

CommunityOverCode

THE ASF CONFERENCE

使用 Apache APISIX 和 KEDA 构建云原生弹性应用



Jintao Zhang - API7.ai

个人介绍

- 张晋涛
- Apache APISIX PMC
- Kubernetes Ingress-NGINX maintainer
- Microsoft MVP
- 『K8S 生态周报』维护者
- 公众号: MoeLove
- <https://github.com/tao12345666333>

CONTENTS

1. 为什么应用需要弹性伸缩
2. 如何实现弹性伸缩
3. KEDA 的价值
4. 如何用 APISIX 与 KEDA 构建弹性应用



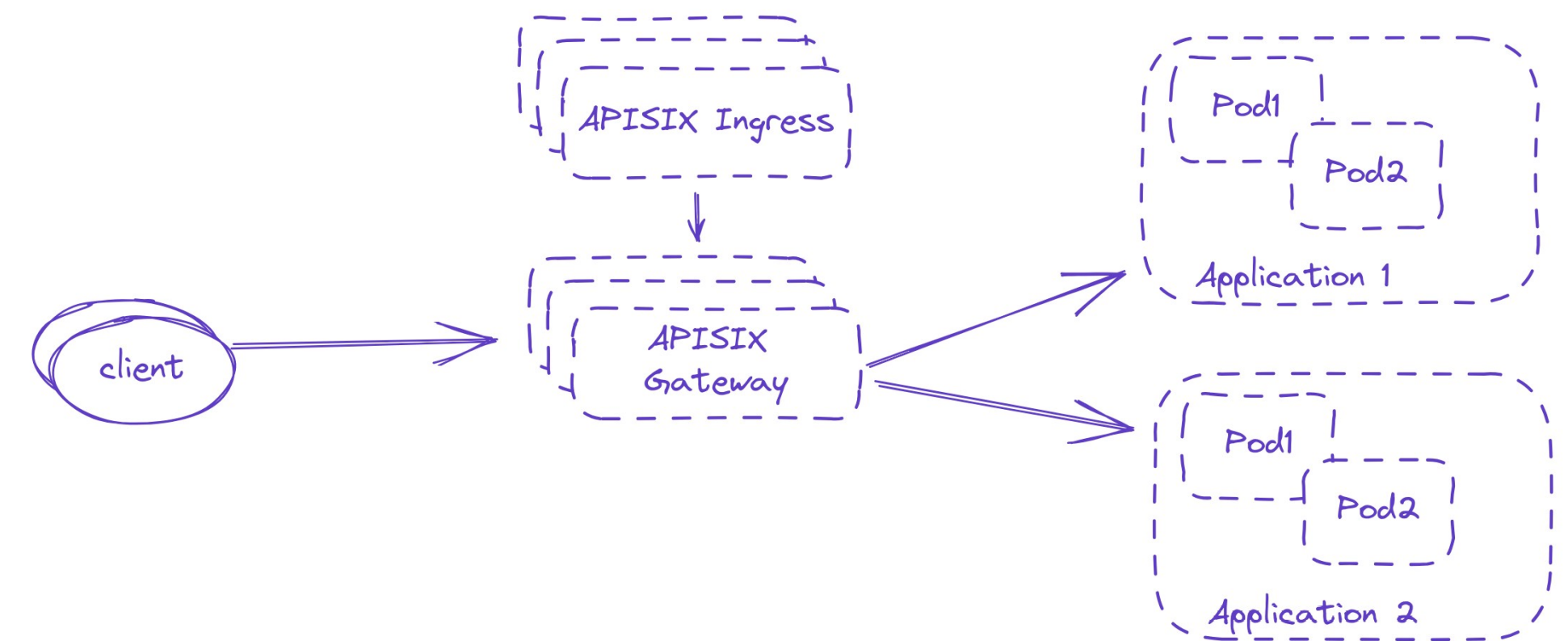
CommunityOverCode

THE ASF CONFERENCE

为什么应用需要 弹性伸缩

云原生应用

- Pod 是最小的调度单元
- Kubernetes 通过控制器来管理应用的副本
- Kubernetes 中可以为 Pod 设置可用资源容量



为什么需要弹性伸缩 / 灵活扩缩容

- 故障自愈：保证总是有足够的 / 预期的副本提供服务
- 应对不同规模的负载
- 提升服务质量：业务高峰时，避免影响用户体验
- 成本优化：业务低谷时，避免资源浪费
- 支持业务快速迭代

CommunityOverCode

THE ASF CONFERENCE

如何实现弹性伸缩

Horizontal Pod Autoscaler (HPA)

- 变更 Pod 数量
- 作用对象 Deployment、StatefulSet 等可扩展 / 缩容对象
- 通过 HorizontalPodAutoscaler 资源定义规则
- 使用 Metrics Server 的指标
- 持续动态进行调整

```
1 apiVersion: autoscaling/v2
2 kind: HorizontalPodAutoscaler
3 metadata:
4   name: moelove-hpa
5 spec:
6   scaleTargetRef:
7     apiVersion: apps/v1
8     kind: Deployment
9     name: moelove-blog
10  minReplicas: 1
11  maxReplicas: 10
12  metrics:
13  - type: Resource
14    resource:
15      name: cpu
16      target:
17        type: Utilization
18        averageUtilization: 50
```

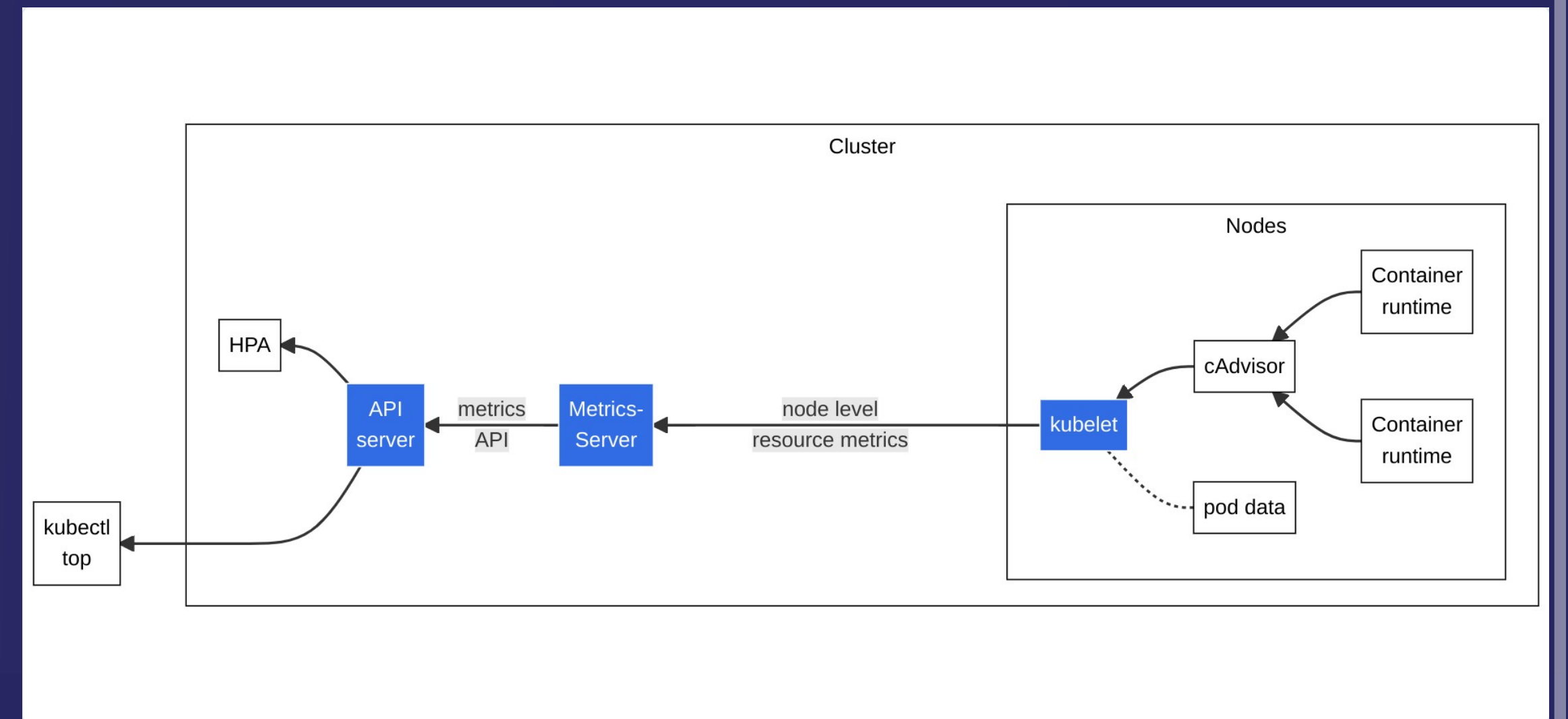

Vertical Pod Autoscaler (VPA)

- 变更 Pod 可使用的资源 CPU/Memory
- 通过 VerticalPodAutoscaler 资源定义规则
- 使用 Metrics Server 的指标
- 持续动态进行调整

```
1 apiVersion: autoscaling.k8s.io/v1
2 kind: VerticalPodAutoscaler
3 metadata:
4   name: my-app-vpa
5 spec:
6   targetRef:
7     apiVersion: "apps/v1"
8     kind:      Deployment
9     name:      my-app
10  updatePolicy:
11    updateMode: "Auto"
```

Metrics Server

- <https://github.com/kubernetes-sigs/metrics-server>
- 聚合相关组件的 metrics
- 通过 metrics API 提供 (`metrics.k8s.io`)
- `custom.metrics.k8s.io` 扩展
- `external.metrics.k8s.io` 扩展



当前 HPA 方案的不足

- 依赖 metrics API
- 可度量的指标有限
 - Container resource metrics (v1.27 beta)
 - custom metrics (v1.23 stable)
 - multiple metrics (v1.23 stable)
- 不能 / 不易满足实际应用中多样化的弹性伸缩条件
 - 定时
 - 与业务逻辑关联
 - 外部系统触发
- ...

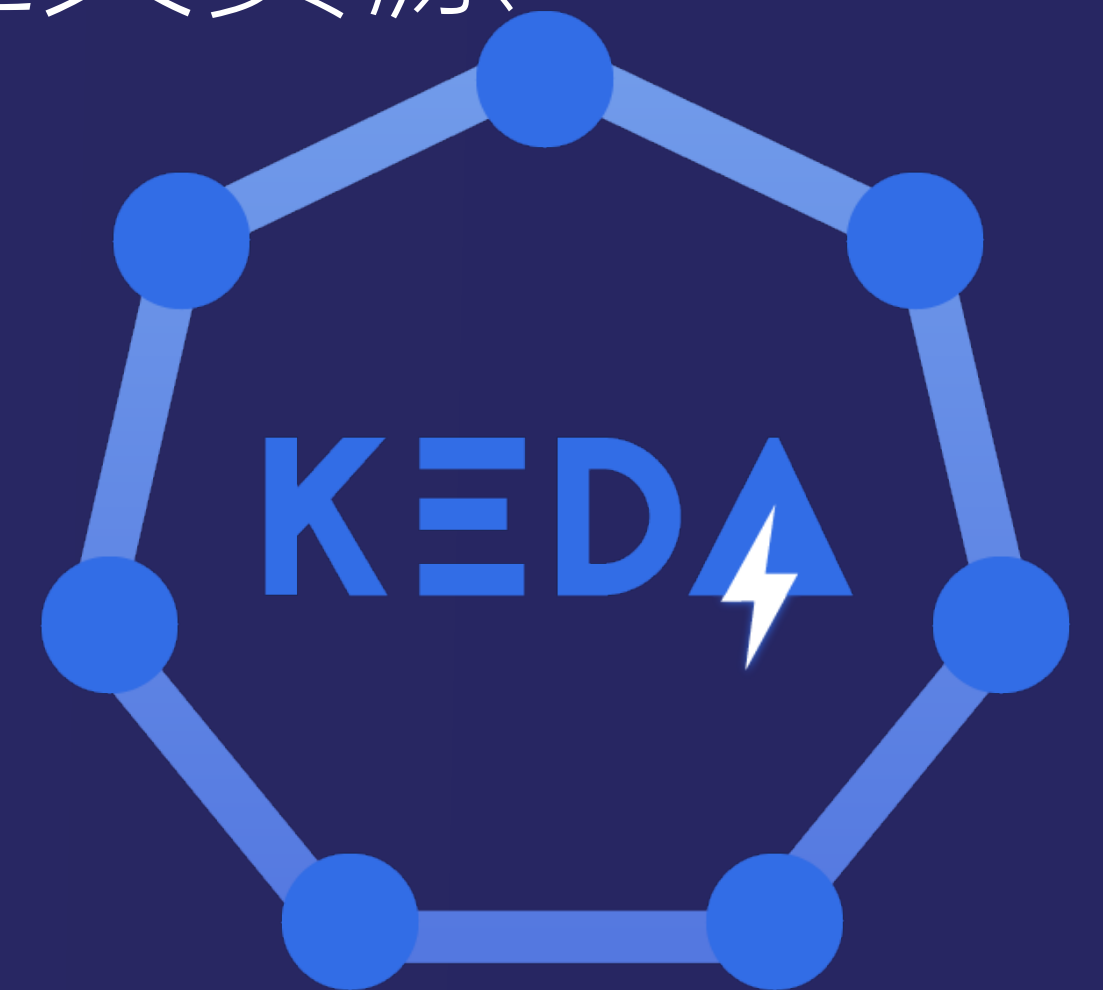
CommunityOverCode

THE ASF CONFERENCE

KEDA 的价值

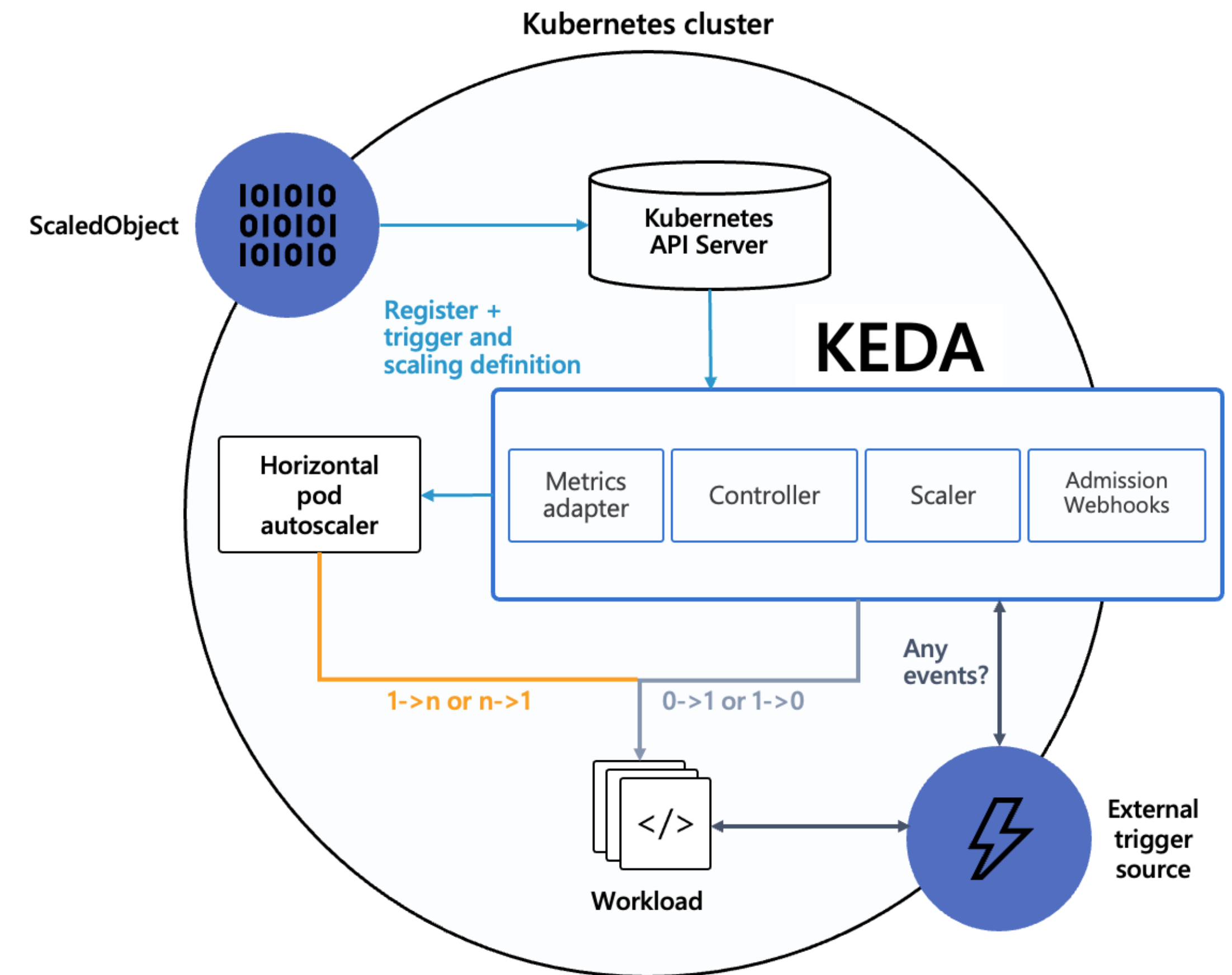
KEDA 是什么

- Kubernetes Event-driven Autoscaling
- 50+ 内置扩展器
- 可缩容至 0 副本
- 支持多种资源扩缩容，包括含有 `/scale` 的自定义资源
- Azure Function 支持



KEDA 如何工作

- Agent
- Metrics: custom metrics
- Admission webhooks
- CRDs:
 - ScaledObjects
 - ScaledJobs
 - TriggerAuthentication
 - ClusterTriggerAuthentication



KEDA 支持的扩展器

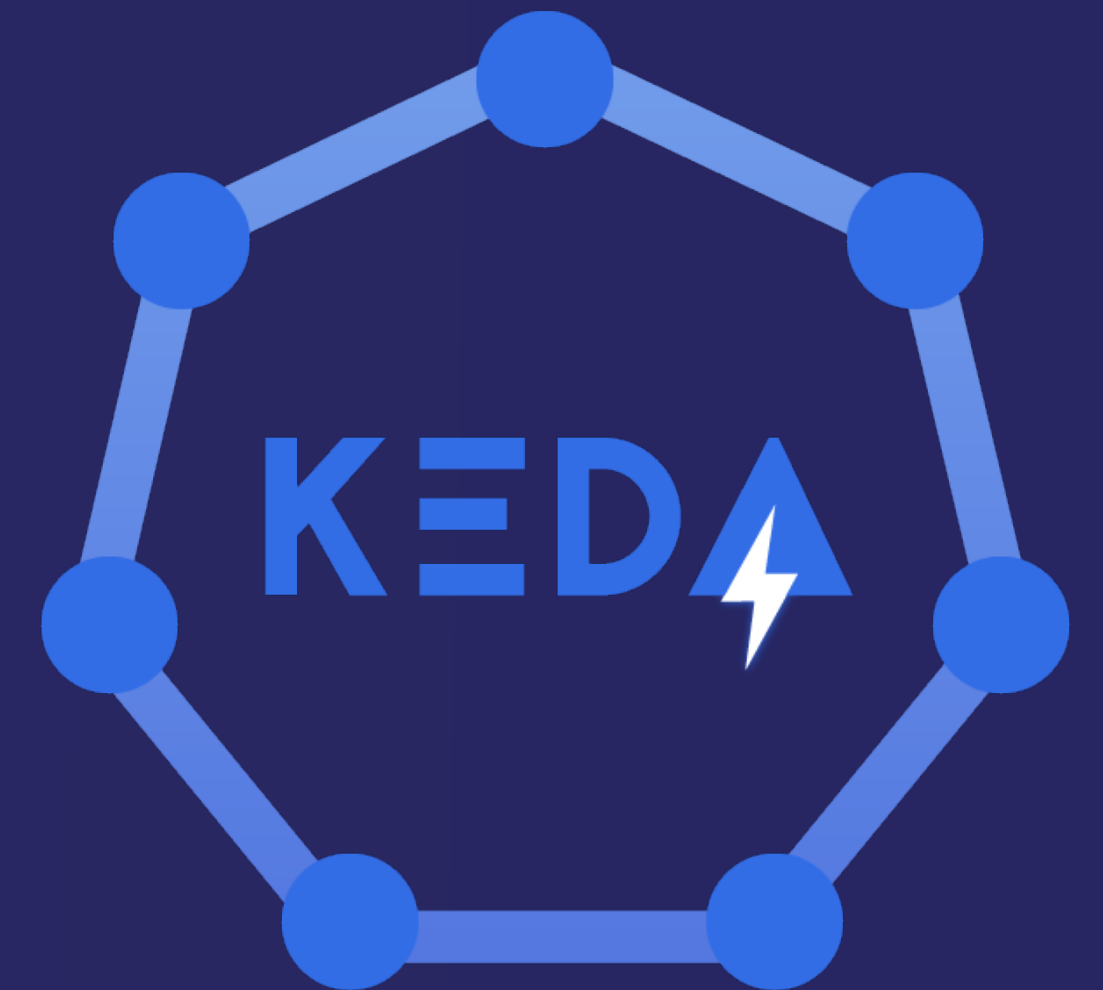
- Cron
- CPU
- External
- MQ
- DB
- ...

KEDA has a wide range of **scalers** that can both detect if a deployment should be activated or deactivated, and feed custom metrics for a specific event source. The following scalers are available:



KEDA 如何解决默认 HPA 的不足

- 丰富的扩展器, **cron**, 资源, **MQ** 等
- 允许多种指标 & 集成
- 可与业务系统进行关联



CommunityOverCode

THE ASF CONFERENCE

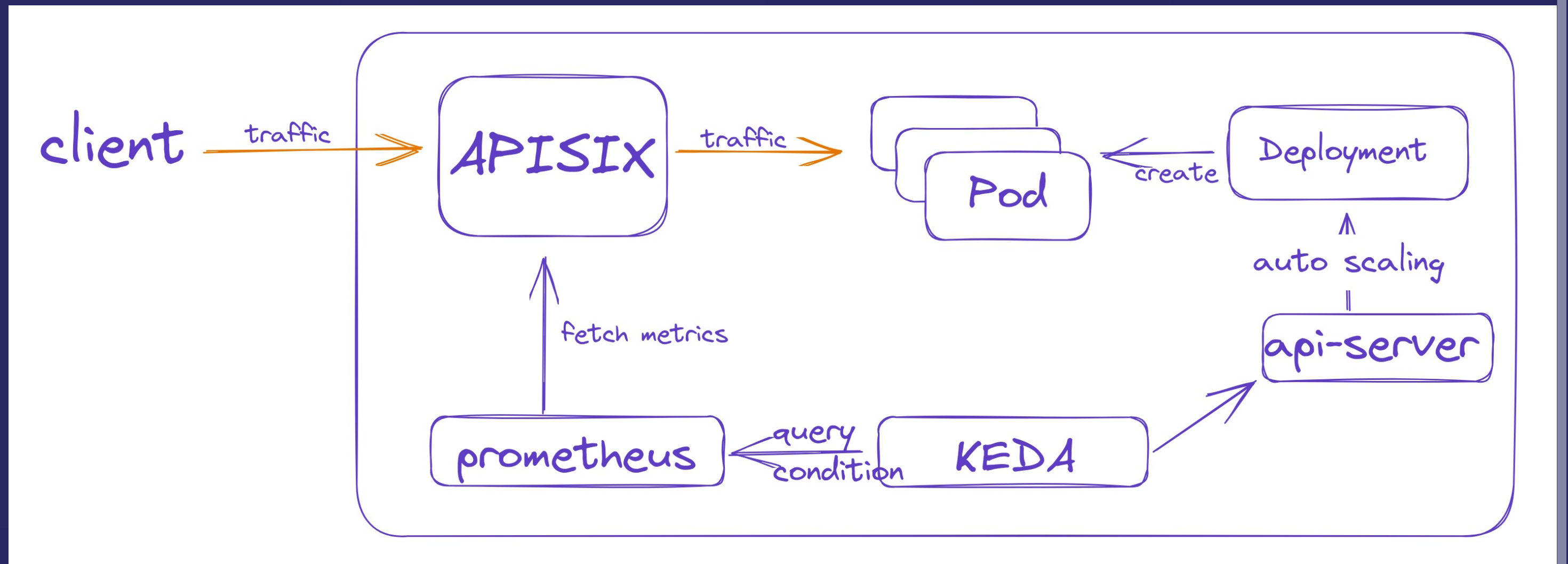
APISIX & KEDA 的价值

为什么将 APISIX 与 KEDA 集成

- Apache APISIX 是 API Gateway
- 业务流量的入口
- 网关侧更易了解整个业务的流量情况
- 可对整体业务进行按比例缩放

如何集成

- Apache APISIX 开启 Prometheus 插件
- KEDA 利用 Prometheus 扩展器查询 APISIX 的 metrics
 - nginx_http_current_connections
 - http_requests_total
 - upstream_status
 - http_status
 - http_latency
 - bandwidth



CommunityOverCode

THE ASF CONFERENCE

Thanks

Jintao Zhang | Apache APISIX PMC
<https://github.com/tao12345666333>

WWW.COMMUNITYOVERCODE.ORG

