

Ansible 자동화로 구현한 클라우드 환경 DevOps

- 그룹 보고서 -

제출일	2021. 01. 29
훈련기관	(주)솔데스크
강사님	김기연 강사님
조원	김민영, 김선경, 최시율, 허윤하, 황유지

목 차

1. 개요
 - 1.1. 개요
 - 1.2. 소프트웨어 환경
 - 1.3. 보고서 포맷
 - 1.4. 출처
2. DHCP 서버
 - 2.1. 개요
 - 2.2. DHCP 서버의 개념
 - 2.3. DHCP 동작 원리
 - 2.4. DHCP 서버 구성도
3. 캐시 전용 DNS
 - 3.1. 개요
 - 3.2. 캐시 전용 DNS 정의
 - 3.3. Unbound
 - 3.4. Unbound DNS의 이점
 - 3.5. 인프라 구성도
4. 웹 서버 구축
 - 4.1. 웹 서버 정의
 - 4.2. 기업 내부에서 웹 서버 구축
 - 4.3. 기업 외부에서 웹 서버 구축
 - 4.4. 결론
5. Stratis
 - 5.1. 개요
 - 5.2. stratis 정의
 - 5.3. stratis의 구성요소
 - 5.4. stratis의 장점
 - 5.5. stratis와 일반 filesystem의 차이점
6. iSCSI
 - 6.1. iSCSI
 - 6.2. LVM 및 target서버 구성
 - 6.3. 추가 구성
7. 마무리
 - 7.1. 결론
 - 7.2. 최종 구성도

개요

개요

기업의 자원을 저비용, 고효율로 활용하기 위한 제안서이다. 외부 IDC 업체에 서버 호스팅을 하여 관련된 비용을 절감하며 DHCP 서버, DNS 캐시 서버, stratis와 iscsi 서버 구성을 통해 스토리지 관리를 더 편리하고 쉽게 할 수 있다. 이뿐만 아니라 다양한 서버를 사용할 때 발생하는 트래픽 과부하 등을 방지할 수 있다.

소프트웨어 환경 및 버전

소프트웨어	설명
LINUX	CentOS 8.3
VM	VMware Workstation 15.5.6

보고서 포맷

전체 보고서 : 맑은 고딕 10pt, 줄 간격 150%
텔 작성 : 굴림체 8pt, 표 안에 작성 여백 5

출처

[레드햇 공식 홈페이지](#)

[compozi](#)

[men&mice](#)

[NLnet Labs](#)

강의 Notion

[가비아](#)

D H C P

개요

인터넷의 TCP/IP 프로토콜에서는, 각 컴퓨터들이 고유한 IP 주소를 가져야만 인터넷에 접속할 수 있다. 이 제안서는 DHCP 프로토콜이 IP 주소를 중앙에서 어떻게 관리하고 할당하는지에 대한 내용을 담고 있으며, DHCP 서버의 구성을 포함한다.

DHCP란?

동적 호스트 구성 프로토콜(Dynamic Host Configuration Protocol)을 말하며, 호스트 IP 구성 관리를 단순화하는 IP 표준이다. 동적 호스트 구성 프로토콜 표준에서는 DHCP 서버를 사용해서 IP 주소 및 관련된 기타 구성 세부 정보를 네트워크의 DHCP 사용을 하는 클라이언트에게 동적으로 할당하는 방법을 제공한다.

DHCP를 사용하는 이유

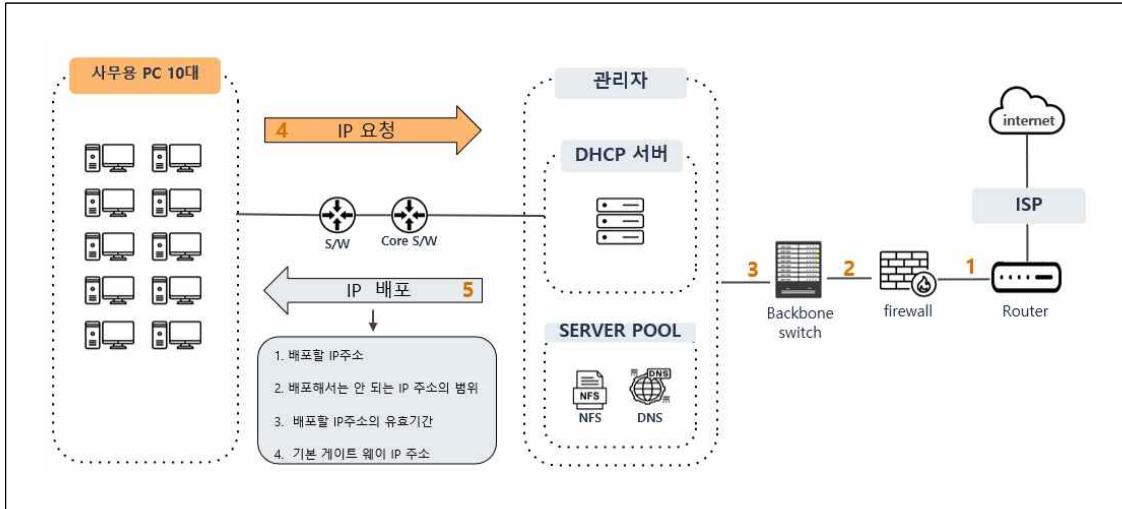
DHCP 서버에서 설정되어 있는 IP를 자동으로 사용자에게 할당해 주기 때문에 각 PC들은 자동으로 IP를 받아오기만 하면 되기 때문에 편하게 IP를 관리할 수 있다.

DHCP 동작 원리

▼ DHCP 동작 원리는 크게 네 단계로 단말과 서버 간에 통신이 이루어진다.

- 1) DHCP Discover : DHCP를 찾기 위해 DHCP 서버를 찾는 메시지를 브로드캐스트로 전송한다.
- 2) DHCP Offer : DHCP 서버가 아닌 컴퓨터들은 클라이언트에 보낸 DHCP Discover를 무시하거나
- 3) DHCP Request: 클라이언트는 IP 주소, 서비스 정보를 이용하여 네트워크 설정을 하고 DHCP 서버의 사용을 위한 DHCP Request를 보낸다.
- 4) DHCP Ack: 사용허가. 클라이언트는 DHCP 서버에서 보내준 정보(IP, DNS, GW, 임대정보)를 설정한다.

▼ DHCP 서버 구성도



▼ DHCP 서버 구성도 설명

1. 사내에는 사무용 PC 10대가 있다. DNS와 NFS가 구성되어 있고, PC들은 IP 요청을 한 상태이다.
2. 라우터 방화벽을 통해 패킷을 필터링하여 발신지 IP /protocol/port/.패킷 등을 선별적으로 인증해 접속 요청에 대해 판단한다.
3. 백본 스위치는 네트워크의 중심에 위치하며 모든 패킷이 지나간다. 방화벽 및 각종 서버가 접속되는 핵심 영역이기 때문에 백본 스위치를 도입하여 병목구간의 대규모 트래픽을 처리할 수 있도록 구성한다.
4. 사내에는 사무용 PC 10대가 있다. PC들은 관리자에게 IP요청을 스위치를 거쳐 전달한다. 관리자는 10대의 PC에게 IP 할당을 위해서는 DHCP를 구축해야 한다. 이미 내부에는 NFS와 DNS가 이미 구성되어 있어 하나의 서버풀로 묶어두고 DHCP 서버를 추가하면 된다.
5. IP 할당을 서비스하는 DHCP 서버는 라우터에서 동작할 수 있다. 관리자는 배포할 IP 주소의 범위나 각종 설정 정보를 등록해준다. DHCP 서버는 배포하도록 설정한 IP 주소의 범위 안에서 아직 미사용 중인 IP주소를 스위치와 코어스위치를 거쳐 배포한다.

캐시 전용 DNS

개요

여러 DNS 서버 종류 중 캐시 전용 DNS 서버의 정의, 기능, 특징을 자세히 알아보고 그중에서도 Unbound에 대해 자세히 알아봅시다. Unbound를 추가한다고 가정할 때 이를 구축하였을 때의 장점과 변경되는 인프라 구성도는 어떻게 되는지 알아봅시다.

캐시 전용 DNS

도메인에 대한 데이터를 관리하지는 않고 요청에 대한 처리만 해주는 DNS Server이다. 보통 본사와 지사가 있는 경우에 지사에서 사용하며, 한 번 사용한 주소지에 대해서는 Client 요청에 대한 응답을 캐시 서버에 남겨두어 재요청하지 않아도 되도록 구성되어 있다.

Unbound

비영리 공익 재단인 NLnet Labs에 의해 개발되었다. 검증된 재귀적인 캐시 DNS 서버 소프트웨어이다. BSD 라이선스로 오픈 소스 형태로 무료로 배포된다.

Unbound DNS의 장점

1. 가볍다

Java 프로토타입을 기반으로 C로 개발되었다. 이 개발자들은 Unbound의 소스코드를 매우 모듈적으로 아주 가볍게 설계했다. 개발자들은 validator, resolver 및 캐시 서버로서 최소한의 요구 사항을 달성할 수 있는 가능한 최소의 솔루션을 설계하기를, 이러한 요구를 충족하는 것 외에도, 서버가 아주 높은 성능을 갖길 원했다.

2. 구성이 쉽다

Unbound는 구성이 아주 쉽다. YAML(Yet Another Markup Language)과 같은 구성 파일을 통해 구성된다. Unbound를 설정하는 데 필요한 구성 지침은 많지 않다. 서비스가 비교적 단순하고 단일 역할을 가지기 때문이다.

3. 높은 성능

Unbound의 가벼운 코드 구조와, 단순하고 모듈적인 설계는 Unbound를 재귀 네임 서버에서 극도로 높은 성능을 이끌어낸다. 초기 벤치마크 테스트 결과, Unbound가 다른 네임 서버에 비해 최대 2배 향상된 성능을 제공했다. Unbound에는 기본적으로 두 가지 작동 모드가 있다.
Threaded mode – 확장성을 높이기 위해 Libevent 교차 컴파일 래퍼 라이브러리 사용한다.
Forked mode – Unbound에서 스레드 되지 않은 프로세스와 Fork를 구분하여 작동할 수 있다.

4. DNSSEC 유효성 검사 지원

Unbound는 DNS 데이터를 보호하기 위한 메커니즘인 DNSSEC 유효성 검사를 처음부터 수행하도록 설계되었다. DNSSEC 검증은 다른 DNS 서버와 마찬가지로 플러그인 또는 볼트온으로 구현되지 않는다. 이는 검증 코드가 Unbound에서 최적화되었기 때문에 Unbound를 다른 솔루션보다 더 높은 성능의 솔루션으로 만든다. trust anchor 관리를 위한 추가 기능이 작업 중에 있으며, 이는 이미 우수한 제품을 향상시키는데 도움이 될 것이다.

5. 소프트웨어 다양성 추가

엔터프라이즈 고객 및 ISP는 이제 Unbound를 통해 DNS 서버의 검증되고 재귀적이며 캐시 전용 계층을 제공하고 입증되고 신뢰할 수 있는 BIND 대안을 도입할 수 있다. Unbound는 대중에게 소프트웨어 다양성을 소개한다. BIND DNS는 오랜 기간 독점이었다. 소프트웨어 다양성은 인터넷과 ISP와 엔터프라이즈에도 좋다. 소프트웨어와 코드 다양성으로 인해 우리는 동일하거나 더 나은 서비스를 제공하기 위해 서로 다른 DNS 공급업체 솔루션을 혼합할 수 있다. 어떠한 공급업체 제품의 버그는 다른 공급업체에서 볼 수 없을 것이다.

6. 프로덕션 준비

surfnet.nl은 2009년 9월에 모든 SOUPnet DNS resolver가 DNSSEC를 사용할 수 있다고 발표했다. 이들의 DNSSEC 유효성 검증은 Unbound DNS 서버 패키지에 의존했다. 주요 통신사가 고객을 위해 Unbound 서비스를 제공하기 시작한다면, 기업에도 준비가 되어 있다는 것은 당연하다.

7. 단일 목적

Unbound가 검증, 재귀 및 캐싱 확인 프로그램으로 코딩되었기 때문에 DNS 서버 솔루션이 수행하는 분할 또는 이중 개인 정보로 인해 어려움을 겪지 않는다. Unbound는 대부분 단일 목적 서버이다. Unbound는 데이터에 대한 권한이 없기 때문에 코드와 기능이 단순화된다. 동적 DNS 업데이트 또는 구역 전송 등을 지원하는 코드가 없다. 대신, 이 단일 목적 서버는 재귀, 유효성 검사 및 캐싱 해상도 등을 지원하도록 코딩된 것 중에서 동급 최강의 서버이다.

8. 보안

Unbound는 단순성과 성능을 희생시키면서 DNS 보안에 소홀히 하지 않았다. 반대로 Unbound는 견고한 접착제, 액세스 제어, 쿼리 ID 및 포트의 최대 임의성, 응답 스크러빙, 사례 보존, 서비스 거부 또는 DoS 보호 기능을 갖춘 DNS Security 기능을 갖추고 있다. 이러한 기능은 Unbound를 가장 안전한 DNS 서버 구현 중 하나로 만드는 일부 기능에 불과하다.

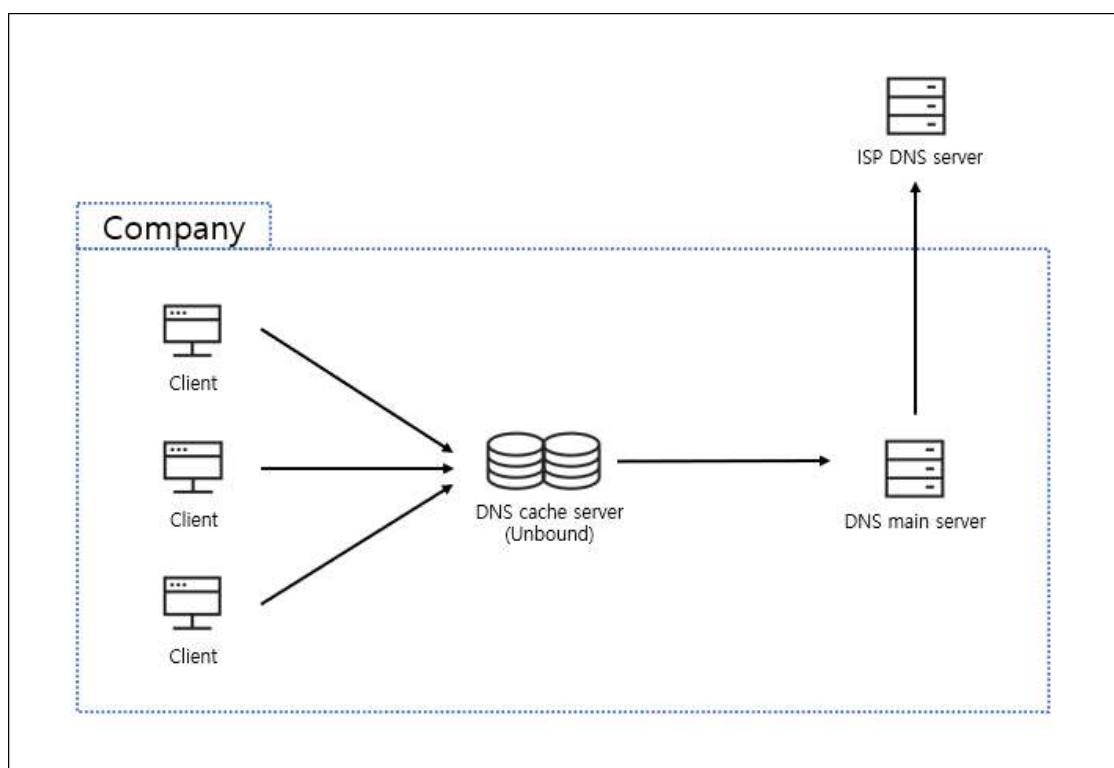
9. 관리성

Unbound에는 원격 관리 기능과 광범위한 네트워크 모니터링 통계 세트를 제공하는 확장 관리 명령 줄 인터페이스 또는 CLI가 있다. Unbound-control은 Secure Sockets Layer 또는 SSL을 사용하여 Unbound를 실행하는 서버에 대한 보안 연결을 사용한다. 명령은 클라이언트에서 전송되고 서버의 응답은 출력으로 표시된다. Unbound-Control-setup이라는 추가 CLI가 제공되어 Unbound-Control 작동을 위한 OpenSSL 공유기 및 구성 지침을 지원한다. 통계 출력은 Unbound가 추적하는 다양한 기준 및 확장된 통계를 그래프로 작성하기 위해 Munin 또는 Campidi와 같은 알려진 용량 계획 도구를 "공급"하는 데 사용할 수 있다.

10. 휴대용 솔루션

Unbound는 Linux, BSD, Solaris SPARC 및 X86, MacOS/X, Windows를 포함한 광범위한 하드웨어 OS 플랫폼에서 실행되도록 포트 되었다. Windows 32비트 사전 컴파일 된 이진 패키지는 NLnet Labs에서 직접 사용하거나 소스 패키지를 다운로드하여 직접 컴파일 할 수 있다.

캐시 전용 DNS 사용 시 인프라 구성도



1. Client에서 DNS 요청
2. DNS cache server의 cache table 확인
 - 2.1. DNS cache server의 cache table에 존재한다면 요청에 대한 응답
 - 2.2. DNS cache server의 cache table에 존재하지 않는다면 DNS main server로 요청
 - 2.2.1. DNS main server에서 ISP DNS server로 요청
 - 2.2.2. ISP DNS server의 cache table을 확인한 뒤 존재하지 않는다면 Root DNS server부터 탐색 후 요청에 대한 응답

S T R A T I S

개요

stratis의 정의, 기능, 특징을 자세히 알아보고 기존에 수동으로 하나씩 생성해야 하고 그에 따른 디스크 용량 관리가 필요한 일반적인 filesystem과 비교해 보면서 stratis를 사용함으로써 가져올 수 있는 이점에 대해 알아봅시다.

stratis 란 무엇인가?

Stratis는 씬 프로비저닝, 스냅샷, 풀 기반 관리 및 모니터링과 같은 고급 스토리지 기능을 쉽게 사용할 수 있도록 지원하는 Linux 로컬 스토리지 관리 툴이다.

Stratis는 물리적 스토리지 장치 풀을 관리하는 서비스로 실행되며, 생성되는 파일 시스템의 볼륨을 투명하게 생성 및 관리한다. Stratis는 기존 스토리지 드라이버 및 도구를 사용하기 때문에 현재 LVM, XFS, 장치 매퍼에서 사용하는 모든 고급 스토리지 기능이 Stratis에서 지원된다. 또한 SELinux map 권한이 추가되어 파일, 디렉터리, 소켓 등에 대한 메모리 매팅 액세스를 제어할 수 있게 되었다

stratis의 장점

stratis는 리눅스를 위한 로컬 스토리지 관리 솔루션이다. 단순성과 사용 편의점에 중점을 두고 있어서 다음과 같은 활동을 용이하게 해준다.

1. 스토리지의 초기 구성
2. 필요시 파일시스템의 크기 자동 확장
3. 고급 스토리지 기능 사용 가능

: Stratis는 기존 스토리지 드라이버 및 도구를 사용하기 때문에 현재 LVM, XFS, 장치 매퍼에서 사용하는 모든 고급 스토리지 기능이 Stratis에서도 지원된다.

stratis 볼륨의 구성 요소

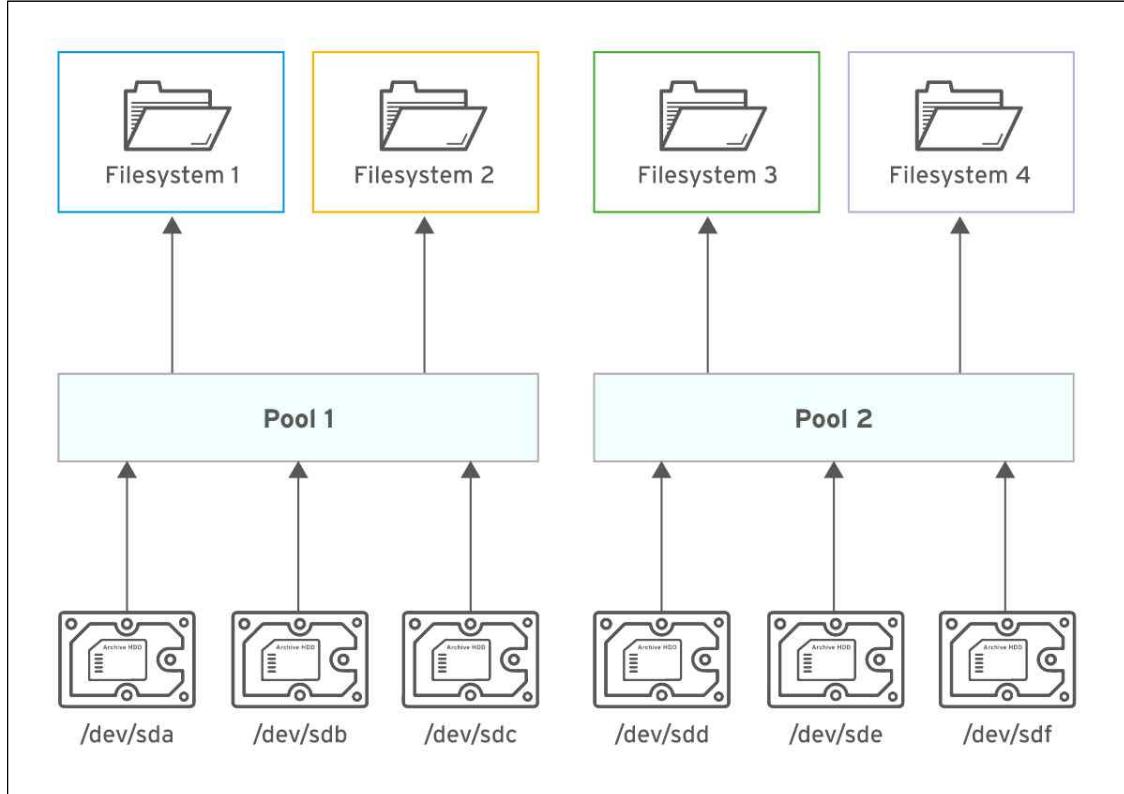
1. 디스크

stratis는 다양한 블록 장치 집합을 사용하여 여러 개의 pool 을 만들 수 있다. 예를 들어, ssd, scsi 등 디스크 장치를 멀티로 사용 가능하다. 각 pool에서는 하나 이상의 파일 시스템을 만들 수 있다.

2. pool

pool은 하나 이상의 블록 장치로 구성된다. 또한 블록 장치의 크기와 동일한 고정된 총 크기를 갖는다.

3.filesystem



파일 시스템은 씬 프로비저닝¹⁾되며 고정된 총 크기가 없다. 파일 시스템의 실제 크기는 저장된 데이터와 함께 증가한다. 데이터 크기가 파일 시스템의 가상 크기에 가까워지면 stratis는 씬 볼륨과 파일 시스템을 자동으로 증가시킨다. 씬 프로비저닝된 파일 시스템을 관리하기 위해서는 stratis-cli 및 stratisd 패키지를 설치하여야 한다. 또한 파일시스템의 기본값은 xfs 파일이다.

※ 주의

Stratis는 저장된 메타데이터를 사용하여 관리되는 풀, 볼륨 및 파일 시스템을 인식한다. 따라서 Stratis가 만든 파일 시스템은 수동으로 다시 포맷하거나 재구성해서는 안 된다. 만일 재구성 볼륨 디스크의 데이터가 모두 제거될 수 있다. Stratis 도구 및 명령으로만 관리해야 한다. Stratis 파일 시스템을 수동으로 구성하면 해당 메타데이터가 손실되어 Stratis가 생성한 파일 시스템을 인식하지 못할 수 있다.

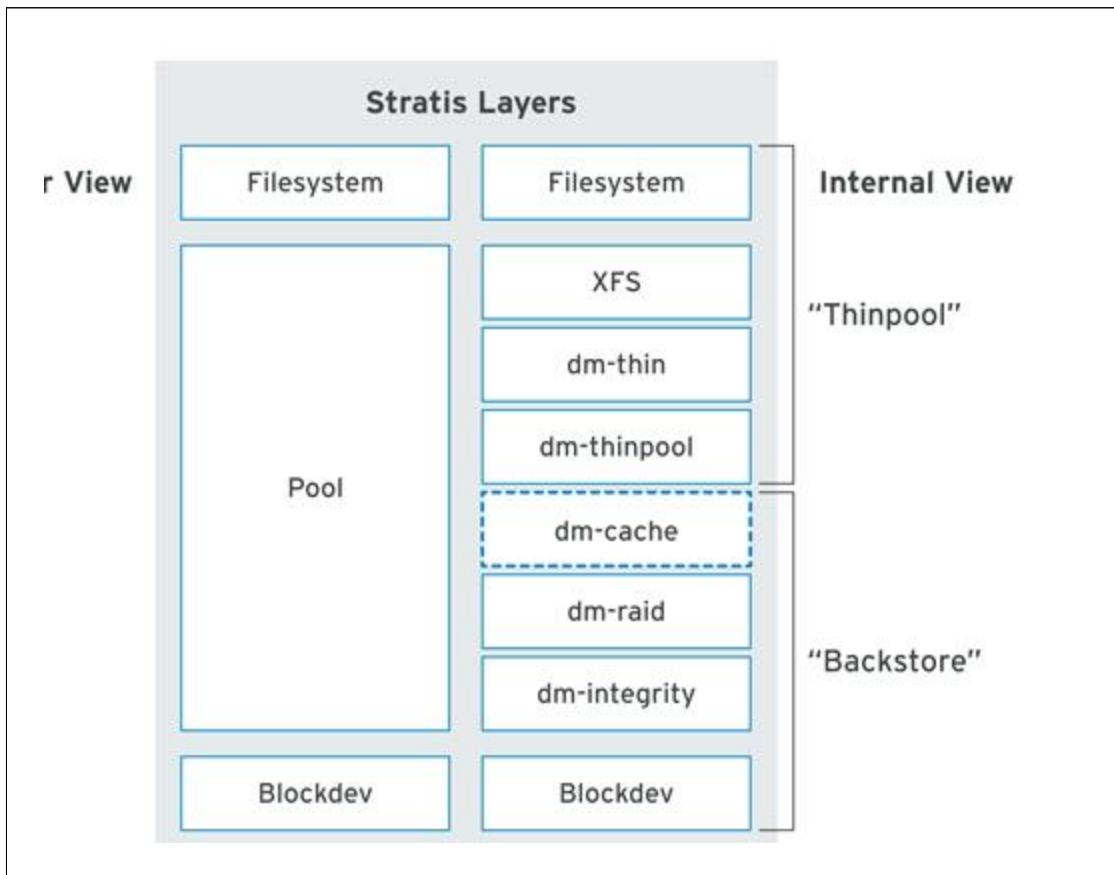
1) Thin provisioning : 모든 용량을 사용하지 않고 사용되는 만큼 조금씩 디스크의 크기를 점유하는 방식.

일반 filesystem과 stratis filesystem 방식 차이점

기존의 스토리지를 관리할 때는

디스크 추가 > 파티션 추가 > 해당 위치에 장치를 연결하기 위한 수동으로 마운트 > 파일 시스템 설정 > (옵션) 영구 마운트

용량이 부족할 경우 마운트를 해제하여 파티션을 늘리거나 디스크를 추가하는 작업을 해 주어야 한다.



stratis 파일 시스템은 크기가 고정되어 있지 않으며 사용되지 않은 블록 공간을 더 이상 할당하지 않는다. 따라서 디스크의 용량에 대해서 따로 설정해 줄 필요가 없다. 이 자동으로 늘려주는 역할 하는 것이 pool이라는 시스템인데 add 명령어를 통해 새롭게 디스크를 추가하여 용량을 늘릴 수 있고 다른 디렉토리에 할당하기 위한 다른 filesystem을 생성할 수도 있다. 또한 성능 향상을 위해서 SSD와 같이 초당 입/출력이 높은 블록 장치를 사용하여 더 빠른 속도로 서비스를 이용하실 수 있다.

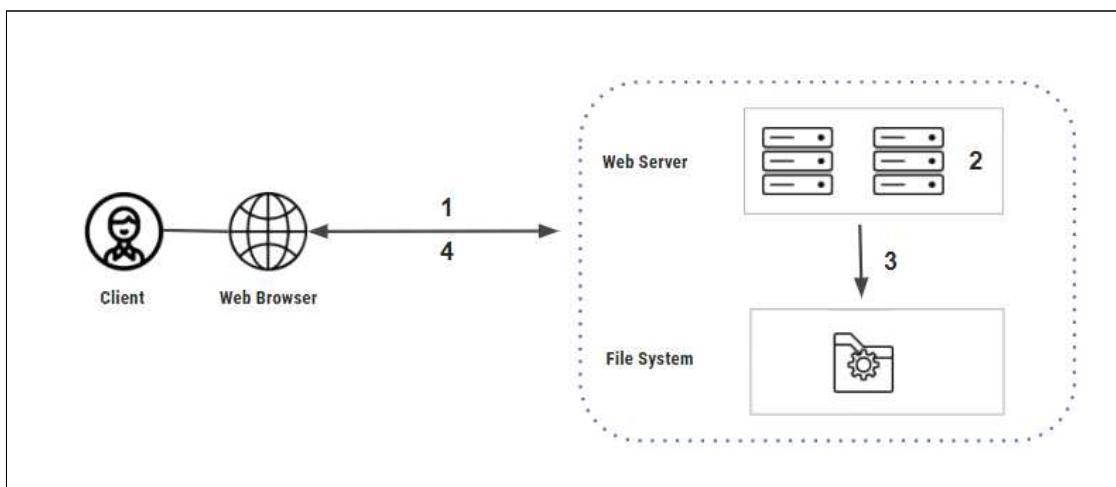
웹 서버 구축

개요

이 제안서는 웹 서버를 구성할 때, 기업 내부에서 구축할 경우와 기업 외부에서 호스팅 할 경우 두 가지를 두고 각각 필요한 시스템 및 필요 하드웨어, 외부 서비스 이용 시 파악해야 하는 요소들을 나열하고 비교하기 위함이다.

웹 서버 정의

웹 서버란 웹 브라우저와 같은 클라이언트로부터 HTTP 프로토콜을 통해 요청을 받아들이고, HTML 문서와 오브젝트 (이미지, 동영상 파일 등)를 전송해 주는 서비스 프로그램이다. 웹 서버 기능을 제공하는 컴퓨터 프로그램을 실행하는 컴퓨터도 웹 서버라고 한다. 대표적인 웹 서버로는 가장 많이 사용되는 아파치(Apache)가 있으며, 엔진엑스(nginx), 마이크로소프트사의 IIS(Internet Information Services) 등이 있다.



현재, 데스크톱 인터넷, 스마트폰을 이용한다면 웹 서버는 필수적으로 거친다. 또한, 전반적인 사회, 경제적 측면에서 인터넷은 빠질 수 없는 매체가 되었다. 인터넷의 탄생에 맞춰 기업들은 자사의 대표 홈페이지나 업무 프로세스를 위한 웹 서버를 구축하기 시작했으며, 현재는 없으면 기업을 제대로 운영하기 힘들 정도의 수준으로 도달하였다. 기업은 앞으로 자사의 목적에 맞게 웹 서버를 효율적으로 활용하는 것이 더더욱 중요하다. 목적에 따라 자사에서 서버 구축을 하거나 외부 업체를 통해 웹 서버 구축을 할 수 있다.

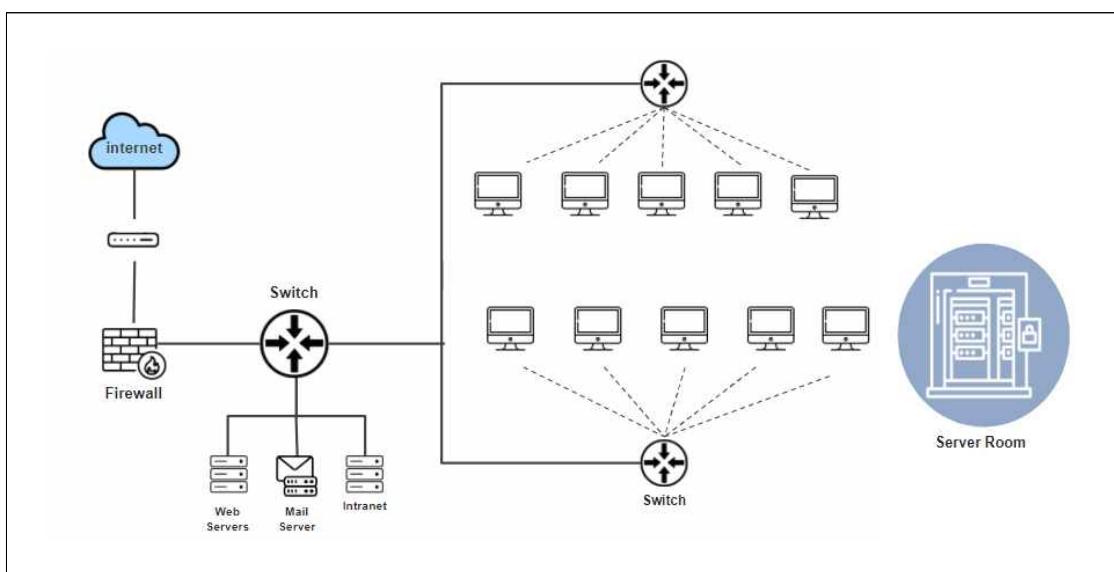
웹 서버 구축

웹 서버를 기업 내부에서 구축할 경우와 외부에서 구축한 경우를 비교했다.

1. 기업 내부에서 웹 서버 구축

기업 내부에서 웹 서버를 구축할 경우, 기업 내에 서버실이 따로 있어야 하는데, 이 서버실에는 네트워크, 스토리지 관련 기기들이 있다. 이 장비는 모두 몇 천만 원대의 비용이 들며, 설치비용도 따로 들며, 유지 및 보수 관리를 위한 에어컨 등 기기와 인력도 필수이다. 이뿐 만 아니라, 보안 장비도 따로 구비해야 한다.

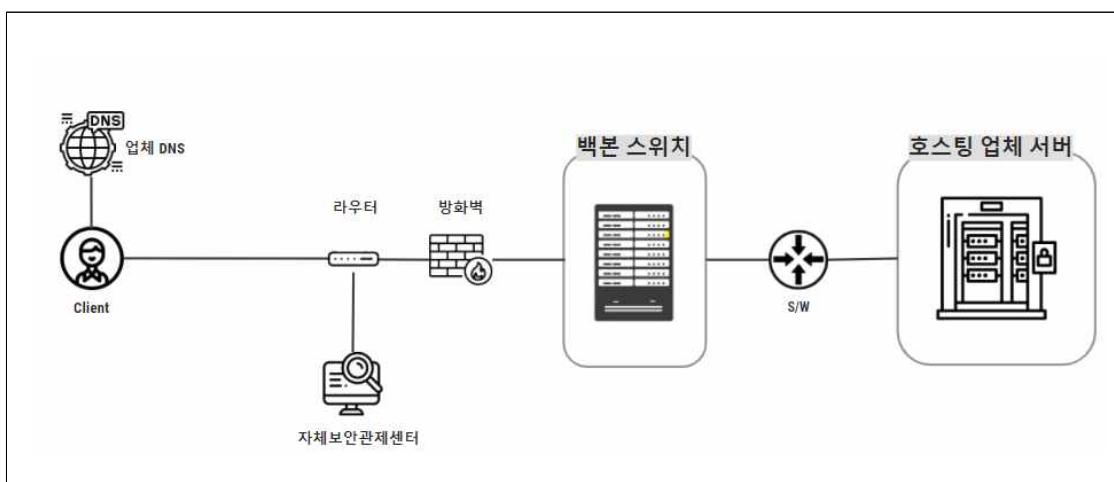
기업 내부에서 쓰는 웹 서버 구축은 따로 내부 DNS 서버를 쓰며, 외부와 통신이 되지 않는 인트라넷처럼 쓸 수 있으며, 라우터로 외부와 통신을 하며 인터넷 사용이 가능하다.



2. 기업 외부에서 웹 서버 구축

기업 외부에서 웹 서버를 구축할 경우 외부 업체에게 외주를 맡길 수 있다. 기업에 목적에 맞는 업체를 선정하여, 전적으로 웹 서버 구축을 맡기거나, 호스팅 업체에 있는 적절한 상품을 구매해 어느 정도 기반이 닦여 있는 상태에서 웹 서버를 구축할 수 있다.

호스팅 업체에서 구현하는 경우 서버를 아예 빌릴 수 있는 서버 호스팅이 있기 때문에 서버실이 따로 필요하지 않다. 업체의 DNS 서버를 사용 가능하며, 필요한 하드웨어만큼 임대, 업체의 인프라와 기술력으로 서버와 네트워크 지원, 모니터링도 가능하기 때문에 그만큼의 시간, 비용을 절감하고 관리의 부담이 적다. 또한, 보안 측면에서도 컨설팅, 솔루션과 다양한 상품이 있어 중요한 문제로 대두되고 있는 보안에 대해서도 관리를 용이하게 할 수 있다.



결론

이 제안서는 기업에서 네트워크, 하드웨어 자원을 고효율, 저비용으로 사용할 수 있도록 기술하였다. 업체의 인프라와 기술력, 보안 관리로 서버 운영과 업무에 집중이 가능하고 서버를 직접 구축하는 것에 비하면 비용에 대한 부담을 줄일 수 있다.

iSCSI Target 구성

개요

iscsi를 구성하여 블록 스토리지를 지원할 계획이다. 우선 target 서버를 설정하고 네트워크 병목현상을 완화하기 위해 iscsi 스위치를 연결하여 구성한다. 기존에 존재하는 서버들이 같은 네트워크 안에서 트래픽을 보내온다면 데이터 손실의 우려가 있으므로 현재 사용하고 있는 네트워크가 아닌 iscsi 전용선을 하나 구성하는 것이다.

1. LVM을 이용하여 iscsi target 서버를 구축
2. iSCSI 전송을 위한 NIC 장착
3. 연결한 NIC를 티밍 작업을 통해 로드 밸런싱 되도록 설정

LVM

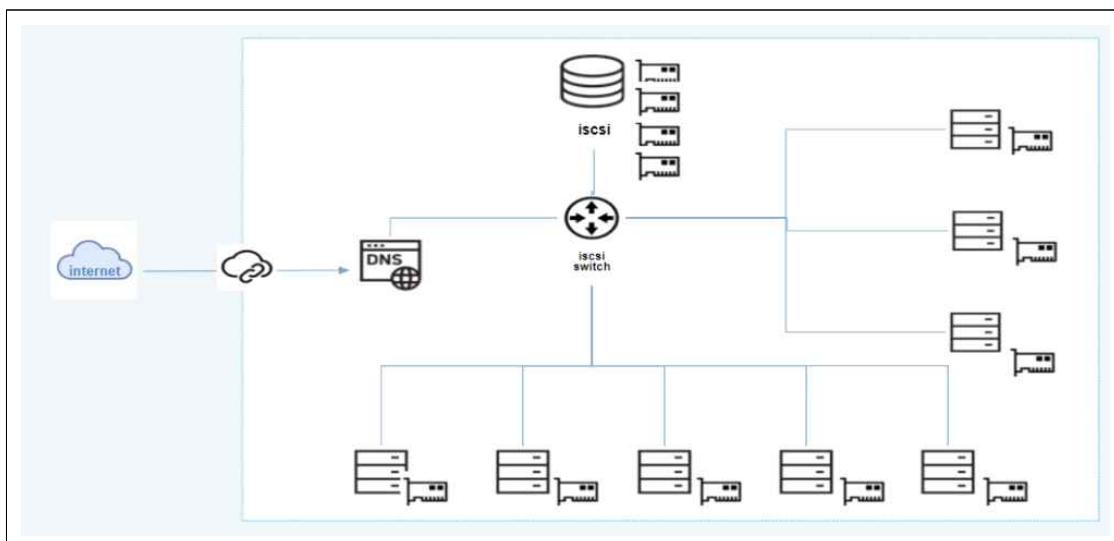
iscsi를 구성하여 블록 스토리지 target 서버를 지원할 계획이다. 이를 위해선 블록 디바이스가 필요한데, 장착되어 있는 디바이스를 사용하기보다 훨씬 더 확장성이 있고 다루기 간편한 LVM을 사용한다. 이는 다음에 있을 용량 확보나 남아있는 용량을 쉽게 쓰기 위함이다. LVM은 기존의 디바이스를 사용하여 구성할 경우 생기는 용량 관리의 불편함을 해결해 준다.

iSCSI Target 구성

생성한 LVM을 토대로 iscsi target을 구축한다. 기업의 서버들이 initiator가 되어 생성한 Target 서버에 접속을 하고 원하는 블록 스토리지를 할당받을 수 있다.

추가 구성

<구성도>



위에 나와 있는 인프라 구성도를 보면 iSCSI Target 서버를 구축만 하는 것이 아니라 추가적인 요소들도 확인하여 구상하는 것이 목표이다.

첫 번째 조건은 기존에 존재하는 서버들의 네트워크가 아닌 iscsi 전용의 네트워크를 사용해야 한다. 호스트 서버들에 iSCSI 전용 NIC를 장착하고 데이터 전송을 하게 한다. 또한, iSCSI Target 서버에도 NIC를 여러 개 장착하여 다음에 나올 두 번째 조건도 부합하도록 사전 작업을 해준다.

두 번째 조건은 트래픽을 분산시키는 것이다. 여러 대의 호스트(initiator)가 iSCSI에 접속하려고 한다. 이 경우 중앙의 스위치가 없다면 여러 트래픽이 막무가내로 전송되기 때문에 과부하가 걸릴 수 있다. 이런 병목현상을 완화하고자 iSCSI 서버(target)에 패킷을 전송하는 형태가 아닌 중앙에 스위치를 하나 두어 트래픽들을 하나로 모아 해당 스위치와 iscsi 서버를 연결하는 형태로 구성한다. 또한, 위 조건에서 iSCSI Target 서버에 장착한 여러 개의 NIC를 라운드로бин이나 로드밸런싱으로 구성하여 들어오는 트래픽을 잘 분산할 수 있도록 한다. 들어오는 트래픽 순서대로 지정되는 라운드로빈 모드보단 점유량을 보고 트래픽을 분산시키는 로드밸런싱 모드가 적합하다.

마 무 리

결론

제안서의 구성으로 비용과 추가적인 인력을 줄일 수 있으며, 결론적으로 서버 사용에 집중할 수 있으며 업무에 차질이 생기지 않도록 하는 효율적인 구성이다.

최종 구성도

제안서의 내용을 도식화하여 작성한 구성도이다.

