오픈테크넷 서밋 2021

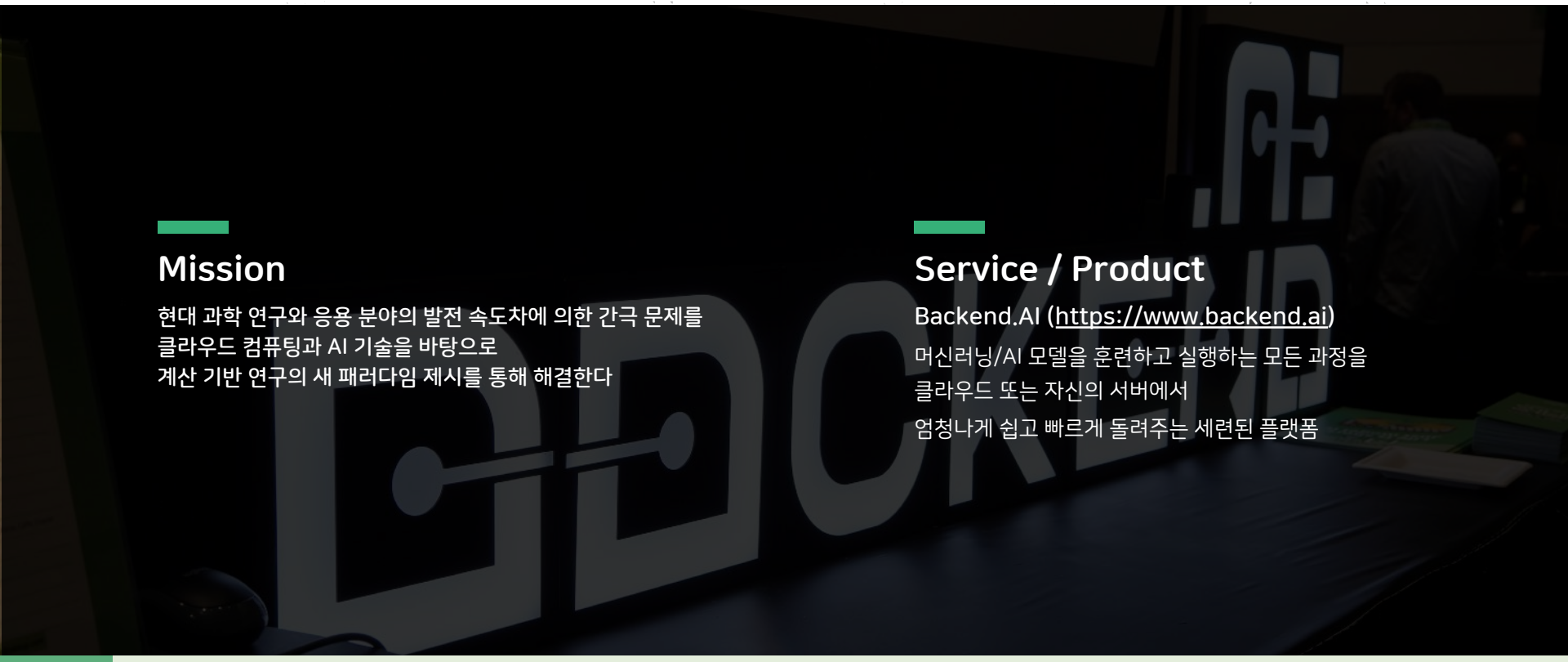
프라이빗 클라우드 및 air-gapped 클러스터 환경에서의 오픈소스 패키지 배포 및 활용

신정규

래블업 주식회사 / Google Developers Expert (ML/DL)

Lablup Inc.

Make AI Accessible



Mission

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

Service / Product

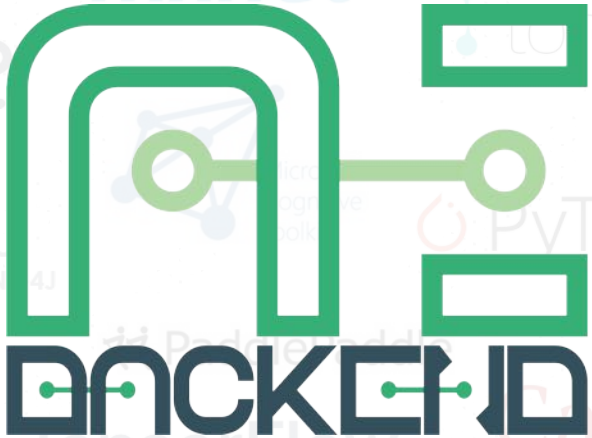
Backend.AI (https://www.backend.ai)

머신러닝/AI 모델을 훈련하고 실행하는 모든 과정을
클라우드 또는 자신의 서버에서
엄청나게 쉽고 빠르게 돌려주는 세련된 플랫폼

Backend.AI

https://www.backend.ai

AI 프레임워크용 엔터프라이즈 클러스터 백엔드

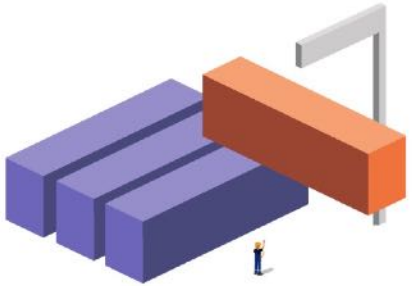


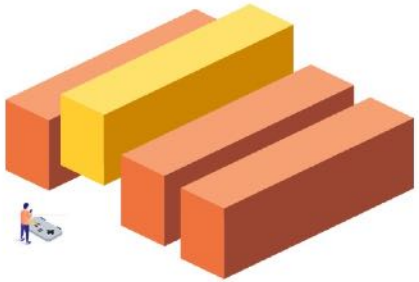
차별점 :

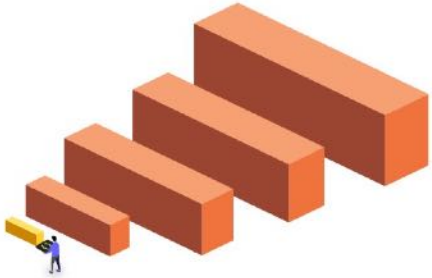
- 분할 GPU 자원 할당 및 공유
- 쉬운 GUI 와 강력한 CLI / SDK 제공
- 훌륭한 관리 편의성과 높은 사용률

Motivation

You make AI ✨



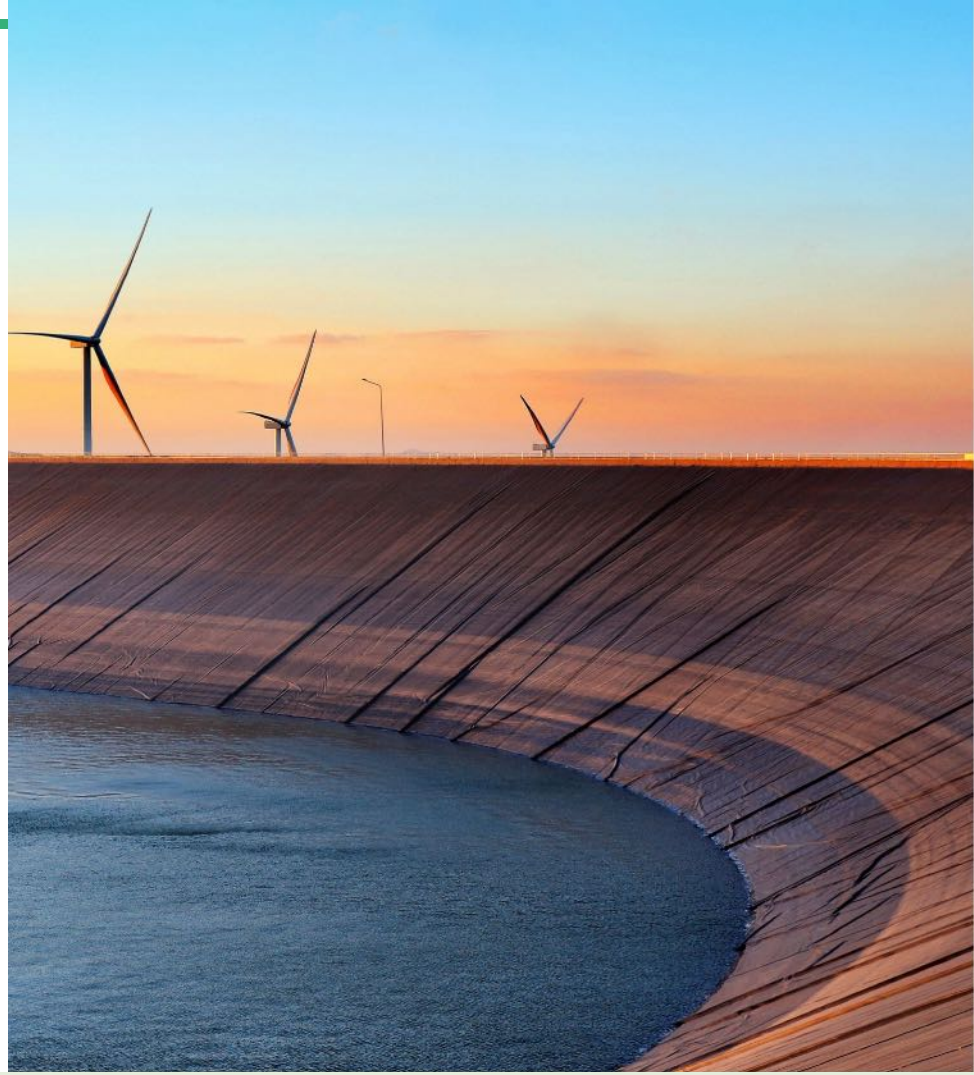




We do 👷🛠️

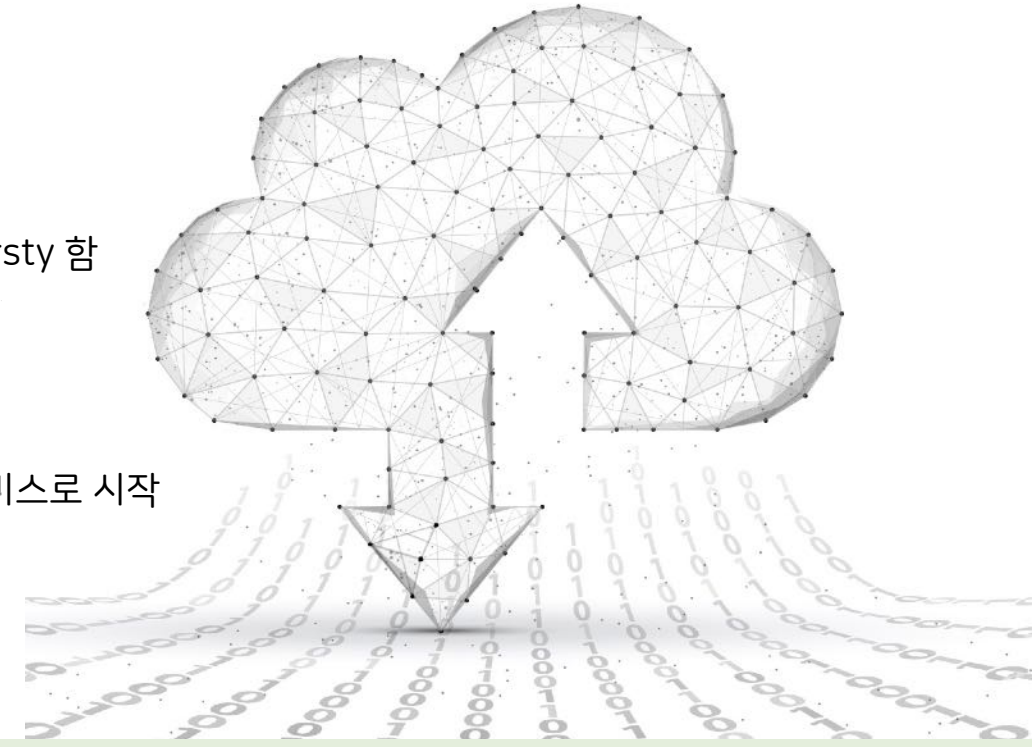
3 이야깃거리

- 프라이빗 클라우드의 시대
  - 퍼블릭 클라우드의 그늘
- 프라이빗 클라우드의 난제
  - 오픈소스의 홍수
  - 편의성과 보안의 교환
- 프라이빗 클라우드 + 컨테이너 + AI
  - 컨테이너+AI: 장점과 단점
  - 접근 방법 바꾸기
- Backend.AI Reservoir
  - 요약 및 상세 구조
  - 사례
  - 오픈소스화
- 마무리

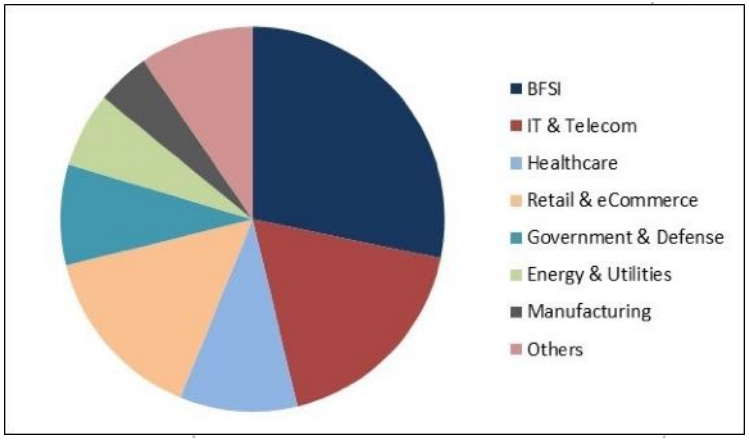


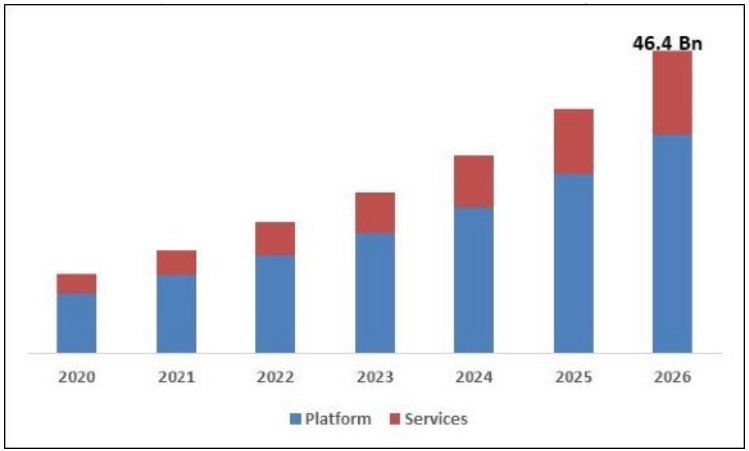
4 클라우드의 시대

- **개념의 변천**
  - 메인 프레임(1980s) - Network Computer (1990s) – 데이터 서버 (2000s)– 클라우드 (2010s)
  - 필요한 만큼의 양을 필요할 때 사용
  - 스토리지에서 연산 자원까지
- **시작**
  - 대규모 연산 자원 이용의 특징: 이용 패턴이 bursty 함
  - 그렇다면 나머지 유휴 시간에는?
- **아마존 웹 서비스**
  - 자체적인 서비스 백엔드를 외부에 공개하는 서비스로 시작
  - 현재는 아마존의 주요 서비스로 성장



클라우드 컴퓨팅과 산업





클라우드 최상위 산업분야

- BFSI (Banking, financial services and insurance)
- IT & Telecom
- 헬스케어
- 운송
- AI 도시 & 스마트 팩토리

성장 중인 개발자 생태계

클라우드 플랫폼 시장

- 26% CAGR, 2026년 USD 46.4 B 로 성장 예측
- 데스크탑 및 서버 기반 시장
  - 25.8%의 CAGR로 성장 예측 (2020-2026)
- 모바일 기반 시장
  - 26%의 CAGR로 성장(2020-2026)
- GPU 클라우드
  - 2,000곳 이상의 기업이 GPU 클라우드 사용

6 클라우드 시장 예측

- 급격한 속도로 인프라 구조의 교체 진행중
- 2021년까지의 예측

	2017	2018	2019	2020	2021
Cloud Business Process Services (BPaaS)	42.6	46.4	**50.1**	54.1	58.4
Cloud Application Infrastructure Services (PaaS)	11.9	15.0	**18.6**	22.7	27.3
Cloud Application Services (SaaS)	60.2	73.6	**87.2**	101.9	117.1
Cloud Management and Security Services	8.7	10.5	**12.3**	14.1	16.1
Cloud System Infrastructure Services (IaaS)	30.0	40.8	**52.9**	67.4	83.5
Total Market	153.5	186.4	**221.1**	260.2	302.5

Worldwide Public Cloud Service Revenue Forecast (Billions of U.S. Dollars)

http://www.gartner.com/newsroom/id/3871416

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

프라이빗 클라우드의 대두

- 퍼블릭 클라우드의 그늘
  - 비용
    - ✓ 항상 감소하지는 않음. (일반적으로는 증가함)
    - ✓ 예: GPU 클라우드: ML/AI, GPU 기반 빅데이터 처리의 경우
  - 퍼블릭 클라우드 위의 프라이빗 클라우드 구현
    - ✓ 동일 스택 기반의 마이그레이션: 관리 소요가 크게 줄어들지 않음
  - 보안
    - ✓ 데이터를 퍼블릭 클라우드로 이전할 수 없는 경우
    - ✓ 데이터를 퍼블릭 클라우드에 내 줄 수 없는 경우
- 프라이빗 클라우드
  - 자체적으로 "클라우드같은" 온프레미스 클러스터 운영
  - 온프레미스와의 차잇점: "Scalability"
  - 퍼블릭 클라우드들의 프라이빗 클라우드 서비스
    - ✓ AWS OutPost, Google Cloud Anthos, Azure Stack

프라이빗 클라우드의 난제

- **오픈소스의 홍수**
  - 빠른 변화 (가상화 기반 -> 컨테이너 기반 MSA)
  - 증가하는 상호 작용
  - 패키지 저장소: 온라인을 통한 배포
- **편의성과 보안의 교환**
  - 개인정보 같은 민감 정보 관리 강화 (법률, 규정, 시장 요구 등)
  - 더욱 복잡해진 네트워크 & 스토리지 요소
  - 보안을 강화할 수록 “인터넷이 안돼요”
- **복잡해진 시장 요구와 아키텍처**
  - 데이터 중심의 IT 서비스 변화로 다양한 클라우드 요구사항 대두
  - 빅데이터 분석과 같은 대규모 DB에 대한 분석
  - 인공지능 맞춤형 서비스

갭 메우기

VM

공용 VM 이미지 빌드 및 제공
프라이빗 저장소 운영과 연계
패키지 설치 지원

KVM
Xen
Oz
DiskImage-Builder

Container

필요한 패키지들을 담은
컨테이너 이미지 빌드 및 배포
컨테이너 빌더 솔루션

Docker
Buildah / Podman
Buildkit
Img
Bazel
Makisu

Repository

오픈소스 및 자체 패키지
설치 및 업데이트 지원을 위한
내부 서빙용 저장소 서비스

Docker Enterprise
Artifactory
ProGet
Cloudsmith Package
Helm
Apache Archiva

컨테이너 기반의 클라우드 런타임 / 장점

- **장점**
  - 재현 가능한 환경을 구축하기에 용이함
    - ✓ 운영체제 커널만 공유하고 사용자레벨의 모든 라이브러리와 패키지를 별도로 패키징 및 실행
    - ✓ VM에 비해 가벼운 이미지 용량 및 레이어 기반의 파일시스템으로 빠른 배포
  - 빠른 성능
    - ✓ 가상 하드웨어를 프로비저닝하지 않고 운영체제 커널 내부의 namespace만 구분하므로 cold start 지연시간 적음
    - ✓ 가상 하드웨어 계층이 없으므로 연산 실행할 때 시스템 하드웨어를 native 성능 그대로 활용 (HPC/AI 워크로드에서 특히 중요)

컨테이너 기반의 클라우드 런타임 / 단점

- 단점
  - 사용자의 개념 이해 및 적응 필요 (높은 재현 가능성이라는 장점과 대응하는 불편 사항)
    - ✓ 컨테이너 내부의 변경사항은 기본적으로 volatile 특성을 가짐 (종료 후 사라짐)
    - ✓ 데이터 보존을 위해서는 스토리지 볼륨에 대한 사용자의 이해 필요
    - ✓ VM은 원하는 환경 구성 후 그 자체를 통째 이미지로 뜨지만, 컨테이너는 별도의 recipe 파일을 작성하여 이미지를 만들고 관리
  - 호스트의 연산 자원 노출
    - ✓ /proc/cpuinfo, /proc/meminfo 및 /sys/cgroup 등의 가상화가 되지 않음
    - ✓ 경우에 따라 연산라이브러리의 과도한 멀티쓰레딩 이슈 발생
    - ✓ lxcfs 등 이를 보완해주는 패키지들이 존재하지만 호환성 문제가 있음
  - 드라이버 호환 이슈
    - ✓ RDMA, GPU 등 특정 가속 하드웨어들을 활용하기 위해서는 드라이버 및 런타임 차원의 지원이 필요함

프라이빗 클라우드 + 컨테이너 + AI

- **모델 훈련 파이프라인: 고정시킬 수 없는 컨테이너 환경**
  - 너무 높은 소프트웨어 구성 복잡도
  - 딥 러닝 프레임워크: TensorFlow 1.X, 2.X, PyTorch 1.0~1.9, JAX, Haiku…
    - ✓ 사용하는 라이브러리 구성이 천차만별임
    - ✓ 충돌하지 않는 라이브러리 구성이 숙제
  - 개발환경 통일 불가 → 공통 컨테이너 이미지만으로 다양한 개발자·연구자 요구사항 충족 어려움
    - ✓ 결국 직접 그때그때 필요한 패키지를 직접 설치해볼 수 있어야 함

- **모델 배포 파이프라인: 컨테이너 용량과 자유도 사이의 균형**
  - 고전적 시도: 모델마다 컨테이너 이미지 하나씩 빌드 및 배포
  - 컨테이너 이미지 용량 문제
    - ✓ 한 노드에 동시 배치하는 이미지의 가짓수·다양성 제한 발생
    - ✓ Auto-scaling 과정에서 cold start 지연시간 증가

해결책: Backend.AI 의 사례

- **공통 이미지 + 상황에 따른 라이브러리 / 프레임워크 레이어의 결합**
  - 컨테이너 실행 후 상황에 따라 자유로운 패키지 설치 허용 필요
- **오픈소스 저장소 기반 패키지 설치 자유도 확보**
  - 사용자 홈디렉토리에 탑재한 스토리지 볼륨에 오픈소스 패키지 설치
    (예: ~/.local, ~/.config)
  - 사용자별로 이미지와 별개인 자신만의 custom 패키지 집합을 보유 및 관리할 수 있음
- **문젯점**
  - Air-gapped 환경에서 오픈소스 패키지를 어떻게 설치할 것인가?
- **대안**
  - 오픈소스 솔루션 + 엔터프라이즈 서비스를 하나 만들자!

Backend.AI Reservoir

처음부터 이렇게 커질 줄 알았던 프로젝트는 아니었습니다;

Backend.AI Reservoir : 요약

- **Backend.AI 용의 로컬 패키지 저장소 서비스**[BETA]
  - Lablup 저장소 허브 서비스와 로컬 패키지 서버로 구성
  - Backend.AI Enterprise 의 추가 컴포넌트 형태로 제공

- **프라이빗 클라우드, 폐쇄 환경, 오프라인 사이트에 최적 솔루션**
  - 보안 체크를 거친 패키지만 제공
  - 오픈소스 라이선스 확인 파이프라인

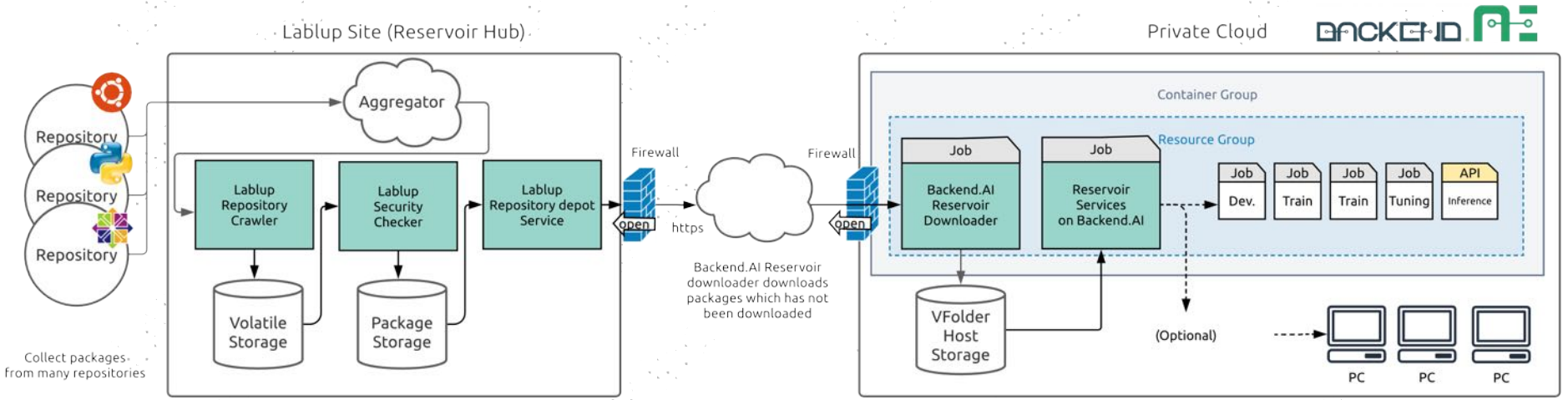
- **지원 패키지 저장소 (Backend.AI 20.09)**
  - 언어: PyPI, CRAN (R)[BETA]
  - 운영체제
    - ✓ Ubuntu (18.04 / 20.04)
    - ✓ DGX OS (4 / 5)
    - ✓ CentOS (7 / 8) 패키지

- **제공**
  - Backend.AI 클러스터 내부 / 외부 (옵션) 를 대상으로 하는 독립 패키지 저장소 서비스

Backend.AI Reservoir : 기능

- **프로그래밍 언어 / 운영체제를 대상으로 한 패키지 저장소**
- **Backend.AI 서비스 세션 형태 제공**
  - 패키지 저장소를 위한 자원 관리
  - 재시작 및 연산자원 추가 할당을 통한 쉬운 스케일업/다운
- **Backend.AI 및 다른 서비스/플랫폼과의 통합**
  - Backend.AI 클러스터와의 즉각적인 통합
  - Backend.AI 클러스터 외부 대상의 패키지 저장소 서비스 제공
- **최신 상태 유지**
  - Lablup 저장소 허브와의 주기적 동기화를 통환 패키지 최신화 (필요시)

Backend.AI Reservoir : 전체 설계



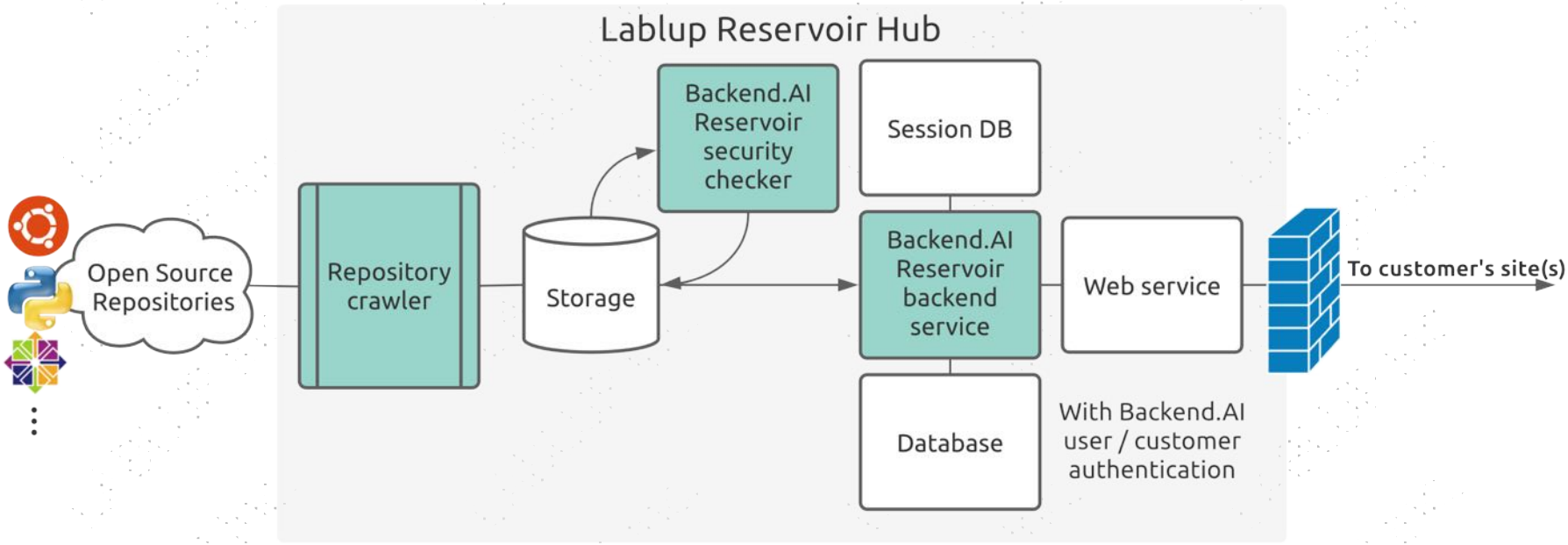
Reservoir Hub firewall ingress port open

source IP : Customer site
source port : Any
target IP : Repository depot service
target port : Reservoir service port

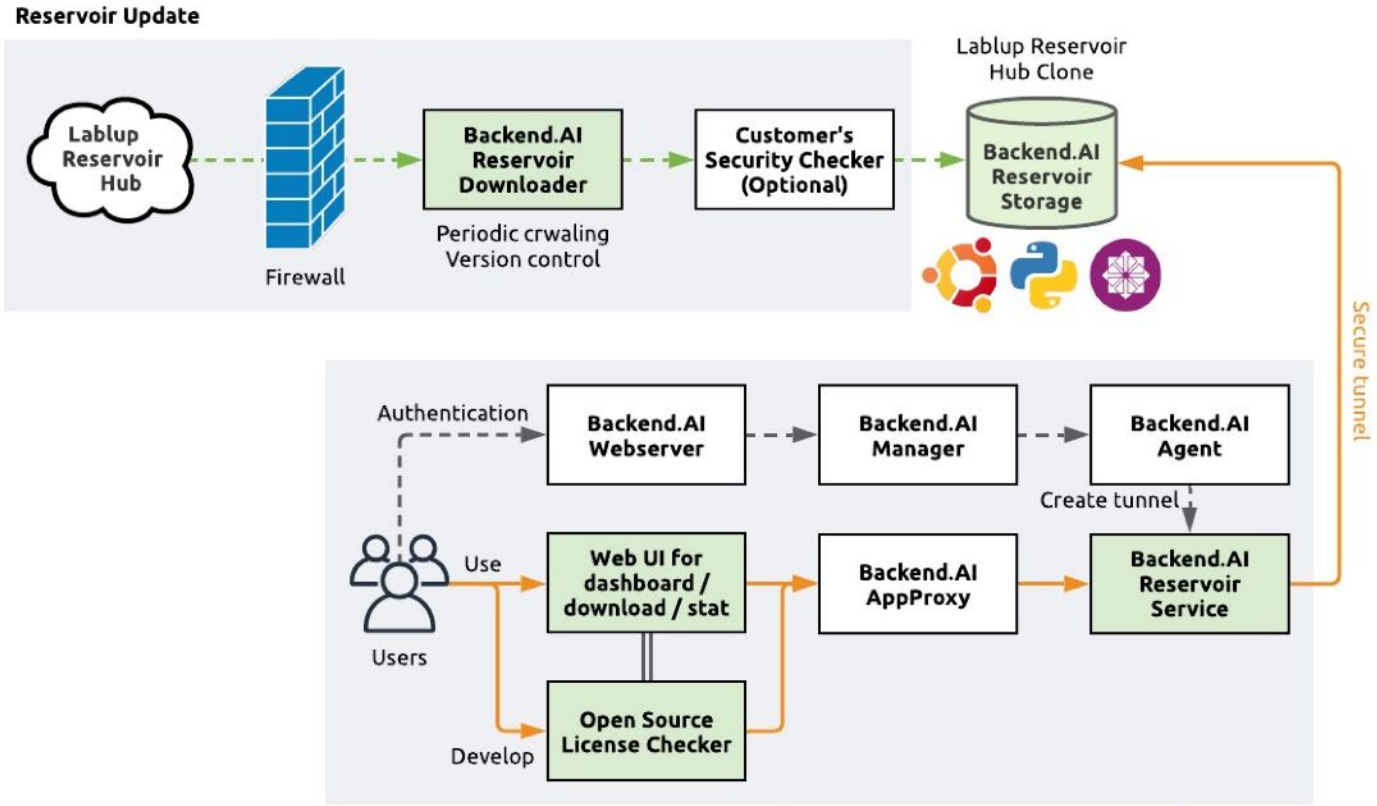
Private cloud firewall egress port open

source IP : Reservoir downloader IP
source port : Any
target IP : Repository depot service
target port : Reservoir service port

Backend.AI Reservoir : 상세 구조 (패키지 허브 서비스)



Backend.AI Reservoir : 상세 구조 (프라이빗 클라우드)



Using the Backend.AI Reservoir with Authentication

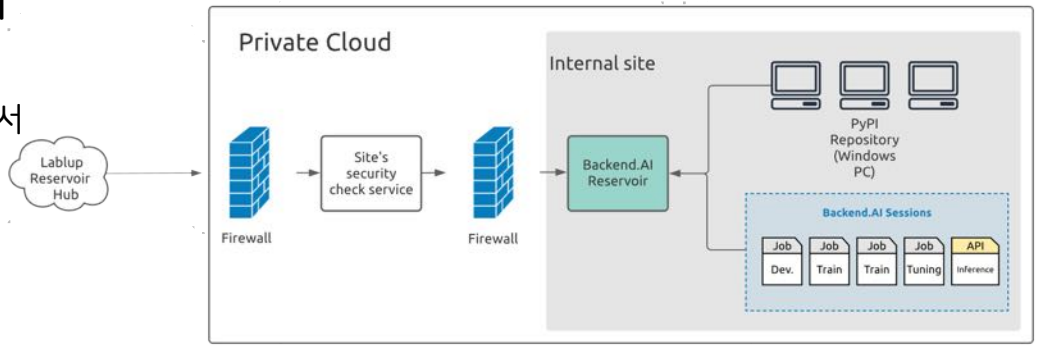
Backend.AI Reservoir : 시나리오

- **올인원 오프라인 사이트 구축**
  - 고객 사이트의 PC 및 서버는 외부 네트워크가 아닌 내부 네트워크에 있는 패키지 저장소에 액세스 할 수 있음
    - ✓ Python, Ubuntu 및 CentOS
  - 오픈소스 패키지 다운로드를 위해 방화벽 예외 추가가 필요 없어짐
    - ✓ Lablup 사이트의 Repository Hub가 인터넷에서 다양한 패키지 세트를 수집 및 보안 확인
    - ✓ 프라이빗 클라우드 사이트의 경우 저장소 동기화를 위해서만 Lablup 사이트에 억세스함
  - 오픈소스 라이선스 체크 과정의 일원화
    - ✓ 패키지 반입 과정에서의 사전 체크 파이프라인
    - ✓ 내부에서 개발한 소프트웨어의 오픈소스 라이선스 체크 지원

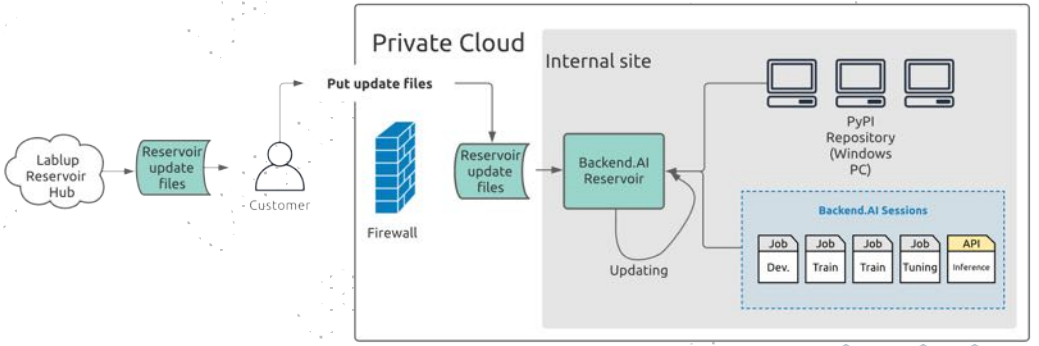
- **패키지 저장소 캐시 서비스**
  - 패키지 요청 및 다운로드시의 트래픽 감소
  - 외부 연결이 불가능하거나 대역폭이 제한된 경우에도 안정적으로 사용할 수 있는 패키지 저장소 서비스 제공

Backend.AI Reservoir : 사례

- 사용자의 요구에 따른 구축 / 업데이트 패턴 사례
- 자동 업데이트 파이프라인 (국내)
  - 자체 보유한 보안 기술을 사용하여 고객 사이트에서 추가 보안 검토가 이루어 져야 Site내로 반입 가능하도록 구축
  - 고객 사 내 Backend.AI session 및 Windows 개발 PC에서 Reservoir 저장소 사용



- 완전 차단망 업데이트 (해외)
  - 외부망과의 직접 연결 지양: 자동 업데이트 불가
  - Update file 공유 후 내부 반입은 고객사 인력에 의해 진행
  - 지정한 특정 위치에 해당 업데이트 파일들을 두면 Reservoir가 업데이트를 수행
  - 고객 사 내 Backend.AI session 및 Windows 개발 PC에서 Reservoir 저장소 사용



Backend.AI Reservoir : 오픈소스 정책

- **Backend.AI Reservoir 컨테이너: 오픈소스**
  - 쉽게 저장소를 운영할 수 있도록 갖춰진 패키지 서버
  - 오픈소스 패키지 서버 기반 이미지
    - ✓ 패키지 서버 솔루션 라이선스에 따라 각각 MIT, Apache License 또는 GPLv3로 배포
  - Backend.AI 가 없어도 간단한 실행 및 운영 가능
    - ✓ 이 경우 통계, 트래픽 제한, 자동 동기화 등의 특화 기능은 Backend.AI 가 필요하여 사용할 수 없음.
  - Docker Hub를 통해 제공 예정

- **Lablup Reservoir 패키지 허브 서비스**
  - 보안 유효성 검증 및 미통과 패키지 제외
  - 점진적 업데이트 지원 (월 단위)
  - 엔터프라이즈용 유료 서비스 형태로 제공
    - ✓ 업데이트 트래픽 용량: 월 평균 2TB 이상
    - ✓ 사이트 특성에 따라 오프라인 전달 / 업데이트 서비스

Backend.AI Reservoir : 로드맵

Alpha *Beta* *Production* *Expansion*

2020.03
프로젝트 시작
- 내부 개발 및 테스트
- 저장소 캐시 서비스

2020.6
v0.2 릴리즈
- 코드 안정화
- 수동 설치 지원
- Diff 알고리즘 기반 패키지 동기화 지원

2021.1
v0.4 릴리즈
- 자동 패키지 수집 Watcher
- DGX OS 저장소 지원
- OS 버전별 패키지 저장소 지원

2022.3
v22.03 릴리즈
- Backend.AI Control-Panel을 통한 통합 대시보드 지원 (beta)
- 패키지 서버 자원 오토스케일
- Backend.AI Storage-Proxy 서비스 통합
- FOSSLight 통합 테스트

2020.07
v0.1
- 초기 프로토타입
- PyPI 저장소 클론

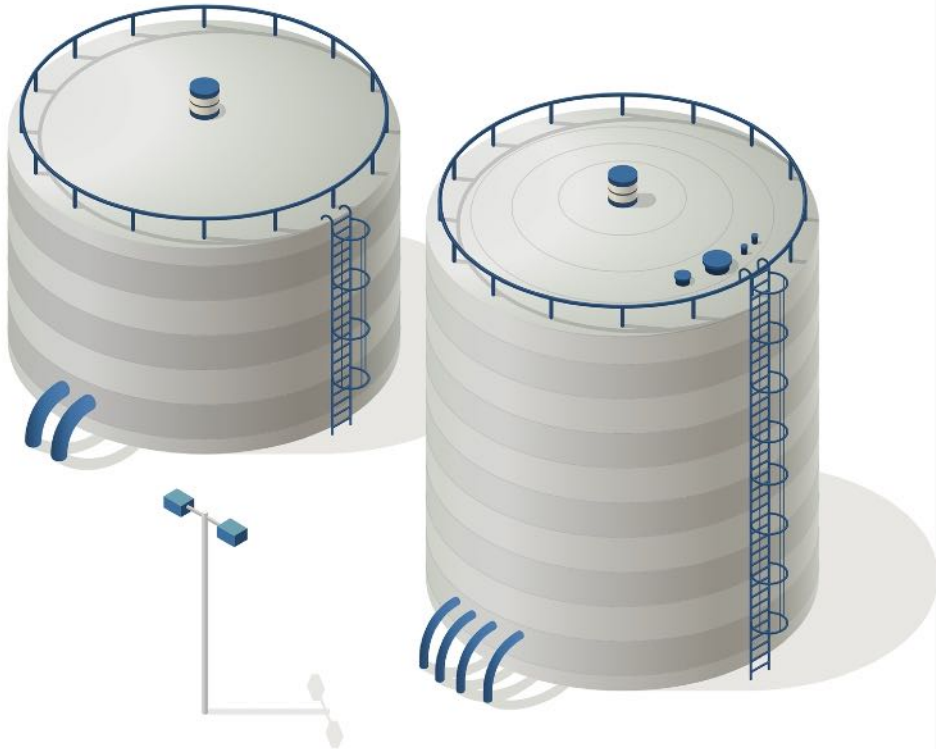
2020.10
v0.3 릴리즈
- Ubuntu 저장소 지원
- CentOS 저장소 지원
- 프로젝트 베타 런칭

2021.05
v21.05 / Enterprise R2
- Downloader / Transferer / Restorer (w/ watcher) 전체 파이프라인 지원
- Downloader에서 향상된 diff 알고리즘

2021.11
v21.11 / Enterprise R2
- **공식 공개**
- 클라이언트 측 패키지 버저닝 관리 지원
- 향상된 PyPI 인덱스 캐싱

마무리

- 프라이빗 클라우드의 시대
  - 퍼블릭 클라우드의 그늘
- 프라이빗 클라우드의 난제
  - 오픈소스의 홍수
  - 편의성과 보안의 교환
- 프라이빗 클라우드 + 컨테이너 + AI
  - 컨테이너+AI: 장점과 단점
  - 접근 방법 바꾸기
- Backend.AI Reservoir
  - 요약 및 상세 구조
  - 사례
  - 오픈소스화
- 마무리



Backend.AI : 장점

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 관리 컨트롤 패널을 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 / HPC 를

R&D 부터
Business Service,
AI Service 추론 및 제공까지

하나의 일관된 플랫폼을 통해

효과적으로 관리

[1] Nvidia CUDA, Nvidia Jetson, AMD ROCm, Google TPU, Google Coral