

# 웹에서 인공지능 언어모델 BERT 경험하기

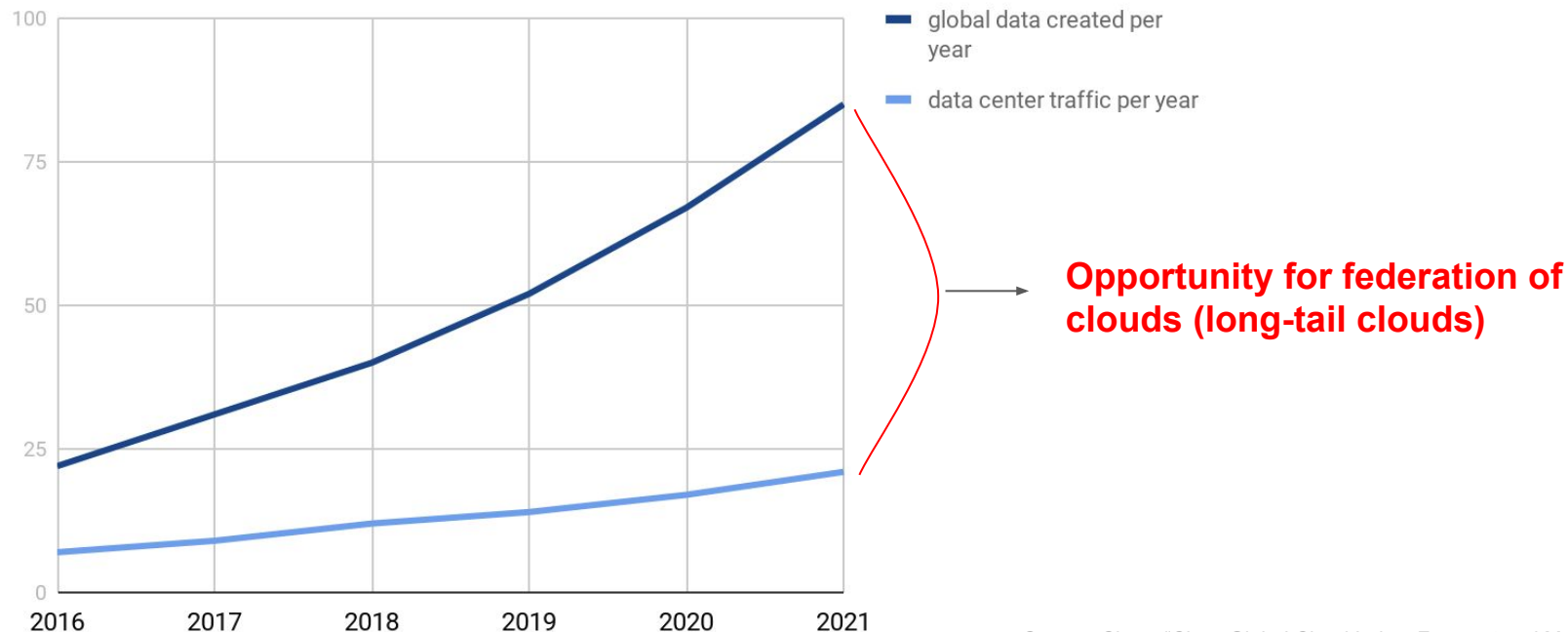
<http://cloud.ainetwork.ai>

김민현 (kimminhyun@comcom.ai)

# Public cloud is not catching up global data needs

→ AI, IoT requires more powerful and diverse clouds.

global data created per year vs. data center traffic per year

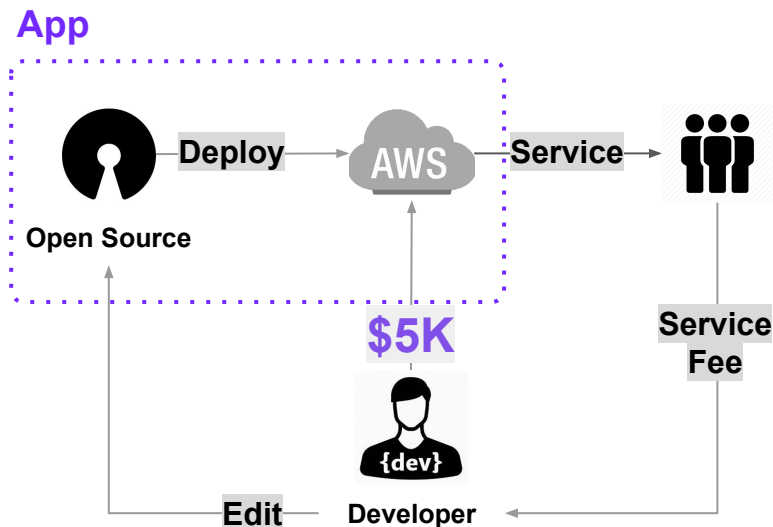


Source: Cisco, "Cisco Global Cloud Index: Forecast and Methodology, 2016–2021 White Paper," Cisco, 2018.



# Over 99% of apps will not make any money

- AS-IS**
- 6,000 apps published / day
  - 99% losing money



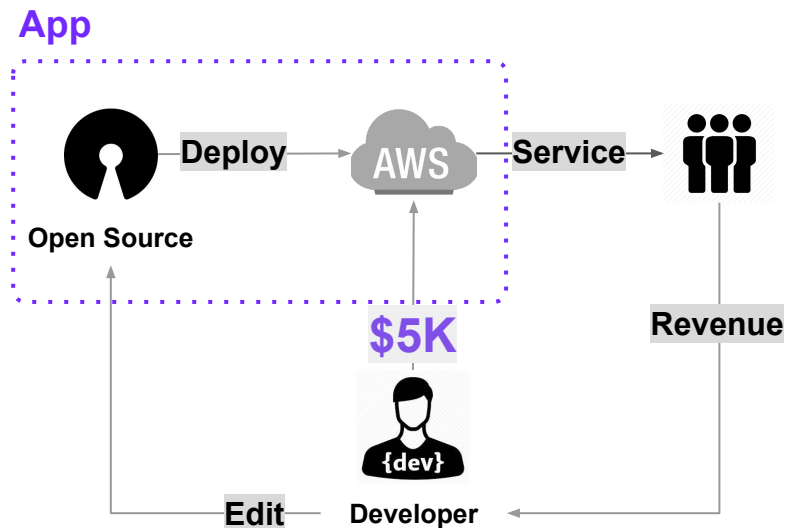
Gartner predicted less than 0.01% of consumer mobile apps will be financially successful.

→ Yet, they still spend avg. \$5K ~ \$50K to maintain their apps

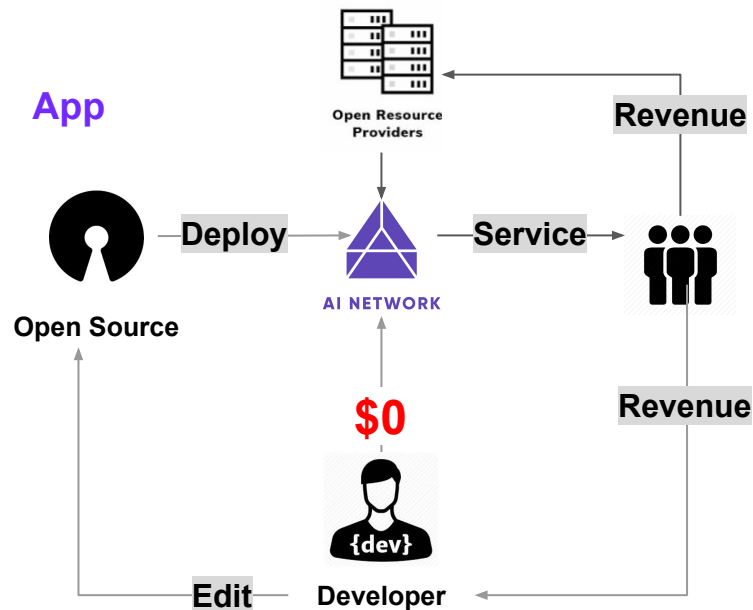


# App operation finally becomes **free!**

- AS-IS**
- 6,000 apps published / day
  - 99% losing money



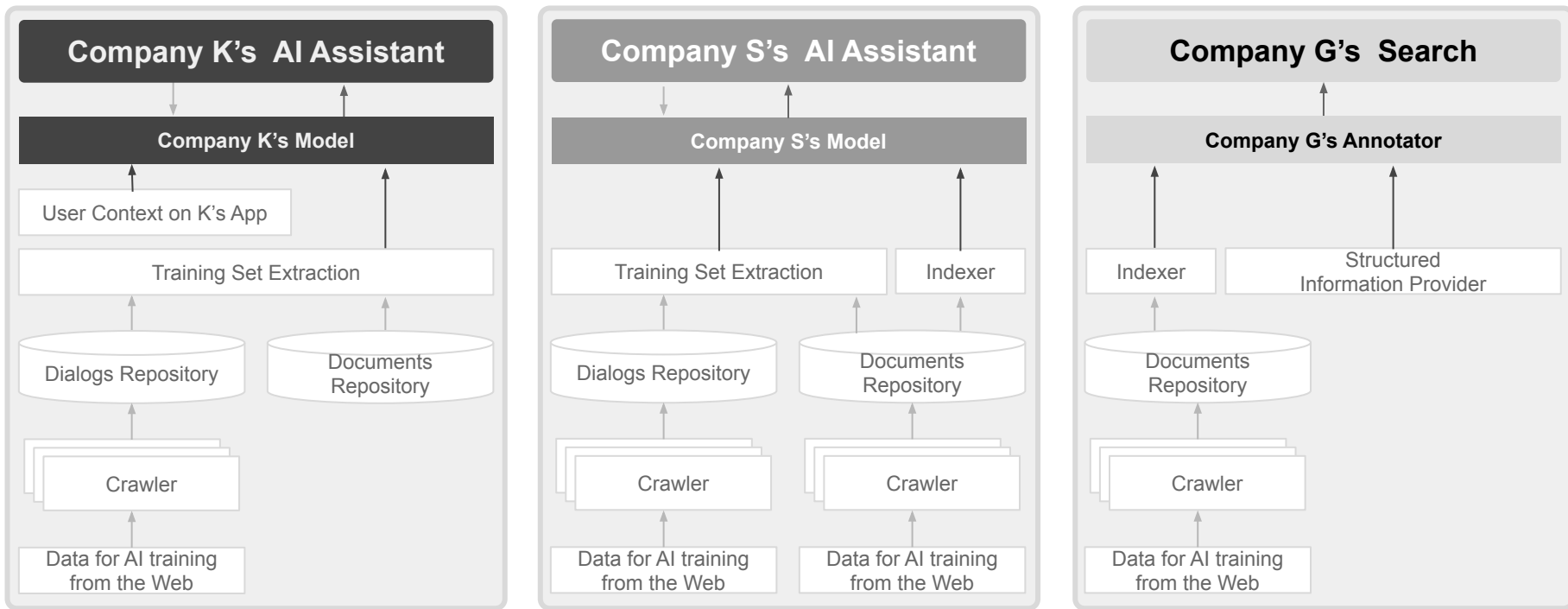
- TO-BE**
- 1M programs expected / day
  - Positive revenue



# Reinventing the wheel costs fortune (\$10M ~ \$350M).

AS-IS

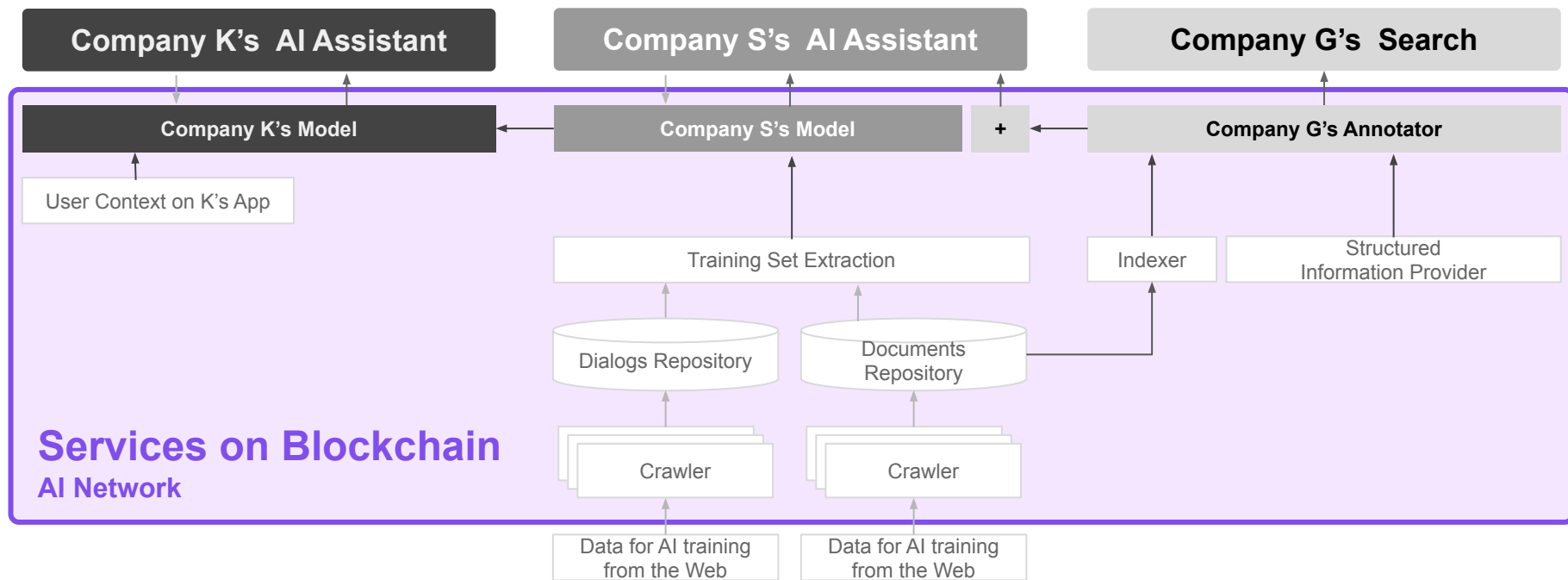
Each companies build their own components to create AI Assistants.



# Single individual can create big impact

TO-BE

**Components can be shared using our blockchain service.**

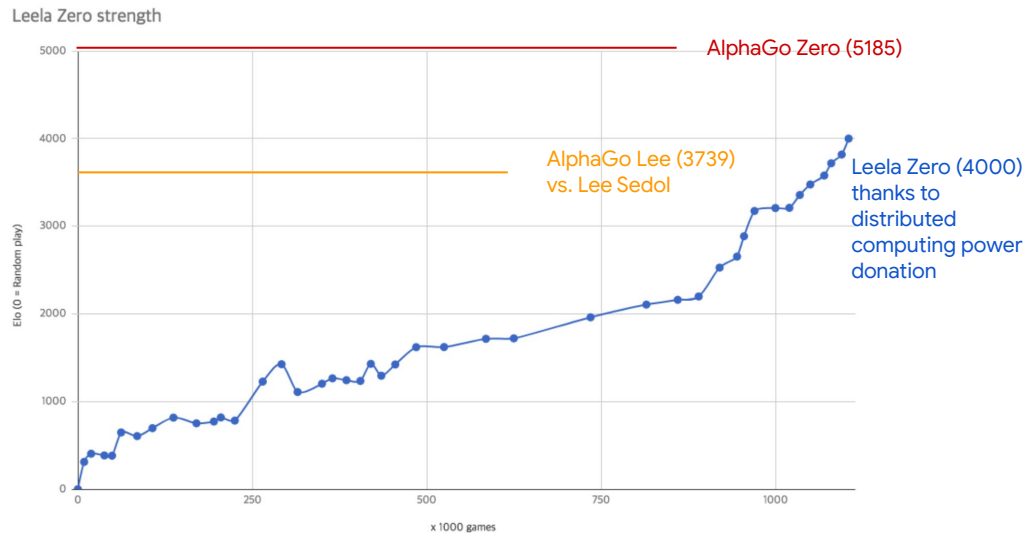


# Rethinking “Leela Zero” project


Leela Zero is a free and open-source computer Go software released on 25 October 2017. Its algorithm is based on DeepMind's paper about AlphaGo Zero and is trained by **distributed computing power**.

## Inspired by the project, we learned:

1. P2P Computation works. Though it is slow.
2. P2P computation works quite differently than centralized computing.
3. Multiple Stakeholders work together: “Author”, “Executor”, and “Resource provider”



# Leela Zero client on Colab's NVIDIA Tesla K80 GPU

 Leela\_zero\_K80.ipynb ☆

파일 수정 보기 삽입 런타임 도구 도움말

실습 모드에서 열기

공유 M

보기

This notebook shows how to run a **Leela Zero client on Google Colab's NVIDIA Tesla K80 GPU**. Thanks to [djinnome from the Leela Chess Zero GitHub who figured it out](#).

Run each cell in order, waiting for the previous one to finish before running the next.

The scripts and cell layout may be messy/redundant, but this should work.

```
[ ] !wget http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1704/x86_64/cuda-repo-ubuntu1704_9.0.176-1_amd64.deb
!apt-get install -y --fix-missing --no-install-recommends dirmngr
!dpkg -i cuda-repo-ubuntu1704_9.0.176-1_amd64.deb
!apt-key adv --fetch-keys https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1704/x86_64/7fa2af80.pub
!apt-get update
!mkdir /usr/lib/nvidia
!apt-get install -y --fix-missing --no-install-recommends linux-headers-generic nvidia-384=384.111-0ubuntu1 nvidia-opencl-dev nvidia-
!apt --fix-broken install
!apt-get install -y --fix-missing --no-install-recommends clinfo cmake git libboost-all-dev libopenblas-dev zlib1g-dev
!apt-get install build-essential qtbase5-dev qttools5-dev qttools5-dev-tools
!clinfo
```

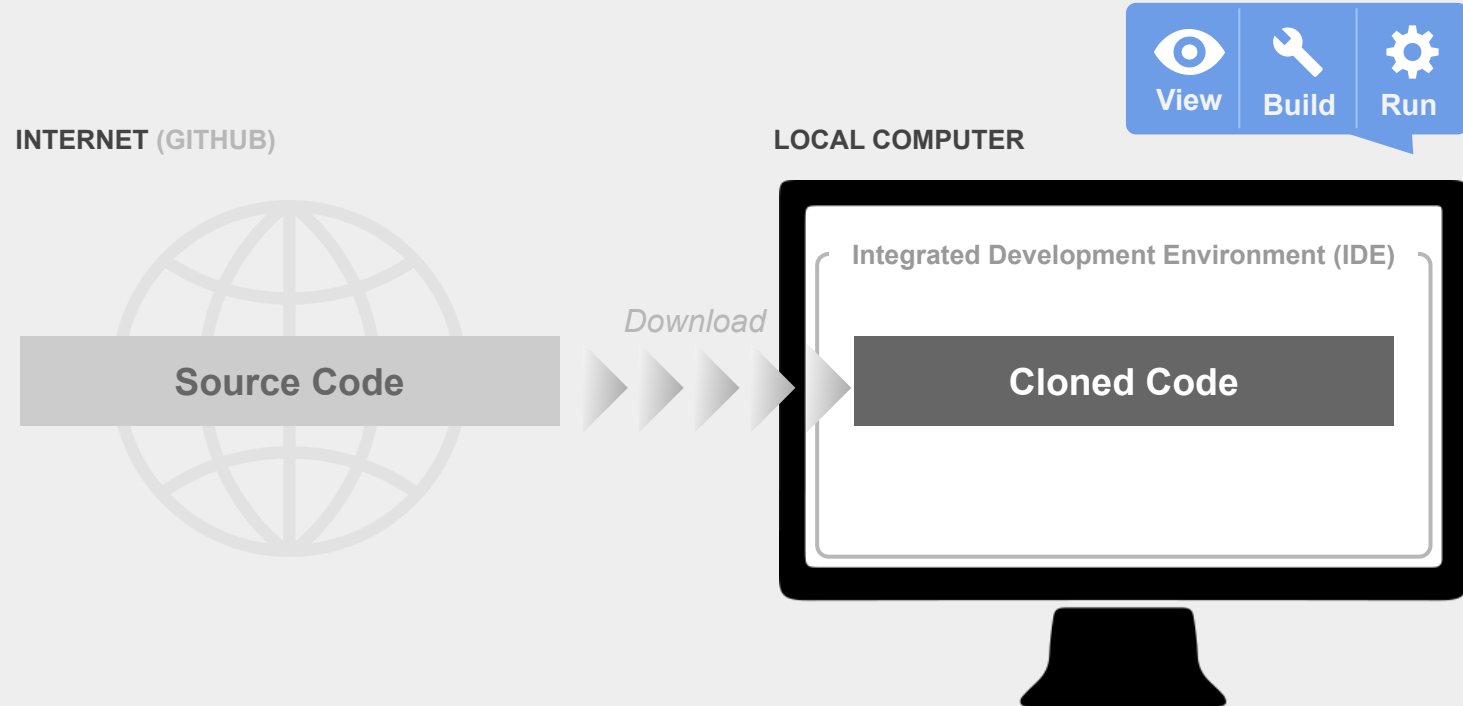
The first cell may take ~10 minutes to complete. You will know when it's done by the proper clinfo output showing the K80; also, the progress indicator will have stopped spinning



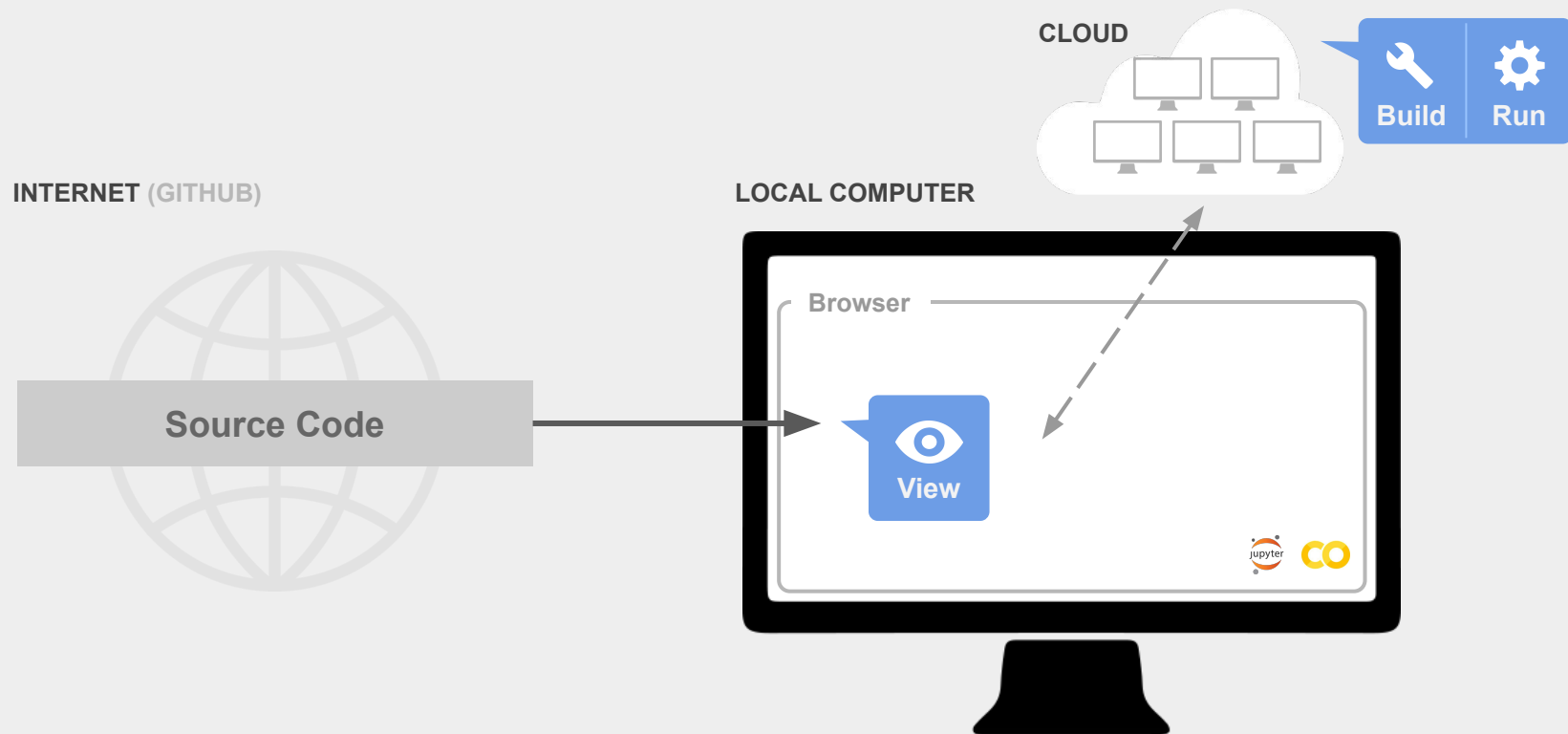
# Google is Cool

1. Google offers **free and unlimited access to the GPU**, but each session will stop running after 12 hours of use and need to be restarted. You must also **keep the browser tab open**. More details are .....
2. Do not use multiple accounts for training. Google has **notified us** they will block users for this.

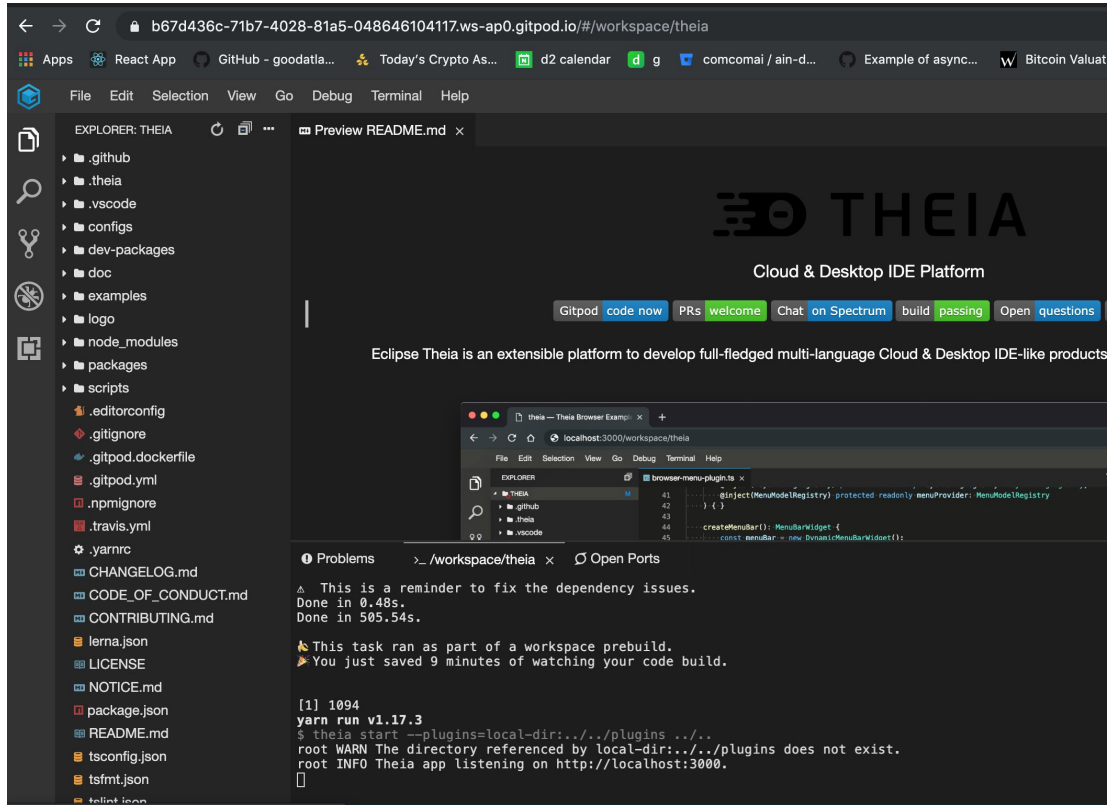
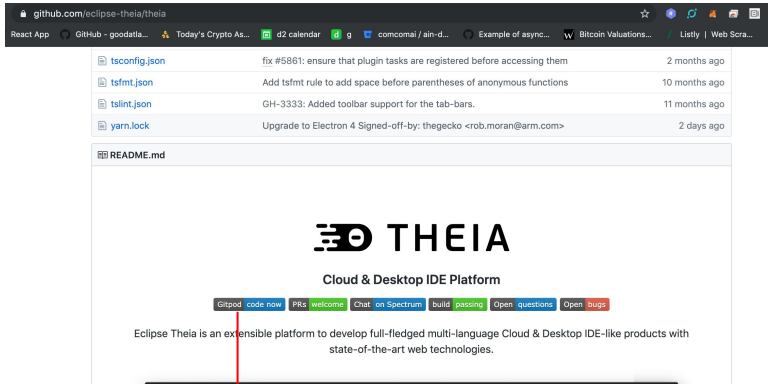
# The Traditional Development Environment



# The New Development Environment



# Cloud & Desktop IDE Platform



[https://deploy.cloud.run/?git\\_repo=https://github.com/GoogleCloudPlatform/cloud-run-hello.git](https://deploy.cloud.run/?git_repo=https://github.com/GoogleCloudPlatform/cloud-run-hello.git)

## Cloud Run Button

---

If you have a public repository, you can add this button to your [README](#) **Google Cloud Run** with a single click.

Try it out with a "hello, world" Go application ([source](#)):



Run on Google Cloud

# Where can I find resource?

AI Network (ainetwork.ai)

**Find me** 18 GPUs, latest CUDA, Tensorflow 1.9, face\_recognition installed



# Bert 환경 준비하기

## 1. 하드웨어 실행환경 준비

12GB이상의 가용 메모리를 가진 GPU가 탑재된 컴퓨터가 필요합니다. 현재 기준으로 약 115만원 정도인 NVIDIA사의 Tesla K40 급 이상의 GPU를 구비하면 되겠습니다.

## 2. 소프트웨어 실행환경 준비

그 다음은 NVIDIA GPU 드라이버([링크](#))와, Docker([링크](#)) 를 설치합니다.

## 3. 사전학습된 모델과 SQuAD 데이터 다운로드

아래 링크에서 원하는 모델과 데이터를 다운로드 받습니다.

- BERT pre-trained 모델 ([링크](#))

- SQuAD 1.1 ([링크](#). 최근 2.0 버전도 올라왔습니다)

# Bert 환경 준비하기

## 1. BERT 코드 다운로드 및 실행

BERT 코드를 GitHub([링크](#))로부터 clone한 후, run\_squad.py를 실행해서 fine-tuning을 수행하면 그 결과로 fine-tuning된 모델을 얻을 수 있습니다. 이때 Tensorflow 라이브러리가 설치된 Docker image (e.g. tensorflow/tensorflow:1.12.0-rc2-gpu)를 사용하면 됩니다.

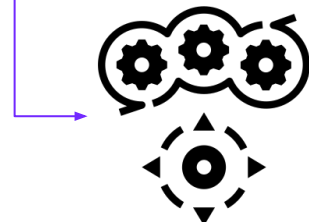
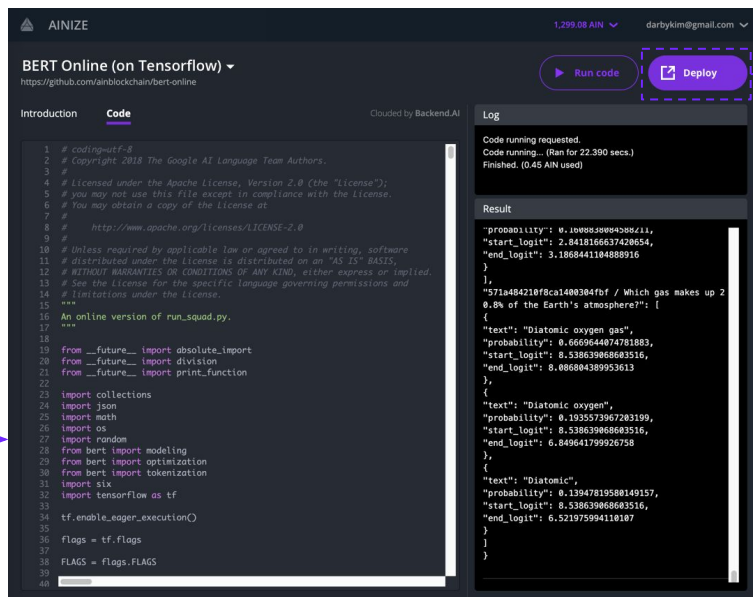
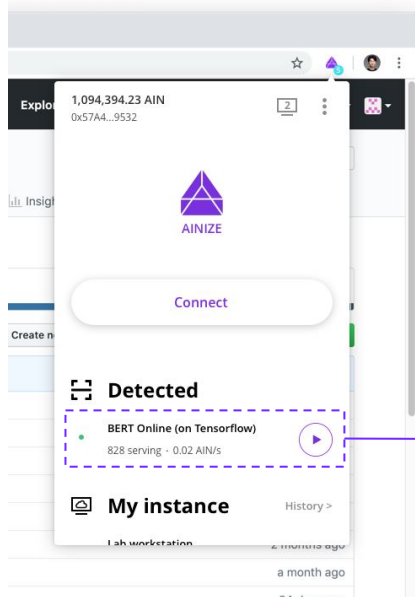
## 2. 미세조정 (Fine-tuning)

에어 서클레이터 & Memory Exceeded 에러를 막기 위해서 Training parameter를 살짝 튜닝...



# Code to Service in a minute

No need to spend huge time and money to configure SW/HW environment



Open AINIZE'd  
code from repo site

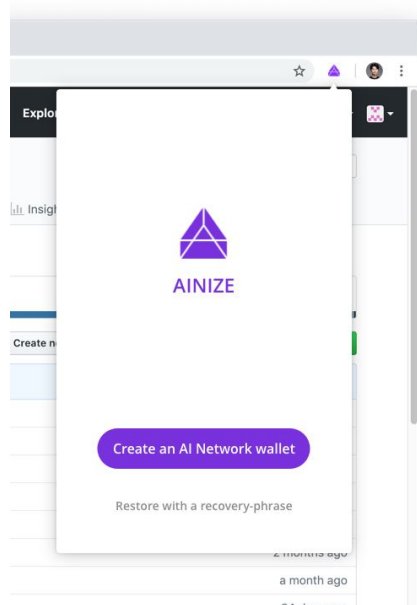
Tweak and Deploy the code  
(PoC site: <https://cloud.ainetwork.ai>)

Code runs  
serverlessly



# Super easy to share resources and earn money.

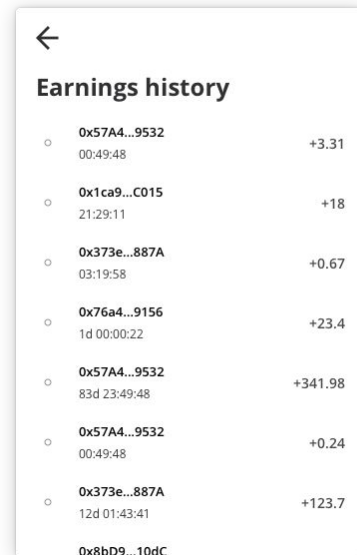
As sharing computers and creating earnings from them are easy, AI Network grows to a resourceful backend.



Create an  
AI Network wallet

`sudo apt-get install ain-worker`  
`ain-worker init`

2 commands of  
worker installation



Joined to AI Network and  
Create earnings



구독해주세요

<http://www.ainize.ai>