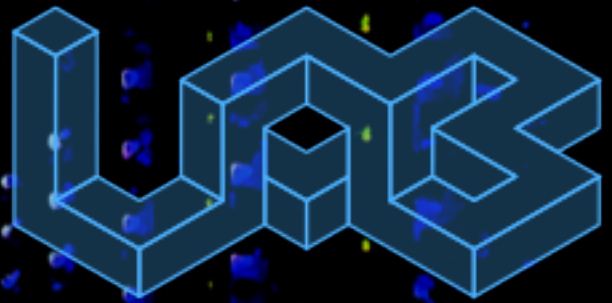
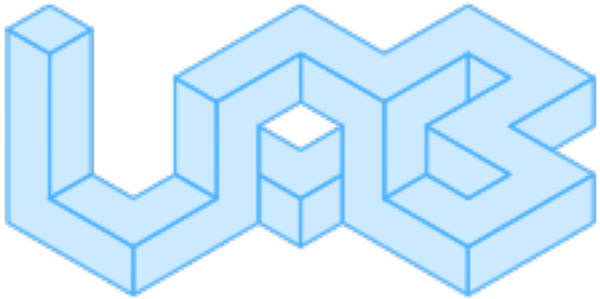
KOREA OpenInfra Days 2018



Backend.AI
오픈소스 머신러닝 인프라 프레임워크

Lablup Inc.



Backend.AI

Backend.AI: Make AI Accessible

문제 의식

- 오픈소스와 클라우드의 발달 – 정말로 AI 하기가 쉬워졌을까?

신이 AI 분야를 만들 때

http://kr.vonvon.me/quiz/329,
http://kr.vonvon.me/quiz/1633

기계학습 모델을 훈련하고

실행하는 **모든 과정**을

클라우드 또는 **자신의 서버**에서

엄청나게 **쉽고 빠르게** 돌려주는

세련된 플랫폼

우리에게 필요한 것

쉽게

- 원래 사용하던 방식과 같도록
- 최소한의 환경설정만으로

빠르게

- GPU를 잘 활용할 수 있도록
- 원하는 만큼 연산자원을 바로바로

값싸게

- 사용한 만큼만 지불하도록
- 비용 제한에 맞춰 성능 조정을 자동으로

함께

- 한번 만들면 누구나 똑같이 재현하고 재사용하도록
- 다른 사람들과 충돌 없이 자원을 공유할 수 있도록

어디서나

- 내가 가진 서버를 지금 바로 활용하거나
- 이도저도 귀찮다면 클라우드에 맡겨서

Backend.AI 목표

쉽게

- Jupyter, Visual Studio Code, IntelliJ, Atom 플러그인 제공
- API key만 설정하면 끝!

빠르게

- 컨테이너와 GPU 기술을 결합
- 1초 이내에 연산 세션이 뜨도록

값싸게

- 밀리초/KiB 단위까지 정밀한 자원 사용 측정
- (향후 지원 예정!)

함께

- 컨테이너를 활용한 언어별·버전별 가상 환경 제공
- 시스템콜 샌드박싱 + Docker의 자원제한 고도화

어디서나

- 오픈소스 버전 제공 (www.backend.ai)
- 클라우드 버전 제공 (cloud.backend.ai)

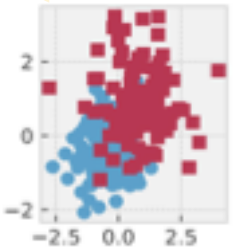
Backend.AI 개념도

기계학습 및 과학연산 코드

```
import tensorflow as tf
import matplotlib
v1 = tf.Variable(…, name="v1")
v2 = tf.Variable(…, name="v2")
...
plot(...)
```

개인·조직 사용자

REST/GraphQL API



실행 결과

응용 서비스 및 애플리케이션


클라우드 서비스


웹 서비스


모바일 및 IoT 장비

Backend.AI 플랫폼


오토스케일링


보안격리


버전관리


언어별 SDK


자원할당


모니터링

클라우드 인프라 (IaaS)

온-프레미스 클러스터

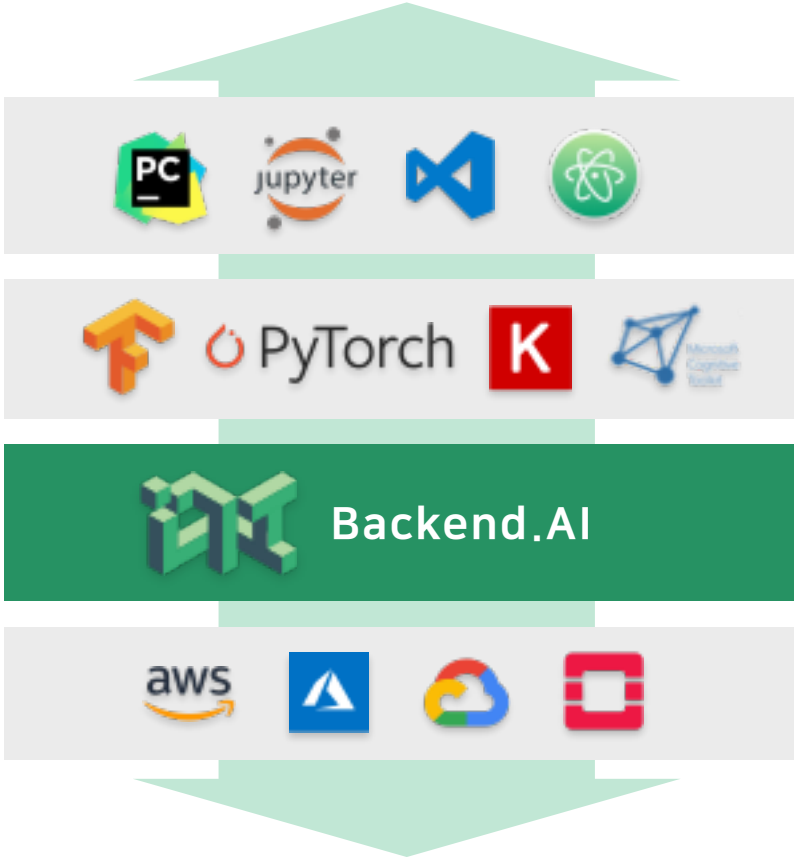
Backend.AI 포지셔닝

개발자를 위한
사용자 인터페이스

GPU를 활용하는
다양한 도구

오케스트레이션 계층

가상화 솔루션
및 클라우드 서비스



5대 핵심목표

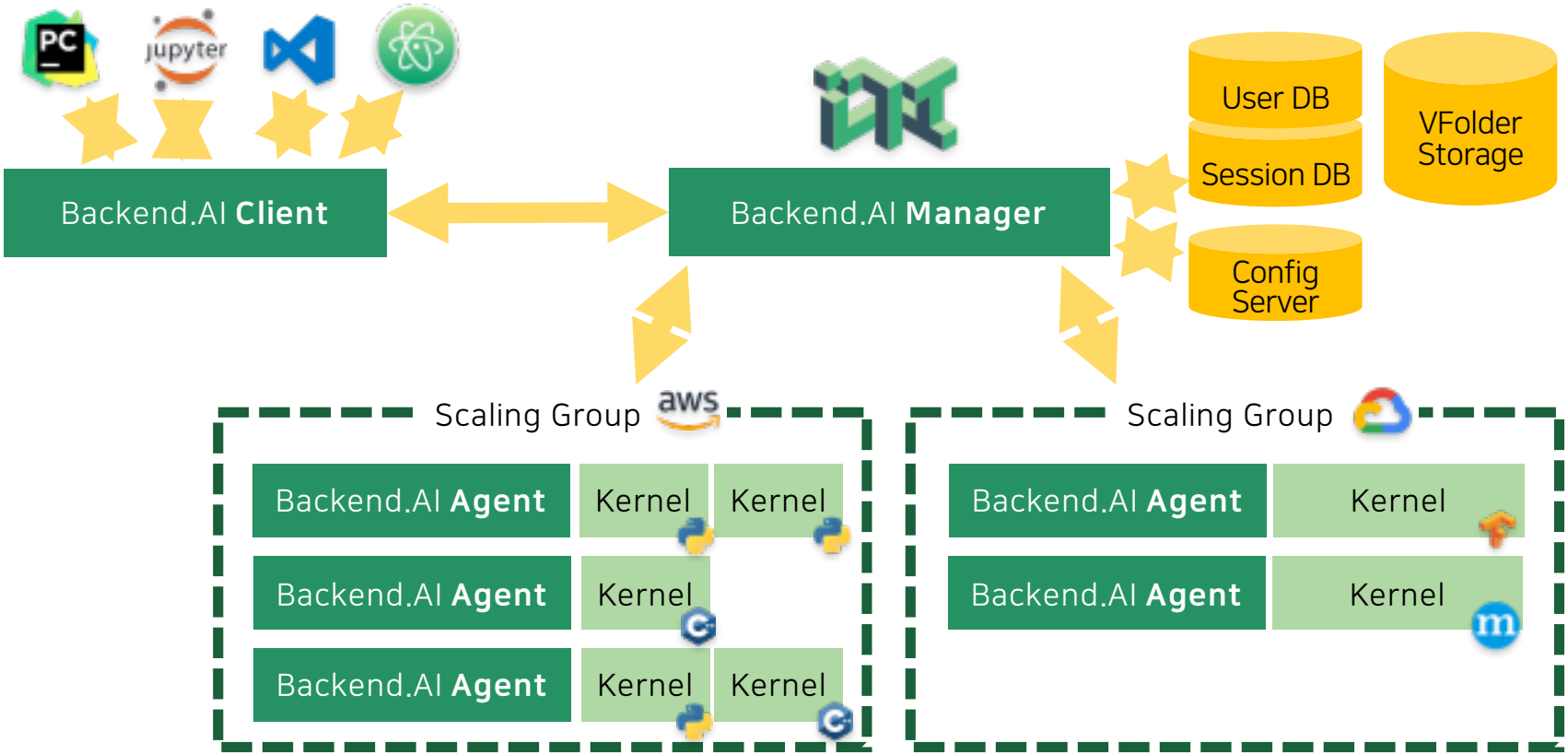
쉽게 **함께**

빠르게 **어디서나**

값싸게

- 보안 격리된 컨테이너 기반 가상화로 서버 수준 고밀도 자원 공유
- 오토스케일링 및 컨테이너 버전 관리
- 컨테이너와 GPU 연결 기능 제공
- 사전 정의된 목적 특화 컨테이너 제공
- 사용자별 자원 사용량 추적
- Jupyter, VSCode, Atom, CLI/IDE 등 다양한 사용자 개발환경 플러그인 지원
- 완전관리형 클라우드 / 설치형 오픈소스

Backend.AI 컴포넌트 구성도



Backend.AI 사용 시나리오

🤔 GPU가 없는 랩탑을 들고다니며 딥러닝 AI를 개발하고 싶을 때

자신의 GPU 워크스테이션을 cloud.backend.ai에 등록 또는 클라우드 요금제 구입 → 갖고다닐 개인 랩탑에 Backend.AI Client 설치 → 😎

🤔 사내 고성능 GPU 서버 1대를 여러 연구원이 공유하면서 쓰고 싶을 때

사내 서버에 Backend.AI 설치 (오픈소스 버전 무료 사용 가능) → 각 연구원 PC에 Backend.AI Client 설치 → 😎

🤔 망분리 적용된 사내 GPU 서버를 조직 단위로 유연하게 공유하고 싶을 때

사내 서버에 Backend.AI 설치 (엔터프라이즈 구입 후 오프라인 설치, SSO 연동, 보안 로깅 추가기능 사용) → 각 연구원 PC에 Backend.AI Client 설치 → 😎

Backend.AI

기계학습 모델을 훈련하고

실행하는 **모든 과정**을

클라우드 또는 **자신의 서버**에서

엄청나게 **쉽고 빠르게** 돌려주는

세련된 플랫폼

- 다양한 기계학습 라이브러리 지원
  - TensorFlow, PyTorch, Caffe 등
- 여러 버전의 라이브러리 동시 지원
  - 예) TensorFlow 1.2, 1.3, 1.4, 1.5를 동일 서버팜에서 동시 운영 지원
- 기계학습 라이브러리 자동 업데이트 지원
- CPU / GPU / RAM 동적 할당
  - 훈련마다 다른 연산 자원 할당 지원
    - ✓ 예) 4CPU + 2GPU + 64GB
  - 멀티 GPU 지원
  - GPU 램 파티셔닝 지원 (TensorFlow)
- GPU 지원
  - Nvidia CUDA 기반 가속 (TensorFlow / PyTorch, Caffe)
  - AMD ROCm 기반 가속 (TensorFlow)*

*베타 테스트 중

Backend.AI

기계학습 모델을 훈련하고

실행하는 **모든 과정**을

클라우드 또는 **자신의 서버**에서

엄청나게 **쉽고 빠르게** 돌려주는

세련된 플랫폼

- 온라인 모델 사용 (1.4*)
  - use.backend.AI 기반의 모델 서빙
  - 함수처럼 딥러닝 모델 사용
  - Backend.AI SDK를 통한 다양한 언어 지원
    - ✓ Python / JAVA BETA / Node.js / JavaScript (on browser) / Microsoft Excel / PHP BETA
- 개발한 모델의 서빙 지원 (1.5†)
  - 직접 개발한 모델의 서빙 지원
  - 기존 모델의 데이터 기반 커스텀 트레이닝 지원

*베타 테스트 중 (2018년 상반기)

† 개발중 (2018년 하반기)

Backend.AI

기계학습 모델을 훈련하고

실행하는 **모든 과정**을

클라우드 또는 **자신의 서버**에서

엄청나게 **쉽고 빠르게** 돌려주는

세련된 플랫폼

- 온프레미스 서버 설치
  - 물리 서버 및 VM 설치 모두 지원
  - OpenStack 설치 지원*
- 다양한 클라우드 지원
  - Amazon, Microsoft, Google 클라우드 지원
- 이기종 클라우드 통합 지원
  - 예) Amazon + Microsoft
  - 예) On-premise + Amazon
  - 클라우드 통합을 위한 편의 지원

*베타 테스트 중

Backend.AI

기계학습 모델을 훈련하고

실행하는 **모든 과정**을

클라우드 또는 **자신의 서버**에서

엄청나게 **쉽고 빠르게** 돌려주는

세련된 플랫폼

- 연구자 및 개발자를 위한 IDE 통합
  - IntelliJ IDEA (PyCharm 포함), Visual Studio Code, ATOM editor, Jupyter Notebook 통합
  - TensorBoard 등의 모니터링 툴 설치 및 접근 지원
- 프로토타이핑 스케일링 시나리오
  - 로컬에서 테스트 후 스케일
  - 모든 워크로드를 서버에서
- 확장을 위한 Backend.AI SDK
  - Python 3 / Node.js / JAVA / PHP SDK 지원

Backend.AI

기계학습 모델을 훈련하고

실행하는 **모든 과정**을

클라우드 또는 **자신의 서버**에서

엄청나게 **쉽고 빠르게** 돌려주는

세련된 플랫폼

- API 기반
  - 완전한 문서 지원 (영어)
  - 온/오프라인 컴포넌트 설치
    - ✓ pip / npm / composer 기반의 설치 지원
  - 오픈소스 생태계
- 완전 비동기 I/O 기반의 코드
  - 짧은 지연시간
  - 각 컴포넌트 동작의 독립성 향상
- 모던 언어 및 컨테이너 환경 기반
  - Python 3.6 + aiodocker
  - Docker 기반의 컨테이너 가상화

Backend.AI 작동 예시

실행하고자 하는 소스 코드

```
import tensorflow as tf
hello = tf.constant('Hello, TensorFlow!')
sess = tf.Session()
print(sess.run(hello))
```

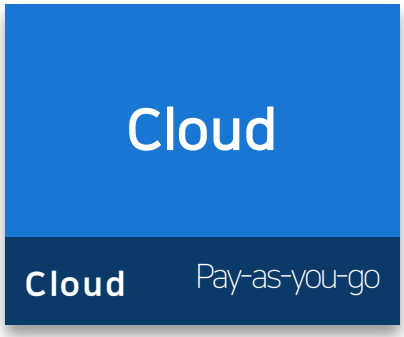
자신의 컴퓨터에는 개발환경이 없음

```
(backend.ai-client-py) ❯ python main.py
Traceback (most recent call last):
  File "main.py", line 1, in <module>
    import tensorflow as tf
ImportError: No module named 'tensorflow'
```

Backend.AI Cloud를 통해 실행하면 OK!

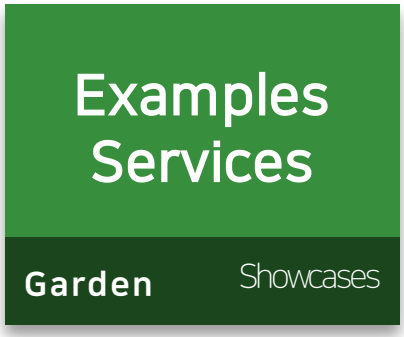
```
(backend.ai-client-py) ❯ backend.ai run python-tensorflow:latest main.py
✔ Session eae5e62d532b3dd94d96bf5099446021 is ready.
Uploading files: 100%|                                | 109/109 [00:00<00:00, 1.33kbytes/s, file=main.py]
✔ Uploading done.
python-kernel: skipping build phase due to missing "setup.py" file
✔ Build finished. (exit code = 0)
b'Hello, TensorFlow!'
✔ Finished. (exit code = 0)
```

Backend.AI 플랫폼



계산 기반의 과학·공학 연구, 딥러닝 모델 훈련 및 엄청나게 간편한 코딩 환경을 구축하는 Scalable PaaS

자신만의 Backend.AI 서버팜을 설치하고 수정하고 개발할 수 있는 오픈소스 버전



Backend.AI 및 Lablup.AI 통합 솔루션 사용자들을 위한 문서, 포럼 및 쇼케이스들

Backend.AI

codeonweb.com

Backend.AI 오픈소스

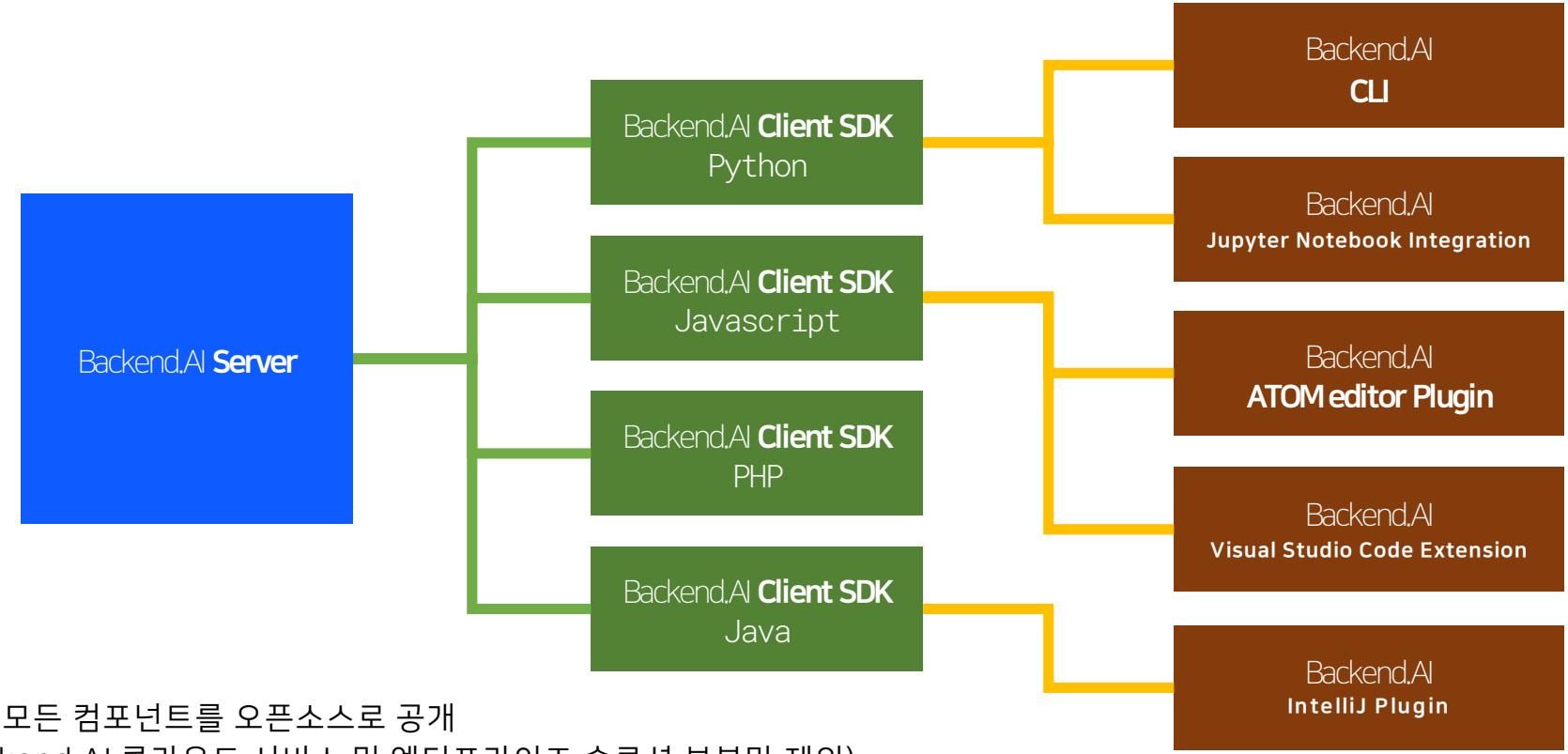
- Backend.AI Ground
  - 오픈소스 (코어 + 라이브러리 + 플러그인)
  - 설치형 소프트웨어 / 프레임워크
  - 듀얼 라이센스 채택
    - ✓ 비상업용: LGPLv3
    - ✓ 상업용: 별도 commercial license 및 사용 계약
- Backend.AI Cloud
  - Backend.AI Ground 기반으로 Lablup에서 직접 운영하는 클라우드
- Backend.AI Enterprise
  - 상업 용도의 별도 계약에 따른 엔터프라이즈 솔루션
  - 온프레미스 / 서포트 플랜

Backend.AI 프로젝트

- https://github.com/lablup/backend.ai
  - 2016년 11월 v0.1 릴리즈
  - 2017년 10월 v1.0 릴리즈
    - ✓ 개발 매뉴얼 제공
  - 2018년 1월 v1.1 / 2월 v1.2 / 3월 v1.3 릴리즈
    - ✓ 코드 안정화
    - ✓ 설치 매뉴얼 및 싱글 클러스터용 자동 설치 지원
    - ✓ 플러그인 아키텍처 도입

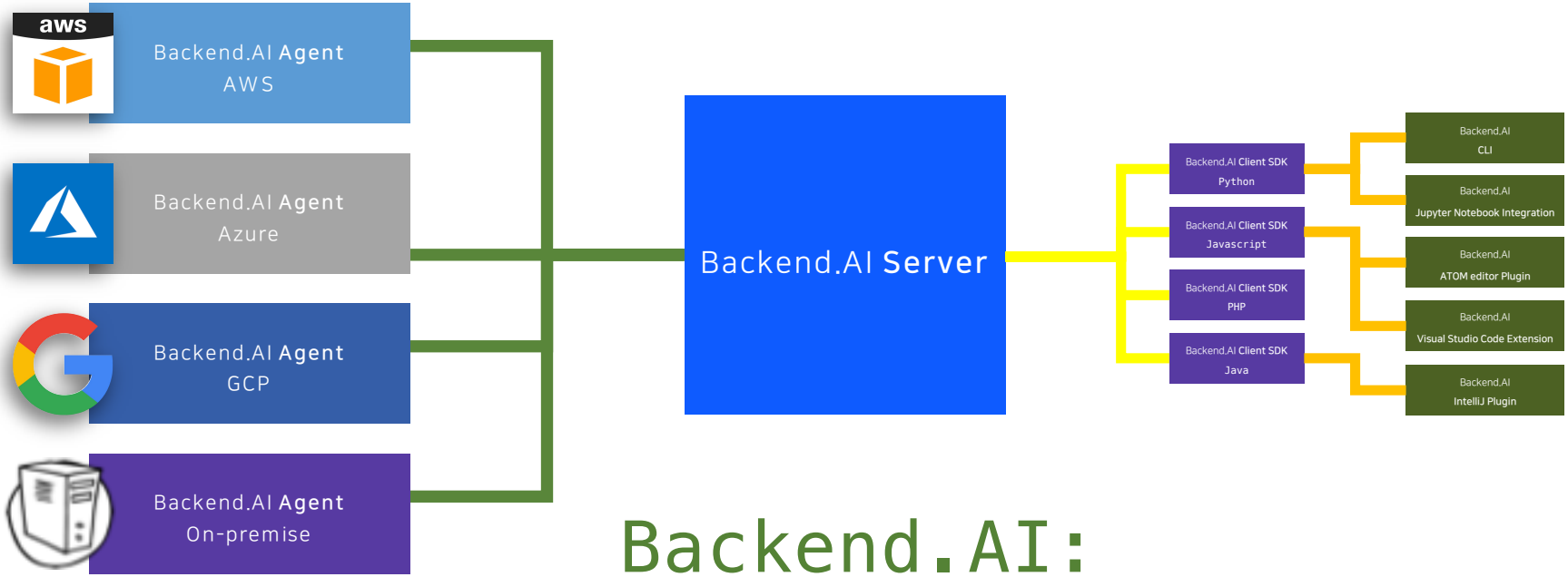
- 향후 로드맵
  - 스케줄러 기능 강화
  - 대규모 HPC 클러스터 오케스트레이션 기능 강화
  - 쉬운 Hybrid cloud 및 on-premise 연동
  - 오토스케일링 기능 강화
    - ✓ 장시간 연산 세션을 위한 scale-in protection
    - ✓ cpu/memory/gpu slot 여유 용량에 따라 cold/hot instance group 운영

Backend.AI 개괄



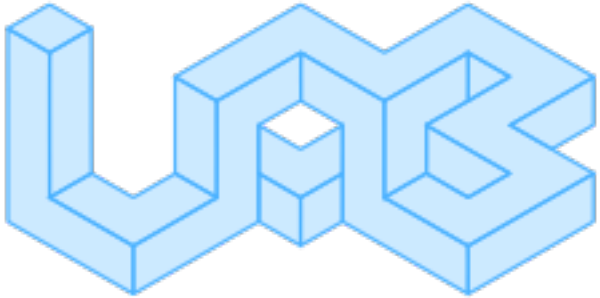
거의 모든 컴포넌트를 오픈소스로 공개
(backend.AI 클라우드 서비스 및 엔터프라이즈 솔루션 부분만 제외)

Backend.AI 개괄



Backend.AI:

The **only All-in-one framework** for
Machine Learning Training PaaS



데모

Backend.AI 사용 데모

Backend.AI 주요 기능

- 빠른 시작
  - 클라우드 가입만으로 바로 사용 가능
  - 사용자의 요청 즉시 가상 프로그래밍 환경 생성
- 다양한 요구사항 충족
  - 모든 주요 프로그래밍 언어와 런타임 지원
    Python, R, Julia, Octave, PHP, Go, C/C++, Java, NodeJS, Lua, Haskell, Rust
  - 동일한 기계학습 라이브러리의 여러 버전 동시 지원

    TensorFlow, Caffe, PyTorch, Keras
- 친숙한 사용자 경험 + 개발자 친화적 프레임워크
  - 기존 연구·개발자들에게 익숙한 환경과의 통합 지원 (코드편집기, 웹기반 연구노트)

    Jupyter Notebook, Visual Studio Code, Atom Editor, IntelliJ beta
  - `$ backend.ai run` 명령줄 및 클라우드 인터프리터·컴파일러 지원
  - 개발자를 위한 HTTP 기반 공용 API 및 언어별 SDK 제공
    Python, Javascript (Node.js), JAVA beta ,PHP beta

지원 프로그래밍 언어

- 현재
  - Python 3, R, PHP 5/7, node.js, JavaScript, Julia, Lua, Octave, Go, C/C++, Rust, Java, Haskell
  - 딥러닝 환경: TensorFlow, **PyTorch**, Keras, Caffe
- 테스트중
  - Swift, TypeScript, C# (.NET Core)

기술적 강점

- Backend.AI Cloud / Open Source
  - API 중심 구현
    - ✓ API 기반으로 확장가능하고 스케일러블하며 재사용가능하고 타 솔루션과 유연하게 결합 가능
  - 다양한 프로그래밍 언어 지원
    - ✓ 다양한 프로그래밍 언어 및 환경을 지원하므로 거의 모든 사용 사례 지원
- 머신러닝용 컨테이너 관련 기술들을 단일 프레임웍으로 제공하는 유일한 솔루션
  - 짧은 지연시간과 고밀도의 컨테이너 풀링
    - ✓ 스케일시 초단위의 컨테이너 스포닝
  - 멀티테넌트 환경에서의 GPU 가속 지원
    - ✓ Faster native GPU performance compared to VM-based solutions
  - 동적 샌드박싱: 프로그래밍 및 재작성 가능한 syscall 필터
    - ✓ Apparmor/seccomp 등 대비 풍부한 프로그래밍 가능한 정책 지원
  - Docker 기반의 레거시 앱 리소스 제한
    - ✓ 예) 현재 Docker의 경우 다양한 머신러닝 컴포넌트들의 CPU 코어 제한을 강제할 수 없음 (OpenBLAS (matrix calculation library) 등).

Backend.AI와 기존 유사 오픈소스 비교

- 기존 오픈소스 구현체들의 한계 해결
  - 특정 프로그래밍 언어 지원 → 범용 언어 지원
  - REPL 구현체들의 취약한 보안 → 보안 중심의 원천적 설계로 실서비스 구축이 가능
  - 기계학습 개발자를 위한 편의 기능 제공 (가상폴더 등)
  - 연산 가속을 위한 보조프로세서 지원 (GPU, TPU, ROCm 등)
  - 연산에 필요한 자원의 자동 스케일링
  - 온디맨드 대규모 코드 연산 기능 제공

Backend.AI 의 특징 비교 (1/2)

- vs. Anaconda Cluster
  - Anaconda Cluster는 고정된 수의 서버를 가진 클러스터에 ssh 기반으로 각종 데이터 분석용 Python/Java 프로그램들을 자동 설치하고 마스터 서버를 통해 통합 실행하는 기능 제공
  - Backend.AI는 클라우드 기반의 서버 자동 스케일링과 컨테이너 기반의 보안 격리된 실행 환경을 제공하여 여러 사용자가 클러스터뿐만 아니라 서버 수준에서도 고밀도 자원 공유 가능

- vs. Google Colaboratory / AWS SageMaker / Azure MLStudio / 각종 MLaaS
  - 클라우드 플랫폼 기반의 관리형 기계학습 실습 환경 제공
  - Backend.AI는 자신의 서버나 벤더 상관 없이 자가 소유한 클라우드 인프라에 직접 설치 가능함

Backend.AI 의 특징 비교 (2/2)

- vs. Kubernetes (Google Borg) / Apache Mesos / Apache Aurora
  - Mesos와 k8s 모두 클러스터 자원 할당 및 관리를 위해 일반화된 오픈소스 솔루션
  - Aurora는 Mesos를 기반으로 service 및 job pipeline 관리 추가 제공
  - Backend.AI는 머신러닝 개발자들을 위해 특화되어 가상폴더, AI 프레임워크 버전 관리를 지원하고 오토스케일링 등 클라우드 환경에 적합한 기능을 추가 제공함

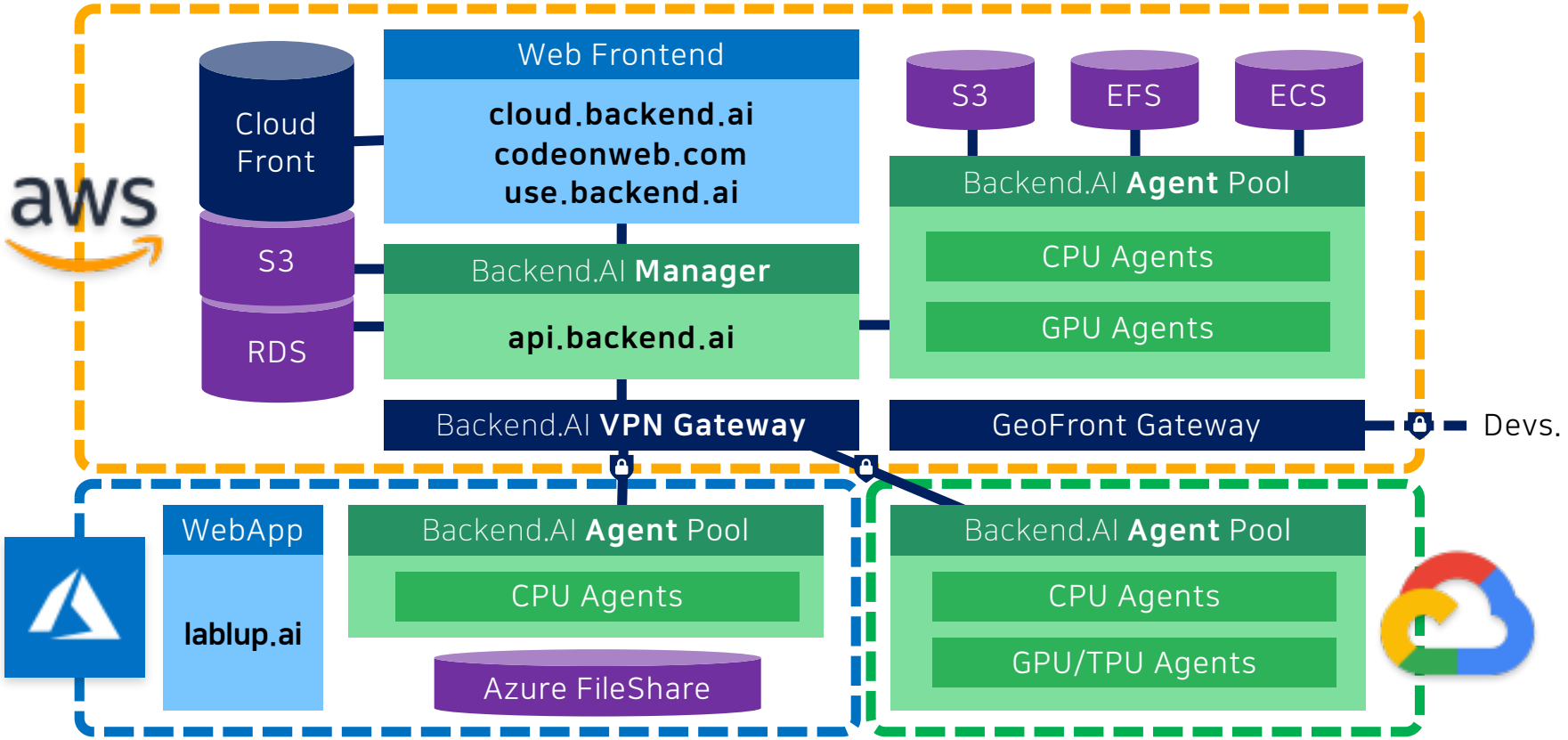
- vs. Ansible / Chef / Puppet / Terraform / ...
  - 대규모 서버 및 컨테이너를 프로비저닝하고 인프라 구성을 코드화하여 관리할 수 있는 DevOps용 도구
  - Backend.AI는 이런 기술들을 종합적으로 활용하여, 기계학습 개발자들이 인프라에 대한 고민을 하지 않도록 한 단계 더 감싸주는 역할을 함

Backend.AI 기술 현황 / 로드맵

기술 특징		nvidia-docker	Docker Swarm	OpenStack	Kubernetes	Apache Mesos	Backend.AI (현재)	Backend.AI (목표)
GPU Support	GPU Assignment & Isolation	✔			✔	✔	✔	✔
	Heterogeneous Accelerators							✔
	Partial GPU Sharing						✔ *	✔
Security	Sandboxing via Hypervisor/Container	✔	✔	✔	✔	✔	✔	✔
	Programmable Sandboxing						✔	✔
Virtualization	VM (Hypervisor)			✔	✔ **	✔ **	✔ **	✔ **
	Docker Container	✔	✔	✔	✔	✔	✔	✔
Scheduling	Availability-slot based		✔	✔	✔	✔	✔	✔
	Advanced				✔ ***	✔		✔
Integration	Modern AI Farmeworks							✔

* TensorFlow 프레임워크에 한하여 기술 테스트 완료
** VM 관리를 직접 수행하지 않고 클라우드 벤더 API 또는 OpenStack에 의존
*** slot 기반이란 여지란 label 기능으로 정해진 자동화된 기능

Backend.AI 다중 클라우드 구성 사례



Backend.AI 응용 예시

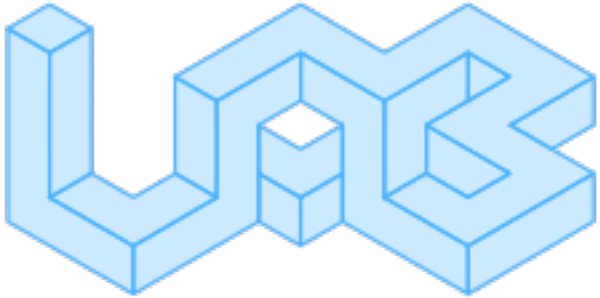
- GPU 클라우드 서비스의 손쉬운 구축
  - 국산 공개SW 기반 기술인 Backend.AI를 활용한 GPU 클라우드 사업화 가능
- 의료기관 및 금융기관 등 고도보안 망분리 환경에서의 사설 GPU 클라우드 구축
  - (엔터프라이즈 에디션) 오프라인 설치, SSO, 보안로깅 등 추가 지원
- 직접 보유한 GPU 서버팜의 용량이 부족할 때 cloud.backend.ai로 동적 용량 확장
  - 또는 반대로 cloud.backend.ai에 자신의 GPU 워크스테이션을 등록하여 원격 관리
- 고성능 컴퓨팅(HPC) 및 과학시각화(HPCV) 지원
  - 클러스터의 서비스와 애플리케이션 배포 단순화로 최신 버전 유지 및 성능 향상

감사합니다!

질문은 contact@lablup.com 으로!

덧) 옆에서 열리는 **대한민국 금융대전**에서
래블업 부스를 운영 중입니다!

Lablup Inc.	https://www.lablup.ai
Backend.AI	https://www.backend.ai
Backend.AI GitHub	https://github.com/lablup/backend.ai
Backend.AI Cloud	https://cloud.backend.ai
CodeOnWeb	https://www.codeonweb.com



Appendix

Backend.AI

Q&A

- Backend.AI 자체의 성능 오버헤드
  - agent: Python asyncio 기반의 싱글쓰레드 데몬으로, CPU 1코어 + RAM/Disk 1 GiB 정도면 충분
  - manager: 2코어 이상 독립 서버 또는 가상 머신 할당 권장 (PostgreSQL, etcd, Redis 구동)
- Backend.AI 최소 실행 환경
  - OS: Ubuntu 16.04+ / CentOS 7.2+
  - SW: Docker 17.03+ (18.03 권장), Python 3.6
  - nvidia-docker는 옵션
- 설치 환경별 스케일링 정책
  - 클라우드 : 오토스케일링
  - 온-프레미스 : 스케줄링 (자원활용률 극대화)
- NSML과 같은 "연산 라이브러리" 제공 여부
  - Backend.AI 자체는 오케스트레이션 역할에 집중
  - 사용자가 직접 또는 래블업에 요청하여 특정 연산 라이브러리가 포함된 Docker 이미지 빌드 및 사용 가능

다중 컨테이너 단일 GPU 공유 (상세)

- 개발 내용
  - TensorFlow 버그 패치 (v1.0 이후 모든 버전 해당)
    - ✓ GPU Memory 제한 옵션(`per_process_gpu_memory_fraction`) 제공
    - ✓ 런타임에만 설정 가능하고, 그나마 Keras 모듈을 먼저 불러오면 버그로 인해 해당 옵션을 무시함
    - ✓ 전역설정은 개발 철학으로 인해 외부 기여 반려됨 (TensorFlow Issue #8040, #8136, #14585)
  - Backend.AI에서는 TensorFlow 코어를 직접 패치해야만 메모리 제한이 가능함을 확인한 상태
    - ✓ 본 과제를 통해 해당 패치를 Backend.AI와 연동할 수 있도록 개선 및 적용
    - ✓ TensorFlow뿐만 아니라, 다른 GPU 기반 코드에서도 강제할 수 있는 일반화된 방법 조사 및 개발
    - ✓ 현재 nvidia-docker도 GPU 메모리 제한 지원하지 않으나(nvidia-docker Issue #408), NVIDIA와의 기술 협력을 통해 가능성 모색 예정