

제 38회 IT 인프라 & 네트워크 전문가 따라잡기 'N.EX.T' [정기 기술 세미나]

Stateless GPU 클러스터 구축 및 자작 모니터링 툴 개발기

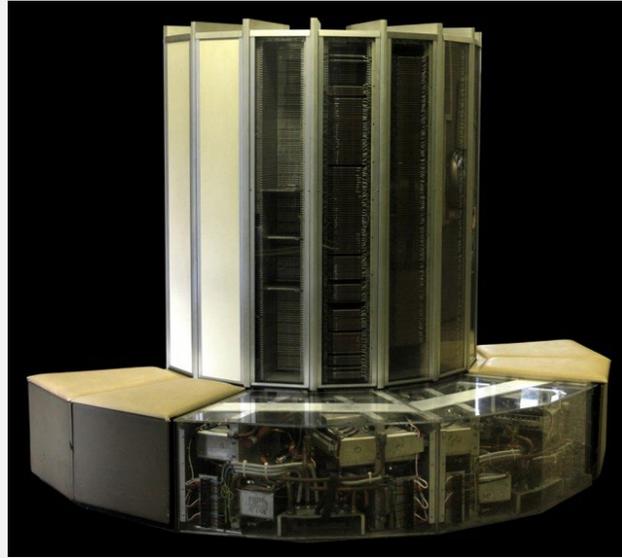
고등과학원 거대수치계산연구센터 김호영
(한국초고성능컴퓨팅포럼 인프라분과 회원)

2025.06.14.

초창기의 슈퍼컴퓨터(HPC)



Control Data Corporation
(CDC) 6600
Released in 1964

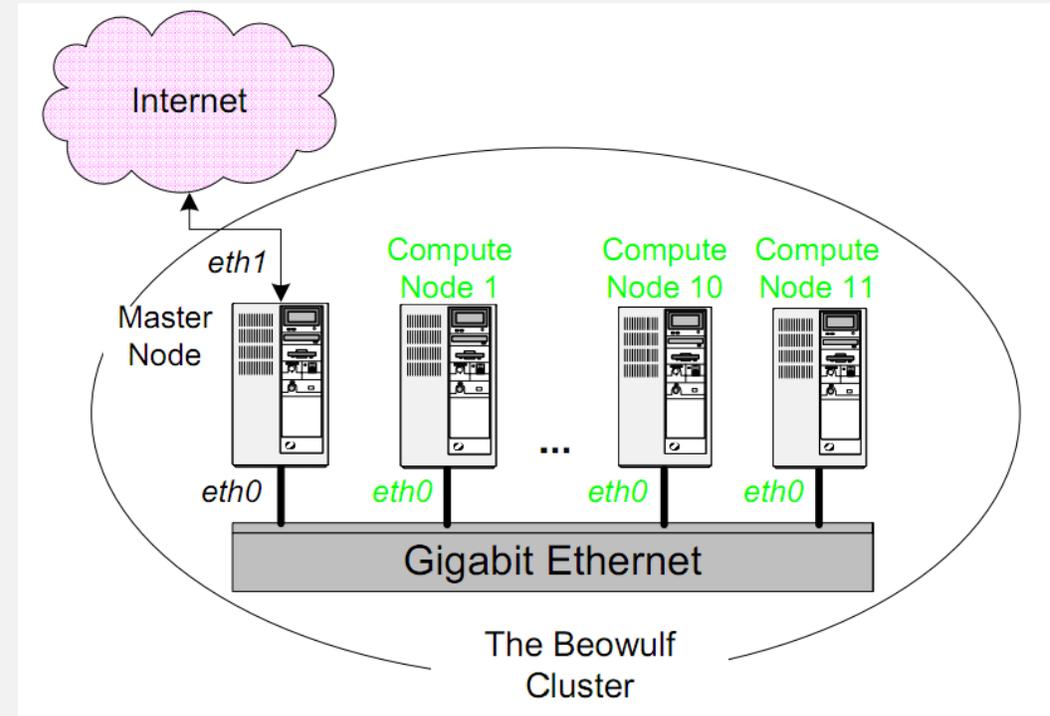


Cray 1
Released in 1976



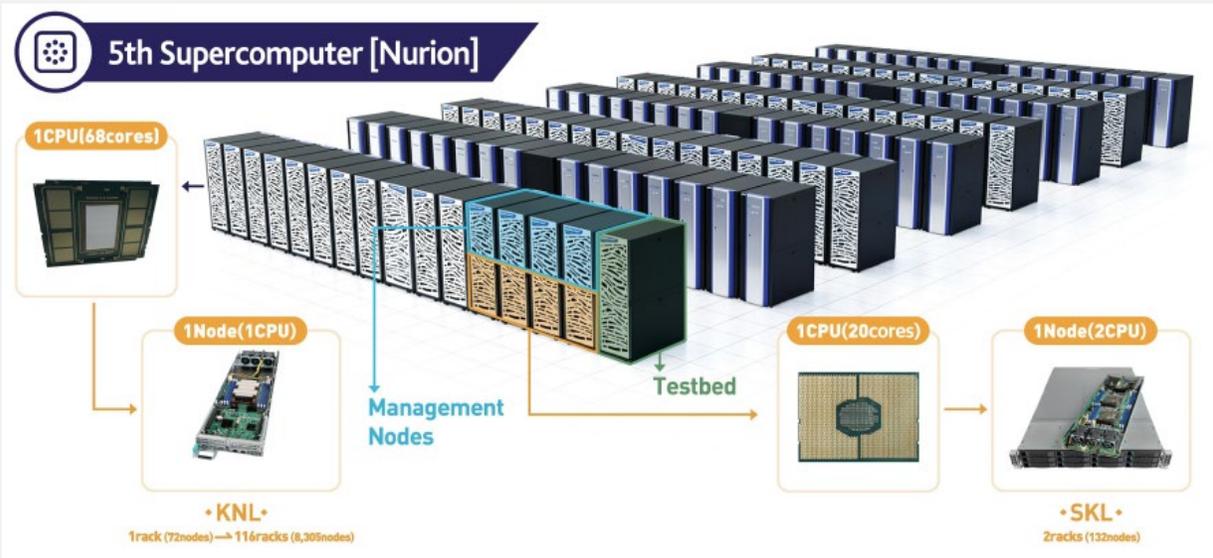
NEC SX-3
Released in 1990

The Beowulf parallel workstation project



The first Beowulf Linux commodity cluster was constructed at [NASA's Goddard Space Flight Center in 1994](#) and its origins are a part of the folklore of high-end computing. In fact, the conditions within Goddard that brought the idea into being were shaped by rich historical roots, strategic pressures brought on by the ramp up of the Federal High-Performance Computing and Communications Program, growth of the open software movement, microprocessor performance trends, and the vision of key technologists. This multifaceted story is told here for the first time from the point of view of NASA project management.

현대의 슈퍼컴퓨터(HPC, Cluster)



Nurion
KISTI, Korea

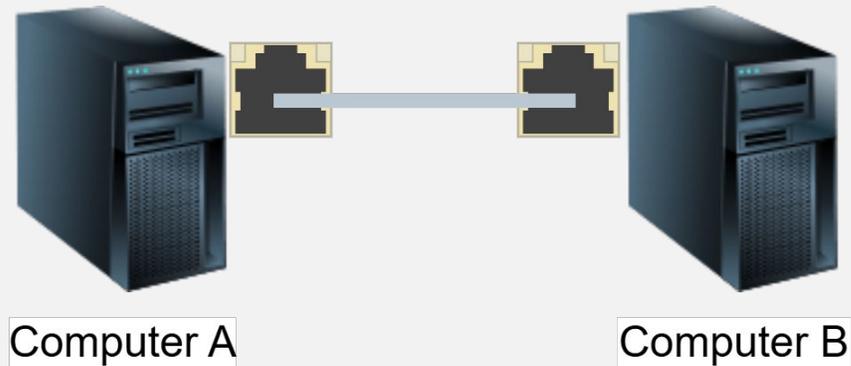


Piz Diant
Swiss National Supercomputing Centre , Switzerland

누구나 쉽게 만드는 HPC

```
Computer_A > ssh-keygen -t rsa
```

```
Computer_A > ssh-copy-id -i ~/.ssh/id_rsa.pub Computer_B
```



데이터 교환을 위해 양측 컴퓨터 간에 SSH 접근이 가능해야 함.

하지만 월급을 받지 못함



오늘의 주제
임금체불 대처방법

노알남

0:25 / 7:30

사장님이 월급을 안줄때 끝까지 받아내는 방법(임금체불 ep.01)

노알남
구독자 1.02만명

구독

226 | | 공유 | 저장 | ...

The image shows a YouTube video player. The video title is '오늘의 주제 임금체불 대처방법' (Today's Topic: How to Deal with Salary Non-payment). The video is from the channel '노알남' (Noalnam), which has 1.02 million subscribers. The video title below the player is '사장님이 월급을 안줄때 끝까지 받아내는 방법(임금체불 ep.01)'. The video progress bar shows 0:25 / 7:30. The video player interface includes standard YouTube controls like play, volume, and full screen. Below the video player, there are interaction buttons for likes (226), comments, share, save, and a menu icon.

HPC in the Movie



영화 : 마션(2015)

HPC in the Real World



많은 서버
—

빠른
네트워크
—

고속
스토리지
—

소프트웨어
구축
—

유지보수
—

모니터링
—

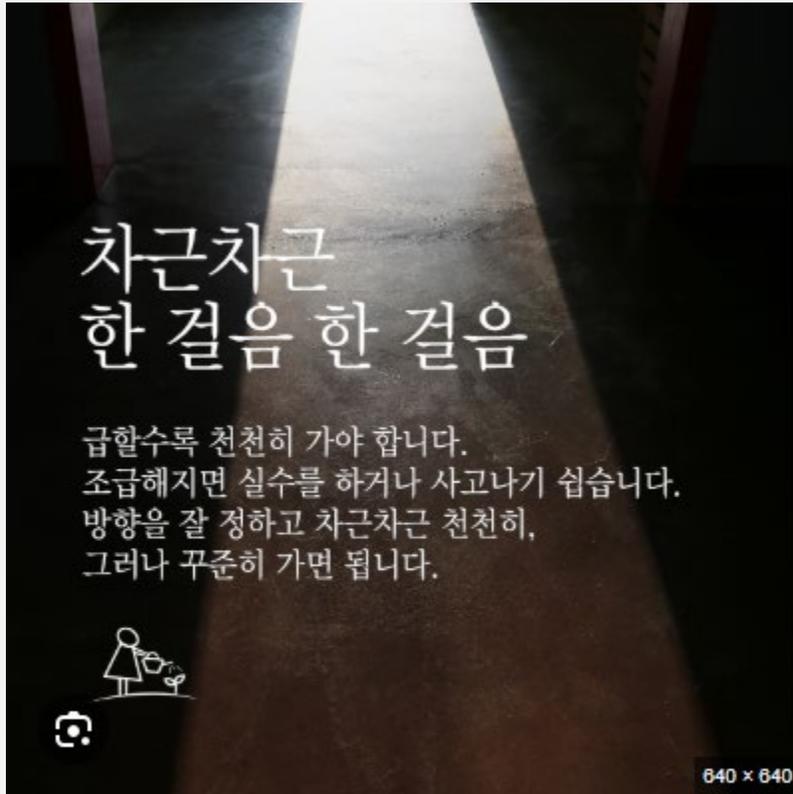
GPU
—

사용자 관리
—

작업
스케줄링
—

환경설비
—

하나씩 차근차근 그리고 많은 예산



차근차근 한 걸음 한 걸음

급할수록 천천히 가야 합니다.
조급해지면 실수를 하거나 사고나기 쉽습니다.
방향을 잘 정하고 차근차근 천천히,
그러나 꾸준히 가면 됩니다.



640 x 640

고도원의 아침편지 - 차근차근 한 걸
음 한 걸음

방문 >



(준비물 : 한도 없는 신용카드)

하지만 월급을 받지 못함

The image shows a YouTube video player with a blue background. The main title is "오늘의 주제" (Today's Topic) and the subtitle is "임금체불 대처방법" (How to Deal with Salary Non-payment). The video features a man in a suit and glasses speaking. The video progress bar shows 0:25 / 7:30. The video title is "사장님이 월급을 안줄때 끝까지 받아내는 방법(임금체불 ep.01)". The channel name is "노알남" with 1.02 million subscribers. The video has 226 likes and a share button.

오늘의 주제
임금체불 대처방법

노알남

0:25 / 7:30

사장님이 월급을 안줄때 끝까지 받아내는 방법(임금체불 ep.01)

노알남
구독자 1.02만명

226 | 공유 | 저장

클러스터 구축의 목표를 세운다.

- 세계 최고의 슈퍼컴퓨터(클러스터)를 만든다.
Top500 1위를 해서 국가 경쟁력을 드높인다.
- 특정 소프트웨어에 탁월한 성능을 내는 클러스터를 만든다.
머신러닝, 추론 전용, 데이터 분석 전용, 실시간 고속 네트워킹 서비스
- 범용으로 가성비 있는 클러스터를 만든다.
다양한 사용자가 다양한 분야에서 사용할 수 있는 시스템을 만든다.

클러스터 구축의 목표를 세운다.

- 세계 최고의 슈퍼컴퓨터(클러스터)를 만든다.
Top500 1위를 해서 국가 경쟁력을 드높인다.
- 특정 소프트웨어에 탁월한 성능을 내는 클러스터를 만든다.
머신러닝, 추론 전용, 데이터 분석 전용, 실시간 고속 네트워킹 서비스

- 범용으로 가성비 있는 클러스터를 만든다.
다양한 사용자가 다양한 분야에서 사용할 수 있는 시스템을 만든다.

범용적인 HPC 클러스터란?



모든 사용자를
만족시키는
시스템

장점 : 할 일이 없다.

범용적인 HPC 클러스터란?



모든 사용자를
만족시키는
시스템

단점 : 그런 시스템은 존재하지 않는다.

범용적인 HPC 클러스터란?



User

User

User

User

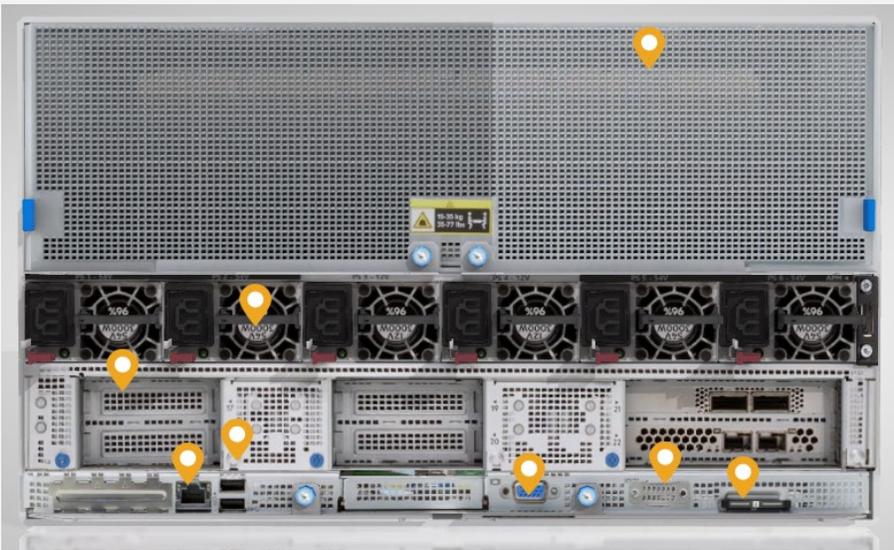
사용자 모두의
요구사항^(불만)을
이성적, 논리적, 감
성적(?)으로 이해시
킬 수 있는
시스템

이제 진짜 시작

Server 를 준비



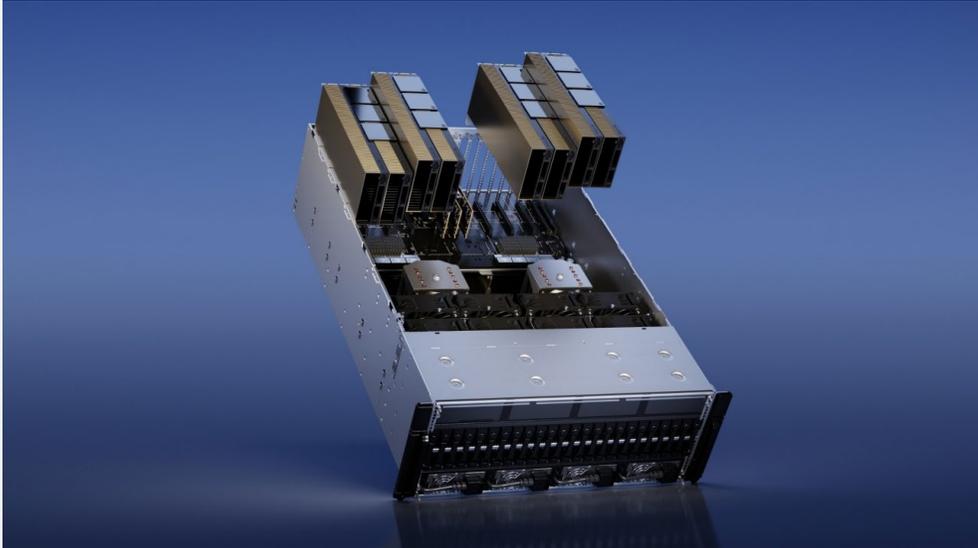
- CPU : AMD EPYC 7543 32-Core ×2EA
- Memory : 1TB
- Network(Ethernet) : 10G
- Network(Infiniband) : 200G HDR
- DISK : NVMe 7TiB ×1EA (캐쉬용)
- IPMI : 1G (시스템 관리용)



- OS용 디스크는 탑재하지 않음(Stateless)
Disk 2EA, H/W RAID 카드 비용 감소

× 20EA

GPU 를 준비



- PCIe 타입 (고장나면 1개씩 교체 가능)



- SXM 타입 (고장나면 전체를 교체)

Interconnect

NVIDIA NVLink™: 900GB/s

PCIe Gen5: 128GB/s

[하이브리드 방식으로 구성 / 일부 노드는 PCIe, 일부 노드는 SXM]

성능 좋은 서버만 준비하면 끝일까?

```
NVIDIA-SMI 440.64.00 Driver Version: 440.64.00 CUDA Version: 10.2
```

GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util Compute M.
0	Tesla V100-SXM2...	On	00000000:1A:00.0	Off	0
N/A	65C	P0	270W / 300W	930MiB / 32510MiB	76%
1	Tesla V100-SXM2...	On	00000000:1C:00.0	Off	0
N/A	54C	P0	225W / 300W	930MiB / 32510MiB	72%
2	Tesla V100-SXM2...	On	00000000:1D:00.0	Off	0
N/A	54C	P0	225W / 300W	930MiB / 32510MiB	75%
3	Tesla V100-SXM2...	On	00000000:1E:00.0	Off	0
N/A	65C	P0	261W / 300W	930MiB / 32510MiB	73%



고속 전용 네트워크 준비

CPU Cluster



FDR(56G) 108 Ports

GPU Cluster

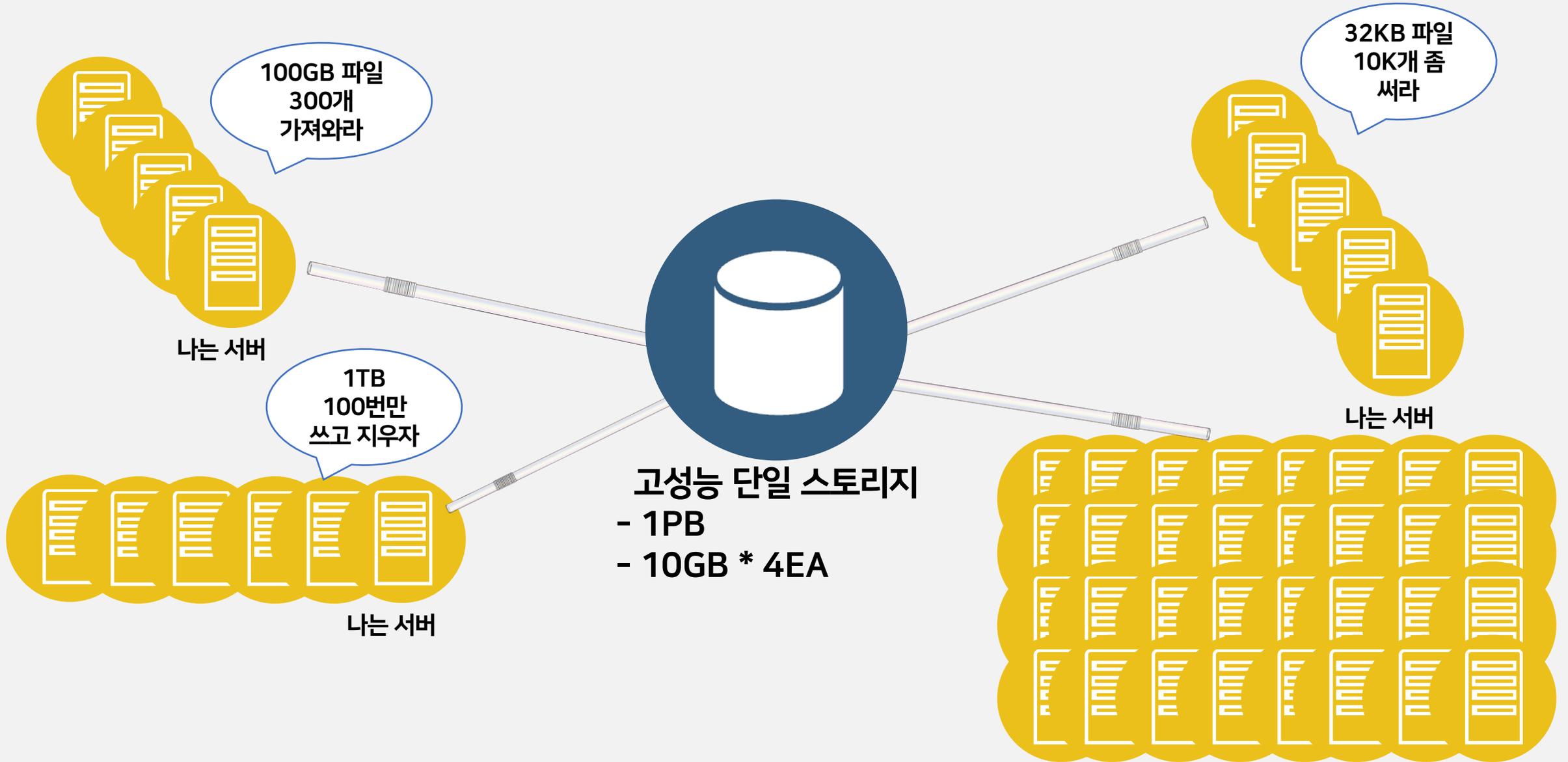


HDR(200G) 40 Ports

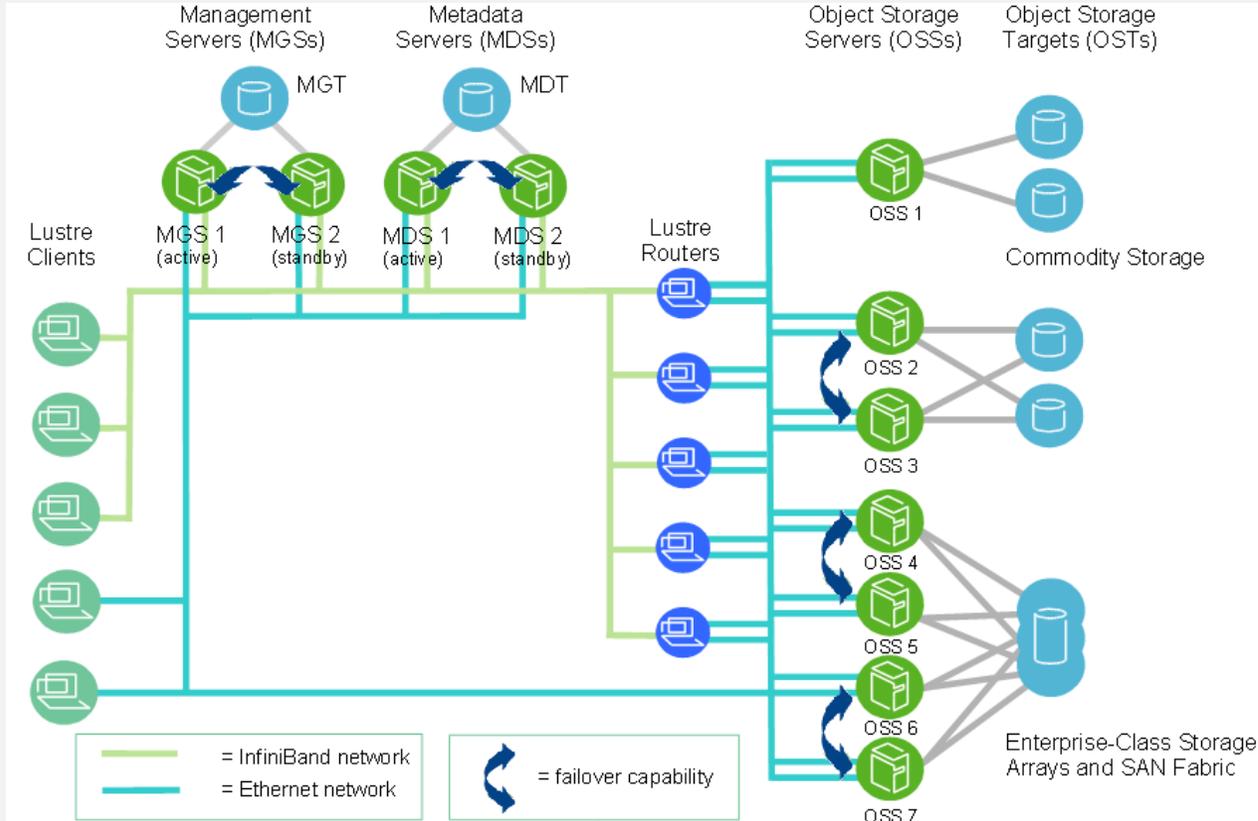
HDR100 80 Ports 가능



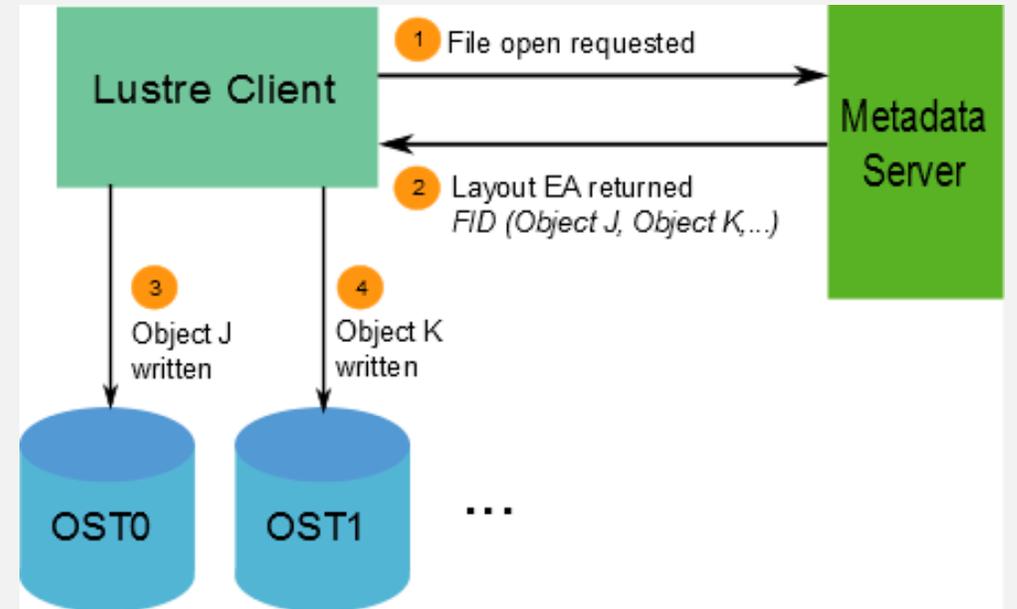
스토리지 준비 : 좋은 스토리지가 필요한 이유



스토리지 준비 : 병렬파일 시스템 Lustre (1)



- ④ inode만 별도로 스토리지에 저장
- ④ 큰 파일은 여러 스토리지에 나눠서 저장
- ④ 작은 파일은 아예 inode 영역에 같이 저장 (<32KB)



스토리지 준비 : 병렬파일 시스템 Lustre (2)

총 용량은 얼마로 할 것인가?

메타데이터 용량은 얼마로 잡을 것 인가?

네트워크는 본딩을 할 것 인가?

Infiniband 네트워크를 추가 구축할까?

클라이언트는 몇 대 까지 붙일 것인가?

모니터링 서버를 따로 둘까?

백업은 어떻게 구성하지?

서버 CPU, 메모리 등 사양은 어떻게 할 것인가?

Kernel 과 Lustre 버전은 얼마로 할까?

MGT와 MDT를 같이 쓸 것인가?

오브젝트 스토리지는 어떻게 구성할 것인가?

각 디스크 RAID는 8+2, 10+2 어떻게 잡지?

Hotspare 는 얼마나 둘 것 인가?

볼륨의 블록 사이즈는?

Striping 정책은 어떻게 할 것인가?

SSD로 구성할까?

SATA 디스크로 구성할까?

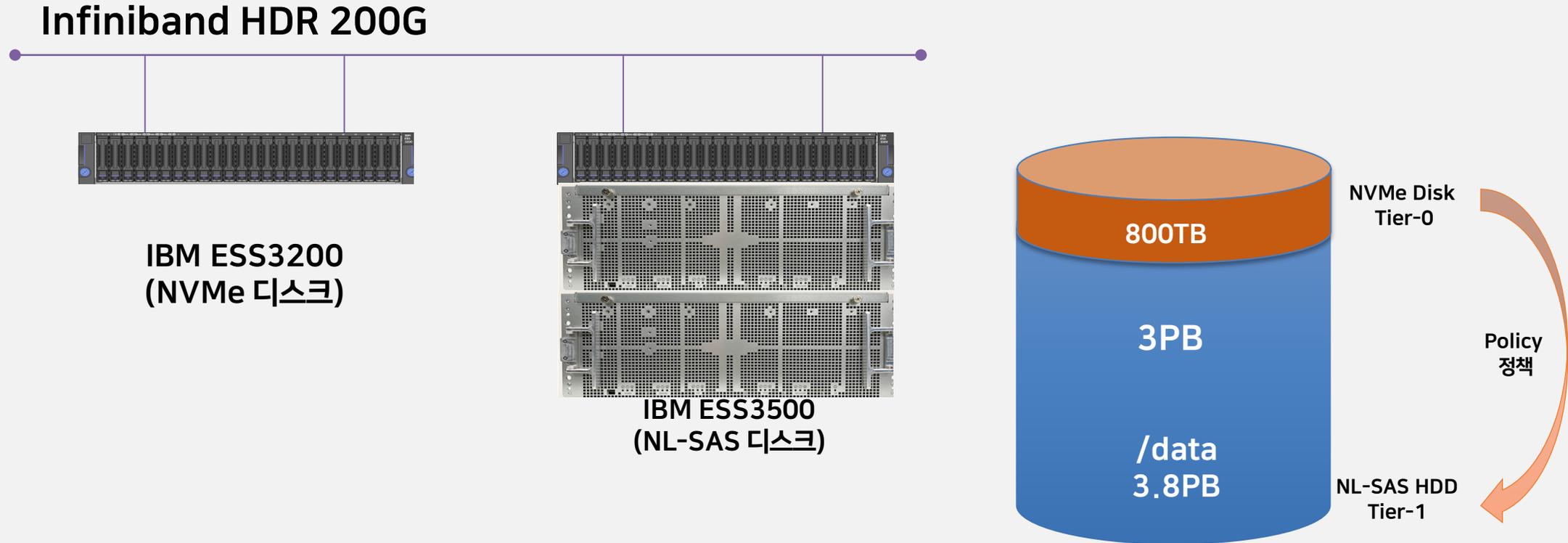
아니면 혼합할까?

Failover는 어디까지 할까?

메타데이터 영역에 데이터도 저장할까?

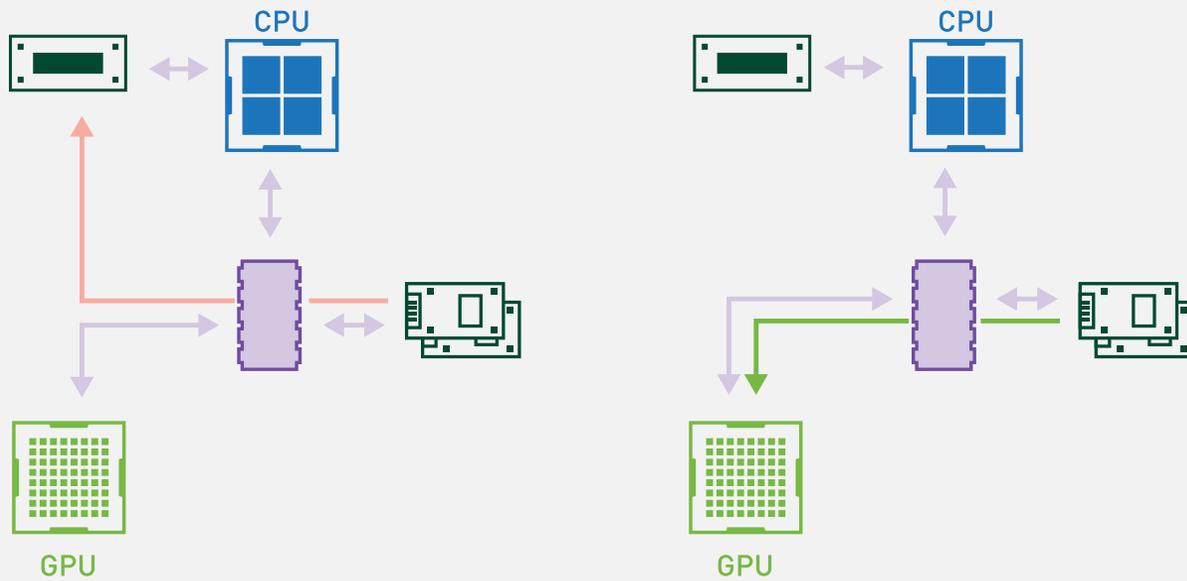


스토리지 준비 : 병렬파일 시스템 GPFS



하나의 볼륨에서 Tier 구조로, 자주 쓰는 데이터, 데이터 크기 등에 따라 정책을 설정하여 효율적으로 활용 가능

스토리지 준비 : GPUDirect Storage (1)



Without GPUDirect Storage

With GPUDirect Storage



GPUDirect Storage는 NVMe 또는 NVMe-oF와 같은 로컬 또는 원격 저장소와 GPU 메모리 간의 직접 데이터 경로를 활성화한다. 이는 CPU 메모리의 중간 버퍼를 통한 추가 복사를 피하고, NIC 또는 저장소 근처의 직접 메모리 액세스(DMA) 엔진을 사용하여 데이터를 GPU 메모리 안팎으로 직접 이동시킴으로써 CPU에 부담을 주지 않는다.

스토리지 준비 : GPUDirect Storage (2)



머리를 거치지 않고 말하는 방식의 장점:

빠르다.

스토리지 준비 : GPUDirect Storage (3)

Lustre 2.15.0 Changelog

•[LU-14798] - NVIDIA GPUDirect Storage Support

GDS On

IoType: READ XferType: GPUD Threads: 64 DataSetSize: 88965120/102400000(KiB) IOSize:
16384(KiB) **Throughput: 8.271321 GiB/sec**, Avg_Latency: 118359.790872 usecs ops: 5430
total_time 10.257581 secs

GDS Off

IoType: READ XferType: CPU_GPU Threads: 64 DataSetSize: 87687168/102400000(KiB)
IOSize: 16384(KiB) **Throughput: 7.456274 GiB/sec**, Avg_Latency: 133003.739459 usecs ops:
5352 total_time 11.215387 secs

스토리지 준비 : 기타 스토리지

백업용 스토리지

시뮬레이션 백업
스토리지 #2

소스코드 저장용
스토리지

시뮬레이션 백업
스토리지 #3

시뮬레이션 백업
스토리지 #1

시뮬레이션 백업
스토리지 #4

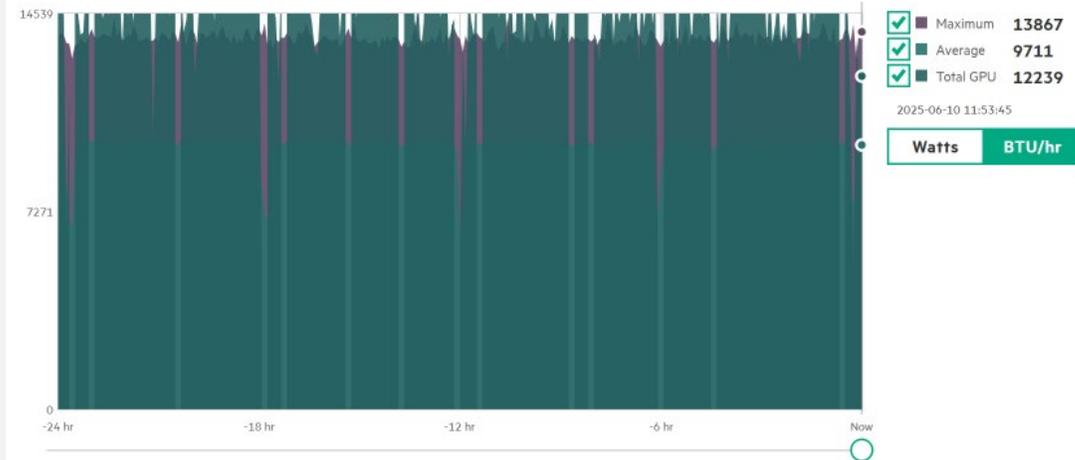
Size	Used	Avail	Use%
52T	40T	12T	78%
66T	17T	49T	26%
88T	80T	3.1T	97%
143T	119T	17T	88%
177T	6.6G	177T	1%
201T	154T	47T	77%
442T	413T	30T	94%
681T	652T	30T	96%
781T	744T	37T	96%
880T	867T	13T	99%
1.4P	1.3P	4.2T	100%
1.6P	1.5P	40T	98%
1.8P	1.8P	50T	98%

Power Supplies

Bay	Present	Status	Capacity
1	✔ OK	✔ Good, In Use	3000 Watts
2	✔ OK	✔ Good, In Use	3000 Watts
3	✔ OK	✔ Good, In Use	3000 Watts
4	✔ OK	✔ Good, In Use	3000 Watts
5	✔ OK	✔ Good, In Use	3000 Watts
6	✔ OK	✔ Good, In Use	3000 Watts

24-Hour History Graph

Power consumption over the past 24 hours at five-minute intervals.



Power Status	Now	Power History	5 min	20 min	24 hr	1 week
Present Power Reading	8363	Maximum Power	13867	14154	14471	14621



구축 시작

누구나 그럴싸하게 클러스터를 구축하는 오픈소스

1. Rocks Cluster (Since 2005) : 2018년 마지막 업데이트
<https://www.rocksclusters.org/>

2. OpenHPC
<https://openhpc.community/>

첫 릴리즈 : OpenHPC v1.0 (12 November 2015)
최근 릴리즈 : [2.9.1 \(12 May 2025\)](#)

OpenHPC 설치 구성

Install Recipes

To aid in the installation of OpenHPC packaged components, a variety of companion installation recipes are available. These can be obtained via installation of the docs-ohpc RPM after the OpenHPC repository has been enabled locally. Alternatively, copies of the guides are also provided for each release track on the OpenHPC GitHub Wiki:

- [OpenHPC 2.x](#)
- [OpenHPC 3.x](#)

[RHEL 8.10 Variants]

- [Latest version of 'ohpc-release' for EL8 \(md5sum=35eed49860df2b0b92331c6d19761879\)](#)
- [Rocky Install Guide \(with Warewulf + OpenPBS\)](#)
- [Rocky Install Guide \(with Warewulf + Slurm\)](#)

[OpenSUSE Leap 15.3]

- [Latest version of 'ohpc-release' for Leap_15 \(md5sum=156d2919b4659b5f15f7706945527cd2\)](#)
- [Install Guide \(with Warewulf + OpenPBS\)](#)
- [Install Guide \(with Warewulf + Slurm\)](#)

OpenHPC 설치 자동화 (2시간이면 Okay)

```
In [7]: 1 BUILD_CMDS=["sh $OPENHPC_BUILD_FOLDER/Phase01.sh 2>&1 | tee /root/build_logs/Phase01.log",
2 "sh $OPENHPC_BUILD_FOLDER/Phase02.sh 2>&1 | tee /root/build_logs/Phase02.log",
3 "sh $OPENHPC_BUILD_FOLDER/Phase03-IB.sh 2>&1 | tee /root/build_logs/Phase03-IB.log",
4 "sh $OPENHPC_BUILD_FOLDER/Phase04-ZFS.sh 2>&1 | tee /root/build_logs/Phase04-ZFS.log",
5 "sh $OPENHPC_BUILD_FOLDER/Phase05-OHPC.sh 2>&1 | tee /root/build_logs/Phase05-OHPC.log",
6 "sh $OPENHPC_BUILD_FOLDER/Phase06-Lustre.sh 2>&1 | tee /root/build_logs/Phase06-Lustre.log",
7 "sh $OPENHPC_BUILD_FOLDER/Phase07-Enable-Lustre.sh 2>&1 | tee /root/build_logs/Phase07-Enable-Lustre.log",
8 "sh $OPENHPC_BUILD_FOLDER/Phase08-NFS.sh 2>&1 | tee /root/build_logs/Phase08-NFS.log",
9 "sh $OPENHPC_BUILD_FOLDER/Phase09_Install_CUDA.sh 2>&1 | tee /root/build_logs/Phase09_Install_CUDA.log",
10 "sh $OPENHPC_BUILD_FOLDER/Phase10_install_GPFS.sh 2>&1 | tee /root/build_logs/Phase10_install_GPFS.log",
11 "sh $OPENHPC_BUILD_FOLDER/Phase11_SetupGPFS.sh 2>&1 | tee /root/build_logs/Phase11_SetupGPFS.log",
12 "sh $OPENHPC_BUILD_FOLDER/Phase12_Zabbix.sh 2>&1 | tee /root/build_logs/Phase12_Zabbix.log",
13 "sh $OPENHPC_BUILD_FOLDER/Phase13-BasicSoftwares.sh 2>&1 | tee /root/build_logs/Phase13-BasicSoftwares.log",
14 "sh $OPENHPC_BUILD_FOLDER/Phase14-IntelOneApi.sh 2>&1 | tee /root/build_logs/Phase14-IntelOneApi.log",
15 "sh $OPENHPC_BUILD_FOLDER/Phase15-Python.sh 2>&1 | tee /root/build_logs/Phase15-Python.log",
16 "sh $OPENHPC_BUILD_FOLDER/Phase100_NODE-CHROOT.sh 2>&1 | tee /root/build_logs/Phase100_NODE-CHROOT.log",
17 "sh $OPENHPC_BUILD_FOLDER/Phase101_NODE-IB.sh 2>&1 | tee /root/build_logs/Phase101_NODE-IB.log",
18 "sh $OPENHPC_BUILD_FOLDER/Phase102_NODE-Lustre.sh 2>&1 | tee /root/build_logs/Phase102_NODE-Lustre.log",
19 "sh $OPENHPC_BUILD_FOLDER/Phase103_NFS-MOUNT.sh 2>&1 | tee /root/build_logs/Phase103_NFS-MOUNT.log",
20 "sh $OPENHPC_BUILD_FOLDER/Phase104_NODE-ZFS.sh 2>&1 | tee /root/build_logs/Phase104_NODE-ZFS.log",
21 "sh $OPENHPC_BUILD_FOLDER/Phase105_NODE_CUDA.sh 2>&1 | tee /root/build_logs/Phase105_NODE_CUDA.log",
22 "sh $OPENHPC_BUILD_FOLDER/Phase106_NODE_GPFS.sh 2>&1 | tee /root/build_logs/Phase106_NODE_GPFS.log",
23 "sh $OPENHPC_BUILD_FOLDER/Phase107-Zabbix.sh 2>&1 | tee /root/build_logs/Phase107-Zabbix.log",
24 "sh $OPENHPC_BUILD_FOLDER/Phase190-Provisioning.sh 2>&1 | tee /root/build_logs/Phase190-Provisioning.log",]
25 "sh $OPENHPC_BUILD_FOLDER/Phase200-SlurmConf.sh 2>&1 | tee /root/build_logs/Phase200-SlurmConf.log",
26 "sh $OPENHPC_BUILD_FOLDER/Phase210-AfterFinish.sh 2>&1 | tee /root/build_logs/Phase210-AfterFinish.log",
27 "sh $OPENHPC_BUILD_FOLDER/Phase211_nv_fabricmanager.sh 2>&1 | tee /root/build_logs/Phase211_nv_fabricmanager.log",
28 "sh $OPENHPC_BUILD_FOLDER/sw/Install_CUDA.sh 2>&1 | tee /root/build_logs/Install_CUDA.log"]
```

OpenHPC 설치 - 1부 헤더노드의 구성

Phase01.sh

Phase02.sh

Phase03-IB.sh

Phase04-ZFS.sh

Phase05-OHPC.sh

Phase06-Lustre.sh

Phase07-Enable-Lustre.sh

Phase08-NFS.sh

Phase09_Install_CUDA.sh

Phase10_install_GPFS.sh

Phase11_SetupGPFS.sh

Phase12_Zabbix.sh

Phase13-BasicSoftwares.sh

Phase14-IntelOneApi.sh

Phase15-Python.sh

OpenHPC 설치 - 1부 헤더노드의 구성

STEP 01

기본 O/S관련 설치

- 기본 레포지터리 변경
- 타임서버, 기본 라우팅
- 방화벽
- 기본 소프트웨어 (vim, screen, ksh, csh, ...)

STEP 02

드라이버 설치 및 파일 시스템 설치

- Infiniband
- Lustre, GPFS, GlusterFS, ZFS, NFSoverRDMA ...

STEP 03

GPU 관련 설치

- nVidia Driver 설치
- CUDA 설치
- HPC Toolkit 설치

STEP 04

OpenHPC 툴 및 모니터링

- Weowulf 툴
- Slurm, Daemon Control, DB, RestAPI
- Zabbix
- Environment module

STEP 05

HPC 관련 소프트웨어 설치

- GNU Software
- Intel OneAPI
- AMD Compiler
- Python
- Julia
- Quantum Espresso
- Mathematica
- Matlab
- Lammps
- Jupyter Hub

OpenHPC 설치 - 2부 이미지 생성 및 프로비저닝

Phase100_NODE-CHROOT.sh
Phase101_NODE_IB.sh
Phase102_NODE_Lustre.sh
Phase103_NFS-MOUNT.sh
Phase104_NODE_ZFS.sh
Phase105_NODE_CUDA.sh
Phase106_NODE_GPFS.sh
Phase107-Zabbix.sh
Phase190-Provisioning.sh
Phase200-SlurmConf.sh
Phase210-AfterFinish.sh
Phase211_nv_fabricmanager.
sh
sw/Install_CUDA.sh

OpenHPC 설치 - 2부 계산노드 배포

STEP 06

계산노드 스크립트 구성

- 호스트이름
- NIC 포트
- IP 주소
- IPoIB 주소
- BMC IP
- 호스트이름 Prefix

STEP 07

계산노드 배포용 OS 이미지 구성

- 헤더노드와 동일하게 구성
- CHROOT을 이용
- Mount Point 설정
- Stateless 방식의 경우 OS이미지는 램에 상주 함.

STEP 08

개별 노드 스크립트 생성

- 구성이 다른 노드의 경우 (DISK, GPU, 서비스 등) 부팅 후 개별 스크립트 동작

STEP 09

계산노드 부팅 및 확인

- IPMI 톨로 일괄 부팅 후 이미지 배포가 제대로 되는지 확인
- 사전작업: 부팅 순서를 PXE 부팅

STEP 10

Job Scheduling 설정

- Slurm, GPU Gres 등

OpenHPC 부연설명 - Environment Module



```
EasyBuild/4.6.2 intel/
amd/aocc/4.1.0 intel/
amd/aocl intel/
autotools intel/
cmake/3.24.2 intel/
cuda/11.2.2 intel/
cuda/11.3.1 intel/
cuda/11.4.4 intel/
cuda/11.5.2 intel/
cuda/11.6.2 intel/
cuda/11.7.1 intel/
cuda/11.8.0 intel/
cuda/12.0.0 intel/
cuda/12.1.1 intel/
cuda/12.2.1 intel/
cuda/12.4.1 (D) intel/
ffmpeg intel/
gnu12/12.2.0 intel/
hdf5_GNU intel/
hdf5_intel intel/
hwloc/2.7.0 intel/
```

```
$ module load gcc/12.4.0
$ which gcc
$ /usr/local/gcc/12.4.0/linux-x86_64/bin/gcc
```

Now we'll switch to a different version of the module

```
$ module switch gcc/14
$ which gcc
/usr/local/gcc/14.2.0/linux-x86_64/bin/gcc
```

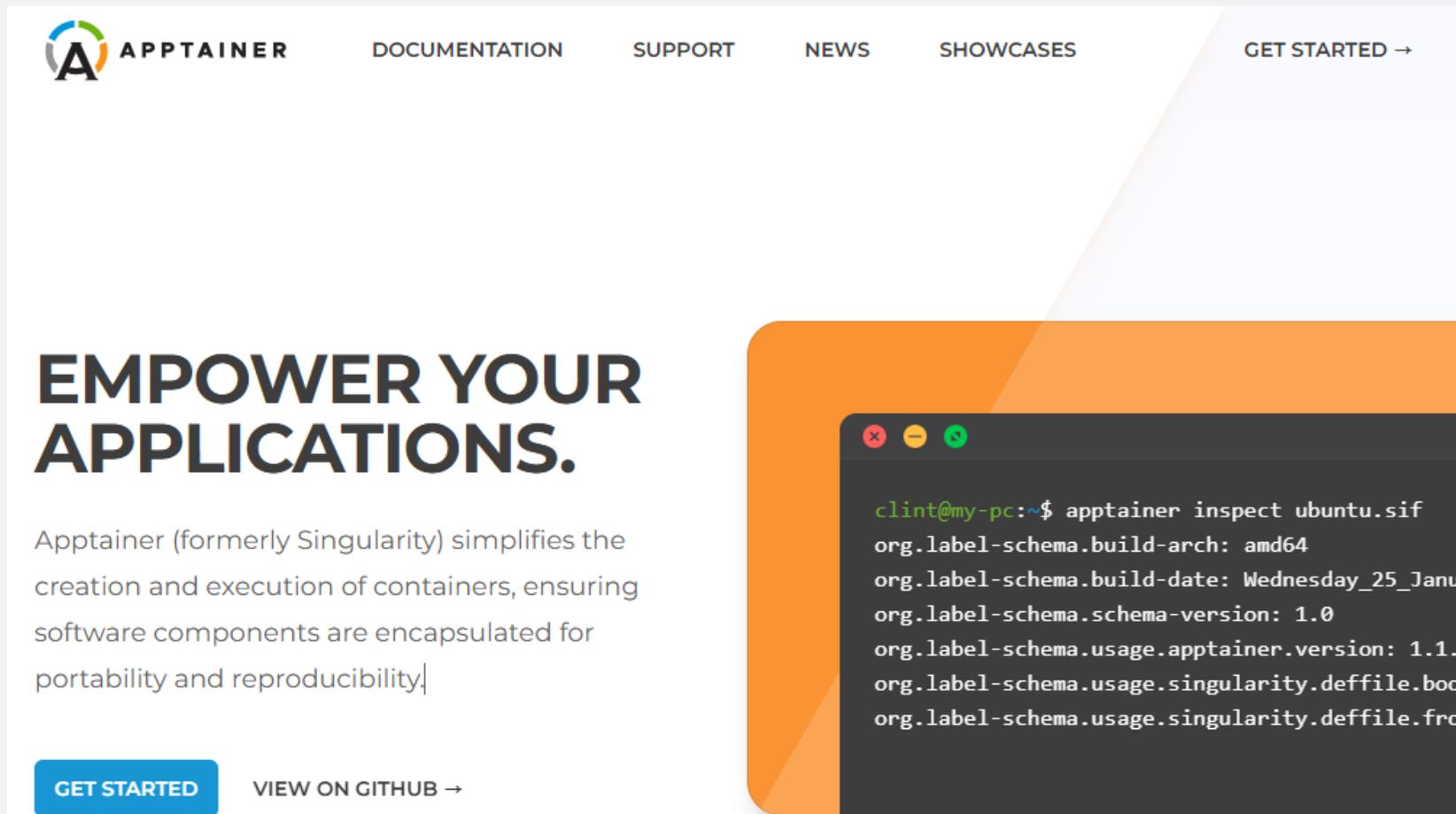
And now we'll unload the module altogether

```
$ module unload gcc
$ which gcc
gcc not found
```

왜 도키는 안 써요?

멀티 사용자 환경에서 Root 권한 문제

그래서 Apptainer를 사용



APPTAINER DOCUMENTATION SUPPORT NEWS SHOWCASES GET STARTED →

EMPOWER YOUR APPLICATIONS.

Apptainer (formerly Singularity) simplifies the creation and execution of containers, ensuring software components are encapsulated for portability and reproducibility.

[GET STARTED](#) [VIEW ON GITHUB →](#)

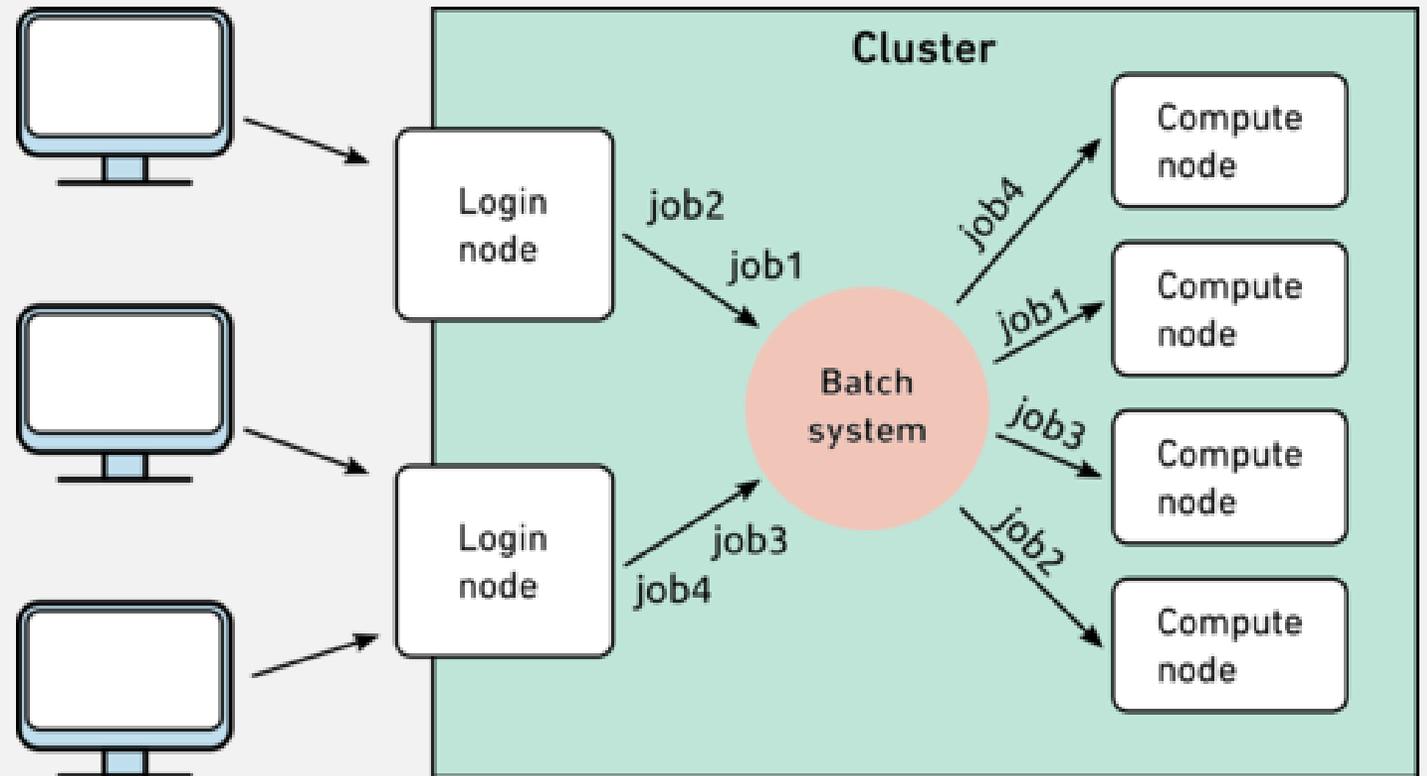
```
clint@my-pc:~$ apptainer inspect ubuntu.sif
org.label-schema.build-arch: amd64
org.label-schema.build-date: Wednesday_25_Janu
org.label-schema.schema-version: 1.0
org.label-schema.usage.apptainer.version: 1.1.
org.label-schema.usage.singularity.deffile.bo
org.label-schema.usage.singularity.deffile.fro
```

작업 스케줄러 - Slurm



```
#!/bin/bash
# Job name:
#SBATCH --job-name=test
# Account:
#SBATCH --account=account_name
# Partition:
#SBATCH --partition=gpu_A5000
# Number of nodes:
#SBATCH --nodes=1
# Number of tasks
#SBATCH --ntasks=1
# Processors per task:
#SBATCH --cpus-per-task=4
#Number of GPUs
#SBATCH --gres=gpu:A5000:1

# Wall clock limit:
#SBATCH --time=00:00:30
## Command(s) to run (example):
./a.out
```



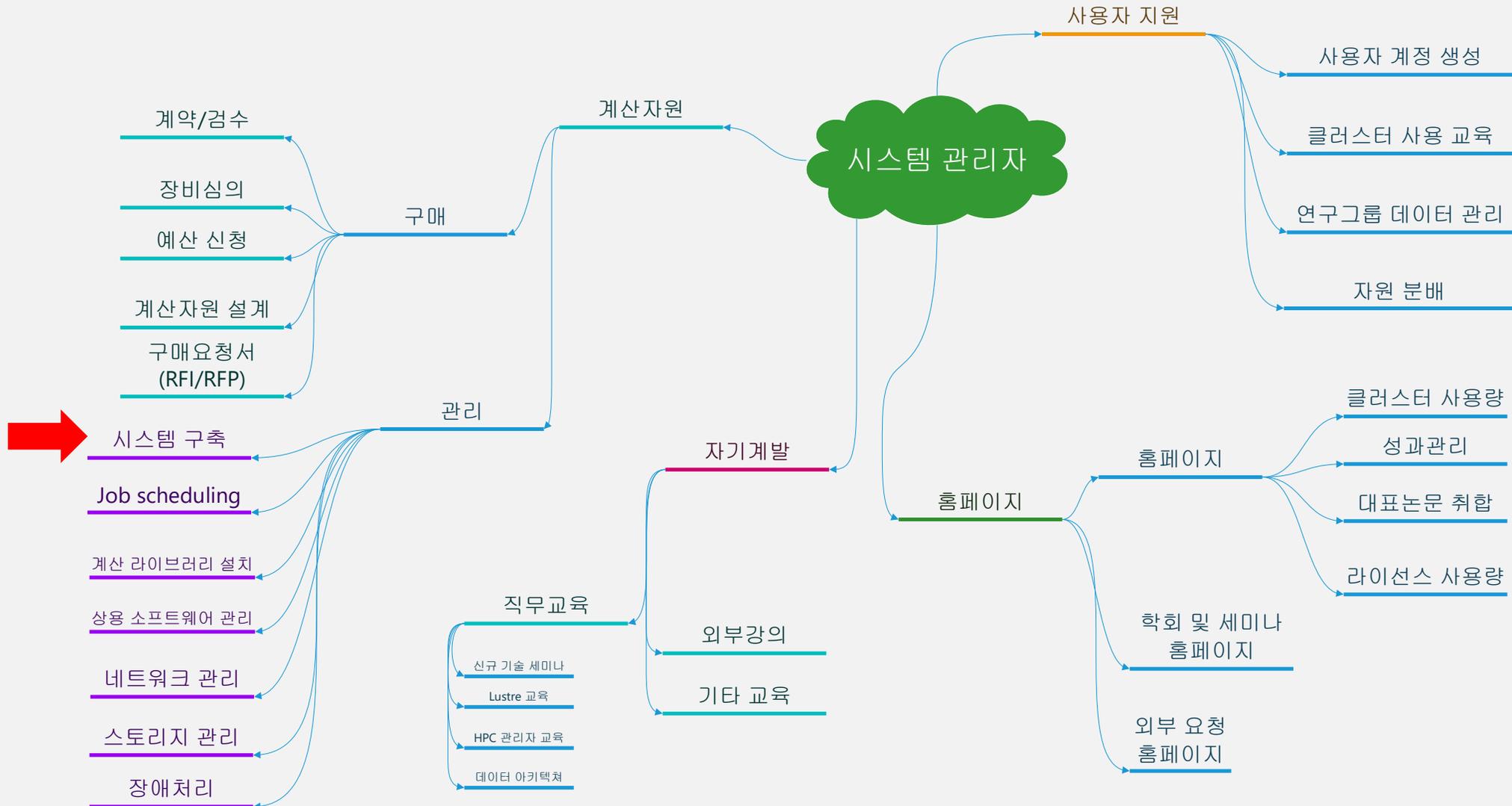
일단 구축은 다 했는데...



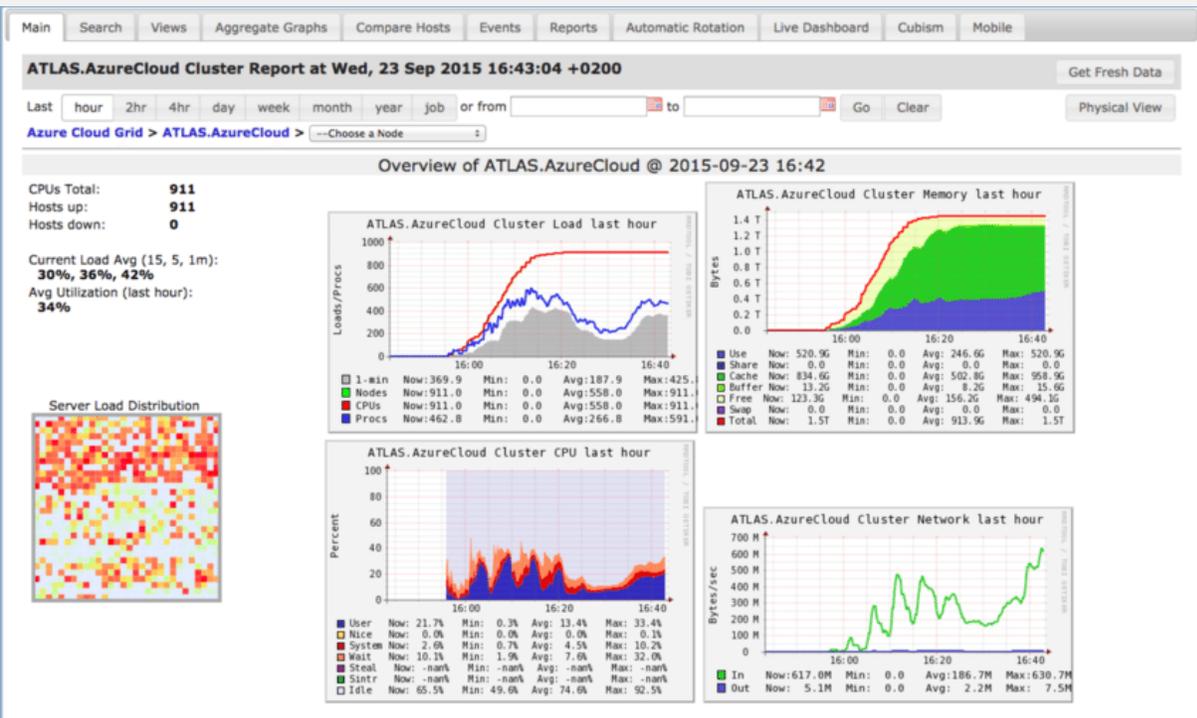
실제 보유하고 있는 시스템의 뒷면



시스템 관리자가 일반적으로 하는 일들



10년 전 모니터링 Ganglia 와 요즘의 Zabbix



그냥 Zabbix 쓰면 안되나요?

YouTube
강성지 (찌야) · 5:07

< > ⋮ ✕



나도... 개발이 하고 싶어요...

시청 >

직접 모니터링 시스템을 구축한 이유

- K-전산에 특화 된 모니터링 시스템이 필요
- 단순 모니터링 뿐 아니라 보고서도 자동 작성
- 다중 역할 사용자가 함께 이용하는 시스템
- 서버 자원 뿐 아니라 부대 시설까지 함께 통합 모니터링
- 프로젝트에 이름 하나 짓고 싶어서... (Volluto Project)

Volluto Project -1

ADMIN / USER / USER-EMAIL

User-Email List

Please select user email list by click servers

checkboxes

Active_User

Select Clusters

Grammar

Syntax

Lexicon

Select Clusters

DNA

RNA

Baekdu

Get List Exclude members Copy List to Clipboard

현재 사용중인 사용자 이메일 리스트
- 긴급 점검 시 현재 이용중인 사용자에게 메일 공지 가능

ADMIN / USER / Password

Password Generator

Generated Password: # \$qQL(T4\$UEL

Generate Password Copy Password

비밀번호 생성기

Volluto Project -2

스토리지 현황 확인 및 디스크 교체 이력

Last Checked: 2025-06-10 23:10:14

Slot	State	Size	Model	SN	IntF	Timestamp	
250:0	Onln	20.008 TB	WUH722222AL5204	WUH722222AL5204	SAS	2025-03-01 00:58:24	
250:1	Onln	20.008 TB	WUH722222AL5204	WUH722222AL5204	SAS	2025-03-01 00:58:24	
250:2	Onln	20.008 TB	WUH722222AL5204	WUH722222AL5204	SAS	2025-03-01 00:58:24	
250:3	Onln	20.008 TB	WUH722222AL5204	WUH722222AL5204	SAS	2025-03-01 00:58:24	
250:4	Onln	20.008 TB	WUH722222AL5204	WUH722222AL5204	SAS	2025-03-01 00:58:24	
250:5	Onln	20.008 TB	WUH722222AL5204	WUH722222AL5204	SAS	2025-03-01 00:58:24	
250:6	Onln	20.008 TB	WUH722222AL5204	WUH722222AL5204	SAS	2025-03-01 00:58:24	
250:7	Onln	20.008 TB	WUH722222AL5204	WUH722222AL5204	SAS	2025-03-01 00:58:24	
250:8	Onln	20.008 TB	WUH722222AL5204	WUH722222AL5204	SAS	2025-03-01 00:58:24	
250:9	Onln	20.008 TB	WUH722222AL5204	WUH722222AL5204	SAS	2025-03-01 00:58:24	
250:10	Onln	20.008 TB	WUH722222AL5204	WUH722222AL5204	SAS	2025-03-01 00:58:24	
0:11	Onln	10.913 TB	ST12000NM004J	ST12000NM004J	SAS	2025-05-27 03:10:03	🕒
0:11	Cpybck	10.913 TB	ST12000NM004J	ST12000NM004J	SAS	2025-05-26 11:10:09	
0:11	UBad	10.913 TB	ST12000NM0027	ST12000NM0027	SAS	2025-05-24 15:10:03	
0:11	Onln	10.913 TB	ST12000NM0027	ST12000NM0027	SAS	2025-03-01 00:58:20	
250:11	Onln	20.008 TB	WUH722222AL5204	WUH722222AL5204	SAS	2025-03-01 00:58:24	

Volluto Project -3

노드 상세 정보

이름	부팅 시간	상태	CPU 로드	메모리
[blurred]	2024-12-19 11:17:33	allocated	2227	87543
[blurred]	2024-12-19 11:19:26	allocated	2021	97109
[blurred]	2025-03-14 16:35:52	allocated	2019	70799
[blurred]	2024-12-19 11:19:28	allocated	2017	73780
[blurred]	2024-12-19 11:17:38	allocated	2229	63106
[blurred]	2024-12-19 11:17:35	allocated	2018	76032
[blurred]	2024-12-19 11:17:38	allocated	2019	88558
[blurred]	2024-12-19 11:17:43	allocated	2015	86150

[새로고침](#)

Name	IDLE	Allocated	Mixed	Down	
[blurred]	12	75	20	2	상세
[blurred]	3		9		상세
[blurred]	11		1		상세
[blurred]	36	58		1	상세
[blurred]	38	14		1	상세
[blurred]	8	61			상세

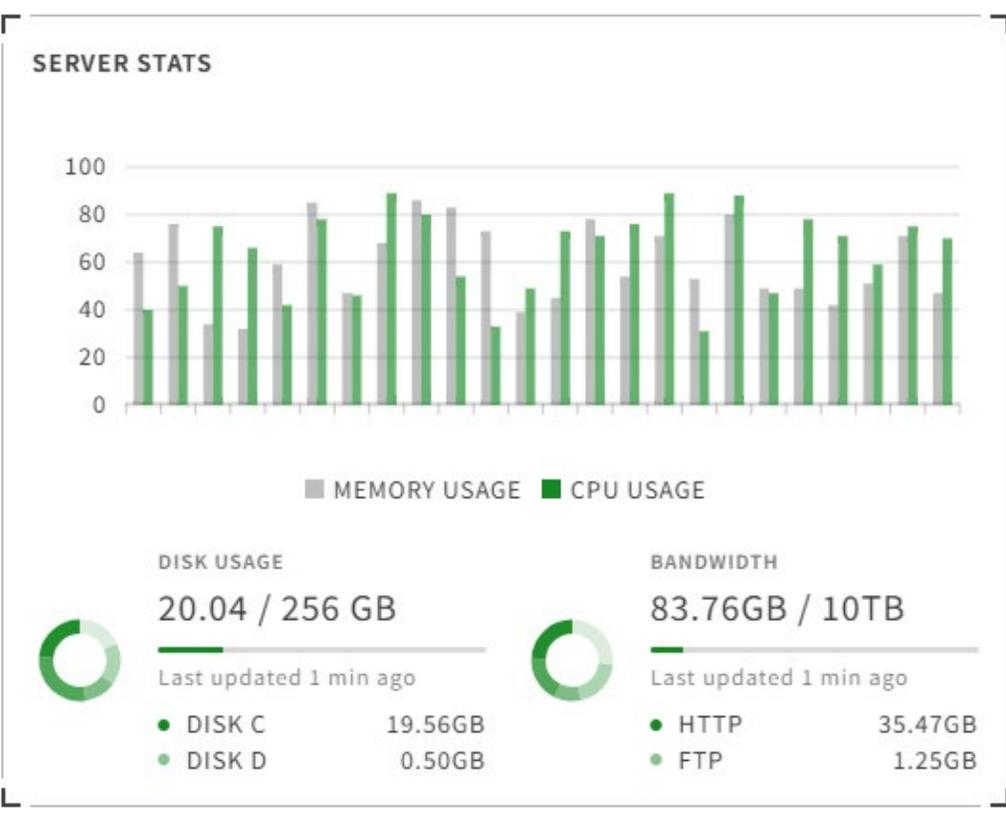
76491 [blurred]

76492 [blurred]

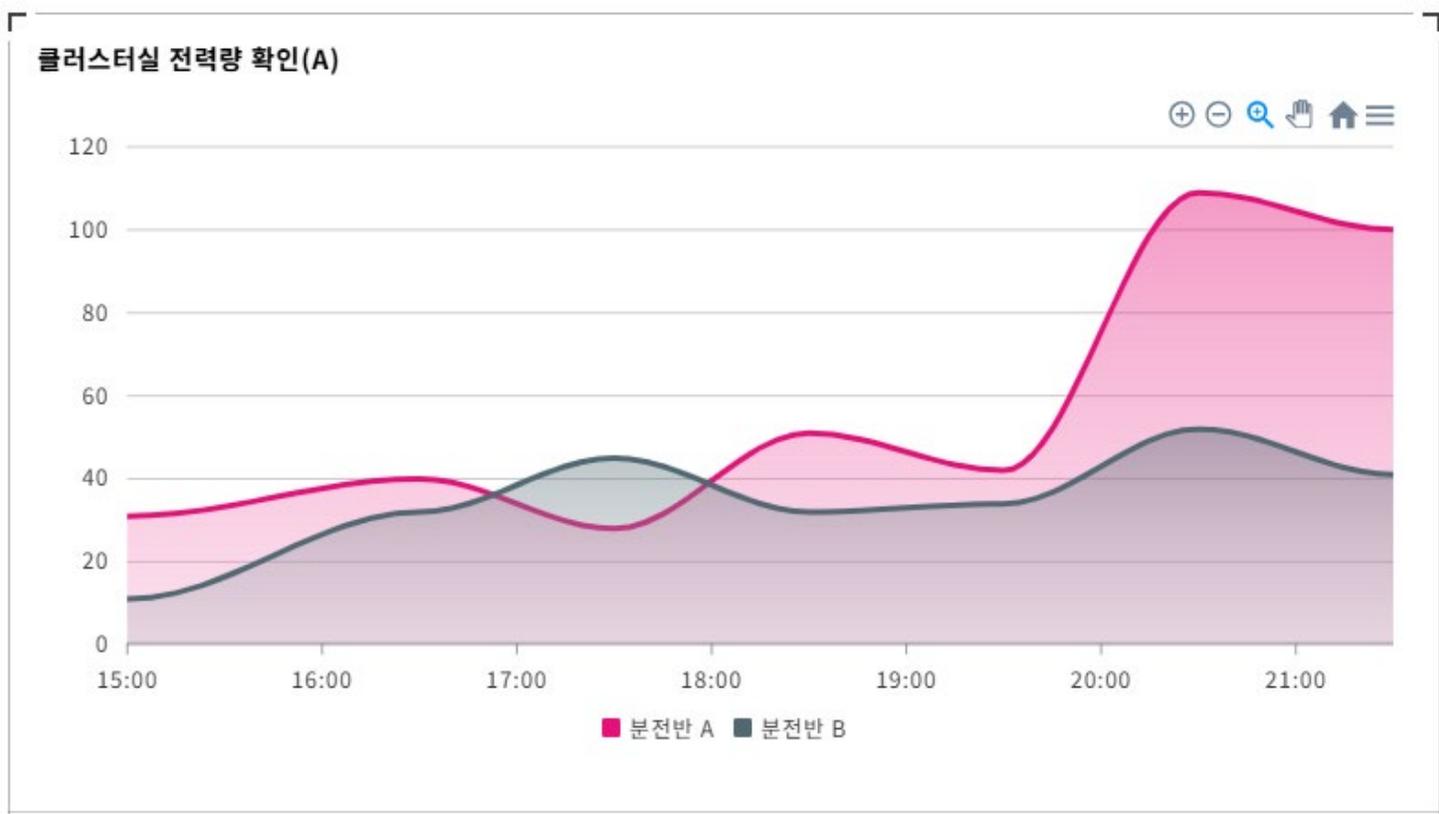
76493 [blurred]

각 클러스터 현황 및 상세 보기

Volluto Project - 4



서버 현황



분전반 현황

일단 모니터링은 맞습니다.

- 계정 신청
- 클러스터 작업 모니터링
- 스토리지 사용량 모니터링
- 공지사항 확인
- 운영자 티켓 요청



관리자

- 계정 발급
- VPN 발급
- 전 시스템 모니터링
- 사용자 티켓 처리
- 유지보수 보고서 확인
- 이메일 처리
- 장애처리

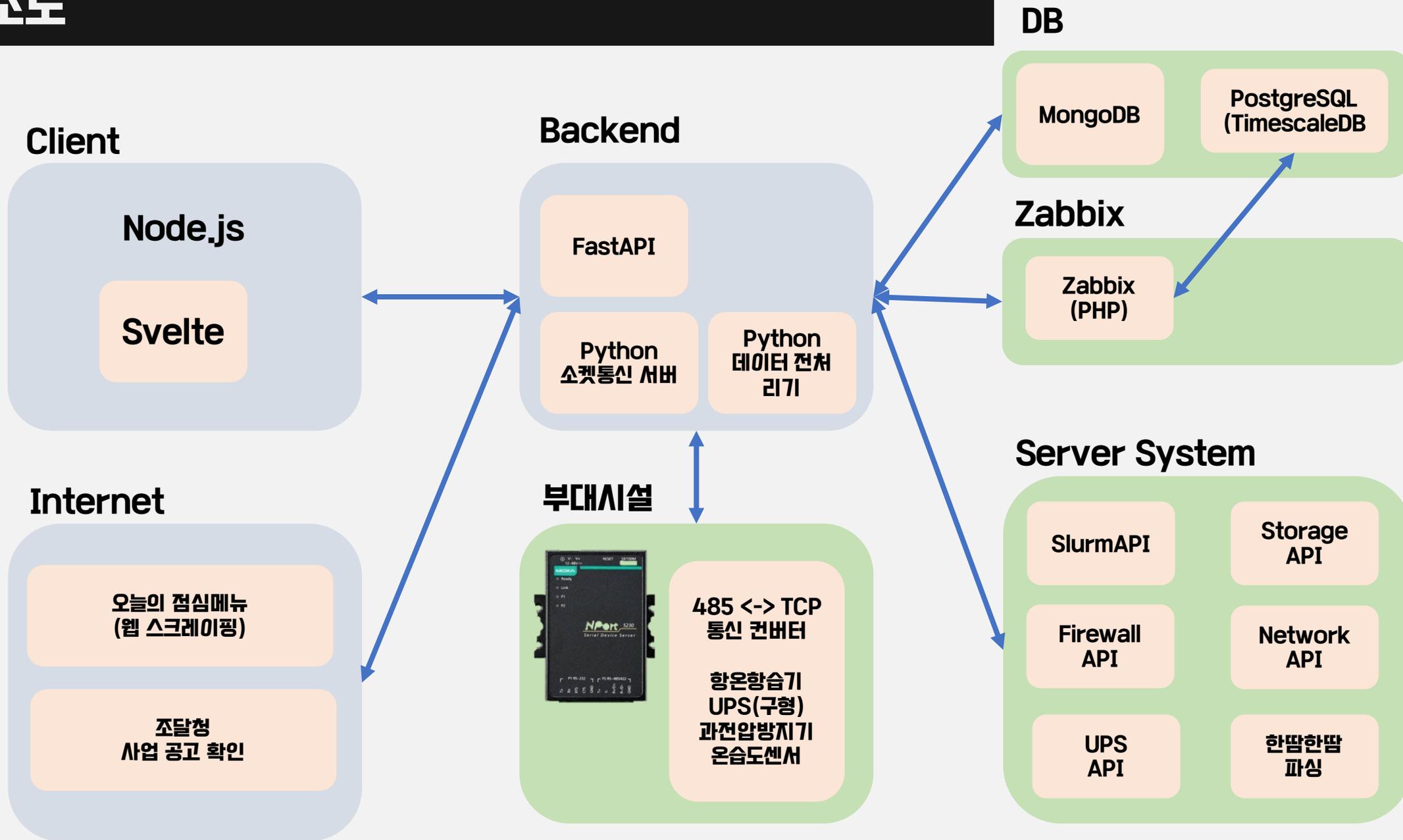
- 장애처리
- 사이트 방문 날짜 신청
- 점검 보고서 제출
- 사용자 티켓 처리
- 여유 파트 재고 확인



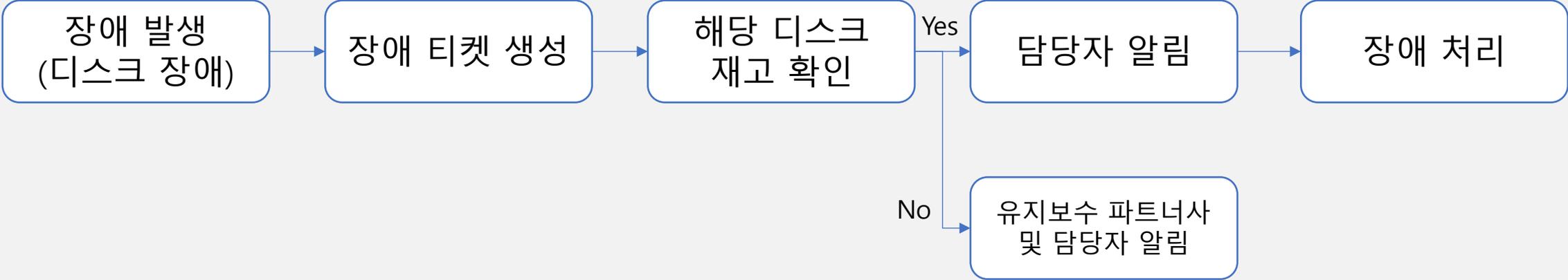
시스템

- 서버 상태 모니터링
- Slurm API 데이터 전송
- 네트워크 상태 모니터링
- 스토리지 상태 모니터링
- 항온항습기 모니터링 제어
- 분전반 모니터링
- UPS 모니터링
- 과전압방지

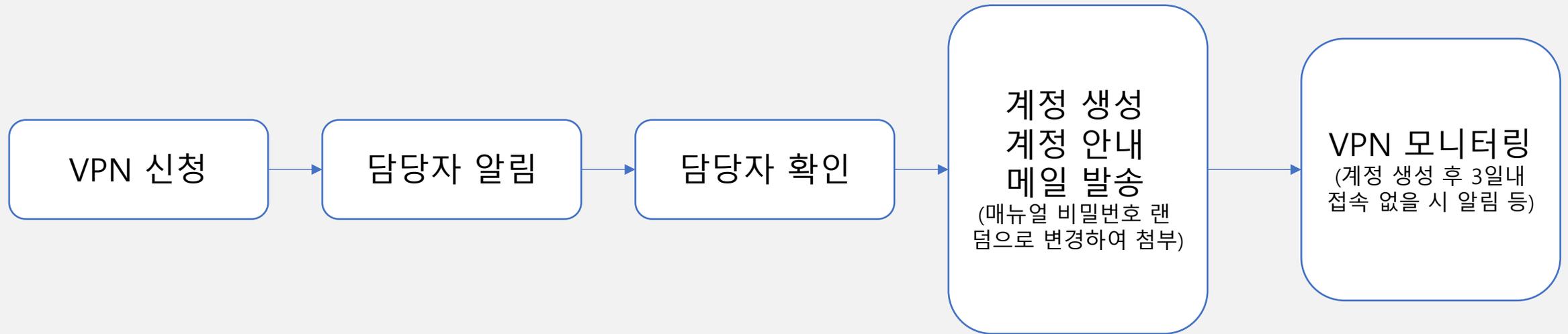
전체 구조도



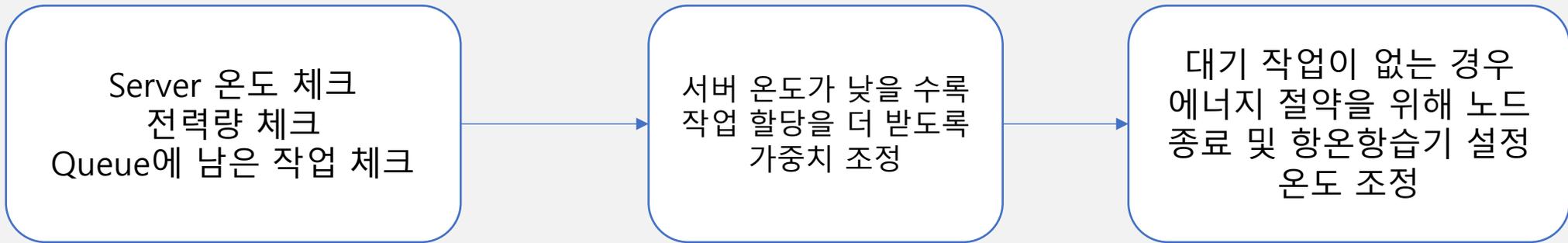
장애 처리 구조도



VPN 발급



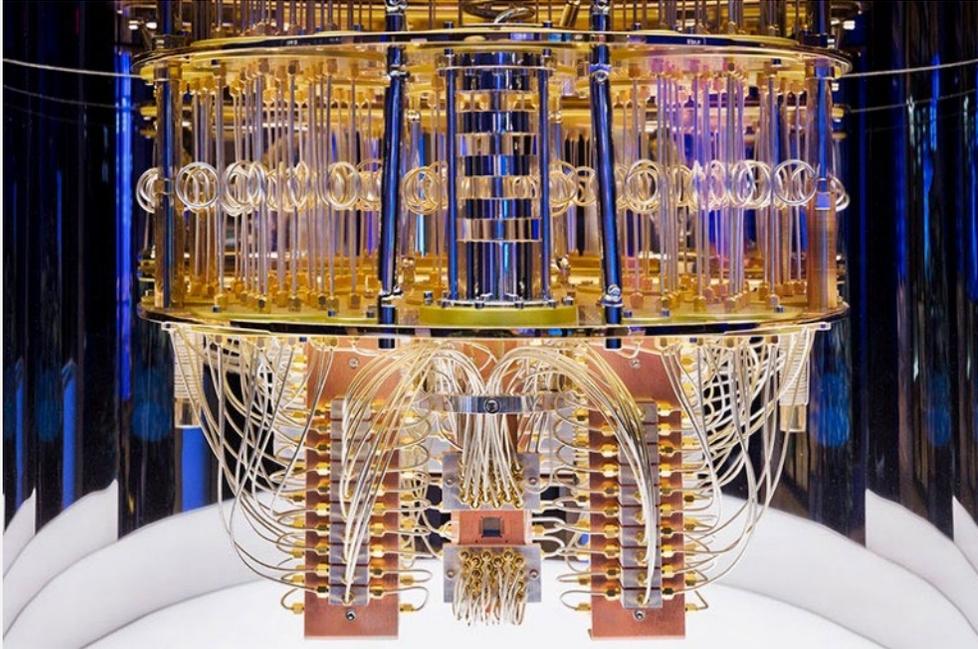
Slurm 노드 가중치 조정



그 밖에...



HPC의 미래



IBM quantum computer
The most quantum computers operate at temperatures between 10 millikelvins (-273.14°C) and 100 millikelvins (-273.15°C)

젠슨 황 "양자 컴퓨터 상용화, 수년 내 이뤄져"...아이온큐 추가↑

| 프랑스 파리 'GTC 개발자 컨퍼런스' 기조연설...지난해 전망치와 상이

컴퓨팅 | 입력 :2025/06/11 19:44 수정: 2025/06/12 09:34

<https://zdnet.co.kr/view/?no=20250611194203>

국가자격 종목별 상세정보

공조냉동기계기사

Engineer Air-Conditioning Refrigerating Machinery

관련부처 국토교통부

시행기관 한국산업인력공단



시험일정

공조냉동기계기사(원서접수시간은 원서접수 첫날 10:00부터 **마지막 날 18:00까지** 임)

감사합니다.