# 네트워크 기술

# NAT&PAT

❖ 개요

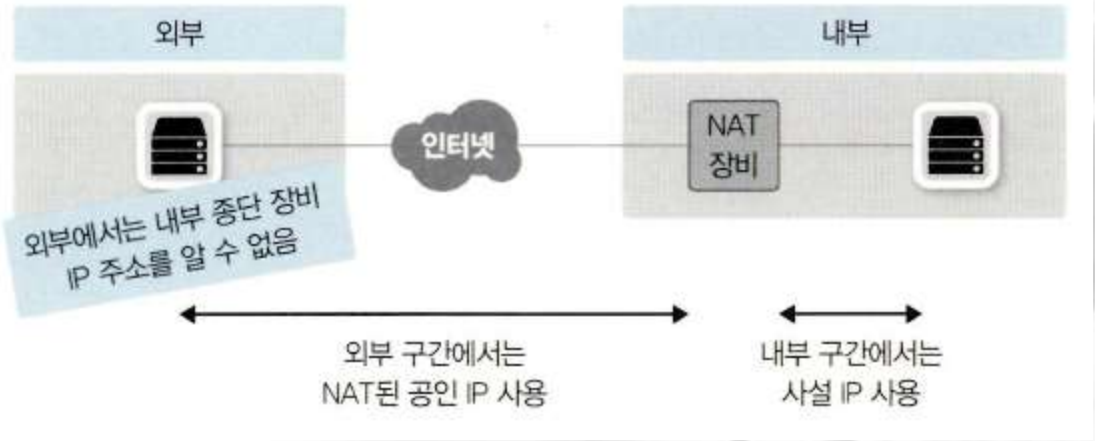
- ✓ NAT(Network Address Translation – 네트워크 주소 변환)는 기본적으로 하나의 네트워크 주소를 다른 하나의 네트워크 주소로 변환하는 기술
- ✓ 1:1 변환이 기본이지만 IP 주소가 고갈되는 문제를 해결하기 위해 1:1 변환이 아닌 여러 개의 IP를 하나의 IP로 변환하기도 하는데 여러 개의 IP를 하나의 IP로 변환하는 기술도 NAT 기술 중 하나이고 NAT로 통칭하여 불리기도 하지만 공식 용어는 NAPT(Network Address Port Translation)
- ✓ 실무에서는 NAPT의 경우 PAT(Port Address Translation)라는 용어로 더 많이 사용
- ✓ NAT는 IP 주소를 다른 IP 주소로 변환해 라우팅을 원활히 해주는 기술이라고 인터넷 표준 문서에서 정의
- ✓ Private IP에서 Public IP로 전환하는 것 뿐 아니라 Public IP에서 Private IP로 주소를 전환할 수 있고 Private IP에서 또 다른 Private IP 나 Public IP에서 또 다른 공인 IP로의 전환도 NAT로 정의될 수 있음
- ✓ IPv4 주소를 IPv6 주소로 변환하거나 그 반대로 IP 주소를 변환하는 기술인 AFT(Address Family Translation)도 NAT 기술의 일종
- ✓ 다양한 NAT 기술과 방법이 존재하지만 NAT가 가장 많이 사용되는 경우는 Private IP 주소에서 Public IP 주소로 전환하는 경우

# NAT&PAT

- 용도 와 필요성
  - ✓ IPv4 주소 고갈 문제의 솔루션으로 NAT가 사용
    - 인터넷 대중화로 갑자기 폭증한 IP 주소 요구를 극복하기 위해 단기, 중기, 장기의 3단계 IP 주소 보존과 전환 전략을 수립
    - 이 전략이 매우 잘 만들어졌고 실제 환경에서도 유용하게 쓰이면서 IPv4 주소 부족 문제가 많이 해소됨
    - 현재는 IPv4 주소 할당이 끝나면서 신규로 할당 가능한 IPv4 주소가 없는 상태이고 IPv6 주소 체계 전환을 일부 분야에서는 매우 많이 진행하고 있음
    - IPv4 주소 보존 전략 중 단기 전략은 Subnetting, 중기 전략은 NAT와 Private IP 체계, 장기 전략은 IPv6 전환
    - NAT를 이용한 중기 전략이 IPv4 주소 보존에 큰 기여를 했는데 외부에 공개해야 하는 서비스에 대해서는 Public IP를 사용하고 외부에 공개할 필요가 없는 일반 사용자의 PC나 기타 종단 장비에 대해서는 Private IP를 사용해 꼭 필요한 곳에만 효율적으로 IP를 사용할 수 있게 함

# NAT&PAT

- ❖ 용도 와 필요성
  - ✓ 보안을 강화하는 데 NAT 기술을 사용
    - IP 주소는 네트워크에서 중복되지 않아야 하고 이 정보를 이용해서 외부에서 식별해서 외부와 통신하게 해줌
    - 외부와 통신할 때 내부 IP를 다른 IP로 변환해 통신하면 외부에 내부 IP 주소 체계를 숨길 수 있음
    - NAT는 주소 변환 후 역변환이 정상적으로 다시 수행되어야만 통신이 가능한데 이 성질을 이용해 복잡한 룰 설정 없이 방향성을 통제할 수 있음
    - 내부 네트워크에서 외부 네트워크로 나가는 방향 통신은 허용하지만 외부에서 시작해 내부로 들어오는 통신은 방어할 수 있음
    - NAT&PAT의 이런 성질을 이용해 보안을 쉽게 강화할 수 있음



# NAT&PAT

- 용도 와 필요성
  - ✓ IP 주소 체계가 같은 두 개의 네트워크 간 통신을 가능하게 해줌
    - IP 네트워크에서 서로 통신하려면 식별 가능한 유일한 IP 주소가 있어야 하는데 Public IP는 인터넷에서 유일한 주소로 IP 주소가 중복되면 안되지만 Private IP는 외부와 통신할 때 Public IP로 변환되어 통신하므로 서로 다른 회사에서 중복해 사용할 수 있음
    - 회사 내부에서 Private IP를 독립적으로 사용한다면 상관없지만 Private IP를 이용해 다른 회사와 직접 연결해야 하거나 회사 간 합병으로 서로 통신해야 한다면 Private IP 주소가 충돌할 수 있음
    - 흔히 대외계라고 부르는 회사 간 통신에서 이런 상황이 많이 발생
    - IP 대역이 같은 네트워크와 통신할 가능성이 높은 대외계 네트워크를 연결하기 위해 출발지와 도착지를 한꺼번에 변환하는 Double NAT 기술을 사용

# NAT&PAT

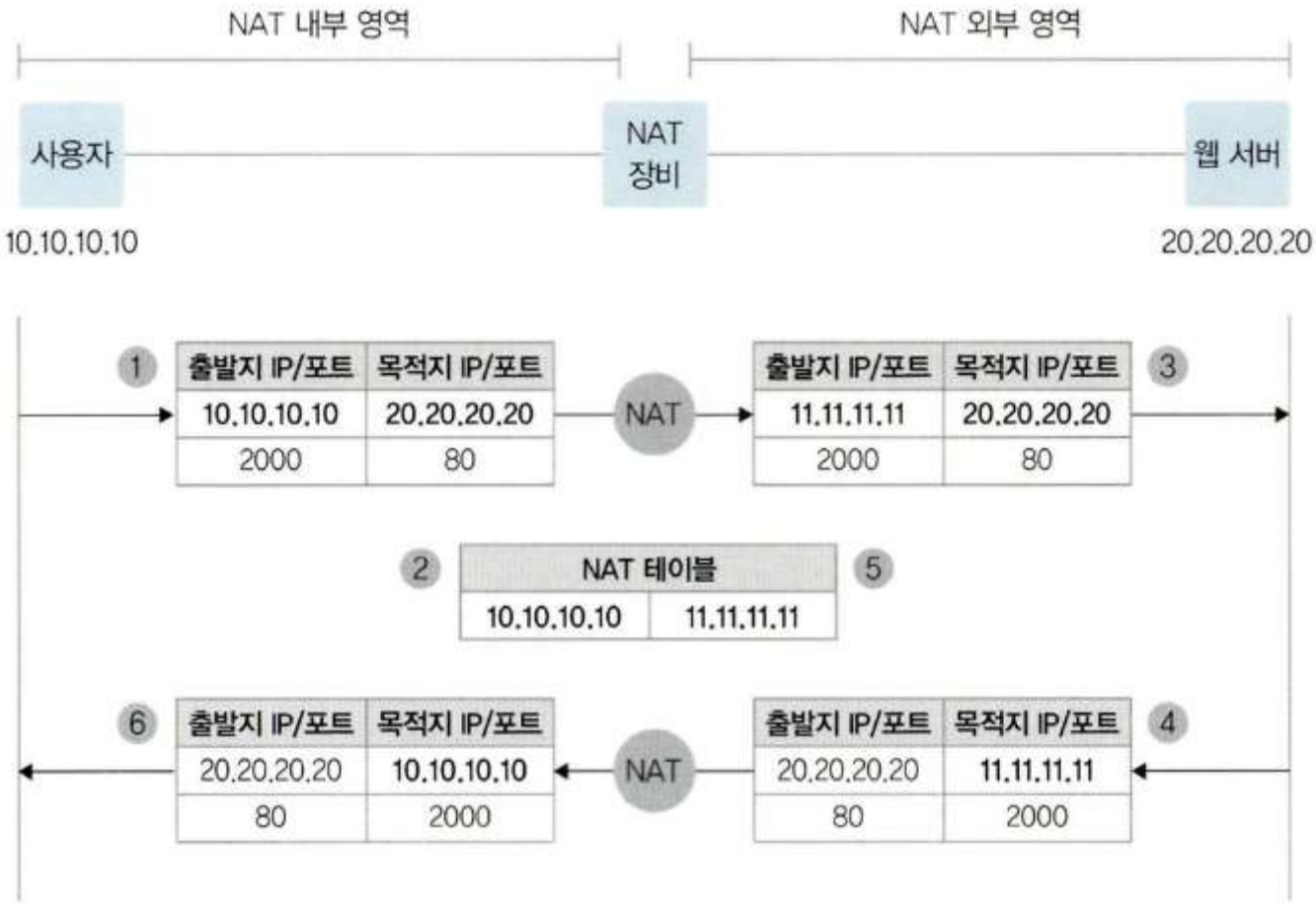
- ❖ 용도 와 필요성
  - ✓ 불필요한 설정 변경을 줄일 수 있음
    - KISA를 통해 인터넷 독립기관으로 직접 등록하고 소유한 IP 주소를 직접 운영하는 경우가 아니라면 통신사업자나 IDC 쪽에서 IP를 할당받아 사용하는데 이 IP들은 자신이 소유한 것이 아니라 임시로 빌려 사용하는 것이므로 회선 사업자를 바꾸거나 IDC를 이전하면 그동안 빌려 써왔던 공인 IP를 더 이상 사용할 수 없고 신규 사업자가 빌려주는 IP로 변경해야 함
    - 사용자가 NAT&PAT를 이용해 내부 네트워크를 구성하고 있었다면 서버와 PC의 IP 주소 변경 없이 회선과 IDC 사업자 이전이 가능
    - 외부에 서비스하던 공인 IP 주소가 변경되므로 DNS 서비스나 NAT를 수행하는 네트워크 장비 설정은 변경해야 하지만 내부 서버나 PC 설정 변경을 최소화할 수 있어 NAT&PAT 기술을 적용하면 복잡한 작업을 많이 줄일 수 있는데 이런 설계는 특정 사업자에 종속되지 않는 유연한 Infrastructure의 기본 요소로 비즈니스 유연성을 높이는데 매우 중요한 기술로 활용

# NAT&PAT

- 단점
  - ✓ 네트워크 운영자 입장에서는 IP가 변환되면 장애가 발생했을 때 문제 해결이 힘듬
  - ✓ 애플리케이션 개발자는 NAT 환경이 대중화되면서 애플리케이션을 개발할 때 더 많은 고려사항이 생김(Port Forwarding)
  - ✓ IPv6 전환은 IPv4 주소 부족 해결이라는 중요한 목표가 있지만 이런 어려움을 주는 NAT를 인터넷에서 없애는 목표도 있음
  - ✓ NAT로 인해 주소가 변환되면서 단말 간 직접적인 연결성이 무너졌고 이로 인해 개발자들이 애플리케이션을 제작할 때 NAT 환경을 항상 고려해야 하는 상황이 되었음
  - ✓ NAT의 이런 한계를 극복하기 위해 NAT 밑에 있는 단말도 직접 연결하게 도와주는 Hole Punching 기술이 나오고 이 기술을 이용하기 위해 애플리케이션이 더 복잡해지는 악순환이 계속됨
  - ✓ 오랜 시간 동안 NAT는 일상생활 속에 많이 적용되었고 그 과정에서 NAT로 인한 오버헤드나 다양한 문제를 신기술로 해결하고 있음

# NAT&PAT

- NAT 동작 방식
  - ✓ 동작 순서



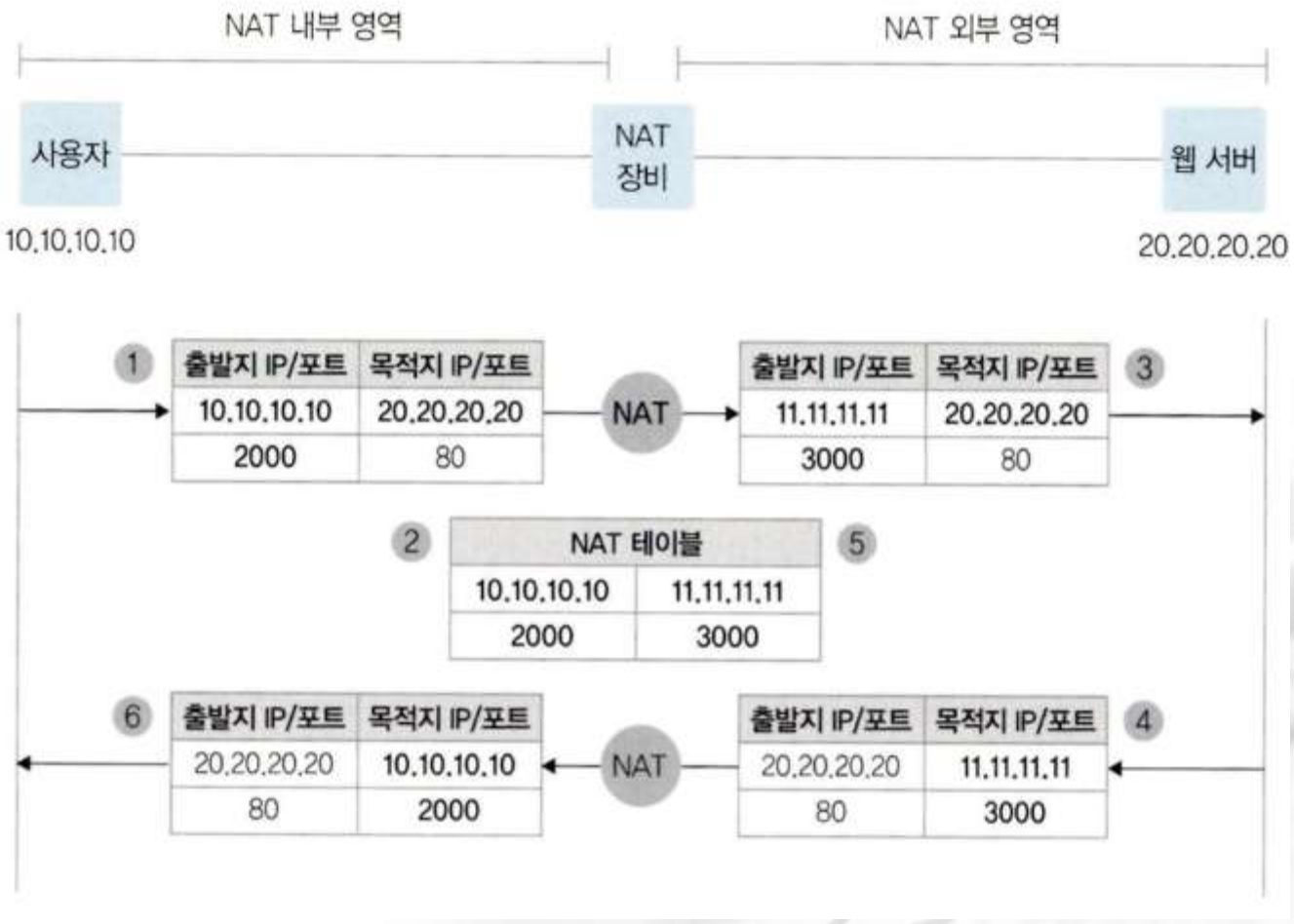
# NAT&PAT

- NAT 동작 방식
  - ✓ 동작 순서
    1. 사용자는 웹 서버에 접근하기 위해 출발지 IP를 10.10.10.10 으로 목적지 IP와 서비스 포트는 20.20.20.20 과 80 으로 패킷을 전송하는데 출발지 서비스 포트는 임의의 포트로 할당하는데 여기서는 2000번 포트로 가정
    2. NAT 역할을 수행하는 장비에서는 사용자가 보낸 패킷을 수신한 후 NAT 정책에 따라 외부 네트워크와 통신이 가능한 공인 IP인 11.11.11.11로 IP 주소를 변경한 후 NAT 장비에서 변경 전후의 IP 주소는 NAT 테이블에 저장
    3. NAT 장비에서는 출발지 주소를 11.11.11.11로 변경해 목적지 웹 서버로 전송
    4. 패킷을 수신한 웹 서버는 사용자에게 응답을 보내는데 응답이므로 수신한 내용과 반대로 출발지는 웹 서버(20.20.20.20)가 되고 목적지는 NAT 장비에 의해 변환된 공인 IP 11.11.11.11로 사용자에게 전송
    5. 웹 서버로부터 응답 패킷을 수신한 NAT 장비는 자신의 NAT 테이블에서 목적지 IP 에 대한 원래 패킷을 발생시킨 출발지 IP 주소가 10.10.10.10인 것을 확인
    6. NAT 변환 테이블에서 확인된 원래 패킷 출발지 IP(10.10.10.10)로 변경해 사용자에게 전송하면 사용자는 최종적으로 패킷을 수신

# NAT&PAT

❖ PAT 동작 방식

✓ 동작 순서



# NAT&PAT

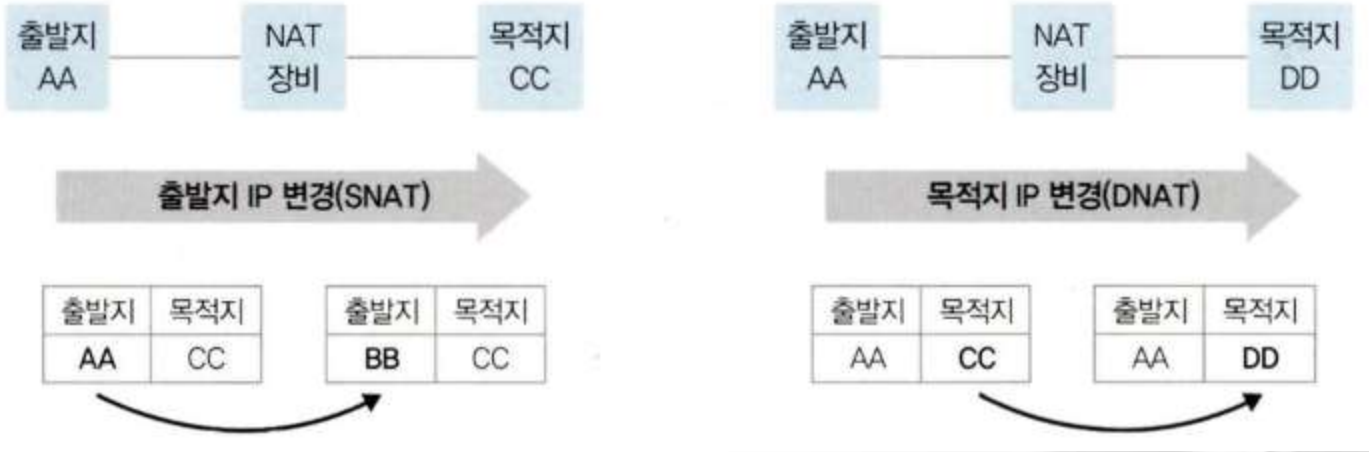
- PAT 동작 방식
  - ✓ 동작 순서
    1. 사용자는 웹 서버에 접근하기 위해 출발지 IP를 10.10.10.10 목적지 IP와 서비스 포트는 20.20.20.20 과 80 으로 패킷을 전송하는데 출발지 서비스 포트는 임의의 포트로 할당하는데 여기서는 2000번 포트로 가정
    2. NAT 장비는 사용자가 보낸 패킷을 받아 외부 네트워크와 통신이 가능한 공인 IP인 11.11.11.11로 변경하지만 출발지에 있는 다수의 사용자가 동일한 공인 IP로 변환되어야 하므로 패킷의 주소 변경 시 출발지 IP뿐만 아니라 출발지의 서비스 포트도 변경되는데 출발지 IP와 출발지 서비스 포트는 NAT 장비에 의해 모두 변경되고 NAT 장비가 이 변경 정보를 NAT 테이블에 기록
    3. NAT 장비에서는 출발지 주소를 11.11.11.11 과 서비스 포트 3000 으로 변경해 목적지 웹 서버로 전송
    4. 패킷을 수신한 웹 서버는 사용자에게 응답을 보내는데 출발지 주소는 웹 서버(20.20.20.20)가 되고 목적지는 NAT 장비에 의해 변환된 공인 IP 11.11.11.11와 서비스 포트 사용자에게 전송
    5. 웹 서버로부터 응답 패킷을 수신한 NAT 장비는 NAT 테이블을 확인해 웹 서버로부터 받은 패킷의 목적지 IP 주소인 11.11.11.11이 원래 10.10.10.10이며 서비스 포트 3000이 원래 2000인 것을 확인
    6. NAT 장비는 NAT 테이블에서 확인한 목적지 IP 주소와 서비스 포트로 패킷을 재작성한 후 사용자에게 전달하는데 사용자는 NAT 장비에서 역변환된 패킷을 받아 웹 페이지를 표시

# NAT&PAT

- ❖ PAT 동작 방식
  - ✓ 동작 방식
    - 다른 IP를 가진 사용자가 동일하게 20.20.20.20 서버로 접속하는 경우 NAT 장비에서는 출발지 IP만 11.11.11.11로 동일하게 변경하고 서비스 포트는 다른 포트로 변경
    - PAT 동작 방식은 NAT와 거의 동일하게 이루어지지만 IP 주소 뿐 만 아니라 서비스 포트까지 함께 변경해 NAT 테이블을 관리하므로 하나의 IP만으로도 다양한 포트 번호를 사용해 사용자를 구분할 수 있음
    - 서비스 포트의 개수는 제한되어 있어 재사용되는데 만약 서비스 포트가 동시에 모두 사용 중이거나 재사용할 수 없을 때는 PAT가 정상적으로 동작하지 않음
    - 동시 사용자가 매우 많을 때는 PAT에서 사용하는 Public IP 주소를 IP 하나가 아닌 풀로 구성해야 함
    - PAT는 다수의 IP가 있는 출발지에서 목적지로 갈 때 NAT 테이블이 생성되고 응답에 대해 NAT 테이블을 참조할 수 있지만 PAT IP가 목적지일 때는 해당 IP가 어느 IP에 바인딩되는지 확인할 수 있는 NAT 테이블이 없으므로 사용할 수 없음

# NAT&PAT

- SNAT & DNAT
  - ✓ 개요
    - SNAT(Source NAT): 출발지 주소를 변경하는 NAT
    - DNAT(Destination NAT): 도착지 주소를 변경하는 NAT



# NAT&PAT

- SNAT & DNAT
  - ✓ 개요
    - SNAT와 DNAT는 트래픽이 출발하는 시작 지점을 기준으로 구분
    - 어떤 주소를 변경해야 하는지는 서비스 흐름과 목적에 따라 결정
    - SNAT와 DNAT의 기준은 NAT가 수행되기 이전의 트래픽이 출발하는 시작 지점
    - 요청 시 SNAT를 해 목적지로 전송하면 해당 트래픽에 대한 응답을 받을 때는 출발지와 목적지가 반대가 되므로 DNAT가 되는데 이때 트래픽을 요청하는 시작 지점만 고려해 SNAT 설정을 해야 함
    - NAT 장비를 처음 통과할 때 NAT 테이블이 생성되므로 응답 패킷이 NAT 장비에 들어오면 별도의 NAT 설정이 없더라도 NAT 테이블을 사용해 반대로 패킷을 변환해줄 수 있기 때문인데 이 과정을 역 NAT 라고 하며 NAT가 정상적으로 수행되려면 역 NAT 과정이 함께 수행되어야 함

# NAT&PAT

- ❖ SNAT & DNAT
  - ✓ 용도
    - ● SNAT
      - ◆ SNAT는 내부망에서 외부망으로 통신할 때 많이 사용
        - ▪ 외부망 주소의 목적지에서 출발지로 다시 응답을 받으려면 출발지 IP 주소 경로가 필요한데 외부망에서는 내부망으로의 경로를 알 수 없으므로 외부망의 목적지로 서비스를 요청할 때 출발지에서는 내부 IP를 외부 IP로 NAT 해 서비스를 요청
        - ▪ 공유기처럼 PAT를 사용하는 경우에 해당
      - ◆ 보안상 SNAT를 사용
        - ▪ 회사에서 다른 대외사와 통신 시 내부 IP 주소가 아니라 별도의 다른 IP로 전환해 전송함으로써 대외에 내부의 실제 IP 주소를 숨길 수 있으며 대외사와 통신해야 하는 Private IP가 대외사의Private IP 대역과 중복될 때 SNAT를 통해 중복되지 않는 다른 IP로 변경해 통신하는 데 사용할 수 있음
        - ▪ 내부망에서 외부망으로 통신해야 하는 경우와 비슷하지만 이 경우에는 변경되는 IP가 반드시 Public일 필요는 없음
        - ▪ 로드 밸런서의 구성에 따라 SNAT를 사용: 출발지와 목적지 서버가 동일한 대역일 때는 로드 밸런서 구성에 따라 트래픽이 로드 밸런서를 거치지 않고 응답할 수 있어 SNAT를 통해 응답 트래픽이 로드 밸런서를 거치게 할 수 있음

# NAT&PAT

- ❖ SNAT & DNAT
  - ✓ 용도
    - DNAT
      - ◆ 서비스 VIP(Virtual IP)로 서비스를 요청하고 로드 밸런서에서는 서비스 VIP를 로드 밸런싱될 서버의 실제 IP로 DNAT해 내보냄
      - ◆ 사내가 아닌 대외망과의 네트워크 구성에도 DNAT를 사용하는데 사내 IP 주소는 중앙에서 일괄적으로 관리되므로 IP가 중복되는 경우가 없지만 사내가 아닌 대외망과의 연동에서는 IP가 중복될 수 있는데 설사 IP가 중복되지 않더라도 IP 주소가 제각각이므로 신규 대외사 와의 연동마다 라우팅을 개별적으로 설정해야 하는 번거로움이 발생하지만 이 경우 대외망에 NAT 장비를 이용해 대외사의 IP를 특정 IP 대역으로 NAT를 수행해야 하면 사내에서는 어떤 대외사든 대외망 전용 NAT 대역으로 변경된 네트워크 대역으로 라우팅을 처리하면 되므로 대외사 추가에 따라 별도 라우팅을 개별적으로 설정할 필요가 없고 사내 IP와 중복되는 IP가 있더라도 라우팅 이슈 없이 구성할 수 있음

# NAT&PAT

- ❖ 동적 NAT & 정적 NAT
  - ✓ 개요
    - 출발지와 목적지의 IP를 미리 매핑해 고정해 놓은 NAT가 정적 NAT이고 반대로 출발지나 목적지 어느 경우든 사전에 정해지지 않고 NAT를 수행할 때 IP를 동적으로 변경하는 것이 동적 NAT
    - 동적 NAT는 출발지와 목적지가 모두 정의된 것이 아니라 다수의 IP 풀에서 정해지므로 최소한 출발지나 목적지 중 한 곳이 다수의 IP로 구성된 IP 풀이나 Range로 설정되어 있어서 NAT가 필요할 때 IP 풀에서 어떤 IP로 매핑될 것인지 판단해 NAT를 수행하는 시점에 NAT 테이블을 만들어 관리하는데 NAT 테이블은 설정된 시간 동안 유지되고 일정 시간 동안 통신이 없으면 다시 사라지므로(NAT 테이블 타임아웃) 동적 NAT의 설정은 서비스 흐름을 고려해 적용
    - 정적 NAT는 출발지와 목적지 매핑 관계가 특정 IP로 사전에 정의된 것이므로 1:1 NAT라고 부르기도 하는데 실제 IP 매핑도 A라는 IP와 B라는 IP가 항상 고정되어 매핑된 상태이므로 서비스 방향에 따라 고려할 필요가 없는데 방향성 없이 서비스 흐름을 고려하지 않고 NAT를 설정 할 수 있음

# NAT&PAT

❖ 동적 NAT & 정적 NAT

✓ 비교

| | 동적 NAT | 정적 NAT |
|---|---|---|
| 설정 | 1:N, N:1, N:M | 1:1 |
| 테이블 | NAT 수행 시 생성 | 사전 생성 |
| 테이블 타임아웃 | 동작 | 없음 |
| 수행 정보 | 실시간으로 확인하거나 별도 로그 저장 필요 | 별도 필요 없음 |

✓ 테이블

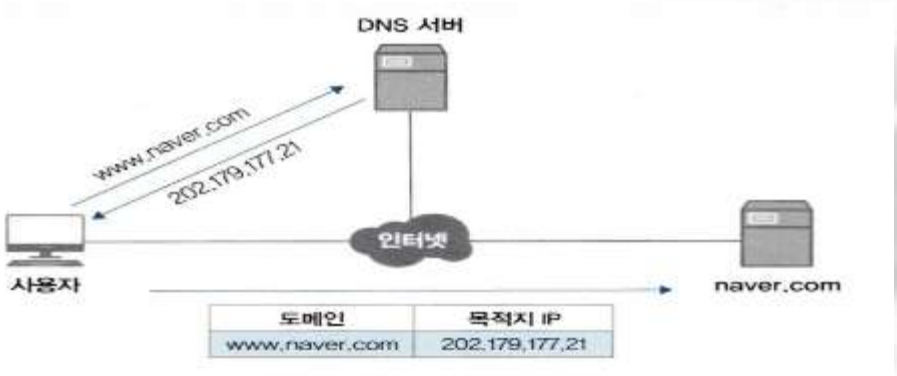

| NAT 테이블 | | |
|---|---|---|
| 실제 IP | NAT IP | 타임아웃 |
| 10.10.10.13 | 11.11.11.1 | 200 |
| 10.10.10.11 | 11.11.11.2 | 3 |


| NAT 테이블 | | |
|---|---|---|
| 실제 IP | NAT IP | 타임아웃 |
| 10.10.10.11 | 11.11.11.1 | -(영구) |
| 10.10.10.12 | 11.11.11.2 | -(영구) |
| 10.10.10.13 | 11.11.11.3 | -(영구) |

# DNS

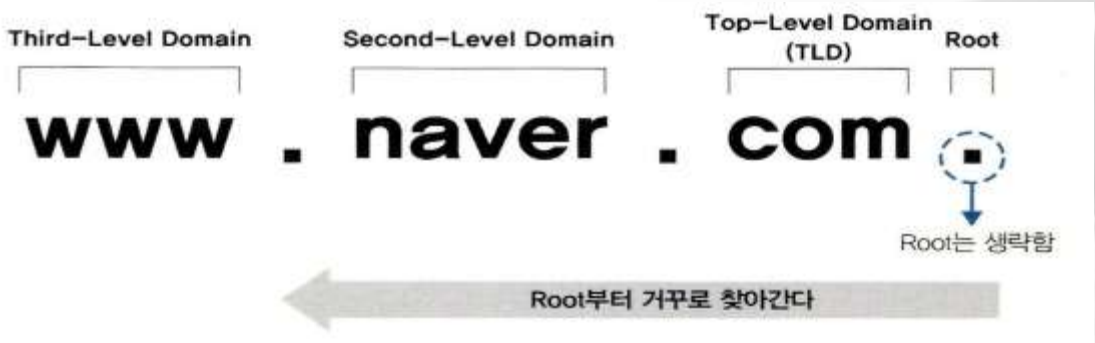
❖ 개요

- ✓ DNS(Domain Name System)는 우리가 흔히 사용하는 웹사이트 주소(예: naver.com)를 컴퓨터가 이해할 수 있는 IP 주소(예: 202.179.177.21)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 naver.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.naver.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

- ❖ 구조 와 명명 규칙
  - ✓ 개요
    - 도메인은 계층 구조여서 수많은 인터넷 주소 중 원하는 주소를 효율적으로 찾아갈 수 있음
    - 역트리 구조로 최상위 루트부터 Top-Level 도메인과 같이 하위 레벨로 원하는 주소를 단계적으로 찾아감
    - 우리가 도메인 주소를 사용할 때는 각 계층의 경계를 .으로 표시하고 뒤에서 앞으로 해석
    - third.second.top.과 같은 형태로 표현하고 Top-Level인 com 그리고 Second-Level인 naver 다음으로 Third-Level인 www와 같이 뒤에서 앞으로 해석
    - 도메인 계층은 최대 128계층까지 구성할 수 있으며 계층별 길이는 최대 63바이트까지 사용할 수 있고 도메인 계층을 구분하는 구분자 .을 포함한 전체 도메인 네임의 길이는 최대 255바이트까지 가능하며 알파벳, 숫자, - 만 사용 가능하며 대소문자 구분 없음
    - 도메인 계층



# DNS

- ❖ 구조 와 명명 규칙
  - ✓ 루트 도메인
    - 루트 도메인은 도메인을 구성하는 최상위 영역
    - DNS 서버는 사용자가 쿼리한 도메인에 대한 값을 직접 갖고 있거나 캐시에 저장된 정보를 이용해 응답
    - DNS 서버에 해당 도메인의 정보가 없으면 루트 도메인을 관리하는 루트 DNS에 쿼리
    - 루트 DNS는 전 세계에 13개가 있고 DNS 서버를 설치하면 루트 DNS의 IP 주소를 기록한 힌트(Hint) 파일을 가지고 있어 루트 DNS 관련 정보를 별도로 설정할 필요가 없음

# DNS

- ❖ 구조 와 명명 규칙
  - ✓ 루트 도메인
    - ● 전 세계 루트 서버 목록 과 루트 서버 관리 기관

a.root-servers.net: Verisign (베리사인) - 미국의 글로벌 보안 및 도메인 관리 기업
b.root-servers.net: USC-ISI (남가주대학교 정보과학연구소) - 미국의 교육 및 연구 기관
c.root-servers.net: Cogent Communications (코전트) - 미국의 대형 통신 인프라 기업 (ISP)
d.root-servers.net: University of Maryland (메릴랜드 대학교) - 미국의 교육 및 연구 기관
e.root-servers.net: NASA Ames Research Center(미항공우주국) - 미국의 연방 정부 연구 기관
f.root-servers.net: ISC(Internet Systems Consortium) - 미국의 인터넷 표준 기술 개발 비영리 단체
g.root-servers.net: US DoD DISA (미 국방부 국방정보시스템국) - 미국의 정부/군사 기관
h.root-servers.net: U.S. Army Research Lab (미 육군 연구소) - 미국의 정부/군사 기관
i.root-servers.net: Netnod (넷노드) - 스웨덴의 인터넷 인프라 및 교환 노드 운영사
j.root-servers.net: Verisign (베리사인) - 미국의 글로벌 보안 및 도메인 관리 기업 (A와 동일)
k.root-servers.net: RIPE NCC (유럽 IP 네트워크 협조 센터) - 유럽·중동 지역의 인터넷 주소 자원 관리 기구
l.root-servers.net: ICANN (국제인터넷주소관리기구) - 전 세계 인터넷 식별자를 총괄하는 국제 비영리 기구
m.root-servers.net: WIDE Project (와이드 프로젝트) - 일본의 산학협력 인터넷 연구 컨소시엄

# DNS

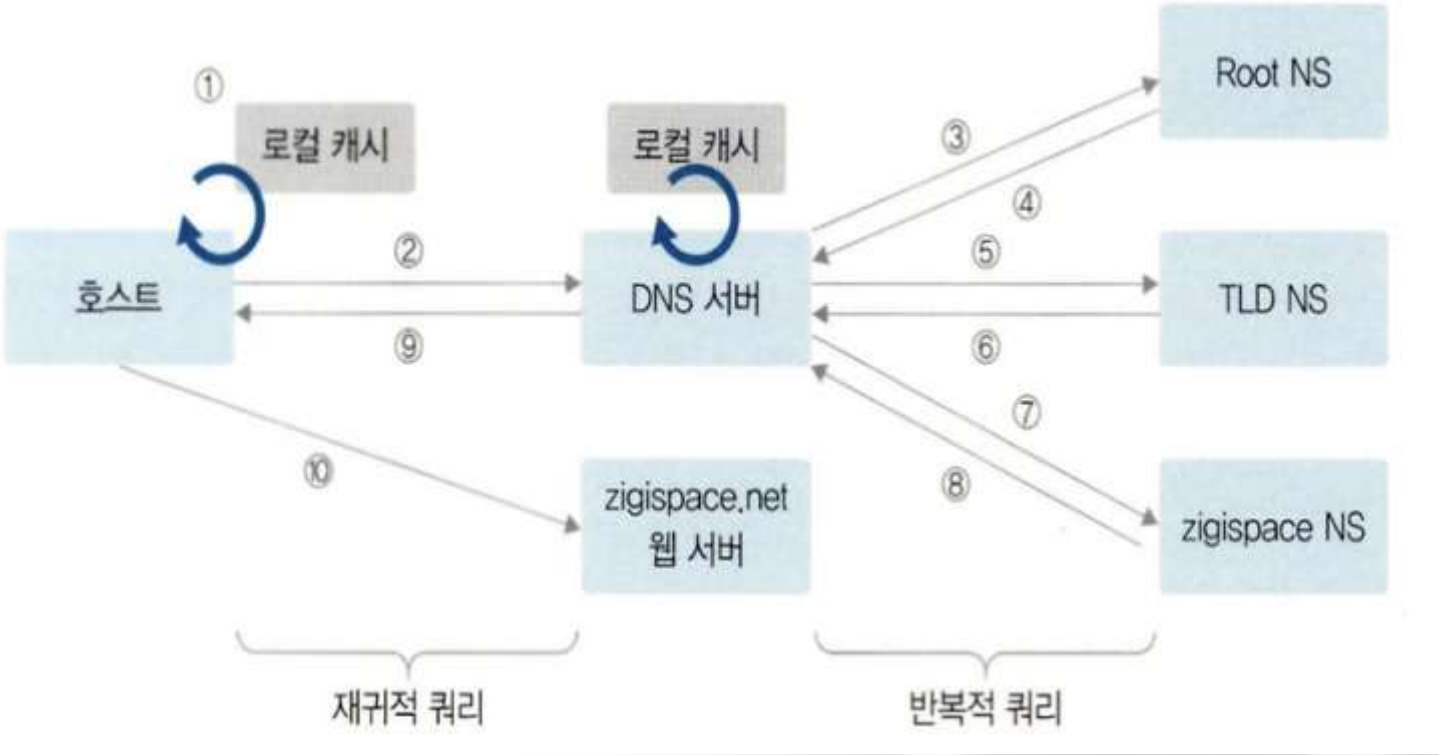
- ❖ 구조 와 명명 규칙
  - ✓ Top-Level Domain(TLD)
    - 최상위 도메인인 TLD는 IANA(Internet Assigned Winibers Authority)에서 구분한 6가지 유형으로 구분
    - 각 유형은 IANA 사이트(https://www.iana.org/domains/root/db) 에서 확인
    - 유형
      - ◆ Generic TLD(gTLD): gTLD는 특별한 제한없이 일반적으로 사용되는 최상위 도메인이며 세 글자 이상으로 구성되는데 초기 gTLD는 1980년대 7개의 gTLD(.com, .edu, .gov, .int, .mil, .net, .org)로 시작했으며 필요에 의해 새로운 gTLD가 지속적으로 만들어지고 있음
      - ◆ Country Code TLD(ccTLD)
        - ccTLD는 국가 최상위 도메인으로 ISO 3166 표준에 의해 규정된 두 글자의 국가 코드를 사용
        - 우리나라는 kr을 사용
        - 일반적으로 ccTLD를 사용하는 경우 Second Level TLD에는 gTLD에서 구분한 것처럼 사이트 용도에 따른 코드를 사용하는데 예를 들어 일반 회사는 co.kr을 사용하고 정부기관은 go.kr을 사용하는 방법
        - 우리나라는 gTLD를 두 글자로 줄여 사용하지만 호주나 대만처럼 gTLD를 그대로 사용하는 나라도 있습니다(com.au, gov.au, com.tw, gov.tw 등)
        - 영국은 ISO 5166 표준이 아닌 uk라는 별도 ccTLD를 사용

# DNS

- ❖ 구조 와 명명 규칙
  - ✓ Top-Level Domain(TLD)
    - ● 유형
      - ◆ Sponsored(sTLD)
        - ▪ sTLD는 특정 목적을 위한 스폰서를 두고 있는 최상위 도메인
        - ▪ 특정 민족 공동체, 전문가 집단, 지리적 위치 등이 속할 수 있음
        - ▪ sTL의 종류에는 aero, asia, edu, museum 등 이 있음
      - ◆ Infrastructure
        - ▪ 운용상 중요한 인프라 식별자 공간을 지원하기 위해 전용으로 사용되는 최상위 도메인
        - ▪ TLD는 arpa
      - ◆ Generic-restricted(grTLD)
        - ▪ grTLD는 특정 기준을 충족하는 사람이나 단체가 사용할 수 있는 최상위 도메인
        - ▪ grTLD의 종류에는 biz, name, pro 등
      - ◆ Test(tTLD)
        - ▪ tTLD는 개발 프로세스에서 테스트 목적으로 사용되는데 test가 있음

# DNS

❖ 동작 방식


①
로컬 캐시
로컬 캐시
Root NS
③
④
②
⑤
호스트
DNS 서버
TLD NS
⑨
⑥
⑦
⑩
⑧
zigispace.net
웹 서버
zigispace NS
재귀적 쿼리
반복적 쿼리

# DNS

- ❖ 주요 레코드
  - ✓ 종류
    - A(IPv4) 레코드
      - ◆ A 레코드는 기본 레코드로 도메인 주소를 IP 주소로 변환하는 레코드
      - ◆ 사용자가 DNS에 질의한 도메인 주소를 A 레코드에 설정된 IP 주소로 응답
      - ◆ 하나의 A 레코드에는 한 개의 도메인 주소와 한 개의 IP 주소가 1:1로 매핑되는데 동일한 도메인을 가진 A 레코드를 여러 개 만들어 서로 다른 IP 주소와 매핑할 수 있음
      - ◆ 반대로 다수의 도메인에 동일한 IP를 매핑한 A 레코드를 만들 수 있음
      - ◆ 서버 한 대에 여러 웹 서비스를 구동해야 한다면 여러 도메인에 동일한 IP를 매핑하고 HTTP 헤더의 HOST 필드에 도메인을 명시해 웹 서버를 구분해 서비스 할 수 있음
    - AAAA(IPv6) 레코드
      - ◆ A 레코드가 IPv4 주소 체계에서 사용되는 레코드라면 AAAA 레코드는 IPv6 주소 체계에서 사용되는 레코드
      - ◆ 역할은 A 레코드와 동일

# DNS

- ❖ 주요 레코드
  - ✓ 종류
    - CNAME(Canonical Name) 레코드
      - ◆ CNAME 레코드는 별칭 이름을 사용하게 해주는 레코드
      - ◆ 레코드값에 IP 주소를 매핑하는 A 레코드와 달리 CNAME 레코드는 도메인 주소를 매핑
      - ◆ 네임 서버가 CNAME 레코드에 대한 질의를 받으면 CNAME 레코드에 설정된 도메인 정보를 확인하고 그 도메인 정보를 내부적으로 다시 질의한 결과 IP 값을 응답
      - ◆ CNAME 레코드의 대표적인 예로 www 가 있음
      - ◆ zigispace.net라는 웹사이트에 접속하려면 보통 zigispace.net 또는 www.zigispace.net 으로 접속하는데 . zigispace.net 과 www.zigispace.net을 각각 A 레코드로 매핑하면 IP 주소가 변경될 때 두 개의 레코드값을 모두 변경해야 하지만 zigispace.net만 A 레코드로 IP 주소를 매핑하고 www.zigispace.net은 CNAME으로서 zigispace.net으로 매핑하면 IP 주소가 변경될 때 zigispace.net만 변경해도 동일한 결괏값을 가져올 수 있음

# DNS

- ❖ 주요 레코드
  - ✓ 종류
    - SOA(Start Of Authority) 레코드
      - ◆ 도메인 영역에 대한 권한을 나타내는 레코드
      - ◆ 현재 네임 서버가 이 도메인 영역에 대한 관리 주체임을 의미하므로 해당 도메인에 대해서는 다른 네임 서버에 질의하지 않고 직접 응답
      - ◆ 도메인 영역 선언 시 SOA 레코드는 필수 항목이므로 반드시 만들어야 함
      - ◆ 도메인 영역에 SOA 레코드를 만들지 않으면 해당 도메인은 네임 서버에서 정상적으로 동작할 수 없음
      - ◆ 그밖에 SOA 레코드는 현재 도메인 관리에 필요한 속성값을 설정하는데 도메인 동기화에 필요한 타이머 값이나 TTL 값과 함께 도메인의 네임 서버나 관리자 정보도 SOA 레코드에서 설정
    - NS(Name Server) 레코드
      - ◆ 도메인에 대한 권한이 있는 네임 서버 정보를 설정하는 레코드
      - ◆ NS 레코드의 경우 권한이 있는 네임 서버 정보를 해당 도메인에 설정하는 역할 외에 하위 도메인에 대한 권한을 다른 네임 서버로 위임(Delegate)하는 역할로도 많이 사용

# DNS

- ❖ 주요 레코드
  - ✓ 종류
    - MX(Mail exchange) 레코드
      - ◆ 메일 서버를 구성할 때 사용되는 레코드
      - ◆ 메일 서버에서 메일을 보낼 때는 MX 레코드를 참조해 동작하는데 우선순위 값을 이용해 다수의 MX 레코드를 선언할 수 있음
      - ◆ 우선순위가 높은(값이 적은) 서버로 메일을 보내고 실패하면 다음 순서의 MX 레코드의 메일 서버에서 처리
    - PTR(Pointer) 레코드
      - ◆ IP 주소에 대한 질의를 도메인 주소로 응답하기 위한 레코드
      - ◆ A 레코드가 정방향 조회용 레코드라면 PTR 레코드는 역방향 조회용 레코드
      - ◆ 하나의 IP에 대한 PTR 레코드는 오직 하나만 가짐
      - ◆ PTR 레코드는 주로 화이트 도메인 구성용으로 사용
    - TXT(TeXT) 레코드
      - ◆ 도메인에 대한 설명과 같이 간단한 텍스트를 입력할 수 있는 레코드
      - ◆ 이 레코드를 특정 기능으로 사용할 수도 있는데 주로 사용되는 곳은 화이트 도메인을 위한 SPF 레코드
      - ◆ TXT 레코드에는 공백도 포함할 수 있고 대소문자를 구분하며 255자까지 사용 가능

# DNS

- ❖ 기타
  - ✓ 도메인 위임
    - 도메인은 그 도메인에 대한 정보를 관리할 수 있는 네임 서버를 지정하지만 도메인 내의 모든 레코드를 그 네임 서버가 직접 관리하지 않고 일부 영역에 대해서는 다른 곳에서 레코드를 관리하도록 위임하기도 하는데 이 방식이 도메인 위임
    - 자신이 가진 도메인 관리 권한을 다른 곳으로 일부 위임해 위임한 곳에서 세부 레코드를 관리하도록 하는 것
    - CDN을 이용하거나 GSLB를 사용하는 것이 대표적인 경우
    - 도메인은 계층 구조여서 특정 계층의 레코드를 위임하면 해당 레코드의 하위 계층은 함께 위임 처리
    - 도메인 위임 기능을 쓰면 특정 영역을 다른 네임 서버에서 관리할 수 있는 권한을 위임해서 특정 영역에 대한 관리 주체를 분리하는 용도로 사용할 수 있어 계열사에서 특정 도메인을 분리하거나 GSLB 등 다양한 용도로 사용할 수 있음

# DNS

- 기타
  - ✓ TTL
    - 도메인의 TTL(Timc To I.ive) 값은 DNS에 질의해 응답받은 결괏값을 캐시에서 유지하는 시간
    - 로컬 캐시에 저장된 도메인 정보를 무작정 계속 갖고 있는 것이 아니라 DNS에 설정된 TTL 값에 따라 그 시간만 로컬 캐시에 저장
    - DNS 서버에서 TTL 값을 늘려 캐시를 많이 이용하면 DNS 재귀적 쿼리로 인한 응답 시간을 많이 줄일 수 있고 결과적으로 전체적인 네트워크 응답 시간이 단축되지만 DNS에서 해당 도메인 관련 정보가 변경되었을 때 TTL 값이 크면 새로 변경된 값으로 DNS 정보 갱신이 그만큼 지연되는 단점이 발생
    - TTL 값이 너무 작으면 DNS의 정보 갱신이 빨라지므로 DNS 쿼리량이 늘어나 DNS 서버 부하가 증가 할 수 있음
    - 서비스의 성질과 도메인 정보의 갱신 빈도에 따라 TTL 값을 적절히 조절하는 것이 좋음
    - 변경이 빈번하지 않다면 TTL 값을 늘려 DNS 부하를 줄이는 것이 좋고 IDC 이전이나 Public IP 또는 서비스 변경이 예정되어 있다면 DNS의 TTL 값을 미리 극도로 줄여 변경을 신속히 적용하는 것이 좋음
    - 윈도우 기본값은 1시간이고 리눅스는 3시간

# DNS

- 기타
  - ✓ 도메인 관련 시간
    - refresh: 보조 네임서버에서 Zone Transfer를 통해 정보를 주기적으로 받아오는 주기
    - retry: 보조 네임 서버가 주 네임 서버로 접근이 불가능할 때 재시도하는 주기
    - expire(다음 날짜 이후 만료): 보조 네임 서버가 주 네임 서버로부터 도메인 정보를 받아오지 못할 때 유지되는 시간으로 해당 시간 동안 도메인 관련 정보를 받아오지 못하면 주 네임 서버에서 삭제된 것으로 간주하고 보조 네임 서버에서도 해당 도메인 정보를 삭제
  - ✓ 화이트 도메인
    - 한국인터넷진흥원(K1SA)에서는 불법적인 방법으로 발송되는 스팸메일 차단 활동을 하고있는데  이를 위해 정상적인 도메인을 인증 관리하는 제도가 '화이트 도메인'
    - 반대로 불법적인 스팸메일을 발송하는 사이트를 실시간 블랙리스트 정보로 관리해 메일 발송을 제한하는데 이 실시간 블랙리스트를 RBL(Realltime Blackhole List. Realtime Blocking List)이라고 합니다
    - 현재 보유 중인 도메인을 화이트 도메인으로 등록하려면 KISA RBL 사이트에서 화이트 도메인으로 등록해야 하는데 이를 위해 사전에 해당 도메인에 SPF 레코드(Sender Policy Framework)가 설정되어 있어야 함
    - SPF 레코드를 통해 사전에 메일 서버 정보를 공개하면 수신 측 메일 서버에서는 해당 도메인을 통해 발송된 메일이 실제 메일 서버에 등록된 정보와 일치하는지 확인할 수 있음

# DNS

- 기타
  - ✓ 한글 도메인
    - 도메인 주소는 영문뿐만 아니라 한글로 주소를 만들 수 있음
    - 사용자가 도메인을 한글로 등록하고 사용하기 위해 DNS에서는 해당 한글을 퓨니코드로 변경하고 이 퓨니코드로 DNS에 도메인을 생성
    - 퓨니코드: 애플리케이션 국제화 도메인 네임(IDNA)기반 하에서 다국어 도메인이 아스키로 변환Encoding ) 된 구문

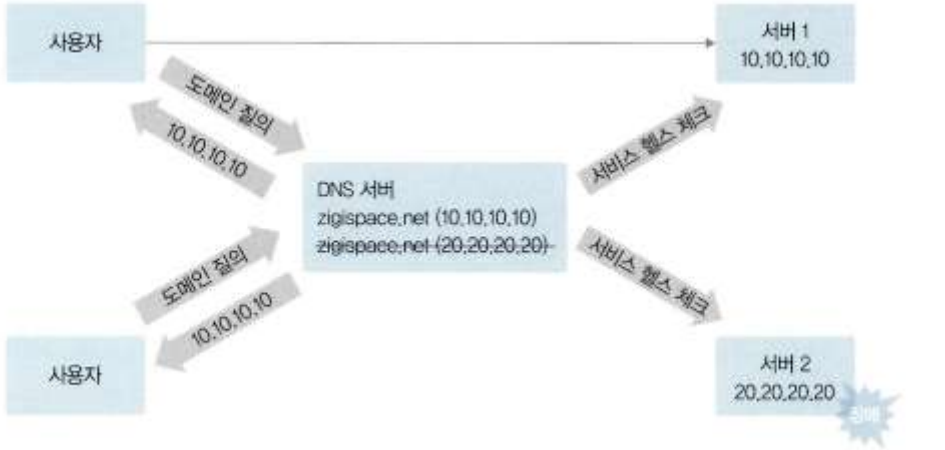
# GSLB

- DNS Load Balancing
  - ✓ DNS에서 동일한 레코드 이름으로 서로 다른 IP 주소를 동시에 설정할 수 있는데 이렇게 설정하면 도메인 질의에 따라 응답받는 IP 주소를 나누어 로드밸런싱할 수 있으며 이것이 DNS 로드밸런싱
  - ✓ DNS만 이용한 로드밸런싱으로는 정상적인 서비스를 할 수 없는데 DNS는 설정된 서비스 상태의 정상 여부를 확인하지 않고 도메인에 대한 질의에 대해 설정된 값을 무조건 응답하므로 DNS에 저장된 레코드와 매핑된 서비스가 모두 정상일 때는 문제가
  - ✓ 없지만 특정 서비스에 문제가 있을 때 DNS 서버는 이것을 감지하지 못해 사용자의 도메인 질의 요청에 비정상 상태인 서비스 IP 주소를 응답한 경우 사용자는 해당 서비스에 접근할 수 없음
  - ✓ DNS 서버에서는 각 레코드에 대한 서비스 체크가 이루어지지 않고 설정된 값에 따라 동작하므로 서비스 가용성 향상 방법으로는 부적합

# GSLB

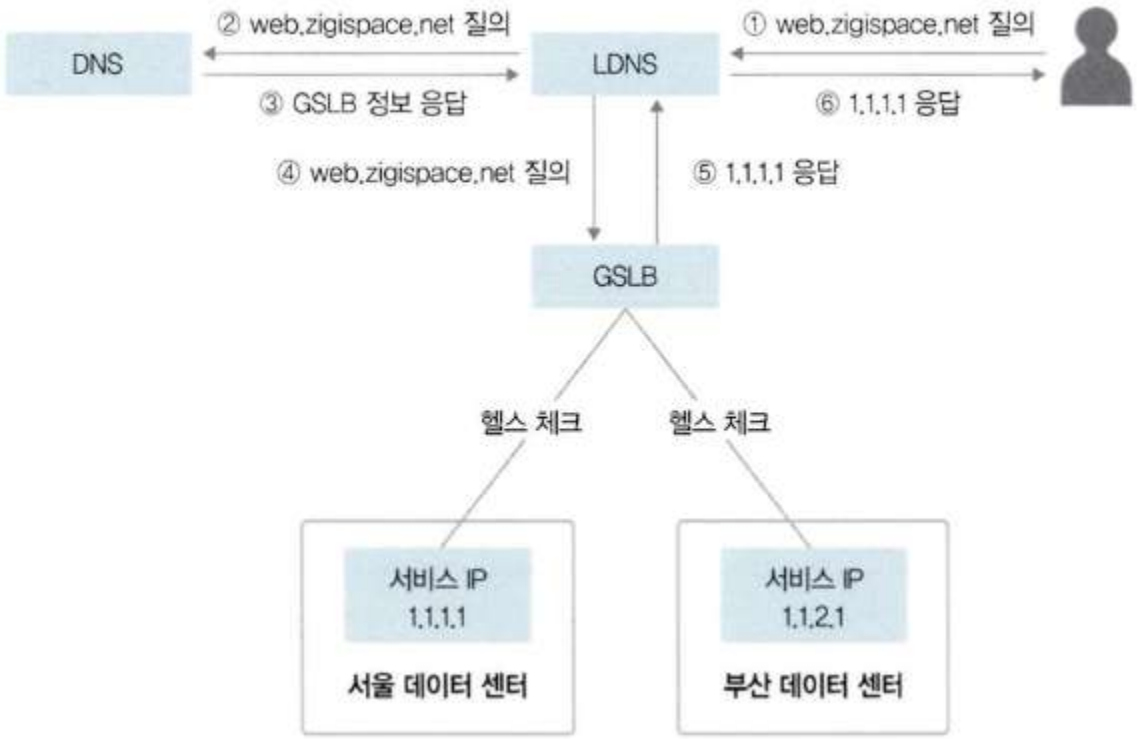
❖ GSLB 개요

- ✓ GSLB(Global Server/Servicc Load Balancing)는 DNS의 이런 문제점을 해결해 도메인을 이용한 로드밸런싱 구현을 도와줌
- ✓ GSLB는 DNS와 동일하게 도메인 질의에 응답해주는 역할과 동시에 로드 밸런서처럼 등록된 도메인에 연결된 서 비스가 정상적 인지 헬스 체크를 수행
- ✓ 등록된 도메인에 대한 서비스가 정상인지 상태를 체크해 정상인 레코드에 대해서만 사용
- ✓ 서버가 장애로 서비스가 불가능할 때 GSLB에서는 등록된 두 개의 레코드 중 장애가 발생한 서버는 응답으로 사용하지 않고 정상적인 서버만 응답에 대한 레코드로 사용
- ✓ GSLB를 인텔리전스 DNS 라고도 부름



# GSLB

❖ GSLB 도메인 질의 흐름

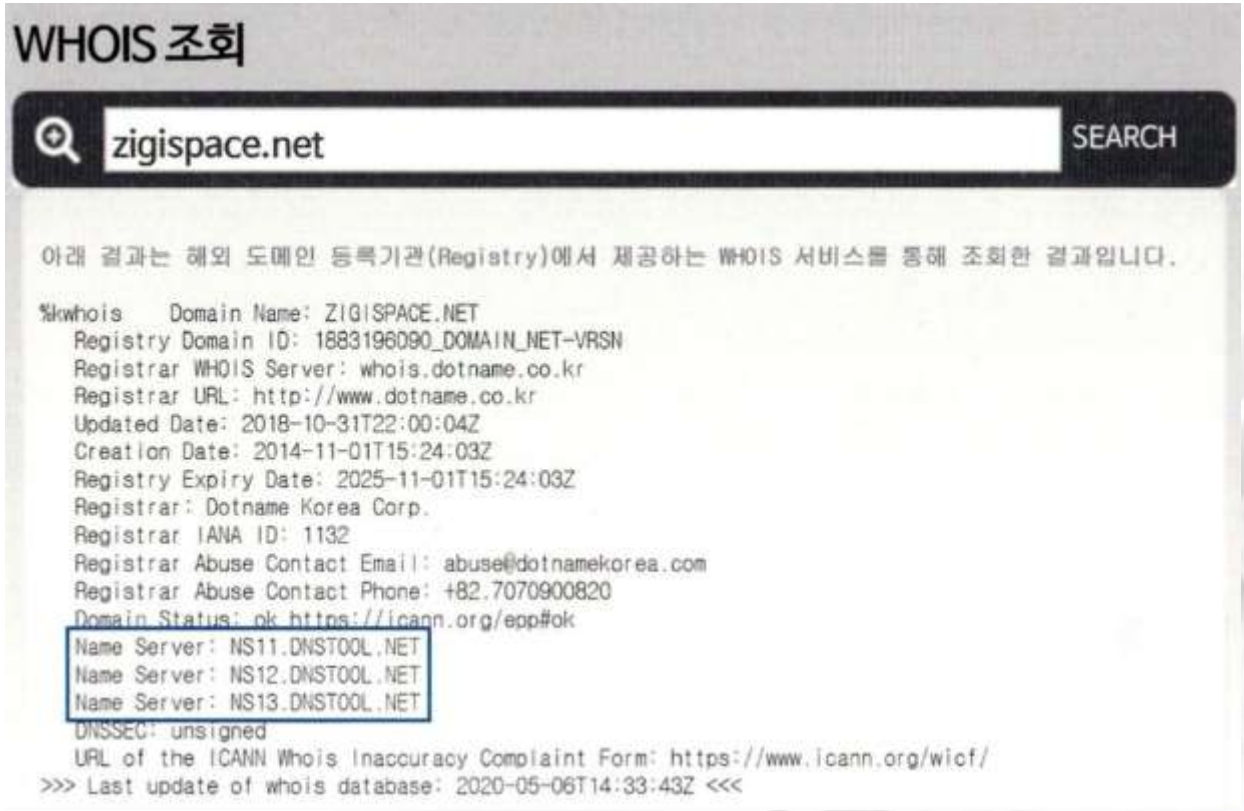


# GSLB

- GSLB 도메인 질의 흐름
  - ✓ 사용자가 web.zigispace.net에 접속하기 위해 DNS에 질의
  - ✓ DNS는 web.zigispace.net을 관리하는 NS 서버를 찾기 위해 root부터 순차 질의
  - ✓ zigispace.net을 관리하는 NS 서버로 web.zigispace.net에 대해 질의
  - ✓ DNS 서버는 GSLB로 web.zigispace.net에 대해 위임했으므로 GSLB 서버가 NS 서버라고 LDNS 에 응답
  - ✓ LDNS는 다시 GSLB로 web.zigispace.net에 대해 질의
  - ✓ GSLB는 web.zigispace.net에 대한 IP 주솟값 중 현재 설정된 분산 방식에 따라 서울 또는 부산 데이터 센터의 IP 주솟값을 DNS에 응답하는데 서울 데이터 센터의 서비스 IP인 1.1.1.1을 응답하는 것으로 가정
  - ✓ GSLB가 응답하는 값은 GSLB에서 설정한 주기에 따라 서울과 부산 데이터 센터로 헬스 체크해 정상적인 값만 응답
  - ✓ GSLB에서 결괏값을 응답받은 LDNS는 사용자에게 web.zigispace.net이 1.1.1.1로 서비스하고 있다고 최종 응답

# GSLB

- 구성 방식
  - ✓ 종류
    - 도메인 자체를 GSLB로 사용
    - 도메인 내의 특정 레코드만 GSLB를 사용
  - ✓ 도메인 자체를 GLSB로 사용



# GSLB

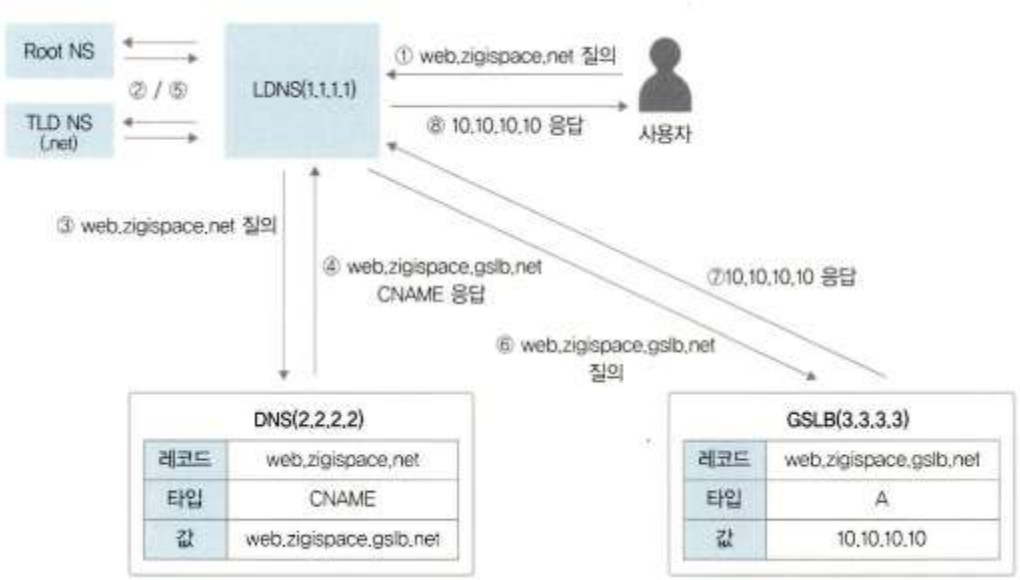
- 구성 방식
  - ✓ 도메인 자체를 GLSB로 사용
    - 도메인 자체를 GSLB로 사용하면 해당 도메인에 속하는 모든 레코드 설정을 GSLB 장비에서 관리
    - 도메인에 대한 모든 레코드를 GSLB에서 설정
    - 도메인 구입 시 도메인에 대한 권한을 갖는 네임 서버를 지정하는데 이 네임 서버가 도메인을 관리
    - 도메인 자체를 GSLB로 사용하는 것은 도메인에 대한 네임 서버를 GSLB로 지정하고 GSLB에서 도메인에 대한 모든 레코드를 등록해 처리하는 방식으로 GSLB 자체가 도메인의 네임 서버 역할
    - 도메인의 레코드 중 헬스 체크 기능이 불필요한 경우뿐만 아니라 모든 레코드에 대한 질의가 GSLB를 통해 이루어지므로 GSLB에 부하를 줌

# GSLB

- ❖ 구성 방식
  - ✓ 도메인 내의 특정 레코드만 GLSB로 사용
    - DNS에서 도메인 설정 시 GSLB를 사용하려는 레코드에 대해서만 GSLB로 처리하도록 설정
    - 대표 도메인에 속한 레코드 중 GSLB 적용이 불필요한 경우가 많아 도메인 내의 특정 레코드에 대해서만 GSLB로 처리를 이관하는 방식을 사용
    - 특정 레코드에 대해서만 GSLB로 처리를 이관하는 방법은 두 가지
      - ◆ 별칭(Alias) 사용(CNAME 레코드 사용)
      - ◆ 위임(Delegation) 사용(NS 레코드 사용)
    - 별칭을 이용해 GSLB를 사용하는 방법은 실제 도메인과 다른 별도의 도메인 레코드로 GSLB에 등록하는 것으로  외부 CDN을 사용하거나 내부에 GSLB를 사용해야 할 도메인이 많은 경우 한꺼번에 관리하기 위해 사용
    - 위임을 이용해 GSLB를 사용하는 경우는 실제 도메인과 동일한 도메인 레코드를 사용하며 도메인 전체를 위임하는 것이 대표적인 예

# GSLB

- ❖ 구성 방식
  - ✓ 도메인 내의 특정 레코드만 GLSB로 사용
    - ● 별칭(Alias) 사용(CNAME 레코드 사용)
      - ◆ DNS 서버에 CNAME 레코드로 CDN과 같은 외부 GSLB를 지정하면 CNAME 레코드의 값으로 등록된 FQDN을 GSLB로 재질의 해 서버를 찾아가는데 CNAME 값으로 등록되는 FQDN이 GSLB가 네임 서버로 등록된 도메인을 사용해 GSLB로 재질의하게 만드는 것



# GSLB

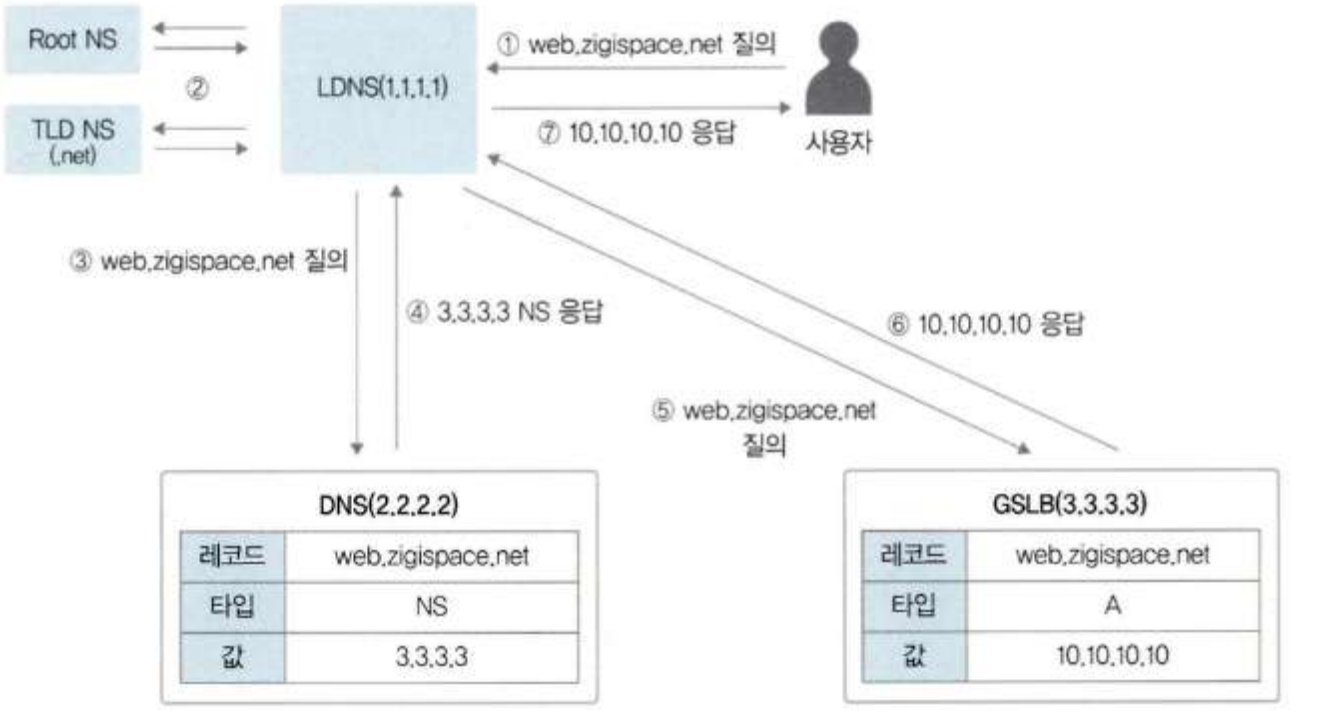
- 구성 방식
  - ✓ 도메인 내의 특정 레코드만 GLSB로 사용
    - 위임(Delegation) 사용(NS 레코드 사용)
      - NS에서 특정 FQDN에 대한 설정을 NS 레코드로 설정하면 해당 FQDN에 대한 값을 NS 레코드의 값으로 설정된 네임 서버로 재질의하는 방식으로 이때 NS 레코드에 지정된 네임 서버가 GSLB
      - NS 레코드를 이용한 위임으로 재질의하는 경우 최초 요청한 FQDN을 그대로 재질의하므로 GSLB에서 관리되는 도메인은 사용자가 최초 호출하는 동일한 FQDN이 됨
      - 사용자가 web.zigispace.net의 IP 주소를 DNS로 질의하고 이 질의를 받은 LDNS(Local DNS)는 root부터 DNS 서버를 순차적으로 찾아 web.zigispace.net을 관리하는 NS 서버에 web.zigispace.net의 IP 주소가 무엇인지 질의
      - DNS(2.2.2.2)서버에는 web.zigispace.net0] NS 레코드로 등록되어 있어 이 결괏값을 LDNS 서버로 응답
      - LDNS NS 레코드로 응답받은 GSLB 주소(3.3.3.3)에 web.zigispace.net에 대해 질의
      - GSLB(3.3.3.3)는 web.zigispace.net에 대한 A 레코드를 확인하여 이 값을 LDNS에 응답
      - LDNS는 최종적으로 사용자에게 이 결괏값을 전달

# GSLB

❖ 구성 방식

✓ 도메인 내의 특정 레코드만 GLSB로 사용

- 위임(Delegation) 사용(NS 레코드 사용)



# GSLB

- 분산 방식
  - ✓ 달성하는 목적
    - 서비스제공의가능 여부를 체크해 트래픽 분산
    - 지리적으로 멀리 떨어진 다른 데이터 센터에 트래픽 분산
    - 지역적으로 가까운 서비스에 접속해 더 빠른 서비스 제공이 가능하도록 분산
  - ✓ GSLB에서 지원되는 분산 방식은 GSLB 장비를 생산하는 벤더와 모델에 따라 조금씩 다를 수 있지만 대부분 다음 두 가지 헬스 체크 모니터링 요소를 지원
    - 서비스 응답 시간 / 지연(RTT/Latency): 서비스 요청에 대한 응답이 얼마나 빠른지 또는 지연이 얼마나 없는지를 확인하고 이것을 이용해 서비스를 분산 처리
    - IP에 대한 지리(Geography) 정보: 서비스 제공이 가능한 각 사이트의 IP 주소에 대한 Geo 값을 확인해 가까운 사이트로 서비스 분산을 처리
  - ✓ 위의 두 가지 요소에 따른 분산 방법은 다르겠지만 기본적으로 추구하는 목표는 같은데 서비스가 가능한 사이트로 트래픽을 분산하는 것은 물론 더 신속히 서비스를 제공할 수 있는 사이트로 접속할 수 있도록 유도하는 것이 궁극적 인 목표
  - ✓ 서비스 응답 시간과 사이트의 Geo 값 모두 사용자가 서비스를 요청했을 때 더 신속한 서비스 응답과 직접적인 연관이 있는 요소이기 때문
  - ✓ 이런 설정은 지리적으로 멀리 떨어진 국내와 해외 사이트로 구성된 경우 더 큰 효과를 발휘
  - ✓ 이런 응답시간이나 IP 주소에 대한 Geo 값은 사용자 기준이 아니라 사용자가 바라보는 Local DNS와 GSLB 간 값이므로 설정에 유의해야 하는데 국내 사용자가 해외 DNS 서버를 Local DNS로 활용하면 사용자의 서비스 접속 시간은 더 길어질 수 있음

# DHCP

❖ 개요

- ✓ 호스트가 네트워크와 통신하려면 물리적 네트워크 구성은 물론 IP 주소 와 같은 네트워크 정보와 DNS 주소도 설정이 필요한데 네트워크 정보를 호스트에 적용하려면 사용자가 IP 정보를 직접 설정하거나 IP 정보를 할당해주는 서버를 이용해 자동으로 설정해야 함
- ✓ 수동으로 IP와 네트워크 정보를 직접 설정하는 것을 정적 할당이라고 하고 자동으로 설정하는 것을 동적 할당 이라고 하며 일반적으로 데이터 센터의 서버 팜과 같은 운영 망에서 사용되는 IP는 주로 정적 할당을 사용하지만 PC 사용자를 위해 운영되는 사무실 네트워크에서는 IP를 자동으로 할당받는 동적 할당 방식을 많이 사용
- ✓ 보안이나 관리 목적으로 사무실의 사용자 PC를 정적 할당해 사용하는 경우가 많았지만 최근 동적 할당 방식을 이용하면서도 보안을 강화하고 쉽게 관리하도록 도와주는 서비스와 장비가 대중화되어 동적 할당 방식이 많이 사용되고 있음
- ✓ 핸드폰이나 PC를 사용할 때 IP를 별도로 설정하지 않아도 네트워크에 쉽게 접속되는데 특별한 경우를 제외하고 동적 할당 방식을 기본으로 사용하며 가정에서 사용하는 공유기도 동적 할당 방식을 기본으로 사용
- ✓ IP를 동적으로 할당하는 데 사용되는 프로토콜이 바로 DHCP(Dynaimic Host Configuration Protocol)
- ✓ DHCP를 사용하면 사용자가 직접 입력해야 하는 IP 주소, 서브넷 마스크, 게이트웨이, DNS 정보를 자동으로 할당받아 사용할 수 있음
- ✓ 별도의 IP 설정 작업이 필요없어 사용자와 관리자 모두 편리하게 네트워크에 접속할 수 있고 사용하지 않는 IP 정보는 회수되어 사용하는 경우에만 재할당되어 사용자 이동이 많고 한정된 IP 주소를 가진 경우 유용하게 사용되고 IP가 자동으로 관리되므로 사용자가 직접 입력하면서 발생할 수 있는 설정 정보 오류나 중복 IP 할당과 같은 문제를 예방할 수 있음

# DHCP

- DHCP 프로토콜
  - ✓ 개요
    - DHCP는 BOOTP(Botstrap Protocol)라는 프로토콜을 기반으로 함
    - DHCP는 BOOTP와 유사하게 동작하지만 BOOTP에서 지원되지 않는 몇 가지 기능이 추가된 확장 프로토콜
    - DHCP와 BOOTP 프로토콜 사이에는 호환성이 있어 BOOTP와 DHCP에서 사용되는 서비스 포트가 같고 BOOTP 클라이언트가 DHCP 서버를 사용하거나 DHCP 클라이언트가 BOOTP 서버를 사용해 정보를 수신할 수도 있음
    - DHCP는 서버와 클라이언트로 동작하며 클라이언트의 서비스 포트는 68(bootpc), 서버의 서비스 포트는 67(boot ps)
    - DHCP 프로토콜에 대한 자세한 내용은 RF2131(https://tools.ietf.org/html/rfc2131)에 기술되어 있고 추가적인 상위 기능은 RFC 3046, 3942에서 찾아볼 수 있음

# DHCP

- DHCP 프로토콜
  - ✓ 서버 구성
    - DHCP 서버는 윈도 서버의 DHCP 서비스를 사용하거나 리눅스의 DHCP 데몬을 사용해 구성할 수 있으며 스위치, 라우터, 방화벽, VPN과 같은 네트워크 보안 장비에서도 DHCP 서비스가 가능
    - 설정 항목
      - IP 주소 풀(IP 범위): 클라이언트에 할당할 IP 주소 범위
      - 예외 IP 주소 풀(예외 IP 범위): 클라이언트에 할당할 IP 주소로 선언된 범위 중 예외적으로 할당하지 않을 대역
      - 임대 시간: 클라이언트에 할당할 IP 주소의 기본 임대 시간
      - 서브넷 마스크(Subnet Mask): 클라이언트에 할당할 IP 주소에 대한 서브넷 마스크 정보
      - 게이트웨이(Router): 클라이언트에 할당할 게이트웨이 정보
      - DNS: 클라이언트에 할당할 DNS 주소

# 네트워크 관련 명령어

- ping(Packet InterNet Groper)
  - ✓ 옵션

| 기능 | 윈도우 (Windows) | 리눅스 (Linux) | 설명 |
|---|---|---|---|
| 횟수 지정 | -n <횟수> | -c <횟수> | 보낼 패킷의 개수를 지정 |
| 무한 핑 | -t | 기본 동작 | 종료는 ctrl + c |
| 패킷 크기 | -l <바이트> | -s <바이트> | 보낼 데이터의 크기 변경 |
| 시간 간격 | 지원 안 함 | -i <초> | 패킷을 보내는 주기 설정 |
| 타임아웃 | -w <밀리초> | -W <초> 나 -t <초> | 응답을 기다리는 제한 시간 |
| IPv4 지정 | -4 | -4 | 명시적으로 IPv4 주소를 사용 |
| IPv6 지정 | -6 | -6 | 명시적으로 IPv6 주소를 사용 |
| 소스 지정 | -S | -I | |
| 홉 기록 | -r <횟수) | | |

# 네트워크 관련 명령어

- ping(Packet InterNet Groper)
  - ✓ 샘플: 리눅스에서 ping을 사용해 목적지 8.8.8.8로 100byte 크기의 패킷을 2회 보내기
    - ping 8.8.8.8 -c 2 -s 100(Windows: ping 8.8.8.8 -n 2 -l 100)

```
PING 8.8.8.8 (8.8.8.8): 100 data bytes
108 bytes from 8.8.8.8: icmp_seq=0 ttl=113 time=41.421 ms
108 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=44.574 ms

--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 41.421/42.998/44.574/1.576 ms
```

# 네트워크 관련 명령어

- ping(Packet InterNet Groper)
  - ✓ 샘플: 리눅스에서 ping을 사용해 목적지 8.8.8.8로 100byte 크기의 패킷을 2회 보내기
    - 결과 해석
      - 108 bytes from 8.8.8.8: icmp_seq=0 ttl=113 time=41.421 ms
        - 108 bytes from 8.8.8.8: 8.8.8.8 서버로부터 108바이트 크기의 응답 패킷을 돌려받았다는 의미(요청 시 100바이트의 데이터를 보냈고, 여기에 ICMP 헤더 등이 붙어 108바이트로 돌아온 것)
        - icmp_seq=0 (또는 1): 보낸 패킷의 번호(순서)로 0번, 1번 패킷 모두 누락 없이 순서대로 잘 도착
        - ttl=113: TTL(Time To Live)은 패킷이 네트워크를 돌아다닐 수 있는 수명으로 라우터(경유지)를 하나 거칠 때마다 1씩 줄어드는데 113이라는 숫자는 상대방 서버를 거쳐 내 컴퓨터로 오는 동안 약 15~16개의 라우터를 통과했음을 짐작게 함(정상적인 연결 상태)
        - time=41.421 ms / 44.574 ms: 가장 중요한 왕복 시간(지연 시간)인데 내 컴퓨터에서 보낸 신호가 구글 서버에 닿았다가 다시 내 컴퓨터로 돌아오는 데 각각 약 0.041초, 0.044초가 걸렸다는 의미로 한국에서 해외 서버나 대형 CDN을 거칠 때 나오는 매우 정상적이고 양호한 속도

# 네트워크 관련 명령어

- ping(Packet InterNet Groper)
  - ✓ 샘플: 리눅스에서 ping을 사용해 목적지 8.8.8.8로 100byte 크기의 패킷을 2회 보내기
    - 결과 해석
      - 2 packets transmitted, 2 packets received, 0.0% packet loss
        - 2 packets transmitted, 2 received: 패킷을 2번 던져서 2번 모두 응답을 받음
        - 0.0% packet loss: 패킷 손실률이 0%인데 중간에 데이터가 유실되지 않고 네트워크 선로가 아주 깨끗하게 연결되어 있음을 의미하는데 만약 인터넷이 불안정하면 이 부분에 10% loss, 50% loss 같은 숫자가 뜨거나 Request timed out(요청 시간 초과)이 발생
      - round-trip min/avg/max/stddev = 41.421/42.998/44.574/1.576 ms
        - min (최소 시간): 가장 빨랐을 때 41.421 ms
        - avg (평균 시간): 전체 평균 42.998 ms
        - max (최대 시간): 가장 느렸을 때 44.574 ms
        - stddev (표준편차): 1.576 ms
          - 표준편차(stddev)는 값이 얼마나 들쭉날쭉한지를 보여주는 지표로 이 수치가 1.5 정도로 매우 낮다는 것은 네트워크 속도가 튀지 않고 일정하게 유지되고 있다는 뜻이므로, 온라인 게임이나 실시간 스트리밍을 하기에 아주 좋은 상태

# 네트워크 관련 명령어

- tcping - windows
  - ✓ 개요
    - Ping은 목적지 단말이 살아 있는지 확인하고 출발지와 목적지까지의 네트워크가 잘 연결되어 있는지 확인하는 명령인데 목적지 단말까지의 중간 경로에 문제가 발생한 경우 icmp error 메시지를 활용해 라우팅 문제가 발생한 경우 일부 파악이 가능하지만 목적지 단말이 잘 살아 있고 중간 경로에 문제가 없더라도 실제 서비스를 위해 사용되는 서비스 포트가 정상 상태인지 ping만으로는 확인할 수 없음
    - 출발지와 목적지 사이에 방화벽과 같은 보안 장비에서 막히지 않는지, 목적지 운영체제에서 운영체제 방화벽이 동작해 차단하는 것은 아닌지, 목적지에서 서비스하기 위해 정상적으로 서비스 포트가 오픈되지 않았는지 추가로 확인해야 하는 경우가 많고 네트워크 경로 체크에서도 Ping에서 사용하는 icmp 메시지가 내부 네트워크의 상태 정보를 외부에 유출할 수 있어 정책적으로 icmp와 traceroute를 차단하는 경우도 많음
    - naver.com은 ping이 막혀 있어 확인이 불가능

      PING naver.com (223.130.200.236): 56 data bytes

      Request timeout for icmp_seq 0

      Request timeout for icmp_seq 1

# 네트워크 관련 명령어

- tcping - windows
  - ✓ 개요
    - 목적지의 실제 서비스 포트로 정상적인 통신이 가능한지 확인하는 작업이 매우 중요
    - tcping 프로그램을 통해 ping을 확인하듯이 서비스 포트가 정상적으로 열려 있는지 확인할 수 있음
    - 운영체제에 포함된 기본 명령은 아니지만 설치해두면 유용하게 사용할 수 있음
    - tcping의 사용법

      tcping [옵션] 목적지_IP 주소
    - tcping에서 사용되는 주요 옵션
      - -n count: tcping을 전송하는 횟수(기본 5회)
      - -t: 중지될 때까지 지정한 호스트로 ping을 지속적으로 전송
      - -i interval: tcping을 전송하는 시간 간격
      - serverport: tcping으로 확인하려는 서비스 포트이며 미 설정 때는 80이 기본값

# 네트워크 관련 명령어

- tcping - windows
  - ✓ 다운로드: https://www.elifulkerson.com/projects/tcping.php
  - ✓ 사용
    - tcping –n 4 naver.com

      Probing 223.130.200.236:80/tcp – Port is open - time=5.612ms

      --- 223.130.200.236 ping statistics ---

      2 packets transmitted, 2 packets received, 0.0% packet loss

      round-trip min/avg/max/stddev = 41.421/42.998/44.574/1.576 ms

# 네트워크 관련 명령어

- traceroute(tracert)
  - ✓ 개요
    - traceroute는 출발지부터 통신하거나 목적지까지의 네트워크 경로를 확인할 때 사용하는 네트워크 명령어
    - Ping은 목적지 단말이 잘 동작하는지 확인하는 데 사용되고 icmp 메시지를 이용해 중간 경로에 문제가 있을 때 이것을 확인할 수 있었지만 traceroute는 중간 경로의 더 상세한 정보를 얻는 데 사용
    - 중간 네트워크 장비가 아닌 출발지 pc에서 목적지까지의 라우팅 경로를 확인할 수 있고 목적지까지의 통신에 문제가 있을 때 어느 구간부터 문제가 발생했는지 찾아낼 수 있으며 목적지까지의 네트워크 응답 시간이 느린 경우 어느 구간에서 응답 시간이 느려지는지도 알아낼 수 있음
    - traceroute는 IP 헤더의 TTL(Time To Live) 필드를 이용하는데 TTL을 1부터 1씩 증가시키면서 목적지에 도달할 때까지 패킷을 반복적으로 전송하면서 경로를 추적
    - 인터넷 구간에서 경로를 찾지 못하거나 잘못 전송된 패킷이 무제한적으로 계속 돌아다니지 않도록 IP 헤더에 패킷이 살아 있을 수 있는 한계를 명시하는데 이 정보가 TTL
    - 인터넷 구간에서 라우터 장비를 하나 지날 때마다 패킷의 TTL 값이 1씩 줄고 TTL이 0이 되는 순간 해당 라우터에서 이 패킷을 드롭시키고 icmp 메시지를 이용해 출발지 단말에 패킷을 드롭한 이유를 알려줌

# 네트워크 관련 명령어

- traceroute(tracert)
  - ✓ 개요
    - TTL이 1일 때는 1홉까지의 장비로 전달되고 TTL이 0으로 만료되면서 해당 장비는 ICMP time exceed 메시지를 출발지로 전달
    - traceroute는 이 메시지를 전달한 장비의 IP를 출력하는 과정을 반복하면서 경로를 추적
    - traceroute 경로 추적은 IP 헤더인 3계층 정보에 의한 경로 추적이므로 3계층 이하의 스위치 장비 추적은 불가능
    - 라우팅이 동작하는 라우터나 L3 스위치와 같은 계층 장비의 경로만 확인할 수 있음
    - traceroute로 두 호스트 간 경로가 의도대로 정상적인 경로인지 확인하려면 각 호스트에서 traceroute를 모두 수행해야 함

# 네트워크 관련 명령어

- traceroute(tracert)
  - ✓ 옵션
    - 리눅스: traceroute 패키지 설치 후 사용
      - -I (대문자 i): ICMP 사용 옵션으로 윈도우 tracert와 똑같이 ICMP 패킷으로 경로를 추적하는데  방화벽 통과 확률이 높음
      - -T:TCP 사용 옵션으로 기본 포트는 80번이며 웹 서버나 특정 TCP 서비스 경로만 추적할 때 유용
      - -p <포트>: 포트 번호 지정 옵션으로 -T 또는 -U 옵션과 함께 쓰며 특정 포트(예: DB 3306 등)를 타겟팅할 때 사용
      - -n: 역방향 DNS 조회를 하지 않는 옵션으로  IP 주소를 도메인 이름으로 바꾸는 과정을 생략하므로 결과가 수 배 이상 빠르게 출력
      - -m <숫자>: 최대 홉(Hop) 수 지정으로 패킷이 거쳐 갈 최대 라우터 개수를 제한(기본값은 보통 30홉)
      - -q <숫자>: 프로브(Prb) 개수 지정 옵션으로 라우터당 보낼 패킷 개수 (기본값 3개이므로 속도를 높이려면 1개로 조정)
      - -w <초>: 타임아웃 시간 설정 옵션으로 응답을 기다리는 제한 시간(초)을 설정
      - -s src_addr: 패킷이 나가는 인터페이스가 아닌 별도 IP로 출발지 IP를 지정

# 네트워크 관련 명령어

- traceroute(tracert)
  - ✓ 옵션
    - Windows
      - -d:  역방향 DNS 조회 안하는 옵션으로 리눅스의 -n과 같은 역할
      - -h <숫자>: 최대 홉(Maximum Hops) 지정 옵션으로 목적지까지 거쳐 갈 최대 라우터 개수를 설정(기본값은 30개)
      - -w <밀리초>: 타임아웃 시간 설정 옵션으로응답을 기다리는 제한 시간을 밀리초(ms) 단위로 설정(예: -w 1000은 1초)
      - -4:IPv4 강제 사용 옵션으로 네트워크 환경이 IPv6를 지원하더라도 명시적으로 IPv4 경로만 추적
      - -6: IPv6 강제 사용 옵션으로 명시적으로 IPv6 경로만 추적
      - 리눅스는 UDP, ICMP(-I), TCP(-T) 등 패킷 종류를 마음대로 바꿀 수 있지만 윈도우의 tracert는 오직 ICMP (Echo Request) 패킷 하나만 고정으로 사용해야 하므로 특정 포트(예: 웹 80번, DB 3306번)를 타겟팅하여 경로를 추적하는 기능은 윈도우 기본 tracert 명령어로는 불가능

# 네트워크 관련 명령어

- traceroute(tracert)
  - ✓ 샘플
    - ● ubuntu
      - ◆ sudo traceroute -n -I google.com

```
traceroute to google.com (142.250.198.174), 30 hops max, 60 byte packets
 1  192.168.0.1  0.979 ms * *
 2  * * *
 3  * * *
 4  10.243.99.25  7.048 ms  6.075 ms  5.012 ms
 5  100.78.30.1  6.733 ms  6.207 ms  6.080 ms
 6  * * *
 7  1.213.112.53  5.928 ms  6.352 ms  5.671 ms
....
19  216.239.47.17  63.199 ms  62.738 ms  61.175 ms
20  142.250.198.174  62.821 ms  62.047 ms  152.686 ms
```

      - ◆ sudo traceroute -T -p 443 naver.com

```
traceroute to naver.com (223.130.192.247), 30 hops max, 60 byte packets
 1  223.130.192.247 (223.130.192.247)  8.134 ms  9.156 ms *
```

# 네트워크 관련 명령어

- traceroute(tracert)
  - ✓ 샘플
    - Windows
      - tracert -d google.com
      - tracert -d -w 1000 google.com
      - tracert -h 50 google.com

# 네트워크 관련 명령어

- tcptraceroute
  - ✓ 개요
    - traceroute는 경로 추적만 가능하며 서비스를 위한 서비스 포트가 정상적으로 열리는지 확인할 수 없음
    - 서비스에 문제가 생겼을 때는 중간 경로에서 차단되었는지, 최종 목적지에서 차단되었는지. 목적지 단말에서 서비스를 제대로 오픈하지 못했는지 확인해야 하는데 이런 여러 가지 사항을 한꺼번에 확인할 수 있는 명령어가 tcptraceroute(tcptrace)
    - tcptraceroute는 traceroute와 유사하게 출발지와 목적지까지의 경로를 확인하지만 실제 서비스 포트를 이용해 경로를 추적하므로 최종 목적지까지 서비스 포트가 정상적으로 열리는지 확인할 수 있고 만약 열리지 않는다면 어느 구간부터 서비스가 막히는지 확인할 수 있음
    - Windows에서는 https://www.elifiilkerson.com/projects/tcproute.php 에서 다운로드 받아서 설치

# 네트워크 관련 명령어

- ❖ tcptraceroute
  - ✓ 실습
    - ● tcptraceroute -n naver.com 80

      Selected device enp0s8, address 192.168.0.100, port 43591 for outgoing packets

      Tracing the path to naver.com (223.130.192.247) on TCP port 80 (http), 30 hops max

      1 223.130.192.247 [open] 122.232 ms 6.448 ms 5.015 ms

# 네트워크 관련 명령어

- ❖ netstat(network statistics)
  - ✓ 개요
    - netstat은 서버의 다양한 네트워크 상태를 확인하는 데 사용하는 명령어
    - 여러 가지 네트워크 관련 정보를 확인할 수 있어 사용 범위가 매우 넓지만 서비스 포트 상태를 확인하는 용도로 가장 많이 사용
    - 현재 서버에서 특정 서비스가 정상적으로 열려 있는지(listening) 또는 외부 서비스와 TCP 세션이 정상적으로 맺어져 있는지(ESTABLISHED), 서비스가 정상적으로 종료되고 있는지(TIME_WAIT, FIN_WAIT, CLOSE_WAIT)여부 등을 netstat 명령어로 확인
    - 이런 정보는 서버의 네트워크 상태를 확인시켜 주므로 서비스에 문제가 발생했을 때 문제 해결을 위한 기본 정보로 활용할 수 있지만 이런 정보를 이용해 현재 서비스 상태를 명확히 이해하고 문제를 분석하려면 TCP 통신에 대한 상태 변화와 동작 방식을 정확히 이해하고 있어야 함
    - netstat 명령어는 서비스의 포트 상태 정보뿐만 아니라 라우팅 테이블이나 인터페이스 패킷 통계정보 확인에도 사용할 수 있음

# 네트워크 관련 명령어

- netstat(network statistics)
  - ✓ 옵션

| 기능 | 윈도우 | 리눅스 | 설명 |
|---|---|---|---|
| 전체 연결 표시 | -a | -a | 연결된(Established) 포트뿐만 아니라 대기 중인(Listening) 포트까지 모두 출력 |
| 숫자로 표시 | -n | -n | 주소나 포트 번호를 이름(예: http, https) 대신 **숫자(예: 80, 443)**로 표시하여 속도가 빨라짐 |
| 프로그램/PID 표시 | -o | -p | 어떤 프로그램(프로세스 ID)이 해당 포트를 쓰고 있는지 찾아줌 |
| 프로토콜 지정 | -p <tcp/udp> | -t (TCP) / -u (UDP) | 특정 프로토콜만 골라서 필터링 |
| 라우팅 테이블 | -r | -r | 현재 컴퓨터의 라우팅 경로 정보를 출력 |
| 통계 보기 | -s | -s | IP, TCP, UDP, ICMP 패킷의 총 전송/수신 통계를 보여줌 |

# 네트워크 관련 명령어

- ❖ netstat(network statistics)
  - ✓ 사용법
    - 윈도우에서 어떤 포트가 열려 있고, 그 포트를 어떤 프로그램(PID)이 쓰고 있는지 한눈에 숫자로 파악: netstat -ano
    - 리눅스 서버에서 현재 정상적으로 구동 중인 서비스(Listening 상태)와 프로세스 명을 확인: sudo netstat -ntlp
    - 사용 중인 프로세스 종료
      - ◆ 윈도우

      # 1. 8080 포트를 쓰는 PID(프로세스 아이디) 찾기

      netstat -ano | findstr 8080

      # 2. 결과 끝에 나온 PID(예: 1234)를 이용해 강제 종료하기

      taskkill /f /pid 1234

      - ◆ 리눅스

      Bash

      # 1. 8080 포트를 쓰는 프로그램 찾기

      sudo netstat -ntlp | grep 8080

      # 2. 결과에 나온 PID(예: 5678)를 이용해 프로세스 죽이기

      sudo kill -9 5678

# 네트워크 관련 명령어

- ❖ netstat(network statistics)
  - ✓ 사용법
    - ● 네트워크 연결 상태(State) 읽는 법
      - ◆ LISTENING: 서버 프로그램이 손님(클라이언트)의 접속을 받기 위해 문을 열어놓고 기다리는 상태(내 컴퓨터가 서버 역할을 하고 있음)
      - ◆ ESTABLISHED: 서로 연결이 완벽하게 수립되어 데이터를 정상적으로 주고받는 상태
      - ◆ TIME_WAIT: 데이터 송수신이 끝나고 연결을 종료하는 과정 중 하나로 혹시 모를 유실된 패킷이 뒤늦게 올까 봐 잠시 포트를 열어두고 대기하는 정상적인 상태

# 네트워크 관련 명령어

- nslookup(name server lookup)
  - ✓ 개요
    - nslookup은 DNS에 다양한 도메인 관련 내용을 질의해 결괏값을 전송받을 수 있는 네트워크 명령
    - nslookup 명령어를 사용하면 운영체제에 설정했던 네트워크 설정 정보를 이용해 DNS 서버 주소로 질의를 보내지만 필요한 경우 옵션값으로 질의하려는 DNS 서버를 변경할 수 있음
    - DNS 서버 변경은 보통 특정 도메인의 변경된 설정값이 외부 DNS로 잘 전파되었는지 확인할 때 사용

# 네트워크 관련 명령어

- nslookup(name server lookup)
  - ✓ 사용법
    - $ nslookup

    > google.com

    Server: 127.0.0.53

    Address: 127.0.0.53#53

    Non-authoritative answer:

    Name: google.com

    Address: 142.250.198.206

    Name: google.com

    Address: 2404:6800:4005:80a::200e

# 네트워크 관련 명령어

- Ipconfig/ifconfig
  - ✓ 사용법
    - 리눅스
      - ifconfig(기본): 현재 활성화된 모든 네트워크 인터페이스(eth0, wlan0, en0 등)의 IP 주소, MAC 주소(ether), 서브넷 마스크(netmask), 송수신 패킷 수(RX/TX)를 조회
      - ifconfig –a: 현재 비활성화(Down)되어 있는 네트워크 카드까지 포함하여 시스템에 존재하는 모든 인터페이스를 강제로 찾아내 보여줌
      - 리눅스 인터페이스 제어
        - sudo ifconfig eth0 down: eth0 카드 비활성화
        - sudo ifconfig eth0 up: eth0 카드 활성화
        - 수동으로 고정 IP 주소 변경: sudo ifconfig eth0 192.168.0.50 netmask 255.255.255.0

# 네트워크 관련 명령어

- ❖ Ipconfig/ifconfig
  - ✓ 개요
    - ● 가장 기본적으로 사용하는 네트워크 인터페이스(네트워크 카드, IP 주소 등) 확인 및 관리 명령어
  - ✓ 사용법
    - ● 윈도우
      - ◆ ipconfig(기본): 현재 컴퓨터에 연결된 모든 네트워크 어댑터(이더넷, 와이파이 등)의 IPv4 주소, 서브넷 마스크, 기본 게이트웨이(공유기 주소)를 아주 간결하게 보여줌
      - ◆ ipconfig /all (상세 보기): 가장 많이 쓰는 옵션으로 기본 정보에 더해 MAC 주소(물리적 주소), DNS 서버 주소, DHCP 활성화 여부 등 컴퓨터의 모든 네트워크 세부 스펙을 낱낱이 출력
      - ◆ 인터넷 연결이 안좋을 때
        - ▪ ipconfig /release: 현재 공유기로부터 할당받은 IP 주소를 반납
        - ▪ ipconfig /renew: 공유기에게 IP 주소를 새로 달라고 요청
        - ▪ ipconfig /flushdns: 컴퓨터에 쌓인 이전 웹사이트의 DNS 캐시(주소록 수첩)를 깨끗이 지움 (특정 사이트만 접속이 안 될 때 유용)

# 네트워크 관련 명령어

- tcpdump
  - ✓ 개요
    - tcpdump는 네트워크 인터페이스로 오가는 패킷을 캡처해 보는 기능의 명령어
    - 평소에는 사용하지 않지만 장애 처리나 패킷 분석이 필요할 때 자주 사용하는 명령어
    - tcpdump는 주로 리눅스 서버에서 사용하지만 리눅스 커널 기반의 네트워크 장비에서도 많이 사용
    - ㅅcpdump 명령어를 사용하면 네트워크 인터페이스로 오가는 모든 패킷을 캡처할 수 있음
    - 전체 패킷을 캡처하면 분석이 어려우므로 보통 옵션을 이용해 분석에 필요한 패킷만 필터링해 캡처

# 네트워크 관련 명령어

- tcpdump
  - ✓ 옵션
    - -i [인터페이스]: 캡처할 네트워크 카드 지정 (예: eth0, any는 모든 인터페이스)
    - -n: 주소 변환을 하지 않음(IP 주소와 포트 번호를 도메인/서비스명 대신 숫자로 그대로 표시 -> 속도가 빨라짐 예: Localhost -> 127.0.0.1)
    - -nn: 서비스 포트를 실제 포트 번호로 표기(예: http -> 80)
    - -X: 패킷의 내용을 헥사(Hex)와 ASCII 문자 형태로 화면에 출력 (데이터 확인용)
    - -c [개수]: 지정한 개수의 패킷만 캡처하고 종료
    - -w [파일명]: 캡처한 패킷을 화면에 뿌리지 않고 .pcap 파일로 저장 (Wireshark에서 분석 가능)
    - -r [파일명]: 저장된 .pcap 파일을 읽어서 화면에 출력
    - src IP주소: 출발지 IP 주소를 지정해 필터링
    - dst IP주소: 목적지 IP 주소를 지정해 필터링
    - host IP주소: 출발지/목적지 상관없이 IP 주소를 지정해 필터링
    - src port 포트번호: 출발지 포트 번호를 지정해 필터링
    - dst port 포트번호: 목적지 포트 번호를 지정해 필터링
    - port 포트번호: 출발지/목적지 상관없이 tcp 포트를 지정해 필터링
    - tcp 또는 udp: tcp 또는 udp 만 필터링

# 네트워크 관련 명령어

- tcpdump
  - ✓ 사용법
    - 웹 서비스에 대한 패킷 캡쳐: sudo tcpdump -i enp0s8 tcp port 80
    - 특정 웹 사이트에 대한 패킷 캡쳐: tcpdump -i eth0 tcp port 80 and host 172.16.10.10
    - ssh를 제외한 전체 패킷 캡쳐: tcpdump -i eth0 not tcp port 22
    - tcpdump 결과를 dumpfile.pacp 파일로 출력: tcpdump -i eth0 -w dumpfiile.pcap
  - ✓ 와이어샤크
    - tcpdump와 같은 역할을 하는 애플리케이션으로 Wireshark가 있음
    - 와이어샤크를 사용하면 패킷 캡처를 더 쉽게 할 수 있고 캡처된 패킷이 필드별로 구분되어 표시됨
    - 패킷별로 Flow도 볼 수 있어 분석할 때 유용
    - tcpdump로 캡처한 패킷을 파일로 출력한 후 와이어샤크로 불러와 볼 수도 있음
    - https://www.wireshark.org/