

Open Infrastructure & Cloud Native Days Korea 2019
핸즈온 워크샵

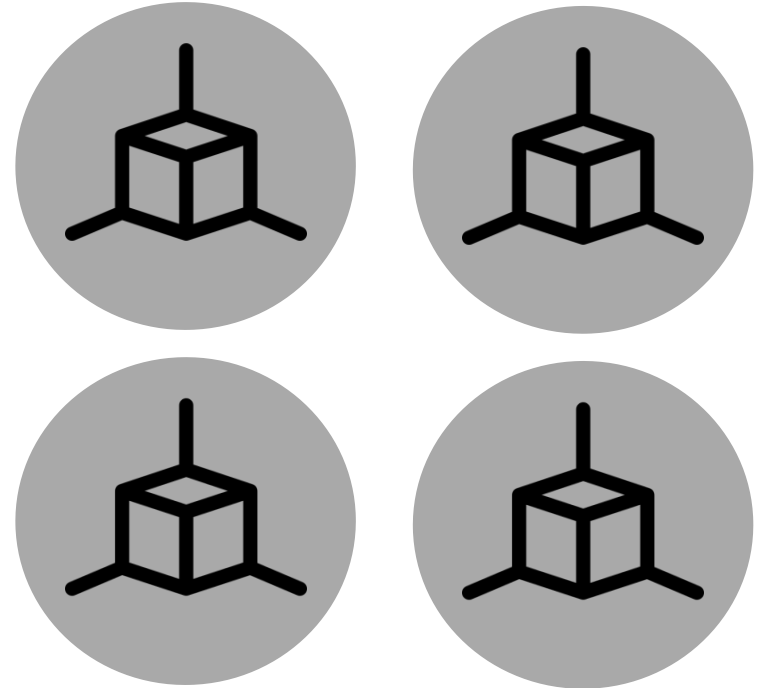
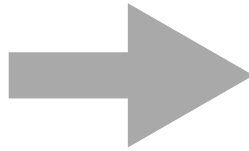
How to scale your ML job with Kubernetes

LG전자 유흥근



오늘의 Workshop 내용

- 손쉽게 머신러닝 학습 확장하기



안녕하세요, 저는요



커피고래 (coffeewhale.com)

- ML, Container

술 좋아하면 술고래,
커피 좋아하면 커피고래

학사 - 컴퓨터 공학

석사 - 텍스트 분석

커피고래

LG전자 스마트데이터실

오늘 다루지 않을 내용

- 쿠버네티스에 대한 상세한 내용

API server? ❌



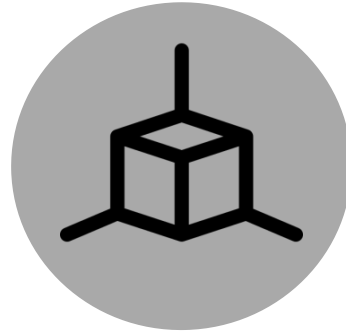
kubernetes

Kubelet? ❌

오늘 다루지 않을 내용

- 모델링에 대한 이야기

새로운 모델 소개? ❌



Model

모델 적용 사례? ❌

발표내용

누구나

손쉽게 기계학습 훈련

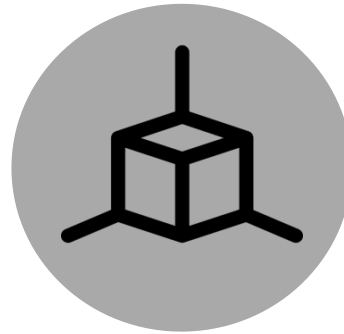
확장시키는 방법

feat. Kubernetes

기계학습 엔지니어링의 중요성

- 쿠버네티스를 이용하여 손쉽게 모델을 학습 시킵시다!

뛰어난 모델



탄탄한 머신러닝 엔지니어링 스킬



kubernetes

기계학습 엔지니어링의 중요성

- 왜 중요할까요?

AI 연구에서 엔지니어링 스킬은 생각보다 중요

조직 이동전에는 연구에 엔지니어링 스킬이 어느정도 필요할지 감을 잡지 못했다. 지금은 엔지니어링 스킬이 연구를 위해 가장 필요한 체력적인 부분이라는 것에 100% 공감하고 있다. 축구 선수로 치면 체력이 좋아야 슈트를 많이 시도할 수 있는 것과 같다.

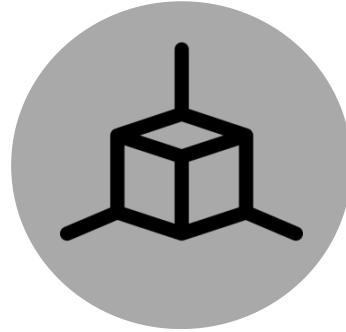
“데이터 과학자에서 AI 연구자로 들어서며...”

SKT T-Brain 전희원

<http://freesearch.pe.kr/archives/4905>

모델

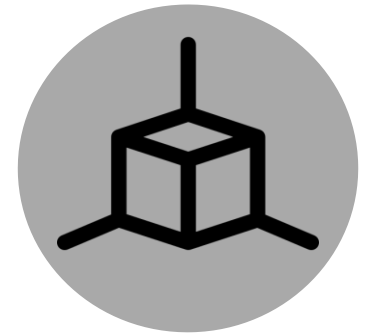
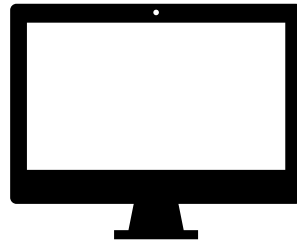
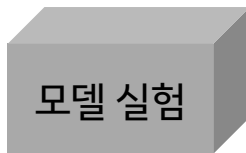
- 모델이란 무엇일까?



Model

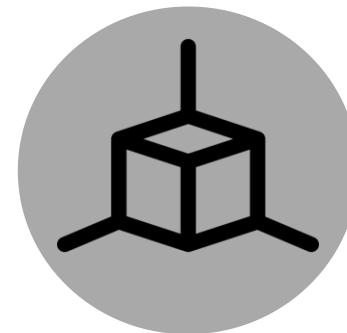
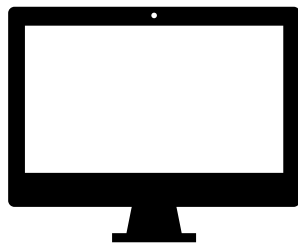
모델

- 기계학습 시대



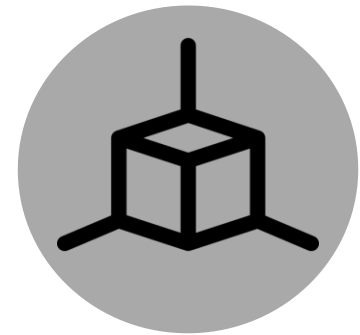
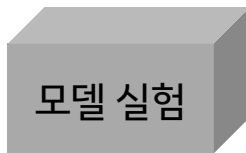
모델

- 기계학습 시대



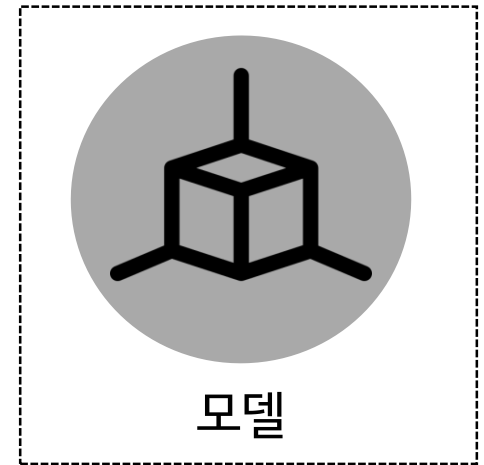
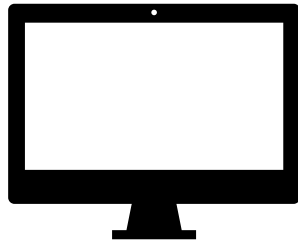
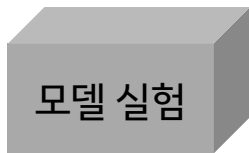
모델

- 기계학습 시대



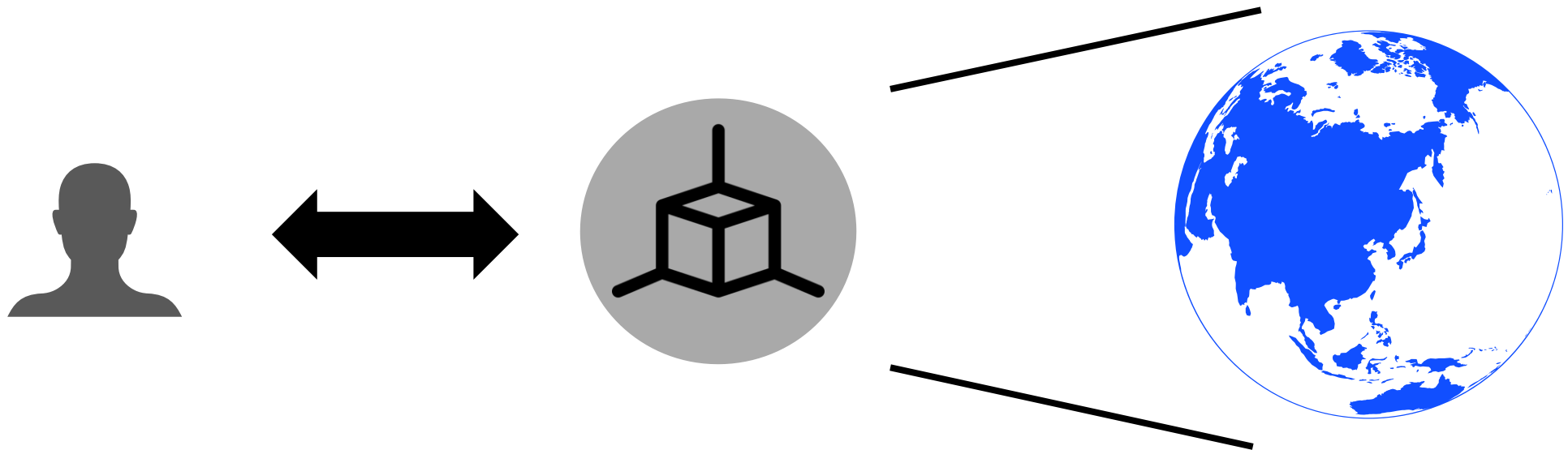
모델

- 기계학습 시대



모델

- 조금 더 근본적인 생각
 - 현실 세계의 모습을 반영하는 매개체



Rule Based 모델

- 나의 주말 계획 모델

```
def my_weekend_plan(status):
```

```
    if status == boring:
```

```
        return play()
```

```
    elif status == hungry:
```

```
        return eat()
```

```
    else:
```

```
        return sleep()
```

Numeric 모델

- 거리 계산 모델

```
def distance_cal(velocity, time):  
    return velocity * time
```


확률적 모델

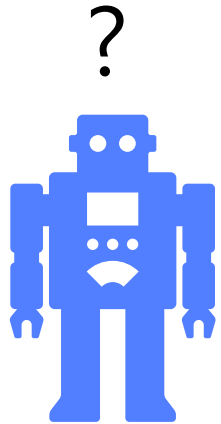
- 동전 앞면이 나올 확률

$$p(\text{H}) = 0.5$$

$$p(\text{T}) = 0.5$$

기계가 학습한 모델

- 사람 손으로 풀기 어려운 복잡한 문제

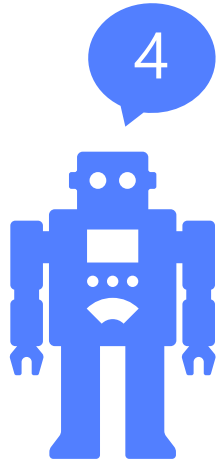


$$y = Ax + Bx^2 + C$$

풀기 어려운 문제

기계가 학습한 모델

- 기계한테 맡겨 보자!

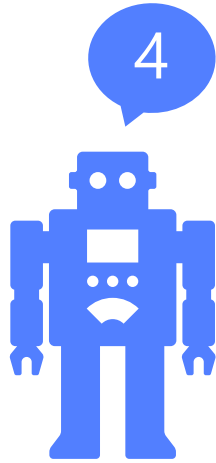


$$y = Ax + Bx^2 + C$$

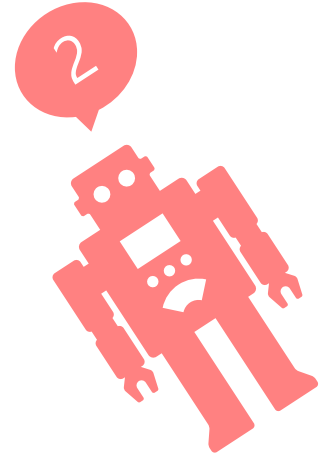
풀기 어려운 문제

기계가 학습한 모델

- 여러가지 경우의 수를 계산

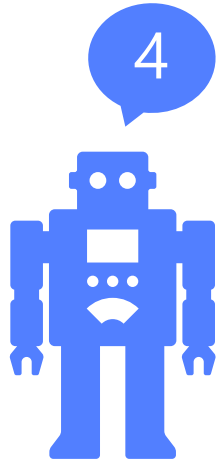


$$y = Ax + Bx^2 + C$$

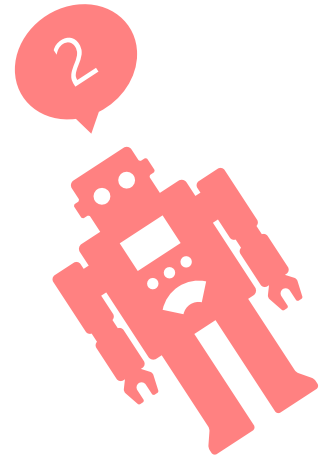
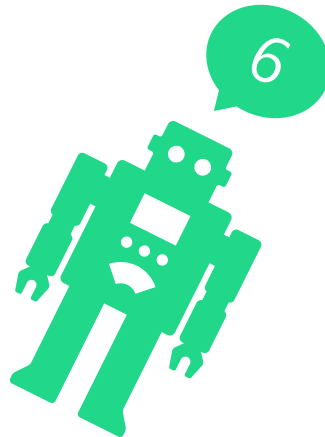


기계가 학습한 모델

- 필연적으로 많은 기계가 필요

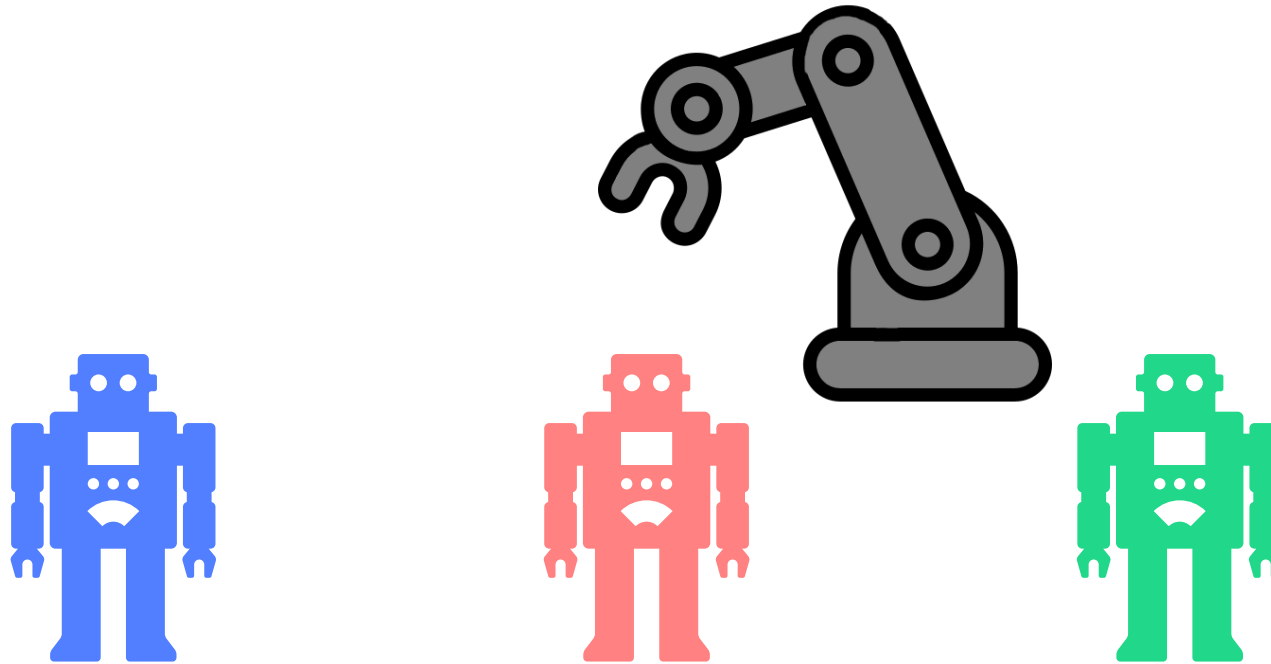


$$y = Ax + Bx^2 + C$$



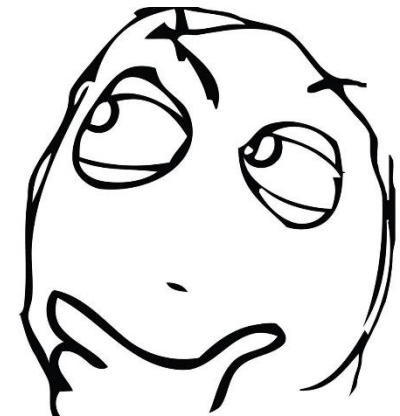
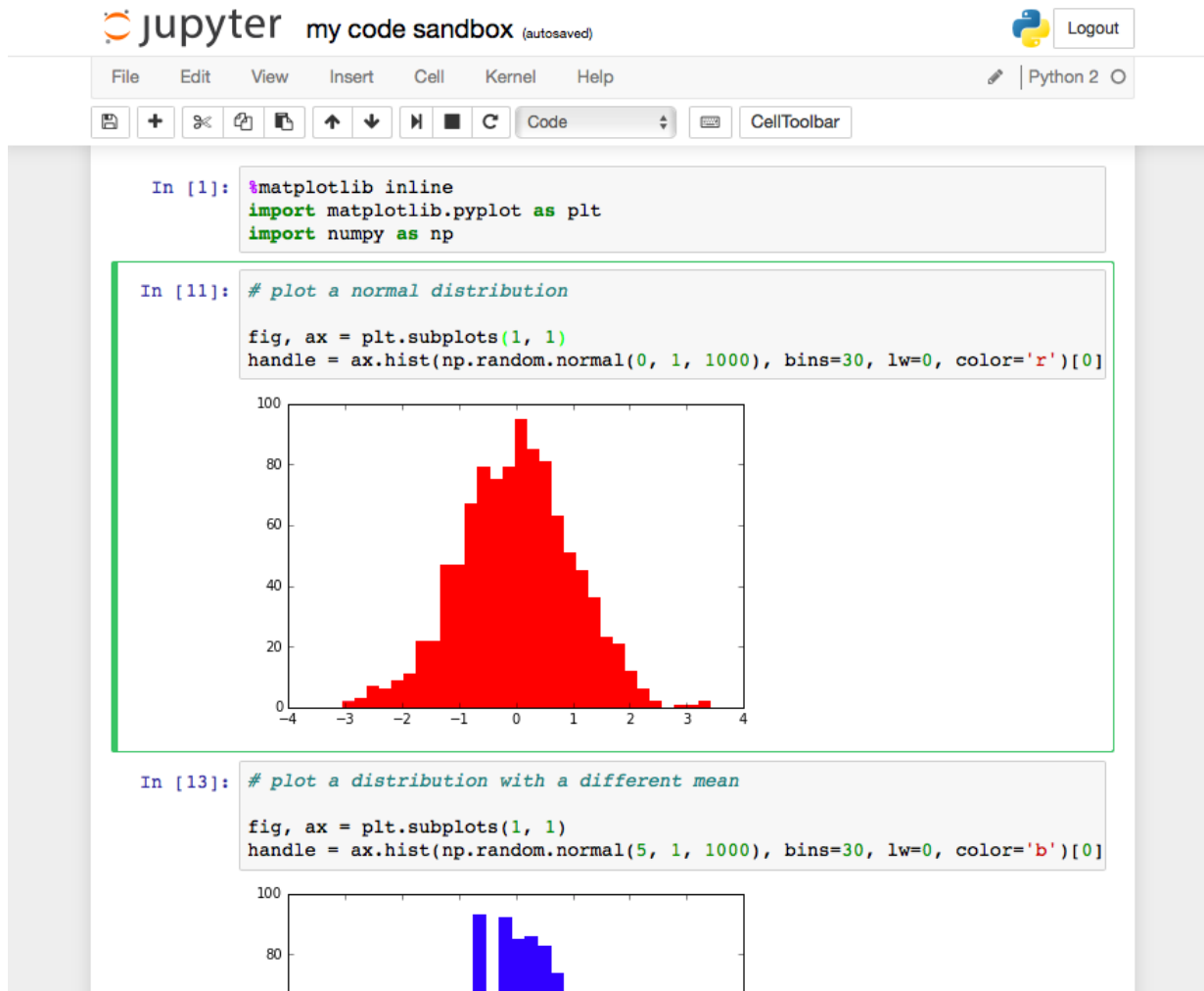
시작하게 된 고민

- 어떻게 하면 많은 기계들을 효율적으로 관리할 수 있을까?



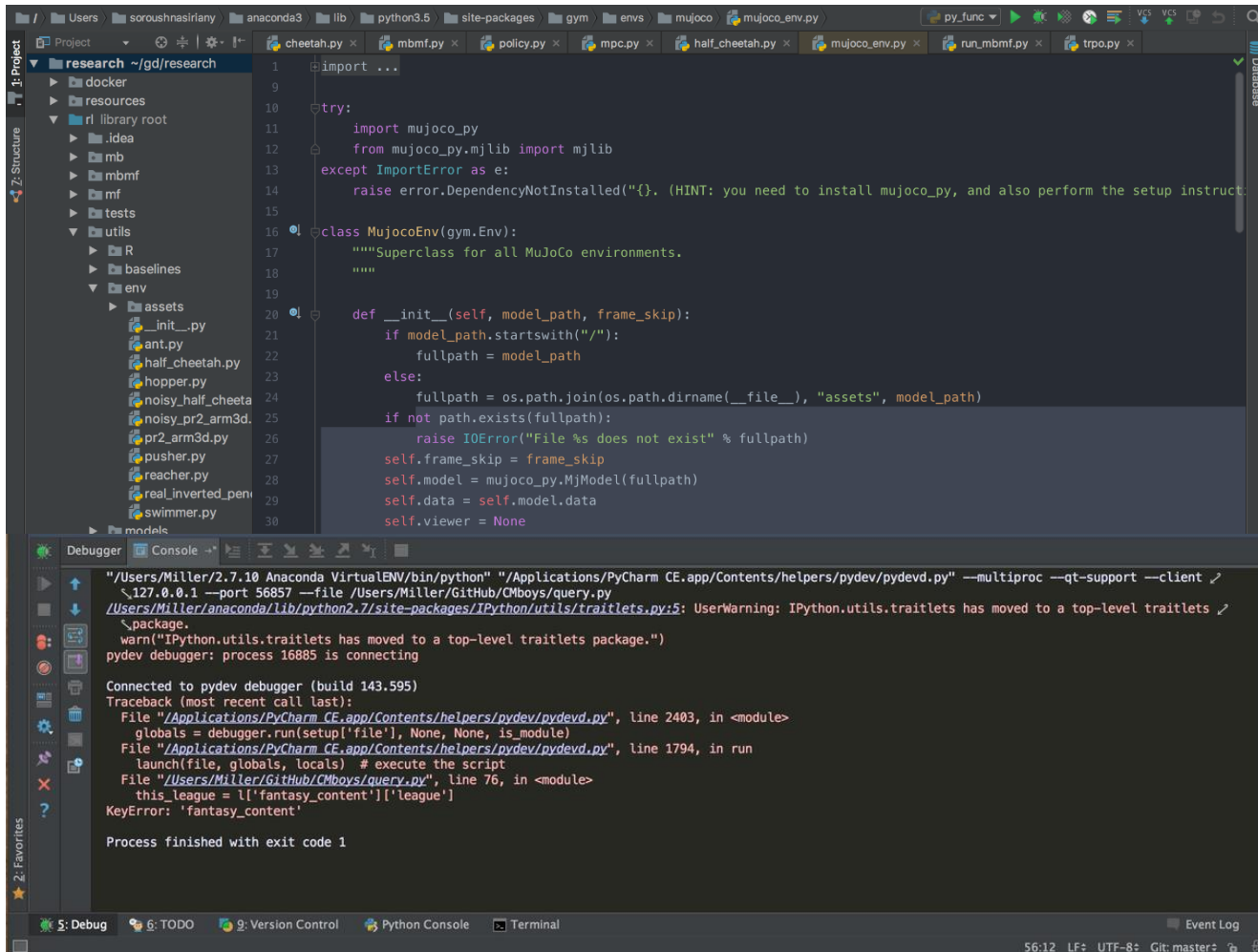
나의 첫 모델 개발기

- 주피터 노트북으로 EDA & Vanilla 모델 개발



나의 첫 모델 개발기

- Run python



The screenshot shows the PyCharm IDE interface. The top pane displays a Python script named `mujoco_env.py` with the following code:

```
1 import ...
2
3 try:
4     import mujoco_py
5     from mujoco_py.mjlib import mjlib
6 except ImportError as e:
7     raise error.DependencyNotInstalled("{} (HINT: you need to install mujoco_py, and also perform the setup instructions from the README of the first repository)".format(e))
8
9
10 class MujocoEnv(gym.Env):
11     """Superclass for all MuJoCo environments.
12     """
13
14     def __init__(self, model_path, frame_skip):
15         if model_path.startswith("/"):
16             fullpath = model_path
17         else:
18             fullpath = os.path.join(os.path.dirname(__file__), "assets", model_path)
19         if not path.exists(fullpath):
20             raise IOError("File %s does not exist" % fullpath)
21         self.frame_skip = frame_skip
22         self.model = mujoco_py.MjModel(fullpath)
23         self.data = self.model.data
24         self.viewer = None
```

The bottom pane shows the console output, which includes a warning about the `IPython.utils.traitlets` package and a `KeyError` exception:

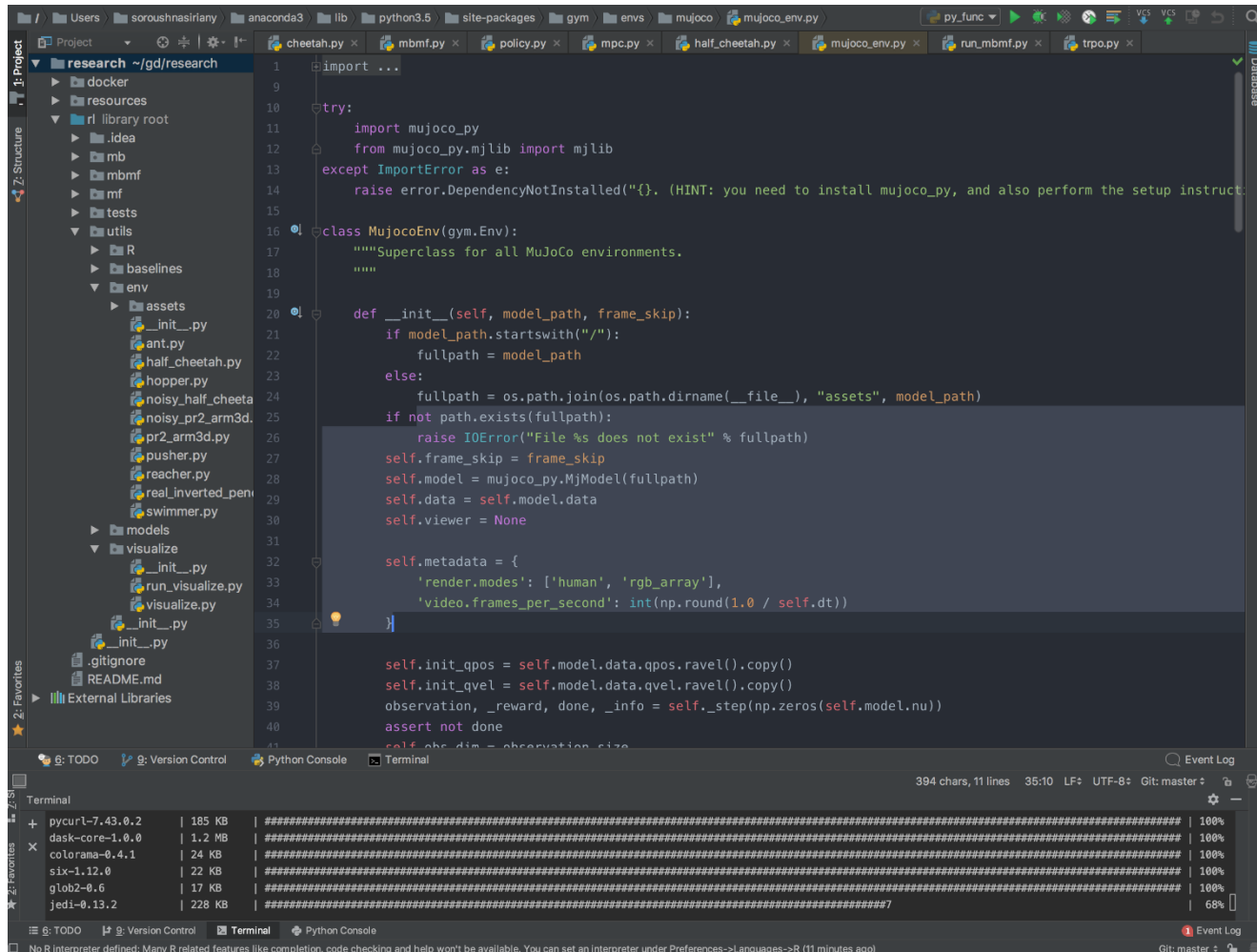
```
"/Users/Miller/2.7.10 Anaconda VirtualEnv/bin/python" "/Applications/PyCharm CE.app/Contents/helpers/pydev/pydevd.py" --multiproc --qt-support --client 127.0.0.1 --port 56857 --file /Users/Miller/GitHub/CMBoys/query.py
/Users/Miller/anaconda/lib/python2.7/site-packages/IPython/utils/traitlets.py:5: UserWarning: IPython.utils.traitlets has moved to a top-level traitlets package.
warn("IPython.utils.traitlets has moved to a top-level traitlets package.")
pydev debugger: process 16885 is connecting

Connected to pydev debugger (build 143.595)
Traceback (most recent call last):
  File "/Applications/PyCharm CE.app/Contents/helpers/pydev/pydevd.py", line 2403, in <module>
    globals = debugger.run(setup['file'], None, None, is_module)
  File "/Applications/PyCharm CE.app/Contents/helpers/pydev/pydevd.py", line 1794, in run
    launch(file, globals, locals) # execute the script
  File "/Users/Miller/GitHub/CMBoys/query.py", line 76, in <module>
    this_league = l['fantasy_content']['league']
KeyError: 'fantasy_content'

Process finished with exit code 1
```


나의 첫 모델 개발기

- Success!



The screenshot shows an IDE with a project structure on the left and a Python script in the center. The script defines a `MujocoEnv` class that inherits from `gym.Env`. It includes methods for initialization, step, and rendering. The terminal at the bottom shows the installation progress of various packages.

```
1 import ...
2
3
4
5
6
7
8
9
10 try:
11     import mujoco_py
12     from mujoco_py.mjlib import mjlib
13 except ImportError as e:
14     raise error.DependencyNotInstalled("{} (HINT: you need to install mujoco_py, and also perform the setup instruct
15
16
17
18
19
20 class MujocoEnv(gym.Env):
21     """Superclass for all MuJoCo environments.
22     """
23
24     def __init__(self, model_path, frame_skip):
25         if model_path.startswith("/"):
26             fullpath = model_path
27         else:
28             fullpath = os.path.join(os.path.dirname(__file__), "assets", model_path)
29         if not path.exists(fullpath):
30             raise IOError("File %s does not exist" % fullpath)
31         self.frame_skip = frame_skip
32         self.model = mujoco_py.MjModel(fullpath)
33         self.data = self.model.data
34         self.viewer = None
35
36         self.metadata = {
37             'render.modes': ['human', 'rgb_array'],
38             'video.frames_per_second': int(np.round(1.0 / self.dt))
39         }
40
41         self.init_qpos = self.model.data.qpos.ravel().copy()
42         self.init_qvel = self.model.data.qvel.ravel().copy()
43         observation, _reward, done, _info = self._step(np.zeros(self.model.nu))
44         assert not done
45         self.obs_dim = observation.size
```

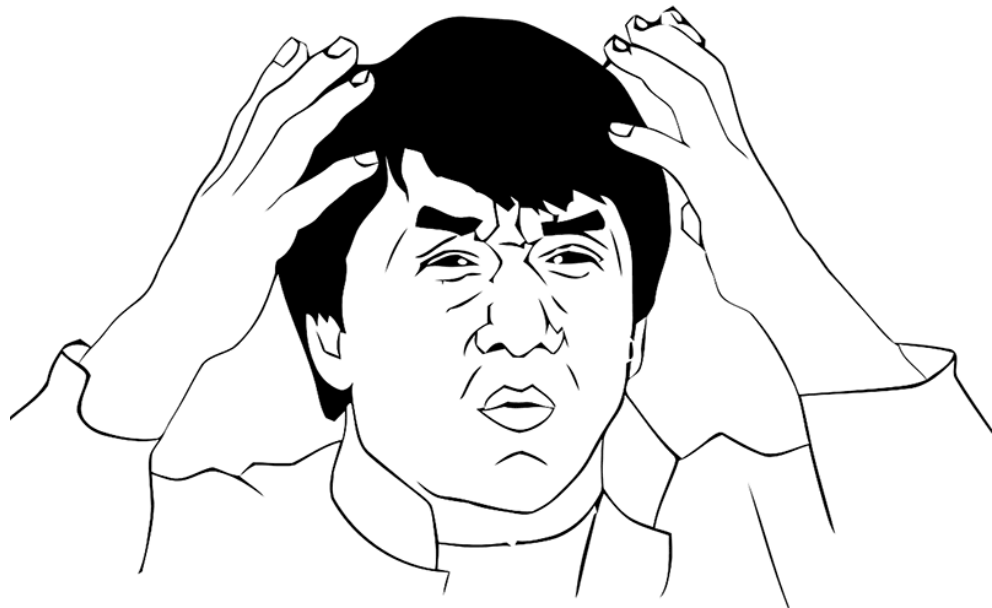
Package	Size	Progress
pycurl-7.43.0.2	185 KB	100%
dask-core-1.0.0	1.2 MB	100%
colorama-0.4.1	24 KB	100%
six-1.12.0	22 KB	100%
glob2-0.6	17 KB	100%
jedi-0.13.2	228 KB	68%



나의 첫 모델 개발기

- 2시간뒤...

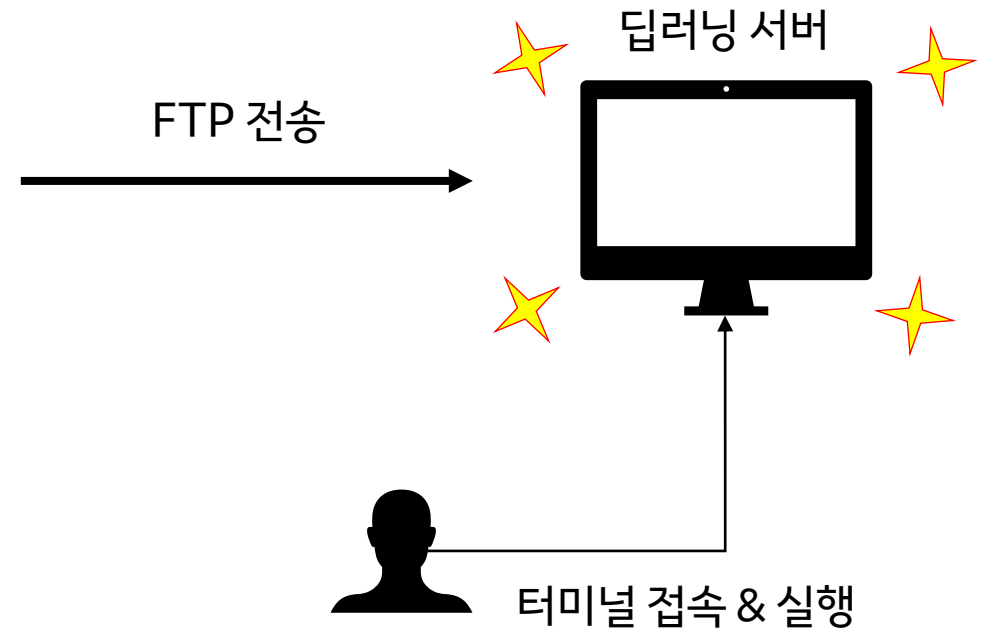
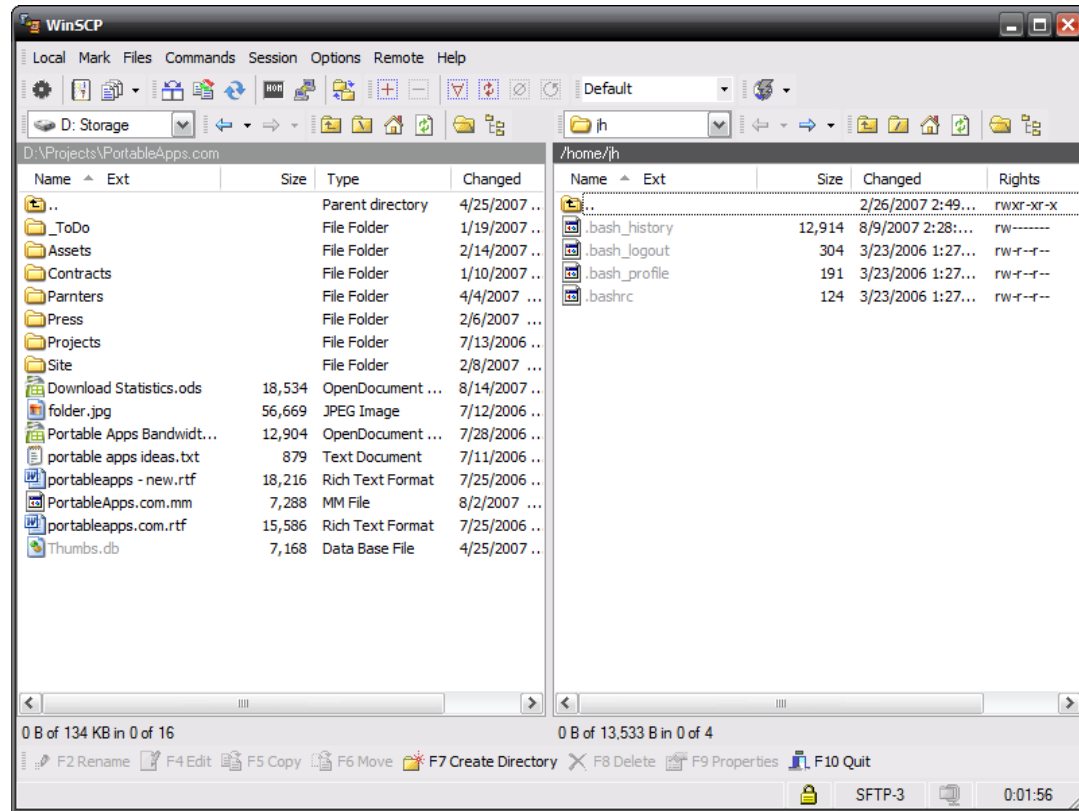
It's too slow!!



```
3%|█          | 3/100 [00:03<01:37, 1.00s/it]
```

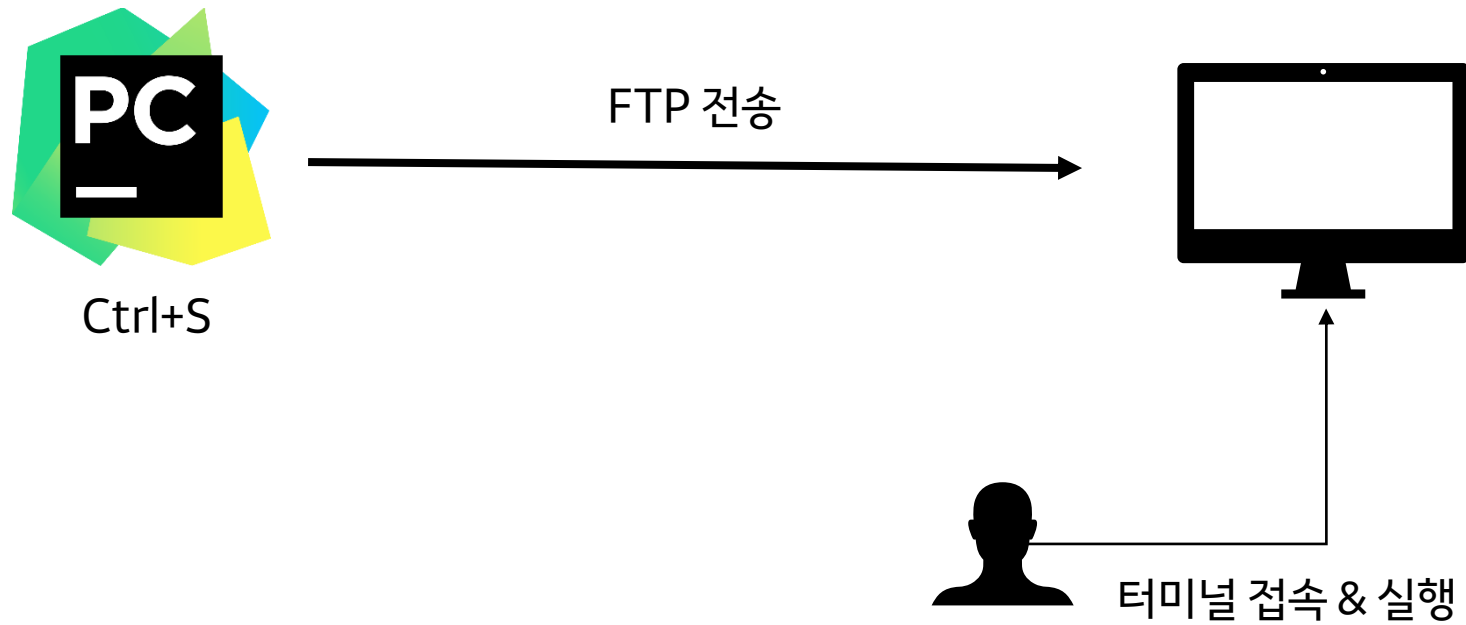
나의 첫 모델 개발기

- FTP로 서버로 파일 전송 후 학습 실행



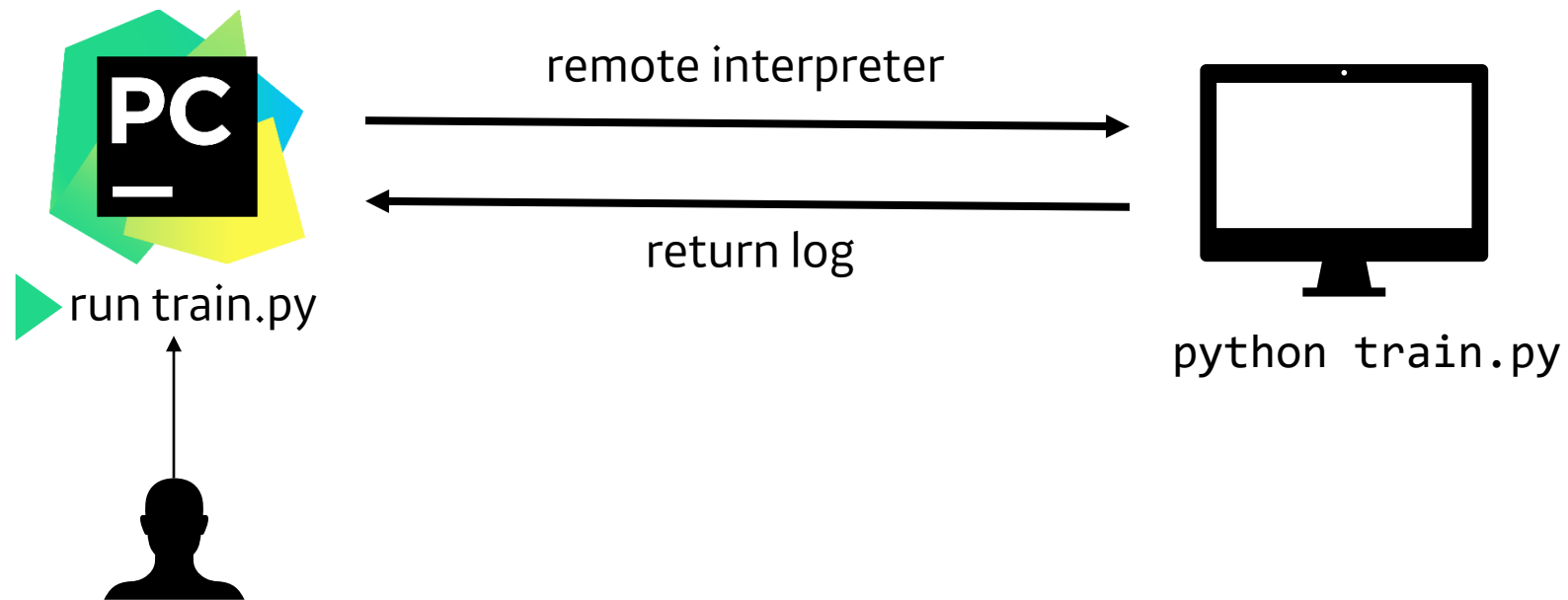
나의 첫 모델 개발기

- 파일 저장 → 자동 FTP 전송



나의 첫 모델 개발기

- Remote execution (professional)



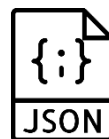
나의 첫 모델 개발기

- 좋아, 이제 모델 실험의 개수를 늘려보자!

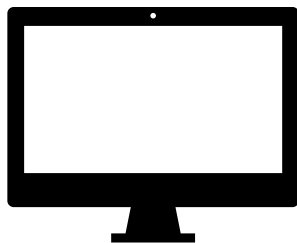


나의 첫 모델 개발기

- 모델 실험 리스트 생성



params.json



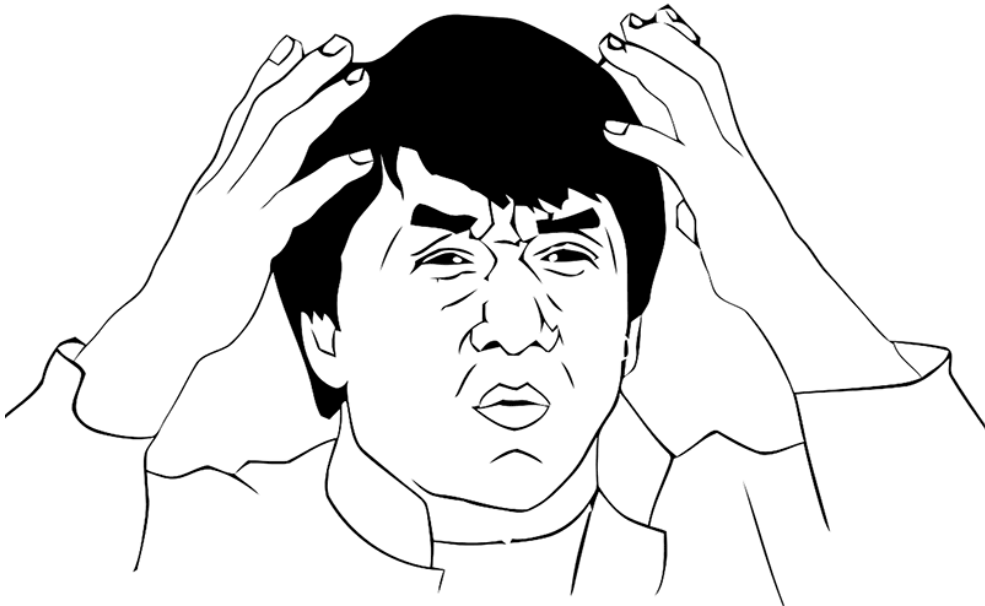
python train.py params.json

```
[
  {
    "model": "CNN-01",
    "preproc": "algo-01",
    "params": [
      "weight=0.5 depth=2",
      "weight=0.3 depth=3"
    ]
  },
  {
    "model": "RNN-01",
    "preproc": "algo-02",
    "params": [
      "dropout=0.5 act=softmax",
      "dropout=0.7 act=tanh"
    ]
  },
  .....
]
```

나의 첫 모델 개발기

- 실험 모델이 많이지다 보니...

It's too slow!!



```
3%|  | 3/100 [00:03<01:37, 1.00s/it]
```

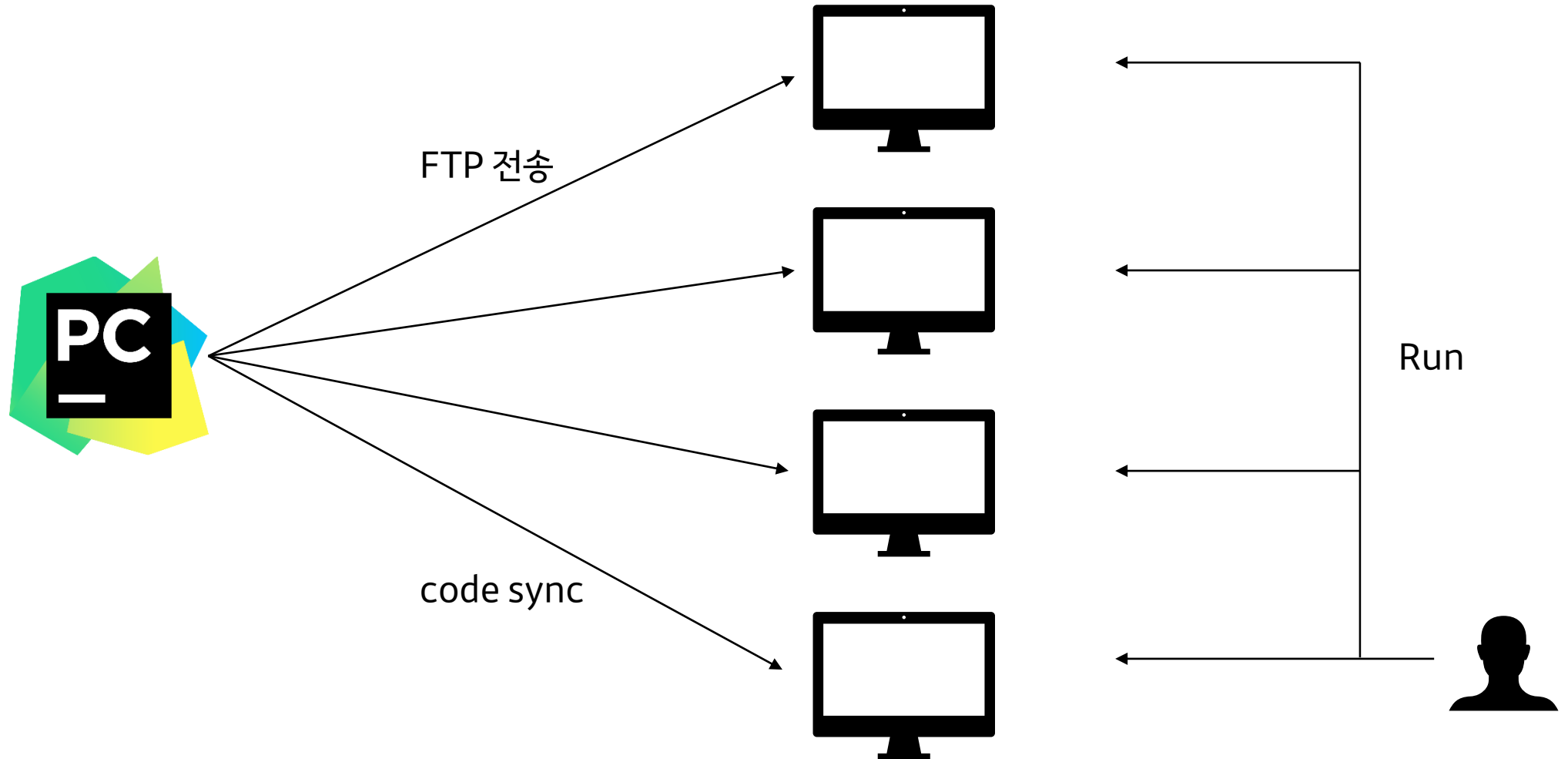

나의 첫 모델 개발기

- 좋아, 이제 서버의 개수를 늘려보자!



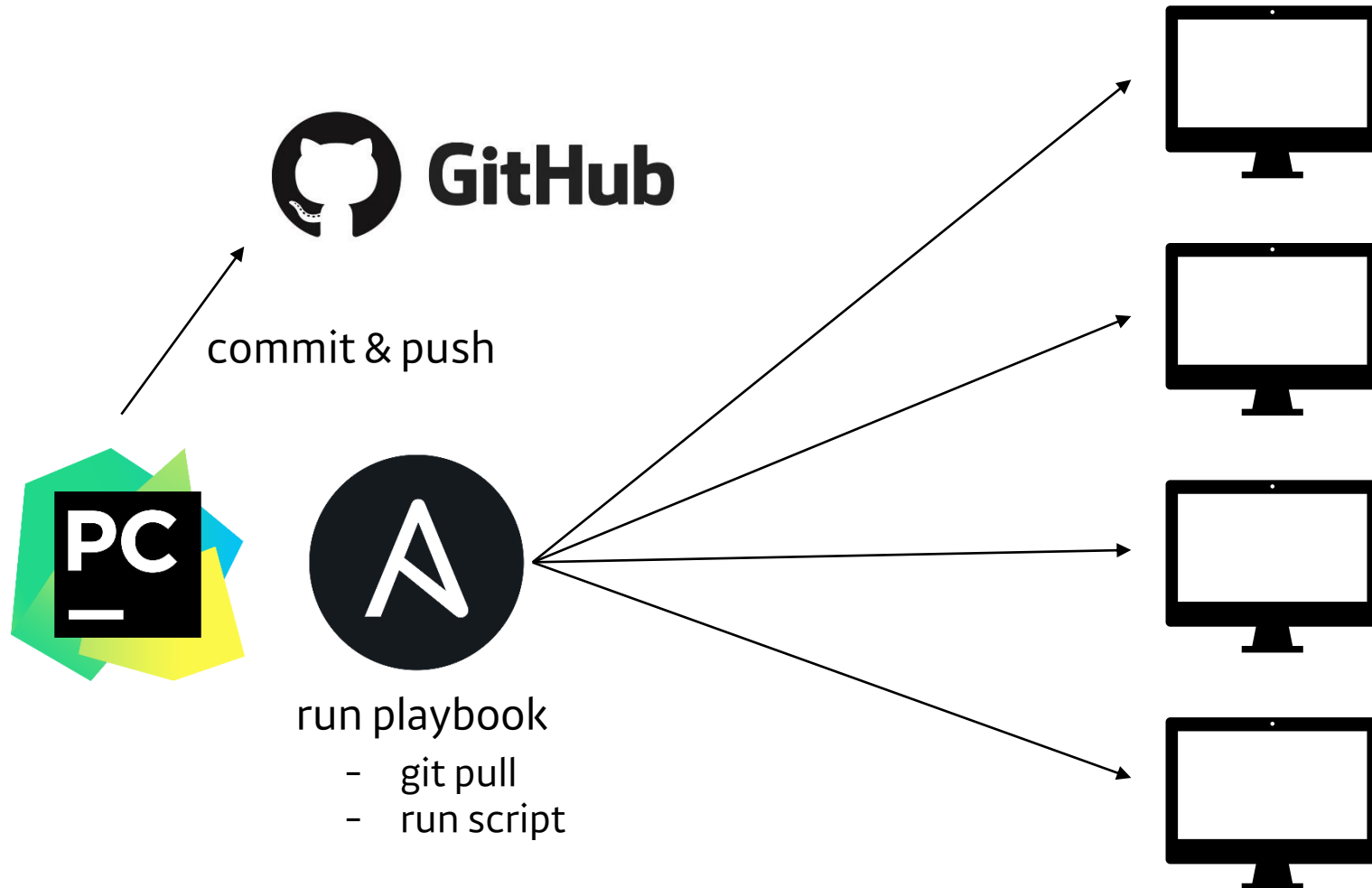
나의 첫 모델 개발기

- 각 서버에 FTP 전송 & Run



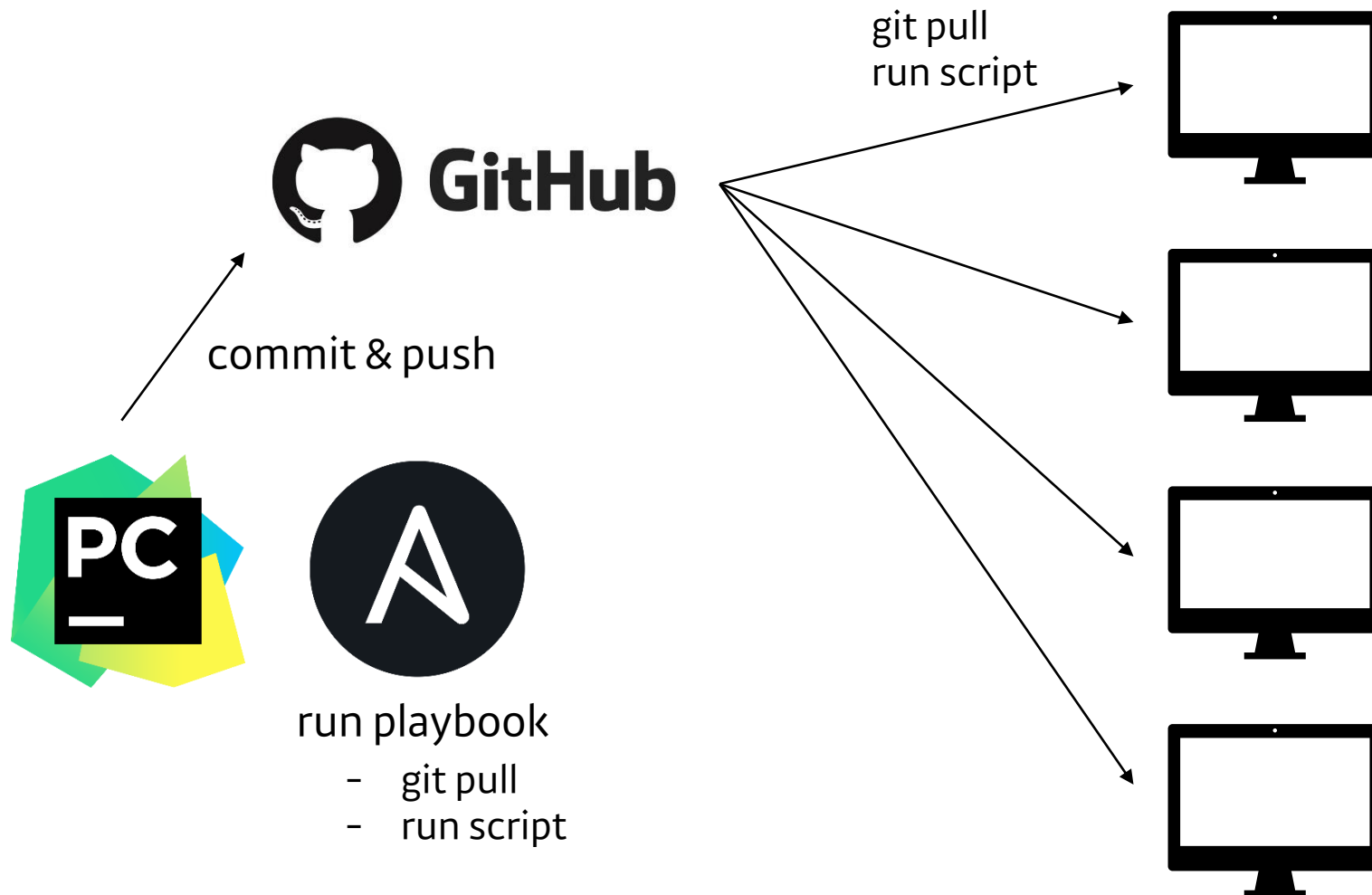
나의 첫 모델 개발기

- 깃&ansible 사용



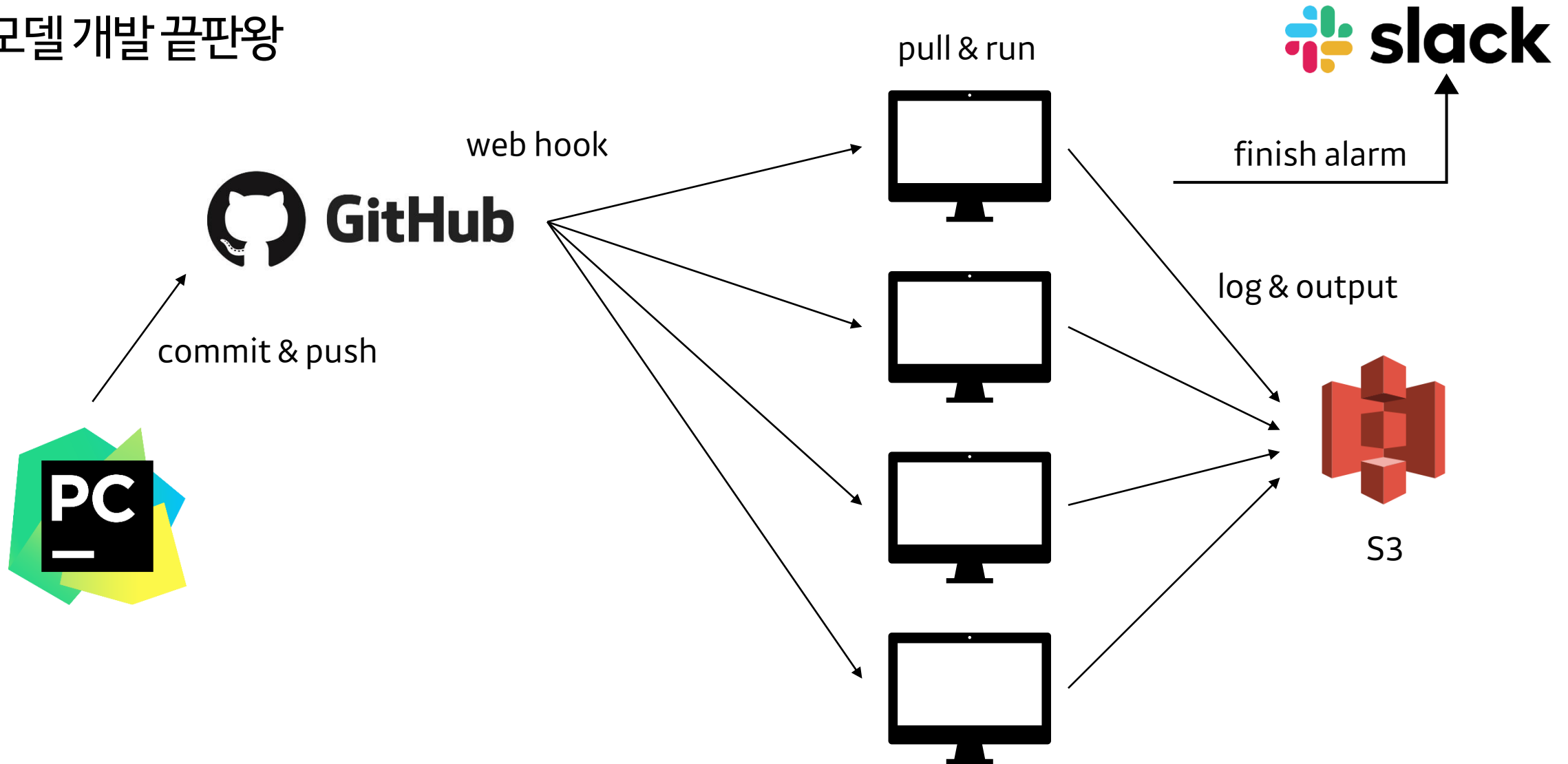
나의 첫 모델 개발기

- 깃&ansible 사용



나의 첫 모델 개발기






















- 모델 개발 끝판왕



여전히 문제

- 어지럽혀지는 커밋 로그

Commits on Sep 6, 2017

a  hongkunyoo committed on Sep 6, 2017	 cf0cc45	
a  hongkunyoo committed on Sep 6, 2017	 e35fe6d	
a  hongkunyoo committed on Sep 6, 2017	 70e53a7	
a  hongkunyoo committed on Sep 6, 2017	 518af7b	
a  hongkunyoo committed on Sep 6, 2017	 84d36ad	
a  hongkunyoo committed on Sep 6, 2017	 b21196d	
a  hongkunyoo committed on Sep 6, 2017	 64f9c40	

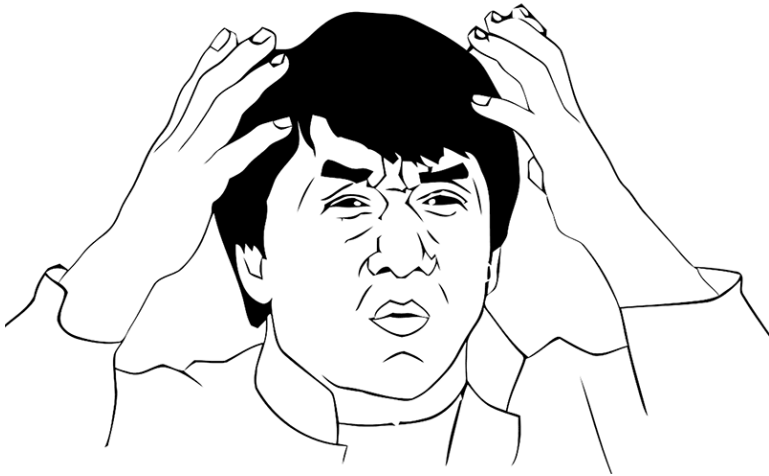
여전히 문제

- 학습 끝나는 시간이 달라서 개별적으로 재실행



여전히 문제

- 어느 서버에서 어떤 모델 실험이 돌아가는지 일일이 찾기가 힘들



여전히 문제

- 자원 모니터링 & 에러 파악 확인

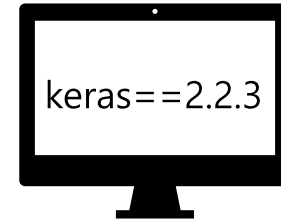
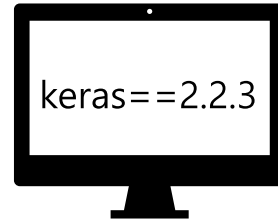
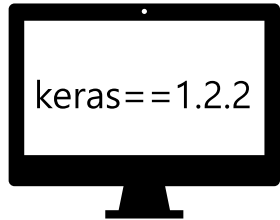
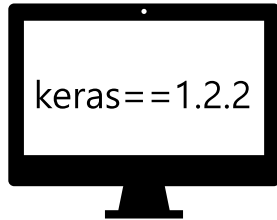


```
Tasks: 205 total, 9 running
Load average: 4.95 2.77 1.34
Uptime: 59 days, 09:32:48

1  [|||||100.0%]
2  [|||||100.0%]
3  [|||||100.0%]
4  [|||||100.0%]
5  [|||||100.0%]
6  [|||||100.0%]
7  [|||||100.0%]
8  [|||||100.0%]
Mem[|||||19612/32231MB]
Swp[|178/32767MB]
```

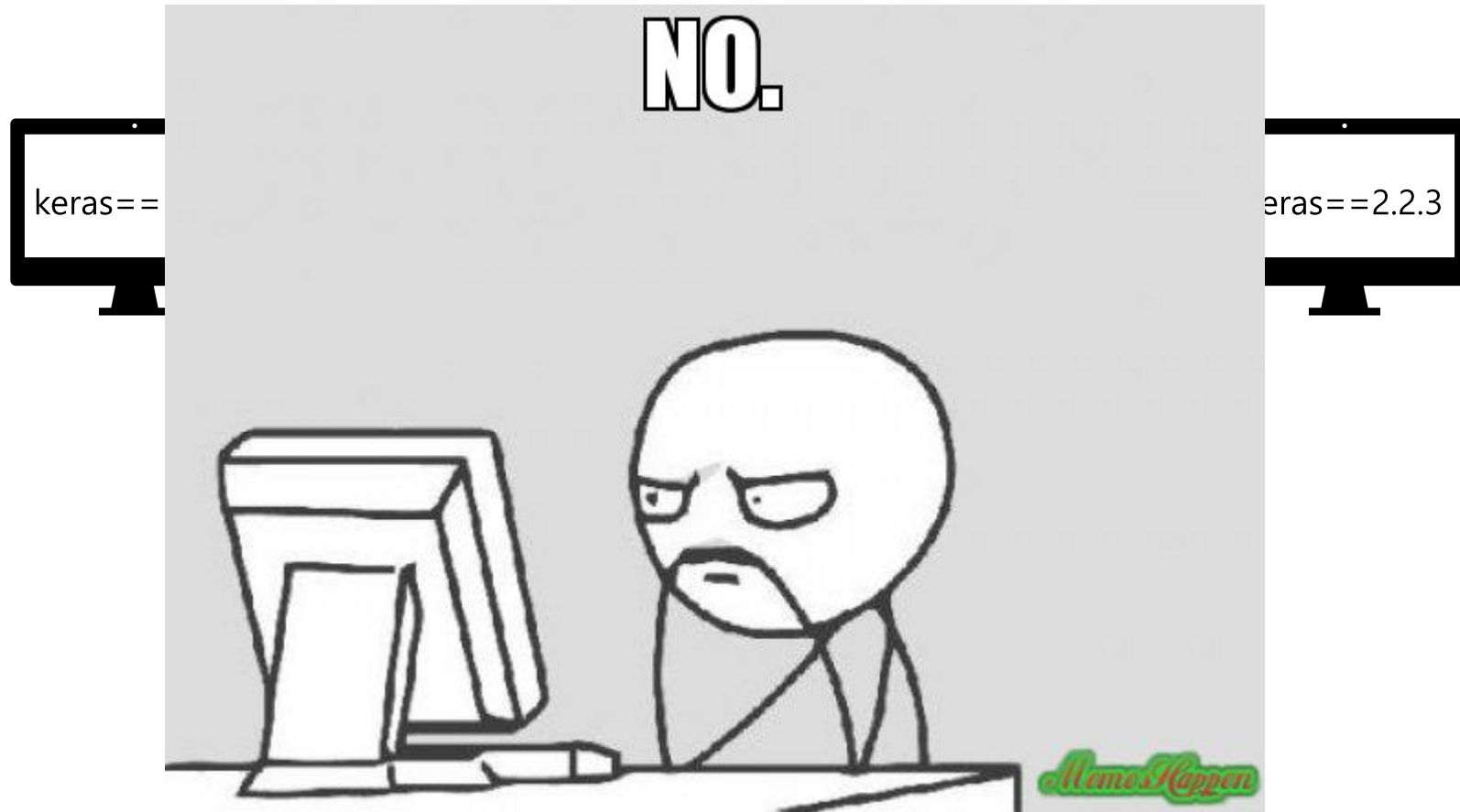
여전히 문제

- 패키지 & 라이브러리 버전 관리



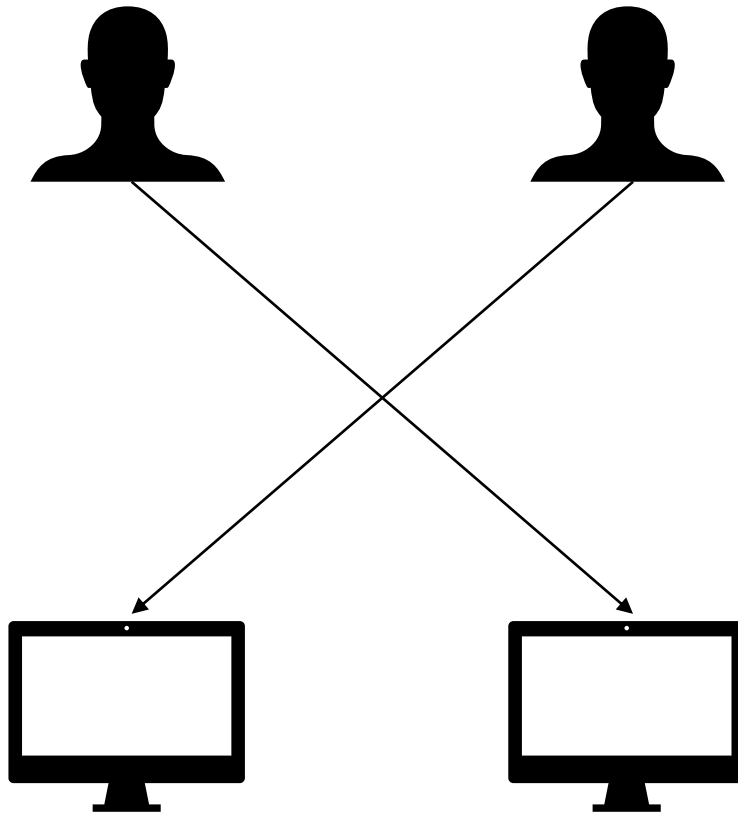
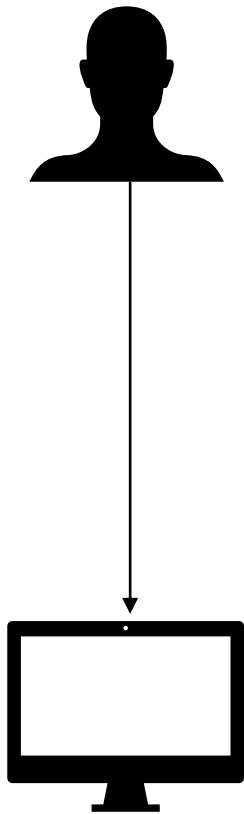
여전히 문제

- 패키지 업데이트 하는 날엔...



근본적인 문제

- 직접 관리



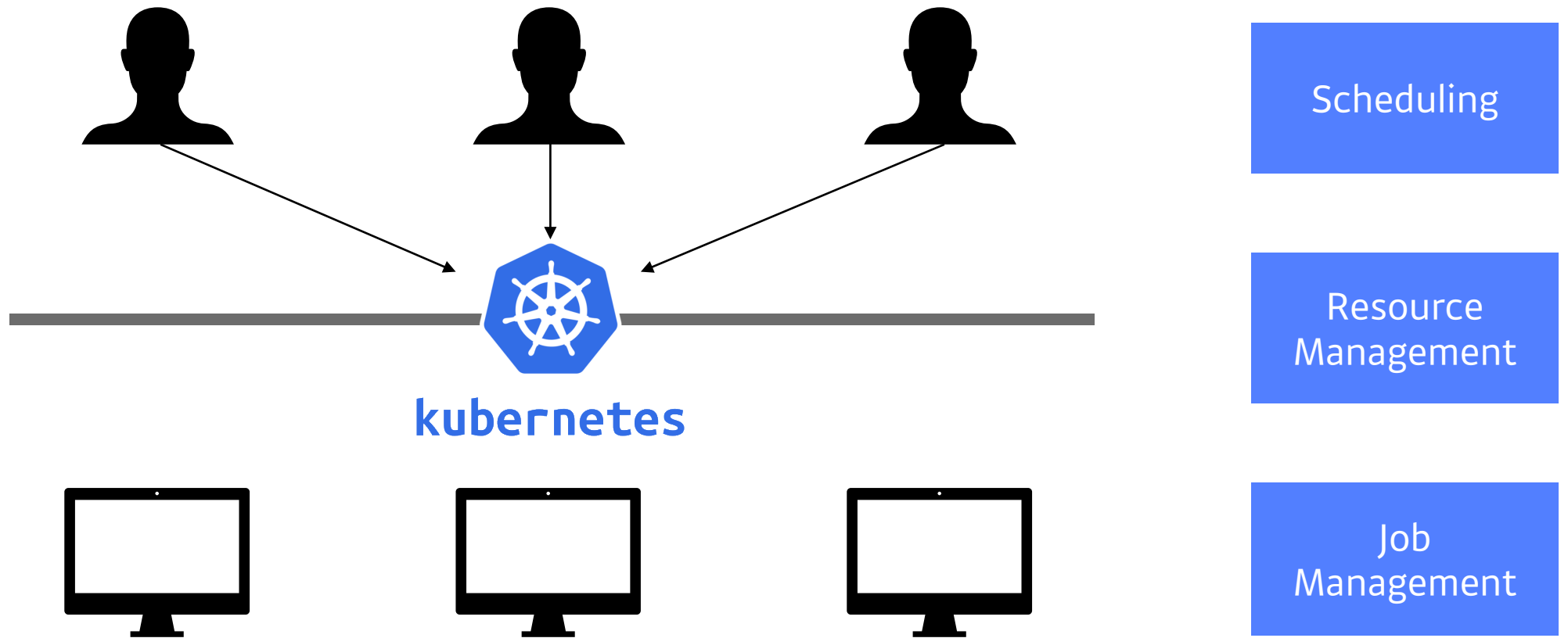
Scheduling

Resource
Management

Job
Management

제안 방법

- 위임 관리

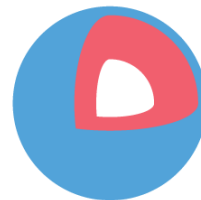


Kubernetes란?

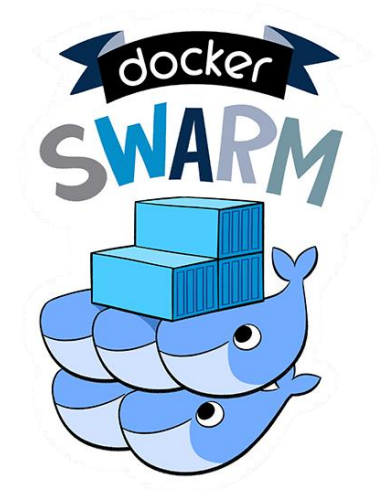
- 도커 Orchestration 툴 중에 하나?



kubernetes



CoreOS

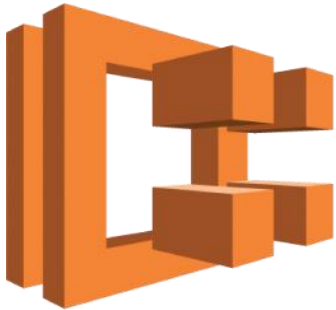


Kubernetes란?

- 도커 Orchestration 툴 중에 하나?



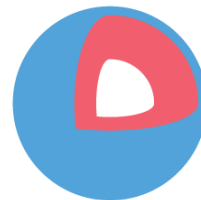
MESOS



Amazon ECS



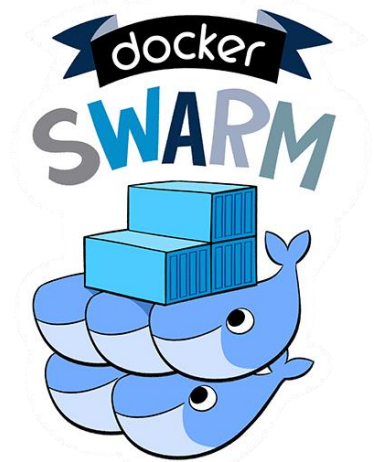
kubernetes



CoreOS



RANCHEROS



Kubernetes란?

- 조금 다르게 정의해 봅시다.
→ 쿠버네티스란 Cluster OS 입니다.



kubernetes

Kubernetes란?

- 응...?

꾸웅?



kubern

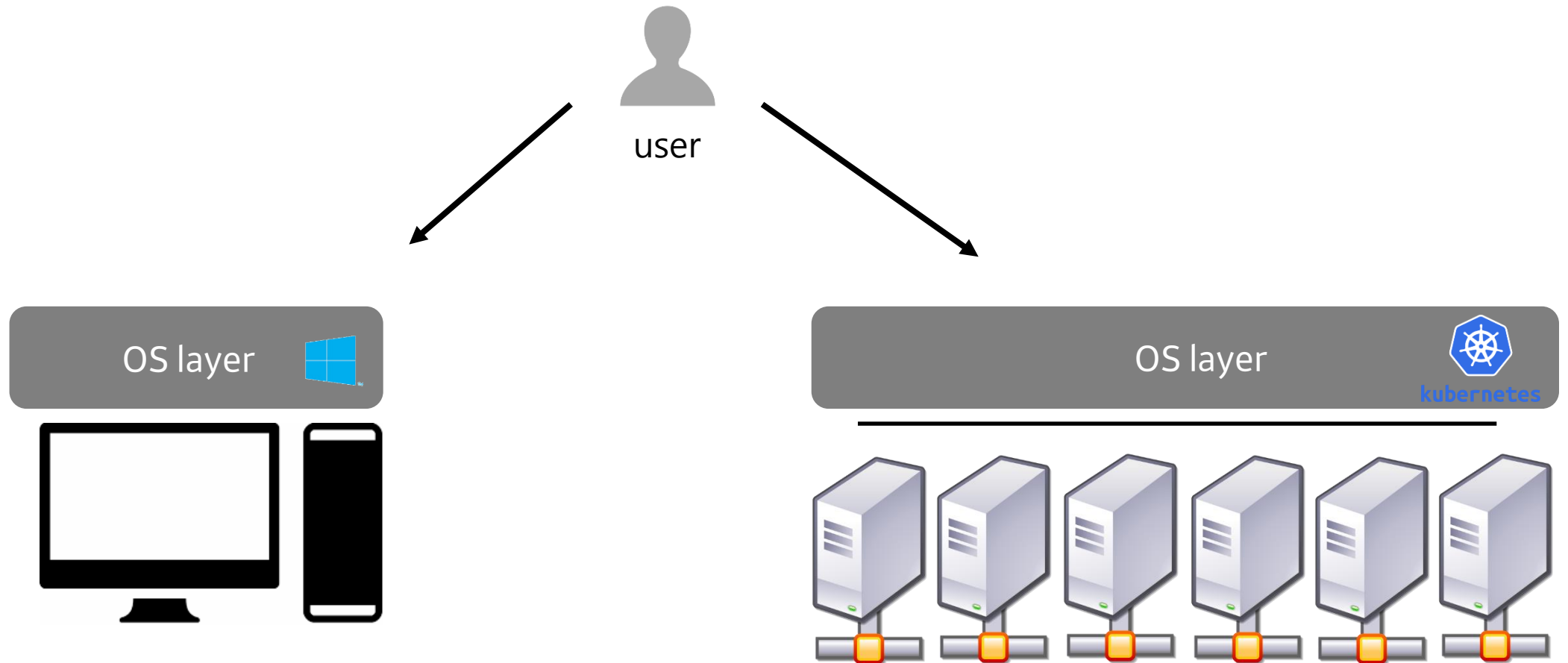


Kubernetes란?

- Cluster
 - Collection of computers to compute large job.
- OS
 - Hardware 추상화 layer (CPU, memory, HDD)
 - 자원 분배 (resource management)
 - 프로세스 스케줄링 (scheduling)
 - User Interface (GUI, CUI)

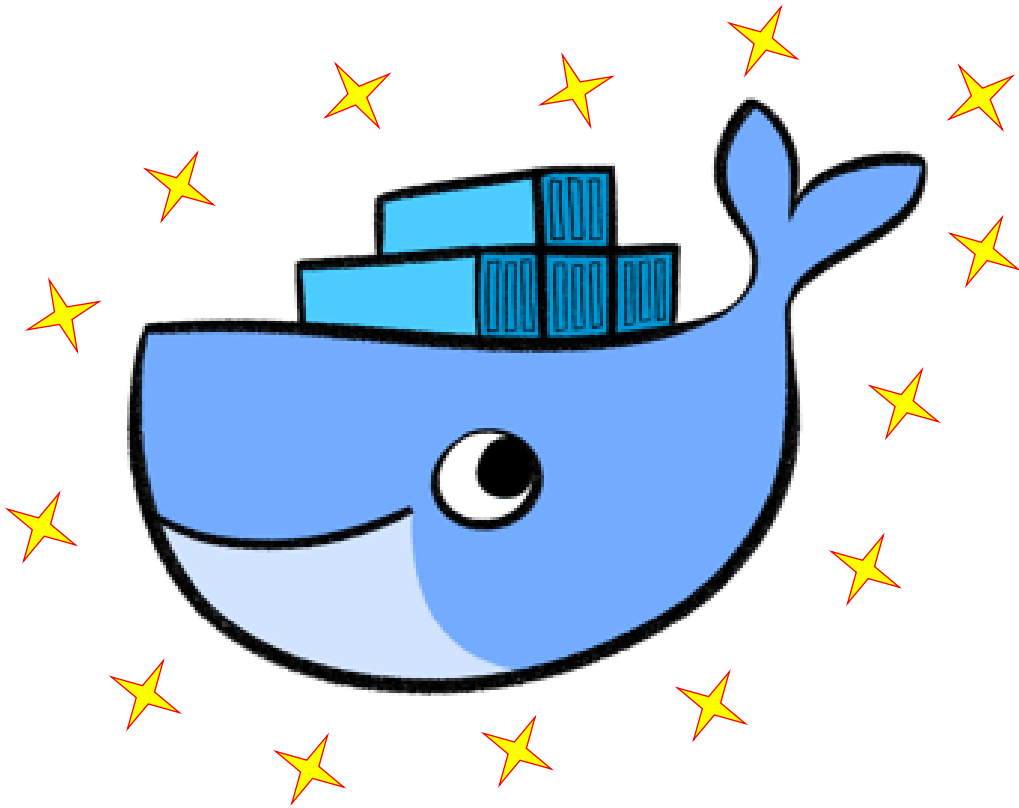
Kubernetes란?

- Kubernetes as Cluster OS



Kubernetes란?

- Docker orchestration tool이라 부르기 어려운 또 다른 이유



kubernetes

저도 도커
orchestration 툴이에요~
데헛ㅎ

Kubernetes란?

- Docker orchestration tool이라 부르기 어려운 또 다른 이유

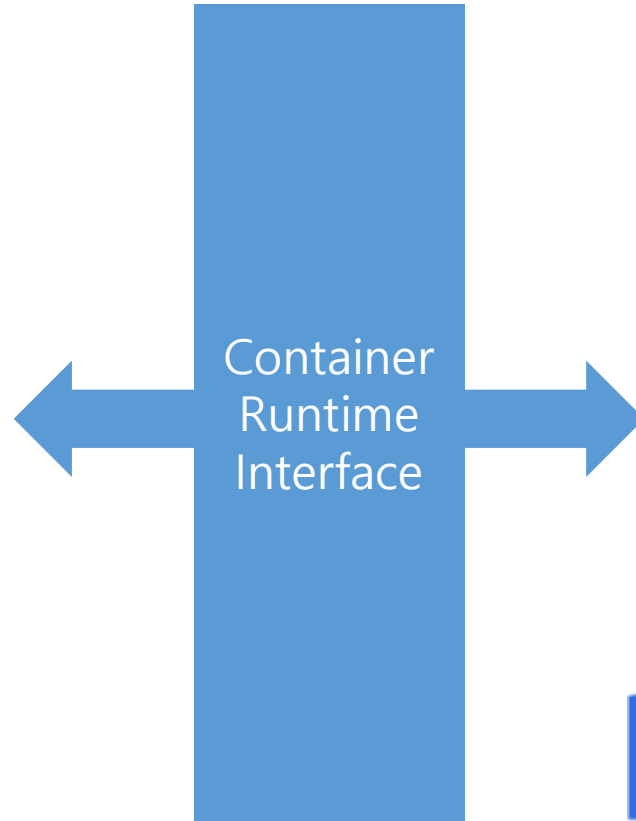
containerd



cri-o



끼양ㅏ

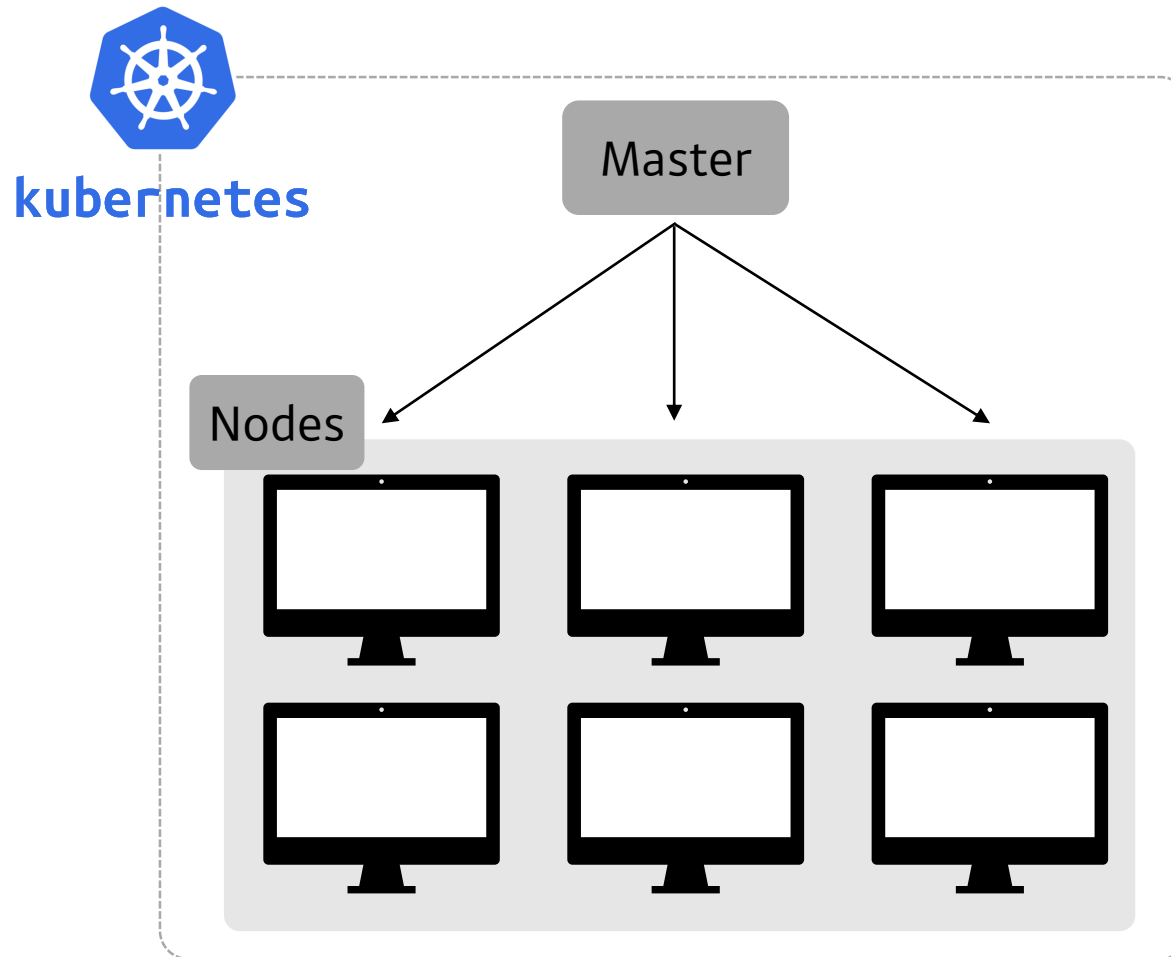


kubernetes

짐은 이제
CRI를 통해서만 대화
하겠노라

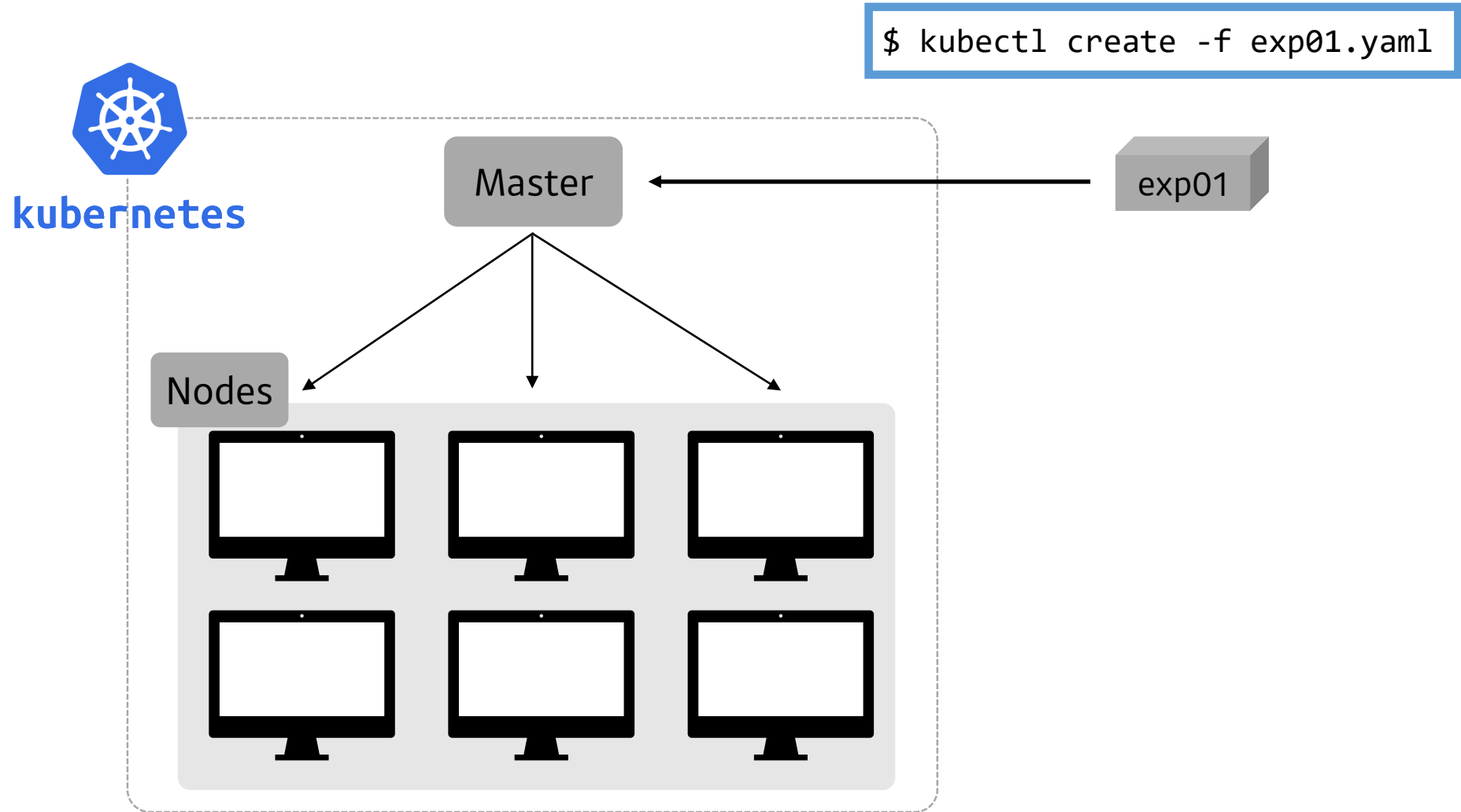
Kubernetes 소개

- Master / Node



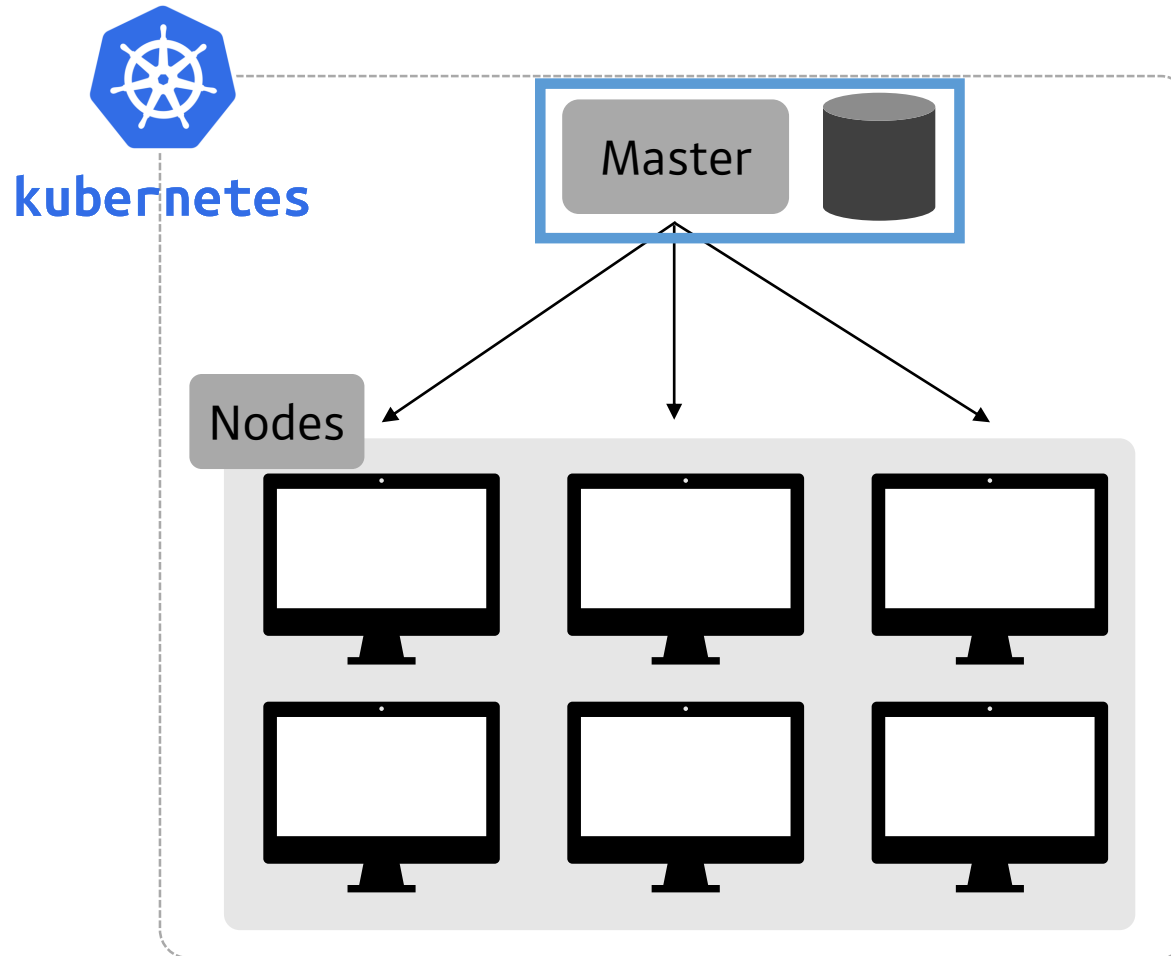
Kubernetes 소개

- kubectl 툴



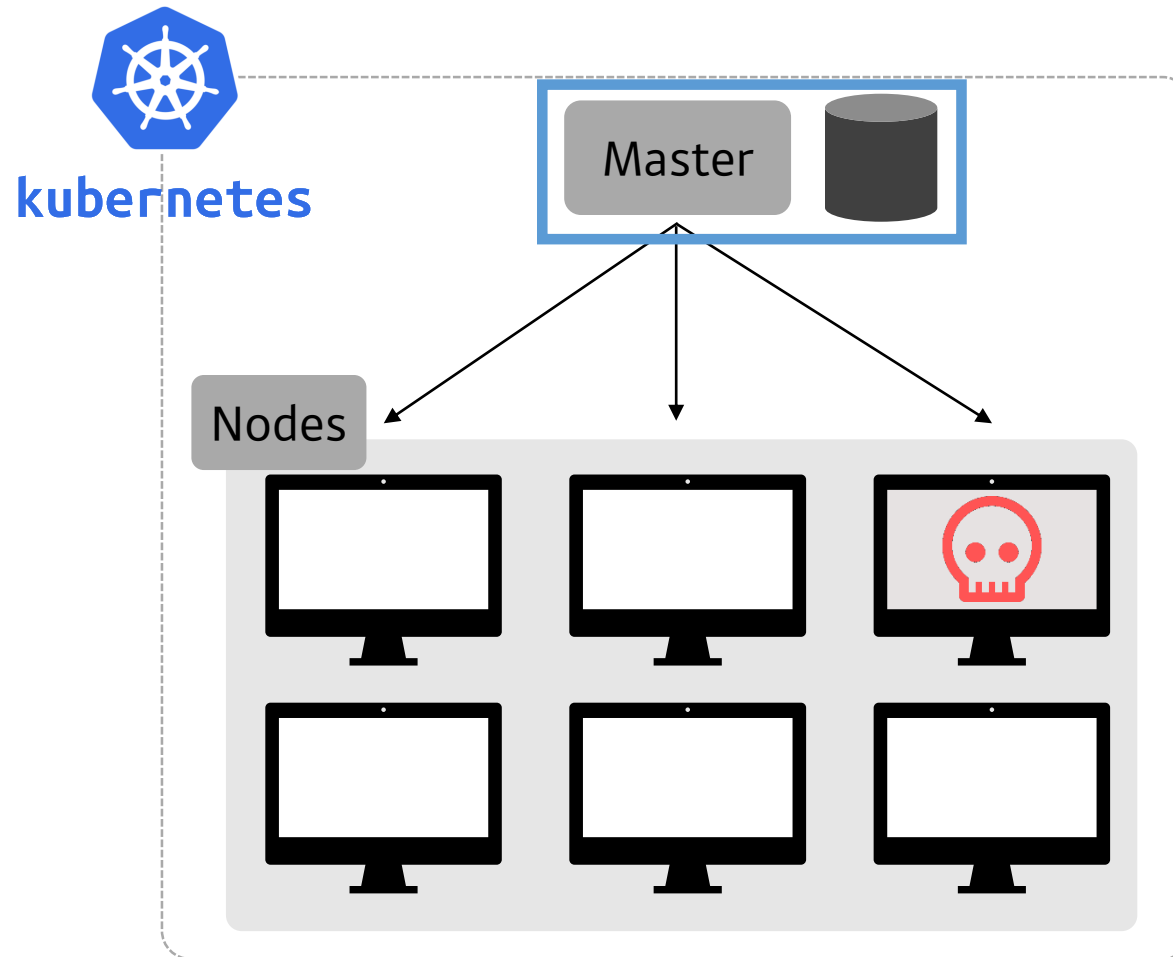
Kubernetes 소개

- 내부 DB



Kubernetes 소개

- 내부 DB



Kubernetes 소개

- 모든 것이 resource

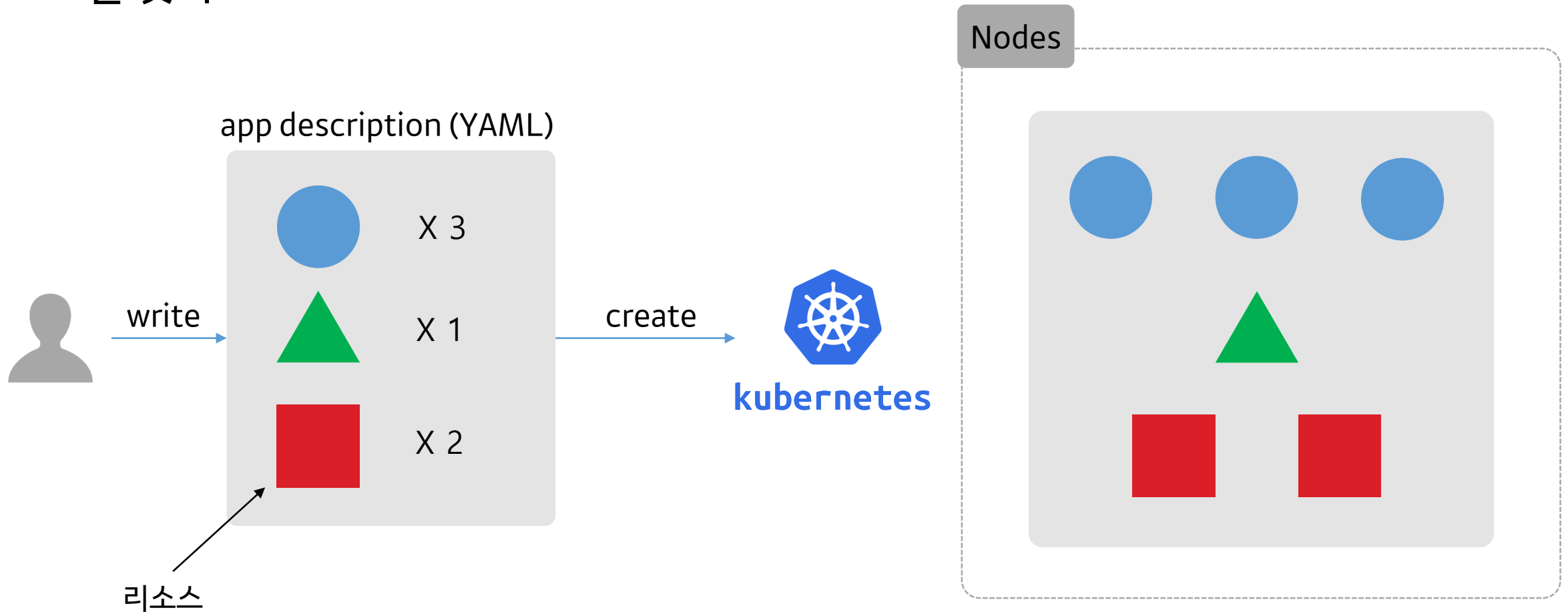
Job

Deployment

Service

Kubernetes 소개

- 모든 것이 resource



Kubernetes 소개

- 모든 것이 resource

Job: 1회성 배치 작업을 실행할 때 사용하는 리소스

Deployment: 데몬처럼 계속 실행되는 리소스

Service: 사용자 request를 받을 수 있는 Endpoint

Kubernetes 소개

```
apiVersion: batch/v1
kind: Job
metadata:
  name: exp01
spec:
  template:
    spec:
      containers:
      - name: ml
        image: $model_image
        command: ["python", "train.py"]
        args: ['2', 'softmax', '0.5']
        resources:
          requests:
            cpu: "2"
            memory: "8Gi"
          limits:
            cpu: "4"
            memory: "16Gi"
      restartPolicy: OnFailure
```



exp01.yaml

Kubernetes 소개

```
apiVersion: batch/v1
kind: Job
metadata:
  name: exp01
spec:
  template:
    spec:
      containers:
      - name: ml
        image: $model_image
        command: ["python", "train.py"]
        args: ['2', 'softmax', '0.5']
        resources:
          requests:
            cpu: "2"
            memory: "8Gi"
          limits:
            cpu: "4"
            memory: "16Gi"
      restartPolicy: OnFailure
```

exp01이라는 Job을

Kubernetes 소개

```
apiVersion: batch/v1
kind: Job
metadata:
  name: exp01
spec:
  template:
    spec:
      containers:
      - name: ml
        image: $model_image
        command: ["python", "train.py"]
        args: ['2', 'softmax', '0.5']
        resources:
          requests:
            cpu: "2"
            memory: "8Gi"
          limits:
            cpu: "4"
            memory: "16Gi"
      restartPolicy: OnFailure
```

이렇게 실행하라
python train.py 2 softmax 0.5

Kubernetes 소개

```
apiVersion: batch/v1
kind: Job
metadata:
  name: exp01
spec:
  template:
    spec:
      containers:
      - name: ml
        image: $model_image
        command: ["python", "train.py"]
        args: ['2', 'softmax', '0.5']
        resources:
          requests:
            cpu: "2"
            memory: "8Gi"
          limits:
            cpu: "4"
            memory: "16Gi"
      restartPolicy: OnFailure
```

이러한 자원을 가지고

Kubernetes 소개

```
apiVersion: batch/v1
kind: Job
metadata:
  name: exp01
spec:
  template:
    spec:
      containers:
      - name: ml
        image: $model_image
        command: ["python", "train.py"]
        args: ['2', 'softmax', '0.5']
        resources:
          requests:
            cpu: "2"
            memory: "8Gi"
          limits:
            cpu: "4"
            memory: "16Gi"
      restartPolicy: OnFailure
```

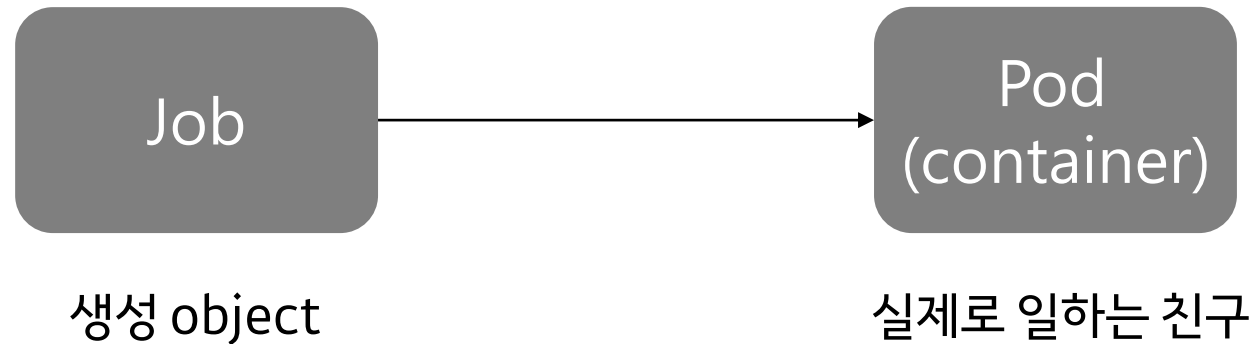
이름 정보

실행 정보

자원 정보

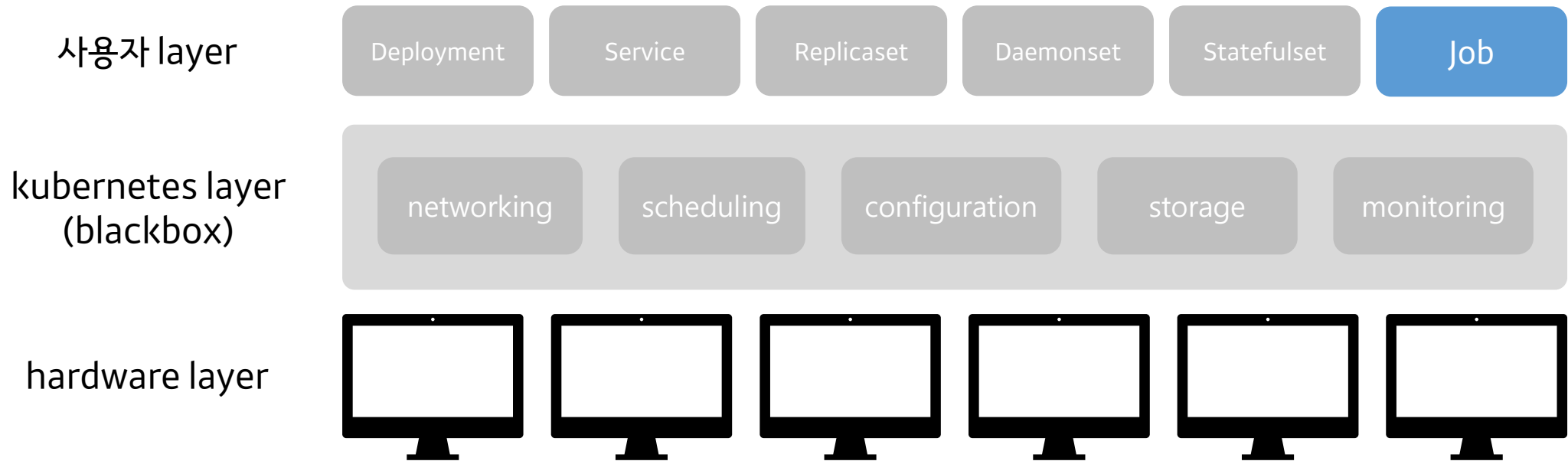
Kubernetes 소개

- `kubectl create -f exp01.yaml`



Kubernetes 소개

- 지금까지 알아본 것
 - Job 리소스 하나만 알아도



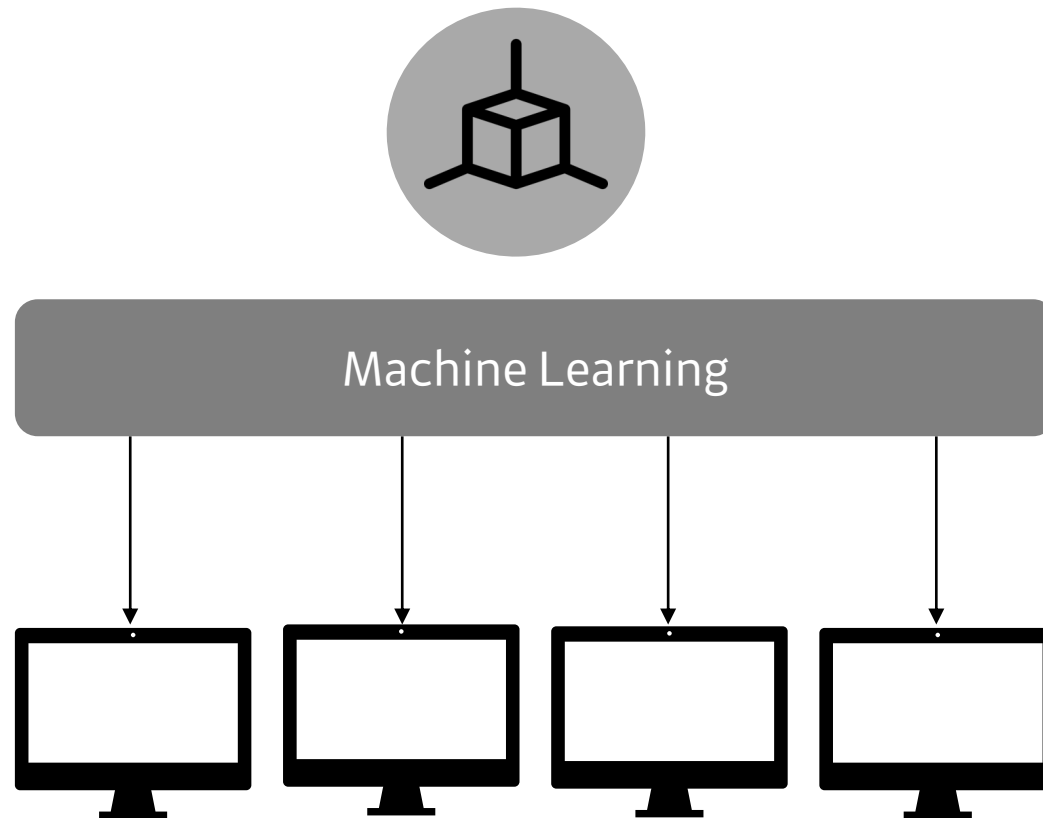
Kubernetes 소개

- 이제 여러분은 분산 기계학습 엔지니어



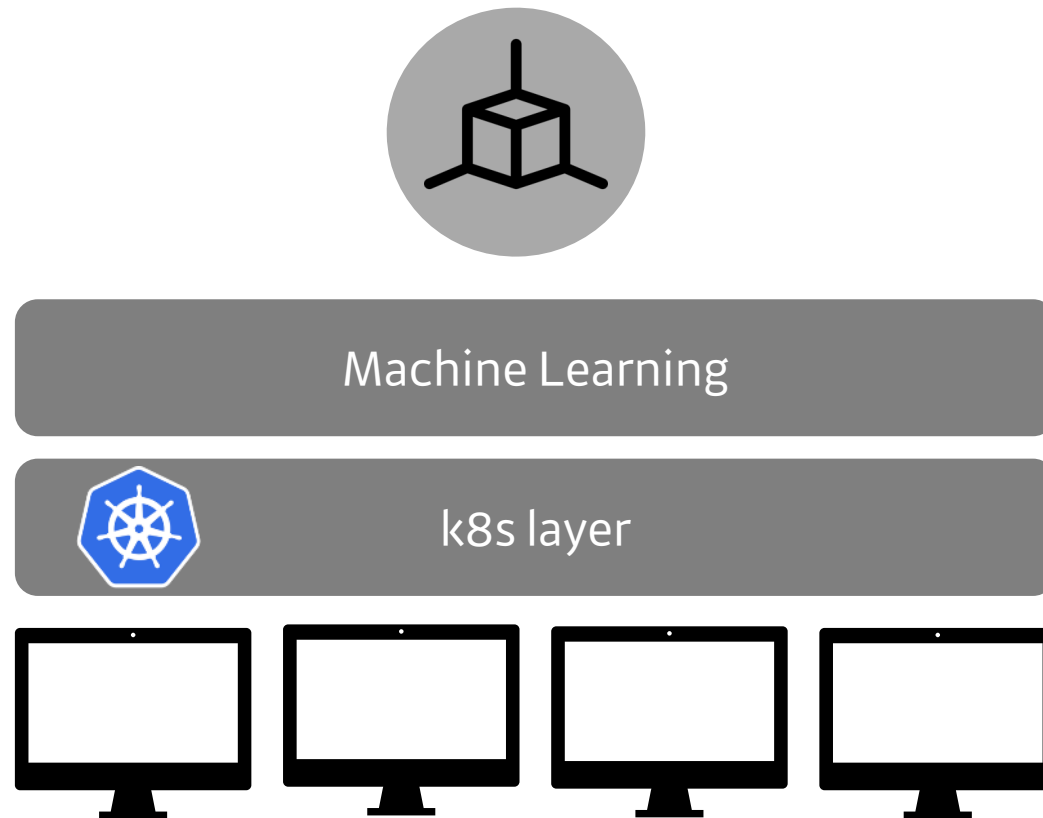
ML with k8s

- 어떤 장점들이 있을까요?



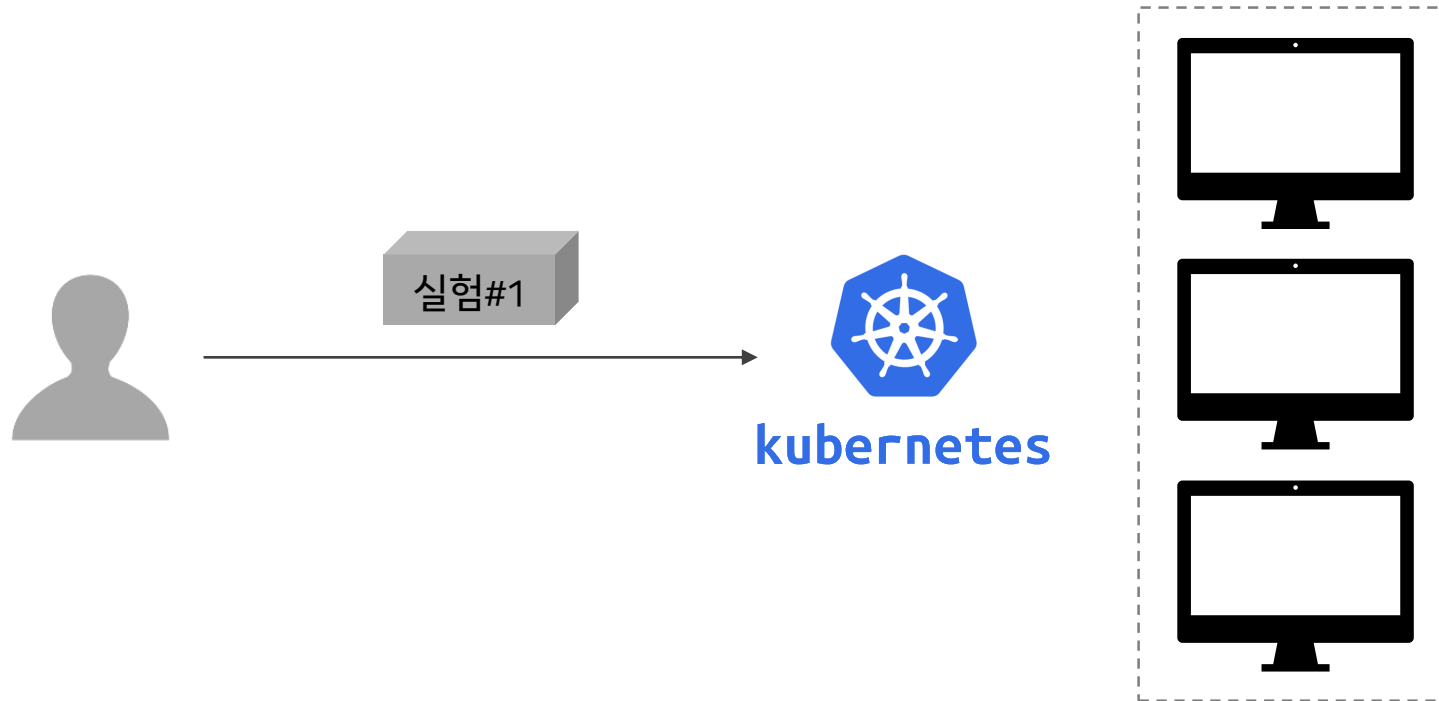
ML with k8s

- 어떤 장점들이 있을까요?



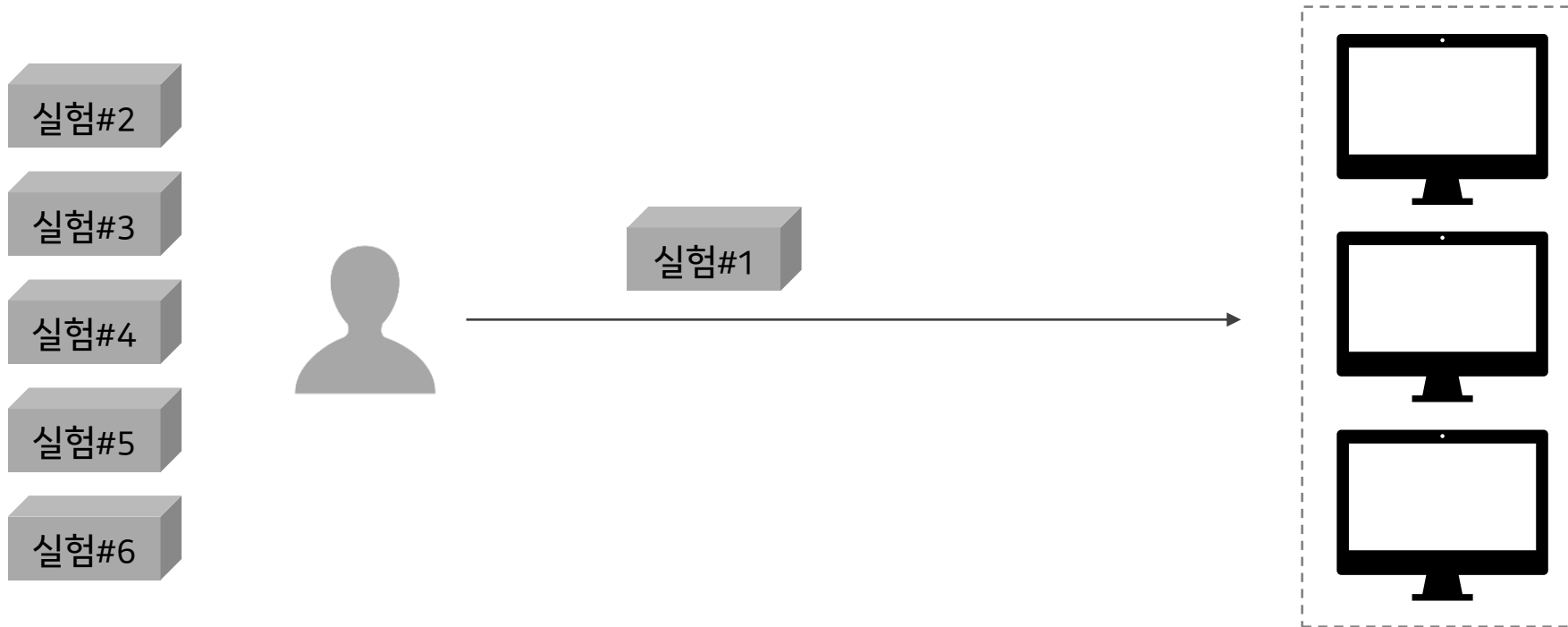
ML with k8s

- 스케줄링이 편리해 집니다.



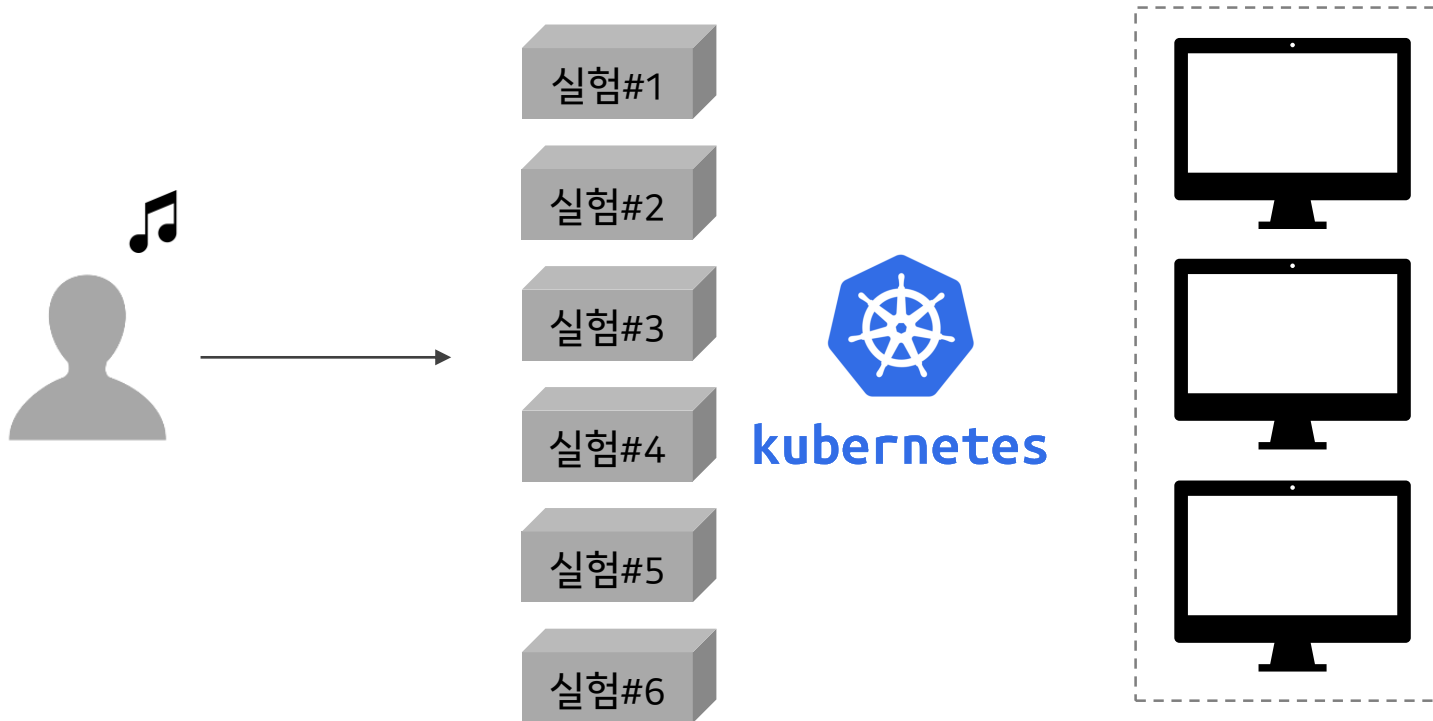
ML with k8s

- 스케줄링이 편리해 집니다.
 - 사람이 직접 스케줄링하던 기계학습 훈련을



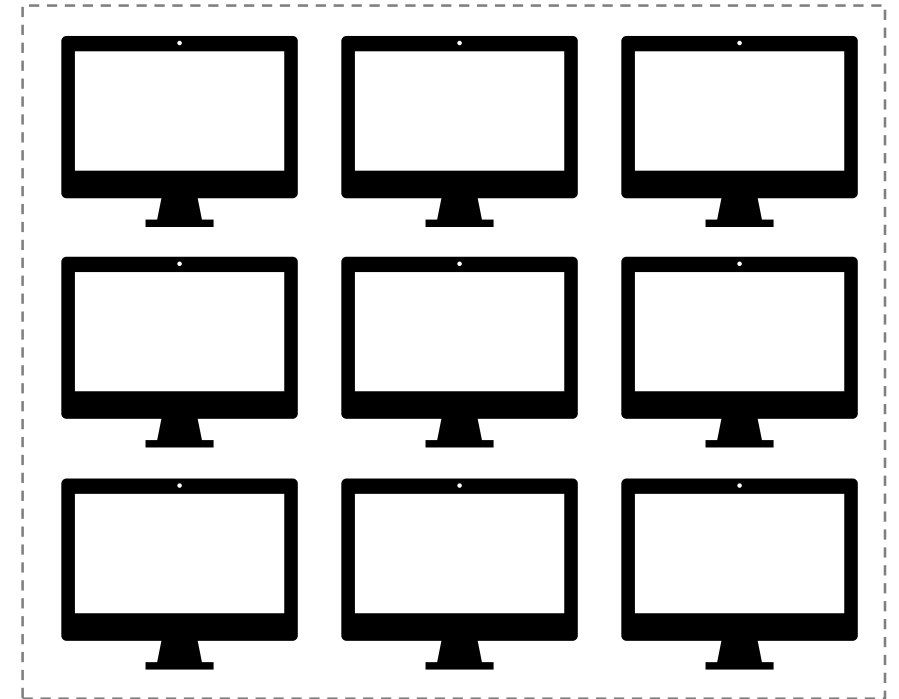
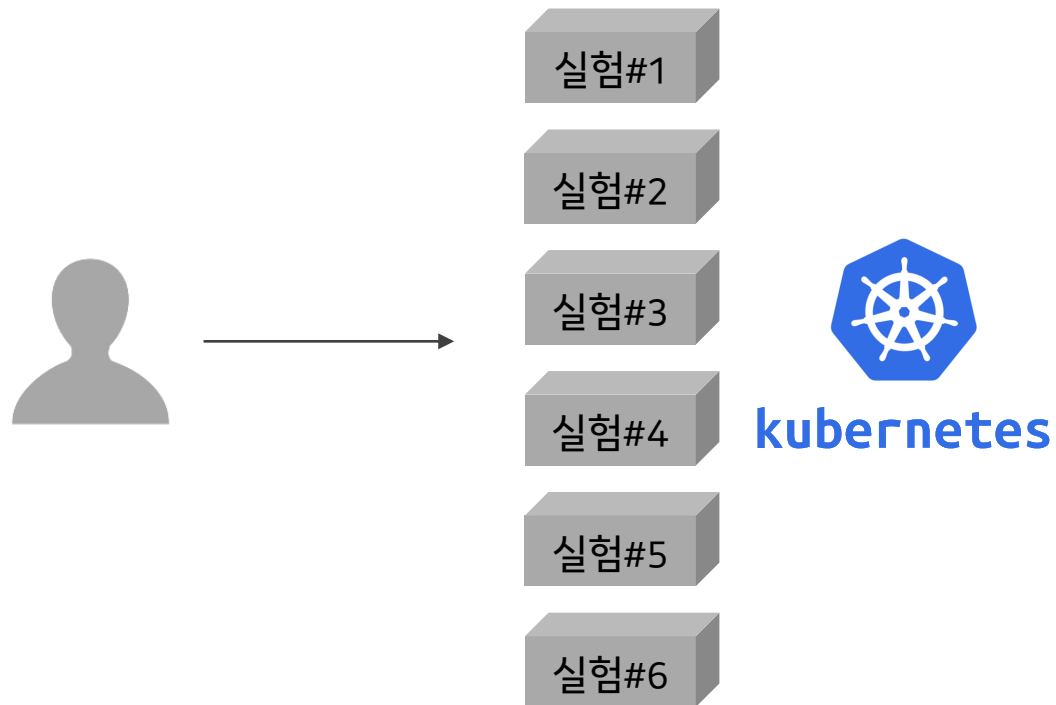
ML with k8s

- 스케줄링이 편리해 집니다.
 - 쿠버네티스에게 맡깁니다.



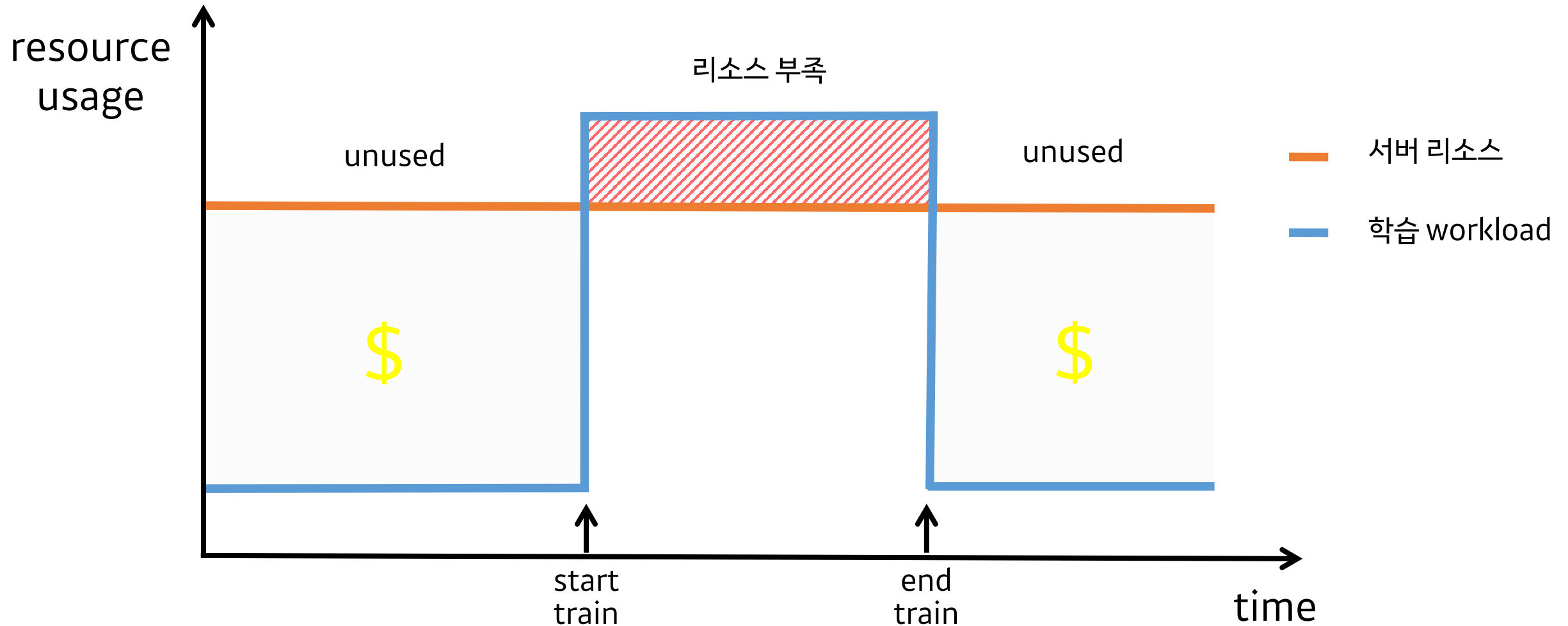
ML with k8s

- 스케줄링이 편리해 집니다.
 - 학습 서버가 증가할수록 그 효과는 커집니다.



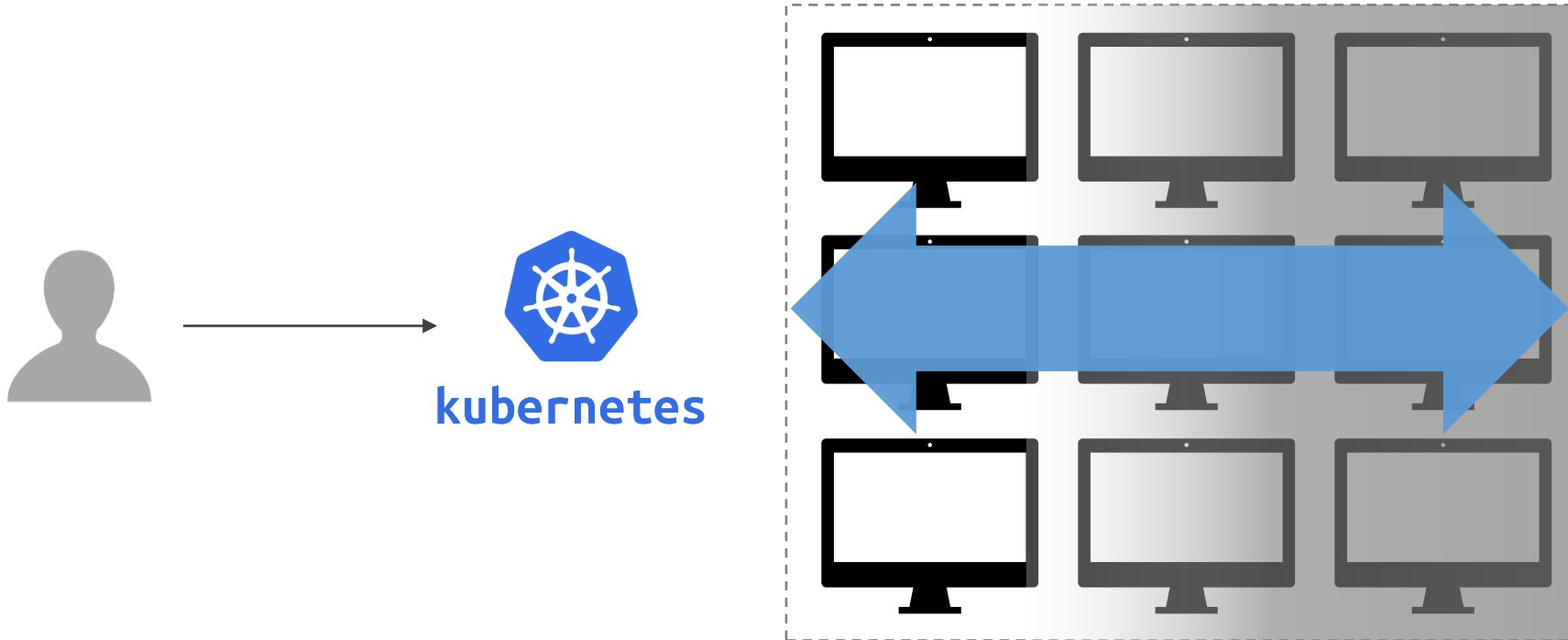
ML with k8s

- 확장이 용이해 집니다.



ML with k8s

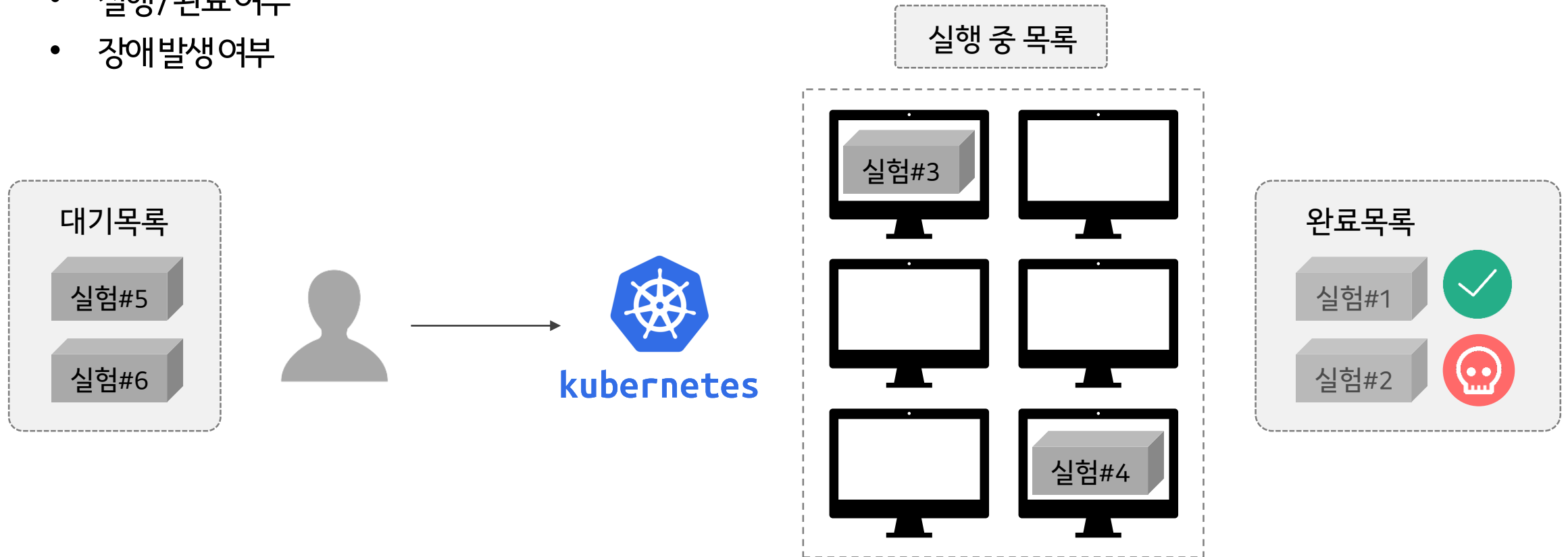
- 확장이 용이해 집니다.



ML with k8s

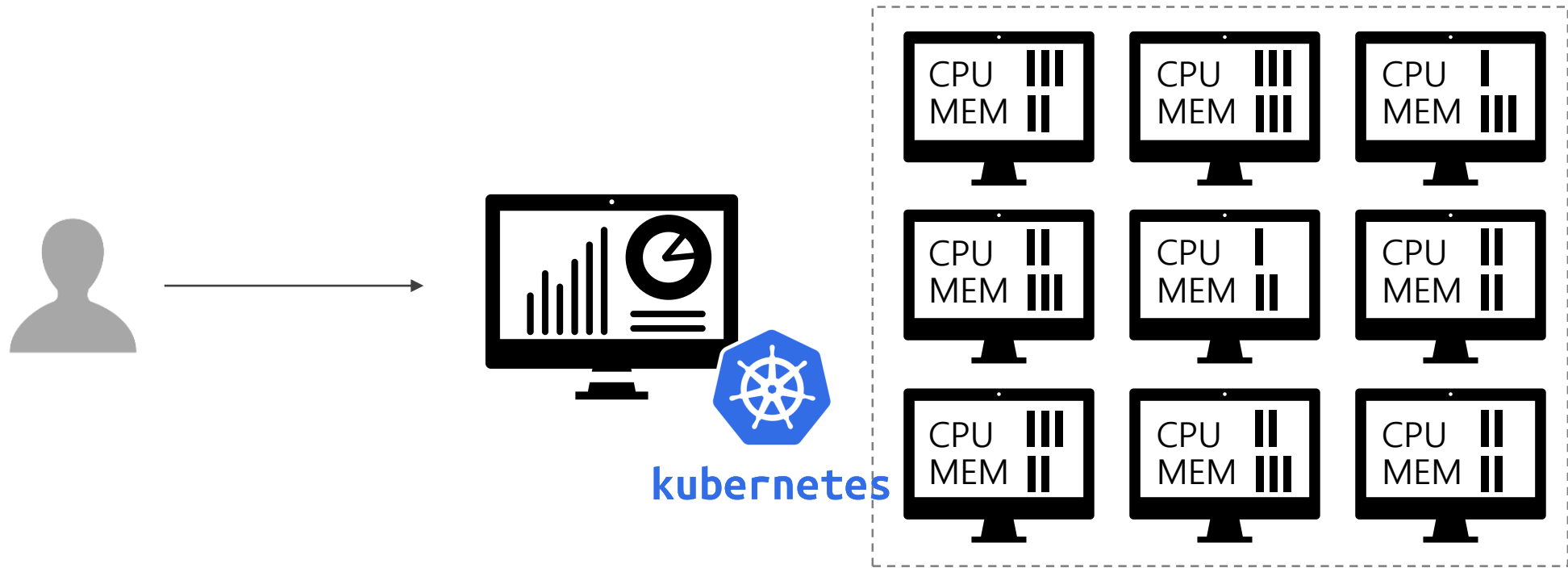
- Job 관리가 편리해 집니다.

- 실행/완료 여부
- 장애 발생 여부



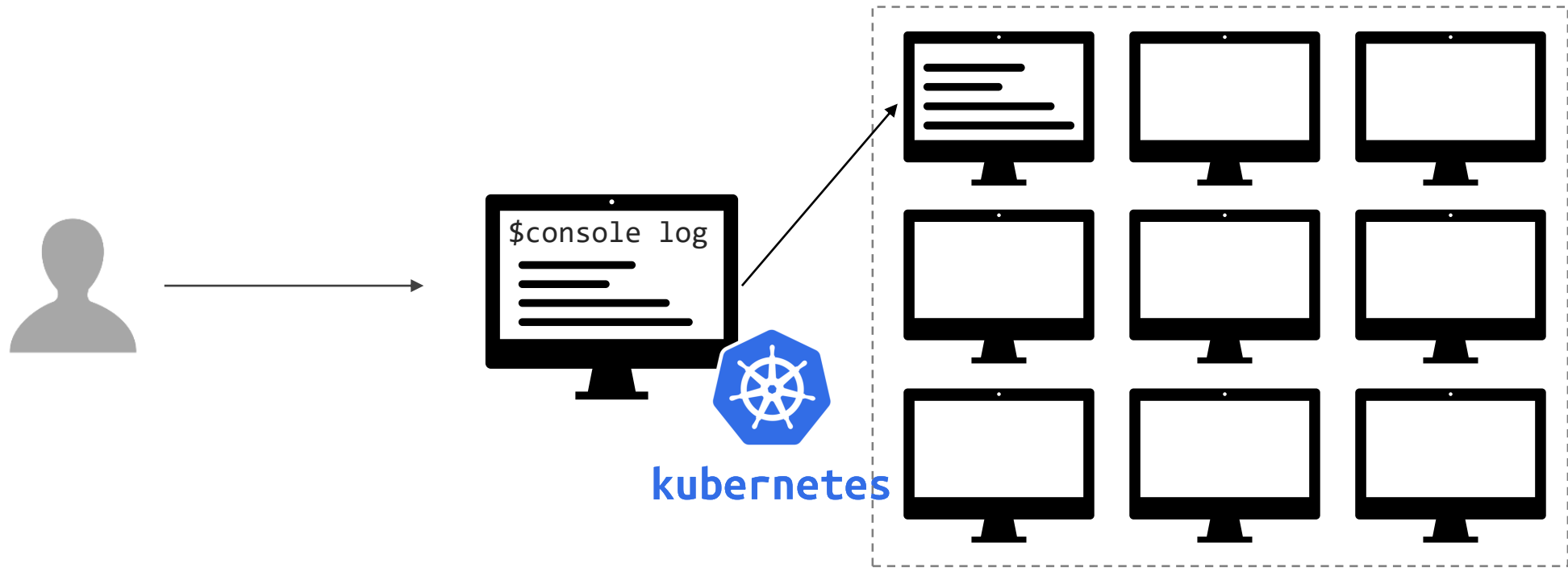
ML with k8s

- 모니터링이 편리해 집니다.
 - 리소스 사용량



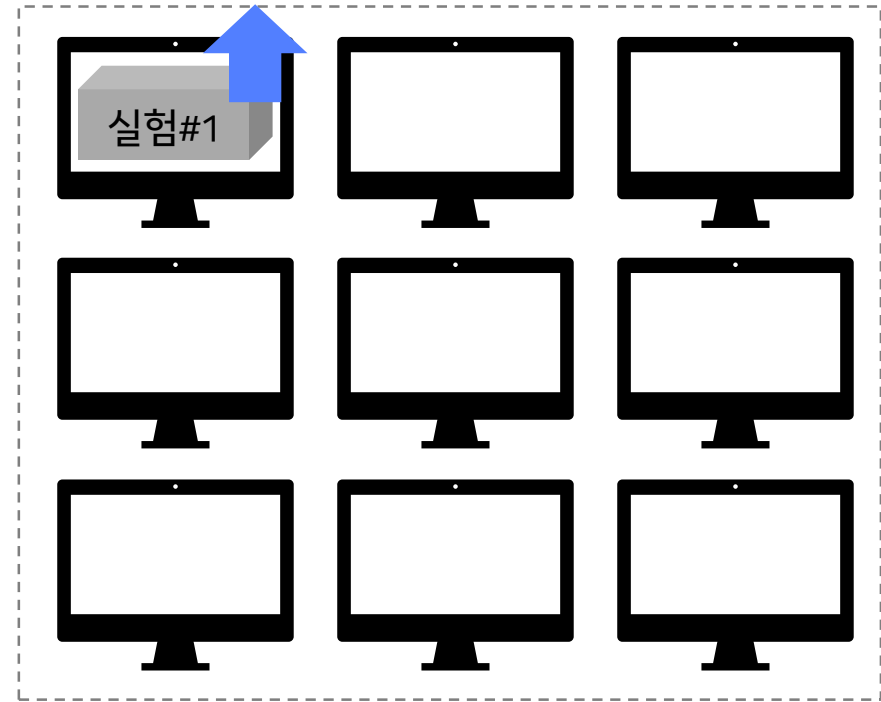
ML with k8s

- 모니터링이 편리해 집니다.
 - 기계학습 로그



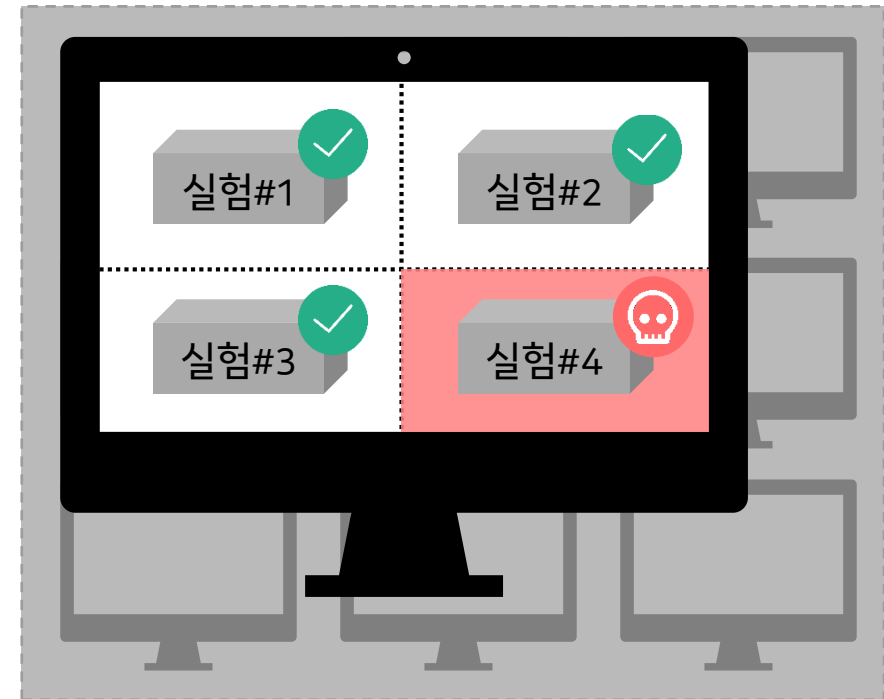
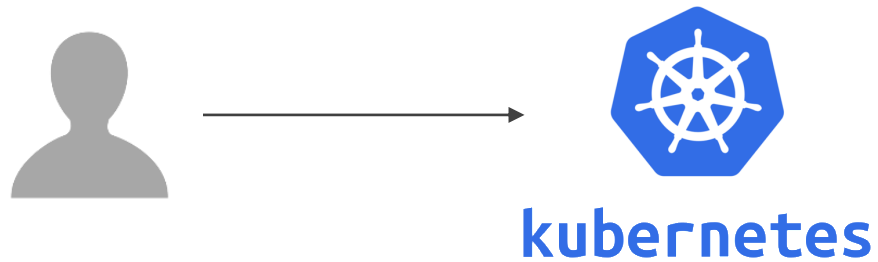
ML with k8s

- 배포가 쉬워집니다.
 - 라이브러리 관리
 - 소스코드 관리



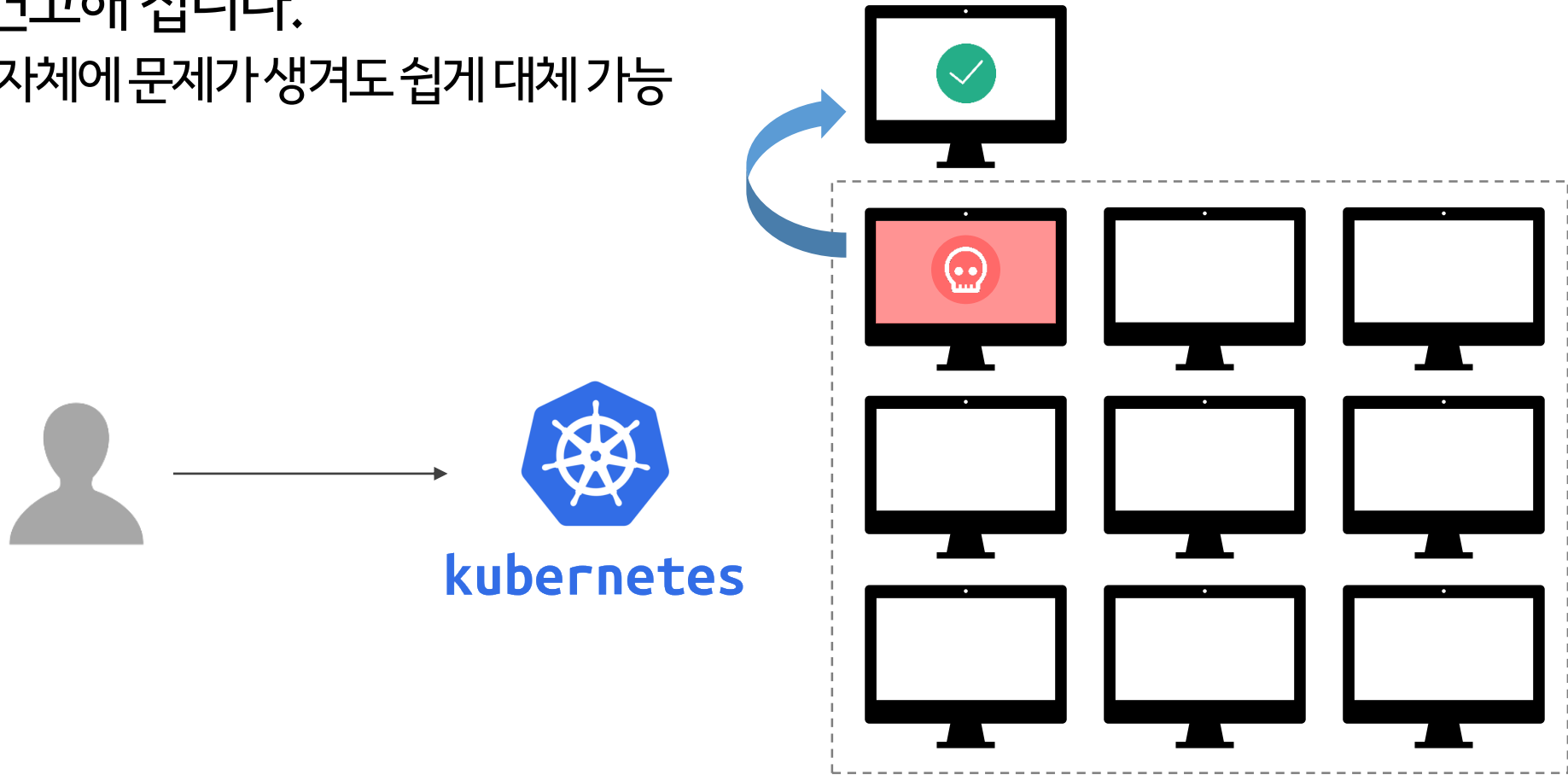
ML with k8s

- 장애에 견고해 집니다.
 - 문제가 되는 한개의 Job만 장애 발생



ML with k8s

- 장애에 견고해 집니다.
 - 서버 자체에 문제가 생겨도 쉽게 대체 가능



ML with k8s

- 이 모든 것이 쿠버네티스 Job만 잘 활용해도 가능합니다!

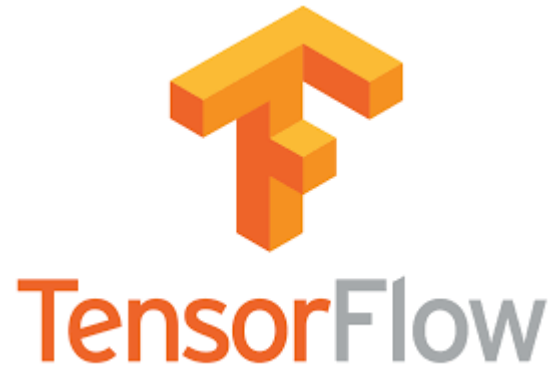


마무리 전

- kubeflow 소개



+



=



마무리 전



Kubeflow



EDA: Jupyter



Train: tf-job



Deploy: tf-serving

<https://www.kubeflow.org/>

마무리 전

- 2014년 Netflix 블로그: 분산 딥러닝 framework 소개



Distributed Neural Networks with GPUs in the AWS Cloud



Netflix Technology Blog [Follow](#)

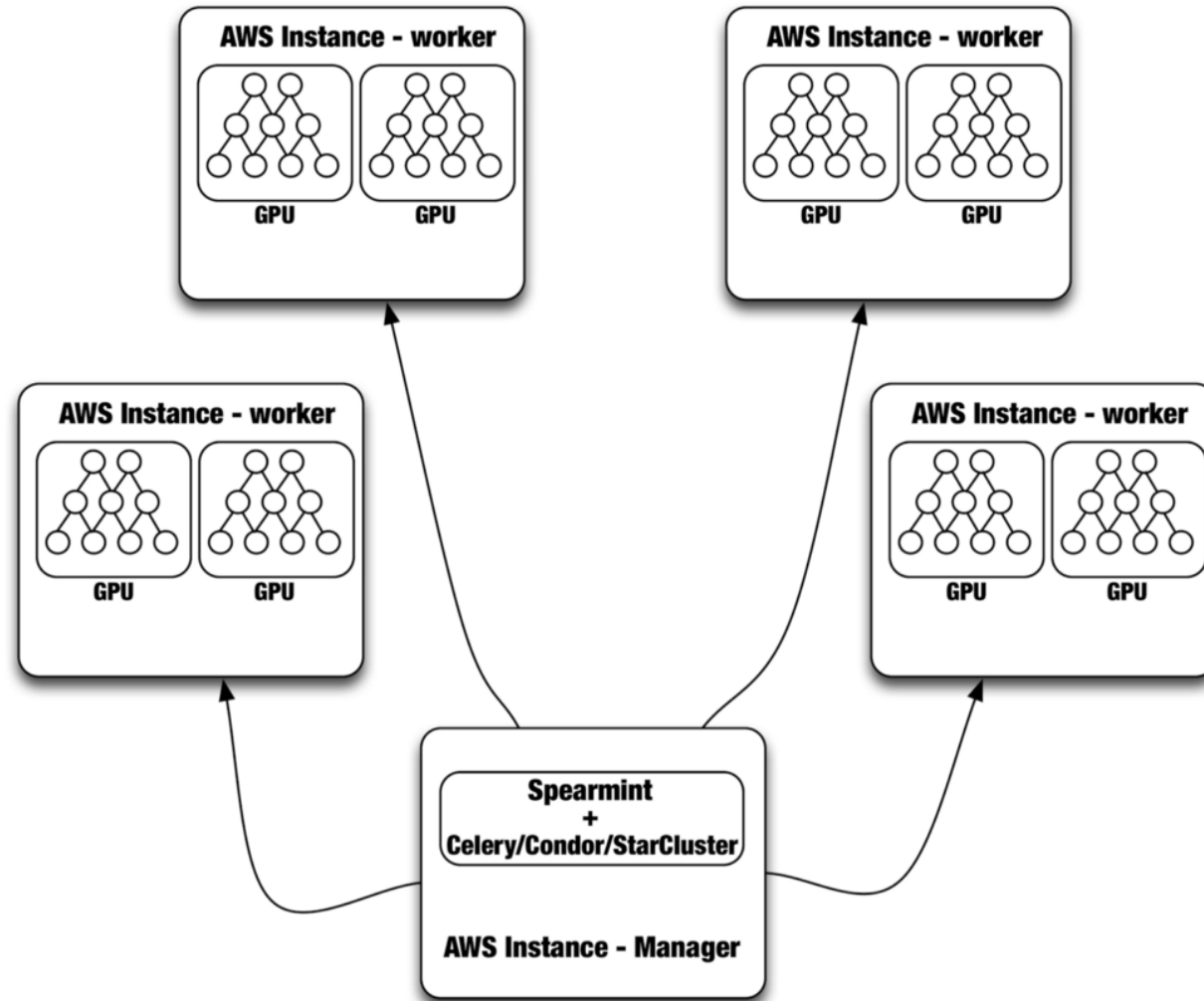
Feb 10, 2014 · 11 min read

by Alex Chen, Justin Basilico, and Xavier Amatriain

As we have described previously on this blog, at Netflix we are constantly innovating by looking for better ways to find the best movies and TV shows for our members. When a new algorithmic technique such as Deep Learning shows promising results in other domains (e.g. Image Recognition, Neuro-

<https://medium.com/netflix-techblog/distributed-neural-networks-with-gpus-in-the-aws-cloud-ccf71e82056b>

마무리 전



AWS EC2 + MIT StarCluster + Celery (distributed Job Queue)

마무리

- 오늘 저와 함께 쿠버네티스를 통해 기계학습 모델들을 쉽게 확장해 봅시다!



kubernetes

