



Ray History Server的历史任务可观测性实践

吴 昆

阿里巴巴高级研发工程师

2025/12/20



吴昆

阿里巴巴 高级研发工程师

个人简介：吴昆就职于阿里巴巴容器服务团队，是容器服务调度领域负责人以及Ray on ACK研发，同时也是Koordinator, KubeRay, Kueue, Kubernetes等社区贡献者。

CONTENT

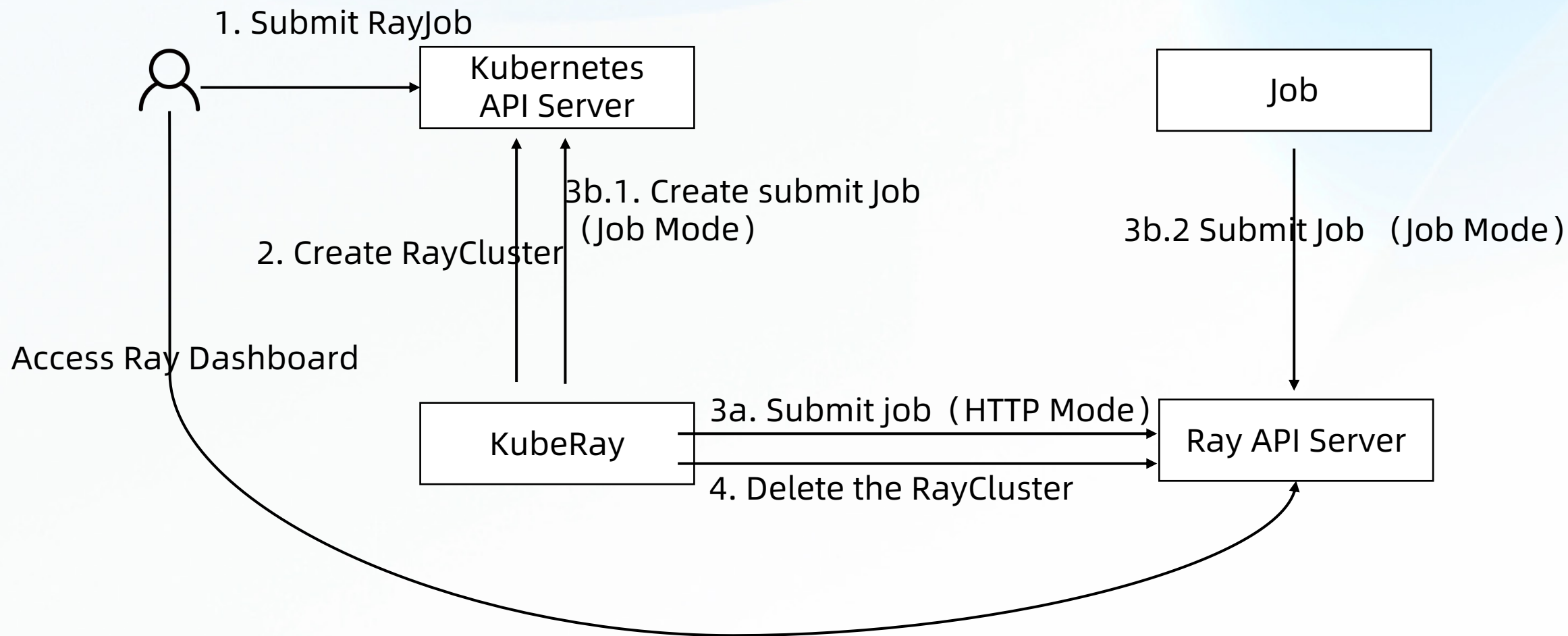
目录

01 为什么需要 History Server

02 History Server 的设计与实现

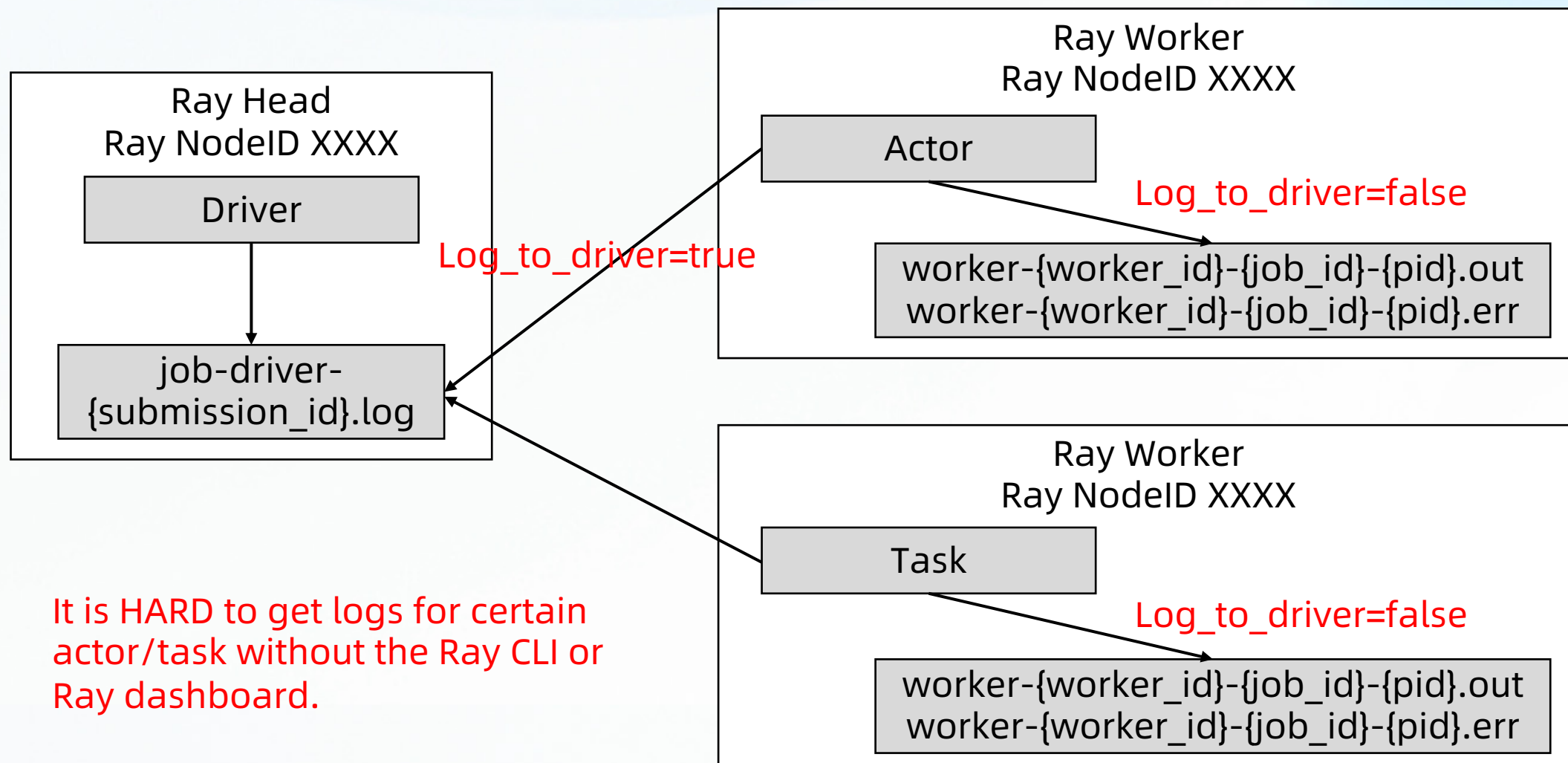
03 History Server 开源 RoadMap

RayJob on Kubernetes



We can not access the dashboard after the job complete

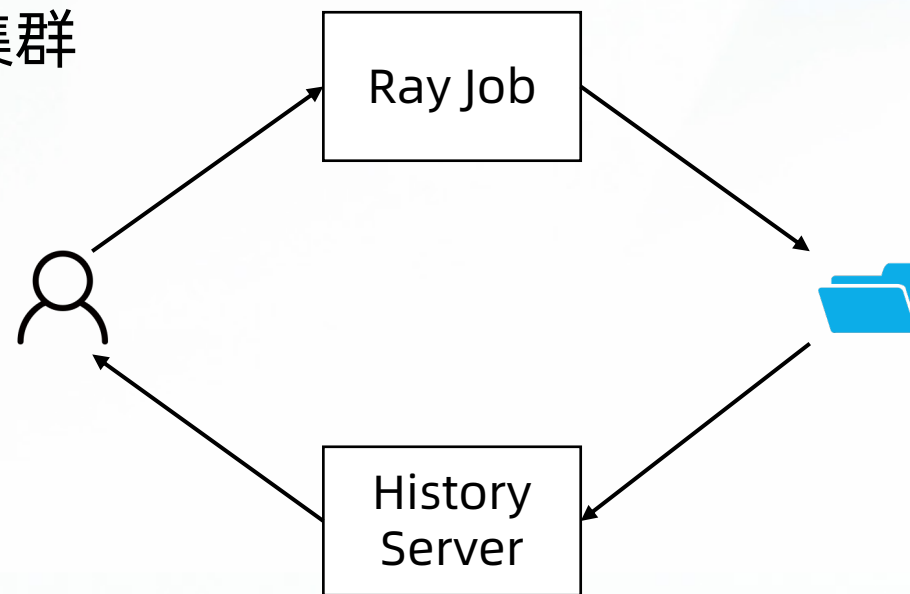
Ray的日志管理方式



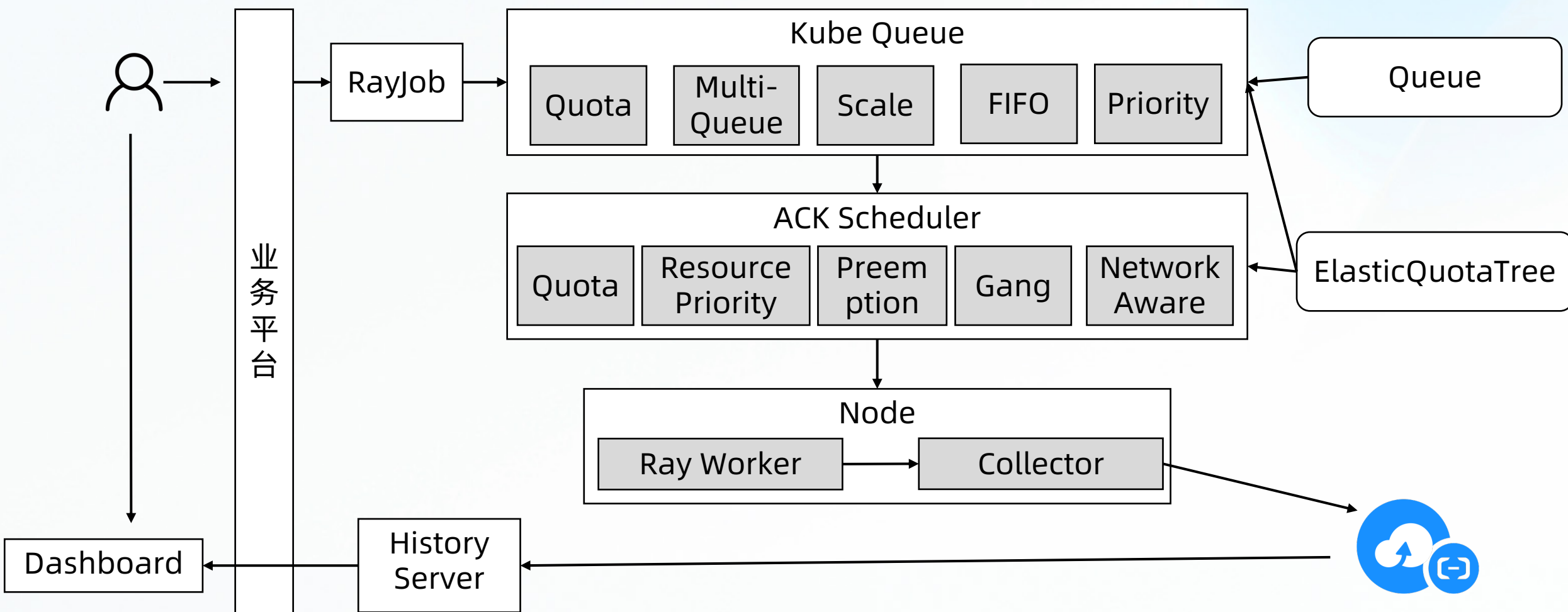
Why History Server ?



- 支持集群销毁后通过Ray Dashboard访问日志以及集群状态，保持一致的用户体验
- 日志以及集群数据存储在后端，避免部署多个Ray APIServer，节省集群资源
- 通过Event收集，HistoryServer可以支持集群状态回放等功能
- 通过Event收集以及重放，减少对Ray API Server的访问，提升Ray API Server性能



Ray on ACK使用场景



History Server 的设计与实现

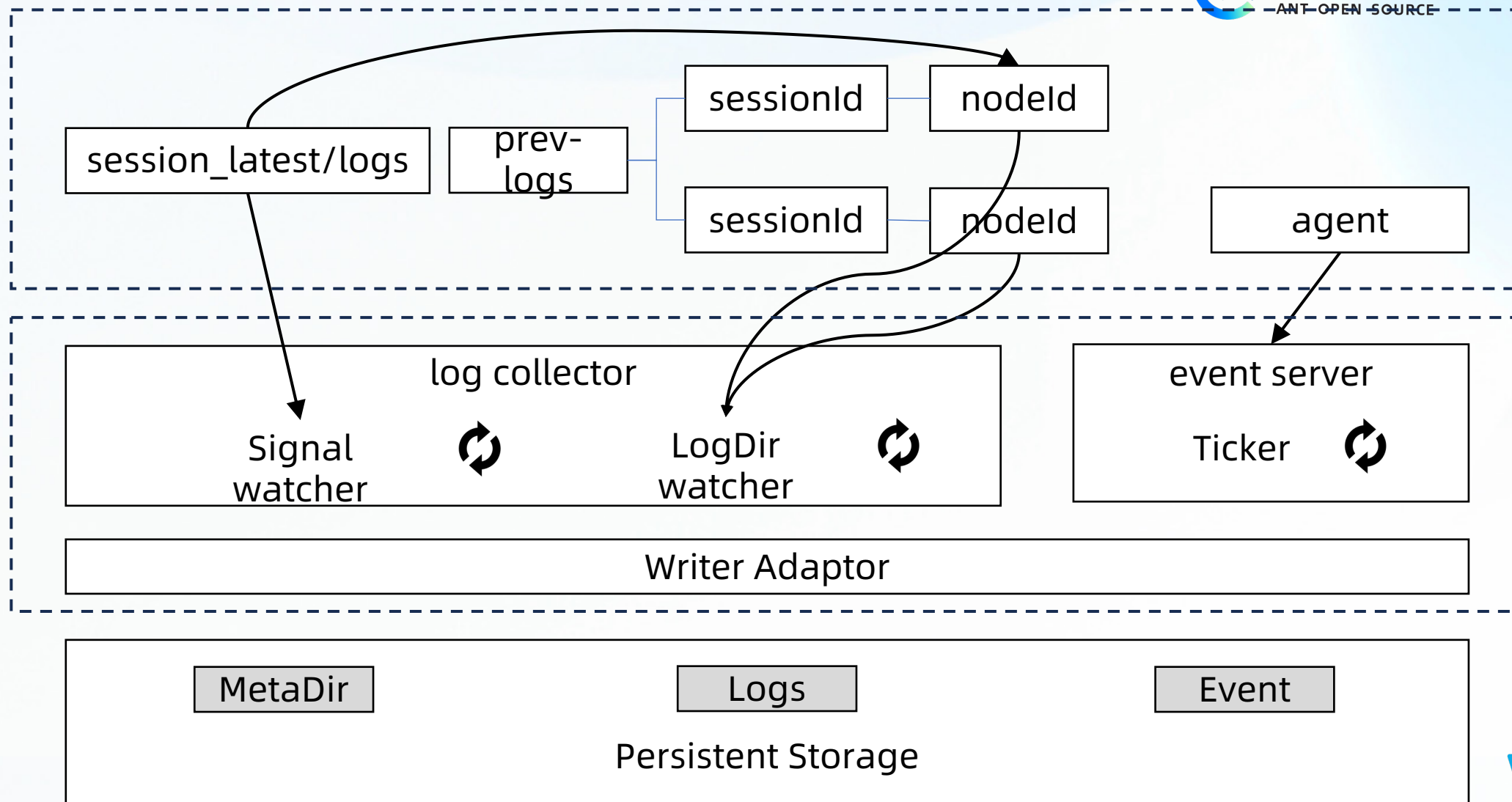


蚂蚁开源

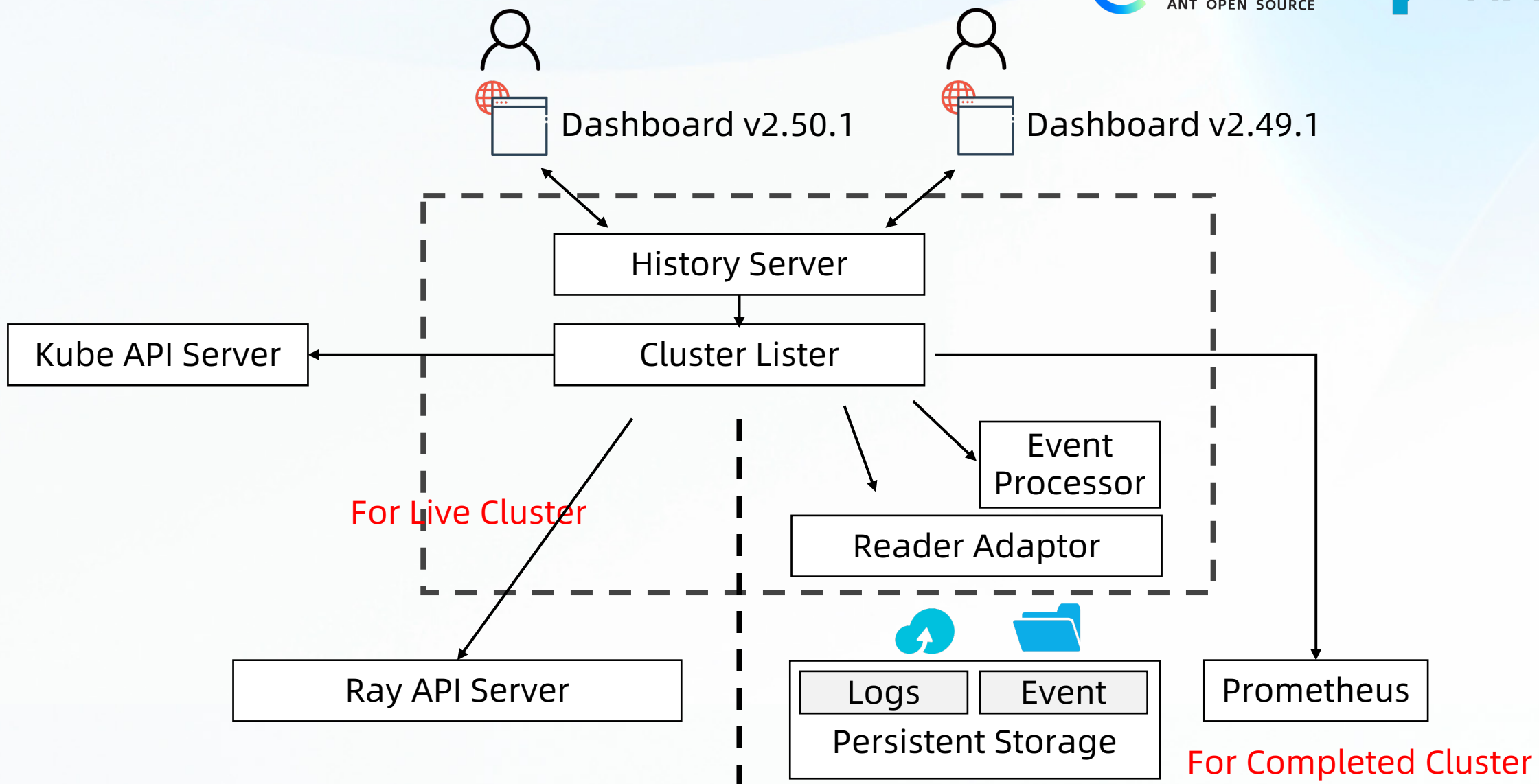
ANT-OPEN-SOURCE



RAY



History Server 的设计与实现



History Server 的设计与实现



KubeRay HistoryServer Dashboard

Instructions: Select a cluster from the list below, choose a dashboard version, and click "Enter Cluster Dashboard" to view the cluster details.

Available Clusters

wukun

Namespace: default
Created: 2025-11-13 14:10:27 +0800 CST
Session: live

Dashboard Version:

Ray Dashboard

Enter Cluster Dashboard

wukun

Namespace: default
Created: 2025-11-12T22:10:30Z
Session: session_2025-11-12_22-10-30_318513_1

Dashboard Version:

Ray Dashboard

Enter Cluster Dashboard

wukun

Namespace: default
Created: 2025-11-12T21:48:38Z
Session: session_2025-11-12_21-48-38_112021_1

Dashboard Version:

1. Select a version

v2.51.0

2. Enter the dashboard

Enter Cluster Dashboard

wukun

Namespace: default
Created: 2025-11-12T21:46:37Z
Session: session_2025-11-12_21-46-37_444456_1

Dashboard Version:

Ray Dashboard

Enter Cluster Dashboard

wukun

Namespace: default
Created: 2025-11-12T20:01:28Z
Session: session_2025-11-12_20-01-28_947657_1

Dashboard Version:

Ray Dashboard

Enter Cluster Dashboard

wukun

Namespace: default
Created: 2025-11-12T19:59:23Z
Session: session_2025-11-12_19-59-23_509375_1

Dashboard Version:

Ray Dashboard

Enter Cluster Dashboard



localhost:8080/#/overview

Overview Jobs Serve Cluster Actors Metrics Logs

GMT+8

Cluster utilization

Set up Prometheus and Grafana for better Ray Dashboard experience

Time-series charts are hidden because either Prometheus or Grafana server is not detected. Follow these instructions to set them up and refresh this page.

Recent jobs

No jobs yet...

View all jobs →

Serve Deployments

No Deployments yet...

View all deployments →

Cluster status and autoscaler

Node count

Set up Prometheus and Grafana for better Ray Dashboard experience

Time-series charts are hidden because either Prometheus or Grafana server is not detected. Follow these instructions to set them up and refresh this page.

No cluster status.

No cluster status.

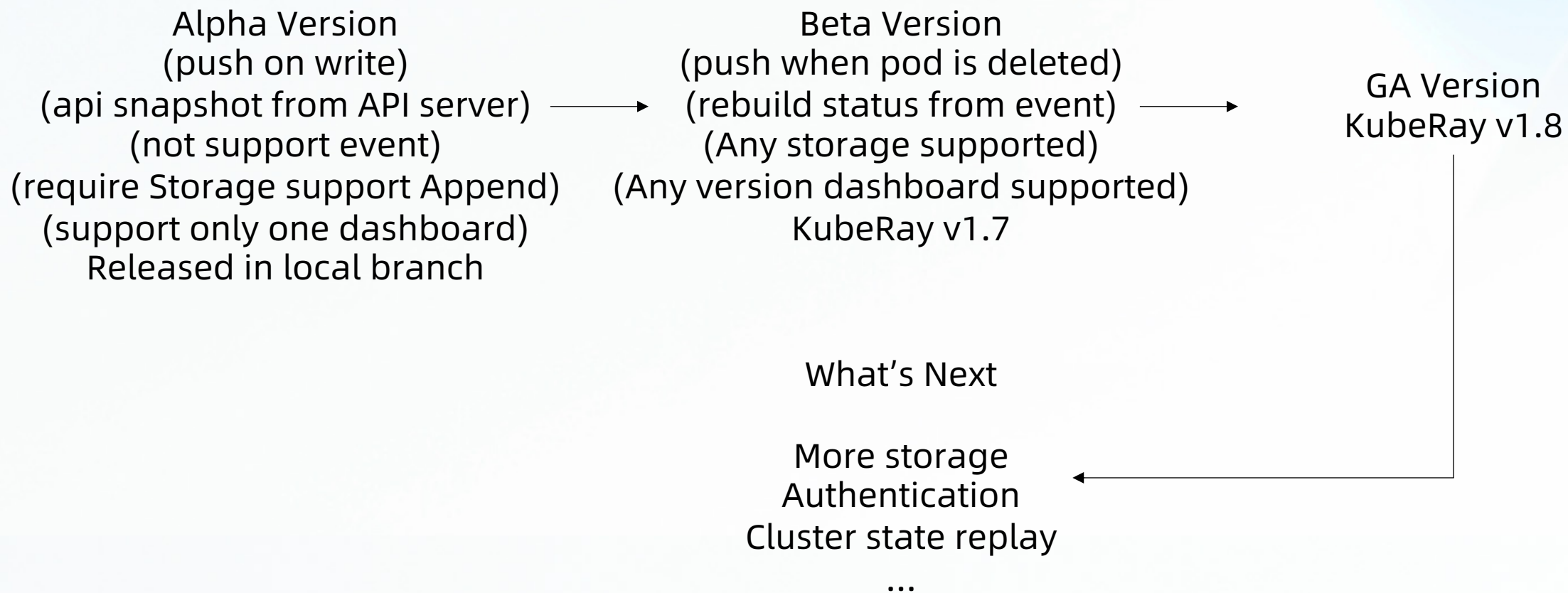
Demo视频



```
1 # How to use history server
2 # 0. Install KubeRay in your Kubernetes cluster
3 kubectl api-resources | grep ray
4 # 1. Implement the interface for different storage backends
5 # 2. Create a ConfigMap for storing the information to access the storage backend
6 # ---
7 # apiVersion: v1
8 # data:
9 #   data: ${Your data}
10 # kind: ConfigMap
11 # metadata:
12 #   name: historyserver-config
13 #   namespace: default
14 kubectl create -f historyserver/config/samples/aliyuncs/configmap.yaml
15 # 3. Create RayCluster with collectors (and sidecar and lifecycle hook to the main container)
16 # ---
17 # ...
18 # containers:
19 #   - name:
20 #     ...
21 #     name: ray-head
22 #     lifecycle:
23 #       postStart:
24 #         exec:
25 #           command:
26 #             - /bin/sh
27 #             - -c
28 #             - |
29 #               while true; do
30 #                 getnodeid() {
31 #                   while true; do
32 #                     nodeid=$(ps -ef | grep raylet | grep node_id | grep -v grep | grep -oP '(?=<node_id>[")' | tr -d '"')
33 #                     if [ -n "$nodeid" ]; then
34 #                       echo "$data" raylet started: \$(ps -ef | grep raylet | grep node_id | grep -v grep | grep -oP '(?=<node_id>[")' | tr -d '"') >> $
35 #                       (nodeid) >> /tmp/ray/init.log
36 #                       echo nodeid > /tmp/ray/raylet_node_id
37 #                       break
38 #                     else
39 #                       echo "$data" raylet not start >> /tmp/ray/init.log"
40 #                       sleep 1
41 #                     fi
42 #                   done
43 #                 }
44 #                 getnodeid
45 #       - name: collector
46 #       image: registry.cn-bjshang.aliyuncs.com/image-yueming/ray-historyserver:v0.0.1
47 #       imagePullPolicy: Always
48 #       env:
49 #         # Your custom InitConfigs here and in config map
50 #       command:
51 #         - collector
52 #         - --ray-head # Worker for worker nodes
53 #         - --runtime-class-name=${Your runtime class name, implemented in previous step}
54 #         - --ray-cluster-name=demo
55 #         - --ray-root-dir=log
56 #       volumeMounts:
57 #         - name: historyserver
58 #         mountPath: /tmp/ray
59 #         - name: config-volume
60 #         mountPath: /var/collector-config
61 #       volumes:
62 #         # Some other configs here, created in previous step
63 #         - name: config-volume
64 #         configMap:
65 #           name: historyserver-config
66 #           items:
67 #             - key: data
68 #             path: data
69 #         # For log storage
70 #         - name: historyserver
71 #         emptyDir: {}
72 kubectl create -f historyserver/config/samples/aliyuncs/raycluster.yaml
73 # 4. Deploy the history server in your cluster
74 kubectl create -f historyserver/config/samples/aliyuncs/historyserver.yaml
75 kubectl create -f historyserver/config/samples/aliyuncs/svc.yaml
76 # 5. Use port-forward to access the history server
77 kubectl port-forward svc/historyserver 8088:30088
78
```

1. Ensure api is installed

History Server RoadMap



谢谢
欢迎大家进群交流

