module(load="imklog" permitnonkernelfacility="on")).
* 전역 지시문 설정:
  + 타임스탬프 포맷: 전통적인 타임스탬프 포맷을 사용합니다($ActionFileDefaultTemplate RSYSLOG\_TraditionalFileFormat).
  + 중복 메시지 필터링: 중복된 메시지는 필터링됩니다($RepeatedMsgReduction on).
  + 파일 권한 설정: 모든 로그 파일의 기본 권한이 설정되어 있습니다
    - $FileOwner syslog
    - $FileGroup adm
    - $FileCreateMode 0640
    - $DirCreateMode 0755
    - $Umask 0022
    - $PrivDropToUser syslog
    - $PrivDropToGroup syslog.
  + 스풀 및 상태 파일 위치: 스풀 및 상태 파일은 /var/spool/rsyslog 디렉토리에 위치합니다($WorkDirectory /var/spool/rsyslog).
  + 설정 파일 포함: /etc/rsyslog.d/ 디렉토리에 있는 모든 .conf 파일이 포함됩니다($IncludeConfig /etc/rsyslog.d/\*.conf).

**5.2 로그 파일 지정**

- 경로 : /etc/rsyslog.d/30-RouterLog.conf

|  |
| --- |
| template(name="CustomFormat" type="string" string="%timegenerated% %hostname% %syslogtag%%msg%\n")  if $fromhost-ip == '172.20.1.254' then {  action(type="omfile" file="/var/log/Backbone2.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.10' then {  action(type="omfile" file="/var/log/st1Multy.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.20' then {  action(type="omfile" file="/var/log/nd2Multy.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.30' then {  action(type="omfile" file="/var/log/rd3Multy.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.40' then {  action(type="omfile" file="/var/log/th4Multy.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.50' then {  action(type="omfile" file="/var/log/th5Multy.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.11' then {  action(type="omfile" file="/var/log/st1SW1.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.12' then {  action(type="omfile" file="/var/log/st1SW2.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.13' then {  action(type="omfile" file="/var/log/st1SW3.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.14' then {  action(type="omfile" file="/var/log/st1SW4.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.15' then {  action(type="omfile" file="/var/log/st1SW5.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.21' then {  action(type="omfile" file="/var/log/nd2SW1.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.22' then {  action(type="omfile" file="/var/log/nd2SW2.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.23' then {  action(type="omfile" file="/var/log/nd2SW3.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.24' then {  action(type="omfile" file="/var/log/nd2SW4.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.25' then {  action(type="omfile" file="/var/log/nd2SW5.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.31' then {  action(type="omfile" file="/var/log/rd3SW1.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.32' then {  action(type="omfile" file="/var/log/rd3SW2.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.33' then {  action(type="omfile" file="/var/log/rd3SW3.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.34' then {  action(type="omfile" file="/var/log/rd3SW4.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.35' then {  action(type="omfile" file="/var/log/rd3SW5.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.41' then {  action(type="omfile" file="/var/log/th4SW1.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.42' then {  action(type="omfile" file="/var/log/th4SW2.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.43' then {  action(type="omfile" file="/var/log/th4SW3.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.44' then {  action(type="omfile" file="/var/log/th4SW4.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.45' then {  action(type="omfile" file="/var/log/th4SW5.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.51' then {  action(type="omfile" file="/var/log/th5SW1.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.52' then {  action(type="omfile" file="/var/log/th5SW2.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.53' then {  action(type="omfile" file="/var/log/th5SW3.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.54' then {  action(type="omfile" file="/var/log/th5SW4.log" template="CustomFormat")  }  if $fromhost-ip == '172.20.3.55' then {  action(type="omfile" file="/var/log/th5SW5.log" template="CustomFormat") |

* 로그 포맷 설정: 로그 메시지의 포맷을 정의합니다.
  + 이 설정에서는 '%timegenerated% %hostname% %syslogtag%%msg%\n' 포맷이 사용되었습니다
  + (template(name="CustomFormat" type="string" string="%timegenerated% %hostname% %syslogtag%%msg%\n")).
* 네트워크 장비 별 로그 처리
  + 특정 IP 주소를 가진 호스트로부터 로그를 받을 경우, 해당 로그를 지정된 파일에 저장하도록 설정합니다. 이를 통해 네트워크 장비 별 로그를 분리하여 저장할 수 있습니다.
  + 예를 들어, '172.20.3.10' 주소를 가진 호스트의 로그는 "/var/log/st1Multy.log" 파일에 저장되며, 이때 'CustomFormat' 포맷이 사용됩니다

**6. CA Server**

- 설계 목적 : 나마을식당의 본사 및 지사에서 보안 기능을 실현하기 위해 CA 서버를 사용합니다. CA 서버는 디지털 인증서를 발급, 관리하며 이를 통해 안전한 통신을 보장합니다.

- 사용 소프트웨어 : OpenSSL

**6.1. 인증서 발급**

- CA서버 자체 인증서 생성

|  |
| --- |
| openssl genrsa -out ca.key 2048  openssl genrsa -aes256 -out lesstif-rootca.key 2048  openssl req -new -key ca.key -out ca.csr  openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt |

**6.2 사용 인증서 인증**

- 본사, 지사의 자체 인증서에 대한 서명 및 인증을 통한 보안을 사용하는 서버에 대한 인증서 인증과정입니다.

- Web 서버 인증서 발급

|  |
| --- |
| openssl genrsa -out ca.key 2048  openssl genrsa -aes256 -out lesstif-rootca.key 2048  openssl req -new -key ca.key -out ca.csr  openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt |

- 본사 인증서 발급

|  |
| --- |
| openssl genrsa -out hqftp.key 2048  openssl req -new -key hqftp.key -out hqftp.csr  openssl x509 -req -extensions v3\_user -in hqftp.csr -CA ca.crt -CAcreateserial -CAkey ca.key -out hqftp.crt |

- 서울 지사 인증서 발급

|  |
| --- |
| openssl genrsa -out seoulftp.key 2048  openssl req -new -key seoulftp.key -out seoulftp.csr  openssl x509 -req -extensions v3\_user -in seoulftp.csr -CA ca.crt -CAcreateserial -CAkey ca.key -out seoulftp.crt |

- 경기지사 인증서 발급

|  |
| --- |
| openssl genrsa -out seoulftp.key 2048  openssl req -new -key seoulftp.key -out seoulftp.csr  openssl x509 -req -extensions v3\_user -in seoulftp.csr -CA ca.crt -CAcreateserial -CAkey ca.key -out seoulftp.crt |

* 개인키 생성: OpenSSL의 'genrsa' 명령을 사용하여 2048비트 RSA 개인키를 생성합니다. 이 키는 'seoulftp.key' 파일에 저장됩니다(openssl genrsa -out seoulftp.key 2048).
* 인증서 서명 요청(CSR) 생성: 'req' 명령을 사용하여 생성된 개인키를 이용해 인증서 서명 요청(Certificate Signing Request, CSR)을 생성합니다. 이 요청은 'seoulftp.csr' 파일에 저장됩니다(openssl req -new -key seoulftp.key -out seoulftp.csr).
* 인증서 서명: 생성된 CSR을 CA의 인증서(ca.crt)와 개인키(ca.key)를 이용하여 서명합니다. 이 때 'x509' 명령을 사용하며, 서명된 인증서는 'seoulftp.crt' 파일에 저장됩니다(openssl x509 -req -extensions v3\_user -in seoulftp.csr -CA ca.crt -CAcreateserial -CAkey ca.key -out seoulftp.crt).

## 서버 작동 테스트

* WEB 서버

- 웹 서버는 HTTP 프로토콜을 통해 클라이언트의 요청에 응답하며, 네트워크의 인터페이스를 제공합니다.

|  |  |  |
| --- | --- | --- |
| **작동 순서도** | | |
|  | | |
| **설명** | 1. 사용자가 URL (도메인 주소)를 통해 웹 서버에 접속을 시도하면, 클라이언트 시스템은 먼저 해당 도메인의 IP 주소를 확인하기 위해 설정된 DNS 서버에 쿼리를 보냅니다. 2. DNS 서버는 클라이언트에게 해당 도메인의 IP 주소를 포함한 정보를 응답으로 전송합니다. 클라이언트는 이 정보를 받아 해당 웹 서버의 IP 주소를 알아냅니다. 3. DNS를 통해 얻은 IP 주소를 활용하여 클라이언트는 웹 서버에 접근합니다. 이 때, 웹 서버와 클라이언트 사이의 통신은 HTTP 또는 HTTPS 프로토콜을 이용하며, 이는 각각 80번과 443번 포트를 사용합니다. 4. 웹 서버에 성공적으로 접속하면, 서버는 클라이언트에게 웹 페이지를 보여주기 위한 데이터를 전송하며, 클라이언트는 이 데이터를 해석하여 사용자에게 웹 페이지를 표시합니다. | |
| **결과** |  |  |

* DNS 서버
* DNS 서버는 도메인 이름을 IP 주소로 변환하는 역할을 합니다. 이는 네트워크 내의 모든 서버와 클라이언트의 통신을 용이하게 합니다.

|  |  |
| --- | --- |
| **설계 설명** | |
|  | |
| **설명** | Docker를 사용하여 Bind9 DNS 서버 컨테이너를 생성하였습니다. 생성 과정에서 볼륨 옵션(-v)을 활용하여 named.conf 파일과 zone 파일들을 호스트 서버의 지정된 폴더로 마운트하였습니다. 이를 통해, DNS 서버는 호스트 서버에 저장된 설정 파일들을 바로 참조하여 사용할 수 있게 하였습니다. |
|  | |
| **설명** | 현재 DNS 컨테이너가 정상적으로 작동 중임을 확인하였습니다. |
|  | |
| **설명** | 라우터에서는 기본적으로 사용할 DNS 서버(172.20.1.2)와 장애 발생 시 대비한 보조 DNS 서버(172.20.2.2)를 설정하였습니다. 또한, "ip domain lookup" 명령어를 사용하여 라우터에서 DNS 질의가 가능하도록 설정하였습니다. 이렇게 함으로써, 메인 DNS 서버에 장애가 발생했을 때에도 보조 DNS 서버를 통해 도메인 조회가 계속 이루어질 수 있도록 하였습니다.. |

|  |  |
| --- | --- |
| **작동 순서도** | |
|  | |
| **설명** | 1. 사용자가 PC에서 'namaul.thebone.com'라는 도메인 주소를 입력하면, 이는 DHCP에 의해 설정된 DNS 서버로 패킷을 전송하기 위한 DNS 쿼리를 생성합니다. 이는 UDP 53번 포트를 사용하여 통신합니다. 2. DNS 쿼리를 받은 DNS 서버는 자신의 레코드에 해당 도메인이 존재하는지 확인합니다. 만약 존재한다면, 해당 서버의 IP 주소를 반환하고, 존재하지 않는다면 상위 DNS 서버인 Root DNS로 쿼리를 재전송합니다. 3. DNS 서버로부터 도메인에 대한 IP 주소를 수신한 PC는 이 IP 주소를 이용하여 웹 통신을 진행합니다. 이로써 사용자는 원하는 도메인에 접근할 수 있게 됩니다. |
| **결과 (1)** |  |
| **설명** | 라우터에서 URL 주소로 ping 명령을 전송하여 DNS 조회가 성공하는지 확인하였습니다. 이를 통해, 주 DNS 서버가 도메인 조회 요청에 정상적으로 응답하며, 이로 인해 라우터가 해당 URL에 대한 IP 주소를 정확하게 얻어냄을 알 수 있었습니다. 이 결과는 주 DNS 서버의 정상적인 운영을 나타냅니다. |
| **결과 (2)** |  |
| **설명** | 주 DNS 서버(172.20.1.2)에서 발생한 장애로 인해 도메인 조회가 불가능할 때, 보조 DNS 서버의 작동을 확인하였습니다. 이에 따라 시스템의 DNS 조회 요청이 주 DNS 서버에서 실패한 경우 보조 DNS 서버로 자동 전환되는 기능이 정상적으로 동작하는 것을 확인하였습니다. |

* DHCP 서버
* DHCP 서버는 네트워크에 연결된 장치들에게 동적으로 IP 주소를 할당하는 역할을 합니다.

|  |  |
| --- | --- |
| **설계 확인** | |
|  | |
| **설명** | Docker를 사용하여 Bind9 DNS 서버 컨테이너를 생성하였습니다. 생성 과정에서 볼륨 옵션(-v)을 활용하여 named.conf 파일과 zone 파일들을 호스트 서버의 지정된 폴더로 마운트하였습니다. 이를 통해, DNS 서버는 호스트 서버에 저장된 설정 파일들을 바로 참조하여 사용할 수 있게 하였습니다. |
|  | |
| **설명** | 분배 L3 스위치와 백본 라우터 (Backbone 3, 4)의 연결 지점에서, 각 부서별 VLAN 인터페이스에 IP가 적절하게 할당되어 있음을 확인하였습니다. 또한, ip helper-address 명령어를 통해 DHCP relay agent 기능이 활성화된 상태임을 확인했습니다. 이 후, 라우터가 DHCP 기능을 활용할 수 있도록 service dhcp 명령어를 실행하였습니다. |

|  |  |
| --- | --- |
| **작동 순서도** | |
|  | |
| **설명** | 1. PC가 IP를 자동 할당받기 위해 'Discover' 메시지를 전송합니다. 현재 PC가 IP를 가지고 있지 않으므로, 소스 IP는 '0.0.0.0', 목적지 IP는 '255.255.255.255'를 사용하여 브로드캐스트로 전송합니다.  * 이 때, DHCP 서버가 같은 네트워크 대역에 존재하지 않는 경우, 해당 패킷을 수신하지 못하게 되므로 라우터에서 'ip helper-address' 명령어를 사용하여 DHCP Relay agent 기능을 활성화 해야 합니다.  1. Discover 메시지를 수신한 DHCP 서버는 사용 가능한 IP 주소 중 하나를 선택하여 'Offer' 메시지를 PC에 전송합니다. 2. Offer 메시지를 수신한 PC는 해당 IP를 사용하기 위해 'Request' 메시지를 전송합니다. 3. Request 메시지를 수신한 DHCP 서버는 해당 IP 할당을 확정하고 'Ack' 메시지를 PC에 전송합니다.  * 이후, IP의 임대 기간이 만료될 때마다 PC와 DHCP 서버는 'Discover'와 'Offer' 메시지 교환 과정을 생략하고, 'Request'와 'Ack' 메시지만 교환하게 됩니다. 이를 통해 IP 재할당 과정이 간소화됩니다. |
| **결과 (1)** |  |
| **설명** | PC가 DHCP 기능을 사용하여 해당 부서(VLAN)에 맞는 IP 주소를 성공적으로 할당받았습니다. 이를 통해 PC가 각 VLAN에 따라 적절한 IP 주소를 가질 수 있음을 확인하였습니다. 이 결과는 해당 부서별로 정확한 네트워크 구성이 이루어졌음을 보여줍니다. |

* Log 서버
* Log 서버는 네트워크의 모든 이벤트와 활동을 기록합니다. 이는 네트워크의 안정성과 보안을 유지하는 데 중요하며, EEM 기술을 활용하여 이벤트 처리를 자동화할 수 있습니다.

|  |  |
| --- | --- |
| **설계 확인** | |
|  | |
| **설명** | Docker를 활용하여 rsyslog 서비스를 위한 컨테이너를 생성하고 실행합니다. 이 과정에서 볼륨 옵션(-v)을 사용하여 각 네트워크 장비에 해당하는 로그 파일들이 저장될 폴더를 지정합니다 |
|  | |
| **설명** | Log 데이터를 전송할 네트워크 장비들 (예: 라우터, 스위치 등)에서, 전송할 Log 서버를 지정합니다. 이를 통해 네트워크 장비로부터 로그 데이터가 올바르게 해당 로그 서버로 전송될 수 있도록 설정합니다. |

|  |  |
| --- | --- |
| **작동 순서도** | |
|  | |
| **설명** | 1. Syslog 메시지 생성 및 전송   네트워크 장비의 상태 변화나 사용자가 알아야 할 중요한 접근이 발생하면, 해당 정보를 포함한 Syslog 메시지를 Log Server로 전송합니다. 대다수의 네트워크 장비는 중요도에 따라 로그 레벨을 구분하여 로그 데이터 전송이 가능합니다.   1. Log Server에서의 로그 데이터 처리   Log Server는 들어오는 Syslog 메시지의 출처 IP를 확인하여 특정 네트워크 장비를 식별합니다. 이후 식별된 로그 정보를 사전에 정의된 형식의 파일로 저장하게 됩니다. 이를 통해 로그 데이터는 적절하게 분류 및 저장됩니다. 이 과정은 모든 네트워크 장비와 로그 서버 간에 일관되게 이루어집니다. |
| **결과 (1)** |  |
| **설명** | 사전에 정의된 로그 데이터 저장 폴더에서, 각 네트워크 장비로부터 받아온 로그 파일들이 정상적으로 저장되어 있는 것을 확인하였습니다. 이를 통해, 로그 데이터의 일관된 분류 및 저장이 잘 이루어지고 있음을 확인할 수 있습니다. |

* FTP 서버
* FTP 서버는 네트워크를 통해 파일을 안전하게 전송하는 역할을 합니다.

|  |  |
| --- | --- |
| **설계 확인** | |
|  | |
| **설명** | Docker를 이용하여 vsftpd 컨테이너를 생성하고 실행합니다. 이 컨테이너는 'router' 사용자를 통해 파일 저장 기능을 수행하도록 설정됩니다. 볼륨 옵션(-v)을 사용하여 FTP 파일들이 지정된 폴더에 저장되도록 합니다. |
|  | |
| **설명** | 현재 실행 중인 FTP 컨테이너의 상태를 확인합니다. 이는 컨테이너의 작동 상태를 점검하고 필요한 서비스를 제공하는 데 중요합니다. |

|  |  |
| --- | --- |
| **작동 순서도** | |
|  | |
| **설명** | 1. 초기 접속 요청: FTP 서버에 파일 송수신을 위한 접속을 원하는 장비 (PC, Router, Switch 등)는 FTP 서버에 대해 제어 채널 연결을 요청합니다 (TCP 21번 포트 사용). 2. 데이터 채널 설정: 서버는 이후 데이터를 전송할 수 있는 채널을 설정합니다. 이 때, 두 가지 모드 중 하나를 선택하여 통신이 진행됩니다.  * Passive Mode (수동 모드): 이 모드에서는 서버가 데이터 채널을 오픈하고, 임의로 지정한 포트에서 정보를 교환합니다. * Active Mode (능동 모드): 이 모드에서는 클라이언트가 데이터 채널을 오픈하고, 클라이언트가 서버에 접속하여 정보를 교환합니다. 단, 클라이언트에 방화벽이 설정되어 있으면 동작하지 않을 수 있습니다.  1. 데이터 송수신: 연결이 성립된 이후에 서버와 클라이언트 사이에서 데이터 송수신이 이루어집니다. 이는 파일 업로드, 다운로드 등의 목적으로 사용될 수 있습니다. |
| **결과 (1)** |  |
| **설명** | FTP를 통한 Configuration 파일 전송 검증  라우터는 FTP 명령을 사용하여 running-config 파일을 FTP 서버로 성공적으로 전송하는지 검증합니다. 이 과정은 네트워크 장비 설정의 백업 및 복구에 필수적인 절차입니다. |
| **결과 (2)** |  |
| **설명** | 전송 파일 저장 확인  전송된 파일이 FTP 서버에 마운트된 지정된 폴더에 올바르게 저장되었는지 확인합니다. 이는 데이터 무결성 및 파일 접근 가능성을 보장하는 중요한 절차입니다. |

* CA 서버
* CA 서버는 네트워크 보안을 담당하며, SSL/TLS 인증서를 발급하고 관리합니다.

|  |  |
| --- | --- |
| **작동 순서도** | |
|  | |
| **설명** | 사설 CA서버는 자체적으로 서명한 인증키를 생성합니다. 이 인증키는 PC에 설치되며, 이 과정을 통해 '신뢰할 수 있는 루트 인증 기관'에 등록됩니다. 이렇게 등록된 인증서는 사설 CA서버를 통해 서명받은 다른 인증서들을 신뢰할 수 있는 인증서로 인식합니다. |
| **결과 (1)** |  |
| **설명** | 사설 CA 인증서가 신뢰할 수 있는 루트 인증 기관에 등록된 후, 해당 CA서버에서 인증받은 웹 서버에 HTTPS로 접근하면, 해당 웹페이지는 신뢰할 수 있는 웹페이지로 인식됩니다. |
| **결과 (2)** |  |
| **설명** | 인증서를 사용하여 FTP 통신을 시도하면, 해당 통신은 자동으로 FTPS로 변경됩니다. 이로 인해 파일 전송 과정이 암호화되어 보안이 강화됩니다. 다만, 일부 FTP 클라이언트(예: 파일질라)는 특정 버전의 인증서를 신뢰할 수 없는 인증서로 간주하는 경우가 있을 수 있음을 유의해야 합니다. |