

# Cloud & Container

유현진

통합과정 4학기,

데이터컴퓨팅 연구실(S4-1 311),

컴퓨터과학과

hyeonjin-yu@chungbuk.ac.kr

# 1. 클라우드(Cloud)란

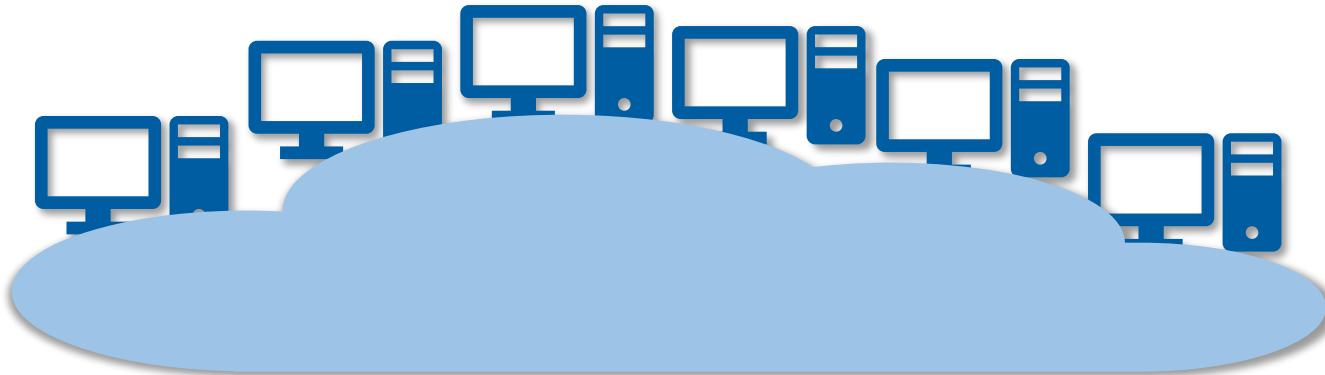
# 1. 클라우드(Cloud)란

## ❖ 1) 클라우드란 무엇인가...

Cloud => 구름에 가려지면 잘 보이지 않음

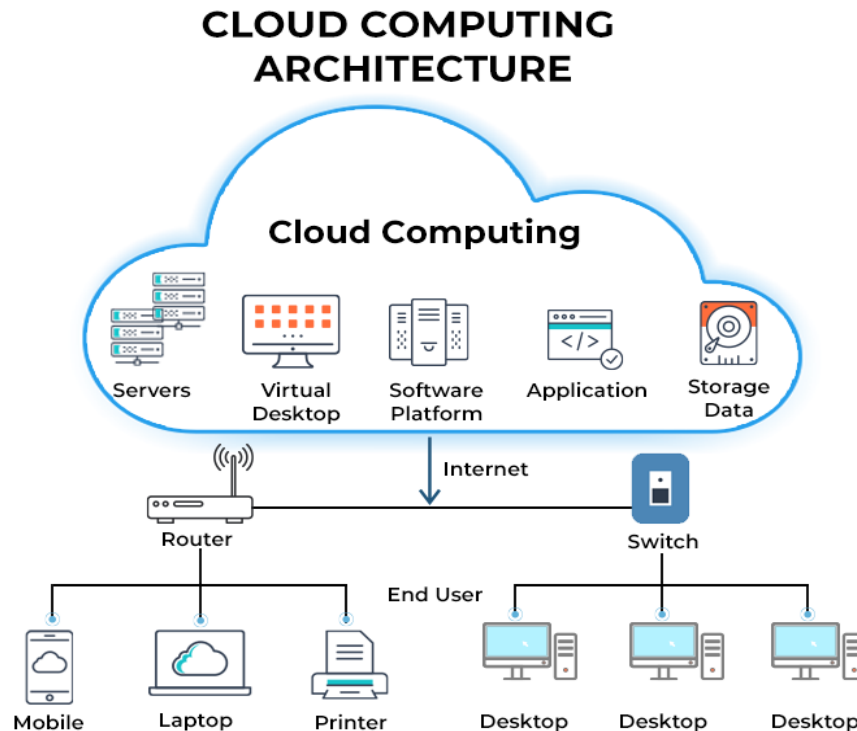


" Computer에 적용 "



# 1. 클라우드(Cloud)란

❖ "사용자의 요구에 맞게 on-demand로 컴퓨팅 자원을 제공하는 서비스"



클라우드 컴퓨팅을 정의하기  
위한 핵심 개념

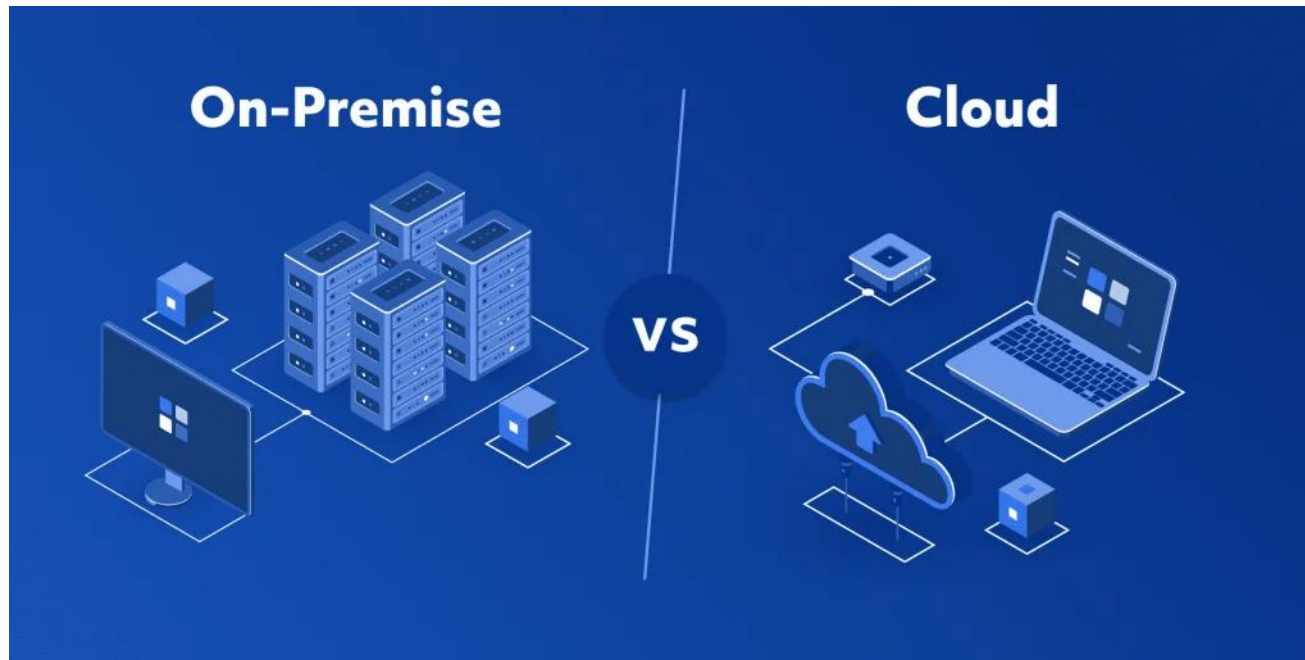
- on-demeand?
- 컴퓨팅 자원?

<https://www.spiceworks.com/tech/cloud/articles/what-is-cloud-computing/>

# 1. 클라우드(Cloud)란

## ❖ 2) On-demand

- 사용자의 요청을 시간/장소의 제약없이 즉각적으로 반응하는 것
  - 클라우드 서비스에서는 사용자 요청을 즉각적으로 대응하는 것이 중요



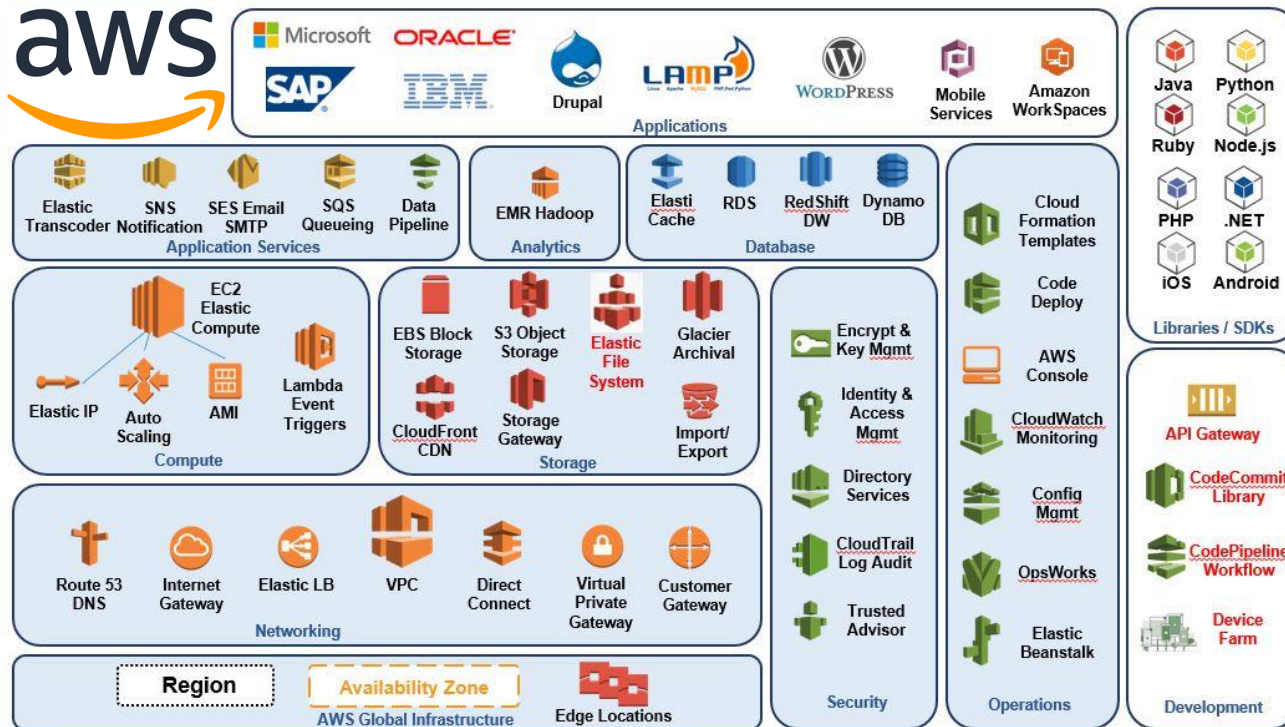
<https://softteco.com/blog/on-premise-vs-cloud>

\* On-premise: IT 시스템과 소프트웨어를 자체적인 물리적인 공간에 직접 설치하고 운영하는 방법

# 1. 클라우드(Cloud)란

## ❖ 2) On-demand 예시 - AWS

- 사용자가 요청한 인프라, 서비스를 별도의 설치과정 없이 바로 사용 가능하도록 제공
- 네트워크만 연결되어 있다면, 제공되는 환경을 시간/장소의 제약없이 사용 가능



<https://hamait.tistory.com/675>

# 1. 클라우드(Cloud)란

## ❖ 3) 컴퓨팅 자원(Computing Resource)

- **컴퓨팅** : 컴퓨터를 이용하여 어떠한 연산을 하는 행위
- **자원** : 컴퓨팅에 필요한 요소로, CPU / Memory / Storage / Network 등이 해당



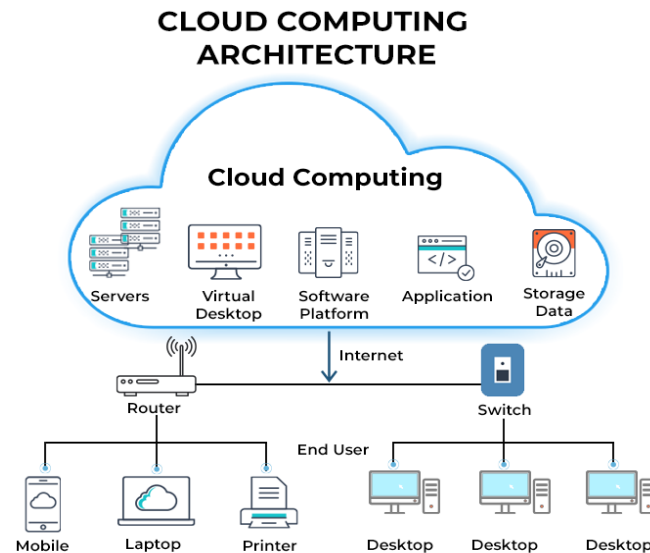


# 1. 클라우드(Cloud)란

"사용자의 요구에 맞게 on-demand로 컴퓨팅 자원을 제공하는 서비스"



사용자 요구에 즉각적으로 반응하여, 원하는 연산을 수행할 수 있도록 그 요구에 맞는 CPU, Memory, Storage를 제공하는 서비스



<https://www.spiceworks.com/tech/cloud/articles/what-is-cloud-computing/>

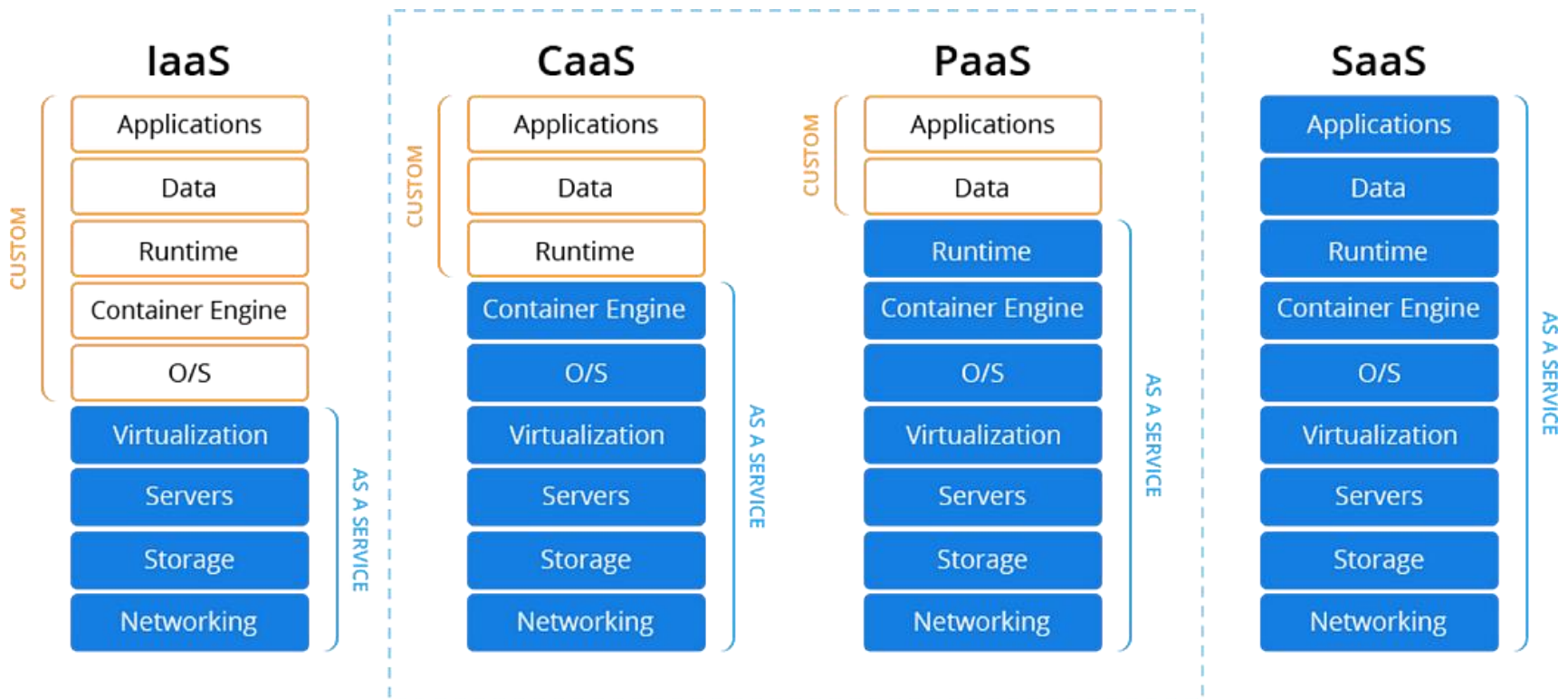


## 2. 클라우드 서비스 모델 종류

## 2. 클라우드 서비스 모델 종류

### ❖ IaaS vs. PaaS vs. SaaS vs. CaaS

- 클라우드 서비스로 어디까지 제공하는지에 따라 분류

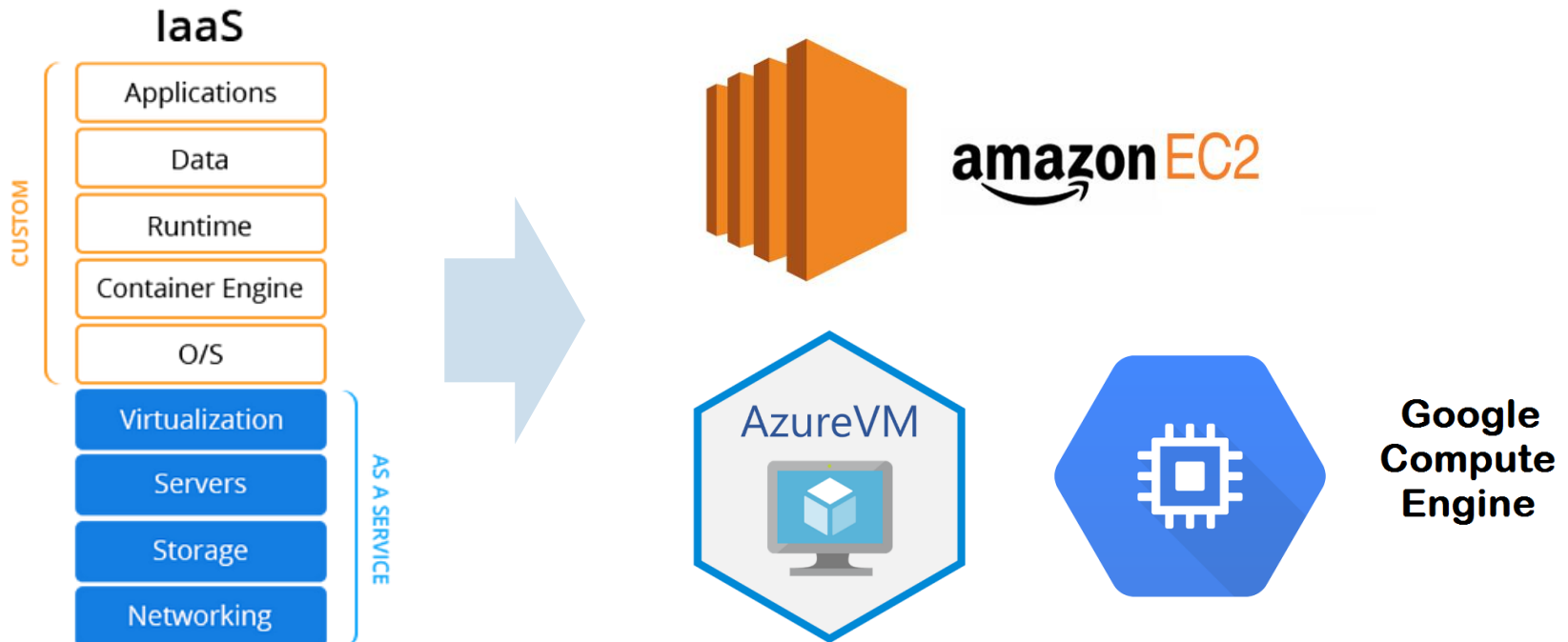


<https://medium.com/@michaelmoopenn/paas-vs-caas-another-insight-864b0703f5bc>

## 2. 클라우드 서비스 모델 종류

### ❖ 1) IaaS = Infrastructure as a Service

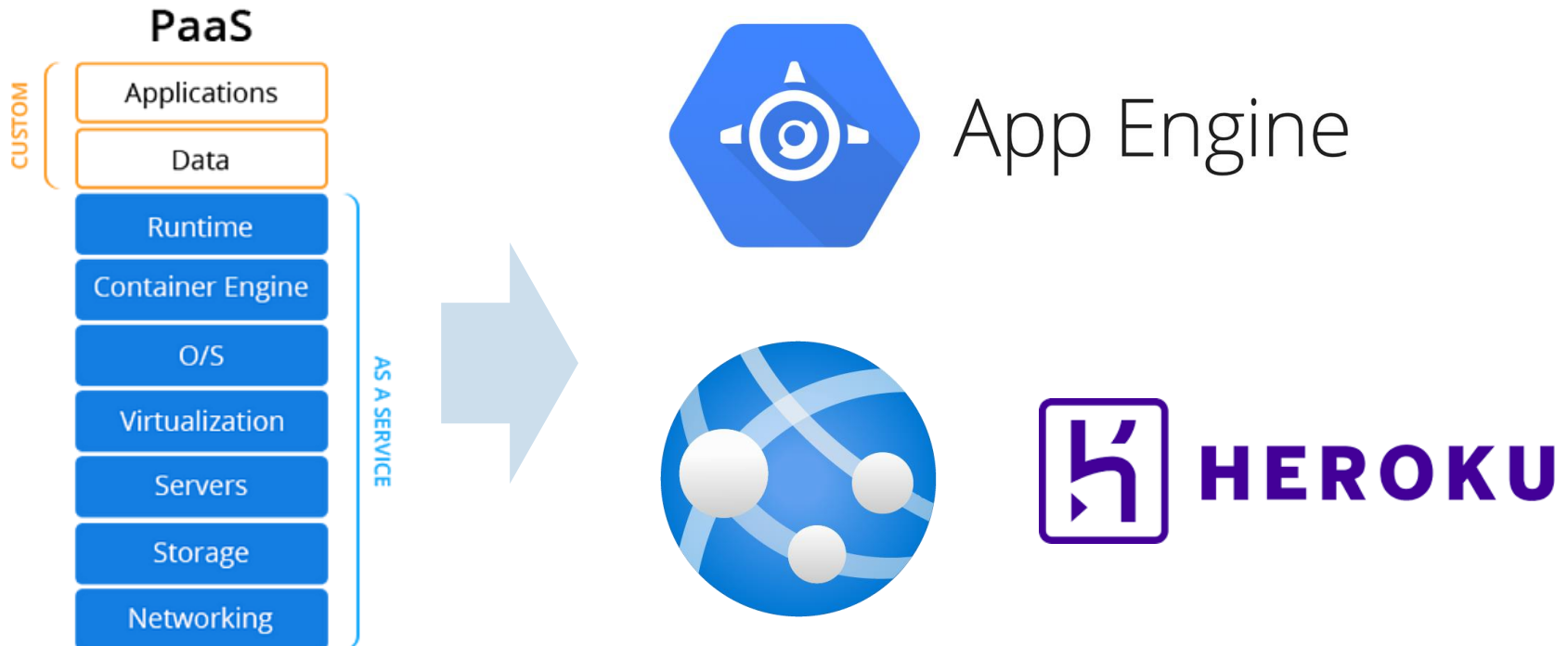
- 인프라를 서비스 형태로 제공하는 모델
  - Ex. 네트워크로 접속할 수 있는 깡통 서버
    - 사용자가 직접 원하는 대로 커스텀



## 2. 클라우드 서비스 모델 종류

### ❖ 2) PaaS = Platform as a Service

- 플랫폼을 서비스 형태로 제공하는 모델
  - Ex. 어플리케이션을 구동하기 위한 사전 준비는 되어 있는 서버
    - 사용자는 원하는 어플리케이션을 실행만 하면 됨



## 2. 클라우드 서비스 모델 종류

### ❖ 3) SaaS = Software as a Service

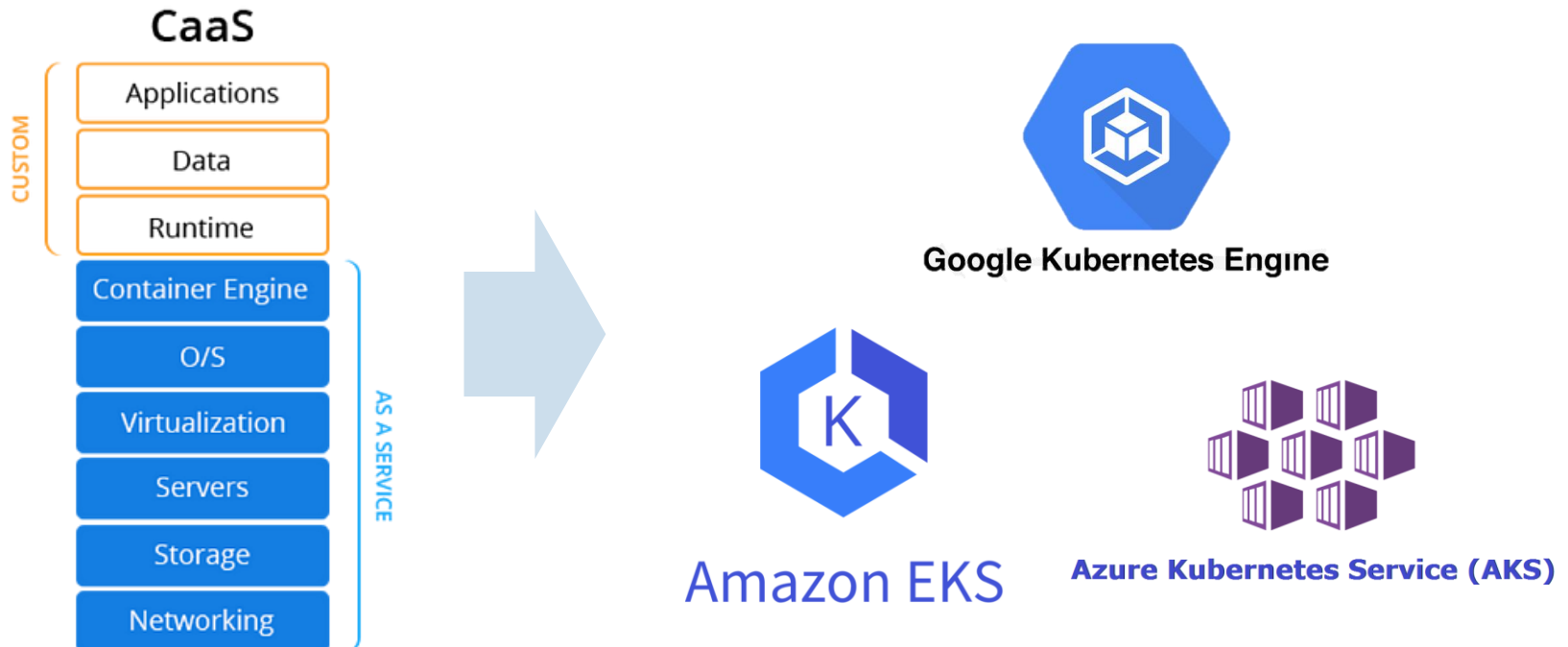
- 소프트웨어를 서비스 형태로 제공하는 모델
  - Ex. 네트워크 상에서 접속하여 이용하는 소프트웨어
    - 보통 달마다 구독료를 내는 형태



## 2. 클라우드 서비스 모델 종류

### ❖ 4) CaaS = Containers as a Service

- 컨테이너를 서비스 형태로 제공하는 모델
  - Ex. 네트워크 상에서 제공되는 컨테이너
    - 구동할 어플리케이션에 맞는 런타임 적용이 필요



# 3. 클라우드 서비스 구현 기술

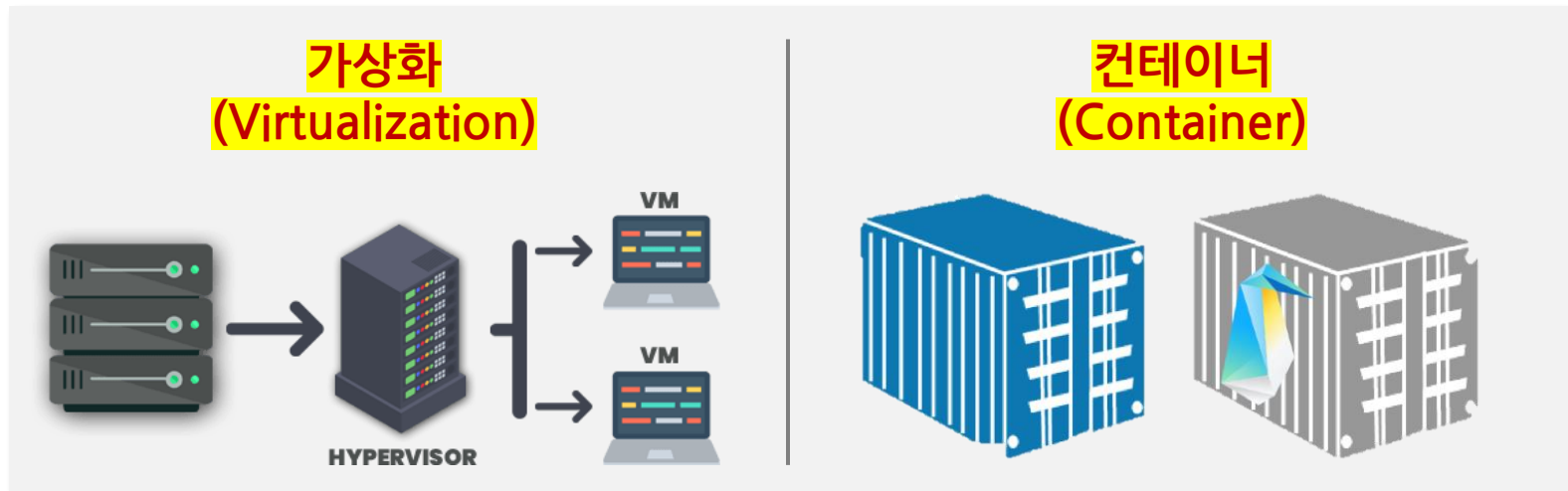


### 3. 클라우드 서비스 구현 기술

“클라우드 서비스를 이용하는 사용자에게 적절한 자원을 제공”

+

“다른 사용자에게 제공된 자원과 충돌이 일어나면 안됨”

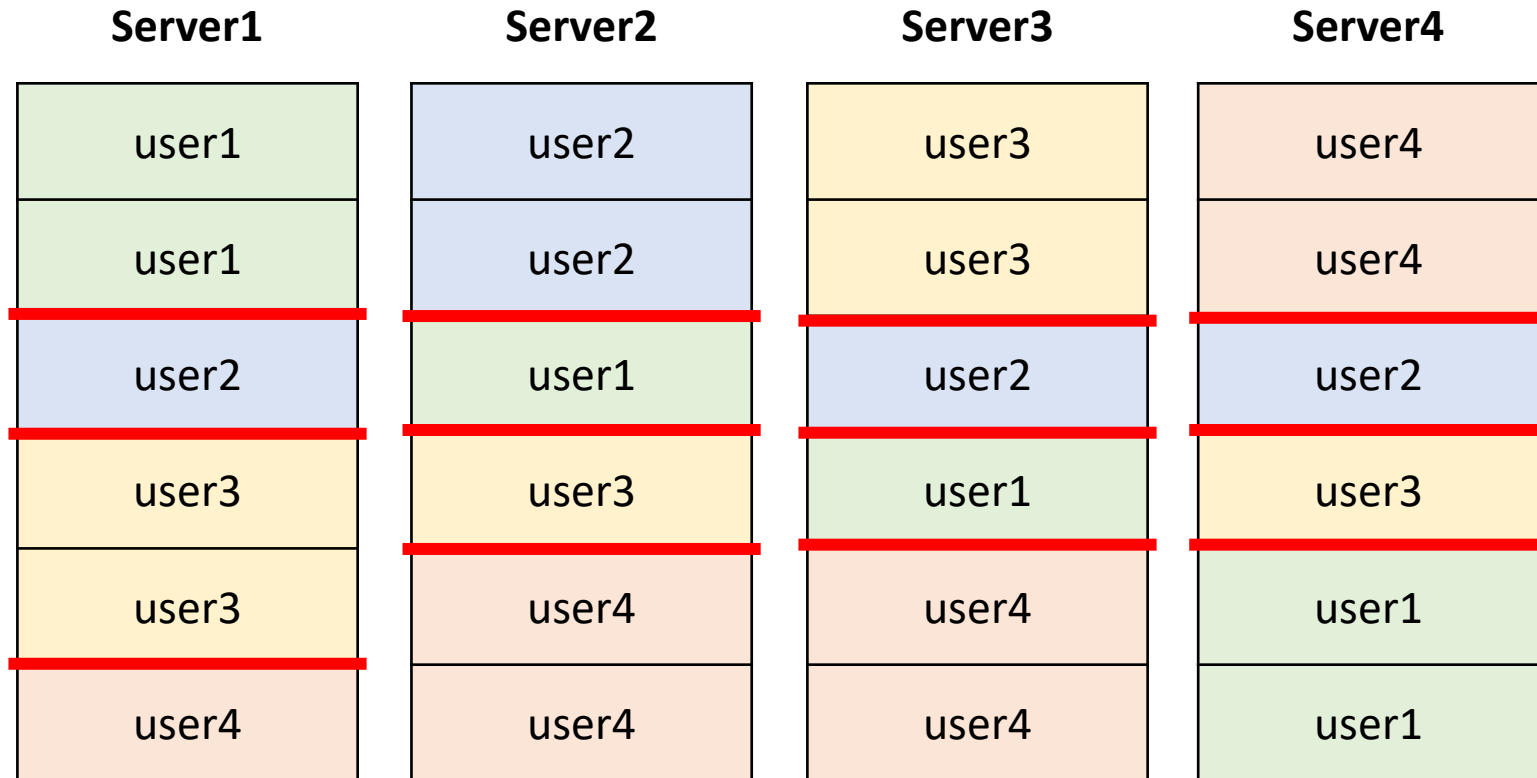


# 3. 클라우드 서비스 구현 기술



## ❖ "서버의 자원을 가상화 & 컨테이너 기술을 통해 제공"

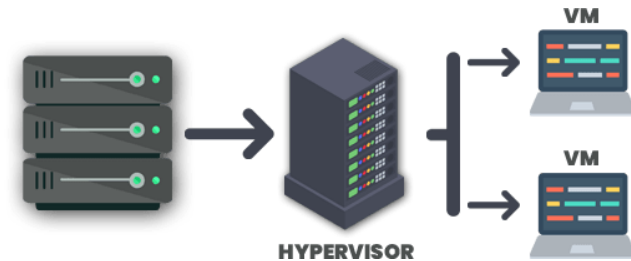
- 각 사용자가 소유한 자원은 격리되어 있기에, 서로 사용하는 자원에 간섭 불가



# 3. 클라우드 서비스 구현 기술

## ❖ 1) 가상화(Virtualization)

- 컴퓨팅 자원을 가상화한 것으로, 가상의 컴퓨터를 생성하는 기술(=가상머신)
  - 호스트와 가상머신 간의 완전한 격리
  - 호스트와 다른 OS 사용 가능



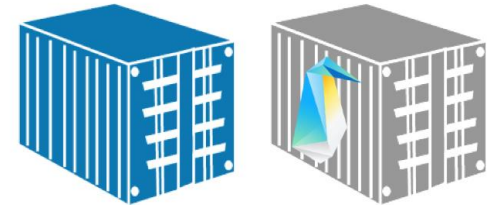
- Ex. VritualBox, WMware, Parallels



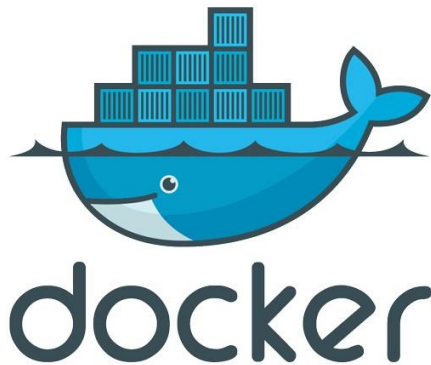
# 3. 클라우드 서비스 구현 방법

## ❖ 2) 컨테이너(Container)

- 다른 프로세스와 분리된 독립적인 공간을 제공하는 기술
  - OS + 하드웨어를 가상화하지 않기에, 굉장히 가벼움
  - 호스트와 다른 계열의 OS 사용 불가



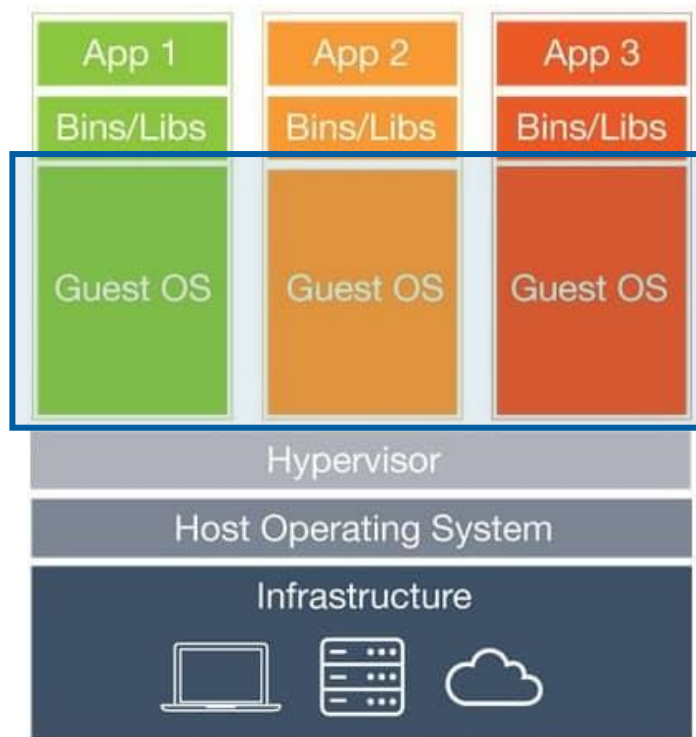
- Ex. Docker, Kubernetes, Singularity



# 3. 클라우드 서비스 구현 방법

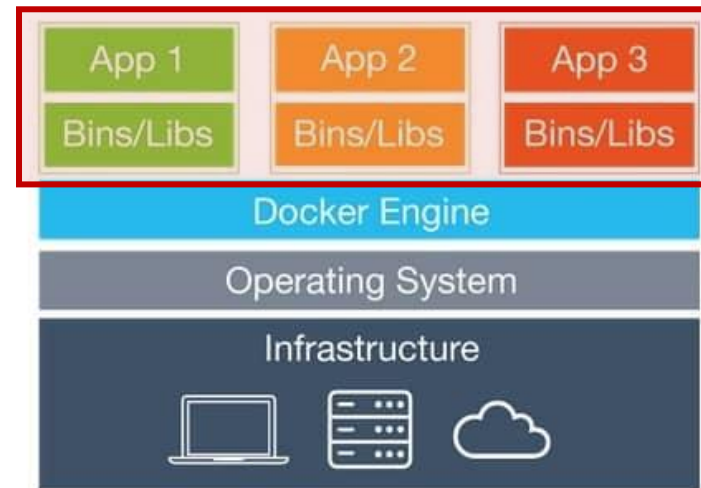
## ❖ 3) 가상화 vs. 컨테이너

<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.justaftermidnight247.com%2Finsights%2Fcontainerisation-cost-saving-and-a-big-leap-forward%2F&psig=AOvVaw0R7lQLmRxGMe8ftlufIAS&ust=1730983602960000&source=images&cd=vfe&opi=89978449&ved=0CBcQjhxqFwoTCPDjv9Dex4kDFQAAAAAAdAAAAABAE>



Virtual Machines

- 호스트와 다른 OS 사용 가능  
(OS 레벨부터 가상화하기 때문)



Containers

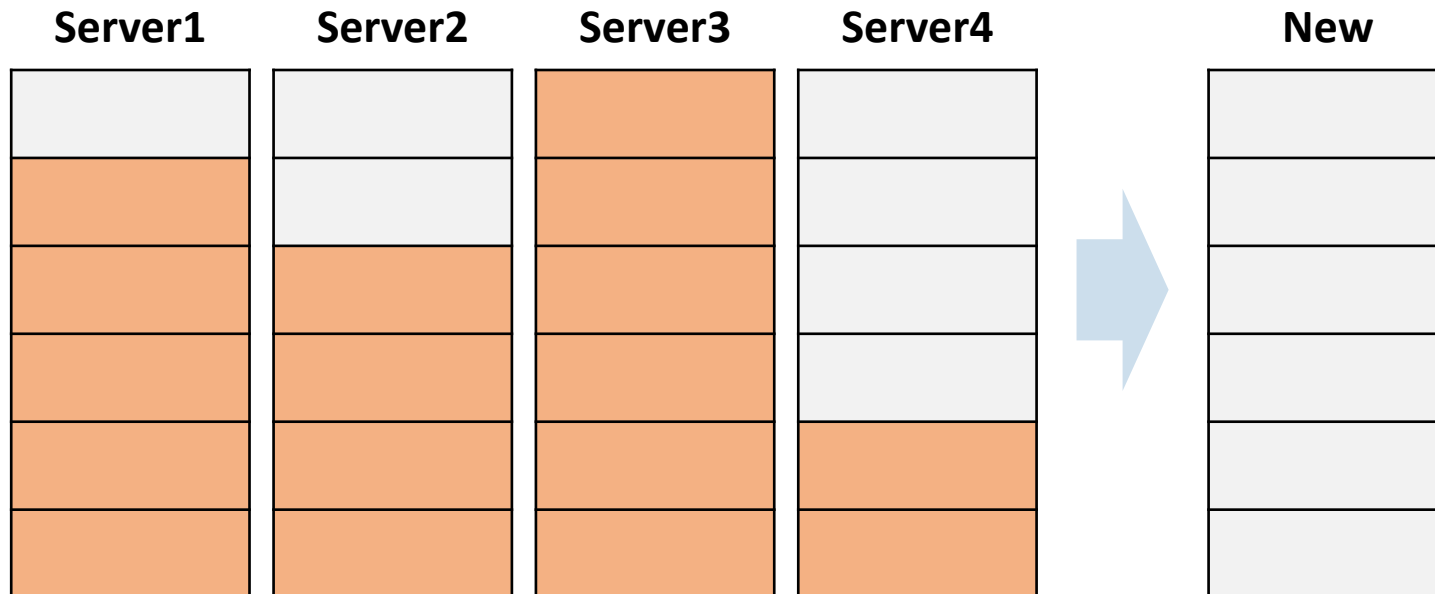
- 호스트와 다른 OS 사용 불가  
(단순히 프로세스 격리 역할)

## 4. 클러스터 컴퓨팅

# 4. 클러스터 컴퓨팅

❖ 컴퓨터는 항상 100% 자원을 사용하지 않음

⇒ 사용되지 않는 자원을 모아 하나의 거대한 자원으로 운영 가능  
(like “티끌 모아 태산”)





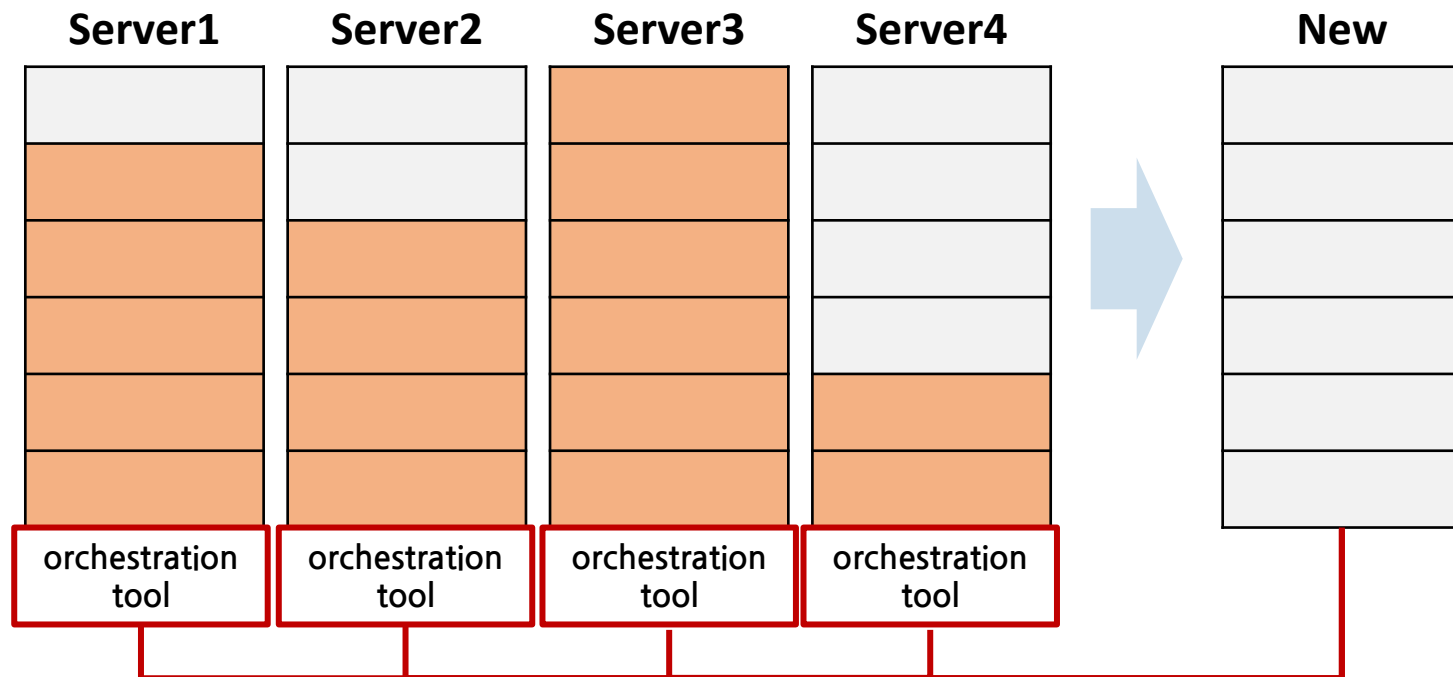
# 4. 클러스터 컴퓨팅

이러한 유휴자원을 모아 활용할 수 있도록 지원하는 기술

= “ 오케스트레이션(Orchestration) ”

- 회색 자원(유휴 자원)을 탐지 및 수집하여 하나의 시스템처럼 운영할 수 있도록 개입

[ 클러스터 컴퓨팅(Cluster Computing) ]



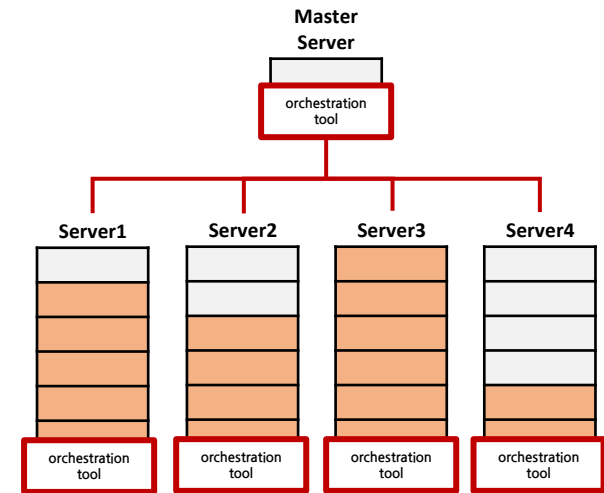
# 5. 클러스터 컴퓨팅 기반 기술

# 5. 클러스터 컴퓨팅 기반 기술

## ❖ 1) 오케스트레이션(Orchestration)

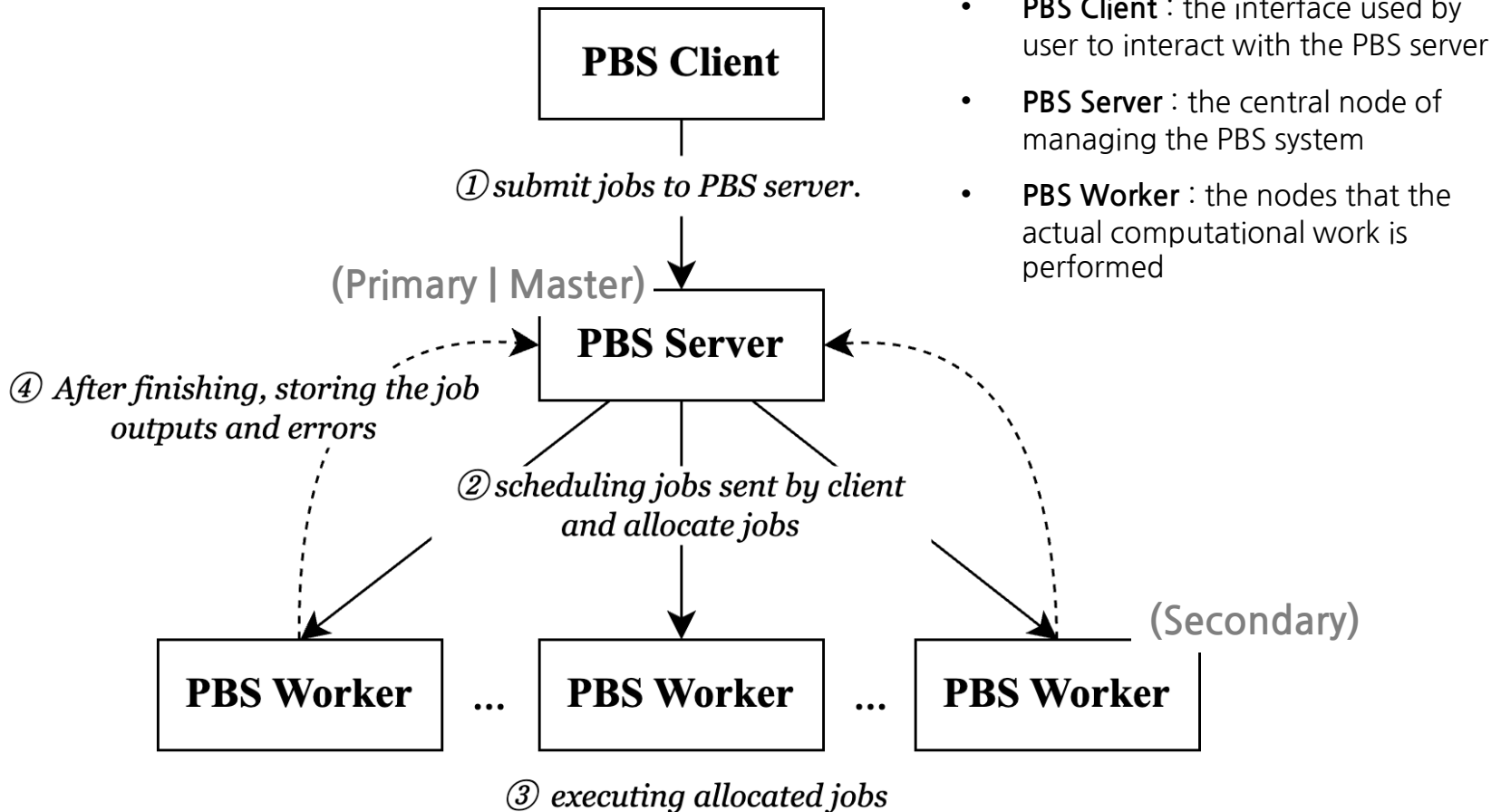
- 수집한 자원의 정보 파악
- 자원을 효율적으로 관리할 수 있도록 스케줄링
- 자원 모니터링

• Ex. HTCondor, Slurm, OpenPBS



# 5. 클러스터 컴퓨팅 기반 기술

## ❖ 2) 대략적인 구조



# 5. 클러스터 컴퓨팅 기반 기술



## ❖ 3) 작업 스크립트 & 클러스터 모니터링

```
[gsdc24a01@alice-kisti-hpc scratch]$ cat check-mount.sh
#!/bin/sh
#PBS -V
#PBS -N check_mount
#PBS -q alice
#PBS -A etc
#PBS -l select=1:ncpus=1:mpiprocs=1:ompthreads=1
#PBS -l walltime=00:10:00
```

```
cd $PBS_O_WORKDIR

module purge
module load craype-mic-knl intel/18.0.3

hostname
ls -l /cvmfs/alice.cern.ch/bin/
echo ===
mount | grep cvmfs
```

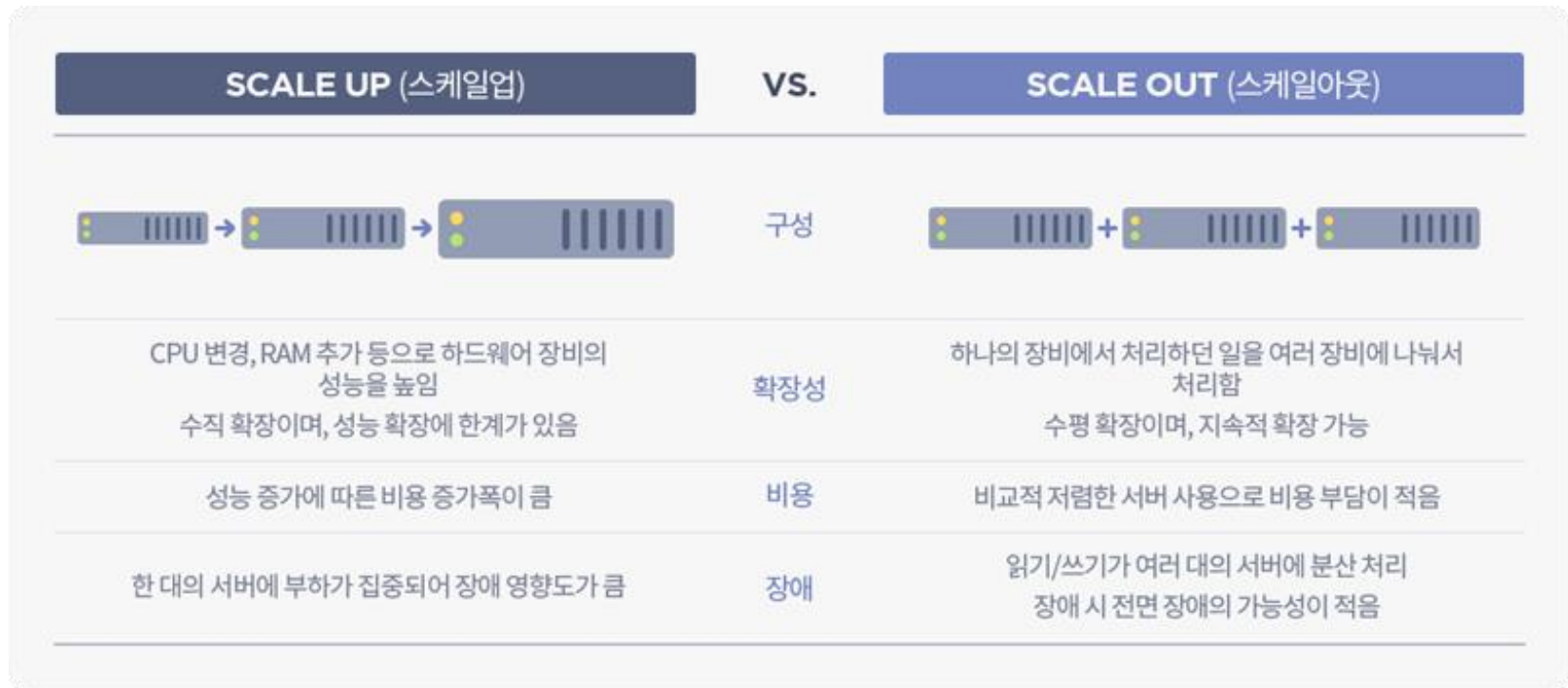
```
[root@alice-kisti-hpc gsdc24a01_backup]# cat ToBeDelete_check_mount.o14286817
node8305
try telnet:
```

```
[gsdc24a01@alice-kisti-hpc scratch]$ pbsnodes -aSj | tail
```

node8296	state-unknown	0	0	0	94gb/94gb	68/68	0/0	0/0	--
node8297	state-unknown	0	0	0	94gb/94gb	68/68	0/0	0/0	--
node8298	state-unknown	0	0	0	94gb/94gb	68/68	0/0	0/0	--
node8299	state-unknown	0	0	0	94gb/94gb	68/68	0/0	0/0	--
node8300	state-unknown	0	0	0	94gb/94gb	68/68	0/0	0/0	--
node8301	free	0	0	0	93gb/93gb	68/68	0/0	0/0	--
node8302	free	0	0	0	93gb/93gb	68/68	0/0	0/0	--
node8303	free	0	0	0	93gb/93gb	68/68	0/0	0/0	--
node8304	state-unknown	0	0	0	94gb/94gb	68/68	0/0	0/0	--
node8305	state-unknown	0	0	0	94gb/94gb	68/68	0/0	0/0	--

# 5. 클러스터 컴퓨팅 기반 기술

## ❖ 4) Scale-Up vs. Scale-Out?



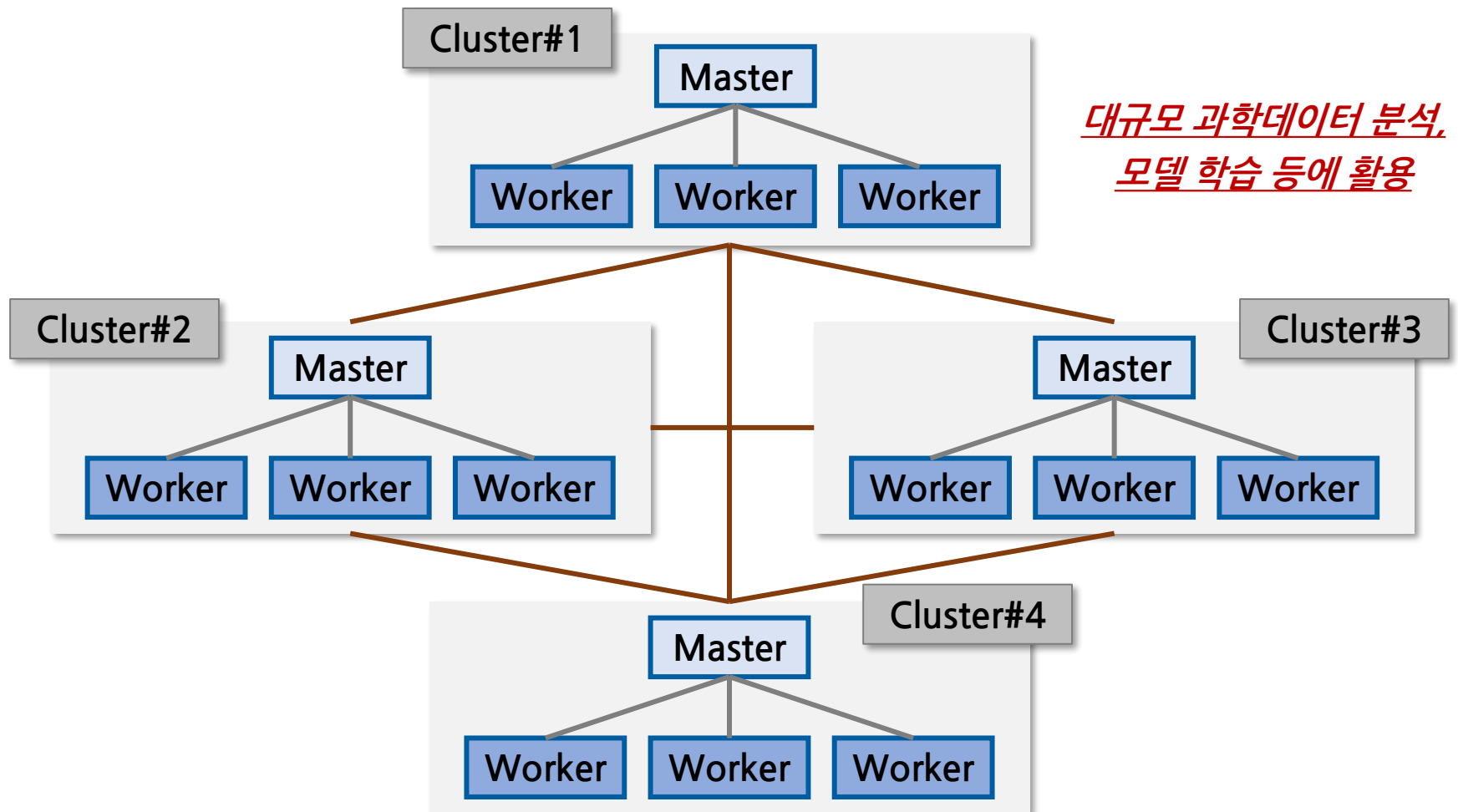
<https://tecoble.techcourse.co.kr/post/2021-10-12-scale-up-scale-out/>

## 6. 그리드 컴퓨팅



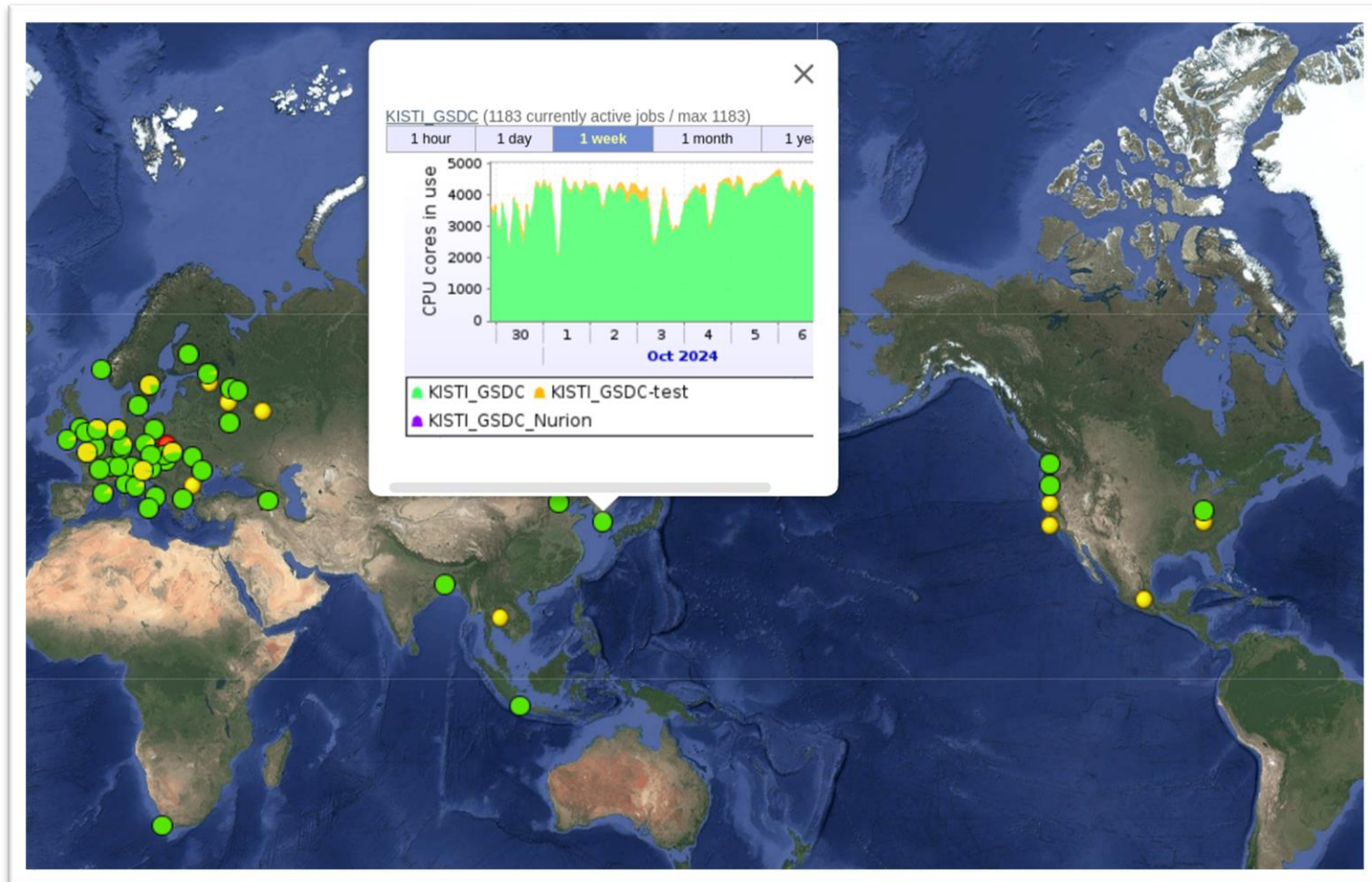
# 6. 그리드 컴퓨팅

“클러스터 컴퓨팅 환경이 모여 구성된 하나의 거대한 컴퓨팅 환경”



# 6. 그리드 컴퓨팅

❖ Ex. CERN Grid Site - WLCG (*Worldwide LHC Computing Grid*)

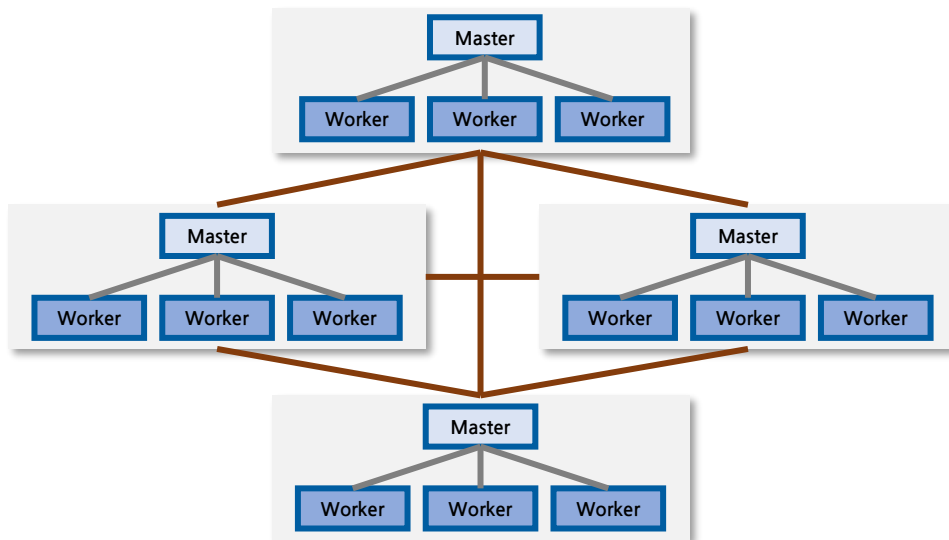


# 6. 그리드 컴퓨팅

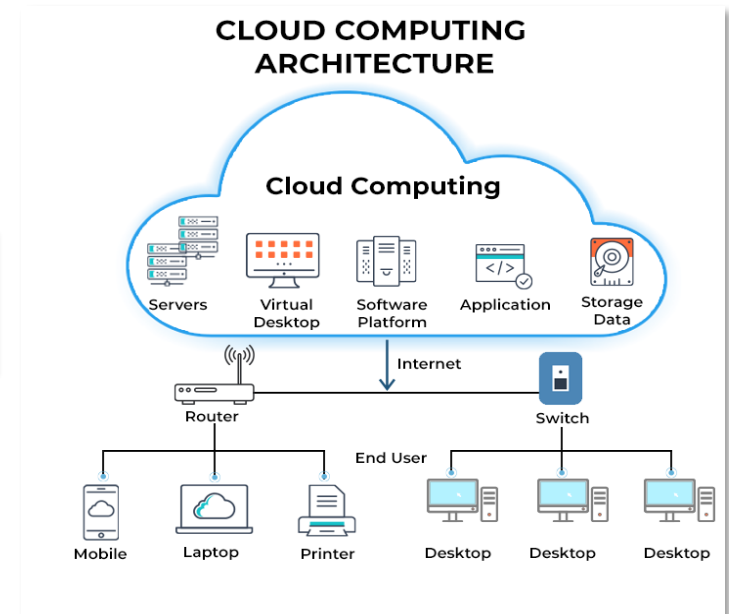
“자원의 활용성을 증진시키기 위한 컴퓨팅”  
= High Throughput Computing (HTC)



[ Grid Computing ]



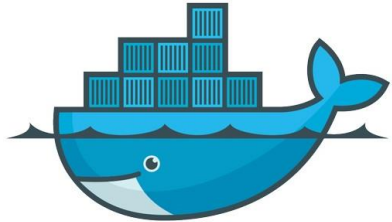

[ Grid Computing ]



## 7. 컨테이너 플랫폼

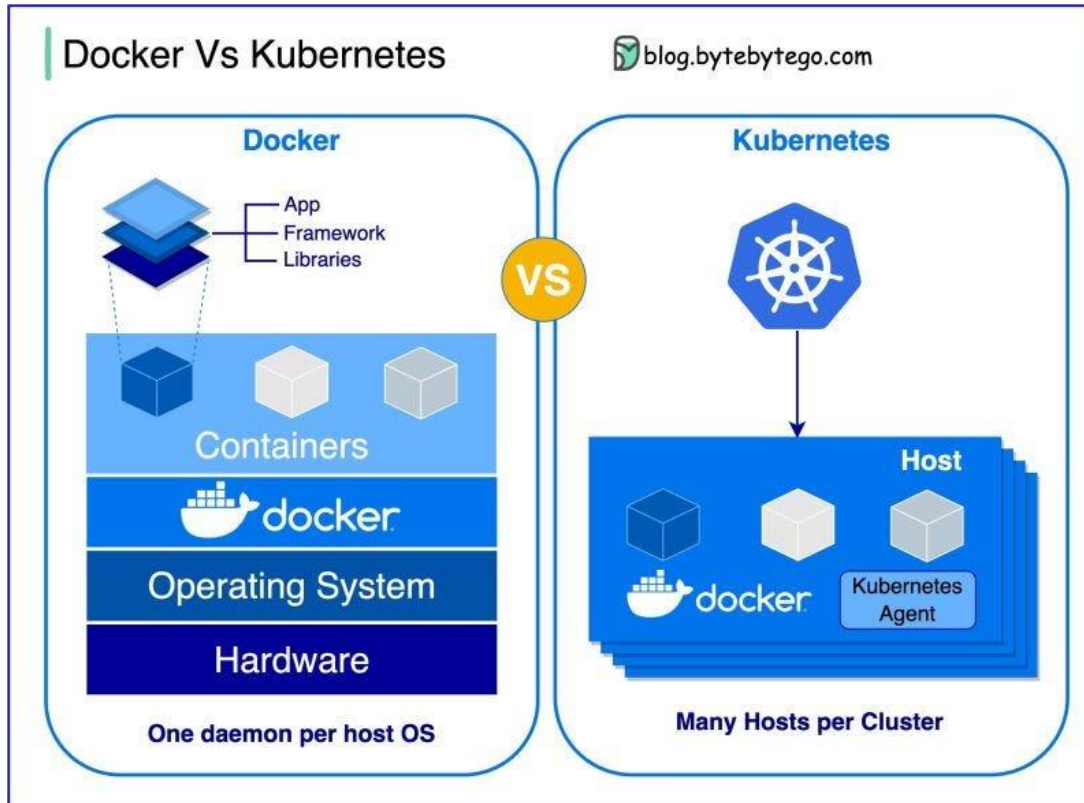
# 7. 컨테이너 플랫폼

❖ 대중적으로 사용되는 컨테이너 플랫폼 = Docker, Kubernetes

Docker	Kubernetes
 docker	 kubernetes
<ul style="list-style-type: none"><li>• 애플리케이션을 컨테이너로 패키징</li><li>• 단일 컨테이너의 배포와 관리에 최적화</li><li>• 단일 컨테이너 관리에 초점 (생성, 실행, 중지)</li><li>• Kubernetes의 컨테이너 런타임으로 사용</li></ul>	<ul style="list-style-type: none"><li>• 대규모 컨테이너 기반 애플리케이션을 오케스트레이션</li><li>• 다수의 컨테이너가 필요한 복잡한 애플리케이션에 적합</li><li>• 클러스터 내의 다수의 컨테이너와 노드를 관리</li><li>• 컨테이너의 자동 배포, 로드 밸런싱, 스케일링, 장애 복구 등을 수행</li></ul>

# 7. 컨테이너 플랫폼

## ❖ Docker 와 Kubernetes 의 차이점



<https://www.google.com/url?sa=i&url=https%3A%2F%2Fsystemweakness.com%2Fdocker-compose-vs-kubernetes-f04acf47abc4&psig=AOvVaw3OtRk7ikVn5qwqJBMF9pn&ust=1730995490551000&source=images&cd=vfe&opi=89978449&ved=0CBQQjRxqFwoTCijLoPGKyikDFQAAAAAABBBK>

### • Docker

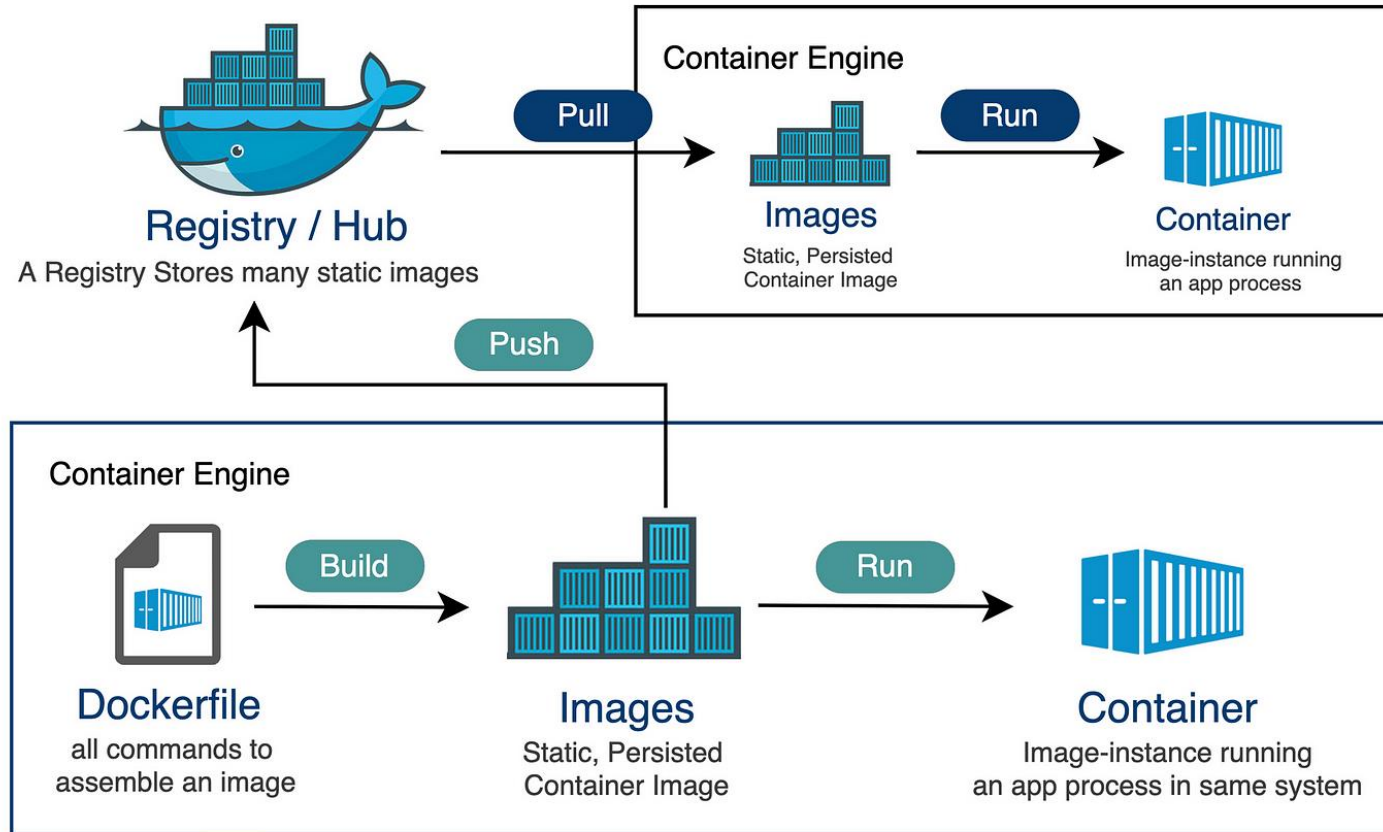
- 호스트마다 Docker Engine 설치
- 어플리케이션 컨테이너화 도구

### • Kubernetes

- 클러스터 내 호스트별로 필요한 데몬만 설치
- 클러스터 단위로 컨테이너 배포 및 관리 가능

# 7. 컨테이너 플랫폼

## ❖ 1) Docker 구조

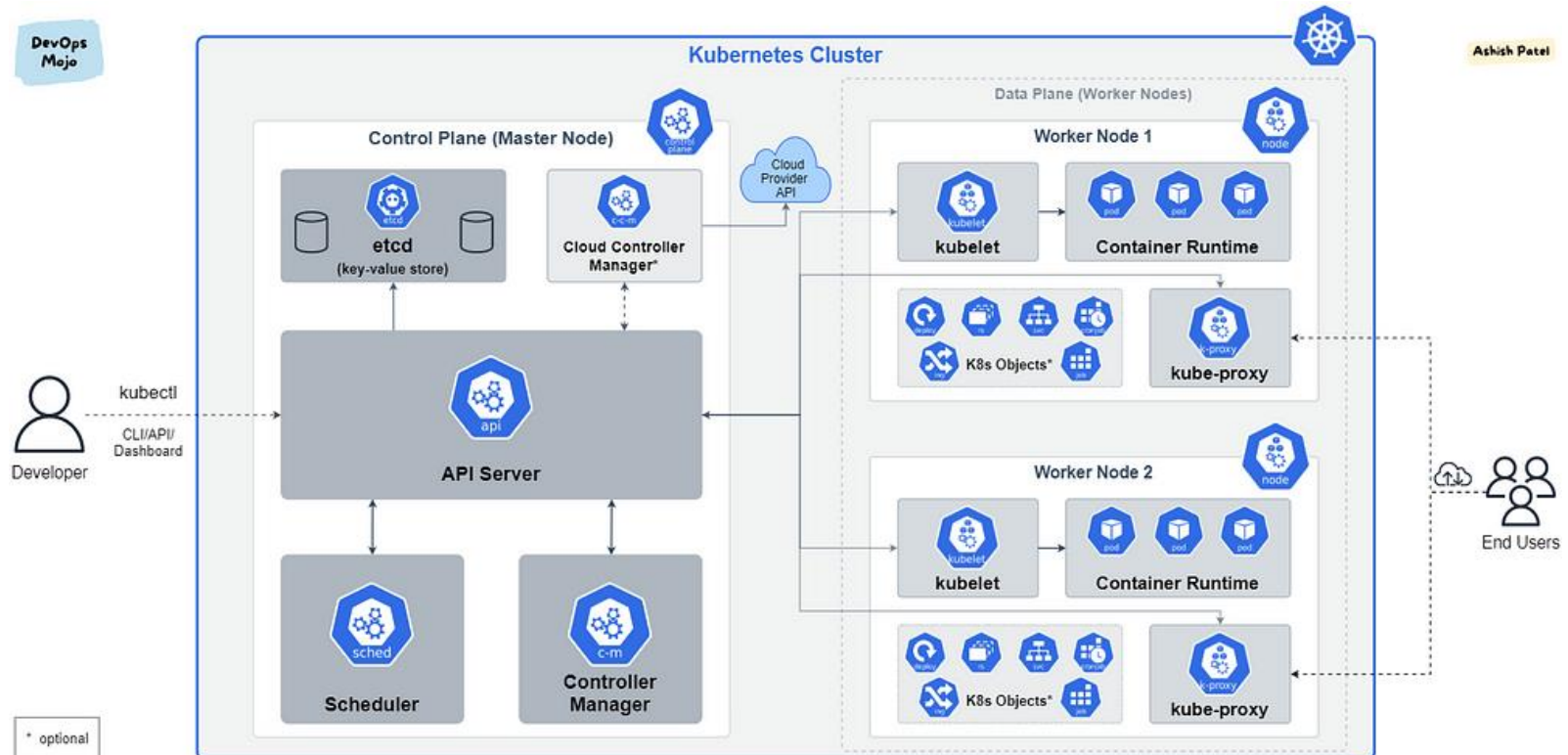


<https://www.google.com/url?sa=i&url=https%3A%2F%2F8grams.medium.com%2Fhow-to-install-container-registry-on-kubernetes-cluster-af1f486ab22a&psig=AOvVaw30tRk7ikVVn5qwqJBMF9pn&ust=1730995490551000&source=images&cd=vfe&opi=89978449&ved=0CBQQjRxqFwoTCIjLoPGKyIkDFQAAAAAdAAAAABbc>



# 7. 컨테이너 플랫폼

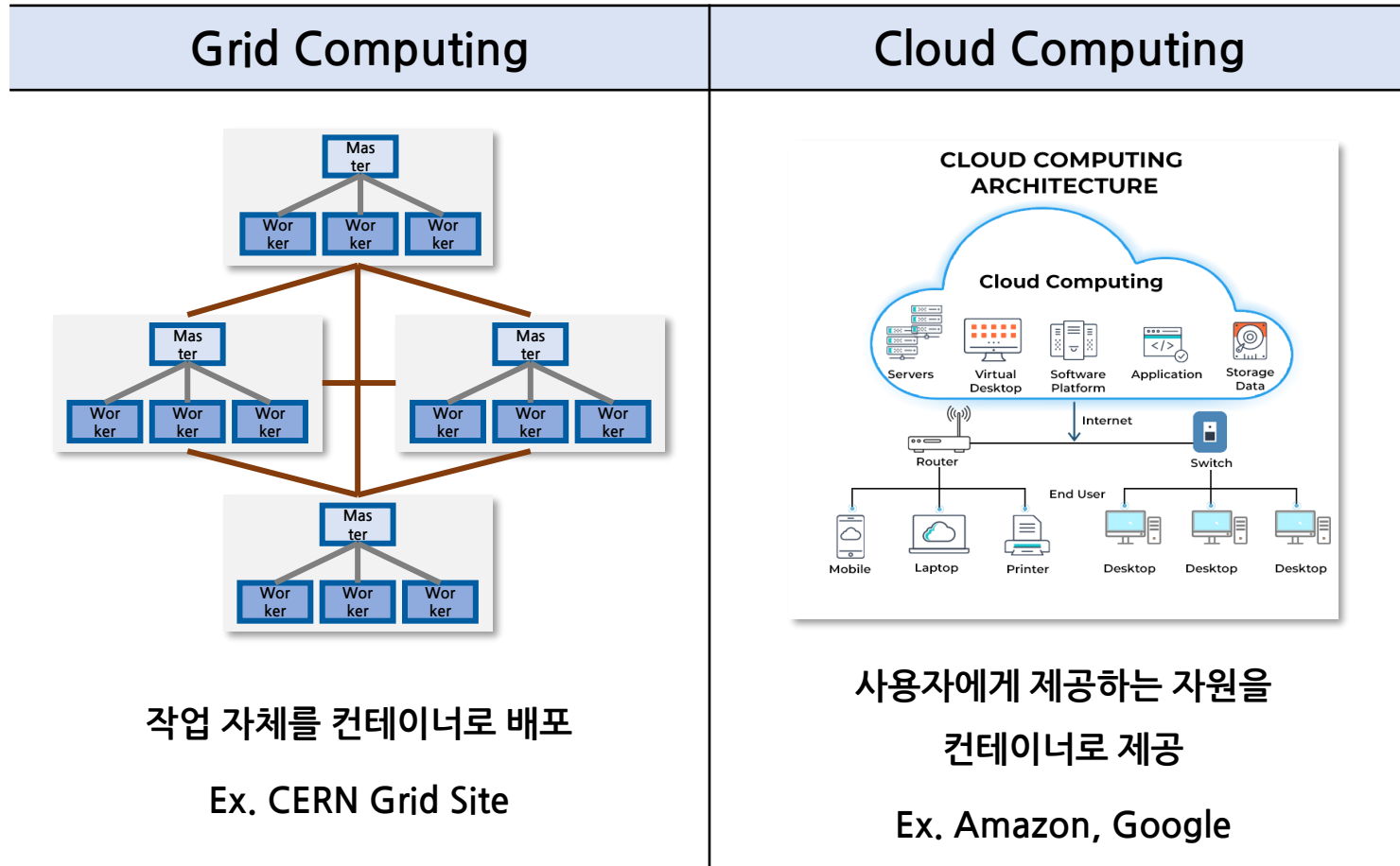
## ❖ 2) Kubernetes 구조



<https://medium.com/devops-mojo/kubernetes-architecture-overview-introduction-to-k8s-architecture-and-understanding-k8s-cluster-components-90e11eb34ccd>

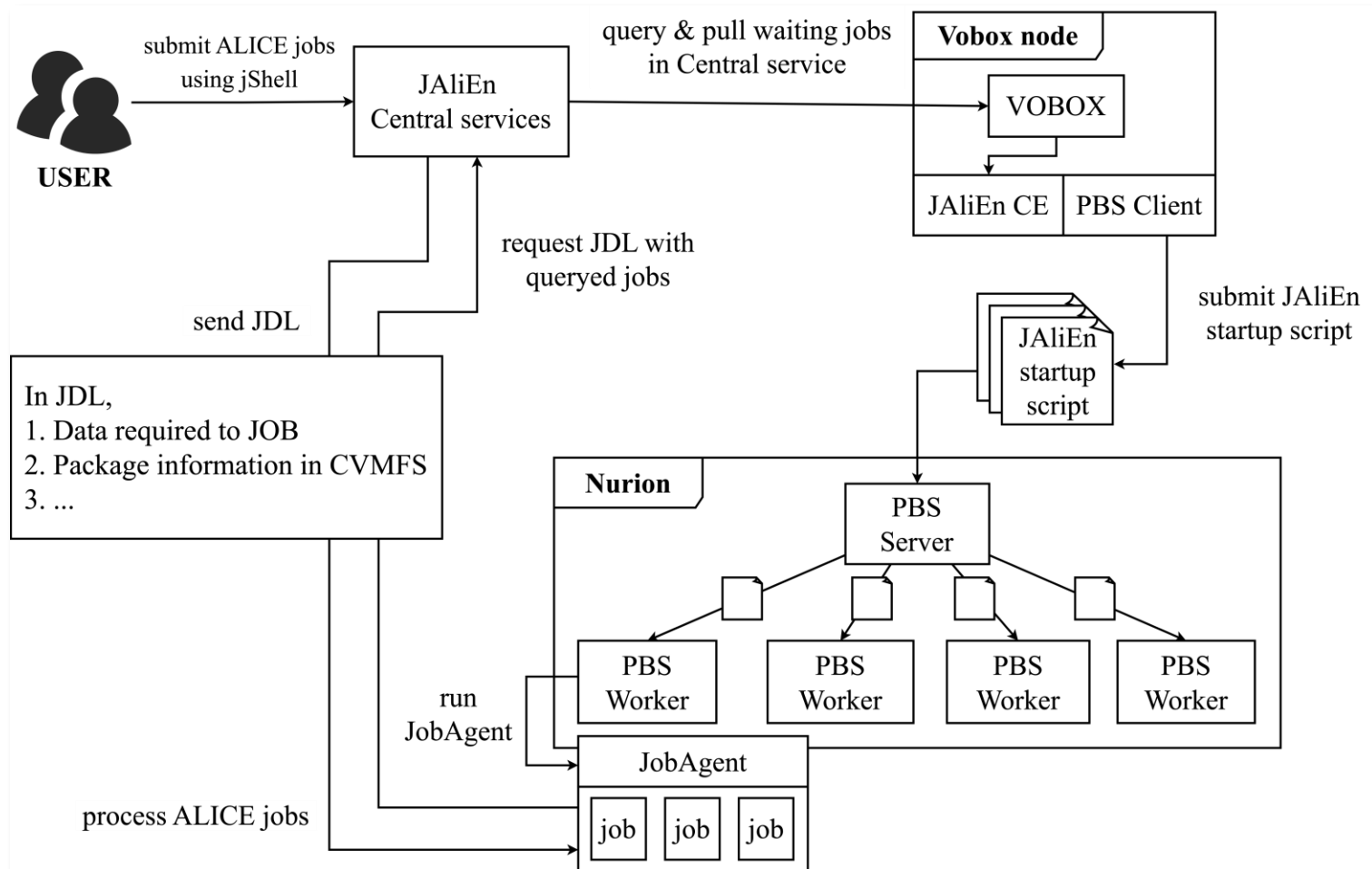
# 7. 컨테이너 플랫폼

## ❖ 컨테이너 플랫폼 활용 예시



# 7. 컨테이너 플랫폼

## ❖ 컨테이너 플랫폼 활용 예시 - Grid Computing

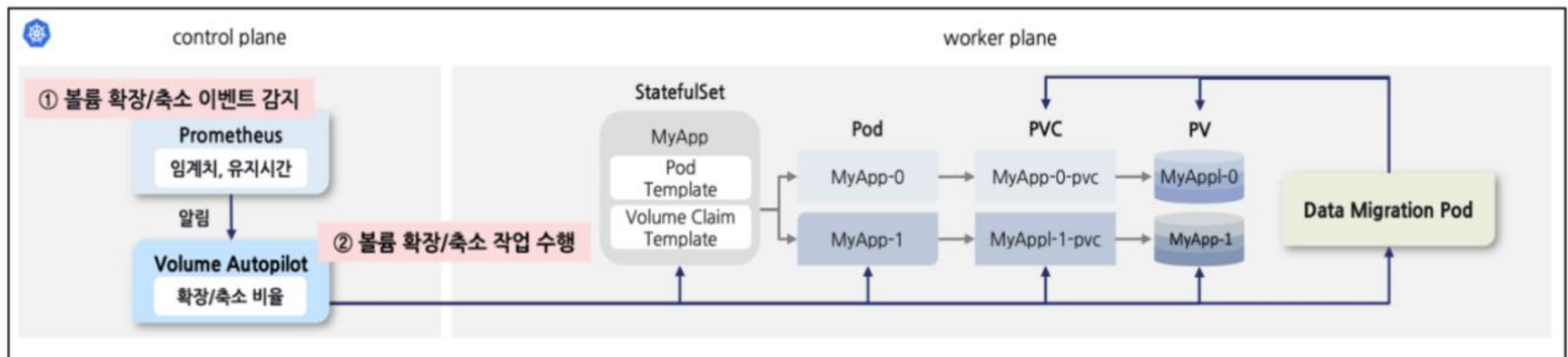
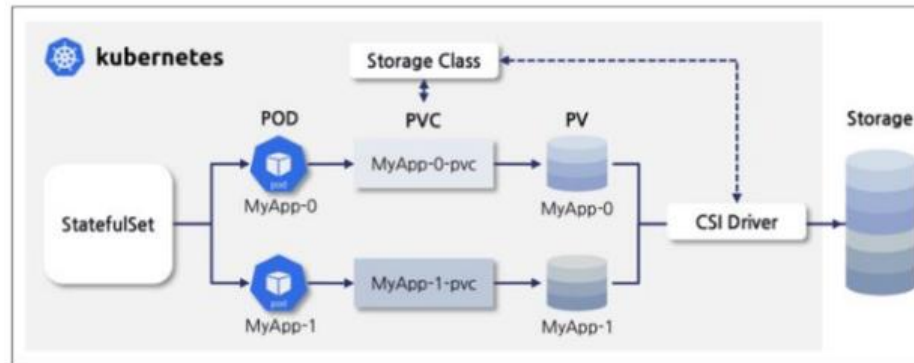


## 8. 클라우드 관련 연구 주제

# 8. 클라우드 관련 연구 주제

## 1) Kubernetes Persistent Volume(PV) 오토스케일러 (기술 개발)

- 컨테이너 서비스에서 발생하는 데이터를 영구 저장하기 위한 볼륨의 자동 확장 및 축소 기능 관련 알고리즘 제안 및 구현



# 8. 클라우드 관련 연구 주제



## 1) Kubernetes Persistent Volume(PV) 오토스케일러 (기술 개발)

- 컨테이너 서비스에서 발생하는 데이터를 영구 저장하기 위한 볼륨의 자동 확장 및 축소 기능 관련 알고리즘 제안 및 구현

## 2) 슈퍼컴퓨터 기반 CERN Grid Site 구축 (인프라 구축)

- 5호기 슈퍼컴퓨터를 활용하여, 대규모 과학데이터 분석 환경을 CERN Grid Site로 제공

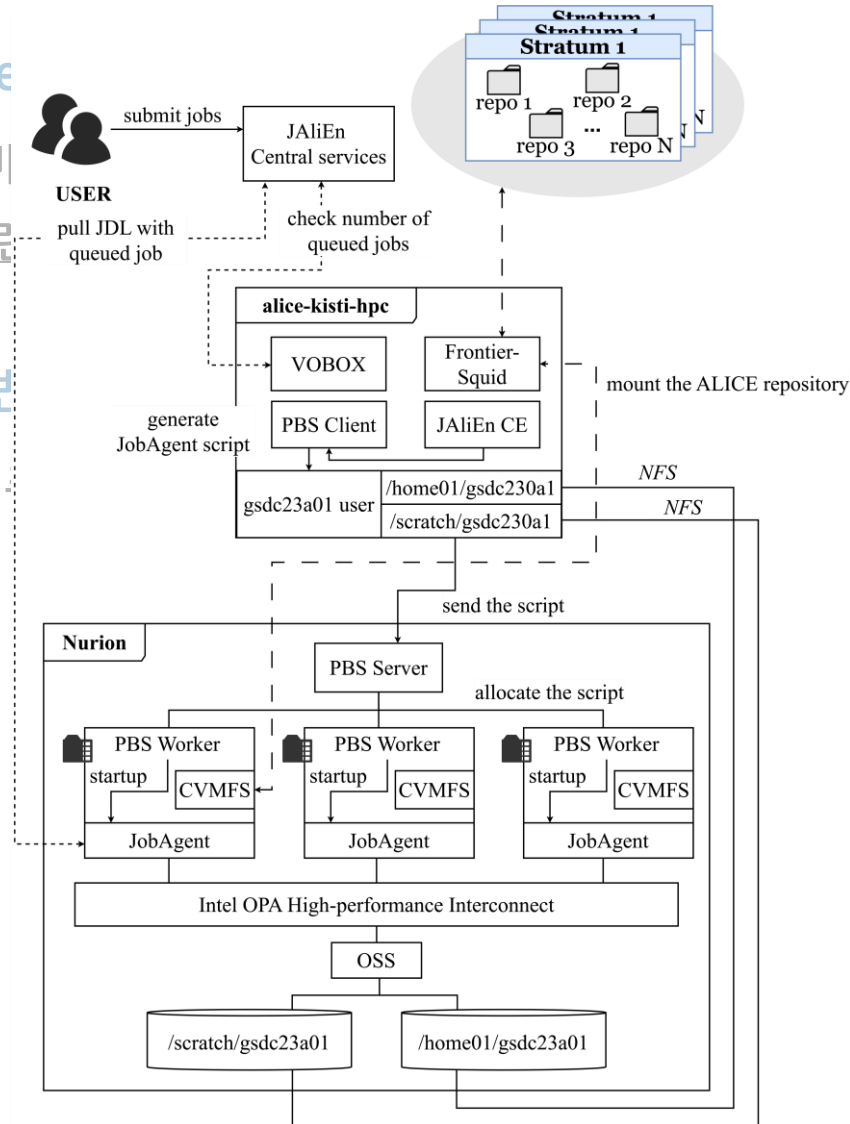
# 8. 클라우드 관련 연구 주제

## 1) Kubernetes Pe

- 컨테이너 서버
- 축소 기능 관련

## 2) 슈퍼컴퓨터 기반

- 5호기 슈퍼컴
- 제공



개발)

볼륨의 자동 확장 및

을 CERN Grid Site로

# 8. 클라우드 관련 연구 주제



## 1) Kubernetes Persistent Volume(PV) 오토스케일러 (기술 개발)

- 컨테이너 서비스에서 발생하는 데이터를 영구 저장하기 위한 볼륨의 자동 확장 및 축소 기능 관련 알고리즘 제안 및 구현

## 2) 슈퍼컴퓨터 기반 CERN Grid Site 구축 (인프라 구축)

- 5호기 슈퍼컴퓨터를 활용하여, 대규모 과학데이터 분석 환경을 CERN Grid Site로 제공

## 3) 다중목적 함수를 활용한 가상머신 재배포 알고리즘 구현 (스케줄링, 최적화)

- 효율적인 가상머신 재배포를 위해 다중목적 함수로 여러 요소를 고려하여, 가능한 적은 서버를 운영할 수 있도록 가상머신을 재배포하는 알고리즘 제안 및 시뮬레이션



# 9. If interesting...



Hyeon-Jin Yu ([연구실 대표](#))

- Dept. of Computer Science
- hyeonjin-yu@chungbuk.ac.kr
- 2023.03 - Current

**Interests :**


- ALICE SW, ALICE HPC

DCLAB Data Computing Laboratory  
데이터 컴퓨팅 연구실


Home News Gallery Members Research Publications Lectures

## Data Computing Laboratory

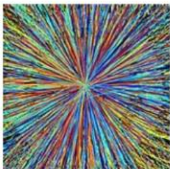
Welcome to the Data Computing Laboratory (DCLab) at [the School of Computer Science, Chungbuk National University](#). Computing plays a pivotal role in various fields, including the ICT industry and scientific research. At DCLab, we conduct research on diverse computing methodologies to address real-world problems using advanced technologies. Our research focuses on various aspects of data-related computing, including cloud computing, big data processing, high-throughput computing, high-performance computing, and quantum computing.




Cloud Computing




Big Data Processing



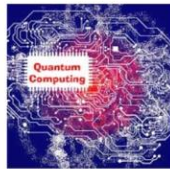
Scientific Computing



High Throughput Computing



High Performance Computing



Quantum Computing