

오픈 테크넷 서밋 2024

Open Source Consulting Inc.

기상청 사례로 알아보는 OpenStack과 Kubernetes를 이용한 효율적인 GPU 활용방안

2024 . 9

발표자: 신호철(hcshin@osci.kr)



목차

Contents

- ✓ 자기소개
- ✓ 오픈소스컨설팅 소개
- ✓ 기상청에서 GPU 가상화의 필요성
- ✓ OpenStack, K8S GPU 가상화 구성
- ✓ Beyond 기상청 GPU 전략
- ✓ Q&A 및 마무리

자기소개



신호철

저는 오픈소스 기술에 대해서 깊은 열정을 가지고 오픈스택과 Ceph 스토리지 기술을 바탕으로 Cafe24에서 퍼블릭 클라우드 서비스 인프라 구축을 총괄하여 서비스를 구축, 운영해왔으며, 이를 통해 대규모 고객과 대규모 데이터를 처리하는 고성능 클라우드 인프라를 지속적으로 확장해 왔습니다.

현재는 [Open Source Consulting Inc.](#)에서 오픈소스 기술을 바탕으로 새로운 클라우드 서비스를 개발하고 있으며, 지속적인 혁신과 기술 발전에 제 열정을 하얗게 불태우고 있으며,

오픈인프라 한국 사용자 모임 & 한국 Ceph User Group 커뮤니티와 함께 성장하고 있습니다

경력 사항

- 2024. 오픈소스컨설팅
- 2007. 카페24
- 2006. 유비다임

대외 활동

- OpenInfra Korea User Group
- Korea Ceph User Group

기술

- OpenStack / Ceph
- K8S
- Pormethues/Grafana
- Jenkins / ArgoCD

오픈소스컨설팅 소개

오픈소스 컨설팅 소개

사업영역



Services

Linux/Rancher/HCI

- 오픈소스 기반 통합 리눅스 환경 구축 및 기술지원 서비스
- **SUSE Linux Enterprise Server**
- Kubernetes 관리 플랫폼 **Rancher**
- HCI 솔루션 **Harvester, Longhorn**
- 컨테이너 보안 솔루션 **NeuVector**

Private Cloud

- Kubernetes 기반의 클라우드 네이티브 환경 구축
- OpenStack, DevOps 컨설팅 구축 및 유지보수 서비스
- 프라이빗(IaaS/PaaS), 멀티 클라우드 유지보수 및 기술 지원

Solutions

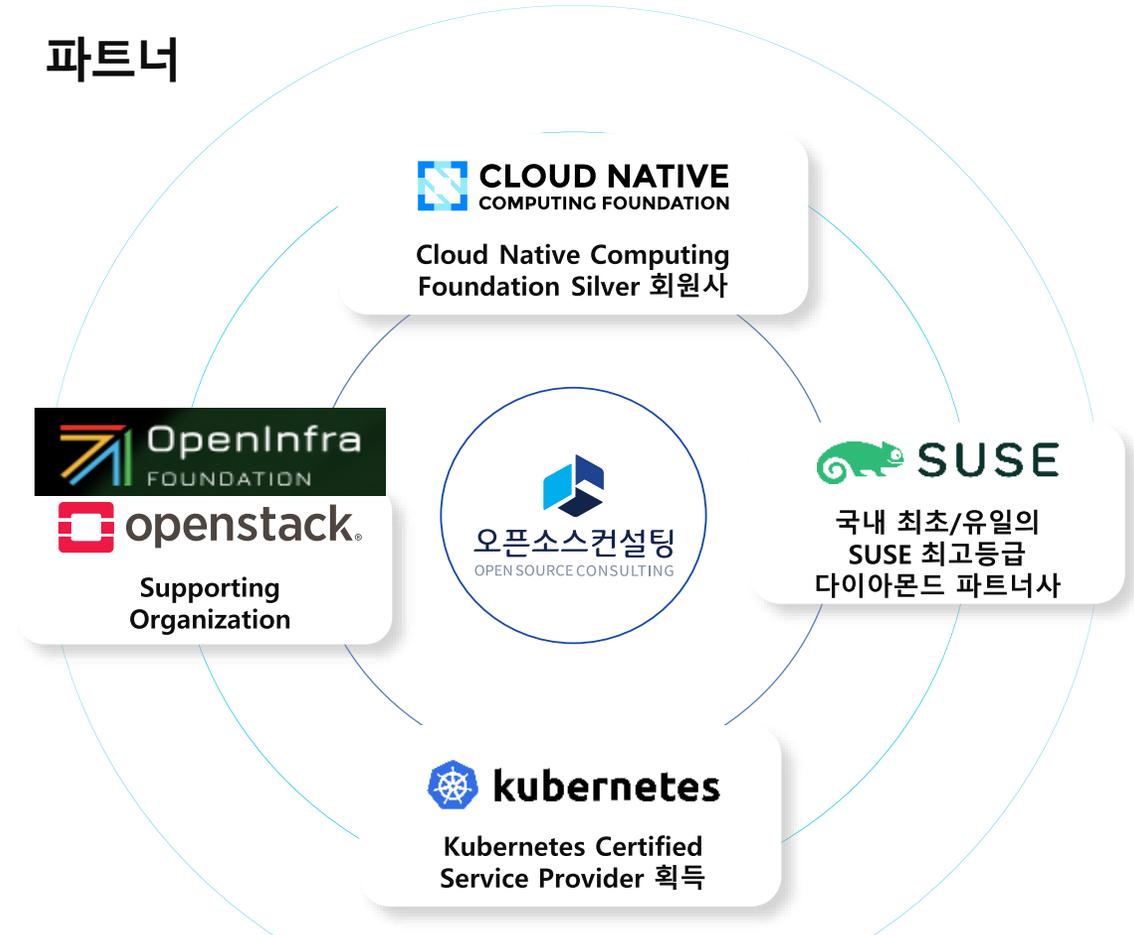
Playce OpenStack

- 100% 오픈소스 기반의 안정적인 All-in-One IaaS 플랫폼

Playce Kube

- 100% 오픈소스 기반의 안정적인 All-in-One PaaS 플랫폼

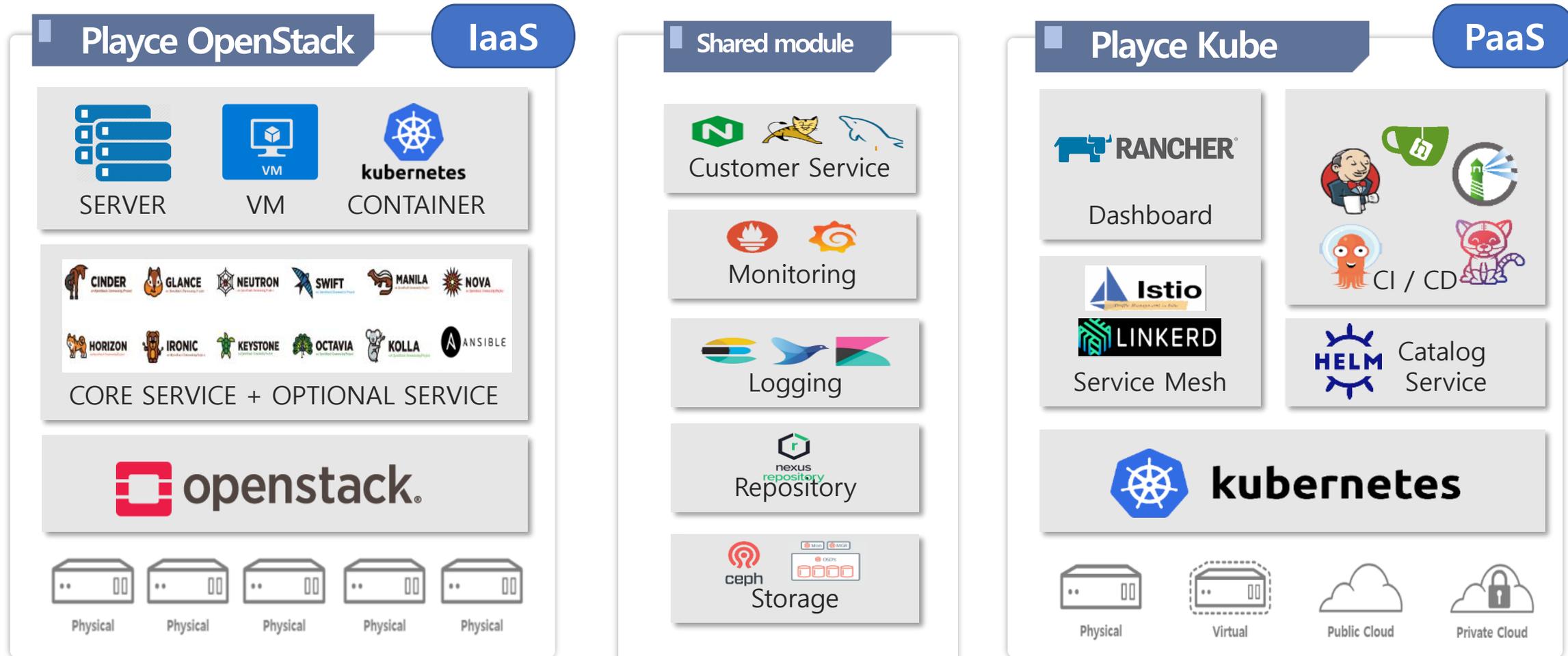
파트너



Sharing Technology.
Growing Together.

Playce Cloud 솔루션 개요 (Playce OpenStack + Playce Kube)

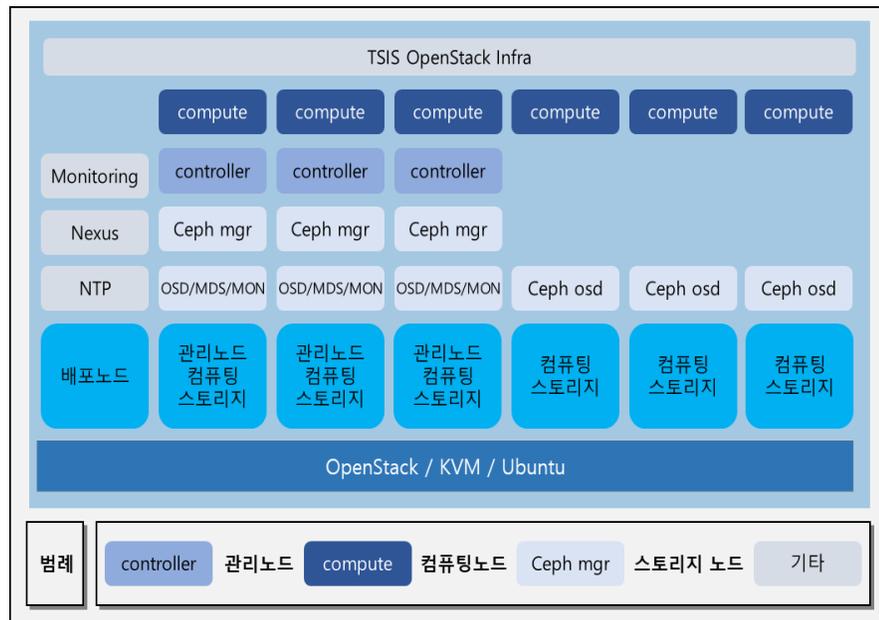
오픈소스를 패키징한 솔루션으로 벤더 종속성을 탈피한 컨설팅 / 구축 / 운영 / 장애 관리와 기술전수를 지원하는 All-in-One 솔루션



오픈스택 - 모든 인프라 기능을 cover 할 수 있는 SDDC All-Stack (from HCI to DataCenter)

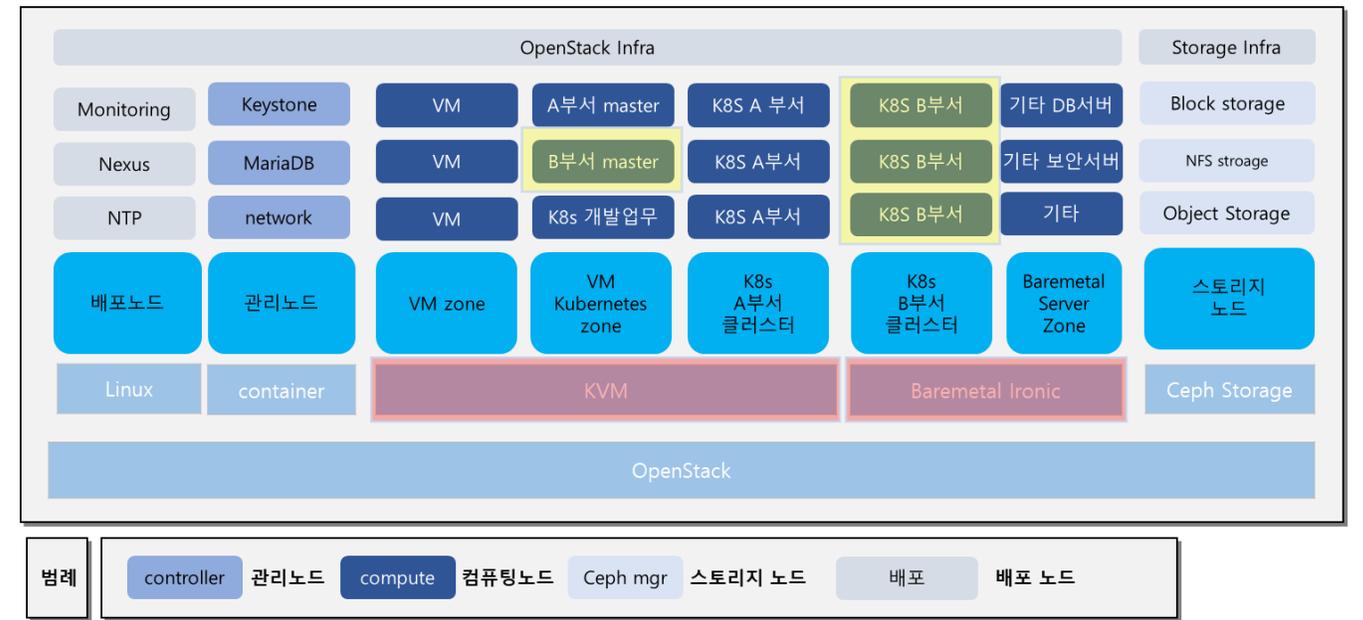
고객사 : T사 내부업무

인프라 구성	<ul style="list-style-type: none"> 서버 댓수 : 7 대 / 224 vCPU 스토리지 용량 : 98 TB
사용모듈	<ul style="list-style-type: none"> 서버 - SDC(Software Defined Computing) 스토리지 - SDS(Software Defined Storage) / SAN / NAS 모니터링 - Grafana / Prometheus



고객사 : 기상청 대외서비스

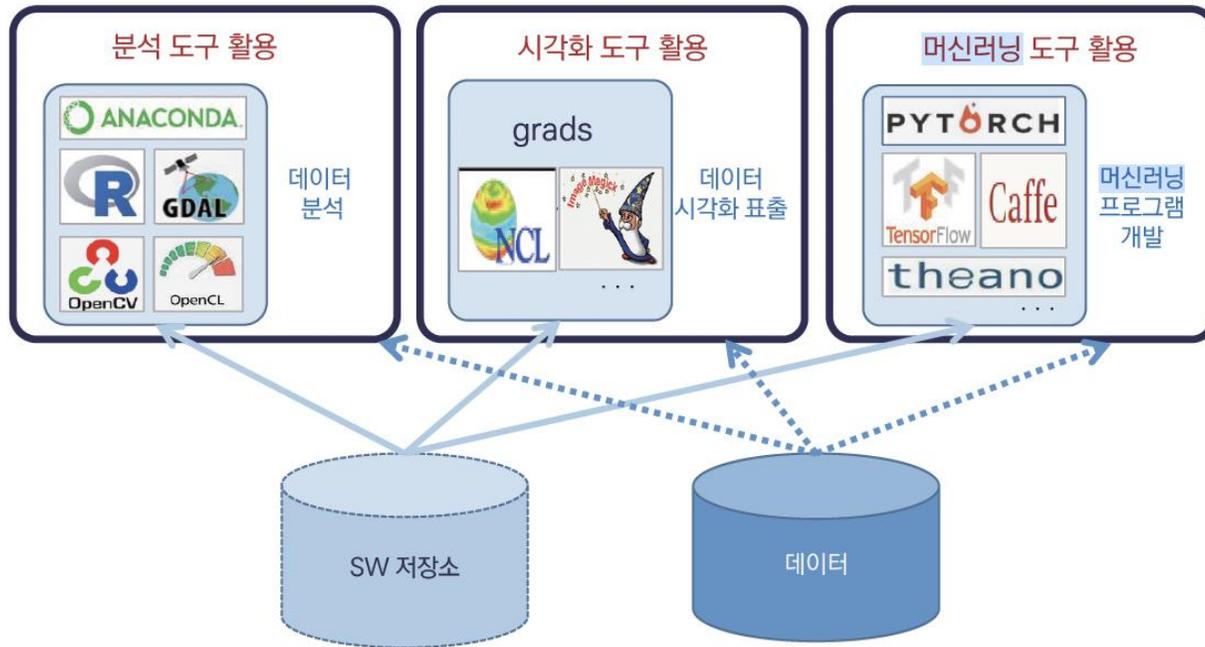
인프라 구성	<ul style="list-style-type: none"> 서버 댓수 : 400 여대 / 12,958 vCPU 스토리지 용량 : 6.7 PB
사용모듈	<ul style="list-style-type: none"> 서버 - SDC(Software Defined Computing) 스토리지 - SDS(Software Defined Storage) / SAN / NAS 네트워크 - SDN(Software Defined Storage) / LBAAS /DVR / FWAAS 모니터링 - Grafana / Prometheus / PaaS - Kubernetes / CICD / Repository / SSO 서울/오창 Active-Active DMZ/내부망 망분리요건 충족



기상청에서 GPU 가상화의 필요성

기상청 분석업무를 위한 GPU의 필요성

기상청은 예전부터 기상데이터 분석을 위한 다양한 분석도구를 활용하고 있었으며, 정확한 기상 예측을 위한 AI 도입 등 다양한 시도를 하고 있습니다.



[그림 4-3-49] K-DASH 데이터 분석 환경

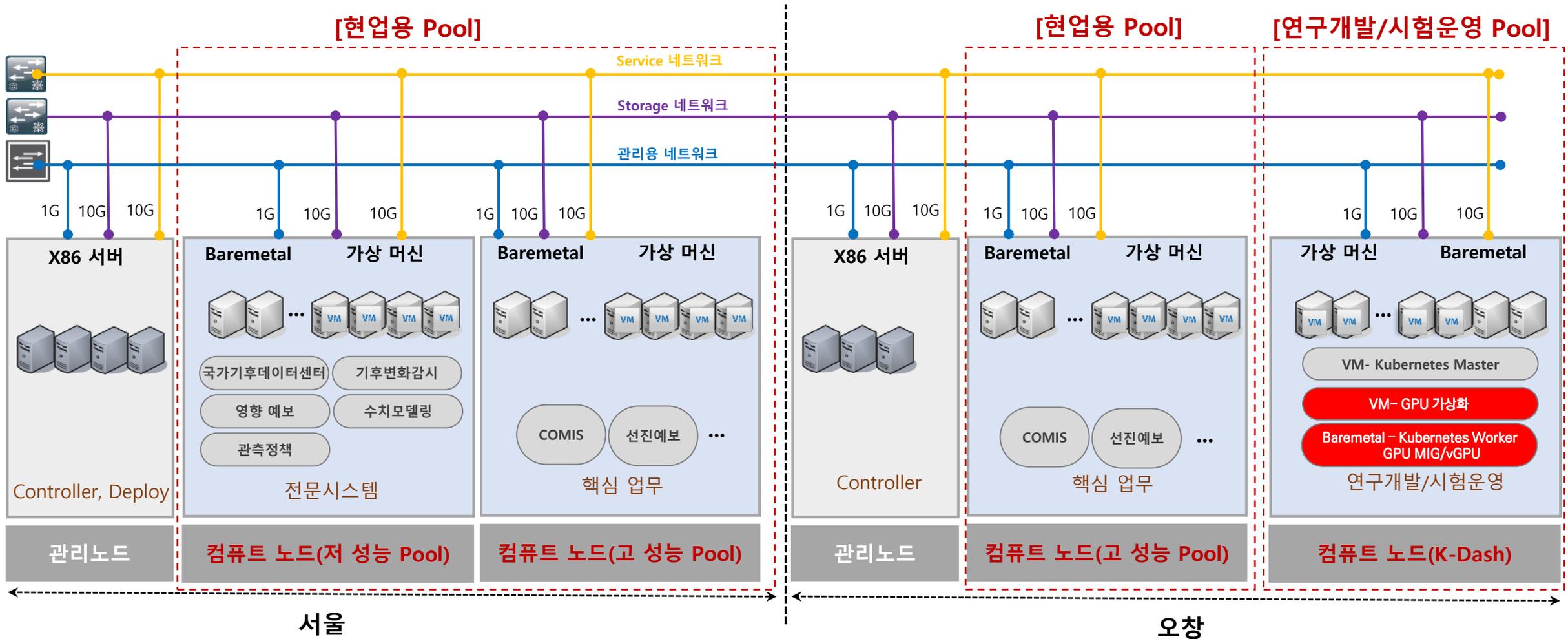
그림 출처 : 종합기상정보시스템 구축백서 <https://www.nanet.go.kr/main.do>



기사 출처 : <https://m.dongascience.com/news.php?idx=45260>

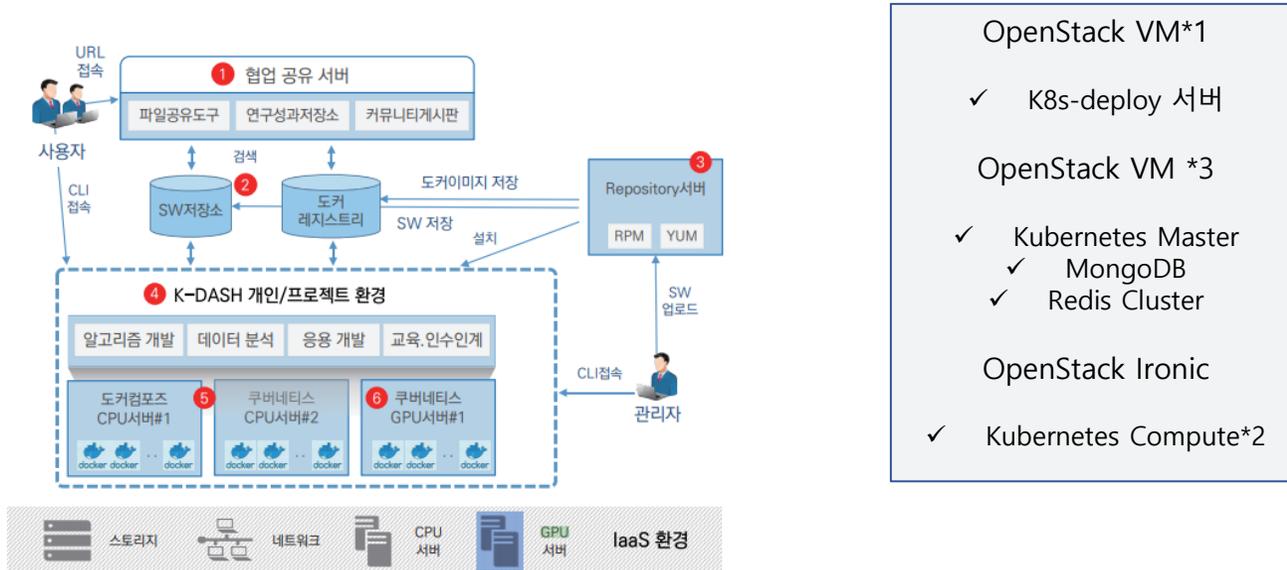
기상청의 오픈스택 구성

오픈스택 VM, Baremetal, K8S기반에 GPU 아키텍처



K-DASH - 초기 구축

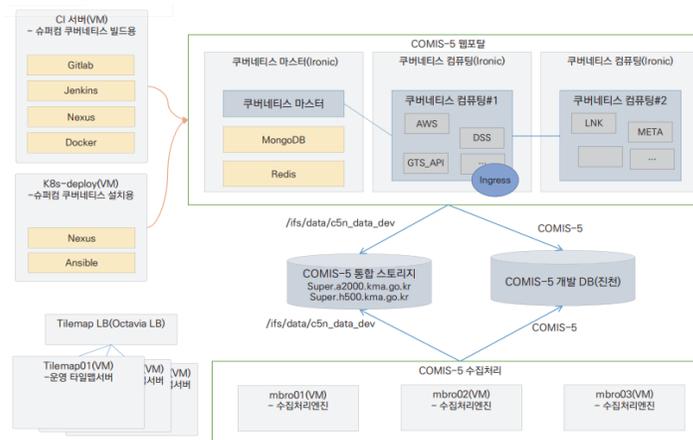
데이터 기반의 과학적 기상업무를 지원하기 위해 청 내에 분산된 데이터와 소프트웨어를 통합적으로 활용할 수 있도록 제공하는 플랫폼



- OpenStack VM*1
 - ✓ K8s-deploy 서버
- OpenStack VM *3
 - ✓ Kubernetes Master
 - ✓ MongoDB
 - ✓ Redis Cluster
- OpenStack Ironic
 - ✓ Kubernetes Compute*2

GPU에 대한 여러 요청사항

- ① 협업공유 서버에 파일공유도구, 연구성과저장소 및 커뮤니티 게시판 구축
- ② NAS에 SW저장소, 도커 레지스트리 구성
- ③ 망 분리가 적용된 기상청의 특성을 감안하여, 사용자가 설치패키지 매니저(rpm, yum 등)로 SW를 설치 시, 내부망 SW 저장소에서 다운로드 되도록 Repository 서버 구성
- ④ K-DASH 개인/프로젝트 환경은 알고리즘 개발, 데이터 분석 등 작업이 가능하도록 구성
- ⑤ CPU서버 2대에 쿠버네티스와 도커 컴포즈를 각각 설치하여 제공
- ⑥ 머신러닝 개발을 위하여 쿠버네티스를 구축하고, 그 노드에 GPU서버에 NVIDIA를 설치하여 제공

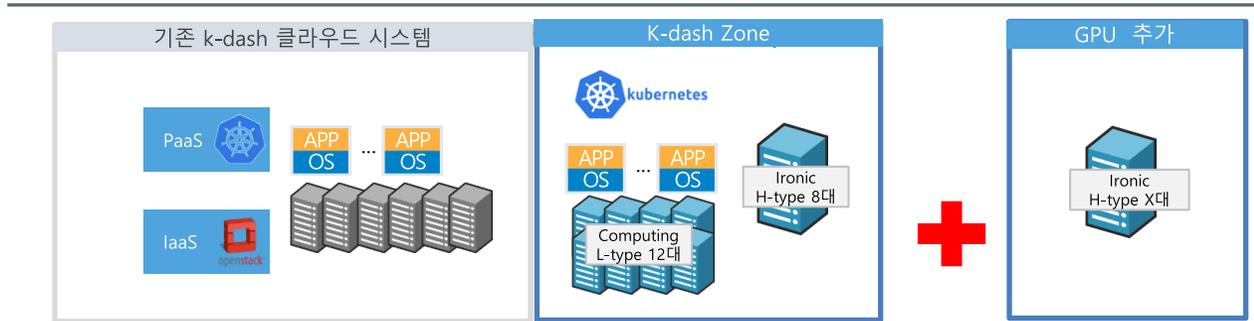


K-DASH - 확장

데이터 기반의 과학적 기상업무를 지원하기 위해 청 내에 분산된 데이터와 소프트웨어를 통합적으로 활용할 수 있도록 제공하는 플랫폼

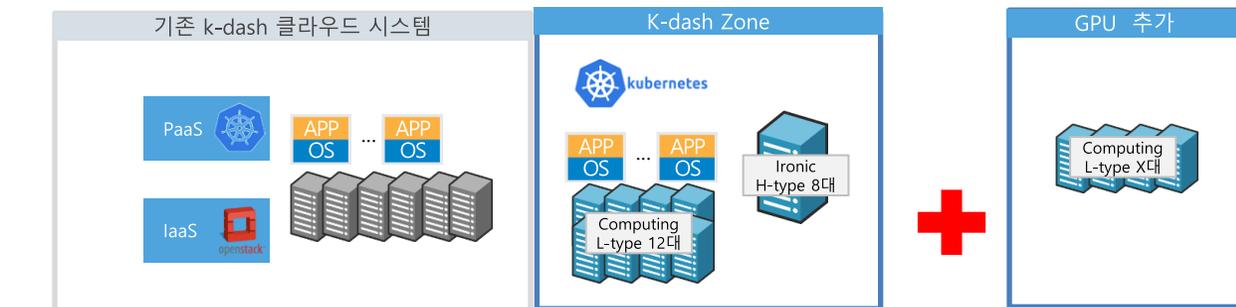
K-DASH SDC 증설 구성도

1안



K-DASH SDC 증설 구성도

2안



GPU 사용이 지속적으로 증가

- 날씨 예측 및 모델링에서 고성능 컴퓨팅의 필요성이 증가

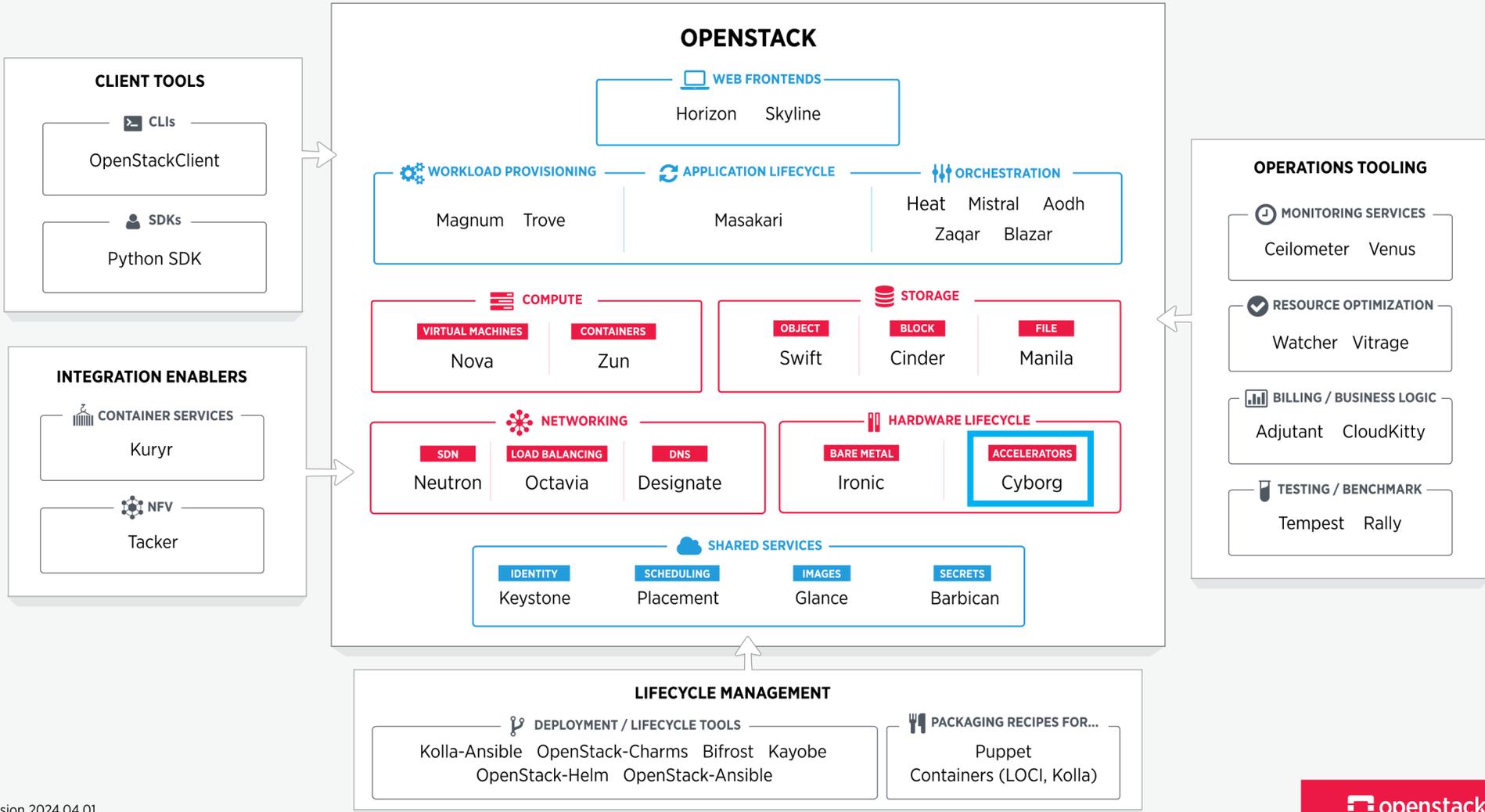
- GPU 서버가 머신러닝, 데이터 분석, 알고리즘 개발과 같은 작업에서 CPU 서버보다 빠르게 대용량 날씨 데이터를 처리하는 데 사용됨.

- 특히 기상청이 머신러닝과 데이터 분석 작업의 성능 향상을 위해 GPU 서버를 지속적으로 확장됨

OpenStack & K8S 기반 GPU 가상화 구성

- 1. Compute node PCI Passthrough (Baremetal)**
- 2. Compute node vGPU**
- 3. K8S vGPU**

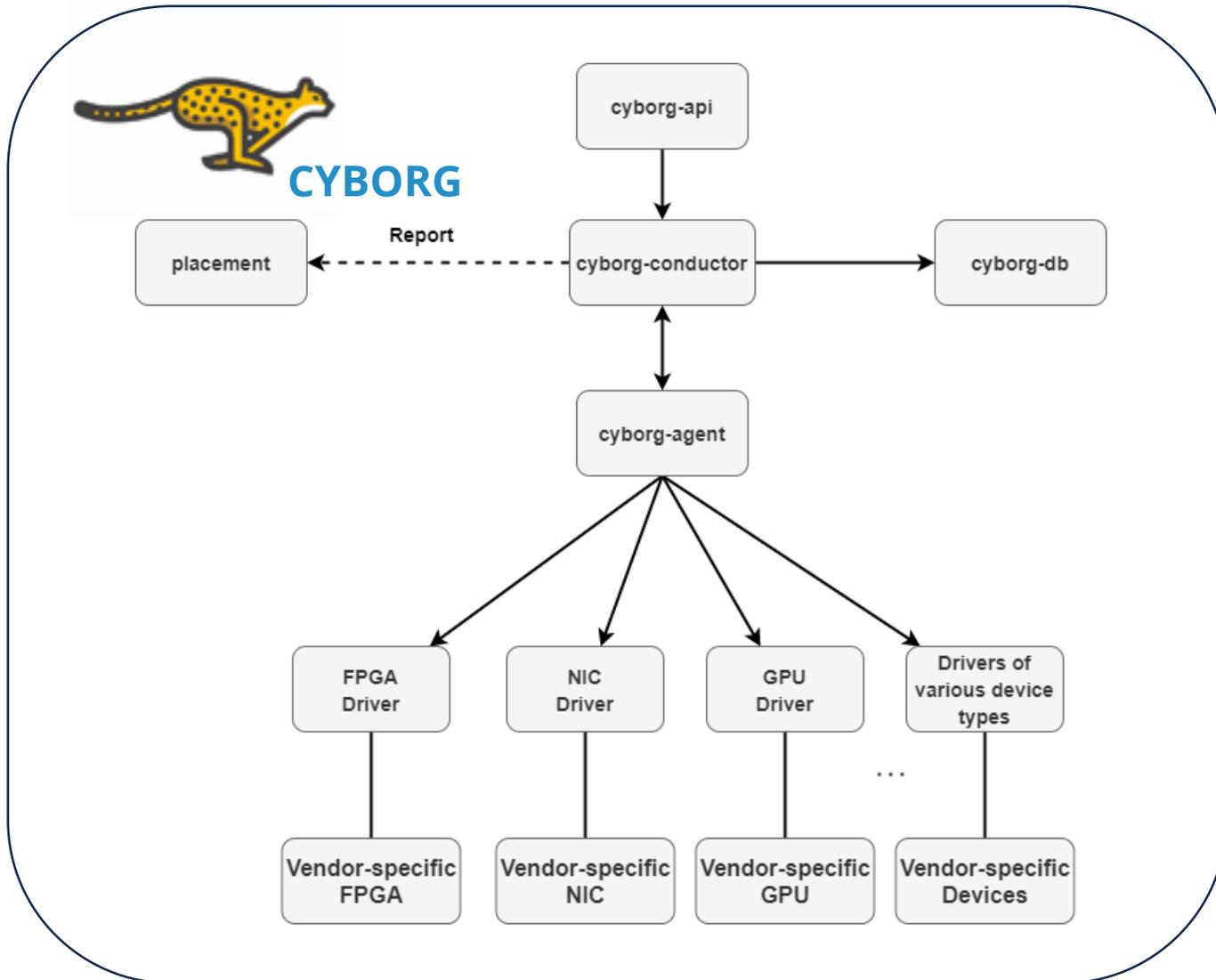
OpenStack landscape



Version 2024.04.01



OpenStack Cyborg-api



Cyborg 아키텍처 설명

1. Cyborg API:

- 사용자나 상위 시스템이 Cyborg에 접근해 가속기와 관련된 요청을 보내는 인터페이스.

2. Cyborg Conductor:

- Cyborg 시스템의 중추 역할을 하며, 다양한 에이전트로부터 수집된 정보를 기반으로 가속기 상태를 관리하고 업데이트.

3. Cyborg Agent:

- 각 하드웨어 장치(GPU, FPGA, NIC 등)에서 동작하며, 해당 장치의 상태를 모니터링하고 드라이버를 통해 제어.

4. 드라이버 레벨:

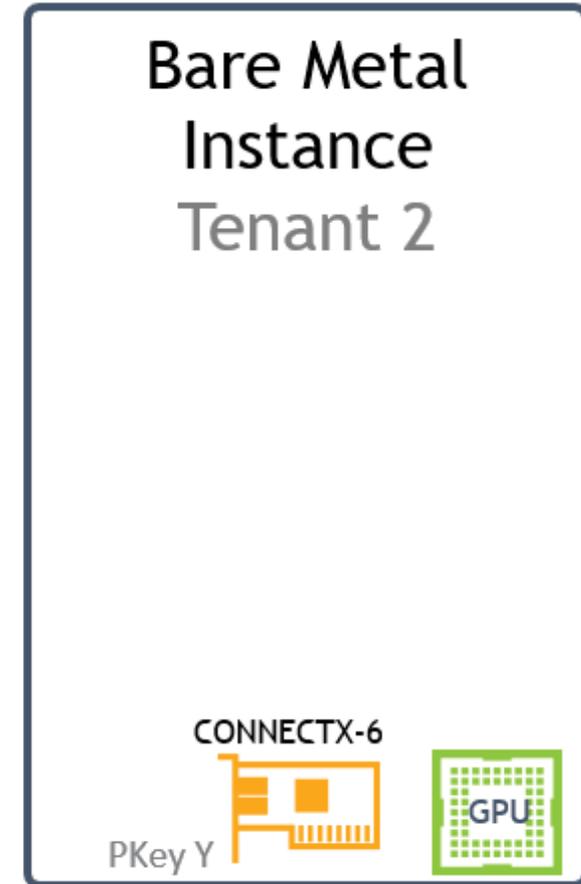
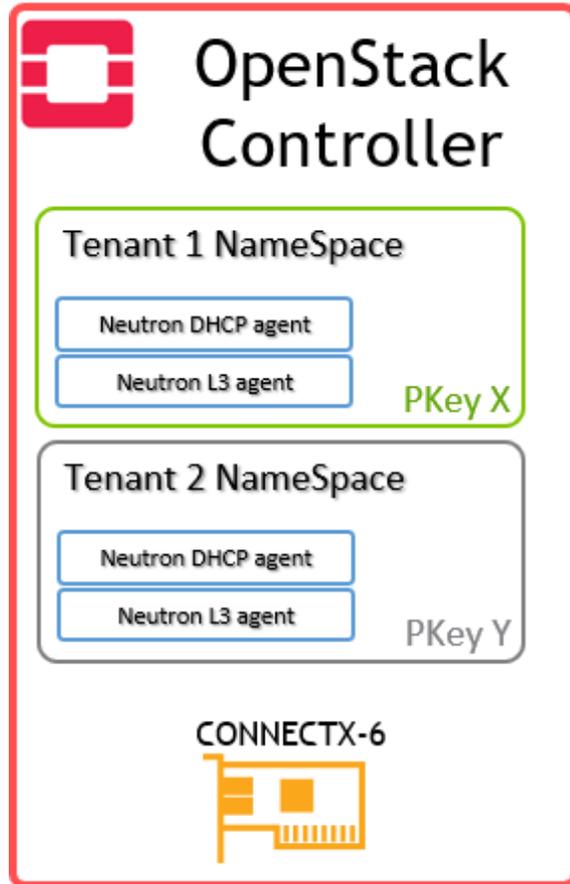
- 각각의 장치(예: **Vendor-specific GPU**, **Vendor-specific NIC**)에 맞는 드라이버가 제공되어 Cyborg 에이전트와 통신.

5. Placement와의 상호작용:

- **Placement** 서비스는 가속기의 자원 할당과 관련된 정보를 수집해 보고. 이를 통해 최적의 가속기 배치를 수행.

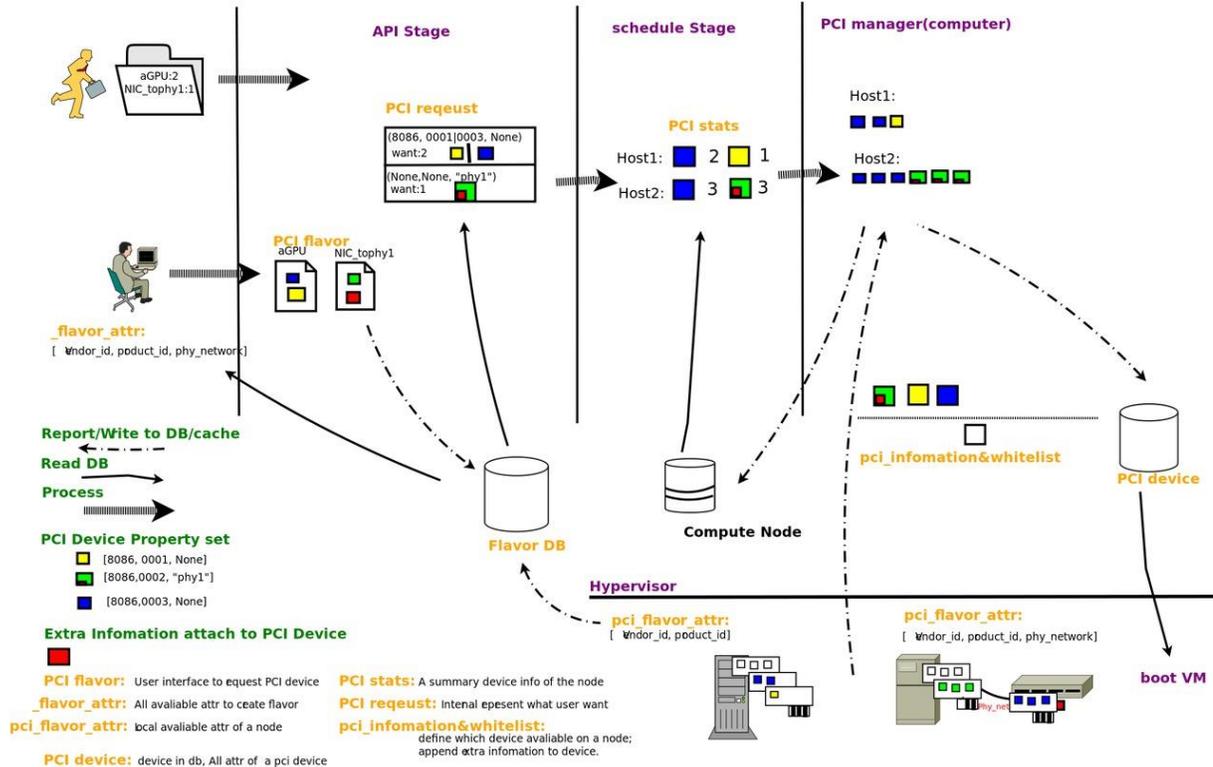
Compute node PCI Passthrough (Baremetal)

Baremetal은 가상화 계층 없이 물리 서버의 자원을 직접 사용하는 방식으로, OpenStack의 Ironic 프로젝트를 통해 지원됩니다



Compute node PCI Passthrough

OpenStack Compute에서의 PCI Passthrough 활용



① API 단계 (API Stage)

- 사용자는 **PCI 요청**을 통해 특정 PCI 장치 (예: aGPU, NIC 등)를 요청합니다.
- 이 요청은 장치의 vendor_id, product_id, 및 물리적 네트워크 정보와 함께 전달됩니다.

② 스케줄링 단계 (Schedule Stage)

- 스케줄러는 각 **호스트(Host)**의 PCI 장치 상태를 확인하고, 호스트가 처리할 수 있는 요청 수를 비교합니다.
- **PCI 통계(PCI Stats)**는 각 호스트가 제공할 수 있는 장치 수와 해당 장치의 상태를 나타냅니다.

③ PCI 관리 단계 (PCI Manager)

- 호스트의 PCI 장치는 pci_information 및 whitelist에 정의된 필터를 통과한 장치만 사용할 수 있습니다.
- 선택된 장치는 할당되어, 가상 머신(VM)에 전달될 준비를 합니다.

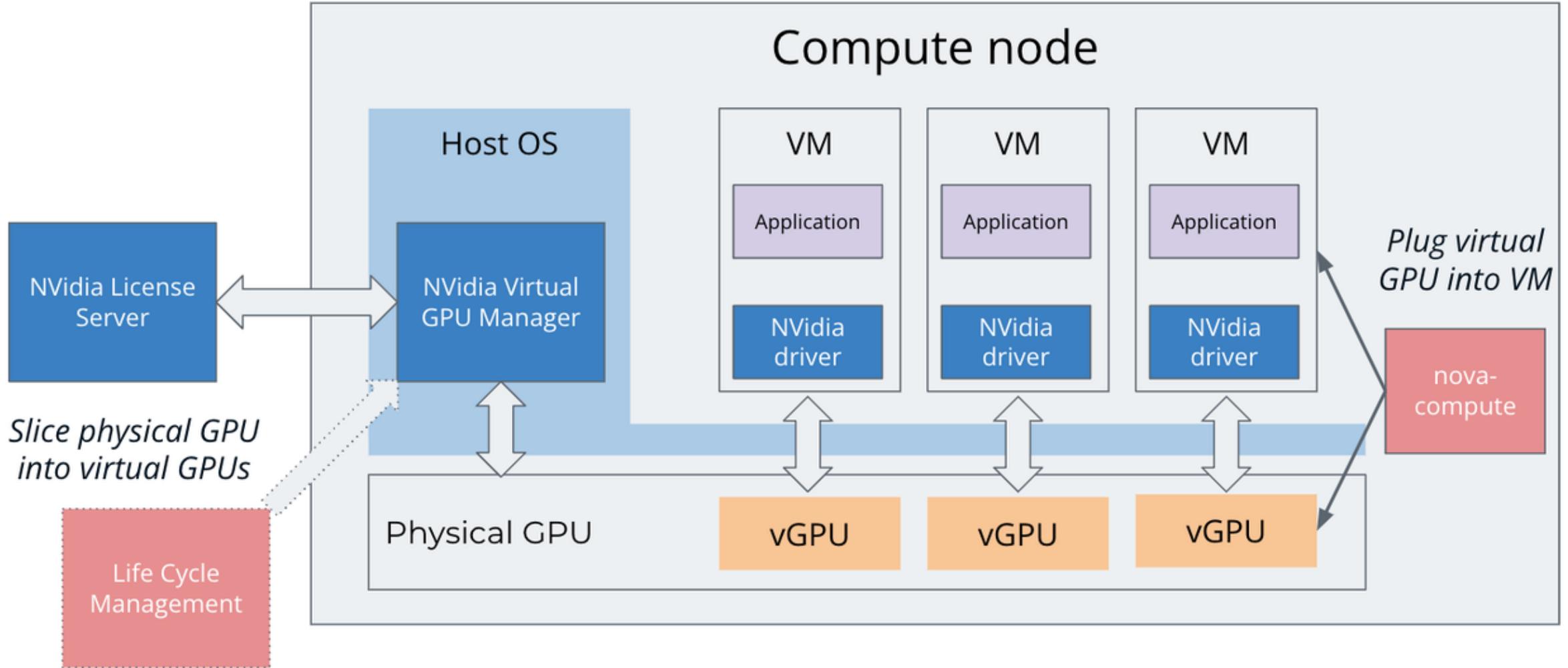
④ PCI 장치 할당 및 부팅

- 선택된 장치는 가상 머신이 시작되면서 할당됩니다.
- VM 부팅 시 필요한 모든 PCI 정보는 가상 머신과 연결됩니다.

⑤ 추가 정보 부착

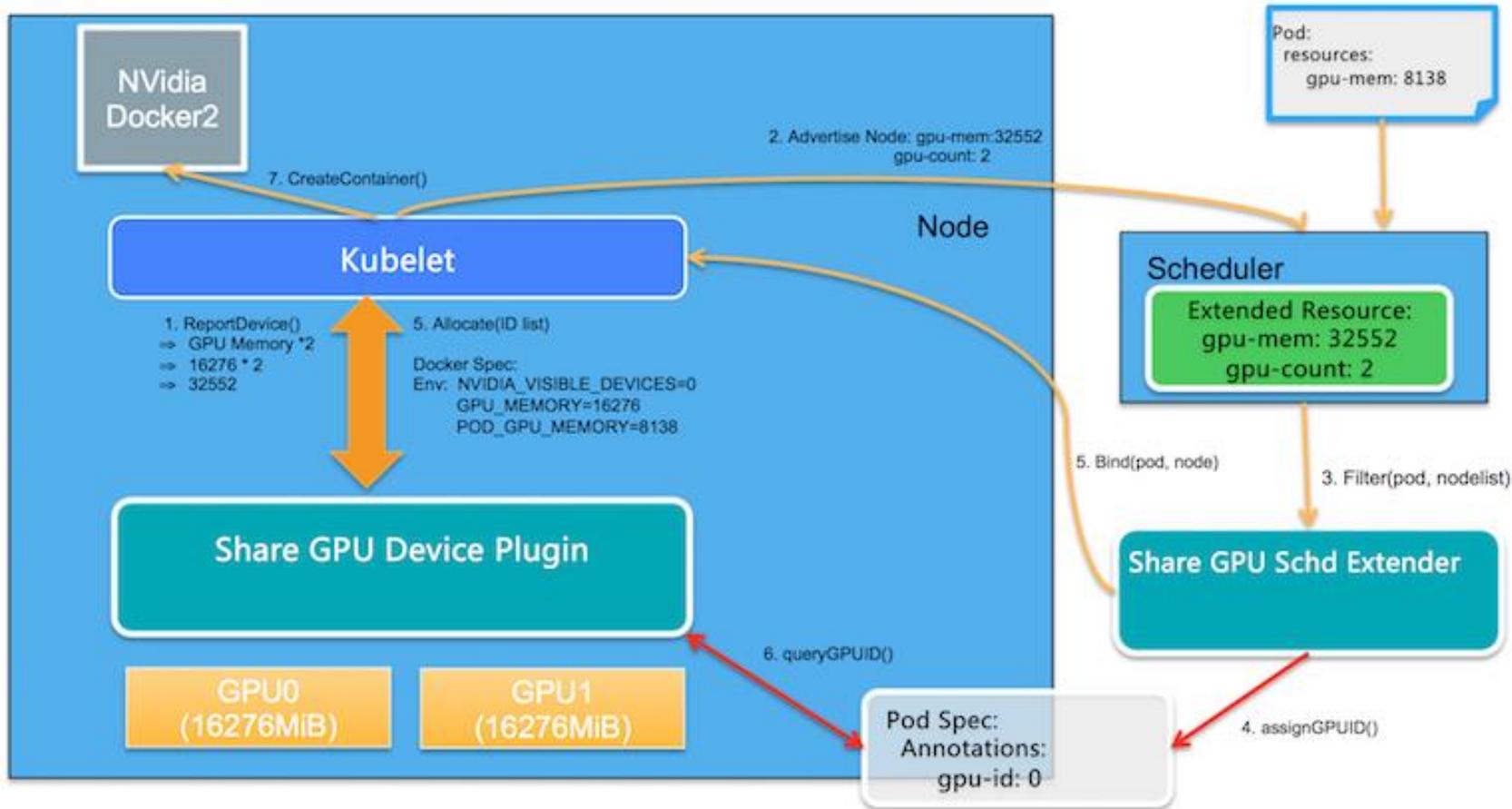
- SR-IOV와 같은 경우, PCI 장치에 추가적인 정보 (예: 네트워크 연결 정보)가 부착될 수 있습니다.

Compute node vGPU



K8S vGPU

Kubernetes Cluster node

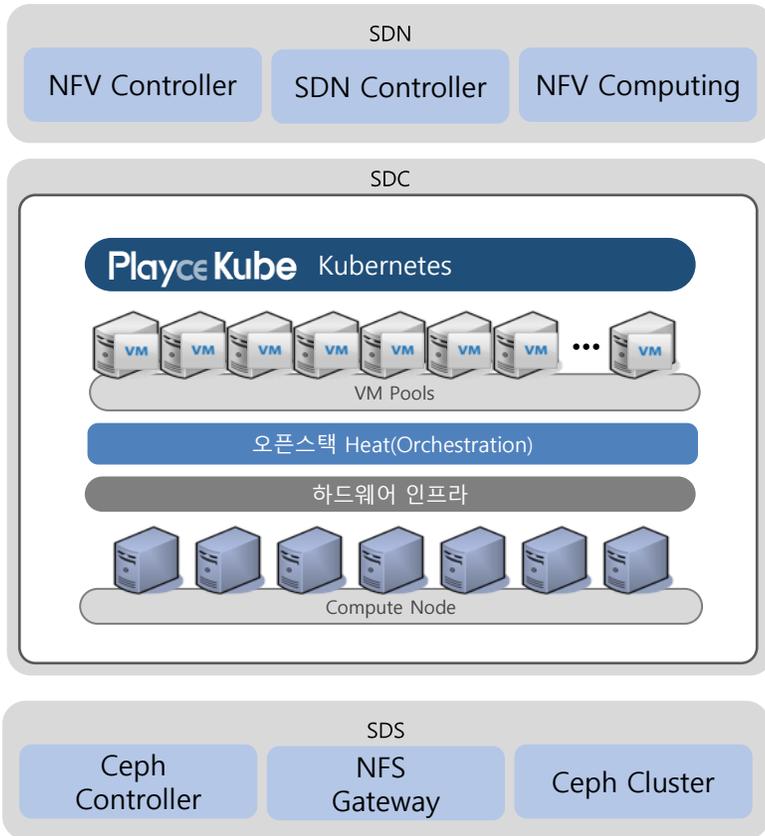


- **Kubelet**은 노드에서 GPU 자원을 관리 및 스케줄링
- **Scheduler**는 Pod의 GPU 리소스 요구 사항에 맞는 노드를 선택하고 GPU를 할당
- **GPU Device Plugin**은 GPU ID를 관리하며, 올바르게 할당된 GPU를 확인하고 처리

Beyond 기상청 GPU 전략

기상청에서 GPU의 여러가지 type을 고민하게 되는 환경 발생

쿠버네티스에서 GPU를 사용하고자 함 - 일부 쿠버네티스는 VM위에 있고, 일부 쿠버네티스는 ironic 베어메탈 위에 있음



VM위에 쿠버네티스

멀티 클러스터 쿠버네티스 구성

필요시 VM단에서 쉽게 확장 가능

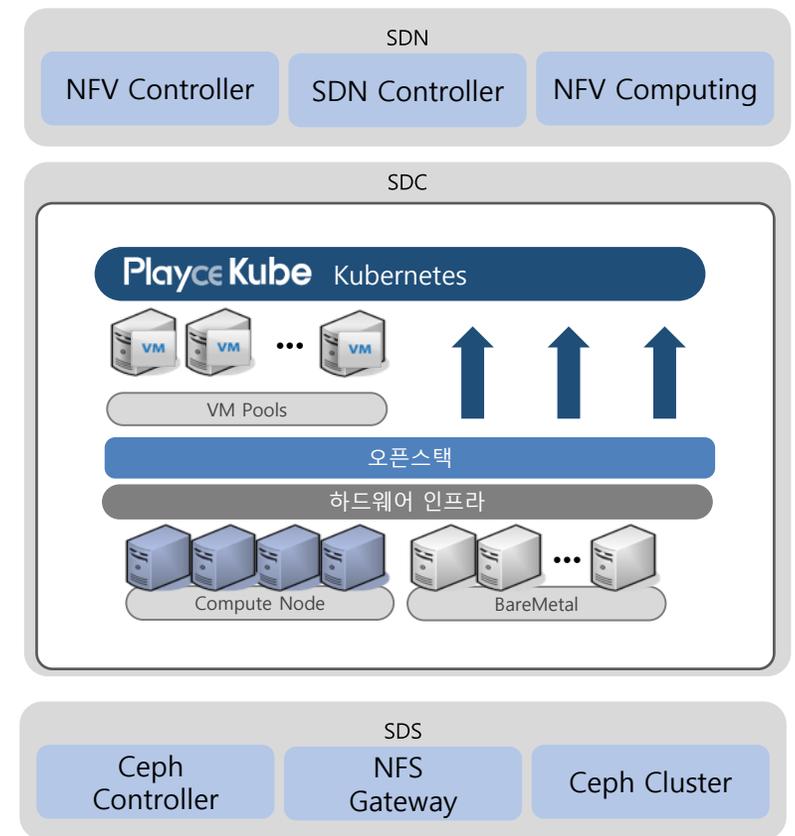
테넌트 내 VM과 연계하여 서비스 제공

BareMetal 서버위에 쿠버네티스

최적성능을 위해 Kubernetes worker 구성

장애시 오픈스택에서 배포/구성 가능

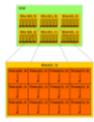
오픈스택의 NFV 기능인 Octavia LoadBalancer 사용



기상청에 필요한 GPU 파티셔닝 전략은?

GPU “CONCURRENCY”

Choices



Single Process
in CUDA



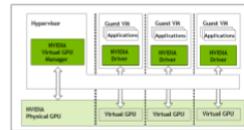
Multi-Process
with CUDA MPS



Time-slicing



MIG



Virtualization with vGPU

Application level

(using the CUDA programming
model APIs - CUDA streams)

GPU System Software / Hardware
(Mostly transparent to CUDA applications)

1. CUDA Streams

- 비동기 실행 모델로 여러 작업(커널, 메모리 복사 등)을 동시에 처리.
- 두 개 이상의 스트림에서 병렬 실행 가능.
- GPU 자원 활용 극대화: 추론 서빙, 여러 모델 병렬 실행.
- 단점: 단일 애플리케이션 내에서만 사용 가능, 하드웨어 격리 제한.

2. CUDA MPS (Multi-Process Service)

- 여러 프로세스의 CUDA 커널을 병렬로 실행하여 GPU 자원 활용 극대화.
- 타임 슬라이싱과 다르게 병렬 실행 가능.
- 메모리 할당 및 스레드 사용률 설정 가능.
- 단점: 에러 격리, 메모리 보호, QoS 제한.

3. 타임 슬라이싱 (Time-Slicing)

- 여러 CUDA 애플리케이션이 GPU 리소스를 다 활용하지 못할 때 시간 분할 방식
- Pascal 아키텍처부터 지원, 컨텍스트 스위칭 비용 발생.
- 단점: 지연 시간 증가, OOM 발생 가능성.
- nvidia-smi를 통해 타임 슬라이스 설정 가능.

4. MIG (Multi-Instance GPU)

- Ampere 아키텍처 기반 GPU를 최대 7개의 인스턴스로 보안 파티셔닝.
- 프로세스, 컨테이너, VM에 격리된 GPU 리소스 제공.
- 각 인스턴스에서 CUDA 스트림, MPS, 타임 슬라이싱 사용 가능.

5. vGPU (Virtual GPU)

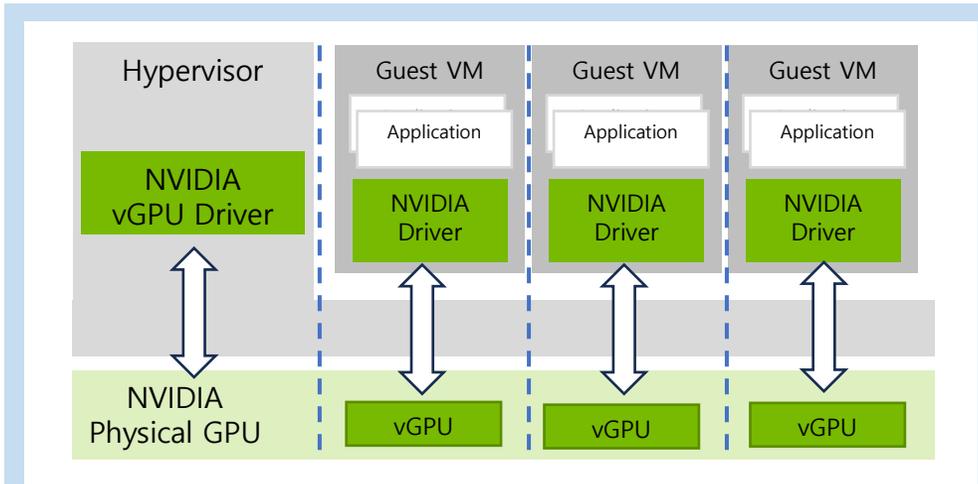
- 가상 머신(VM)이 단일 GPU에 동시 접근할 수 있도록 지원.
- IOMMU 보호 및 라이브 VM 마이그레이션 기능 제공.
- MIG 지원 시, VM 간 병렬 실행 가능.

<https://developer.nvidia.com/blog/improving-gpu-utilization-in-kubernetes>

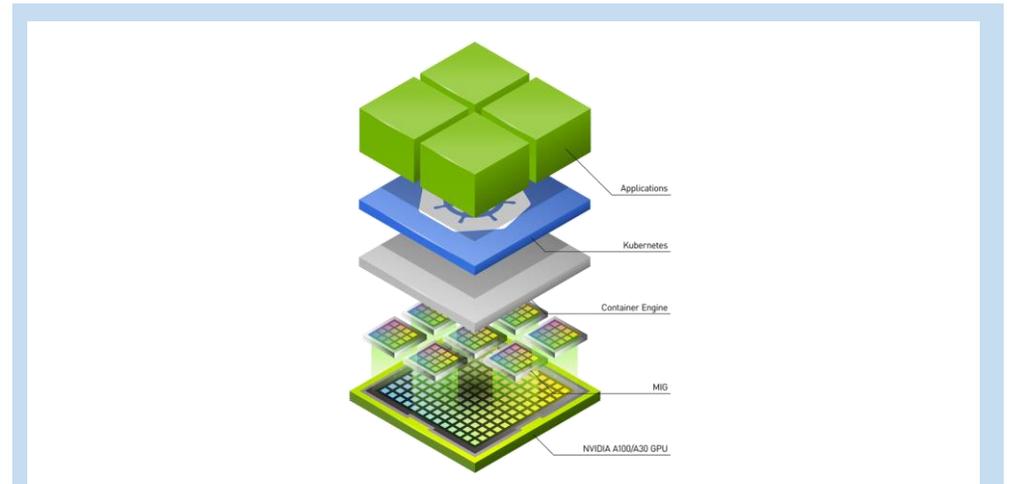
기상청에 필요한 GPU 파티셔닝 전략은?

GPU 파티셔닝을 위한 MIG와 vGPU는 작업 환경과 업무에 맞도록 설계 필요

vGPU



MIG



주요 특징

유연한 자원 공유, 하이퍼 바이저 통합, 동적인 자원 할당

고성능 유지, 고격리성(QOS), 최적화된 자원 할당

적용 업무

데스크톱 가상화(VDI), GPU 활용이 필요한 대량의 VM 활용, 다양한 워크로드(3D 렌더링, AI 작업 등)

AI 추론, 병렬 처리, 고성능 요구 환경

워크로드 종류, 리소스 격리 필요 여부, 유연성을 고려하여 파티셔닝 전략 선택

효율적인 GPU 자원활용 방법 비교

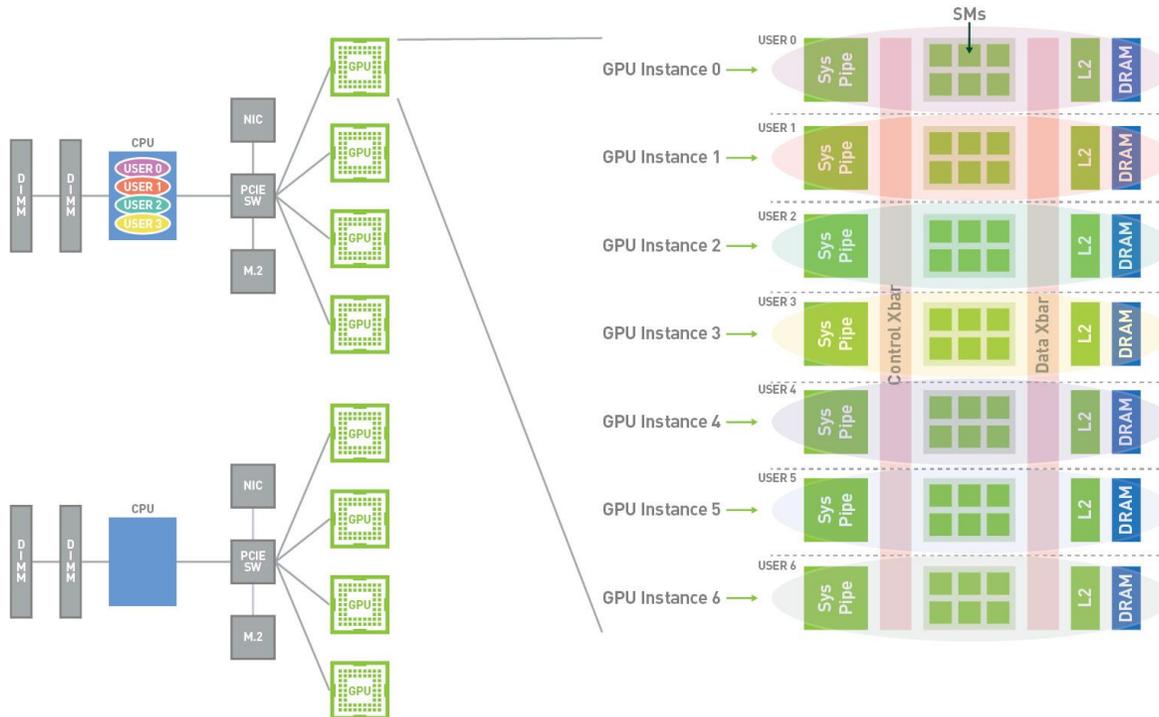
MIG (Multi-Instance GPU) vs vGPU

구분	MIG		vGPU	
	GPU	Memory	GPU	Memory
자원 분할 단위(최소)	1 Instance	12 GB	1 vGPU	4 GB
자원 분할 단위(최대)	7 Instance	94 GB	23 vGPU	94 GB
VM에 자원 할당	Y	Y	Y	Y
Container에 지원 할당	Y	Y	N	N
Bare Metal에 자원 할당	Y	Y	N	N
Over Provisioning 가능	N	N	N	N
장점 (권고 환경)	<ul style="list-style-type: none"> ✓ 최대 7개의 독립적인 Instance로 파티셔닝이 가능하며, 여러 작업을 동시에 수행가능, ✓ 각 인스턴스는 고유 Memory와 Core를 사용하므로 리소스 활용도가 높아짐, ✓ 여러 GPU를 구매하는 대신, 하나의 GPU를 여러 Instance로 나누어 사용 가능하므로 비용절감이 가능 		<ul style="list-style-type: none"> ✓ GPU의 메모리를 정적으로 분할하며, 컴퓨팅은 시분할 방식으로 유연하게 할당할수 있음 ✓ 다양한 워크로드 지원 ✓ 단일 GPU를 여러 VM에서 공유 할수 있음 	
단점 (제약 사항)	<ul style="list-style-type: none"> ✓ 고정된 리소스 분할로 유휴 자원이 발생할 수도 있음 ✓ 여러 인스턴스를 관리해야 하므로 관리에 신경을 써야 함 		<ul style="list-style-type: none"> ✓ 여러 VM이 동시에 GPU를 사용할 경우, 성능에 영향이 있을수 있음 ✓ 초기 설정이 복잡할수 있으며, 세심한 조정이 필요 	
Best Practice (Use Case)	<ul style="list-style-type: none"> ✓ 적절한 Instance 크기가 필요한 업무 ✓ 여러 워크로드를 독립적인 Instance로 분리하여 성능 간섭을 최소화 해야 하는 업무 ✓ Deep Learning, Inference, HPC 등 		<ul style="list-style-type: none"> ✓ 워크로드 특성에 맞는 Profile을 선택하는 업무 ✓ VDI, Graphics, Data Science, 원격작업 	

MIG(Multi-Instance GPU)

MIG (Multi-Instance GPU)

MULTI-INSTANCE GPU ("MIG")



<https://docs.nvidia.com/datacenter/tesla/mig-user-guide/>

MIG란 무엇인가?

- MIG는 하나의 GPU를 여러 인스턴스로 분할하여 사용할 수 있는 기술
- 각 인스턴스는 물리적으로 분리되어 있으며, 독립적으로 자원을 할당받아 사용
- 이 기술을 사용하면 GPU 자원을 더 효율적으로 사용할 수 있습니다.
예) 큰 자원이 필요하지 않은 작업에 전체 GPU를 할당하지 않고, 필요한 만큼만 자원을 나누어 사용

2. 왜 최대 7개의 인스턴스가 가능한가?

- A100 GPU에서 MIG는 물리적인 자원(SMs, 메모리, 대역폭 등)을 7개의 독립적인 GPU 인스턴스로 나눌 수 있음
- SMs(Scheduler Multiprocessors): GPU의 연산 자원
- L2 캐시 및 DRAM: 메모리 자원
- MIG의 하드웨어 아키텍처는 각 인스턴스가 일정한 자원을 차지하게 설계되어 있으며, 물리적 자원의 총량에 따라 최대 7개의 인스턴스로 나누는 것이 가장 효율적인 방식

3. 어떻게 자원이 나눠지는가?

- 각 GPU 인스턴스는 물리 GPU의 일정한 부분을 차지합니다. 사용자 0에서 사용자 6까지 각기 다른 GPU 인스턴스를 할당받으며, Sys Pipe, L2 캐시, DRAM 등의 자원이 분배
- 물리 GPU의 자원을 나누어도 성능 간섭 없이 독립적으로 사용할 수 있음
예) GPU 인스턴스 0은 GPU의 일정 부분을, GPU 인스턴스 1은 다른 부분을 사용하여 각각 독립적인 작업을 수행

NVIDIA vGPU와 MIG

GPU모델 종류

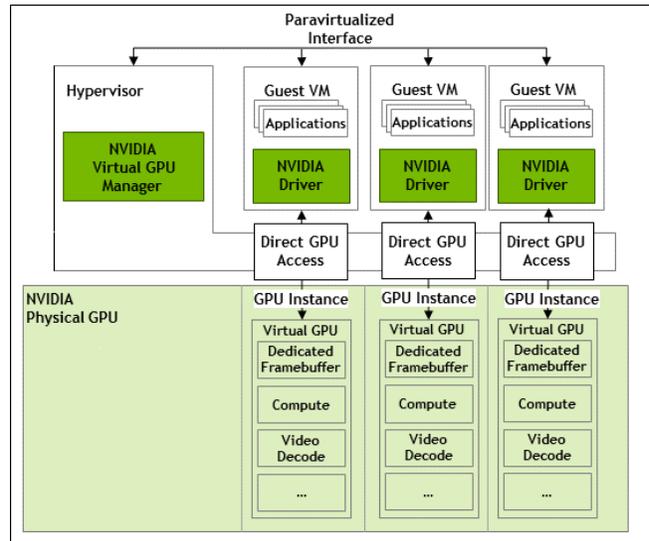
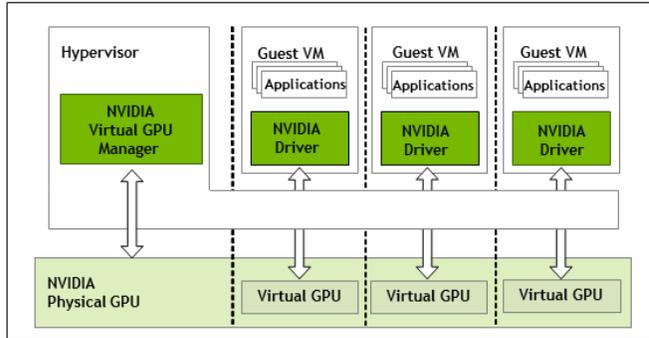
GPU 모델	Ampere 아키텍처	MIG 지원 여부	vGPU 지원 여부
NVIDIA A100	예	지원	지원
NVIDIA A30	예	지원	지원
NVIDIA H100	예	지원	지원
NVIDIA A40	예	미지원	지원
NVIDIA RTX A6000	예	미지원	지원
NVIDIA A10	예	미지원	지원
NVIDIA A16	예	미지원	지원
NVIDIA T4	아니요	미지원	지원
NVIDIA V100	아니요	미지원	지원

NVIDIA vGPU와 MIG

기상청 A100

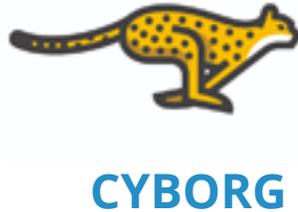
Config	GPC Slice #0	GPC Slice #1	GPC Slice #2	GPC Slice #3	GPC Slice #4	GPC Slice #5	GPC Slice #6	OFA	NVDEC	NVJPG	P2P	GPU Direct RDMA
1	7							1	5	1	No	Supported MemBW proportional to size of the instance
2	4			3				0	2+2	0	No	
3	4			2		1		0	2+1+0	0	No	
4	4			1	1	1		0	2+0+0+0	0	No	
5	3		3					0	2+2	0	No	
6	3		2		1			0	2+1+0	0	No	
7	3		1	1	1			0	2+0+0+0	0	No	
8	2		2		3			0	1+1+2	0	No	
9	2		1	1	3			0	1+0+0+2	0	No	
10	1	1	2		3			0	0+0+1+2	0	No	
11	1	1	1	1	3			0	0+0+0+0+2	0	No	
12	2		2		2		1	0	1+1+1+0	0	No	
13	2		1	1	2		1	0	1+0+0+1+0	0	No	
14	1	1	2		2		1	0	0+0+1+1+0	0	No	
15	2		1	1	1	1	1	0	1+0+0+0+0	0	No	
16	1	1	2		1	1	1	0	0+0+1+0+0+0	0	No	
17	1	1	1	1	2		1	0	0+0+0+0+1+0	0	No	
18	1	1	1	1	1	2		0	0+0+0+0+0+1	0	No	
19	1	1	1	1	1	1	1	0	0+0+0+0+0+0+0	0	No	

Next OpenStack Releases



항목	특징
vGPU (Virtual GPU)	<p>주요 특징:</p> <ul style="list-style-type: none"> • 여러 VM이 동시에 GPU 리소스를 사용할 수 있는 가상화 기술 • VM 간 GPU 리소스 공유로 자원의 효율적 활용 • IOMMU 보호 및 VM 라이브 마이그레이션 지원 • VDI 및 컴퓨팅 워크로드 혼합 환경에서 사용 가능
MIG (Multi-Instance GPU)	<p>주요 특징:</p> <ul style="list-style-type: none"> • 하나의 GPU를 다중 인스턴스로 분할하여 다양한 애플리케이션 지원 • 각 인스턴스가 전용 리소스를 갖추어 상호 간섭 없이 실행 • QoS 보장 및 격리된 환경에서 애플리케이션 실행 • CUDA 스트림, MPS, 타임슬라이싱을 각 인스턴스 내에서 적용 가능

OpenStack Release for MIG

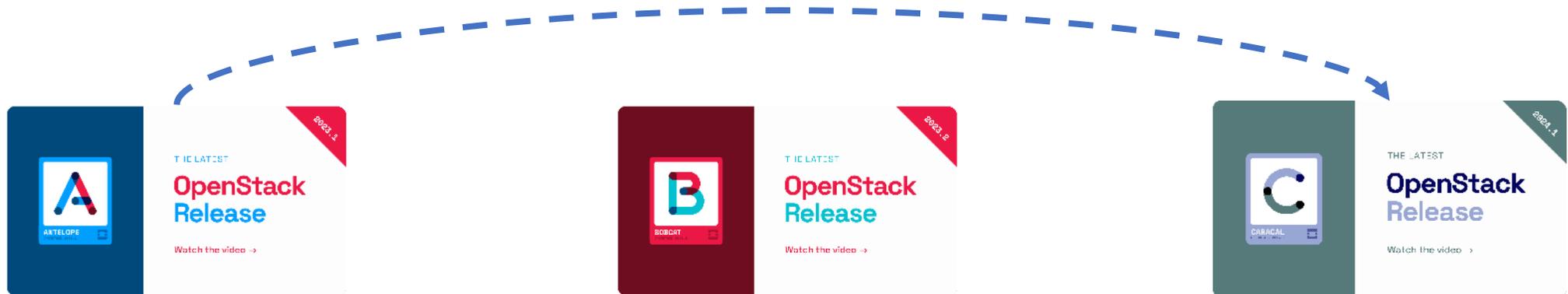


About this project
Cyborg provides a general purpose management framework for accelerators (including GPUs, FPGAs, ASIC-based devices, etc.)

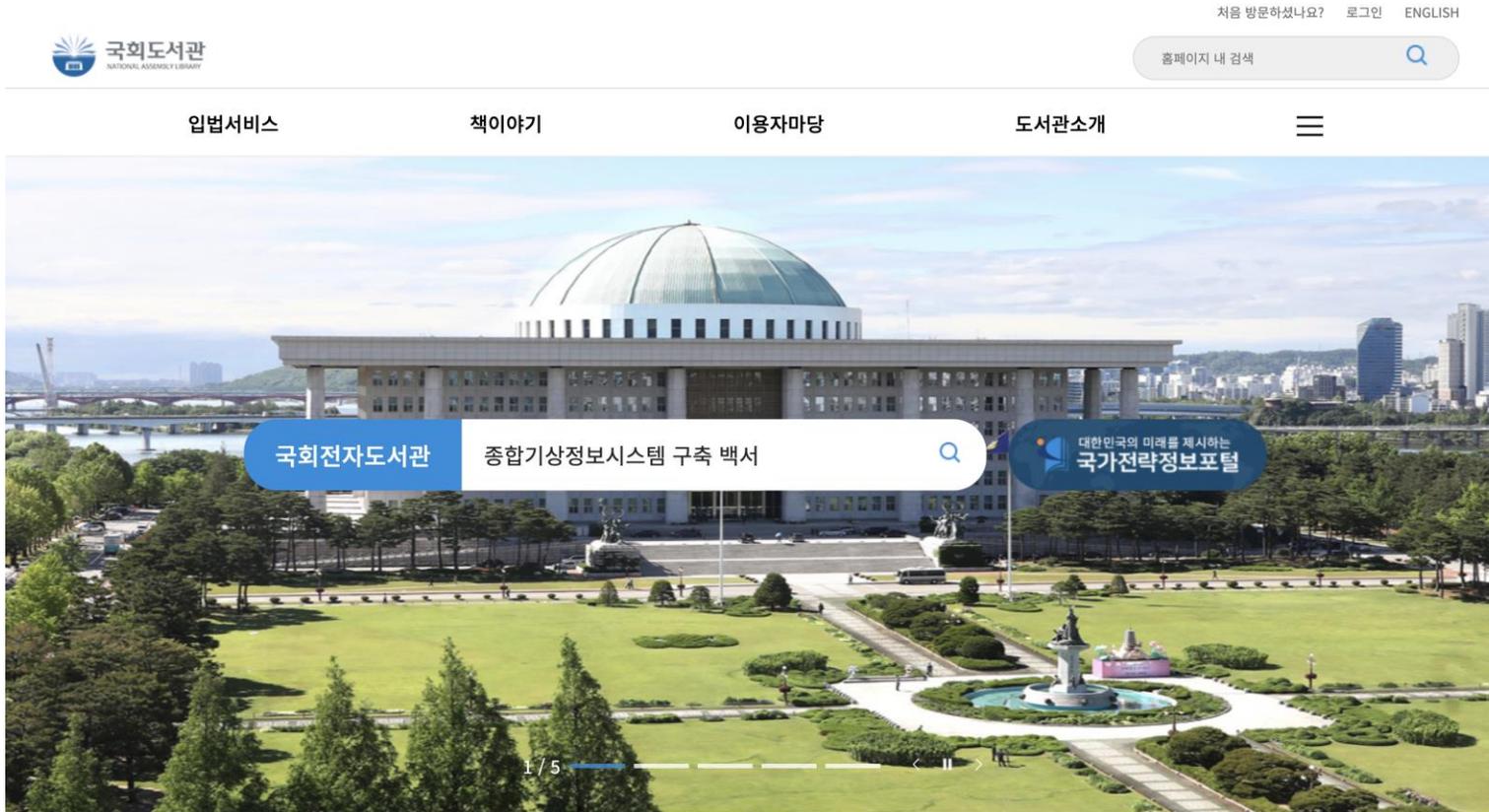
Hardware enablement advanced, extending support to broader vendor footprint: New backend drivers were added to Cinder: DataCore iSCSI and FC, Dell PowerStore NFS, Yadro Tatlin Unified iSCSI, Dell PowerStore NVMe-TCP, and Pure Storage NVMe-RoCE storage drivers. Cyborg now offers an Xilinx FPGA driver, which can manage Xilinx FPGA devices, including discovering devices' info and programming xclbin. The community also proposes a spec of adding NVIDIA **MIG** for A100 devices. Multi-Instance GPU (**MIG**) is new feature in Cyborg that allows GPUs based on the NVIDIA Ampere architecture (such as NVIDIA A100) to be securely partitioned, which is different from VGPU feature; the MIG driver is needed to managed compatible with PGPU and VGPU. In Nova, Virtual IOMMU devices can now be created and attached to an instance when running on a x86 host and using the libvirt driver.

Next OpenStack Release

항목	설명
예전 방식: 매 6개월마다 업그레이드	<ul style="list-style-type: none"> 6개월마다 새로운 릴리스를 설치 하지만 이게 너무 자주 바뀌어서, 큰 회사들은 계속해서 업그레이드를 해야 해서 작업난이도가 높음
새로운 방식: "SLURP" 릴리스	<ul style="list-style-type: none"> 그래서 OpenStack에서는 새로운 방법 도입 "SLURP"라는 특별한 릴리스 개념이 생김 이 릴리스는 1년에 한 번만 나오고, 중간 릴리스는 건너뛴 수 있음 예시: "A 릴리스"가 SLURP라면, 그 다음 "B 릴리스"는 건너뛰고, 다시 "C 릴리스" 때 업그레이드하면 돼요.
어떻게 좋아졌나요?	<ul style="list-style-type: none"> 이제는 매번 업그레이드하지 않아도 돼서 더 편리하고, 문제도 더 쉽게 해결할 수 있어요. SLURP 릴리스는 더 안정적이라서 많은 사람들이 이 릴리스를 선택할 거예요
기억할 점!	<ul style="list-style-type: none"> SLURP 릴리스는 매우 중요해요. 그래서 이 릴리스를 잘 기억하고, 업그레이드할 때 꼭 챙기면 돼요!



기상청



발간등록번호
11-1360000-001768-01

종합기상정보시스템 구축 백서



기상청

출처 : <https://www.nanet.go.kr/main.do>

Thank you.

감사합니다

