

DNS

DNS

❖ 개요

- ✓ DNS(Domain Name System)는 우리가 흔히 사용하는 웹사이트 주소(예: google.com)를 컴퓨터가 이해할 수 있는 IP 주소(예: 142.250.190.46)로 변환해 주는 시스템
- ✓ 인터넷상의 모든 기기는 숫자로 된 IP 주소를 통해 서로를 식별하지만 사람이 이를 모두 외우기 어렵기 때문에 DNS를 통해 문자로 된 이름을 사용
- ✓ 웹 브라우저에 google.com 등과 같은 URL을 사용하는데 실제 원하는 서버에 접근하려면 이 URL을 해당 컴퓨터의 IP 주소로 변환시켜야 하는데 이 일을 네임 서버 또는 DNS 서버라고 하는 컴퓨터가 담당
- ✓ www.google.com을 IP 주소로 변환하는 과정을 이름 해석(resolution)이라고 함
- ✓ 네트워크에서 컴퓨터를 구분하는 유일한 방법은 IP 주소인데 인터넷에 연결된 모든 컴퓨터에는 중복되지 않는 IP 주소가 있음
- ✓ DNS Server를 이용하지 않는다면 IP 주소를 알아내는 과정이 생략되므로 인터넷 속도가 더 빨라짐

DNS

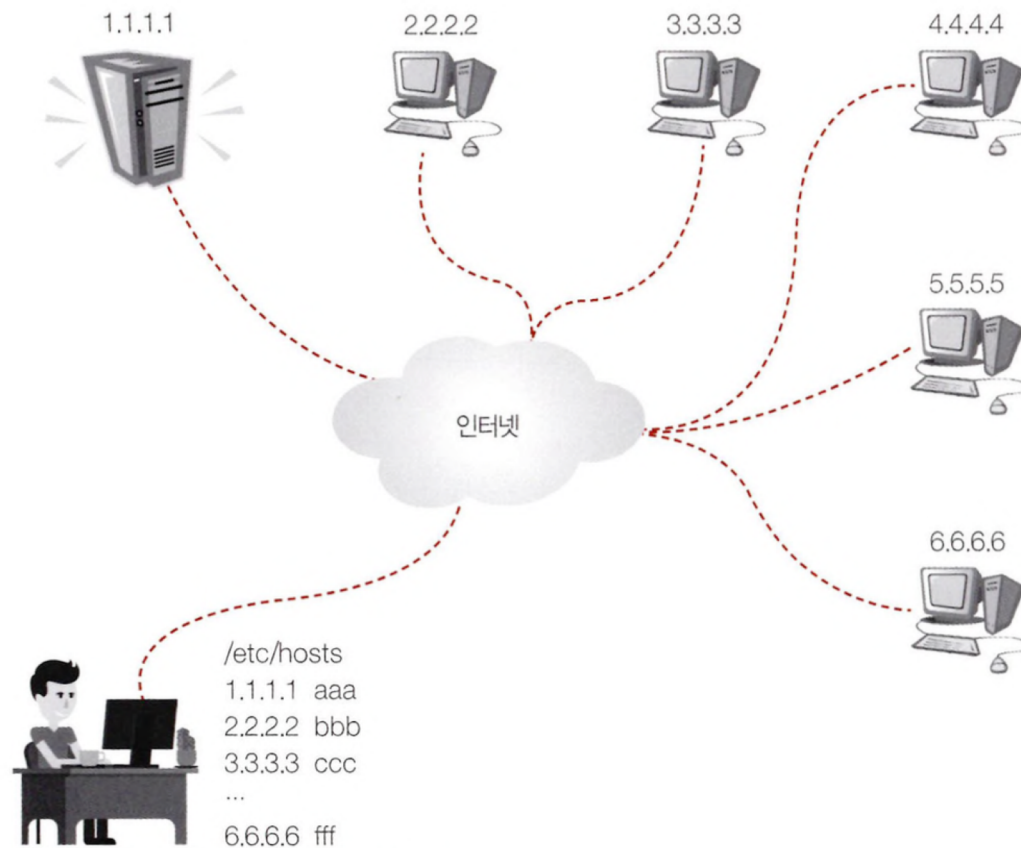
❖ hosts 파일

- ✓ hosts 파일은 도메인 이름을 IP 주소로 직접 매핑하는 로컬 DNS 파일
- ✓ 역할
 - 도메인 → IP 변환을 로컬에서 강제로 지정
 - 127.0.0.1 localhost 로 작성하면 localhost → 127.0.0.1 로 해석됨
 - DNS 조회를 아예 하지 않음
- ✓ 파일 위치
 - Linux / macOS: /etc/hosts
 - Windows: C:\Windows\System32\drivers\etc\hosts
 - 관리자(root) 권한 필요
- ✓ 예시

```
127.0.0.1 localhost
::1      localhost
```

DNS

❖ hosts 파일



DNS

❖ hosts 파일

✓ 주의사항

- 0.0.0.0은 접속 대상 IP가 아니었기 때문에 연결 즉시 끊김 or 실패
- 특정 사이트 접속 차단
0.0.0.0 ads.example.com

✓ hosts 파일 수정 후 체크 사항

- DNS 캐시 삭제 (권장)

◆ macOS

- `sudo dscacheutil -flushcache`
- `sudo killall -HUP mDNSResponder`

◆ Linux (systemd)

- `sudo systemctl restart systemd-resolved`

◆ Windows

- `ipconfig /flushdns`

DNS

❖ hosts 파일

✓ 실제 해석 확인

- ping 도메인명
- getent hosts 도메인명

✓ hosts vs DNS 우선순위

1. hosts 파일
2. DNS 캐시
3. DNS 서버



DNS

❖ hosts 파일

✓ 운영 서버에서 hosts 사용을 피해야 하는 이유

- DNS 변경이 반영되지 않을 수 있음(가장 치명적)

◆ 상황

- DNS에서 서버 IP 변경
- 일부 서버만 계속 예전 IP로 접속

◆ 원인

/etc/hosts

10.0.0.5 api.company.com

- DNS는 바뀌었지만 hosts가 우선
- 서버마다 서로 다른 목적지로 접속

◆ 결과

- 서버 간 통신 장애
- 특정 서버에서만 장애 발생
- 재현 불가 장애

DNS

❖ hosts 파일

✓ 운영 서버에서 hosts 사용을 피해야 하는 이유

- 서버마다 설정이 달라짐(일관성 붕괴)

- ◆ 서버 A: hosts 있음

- ◆ 서버 B: hosts 없음

- 장애 발생 시 원인 추적이 극도로 어려움

- ◆ 현상

- SSH / API / DB 간헐적 실패

- 로그상 네트워크 문제처럼 보임

- ◆ 실제 원인

- 127.0.0.1 db.company.com

- ◆ 네트워크 / 방화벽 / 서버 문제로 오인

DNS

❖ hosts 파일

✓ 운영 서버에서 hosts 사용을 피해야 하는 이유

- 서버 교체 또는 오토스케일링과 충돌

◆ 클라우드 환경

- Auto Scaling
- Immutable Infrastructure
- Blue/Green 배포

◆ 문제

- 새 서버에는 hosts 없음
- 기존 서버만 정상 동작
- 배포 직후 일부 서버만 장애

DNS

❖ hosts 파일

✓ 운영 서버에서 hosts 사용을 피해야 하는 이유

- 보안 사고 위험

- ◆ 공격 시나리오: 10.0.0.99 auth.company.com

- 인증 서버를 가짜 서버로 유도 가능
 - 내부자 공격 / 침해 시 피해 증폭

- 운영 자동화 도구와 충돌

- ◆ 충돌 대상

- Ansible
 - Terraform
 - Kubernetes
 - CI/CD 파이프라인

- ◆ 문제

- 인프라는 DNS 기준
 - hosts는 로컬 수동 설정

DNS

❖ hosts 파일

✓ 운영 서버에서 hosts 사용을 피해야 하는 이유

- IPv4 / IPv6 우선순위 문제

::1 api.company.com

- ◆ SSH가 IPv6 먼저 시도
- ◆ 서버는 IPv4만 리스
- ◆ timeout / hang 발생

DNS

❖ hosts 파일

✓ 허용되는 경우

- 로컬 개발 PC
- 개인 테스트 서버
- 단기 장애 우회 (→ 즉시 원복)



DNS

❖ DNS의 주요 구성 요소

- ✓ DNS Recursive Resolver: 사용자로부터 쿼리를 받는 첫 번째 지점으로 도서관의 사서처럼 요청을 전달받아 실제 주소를 찾아 나섬
- ✓ Root Nameserver: 호스트 이름을 IP 주소로 변환하는 첫 번째 단계로 .com, .net과 같은 TLD 서버의 위치를 알려줌
- ✓ TLD Nameserver(최상위 도메인 네임서버): .com, .org, .kr 등 도메인의 확장자를 관리하며 구체적인 도메인 정보를 가진 네임서버로 안내
- ✓ Authoritative Nameserver(권한 있는 네임서버): 실제 IP 주소가 저장되어 있는 최종 목적지로 여기서 정확한 IP 주소를 회신

DNS

❖ DNS 레코드의 종류

- ✓ DNS 서버에는 다양한 형태의 정보가 저장되는데 이를 레코드(Record)라고 부름
- ✓ 종류
 - A Record: 도메인을 IPv4 주소와 연결 (가장 기본)
 - AAAA Record: 도메인을 IPv6 주소와 연결
 - CNAME: 한 도메인을 다른 도메인으로 연결
 - MX Record: 해당 도메인의 이메일을 수신할 메일 서버를 지정
 - TXT Record: 도메인에 대한 텍스트 정보를 담으며 주로 보안 인증에 사용

DNS

❖ DNS 작동 원리(조회 과정)

- ✓ 사용자 요청: 사용자가 example.com을 입력
- ✓ 리커서 조회: 브라우저가 DNS 리커서에게 IP 주소를 물음
- ✓ 루트 서버 조회: 리커서가 루트 서버에 묻고, 루트 서버는 ".com" TLD 서버로 가라고 답변
- ✓ TLD 서버 조회: 리커서가 .com TLD 서버에 묻고 해당 도메인의 권한 있는 네임서버 주소를 받음
- ✓ 최종 확인: 권한 있는 네임서버가 example.com의 실제 IP 주소를 리커서에게 알려줌
- ✓ 접속 완료: 리커서가 브라우저에 IP를 전달하고 브라우저는 해당 IP로 웹사이트 데이터를 요청

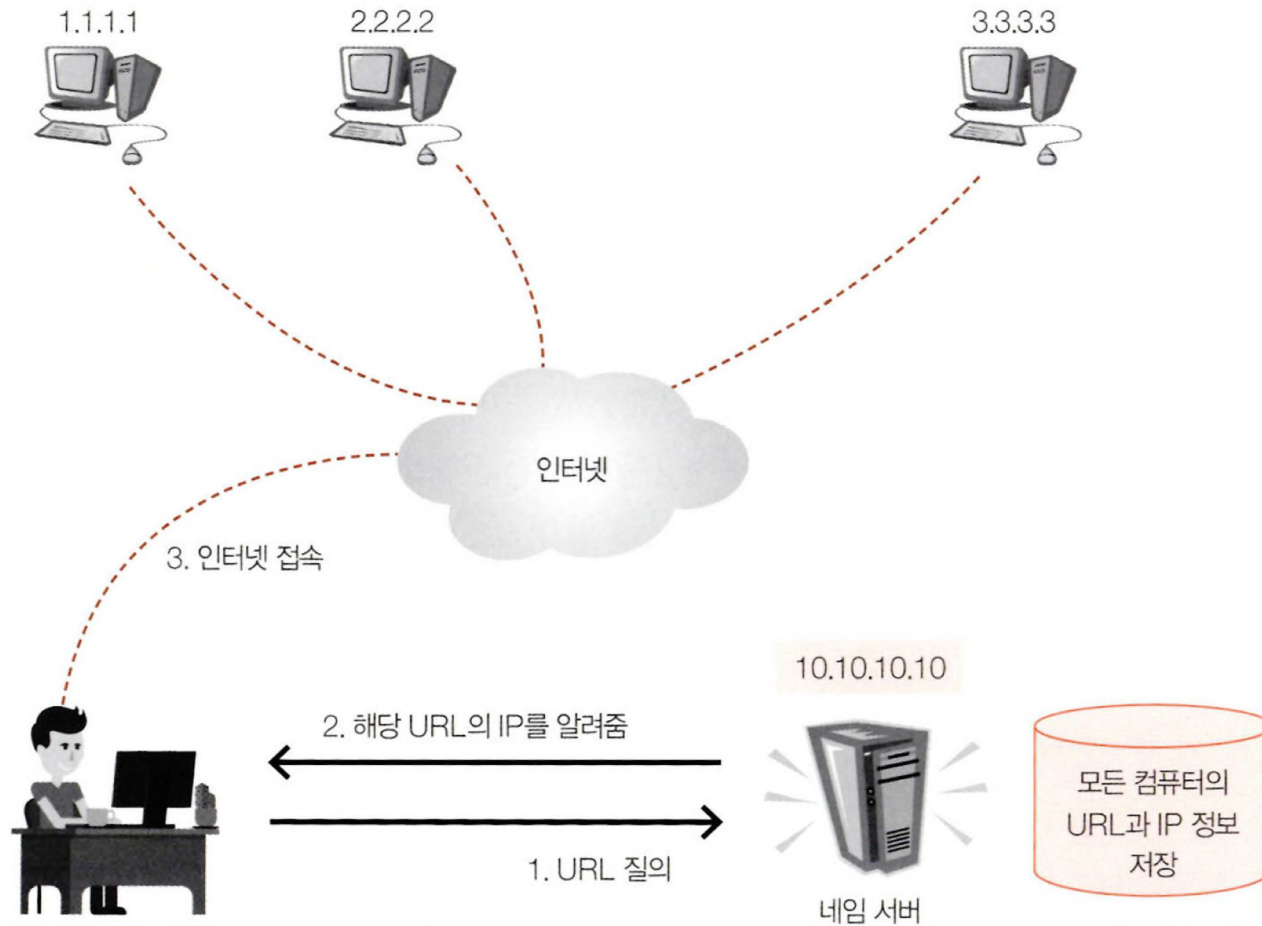
DNS

❖ DNS 캐싱(Caching)

- ✓ 매번 동일한 조회 과정을 거치면 속도가 느려지기 때문에 한 번 찾은 주소는 일정 기간 저장
- ✓ 브라우저/OS 캐싱: 컴퓨터 내부에 먼저 저장된 기록이 있는지 확인
- ✓ TTL(Time To Live): DNS 레코드가 캐시에 저장되는 유효 기간으로 TTL이 지나면 다시 서버에 정보를 요청

DNS

❖ DNS Server



DNS

❖ DNS Server 설정

✓ /etc/resolv.conf 파일 확인

```
nameserver 127.0.0.53  
options edns0 trust-ad  
search .
```

✓ nameserver 설정이 없는 경우

```
#nameserver 127.0.0.53  
options edns0 trust-ad  
search .
```

● ping www.naver.com

ping: www.naver.com: Temporary failure in name resolution

● ping 23.60.184.249

PING 23.60.184.249 (23.60.184.249) 56(84) bytes of data.

64 bytes from 23.60.184.249: icmp_seq=1 ttl=48 time=38.4 ms

64 bytes from 23.60.184.249: icmp_seq=2 ttl=48 time=5.08 ms

64 bytes from 23.60.184.249: icmp_seq=3 ttl=48 time=9.75 ms

64 bytes from 23.60.184.249: icmp_seq=4 ttl=48 time=8.32 ms

DNS

❖ DNS Server 설정

✓ /etc/resolv.conf 파일 수정

- 파일 경로 확인: `ls -l /etc/resolv.conf`
`lrwxrwxrwx 1 root root 39 Aug 6 2025 /etc/resolv.conf -`
`> ../run/systemd/resolve/stub-resolv.conf`
- /etc/resolv.conf 파일 수정: `sudo nano /etc/systemd/resolved.conf`
127.0.0.1 localhost
127.0.1.1 original
The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
23.60.184.249 www.naver.com

DNS

❖ DNS Server 설정

✓ /etc/resolv.conf 파일 수정

- 서비스 재시작: `sudo systemctl restart systemd-resolved`
- `ping www.naver.com`

`ping www.naver.com`

PING www.naver.com (23.60.184.249) 56(84) bytes of data.

64 bytes from www.naver.com (23.60.184.249): icmp_seq=1 ttl=48 time=8.76 ms

64 bytes from www.naver.com (23.60.184.249): icmp_seq=2 ttl=48 time=6.44 ms

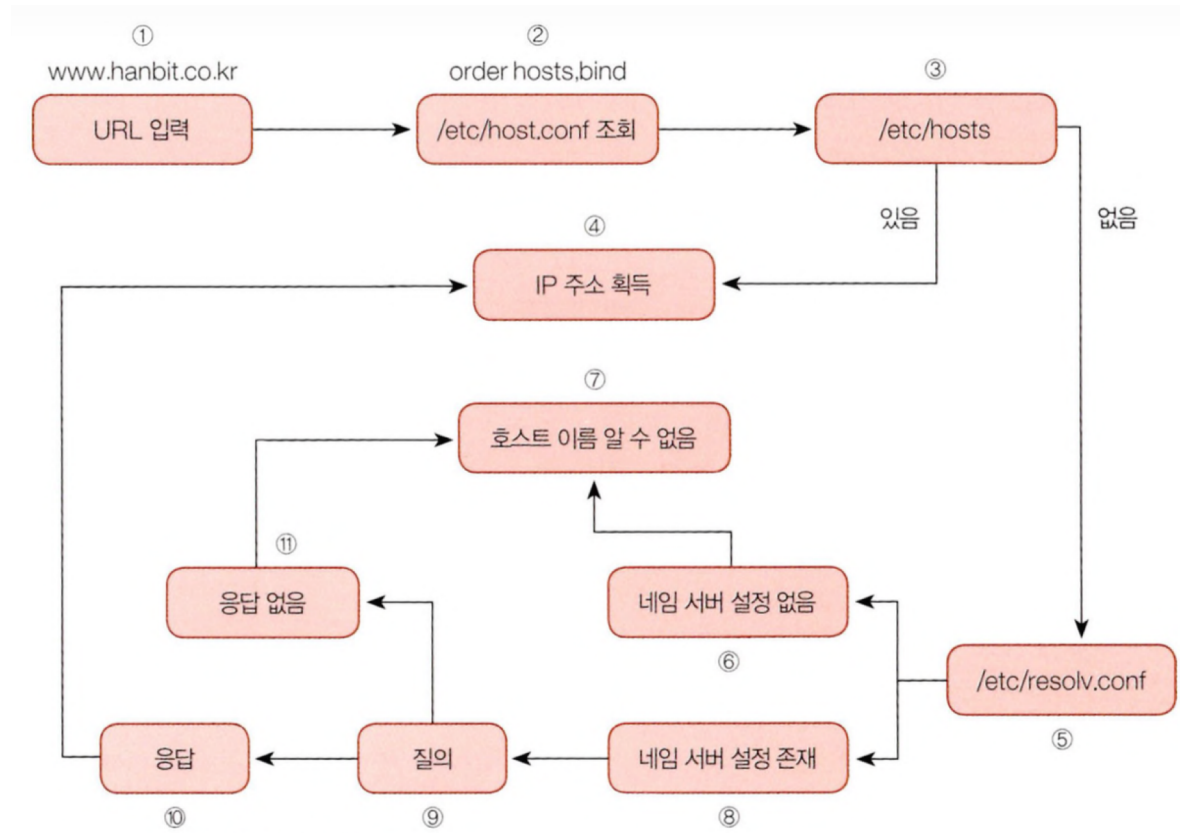
64 bytes from www.naver.com (23.60.184.249): icmp_seq=3 ttl=48 time=38.5 ms

64 bytes from www.naver.com (23.60.184.249): icmp_seq=4 ttl=48 time=9.59 ms

DNS

❖ DNS Server 설정

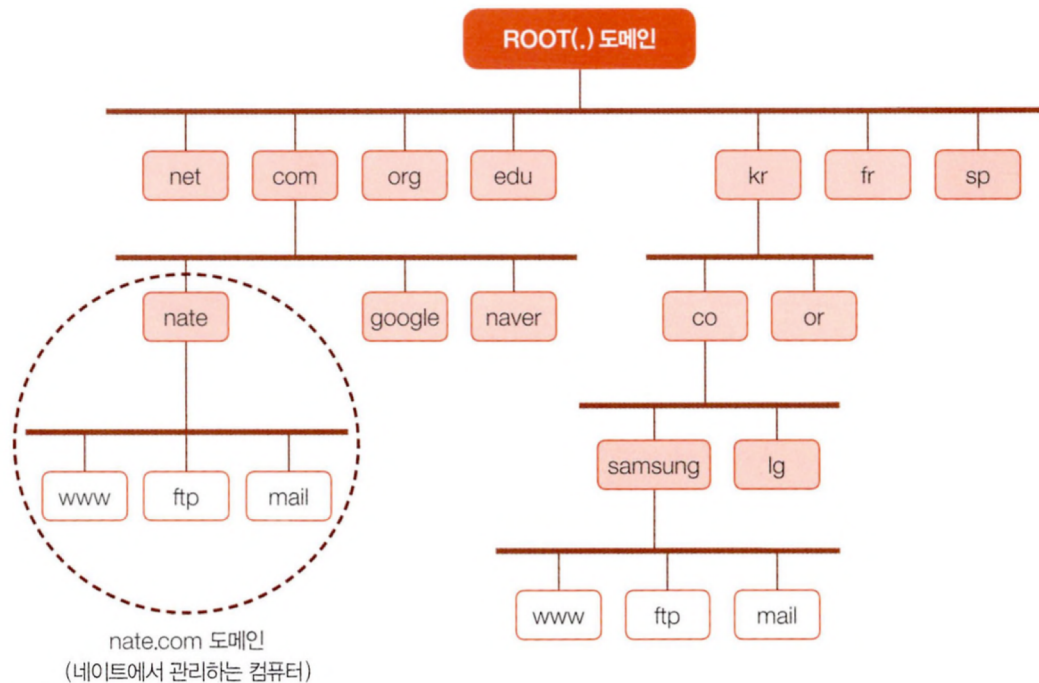
- ✓ IP 주소를 찾아가는 흐름



Name Server

❖ 도메인 이름 체계

- ✓ 트리 구조 형태의 도메인 이름 체계를 가짐
- ✓ ROOT(.) 네임 서버는 1단계 네임 서버인 com 네임 서버, net 네임 서버, org 네임 서버, edu 네임 서버 등과 국가 도메인인 kr, fr, sp 등을 관리하고 1단계 네임 서버는 자신의 하위에 있는 2단계 네임 서버를 관리
- ✓ com 네임 서버는 nate, google, naver 등 2단계 도메인을 관리하는 네임 서버들만 관리



Name Server

❖ 로컬 네임 서버 동작

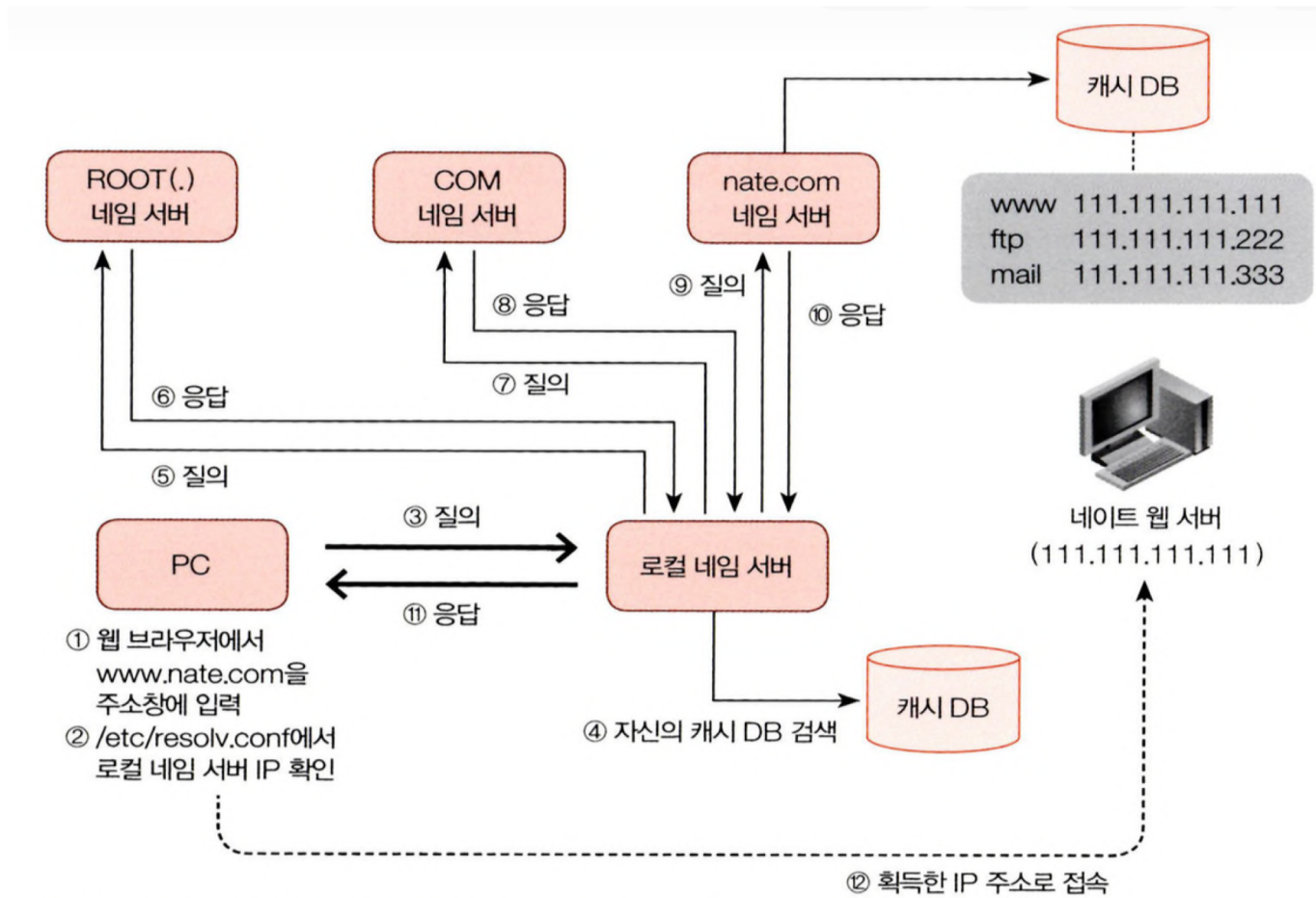
✓ 개요

- 리눅스에는 각자 사용하는 네임 서버가 /etc/resolv.conf 파일에 nameserver IP주소 형식으로 설정되어 있는데 이 네임 서버를 로컬 네임 서버라고 부름
- www.nate.com의 IP 주소를 요구하면 이 로컬 네임 서버에 질문
- 로컬 네임 서버는 의외로 아는 것이 별로 없는데 로컬 네임 서버가 혼자서 전 세계 모든 컴퓨터의 도메인 이름을 관리할 수는 없기 때문
- 로컬 네임 서버는 자신이 아는 도메인 이름이라면 바로 알려주지만 자신이 모를 경우에는 그림과 같은 형태로 작업을 수행

Name Server

❖ 로컬 네임 서버 동작

- ✓ 컴퓨터가 네임 서버를 통해 IP 주소를 획득하는 흐름도

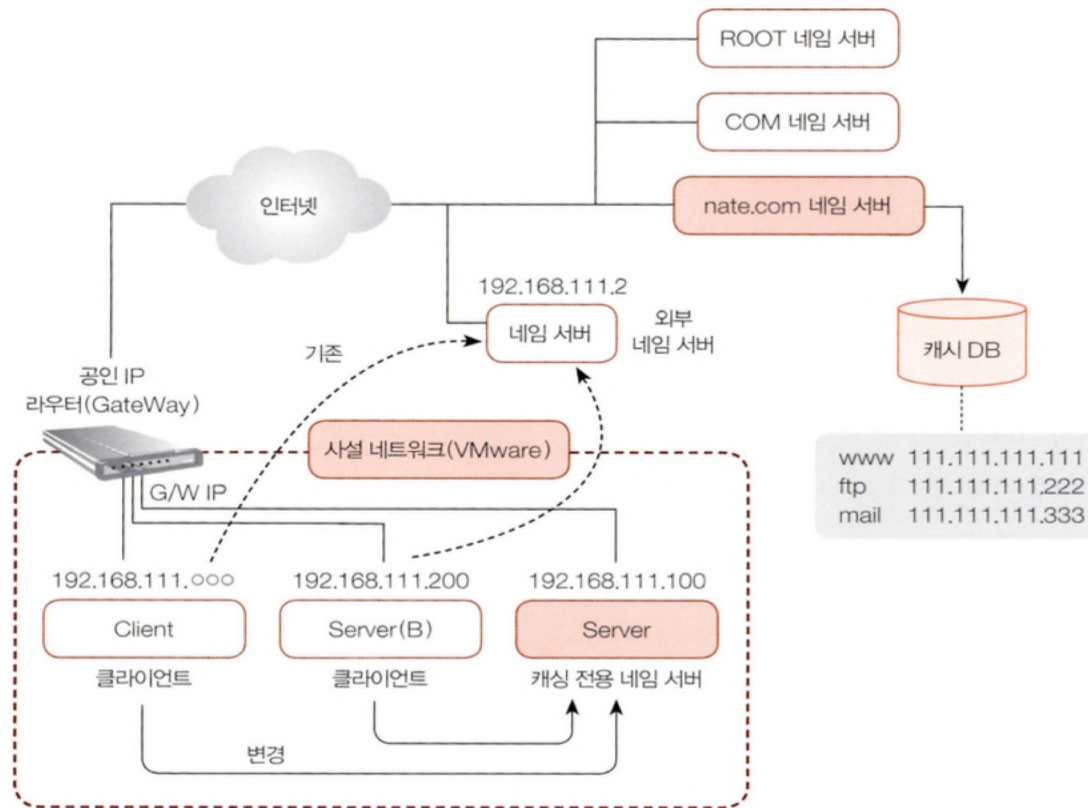


Name Server

❖ Caching-only Nameserver

✓ 개요

- Caching-only Nameserver는 컴퓨터에서 URL로 IP 주소를 얻고자 할 때 해당하는 URL의 IP 주소를 알려주는 네임 서버를 의미



Name Server

❖ Name Server 설치 및 설정

✓ 설치: BIND9 이용

- `sudo apt update`
- `sudo apt install bind9 bind9-utils bind9-doc -y`
- `systemctl status bind9`

✓ 기본 설정 파일: `/etc/bind/`

- `named.conf`: 전체 설정 진입점
- `named.conf.options`: DNS 서버 옵션
- `named.conf.local`: 도메인(zone) 설정
- `db.*`: 실제 DNS 레코드 파일

Name Server

❖ Name Server 설치 및 설정

✓ 설정

- 설정 파일 수정: `sudo vi /etc/bind/named.conf.options`

```
options {  
    directory "/var/cache/bind";  
    dnssec-validation no;  
    recursion yes;  
    allow-query { any; };  
  
    listen-on-v6 { any; };  
};
```

- 서비스 재시작: `systemctl restart named`
- 부팅 시 항상 시작: `systemctl enable named`
- 서비스 확인: `systemctl status named`
- 방화벽 해제: `sudo ufw allow 53`

Name Server

❖ Name Server 설치 및 설정

✓ 동작 확인

- dig @네임서버IP 조회할URL

adam@server:~\$ dig @192.168.0.100 www.google.com

;; QUESTION SECTION:

;www.google.com. IN A

;; ANSWER SECTION:

www.google.com. 300 IN A
142.251.156.119

;; Query time: 269 msec

;; SERVER: 192.168.0.100#53(192.168.0.100) (UDP)

;; WHEN: Fri Apr 17 00:21:17 UTC 2026

;; MSG SIZE rcvd: 199

Name Server

❖ Name Server 설치 및 설정

✓ 동작 확인

● nslookup

```
adam@server:~$ nslookup
```

```
> www.naver.com
```

```
Server:                127.0.0.53
```

```
Address:               127.0.0.53#53
```

```
Non-authoritative answer:
```

```
www.naver.com canonical name = www.naver.com.nheos.com.
```

```
www.naver.com.nheos.com canonical name = www.naver.com.edgekey.net.
```

```
www.naver.com.edgekey.net canonical name = e6030.a.akamaiedge.net.
```

```
Name:                  e6030.a.akamaiedge.net
```

```
Address: 23.207.176.194
```

Name Server

❖ Name Server 설치 및 설정

✓ 클라이언트 설정 및 확인

- `sudo nano /etc/resolv.conf`

```
nameserver 192.168.0.100
options edns0 trust-ad
search .
```

- 확인

```
adam@server:~$ nslookup
```

```
> www.naver.com
```

```
Server:                127.0.0.53
```

```
Address:               127.0.0.53#53
```

```
Non-authoritative answer:
```

```
www.naver.com canonical name = www.naver.com.nheos.com.
```

```
www.naver.com.nheos.com canonical name = www.naver.com.edgekey.net.
```

```
www.naver.com.edgekey.net canonical name = e6030.a.akamaiedge.net.
```

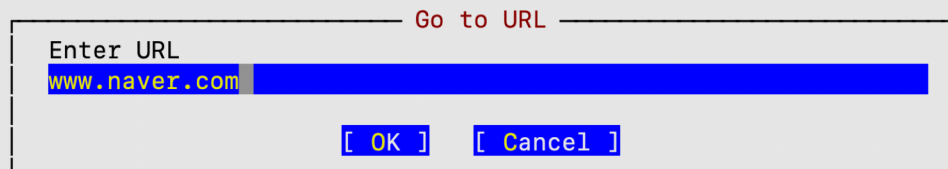
```
Name:                 e6030.a.akamaiedge.net
```

```
Address: 23.207.176.194
```

```
>
```

Name Server

- ❖ Name Server 설치 및 설정
 - ✓ 클라이언트 설정 및 확인
 - Text 기반의 웹 브라우저 이용
 - ◆ 브라우저 설치: `sudo apt -y install elinks`
 - ◆ 브라우저 실행: `elinks`



Name Server

- ❖ Name Server 설치 및 설정
 - ✓ 클라이언트 설정 및 확인
 - Text 기반의 웹 브라우저 이용
 - ◆ 브라우저 실행: elinks



```
adam - NAVER - ELinks - ssh adam@192.168.201.107 -p 20001 - 80x24
NAVER
Link: canonical
Link: alternate (only screen and (max-width: 767px))
상단영역 바로가기 서비스 메뉴 바로가기 새 소식 블록 바로가기 쇼핑 블록 바로가기
관심사 블록 바로가기 MY 영역 바로가기 위젯 보드 바로가기 보기 설정 바로가기
검색
-----
검색
입력 도구
자동완성 /최근 검색어펼치기
https://www.naver.com/
```