

2017
China Kubernetes
End User Conference
kubernetes 中国用户大会 — 2017 —

2017.10.15 / 中国·杭州

七牛AI训练业务的K8S实践

七牛容器云 - 陆龙文

主办：  CLOUD NATIVE
COMPUTING FOUNDATION


caicloud
才云


K8S
Meetup
中国社区

AI训练的业务情况

kubernetes 的优势

基于 kubernetes 的AI训练

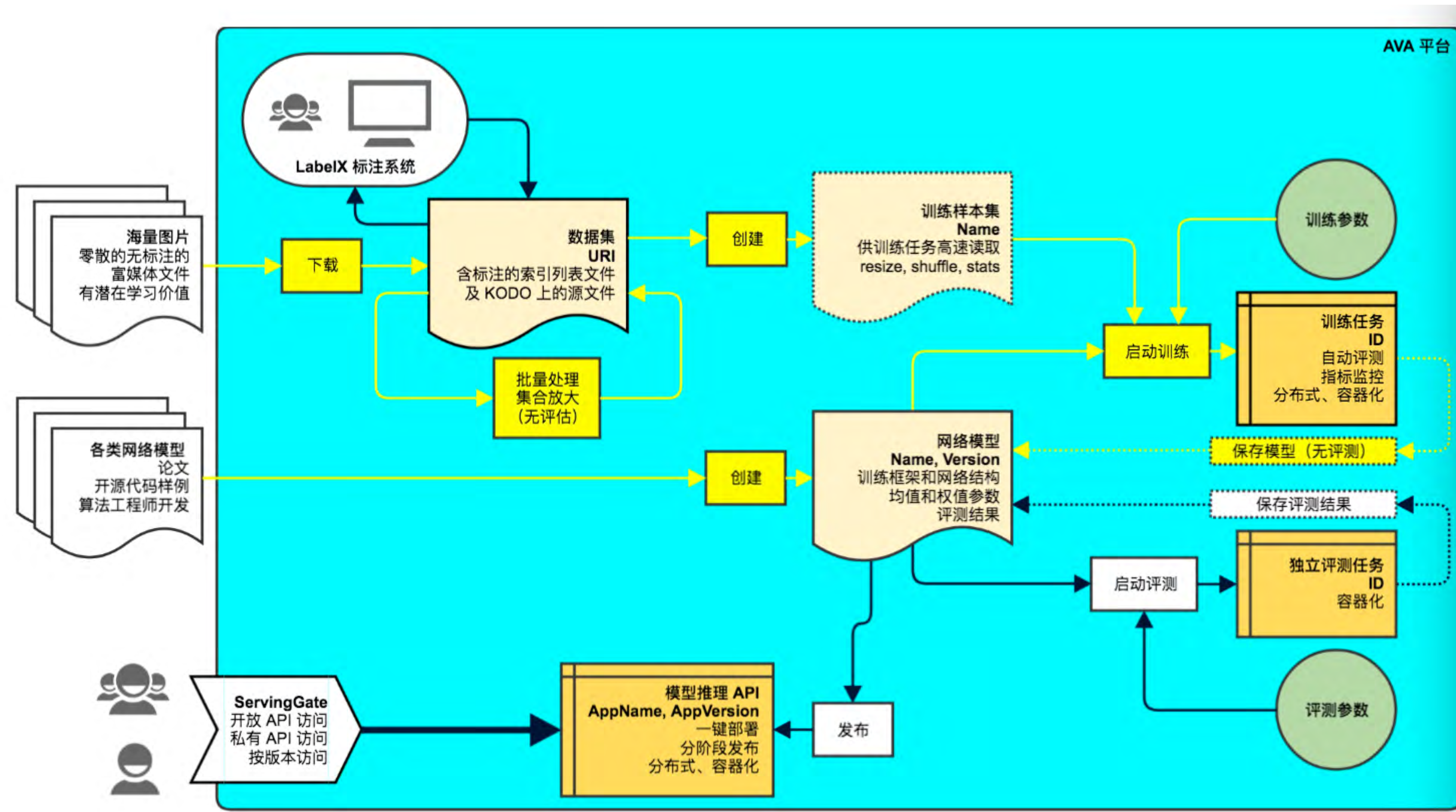
一次踩坑经历

接下来的工作

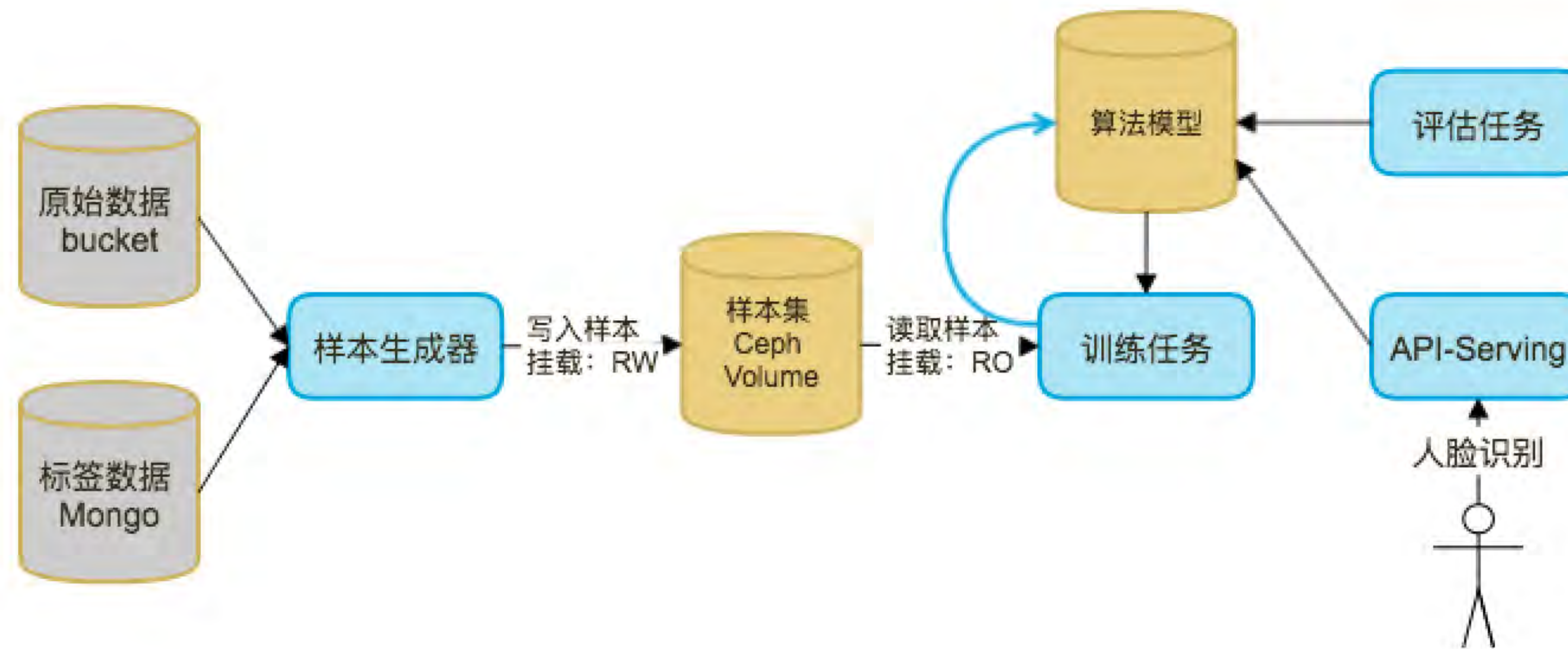
主办：



AI训练流程



AI训练迭代



AI训练的业务情况

kubernetes 的优势

基于 kubernetes 的AI训练

一次踩坑经历

接下来的工作

主办：



AP训练的痛点-流程

训练流程

- 算法工程师通过脚本控制，管理困难
- 失败重试需要人工介入

GPU 资源管理

- 资源多人共用，协调耗费精力
- 资源释放不及时，利用率低

存储

- IOPS要求高，突发读写，目标数据大
- NFS服务单点，扩展性差，性能无法满足需求

kubernetes 的优势

GPU 支持

- kubernetes 初步支持 GPU 资源调度，并且仍在快速完善中

Job 调度

- Job 类型的任务调度方式与训练任务场景相性相合

日志监控

- kubernetes 与 Prometheus、Elastic Search 配合良好

AI训练的业务情况

kubernetes 的优势

基于 kubernetes 的AI训练

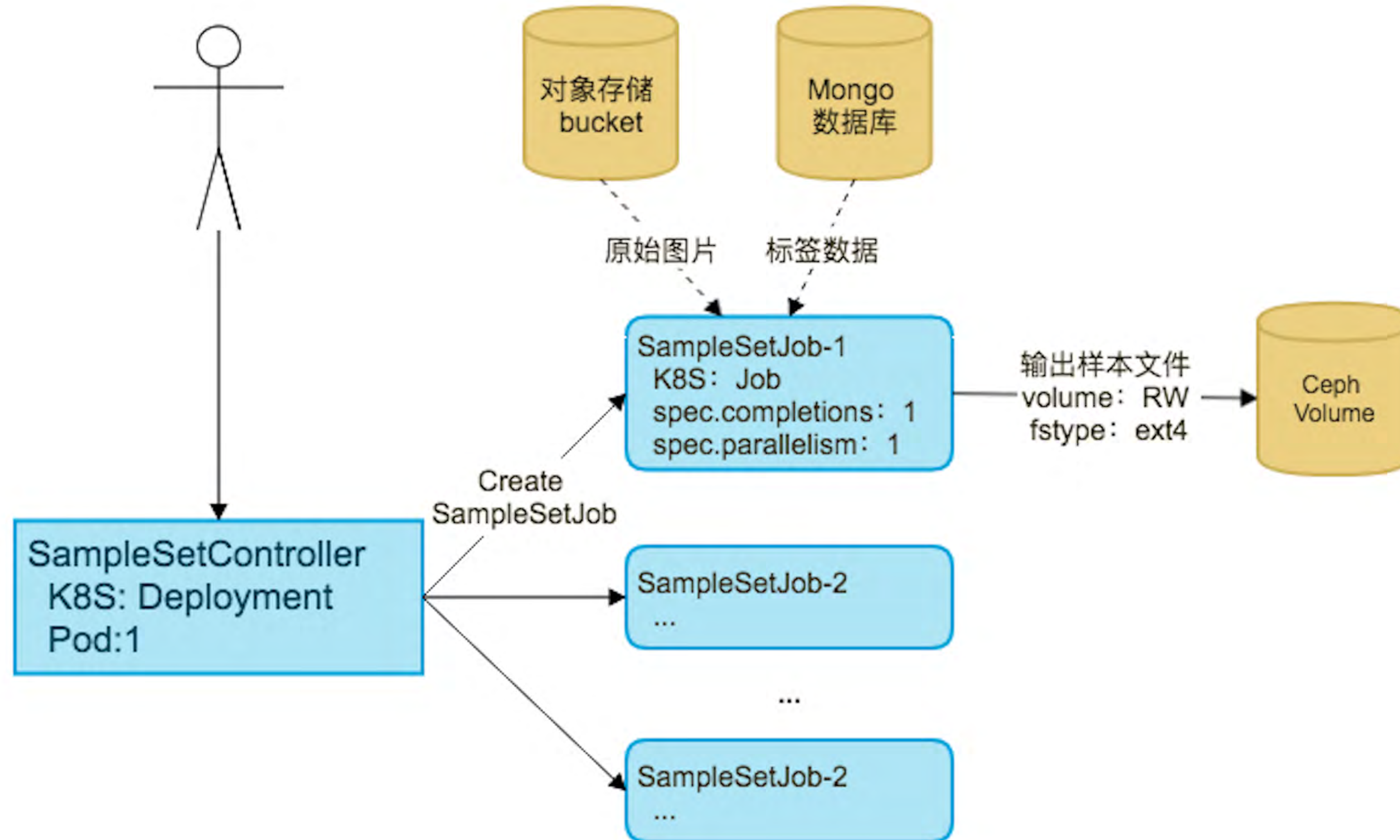
一次踩坑经历

接下来的工作

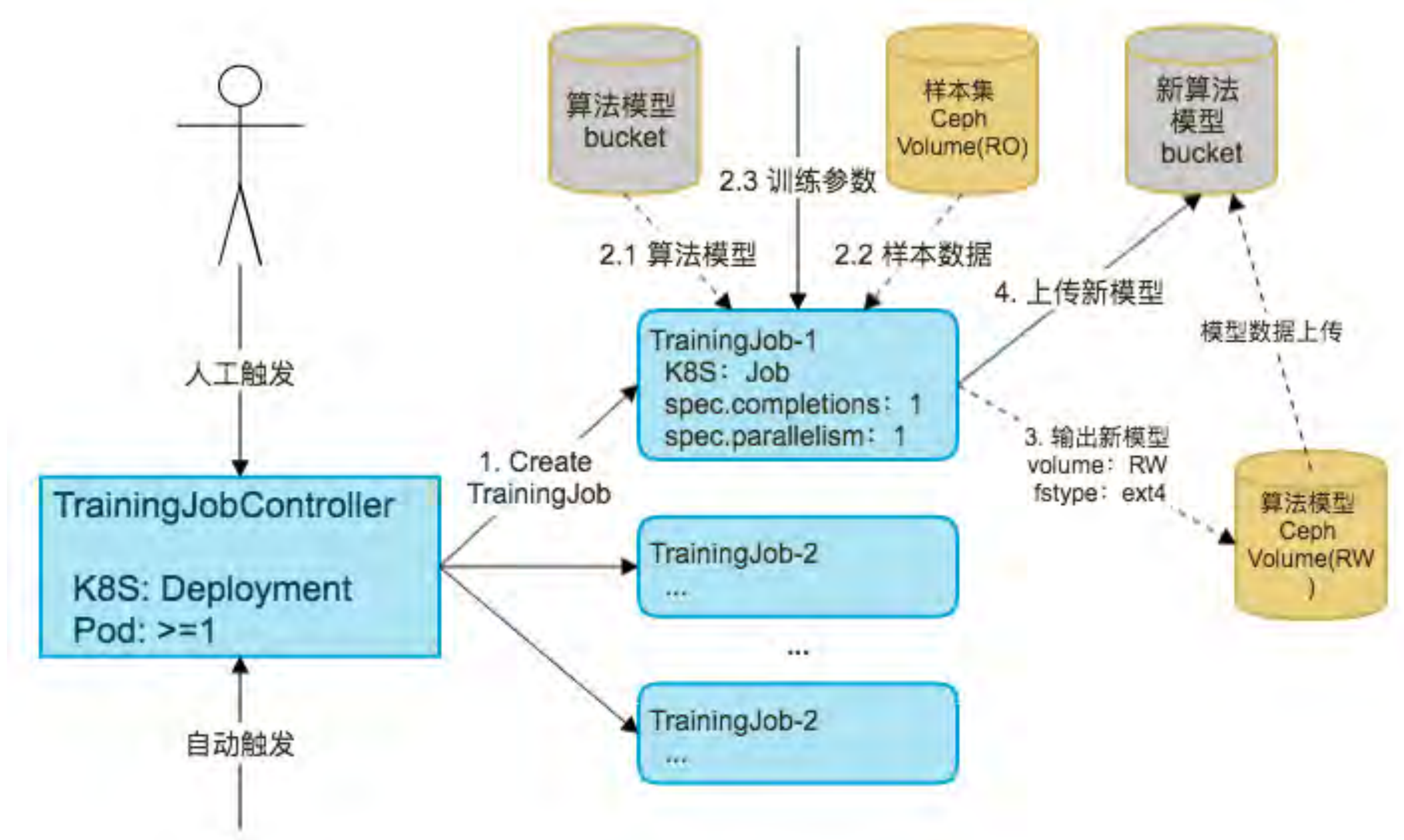
主办：



AI训练 - 生成样本集



AI训练 - 启动训练任务



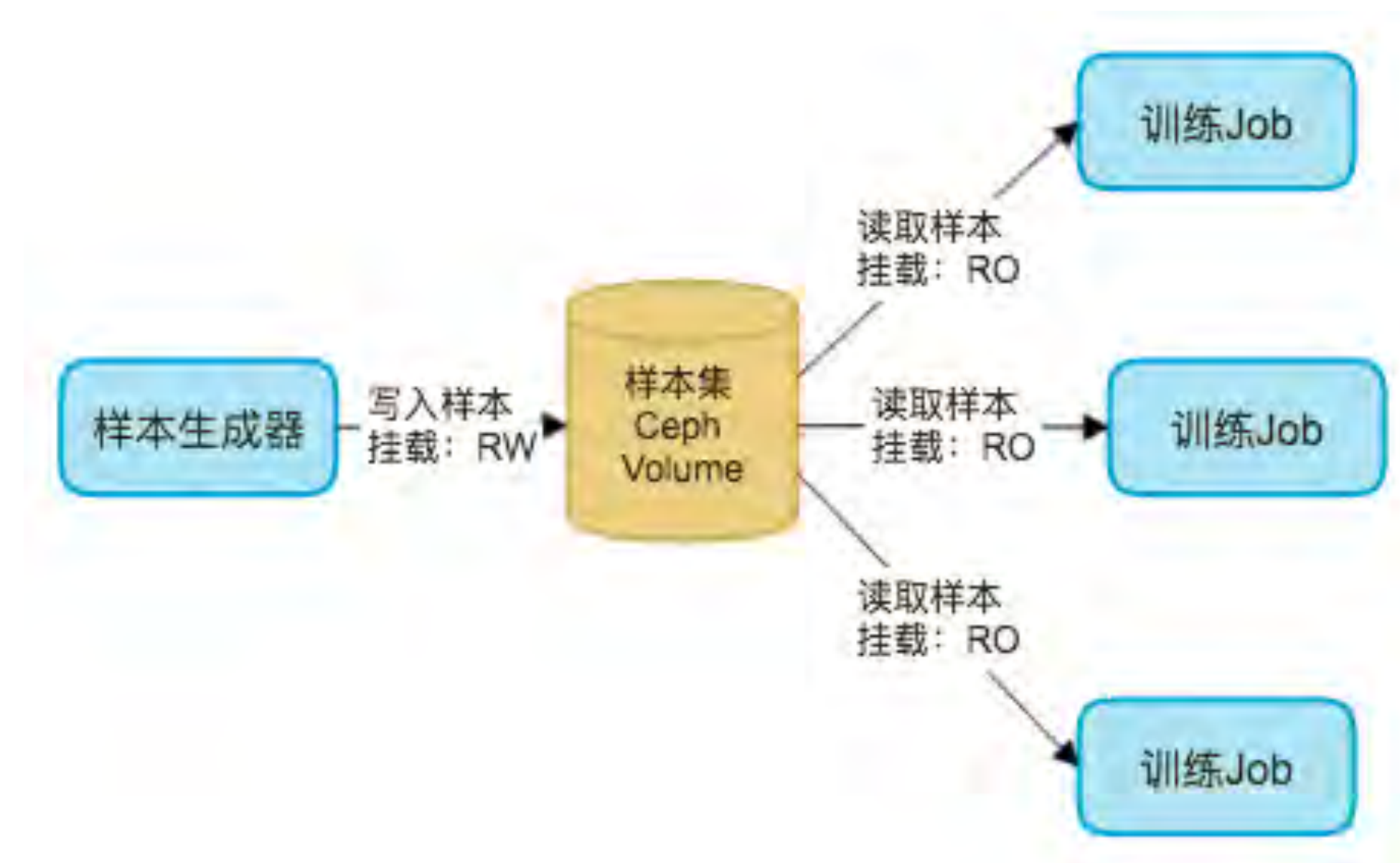
AI训练 - 使用 CEPH 存储

数据集规模大：10T+

- 网络共享卷，无需拷贝数据
- 水平扩展性良好

读写控制：

- 支持一写或者多读
- 生成样本时，不能训练
- 生成样本后，允许多个 Job 并发训练



主办：



CEPH on Kubernetes 的改进

imageFormat 2 的支持

支持 Volume cloning

RDB Provisioner out-of-tree 支持

- 与 K8S 解耦，独立外部进程运行
- 通过 list-watch apiserver 来操作 RBD

GPU 资源规划与分配

使用 Node Label + Node Selector

- Node Label 标注拥有 GPU 资源的物理机
- Node Selector 调度需求 GPU 的训练任务

Labels

```
beta.kubernetes.io/arch: amd64
beta.kubernetes.io/os: linux
kubernetes.io/hostname: xsgpu3
network: 10G   nvidia-driver-version: 384.59
nvidia-gpu-type: Tesla-K80
show fewer labels
```

```
▼ nodeSelectorTerms [1]
  ▼ 0 {1}
    ▼ matchExpressions [1]
      ▼ 0 {3}
        key : nvidia-gpu-type
        operator : In
        ▼ values [1]
          0 : Tesla-K80
```

主办：



GPU 资源规划与分配

使用 limits + request 合理分配资源

- limits 标注任务使用资源上限
- request 标注任务所需资源
- 实现合理超卖，充分利用资源

```
▼ limits {3}
  alpha.kubernetes.io/nvidia-gpu : 1
  cpu : 800m
  memory : 80Gi
▼ requests {3}
  alpha.kubernetes.io/nvidia-gpu : 1
  cpu : 200m
  memory : 1Gi
```

Nvidia GPU Driver

需求：

- Nvidia 驱动：硬件相关（K80、M40），部署于物理机上
- 训练容器：硬件相关，需要对应的的驱动

实现方式：

- 将物理机上的驱动安装路径定义为 HostPath Volume
- 运行容器时候挂载到容器内

```
▼ 2 {2}
  name : path--usr-local-nvidia
  ▼ hostPath {1}
    path : /var/lib/nvidia-docker/volumes/nvidia_driver/latest

▼ volumeMounts [5]
  ▶ 0 {3}
  ▶ 1 {2}
  ▼ 2 {3}
    name : path--usr-local-nvidia
    readOnly : true
    mountPath : /usr/local/nvidia
```

主办：



监控方案

物理机监控

- 基于 Prometheus Node Exporter
- 获取 CPU、内存、磁盘、网络维度信息

容器监控

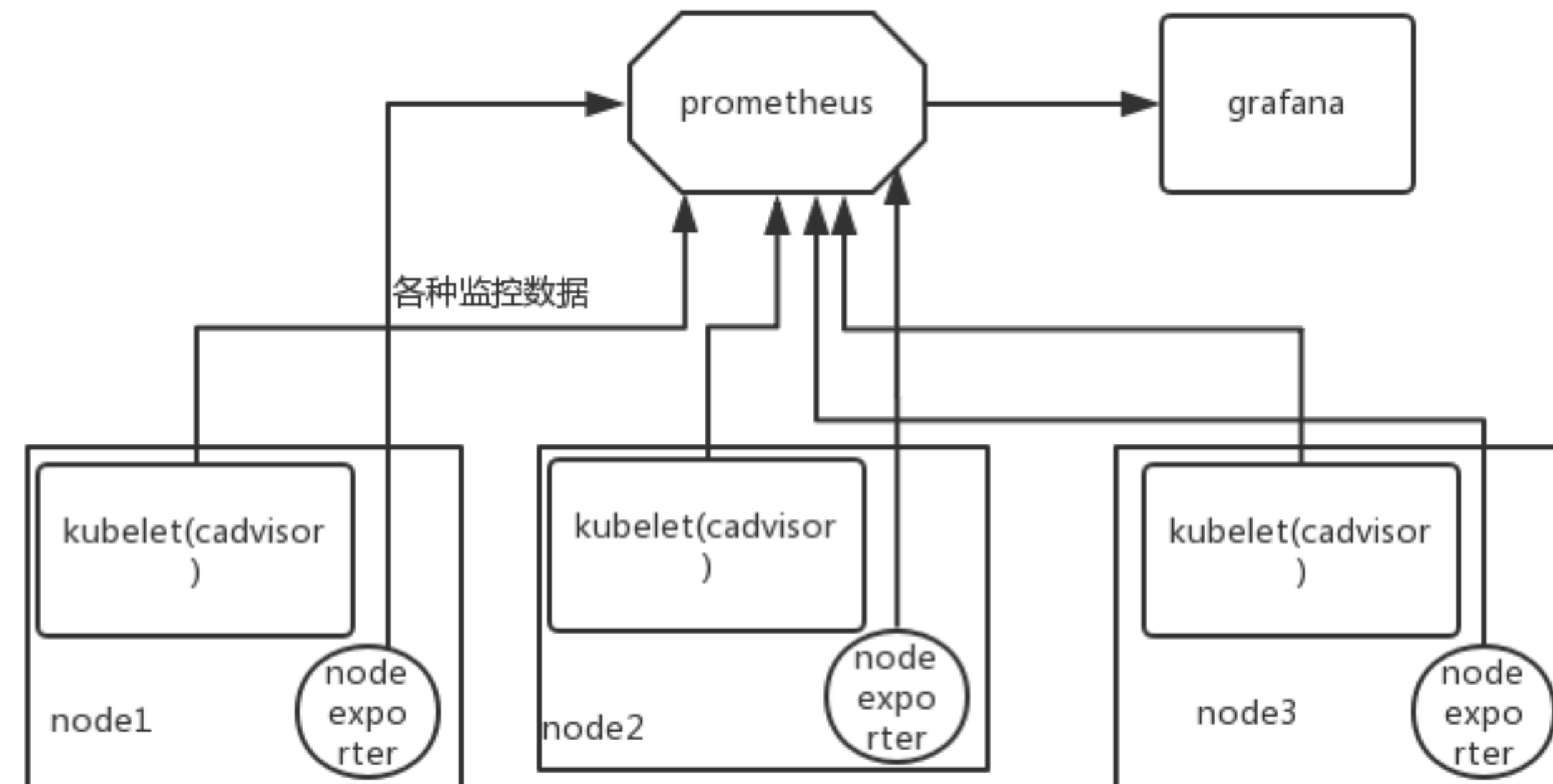
- kubelet 内嵌 cadvisor

监控注册

- Prometheus 从 kubernetes apiserver 获取需监控的资源

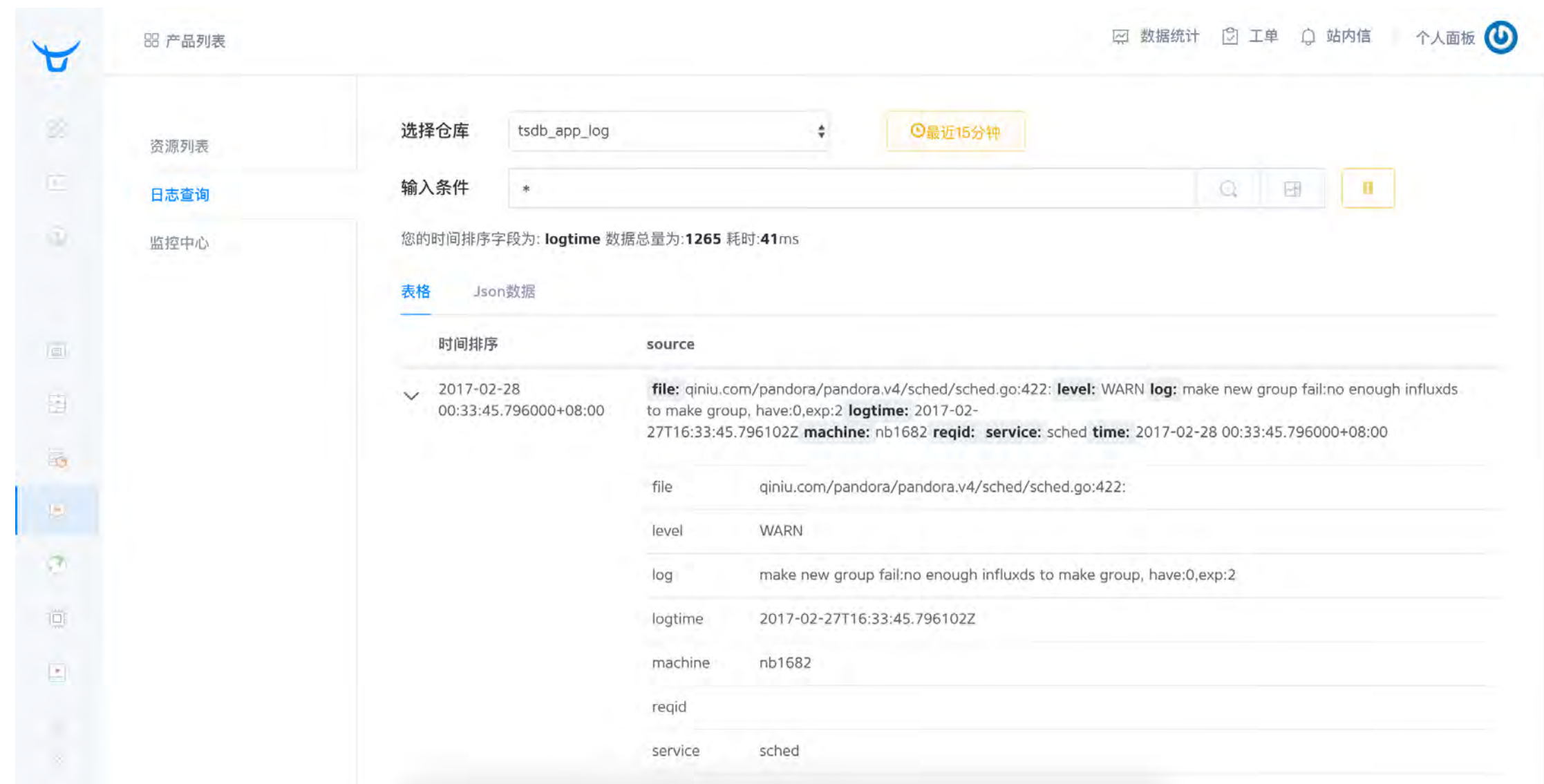
GPU 监控

- 为社区贡献：GPU 使用率、内存使用率、GPU核心使用率



基于 Pandora 日志存储、分析

基于ElasticSearch自研Sharding集群，承载七牛公有云所有日志
用户通过日志分析，快速定位事故原因、持续时间、影响范围等



主办：



AI训练的业务情况

kubernetes 的优势

基于 kubernetes 的AI训练

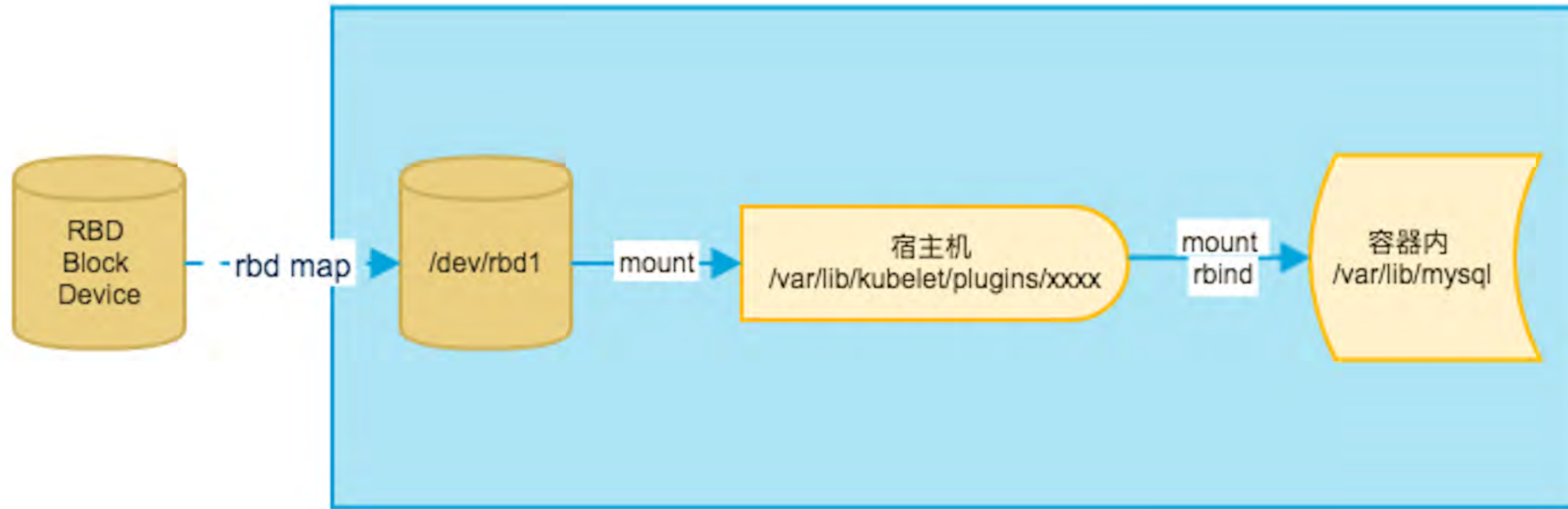
一次踩坑经历

接下来的工作

主办：



容器挂载 CEPH 过程



RBD Map : 虚拟块设备

mount : 格式化后，挂载文件系统到宿主机目录

mount rbind : 挂载宿主机目录到容器进程内

主办：



递归挂载 Volume 的传递性

挂载模式：

shared：相互传递

slave：主从传递

private：互补传递 (1.6 仅支持 private 模式)

例：

原始根目录下挂载：/mnt/a /mnt/b /mnt/c

rbind 挂载：将原始根目录递归挂载到特定目录

原始根目录：卸载 /mnt/b

rbind：卸载 /mnt/b

	原始根目录/	rbind /
Shared	/mnt/a	/mnt/a
Slave	/mnt/a /mnt/c	/mnt/a
Private	/mnt/a /mnt/c	/mnt/a /mnt/b

坑

1. 容器 A 已 RO 方式挂载 RBD Volume: /var/lib/kubelet/plugins/v1
2. Node exporter 以 make-private 方式挂载宿主机根目录
3. 容器 A 销毁后，写在 RBD Volume 成功，但是 rbd umap 失败：因为 node-exporter 仍在 RO 挂载
4. 容器 B 以 RW 的方式挂载同一 RBD Volume，由于 node-exporter 仍在 RO 挂载，导致容器 B 挂载失败
5. Kubernetes 使用 lsblk 获取文件系统失败
6. Kubernetes 触发格式化磁盘，数据丢失

故障原因

Node-exporter

Node exporter 以 DaemonSet 方式运行，挂载了整个物理机根目录
Kubernetes 只支持 mount make-private 的方式，Volume 卸载不具有传递性



Ceph

Ceph red client 与服务端版本不一致，导致 lsblk 获取文件系统类型失败



Kubernetes

在获取文件系统失败时，直接格式化 Volume



主办：



故障反思

部署固化，自动化

固化各个组件部署流程

部署后，检查相关组件的版本号

错误日志收集、告警

收集各个系统组件的 warning、error、fatal 日志，并设置告警，及时跟进

收集宿主机系统日志，设置告警，及时跟进

Kubernetes

梳理 Kubernetes 对 Volume 相关处理逻辑

Node-exporter 不应以 make-private 方式递归挂载根目录

主办：



AI训练的业务情况

kubernetes 的优势

基于 kubernetes 的AI训练

一次踩坑经历

接下来的工作

主办：



自配置监控

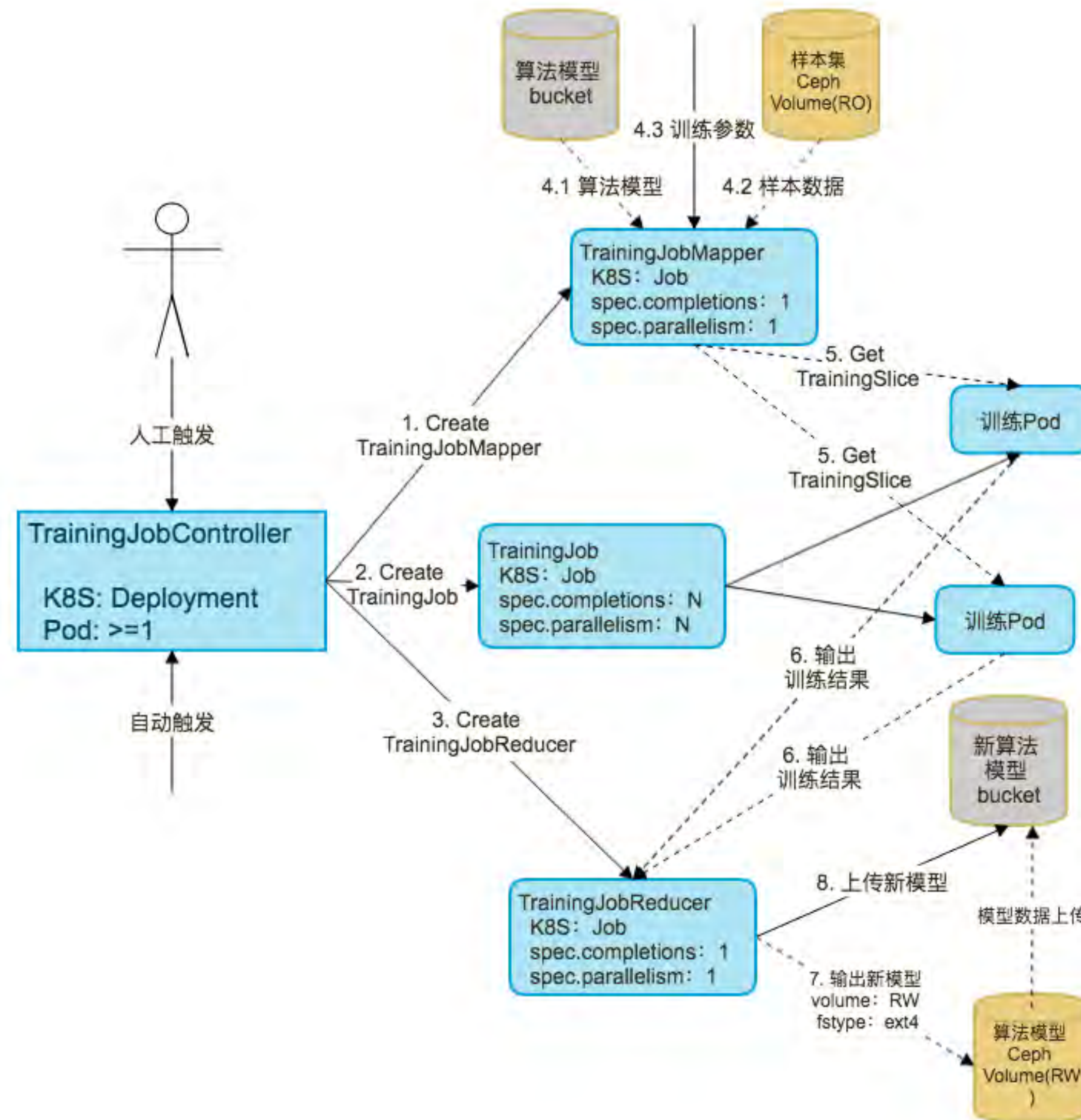
创建 Kubernetes Services 时添加 annotation

- 监控数据抓取服务端口
- 监控数据抓取路径

Prometheus 自动发现服务，抓取监控数据

```
...  
# The service being deployed exposes a metric  
# scraped by Prometheus but metrics are exposed  
prometheus.io/scrape: 'true'  
prometheus.io/path:    /foo/bar/metrics  
prometheus.io/port:   '8080'  
...
```

分布式训练



主办：



Q&A

Thank you for your time



2017
China Kubernetes
End User Conference
kubernetes 中国用户大会 — 2017 —

2017.10.15 / 中国·杭州

主办：

