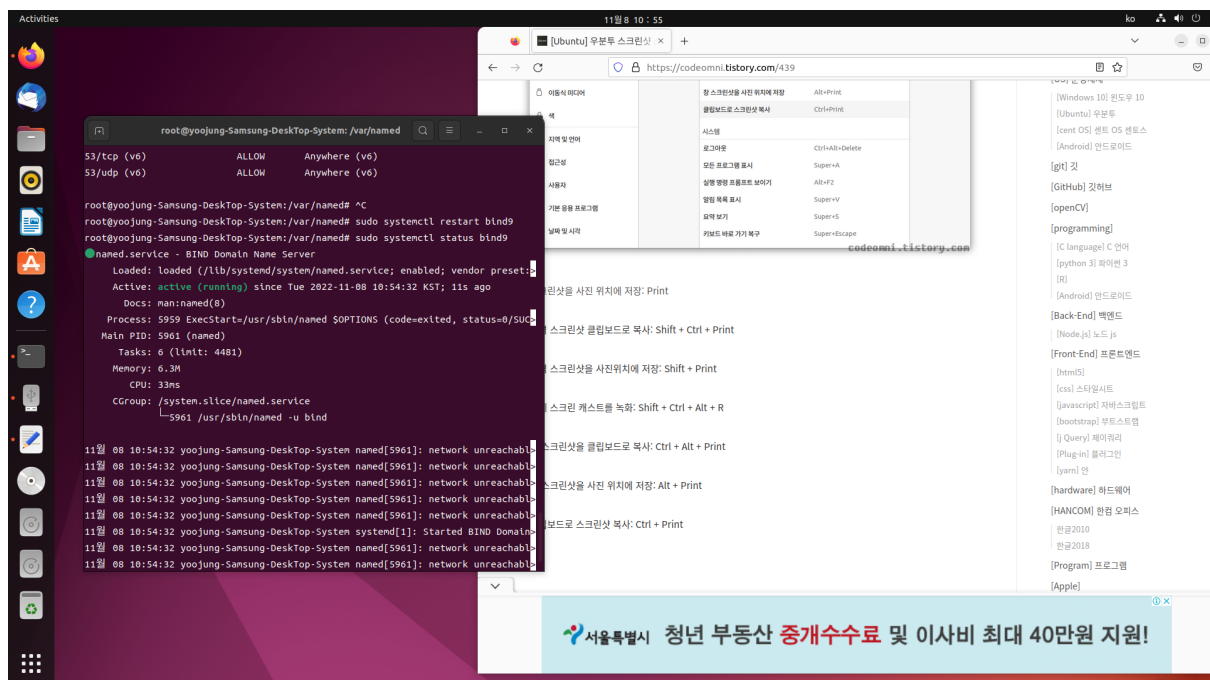


# DNS서버

Created By	
Stakeholders	
Status	
Type	
Created	@2022년 10월 31일 오전 8:21
Last Edited Time	@2022년 11월 30일 오후 2:10
Last Edited By	
날짜	

nslookup(dns서버를 알고 있어야 함)과 ping 이 비슷함 .... ip 주소 나눔

4조 실습



작동 되고 있음은 확인됨

```
su
mkdir -p /usr/local/dns
cd /usr/local/dns
wget https://downloads.isc.org/isc/bind9/9.16.34/bind-9.16.34.tar.xz
tar -xvf bind-9.16.34.tar.xz
cd bind-9.16.34
```

root@yoojung-Samsung-DeskTop-System:/usr/local/dns/bind-9.16.34# ./configure --help  
'configure' configures BIND 9.16 to adapt to many kinds of systems.

Usage: ./configure [OPTION]... [VAR=VALUE]...

To assign environment variables (e.g., CC, CFLAGS...), specify them as VAR=VALUE. See below for descriptions of some of the useful variables.

Defaults for the options are specified in brackets.

Configuration:

-h, --help display this help and exit  
--help=short display options specific to this package  
--help=recursive display the short help of all the included packages  
-V, --version display version information and exit  
-q, --quiet, --silent do not print `checking ...` messages `--cache-file=FILE` cache test results  
`in FILE [disabled]` `-C, --config-cache` alias for `--cache-file=config.cache`  
-n, --no-create do not create output files  
--srcdir=DIR find the sources in DIR [configure dir or `..']

Installation directories:

--prefix=PREFIX install architecture-independent files in PREFIX  
[`/usr/local`]  
--exec-prefix=EPREFIX install architecture-dependent files in EPREFIX  
[PREFIX]

By default, `make install` will install all the files in `/usr/local/bin`, `/usr/local/lib` etc. You can specify an installation prefix other than `/usr/local` using `--prefix'`, for instance `--prefix=$HOME`.

For better control, use the options below.

Fine tuning of the installation directories:

--bindir=DIR user executables [EPREFIX/bin]  
--sbindir=DIR system admin executables [EPREFIX/sbin]  
--libexecdir=DIR program executables [EPREFIX/libexec]  
--sysconfdir=DIR read-only single-machine data [PREFIX/etc]  
--sharedstatedir=DIR modifiable architecture-independent data [PREFIX/com]  
--localstatedir=DIR modifiable single-machine data [PREFIX/var]  
--libdir=DIR object code libraries [EPREFIX/lib]  
--includedir=DIR C header files [PREFIX/include]  
--oldincludedir=DIR C header files for non-gcc [`/usr/include`]  
--datarootdir=DIR read-only arch.-independent data root [PREFIX/share]  
--datadir=DIR read-only architecture-independent data [DATAROOTDIR]  
--infodir=DIR info documentation [DATAROOTDIR/info]  
--localedir=DIR locale-dependent data [DATAROOTDIR/locale]  
--mandir=DIR man documentation [DATAROOTDIR/man]  
--docdir=DIR documentation root [DATAROOTDIR/doc/bind]  
--htmldir=DIR html documentation [DOCDIR]

--dvidir=DIR        dvi documentation [DOCDIR]

--pdfdir=DIR        pdf documentation [DOCDIR]

--psdir=DIR        ps documentation [DOCDIR]

System types:

--build=BUILD    configure for building on BUILD [guessed]

--host=HOST      cross-compile to build programs to run on HOST [BUILD]

Optional Features:

--disable-option-checking ignore unrecognized --enable/--with options

--disable-FEATURE    do not include FEATURE (same as --enable-FEATURE=no)

--enable-FEATURE[=ARG] include FEATURE [ARG=yes]

--disable-maintainer-mode

disable make rules and dependencies not useful (and sometimes confusing) to the casual installer

--disable-largefile    omit support for large files

--enable-shared[=PKGS] build shared libraries [default=yes]

--enable-static[=PKGS] build static libraries [default=yes]

--enable-fast-install[=PKGS]

optimize for fast installation [default=yes]

--disable-libtool-lock avoid locking (might break parallel builds)

--enable-buffer-useinline

define ISC\_BUFFER\_USEINLINE when compiling

[default=yes]

--enable-warn-shadow    turn on -Wshadow when compiling

--enable-warn-error    turn on -Werror when compiling

--enable-developer    enable developer build settings

--enable-fuzzing=<afl|libfuzzer>

Enable fuzzing using American Fuzzy Lop or libFuzzer

(default=no)

--enable-kqueue        use BSD kqueue when available [default=yes]

--enable-epoll        use Linux epoll when available [default=auto]

--enable-devpoll       use /dev/poll when available [default=yes]

--disable-geoip        support GeoIP2 geolocation ACLs if available

[default=yes]

--enable-pthread-rwlock use pthread rwlock instead of internal rwlock implementation

--enable-fips-mode    enable FIPS mode in OpenSSL library [default=no]

--enable-native-pkcs11 use native PKCS11 for public-key crypto [default=no]

--enable-backtrace    log stack backtrace on abort [default=yes]

--enable-symtable    use internal symbol table for backtrace

[all|minimal(default)|none]

--disable-tcp-fastopen disable TCP Fast Open support [default=yes]

--disable-chroot      disable chroot

--disable-linux-caps    disable Linux capabilities

--enable-fixed-rrset    enable fixed rrset ordering [default=no]  
 --enable-dnstatap      enable dnstatap support (requires fstrm, protobuf-c)  
 --enable-querytrace    enable very verbose query trace logging [default=no]  
 --enable-auto-validation  
 turn on DNSSEC validation by default, using the IANA  
 root key [default=yes]  
 --enable-dnsrps-dl     DNS Response Policy Service delayed link  
 [default=\$librpz\_dl]  
 --enable-dnsrps        enable DNS Response Policy Service API  
 --enable-full-report    report values of all configure options

Optional Packages:

--with-PACKAGE[=ARG]   use PACKAGE [ARG=yes]  
 --without-PACKAGE      do not use PACKAGE (same as --with-PACKAGE=no)  
 --with-pic[=PKGS]      try to use only PIC/non-PIC objects [default=use  
 both]  
 --with-aix-soname=aix|svr4|both  
 shared library versioning (aka "SONAME") variant to  
 provide on AIX, [default=aix].  
 --with-gnu-ld          assume the C compiler uses GNU ld [default=no]  
 --with-sysroot[=DIR]   Search for dependent libraries within DIR (or the  
 compiler's sysroot if not specified).  
 --with-python=PATH     specify path to Python interpreter  
 --with-python-install-dir=PATH  
 installation directory for Python modules  
 --with-geoip2          deprecated, use --with-maxminddb  
 --with-maxminddb=PATH   Build with MaxMind GeoIP2 support (auto|yes|no|path)  
 [default=auto]  
 --with-locktype=ARG    Specify mutex lock type (adaptive or standard)  
 --with-libtool         use GNU libtool  
 --with-openssl=DIR     root of the OpenSSL directory  
 --with-pkcs11=PATH     Build with PKCS11 support [no|path] (PATH is for the  
 PKCS11 provider)  
 --with-gssapi=PATH[/path/krb5-config  
 Specify path for system-supplied GSSAPI  
 [default=auto]  
 --with-lmdb=PATH       build with LMDB library [yes|no|path]  
 --with-libxml2         build with libxml2 library [yes|no|auto] (default is  
 auto)  
 --with-libjson         deprecated, use --with-json-c  
 --with-json-c          build with json-c library [yes|no|detect] (default  
 is detect)  
 --with-zlib            build with zlib for HTTP compression [default=yes]  
 --with-purify=PATH     use Rational purify

--with-gperftools-profiler  
 use gperftools CPU profiler  
 --with-readline=LIBSPEC specify readline library [default auto]  
 --with-protobuf-c=path Path where protobuf-c is installed, for dnstap  
 --with-libfstrm=path Path where libfstrm is installed, for dnstap  
 --with-libidn2=PATH enable IDN support using GNU libidn2  
 [yes|no(default)|path]  
 --with-cmocka=detect enable CMocka based tests (default is detect)  
 --with-tuning=ARG Specify server tuning (default or small)  
 --with-dlopen=ARG support dynamically loadable DLZ and DYNDB drivers  
 --with-dnsrps-libname DNSRPS provider library name (librpz.so)  
 --with-dnsrps-dir path to DNSRPS provider library  
 --with-dlz-postgres=PATH  
 Build with Postgres DLZ driver [yes|no|path].  
 (Required to use Postgres with DLZ)  
 --with-dlz-mysql=PATH Build with MySQL DLZ driver [yes|no|path]. (Required  
 to use MySQL with DLZ)  
 --with-dlz-bdb=PATH Build with Berkeley DB DLZ driver [yes|no|path].  
 (Required to use Berkeley DB with DLZ)  
 --with-dlz-filesystem=ARG  
 Build with filesystem DLZ driver [yes|no]. (Required  
 to use file system driver with DLZ)  
 --with-dlz-ldap=PATH Build with LDAP DLZ driver [yes|no|path]. (Required  
 to use LDAP with DLZ)  
 --with-dlz-odbc=PATH Build with ODBC DLZ driver [yes|no|path]. (Required  
 to use ODBC with DLZ)  
 --with-dlz-stub=ARG Build with stub DLZ driver [yes|no]. (Required to  
 use stub driver with DLZ)  
 --with-make-clean run "make clean" at end of configure [yes|no]

Some influential environment variables:

CC C compiler command

CFLAGS C compiler flags

LDLFLAGS linker flags, e.g. -L<lib dir> if you have libraries in a  
 nonstandard directory <lib dir>

LIBS libraries to pass to the linker, e.g. -l<library>

CPPFLAGS (Objective) C/C++ preprocessor flags, e.g. -I<include dir> if  
 you have headers in a nonstandard directory <include dir>

CPP C preprocessor

LT\_SYS\_LIBRARY\_PATH

User-defined run-time library search path.

PKG\_CONFIG path to pkg-config utility

PKG\_CONFIG\_PATH

directories to add to pkg-config's search path

PKG\_CONFIG\_LIBDIR  
path overriding pkg-config's built-in search path  
PYTHON path to python executable  
MAXMINDDDB\_CFLAGS  
C compiler flags for MAXMINDDDB, overriding pkg-config  
MAXMINDDDB\_LIBS  
linker flags for MAXMINDDDB, overriding pkg-config  
MAXMINDDDB\_PREFIX  
value of prefix for libmaxminddb, overriding pkg-config  
LIBUV\_CFLAGS  
C compiler flags for LIBUV, overriding pkg-config  
LIBUV\_LIBS linker flags for LIBUV, overriding pkg-config  
OPENSSL\_CFLAGS  
C compiler flags for OPENSSL, overriding pkg-config  
OPENSSL\_LIBS  
linker flags for OPENSSL, overriding pkg-config  
LIBXML2\_CFLAGS  
C compiler flags for LIBXML2, overriding pkg-config  
LIBXML2\_LIBS  
linker flags for LIBXML2, overriding pkg-config  
JSON\_C\_CFLAGS  
C compiler flags for JSON\_C, overriding pkg-config  
JSON\_C\_LIBS linker flags for JSON\_C, overriding pkg-config  
ZLIB\_CFLAGS C compiler flags for ZLIB, overriding pkg-config  
ZLIB\_LIBS linker flags for ZLIB, overriding pkg-config  
SPHINX\_BUILD  
path to sphinx-build binary used to build the documentation  
LIBIDN2\_CFLAGS  
C compiler flags for LIBIDN2, overriding pkg-config  
LIBIDN2\_LIBS  
linker flags for LIBIDN2, overriding pkg-config  
CMOCKA\_CFLAGS  
C compiler flags for CMOCKA, overriding pkg-config  
CMOCKA\_LIBS linker flags for CMOCKA, overriding pkg-config  
KYUA path to kyua execution engine

Use these variables to override the choices made by `configure` or to help it to find libraries and programs with nonstandard names/locations.

Report bugs to [https://gitlab.isc.org/isc-projects/bind9/-/issues/new?issuable\\_template=Bug](https://gitlab.isc.org/isc-projects/bind9/-/issues/new?issuable_template=Bug).  
BIND home page: <https://www.isc.org/downloads/>.

```
./configure
```

에러발생 ERROR: no acceptable C compiler found in \$PATH

<https://stackoverflow.com/questions/19816275/no-acceptable-c-compiler-found-in-path-when-installing-python> -> apt install build-essential in other terminal

```
./configure
```

에러 -> ERROR: error: The pkg-config script could not be found or is too old.

try : sudo apt install pkg-config

(<https://webcreate.tistory.com/entry/pkgconfig-command-not-found-에러-해결방법>)

try again

```
./configure
```

-> ERROR : error: Python >= 2.7 or >= 3.2 and the PLY package are required for dnsmgr and other Python-based tools. PLY may be available from your OS package manager as python-ply or python3-ply; it can also be installed via pip. To build without Python/PLY, use --without-python.

then try :

```
./configure --without-python
```

(<https://comp.protocols.dns.bind.narkive.com/g2XMLyHb/bind-9-14-configure-error>) -> ERROR: error: libuv not found

then try:

```
apt-get update
sudo apt-get install -y libuv1-dev
```

then ERROR: error: OpenSSL/LibreSSL not found  
See `config.log' for more details

then try:

```
apt install openssl
```

still same error ->  
then try & result:

```
sudo apt-get install libssl-dev
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다... 완료
상태 정보를 읽는 중입니다... 완료
The following additional packages will be installed:
libssl3
제안하는 패키지:
libssl-doc
다음 새 패키지를 설치할 것입니다:
libssl-dev
다음 패키지를 업그레이드할 것입니다:
libssl3
1개 업그레이드, 1개 새로 설치, 0개 제거 및 190개 업그레이드 안 함.
```

```
./configure --without-python
```

-> new ERROR: error: sys/capability.h header is required for Linux capabilities support. Either install libcap or use --disable-linux-caps.

then try:

```
sudo apt-get install git libcap-dev
```

(<https://www.oreilly.com/library/view/security-with-go/9781788627917/f8829d03-9243-42c0-b748-50377c96d293.xhtml>)

still same error

then try :

```
sudo apt-get install libcap-dev
```



**SUCCESS to ./configure --without-python !!!!!!!!!!!!!!!**  
**(Configuration summary:**

**Optional features enabled:**

**Print backtrace on crash (--enable-backtrace)**

**Use symbol table for backtrace, named only (--enable-symtable)**

**DNSSEC validation active by default (--enable-auto-validation)**

**Dynamically loadable zone (DLZ) drivers:**  
**None**

**Features disabled or unavailable on this platform:**

**Small-system tuning (--with-tuning)**

**Allow 'dnstap' packet logging (--enable-dnstap)**

**GeoIP2 access control (--enable-geoip)**

**GSS-API (--with-gssapi)**

**DNS Response Policy Service interface (--enable-dnsrps)**

**Allow 'fixed' rrset-order (--enable-fixed-rrset)**

**Using PKCS#11 for Public-Key Cryptography (--with-native-pkcs11)**

**Very verbose query trace logging (--enable-querytrace)**

**Use GNU libtool (--with-libtool)**

**CMocka Unit Testing Framework (--with-cmocka)**

**Python tools (--with-python)**

**XML statistics (--with-libxml2)**

**JSON statistics (--with-json-c)**

**HTTP zlib compression (--with-zlib)**

**LMDB database to store configuration for 'addzone' zones (--with-lmdb)**

**IDN support (--with-libidn2)**

**Configured paths:**

**prefix: /usr/local**

**sysconfdir: \${prefix}/etc**

**localstatedir: \${prefix}/var**

**Compiler: gcc**

**gcc (Ubuntu 11.3.0-1ubuntu1~22.04) 11.3.0**

Copyright (C) 2021 Free Software Foundation, Inc.

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

For more detail, use --enable-full-report.

)

```
make ->success
make install ->success
```

**!!!!!!setting named.conf !!!!!**

**???no named.conf so.. without python ????**

**python-> sudo apt-get install -y python3-ply**

**/etc/bind/named.conf -> different**

**now try package install // success**

**package-> named.conf files ;**

named.conf , named.conf.options, named.conf.default-zones, named.conf.local

root@yoojung-Samsung-DeskTop-System:/etc/bind# cat named.conf.options

options {

directory "/var/cache/bind";

// If there is a firewall between you and nameservers you want

// to talk to, you may need to fix the firewall to allow multiple

// ports to talk. See <<http://www.kb.cert.org/vuls/id/800113>>

// If your ISP provided one or more IP addresses for stable

// nameservers, you probably want to use them as forwarders.

// Uncomment the following block, and insert the addresses replacing

// the all-0's placeholder.

// forwarders {

// 0.0.0.0;

// };

//=====

// If BIND logs error messages about the root key being expired,

// you will need to update your keys. See <<https://www.isc.org/bind-keys>>

//=====

dnssec-validation auto;

listen-on-v6 { any; };

};

root@yoojung-Samsung-DeskTop-System:/etc/bind# cat named.conf.default-zones

```
// prime the server with knowledge of the root servers
zone "." {
    type hint;
    file "/usr/share/dns/root.hints";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
    type master;
    file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
    type master;
    file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
    type master;
    file "/etc/bind/db.255";
};

zone "bind9.kr" {
    type master;
    file "/etc/bind/db.bind9.zone";
};

try copy and paste but cp mistake
```

Is it possible to undo rm?

Short answer: You can't. rm removes files blindly, with no concept of 'trash'.

now finally make named.conf::

```
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk. See <http://www.kb.cert.org/vuls/id/800113>

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.
```

```
// forwarders {
// 0.0.0.0;
// };

//=====
// If BIND logs error messages about the root key being expired,
// you will need to update your keys. See <https://www.isc.org/bind-keys>
//=====
dnssec-validation auto;

listen-on-v6 { any; };

};
// prime the server with knowledge of the root servers
zone "." {
type hint;
file "/usr/share/dns/root.hints";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
type master;
file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
type master;
file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
type master;
file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
type master;
file "/etc/bind/db.255";
};

zone " bind9.kr " {
type master;
file "/etc/bind/db.bind9.zone";
};

zone " naver.com : IN {
type master;
file "naver.com.zone";
};
```

```
root@yoojung-Samsung-DeskTop-System:~# cd /var/named/
bash: cd: /var/named/: No such file or directory
root@yoojung-Samsung-DeskTop-System:~# cd /var
root@yoojung-Samsung-DeskTop-System:/var# ls
backups crash local log metrics run spool
cache lib lock mail opt snap tmp
root@yoojung-Samsung-DeskTop-System:/var# mkdir named
root@yoojung-Samsung-DeskTop-System:/var# cd named
root@yoojung-Samsung-DeskTop-System:/var/named# vi naver.com.zone
root@yoojung-Samsung-DeskTop-System:/var/named# cat naver.com.zone
$TTL 3H
```

```
@ SOA @ root. (2 1D 1H 1W 1H)
IN NS @
IN A 8.8.8.8
www IN A 8.8.8.8
```

- making new named.conf

```
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, BEFORE you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

```
root@yoojung-Samsung-DeskTop-System:/var# cd named
root@yoojung-Samsung-DeskTop-System:/var/named# ls
naver.com.zone
root@yoojung-Samsung-DeskTop-System:/var/named# vi named.conf
root@yoojung-Samsung-DeskTop-System:/var/named# ufw status
Status: active
```

To	Action	From
23/tcp	ALLOW	Anywhere
53/tcp	ALLOW	Anywhere
53/udp	ALLOW	Anywhere
23/tcp (v6)	ALLOW	Anywhere (v6)
53/tcp (v6)	ALLOW	Anywhere (v6)
53/udp (v6)	ALLOW	Anywhere (v6)

```
root@yoojung-Samsung-DeskTop-System:/var/named# sudo systemctl restart bind9
root@yoojung-Samsung-DeskTop-System:/var/named# sudo systemctl status bind9
• named.service - BIND Domain Name Server
Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset:>
Active: active (running) since Tue 2022-11-08 10:54:32 KST; 11s ago
Docs: man:named(8)
```

```
//////////NOW WE TEST DNS//////////
root@yoojung-Samsung-DeskTop-System:~# ps -ef | grep namedroot
root      6178      5457  0 10:59 pts/1    00:00:00 grep --color=auto namedroot
yoojung@yoojung-Samsung-DeskTop-System:~$ su
Password:
root@yoojung-Samsung-DeskTop-System:/home/yoojung# sudo apt install resolvconf
root@yoojung-Samsung-DeskTop-System:/home/yoojung# sudo vi /etc/resolvconf/resolv.conf.d/head
```

**Dynamic resolv.conf(5) file for glibc resolver(3)  
generated by resolvconf(8)**

**DO NOT EDIT THIS FILE BY HAND -- YOUR  
CHANGES WILL BE OVERWRITTEN**

**127.0.0.53 is the systemd-resolved stub resolver.**

**run "systemd-resolve --status" to see details  
about the actual nameservers.**

nameserver 14.53.90.72

root@yoojung-Samsung-DeskTop-System:/home/yoojung# cat /etc/resolv.conf

**Dynamic resolv.conf(5) file for glibc resolver(3)  
generated by resolvconf(8)**

**DO NOT EDIT THIS FILE BY HAND -- YOUR  
CHANGES WILL BE OVERWRITTEN**

**127.0.0.53 is the systemd-resolved stub resolver.**

**run "systemd-resolve --status" to see details  
about the actual nameservers.**

nameserver 127.0.0.53

root@yoojung-Samsung-DeskTop-System:/home/yoojung# vi /etc/resolv.conf

add nameserver 14.53.90.72

ERROR we can't use any website

root@yoojung-Samsung-DeskTop-System:/home/yoojung# nslookup

naver.com

Server: ::1

Address: ::1#53

Non-authoritative answer:

Name: naver.com

Address: 223.130.200.107

Name: naver.com

Address: 223.130.200.104

Name: naver.com

Address: 223.130.195.95

Name: naver.com

Address: 223.130.195.200

google.com

Server: ::1

Address: ::1#53

Non-authoritative answer:

Name: google.com

Address: 142.250.207.78

Name: google.com

Address: 2404:6800:4005:820::200e

```
root@yoojung-Samsung-DeskTop-System:~# cd /usr/local/dns/sbin
root@yoojung-Samsung-DeskTop-System:/usr/local/dns/sbin# ll
total 135312
drwxr-xr-x 2 root root      4096 11월  2 14:26 ./
drwxr-xr-x 9 root root      4096 11월  2 14:26 ../
-rwxr-xr-x 1 root root    2817320 11월  2 14:26 ddns-confgen*
-rwxr-xr-x 1 root root    9650184 11월  2 14:26 dnssec-cds*
-rwxr-xr-x 1 root root    9619840 11월  2 14:26 dnssec-dsfromkey*
-rwxr-xr-x 1 root root    9617248 11월  2 14:26 dnssec-importkey*
-rwxr-xr-x 1 root root    9620368 11월  2 14:26 dnssec-keyfromlabel*
-rwxr-xr-x 1 root root   10082760 11월  2 14:26 dnssec-keygen*
-rwxr-xr-x 1 root root    9593264 11월  2 14:26 dnssec-revoke*
-rwxr-xr-x 1 root root    9626200 11월  2 14:26 dnssec-settime*
-rwxr-xr-x 1 root root    9822032 11월  2 14:26 dnssec-signzone*
-rwxr-xr-x 1 root root    9636560 11월  2 14:26 dnssec-verify*
-rwxr-xr-x 1 root root   13481144 11월  2 14:26 named*
-rwxr-xr-x 1 root root   10362688 11월  2 14:26 named-checkconf*
-rwxr-xr-x 1 root root    9636808 11월  2 14:26 named-checkzone*
lrwxrwxrwx 1 root root      15 11월  2 14:26 named-compilezone -> named-checkzone*
-rwxr-xr-x 1 root root    9560104 11월  2 14:26 named-journalprint*
-rwxr-xr-x 1 root root    529816 11월  2 14:26 nsec3hash*
-rwxr-xr-x 1 root root   2046856 11월  2 14:26 rndc*
-rwxr-xr-x 1 root root   2816136 11월  2 14:26 rndc-confgen*
lrwxrwxrwx 1 root root      12 11월  2 14:26 tsig-keygen -> ddns-confgen*
```

```
root@yoojung-Samsung-DeskTop-System:/usr/local/dns/sbin# ./named-checkconf
/usr/local/dns/etc/named.conf:61: '{' expected near ''
```

make temp file

```
//
// named.caching-nameserver.conf
//
// Provided by Red Hat caching-nameserver package to configure the
// ISC BIND named(8) DNS server as a caching only nameserver
// (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//
// DO NOT EDIT THIS FILE - use system-config-bind or an editor
// to create named.conf - edits to this file will be lost on
// caching-nameserver package upgrade.
//

options {
listen-on port 53 { any; };
listen-on-v6 port 53 { any; };
directory    "/var/named";
dump-file     "/var/named/data/cache_dump.db";
statistics-file "/var/named/data/named_stats.txt";
memstatistics-file "/var/named/data/named_mem_stats.txt";

    // Those options should be used carefully because they disable port
    // randomization
    query-source    port 53;
    query-source-v6 port 53;

allow-query    { any; };
allow-query-cache { any; };

```

```
};
logging {
channel default_debug {
file "data/named.run";
severity dynamic;
};
};
view localhost_resolver {
match-clients        { any; };
match-destinations { any; };
recursion no;
include "/etc/named.rfc1912.zones";
};
root@yoojung-Samsung-DeskTop-System:/usr/local/dns/sbin# cp -f temp ./named-checkconf
```

DNS 서버가 설치됨을 확인했으나, 테스트는 실패함






## 서버의 정의

DNS 서비스는 전 세계에 배포된 서비스로서, `www.example.com`과 같이 사람이 읽을 수 있는 이름을 `192.0.2.1`과 같은 숫자 IP 주소로 변환하여 컴퓨터가 서로 통신할 수 있도록 합니다. 인터넷의 DNS 시스템은 이름과 숫자 간의 매핑을 관리하여 마치 전화번호부와 같은 기능을 합니다. DNS 서버는 이름을 IP 주소로 변환하여 도메인 이름을 웹 브라우저에 입력할 때 최종 사용자를 어떤 서버에 연결할 것인지를 제어합니다. 이 요청을 **쿼리**라고 부릅니다.

### 포트번호? udp 53, tcp 53

#### 네임서버(DNS)의 port사용

name server 일반 적으로 dns 쿼리시 udp53번 포트를 사용하며, 네임서버간에 자료 전송시 tcp 53번 포트를 이용하여 서비스 하게 됩니다. 즉, udp53번만 열어도 서비스 하는데 지장이 없으며, 네임서버를 1, 2차로 마스터 슬레이브로 구성하셨다면,

 <https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=hellowkorea&logNo=50004233485>



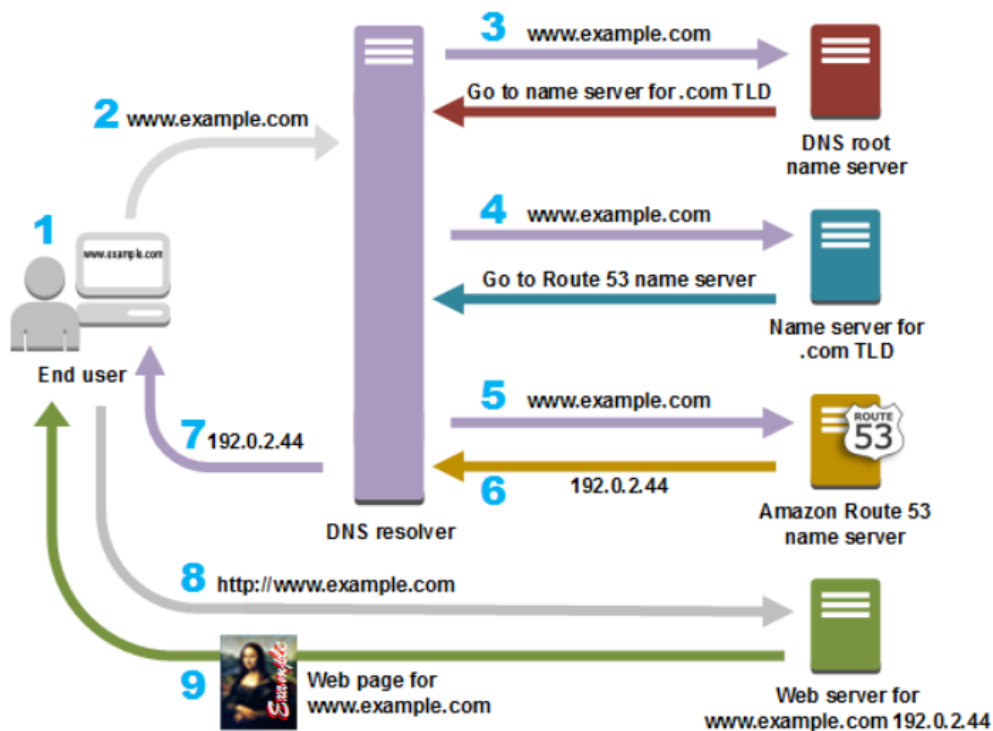
## 1. 역할

### ! DNS(Domain Name Syetem)란?

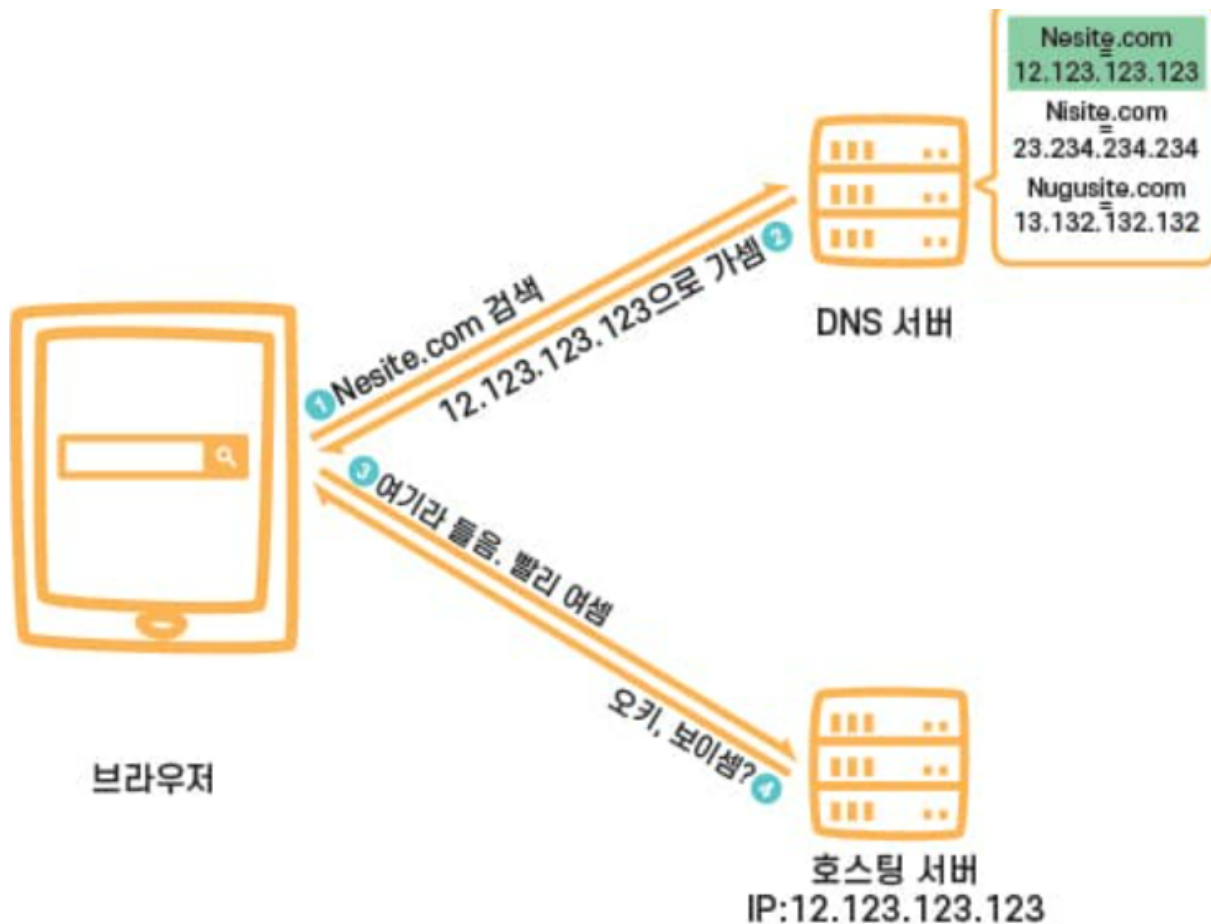
- 웹사이트에 접속 할 때 우리는 외우기 어려운 IP 주소 대신 도메인 이름을 사용한다.
- 도메인 이름을 사용했을 때 입력한 도메인을 실제 네트워크상에서 사용하는 IP 주소로 바꾸고 해당 IP 주소로 접속하는 과정이 필요하다.
- 이러한 과정, 전체 시스템을 DNS(도메인 네임 시스템)라고 한다.
- 이러한 시스템은 전세계적으로 약속된 규칙을 공유한다.
- 상위 기관에서 인증된 기관에게 도메인을 생성하거나 IP 주소로 변경할 수 있는 '권한'을 부여한다.
- DNS는 이처럼 상위 기관과 하위 기관과 같은 '계층 구조'를 가지는 분산 데이터베이스 구조를 가진다.

*192.0.2.1과 같은 숫자 IP 주소로 변환*

## 구성



1. 사용자가 웹 브라우저를 열어 주소 표시줄에 `www.example.com`을 입력하고 Enter 키를 누릅니다.
2. `www.example.com`에 대한 요청은 일반적으로 케이블 인터넷 공급업체, DSL 광대역 공급업체 또는 기업 네트워크 같은 인터넷 서비스 제공업체(ISP)가 관리하는 DNS 해석기로 라우팅됩니다.
3. ISP의 DNS 해석기는 `www.example.com`에 대한 요청을 DNS 루트 이름 서버에 전달합니다.
4. ISP의 DNS 해석기는 `www.example.com`에 대한 요청을 이번에는 `.com` 도메인의 TLD 이름 서버 중 하나에 다시 전달합니다. `.com` 도메인의 이름 서버는 `example.com` 도메인과 연관된 4개의 Amazon Route 53 이름 서버의 이름을 사용하여 요청에 응답합니다.
5. ISP의 DNS 해석기는 Amazon Route 53 이름 서버 하나를 선택해 `www.example.com`에 대한 요청을 해당 이름 서버에 전달합니다.
6. Amazon Route 53 이름 서버는 `example.com` 호스팅 영역에서 `www.example.com` 레코드를 찾아 웹 서버의 IP 주소 `192.0.2.44` 등 연관된 값을 받고 이 IP 주소를 DNS 해석기로 반환합니다.
7. ISP의 DNS 해석기가 마침내 사용자에게 필요한 IP 주소를 확보하게 됩니다. 해석기는 이 값을 웹 브라우저로 반환합니다. 또한, DNS 해석기는 다음에 누군가가 `example.com`을 탐색할 때 좀 더 빠르게 응답할 수 있도록 사용자가 지정하는 일정 기간 `example.com`의 IP 주소를 캐싱(저장)합니다. 자세한 내용은 Time to Live(TTL)를 참조하세요.
8. 웹 브라우저는 DNS 해석기로부터 얻은 IP 주소로 `www.example.com`에 대한 요청을 전송합니다. 여기가 콘텐츠가 있는 곳으로, 예를 들어 웹 사이트 엔드포인트로 구성된 Amazon S3 버킷 또는 Amazon EC2 인스턴스에서 실행되는 웹 서버입니다.
9. `192.0.2.44`에 있는 웹 서버 또는 그 밖의 리소스는 `www.example.com`의 웹 페이지를 웹 브라우저로 반환하고, 웹 브라우저는 이 페이지를 표시합니다.



[https://gentlysallim.com/dns란-뭐고-네임서버란-뭔지-개념정리/#!/lightbox\[257f85eddef0a477cd5\]/0](https://gentlysallim.com/dns란-뭐고-네임서버란-뭔지-개념정리/#!/lightbox[257f85eddef0a477cd5]/0)

## DNS Query(쿼리)란?

DNS 클라이언트와 DNS 서버는 DNS 쿼리를 교환한다. DNS 쿼리는 Recursive(재귀적) 또는 Iterative(반복적)으로 구분된다. 아래의 용어가 위 동작 과정에 이해를 방해할 수도 있기에 추가하였습니다. 단순한 개념으로 아래와 같은 방식으로 요청과 응답을 한다고 이해하시면 됩니다.

## Recursive Query (재귀적 질의)

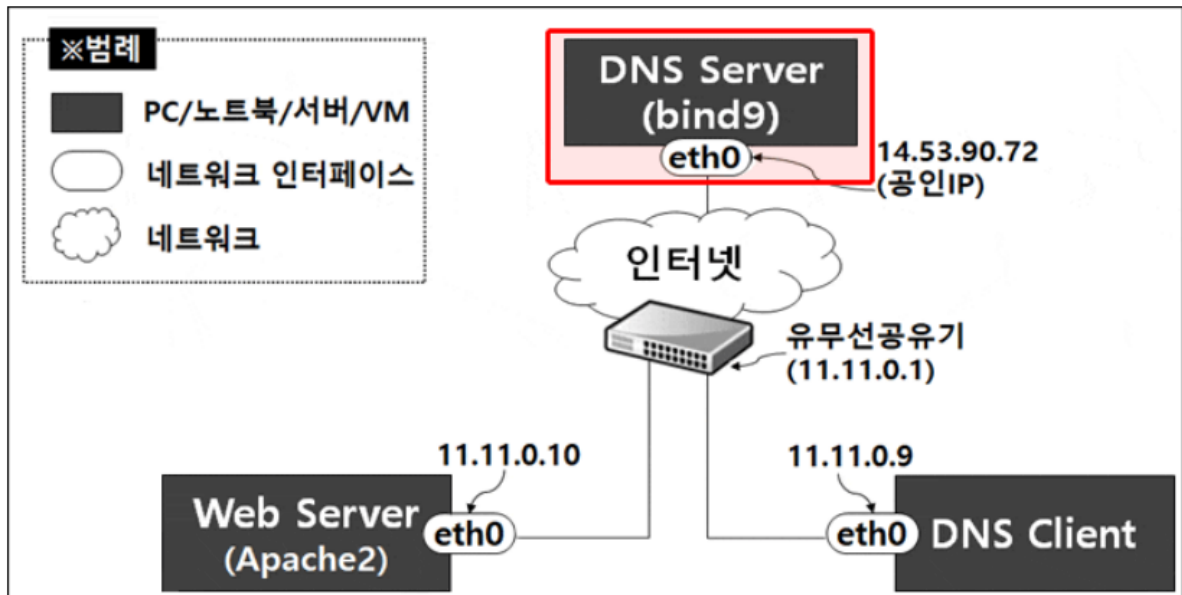
결과물(IP 주소)를 돌려주는 작업이다. (결과적으로 Recursive 서버가 Recursive 쿼리를 웹 브라우저 등에게 돌려주는 역할을 한다.) Recursive나 Iterative나 가가가가 아닌가? 뭐지? 엄청 혼란스러웠다... Recursive 쿼리를 받은 Recursive 서버는 Iterative 하게 권한 있는 네임 서버로 Iterative 쿼리를 보내서 결과적으로 IP 주소를 찾게 되고 해당 결과물을 응답한다. 사전적 의미로써 재귀로, 응답을 돌려주는 쿼리라고 생각하자.

## Iterative Query (반복적 질의)

Recursive DNS 서버가 다른 DNS 서버에게 쿼리를 보내어 응답을 요청하는 작업이다. Recursive 서버가 권한 있는 네임 서버들에게 반복적으로 쿼리를 보내서 결과물(IP 주소)을 알아낸다. Recursive 서버에 이미 IP 주소가 캐시 되어있다면 이 과정은 건너 뛴다.

출처: <https://hanamon.kr/dns란-도메인-네임-시스템-개념부터-작동-방식까지/>

# 1.배치위치



bind9으로 Ubuntu 20.04 LTS에 DNS 서버 구축 및 동작 테스트하기

본 포스팅의 실습구조를 살펴보면 아래 [그림 1] 과 같이, 일단 Ubuntu 20.04 LTS 가 설치된 PC(혹은 VM이나 서버)를 3대 필요하다. 준비된 1대의 PC에는 네트워크 인터페이스 'eth0'에 공인IP주소 '14.53.90.72'가 할당되어 인터넷에 연결된 상태

[https://m.blog.naver.com/love\\_tolty/222690840923](https://m.blog.naver.com/love_tolty/222690840923)

**bind9 실습편(1)**  
**bind9으로 Ubuntu 20.04에 DNS서버**

## 설치방법

- 준비사항

DNS서버는 애플리케이션 프로토콜중의 하나인 DNS 프로토콜을 사용한다. DNS 프로토콜은 네트워크를 경유하여 DNS 구현요소 간에 DNS 질의(DNS Query)와 응답(DNS Response)을 수행하기 위한 클라이언트/서버 모델의 애플리케이션 프로토콜이다.

DNS 프로토콜은 TCP 및 UDP 포트번호 53번을 사용한다. DNS 질의의 대부분은 UDP 포트 53번을 사용하여 질의와 응답이 이루어진다. 그러나 UDP 헤더 이후의 DNS 헤더를 포함한 DNS 메시지 영역의 길이가 512 바이트를 초과하는 경우에는 TCP 53번 포트를 사용하는 TCP 연결을 통한 DNS 질의와 응답이 이루어지는 매커니즘이 존재한다. 또한 동일한 도메인 존을 가지고 있는 네임서버간의 도메인 존 데이터 송수신을 위한 존 트랜스퍼(Zone Transfer)를 수행하는 경우에는 많은 데이터 전송요청이 이루어지므로 TCP 53번을 사용한다.

고로 방화벽 및 공유기에서 TCP/UDP 53번 포트를 개방한다.

iptables를 활성화 해둔 경우 아래의 순서로 포트 개방

#### BASH

```
iptables -A INPUT -p udp -m state --state NEW --dport 53 -j ACCEPT
iptables -A INPUT -p tcp -m state --state NEW --dport 53 -j ACCEPT
service iptables save
service iptables restart
```

- 우분투 포트 개방 : <https://pinggoopark.tistory.com/158>

```
방화벽 정지 후 상태 확인
ufw disable
ufw status
방화벽 실행 후 상태 확인
ufw enable
ufw status
22번 포트 허용하고 리스트 확인
ufw allow 22
ufw status
22번 포트 막고 리스트 확인하기
ufw deny 22
ufw status
방화벽 설정된 포트 삭제하고 상태확인하기
ufw delete deny 22
ufw status
방화벽 끄고 방화벽 상태 확인하기
ufw disable
ufw status
우분투에서 특정 포트의 방화벽을 여는 명령어는 다음과 같습니다.
```

#### 우분투 포트 방화벽 열기

우분투에서 특정 포트의 방화벽을 여는 명령어는 다음과 같습니다. 만약 10000번 포트를 열고 싶다면, \$ sudo iptables -I INPUT 1 -p tcp --dport 10000 -j ACCEPT 로 명령해주면 됩니다.

<https://bskyvision.com/entry/%EC%9A%B0%EB%B6%84%ED%88%AC-%ED%8F%AC%ED%8A%B8-%EB%B0%A9%ED%99%94%EB%B2%BD-%EC%97%B4%EA%B8%B0>

만약 10000번 포트를 열고 싶다면,

**\$ sudo iptables -I INPUT 1 -p tcp --dport 10000 -j ACCEPT**

로 명령해주면 됩니다.



`sudo ufw allow 53/tcp, sudo ufw allow 53/udp` 로 했음!

## 1 패키지로 설치

### BIND9 설치해야 함

#### DNS 네임 서버를 직접 구축해보자! (feat. BIND9)

오늘은 해당 기능을 활용해서 간단하게 자체 DNS 네임 서버를 구축해보는 방법을 소개 드리겠습니다! 서버 내부에서만 동작하는 아주 간단한 예제이지만, DNS를 이해하는 데에 조금이나마 도움이 되셨으면 좋겠습니다. nslookup 명령어는 도메인

🔗 <https://tech.ktcloud.com/66>

```
nslookup cloud.kt.com
63.1
63.1#53
```

- `sudo apt-get install bind9`
- `/etc/resolv.conf` 파일 수정

우리는 이 서버의 네임 서버가 자기 자신이 되도록 해야 하기 때문에, 아래와 같이 `nameserver` 값에 `127.0.0.1` 을 추가해 봅시다.

```
nameserver 127.0.0.53
nameserver 127.0.0.1
options edns0
search csa77fcloud.internal
```

- `cd /etc/bind` 명령어를 통해 `bind` 디렉토리로 이동한 후, `vim named.conf.default-zones` 명령어를 통해 파일을 열어봅시다.

```
cd /etc/bind 명령어를 통해 bind 디렉토리로 이동한 후,  
vim named.conf.default-zones 명령어를 통해 파일을 열어봅시다.
```

이 파일에는 추가된 zone에 대한 정보가 담겨있습니다. 여기서 **zone**이란 **도메인의 정보**를 말합니다.

우리는 지금 *bind9.kr* 이라는 도메인을 추가하고자 하므로, 해당 파일 맨 아래에 이와 같이 입력하였습니다.

```
zone "bind9.kr" {  
    type master;  
    file "/etc/bind/db.bind9.zone";  
};
```

- *type*: master / slave 중 어느 타입인지 설정합니다. 이번에는 master-slave 구조를 만들지 않고 단일 서버로만 테스트 해 볼 예정이기 때문에 master로 설정합니다.
- *file*: zone file의 위치를 설정합니다.

- *bind9.kr* 이라는 도메인에 대한 zone 파일을 설정

*cd /etc/bind* 명령어를 통해 *bind* 디렉토리로 이동합니다.

이 디렉토리 안에 있는 *db.local* 파일이 기본적인 zone 파일인데요, 한 번 살펴볼까요?

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@       IN      NS      localhost.
@       IN      A       127.0.0.1
@       IN      AAAA    ::1
```

- *TTL*: Time to live. 캐시에 zone 정보가 남아있는 데이터 유효 기간
- *@*: root 도메인
- *SOA Record*: zone에 대한 필수적인 정보를 가지고 있는 레코드
- *NS Record*: 해당 도메인의 IP 주소를 찾기 위해 가야 할 네임 서버 정보를 담고 있는 레코드
- *A Record*: 호스트 네임에 주어진 도메인에 매핑되는 IPv4 형식의 IP 주소를 저장하는 레코드
- *AAAA Record*: 호스트 네임에 주어진 도메인에 매핑되는 IPv6 형식의 IP 주소를 저장하는 레코드

우리의 목표에 맞춰서 이 파일을 기준으로 `db.bind9.zone` 파일을 한 번 만들어 봅시다.

`cp db.local db.bind9.zone` 명령을 통해 default zone 파일을 복사한 후, `vim db.bind9.zone` 을 통해 zone 파일을 편집합니다.

```
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA     ns.bind9.kr. master.yes.kr. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS      ns.bind9.kr.
@         IN      A       1.1.1.1
dns       IN      A       2.2.2.2
kt        IN      A       3.3.3.3
ns        IN      A       4.4.4.4
www       IN      CNAME   dns.bind9.kr.
```

- `@ IN SOA ns.bind9.kr. master.yes.kr.`
  - `ns.bind9.kr.` : 해당 zone의 마스터 네임 서버
  - `master.yes.kr.` : zone 관리자 연락처
    - 관리자 연락처의 경우에는 전자 메일 주소로, 첫 번째 .(온점)이 @로 대체됩니다. 예시의 경우에는 전자 메일 주소가 `master@yes.kr` 이 되는 샘플입니다.
- `CNAME Record`: 도메인 네임(alias name)을 다른 이름(canonical name)으로 매핑시키는 레코드

#### • 테스트 :

`sudo systemctl restart bind9` 을 통해 BIND9을 재시작 해주고,

`sudo systemctl status bind9`을 통해 상태를 확인하여 아래와 같이 active 상태가 되었는지 확인합니다.

`nslookup` 명령어를 통해 DNS 설정이 잘 되었는지 확인해 봅니다.

#### 우분투, DNS서버(bind9) 네임서버 설치하기

도메인이 하나 있다면, 도메인 벤더와 관계없이 직접 서브 도메인을 관리하고 싶어 질 수 있습니다. 이런 경우 우분투에서는 bind9 라는 네임서버를 설치하여 도메인을 관리할 수 있습니다. 우분투, DNS서버(bind9) 네임서버 설치하기 ◆ 도움이 되

🔗 <https://cocatv.tistory.com/131>



또 다른 방법1

<https://www.youtube.com/watch?v=AECK6Egynlk>

또 다른 방법 2



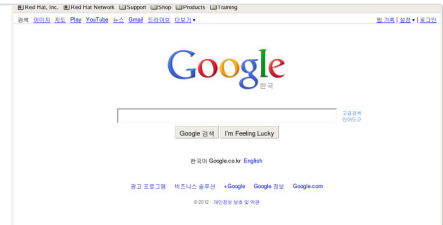
## 2 소스로 설치

### /usr/local 에 설치해야 함

#### 리눅스 네임서버 구축 (소스 컴파일 방법)

rpm과 소스compile 방법의 차이는 옵션을 설정할수 있느냐 없느냐의 차이인 것 같음. rpm의 단점 중 하나가 설정, 옵션을 내가 할 수가 없다 라고 했음 소스 컴파일 방법으로 할 경우 설치 경로등 각종 옵션들을 우리가 설정할 수 있기 때문에 이 방법

☛ <https://hayleyfish.tistory.com/21>



#### 1. BIND 데몬 설치

우선 <http://pkgs.fedoraproject.org/repo/pkgs> (리눅스에서 쓸 수 있는 각종 소스들이 많이 있음, 좋은 곳임ㅋㅋ)에 가서 네임서버의 서버프로세스인 BIND 데몬을 받자(OS가 페도라 4이기 때문에 최신버전은 쓰지 않았다. 나는 9.5.0 을 쓸거임)

우선 설치할 디렉토리를 만들고

```
[root@server103 ~]# mkdir -p /usr/local/dns
[root@server103 ~]# cd /usr/local/dns
```

필요한 bind데몬을 다운받고 압축풀기

```
[root@server103 dns]# wget http://pkgs.fedoraproject.org/repo/pkgs/bind/bind-9.5.0.tar.gz/066484717db1d1b1b4092ddcf5d0eb6e/bind-9.5.0.tar.gz
[root@server103 dns]# tar zxvf bind-9.5.0.tar.gz
[root@server103 dns]# cd bind-9.5.0
```

압축본 디렉토리에 가서 목록들을 쭉욱 보면 configure Makefile 등등이 보일 것이다.아까 설명했듯이 configure -> make -> make install 이 과정을 따라하면 됨.(압축을 풀면 README 파일이 있는데 읽어보자, 다양한 정보가 나와있음, 설치 옵션들이 라등가 등등)

```
[root@server103 bind-9.5.0]# ./configure --help이 명령어는 configure 할때 쓰는 옵션들 설명이 나온다.prefix
옵션 = 설치경로 옵션
```

```
[root@server103 bind-9.5.0]# ./configure --prefix=/usr/local/dns(끝나면 WARNING해서 openssl 뭐시깡이가
나오는데 가볍게 토스~)
```

```
[root@server103 bind-9.5.0]# make && make install
```

#### 2. named.conf 설정하기 + zonefile

네임서버 데몬(bind)이 시작될 때 제일 먼저 읽는 파일이 바로 named.conf (설정파일) 파일이다.네임 서버는 설정 파일이 한글 자라도 틀리면 데몬 자체가 가동하지 않아서 대소문자 공백 주의해서 입력할 것!!named.conf 가 어디있느냐하면 바로 우리가 설치한 곳 안에 etc 폴더에 있다 (/usr/local/dns/etc/naemd.conf)(없으면 etc 폴더 만들고 named.conf 새로 만들어야함or r/etc/ 에 가면 named.conf가 있는데 복사해오기 -->cp /etc/named.conf /usr/local/dns/etc/named.conf) 나는 그냥 복사해왔음.. 다 쓰기 귀찮아.....

named.conf 설정 형식 중 주요한 부분 설명

```
options {
    directory "/var/named";          ---> 네임서버 데이터베이스 파일(=zone파일)이 들어있는 디렉토리          dump-f
    ile "/var/named/data/cache_dump.db"; ---> 정보가 갱신될 때 저장되는 파일
    statistics-file "/var/named/data/named_stats.txt"; ---> 통계처리 용도의 파일          ~~ 생략 ~~

    // query-source address * port 53;
};
zone "도메인이름" IN {
    type master;          ---> 마스터 네임 서버는 master
    file "zone파일명";    ---> 위 options에서 directory에 생성될 "도메인이름" 의 상세설정 파일 (zonefile)

    allow-update { ip주소 또는 none; };    ---> 2차 네임 서버의 주소, 생략하면 none이 된다.
};
```

/etc/naemd..conf 복사해 왔을 때 line 22, 23, 67 (나는 이랬음) 이 행들앞마다 주석처리를 해주삼.# << 요걸 맨 앞에다가 다 붙여주기 (나중에 이거때문에 에러가 나서...)이렇게 바꿔주기  
#controls {# inet 127.0.0.1 allow { localhost; } keys { rndckey; };#};

#include "/etc/rndc.key";  
+ 맨 마지막 부분에  
zone "naver.com" IN { type master; file "naver.com.zone";};  
를 넣어준다.저장하고 닫기

이제 naver.com.zone 파일을 생성하러 갑시다.아까 options에서도 봤듯이directory "/var/named"; ---> 네임서버 데이터베이스 파일(=zone파일)이 들어있는 디렉토리zonefile들이 있어야 할 곳은 /var/named

[root@server103 etc]# cd /var/named/[root@server103 named]# vi naver.com.zone  
이걸 작성

```
$TTL      3H      SOA      @      root.      ( 2 1D 1H 1W 1H )      IN      NS      @
IN      A      (구글 ip주소)
www      IN      A      (구글 ip주소)
```

zone file 존파일의 문법 요약

\$TTL:Time To Live 약자, www.naver.com의 호스트 이름을 질의해 갔을 때, 질의해간 다른 네임 서버가 해당 ip주소를 캐시에 저장하는 기간3H는 3시간을 의미함

@:named.conf 에 정의된 naver.com를 의미함 (naver.com으로 고쳐써도 되지만 기니까 ㅇㅇ)

SOA:Start Of Authority 약자. 권한의 시작을 뜻한다.괄호안의 숫자는 시간을 의미하는데 차례대로 serial, refresh, retry, expire, minimum, hour, day, week를 의미)

IN:클래스 이름으로 internet을 의미함

NS:Name Server의 약자로 설정된 도메인의 네임 서버 역할을 하는 컴퓨터를 지정함

A:호스트 이름에 상응하는 IP주소를 지정함

참고!! IN A (구글 ip주소) --> 그냥 도메인 네임만 찾을때 가는 주소

www IN A (구글 ip주소) --> www.도메인네임 찾을때 가는 주소, 여기다가 만약 다음 ip주소를 친다면 naver.com으로 갈때는 구글로 www.naver.com으로 가면 다음으로 뜨는 것이다.

데몬을 실행시키기 전에 named.conf 파일과 zone file의 에러를 체크하기 위해서/usr/local/dns/sbin에 가봅시다.

```
[root@server103 sbin]# cd /usr/local/dns/sbin
[root@server103 sbin]# lltotal 20288-rwxr-xr-x 1 root root 1117356 Dec 2 15:07 dnsssec-keygen-rwxr-xr-x 1 root root 3041724 Dec 2 15:07 dnsssec-signzone-rwxr-xr-x 2 root root 4289987 Dec 2 15:07 lwresd-rwxr-xr-x 2 root root 4289987 Dec 2 15:07 named-rwxr-xr-x 1 root root 3178669 Dec 2 15:07 named-checkconf-rwxr-xr-x 1 root root 2990301 Dec 2 15:07 named-checkzonelrwxrwxrwx 1 root root 15 Dec 2 15:07 named-compilezone -> named-checkzone-rwxr-xr-x 1 root root 739791 Dec 2 15:07 rndc-rwxr-xr-x 1 root root 1059815 Dec 2 15:07 rndc-confgen
named-checkconf, named-checkzone 이것들이 체크해주는거임
```

```
[root@server103 sbin]# ./named-checkconf[root@server103 sbin]# ./named-checkzone naver.com /var/named/naver.com.zonezone naver.com/IN:loaded serial 20K
```

이렇게 아무에러 없이 나오면 되는거고, 에러가 뿔 나오면 그걸 고쳐주면 된다.

++현 위치가 sbin이 아닐때는 절대경로로 하면 된다.

```
[root@server103 named]# /usr/local/dns/sbin/named-checkconf
root@server103 named]# /usr/local/dns/sbin/named-checkzone naver.com /var/named/naver.com.zone
```

### 3. BIND 실행

그 다음 DNS 데몬 bind를 실행시켜보자.[root@server103 sbin]# ./named

```
[root@server103 sbin]# ps -ef | grep namedroot      15058      1  0 16:40 ?          00:00:00 ./namedr
oot      15068  3571  0 16:41 pts/2    00:00:00 grep named
```

잘 실행되고 있네요

실행이 안되었다면 문제가 있는것문제를 확인하기 위해서는/var/log/messages를 보면 되는데 여기는 리눅스 모든 에러가 기록되고 있는 곳이다.tail 10 /var/log/messages 로 10줄만 보고에러를 찾아내서 고친다.(9.3.1 버전은 named.pid오류가 나는데 아까 named.conf 에가서 options 안에 pid-file "/var/named/named/pid"; 를 입력)

이제 DNS서버를 나로 돌립시다. 항상 DNS정보를 확인하는 곳은 /etc/resolv.conf[root@server103 sbin]# vi /etc/resolv.confnameserver 에 내 아이피주소를 넣는다. 저장하고 나와서..네트워크 서비스 재시작을 꼭 합니다.  
[root@server103 sbin]# service network restart

그럼 확인해봅시다.우리는 네이버주소를쳤을때 네이버 화면이 아닌 구글 화면이나와야 함!!!!  
리눅스안의 웹 브라우저를 띄우고 naver.com 이나 www.naver.com 을 쳤을때 네이버 화면이 아닌 구글 화면 이 나오면 성공!

출처:

<https://hayleyfish.tistory.com/21>

[DB네 생선가게:티스토리]

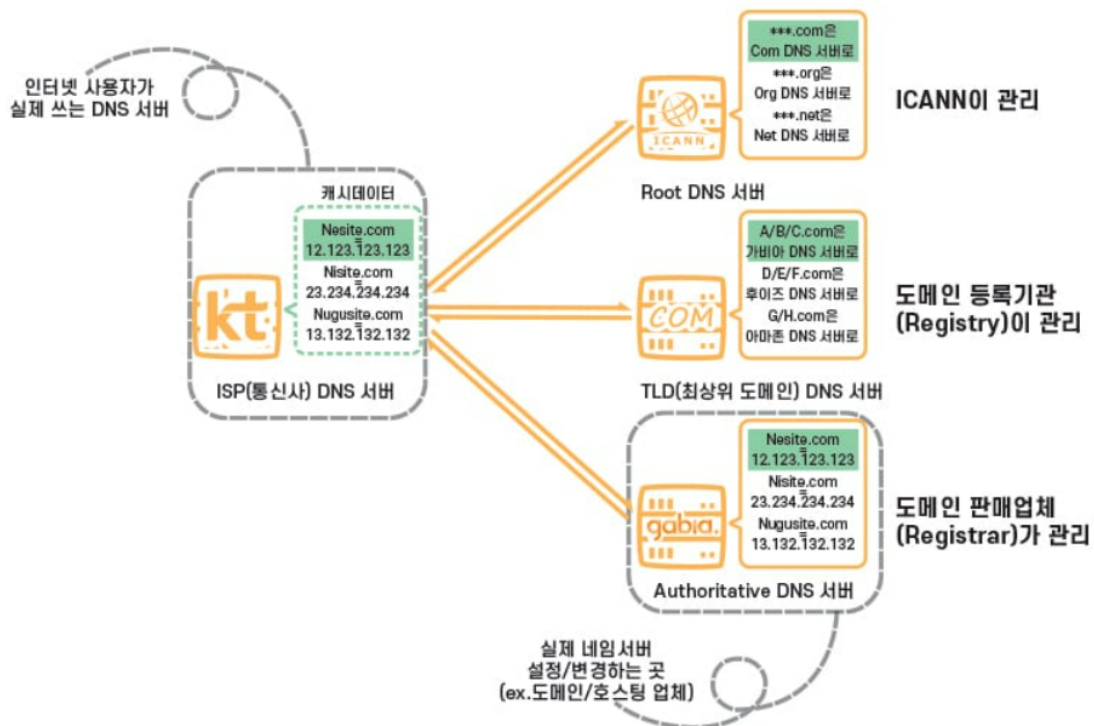
## DNS서버(네임서버) 설치

설치환경 : CentOS 6.4 minimal / bind-9.8.2 필요요소 : Bind 패키지, 도메인주소 (예: uzuro.com) 및 IP주소(예: 123.456.789.00) 설정포트 : tcp 53 / udp 53 1. 도메인 주소 DNS서버를 구축하기 위해서 도메인 주소를 확보한다. 일반적으로 도메

<https://webdir.tistory.com/163>



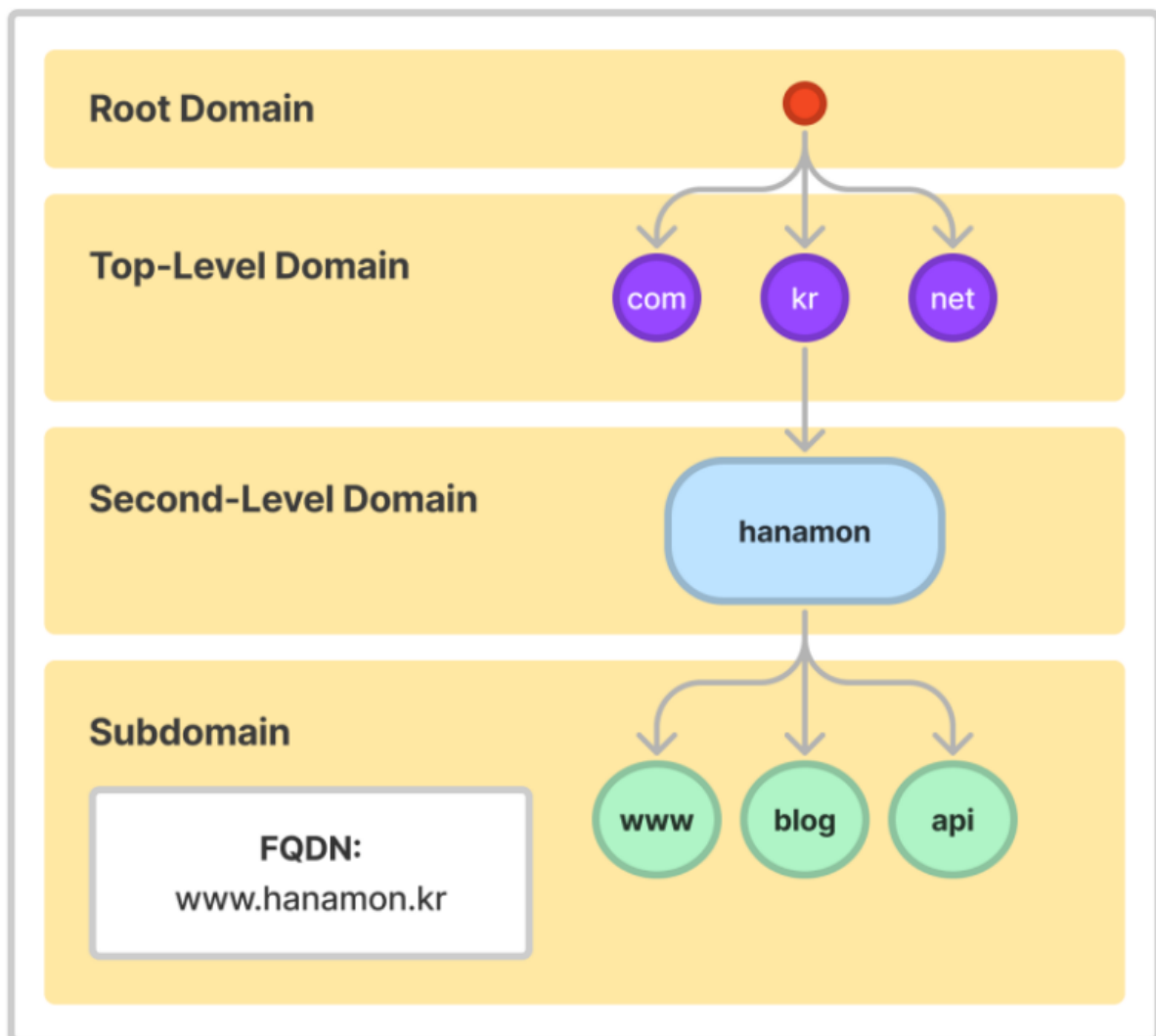
## 연동방법



<한국 DNS 서버들로 예를 들어 본 DNS 서버 구조>

- **Root DNS Server:** ICANN이 직접 관리하는 서버로, TLD DNS 서버 IP들을 저장해두고 안내하는 역할을 함.


- **TLD(최상위 도메인) DNS Server:** 도메인 등록 기관(Registry)이 관리하는 서버로, Authoritative DNS 서버 주소를 저장해두고 안내하는 역할을 함. 어떤 도메인 묶음이 어떤 Authoritative DNS Server에 속하는지 아는 이유는 도메인 판매 업체(Registrar)의 DNS 설정이 변경되면 도메인 등록 기관(Registry)으로 전달이 되기 때문임.
- **Authoritative DNS Server:** 실제 개인 도메인과 IP 주소의 관계가 기록/저장/변경되는 서버. 그래서 권한의 의미인 Authoritative가 붙음. 일반적으로 **도메인/호스팅 업체의 '네임서버'**를 말하지만, 개인 DNS 서버 구축을 한 경우에도 여기에 해당함.
- **Recursive DNS Server:** 인터넷 사용자가 가장 먼저 접근하는 DNS 서버임. 위 3개의 DNS 서버를 매번 거친다면 효율이 구데기일 수밖에 없으니, 한 번 거친 후 얻은 데이터를 일정 기간 (TTL/Time to Live) 동안 **캐시라는 형태로 저장해 두는 서버**임. 직접 도메인과 IP 주소의 관계를 기록/저장/변경하지는 않고 캐시만을 보관하기 때문에, Authoritative와 비교되는 의미로 반복의 Recursive가 붙음. 대표적인게 KT/LG/SK와 같은 **ISP(통신사) DNS 서버**가 있고, 브라우저 우회 용도로 많이 쓰는 구글 DNS, 클라우드플레어와 같은 Public DNS 서버가 있음.



## 장단점

## 7 최고의 DNS 서버 (무료 및 공용)

인터넷의 작동에 대해 아는 경우 도메인 이름 서버 또는 DNS에 대해 들어 보셨습니까? DNS는 오늘날 우리가 사용하고 알고있는 인터넷이 존재하지 않는 인터넷 빌딩 블록 중 하나입니다. 인터넷 서비스 제공 업체 또는 ISP는 기본 DNS 서버를 제공

 <https://ko.gadget-info.com/16658-7-best-dns-servers-free-and-public>

