

2022 A Basic CS skill: ABC Winter School

Potpourri

Team 8

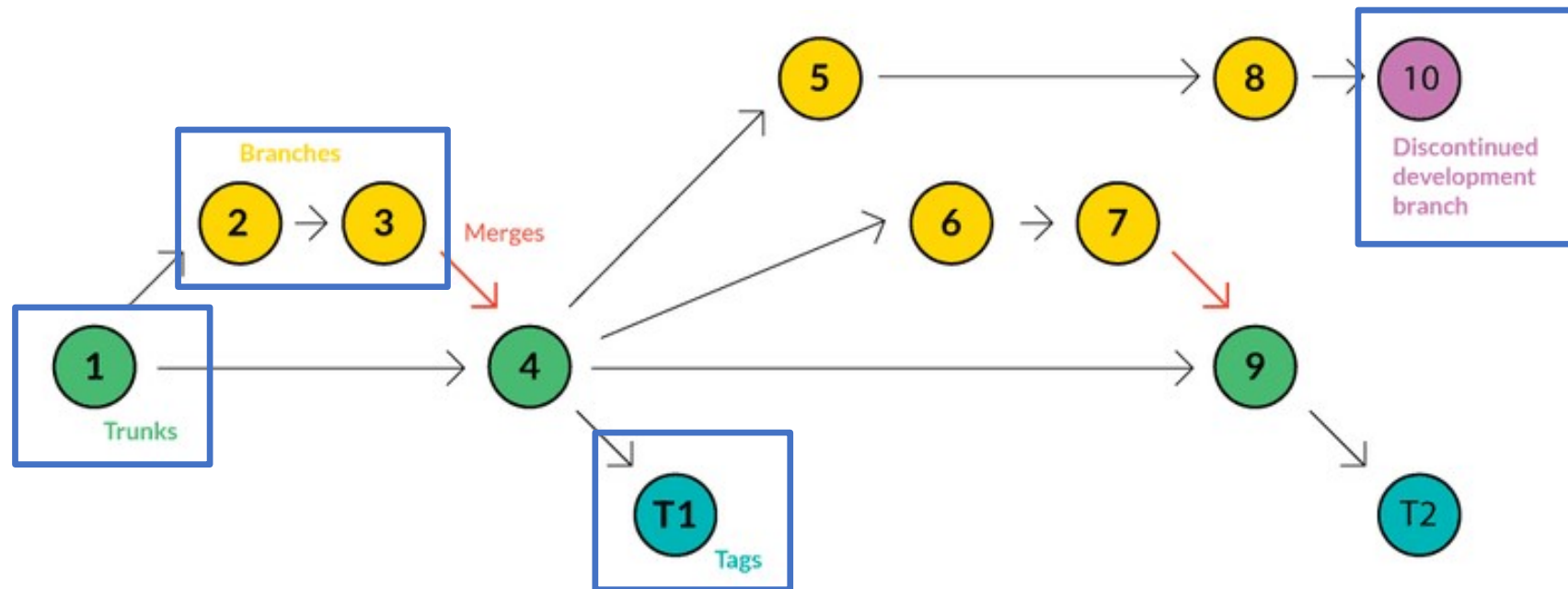
2022 / 02 / 16



ULSAN NATIONAL INSTITUTE OF
SCIENCE AND TECHNOLOGY

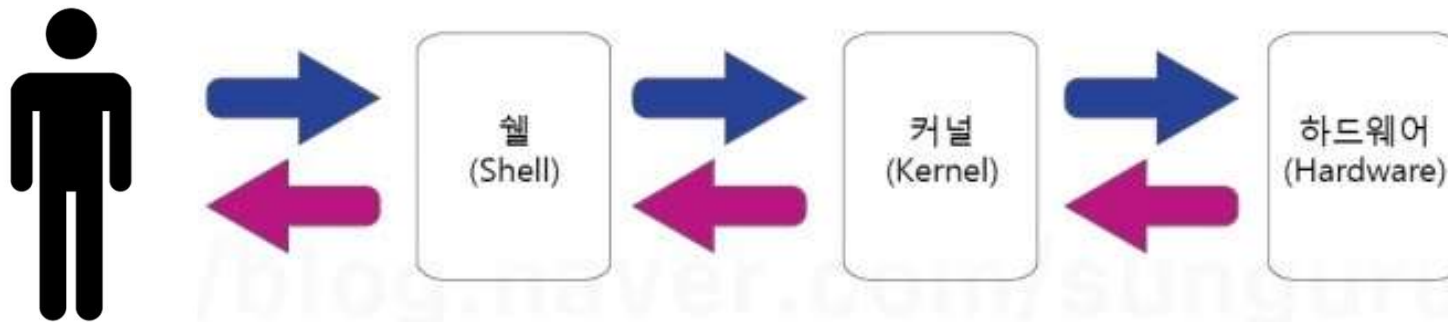
지난 시간 복습 - Git

- Git, version control



지난 시간 복습 - Shell

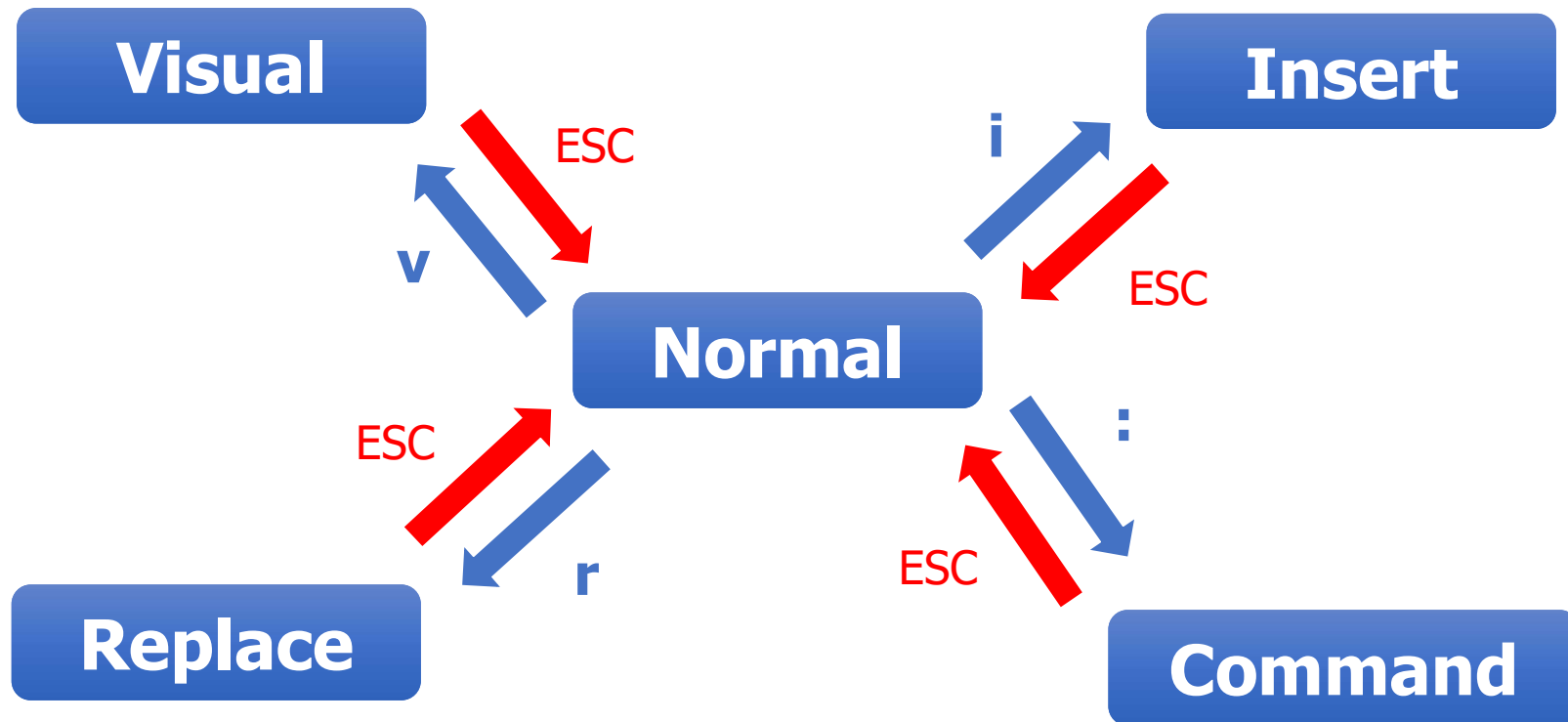
- 명령어와 프로그램을 실행 할 때 사용하는 인터페이스.
- **Linux Kernel**과 **사용자** 사이에 다리 역할을 합니다.



* Kernel : 컴퓨터를 관리하는 운영체제의 핵심 부분 (이라고만 생각 합시다)

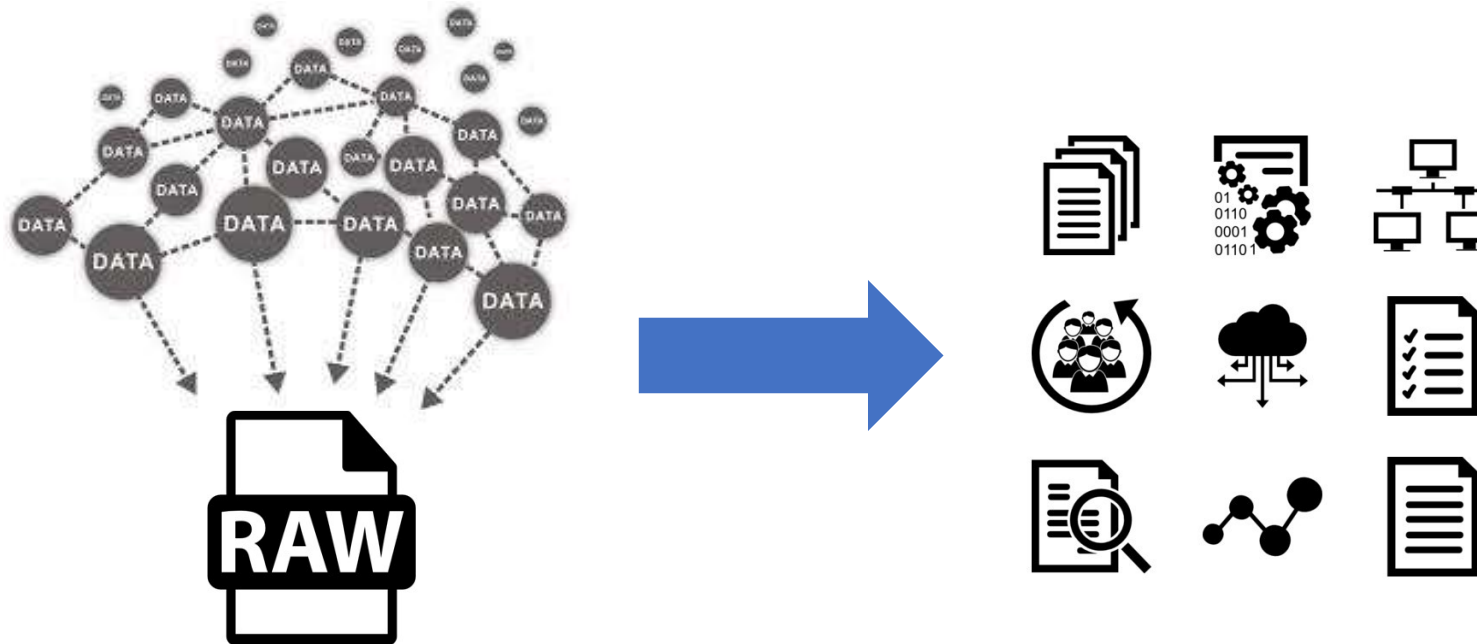
지난 시간 복습 - Vim

- Vim mode



지난 시간 복습 - Data Wrangling

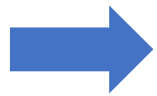
- Data Munging이라고도 합니다.



- 정렬되지 않는 날 것의 데이터를 사용하기 좋게 정렬하고 분류하는 모든 작업을 의미합니다.

지난 시간 복습 - Metaprogramming

- 자기 자신, 또는 다른 프로그램을 데이터 취급하여 프로그래밍 하는 것을 의미 (Wikipedia)
- 프로그램을 운영하기 위한 프로그램을 프로그래밍 하는 것

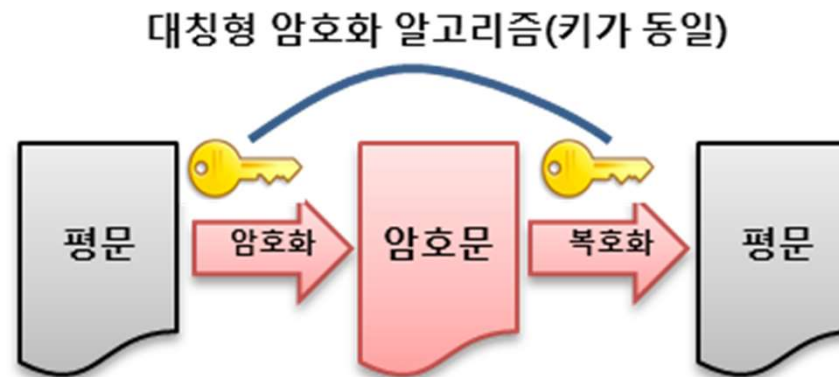


프로그램을 Build 한다

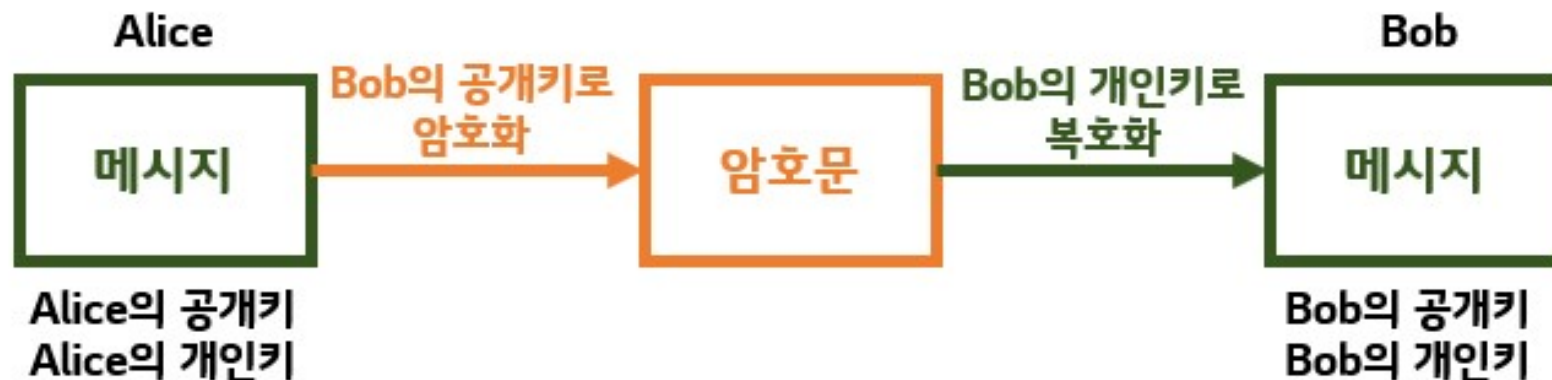
프로그램의 Dependency를 관리한다

지난 시간 복습 - Cryptography

- 대칭 암호화 방식



- 비대칭 암호화 방식



Daemon

- 사용자가 직접 제어하거나 상호작용 하지 않고, 백그라운드에서 여러 작업을 하는 프로그램을 의미합니다.
- 보통 데몬 프로그램은 'd'라는 이름을 끝에 달고 있습니다.
 - Ex) systemd, sshd, httpd 등등...
- 일반적으로 시스템이 시작될 때 데몬 프로그램 역시 시작됩니다.
- 네트워크 요청, 하드웨어 동작, 그 외 프로그램에 반응하는 기능을 담당합니다.

Daemon

- 일반적인 백그라운드 프로그램과 다른 점?



일반적인 백그라운드 프로그램은 터미널을 종료하면 같이 종료되지만,
데몬 프로그램은 터미널을 종료해도 그대로 서버에 남아 있습니다.

- Ex) 웹 서버의 경우를 생각 해봅시다.



Markdown

- 웹 페이지를 구성하는 형태 -> HTML
- HTML의 태그는 사용자가 글을 작성할 때 어렵기 때문에 상대적으로 더 간단한 마크다운을 사용합니다.
- 마크다운은 사용자가 쓴 글을 HTML의 형태로 변환시켜 줍니다.
- Ex) Github의 README.md, Notion, 등등...

```
---
title: "AlloX: Compute Allocation in Hybrid Clusters"
author: Hyunjoon Jeong
layout: post
category: review
---
```

Github에 md 파일

title	author	layout	category
AlloX: Compute Allocation in Hybrid Clusters	Hyunjoon Jeong	post	review

md 파일이 변환된 형태

API (Application Program Interface)

- 응용 프로그램에서 사용할 수 있도록, OS나 언어가 제공하는 기능을 제어할 수 있도록 만든 인터페이스를 의미합니다. (Wiki)
- 쉽게 말해서, 프로그램과 또다른 프로그램을 이어주는 다리 역할을 합니다.
- Ex) Google의 Geocoding API, NAVER Maps API



지도 프로그램의 모든
것을 구현하기 힘들



지도 API 사용

API

- Google APIs 플랫폼이나 NAVER Developers, Kakao Developers등 여러 API를 지원하는 플랫폼들이 있습니다.
- 물론 사용량에 따라 완전 공짜는 아닙니다.



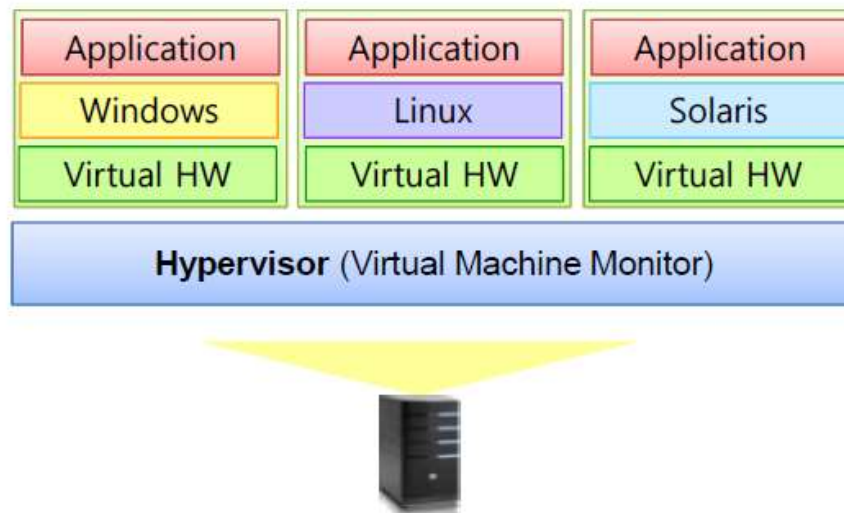
The screenshot shows a table with the following columns: 이름 (Name), ↓ 요청 (Requests), 오류율(%) (Error Rate (%)), 지연 시간, 중앙값(밀리초) (Latency, Median (milliseconds)), and 지연 시간, 95%(밀리초) (Latency, 95% (milliseconds)). The table lists two APIs: Geocoding API and Maps JavaScript API. The '요청' column values are 19 and 2 respectively. A blue box highlights the '요청' column header and its values.

이름	↓ 요청	오류율(%)	지연 시간, 중앙값(밀리초)	지연 시간, 95%(밀리초)
Geocoding API	19	0	453	965
Maps JavaScript API	2	0	16	31

보통은 이러한 요청 횟수에 따라 돈을 부과합니다.

Virtual Machine (VM)

- OS를 포함해 컴퓨터의 시스템 전체를 가상화 시킨 환경을 만드는 툴을 의미합니다.
- 하나의 물리적인 장치(서버 컴퓨터)에서 여러 개의 다른 OS 커널 환경을 구동할 수 있습니다.
- VMWare에서 만든 WorkStation이 대표적인 VM 툴 입니다.

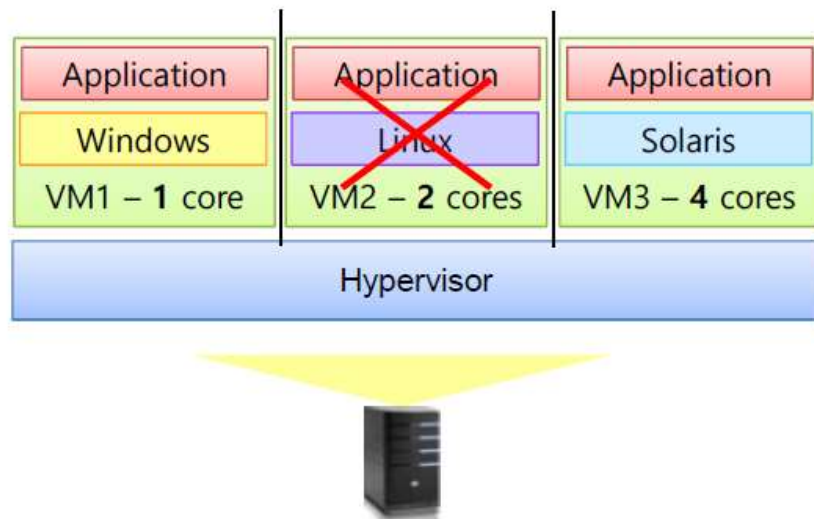


→ 서로 다른 OS 환경

→ 하나의 서버 컴퓨터

Virtual Machine (VM)

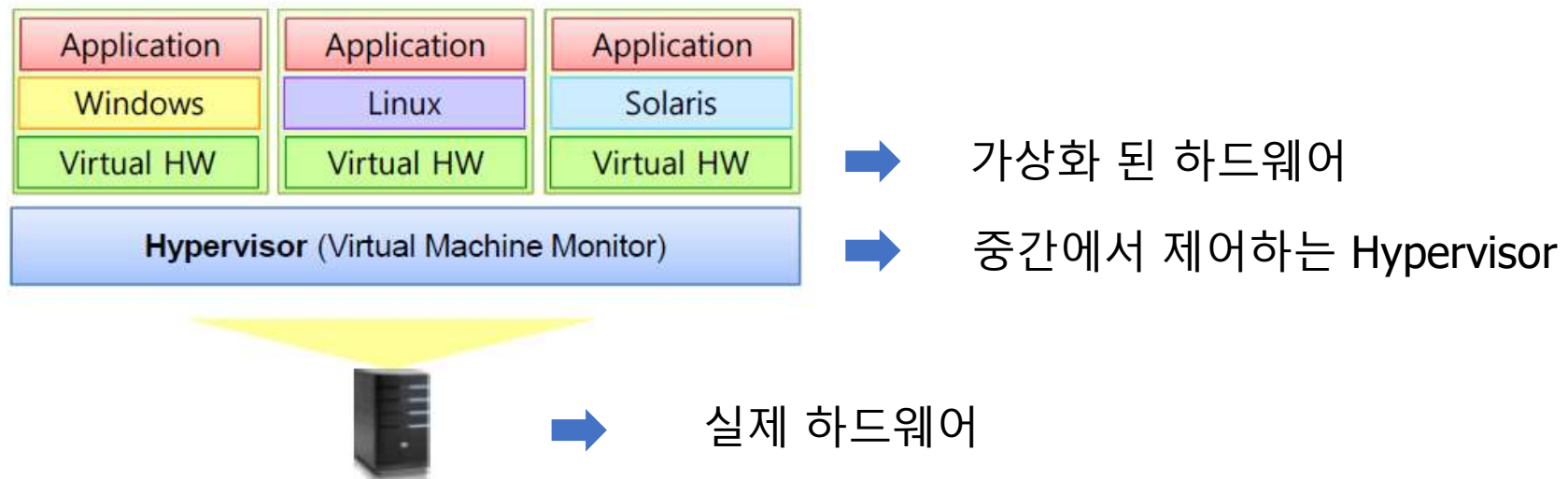
- VM은 하드웨어까지 가상화를 합니다.
- 각 VM은 자신이 해당 머신(서버 컴퓨터)에 있는 자원을 전부 독점하여 쓰고 있다고 착각을 하게 됩니다.
- 또한 각 VM은 서로 간섭할 수 없습니다. (isolation)



Isolation 덕분에 하나의 VM에서 오류가 발생하면 발생한 오류는 다른 VM에 영향을 끼치지 않습니다.

Virtual Machine (VM)

- 머신 안에 모든 VM은 Hypervisor(VMM)에 의해 관리됩니다.
- Hypervisor는 VM들이 사용하는 자원에 대한 관리나 VM간의 통신, 또는 VM들이 시스템에 접근하는 것에 대해 관리하게 됩니다.

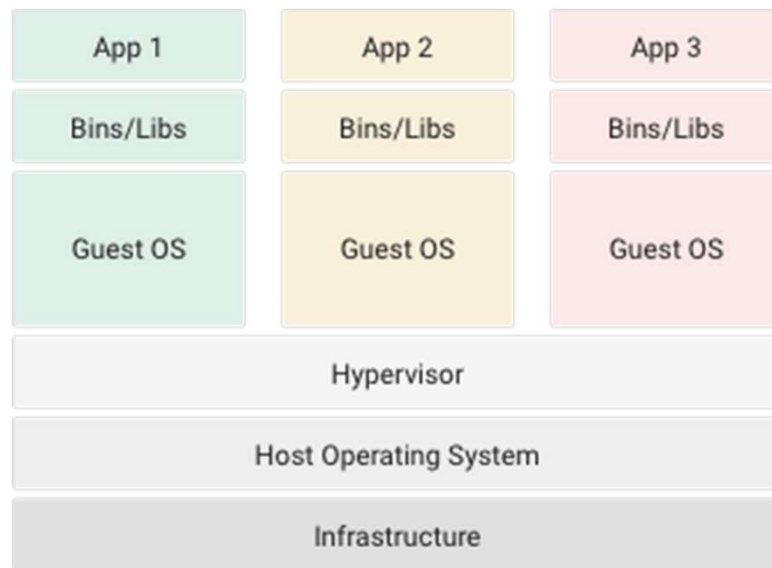


Virtual Machine (VM)

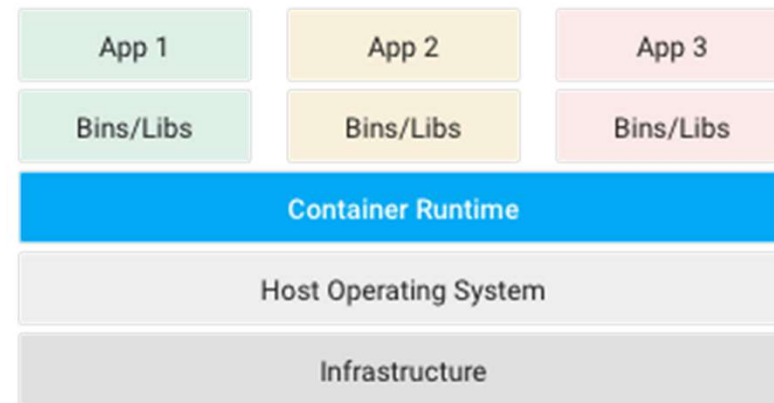
- VM은 Hypervisor를 이용해 각 VM들끼리 간섭을 통제한다는 장점이 있지만 그만큼 단점도 존재합니다.
- 하드웨어까지 가상화를 해야 하기 때문에 컴퓨터의 자원을 많이 사용합니다.
- 또한 이러한 무거운 특성 때문에 실행 시간도 오래 걸립니다.
- 따라서 하나의 응용 프로그램 사용을 위해 VM을 사용하는 것은 효율적이지 않습니다.

Container

- OS level 가상화라고도 합니다. (앞서 설명한 것은 Hardware level 가상화)



Virtual Machines



Containers

Container

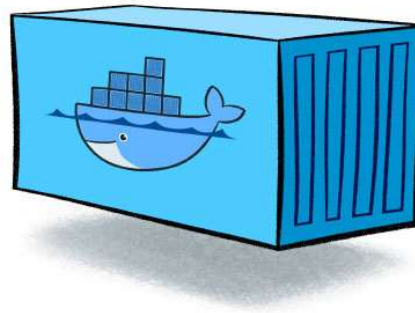
- VM과는 다르게 개별적인 OS 커널이 없고 Hypervisor 대신 container를 실행해주는 프로그램이 존재합니다.
- 단, container에 따라 최소화 된 OS 커널이 존재할 수 있습니다.
- Container 생성 및 관리에 주로 Docker 프로그램을 사용 합니다.
- Container 역시 서로 간섭할 수 없도록 isolation 되어 있습니다.
- 또한 Hardware까지 가상화 하는 것에 비해 매우 가볍습니다.

Docker

- Container를 이용해 각종 응용 프로그램을 빠르게 구축, 테스트, 배포할 수 있도록 도와주는 SW 플랫폼 입니다.
- Linux 환경에서 Container를 생성하고 사용할 수 있도록 도와줍니다.



응용 프로그램



도커 컨테이너



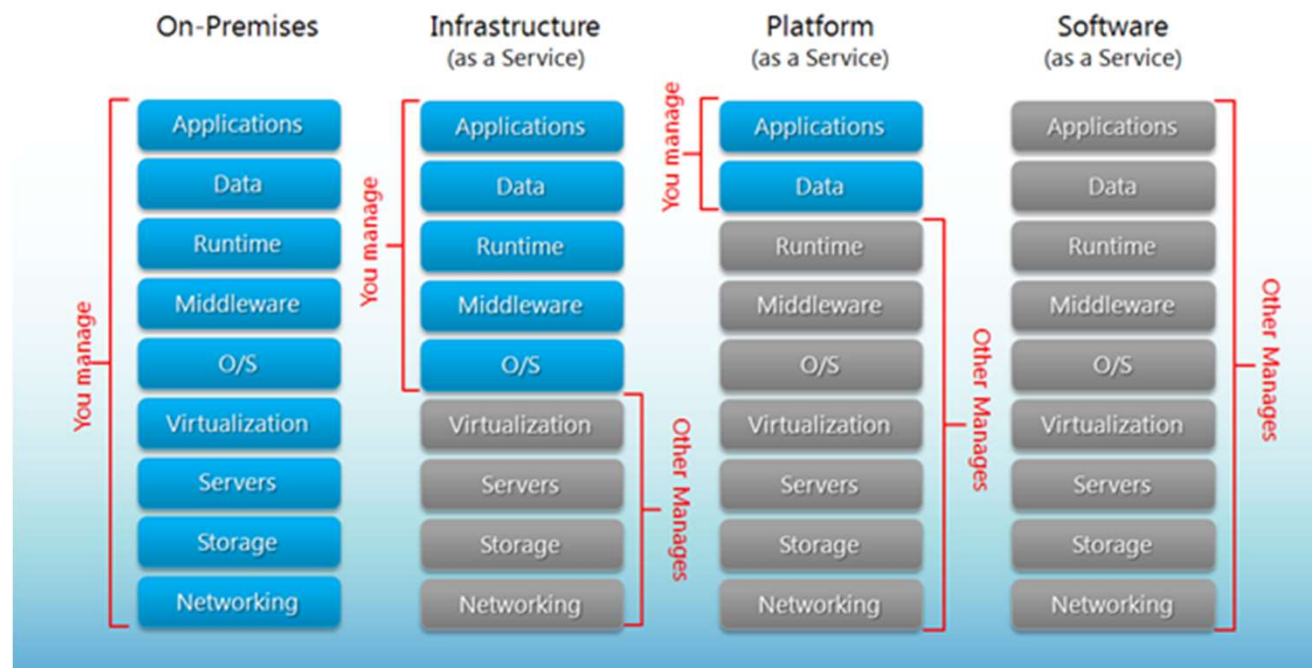
서버

Kubernetes

- Docker container를 타겟으로 하는 Container Orchestration 소프트웨어입니다.
- 즉, Docker container의 확장, 배포, 관리, 네트워킹을 쉽게 관리할 수 있도록 도와주는 SW입니다.
- Kubernetes의 경우, container에 문제가 발생했을 때, 이를 감지하고 빠르게 복구할 수 있도록 도와주는 자동 복구를 탑재하고 있습니다.
- 특히 클러스터에서 다수의 Docker container를 사용할 때 Kubernetes를 많이 사용합니다.
- <https://kubernetes.io/ko/docs/concepts/overview/what-is-kubernetes/>

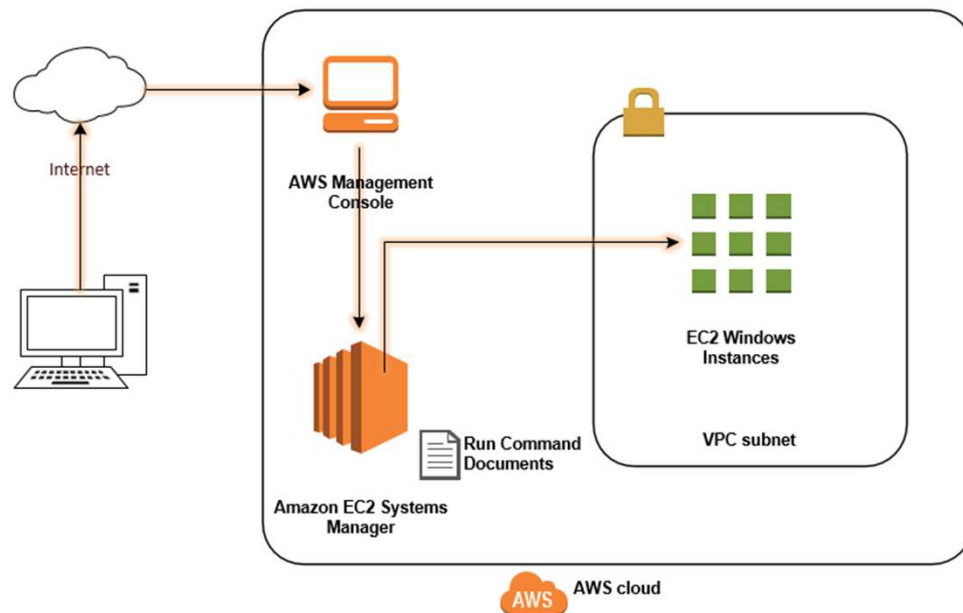
PaaS, IaaS, SaaS

- 서비스를 제공하는 측에서 사용자에게 얼마 만큼의 서비스를 제공하는가에 따라 결정 됩니다.
- 주로 클라우드 서비스에서 많이 사용 되는 용어 입니다.



AWS EC2

- IaaS의 대표적인 예시입니다.
- 사용자는 AWS로부터 “인스턴스”라고 불리는 가상의 서버를 할당 받아서 그 위에 사용자가 원하는 응용 프로그램을 실행할 수 있도록 합니다.
- 실행 중인 인스턴스의 사용량이나 실행 시간에 대해 비용을 지불해야 합니다.



EC2

Thank you

