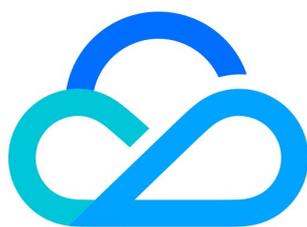


# 容器服务

## TKE Serverless 集群指南



腾讯云

## 【 版权声明 】

©2013-2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

## 【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

# 文档目录

## TKE Serverless 集群指南

TKE Serverless 集群概述

购买 TKE Serverless 集群

地域和可用区

计费概述

购买限制

欠费说明

TKE Serverless 集群管理

创建集群

连接集群

集群生命周期

注意事项

Kubernetes 对象管理

工作负载管理

服务管理

其他资源管理

Annotation 说明

Serverless 集群全局配置说明

运维中心

日志采集

使用 CRD 采集日志到 Kafka

使用 CRD 采集日志到 CLS

开启日志采集

通过控制台配置日志采集

通过 YAML 配置日志采集

审计管理

审计日志

审计仪表盘

事件管理

事件日志

事件仪表盘

监控和告警

常见问题

# TKE Serverless 集群指南

## TKE Serverless 集群概述

最近更新时间：2024-11-14 17:25:32

### ① 说明：

Serverless 容器服务现已升级为 TKE 标准集群 + 超级节点模式。为了进一步优化用户体验和提升服务质量，我们已关闭新建 Serverless 集群的入口。如果您需要使用超级节点，请前往容器服务控制台，新建 TKE 标准集群并添加超级节点，操作详情请参见 [新建超级节点](#)。存量用户的 Serverless 集群暂不受此次变更的影响，您仍可按照现有文档进行操作。

## 什么是 TKE Serverless 集群？

TKE Serverless 集群是腾讯云容器服务推出的无需用户购买节点即可部署工作负载的服务模式。TKE Serverless 集群完全兼容原生 Kubernetes，支持使用原生方式购买及管理资源，按照容器真实使用的资源量计费。TKE Serverless 集群还扩展支持腾讯云的存储及网络等产品，同时确保用户容器的安全隔离，开箱即用。

## 相关概念

### 容器和镜像

容器是进程级别的虚拟化技术，将系统资源进行隔离和控制，让原来全局的资源仅能在容器内进程使用。镜像类似于轻量化的虚拟机快照，也可理解为容器的静态形式。镜像定义了容器运行的一切文件和依赖关系，保证了容器运行的一致性。容器技术通过把应用程序及其依赖全部打包成镜像，再使用镜像生成资源隔离的环境来运行程序，简单高效的实现了应用程序运行时的独立性和环境一致性。

### Kubernetes

Kubernetes 是 Google 基于 Borg 开源的容器编排调度引擎，是 CNCF（Cloud Native Computing Foundation）最重要的组件之一。提供了生产级别的应用编排、容器调度、服务发现、自动扩缩容等功能，详情请参见 [Kubernetes 官方文档](#)。

## 产品优势

### 原生支持

TKE Serverless 集群紧跟社区，支持最新的 Kubernetes 版本及原生的 Kubernetes 集群管理方式。以插件的形式扩展支持腾讯云系列产品，例如存储、网络、负载均衡等服务，开箱即用。

### 无服务器

TKE Serverless 集群是一种全托管的 Kubernetes 服务，意味着用户无须管理任何计算节点。TKE Serverless 集群以 Pod 的形式交付计算资源，支持用户使用 Kubernetes 原生的方式购买、退还及管理云资源。

### 安全可靠

基于腾讯云成熟的虚拟化技术和网络架构，提供99.95%以上的服务可用性。腾讯云保证用户间 TKE Serverless 集群的虚拟化隔离和网络隔离，支持用户通过安全组、网络 ACL 等产品为具体服务配置网络策略。

## 秒级伸缩

通过腾讯云自研的轻量虚拟化技术，确保更快的资源创建效率，用户可秒级创建或删除容器服务。TKE Serverless 集群支持设置 Kubernetes 原生 HPA 的方式，可让服务根据实际负载进行自动伸缩。

## 降低成本

无服务器的形态决定了 TKE Serverless 集群能为用户带来更高的资源利用率和更低的运维成本，灵活高效的弹性伸缩能力保证容器服务仅会使用当前负载需要的资源量。

## 服务集成

TKE Serverless 集群能够和腾讯云的大部分业务做到高度集成，例如存储产品云硬盘 CBS、文件存储 CFS 及对象存储 COS、云数据库 TencentDB 系列产品、私有网络 VPC 系列产品等，提供了满足各类业务需求的解决方案。

## 使用限制

详情请参见 [购买限制](#) 和 [资源规格](#)。

## 产品定价

TKE Serverless 集群是全托管的无服务器 Kubernetes 服务，提供以下几种计费模式：按量计费、包年包月、预留券和竞价模式。详情请参见 [产品定价](#)。

## 与容器服务的对比

| 特征         | TKE 标准集群                        | TKE Serverless 集群                                    |
|------------|---------------------------------|--|
| Kubernetes | 原生支持                            | 原生支持，存在由无计算节点造成的功能缺省，详情请参见 <a href="#">注意事项</a>      |
| 私有网络       | 支持                              |  |
| 计算节点       | 需要用户自主购买并管理 CVM，裸金属等计算节点        | 无需购买节点   |
| 管理方式       | 支持原生 Kubernetes API、Kubectl 等方式 |  |
| 集群         | 支持创建、管理多个集群                     |  |
| 命名空间       | 原生支持                            |  |
| 工作负载       | 原生支持                            | 支持，DaemonSet 请参见 <a href="#">超级节点上支持运行 Daemonset</a> |
| 服务         | 原生支持，集成 CLB 插件                  |  |
| 存储         | 原生支持，集成 CBS、CFS 等插件             |  |

## 应用场景

### 微服务场景

使用 TKE Serverless 集群来运行微服务，免除用户对计算节点的运维工作。服务可根据负载情况自动伸缩，使用最合理的资源量来承载应用，降低资源使用成本。

### 离线计算场景

使用 TKE Serverless 集群运行离线计算任务，只需准备容器镜像，即可快速部署任务负载。另外，TKE Serverless 集群仅收取任务真实运行时间所使用算力的费用，任务结束 Pod 自动释放即结束计费。

### 在线推理场景

TKE Serverless 集群支持使用 CPU、GPU 以及 vGPU 来运行在线推理服务，丰富的资源规格和弹性伸缩的负载，使运行服务更高效、更经济。

## 相关服务

- 存储：如需使用云硬盘或文件存储作为容器的持久化存储，可使用 [云硬盘 CBS](#) 和 [文件存储 CFS](#)。
- 网络：
  - 如需创建并管理您的私有网络，例如创建私有网络和子网、建立对等连接、使用 NAT 网关、配置路由表、配置安全策略等，可使用 [私有网络 VPC](#)。
  - 如需管理服务的内外网访问配置，可使用 [负载均衡 CLB](#)。
- API：如需使用腾讯云 API 来访问腾讯云的产品和服务，请参见 [腾讯云 API 文档](#)。

# 购买 TKE Serverless 集群 地域和可用区

最近更新时间：2024-11-14 17:25:32

## 说明：

Serverless 容器服务现已升级为 TKE 标准集群 + 超级节点模式。为了进一步优化用户体验和提升服务质量，我们已关闭新建 Serverless 集群的入口。如果您需要使用超级节点，请前往容器服务控制台，新建 TKE 标准集群并添加超级节点，操作详情请参见 [新建超级节点](#)。存量用户的 Serverless 集群暂不受此次变更的影响，您仍可按照现有文档进行操作。

## 地域

### 简介

地域（Region）是指物理的数据中心的地理区域。腾讯云不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过下表查看 TKE Serverless 集群当前支持的地域、可用区及资源类型，其他地域将相继开放。

### 相关特性

- 不同地域之间的网络完全隔离，不同地域之间的云产品默认不能通过内网通信。
- 不同地域之间的云产品，可以通过 [公网 IP](#) 访问 Internet 的方式进行通信。处于不同私有网络的云产品，可以通过 [云联网](#) 进行通信，此通信方式较为高速、稳定。
- [负载均衡](#) 当前默认支持同地域流量转发，绑定本地域的云服务器。如果开通 [跨地域绑定](#) 功能，则可支持负载均衡跨地域绑定云服务器。

## 说明：

针对地域为深圳 / 上海 / 北京金融专区的金融行业监管要求定制的合规专区，具有高安全，高隔离性的特点。目前提供云服务器、金融数据库、Redis 存储、人脸识别等服务，已认证通过的金融行业客户可 [在线咨询](#) 申请使用专区。详见 [金融专区介绍](#)。

## 可用区

### 简介

可用区（Zone）是指腾讯云在同一地域内电力和网络互相独立的物理数据中心。其目标是能够保证可用区间故障相互隔离（大型灾害或者大型电力故障除外），不出现故障扩散，使得用户的业务持续在线服务。通过启动独立可用区内的实例，用户可以保护应用程序不受单一位置故障的影响。

您可以通过 API 接口 [查询可用区列表](#) 查看完整的可用区列表。

### 相关特性

处于相同地域不同可用区，但在同一个私有网络下的云产品之间均通过内网互通，可以直接使用 [内网 IP](#) 访问。

**说明：**

内网互通是指同一账户下的资源互通，不同账户的资源内网完全隔离。

## TKE Serverless 集群支持地域

### 中国

| 地域                            | 可用区  |
|-------------------------------|--|
| 华南地区（广州）<br>ap-guangzhou      | 广州三区<br>ap-guangzhou-3   |
|                               | 广州四区<br>ap-guangzhou-4   |
|                               | 广州六区<br>ap-guangzhou-6   |
|                               | 广州七区<br>ap-guangzhou-7   |
| 华南地区（深圳金融）<br>ap-shenzhen-fsi | 深圳金融一区（仅限金融机构和企业 <a href="#">在线咨询</a> 申请开通）<br>ap-shenzhen-fsi-1 |
|                               | 深圳金融二区（仅限金融机构和企业 <a href="#">在线咨询</a> 申请开通）<br>ap-shenzhen-fsi-2 |
|                               | 深圳金融三区（仅限金融机构和企业 <a href="#">在线咨询</a> 申请开通）<br>ap-shenzhen-fsi-3 |
| 华东地区（上海）<br>ap-shanghai       | 上海二区<br>ap-shanghai-2  |
|                               | 上海三区<br>ap-shanghai-3  |
|                               | 上海四区<br>ap-shanghai-4  |
|                               | 上海五区<br>ap-shanghai-5  |
| 华东地区（上海金融）<br>ap-shanghai-fsi | 上海金融一区（仅限金融机构和企业 <a href="#">在线咨询</a> 申请开通）                      |

|                              |  |
|------------------------------|--|
|                              | ap-shanghai-fsi-1  |
|                              | 上海金融二区（仅限金融机构和企业 <a href="#">在线咨询</a> 申请开通）<br>ap-shanghai-fsi-2 |
|                              | 上海金融三区（仅限金融机构和企业 <a href="#">在线咨询</a> 申请开通）<br>ap-shanghai-fsi-3 |
| 华东地区（南京）<br>ap-nanjing       | 南京一区<br>ap-nanjing-1   |
|                              | 南京二区<br>ap-nanjing-2   |
| 华北地区（北京）<br>ap-beijing       | 北京三区<br>ap-beijing-3   |
|                              | 北京四区<br>ap-beijing-4   |
|                              | 北京五区<br>ap-beijing-5   |
|                              | 北京六区<br>ap-beijing-6   |
|                              | 北京七区<br>ap-beijing-7   |
| 华北地区（北京金融）<br>ap-beijing-fsi | 北京金融一区（仅限金融机构和企业 <a href="#">在线咨询</a> 申请开通）<br>ap-beijing-fsi-1  |
|                              | 北京金融二区（仅限金融机构和企业 <a href="#">在线咨询</a> 申请开通）<br>ap-beijing-fsi-2  |
|                              | 北京金融三区（仅限金融机构和企业 <a href="#">在线咨询</a> 申请开通）<br>ap-beijing-fsi-3  |
| 西南地区（成都）<br>ap-chengdu       | 成都一区<br>ap-chengdu-1   |
|                              | 成都二区<br>ap-chengdu-2   |
| 港澳台地区（中国香港）<br>ap-hongkong   | 香港二区（中国香港节点可用于覆盖港澳台地区）<br>ap-hongkong-2                          |

## 其他国家和地区

| 地域                         | 可用区                                       |
|----------------------------|---|
| 亚太东南（新加坡）<br>ap-singapore  | 新加坡一区（新加坡节点可用于覆盖亚太东南地区）<br>ap-singapore-1 |
|                            | 新加坡二区（新加坡节点可用于覆盖亚太东南地区）<br>ap-singapore-2 |
| 亚太东南（雅加达）<br>ap-jakarta    | 雅加达一区（雅加达节点可用于覆盖亚太东南地区）<br>ap-jakarta-1   |
| 亚太东北（首尔）<br>ap-seoul       | 首尔一区（首尔节点可用于覆盖亚太东北地区）<br>ap-seoul-1       |
|                            | 首尔二区（首尔节点可用于覆盖亚太东北地区）<br>ap-seoul-2       |
| 亚太东北（东京）<br>ap-tokyo       | 东京二区（东京节点可用区覆盖亚太东北地区）<br>ap-tokyo-2       |
| 亚太东南（曼谷）<br>ap-bangkok     | 曼谷一区（曼谷节点用户覆盖亚太东南地区）<br>ap-bangkok-1      |
| 美国东部（弗吉尼亚）<br>na-ashburn   | 弗吉尼亚一区（弗吉尼亚节点用户覆盖美国东部地区）<br>na-ashburn-1  |
|                            | 弗吉尼亚二区（弗吉尼亚节点用户覆盖美国东部地区）<br>na-ashburn-2  |
| 欧洲地区（法兰克福）<br>eu-frankfurt | 法兰克福一区（法兰克福节点可用于覆盖欧洲地区）<br>eu-frankfurt-1 |

# 计费概述

最近更新时间：2024-11-14 17:25:32

## 说明：

Serverless 容器服务现已升级为 TKE 标准集群 + 超级节点模式。为了进一步优化用户体验和提升服务质量，我们已关闭新建 Serverless 集群的入口。如果您需要使用超级节点，请前往容器服务控制台，新建 TKE 标准集群并添加超级节点，操作详情请参见 [新建超级节点](#)。存量用户的 Serverless 集群暂不受此次变更的影响，您仍可按照现有文档进行操作。

## 计费模式

TKE Serverless 集群以超级节点维度承载资源，超级节点提供以下几种计费模式：按量计费、包年包月、预留券和竞价模式，分别适用于不同场景下的用户需求。详情可参见 [计费模式说明](#)。

## 计费方式

TKE Serverless 集群中包年包月的超级节点会根据购买资源的总规格进行计费，按量计费的超级节点会根据节点上实际调度的 Pod 规格及运行时长进行计费，具体计算方式请参见 [产品定价](#)。

## 其他费用

如在使用 TKE Serverless 集群时使用到 [负载均衡 CLB](#)、[云硬盘 CBS](#)、[文件存储 CFS](#) 等其他收费产品时，按原产品计费原则计费，具体细节请参考各产品购买指南。

# 购买限制

最近更新时间：2024-11-14 17:25:32

## 说明：

Serverless 容器服务现已升级为 TKE 标准集群 + 超级节点模式。为了进一步优化用户体验和提升服务质量，我们已关闭新建 Serverless 集群的入口。如果您需要使用超级节点，请前往容器服务控制台，新建 TKE 标准集群并添加超级节点，操作详情请参见 [新建超级节点](#)。存量用户的 Serverless 集群暂不受此次变更的影响，您仍可按照现有文档进行操作。

## 使用要求

在使用 TKE Serverless 集群前，您需要 [注册腾讯云账号](#)，并完成 [实名认证](#)。

## 支持地域

TKE Serverless 集群支持的地域请参见 [地域和可用区](#)，资源规格信息请参考 [资源规格](#)。

## 资源配额

### 集群与 Pod 限制

| 资源               | 限制（个） | 说明                            |
|------------------|-------|-------------------------------|
| 同一地域内最大集群数量      | 5     | 包括创建中、运行中状态的集群                |
| 同一集群最大 Pod 规模    | 500   | 包括所有 Namespace、所有负载、任何状态的 Pod |
| 单个工作负载最大 Pod 副本数 | 100   | 包括负载内任何状态的 Pod                |
| 同一地域最大容器实例规模     | 500   | 包括任何状态的容器实例                   |

### 其他相关限制

当使用 TKE Serverless 时，您的所有 Pod 都是云上独立的计算、网络实例，等同于一台云服务器实例，故还会受到下述约束：

- 创建工作负载时，工作负载的每个 Pod 会默认关联一个 [安全组](#)，且同一个工作负载的所有 Pod 副本会关联同一个安全组，每一个 Pod 对安全组来说等同于一台云服务器实例，此时会受 [单个安全组关联的云服务器实例数](#) 的配额约束。
- 当使用 CLB 类型 Service 时，每一个绑定 CLB 的 Pod 对该 CLB 来说等同于一台云服务器实例，此时会受 [一个负载均衡实例的转发规则可绑定的服务器数量](#) 的配额约束。

### 申请提升配额操作指引

若您需要超过以上配额的资源，可填写提升配额申请，由腾讯云对您的实际需求进行评估，评估通过之后将为您提升配额。

- 请 [提交工单](#)，选择 [其他问题](#) > [立即创建](#)，进入创建工单信息填写页面。

2. 在问题描述中填写“期望提升 TKE Serverless 相关配额”，注明目标地区、提高配额的对象、目标配额，并按照页面提示填写您可用的手机号等信息。
3. 填写完成后，单击**提交工单**即可。

# 欠费说明

最近更新时间：2025-04-03 10:53:43

## 说明：

Serverless 容器服务现已升级为 TKE 标准集群 + 超级节点模式。为了进一步优化用户体验和提升服务质量，我们已关闭新建 Serverless 集群的入口。如果您需要使用超级节点，请前往容器服务控制台，新建 TKE 标准集群并添加超级节点，操作详情请参见 [新建超级节点](#)。存量用户的 Serverless 集群暂不受此次变更的影响，您仍可按照现有文档进行操作。

## 欠费说明

TKE Serverless 集群当前仅对集群内资源进行计费，其中包含后付费资源和预付费资源，从您的**账户欠费**（账户余额被扣为负值时刻）起：

- 后付费资源（按量计费 Pod）将在下一个计费周期立即停服并释放底层资源。
- 预付费资源（包月超级节点、包月预留券）将不受账户欠费影响，包月超级节点仍可调度 Pod，包月预留券仍可抵扣同地域同规格的按量计费 Pod。

## 欠费处理

TKE Serverless 集群本身暂不收取集群管理费用，欠费后集群的控制面仍然正常工作，仅对用户的数据资源层面进行处理：

## 注意：

下表仅列出了 TKE Serverless 计算资源的处理，集群中的 **CLB** 等资源的欠费处理逻辑请参考对应产品的欠费说明。

| 资源类型  | 说明  |
|-------|---|
| 集群控制面 | 正常工作  |
| 后付费资源 | 在下一个计费周期立即停服并释放底层资源，用户 YAML 等数据资源不会删除，待账号不再为欠费状态后，集群内 Pod 可重新创建（若 Pod 的 restartPolicy 指定为“Always”）。 |
| 预付费资源 | 预付费资源将正常使用至预付费资源的生命周期结束，预付费的超级节点上仍可调度规则内的 Pod，预付费的预留券仍可用于抵扣规则内的 Pod，但超出预付费范围外的资源将进入 Pending 状态。     |

# TKE Serverless 集群管理

## 创建集群

最近更新时间：2024-11-13 11:32:21

### 说明：

Serverless 容器服务现已升级为 TKE 标准集群 + 超级节点模式。为了进一步优化用户体验和提升服务质量，我们已关闭新用户新建 Serverless 集群的入口。如果您需要使用超级节点，请前往容器服务控制台，新建标准集群并添加超级节点，操作详情请参见 [新建超级节点](#)。存量用户的 Serverless 集群将不受此次变更的影响，您仍可按照现有文档进行操作。

本文介绍通过腾讯云容器服务控制台创建 TKE Serverless 集群。

## 前提条件

请前往 [CAM 管理控制台](#) 开通相应的服务权限。

## 操作步骤

1. 登录容器服务控制台，选择左侧导航栏中的 [集群](#)。
2. 单击集群列表上方的新建。
3. 在集群类型中，选择 **Serverless 集群类型**，单击创建。

#### 4. 在创建集群页面，根据以下提示设置集群信息。如下图所示：

集群名称

Kubernetes版本

所在地域  上海 北京 成都 中国香港 新加坡 曼谷 孟买 首尔 东京 弗吉尼亚 法兰克福 南京 雅加达 多伦多

处在不同地域的云产品内网不通，购买后不能更换。建议选择靠近您客户的地域，以降低访问延时、提高下载速度。

集群网络

如现有的网络不合适，您可以去控制台[新建私有网络](#)

超级节点配置

可用区  广州四区 广州六区 广州七区

计费模式   New

容器网络

| 子网ID | 子网名称 | 可用区 | 剩余IP数 |
|------|------|-----|-------|
| 暂无数据 |      |     |       |

Pod 会直接占用所选子网 IP，请尽量选择 IP 数量充足且与其他产品使用无冲突的子网。如现有的子网不合适，您可以去控制台[新建子网](#)

节点名称

Service CIDR

指定Kubernetes Service 分配的IP段，不能与VPC网段冲突

集群描述

Prometheus 监控服务   [创建 TMP 实例](#)

当前 VPC 没有TMP实例，请先[创建](#)，或在[TMP控制台](#)进行跨地域关联

[高级设置](#)

- **集群名称：**创建的 Serverless 集群名称，不超过60个字符。
- **Kubernetes版本：** Serverless 集群支持1.16以上的多个 Kubernetes 版本选择，各版本特性对比请查看 [Supported Versions of the Kubernetes Documentation](#)。
- **所在地域：**建议您根据所在地理位置选择靠近的地域，可降低访问延迟，提高下载速度。
- **集群网络：**该字段展示所在地域下您已创建的全部 VPC 网络，请选择合适的网络作为集群网络，若无可用网络信息或现有网络不合适，可跳转至控制台新建私有网络，详情请参见 [私有网络（VPC）](#)，私有网络中的子网将用于 serverless 集群的容器网络。
- **超级节点配置：**

#### 按量计费

- **可用区：**选择超级节点所在的可用区。
- **计费模式：**全地域支持按量计费模式。
- **容器网络：**为集群内容器分配在容器网络地址范围内的 IP 地址。集群内超级节点的 Pod 会直接占用 VPC 子网 IP（剩余 IP 数将限制调度至超级节点上的 Pod 数），请尽量选择 IP 数量充足且与其他产品使用无冲突的子网。

超级节点上的 Pod 会直接运行在用户已指定的 VPC 网络上，每个 Pod 在生命周期内都会绑定一个指定 VPC 内的弹性网卡。您可前往 [弹性网卡列表](#) 查看 Pod 关联的网卡。

按量计费模式下，支持选择多个子网，每个子网对应创建一个按量计费的超级节点，按量计费的超级节点创建时不计费，当实际创建工作负载后 Pod 调度至按量计费超级节点时开始按照 Pod 规格和实际运行时长计费。

#### ⚠ 注意：

- 建议为容器网络配置多个可用区，这样您的工作负载在部署时会自动打散分布在多个可用区，可用性更高。
- 请确保为容器网络分配 IP 充足的子网，避免创建大规模工作负载时因为 IP 资源耗尽无法创建 Pod。

## 包年包月

- **可用区：**选择超级节点所在的可用区。
- **计费模式：**北京、上海、广州、南京支持包年包月计费模式。
- **容器网络：**为集群内容器分配在容器网络地址范围内的 IP 地址。集群内超级节点的 Pod 会直接占用 VPC 子网 IP（剩余 IP 数将限制调度至超级节点上的 Pod 数），请尽量选择 IP 数量充足且与其他产品使用无冲突的子网。超级节点上的 Pod 会直接运行在用户已指定的 VPC 网络上，每个 Pod 在生命周期内都会绑定一个指定 VPC 内的弹性网卡。您可前往 [弹性网卡列表](#) 查看 Pod 关联的网卡。

包年包月计费模式支持选择单个子网，如下图所示：

超级节点配置

可用区： 广州三区  广州四区  广州六区  广州七区

计费模式： 按量计费  包年包月 New

| 子网ID  | 子网名称     | 可用区  | 剩余IP数 |                                  |
|---|----------|------|-------|----------------------------------|
| <input checked="" type="radio"/> subnet-... | sub2     | 广州三区 | 241   | 支持 INTEL_AMD,GPU\NVIDIA 1/4 T... |
| <input type="radio"/> subnet-...            | test-gzs | 广州三区 | 245   | 支持 INTEL_AMD,GPU\NVIDIA 1/4 T... |

Pod 会直接占用所选子网 IP，请尽量选择 IP 数量充足且与其他产品使用无冲突的子网。如现有的子网不合适，您可以去控制台 [新建子网](#)

节点名称：

CPU 总规格：

CPU 内存比：

购买时长： 1个月  2个月  3个月  4个月  5个月  6个月  1年  2年  3年 [其他时长](#)

自动续费： 账户余额足够时，设备到期后自动续费

[添加节点](#)

- **CPU 总规格：**可创建规格50-5000，总规格表示该包年包月超级节点上可调度的 Pod 的 CPU 总核数。
- **CPU 内存比：**可选择1:2和1:4规格的 CPU 内存比，这里指该包年包月超级节点上的 CPU 与内存总规格的比值，非超级节点上调度的 Pod 的 CPU 内存比，包年包月超级节点支持调度 CPU 内存比为1:1到1:4规格的

Pod。

- **购买时长：**包年包月超级节点的购买时长。
- **自动续费：**支持开启包月超级节点自动续费能力。

**⚠ 注意：**

- 建议为容器网络配置多个可用区，这样您的工作负载在部署时会自动打散分布在多个可用区，可用性更高。
- 请确保为容器网络分配 IP 充足的子网，避免创建大规模工作负载时因为 IP 资源耗尽无法创建 Pod。

- **Service CIDR：**集群的 ClusterIP Service 默认分配在所选 VPC 子网中，请尽量选择 IP 数量充足且与其他产品使用无冲突的子网。
- **集群描述：**创建集群的相关信息，该信息将显示在“集群信息”页面。
- **高级设置：**
  - **CoreDNS：**会自动在集群命名空间 kube-system 中部署 2 副本的 Deployment:coredns，该服务默认不收取费用，同时不建议进行修改。
  - **标签：**腾讯云标签服务，用于从不同维度对资源分类管理。

5. 单击**完成**即可开始创建，可在集群列表页面查看集群的创建进度。

# 连接集群

最近更新时间：2025-04-17 17:36:02

## 说明：

Serverless 容器服务现已升级为 TKE 标准集群 + 超级节点模式。为了进一步优化用户体验和提升服务质量，我们已关闭新建 Serverless 集群的入口。如果您需要使用超级节点，请前往容器服务控制台，新建 TKE 标准集群并添加超级节点，操作详情请参见 [新建超级节点](#)。存量用户的 Serverless 集群暂不受此次变更的影响，您仍可按照现有文档进行操作。

## 操作场景

您可以通过 Kubernetes 命令行工具 Kubectl 从本地客户端机器连接到 TKE Serverless 集群。本文档指导您如何连接集群。

## 说明：

开启集群公网访问前，请确保您的集群内存在节点资源。

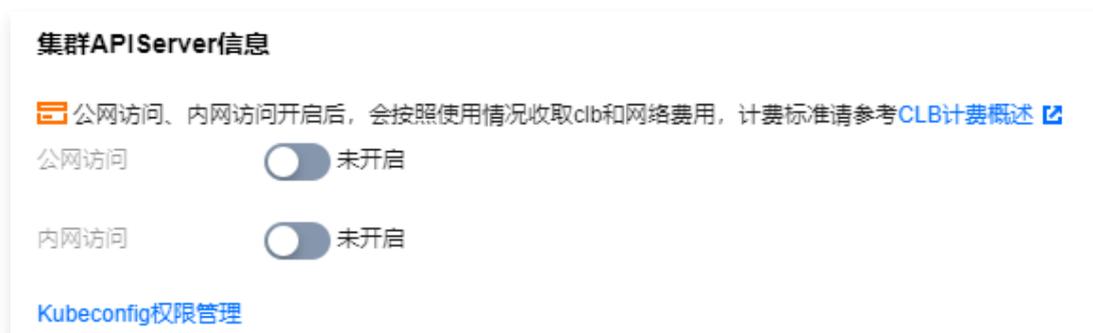
## 方案一：通过 Cloud Shell 连接集群

TKE Serverless 集成了腾讯云 [Cloud Shell](#)，您可以在腾讯云控制台上实现一键连接集群的能力，通过 kubectl 实现对集群的灵活操作。

## 操作步骤

### 步骤1：开启集群公网访问

1. 登录容器服务控制台，选择左侧导航栏中的 [集群](#)。
2. 在[集群管理](#)页面，选择集群所在地域，单击目标集群 ID，进入集群详情页。
3. 在[集群基本信息](#)页面，查看集群访问开启状态，如下图所示：



4. 单击  开启公网访问。开启公网访问时，需配置相关参数，如：
  - **安全组**：开启公网访问后，会自动分配一个外网 CLB 作为访问端口。您可以通过安全组来配置来源授权，我们会将安全组绑定到外网 CLB 上，以达到访问控制的效果。集群访问代理的流量默认走443端口，请确认您选择的安全组内针对来源 IP 已放通443端口，以确保开启功能后集群可以正常访问。

- **运营商类型、网络计费模式、带宽上限**：CLB 相关参数，详情请参见 [CLB 创建指南](#)，根据实际需求进行设置。
- **访问方式**：选择公网域名后，您需要传入自定义域名，我们会为您传入的域名进行安全签名，您需要自行配置公网解析。选择 CLB 默认域名后，您无需再手动配置域名解析等操作。

5. 单击**保存**。确认公网访问已开启。

## 步骤2：使用 Cloud Shell 连接集群

1. 登录容器服务控制台，选择左侧导航栏中的 **集群**。
2. 在**集群管理**页面，选择集群所在地域，单击目标集群右侧的**更多 > 连接集群**，如下图所示：



3. 在控制台下方出现 **Cloud Shell** 入口，您可以直接在命令框里面输入 `kubectl` 指令，如下图所示：



## 方案二：通过本地计算机连接集群

### 前提条件

请安装 [curl](#)。

### 操作步骤

#### 步骤1：安装 Kubectl 工具

1. 参考 [Installing and Setting up kubectl](#)，安装 Kubectl 工具。您可根据操作系统的类型，选择获取 Kubectl 工具的方式：

**⚠ 注意:**

- 如果您已经安装 Kubectl 工具，请忽略本步骤。
- 请根据实际需求，将命令行中的“v1.18.4”替换成业务所需的 Kubectl 版本。客户端的 Kubectl 与服务端的 Kubernetes 的最高版本需保持一致，您可以在基本信息的“集群信息”模块里查看 Kubernetes 版本。

**Mac OS X 系统**

执行以下命令，获取 Kubectl 工具：

```
curl -LO https://storage.googleapis.com/kubernetes-  
release/release/v1.18.4/bin/darwin/amd64/kubectl
```

**Linux 系统**

执行以下命令，获取 Kubectl 工具：

```
curl -LO https://storage.googleapis.com/kubernetes-  
release/release/v1.18.4/bin/linux/amd64/kubectl
```

**Windows 系统**

执行以下命令，获取 Kubectl 工具：

```
curl -LO https://storage.googleapis.com/kubernetes-  
release/release/v1.18.4/bin/windows/amd64/kubectl.exe
```

2. 此步骤以 Linux 系统为例。执行以下命令，添加执行权限。

```
chmod +x ./kubectl  
sudo mv ./kubectl /usr/local/bin/kubectl
```

3. 执行以下命令，测试安装结果。

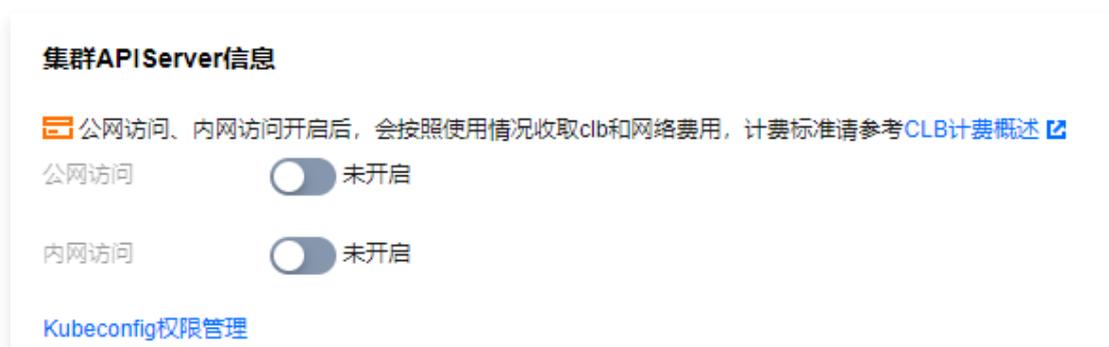
```
kubectl version
```

如若输出类似以下版本信息，即表示安装成功。

```
Client Version: version.Info{Major:"1", Minor:"5", GitVersion:"v1.5.2",
GitCommit:"08e099554f3c31f6e6f07b448ab3ed78d0520507", GitTreeState:"clean",
BuildDate:"2017-01-12T04:57:25Z", GoVersion:"go1.7.4", Compiler:"gc",
Platform:"linux/amd64"}
```

## 步骤2：开启集群访问

1. 登录容器服务控制台，选择左侧导航栏中的 **集群**。
2. 在**集群管理**页，选择集群所在地域，单击目标集群 ID，进入集群详情页。
3. 在集群基本信息页，查看集群访问开启状态，如下图所示：



### 开启公网访问

开启公网访问时，需配置相关参数：

- **安全组**：开启公网访问后，会自动分配一个外网 CLB 作为访问端口。您可以通过安全组来配置来源授权，我们会将安全组绑定到外网 CLB 上，以达到访问控制的效果。集群访问代理的流量默认走443端口，请确认您选择的安全组内针对来源 IP 已放通443端口，以确保开启功能后集群可以正常访问。
- **运营商类型、网络计费模式、带宽上限**：CLB 相关参数，详情请参见 [CLB 创建指南](#)，根据实际需求进行设置。
- **访问方式**：选择公网域名后，您需要传入自定义域名，我们会为您传入的域名进行安全签名，您需要自行配置公网解析。选择 CLB 默认域名后，您无需再手动配置域名解析等操作。

### 开启内网访问

开启内网访问时，需配置相关参数：

- **负载均衡器**：
  - **自动创建**：自动创建CLB用于公网/内网访问 Service，CLB 的生命周期由 TKE 管理。

- 使用已有：使用当前未被TKE使用的应用型 CLB 用于公网/内网访问 Service，请勿手动修改由 TKE 创建的 CLB 监听器。
- **LB 所在子网**：默认不开启。开启内网访问时，需配置一个子网，开启成功后将在已配置的子网中分配 IP 地址。
- **访问方式**：选择内网域名后，您需要传入自定义域名，我们会为您传入的域名进行安全签名，您需要自行配置内网解析。选择内网 IP 后，我们会为您分配内网 IP 并安全签名。

### 步骤3：获取 KubeConfig

TKE 提供了两种 KubeConfig，分别用于公网访问和内网访问。开启集群访问后，即可按如下步骤获取对应的 KubeConfig：

1. 在**集群详情 > 基本信息**中，查看“**集群 API Server 信息**”。
2. 在对应集群访问的开关下方，复制或下载 **KubeConfig**，或查看外网访问安全组、访问域名（开启访问时配置）、访问 IP。如下图所示：



### 步骤4：配置 KubeConfig 并访问 Kubernetes 集群

1. 根据实际情况进行集群凭据配置。
1. 配置前，请判断当前访问客户端是否已经配置过任何集群的访问凭证：
  - 否，即 `~/.kube/config` 文件内容为空，可直接复制已获取的 KubeConfig 访问凭证内容并粘贴入 `~/.kube/config` 中。若客户端无 `~/.kube/config` 文件，您可直接创建。
  - 是，您可下载已获取的 KubeConfig 至指定位置，并依次执行以下命令以合并多个集群的 config。

```
KUBECONFIG=~/.kube/config:~/Downloads/cls-3jju4zdc-config kubectl config view --merge --flatten > ~/.kube/config
```

**说明：**

此处需要将文件下载到需要连接集群的客户端内，并且注意文件名。

```
export KUBECONFIG=~/.kube/config
```

其中，`~/Downloads/cls-3jju4zdc-config` 为本集群的 KubeConfig 的文件路径，请替换为下载至本地后的实际路径。

**注意：**

您需要修改集群的 kubeconfig 文件中的 user 和 name，以确保每个集群都有唯一的配置，这样才能保证切换集群后的正常访问。

2. 完成 KubeConfig 配置后，依次执行以下命令查看并切换 context 以访问本集群。

```
kubectl config get-contexts
```

**说明：**

下面例子中的“cls-3jju4zdc-context-default”为上一步 `kubectl config get-contexts` 中查到的集群名称 NAME。

```
kubectl config use-context cls-3jju4zdc-context-default
```

3. 执行以下命令，测试是否可正常访问集群。

```
kubectl get node
```

成功响应示例：

```
[root@ ~]# kubectl get node
NAME                STATUS    ROLES    AGE   VERSION
node1                Ready    <none>   27m   v1.26.1-tke.1
node2                Ready    <none>   27m   v1.26.1-tke.1
```

如果无法连接请查看是否已经开启公网访问或内网访问入口，并确保访问客户端在指定的网络环境内。

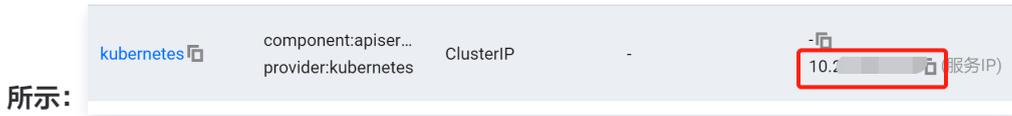
## 方案三：通过集群内节点连接集群

### 操作步骤

#### 步骤1：获取 Kubernetes 的 service IP

1. 登录容器服务控制台，选择左侧导航栏中的 **集群**。

2. 在**集群管理**页面，选择集群所在地域，单击目标集群 ID，进入集群详情页。
3. 在集群详情页中，选择左侧的**服务与路由** > **Service** 获取 default 命名空间下 Kubernetes 的 service IP，如下图



### 说明：

Kubernetes service 是 ClusterIP 模式，仅适用于集群内访问。

## 步骤2：配置 KubeConfig 并访问 Kubernetes 集群

1. 调用云 API 接口 [DescribeClusterKubeconfig](#) 获取当前使用账号的凭证信息 Kubeconfig 文件。
2. 使用步骤1中获取的 kubernetes service ip，替换通过 API 获取的 Kubeconfig 中的 clusters.cluster.server 字段为 https://<IP>:443，并整理格式如下图所示：



3. 登录任意一台集群内节点，使用上步生成的 Kubeconfig 内容替换节点上 `~/.kube/config` 文件内容。
4. 执行以下命令，测试是否可正常访问集群。

```
kubectl get node
```

成功响应式例：

```
[root@... .kube]# kubectl get node
NAME                STATUS    ROLES    AGE   VERSION
...                Ready    <none>   27m   v1.26.1-tke.1
...                Ready    <none>   27m   v1.26.1-tke.1
```

## 相关说明

### Kubectl 命令行介绍

Kubectl 是一个用于 Kubernetes 集群操作的命令行工具。本文涵盖 kubectl 语法、常见命令操作并提供常见示例。有关每个命令（包括所有主命令和子命令）的详细信息，请参见 [kubectl 参考文档](#) 或使用 `kubectl help` 命令查看详细帮助。

# 集群生命周期

最近更新时间：2025-03-28 14:51:42

## 说明：

Serverless 容器服务现已升级为 TKE 标准集群 + 超级节点模式。为了进一步优化用户体验和提升服务质量，我们已关闭新建 Serverless 集群的入口。如果您需要使用超级节点，请前往容器服务控制台，新建 TKE 标准集群并添加超级节点，操作详情请参见 [新建超级节点](#)。存量用户的 Serverless 集群暂不受此次变更的影响，您仍可按照现有文档进行操作。

## 集群生命周期说明

| 状态  | 说明                             |
|-----|--------------------------------|
| 创建中 | TKE Serverless 集群正在创建，正在申请云资源。 |
| 运行中 | 集群正常运行。                        |
| 闲置中 | 集群处于闲置状态。                      |
| 激活中 | 集群从闲置状态中调整到运行中状态的过程。           |
| 删除中 | 集群在删除中，将同步销毁和释放云资源。            |
| 异常  | 集群中存在异常，如网络不可达等。               |

## 集群闲置状态说明

为了减少资源空置成本，引导用户更合理地规划和管理集群资源，TKE Serverless 上线了闲置集群回收功能，将没有存量 Pod，且连续7天没有创建/删除 Pod 的集群视为闲置集群，被标记为闲置集群的 TKE Serverless 集群将展示为“闲置中”状态。

闲置集群定义条件：

- 没有存量的由用户侧创建的 Pod。
- 连续7天没有 Pod 创建和删除。
- 距离上次集群闲置变正常状态时间间隔至少3天。
- 距离新建集群时间超过7天。

状态为“闲置中”的集群，会保留原集群的所有信息，但集群为不可用状态，不支持查看集群监控，不支持查看集群详情，并限制了 APIServer 相关操作，您可通过单击操作栏中**更多** > **激活唤醒集群**，也可以单击**删除**删除集群。

闲置集群激活后集群与原集群功能和配置完全一致。需要注意的是，如果重新被激活的集群连续3天保持没有 Pod 的状态，且没有创建/删除 Pod 操作，会再次被标记为闲置集群。

# 注意事项

最近更新时间：2025-03-28 14:51:42

## ❗ 说明：

Serverless 容器服务现已升级为 TKE 标准集群 + 超级节点模式。为了进一步优化用户体验和提升服务质量，我们已关闭新建 Serverless 集群的入口。如果您需要使用超级节点，请前往容器服务控制台，新建 TKE 标准集群并添加超级节点，操作详情请参见 [新建超级节点](#)。存量用户的 Serverless 集群暂不受此次变更的影响，您仍可按照现有文档进行操作。

## 计费方式

TKE Serverless 集群是付费产品，采取按量计费模式。计费详情请参见 [计费概述](#)、[产品定价](#)、[购买限制](#)。

## Pod 规格配置

Pod 的规格配置是容器运行时可用资源和使用服务计费的依据，请务必了解 TKE Serverless 集群支持的 Pod 规格配置和指定方法，详情请参见 [资源规格](#) 和 [指定资源规格](#)。

## Pod 临时存储

每个 Pod 创建时会分配不超过20GiB的临时镜像存储。

## ⚠ 注意：

- 临时镜像存储将于 Pod 生命周期结束时删除，请勿用于存储重要数据。
- 由于存储镜像，实际可用空间小于20GiB。
- 重要数据、超大文件等推荐挂载 Volume 持久化存储。

## Kubernetes 版本

不支持1.14以下版本。

## 其他说明

TKE Serverless 集群没有 Node，所以不支持部分依赖 Node 组件，例如 Kubelet、Kube-proxy 的功能。

## Node

- 暂不支持添加、管理物理节点。
- TKE Serverless 集群的节点会被“超级节点”替代，每个超级节点对应一个容器网络中指定的 VPC 子网。
- 超级节点也支持节点亲和性、污点、封锁等调度操作。
- 超级节点上可调度的 Pod 数只受对应 VPC 子网的 IP 数限制。

## 容器网络

- TKE Serverless 集群的容器网络采用的是与云服务器、云数据库等云产品平级的 VPC 网络，集群中的每个 Pod 都会占用一个 VPC 子网 IP。
- Pod 与 Pod、Pod 与其他同 VPC 云产品间直接通过 VPC 网络通信，没有性能损耗。

## Pod 隔离性

TKE Serverless 集群中的 Pod 拥有与云服务器完全一致的安全隔离性。Pod 在腾讯云底层物理服务器上调度创建，创建时会通过虚拟化技术保证 Pod 间的资源隔离。

## 节点内核

仅支持自定义 net 开头的内核参数。

## Service

TKE Serverless 集群没有真实节点，部署 NodePort 类型的服务不会生效。

## Volume

支持 Hostpath 类型的数据卷，详情请参见 [存储卷说明](#)。

由于 TKE Serverless 集群没有节点，虽然集群依旧兼容 Pod 中与 Host 相关的参数（例如 Hostpath、Hostnetwork: true、DnsPolicy: ClusterFirstWithHostNet 等），但不一定能满足预期效果。例如您希望使用 Hostpath 共享数据，但可能调度到同一个超级节点上的两个 Pod 实际为不同子机的 Hostpath。因此建议您在 TKE Serverless 集群运行的任务，不要强依赖与 Host 相关的参数。

## 端口限制

在 TKE Serverless 集群中，9100 端口被保留，不可使用。

# Kubernetes 对象管理

## 工作负载管理

最近更新时间：2025-04-01 16:11:02

### 说明：

Serverless 容器服务现已升级为 TKE 标准集群 + 超级节点模式。为了进一步优化用户体验和提升服务质量，我们已关闭新建 Serverless 集群的入口。如果您需要使用超级节点，请前往容器服务控制台，新建 TKE 标准集群并添加超级节点，操作详情请参见 [新建超级节点](#)。存量用户的 Serverless 集群暂不受此次变更的影响，您仍可按照现有文档进行操作。

本文介绍如何在 TKE Serverless 集群中选择多种工作负载形式来运行您的服务。

### 注意

- Serverless 集群没有 Node，工作负载在创建时会根据参数设置为每个 Pod 分配实际的资源。详情请参见 [指定资源规格](#)。
- 如果您需要通过 yaml 来创建、管理您的 Kubernetes 对象，可通过指定 annotation 完成。详情请参见 [Annotation 说明](#)。

## 工作负载类型介绍

### Deployment

Deployment 声明了 Pod 的模板和控制 Pod 的运行策略，适用于部署无状态的应用程序。您可以根据业务需求，对 Deployment 中运行的 Pod 的副本数、调度策略、更新策略等进行声明。

### StatefulSet

StatefulSet 主要用于管理有状态的应用，可以创建具有持久性标识符的 Pod。Pod 迁移或销毁重启后，标识符仍会保留。在需要持久化存储时，您可以通过标识符对存储卷进行一一对应。如果应用程序不需要持久的标识符，建议您使用 Deployment 部署应用程序。

### Job

Job 控制器会创建 1 - N 个 Pod，这些 Pod 按照运行规则运行，直至运行结束。Job 可用于批量计算和数据分析等场景，它可以通过设置重复执行次数、并行度、重启策略等参数来满足业务需求。

Job 运行完毕后，将不再创建新的 Pod，仅存在原 Pod 运行记录，但底层真实资源已释放，因此运行结束后无法在控制台“日志”中查询到对应 Pod 的日志。

### CronJob

一个 CronJob 对象类似于 crontab ( cron table ) 文件中的一行。它根据指定的预定计划周期性地运行一个 Job，格式可以参考 Cron。

Cron 格式说明如下:

```
# 文件格式说明
# —分钟 (0 - 59)
# | —小时 (0 - 23)
# | | —日 (1 - 31)
# | | | —月 (1 - 12)
# | | | | —星期 (0 - 6)
# | | | | |
# * * * * *
```

## 前提条件

- 已创建状态为“运行中”的 Serverless 集群，详情请参见 [创建集群](#)。
- 集群有合适的且为 Active 状态的命名空间。

## 操作步骤

1. 登录 [容器服务控制台](#)，选择左侧导航栏中的**集群**。
2. 在集群管理页面，选择 Serverless 集群 ID，进入该集群基本信息页面。
3. 在左侧导航中选择工作负载类型，单击**新建**。
4. 在新建工作负载页面，填写工作负载名，并根据实际需求设置参数。
  - 各类型工作负载的具体参数设置请参考：
    - [Deployment 管理](#)
    - [StatefulSet 管理](#)
    - [CronJob 管理](#)
    - [Job 管理](#)
  - 其他操作指引请参考：
    - [设置工作负载的资源限制](#)
    - [设置工作负载的调度规则](#)
    - [设置工作负载的健康检查](#)
    - [设置工作负载的运行命令和参数](#)

# 服务管理

最近更新时间：2023-05-17 15:40:45

## 操作场景

本文档介绍工作负载如何通过 Service 和 Ingress 对外暴露服务入口。

## 服务类型介绍

### Service

Service 定义访问后端 Pod 的访问策略，提供固定的虚拟访问 IP。您可以通过 Service 负载均衡地访问到后端的 Pod。

Service 支持以下类型：

- 公网访问：使用 Service 的 Loadbalance 模式，自动创建公网 CLB。公网 IP 可直接访问到后端的 Pod。
- 集群内访问：使用 Service 的 ClusterIP 模式，用于集群内访问。
- VPC 内网访问：使用 Service 的 Loadbalance 模式，自动创建内网 CLB。指定

```
annotations:service.kubernetes.io/qcloud-loadbalancer-internal-subnetid: subnet-xxxxxxx
```

，VPC 内网即可通过内网 IP 直接访问到后端的 Pod。

#### ⚠ 注意

当 Service 类型为公网访问时，默认不为该 Service 开启 ClusterIP。可通过在 yaml 中增加如下 annotations 开启 ClusterIP：

```
service.kubernetes.io/qcloud-clusterip-loadbalancer-subnetid: #Service  
CIDR使用的子网ID
```

## Ingress

Ingress 是允许访问到集群内 Service 的规则集合，您可以通过配置转发规则，实现不同 URL 可以访问到集群内不同的 Service。

为了使 Ingress 资源正常工作，集群必须运行 Ingress-controller。腾讯云容器服务在集群内默认启用了基于腾讯云负载均衡器实现的 `l7-lb-controller`，支持 HTTP、HTTPS 及 nginx-ingress 类型，您可以根据您的业务需要选择不同的 Ingress 类型。

## 前提条件

- 已创建状态为“运行中”的 Serverless 集群，详情请参见 [创建集群](#)。
- 集群有合适的且为 Active 状态的命名空间。

## 操作步骤

具体操作请参考 [Service 管理](#) 和 [Ingress 管理](#)。

## 注意事项

- Serverless 集群内创建 ClusterIP Service 会占用 Service CIDR 所选子网 IP，请确保该子网剩余 IP 足够。
- Serverless 集群 Service 创建的 CLB 会直接绑定 Endpoint 内所有 Pod 的弹性网卡。
- Serverless 集群 Service 仅支持 [应用型负载均衡](#)。
- 使用已有 CLB 创建 Service 时，仅支持使用当前未创建监听器的 CLB。

# 其他资源管理

最近更新时间：2025-04-01 16:11:02

## 说明：

Serverless 容器服务现已升级为 TKE 标准集群 + 超级节点模式。为了进一步优化用户体验和提升服务质量，我们已关闭新建 Serverless 集群的入口。如果您需要使用超级节点，请前往容器服务控制台，新建 TKE 标准集群并添加超级节点，操作详情请参见 [新建超级节点](#)。存量用户的 Serverless 集群暂不受此次变更的影响，您仍可按照现有文档进行操作。

本文档介绍通过腾讯云容器服务控制台管理其他 Kubernetes 资源，例如命名空间、配置、存储等。

## 前提条件

已创建状态为“运行中”的 Serverless 集群，详情请参见 [创建集群](#)。

## 操作步骤

1. 登录容器服务控制台，选择左侧导航栏中的 [集群](#)。
2. 在集群管理页面，单击 Serverless 集群 ID。
3. 在集群详情页面，您可参考以下文档进行其他资源管理：
  - 命名空间的操作管理请参考 [Namespaces](#)。
  - 自动伸缩的操作管理请参考 [自动伸缩](#) 和 [自动伸缩指标说明](#)。
  - 配置资源的操作管理请参考 [ConfigMap 管理](#) 和 [Secret 管理](#)。
  - 存储资源的操作管理请参考 [存储管理概述](#)。

# Annotation 说明

最近更新时间：2023-05-17 15:40:26

通过在 YAML 文件中定义 Annotation（注解）的方式，可以实现超级节点丰富的自定义能力。您可以从 [Annotation 说明](#) 中了解更多通过注解对超级节点进行常见配置的操作。

# Serverless 集群全局配置说明

最近更新时间：2023-07-03 10:34:21

## 操作场景

TKE Serverless 集群支持通过 configmap 进行全局配置。在 TKE 弹超级节点场景以及纯 TKE Serverless 集群场景下，如果用户需要批量对每个超级节点或每个 Pod 设置 annotation，此时在超级节点维度或 Pod 维度进行配置会相对繁琐，对业务 yaml 的侵入性也较大，因此 TKE Serverless 集群提供全局配置的能力，用户可以通过 configmap 进行全局配置来实现对集群内每个 Pod 注入 Annotation 的能力。

## 操作步骤

1. 在 kube-system 下新建一个 eks-config 的 configmap。
2. 填入相应的参数设置，使其对所有 TKE Serverless 集群的 Pod 生效。

全局配置参考如下：

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: eks-config
  namespace: kube-system
data:
  pod.annotations: |
    eks.tke.cloud.tencent.com/resolv-conf: |
      nameserver 183.60.83.19
    eks.tke.cloud.tencent.com/host-sysctls: '[{"name":
"net.core.rmem_max", "value": "26214400"}]'
```

## 配置优先级说明

全局配置的优先级最低，其次是超级节点维度的配置，优先级最高的为 Pod 本身的配置，若配置冲突时，按照优先级生效。

# 运维中心

## 日志采集

### 使用 CRD 采集日志到 Kafka

最近更新时间：2023-07-11 16:26:01

TKE Serverless 集群不仅支持上传日志到 CLS，也支持采集日志到自建 Kafka 或者 CKafka。

## 创建 CRD

若需要采集日志到 Kafka，只需定义 CRD 即可。具体模板如下：

```
apiVersion: cls.cloud.tencent.com/v1
kind: LogConfig                                ## 默认值
metadata:
  name: test                                    ## CRD资源名，在集群内唯一
spec:
  kafkaDetail:
    brokers: xxxxxx                            # 必填，broker地址，一般是域名:端口，多个地址以“,”分隔
    topic: xxxxxx                              # 必填，topicID
    messageKey:                                # 选填，指定pod字段作为key上传到指定分区
      valueFrom:
        fieldRef:
          fieldPath: metadata.name
    timestampKey:                              #时间戳的key，默认是@timestamp
    timestampFormat:                          #时间戳的格式，默认是double
  inputDetail:
    type: container_stdout                    ## 采集日志的类型，包括
    container_stdout (容器标准输出)、container_file (容器文件)
    containerStdout:                          ## 容器标准输出
      namespace: default                      ## 采集容器的kubernetes命名空间，如果
      不指定，代表所有命名空间
      allContainers: false                    ## 是否采集指定命名空间中的所有容器的标
      准输出
      container: xxx                          ## 采集日志的容器名，此处可填空
      includeLabels:                          ## 采集包含指定label的Pod
        k8s-app: xxx                          ## 只采pod标签中配置"k8s-app=xxx"的
      pod产生的日志，与workloads、allContainers=true不能同时指定
      workloads:                              ## 要采集的容器的Pod所属的kubernetes
      workload
      -namespace: prod                        ## workload的命名空间
      name: sample-app                        ## workload的名字
```

```

    kind: deployment                                ## workload类型，支持deployment、
daemonset、statefulset、job、cronjob
    container: xxx                                ## 要采集的容器名，如果填空，代表
workload Pod中的所有容器

    containerFile:                                ## 容器内文件
    namespace: default                            ## 采集容器的kubernetes命名空间，必须
指定一个命名空间
    container: xxx                                ## 采集日志的容器名，此处可填*
    includeLabels:                                ## 采集包含指定label的Pod
    k8s-app: xxx                                  ## 只采pod标签中配置"k8s-app=xxx"的
pod产生的日志，与workload不能同时指定

    workload:                                     ## 要采集的容器的Pod所属的kubernetes
workload
    name: sample-app                              ## workload的名字
    kind: deployment                              ## workload类型，支持deployment、
daemonset、statefulset、job、cronjob
    logPath: /opt/logs                            ## 日志文件夹，不支持通配符
    filePattern: app_*.log                        ## 日志文件名，支持通配符 * 和 ? , *
表示匹配多个任意字符，? 表示匹配单个任意字符

```

## 注意事项

若无法采集日志，请销毁重建 Pod 重试。

# 使用 CRD 采集日志到 CLS

## 开启日志采集

最近更新时间：2023-10-23 18:17:11

### 操作场景

在 TKE Serverless 集群中，用户可以通过自定义资源（CustomResourceDefinitions，CRD）的方式配置日志采集。CRD 对 Pod 无侵入性，支持单行、多行、分隔符、完全正则、JSON 等多种日志解析方式，将标准输出、容器内文件日志发送至 [腾讯云日志服务 CLS](#)，提供检索分析、可视化应用、日志下载消费等服务。**推荐使用 CRD 配置日志采集。**

#### ⚠ 注意：

- 使用 CRD 配置日志采集目前只对2021年5月25号后新建的 Pod 生效，若需为旧 Pod 配置日志采集，请销毁重建。
- 若采集的 Pod 同时配置环境变量及 CRD 采集日志，会造成重复采集、重复计费。故使用 CRD 配置日志采集时，请删除相关环境变量。

### 概念

- **日志规则：**用户可以使用日志规则指定日志的采集源、消费端、日志解析方式、配置过滤器及上传解析失败等能力。日志采集器会监测日志采集规则的变化，变化的规则会在最多10s内生效。
- **日志源：**包含指定容器标准输出及容器内文件。
  - 在采集容器标准输出日志时，用户可选择所有容器、或指定工作负载和指定 Pod Labels 内的容器服务日志作为日志的采集源。
  - 在采集容器文件路径日志时，用户可指定工作负载或 Pod Labels 内容器的文件路径日志作为采集源。
- **消费端：**用户选择日志服务 CLS 的日志集和日志主题作为消费端。
- **提取模式：**日志采集 Agent 支持将采集到的日志以单行文本、JSON、分隔符、多行文本和完全正则的形式发送至用户指定的日志主题。
- **过滤器：**开启过滤器后可以根据用户指定的规则采集部分日志，key 支持完全匹配，过滤规则支持正则匹配，例如仅采集 ErrorCode = 404 的日志。
- **上传解析失败：**开启后，所有解析失败的日志，均以作为键名称（Key），原始日志内容作为值（Value）进行上传。关闭时会丢弃失败的日志。

### 操作步骤

#### 首次授权

首次使用 TKE Serverless 集群的日志采集功能，需要您对 CLS 等相关权限进行授权，以保证将日志正常上传到 CLS。

1. 登录 [容器服务控制台](#)，选择左侧导航栏中的 [运维功能管理](#)。
2. 在运维功能管理页面上方选择地域和 Serverless 集群，并单击需要开启日志采集的集群右侧的**设置**。

3. 在设置功能中，单击**访问管理**，为服务授权。如下图所示：

ⓘ 当前账号 [\[账号ID\]](#)，尚未授权腾讯云容器服务(TKE)操作CLS等云资源的权限，请前往 [访问管理](#) 完成授权。完成授权后，才能继续使用腾讯云容器服务，感谢您的配合

完成授权后，会默认为您的账号绑定角色 TKE\_QCSLinkedRoleInEKSLog，该角色配置的预设策略为 QcloudAccessForTKELinkedRoleInEKSLog。

#### ⓘ 说明

您只需在首次使用日志采集功能时进行授权，后续使用无需再操作。若您删除了以上角色，则会再次触发授权操作。

## 开启日志采集

完成授权后，即可开启日志采集。

1. 登录 [容器服务控制台](#)，选择左侧导航栏中的 [运维功能管理](#)。
2. 在运维功能管理页面上方选择地域和 Serverless 集群，并单击需要开启日志采集的集群右侧的**设置**。
3. 在设置功能中，单击日志采集**编辑**，勾选**开启日志采集**。
4. 单击**确定**。

## 下一步操作

开启日志采集后，您可根据以下操作使用 CRD 配置 TKE Serverless 集群的日志采集功能：

- [通过控制台配置日志采集](#)
- [通过 YAML 配置日志采集](#)

## 常见问题

如遇问题，您可先查询 [Serverless 集群日志采集相关问题](#)。如果您的问题仍未解决，请 [联系我们](#)。

# 通过控制台配置日志采集

最近更新时间：2025-07-01 14:42:22

本文介绍通过控制台配置 TKE Serverless 集群的日志采集功能。

## 前提条件

登录 [容器服务控制台](#)，并为 Serverless 集群开启日志采集功能。操作详情请参见 [开启日志采集](#)。

## 操作步骤

为集群开启日志采集功能后，您可根据以下操作进行配置：

### 配置日志规则

开启日志采集后，需要配置日志规则，确认日志源、消费端、日志解析方式等。

1. 登录 [容器服务控制台](#)，选择左侧导航栏中的 [日志管理](#) > [日志规则](#)。
2. 在 [日志规则](#) 页面上方选择地域和 TKE Serverless 集群，单击 [新建](#)。
3. 在 [新建日志采集规则](#) 页面中配置日志服务消费端，在 [消费端](#) > [类型](#) 中选择 [CLS](#)。
  - 日志集：根据日志所在地域展示您已经创建的日志集，若现有日志集不合适，您可以在 [日志服务控制台](#) 新建日志集。操作详情见 [创建日志集](#)。
  - 日志主题：选择日志集下对应的日志主题，支持“自动创建”和“选择已有”日志主题两种模式。

#### ⚠ 注意：

- 日志服务 CLS 目前只能支持同地域的容器集群进行日志采集上报。
- 日志集和日志主题在日志规则完成后不可更新。

4. 选择采集类型并配置日志源。目前采集类型支持 [容器标准输出](#) 和 [容器文件路径](#)。

#### 采集容器标准输出日志

选择容器标准输出采集类型，并根据自身需求，配置日志源。如下图所示：

类型  容器标准输出  容器文件路径

采集集群内任意服务下的容器日志，仅支持Stderr和Stdout的日志。 [查看示例](#)

日志源  所有容器  指定工作负载  指定 Pod Labels

全部命名空间  全部命名空间  指定命名空间

该类型日志源支持采集：

- **所有容器**：所有 Namespace 或某个 Namespace 下的所有容器。
- **指定工作负载**：某 Namespace 下，指定工作负载下的某些容器，并支持添加多个 Namespace。
- **指定 Pod Labels**：某 Namespace 下，指定多个 Pod Labels，采集符合该 Labels 的所有容器。

### 采集容器内文件日志

选择容器文件路径采集类型，并根据自身需求，配置日志源。如下图所示：

类型  容器标准输出  容器文件路径

采集集群内指定容器内的文件日志。 [查看示例](#)

日志源  指定工作负载  指定 Pod Labels

工作负载选项

容器名

采集路径

该类型日志源支持采集：

- **指定工作负载**：某 Namespace 下，指定工作负载下的容器的指定文件路径。
- **指定 Pod Labels**：某 Namespace 下，指定多个 Pod Labels，采集符合该 Labels 的所有容器的指定文件路径。

采集文件路径支持文件路径和通配规则，例如当容器文件路径为 `/opt/logs/*.log`，可以指定采集路径为 `/opt/logs`，文件名为 `*.log`。

**注意：**

如果选择采集类型为“容器文件路径”时，对应的“容器文件路径”不能为软链接，否则会导致软链接的实际路径在采集器的容器内不存在，采集日志失败。

### ❗ 说明：

对于容器的标准输出及容器内文件，除了原始的日志内容，还会带上容器或 Kubernetes 相关的元数据（例如，产生日志的 Pod name）一起上报到 CLS，方便用户查看日志时追溯来源或根据容器标识、特征（例如，容器名、Labels）进行检索。

容器或 Kubernetes 相关的元数据请参考下方表格：

| 字段名                    | 含义   |
|------------------------|--|
| cluster_id             | 日志所属的集群 ID。  |
| container_name         | 日志所属的容器名称。   |
| image_name             | 日志所属容器的镜像名称 IP。  |
| namespace              | 日志所属 pod 的 Namespace。  |
| pod_uid                | 日志所属 pod 的 UID。  |
| pod_name               | 日志所属 pod 的名字。  |
| pod_ip                 | 日志所属 pod 的 IP。   |
| pod_label_{label name} | 日志所属 pod 的 label（例如一个 pod 带有两个 label: app=nginx, env=prod, 则在上传的日志会附带两个 metadata : pod_label_app:nginx, pod_label_env:prod）。 |

### 5. 配置采集策略。您可以选择**全量**或者**增量**。

- **全量**：全量采集指从日志文件的开头开始采集。
- **增量**：增量采集指从距离文件末尾1M处开始采集（若日志文件小于1M，等价于全量采集）。

### 6. 单击**下一步**，选择日志解析方式。

- **编码模式**：支持**UTF-8**和**GBK**。
- **提取模式**：支持多种类型的提取模式，详情如下：

| 解析模式 | 说明   | 相关文档                   |
|------|--|------------------------|
| 单行全文 | 一条日志仅包含一行的内容，以换行符 \n 作为一条日志的结束标记，每条日志将被解析为键值为 CONTENT 的一行完全字符串，开启索引后可通过全文检索搜索日志内容。日志时间以采集时间为准。 | <a href="#">单行全文格式</a> |

|           |  |             |
|-----------|--|-------------|
| 多行全文      | 指一条完整的日志跨占多行，采用首行正则的方式进行匹配，当某行日志匹配上预先设置的正则表达式，就认为是一条日志的开头，而下一个行首出现作为该条日志的结束标识符，也会设置一个默认的键值 CONTENT，日志时间以采集时间为准。  | 多行全文格式      |
| 单行 - 完全正则 | 指将一条完整日志按正则方式提取多个 key-value 的日志解析模式，您需先输入日志样例，其次输入自定义正则表达式，系统将根据正则表达式里的捕获组提取对应的 key-value。   | 单行 - 完全正则格式 |
| 多行 - 完全正则 | 适用于日志文本中一条完整的日志数据跨占多行（例如 Java 程序日志），可按正则表达式提取为多个 key-value 键值的日志解析模式，您需先输入日志样例，其次输入自定义正则表达式，系统将根据正则表达式里的捕获组提取对应的 key-value。  | 多行 - 完全正则格式 |
| JSON      | JSON 格式日志会自动提取首层的 key 作为对应字段名，首层的 value 作为对应的字段值，以该方式将整条日志进行结构化处理，每条完整的日志以换行符 \n 为结束标识符。   | JSON 格式     |
| 分隔符       | 指一条日志数据可以根据指定的分隔符将整条日志进行结构化处理，每条完整的日志以换行符 \n 为结束标识符。日志服务在进行分隔符格式日志处理时，您需要为每个分开的字段定义唯一的 key，无效字段即无需采集的字段可填空，不支持所有字段均为空。   | 分隔符格式       |
| 组合解析      | 当您的日志结构太过复杂，涉及多种解析模式，单种解析模式（如 Nginx 模式、完全正则模式、JSON 模式等）无法满足日志解析需求时，您可以使用 LogListener 组合解析格式解析日志，此模式支持用户在控制台输入代码（JSON 格式）用来定义日志解析的流水线逻辑。您可添加一个或多个 Loglistener 插件处理配置，Loglistener 会根据处理配置顺序逐一执行。 | 组合解析格式      |

### ⚠ 注意：

一个日志主题目前仅支持一个采集配置，请保证选用该日志主题的所有容器的日志都可以接受采用所选的日志解析方式。若在同一日志主题下新建了不同的采集配置，旧的采集配置会被覆盖。

- 过滤器：LogListener 仅采集符合过滤器规则的日志，Key 支持完全匹配，过滤规则支持正则匹配，如仅采集 `ErrorCode = 404` 的日志。您可以根据需求开启过滤器并配置规则。

7. 单击完成。

## 更新日志规则

1. 登录 [容器服务控制台](#)，选择左侧导航栏中的 [日志管理](#) > [日志规则](#)。
2. 在 [日志规则](#) 页面中，选择需要更新的日志规则，单击右侧的 [编辑收集规则](#)。
3. 根据需求更新相应配置，单击完成。

# 通过 YAML 配置日志采集

最近更新时间：2023-12-11 14:56:41

本文介绍通过 YAML 方式使用 CRD 配置 TKE Serverless 集群的日志采集功能。

## 前提条件

登录 [容器服务控制台](#)，并为 Serverless 集群开启日志采集功能。操作详情请参见 [开启日志采集](#)。

## 创建 CRD

您只需要定义 LogConfig CRD 即可创建采集配置，采集组件根据 LogConfig CRD 的变化修改相应的日志服务 CLS 日志主题，并设置绑定的机器组。CRD 的格式如下：

## clsDetail 字段说明

### ⚠ 注意：

- topic 指定后不允许修改。
- 如果选择采集类型为“容器文件路径”时，对应的“容器文件路径”不能为软链接，否则会导致软链接的实际路径在采集器的容器内不存在，采集日志失败。

```
clsDetail:
  ## 自动创建日志主题，需要同时指定日志集和主题的名称
  logsetName: test          ## CLS日志集的名称，若无该名称的日志集，会自动创建，若有，会在该日志集下创建日志主题
  topicname: test          ## CLS日志主题的名称，若无该名称的日志主题，会自动创建

  # 选择已有日志集日志主题，如果指定了日志集未指定日志主题，则会自动创建一个日志主题
  logsetId: xxxxxx-xx-xx-xx-xxxxxxx ## CLS日志集的ID，日志集需要在CLS中提前创建
  topicid: xxxxxx-xx-xx-xx-xxxxxxx ## CLS日志主题的ID，日志主题需要在CLS中提前创建，且没有被其它采集配置占用

  logType: json_log ## 日志采集格式，json_log代表 json 格式，delimiter_log代表分隔符格式，minimalist_log代表单行全文格式，multiline_log代表多行全文格式，fullregex_log代表完全正则格式。默认为minimalist_log
  logFormat: xxx ## 日志格式化方式
  period: 30 ## 生命周期，单位天，可取值范围1~3600。取值为3640时代表永久保存
  partitionCount: ## Integer 类型，日志主题分区个数。默认创建1个，最大支持创建10个分区。
  tags: ## 标签描述列表，通过指定该参数可以同时绑定标签到相应的日志主题。最大支持9个标签键值对，同一个资源只能绑定到同一个标签键下。
    - key: xxx ## 标签key
```

```

value: xxx                                ## 标签value
autoSplit: false                          ## boolean 类型，是否开启自动分裂，默认值为
true
maxSplitPartitions:
storageType: hot                          ## 日志主题的存储类型，可选值 hot（标准存储），
cold（低频存储）；默认为hot。
excludePaths:                             ## 采集黑名单路径列表
- type: File                               ## 类型，选填File或Path
  value: /xx/xx/xx/xx.log                  ## type 对应的值
indexs:                                    ## 创建 topic 时可自定义索引方式和字段
- indexName:                               ## 需要配置键值或者元字段索引的字段，元字段Key无需额外添加__TAG__。
前缀，与上传日志时对应的字段Key一致即可，腾讯云控制台展示时将自动添加__TAG__前缀
  indexType:                               ## 字段类型，目前支持的类型有：long、text、double
  tokenizer:                               ## 字段的分词符，其中的每个字符代表一个分词符；仅支持英文符号及
\n\t\r; long及double类型字段需为空；text类型字段推荐使用 @&?|#()=',';:<>[]{} /
\n\t\r\ 作为分词符；
  sqlFlag:                                 ## boolean 字段是否开启分析功能
  containZH:                               ## boolean 是否包含中文
region: ap-xxx                             ## topic 所在地域，用于跨地域投递
userDefineRule: xxxxxxxx                  ## 用户自定义采集规则，Json格式序列化的字符串
extractRule: {}                           ## 提取、过滤规则。如果设置了ExtractRule，
则必须设置LogType

```

## inputDetail 字段说明

```

inputDetail:
  type: container_stdout ## 采集日志的类型，包括container_stdout（容器标准输出）、
container_file（容器文件）、host_file（主机文件）

  containerStdout:        ## 容器标准输出
    namespace: default   ## 采集容器的kubernetes命名空间。支持多个命名空间，如果有多个命名空间使用","分隔，如：default,namespace。如果不指定，代表所有命名空间。注意：与
excludeNamespace 不能同时指定
    excludeNamespace: nm1,nm2 ## 排除采集容器的kubernetes命名空间。支持多个命名空
间，如果有多个命名空间使用","分隔，如：nm1,nm2。如果不指定，代表所有命名空间。注意：与
namespace 不能同时指定
    nsLabelSelector: environment in (production),tier in (frontend) ## 根据命名
空间label 筛选符合的 namespace
    allContainers: false   ## 是否采集指定命名空间中的所有容器的标准输出。注
意：allContainers=true 时不能同时指定 workload, includeLabels 和 excludeLabels
    container: xxx        ## 采集日志的容器名，为空时，代表采集所有符合容器的日志
名。注意：与
    excludeLabels:       ## 采集不包含包含指定label的Pod，与workload, namespace 和
excludeNamespace 不能同时指定

```

```
key2: value2 ## 支持匹配同一个key下多个value值的pod, 例填写enviroment =
production,qa表示当key为enviroment, value值为production或qa时, 均会被排除, 注意输入
多个value值时请使用逗号隔开。如果同时指定了 includeLabels, 则匹配与 includeLabels 交集
的pod
```

```
includeLabels: ## 采集包含指定label的Pod, 与workload, namespace 和
excludeNamespace 不能同时指定
```

```
key: value1 ## 收集规则收集的日志会带上metadata, 并上报到消费端。支持匹配同一个
key下多个value值的pod, 例填写enviroment = production,qa表示当key为enviroment,
value值为production或qa时, 均会被匹配, 注意输入多个value值时请使用逗号隔开。 如果同时指
定了 excludeLabels, 则匹配与 excludeLabels 交集的pod
```

```
metadataLabels: ## 指定具体哪些pod label被当做元数据采集, 如果不指定,
则采集所有pod label为元数据
```

```
- label1
```

```
customLabels: ## 用户自定义metadata
```

```
label: l1
```

```
workloads:
```

```
- container: xxx ## 要采集的容器名, 如果不指定, 代表workload Pod中的所有容器
```

```
kind: deployment ## workload类型, 支持deployment、daemonset、
statefulset、job、cronjob
```

```
name: sample-app ## workload的名字
```

```
namespace: prod ## workload的命名空间
```

```
containerFile: ## 容器内文件
```

```
namespace: default ## 采集容器的kubernetes命名空间, 必须指定一个命名空间
```

```
excludeNamespace: nm1,nm2 ## 排除采集容器的kubernetes命名空间。支持多个命名空
间, 如果有多个命名空间使用", "分隔, 如: nm1,nm2。 如果不指定, 代表所有命名空间。 注意: 与
namespace 不能同时指定
```

```
nsLabelSelector: environment in (production),tier in (frontend) ## 根据命名
空间label 筛选符合的 namespace
```

```
container: xxx ## 采集日志的容器名, 为 * 时, 代表采集所有符合容器的日志名
```

```
logPath: /var/logs ## 日志文件夹, 不支持通配符
```

```
filePattern: app_*.log ## 日志文件名, 支持通配符 * 和 ? , * 表示匹配多个任意字
符, ? 表示匹配单个任意字符
```

```
customLabels: ## 用户自定义metadata
```

```
key: value
```

```
excludeLabels: ## 采集不包含包含指定label的Pod, 与workload不能同时指定
```

```
key2: value2 ## 支持匹配同一个key下多个value值的pod, 例填写enviroment =
production,qa表示当key为enviroment, value值为production或qa时, 均会被排除, 注意输入
多个value值时请使用逗号隔开。如果同时指定了 includeLabels, 则匹配与 includeLabels 交集
的pod
```

```
includeLabels: ## 采集包含指定label的Pod, 与workload不能同时指定
```

```
key: value1 ## 收集规则收集的日志会带上metadata，并上报到消费端。支持匹配同一个
key下多个value值的pod，例填写enviroment = production,qa表示当key为enviroment，
value值为production或qa时，均会被匹配，注意输入多个value值时请使用逗号隔开。 如果同时指
定了 excludeLabels，则匹配与 excludeLabels 交集的pod
  metadataLabels:      ## 指定具体哪些pod label被当做元数据采集，如果不指定，则采集
  所有pod label为元数据
  - label1             ## pod label
  workload:
    container: xxx     ## 要采集的容器名，如果不指定，代表workload Pod中的所有容器
    name: sample-app   ## workload的名字
```

## 日志解析格式

### 单行全文格式

单行全文日志是指一行日志内容为一条完整的日志。日志服务在采集的时候，将使用换行符 `\n` 来作为一条日志的结束符。为了统一结构化管理，每条日志都会存在一个默认的键值 `__CONTENT__`，但日志数据本身不再进行日志结构化处理，也不会提取日志字段，日志属性的时间项由日志采集的时间决定。详情请参见 [单行文本格式](#)。

假设一条日志原始数据为：

```
Tue Jan 22 12:08:15 CST 2019 Installed: libjpeg-turbo-static-1.2.90-
6.e17.x86_64
```

LogConfig 配置参考示例如下：

```
apiVersion: cls.cloud.tencent.com/v1
kind: LogConfig
spec:
  clsDetail:
    topicId: xxxxxx-xx-xx-xx-xxxxxxxxx
    # 单行日志
    logType: minimalist_log
```

采集到日志服务的数据为：

```
__CONTENT__:Tue Jan 22 12:08:15 CST 2019 Installed: libjpeg-turbo-static-
1.2.90-6.e17.x86_64
```

### 多行全文格式

多行全文日志是指一条完整的日志数据可能跨占多行（例如 Java stacktrace）。该情况下无法使用换行符 `\n` 作为日志的结束标识符，为了使日志系统明确区分每条日志，采用首行正则的方式进行匹配，当某行日志匹配预先设置的正则表达式，即为一条日志的开头，而下一行首出现则作为该条日志的结束标识符。多行全文也会设置一个默认的键值 `__CONTENT__`，但日志数据本身不再进行日志结构化处理，也不会提取日志字段，日志属性的时间项由日志采集的时间决定。详情请参见 [多行文本格式](#)。

假设一条多行日志原始数据为：

```
2019-12-15 17:13:06,043 [main] ERROR com.test.logging.FooFactory:
java.lang.NullPointerException
    at com.test.logging.FooFactory.createFoo(FooFactory.java:15)
    at com.test.logging.FooFactoryTest.test(FooFactoryTest.java:11)
```

LogConfig 配置的参考如下：

```
apiVersion: cls.cloud.tencent.com/v1
kind: LogConfig
spec:
  clsDetail:
    topicId: xxxxxx-xx-xx-xx-xxxxxxxx
    # 多行日志
    logType: multiline_log
    extractRule:
      # 只有以日期时间开头的行才被认为是新一条日志的开头，否则就添加换行符\n并追加到当前日志的尾部
      beginningRegex: \d{4}-\d{2}-\d{2}\s\d{2}:\d{2}:\d{2},\d{3}\s.+
```

采集到日志服务的数据为：

```
__CONTENT__: 2019-12-15 17:13:06,043 [main] ERROR
com.test.logging.FooFactory:
java.lang.NullPointerException
    at com.test.logging.FooFactory.createFoo(FooFactory.java:15)
    at com.test.logging.FooFactoryTest.test(FooFactoryTest.java:11)
```

### 单行-完全正则格式

完全正则格式通常用来处理结构化的日志，指将一条完整日志按正则方式提取多个 key-value 的日志解析模式。详情请参见 [完全正则格式](#)。

假设一条日志原始数据为：

```
10.135.46.111 - - [22/Jan/2019:19:19:30 +0800] "GET /my/course/1
HTTP/1.1" 127.0.0.1 200 782 9703 "http://127.0.0.1/course/explore?
filter%5Btype%5D=all&filter%5Bprice%5D=all&filter%5BcurrentLevelId%5D=all
&orderBy=studentNum" "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:64.0)
Gecko/20100101 Firefox/64.0" 0.354 0.354
```

LogConfig 配置的参考如下：

```
apiVersion: cls.cloud.tencent.com/v1
kind: LogConfig
spec:
  clsDetail:
    topicId: xxxxxx-xx-xx-xx-xxxxxxxxx
    # 完全正则格式
    logType: fullregex_log
    extractRule:
      # 正则表达式，会根据 () 捕获组提取对应的value
      logRegex: (\S+) [^\[]+(\[[^:]+\d+:\d+:\d+\s\S+)\s"
(\w+)\s(\S+)\s("[^"]+)"\s(\S+)\s(\d+)\s(\d+)\s(\d+)\s"("[^"]+)"\s"
("[^"]+)"\s+(\S+)\s(\S+).*
      beginningRegex: (\S+) [^\[]+(\[[^:]+\d+:\d+:\d+\s\S+)\s"
(\w+)\s(\S+)\s("[^"]+)"\s(\S+)\s(\d+)\s(\d+)\s(\d+)\s"("[^"]+)"\s"
("[^"]+)"\s+(\S+)\s(\S+).*
      # 提取的key列表，与提取的value的一一对应
      keys: ['remote_addr', 'time_local', 'request_method',
'request_url', 'http_protocol', 'http_host', 'status', 'request_length',
'body_bytes_sent', 'http_referer', 'http_user_agent', 'request_time',
'upstream_response_time']
```

采集到日志服务的数据为：

```
body_bytes_sent: 9703
http_host: 127.0.0.1
http_protocol: HTTP/1.1
http_referer: http://127.0.0.1/course/explore?
filter%5Btype%5D=all&filter%5Bprice%5D=all&filter%5BcurrentLevelId%5D=all
&orderBy=studentNum
http_user_agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:64.0)
Gecko/20100101 Firefox/64.0
remote_addr: 10.135.46.111
request_length: 782
```

```
request_method: GET
request_time: 0.354
request_url: /my/course/1
status: 200
time_local: [22/Jan/2019:19:19:30 +0800]
upstream_response_time: 0.354
```

## 多行-完全正则格式

多行-完全正则模式适用于日志文本中一条完整的日志数据跨占多行（例如 Java 程序日志），可按正则表达式提取为多个 key-value 键值的日志解析模式。若不需要提取 key-value，请参阅多行全文格式进行配置。详情请参见[多行-完全正则格式](#)。

假设一条日志原始数据为：

```
[2018-10-01T10:30:01,000] [INFO] java.lang.Exception: exception happened
  at TestPrintStackTrace.f(TestPrintStackTrace.java:3)
  at TestPrintStackTrace.g(TestPrintStackTrace.java:7)
  at TestPrintStackTrace.main(TestPrintStackTrace.java:16)
```

LogConfig 配置的参考如下：

```
apiVersion: cls.cloud.tencent.com/v1
kind: LogConfig
spec:
  clsDetail:
    topicId: xxxxxx-xx-xx-xx-xxxxxxxx
    # 多行-完全正则格式
    logType: multiline_fullregex_log
    extractRule:
      # 行首完全正则表达式，只有以日期时间开头的行才被认为是新一条日志的开头，否则就添加换行符\n并追加到当前日志的尾部
      beginningRegex: \[\d+-\d+-\d+\:\d+\:\d+\,\d+\]\s\[ \w+\]\s.*
      # 正则表达式，会根据()捕获组提取对应的value
      logRegex: \[(\d+-\d+-\d+\:\d+\:\d+\,\d+)\]\s\[ (\w+)\]\s(.*)
      # 提取的 key 列表，与提取的 value 的一一对应
    keys:
      - time
      - level
      - msg
```

根据提取的 key，采集到日志服务的数据为：

```
time: 2018-10-01T10:30:01,000
level: INFO
msg: java.lang.Exception: exception happened
    at TestPrintStackTrace.f(TestPrintStackTrace.java:3)
    at TestPrintStackTrace.g(TestPrintStackTrace.java:7)
    at TestPrintStackTrace.main(TestPrintStackTrace.java:16)
```

## JSON 格式

JSON 格式日志会自动提取首层的 key 作为对应字段名。首层的 value 作为对应的字段值，以该方式将整条日志进行结构化处理，每条完整的日志以换行符 `\n` 为结束标识符。详情请参见 [JSON 格式](#)。

假设一条 JSON 日志原始数据为：

```
{"remote_ip":"10.135.46.111","time_local":"22/Jan/2019:19:19:34
+0800","body_sent":23,"responsetime":0.232,"upstreamtime":"0.232","upstre
amhost":"unix:/tmp/php-
cgi.sock","http_host":"127.0.0.1","method":"POST","url":"/event/dispatch"
,"request":"POST /event/dispatch HTTP/1.1","xff":"-
","referer":"http://127.0.0.1/my/course/4","agent":"Mozilla/5.0 (Windows
NT 10.0; WOW64; rv:64.0) Gecko/20100101
Firefox/64.0","response_code":"200"}
```

LogConfig 配置的参考如下：

```
apiVersion: cls.cloud.tencent.com/v1
kind: LogConfig
spec:
  clsDetail:
    topicId: xxxxxxx-xx-xx-xx-xxxxxxxxx
    # JSON格式日志
    logType: json_log
```

采集到日志服务的数据为：

```
agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:64.0) Gecko/20100101
Firefox/64.0
```

```
body_sent: 23
http_host: 127.0.0.1
method: POST
referer: http://127.0.0.1/my/course/4
remote_ip: 10.135.46.111
request: POST /event/dispatch HTTP/1.1
response_code: 200
responsetime: 0.232
time_local: 22/Jan/2019:19:19:34 +0800
upstreamhost: unix:/tmp/php-cgi.sock
upstreamtime: 0.232
url: /event/dispatch
xff: -
```

## 分隔符格式

分隔符日志是指一条日志数据可以根据指定的分隔符将整条日志进行结构化处理，每条完整的日志以换行符 `\n` 为结束标识符。日志服务在进行分隔符格式日志处理时，您需要为每个分开的字段定义唯一的 key。详情请参见 [分隔符格式](#)。

假设原始日志为：

```
10.20.20.10 ::: [Tue Jan 22 14:49:45 CST 2019 +0800] ::: GET
/online/sample HTTP/1.1 ::: 127.0.0.1 ::: 200 ::: 647 ::: 35 :::
http://127.0.0.1/
```

LogConfig 配置的参考如下：

```
apiVersion: cls.cloud.tencent.com/v1
kind: LogConfig
spec:
  clsDetail:
    topicId: xxxxxx-xx-xx-xx-xxxxxxxxx
    # 分隔符日志
    logType: delimiter_log
    extractRule:
      # 分隔符
      delimiter: ':::'
      # 提取的 key 列表，与被分割的字段一一对应
      keys: ['IP', 'time', 'request', 'host', 'status', 'length',
'bytes', 'referer']
```

采集到日志服务的数据为：

```
IP: 10.20.20.10
bytes: 35
host: 127.0.0.1
length: 647
referer: http://127.0.0.1/
request: GET /online/sample HTTP/1.1
status: 200
time: [Tue Jan 22 14:49:45 CST 2019 +0800]
```

## 采集日志的类型

### 容器标准输出

#### 示例1：采集 default 命名空间中的所有容器的标准输出

```
apiVersion: cls.cloud.tencent.com/v1
kind: LogConfig
spec:
  inputDetail:
    type: container_stdout
    containerStdout:
      namespace: default
      allContainers: true
  ...
```

#### 示例2：采集 production 命名空间中属于 ingress-gateway deployment 的 pod 中的容器的标准输出

```
apiVersion: cls.cloud.tencent.com/v1
kind: LogConfig
spec:
  inputDetail:
    type: container_stdout
    containerStdout:
      allContainers: false
      workloads:
        - namespace: production
          name: ingress-gateway
          kind: deployment
```

...

**示例3: 采集 production 命名空间下 pod 标签中包含 “k8s-app=nginx” 的 pod 中的容器的标准输出**

```
apiVersion: cls.cloud.tencent.com/v1
kind: LogConfig
spec:
  inputDetail:
    type: container_stdout
    containerStdout:
      namespace: production
      allContainers: false
      includeLabels:
        k8s-app: nginx
  ...
```

## 容器文件

**示例1: 采集 production 命名空间下属于 ingress-gateway deployment 的 pod 中的 nginx 容器中 /data/nginx/log/ 路径下名为 access.log 的文件**

```
apiVersion: cls.cloud.tencent.com/v1
kind: LogConfig
spec:
  topicId: xxxxxx-xx-xx-xx-xxxxxxxx
  inputDetail:
    type: container_file
    containerFile:
      namespace: production
      workload:
        name: ingress-gateway
        type: deployment
      container: nginx
      logPath: /data/nginx/log
      filePattern: access.log
  ...
```

**示例2: 采集 production 命名空间下 pod 标签包含 “k8s-app=ingress-gateway” 的 pod 中的 nginx 容器中 /data/nginx/log/ 路径下名为 access.log 的文件**

```
apiVersion: cls.cloud.tencent.com/v1
kind: LogConfig
spec:
```

```
inputDetail:
  type: container_file
  containerFile:
    namespace: production
    includeLabels:
      k8s-app: ingress-gateway
    container: nginx
    logPath: /data/nginx/log
    filePattern: access.log
  ...
```

## 元数据 (Metadata)

容器标准输出 (container\_stdout) 以及容器文件 (container\_file)，除原始的日志内容外，还需携带容器场景的元数据 (例如产生日志的容器 ID) 一起上报至日志服务。方便用户查看日志时追溯来源或根据容器标识、特征 (例如容器名及 labels) 进行检索。

元数据如下表：

| 字段名                    | 含义   |
|------------------------|--|
| cluster_id             | 日志所属的集群 ID。  |
| container_name         | 日志所属的容器名称。   |
| image_name             | 日志所属容器的镜像名称 IP。  |
| namespace              | 日志所属 pod 的 namespace。  |
| pod_uid                | 日志所属 pod 的 UID。  |
| pod_name               | 日志所属 pod 的名字。  |
| pod_ip                 | 日志所属 pod 的 IP。   |
| pod_label_{label name} | 日志所属 pod 的 label (例如一个 pod 带有两个 label: app=nginx, env=prod, 则在上传的日志会附带两个 metadata: pod_label_app:nginx, pod_label_env:prod)。 |

# 审计管理

## 审计日志

最近更新时间：2025-02-27 17:55:12

### 简介

审计日志是基于 [Kubernetes Audit](#) 对 kube-apiserver 产生的可配置策略的 JSON 结构日志的记录存储及检索功能。本功能记录了对 kube-apiserver 的访问事件，会按顺序记录每个用户、管理员或系统组件影响集群的活动。

### 功能优势

审计日志功能提供了区别于 metrics 的另一种集群观测维度。开启审计日志后，Kubernetes 可以记录每一次对集群操作的审计日志。每一条审计日志是一个 JSON 格式的结构化记录，包括元数据（metadata）、请求内容（requestObject）和响应内容（responseObject）三个部分。其中元数据（包含了请求的上下文信息，例如谁发起的请求、从哪里发起的、访问的 URI 等信息）一定存在，请求和响应内容是否存在取决于审计级别。通过日志可以了解到以下内容：

- 集群里发生的活动。
- 活动的发生时间及发生对象。
- 活动的触发时间、触发位置及观察点。
- 活动的结果以及后续处理行为。

### 阅读审计日志示例

```
{
  "kind": "Event",
  "apiVersion": "audit.k8s.io/v1",
  "level": "RequestResponse",
  "auditID": "0a4376d5-307a-4e16-a049-24e017*****",
  "stage": "ResponseComplete",
  // 发生了什么
  "requestURI": "/apis/apps/v1/namespaces/default/deployments",
  "verb": "create",
  // 谁发起的
  "user": {
    "username": "admin",
    "uid": "admin",
    "groups": [
      "system:masters",
      "system:authenticated"
    ]
  },
  // 从哪里发起
  "sourceIPs": [
```

```
"10.0.6.68"
],
"userAgent": "kubectl/v1.16.3 (linux/amd64) kubernetes/ald64d8",
// 发生了什么
"objectRef": {
  "resource": "deployments",
  "namespace": "default",
  "name": "nginx-deployment",
  "apiGroup": "apps",
  "apiVersion": "v1"
},
// 结果是什么
"responseStatus": {
  "metadata": {
  },
  "code": 201
},
// 请求及返回具体信息
"requestObject": Object{...},
"responseObject": Object{...},
// 什么时候开始/结束
"requestReceivedTimestamp": "2020-04-10T10:47:34.315746Z",
"stageTimestamp": "2020-04-10T10:47:34.328942Z",
// 请求被接收/拒绝的原因是什么
"annotations": {
  "authorization.k8s.io/decision": "allow",
  "authorization.k8s.io/reason": ""
}
}
```

## TKE Serverless 审计日志策略

### 审计级别 (level)

和一般日志不同，Kubernetes 审计日志的级别更像是一种 verbose 配置，用于标示记录信息的详细程度。一共有4个级别，可参考以下表格内容：

| 参数              | 说明                                     |
|-----------------|--|
| None            | 不记录。                                   |
| Metadata        | 记录请求的元数据（例如用户、时间、资源、操作等），不包括请求和响应的消息体。 |
| Request         | 除了元数据外，还包括请求消息体，不包括响应消息体。              |
| RequestResponse | 记录所有信息，包括元数据以及请求、响应的消息体。               |

## 审计阶段 (stage)

记录日志可以发生在不同的阶段，参考以下表格内容：

| 参数               | 说明                               |
|------------------|----------------------------------|
| RequestReceived  | 一收到请求就记录。                        |
| ResponseStarted  | 返回消息头发送完毕后记录，只针对 watch 之类的长连接请求。 |
| ResponseComplete | 返回消息全部发送完毕后记录。                   |
| Panic            | 内部服务器出错，请求未完成。                   |

## 审计策略

TKE Serverless 默认收到请求即会记录审计日志，且大部分的操作会记录 RequestResponse 级别的审计日志。但也会存在如下情况：

- get、list 和 watch 会记录 Request 级别的日志。
- 针对 secrets 资源、configmaps 资源或 tokenreviews 资源的请求会在 Metadata 级别记录。

以下请求将不会进行记录日志：

- `system:kube-proxy` 发出的监视 endpoints 资源、services 资源或 services/status 资源的请求。
- `system:unsecured` 发出的针对 kube-system 命名空间中 configmaps 资源的 get 请求。
- kubelet 发出的针对 nodes 资源或 nodes/status 资源的 get 请求。
- `system:kube-controller-manager`、`system:kube-scheduler` 或 `system:serviceaccount:endpoint-controller` 发出的针对 kube-system 命名空间中 endpoints 资源的 get 和 update 请求。
- `system:apiserver` 发出的针对 namespaces 资源、namespaces/status 资源或 namespaces/finalize 资源的 get 请求。
- 对与 `/healthz*`、`/version` 或 `/swagger*` 匹配的网址发出的请求。

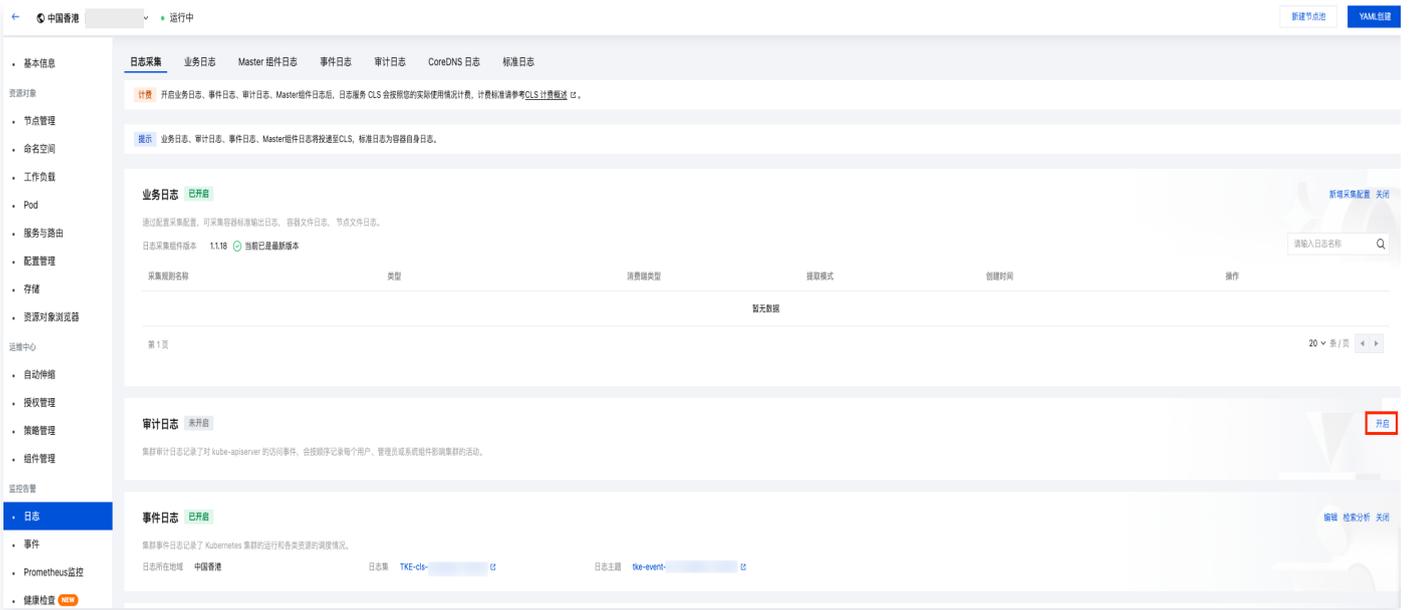
## 操作步骤

### 开启审计日志

#### ⚠ 注意：

开启审计日志功能需要重启 kube-apiserver，建议不要频繁开关。

1. 登录 [容器服务控制台](#)，在左侧导航栏中选择**集群**。
2. 在集群列表中，单击目标集群 ID，进入集群详情页。
3. 选择左侧导航栏中的**日志**，在日志采集页签中单击**审计日志**右侧的**开启**。如下图所示：



4. 在开启审计日志中，选择日志所在地域，并配置日志集和日志主题。如下图所示：



5. 单击**确定**，即可开启审计日志。

# 审计仪表盘

最近更新时间：2025-03-21 10:13:22

## 操作场景

TKE Serverless 为用户提供了开箱即用的审计仪表盘。在集群开启审计日志功能后，TKE Serverless 将自动为该集群配置审计总览、K8S 对象操作概览、聚合检索仪表盘。还支持用户自定义配置过滤项，同时内置 CLS 的全局检索，方便用户观测和检索各类集群操作，以便于及时发现和定位问题。

## 功能介绍

审计检索中配置了四个大盘，分别是审计总览、K8S 对象操作概览、聚合检索、全局检索。请按照以下步骤进入审计日志页面，开始使用对应功能：

1. 登录 [容器服务控制台](#)。
2. 开启审计日志功能，详情请参见 [审计日志](#)。
3. 在集群列表中，单击目标集群 ID，进入集群详情页。
4. 选择左侧导航栏中的日志，选择审计日志页面。

## 审计总览

当您需要观测整个集群 APIserver 操作时，可在“审计总览”页面设置过滤条件，查看核心审计日志的汇总统计信息，并展示一个周期内的数据对比。例如，核心审计日志的统计数、分布情况、重要操作趋势等。

在过滤项中，您可根据自己需求进行个性化配置，如下图所示：

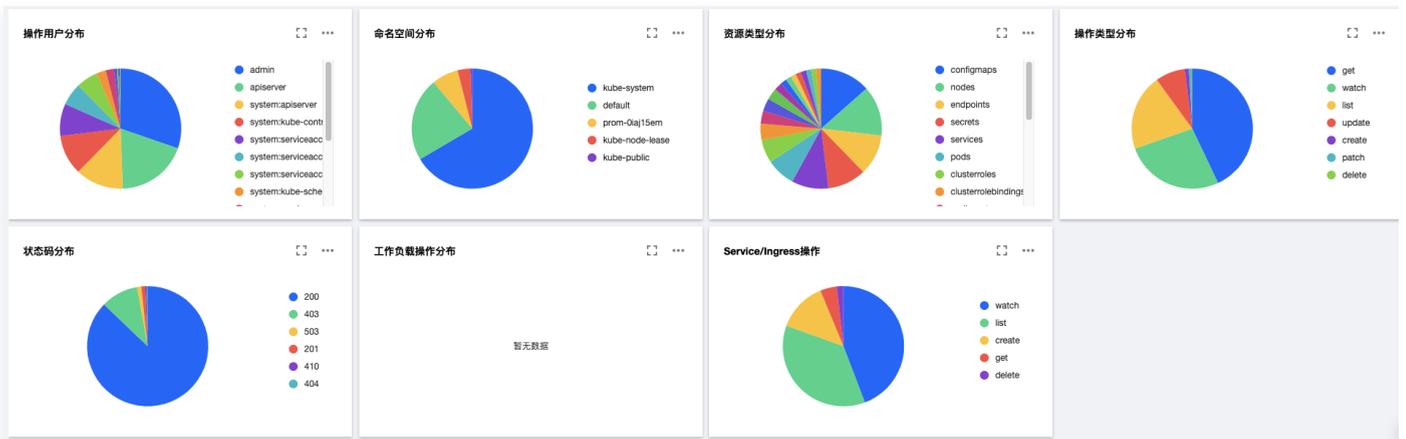


您还可在该页面中查看更多统计信息，如下所示：

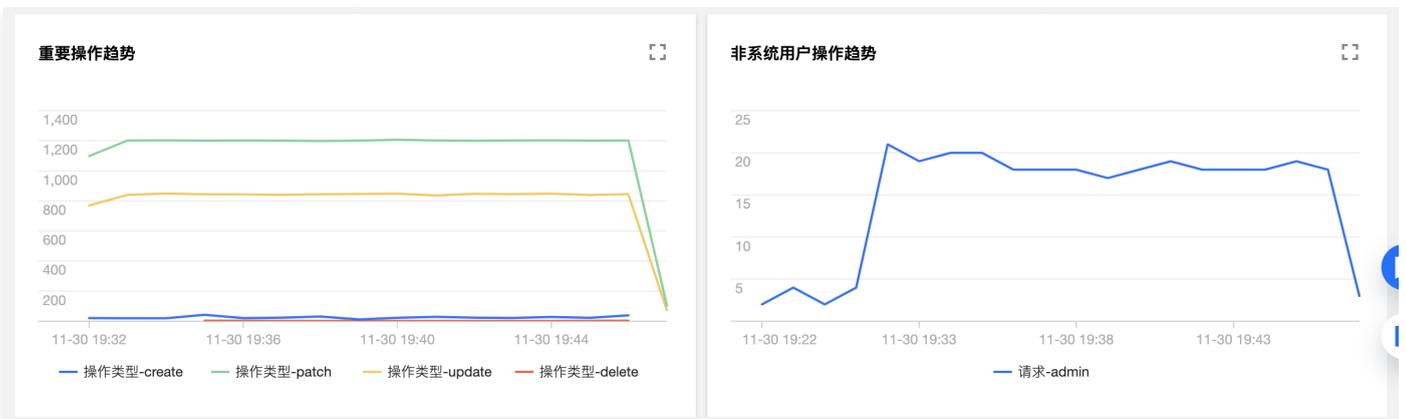
- 核心审计日志的统计数仪表盘：



● 分布情况仪表盘：



● 重要操作趋势仪表盘：



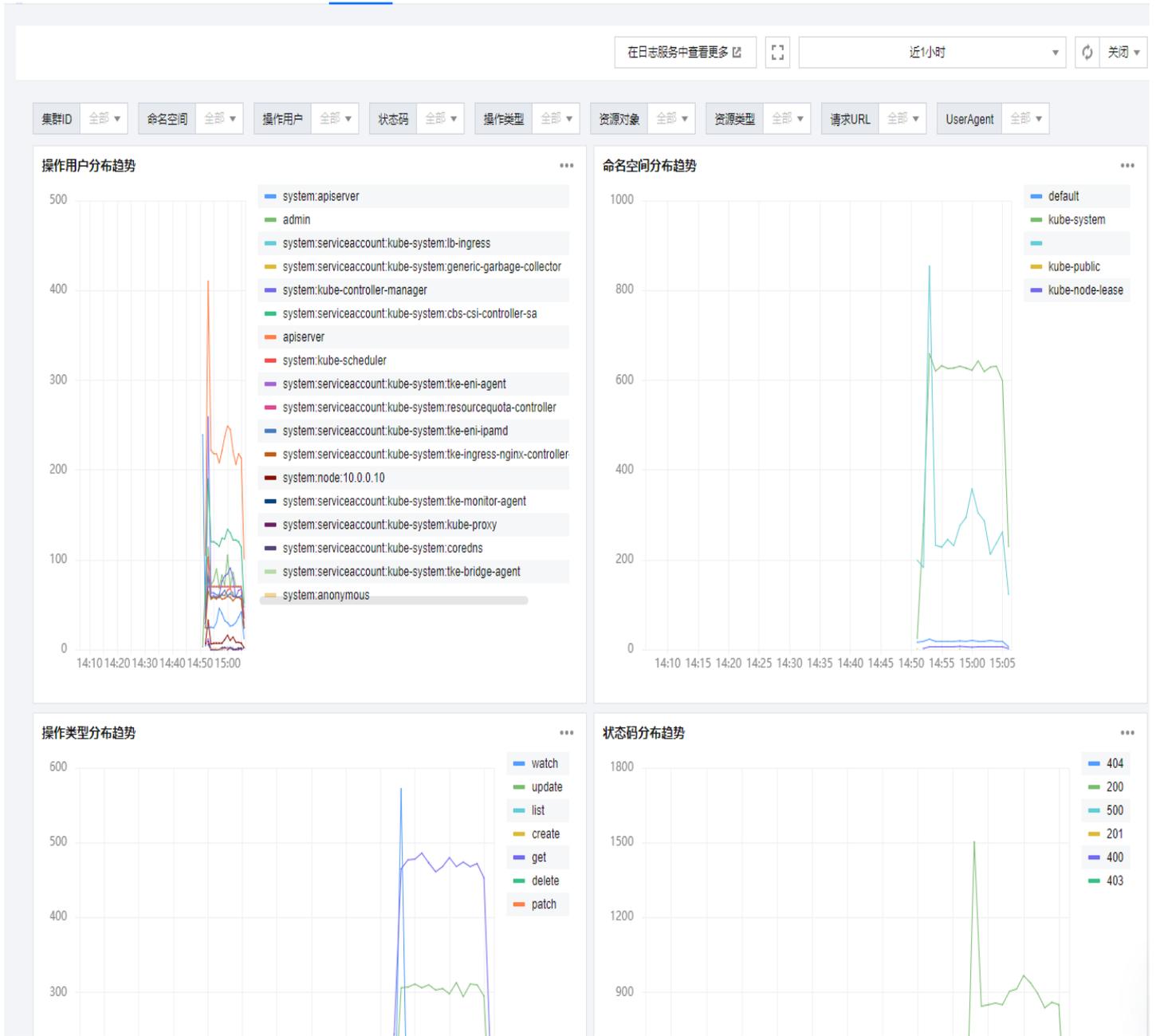
## K8S 对象操作概览

当您需要排查 K8S 对象（例如某个工作负载）的相关问题时，可在切换至 **K8S 对象操作概览** 页签，在此页面设置过滤条件，查看各类 K8S 对象的操作概览、对应的操作用户、相应的审计日志列表，以查找更多的细节。如下图所示：



## 聚合检索

当您需要观测某个维度下审计日志的分布趋势，可在“聚合检索”页面设置过滤条件，查看各类重要操作的时序图。纬度包括操作用户、命名空间、操作类型、状态码、资源类型以及相应的审计日志列表。如下图所示：



## 全局检索

全局检索仪表盘内嵌了 CLS 的检索分析页面，方便用户在容器服务控制台也能快速检索全部审计日志。如下图所示：

检索分析
上海 1
主题类型 日志主题
日志类 TKE-cls-
日志主题 tke-audit-cls-
多主题检索
监控统计
产品文档

<> 语句模式(CQL) 收藏夹 历史记录

HOT 推荐仪表盘 告警 健康监控 采集配置 索引配置 更多

1 e.g. \_\_SOURCE\_\_: AND "http/1.0" ☆ ⚙ 近15分钟 🔍

原始日志 统计图表

添加到仪表盘 添加告警策略 下载

搜索 🔍

日志条数 14,203 2023-07-25 14:52:14.628 - 2023-07-25 15:07:14.628

换行
  行号
  日志时间
  JSON格式化
 版面: 默认配置

| 行号 | 日志时间               | 原始日志  |
|----|--------------------|---|
| 1  | 07-25 15:07:10.010 | <pre>auditID: requestReceivedTimestamp: 2023-07-25T07:07:09.952086Z objectRef.uid: objectRef.apiVersion: v1 objectRef.resource: configmaps objectRef.resourceVersion: objectRef.name: qcloud-tke-nginx-ingress-controller objectRef.namespace: kube-system level: Metadata kind: Event verb: update annotations.authorization.k8s.io/decision: allow annotations.authorization.k8s.io/reason: RBAC:</pre> |

## 基于仪表盘配置告警

您可以通过以上预设的仪表盘配置告警，达到您所设置的条件则触发告警。操作详情如下：

1. 单击需要配置告警的仪表盘右侧的**添加到监控告警**。
2. 在 **日志服务控制台** 中，选择**监控告警 > 告警策略**，新建告警策略。详情可参见 [配置告警策略](#)。

# 事件管理

## 事件日志

最近更新时间：2025-02-27 17:55:12

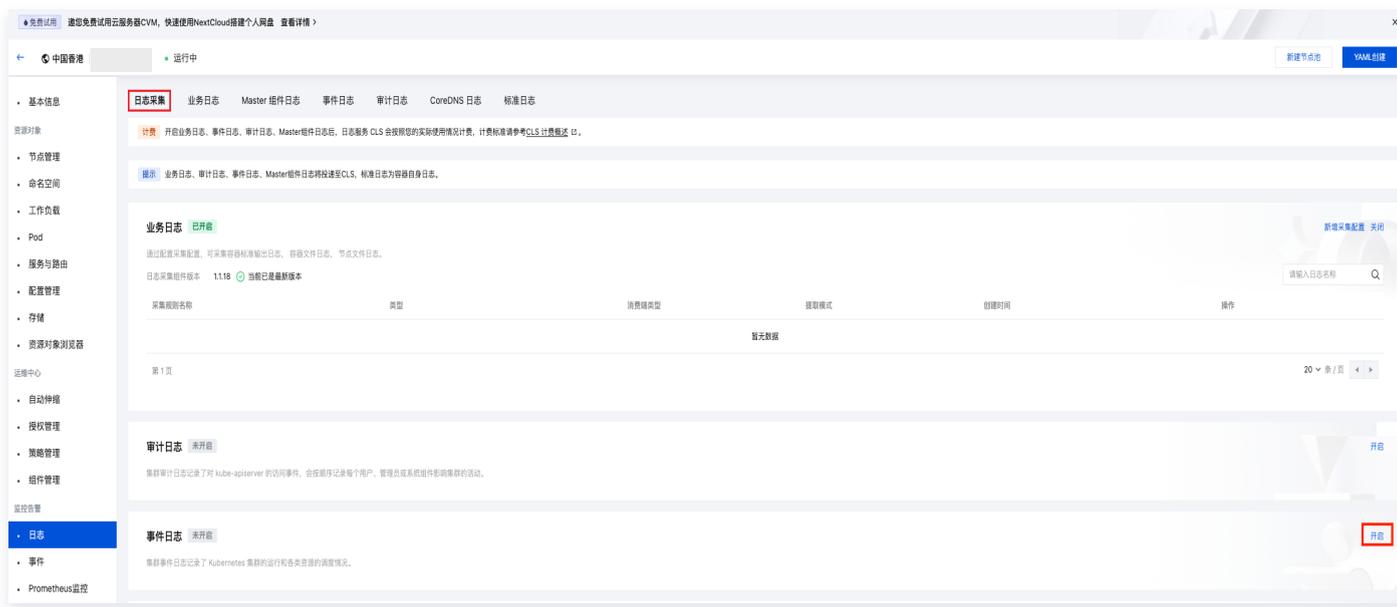
### 操作场景

Kubernetes Events 包括 Kubernetes 集群的运行和各类资源调度情况，有助于维护人员日常观察资源的变更以及定位问题。TKE Serverless 支持为您的所有集群配置事件持久化功能，还支持使用腾讯云提供的 PAAS 服务和开源软件对事件流水进行检索。开启集群事件持久化后，TKE Serverless 会将您的集群事件实时导出至配置的存储端。本文档指导您如何开启集群事件持久化存储，您可参考本文开始使用事件日志功能。

### 操作步骤

#### 开启事件日志

1. 登录 [容器服务控制台](#)，在左侧导航栏中选择**集群**。
2. 在集群列表中，单击目标集群 ID，进入集群详情页。
3. 选择左侧导航栏中的**日志**，在日志采集页签中单击**事件日志**右侧的**开启**。如下图所示：



4. 在**开启事件日志**对话框中，选择日志所在地域，并配置日志集和日志主题。如下图所示：

### 开启事件日志 ×

**计费** 开启事件存储后，日志服务CLS会按照您的实际使用情况计费，计费标准请参考[CLS计费概述](#) 🔗

日志所在地域  修改

日志集

- 日志所在地域：CLS 支持跨地域日志投递，您可通过单击修改选择日志投递的目标地域。
- 日志集：根据日志所在地域展示您已经创建的日志集，若现有日志集不合适，您可以在 [日志服务控制台](#) 新建日志集。操作详情见 [创建日志集](#)。

**注意：**

每个日志集下最多可以创建500个日志主题及指标主题。

- 日志主题：选择日志集下对应的日志主题，支持“自动创建”和“选择已有”日志主题两种模式。

5. 单击**确定**，即可开启事件日志。

## 更新日志集或日志主题

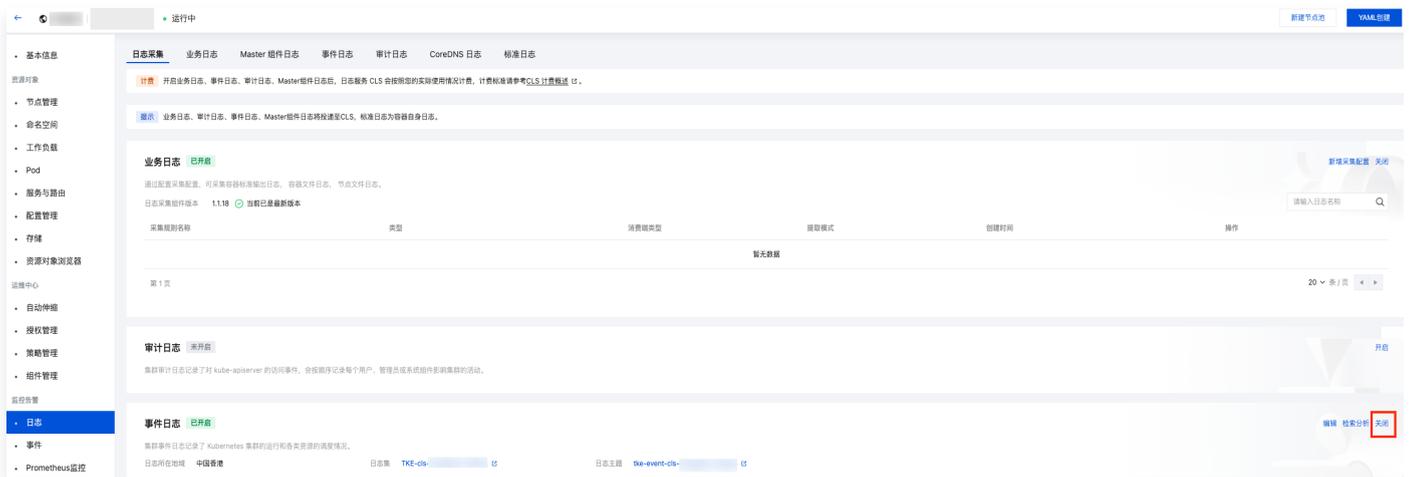
1. 登录 [容器服务控制台](#)，在左侧导航栏中选择集群。
2. 在集群列表中，单击目标集群 ID，进入集群详情页。
3. 选择左侧导航栏中的**日志**，在日志采集页签中单击事件日志右侧的**编辑**。
4. 在编辑事件日志中，重新选择**日志集和日志主题**。如下图所示：



5. 单击确定即可更新日志集和日志主题。

## 关闭事件日志

1. 登录 [容器服务控制台](#)，在左侧导航栏中选择集群。
2. 在集群列表中，单击目标集群 ID，进入集群详情页。
3. 选择左侧导航栏中的日志，在日志采集页签中单击事件日志右侧的关闭。如下图所示：

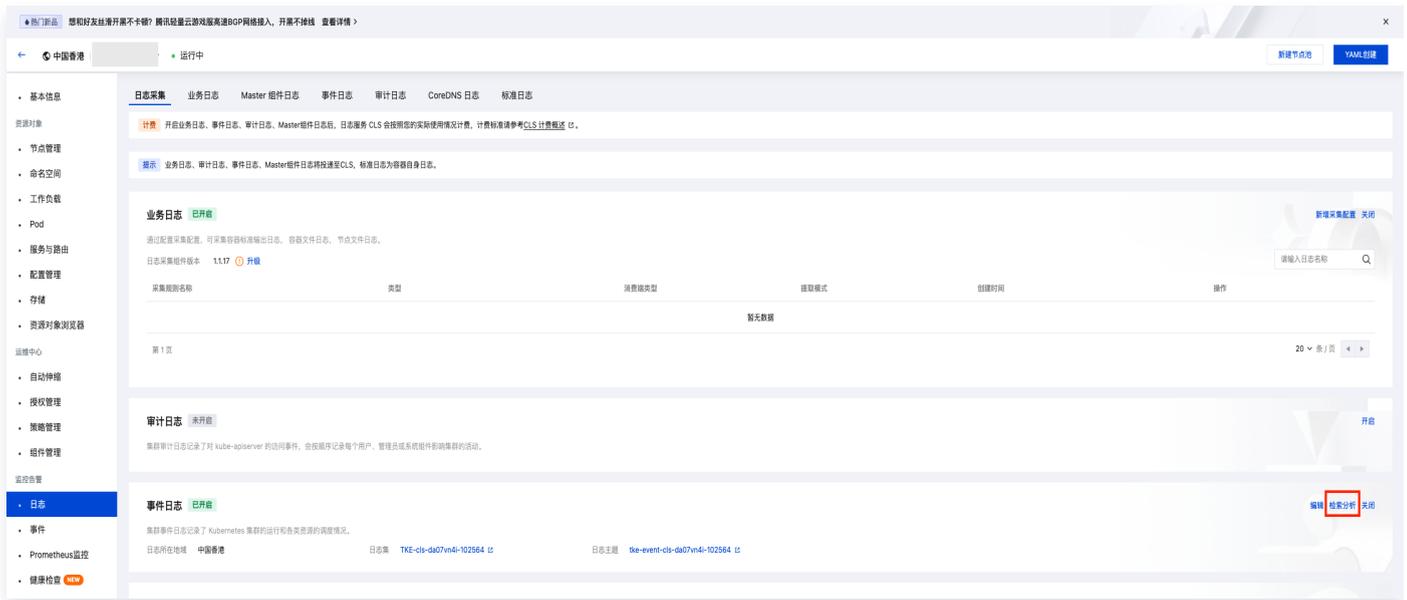


4. 在关闭事件日志页面，单击**确定**，即可关闭事件日志。如果不再需要事件日志，可勾选删除日志数据。如下图所示：



## 查看事件

1. 登录 [容器服务控制台](#)，在左侧导航栏中选择集群。
2. 在集群列表中，单击目标集群 ID，进入集群详情页。
3. 选择左侧导航栏中的日志，在日志采集页签中单击事件日志右侧的检索分析。如下图所示：



4. 在事件日志页面，单击全局检索，按需自行配置显示字段并进行检索分析。

**说明：**

当您开启事件日志时，将默认为您的 Topic 开启索引。详情请参见 [日志检索分析](#)。

# 事件仪表盘

最近更新时间：2025-03-05 14:06:02

## 操作场景

TKE Serverless 为用户提供了开箱即用的事件仪表盘。在 Serverless 集群开启事件日志功能后，TKE Serverless 将自动为集群配置各类事件总览大盘和异常事件的聚合检索分析仪表盘。还支持用户自定义配置过滤项，同时内置 CLS 的事件全局检索，实现在容器服务控制台 全面观测、查找、分析、定位问题的能力。

## 功能介绍

事件检索中配置了三个大盘，分别是“事件总览”、“异常事件聚合检索”、“全局检索”。请按照以下步骤进入“事件检索”页面，开始使用对应功能：

1. 登录 [容器服务控制台](#)。
2. 开启“事件日志功能”，详情请参见 [事件日志](#)。
3. 在左侧导航栏中，选择[运维中心](#) > [日志管理](#)，在日志规则页面选择地域和集群类型。
4. 在[事件日志](#)页签查看集群事件详情。

## 事件总览

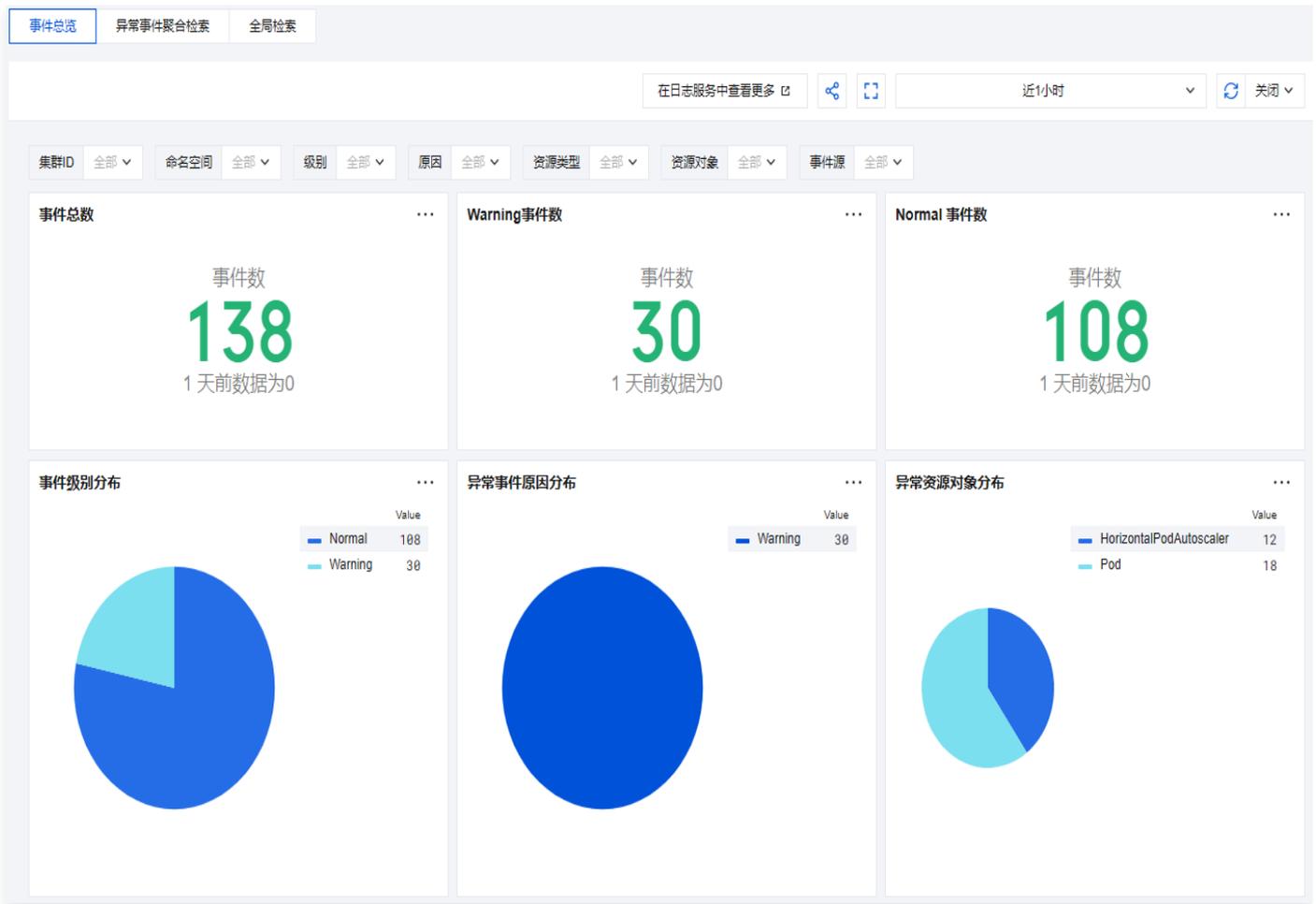
在[事件日志](#) > [事件总览](#)页面，可根据集群 ID、命名空间、级别、原因、资源类型、资源对象、事件源等维度过滤事件，查看核心事件的汇总统计信息，并展示一个周期内的数据对比。例如，事件总数及分布情况、节点异常、Pod OOM、重要事件趋势等仪表盘以及异常 TOP 事件列表。

在过滤项中，您可根据自己需求进行个性化配置。如下图所示：

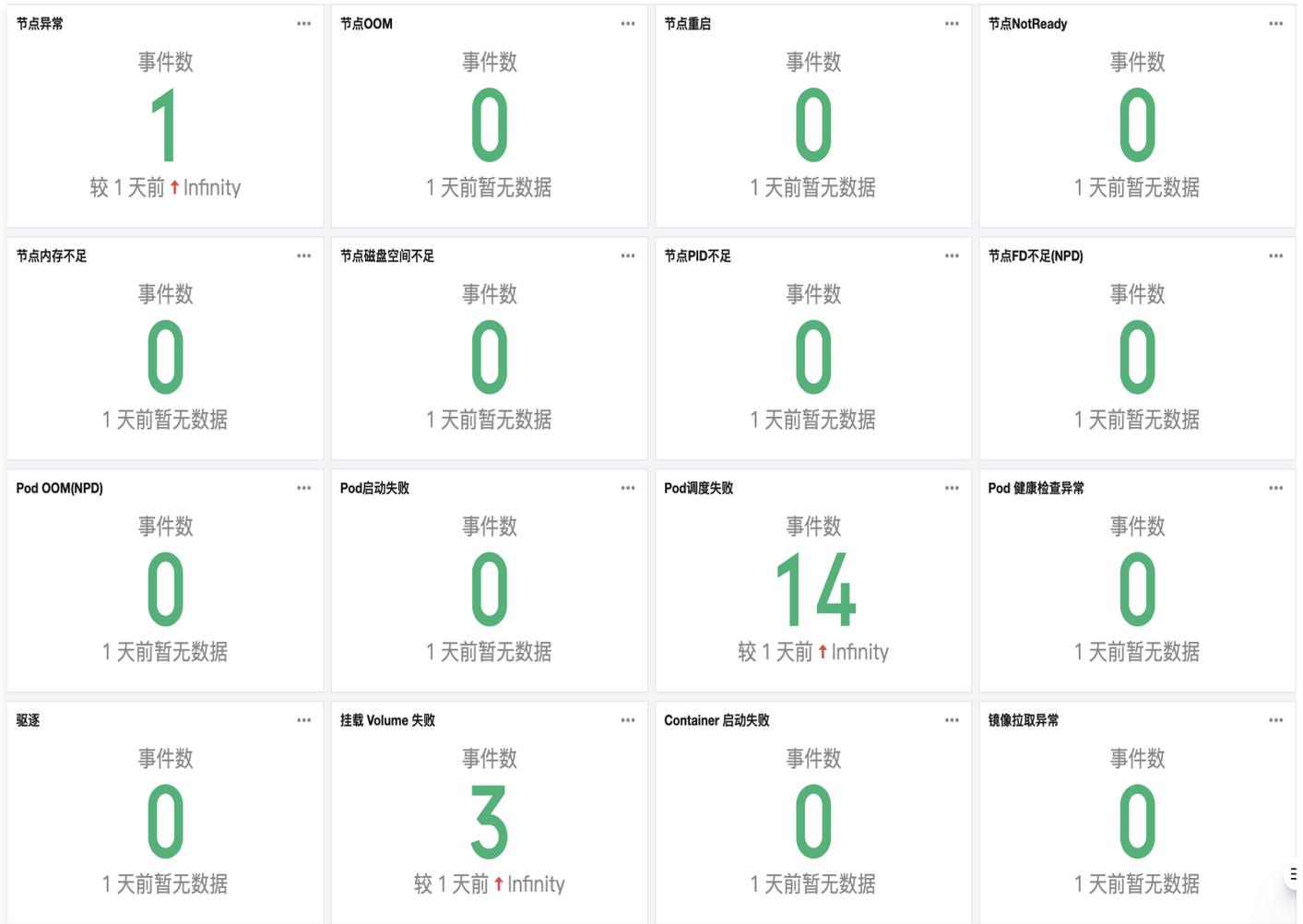


您还可在该页面中查看更多统计信息，如下所示：

- 事件总数及级别分布情况，异常事件的原因、对象分布情况检索如下图所示：



- 各类常见事件的汇总信息检索如下图所示：

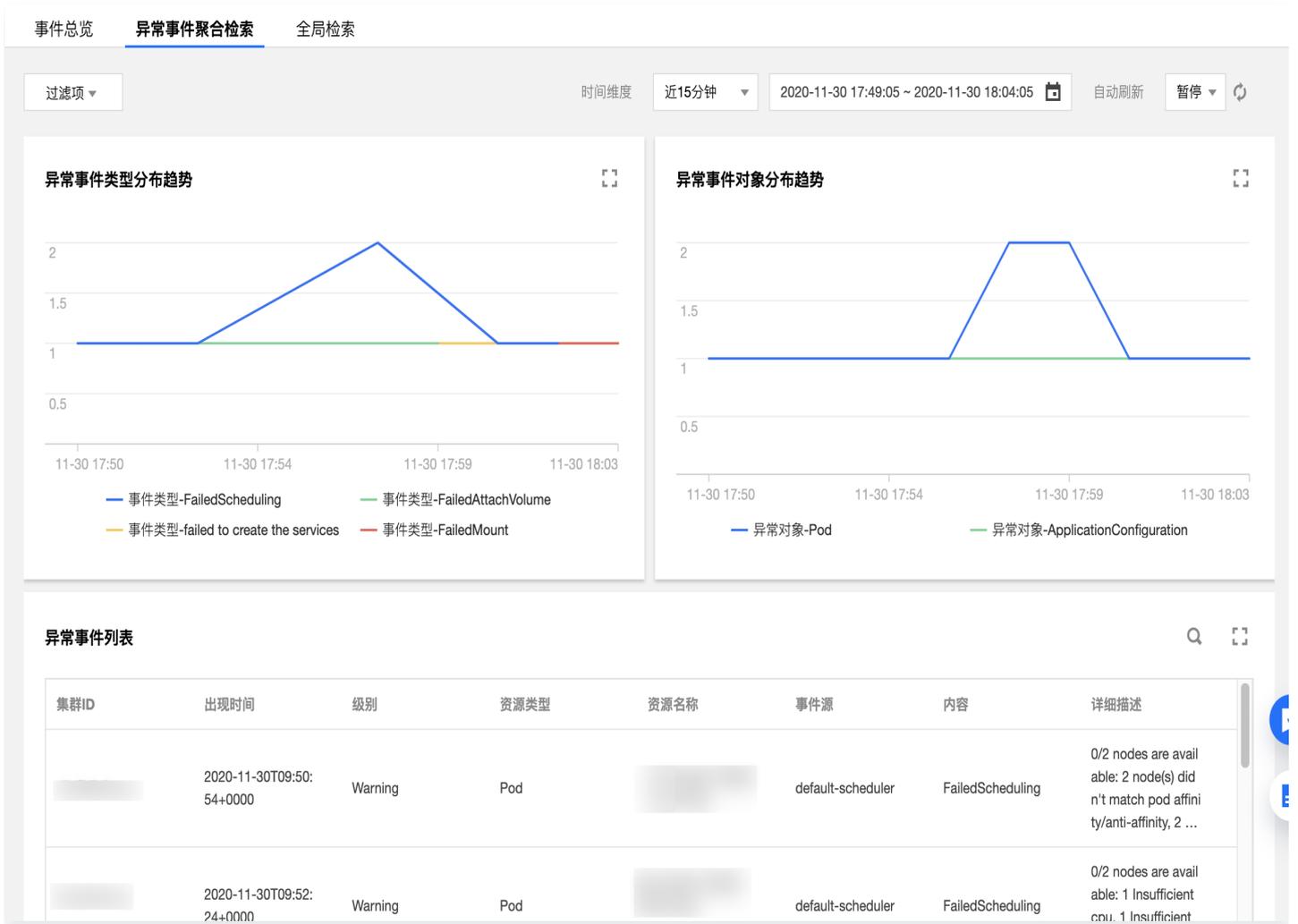


- 事件趋势及异常 TOP 事件列表检索如下图所示：



## 异常事件聚合检索

在“异常事件聚合检索”页面设置过滤条件，查看某个时间段内各类异常事件的 reason 和 object 分布趋势。在趋势下方展示了可供搜索的异常事件列表，帮助您快速定位到问题。如下图所示：



## 全局检索

全局检索仪表盘内嵌了 CLS 的检索分析页面，方便用户在容器服务控制台也能快速检索全部事件。如下图所示：

事件总览 异常事件聚合检索 全局检索

所属日志主题信息：广州 / TKE-cls- / tke-event-cls-

交互模式 收藏夹 历史记录 语句模板 推荐仪表盘 告警 采集配置 索引配置 更多

+ 添加检索条件 近15分钟

+ 添加统计语句

原始日志 统计图表 添加到仪表盘 添加告警策略 下载

字段列表 搜索 显示原始日志 可用字段

clusterId event.count event.eventTime event.firstTimestamp

日志条数 39

原始 表格 换行 行号 日志时间 更多配置 全屏查看

| 行号 | 日志时间               | 原始日志  |
|----|--------------------|---|
| 1  | 02-11 17:07:30.000 | clusterId: c1s- event.firstTimestamp: 2025-02-11T03:27:15Z event.reason: FailedGetResourceMetric event.metadata: {"name": "tke-kube-state-metrics-", "namespace": "kube-system", "uid": "-", "resourceVersion": "26290347", "creationTimestamp": "2025-02-11T03:27:15Z", "managedFields": [{"manager": "kube-controller-manager", "operation": "Update", "apiVersion": "v1", "time": "2025-02-11T09:07:29Z"}]} event.involvedObject.uid: event.involvedObject.apiVersion: autoscaling/v2 event.involvedObject.kind: HorizontalPodAutoscaler event.involvedObject.reso |

## 基于仪表盘配置告警

您可以通过以上预设的仪表盘配置告警，达到您所设置的条件则触发告警。操作详情如下：

1. 单击需要配置告警的仪表盘右侧的**添加到监控告警**。如下图所示：

Normal 事件数

事件数

108

1 天前数据为0

- 预览检索分析语句
- 在检索分析查看
- 添加到监控告警
- 保存为指标
- 全屏查看
- 数据导出

2. 在 **日志服务控制台** 中，选择**监控告警 > 告警策略**，新建告警策略。详情请参见 **配置告警策略**。

# 监控和告警

最近更新时间：2024-04-22 15:37:11

本文档介绍 TKE Serverless 集群提供的集群、工作负载、Pod、Container 4个层面的监控数据收集和展示功能。

## 前提条件

已创建状态为“运行中”的 Serverless 集群，详情请参见 [创建集群](#)。

## 操作步骤

1. 登录容器服务控制台，选择左侧导航栏中的 [集群](#)。
2. 在集群管理页面，选择 Serverless 集群 ID。
3. 在集群资源管理页面，参考以下文档查看监控及设置告警：
  - [查看监控数据](#)
  - [设置告警](#)

## 监控及告警指标列表

### 监控

目前 TKE Serverless 集群提供了以下维度的监控指标，所有指标均为统计周期内的**平均值**。

#### ⚠ 注意

- 工作负载使用的 PV 的详细监控指标请参考 [云硬盘监控](#)、[文件存储监控](#)。
- Service 关联的 CLB 详细的网络监控指标请参考 [负载均衡监控](#)。
- 腾讯云可观测平台创建告警策略指引，详情请参见 [创建告警策略](#)。

### 集群监控指标

| 指标      | 单位 | 说明                       |
|---------|----|--------------------------|
| CPU 使用量 | 核  | 集群当前所有运行中 Pod 使用的 CPU 规模 |
| 内存使用量   | B  | 集群当前所有运行中 Pod 使用的内存规模    |

### 工作负载监控指标

| 指标       | 单位 | 说明                    |
|----------|----|-----------------------|
| Pod 重启次数 | 次  | 工作负载内所有 Pod 的重启次数之和   |
| CPU 使用量  | 核  | 工作负载内所有 Pod 的 CPU 使用量 |

|                   |   |                                |
|-------------------|---|--------------------------------|
| CPU 利用率（占 Pod 规格） | % | 工作负载内所有 Pod 的 CPU 使用量占分配资源总量之比 |
| 内存使用量             | B | 工作负载内所有 Pod 的内存使用量             |
| 内存利用率（占 Pod 规格）   | % | 工作负载内所有 Pod 的内存使用量占分配资源总量之比    |

## Pod 监控指标

| 指标                 | 单位 | 说明                            |
|--------------------|----|-------------------------------|
| 异常状态               | -  | Pod 的状态，正常或异常                 |
| CPU 使用量            | 核  | Pod 的 CPU 使用量                 |
| CPU 利用率（占 Request） | %  | Pod 的 CPU 使用量和设置的 Request 值之比 |
| CPU 利用率（占 Limit）   | %  | Pod 的 CPU 使用量和设置的 Limit 值之比   |
| CPU 利用率（占 Pod 规格）  | %  | Pod 的 CPU 使用量占 Pod 分配总量之比     |
| 内存使用量              | B  | Pod 的内存使用量，含缓存                |
| 内存利用率（占 Request）   | %  | Pod 的内存使用量和设置的 Request 值之比    |
| 内存利用率（占 Limit）     | %  | Pod 的内存使用量和设置的 Limit 值之比      |
| 内存利用率（占 Pod 规格）    | %  | Pod 的内存使用量占 Pod 分配总量之比        |

## Container 监控指标

| 指标                 | 单位 | 说明                                  |
|--------------------|----|-------------------------------------|
| CPU 使用量            | 核  | Container 的 CPU 使用量                 |
| CPU 利用率（占 Request） | %  | Container 的 CPU 使用量和设置的 Request 值之比 |
| CPU 利用率（占 Limit）   | %  | Container 的 CPU 使用量和设置的 Limit 值之比   |
| 内存使用量              | B  | Container 的内存使用量，含缓存                |
| 内存利用率（占 Request）   | %  | Container 的内存使用量和设置的 Request 值之比    |
| 内存利用率（占 Limit）     | %  | Container 的内存使用量和设置的 Limit 值之比      |

## 告警

目前 TKE Serverless 集群提供了以下维度的告警指标，所有指标均为统计周期内的平均值。

## Pod 告警指标

| 指标                 | 单位 | 说明                            |
|--------------------|----|-------------------------------|
| CPU 利用率（占 Pod 规格）  | %  | Pod 的 CPU 使用量占 Pod 分配总量之比     |
| 内存利用率（占 Pod 规格）    | %  | Pod 的内存使用量占 Pod 分配总量之比        |
| CPU 利用率（占 Request） | %  | Pod 的 CPU 使用量和设置的 Request 值之比 |
| 内存利用率（占 Request）   | %  | Pod 的内存使用量和设置的 Request 值之比    |
| CPU 利用率（占 Limit）   | %  | Pod 的 CPU 使用量和设置的 Limit 值之比   |
| 内存利用率（占 Limit）     | %  | Pod 的内存使用量和设置的 Limit 值之比      |
| Pod 重启次数           | 次  | Pod 的重启次数                     |
| Pod Ready          | -  | Pod 的状态，默认 False 时告警          |
| CPU 使用量            | 核  | Pod 的 CPU 使用量                 |
| 内存使用量              | MB | Pod 的内存使用量，含缓存                |

# 常见问题

最近更新时间：2024-11-27 17:59:22

## TKE Serverless 集群相关问题

- [为什么 Pod 规格与填写的 Request/Limit 不一致？](#)
- [如何新增或修改 TKE Serverless 集群的容器网络？](#)
- [Pod 因子网 IP 耗尽调度失败时，该如何处理？](#)
- [TKE Serverless 集群安全组使用指引和说明有哪些？](#)
- [如何设置容器终止消息？](#)
- [如何使用 Host 参数？](#)
- [如何挂载 CFS/NFS？](#)
- [如何通过镜像复用加快容器启动速度？](#)
- [复用镜像异常问题说明](#)
- [挂载自建的 nfs 时，事件报 Operation not permitted](#)
- [Pod 磁盘满了（ImageGCFailed）如何处理？](#)
- [9100 端口问题](#)

## 负载均衡相关问题

- [TKE Serverless 会为哪些 Ingress 创建 CLB 实例？](#)
- [如何查看 TKE Serverless 为 Ingress 创建的 CLB 实例？](#)
- [TKE Serverless 会为哪些 Service 创建 CLB 实例？](#)
- [如何查看 TKE Serverless 为 Service 创建的 CLB 实例？](#)
- [为什么 Service 的 ClusterIP 无效（无法正常访问）或没有 ClusterIP？](#)
- [如何指定 CLB 实例类型（公网或内网）？](#)
- [如何指定使用已有 CLB 实例？](#)
- [如何查看 CLB 实例的访问日志？](#)
- [为什么 TKE Serverless 没有为 Ingress 或 Service 创建 CLB 实例？](#)
- [如何在多个 Service 中使用相同的负载均衡？](#)
- [为什么访问 CLB VIP 时失败？](#)

## 超级节点相关问题

- [如何禁止 Pod 调度到某个超级节点？](#)
- [如何禁止 TKE 普通集群在资源不足时自动调度到超级节点？](#)
- [如何手动调度 Pod 到超级节点？](#)
- [如何强制调度 Pod 到超级节点，无论超级节点是否支持该 Pod？](#)
- [如何自定义超级节点 DNS？](#)

## 镜像仓库相关问题

- [TKE Serverless 集群如何使用容器镜像服务 TCR?](#)
- [TKE Serverless 集群如何使用自建的自签名镜像仓库或 HTTP 协议镜像仓库?](#)

## 日志采集相关问题

- [集群配置日志采集后，为什么在日志服务控制台查看不到日志?](#)
- [配置好日志规则后，日志在哪查看?](#)

## Prometheus 监控相关问题

- [TKE Serverless 集群如何对接腾讯云原生 Prometheus 监控?](#)
- [TKE Serverless 集群如何对接自建 Prometheus?](#)