

Backend.AI를 활용하여 원격 공동 연구를 위한 대규모 AI/MLOps 플랫폼 서비스 구현하기



신정규

래블업 주식회사 / Google Developers Expert (ML/DL)

안녕하세요!

Lablup Inc. : Make AI Accessible

Open-source machine learning cluster
platform: Backend.AI

<https://www.backend.ai>

Google Developer Expert

ML / DL GDE (Context retrieval)

Sprint Master

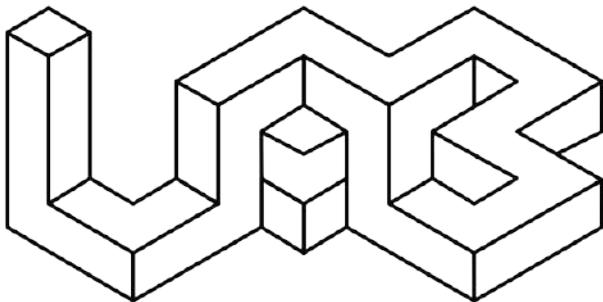




Lablup Inc.
Make AI Accessible

미션

현대 과학 연구와 응용 분야의 발전 속도차에 의한 간극 문제를
클라우드 컴퓨팅과 AI 기술을 바탕으로
계산 기반 연구의 새 패러다임 제시를 통해 해결한다



서비스/제품

Backend.AI (<https://www.backend.ai>)
기계학습 모델을 훈련하고 실행하는 모든 과정을
클라우드 또는 자신의 서버에서
엄청나게 쉽고 빠르게 돌려주는 세련된 플랫폼

딥러닝 기술의 발전과 대중화



하드웨어 연산 능력의 향상
초 대규모 행렬 연산 지원의
등장

통신망의 발전 및
IoT 보급 기반
빅데이터 수집

딥러닝 소프트웨어 스택 발전

딥러닝 기술의 발전과 대중화: 하드웨어 연산 능력의 향상



하드웨어 연산 능력의 향상
초 대규모 행렬 연산 지원의
등장

통신망의 발전 및
IoT 보급 기반
빅데이터 수집

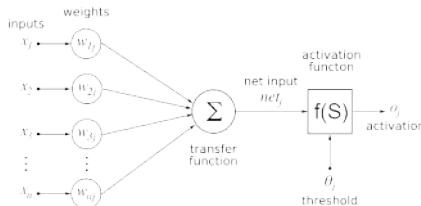
딥러닝 소프트웨어 스택 발전

- 뉴럴넷의 특징
 - 기본적으로 행렬 연산: GPU에 적합
 - 성능차: 동일 전성비 기준 10^2 배 이상
- 환경의 변화로 이어짐
 - 머신러닝의 발전 → GPU 수요 증가 → GPU 클라우드 서비스 등장 → 대규모 GPU/ASIC 자원의 등장
- AI Accelerator: ASIC 시장의 활성화
 - Google TPU (2016~), Intel Nervana (2017) and Movidius (2017)
- GPU를 왕창 모아 놓으면 무엇이 가능해지는가?
 - 초대규모 행렬 연산자원 Ultra-large scale calculation resources의 등장

하드웨어의 발전: 딥러닝과 GPU의 시너지



Deep Learning = 수십억 개 이상의 매개변수를 갖는 행렬 계산의 반복



$$\begin{aligned} & \text{activation function: } f(S) \\ & \text{threshold: } \theta_j \\ & \text{net input: } net_j \\ & \text{summation: } \Sigma \\ & \text{transfer function: } \Sigma \\ & \text{weights: } w_{ij} \\ & \text{inputs: } x_i \end{aligned}$$
$$1 \cdot 7 + 2 \cdot 9 + 3 \cdot 11 = 58$$
$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 7 \\ 9 \\ 11 \end{bmatrix} \begin{bmatrix} 8 \\ 10 \\ 12 \end{bmatrix}$$

6000만 파라미터
700경 계산량

2015

2015 Microsoft ResNet

3억 파라미터
2000경 계산량

2016

2016 Baidu Deep Speech 2

87억 파라미터
1.05해 계산량

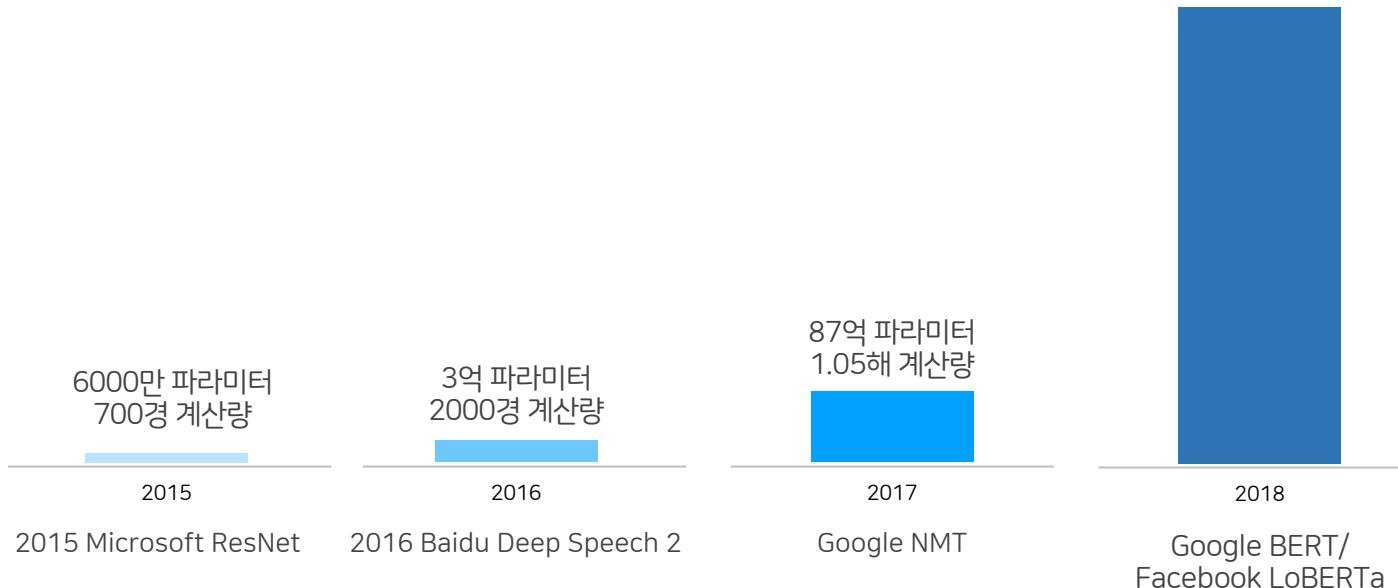
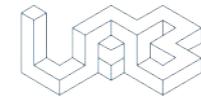
2017

Google NMT

Reference: NVIDIA 2017 "A NET COMPUTING ERA"

계산량 산정: GOPS * bandwidth

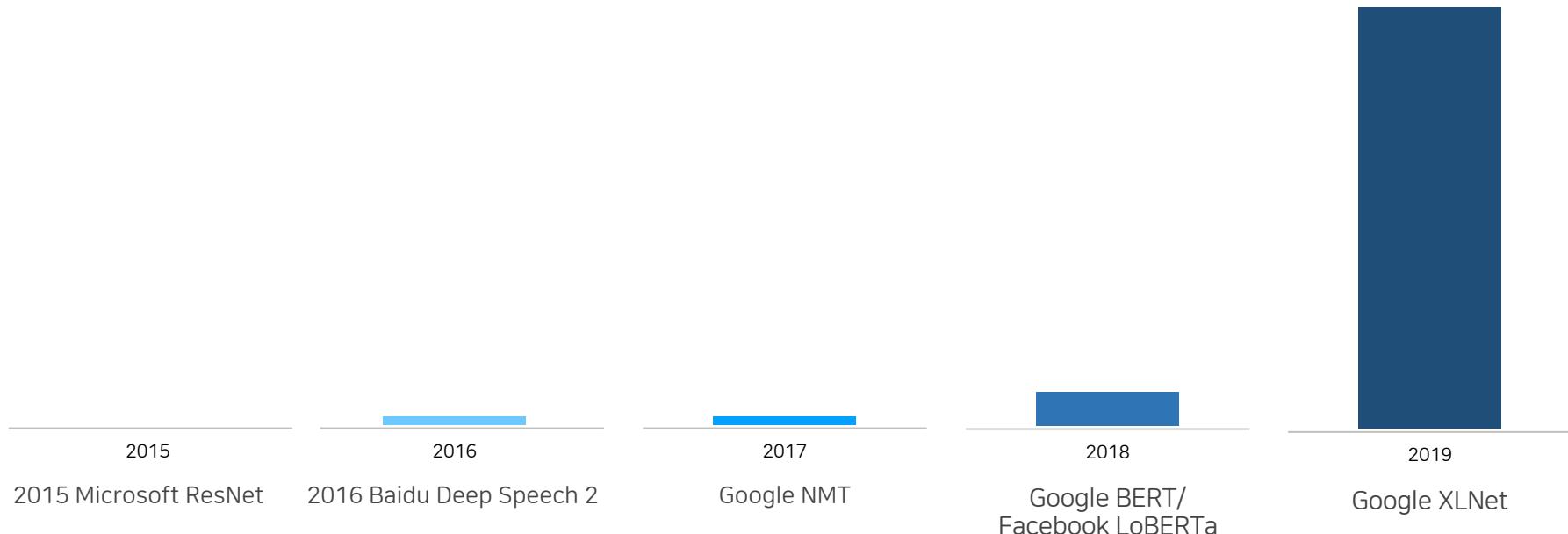
하드웨어의 발전: 딥러닝 모델의 복잡도의 급격한 증가



Reference: NVIDIA 2017 "A NEW COMPUTING ERA" / Google BERT

계산량 산정: GOPS * bandwidth

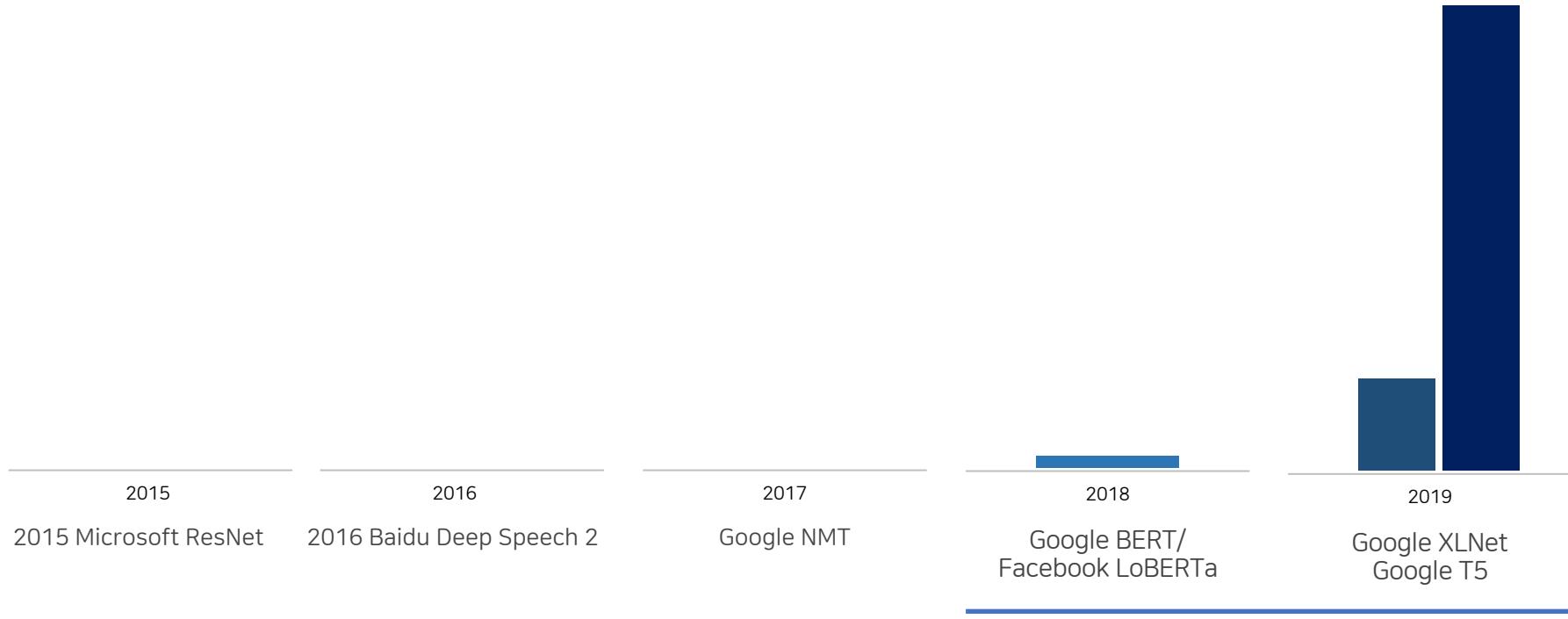
하드웨어의 발전: 딥러닝 모델의 복잡도의 급격한 증가



Reference: NVIDIA 2017 "A NEW COMPUTING ERA"

계산량 산정: GOPS * bandwidth

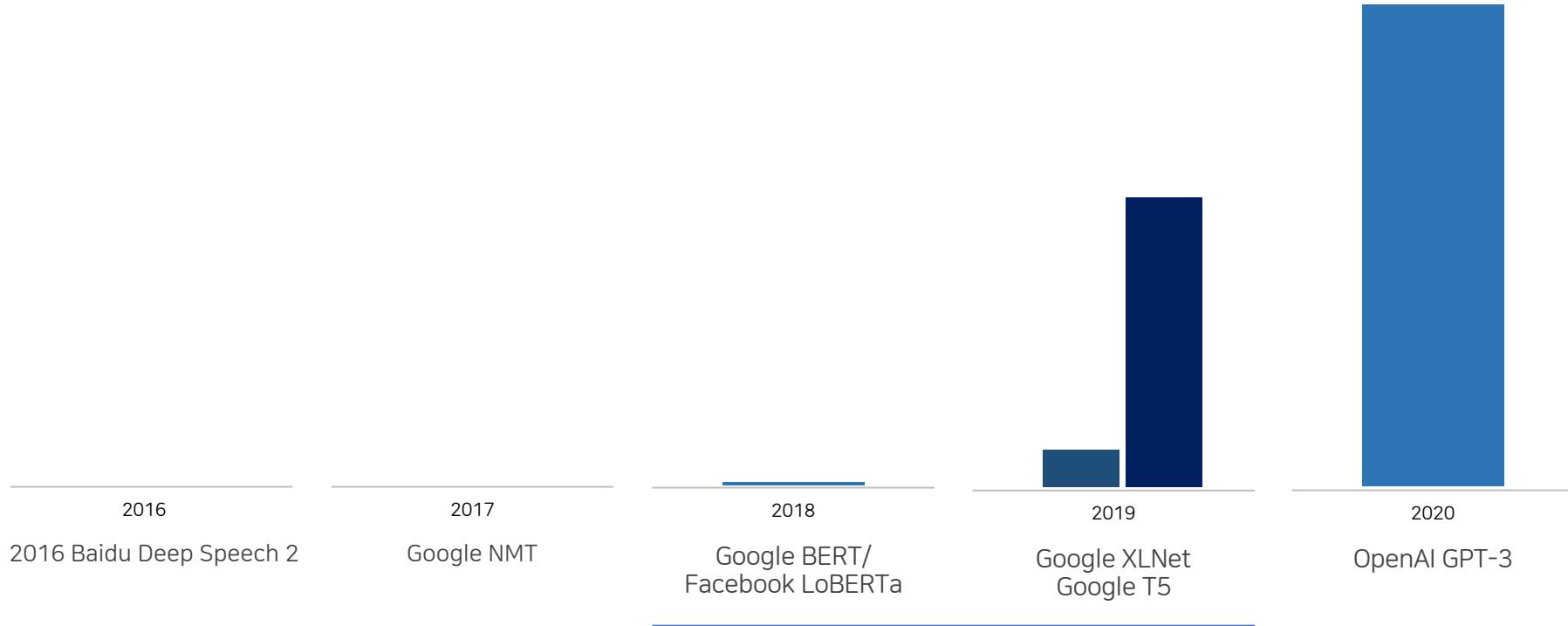
하드웨어의 발전: 딥러닝 모델의 복잡도의 급격한 증가



Reference: NVIDIA 2017 "A NEW COMPUTING ERA"

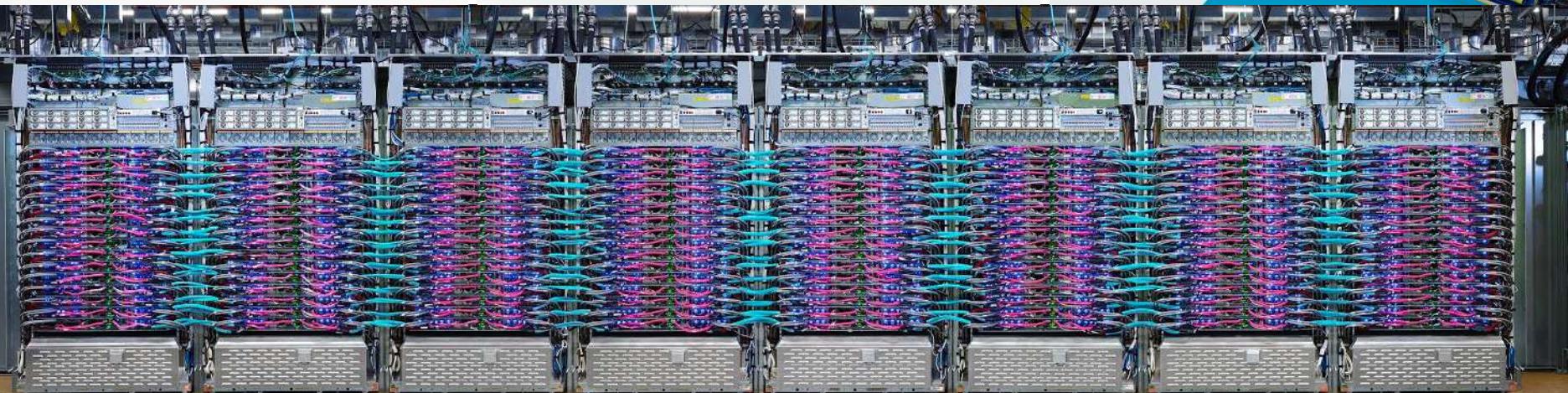
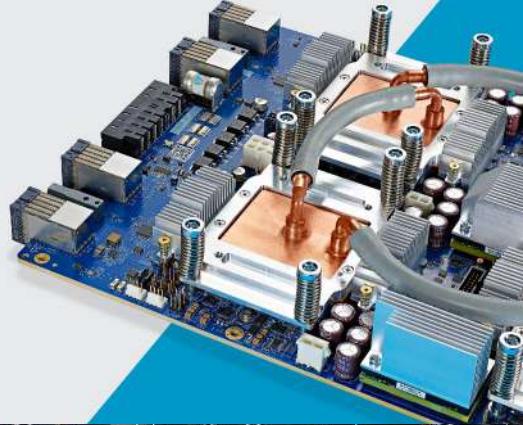
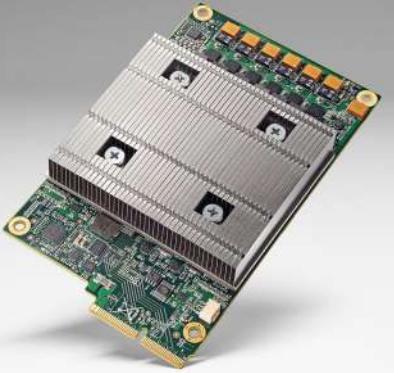
계산량 산정: GOPS * bandwidth

하드웨어의 발전: 딥러닝 모델의 복잡도의 급격한 증가



Reference: NVIDIA 2017 "A NEW COMPUTING ERA"

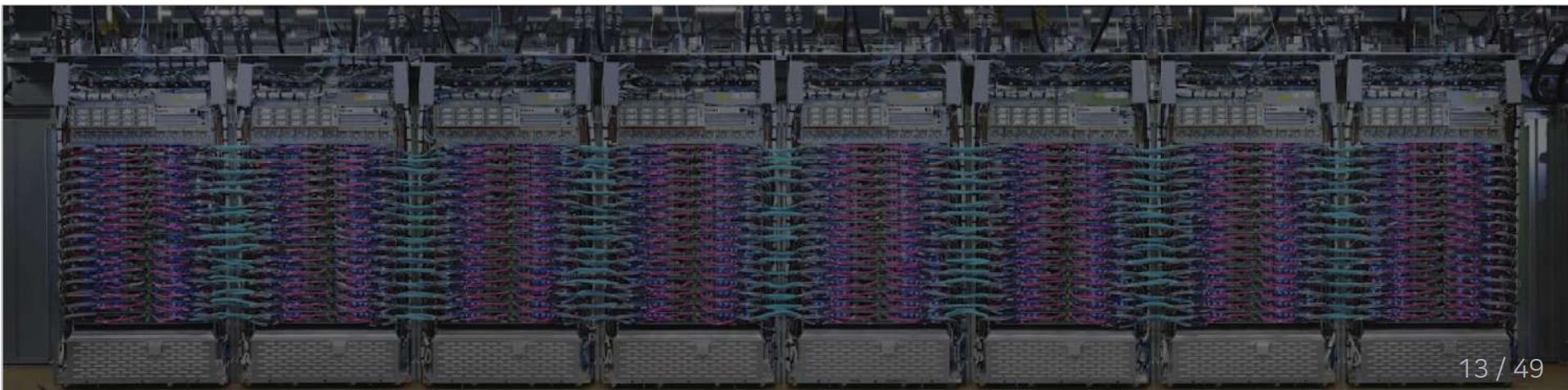
계산량 산정: GOPS * bandwidth



하드웨어의 발전: Cloud TPU Pod



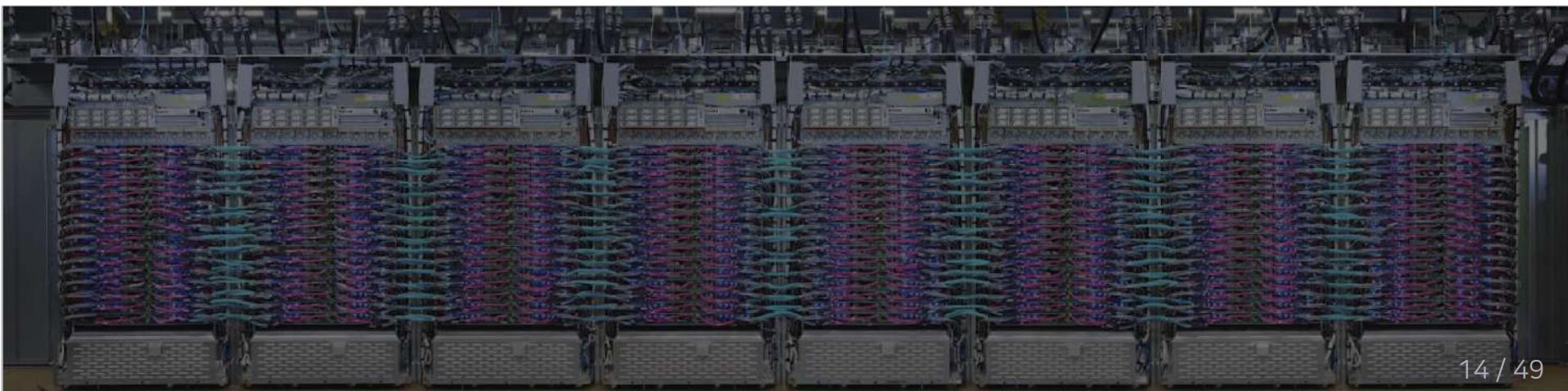
- TPU Pods (Google Cloud)
 - 초 대규모 행렬 연산 자원 기반 모델링의 가능성 시연
 - XLNet 공개 (2019년 6월)
 - ✓ BERT를 능가하는 성능의 언어 모델을
 - ✓ Cloud TPU Pod를 써서 2.5일만에 트레이닝
 - ✓ 비용: 2.5일에 2.9억원



하드웨어의 발전: Cloud TPU Pod



- TPU Pods (Google Cloud)
 - 초 대규모 행렬 연산 자원 기반 모델링의 가격 접근 장벽 현실화
 - Google T5 공개 (2019년 10월)
 - ✓ XLNet을 능가하는 성능의 언어 모델을
 - ✓ Cloud TPU Pod를 사용하여 약 2주일 만에 트레이닝
 - ✓ 훈련 비용: 약 30억원



하드웨어의 발전: AlphaStar

- AlphaStar (2019)

- 스타크래프트 2를 플레이하는 인공지능 모델
 - TPUv3 32개 / 44일간 사용
 - 에이전트 12개 실행

- 사용량 계산

- 1 TPUv3 = 420 TFLOPS
 - 총 사용량: 600 ZFLOPS (1 ZFLOPS = 1B TFLOPS = 10^{21} FLOPS)
 - 비교: AlphaGo Zero = 200 ZFLOPS

- 실제 비교

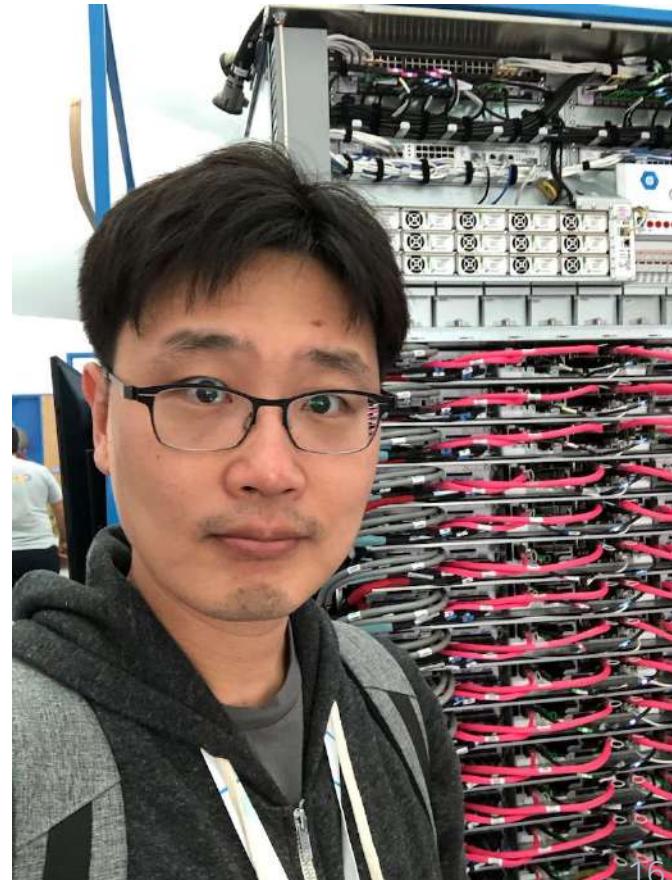
- Nvidia V100 GPU (~\$10,000) = 125 TFLOPS
 - 런타임 -> 150 년 이상



하드웨어의 발전: GPU 클라우드 인프라의 보급



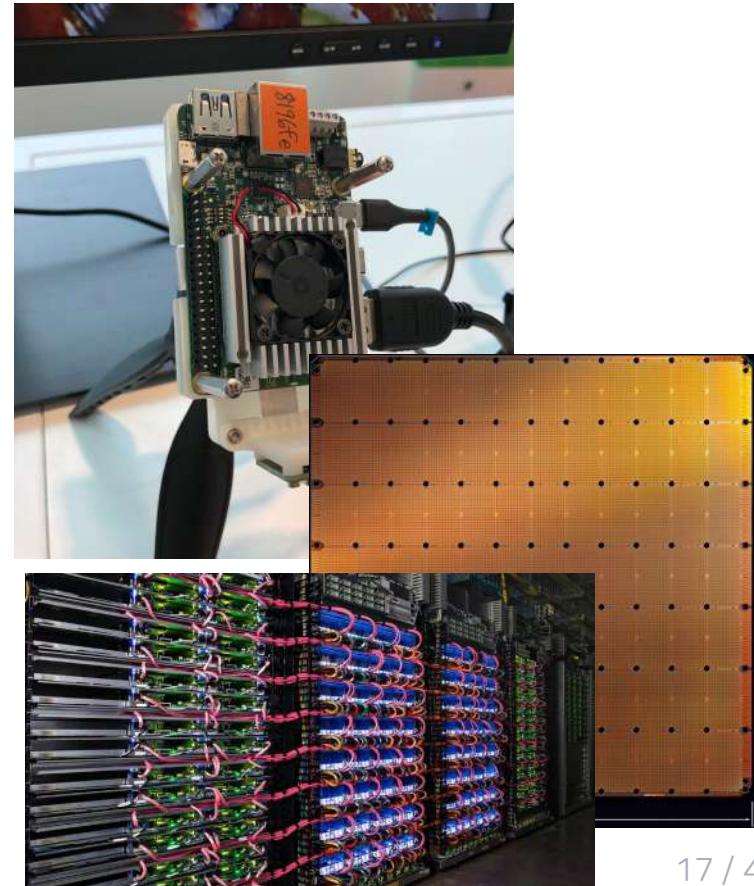
- 대부분의 인프라스트럭처가 클라우드화 됨
 - 더이상 유휴자원의 재판매가 아님
 - 클라우드 전문 사업
- 머신러닝 시대의 도래
 - 유휴자원 사용 이상의 자원이 요구되는 시점. 왜?
 - 기존에는 필요하지 않았던 리소스가 중요해졌음
 - GPU, special-purpose ASIC, DSP (e.g., TPU)
- GPU 클라우드 기반의 대규모 연산자원 대중화
 - 폭발적인 수요 증가 추세 (2017~)
 - 전성비 / 실사용 측면에서 기술적으로 엄청난 개선
 - 전용 ASIC / 전용 인스턴스 도입
 - ✓ AWS / GCP / MS Azure



하드웨어의 발전: AI 전용 가속기 시장의 개막



- 모바일: On-device 머신러닝 (엣지 디바이스)
 - ASIC + GPU + CPU 형태의 AP
 - 모바일 대상 공급
 - ✓ Apple NeuralEngine (2017~), Huawei NPU (2017~), Google Neural Core (2019)
- 서버: GPU 클라우드용 하드웨어
 - TPU (Tensor Processing Unit), Google (2016~)
 - Intel Nervana + Amazon AWS (2017~)
 - DGX-Family (DGX-1/2, DGX-station), Nvidia (2017~)
 - Loihi / Nahuku: Simulated NeuralNet, Intel (2018~)
 - Gaudi / Goya, Habana, Intel (2019~)
 - WSE, Cerebras (2019)
 - IPU, Graphcore (2019)



딥러닝 기술의 발전과 대중화: 딥러닝 소프트웨어 스택 발전



하드웨어 연산 능력의 향상
초 대규모 행렬 연산 지원의
등장

통신망의 발전 및
IoT 보급 기반
빅데이터 수집

딥러닝 소프트웨어 스택 발전



- **프레임워크**

- Theano (MILA, 2011~2017)
- Caffe (Berkeley, 2012~2017)
- Chainer (Preferred Networks, 2015~)
- TensorFlow (Google, 2015~)
- PyTorch (Facebook, 2017~)
- Gluon (Amazon, 2017~)
- MXnet (Amazon, 2017~)

- **라이브러리**

- cuDNN / NCCL (Nvidia)
- MKL-dnn (Intel)
- MIOpen / RCCL (AMD)

- 인공신경망 모델 개발
 - 기본 설계는 여전히 연구진이 진행
 - 최적화는 완전히 AI 자동화에 위탁
 - 현재 타 분야에서 일어나는 일들과 유사함: 건축, 기기 설계 분야

- 장점
 - 미시 최적화 부분을 자동화할 수 있음
 - 지속적으로 정확도를 올려나갈 수 있음

- 단점
 - 최종 결과물의 크기가 필요 이상으로 커질 수 있음
 - 최적화한 모델에서 오버피팅 문제가 발생했을 경우, 모델 구조의 복잡성으로 인하여 디버그하기가 까다로움

- AutoML (Google, 2017~)
 - 대규모 연산 자원을 사용한 뉴럴넷 설계
 - 기본 네트워크를 바탕으로 끊임없이 세대를 거치며 최적화를 거치는 방법 채택
 - 비전, 자연어, 번역용 AutoML 공개 (2018. 7, Google CloudML)
- 오픈소스 AutoML 구현
 - Auto-sklearn : 자동화된 estimator 제공 (2015)
 - AutoKeras : 가우시안 기반의 최적화 (2018~)
 - Microsoft NNI : 다양한 AutoML 알고리즘 제공 (2018~)
 - IDSIA Sacred: 하이퍼파라미터 튜닝 지원 프레임워크 (2018~)
 - Amazon AutoGluon (2020~)
- 공통점
 - 기존 CPU 클라우드로는 감당할 수가 없음 (엄청난 연산량이 필요함)
 - 일반 뉴럴넷 학습 자원의 $10^3 \sim 10^4$ 배
 - 급격한 오픈소스화 진행 중

- 개인화

- 각 분야마다 인공 신경망 템플릿을 미리 작성해 두고, 이를 기반으로 자동 최적화 도구 실행
- Azure Cognitive Toolkit (2017) / Google CloudML (2018)
- 추가 레이어를 별도로 두는 방식 / 모델 자체의 가중치를 조절하는 방식이 혼재
- 전이 학습 (Transfer Learning) 기반 접근 시도 시작 (2019~)

- 데이터 기반 최적화

- 공개된 신경망을 기반으로 자신의 용도에 알맞는 신경망이 되도록 데이터를 추가로 집어넣어 학습
- 실시간 데이터 파이프라인을 통해 동적 평형 상태에 있는 시스템에서 항상 최적화된 결과 도출

딥러닝 기술의 발전과 대중화: 새로운 문제의 대두



하드웨어 연산 능력의 향상
초 대규모 행렬 연산 지원의
등장

통신망의 발전 및 보급 기반
빅데이터 수집

딥러닝 소프트웨어 스택 발전

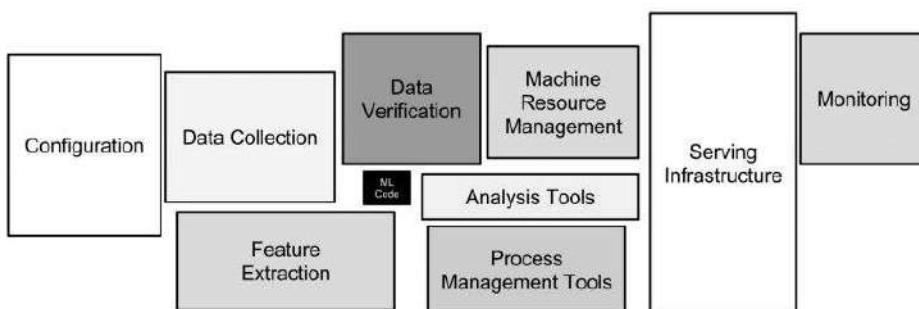
딥러닝 기술의 발전과 대중화
새로운 문제들이 대두됨

- 인공지능/HPC 기술의 폭발적인 성장
 - 하드웨어의 연산(computing) 능력의 지수적 향상
 - 다양한 빅데이터의 수집
 - HPC/딥러닝 모델을 쉽게 개발할 수 있는 소프트웨어 라이브러리의 발전
- 결과
 - 머신러닝 방법론의 특성: 광범위한 분야에 다양한 스케일로 적용 가능
 - ✓ 각 분야의 도메인 전문가 또는 취미로 배우고자 하는 일반인, 학생 사이의 수요가 크게 증가
 - 머신러닝 기반 / 머신러닝 융합 기술의 보급
 - ✓ ICT 산업, 제조, 교통, 환경 / 교육, 의료, 금융, 공공 부문

- **플랫폼 구축의 기술적 난이도 상승에 따른 새 Ops필요성 증가**
 - 다른 분야와의 대표적 차이점: 빠른 변화 사이클 (하드웨어 및 소프트웨어 플랫폼, AI 프레임워크, 모델 등)
 - 구축이 끝나는 시점에서는 이미 구세대 플랫폼이 됨
 - 이로 인한 관리의 어려움
- 딥러닝 대상 워크로드 최적화
 - 하이퍼파라미터 튜닝
 - AutoML
- 파이프라인 단계에 따른 이기종 자원 요구 및 스케줄러
 - 훈련시: 연구 클러스터 플랫폼의 잡 스케줄러에 가까운 구조
 - 추론시: 서비스 오퍼레이터에 가까운 구조

Hidden Technical Debt in Machine Learning Systems

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips
{dsculley, gholt, dg, edavydov, toddphillips}@google.com
Google, Inc.



“Only a fraction of real-world ML systems
is composed of ML code”

- 비자동화된 부분이 상당히 많음
 - 설정, 데이터 수집, 검증, 특징 추출, 리소스 관리, 인프라스트럭처 관리 등
- 각 부분별 새로운 자동화 도입
 - 클라우드 벤더들의 서비스 제공
 - AWS
 - ✓ Sagemaker Zero
 - Google
 - ✓ ML Kit / Cloud Composer
 - Microsoft
 - ✓ Azure Cognition Toolkit

- MLOps
 - DevOps 개념을 머신러닝에 특화한 개념
 - DevOps CI/CD를 구축하기 위한 Pipeline 의 머신러닝 버전
- DevOps / CI/CD 와의 차이점
 - 각 컴포넌트들의 연산자원 요구가 매우 불균형
 - 각 컴포넌트들의 발전에 따른 변경사항이 엄청나게 다양
 - 개발 과정 대신 실험 과정이 필요함
 - 배포 단계가 굉장히 다양하고 복잡함
 - 배포시 필요 리소스가 다양함
- DevOps 대비 추가로 요구되는 점
 - 파이프라인 스키마
 - 코드 검증은 모델 결과 검증으로 대체
 - 개발 단계 대신 실험+자동 모델 개선 루프가 필요

- MLOps 플랫폼

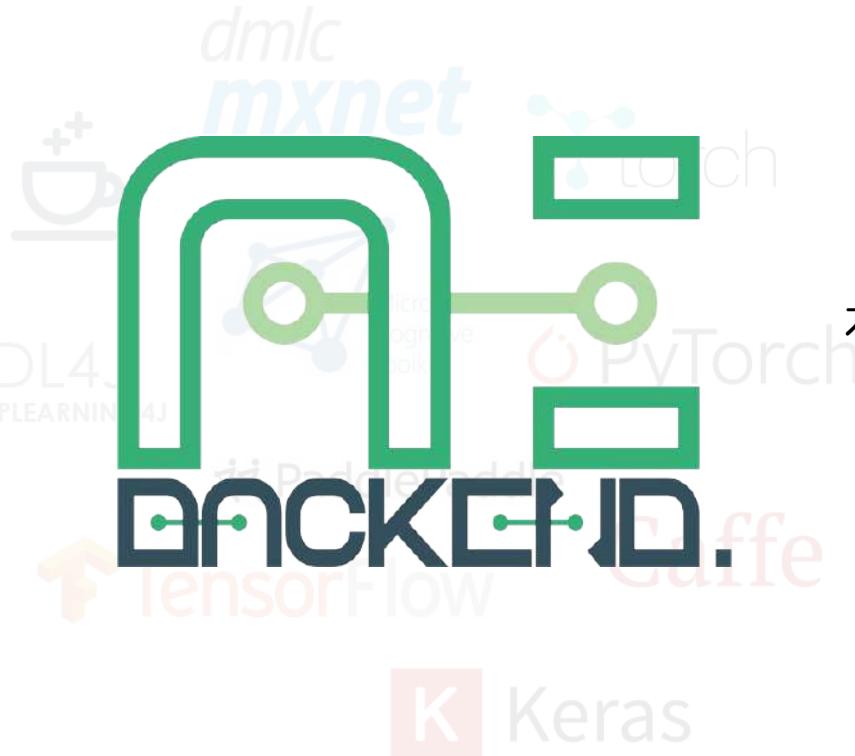
- AirFlow + TensorFlow Extended
 - ✓ Apache AirFlow: AirBnB가 개발한 일반 용도의 워크플로우 관리 플랫폼
 - ✓ TensorFlow Extended의 오케스트레이터로 연결하여 사용
 - ✓ GCP에서 지원 (Google Cloud Composer)
 - ✓ ETL 보다 훨씬 넓은 범위를 커버
- KubeFlow
 - ✓ Kubernetes 기반의 MLOps 플랫폼
 - ✓ TensorFlow Extended 및 TensorFlow에 특화
- MLFlow
 - ✓ Databricks에서 만든 MLOps 플랫폼
 - ✓ Tracking, Projects, Models를 효율적으로 분리하여 지원
 - ✓ TensorFlow, PyTorch 등 프레임워크에 의존성 없이 지원

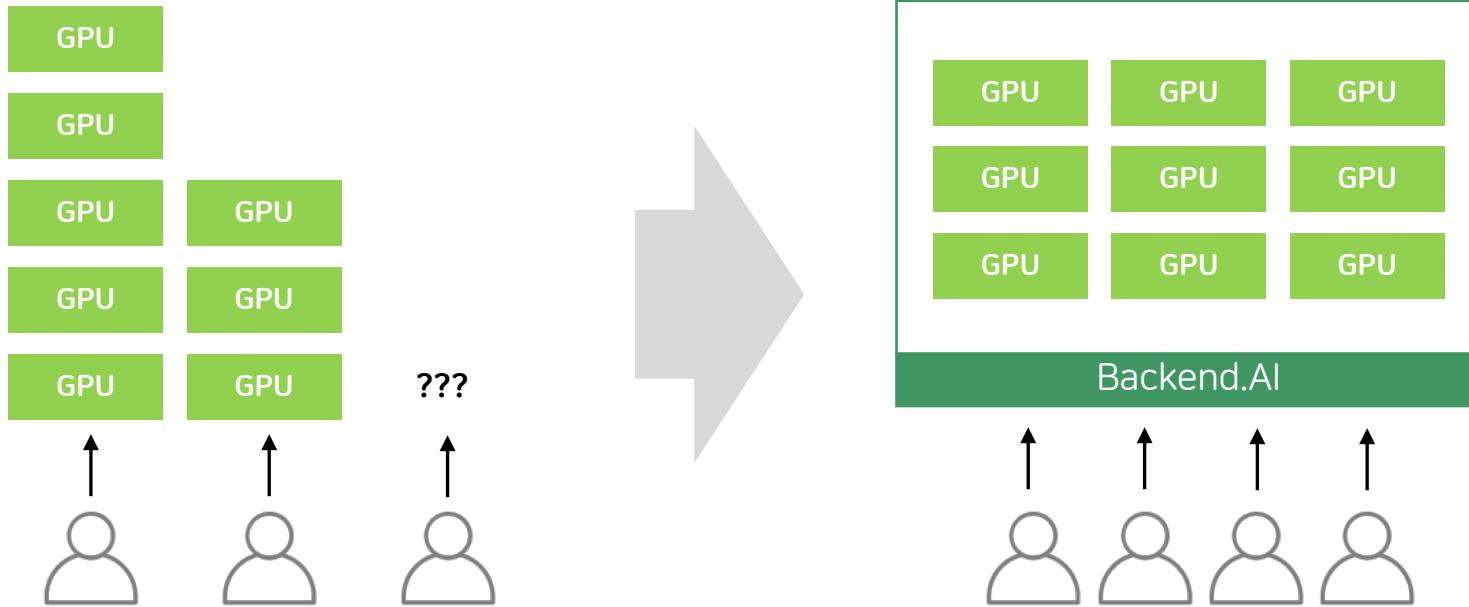


딥러닝 및 고성능 컴퓨팅 연구에 특화된
오픈소스 연산 지원 통합 플랫폼

자체적으로 최적화한 과학 연산 소프트웨어를
GPU 분할 가상화 기반으로 제공하여

과학자, 데브옵스, 기업 및 AI 애호가들이
효율적으로 규모를 확장할 수 있게 도움

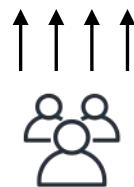
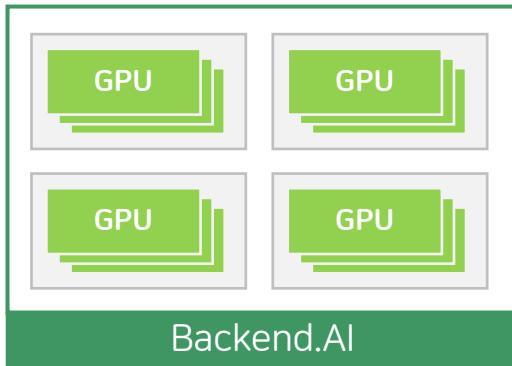




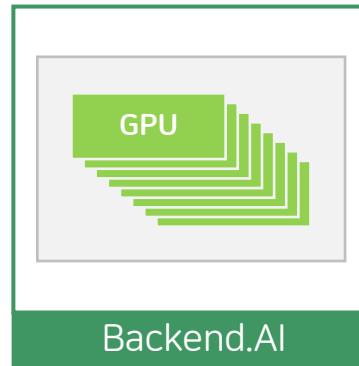
- 관리자가 사용자별로 GPU 할당
- 놀거나 부족한 자원 발생
- SW 유지관리 어려움

- Backend.AI가 GPU 통합 관리
- 동적으로 그때그때 필요한 만큼만 사용
- 컨테이너 기반으로 표준화된 SW 생태계

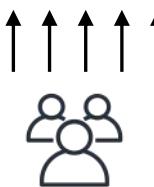
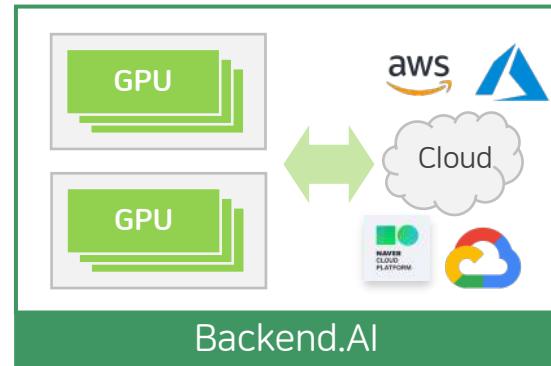
Backend.AI: 사용 시나리오



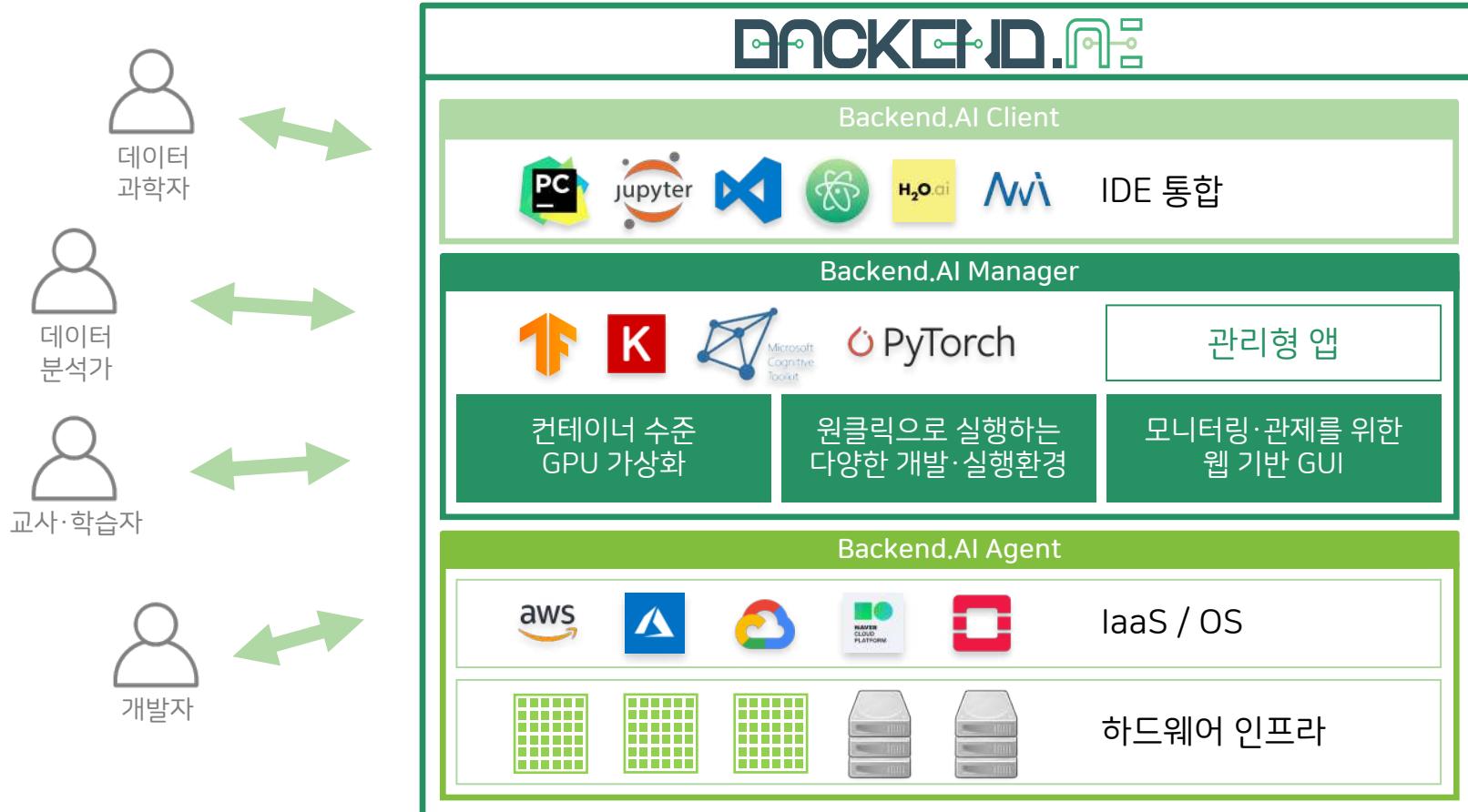
GPU 클러스터 구축



고성능 GPU 공유



클라우드로의 동적 용량 확장



Backend.AI: 플랫폼 개요 (상세)



성능

- VM 및 Kubernetes 기반 솔루션들 대비 동일 하드웨어로 고성능 달성
- AI, ML, HPC, 수치해석 등 연구 개발에 최적화
- 고성능 컴퓨팅에 특화된 다양한 배치, 자원 할당 및 병목 제거 구현

편의성

- ML / HPC 전문가들이 직접 만든 플랫폼
- 일관된 플랫폼 GUI (웹 및 데스크탑 앱) / 관리 동작 및 스크립팅을 위한 CLI
- 시스템 관리용 허브를 통한 상세한 관리자 제어 기능

확장성

- 완전 문서화된 API 및 SDK (Python, Node.js) 제공
- 다양한 GPU 및 머신러닝 가속 H/W 지원^[1]
- On-Prem에서 Public Cloud, Hybrid 클라우드까지 쉬운 확장

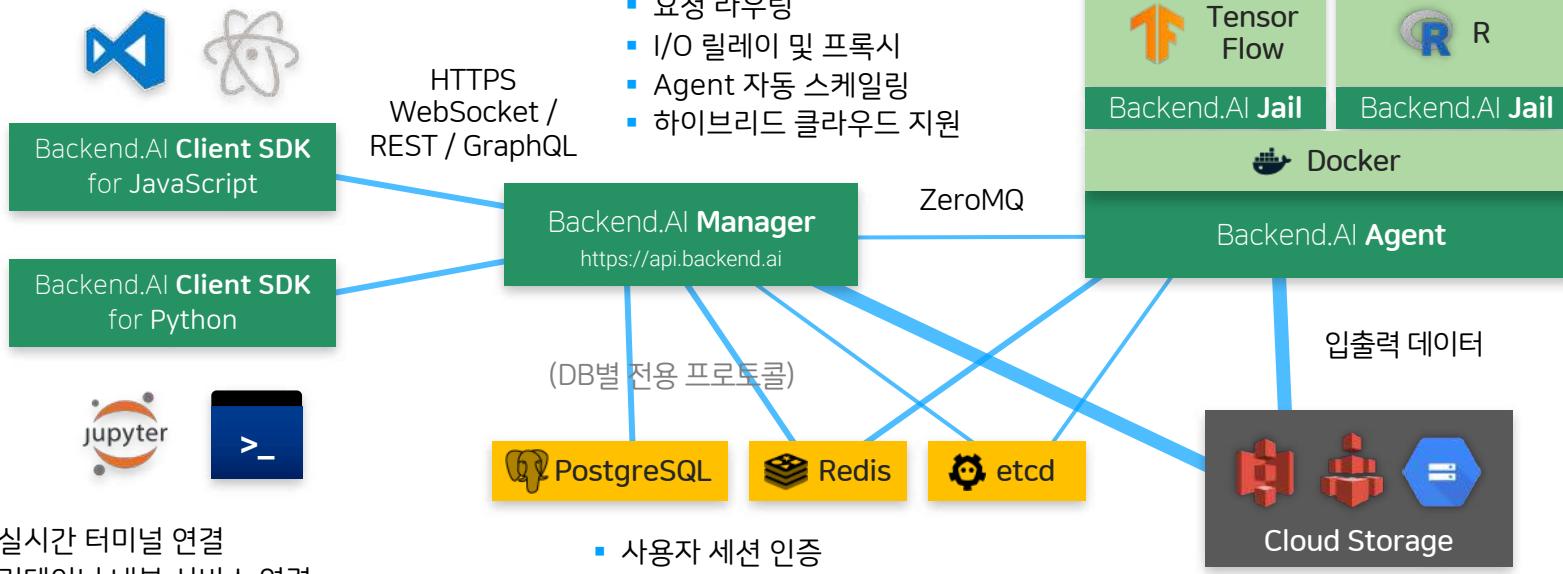
비용 절감

- GPU 분할 가상화(Fractional GPU™)를 통한 고가 GPU의 활용성 증대 및 고가용성 달성
- 더 적은 하드웨어로 동일한 성능, 동일한 하드웨어로 더 높은 성능 제공
- 강력한 장애 대응 (연속성 Fail Over, 쉬운 장애 원인 분석 및 로그 API / 로그 솔루션 통합)



AI, ML, HPC를
R&D 부터
Business Service,
AI Service 추론 및 제공까지
하나의 일관된 플랫폼을 통해
효과적으로 관리

Backend.AI: 플랫폼 구성요소



- 실시간 터미널 연결
- 컨테이너 내부 서비스 연결
- 관리자용 모니터링 기능 연동
- 멀티미디어 출력 처리

- 사용자 세션 인증
- 실시간 사용량 추적
- 자동 rolling upgrade

- 사용자별 가상폴더 관리
- 권한 기반 가상폴더 초대 및 공유
- 예제 데이터집합 제공

- 컨테이너 기반 GPU 스케일링

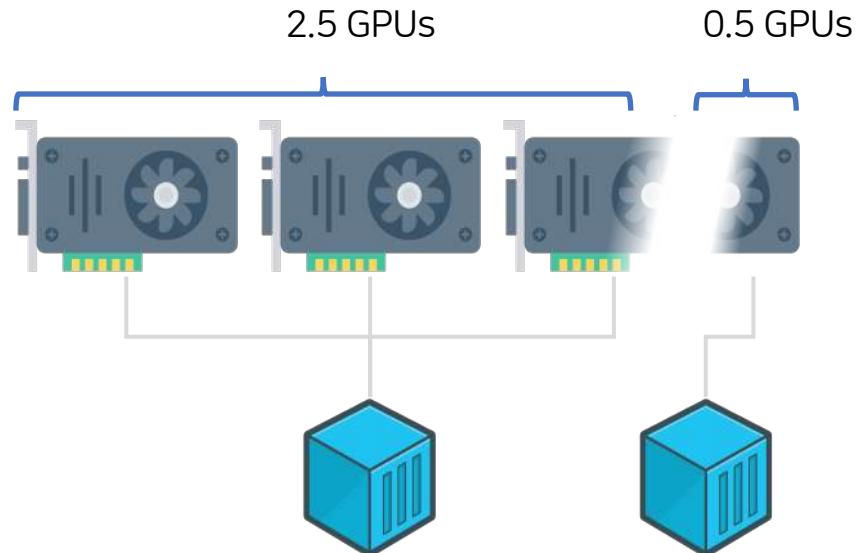
- 컨테이너별로 CUDA SMP 및 GPU RAM을 나눠줌
 - ✓ 예) 2.5 GPUs, 0.3 GPUs를 각 컨테이너에 할당
- 단일 GPU 공유 : 교육 및 추론 워크로드에 적합
- 다중 GPU 할당 : 모델 훈련 등 대규모 워크로드에 적합
- 자체 개발한 CUDA 가상화 계층으로 구현

- NVIDIA 플랫폼 통합

- DGX 패밀리 최적화
- Nvidia에서 직접 개발하는 NGC (Nvidia GPU cloud) 환경 지원

- 혜택

- 고가의 하드웨어인 GPU의 사용률 향상을 통한 구입 비용 감소
- 고가의 훈련용 GPU를 분할하여 추론 GPU로 운영
 - ✓ 노후 장비 활용성 최대화 (훈련 → 추론용으로 전환)



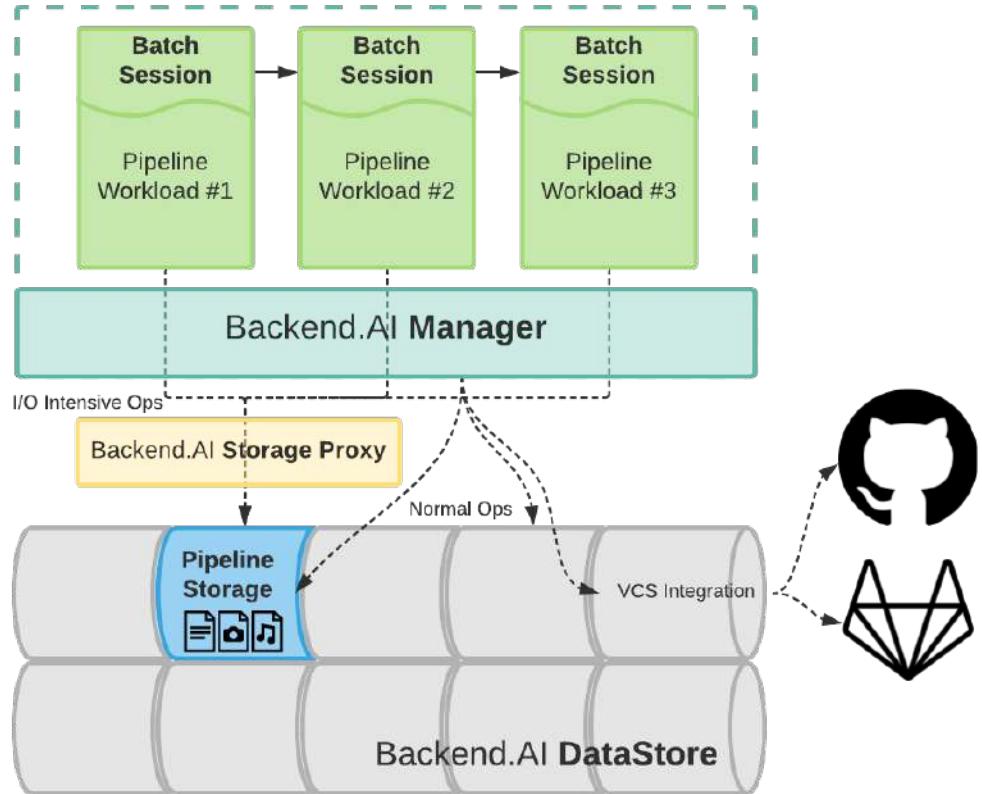
GPU 부분할당 예시 (2.5 / 0.5 slots)

- COVID-19가 가져온 환경 변화
 - 원격 근무 / 원격 연구
 - 원격지 협업 가속
- 원격 연구 협업 플랫폼 요구
 - 넓은 플랫폼 접근성
 - ✓ 데스크탑, 태블릿, 모바일 기기 등
 - 모델 개발 자원 관리 및 공유
 - ✓ 조직 / 팀 별 공용 데이터
 - ✓ 연산 자원 공유 및 제약
 - ✓ 개인 / 조직별 사용량 및 효율 모니터
 - 원격 플랫폼 통합
 - ✓ 보안 연결 지원
 - ✓ 타사 원격 근무 플랫폼 통합

- 넓은 플랫폼 접근성
 - 데스크탑용 앱
 - 웹 접속 지원 및 태블릿 / 랩탑 / ChromeOS용 웹 앱 제공
- 모델 개발 자원 관리 및 공유: 그룹 정책
 - 프로젝트 그룹 / 조직별 관리 기능
 - 조직 / 팀 별 공용 데이터
 - 연산 자원 공유 및 제약
 - 개인 / 조직별 사용량 및 효율 모니터
- 원격 플랫폼 통합
 - SSH/SFTP 터널링 기반의 보안 환경 제공
 - VPN 연결 과정 없이 보안 연결 구현 및 지원
 - GUI + 포트 터널링용 데스크탑 앱 제공
 - LDAP 연동

- MLOps의 표준이 없음
 - 같은 조직 안에서도 연구자에 따라 사용 패턴이 너무 다름
 - AirFlow, KubeFlow, MLFlow 등의 플랫폼의 발전이 ML 프레임워크만큼 빨리 변함
- MLOps 커뮤니티의 변화가 심함
 - 2018년 하반기부터 개발: 중간에 경험한 스펙 변화
 - ✓ 서빙의 경우: TensorFlow GraphDef → TF SavedModel → ONNX, PyTorch 추가
 - ✓ 훈련의 경우: TensorFlow 1.6 → TensorFlow 1.15 / 2.3, PyTorch 0.9 → PyTorch 1.6
 - ✓ 하위 버전 호환성이 없음
- Backend.AI MLOps의 접근 철학
 - 파이프라인은 자체적으로 완결성이 있어야 함
 - ✓ 이식성 및 휴대성
 - ✓ 데이터 + 코드 + 환경
 - ✓ 재사용성 및 재현성 필요

- **파이프라인 전용 스토리지 폴더에 기반한 입출력 데이터 연계**
 - 파이프라인 배치 작업 실행 시 모든 컴포넌트가 공통으로 사용하는 전용 스토리지 폴더를 자동 생성/마운트함
 - 다양한 종류의 데이터를 전용 스토리지 폴더 임의 위치에 자유롭게 저장하고 코드 기반으로 유연하게 연계
 - ✓ 공용 파일 시스템 기반 데이터 입출력을 통해 코드 상 별도 문법 적용 필요 없음
 - 파이프라인 GUI를 통해 입출력 데이터 위치 설정 후 컴포넌트 모듈 범위 데이터 변수 자동 생성 (20.09 / 21.03 도입)
 - ✓ 자주 사용되는 형식의 데이터 자동 파싱 지원으로 편리한 입출력 데이터 접근
- **유연한 데이터 연계 활용**
 - 메타 데이터 기반의 파이프라인 컴포넌트 및 데이터 재구축/재사용
 - 사용자별, 그룹별 파이프라인 스토리지, 데이터, 모델 공유/연계
 - 스토리지 폴더 및 데이터의 서드파티 서비스 원클릭 연계 (ex: Github/Gitlab으로의 반출 및 반입)



Pipeline Storage: 각
파이프라인별로 수행되는 모든 작업은
공유 스토리지를 이용하여 입출력을
연계함

공유 스토리지 자체를 새 파이프라인의
기반 템플릿 데이터와 모델로 사용

- PyTorch 모델 추론 지원
 - PyTorch Serve (TorchServe) 및 Backend.AI + Triton 두가지 방법 지원
 - 권장 방법: Triton inference server
- Nvidia Triton inference server 기반의 추론 서버 기능
 - Backend.AI + NGC 컨테이너 통합 기능
 - CPU 및 GPU 기반 서빙
 - 지원 포맷
 - ✓ TensorFlow GraphDef, TF SavedModel, ONNX, Pytorch, Caffe2 NetDef
- 추론 시작
 - Data&Storage - 모델 탭에서 바로 추론 시작
 - ✓ 추론 모니터링은 세션 탭에서 시작 (20.03) / 20.09에서 전용 메뉴 제공 예정
 - Stateful / Stateless 모드 지원 (Triton 통합 기능 사용시)

- Backend.AI + MLFlow integration 기반의 실험결과 모니터링
 - mlruns 가상폴더 마운트 기반의 분산처리
 - ✓ 기본값을 사용하거나, 임의의 값 지정 가능
 - 계산 결과가 지정한 mlruns 하위에 저장됨
 - mlruns모니터 세션을 통해 모든 실험 결과를 실시간으로 열람
- Backend.AI + Sacred + Omniboard 기반의 실험 설계
 - Omniboard 기반의 Sacred Experiments 시각화 지원 (베타)
- Backend.AI + NNI 기반의 AutoML 실행 및 모니터링
 - NNI 통합 지원 (Backend.AI 20.03 이상)
 - NNI Board를 통한 AutoML 시각화 지원

Backend.AI Pipeline (데모)

Backend.AI

Backend.AI Console

현재 프로젝트
Lablup

사용자 이름
Lablup User

요약

대시보드

시작

+ 시작

- 파일 업로드
- 새 키페어 생성
- 키페어 관리

자원 사용량

리소스 그룹
HPC

CPU	0/198
RAM	0.00/737.77GB
세션	0/50

현재 리소스 그룹 (HPC)
사용자 리소스 제한

시스템 자원

3 연결된 노드
13 활성 세션

CPU	97/429 코어 예약. 사용 12.08% (util. 5183.98 %)
RAM	432.00/2,765.90 GB 예약. 사용 121.12 GB (15.62 %)
GPU	18.5/96 CUDA fGPUs Fractional GPU 가상화 사용중.

공지

Welcome to Backend.AI cloud open-beta!

We updated Backend.AI cloud to the latest beta. We are inspecting areas where problems may arise.

status healthy
All systems normal.

관리

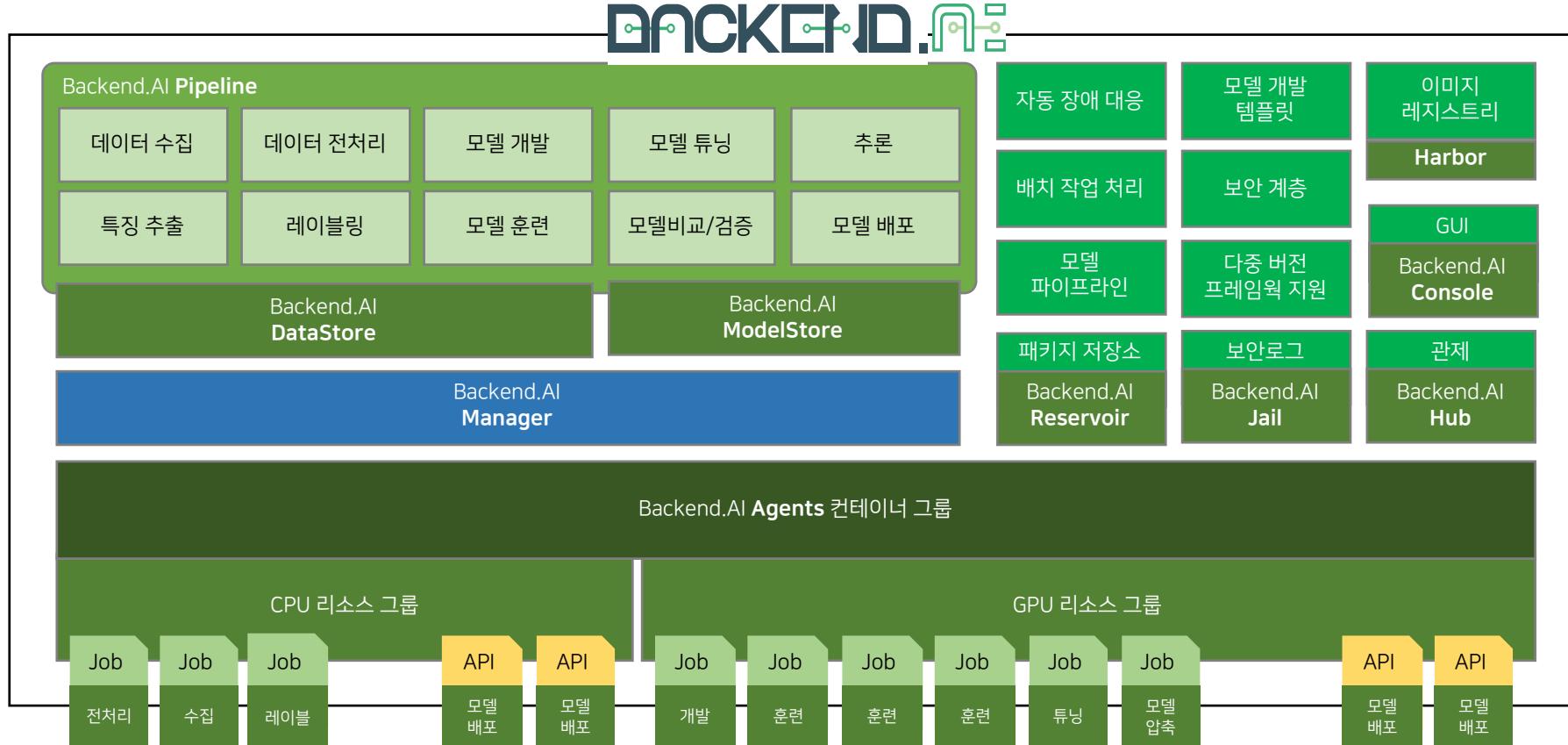
Manager version 20.03.0rc1.dev0
Console version 20.07.9 ✓

- 실행 환경 이미지 업데이트
- 자원 확인
- 시스템 설정 변경
- 시스템 유지보수

Backend.AI MLFlow 통합 (데모)

44 / 49

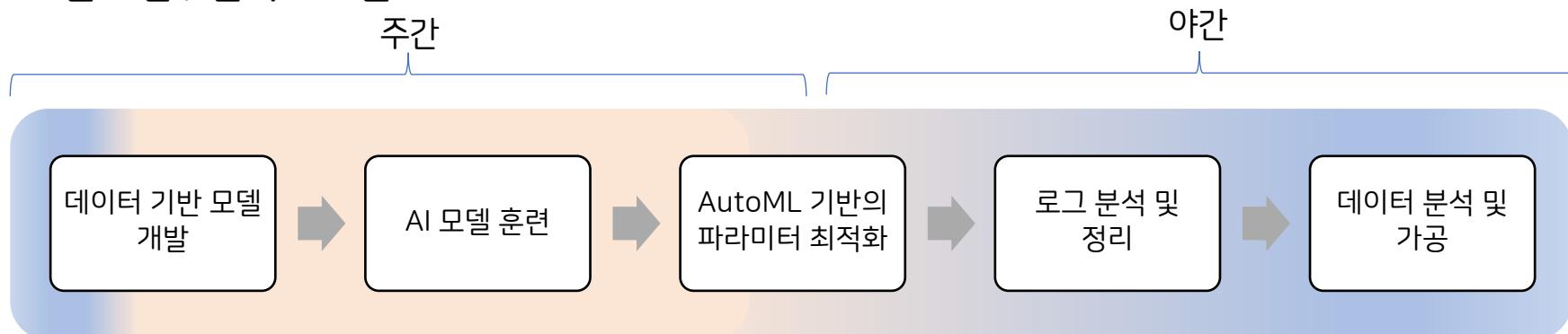
Backend.AI MLOps: 전체 시스템 구성



- 하이브리드 워크로드의 예

- 머신러닝 모델 개발 (주간) / GPU 전체 사용
- 머신러닝 모델 훈련 및 최적화 (야간) / GPU 가상화 최적 조건
- 데이터 분석 (야간) / CPU 중심의 워크로드
- 로그 분석 (새벽) / CPU 중심의 워크로드

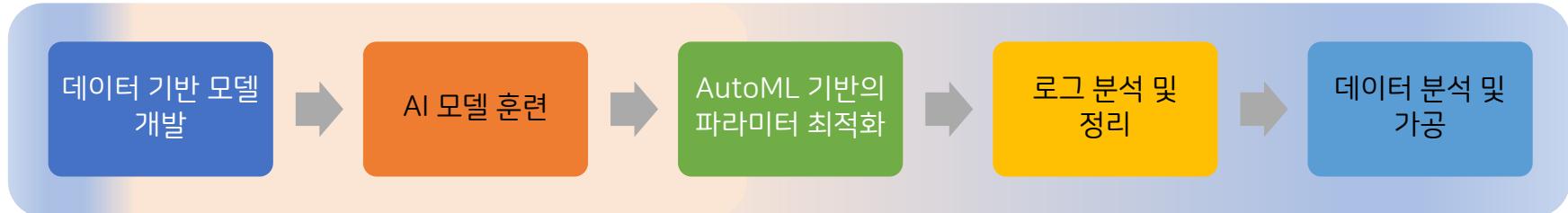
- 모델 개발 / 분석 사이클



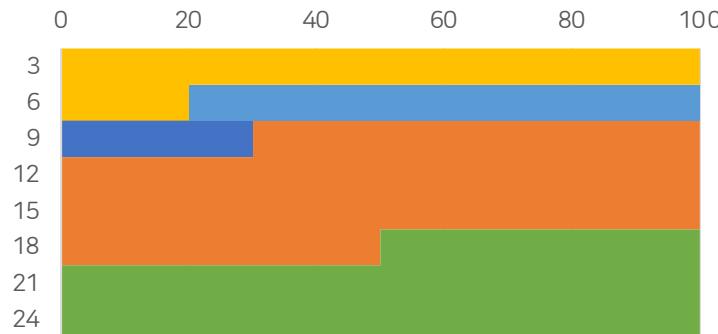
Backend.AI MLOps: 멀티 워크로드 스케줄링



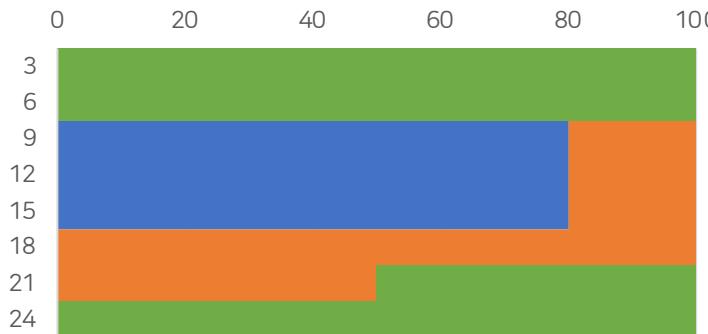
- 하이브리드 워크로드 배치



CPU 할당 시나리오



GPU 할당 시나리오





되짚어보기

- 딥러닝 보급의 3요소
- AI/MLOps 솔루션의 대두
- Backend.AI 소개
- Backend.AI 기반의 MLOps 구현
- 파이프라인 / MLOps 통합 시연
- 구축 구조 및 예시

Make AI Accessible!

더 많은 정보를 원하신다면,

Lablup Inc.

<https://www.lablup.com>

Backend.AI

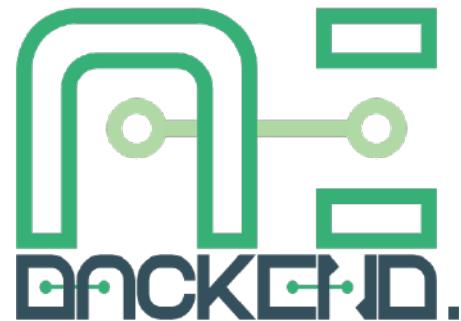
<https://www.backend.ai>

Backend.AI GitHub

<https://github.com/lablup/backend.ai>

Backend.AI 클라우드 (beta)

<https://cloud.backend.ai>





Appendix

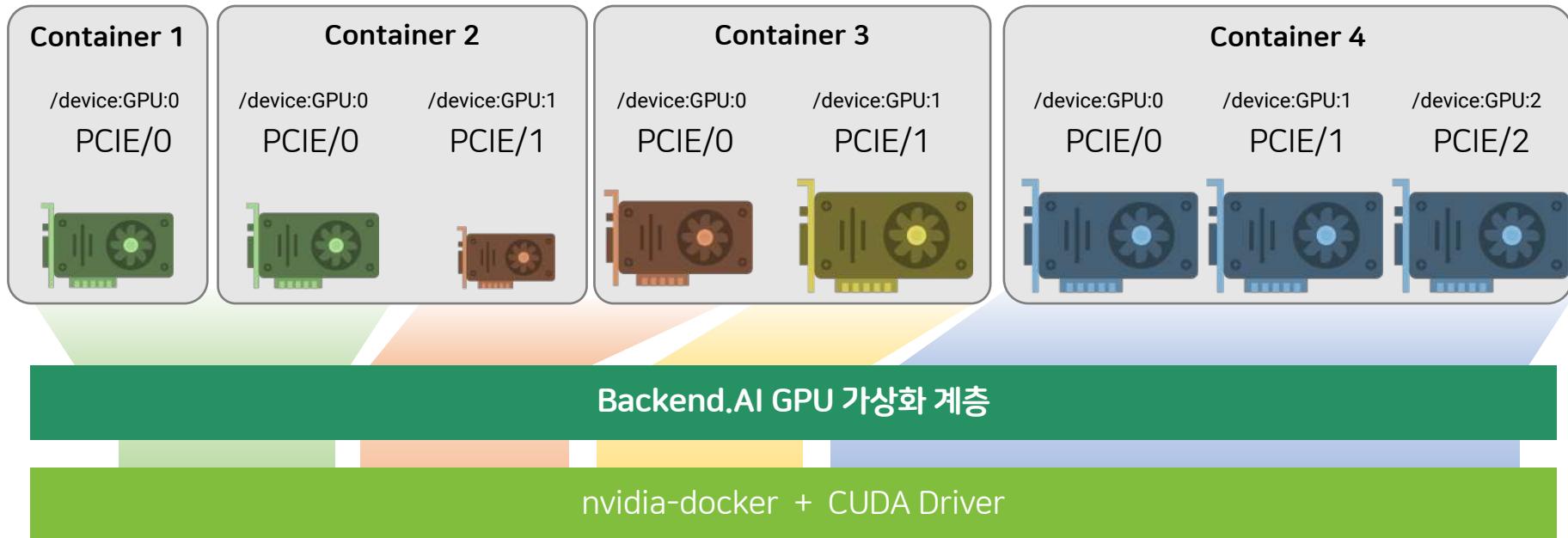
Backend.AI: The most efficient way to build and train your machine learning models



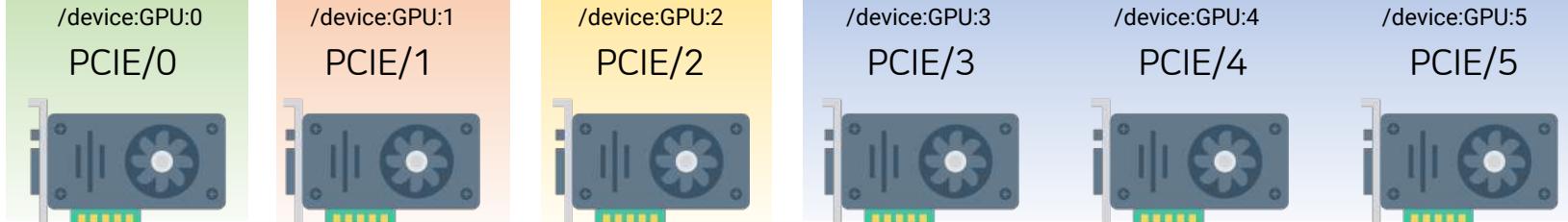
- 머신러닝 컨테이너 관련 기술들을 단일 프레임워크로 제공하는 유일한 솔루션
 - 시장에 다수 존재하는 오케스트레이션 계층은 머신러닝이 아닌 기능(예: scheduling) 및 목적(예: microservice 호스팅)에 최적화
 - 실제 머신러닝 연구자·개발자들이 겪는 고충을 둘어서 해결하는 제품은 부족함
- Backend.AI
 - GPU 최적화 기술 보유
 - ✓ NVIDIA 파트너십을 바탕으로 CUDA 최적 솔루션 구현
 - ✓ 업계 유일 컨테이너 기반 다중/부분 GPU 공유(fractional GPU™ scaling) 지원
 - 동적 샌드박싱: 프로그래밍 및 재작성 가능한 syscall 필터
 - ✓ apparmor/seccomp 등 대비 풍부한 프로그래밍 가능한 정책 지원
 - Docker 기반의 레거시 앱 리소스 제한
 - ✓ OpenBLAS와 같은 수치연산 라이브러리가 인식하는 CPU 코어 수 보정 등
 - ✓ glibc 오버라이딩을 통한 강력한 시스템 자원 제어



Backend.AI: GPU 분할 공유 & 다중 GPU 할당



Host-side
view:



Backend.AI + DGX: Backend.AI DGX 클러스터

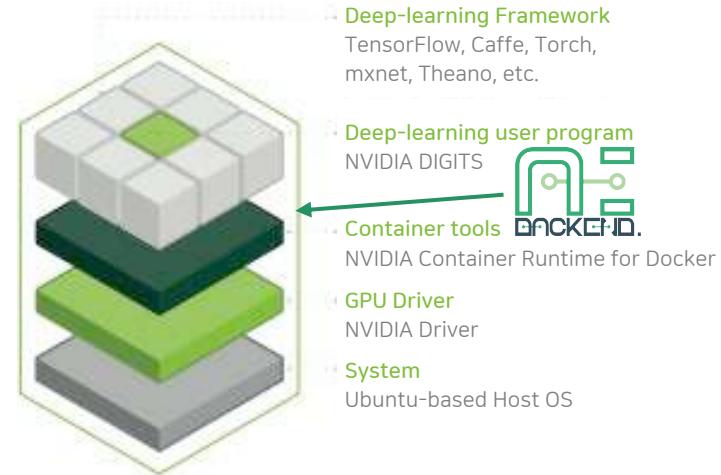


- NVIDIA DGX station/DGX-1/DGX-2/DGX-A100

- 단일 노드로 가장 강력한 다중 GPU 계산 시스템
 - ✓ Ubuntu 기반 운영체제
(GTC 2019에서 RedHat 공식지원 발표)
 - ✓ NVLink / NVSwitch를 통한 고속 GPU간 네트워킹
 - ✓ 다양한 크기의 GPU 워크로드를 돌려볼 수 있음

- Backend.AI와 DGX 패밀리의 통합

- NVIDIA가 제공하는 컨테이너 런타임에서 부족한 기능 제공
 - ✓ 다중 사용자를 위한 GPU 공유 및 할당
 - ✓ 기계학습 파이프라인을 위한 기본 구성요소 제공
 - ✓ CPU/GPU Topology를 고려한 스케줄링
 - NVIDIA Inception Partner로서 기술 협업 수행



SYSTEM SPECIFICATIONS

GPUs	16X NVIDIA® Tesla V100
GPU Memory	512GB total
Performance	2 petaFLOPS
NVIDIA CUDA® Cores	81920
NVIDIA Tensor Cores	10240
NVSwitches	12
Maximum Power Usage	10 kW
CPU	Dual Intel Xeon Platinum 8168, 2.7 GHz, 24-cores
System Memory	1.5TB

Backend.AI: NVIDIA NGC 플랫폼 통합



• NGC (NVIDIA GPU Cloud)

- nvidia-docker를 위해 최적화된 컨테이너 이미지 모음
 - ✓ NVIDIA가 직접 최적화 옵션 및 라이브러리 의존성 관리
- 이번 GTC 2019에서 model store로의 확장 발표
 - ✓ 개발완료된 딥러닝 모델들을 사용자·조직 간 공유
 - ✓ 어느 정도 학습된 딥러닝 모델을 기반으로 추가 데이터를 넣어 학습(transfer learning) 지원
 - ✓ Model script를 통한 보다 쉽고 빠른 모델 학습 환경 구축

• Backend.AI의 NGC 지원

- 모든 NGC 기반 이미지 실행 지원
- NVIDIA 권장 옵션 적용 (docker shm 제한 등 / 수정 기능도 제공)
- GPU 부분 공유 기능 적용
- NGC model store 및 model script 지원 (예정)

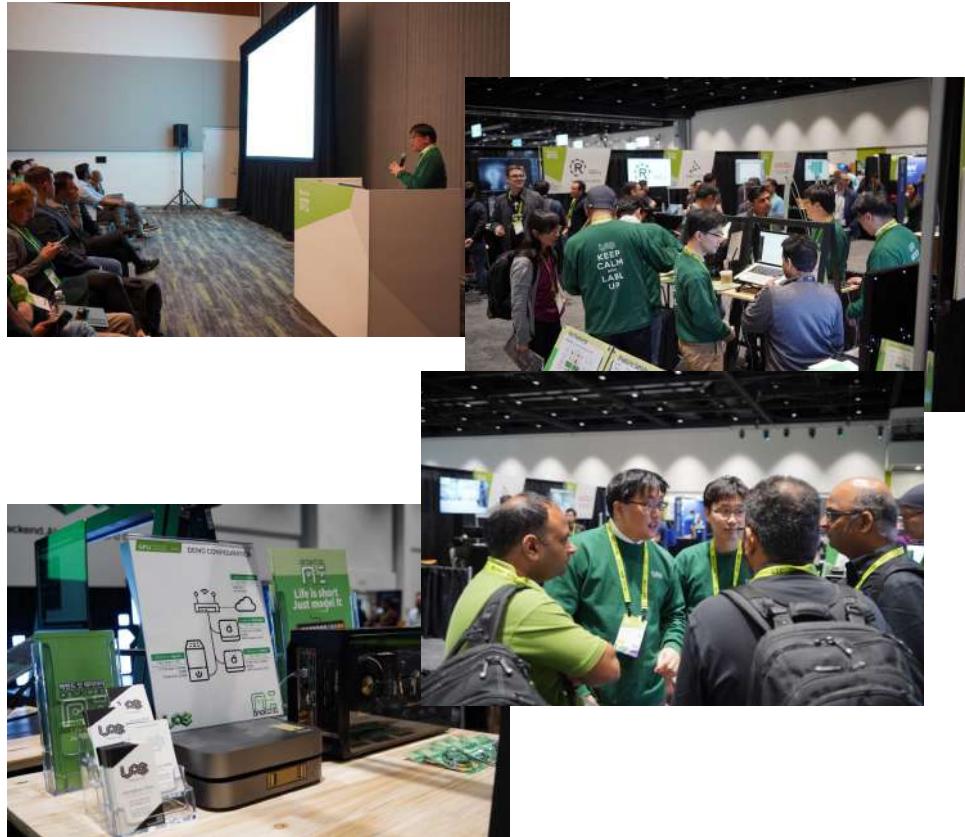
A screenshot of the NVIDIA NGC Accelerated Software interface. The top navigation bar includes 'ACCELERATED SOFTWARE', 'CONTAINERS', 'MODELS', and 'MODEL SCRIPTS'. Below this, there are two tabs: 'ALL CONTAINERS' (selected) and 'HIGH PERFORMANCE COMPUTING'. A search bar is present. The main area displays several container options:

- Caffe2:** Caffe2 is a deep-learning framework designed to easily express all model types, for example, CNN, RNN, and more, in a friendly python-based API, and execute...
Version: 18.08-py3
Built by NVIDIA
Last updated: 08/26/18
- NVIDIA:** CUDA GL
CUDA is a parallel computing platform and programming model that enables dramatic increases in computing performance by harnessing the power of the NVIDIA GPU...
Version: 10.1-runtime-ubuntu18.04
Built by NVIDIA
Last updated: 03/16/18
- NVIDIA:** CUDA
CUDA is a parallel computing platform and programming model that enables dramatic increases in computing performance by harnessing the power of the NVIDIA GPUs.
Version: latest
Built by NVIDIA
Last updated: 03/16/19
- Torch:** Torch is a scientific computing framework that offers popular neural network and optimization libraries that are easy to use yet provide maximum flexibility to build c...
Version: 10.0-py2
Built by NVIDIA
Last updated: 05/20/18
- PGI:** PGI Compilers
PGI Compilers & Tools enable development of performance-portable HPC applications with uniform source code across the most widely used parallel proc...
Version: 18.1
Built by PGI Compilers...
Last updated: 01/08/19
- MATLAB:** MATLAB is a programming platform designed for engineers and scientists. The MATLAB Deep Learning Container provides algorithms, pretrained models, and apps ...
Version: 9.0.1.16
Built by MathWorks
Last updated: 10/25/18

Backend.AI: @NVIDIA GTC Silicon Valley 2019



- DGX User Group Meetup
 - DGX 구축 사례 및 고객 요구사항 청취
- NGC User Group Meetup
 - Backend.AI의 NGC 통합 사례 발표
- Main Session Talk
 - Backend.AI 기술 소개
- Inception Startup 부스 운영
 - 컨테이너 수준 GPU 가상화 기술 시연
 - NVIDIA의 CUDA 개발자들과 직접 질의응답



Backend.AI: 경쟁 기술/플랫폼 비교



	기술 사양	nvidia-docker	Docker Swarm	OpenStack	Kubernetes	Apache Mesos	Backend.AI
GPU 지원	컨테이너별 GPU 할당	✓			✓	✓	✓
	이기종 가속기 지원 (AMD, Google TPU 등)						✓ *
	GPU 부분공유 (fractional scaling)						✓
보안	Hypervisor 및 컨테이너에 의한 가상화	✓	✓	✓	✓	✓	✓
	프로그래밍 가능한 샌드박싱						✓
가상화	VM (Hypervisor)			✓	✓ **	✓ **	✓ **
	Docker 컨테이너	✓	✓	✓	✓	✓	✓
스케줄링	슬롯 기반		✓	✓	✓	✓	✓
	고급 (DRF 등)				✓ ***	✓	✓
통합	최신 AI 프레임워크의 파이프라인 통합						✓

* 베타 버전 (20.03에 정식 포함)

** 설치된 클라우드 벤더나 VM 환경에서 간접 지원

*** 슬롯 기반의 한계가 있으나 label을 활용한 다양한 사용자화 지원



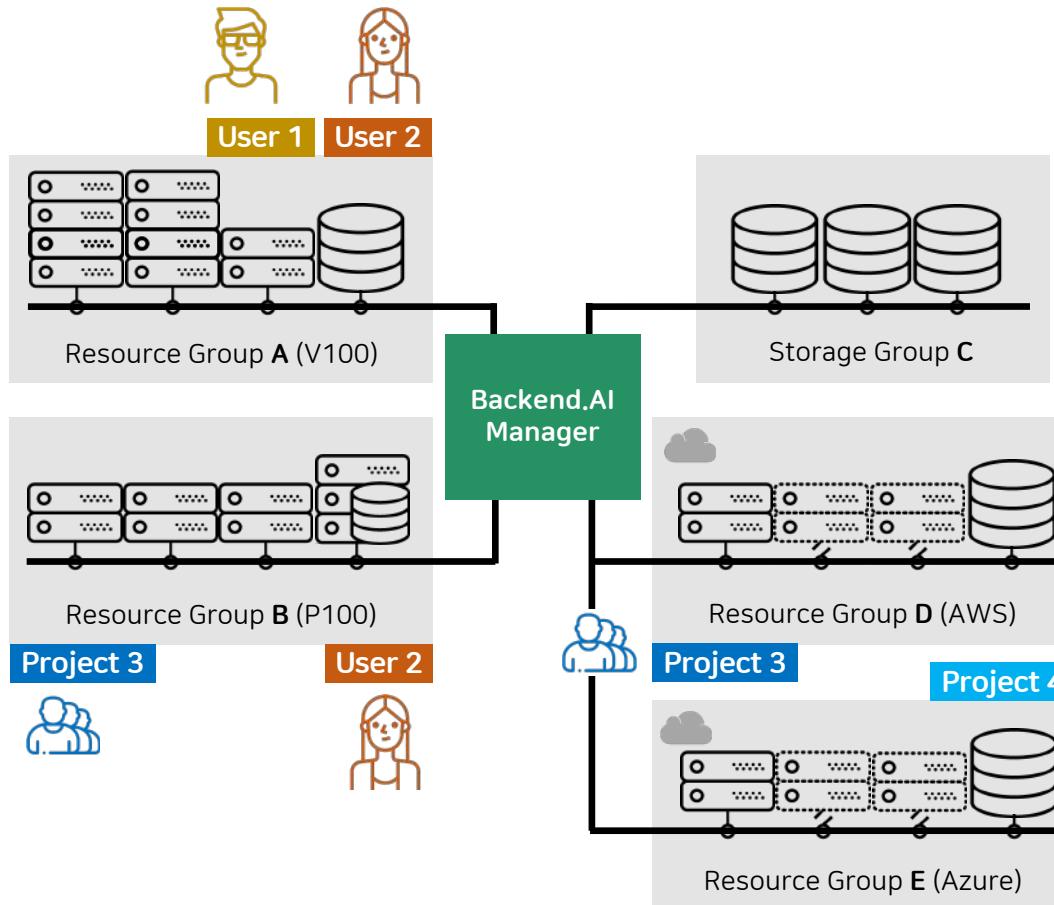
유연한 자원 관리

Backend.AI: The most efficient way to build and train your machine learning models



- **리소스 그룹**
 - 관리하는 하드웨어 자원을 묶어서 그룹화할 수 있음
 - 사용자, 프로젝트 및 도메인마다 사용 가능한 리소스 그룹을 지정할 수 있음
 - 자원 요청을 특정 리소스 그룹 안에서만 할당하도록 할 수 있음
 - 클라우드 상에서 오토스케일 구현시 리소스 그룹 단위로 적용할 수 있음
 - 팀별 워크로드 관리 및 적정 리소스 제공을 통한 비용 최적화
- **예**
 - 장치 성능 별 리소스 그룹 : A100 / V100 / P100 / K80 등
 - 노드 별 리소스 그룹 : 서버 / 워크스테이션 / IDC 등
 - 클라우드 별 리소스 그룹 : AWS / GCP / MS Azure / Naver Cloud 등
- **응용**
 - 특정 하드웨어 또는 GPU를 특정 사용자, 프로젝트, 팀 또는 도메인에만 할당할 경우
 - CPU / GPU / 저장장치별로 노드 그룹을 구분하고 싶을 경우
 - 멀티 네트워크 클러스터의 경우, 물리적으로 동일한 네트워크 안에 있는 노드들을 묶어 관리하고 싶을 경우

유연한 자원 관리: 리소스 그룹 시나리오



• 리소스 그룹 응용 예

- RG A: NVIDIA V100 GPU 그룹
- RG B: NVIDIA P100 GPU 그룹
- 이 경우 User 1은 V100만을, User 2는 V100 및 P100을 모두 사용할 수 있음
- Project 3은 P100 그룹 및 AWS 클라우드를 사용할 수 있음
- Project 4는 Microsoft Azure 클라우드만 사용할 수 있음



사용자 친화 GUI

Backend.AI: The most efficient way to build and train your machine learning models



사용자 친화 GUI: 자원 현황 확인



The screenshot displays the Backend.AI Cloud (Beta) user interface, specifically the resource monitoring dashboard. The top navigation bar includes the Labup logo, the title "Backend.AI Cloud (Beta)", the current project "gardener", the user "Labup User", and a settings icon.

The left sidebar contains a navigation menu with the following items:

- 요약 (Summary)
- 세션 (Sessions)
- 가져오기 & 실행 (Pull & Run)
- 데이터 & 폴더 (Data & Folders)
- 동기 (Sync)
- 관리 (Management)
- 사용자 (User)
- 실행 환경 (Execution Environment)
- 리소스 (Resources)
- 환경 설정 (Environment Settings)
- 관리 (Management)
- 정보 (Information)

The main content area features several cards:

- 대시보드 (Dashboard):** Includes sections for "시작" (Start), "파일 업로드" (File Upload), "새 키페이 생성" (Create New KeyPair), and "키페이 관리" (KeyPair Management). A prominent red-bordered button labeled "+ 시작" (Start) is located in the "시작" section.
- 자원 사용량 (Resource Usage):** Shows resource utilization for the "nvidia-GPU" group. It includes metrics for CPU, RAM, and GPU usage across 13 nodes.
- 시스템 자원 (System Resources):** Displays system resources across 3 nodes, including 16 CUDA GPUs. It shows resource usage for CPU, RAM, and GPU, with a legend indicating reserved, used, and total resources.
- 공지 (Announcement):** Welcomes users to the Backend.AI cloud open-beta.
- 관리 (Management):** Provides information about the Manager and Console versions (Manager version: 20.03.0rc1.dev0, Console version: 20.07.6 ✓) and links to system status and configuration.

The bottom status bar indicates "status: healthy".

사용자 친화 GUI: 환경 및 자원 선택 (즉시 생성)



The screenshot displays the BackendAI Cloud (Beta) user interface, specifically the session management and configuration features.

Left Sidebar: Includes sections for BackendAI Cloud (Beta), Session Management (Session, Session Details, Session History, Session Details), Resource Management (User, Running Environments, Resources, Environment Settings, Management, Information), and Help.

Middle Panel: Shows a list of sessions with columns for #, User ID, Session ID, and Session Name.

Center Overlay: A modal window titled "새 세션 시작" (New Session Start) is open, listing available languages and environments:

- Language:** Python, GCC, Fortran
- Machine Learning / Deep Learning:** TensorFlow (selected), PyTorch, MXNet
- Platform:** Labup Platform, TensorFlow, PyTorch, DIGITS, Caffe, Caffe 2, TensorFlow (Intel)
- Scientific Language:** R

Right Panel: A table showing a list of active sessions, including columns for Session ID, User, Language, Usage, Start Time, and Duration.

세션 ID	현재 프로젝트	사용자 이름	언어	사용량	시작시간	예약시간
bayo	gardener	Labup User	TensorFlow	Nvidia-GPU 8.00GB	7/26/2020, 5:04:01 PM	00:56
pwSr	gardener	Labup User	TensorFlow	Nvidia-GPU 8.00GB	7/26/2020, 4:31:24 PM	01:29
3	minieklam.com	Labup User	PyTorch	Nvidia-GPU 8.00GB	7/26/2020, 4:06:09 PM	01:54
4	gardener@never.com	Labup User	TensorFlow	Nvidia-GPU 8.00GB	7/26/2020, 4:05:33 PM	01:57
5	syokuk@gmail.com	Labup User	PyTorch	Nvidia-GPU 8.00GB	7/26/2020, 3:52:07 PM	02:01
6	rmitra@gmail.com	Labup User	DIGITS	Nvidia-GPU 8.00GB	7/26/2020, 3:42:33 PM	02:18
7	tykim@korea.ac.kr	Labup User	Caffe	Nvidia-GPU 8.00GB	7/26/2020, 3:23:34 PM	02:37
8	pavilH@gmail.com	Labup User	Caffe 2	Nvidia-GPU 6.00GB	7/26/2020, 2:49:37 PM	03:11

사용자 친화 GUI: 환경 및 자원 선택 (즉시 생성)



BackendAI Cloud (Beta)

세션

실행 중 종료

세션

가져오기 & 실행

데이터 & 폴더

통계

관리

사용자

실행 환경

리소스

환경 설정

관리

정보

제작 개발 툴을 지원합니다.

Lacb python 3.6.5

②

⚙️

새 세션 시작

실행 환경* TensorFlow

버전* 2.3 / Python 3.6 / CUDA10.1

리소스 그룹* nvidia-GPU

세션 이름 (옵션)

마운트할 폴더 CORD-19

자원 할당* medium (2CPU 6GB 1GB)

Name	CPU	RAM	공유 메모리	가속기
large	4	8GB	1GB	
large-gpu	4	8GB	1GB	2 GPU
medium	2	6GB	1GB	
medium-gpu	2	6GB	1GB	1 GPU
research-gpu	8	48GB	2GB	4.5 GPU
small	1	4GB	64MB	

+ 시작

현재 프로젝트 gardener 사용자 이름 Labup User

+ 시작

작업	사용량	시작시간	예약시간
V100-GPU	No data	7/26/2020, 5:04:01 PM	02:00
V100-GPU	No data	7/26/2020, 4:31:24 PM	02:35
V100-GPU	No data	7/26/2020, 4:06:09 PM	03:05
V100-GPU	No data	7/26/2020, 4:05:33 PM	03:07
V100-GPU	No data	7/26/2020, 3:52:07 PM	03:39
V100-GPU	No data	7/26/2020, 3:42:33 PM	03:28
V100-GPU	No data	7/26/2020, 3:23:34 PM	03:47
V100-GPU	No data	7/26/2020, 2:12:55 PM	04:58

사용자 친화 GUI: 인터랙티브 개발 툴 지원



The screenshot shows a Jupyter Notebook interface running on TensorFlow 2.1 on Python 3.6 (CUDA 10.0). The left sidebar displays a file tree with folders like 'labels', 'prototyping', 'id_container', and the current notebook 'Untitled.ipynb'. The main area shows three code cells:

```
[1]: print("Hello World")
Hello World

[2]: import tensorflow as tf
print(tf.__version__)
2.1.0

[3]: from tensorflow.python.client import device_lib
print(device_lib.list_local_devices())
In [3]:
{
  "name": "/device:CPU:0",
  "device_type": "CPU",
  "memory_limit": 268435456,
  "locality": {},
  "incarnation": 14007344171145880303,
  "name": "/device:XLA_CPU:0",
  "device_type": "XLA_CPU",
  "memory_limit": 17179869184,
  "locality": {},
  "incarnation": 4361465032883859367,
  "physical_device_desc": "device: XLA_CPU device",
  "name": "/device:XLA_GPU:0",
  "device_type": "XLA_GPU",
  "memory_limit": 17179869184,
  "locality": {},
  "incarnation": 800888443607757140,
  "physical_device_desc": "device: XLA_GPU device"
}
```

The status bar at the bottom indicates "TensorFlow 2.1 on Python 3.6 (CUDA 10.0) | Idle".

사용자 친화 GUI: 데이터 관리



Backend.AI
Cloud (Beta)

데이터 & 폴더

현재 프로젝트: gardener 사용자 이름: Lablup User

폴더 자동 마운트 폴더 공유 데이터 모델

+ 새 폴더

#	이름	id	위치	종류	권한	제어
1	prototyping	bc5982027c1341990188060@024cc613	L3	파일	R W D	
2	labels	21bdcd9311b44b5fa0081cb3c27dc384	L3	파일	R W D	
3	tfrecords-output	fb1ca06f7af24c61a3145c5fbc7b2841	L3	파일	R W D	
4	CORD-19	647b6a40:c7e94bd881b087602b16026d	L3	파일	R W D	
5	kernel-test	e65d1253fd384ce69b6f2ef33d8b1eac	L3	파일	R W D	

관리

사용자

실행 환경

리소스

환경 설정

관리

정보

약관 · 개인정보보호 · 대하여

Lablup Inc. 2018-2023

?

⚙️

사용자 친화 GUI: CPU 노드 / GPU 노드 통합 관리



Backend.AI Cloud (Beta)

연산 자원

연결중 종료됨 보수중 자원 그룹

#	엔드포인트	지역	시작	자원	자원 그룹	상태	제어
1	i-baryon0x tcp://172.32.0.4:6001	local	2019. 8. 7. 오후 6:50:18	79코어 168.19GB 753GB 20.55GB 48GPU	nvidia-GPU	Agent 20.03.0b2 CUDA Plugin 1.4.0 Fractional GPU™ CUDA 10.2	
2	i-turbo tcp://172.32.0.13:6001	local	2020. 1. 13. 오후 4:31:08	95코어 0.40GB 100.5GB 10.21GB 48GPU	nvidia-RTX	Agent 20.03.0b2 CUDA Plugin 1.4.0 Fractional GPU™ CUDA 10.2	
3	i-hydro01 tcp://172.32.0.221:6001	local	2020. 1. 12. 오후 11:48:48	255코어 0.60GB 100.6GB 11.93GB	HPC	Agent 20.03.0b2 CUDA Plugin 1.4.0 CUDA Disabled	

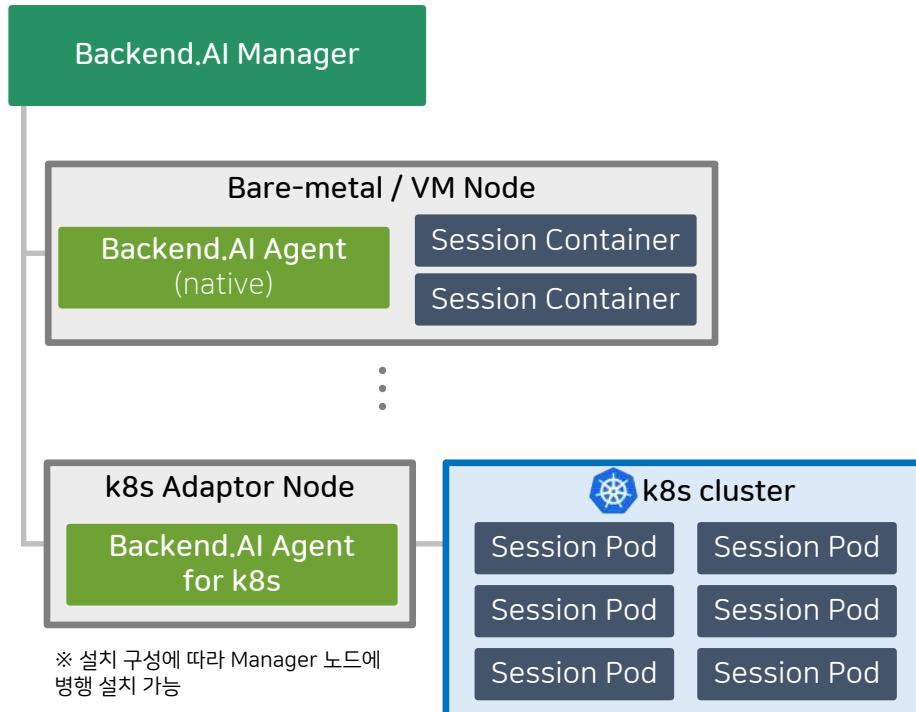
관리 사용자 실행 환경 리소스 환경 설정 관리 정보

액션 | 개인정보보호 | 대하여
Lablup Inc. 2018-2022

GPU Node

CPU 전용 Node

- GPU 워크로드는 GPU 노드에 배치 (딥러닝, 구조 설계, 이미지 분석 등)
- CPU만 사용하는 워크로드는 CPU 노드로 자동 배치 (CPU 고성능컴퓨팅)



- **k8s 연동 어댑터**

- k8s cluster 전체가 하나의 Backend.AI Agent로 가상화하여 마치 하나의 연산 노드인 것처럼 사용
- Native Backend.AI Agent 노드를 함께 활용할 수 있음
- Fractional GPU scaling은 지원하지 않음

- **스케줄링 방식**

- "k8s" Agent로의 세션 할당은 기존 Backend.AI의 스케줄러가 담당
- k8s 내에서의 pod 실행은 기존 k8s 스케줄러가 담당
- k8s cluster의 용량이 꽉 찬 경우 Backend.AI 쪽 스케줄러의 대기 큐 적용



구축 사례

Backend.AI: The most efficient way to build and train your machine learning models





- **클러스터 팜 구축**
 - 최적의 Backend.AI 클러스터를 구축할 수 있도록 하드웨어+소프트웨어+네트워크 구축 컨설팅 제공
 - 사용자 그룹의 워크로드에 따라 최적화된 형태의 GPU 도입 컨설팅 및 토플로지 설계
- **확장성 고려 설계**
 - 이후 시스템 확장을 고려하는 경우, 유연한 확장을 제공하기 위한 랙 모듈 단위의 설계 제공
- **어플라이언스 에디션**
 - 특정 장비 도입 시, 해당 장비에 특화된 기능을 유기적으로 통합하여 제공
 - 컴퓨팅 장비 / 스토리지 장비 등
 - ✓ DGX 시스템 통합 제공
 - ✓ 이후 계속 추가 예정
 - 라이센싱, 모니터링 및 성능 최적화 명령어 통합

Backend.AI 구축 사례: cloud.backend.ai (멀티클라우드+하이브리드 클라우드)

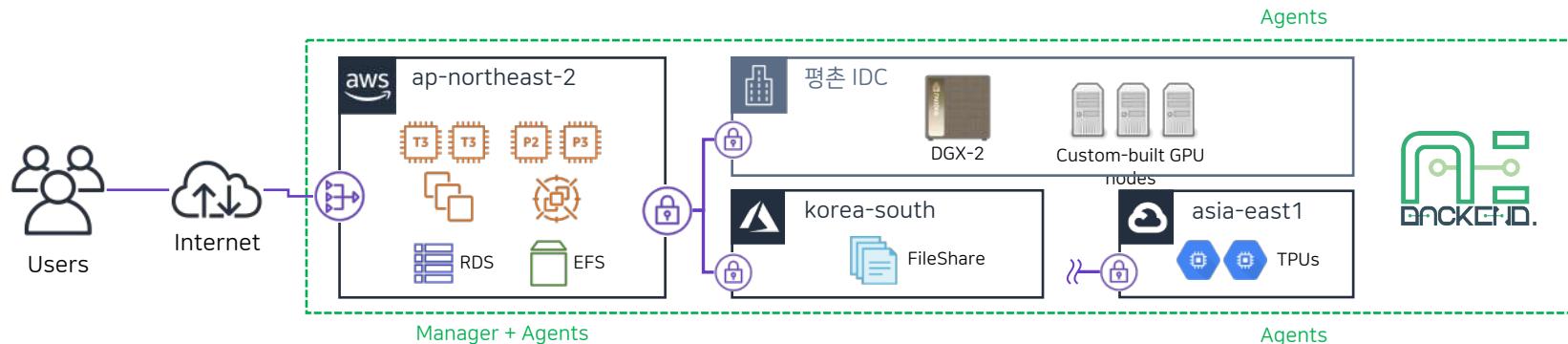


- 래블업에서 직접 운영하는 Backend.AI cloud 서비스 (B2B / B2C)

- B2B : 다양한 고객 수요에 특화된 Full-featured machine learning cloud 유료 공급
- B2C : 가입만으로 바로 Backend.AI를 실행해볼 수 있는 베타 서비스(Invitation 필요)
- <https://cloud.backend.ai>

- 구성

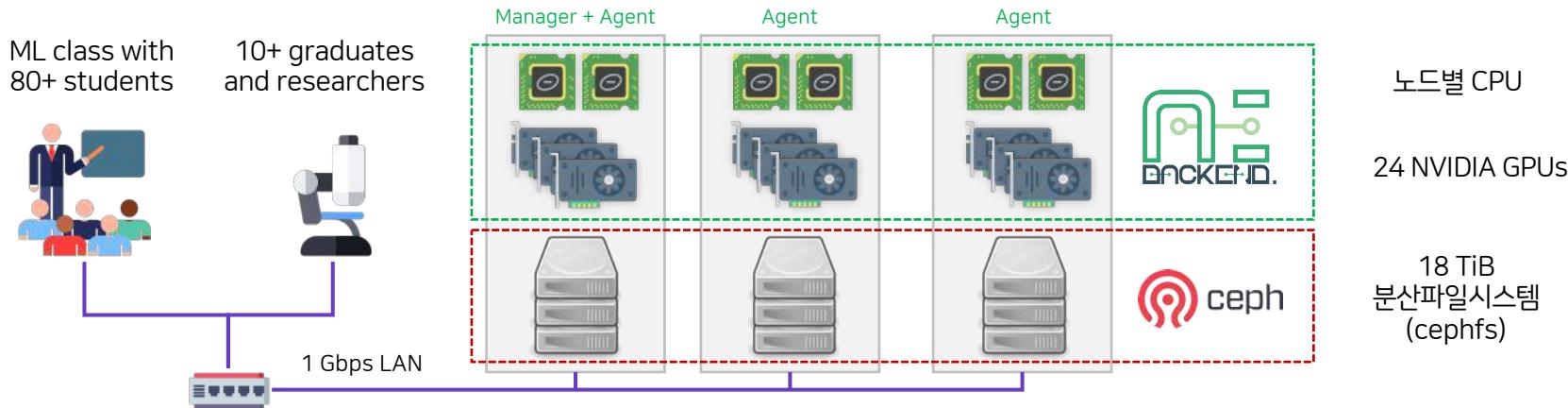
- AWS + Azure + GCP 통합 (평균 20 노드 / 가변 150 노드)
- Google TPU 지원 (beta)
- 데이터 저장을 위해 Azure FileShare + AWS EFS (Elastic File System) + On-prem 스토리지 연동



Backend.AI 구축 사례: 대학A (연구/실습 워크로드 통합 제공)

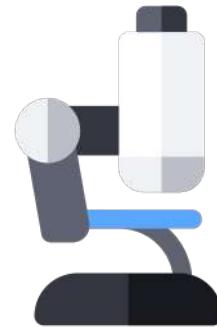


- 금융권 실무자 MBA 수강생 및 연구자들을 위한 GPU 서버팜 구축
 - 3대 서버 24개의 GPU로 80명 이상의 수강생들이 동시에 실습
 - 다수 연구자들은 동일 시스템을 이용하여 모델링 수행
- 구성
 - 4 노드 / 멀티 GPU 조합 (24식)
 - 노드별 대용량 HDD를 LAN으로 묶어 18 TiB ceph 분산 파일시스템 구축 및 연동
 - 시기에 따라 순차적 확장 (2019년 3분기 1차 확장 / 2020년 2분기 (예정))



- 고객 혜택

- 제한된 갯수의 GPU를 가지고 훨씬 많은 인원 수의 사용자들이 동시에 사용
- 수강생·연구자로 구분된 resource policy 적용
- 학내 사용 가능한 Web GUI로 주요 기능을 제공하여 전담 관리자 없이도 최적 운영
- 학기 중: AI 수업 및 연구 워크로드를 시간대에 따라 탄력 조정하며 사용
- 방학 기간: 기업 AI 위탁 교육 플랫폼으로 이용



Backend.AI 구축 사례: K1연구소 (하이브리드 클라우드 구성)



- CPU 워크로드에 특화한 랩 단위의 고성능 컴퓨팅 클러스터 플랫폼 (빅데이터 분석용)

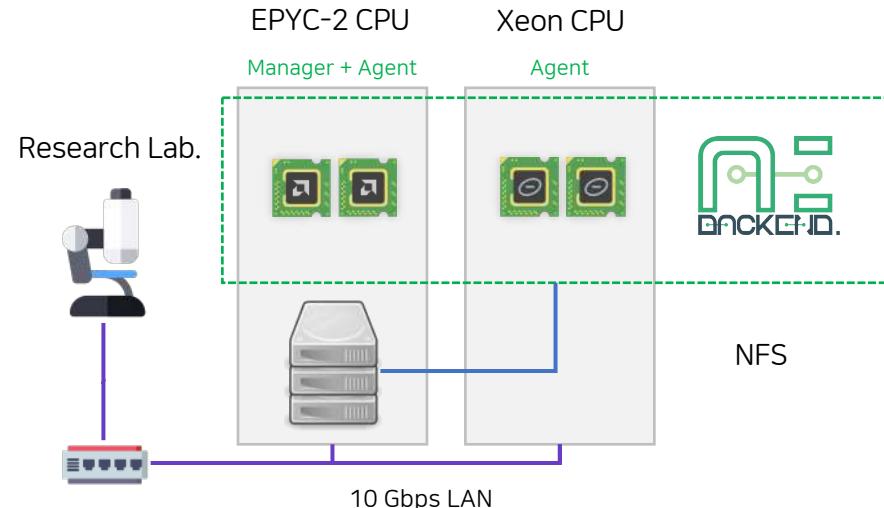
- 빅데이터 가공 및 필터링 / 데이터 분석
 - 예측 모델 파라미터 튜닝

- 구성

- 180코어 / 2 노드 (128 EPYC-2 + 52 Xeon)
 - ✓ 서로 다른 리소스 그룹으로 묶음
 - ✓ AVX512 사용 (Xeon) / 멀티코어 워크로드 (EPYC-2)
 - ✓ GPU 없음
 - Python, C++ 사용
 - 클라우드 인스턴스로 추가 확장 사용
 - ✓ AWS EC2 / p2.xlarge 또는 p2.8xlarge 를 워크로드에 따라 연결

- 고객 혜택

- 기관 내외 조직이 공동 연구 프로젝트 수행시 외부 기관에 공동 연구 환경 제공
 - 하나의 플랫폼 / 하나의 코드 / 하나의 데이터레이아웃을 다양한 CPU 플랫폼 중 필요에 따라 즉시 선택하여 이용



Backend.AI 구축 사례: K2연구소 (appliance 대여)



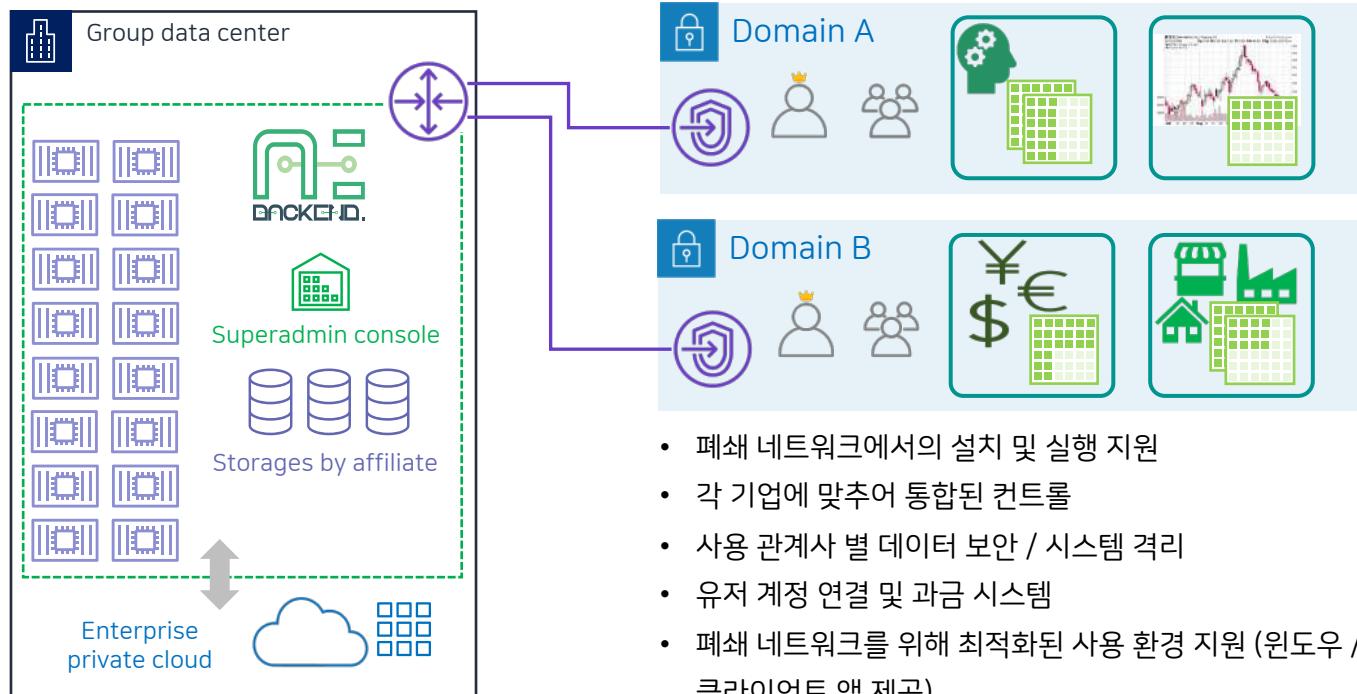
- 질병 연구 프로젝트 (2018~)
 - 데이터 분석 및 실험 데이터로부터 feature 추출
 - 딥러닝 모델 훈련
 - 실험 데이터 아카이빙 및 즉시 분석
 - 다수 사용자들의 동시 연구 코드 작업
- 구성
 - Skylake 기반 4 계산 노드
 - ✓ Backend.AI 매니저는 Lablup에서 대여
 - ✓ Agent는 필요에 따라 순차적으로 추가
 - 데이터 스토리지 1 노드 (Synology)
- 고객 혜택
 - **잘 구성된 온사이트 데이터 프로세싱**을 위한 온프레미스 솔루션 사용
 - **기존 연구용 코드를 거의 수정하지 않고**, 데이터로부터 결과로 이어지는 워크플로우 도입



Backend.AI 구축 사례: 그룹A (다중 계층 조직 관리)



- 그룹 내 관계사 별 독립 GPU 클라우드 오케스트레이션 / 스케줄링 레이어 제공

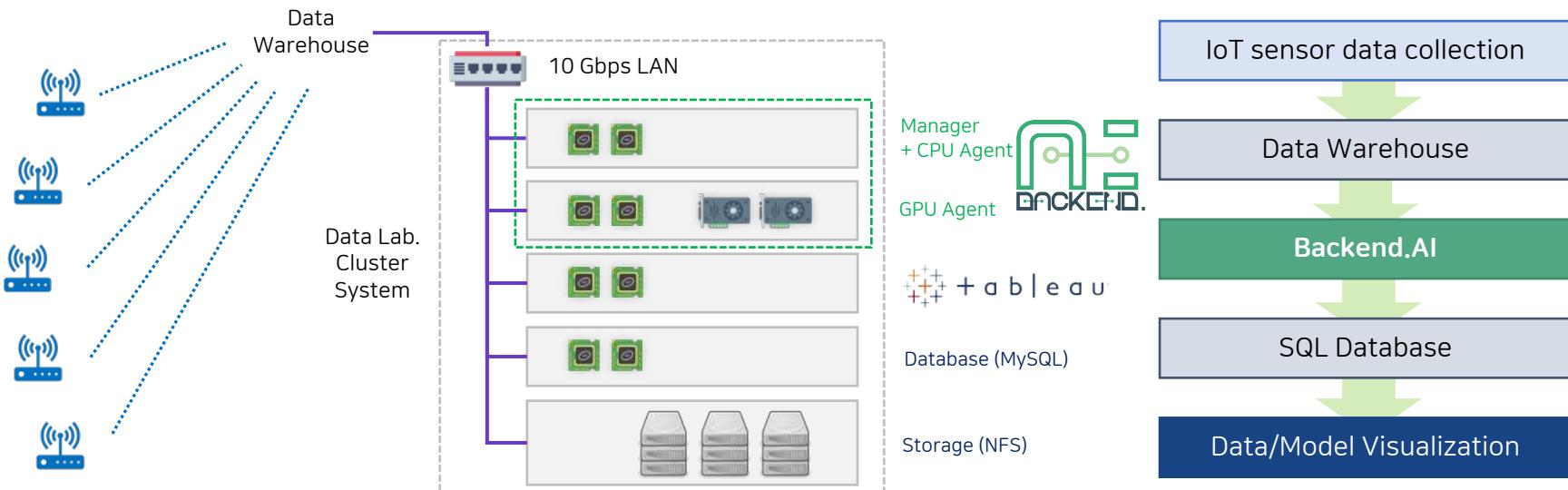


Backend.AI 구축 사례: 그룹B (Smart Factory / IoT 파이프라인)



• IoT-AI-BI 파이프라인 구축

- 도입 규모 : 1 rack (GPU + CPU + Storage) / 연산자원 규모 지속적으로 확장 예정
- 연구 내용 : 대규모 규모 공장 내의 자재 배치·재고 현황 모니터링 및 운송·보관 경로 최적화
- IoT를 활용한 실시간 센싱 데이터 수집 및 이를 자동으로 가공하여 지속적으로 AI 모델을 업데이트하는 파이프라인 구축



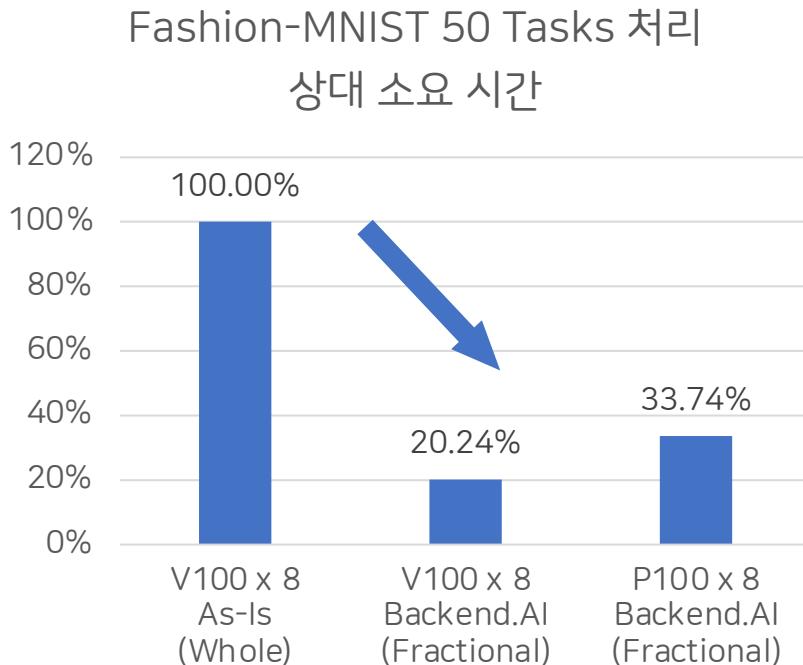


GPU 가상화 성능

Backend.AI: The most efficient way to build and train your machine learning models



Backend.AI: GPU 가상화 성능 벤치마크



- 예제코드 : fashion-MNIST
- V100/P100 GPU 클러스터 (1-node 8 GPU)
 - 실행 모델 수 : 50 개
 - 비교 Case
 - ✓ V100 As-Is 단독모드 (Whole)
 - ✓ V100 Backend.AI 공유모드 (Fractional *)
 - ✓ P100** Backend.AI 공유모드 (Fractional)

* Note, Fraction : SMP 4개, GPU Memory 1 GiB

** Note, P100은 V100 보다 한 세대 이전 GPU



GPU 가상화와 자원 관리 자동화로
고가 GPU 자원의 가동률 제고와 최적 활용 달성

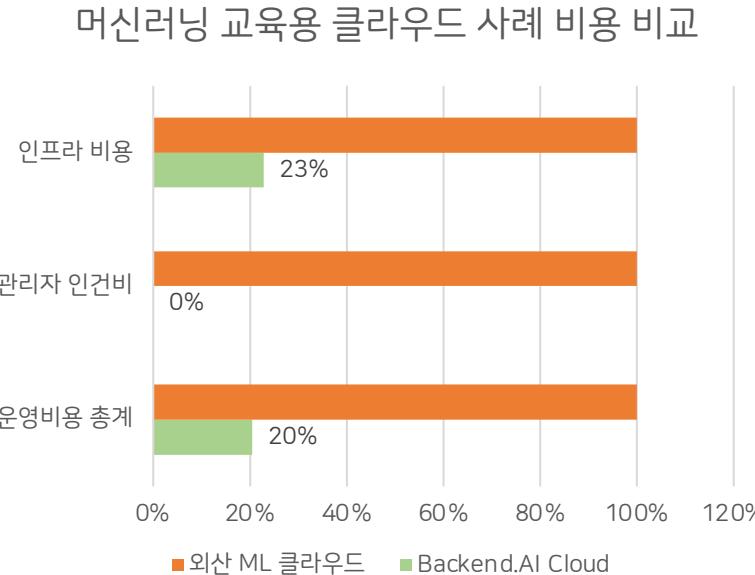


Backend.AI 를 통한 비용 절감

Backend.AI: The most efficient way to build and train your machine learning models



사례: 머신러닝 교육용 클라우드



- **머신러닝 교육 및 개발 클라우드 서비스 공급**
 - 사용자 25인
 - 교육 기간 학기당 2개월
- **GPU 가상화를 통한 교육/개발 각각에 대한 최적 활용 제공**
 - GPU 분할 공유를 통한 자원 최적 활용으로 인프라 비용 75% 이상 절감
- **GUI를 통한 자원 할당 관리 자동화, 개발 환경 제공**
 - Web GUI로 주요 기능을 자동화하여 제공, 전담 관리자 없이도 최적 운영
 - 장기적 유지보수 부담 원천 제거



GPU 가상화를 통한 인프라 비용 저감과 관리 부담 제거는 설치형 솔루션에도 공히 적용됨.



로드맵

Backend.AI: The most efficient way to build and train your machine learning models





Cloud

Fits with your needs
instantly

계산 기반 과학 및
딥러닝 모델링을 위한
On-demand GPU
개발환경 서비스

Open Source

Get the most out of
your hardware

자신만의 Backend.AI
서버팜을 설치하고
개발할 수 있는
오픈소스 버전

Enterprise

End-to-end
ML Infra Manager

대규모 및 고급
기업·기관 사용자를
위한
전용 GPU 클라우드
구축 솔루션

BETA

MLOps

Super-convenient
ML Pipeline Interface

딥러닝 서비스 구축 및
대규모 훈련을 위한
종단간 기계학습
관리 도구 및 인터페이스



Backend.AI Ground

- 오픈소스 설치형 프레임워크
- 코어 + 라이브러리 + 플러그인
- 듀얼 라이선스
 - 비상업용: GPLv3
 - 상업용: 별도 계약

Backend.AI Cloud

- Ground 기반으로 Lablup에서 직접 운영하는 클라우드
- HTTP 기반 Open API 제공으로 외부 서비스 연동 개발 가능
 - 예: 웹 기반 연구환경, 웹 기반 기계학습 실습 환경, Online Judge 구축
- Freemium 모델

Backend.AI Enterprise

- **유료 엔터프라이즈 솔루션**
 - 데이터 보안 환경(금융기관, 의료기관 등)을 위한 AI 개발용 전사 통합 플랫폼
 - 인터넷 속도에 영향 없는 오프라인형 대규모(100명 이상) 기계학습 개발자 교육용 플랫폼
 - 서비스 선탑재 appliance 옵션
- **온프레미스 구독 / 서포트 플랜 제공**
 - 초기 설치 및 기술 문의 지원
- **고급 관리 기능 제공**
 - 폐쇄망 전용 오프라인 설치 도구
 - 고가용성 구성 지원
 - 도메인 / 조직 관리 전용 서비스
 - 샌드박스 보안 로깅
 - Fractional GPU scaling
 - 사설 Docker Registry 연동

MLOps

- MLOps를 위한 일관된 사용 체계
 - 파이프라인용 모듈 인터페이스
 - 파이프라인 스케줄링 및 실행 기능
 - 배치 스케줄링 및 실행 예약
 - MLFlow 통합 지원
- Ground 또는 Cloud가 제공하는 런타임 환경을 활용하는 웹 앱
- Ground와 동일한 라이선스 정책

Backend.AI: 제품 포트폴리오 (Enterprise R2)



Essential

엔터프라이즈 환경을 위한
딥러닝 연구 플랫폼

Pro

서비스 운영 및
프로덕션 모델 개발용
토탈 솔루션

Edu

교육 실습 및
공동 연구에 최적화된
딥러닝 플랫폼

Reservoir

완전 폐쇄환경 운영을 위한
Backend.AI와 통합된
패키지 저장소

Manager

Agent

Console

Environment Hub

Admin. Hub

fGPU™ Plugin

Essential

AutoML

Model Serving

Pipeline

ModelStore

DataStore

Essential

Education

ModelStore

DataStore

PyPi

APT

RPM / Yum

Security Checker

BETA

Backend.AI: 설치 요구 조건



Specification		Description	
Overall	Minimum operation spec.	S/W requirements	OS
			Ubuntu 16.04 / RHEL CentOS 7.5 or later Docker 18.09 or later nvidia-docker v1 / v2 (optional) CUDA 8.0 or later (optional) Backend.AI Agent requires the root account or the following Linux capabilities: cap_sys_ptrace, cap_sys_admin, cap_dac_override
H/W requirements		Manager	Max. manageable agent nodes : 2,000 Max. multiple containers for a single session : 20
CPU	Capacity limits	Max. manageable containers : 100 Max. GPUs on a node : 16	
		Max. GPUs in a single container : 16 < 10% CPU 1core usage for simultaneous sessions by 50 users (in case 2.3GHz Intel Sandybridge)	
GPU	Performance	Depend on user code, H/W, computation framework configuration	
		Overhead of Backend.AI and docker : <1%	
Network	Manager		
	Computation		
Storage	Computation		
	Computation		

Backend.AI: 개발 로드맵 (~현재)



Alpha

Beta

Production-ready

2015.8

프로젝트 공개 (PyCon KR)

2018.1~3

v1.1 ~ v1.3 릴리즈

- 코드 안정화
- 설치프로그램 추가
- 플러그인 구조
- 브랜치 관리 규칙 적용

2016.11

v0.9 릴리즈

- GitHub을 통한 최초 코드 공개
- LGPLv3 라이선스 적용 (서버)
- MIT 라이선스 적용 (클라이언트)

2017.10

v1.0 릴리즈

- REPL 기능 안정화
- 개발 매뉴얼 제공
- 가상폴더 기능 추가
- PyPI 공개 (pip install)

2018.9

v1.4 릴리즈

- GPU 부분공유 기능 첫 구현

2018.12

v18.12 릴리즈

- 버전 번호 부여 정책 변경 (연.월)
- 공개 및 사설 Docker registry 연동
- Google TPU 지원 추가

2019.3

v19.03 릴리즈

- GPU 스케일링 개선
- Nvidia GPU Cloud 통합
- 관리자용 기능 확장
- 부하 테스트 적용 및 안정화

2019.9

v19.09 (Enterprise R1/R2 *19.09.15 이후) 릴리즈

- GPU 부분 공유 및 가상화 고도화
- 연산 자원 리소스 그룹 기능
- 이메일 기반의 사용자 관리 기능
- 도메인별 관리 기능 및 SSO 지원
- 엔터프라이즈용 허브 기능 / 고가용성 지원

2020.6

v20.03 릴리즈

- 표준화된 파이프라인 모듈 인터페이스
- 실행 시작 기준 스케줄링
- 종단간 파이프라인 작성 기능 추가 (베타)
- 멀티 매니저 / 원격 매니저 지원
- k8s pod 연동 지원 (베타)

시간

Backend.AI: 개발 로드맵 (2020~)



Enterprise R2 (20.03/09)

2020.6

v20.03 릴리즈

- 표준화된 파이프라인 모듈 인터페이스
- 실행 시작 기준 스케줄링
- 중단간 파이프라인 기능 (beta)
- 멀티 매니저 / 원격 매니저 지원
- LustreFS / GlusterFS 분산처리 파일 시스템 지원
- DGX-A100 지원 (G/A)
- k8s pod 연동 지원 (beta)
- Google TPU 지원 (beta)
- AMD ROCm 지원 (beta)
- DGX 통합 지원 (beta)
- GUI 리눅스 데스크탑 터널링 지원 (beta)

Enterprise R3 (21.03/09)

2021.3

v21.03 릴리즈

- 데이터스토어 지원 (G/A)
- 모델 스토어 지원 (G/A)
- 익명 모델 배치 지원
- Lambda 방식의 오토스케일 모델 배치 지원
- Web Console 지원 (G/A)

2020.9

v20.09 릴리즈

- 멀티컨테이너 세션 지원 (G/A)
- 데이터 파이프라인
- 비공개/개인 모델 배치 지원
- Storage Proxy 기반 PureStorage 통합 지원
- MLFlow 통합
- Callosum 보안 터널 연결 도입 (G/A)
- DGX 클러스터 시스템 통합 지원 (G/A)
- k8s pod 연동 지원 (G/A)
- Google TPU 지원 (G/A)
- AMD ROCm 지원 (G/A)
- CUDA MPX 지원 (beta)
- 데이터스토어 지원 (beta)
- 모델 스토어 지원 (beta)

시간

