

## 2. 클라우드 컴퓨팅 기술

- 클라우드 인프라
- 가상화 기술 (서버 가상화, 네트워크 가상화, 스토리지 가상화)

### 2.1 클라우드 인프라

**클라우드 인프라**는 클라우드 컴퓨팅 서비스가 실행되는 하드웨어 및 소프트웨어 리소스의 모음입니다. 클라우드 제공업체는 서버, 스토리지, 네트워크와 같은 수천 개의 IT 인프라 구성 요소를 갖춘 다수의 데이터센터를 유지 관리합니다.

이러한 데이터 센터에는 서버, 스토리지, 네트워킹 장비 등 물리적 IT 환경에 있는 거의 모든 컴퓨팅 리소스가 들어 있는 랙(Rack) 또는 표준화된 컨테이너가 있습니다.

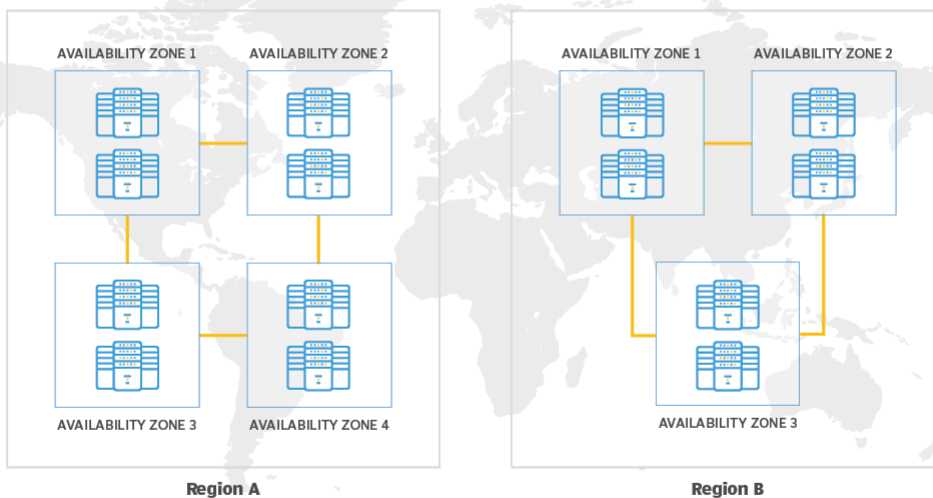
- **컴퓨팅 리소스:** 클라우드 제공업체는 가상 서버, 베어메탈 서버, “서버리스” 컴퓨팅 리소스 등 다양한 컴퓨팅 옵션을 제공합니다. 클라우드 데이터 센터의 대부분의 서버는 하이퍼바이저를 실행하여 가상화 기술을 기반으로 하는 소프트웨어 기반 컴퓨터인 가상 서버 또는 가상 머신 (줄여서 VM이라고도 함) 을 생성합니다. 랙에 있는 다른 서버는 가상화되지 않은 물리적 서버인 베어메탈 (Bare Metals) 서버입니다. 고객은 필요할 때 VM 및 베어메탈 서버를 프로비저닝하고 해당 서버에서 워크로드를 실행할 수 있습니다. 또한 클라우드 사용자는 가상 시스템 상단의 추상화 계층인 서버리스 컴퓨팅 리소스에서 워크로드를 실행할 수 있습니다.
- **스토리지:** 정보 및 데이터는 파일, 코드, 문서, 이미지, 비디오, 백업, 스냅샷 및 데이터베이스로 구성될 수 있으며 클라우드의 다양한 유형의 스토리지 옵션에 저장할 수 있습니다. 베어메탈 서버와 가상 서버는 로컬 드라이브의 기본 스토리지와 함께 프로비저닝됩니다.
- **네트워킹:** 클라우드 데이터 센터의 네트워킹 인프라에는 라우터 및 스위치와 같은 기존 네트워킹 하드웨어가 포함되지만, 클라우드 사용자에게 더 중요한 것은 클라우드 공급자가 API를 통해 특정 네트워킹 리소스를 가상화하거나 프로그래밍 방식으로 사용할 수 있는 SDN (Software Defined Networking) 옵션을 제공한다는 것입니다. 이를 통해 클라우드에서 네트워크 프로비저닝, 구성 및 관리를 더 쉽게 수행할 수 있습니다.
  - 클라우드의 서버를 프로비저닝할 때는 공용 및 사설 네트워크 인터페이스를 설정해야 합니다. 이름에서 알 수 있듯이 공용 네트워크 인터페이스는 서버를 공용 인터넷에 연결하는 반면, 사설 인터페이스는 다른 클라우드 리소스에 대한 연결을 제공하고 보안을 유지하는 데 도움이 됩니다. 물리적 IT 환경에서와 마찬가지로 클라우드의 네트워크 인터페이스에는 IP 주소와 서브넷이 자동으로 할당되거나 구성되어 있어야 합니다. 클라우드 환경에서는 보안 그룹과 ACL (액세스 제어 목록) 을 설정하여 리소스에 액세스할 수 있는 네트워크 트래픽과 사용자를 구성하는 것이 훨씬 더 중요합니다.
  - 클라우드 내 리소스의 보안 강화 및 격리를 위해 대부분의 클라우드 공급자는 VLAN (가상 근거리 통신망), VPC (가상 사설 클라우드) 및 VPN (가상 사설망) 을 제공합니다. 방화벽, 로드 밸런서, 게이트웨이 및 트래픽 분석기와 같은 일부 기존 하드웨어 어플라이언스도 가상화하여 클라우드에서 서비스로 제공할 수 있습니다.
  - 클라우드 제공업체가 제공하는 또 다른 네트워킹 기능으로는 CDN (Content Delivery Networks) 이 있습니다. CDN은 콘텐츠를 전 세계 여러 지점에 배포하여 콘텐츠에 액세스하는 사용자가 가장 가까운 지점에서

콘텐츠를 가져와 더 빠르게 액세스할 수 있도록 합니다.

**가용영역(Availability Zone)**은 1개 이상의 물리 데이터센터를 묶은 논리적인 데이터센터를 의미합니다. 즉, 가용영역은 1개의 데이터센터일 수도 있고, 2개 이상의 데이터센터를 묶어놓은 것일 수도 있지만, 개념적으로는 하나의 가상 데이터센터라고 볼 수 있습니다. 가용영역 간에는 독립적인 전원, 냉온습, 물리적 보안 등의 시설을 갖추게 되며, 물리적으로 떨어져 있기 때문에 데이터센터의 물리적인 장애가 발생하더라도 가용영역 간에 영향을 미치지 않습니다.

**리전(Region)**은 이러한 가용영역이 2개 이상 구성된 지리적 영역입니다. 각 리전은 보통 나라나 대륙을 기준으로 나뉘며, 다른 리전과 완전히 독립적이어서, 한 리전에서 문제가 발생하더라도 다른 리전에는 영향을 미치지 않습니다. 또한, 각 리전은 로컬 사용자에게 낮은 대기 시간을 제공하고, 데이터 주권(데이터가 물리적으로 위치한 나라의 법을 따름)을 준수할 수 있게 합니다.

## Availability zones vs. regions



출처: [https://cdn.ttgtmedia.com/rms/onlineimages/aws\\_availability\\_zones\\_vs\\_regions-f.png](https://cdn.ttgtmedia.com/rms/onlineimages/aws_availability_zones_vs_regions-f.png)

## 2.2 가상화 기술

가상화(Virtualization)는 물리적인 컴퓨터 자원을 추상화하여 하나의 물리적 장치에서 여러 가상 환경을 실행할 수 있도록 하는 기술입니다. 이를 통해 컴퓨팅 자원을 보다 효율적으로 사용하고, 하드웨어와 소프트웨어의 독립성을 확보할 수 있습니다.

가상화는 주로 다음과 같은 종류로 나뉩니다:

- **서버 가상화:** 물리적 서버 하나를 여러 개의 가상 서버로 분리하여 사용하는 방식입니다. 이를 통해 여러 운영체제를 하나의 서버에서 동시에 실행할 수 있습니다.
- **네트워크 가상화:** 물리적인 네트워크를 논리적인 네트워크로 분할하여 관리할 수 있는 기술입니다. 이를 통해 네트워크 자원의 활용도를 높이고, 보안성을 향상시킬 수 있습니다.
- **스토리지 가상화:** 여러 개의 물리적인 스토리지를 하나의 가상 스토리지로 통합하여 사용하는 기술입니다. 이를 통해 스토리지 자원을 보다 효율적으로 관리할 수 있습니다.

가상화는 클라우드 컴퓨팅, 데이터센터의 서버 통합, 개발 및 테스트 환경 등 다양한 분야에서 널리 활용되고 있습니다.

## 2.2.1 서버 가상화

서버 가상화 기술을 가능하게 하는 방식은 크게 하이퍼바이저 방식과 컨테이너 방식으로 구분된다.

### 2.2.1.1 하이퍼바이저 방식

하이퍼바이저는 단순히 물리적 서버 또는 호스트 위에서 실행되는 일종의 소프트웨어로서, 물리적 서버에서 리소스를 가져와 가상머신(Virtual Machine)에 할당하는 일을 수행합니다. 하이퍼바이저에는 두 가지 주요 유형이 있습니다.

- **유형 1 (베어메탈 하이퍼바이저):** 물리적 서버 위에 직접 설치되는 하이퍼바이저입니다. 베어메탈 하이퍼바이저라고도 합니다. 이러한 하이퍼바이저는 가장 자주 사용되는 하이퍼바이저 유형으로, 가장 안전하고, 지연 시간을 줄이며, 시장에서 가장 많이 사용되는 하이퍼바이저입니다. VMware, ESXi, Microsoft Hyper-V, 오픈 소스 KVM 등을 예로 들 수 있습니다.
- **유형 2 (호스트형 하이퍼바이저):** 물리적 서버와 하이퍼바이저 사이에 호스트 OS 계층이 있으므로 유형 1의 하이퍼바이저보다 지연시간이 더 길습니다. 이들은 주로 최종 사용자 가상화에 사용되며 시장에서 오라클, 버추얼박스 또는 VMware 워크스테이션이라고 하는 제품을 볼 수 있습니다.



\*이미지 출처 : 안성원, 클라우드 컴퓨팅과 인공지능의 만남, IT데일리 전문가 강좌, 2017.7  
소프트웨어정책연구소, 클라우드 보안의 핵심이슈와 대응책, 2017.

하이퍼바이저 방식의 몇 가지 주요 이점은 다음과 같습니다.

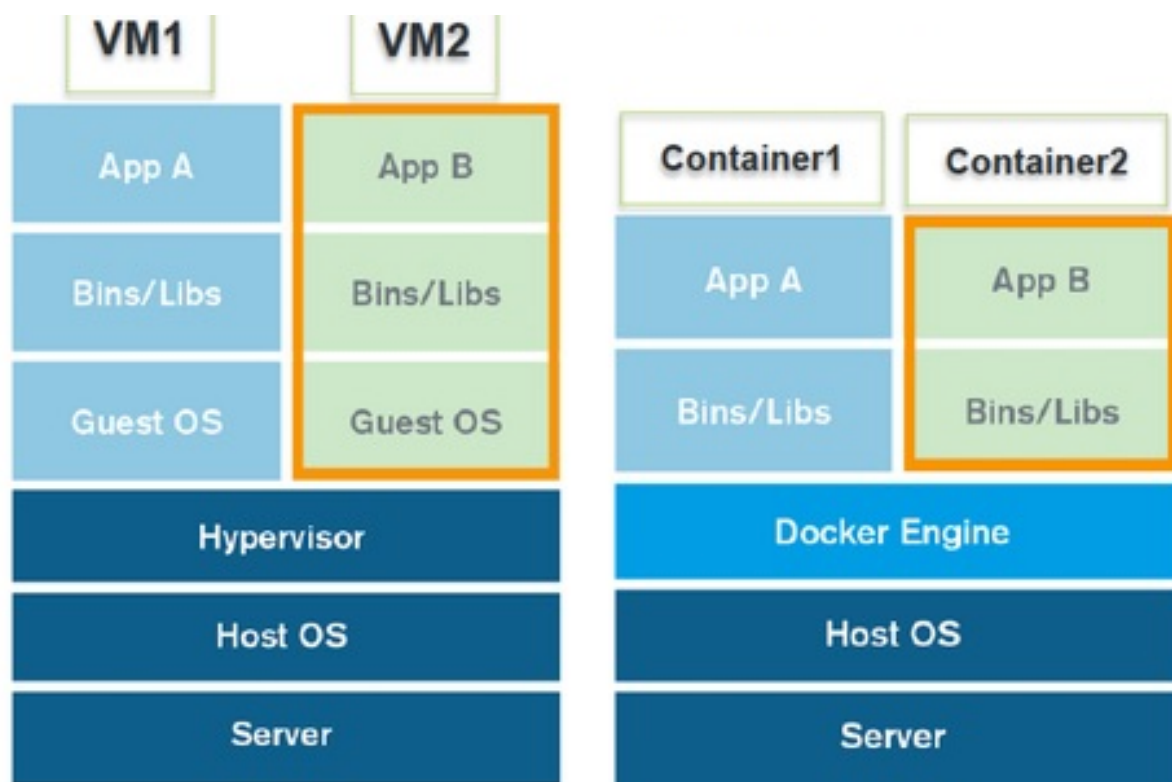
1. **비용 절감.** 하나의 인프라에서 여러 가상머신을 실행할 수 있으므로, 물리적으로 많은 서버를 유지 관리할 필요가 없게되고, 이로써 유지 관리 비용을 절감할 수 있습니다.
2. **민첩성과 속도.** 가상머신을 가동시키는 것은 비교적 쉽고 빠릅니다. 개발자가 개발 테스트 시나리오를 실행할 수 있도록 새로운 환경을 요구한다면, 완전히 새로운 물리적인 환경을 프로비저닝하는 것보다 가상화를 통해 가상머신을 생성하여 가동하는 것이 훨씬 간단하고 신속합니다.
3. **다운타임 감소.** 라이브 마이그레이션은 가동 중인 가상 머신을 물리적 서버 간에 실시간으로 이동시킬 수 있는 기술입니다. 이를 통해 서버의 하드웨어 유지보수가 필요하거나 문제가 발생했을 때, 가상 머신을 중단하지 않고 다른 서버로 옮겨 갈 수 있습니다. 이 과정에서 사용자에게 거의 영향을 미치지 않아 다운타임을 크게 줄일 수 있습니다.

### 2.2.1.2 컨테이너 방식

하이퍼바이저가 호스트 OS 없이 직접적으로 가상머신을 구축할 수 있다 하더라도 하이퍼바이저 위에 여러 개의 OS를 구동시키는 것 자체가 많은 오버헤드를 야기시킵니다. 컨테이너 가상화는 호스트 OS상에 논리적인 영역(컨테이너)을 만들고, 애플리케이션을 동작시키는 데 필요한 라이브러리와 애플리케이션만을 컨테이너에 포함시켜 게스트 OS 없이 개별 서버와 같은 실행 환경을 구축하는 방식을 말합니다.

여기서 컨테이너란 애플리케이션을 실행하는 데 필요한 모든 파일, 라이브러리, 설정 파일 등을 하나로 묶어 독립된 실행 환경을 제공하는 가상화 기술입니다. 컨테이너는 운영체제 수준에서 격리된 상태로 애플리케이션을 실행할 수 있으며, 각 컨테이너는 호스트 운영체제의 커널을 공유하지만, 파일 시스템, 네트워크, 프로세스 등은 각 컨테이너별로 독립적입니다.

컨테이너는 애플리케이션과 그에 필요한 모든 것을 함께 패키징하여 어느 환경에서든 동일하게 실행될 수 있습니다. 이를 통해 애플리케이션이 개발 환경, 테스트 환경, 프로덕션 환경 간에 일관되게 동작할 수 있습니다.



출처:

[https://hub.alfresco.com/legacyfs/online/alfresco/24274\\_vms-vs-containers.png](https://hub.alfresco.com/legacyfs/online/alfresco/24274_vms-vs-containers.png)

컨테이너의 특징은 다음과 같습니다.

- **경량화:** 가상 머신(VM)과 달리 컨테이너는 운영체제를 독립적으로 포함하지 않고 호스트 운영체제의 커널을 공유합니다. 이로 인해 자원 소비가 적고, 컨테이너의 시작과 종료가 매우 빠릅니다.
- **이식성:** 컨테이너는 운영체제 수준에서의 격리를 통해 애플리케이션이 어느 환경에서도 동일하게 동작하도록 보장합니다. 개발자 컴퓨터에서 실행한 애플리케이션을 클라우드나 다른 서버 환경에 배포할 때 환경 설정 문제로 인한 오류를 줄일 수 있습니다.
- **독립성과 격리:** 컨테이너는 각각 격리된 환경에서 동작하므로, 하나의 컨테이너에서 문제가 발생해도 다른 컨테이너에 영향을 미치지 않습니다. 각 컨테이너는 독립적인 파일 시스템, 네트워크 설정, 프로세스 공간을 가집니다.
- **빠른 배포 및 확장:** 컨테이너는 빠르게 생성, 삭제, 복제할 수 있으며, 대규모 애플리케이션 확장이 가능합니다. 컨테이너 기반 오케스트레이션 도구(Kubernetes 등)를 사용하면 수천 개의 컨테이너를 자동으로 관리하고 확장할 수 있습니다.

컨테이너의 구성 요소는 다음과 같습니다.

- **이미지(Image):** 컨테이너는 이미지를 기반으로 실행됩니다. 이미지에는 애플리케이션 코드, 필요한 라이브러리, 설정 파일 등이 포함됩니다. 컨테이너 이미지는 한번 작성된 후 여러 환경에서 재사용할 수 있습니다.
- **컨테이너(Container):** 이미지를 기반으로 실행되는 실행 가능한 인스턴스가 컨테이너입니다. 컨테이너는 애플리케이션을 실행하는 독립적인 환경을 제공하며, 가동 중인 컨테이너는 애플리케이션이 운영되는 상태입니다.
- **레지스트리(Registry):** 컨테이너 이미지를 저장하고 관리하는 저장소입니다. Docker Hub와 같은 퍼블릭 레지스트리나 프라이빗 레지스트리를 통해 이미지를 업로드하거나 다운로드할 수 있습니다.

## 컨테이너의 장점

- **자원 효율성:** 가상 머신보다 경량화되어 자원 사용이 적고, 빠르게 실행됩니다.
- **일관성:** 개발 환경, 테스트 환경, 프로덕션 환경에서 동일한 동작을 보장합니다.
- **확장성:** 클라우드 환경에서 수많은 컨테이너를 쉽게 배포하고 관리할 수 있습니다.
- **이식성:** 애플리케이션과 모든 의존성을 하나로 묶어 어떤 시스템에서도 일관되게 실행됩니다.

## 대표적인 컨테이너 도구

- **Docker:** 가장 널리 사용되는 컨테이너 플랫폼으로, 컨테이너 이미지를 만들고, 관리하며, 실행할 수 있는 기능을 제공합니다.
- **Kubernetes:** 컨테이너를 대규모로 관리하고 자동화하는 오케스트레이션 도구입니다.
- **Podman:** Docker와 유사하지만, 데몬리스(Daemonless) 방식으로 컨테이너를 관리할 수 있는 도구입니다.

컨테이너 기술은 개발 및 운영 환경을 통합하고 애플리케이션의 배포, 확장, 이식성을 크게 향상시켜 DevOps와 클라우드 네이티브 애플리케이션 개발에서 핵심적인 역할을 합니다.

## 2.2.2 네트워크 가상화

**네트워크 가상화(Network Virtualization)**는 물리적인 네트워크 인프라를 추상화하여 소프트웨어 기반의 논리적인 네트워크를 생성하고, 네트워크 자원(예: 스위치, 라우터, 방화벽 등)을 유연하고 효율적으로 관리할 수 있게 해주는 기술입니다. 이를 통해 네트워크 하드웨어와 무관하게 네트워크 자원의 배치, 구성, 관리가 가능해져 더 유연하고 확장 가능한 네트워크 환경을 제공합니다.

네트워크 가상화는 서버 가상화와 유사한 개념으로, 물리적 네트워크 자원을 여러 개의 논리적 네트워크로 나누거나, 여러 네트워크 자원을 통합하여 하나의 네트워크로 구성할 수 있습니다. 이 기술은 특히 클라우드 컴퓨팅 환경에서 중요한 역할을 하며, 네트워크 운영을 자동화하고 최적화하는 데 필수적입니다.

네트워크 가상화를 실현할 수 있는 대표적인 기술로 NFV (Network Function Virtualization)와 SDN (Software Defined Network)이 있습니다.

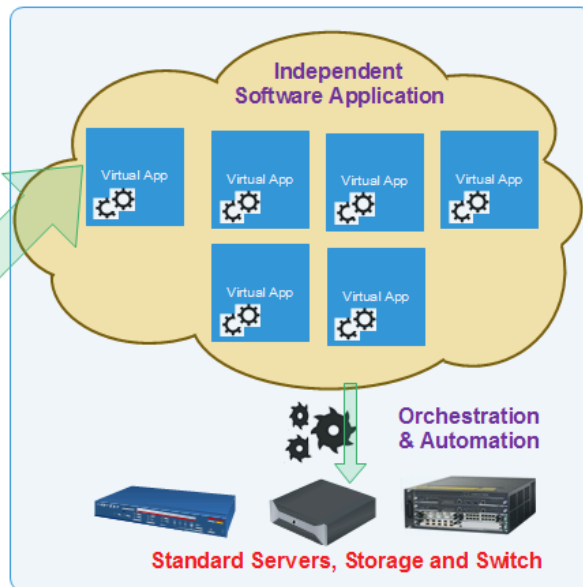
- **NFV (Network Function Virtualization) :** NFV는 네트워크 장비의 도입 없이 네트워크 기능을 소프트웨어적으로 가상화하여 범용 서버에서 구축하는 기술을 말합니다. 일반적으로 네트워크 구축을 위해 라우터, 게이트웨이, 방화벽과 같은 고가의 전용 네트워크 장비를 사용하였으나, NFV 기술을 통해 로드밸런서, 방화벽 등의 전용 하드웨어의 기능을 소프트웨어로 구현하여 온디맨드 서비스로 제공할 수 있습니다.



## Traditional Network Appliance Approach



## Network Virtualisation Approach



출처:

<https://techplayon.com/wp-content/uploads/2017/08/NFV-Comp.png>

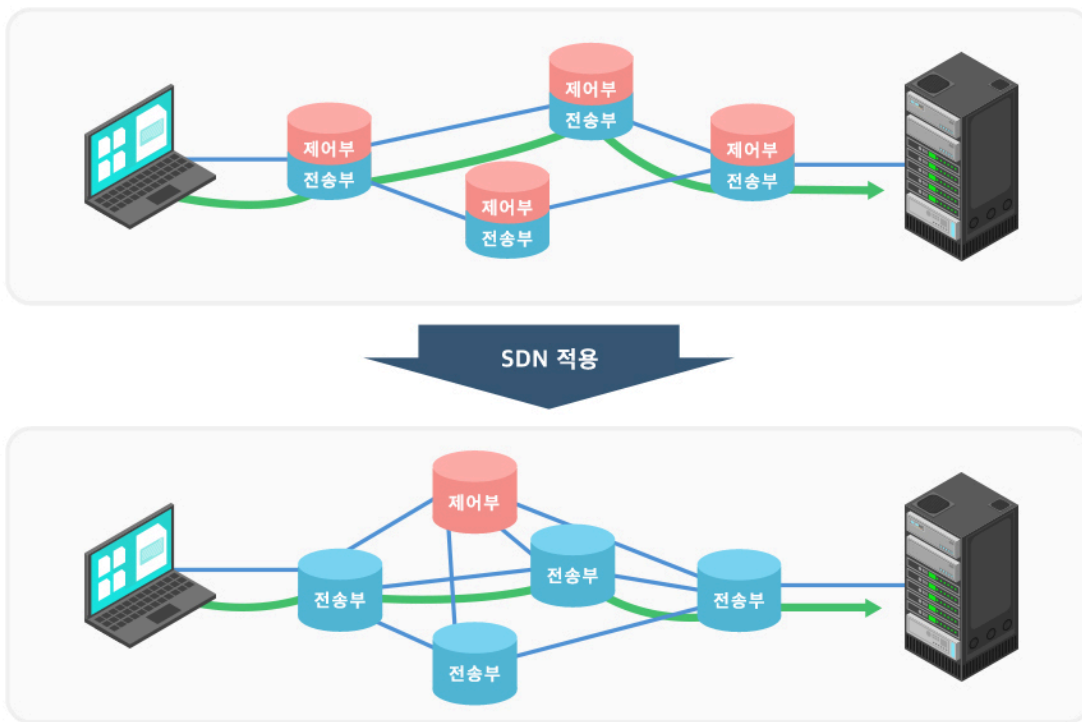
### ○ 장점

- **비용절감:** 전용 하드웨어 대신 상용 서버(COTS, Commercial Off-The-Shelf)를 사용하여 비용을 크게 절감할 수 있습니다. 하드웨어 의존도를 낮추면서도 네트워크 기능을 유연하게 제공할 수 있습니다
- **유연성 및 확장성:** 필요한 네트워크 기능을 소프트웨어로 쉽게 배포하고, 필요에 따라 기능을 확장하거나 축소할 수 있습니다. 트래픽 증가에 따라 물리적인 장비를 추가하지 않고도 네트워크 기능을 확장할 수 있습니다
- **서비스 민첩성:** 새로운 서비스를 빠르게 제공하거나 기존 서비스를 확장할 수 있어, 비즈니스 요구사항에 빠르게 대응할 수 있습니다. 네트워크 환경이 소프트웨어로 관리되기 때문에 신속한 변화가 가능합니다.

### ○ 단점

- **성능 및 안전성 문제:** 범용 서버에 구성하다 보니, 전용장비에 비해 성능과 안정성이 떨어질 수 있습니다.
- **가상화 환경의 보안 취약점:** NFV는 가상화 기술을 기반으로 하기 때문에 기존의 하드웨어 기반 네트워크보다 보안 취약점이 많을 수 있습니다. 하이퍼바이저, 가상 머신, 컨테이너 등 가상화 계층에서 발생할 수 있는 보안 위협이 증가할 수 있습니다.

- **SDN (Software Defined Network):** SDN은 전통적으로 라우터, 스위치 등 하드웨어 중심의 네트워크 인프라를 소프트웨어의 형태로 진화시켜 유연성과 민첩성을 향상시킬 수 있는 네트워크 기술을 말합니다.



출처: [https://img1.daumcdn.net/thumb/R1280x0/?](https://img1.daumcdn.net/thumb/R1280x0/?scode=mtistory2&fname=https%3A%2F%2Fblog.kakaocdn.net%2Fdn%2FUx31d%2FbtrastS3hRc%2FBHUtpzFMPSQBbrE4arJiVK%2Fimg.png)

[scode=mtistory2&fname=https%3A%2F%2Fblog.kakaocdn.net%2Fdn%2FUx31d%2FbtrastS3hRc%2FBHUtpzFMPSQBbrE4arJiVK%2Fimg.png](https://img1.daumcdn.net/thumb/R1280x0/?scode=mtistory2&fname=https%3A%2F%2Fblog.kakaocdn.net%2Fdn%2FUx31d%2FbtrastS3hRc%2FBHUtpzFMPSQBbrE4arJiVK%2Fimg.png)

기존 네트워크에서의 라우터는 패킷을 전송하는 네트워크 장치로서, 제어부와 전송부로 구성되어 있습니다. 제어부는 라우팅 라우팅 프로토콜을 실행해 라우팅 테이블을 만들고, 전송부는 라우팅 테이블을 참조해 패킷을 이동시키는 역할을 수행합니다. 제어부와 전송부가 하나의 네트워크 장비에 결합되어 있으므로, 엔터프라이즈 규모의 네트워크를 관리하기 위해서는 전체 라우터를 개별적으로 관리해야 할 뿐만 아니라 라우터 장비가 벤더 종속적이므로, 벤더별 프로토콜이 달라 운영 비용 증가하는 문제가 존재합니다.

SDN은 제어부와 전송부를 분리하여, 중앙에서 제어계층을 관리하고, 개별 라우터는 전송기능만 담당하도록 합니다. 따라서 네트워크 운영자는 중앙에서 네트워크를 소프트웨어로 프로그래밍하여 자동화된 관리 및 정책을 적용할 수 있습니다.

#### ○ 장점

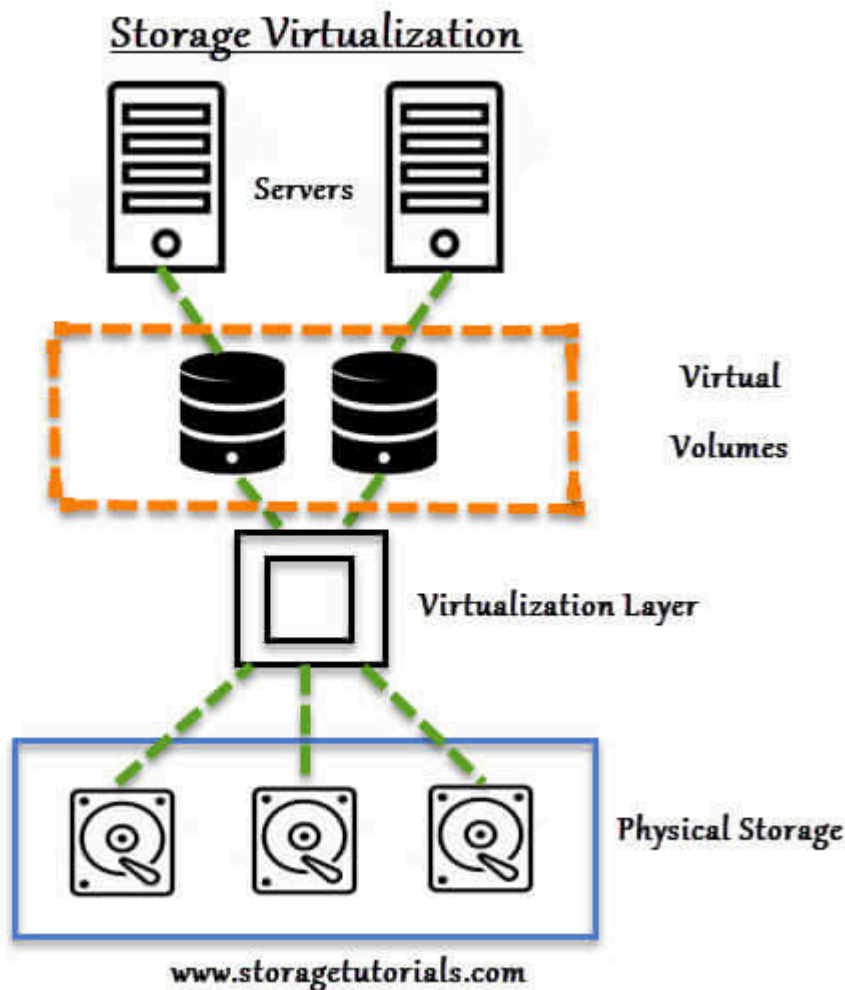
- **유연한 네트워크 관리:** SDN을 통해 네트워크 관리를 소프트웨어 방식으로 할 수 있으므로, 새로운 네트워크 서비스나 트래픽 정책을 쉽게 도입하고 적용할 수 있습니다. 네트워크 장비를 직접 설정할 필요 없이 중앙에서 간단하게 트래픽 경로를 변경하거나 설정할 수 있습니다.
- **자동화 및 운영 효율성:** SDN은 자동화를 통해 네트워크 운영을 효율적으로 만들 수 있습니다. 예를 들어, 네트워크 트래픽이 증가할 경우 트래픽을 자동으로 재분배하거나 대역폭을 동적으로 조정할 수 있습니다. 이를 통해 네트워크 가용성과 성능을 최적화할 수 있습니다.
- **비용 절감:** SDN은 특정 벤더에 종속된 네트워크 장비에 대한 의존도를 줄이고, 일반적인 상용 하드웨어(COTS)를 사용할 수 있게 합니다. 네트워크 자원의 효율적인 사용이 가능해져 하드웨어 비용을 절감할 수 있습니다.
- **보안 및 정책 관리 강화:** 네트워크 보안 정책을 중앙에서 관리하고 일관되게 적용할 수 있습니다. 보안 위협이 감지되면 즉각적으로 네트워크 설정을 변경하여 위협에 대응할 수 있으며, 특정 트래픽을 실시간으로 차단하거나 경로를 변경하는 등의 유연한 보안 관리가 가능합니다.

#### ○ 단점

- **초기 도입 비용:** 기존 네트워크 인프라를 SDN으로 전환하는 데는 초기 비용과 시간이 많이 소요될 수 있습니다.
- **기술 성숙도:** SDN은 상대적으로 새로운 기술이기 때문에 일부 기능은 아직 성숙하지 않았으며, 표준화와 상호 운용성 문제가 존재할 수 있습니다.
- **보안 위험:** SDN 컨트롤러가 네트워크의 중앙 관리 시스템 역할을 하기 때문에, SDN 컨트롤러가 해킹당하거나 문제가 발생할 경우 네트워크 전체가 위험에 노출될 수 있습니다.

### 2.2.3 스토리지 가상화

스토리지 가상화(Storage Virtualization)는 물리적인 스토리지 리소스를 추상화하고 통합하여 사용어나 애플리케이션이 이를 하나의 논리적 단위로 인식하도록 만드는 기술입니다. 이 기술을 사용하면 여러 물리적 스토리지 장치가 하나의 가상 스토리지 풀로 통합되고, 이를 통해 스토리지 리소스를 보다 효율적으로 관리하고 사용할 수 있습니다.



스토리지 가상화의 주요 특징과 이점은 다음과 같습니다.

1. **물리적 리소스와의 추상화** : 스토리지 가상화는 물리적 스토리지 장치의 세부 사항을 숨기고, 사용어나 애플리케이션에게 논리적인 스토리지 공간만을 제공합니다. 물리적으로 여러 스토리지 디바이스가 존재하더라도, 사용자는 이러한 복잡성을 인식할 필요 없이 단일 스토리지로 접근할 수 있습니다.
2. **스토리지 관리의 간소화** : 물리적 장치의 구성을 변경하거나 추가하는 경우에도, 가상화된 스토리지를 통해 시스템 운영에 큰 영향을 미치지 않고 관리할 수 있습니다. 이로 인해 IT 관리자는 보다 유연하게 스토리지를 관리하고, 운영에 미치는 영향 없이 스토리지 리소스를 추가, 제거 또는 재구성할 수 있습니다.
3. **리소스 최적화** : 가상화된 스토리지는 물리적 장치의 용량을 통합하여 사용하는 것이므로, 스토리지 리소스를 더 효율적으로 사용할 수 있습니다. 예를 들어, 특정 스토리지 장치에 여유 공간이 있을 때 가상화 레이어를 통해 자동으로 이를 활용할 수 있어 전체적인 자원 사용률을 높일 수 있습니다.



4. **유연한 확장성** : 스토리지 가상화를 사용하면 새로운 스토리지 장치를 쉽게 추가할 수 있으며, 이를 논리적으로 기존 가상화된 스토리지 풀에 통합할 수 있습니다. 즉, 물리적인 스토리지 장치가 부족해질 때마다 시스템을 중단할 필요 없이 스토리지를 확장할 수 있는 유연성을 제공합니다.
5. **데이터 마이그레이션의 용이성** : 스토리지 가상화는 물리적 장치 간의 데이터 마이그레이션을 보다 원활하게 합니다. 예를 들어, 데이터가 특정 물리적 스토리지에서 다른 장치로 이동해야 할 때, 가상화된 환경에서는 이러한 이동이 사용자에게 투명하게 수행될 수 있으며, 서비스 중단을 최소화할 수 있습니다.
6. **스토리지 중복 제거 및 압축** : 가상화된 스토리지를 통해 데이터 중복 제거(Deduplication) 및 압축과 같은 고급 기능을 구현할 수 있습니다. 이는 스토리지 용량을 더욱 효율적으로 사용하게 하며, 저장 공간을 절약하는 데 도움을 줍니다.
7. **고가용성 및 데이터 보호** : 스토리지 가상화는 중복성을 통해 고가용성을 제공하고, 장애가 발생한 경우 데이터를 자동으로 복구할 수 있는 기능을 제공합니다. 가상화된 스토리지 풀에서는 데이터가 여러 물리적 장치에 분산 저장될 수 있으며, 이를 통해 장애 발생 시에도 데이터를 지속적으로 사용할 수 있습니다.