ALIBABA CLOUD

阿里云

容器服务Kubernetes版
边缘容器服务ACK@Edge用户指南

文档版本：20211129

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.ACK@Edge概述

阿里云边缘容器服务ACK@Edge是阿里云容器服务针对边缘计算场景推出的云边一体化协同托管方案。本文介绍阿里云边缘托管Kubernetes集群的产生背景和主要功能。

产品简介

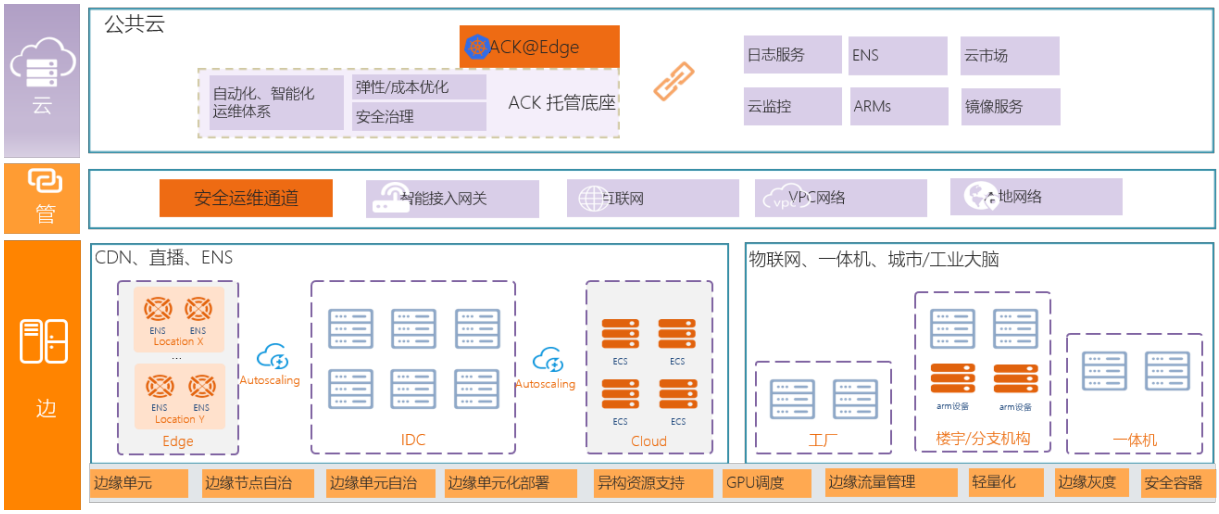
随着互联网智能终端设备数量的急剧增加，以及5G和物联网时代的到来，传统云计算中心集中存储、计算的模式已经无法满足终端设备对于时效、容量、算力的需求，将云计算的能力下沉到边缘侧、设备侧，并通过中心进行统一交付、运维、管控，将是云计算的重要发展趋势。

阿里云边缘托管Kubernetes集群在云端提供一个标准、安全、高可用的Kubernetes集群，整合阿里云虚拟化、存储、网络和安全等能力，并简化集群运维工作，让您专注于容器化应用的开发与管理。ACK@Edge具有以下特点：

- 支持云端托管，帮助您快速构建边缘计算的云原生基础设施。
- 支持多种边缘计算资源的快速接入，包括IoT网关设备、端设备、CDN资源、自建IDC资源等。
- 支持X86和ARM架构。
- 支持丰富的应用场景，包括边缘智能、智慧楼宇、智慧工厂、音视频直播、在线教育、CDN等。

阿里云边缘托管Kubernetes集群，采用非侵入方式增强，提供边缘自治、边缘单元、边缘流量管理、原生运维API支持等能力，以原生方式支持边缘计算场景下的应用统一生命周期管理和统一资源调度。

功能介绍



阿里云边缘托管Kubernetes集群支持对边缘计算场景的容器应用和资源全生命周期管理，具有以下功能：

- 通过控制台一键创建高可用的边缘Kubernetes集群，并提供集群的扩容、升级、日志、监控等生命周期管理运维能力。
- 支持丰富的异构边缘节点资源，包括自建IDC资源、ENS、IoT设备、X86、ARM架构等；并支持异构资源的混合调度。
- 面向边缘计算弱网络连接场景，提供节点自治和网络自治能力，保证边缘节点和边缘业务的高可靠运行。
- 提供反向运维网络通道能力。
- 提供边缘单元管理、单元化部署、单元流量管理能力。

ACK@Edge交流群

如果您对于ACK@Edge有任何疑问，欢迎使用钉钉扫描二维码或者搜索群号21976595加入钉钉交流群。

相关文档

- 创建边缘托管版集群
- 升级边缘集群
- 扩容边缘集群

2.ACK@Edge计费说明

阿里云边缘容器服务ACK@Edge收取节点的管理费用，同时您在使用过程中创建的相关资源，也会向您收取相应的资源费用。本文介绍边缘容器服务计费说明、欠费说明以及计费常见问题。

ACK@Edge标准版计费说明

- 边缘容器收取节点的管理费用。例如，您的集群下有3台4核8GB的节点（无论是ECS、ENS、线下节点），边缘容器服务向您收取的费用为0.2*3*4 = 2.4元/天，使用时间不足一天，按一天计算。

计费方式	价格
按量计费	0.2元/vCPU/天

- 在使用边缘集群过程中所创建的相关阿里云产品资源，例如ECS、ENS、SLB、NAT网关等，按照相应资源的价格计费。
  - 云服务器计费概览
  - 边缘节点服务ENS计费概览
  - NAT网关计费说明
  - 负载均衡按量计费

ACK@Edge Pro版计费说明

Edge Pro版集群额外收取集群管理费，同时在使用过程中所创建的其他相关云产品资源，也会向您收取相应的费用。

计费方式	价格
按量计费	每个集群0.64元/小时

欠费说明

- 计费出账

  边缘容器服务费用的计费周期为24小时，即阿里云将在下一个自然日就您上一个自然日的服务使用进行计量、出具账单。并从您的阿里云账户中按账单金额扣划服务费用。账单出账时间通常在当前计费周期结束后8至10个小时内。

- 欠费释放

  如果您的账户余额不足以支付账单金额，边缘容器服务将处于欠费状态。欠费24小时内，边缘容器服务仍可正常提供服务。如果欠费达到24小时并且仍未缴清账单，边缘容器服务暂停服务状态（您将无法访问集群API Server，但节点上的业务仍可继续运行），计费也将停止。如超过7天仍处于欠费状态，阿里云将从您的集群中移除相应节点（但不会释放）并删除您的边缘托管集群，由于集群删除造成的容器实例的释放不可恢复。

计费常见问题

- 为什么会突然增加或减少每天的账单金额？
  - 如果您在使用过程中扩容或缩容了节点，系统会按照集群所管理的节点的总vCPU数调整计费，在第二天的出账中体现新的费用。

- 当未通过容器服务控制台移除节点时（如：使用 `kubectl delete node` 移除节点），被移除的节点在移除当天有可能还会被计费。因此请通过容器服务控制台移除节点。

- 删除了边缘集群，为什么今天还会收到账单并被扣费？

  边缘集群费用以每天0时0分0秒至23时59分59秒为周期进行计费，第二天收费。所以昨天删除边缘集群，昨天的计费系统已经计入，今天会出账单进行扣费，明天就不会出账单扣费了。

- 如何停止边缘集群的计费？

  删除所有地域的边缘集群，删除集群前请注意备份应用和数据。

- NotReady状态的节点是否会计费？

  节点状态不论是Ready还是NotReady，边缘集群都会对节点进行管理，因此NotReady状态的节点仍然会计费。您必须通过容器服务控制台移除节点，相关节点才不会计费。

ACK@Edge交流群

如果您对于ACK@Edge有任何疑问，欢迎使用钉钉扫描二维码或者搜索群号21976595加入钉钉交流群。

3.ACK@Edge版本发布说明

3.1. ACK@Edge发布Kubernetes 1.18版本说明

阿里云边缘容器服务ACK@Edge是基于容器服务ACK针对边缘计算场景推出的云边一体化托管方案。本文介绍ACK@Edge发布Kubernetes 1.18版本所做的变更内容。

云边运维通道和运维监控

该版本对云边运维通道和运维监控方案进行了优化，具体如下：

- tunnel-server拦截并处理边缘运维监控流量的方式，从基于单机iptables规则改为基于集群内DNS解析。
- 依赖通道能力的监控组件metrics-server、prometheus等，可以不必与tunnel-server部署在同一节点。
- tunnel-server支持多副本部署并实现全局负载均衡。
- 云边运维通道增加 `meta server` 模块，用于处理 `prometheus metrics` 和 `debug/pprof` ，其中tunnel-server的访问端点为 `http://127.0.0.1:10265` ，edge-tunnel-agent的访问端点为 `http://127.0.0.1:10266` ，访问端点中的端口可以通过组件启动参数 --meta-port配置。

边缘节点自治

该版本对边缘缓存、健康检查、服务端点及流量统计等进行优化，同时对边缘流量自治和边缘侧应用通过InCluster模式访问kube-apiserver进行了增强，具体如下：

- edge-hub支持边缘侧Service流量拓扑功能，该功能不再依赖Kubernetes相关FeatureGate。
- edge-hub自动修改边缘侧 `kubernetes service` 的endpoint为kube-apiserver的公网endpoint，支持边缘侧应用通过InCluster模式访问Kubernetes集群。
- edge-hub支持缓存CRD资源，例如：用于保存Flannel网络信息CRD（ `nodenetworkconfigurations` ）的缓存。
- edge-hub对云端健康检查的机制进行了优化，使用 `Lease` 心跳替换 `healthz` 请求。
- edge-hub监听端口拆分，由 `10261` 拆分为 `10261` 和 `10267` ，其中 `10261` 用于处理请求转发， `10267` 用于处理edge-hub本地请求（例如： `yurthub` 的 `liveness probe` 、 `metrics` 和 `pprof` 等）。
- edge-hub增加 `metrics` 指标：node_edge_hub_proxy_traffic_collector，用于展示边缘节点上各个组件（例如：kubelet、kube-proxy）访问Kubernetes资源（例如：Pod、Deployment）时所产生的流量。

边缘单元化管理

单元化部署（UnitedDeployment）新增Patch功能，该功能支持对每个节点池NodePool的部署配置做定制。例如，当同一个UnitedDeployment中，不同节点池使用各不相同的本地镜像仓库时，您可以通过设置Patch字段，修改每个NodePool所使用的镜像地址。

边缘节点接入

边缘集群支持接入OS是Ubuntu 20.04的节点。

边缘网络

- Flannel云边网络流量优化：不再list-watch node，改为list-watch自定义CRD，从而降低云边网络流量。

- 边缘流量管理Annotation调整
  - 1.16版本的相关**Annotation Key**参数说明如下：

Annotation Key	Annotation Value	说明
openyurt.io/topologyKeys	kubernetes.io/hostname	限制Service只能被本节点访问。
openyurt.io/topologyKeys	kubernetes.io/zone	限制Service只能被本节点池的节点访问。
无	无	对Service不做任何拓扑限制。

  - 在1.18版本，对上表第二行中的参数openyurt.io/topologyKeys的**Annotation Value**做了调整。该参数支持两个值：`kubernetes.io/zone` 和 `openyurt.io/nodepool`，这两个值都用于限制Service只能被本节点池的节点访问，且推荐您使用 `openyurt.io/nodepool`。

3.2. ACK@Edge发布Kubernetes 1.16版本说明

阿里云边缘容器服务ACK@Edge是基于容器服务ACK针对边缘计算场景推出的云边一体化托管方案。本文介绍ACK@Edge发布Kubernetes 1.16版本所做的变更内容。

Kubernetes Core

该版本在容器服务ACK 1.16版本之上，针对边缘计算场景做的变更主要为以下两点：

- 修复当节点*cpuacct.stat*文件中超出4条记录时**kubelet**启动失败问题。
- Kube-Proxy支持IPVS（IP Virtual Server）模式。
- **kubelet**支持通过指定网卡名来配置节点InternalIP。

更多关于ACK版本变更信息，请参见ACK发布Kubernetes 1.16版本说明。

边缘节点自治

该版本边缘节点自治在稳定性方面做了加强，主要变更包括以下几个方面：

- 缓存数据丢失时，Client获取的返回数据从空字符串修改为404。
- **edge-hub**证书存储目录从*/etc/kubernetes/edge-hub*调整为*/var/lib/edge-hub*。
- **edge-hub**证书名称从**edge-hub.kubeconfig**调整为**edge-hub.conf**, **bootstrap-edge-hub-current.conf --> bootstrap-hub.conf**。
- 增加**prometheus metrics**接口。
- **iptables**性能优化：为127.0.0.1:10261和169.254.2.1:10261地址增加**iptables notrack**。

更多信息，请参见边缘网络自治。

云边运维通道

该版本对云边运维通道的性能进行了优化，主要变更包括以下几个方面：

- 隧道底层通信库协议从普通TCP协议切换成**gRPC**，基于**gRPC**的压缩效率，云边通信数据量最大可减少40%。
- 为**edge-tunnel-agent**增加证书申请及自动更新功能，解耦了对节点证书依赖，同时**edge-tunnel-agent**证书存储目录修改为*/var/lib/edge-tunnel-agent/pki*。
- 增加**prometheus metrics**。

- edge-tunnel-agent Pod部署依赖节点Label调整为 `alibabacloud.com/is-edge-worker: "true"` 。

更多信息，请参见边缘运维通道。

运维监控组件

该版本对metrics-server进行了版本升级，同时安全性也进行增强，主要变更包括以下两点：

- metrics-server版本从V0.2.1升级到V0.3.8。
- 云监控对接方式调整为更通用的Token对接。

边缘单元化管理

边缘单元化管理是该版本新增功能，增加新的组件yurt-app-manager ，主要特性包括以下几个方面：

- 通过NodePool实现节点的单元化管理。
- 通过UnitedDeployment实现应用的单元化部署。
- 通过Service拓扑配置来实现流量在节点池内或节点闭环。

更多关于边缘单元化管理的信息，请参见边缘节点池概述。

增强型网络节点池

增强型网络节点池是该版本的新增功能，主要特性包括以下两个方面：

- 提供更稳定、更安全的云边通信通道。
- 支持私网环境的边缘应用通过容器网络与云端应用通信。

更多信息，请参见创建增强型网络边缘节点池。

容器运行时

该版本容器运行时主要变更包括以下两点：

- ARM和ARM64 runC版本升级到1.0.0-rc10。
- Cgroup Driver由Cgroupfs变更为Systemd。

CNI插件

该版本增强了CNI插件的稳定性，修复不同命名空间中相同Pod名导致Pod IP分配失败的问题。

边缘节点接入

该版本对边缘节点接入流程进行了优化并新增接入参数，主要包括以下几个方面：

- 优化边缘节点接入流程，新增CIDR（Classless Inter-Domain Routing）冲突校验。
- 支持给节点设置可分配的IP数量。
- 新增labels、nodeIface、annotations和taints等参数。
- 支持Ubuntu 5.4内核系统接入。

更多信息，请参见添加边缘节点。

OpenAPI变更

节点池API支持边缘节点池，更多信息，请参见节点池。

4.边缘托管集群管理

4.1. 创建边缘托管版集群

边缘托管集群服务从云到端将云计算的能力下沉到边缘侧、终端设备侧，并通过容器服务控制台进行统一交付、运维、管控，通过粘合云计算核心能力和边缘算力，是构筑在边缘基础设施之上的云计算平台。您可以通过容器服务控制台非常方便地创建Kubernetes边缘托管版集群。

前提条件

您需要开通容器服务、弹性伸缩（ESS）服务和访问控制（RAM）服务。

登录容器服务管理控制台、RAM管理控制台和弹性伸缩控制台开通相应的服务。

> 说明
>
> - 用户账户需有100元的余额并通过实名认证，否则无法创建按量付费的ECS实例和负载均衡。
> - 随集群一同创建的负载均衡实例只支持按量付费的方式。
> - Kubernetes集群仅支持专有网络VPC。
> - 每个账号默认可以创建的云资源有一定的配额，如果超过配额创建集群会失败。请在创建集群前确认您的配额。如果您需要提高您的配额，请提交工单申请。
>   - 每个账号默认最多可以创建100个安全组。
>   - 每个账号默认最多可以创建60个按量付费的负载均衡实例。
>   - 每个账号默认最多可以创建20个EIP。

背景信息

随着互联网智能终端设备数量的急剧增加以及数据和业务下沉的诉求增多，边缘计算规模和业务复杂度已经发生了翻天覆地的变化，边缘智能、边缘实时计算、边缘分析等新型业务不断涌现。传统云计算中心集中存储、计算的模式已经无法满足边缘设备对于时效、容量、算力的需求。边缘托管版是针对边缘计算场景推出的云边一体化协同托管方案。该类型托管集群在“云端提供一个标准、安全、高可用的Kubernetes集群，整合阿里云虚拟化、存储、网络和安全等能力，并简化集群运维工作，让您专注于容器化的应用的开发与管理。同时，在边缘端支持各种异构边缘计算力快速接入（边缘设备被云端的管控中心接管），涵盖IoT网关设备、终端设备、CDN资源、自建IDC资源等，支持X86和ARM架构。目前边缘托管版集群已经广泛应用于边缘智能、智慧楼宇、智慧工厂、音视频直播、在线教育、CDN等领域。

操作步骤

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面中，单击页面右上角的**创建集群**。
4. 单击**ACK边缘托管版**页签，然后完成集群配置。

   基础选项配置

配置项	描述

配置项	描述
集群名称	填写集群的名称。 **说明** 集群名称应包含1~63个字符，可包含数字、汉字、英文字符或短划线（-）。
集群规格	选择集群规格，支持**标准版**和**Pro版**。 选中**标准版**创建ACK@Edge集群。
地域	选择集群所在的地域。
资源组	将鼠标悬浮于页面上方的**账号全部资源**，选择资源组。在控制台页面顶部选择的资源组可过滤出该资源组内的专有网络及对应的虚拟交换机。在创建集群时，只显示过滤的专有网络实例及专有网络对应的虚拟交换机实例。 
Kubernetes版本	显示当前支持的Kubernetes版本。
专有网络	设置集群的网络，您可以选择普通VPC和共享VPC。 • 共享VPC：VPC的所有者账号（资源所有者）可以将其账号下的VPC内的交换机资源共享给其组织内的其他账号使用。 • 普通VPC：不具备共享功能的VPC。 **说明** Kubernetes集群仅支持专有网络。您可以在已有VPC列表中选择所需的VPC。如果没有您需要的专有网络，可以通过单击**创建专有网络**进行创建，请参见创建和管理专有网络。
虚拟交换机	设置虚拟交换机。 您可以在已有虚拟交换机列表中，根据**可用区**选择1~3个交换机。如果没有您需要的交换机，可以通过单击**创建虚拟交换机**进行创建，请参见使用交换机。

配置项	描述
节点IP数量	如果您选择的网络模式为**Flannel**，您需设置**节点IP数量**。 **说明** ○ **节点IP数量**是指可分配给一个节点的IP数量，建议保持默认值。 ○ 根据您所选择的专有网络和节点IP数量，ACK将为您推荐可用的Pod网络CIDR和Service CIDR，并给出相应配置下集群内可允许部署的主机数量以及每台主机可容纳的Pod数量。请您根据集群规模的实际需求，在推荐配置的基础上进行修改。
Pod网络CIDR	网络插件选择**Flannel**时，需要配置**Pod网络CIDR**。 **Pod网络CIDR**和**Service CIDR**两者都不能与VPC及VPC内已有Kubernetes集群使用的网段重复，创建成功后不能修改。且Service地址段也不能和Pod地址段重复，有关Kubernetes网络地址段规划的信息，请参见Kubernetes集群网络规划。
Service CIDR	
配置SNAT	创建集群时，默认不开通公网。如果您选择的VPC不具备公网访问能力，选中**为专有网络配置SNAT**后，ACK将为您创建NAT网关并自动配置SNAT规则。
API Server访问	ACK默认为API Server创建一个内网SLB实例，您可修改SLB实例规格。更多信息，请参见实例规格。 **注意** 删除默认创建的SLB实例将会导致无法访问API Server。 您可设置是否开放**使用EIP暴露API Server**。API Server提供了各类资源对象（Pod，Service等）的增删改查及Watch等HTTP Rest接口。 ○ 如果选择开放，ACK会创建一个EIP，并挂载到SLB上。此时，Master节点的6443端口（对应API Server）暴露出来，您可以在外网通过kubeconfig连接并操作集群。 ○ 如果选择不开放，则不会创建EIP，您只能在VPC内部用kubeconfig连接并操作集群。 **说明** 通常边缘节点需要通过公网和云端API server交互，因此若不勾选**使用EIP暴露API Server**，边缘节点将无法连接到云端集群，所创建集群也将无法在边缘场景下使用。
RDS白名单	设置RDS白名单。将节点IP添加到RDS实例的白名单中。 **说明** 允许白名单RDS访问Kubernetes集群，RDS必须在当前集群的VPC内。

配置项	描述
安全组	支持选择**自动创建普通安全组**、**自动创建企业级安全组**、**选择已有安全组**。有关安全组的详细内容，请参见安全组概述。 **说明** ○ 只有白名单用户可以使用**选择已有安全组**功能。请提交工单申请。 ○ 指定已有安全组时，系统默认不会为安全组配置额外的访问规则，可能会导致访问异常，请自行管理安全组规则。关于如何管理安全组规则，请参见最小化集群访问规则。
集群删除保护	设置是否启用集群删除保护。为防止通过控制台或API误释放集群。
资源组	创建的集群将归属于选择的资源组。一个资源只能归属于一个资源组。根据不同的业务场景，您可以将资源组映射为项目、应用或组织等概念。更多信息，请参见资源组。

高级选项配置

配置项	描述
kube-proxy代理模式	支持iptables和IPVS两种模式。 ○ iptables：成熟稳定的kube-proxy代理模式，Kubernetes Service的服务发现和负载均衡使用iptables规则配置，但性能一般，受规模影响较大，适用于集群存在少量的Service。 ○ IPVS：高性能的kube-proxy代理模式，Kubernetes Service的服务发现和负载均衡使用Linux ipvs模块进行配置，适用于集群存在大量的service，对负载均衡有高性能要求的场景。
标签	为集群绑定标签。输入键和对应的值，单击**添加**。 **说明** ○ 键是必需的，而*值*是可选的，可以不填写。 ○ 键不能是aliyun、http://、https://开头的字符串，不区分大小写，最多64个字符。 ○ *值*不能是http:// 或https://，可以为空，不区分大小写，最多128个字符。 ○ 同一个资源，标签键不能重复，相同标签键（Key）的标签会被覆盖。 ○ 如果一个资源已经绑定了20个标签，已有标签和新建标签会失效，您需要解绑部分标签后才能再绑定新的标签。

5. 单击**下一步：Worker配置**，完成Worker节点配置。

> **说明** 在Kubernetes边缘托管集群中，至少需要配置1个Worker节点，用于部署云端的管控组件。

配置项	描述
实例规格	支持选择多个实例规格。详情请参见实例规格族。 **说明** 边缘托管集群的日志、监控、反向通道等一些增强特性需要在云端部署组件，因此默认创建至少一个ECS实例作为Worker节点。
已选规格	呈现选中的规格。
数量	新增Worker实例（ECS实例）的数量。
系统盘	配置Worker节点系统盘。支持高效云盘、SSD云盘。 **说明**
挂载数据盘	支持**ESSD云盘**、**SSD云盘**和**高效云盘**。 **说明** ○ 支持选中**开启云盘备份**以备份云盘数据。 ○ **ESSD云盘**支持自定义**性能级别**。 ESSD云盘容量越大，可供选择的性能级别越高（460 GiB容量以上可选PL2，1260 GiB以上可选PL3）。更多信息，请参见容量范围与性能级别的关系。
登录方式	**说明** 当您勾选**云监控插件**或**日志服务**时，需要给ECS设置登录方式。
密钥对	

6. 单击**下一步：组件配置**，完成组件配置。

配置项	描述
云监控插件	设置是否启用云监控插件。您可以选中**在ECS节点上安装云监控插件**，从而在云监控控制台查看所创建ECS实例的监控信息。

配置项	描述
日志服务	设置是否启用日志服务，您可使用已有Project或新建一个Project。默认选中**使用日志服务**。创建应用时，您可通过简单配置，快速使用日志服务，详情参见通过日志服务采集Kubernetes容器日志。
工作流引擎	设置是否使用AGS。 **说明** 当前只有白名单用户可以使用该功能。 ○ 如果选中AGS，则创建集群时系统自动安装AGS工作流插件。 ○ 如果不选中，则需要手动安装AGS工作流插件，请参见AGS命令行帮助。

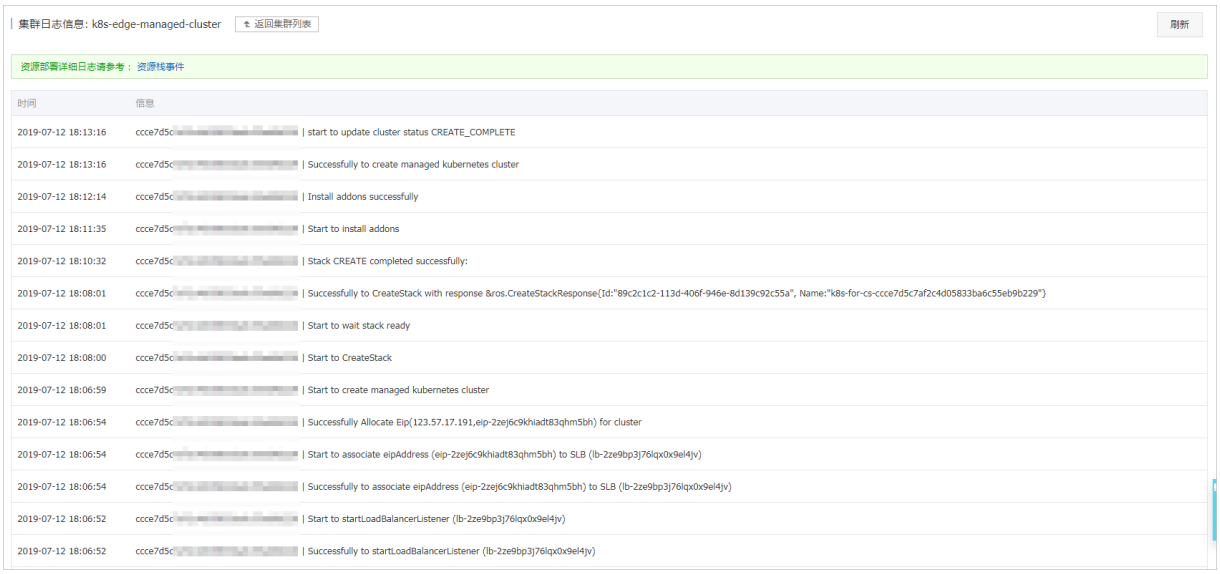
7. 单击**下一步：确认配置**。
8. 选择服务协议并单击**创建集群**。

说明 一个Kubernetes边缘托管版集群的创建时间一般约为十分钟。

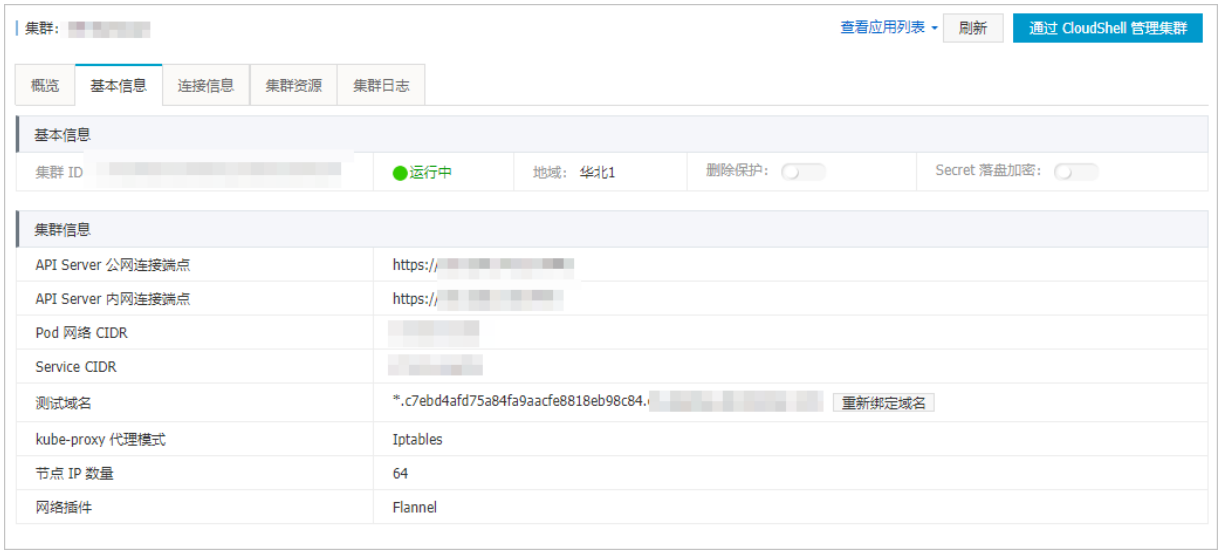
执行结果

集群创建成功后，您可以在容器服务管理控制台的**集群列表**页面查看所创建的集群。

您可以单击操作列的**查看日志**，进入**集群日志信息**页面查看集群的日志信息。您也可以在集群日志信息页面中，单击**资源栈事件**查看更详细的信息。



在**集群列表**页面中，找到刚创建的集群，单击集群名称或者**操作**列下的**详情**，单击**基本信息**和**连接信息**页签，查看集群的基本信息和连接信息。



其中：

- **API Server公网连接端点**：Kubernetes的API Server对公网提供服务的地址和端口，可以通过此服务在用户终端使用kubectl等工具管理集群。
- **API Service内网连接端点**：Kubernetes的API server对集群内部提供服务的地址和端口，此IP为负载均衡的地址。
- **测试域名**：为集群中的服务提供测试用的访问域名。服务访问域名后缀是 `<cluster_id>.<region_id>.alicontainer.com`

  > **说明** 单击**重新绑定域名**，您可以重新绑定访问域名。

您可以通过kubectl工具连接集群，执行 `kubectl get node` 查看集群的节点信息。

```
Type "kubectl" to manage your kubenetes cluster [redacted]

shell@Alicloud:~$ kubectl get node
NAME                                STATUS   ROLES    AGE   VERSION
cn-beijing.i-2zehvxttbua2aw800uin   Ready    <none>   11m   v1.12.6-aliyun.1
shell@Alicloud:~$
```

相关文档

- ACK@Edge概述
- 升级边缘集群
- 添加边缘节点

4.2. 升级边缘集群

您可以通过容器服务管理控制台，可视化升级您集群的Kubernetes边缘版版本。升级集群的过程包含升级前置检查、升级Master（独占版会展示当前正在升级的Master编号）、升级Node（会展示已经升级的节点数和总节点数）。

前提条件

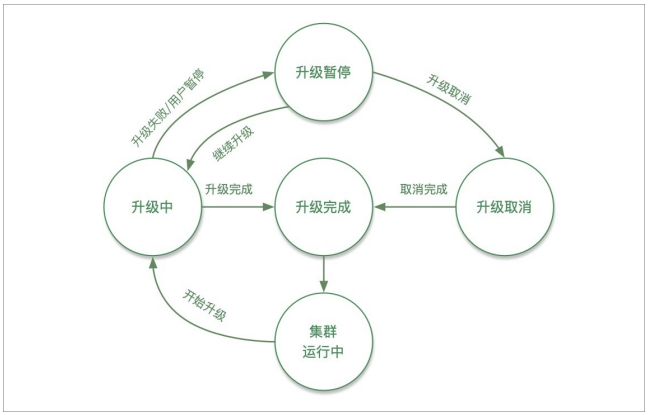
- 您已成功创建一个Kubernetes边缘版集群，请参见创建边缘托管版集群。
- 您已在本地安装Docker，请参见Install Docker。

背景信息

您可以在Kubernetes**集群列表**页面查看您集群的Kubernetes边缘版版本，以及当前是否有新的版本可供升级。

功能原理

下面主要为您介绍集群升级过程中的相关功能及实现原理。



- 集群升级策略

  集群升级策略定义了您将使用怎样的策略对集群进行升级。目前默认策略为分批升级。分批升级会在**升级Node**阶段对集群内的节点进行分批升级。其具体策略为：

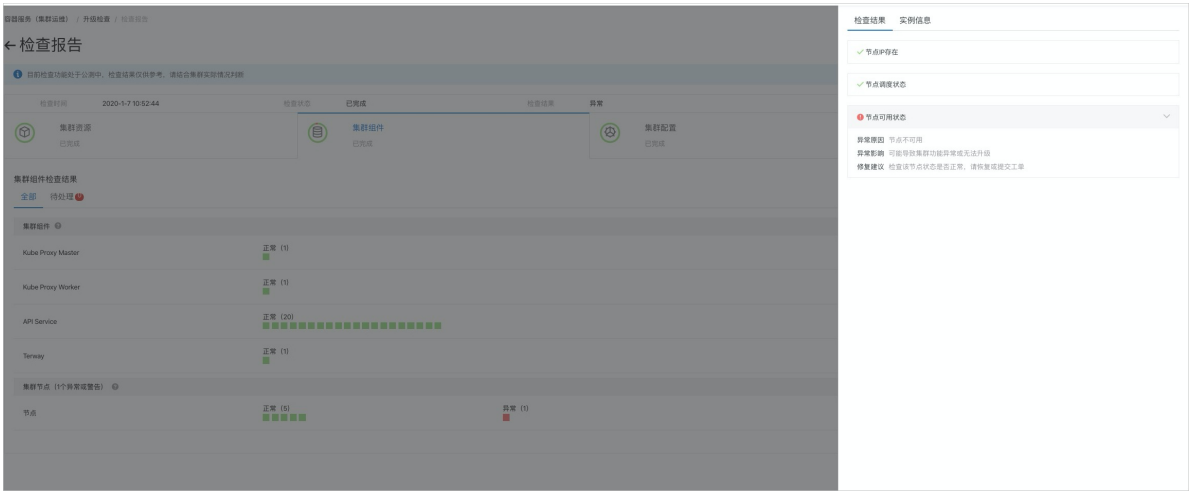
  - 第一批升级的节点数为1，后续的批次以2的幂数进行增长。暂停后重新恢复升级的第一批次为1，后续也是以2的幂数进行增长。
  - 每一批节点的最大数量不会超过节点总数的10%。

- 集群升级前置检查

  在您开始集群升级之后，我们会为您自动启动集群升级前置检查。该检查会对集群进行多项健康检查，以确保您的集群可以顺利的完成此次升级。

  如果您的集群存在不合理配置或者潜在风险，则无法通过前置检查，如下图所示。

单击**查看详情**按钮。系统跳转到集群运维页面，查看具体的失败原因。



- 集群升级暂停

  通过集群升级暂停功能，您可以在集群升级的任意阶段对其升级进程进行暂停。

  > **说明**
  > - 暂停升级之后，当前批次已经开始升级的节点会完成升级。还未开始升级的节点不会升级。
  > - 集群暂停状态为集群升级的中间状态，建议您不要在此时对集群进行操作，并尽快完成升级过程。

  您可以在集群成功暂停之后，单击**继续**，恢复集群的升级进程。

  如果集群升级过程中发生错误，集群升级进程会被系统所暂停。具体失败原因会展示在页面下方详情中。

- 集群升级取消

  您可以在暂停升级后，单击**取消**，对本次升级进行取消操作。

  > **说明**
  > - 取消升级之后，当前批次已经开始升级的节点会完成升级。还未开始升级的节点不会升级。
  > - 已经完成升级的节点不受影响。

注意事项

- 集群升级需要机器可以公网访问，以便下载升级所需的软件包。
- 集群升级Kubernetes边缘版过程中，集群上的应用不会中断。如果应用强依赖于API Server可能会有短暂影响。
- 如果您对Kubernetes集群有过任何的配置更改（例如，打开了swap分区），则升级过程有可能失败。
- 集群升级过程中您可以在一批节点升级完成后中断进程，此时集群处于升级的中间状态，我们建议您不要对集群进行操作，并尽快完成升级过程。处于中间状态的集群会在15日之后关闭升级过程，同时清理一切升级相关的事件和日志信息。
- 集群升级过程中，如非发生错误，请勿修改kube-upgrade命名空间下面的相关资源。
- 如集群升级失败，升级过程会暂停，您需要分析失败原因并清理kube-upgrade命名空间下失败的Pod，确认修复成功后重启升级过程。如需帮助，请联系在线客服。

准备工作

> **说明** 如果您在非生产环境中有待升级的集群，我们强烈建议您先对该集群进行升级验证，再在生产环境中启动集群升级。

请在集群升级前检查集群的健康状况，确保集群已具备升级条件。

1. 登录容器服务管理控制台。
2. 预升级节点。
   i. 在控制台左侧导航栏中，单击**集群**。
   ii. 在**集群列表**页面，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
   iii. 在**集群信息**页面单击**连接信息**页签，然后单击**公网访问**页签，复制集群的**KubeConfig**内容到计算机*$HOME/.kube/config*中。
   iv. 在本地手动执行以下命令。

   ```
   docker run -it -v ~/.kube:/root/.kube registry.cn-hangzhou.aliyuncs.com/edge-kubernetes/node-preprocess:v0.1.0云端节点列表
   ```

   > **说明** 命令需要输入云端节点的列表，没有云端节点的集群不需要该参数。
3. 在控制台左侧导航栏中，单击**集群**。
4. 在目标集群右侧**操作**列，单击**更多 > 集群检查**。
5. 在**容器服务运维中心**左侧导航栏单击**评测 > 升级检查**。
6. 在**升级检查**页面单击**执行升级检查**。
7. 在弹出**升级检查**面板中，勾选**注意事项**后，单击**执行检查**。
   检查完成后，单击**查看详情**。

   当检查报告中**检查结果**为**正常**时，表示升级检查成功，您可以进行集群升级操作。

如果检查结果异常可以自行修复，也可以通过提交工单，请阿里云工程师协助修复。

操作步骤

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面单击目标集群**操作**列下的**更多 > 集群升级**。
4. 弹出升级提示页面，单击**确定**。

   此时，您可以可视化的看到升级的全过程。

升级完成后，您可以在Kubernetes集群列表页面查看集群Kubernetes边缘版的版本，确认升级成功。

相关文档

- ACK@Edge概述
- 创建边缘托管版集群

4.3. 升级边缘集群组件

本文介绍如何升级边缘集群组件来解决很多时候集群已经是最新版本，但某些组件需要进行更小粒度的版本操作的场景。

前提条件

- 您已成功创建一个Kubernetes边缘版集群，请参见创建边缘托管版集群。
- 您已在本地安装Docker，请参见Install Docker。

操作步骤

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面选择目标集群，在该集群右侧**操作**列，选择**更多 > 系统组件管理**。
4. 在**组件管理**页面，您可以进行以下操作：
   - 选择需要安装的组件，单击**安装**。
   - 选择需要卸载的组件，单击**卸载**。
   - 选择需要升级的组件，单击**升级**。
   - 选择需要修改管控面参数的组件，单击**配置**。

     说明 仅Pro版集群支持自定义托管组件Kube API Server和Kube Controller Manager（KCM）的参数。更多信息，请参见自定义ACK Pro集群的管控面参数。

如果您的集群版本为1.12.6-aliyunedge.1时，**edge-tunnel-server**和**edge-tunnel-agent**组件需按如下操作步骤升级。

1. 手动删除**frps/frpc**组件相关的Ds、Deployment、Service等，删除步骤如下：
   i. 登录容器服务管理控制台。
   ii. 在控制台左侧导航栏中，单击**集群**。
   iii. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
   iv. 在**集群信息**页面选择**连接信息**页签，然后选择**公网访问**页签，复制集群的**kubeconfig**内容到计算机*$HOME/.kube/config*中。
   v. 执行以下命令。

   ```
   docker run -v ~/.kube:/root/.kube registry.cn-hangzhou.aliyuncs.com/acs/edge-upgrade-addon:v1.0 tunnel
   ```

2. 在集群管理页左侧导航栏中，选择**运维管理 > 组件管理**。
   - 单击组件**edge-tunnel-server**右侧**操作**列下的**升级**。
   - 单击组件**edge-tunnel-agent**右侧**操作**列下的**升级**。

相关文档

- 升级边缘集群

4.4. 扩容边缘集群

通过容器服务管理控制台，您可以对Kubernetes边缘集群的边缘节点进行扩容。您无需通过购买边缘节点，再通过添加已有节点的方式进行集群扩容，可直接在扩容界面购买并自动扩容。本文介绍如何基于阿里云边缘节点服务ENS（Edge Node Service）扩容边缘集群及如何通过增加ECS节点方式扩容边缘集群的管控节点。

前提条件

- 创建Kubernetes边缘托管版集群。
- 您已开通ENS，请参见开通ENS服务。

操作步骤

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面，选择目标边缘托管版集群，在该集群右侧**操作**列下单击**集群扩容**。
4. 在**集群扩容**页面，扩容边缘集群。

   对边缘集群进行扩容，ACK提供**边缘节点扩容**和**云端管控扩容**两种方式：

   - **边缘节点扩容**

     您可以通过增加边缘节点的方式扩容边缘集群。单击**边缘节点扩容**页签，设置边缘节点的配置项。

配置项	描述
集群名称	默认显示您创建的边缘托管版集群名称。
节点	设置扩容节点的地域。
实例规格	设置节点的实例规格。支持选择多个实例规格。
已有Worker数	显示已经拥有的Worker节点数量。
伸缩数量	需要扩容的Worker节点数量。
系统盘	设置扩容节点的系统盘，最小20GiB。默认为普通云盘。
数据盘	设置是否挂载数据盘。
镜像挂载	默认创建一个操作系统为centos_7_04_64_20G_alibase_20171211的镜像。
公网带宽计费	阿里云边缘节点服务ENS带宽计费方式默认按量后付。如果您的月带宽用量超过10Gbps，您也可以联系商务经理获取更灵活优惠的月95带宽计费方式。带宽计费详情请参见带宽计费。
登录密码	设置节点的登录密码。
确认密码	确认设置节点登录密码。
付费类型	只支持包年包月付费类型。
购买时长	选择包年包月的购买时长，支持选择1、2、3、6或12个月。

配置项	描述
自动续费	设置是否自动续费。

- **云端管控扩容**

  您还可以通过增加ECS节点的方式扩容边缘集群的管控节点。单击**云端管控扩容**页签，设置管控节点的配置项。有关**云端管控扩容**的配置项说明，请参见集群扩容配置项。

5. 在**集群扩容**页面右侧，单击**提交**。
6. 在**当前配置确认**页面，阅读并选中服务协议，单击**确定**。

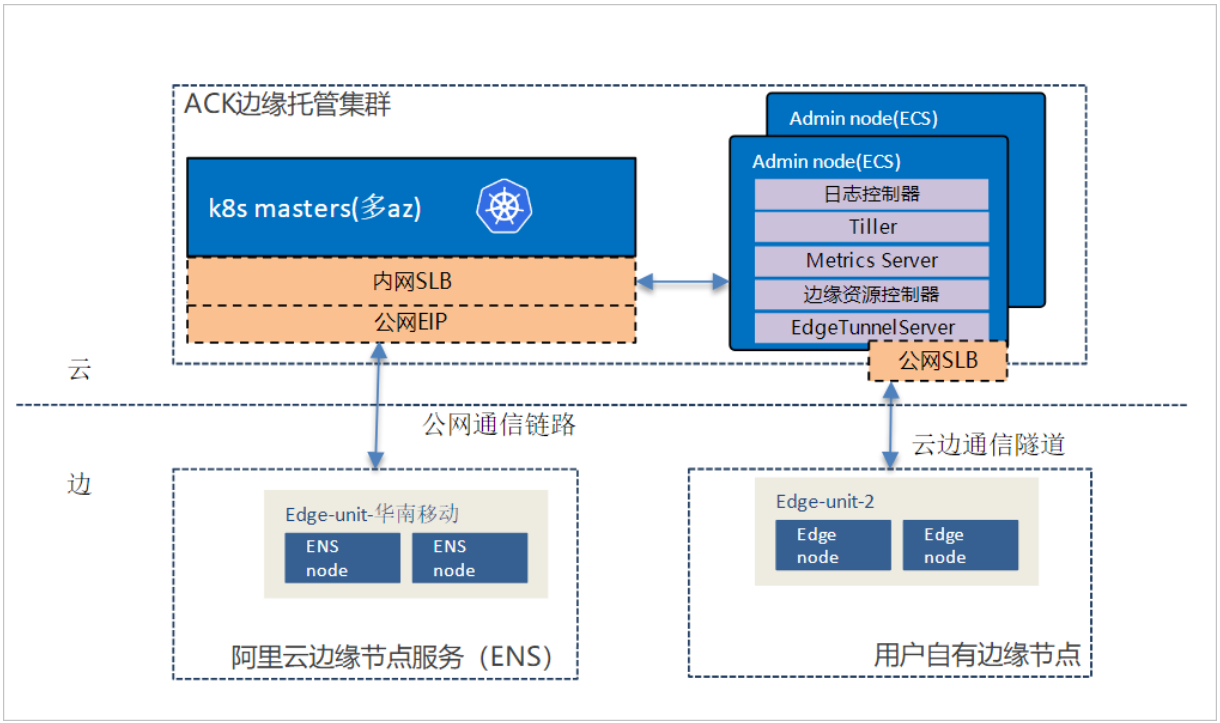
执行结果

返回**集群列表**页面，目标集群的**集群状态**列显示**伸缩中**字样，说明扩容正在执行中。当**集群状态**显示为**运行中**时，说明成功扩容边缘集群。

4.5. 边缘集群云端ECS节点说明

本文主要介绍边缘托管集群中存在的至少一个云服务器ECS（Elastic Compute Service）节点的作用和增、删操作。

边缘计算云端管控节点



在边缘托管集群创建过程中，平台会默认为您创建至少一个ECS实例，并接入到集群管控。该实例主要用来部署云端管控应用，也支持您自定义的云端管控应用部署。通过默认自带**node-role.alibabacloud.com/addon: Effect: NoSchedule**污点（Taints），来保证边缘业务不会部署在云端管控节点。截止1.14.8-aliyunedge.1版本，中心管控节点上默认安装的管控应用有：

- alibaba-log-controller：日志服务LOG（Log Service，原SLS）控制器。
- alicloud-monitor-controller：ECS云监控服务控制器。
- metric-server：集群监控服务端。

- edge-tunnel-server：反向运维通道服务端，支持通过原生Kubernetes API获取边缘节点、容器监控、SSH远程执行命令等操作。

将应用部署到云端管控节点

如果您需要在云端中心管控节点上部署自定义的管控应用，例如各种类型的operator。您需要配置容忍上述提到的污点（Taints）和对应的节点选择器（NodeSelector），配置片段如下。

```
...
 nodeSelector:
  alibabacloud.com/is-edge-worker: 'false'
  beta.kubernetes.io/arch: amd64
  beta.kubernetes.io/os: linux
 tolerations:
  - effect: NoSchedule
    key: node-role.alibabacloud.com/addon
    operator: Exists
...
```

新增云端管控节点

如果您需要在集群中新增云端管控节点，可以按照以下步骤操作（后续支持基于ECS的自动扩缩容能力）。

1. 在集群所在VPC购买ECS。

   购买ECS的步骤，请参见创建ECS实例。

   > **说明** 操作系统选择CentOS 7.6。

2. 在OpenAPI开发者门户上使用AttachInstances API，添加ECS至集群。

   body参数如下：

   ```
   {
    "password": "Helloxxx!",
    "tags":[],
    "instances": [
     "i-uf65mbpn1x8xxxxxx"
    ]
   }
   ```

参数	说明
password	ECS实例密码。密码规则为8~30个字符，且至少同时包含三项（大小写字母、数字和特殊符号）。
tags	给节点打tag标签： ○ key：标签名称。 ○ value：标签值。
instances	已有实例的数组。

   更多API信息，请参见添加已有实例至节点池。

相关文档

- ACK@Edge概述
- 创建边缘托管版集群

5.ACK@Edge Pro版集群

5.1. ACK@Edge Pro版集群介绍

ACK@Edge Pro版集群针对企业大规模生产环境，在ACK@Edge标准版基础上进一步增强了可靠性、安全性，并且提供可赔付的SLA的Edge Pro Kubernetes集群。

ACK@Edge Pro版集群是在原ACK@Edge标准版集群的基础上发展而来的集群类型，继承了原边缘标准版集群的所有优势，例如Master节点托管、Master节点高可用等。同时，相比原边缘标准版进一步增强了集群的可靠性、安全性和调度性，并且支持赔付标准的SLA，适合生产环境下有着大规模业务，对稳定性和安全性有高要求的企业客户。

使用场景

- 互联网企业，大规模业务上线生产环境，对管控的稳定性、可观测性和安全性有较高要求。
- 大数据计算企业，大规模数据计算、高性能数据处理、高弹性需求等类型业务，对集群稳定性、性能和效率有较高要求。
- 开展中国业务的海外企业，对有赔付标准的SLA以及安全隐私等非常重视。
- 金融企业，需要提供赔付标准的SLA。

功能特点

- 更可靠的托管Master节点：ETCD容灾和备份恢复，冷热备机制最大程度保障集群数据库的可用性；管控组件的关键指标可观测，助力您更好地预知风险。
- 更安全的容器集群：管控面ETCD默认采用加密盘存储；数据面通过安装kms-plugin组件实现Secrets数据落盘加密。开放安全管理，并提供针对运行中容器更强检测和自动修复能力的安全管理高级版。
- 更智能的容器调度：集成更强调度性能的kube-scheduler，支持本地存储LVM调度，支持多种智能调度算法，支持NPU调度，优化在大规模数据计算、高性能数据处理等业务场景下的容器调度能力。
- SLA保障：提供赔付标准的SLA保障，集群API Server的可用性达到99.95%。

定价

ACK@Edge Pro版集群的计费详情，请参见ACK@Edge计费说明。

对比

ACK@Edge Pro版集群和ACK@Edge标准版集群的对比详情如下表。

分类	功能	ACK@Edge	
		ACK@Edge Pro版集群	ACK@Edge标准版集群
集群规模	不涉及	最大1000节点	最大100节点（现有集群不受影响，可以升级到Pro版）
SLA	不涉及	99.95%（支持赔付）	99.9%（不支持赔付）
	自定义参数设置机制	✓	✗

API Server分类	功能	ACK@Edge	
		ACK@Edge Pro版集群	ACK@Edge标准版集群
	可用性监控	✓	✗
ETCD	高频冷热备机制，异地容灾	✓	✗
	可观测性监控指标	✓	✗
安全管理	开放高级版（支持数据加密，请参见使用阿里云KMS进行Secret的落盘加密）	✓	✗

开服地域

- 亚太

地域名称	所在城市	Region ID
华北2	北京	cn-beijing
华北3	张家口	cn-zhangjiakou
华北5	呼和浩特	cn-huhehaote
华北6	乌兰察布	cn-wulanchabu
华东1	杭州	cn-hangzhou
华东2	上海	cn-shanghai
华南1	深圳	cn-shenzhen
华南2	河源	cn-heyuan
西南1	成都	cn-chengdu
中国（香港）	中国香港	cn-hongkong
日本	东京	ap-northeast-1
新加坡	新加坡	ap-southeast-1
澳大利亚	悉尼	ap-southeast-2
马来西亚	吉隆坡	ap-southeast-3

地域名称	所在城市	Region ID
印度尼西亚	雅加达	ap-southeast-5

- 美洲与欧洲

地域名称	所在城市	Region ID
美国西部	硅谷	us-west-1
美国东部	弗吉尼亚	us-east-1
英国	伦敦	eu-west-1
德国	法兰克福	eu-central-1

- 印度

地域名称	所在城市	Region ID
印度	孟买	ap-south-1

5.2. 创建ACK@Edge Pro版集群

ACK@Edge Pro版集群相比原边缘标准版集群进一步提升了集群的可靠性、安全性和调度性能，并且支持赔付标准的SLA，适合生产环境下有着大规模业务，对稳定性和安全性有高要求的企业客户。本文介绍如何通过容器服务控制台创建ACK@Edge Pro版集群。

前提条件

登录RAM管理控制台和弹性伸缩控制台开通相应的服务。

说明

您在使用集群过程中，请注意以下限制：

- 用户账户需有100元的余额并通过实名认证，否则无法创建按量付费的ECS实例和负载均衡。
- ACK集群仅支持专有网络VPC。
- 每个账号默认可以创建的云资源有一定的配额，如果超过配额创建集群会失败。请在创建集群前确认您的配额。
  - 关于ACK集群配额限制的详情，请参见ACK集群配额限制。

    注意 每个账户初始默认状况下VPC路由条目不超过48条，意味着您的Kubernetes集群的网络模式是Flannel时，集群的节点数最大不能超过48个（网络模式是Terway则不受该影响）。如集群需要更多节点数，您需要先对目标VPC提交工单，申请提高配额。
  - 每个账号默认最多可以创建100个安全组。
  - 每个账号默认最多可以创建60个按量付费的负载均衡实例。
  - 每个账号默认最多可以创建20个EIP。

操作步骤

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面中，单击页面右上角的**创建集群**。
4. 在**ACK边缘托管版**页签，完成ACK@Edge Pro版集群配置。
   i. 完成集群基础选项配置。

配置项	描述
账号全部资源组	将鼠标悬浮于页面上方的**账号全部资源**，选择资源组。在控制台页面顶部选择的资源组可过滤出该资源组内的专有网络及对应的虚拟交换机。在创建集群时，只显示过滤的专有网络实例及专有网络对应的虚拟交换机实例。 
集群名称	填写集群的名称。 **说明** 集群名称应包含1~63个字符，可包含数字、汉字、英文字符或短划线（-）。
集群规格	选择集群规格，支持**标准版**和**Pro版**。 选中**Pro版**创建ACK@Edge Pro版集群。

配置项	描述
地域	选择集群所在的地域。
Kubernetes版本	显示当前ACK@Edge Pro支持的Kubernetes版本。
专有网络	设置集群的网络，您可以选择普通VPC和共享VPC。 ■ 共享VPC：VPC的所有者账号（资源所有者）可以将其账号下的VPC内的交换机资源共享给其组织内的其他账号使用。 ■ 普通VPC：不具备共享功能的VPC。 **说明** Kubernetes集群仅支持专有网络。您可以在已有VPC列表中选择所需的VPC。如果没有您需要的专有网络，可以通过单击**创建专有网络**进行创建，请参见创建和管理专有网络。
虚拟交换机	设置虚拟交换机。 您可以在已有虚拟交换机列表中，根据**可用区**选择1~3个交换机。如果没有您需要的交换机，可以通过单击**创建虚拟交换机**进行创建，请参见使用交换机。
节点IP数量	如果您选择的网络模式为**Flannel**，您需设置**节点IP数量**。 **说明** ■ **节点IP数量**是指可分配给一个节点的IP数量，建议保持默认值。 ■ 根据您所选择的专有网络和节点IP数量，ACK将为您推荐可用的Pod网络CIDR和Service CIDR，并给出相应配置下集群内可允许部署的主机数量以及每台主机可容纳的Pod数量。请您根据集群规模的实际需求，在推荐配置的基础上进行修改。
Pod网络CIDR	网络插件选择**Flannel**时，需要配置**Pod网络CIDR**。 **Pod网络CIDR**指定Flannel网络插件需要配置Pod网络CIDR，网段不能和VPC及VPC已有Kubernetes集群使用的网段重复，创建成功后不能修改。而且Service地址段不能和Pod地址段重复，有关Kubernetes网络地址段规划的信息，请参见Kubernetes集群网络规划。
Service CIDR	设置**Service CIDR**。您需要指定**Service CIDR**，网段不能与VPC及VPC内已有Kubernetes集群使用的网段重复，创建成功后不能修改。而且Service地址段也不能和Pod地址段重复，有关Kubernetes网络地址段规划的信息，请参见Kubernetes集群网络规划。
配置SNAT	创建集群时，默认不开通公网。如果您选择的VPC不具备公网访问能力，选中**为专有网络配置SNAT**后，ACK将为您创建NAT网关并自动配置SNAT规则。

配置项	描述
API Server访问	ACK默认为API Server创建一个内网SLB实例，您可修改SLB实例规格。更多信息，请参见实例规格。 **注意** 删除默认创建的SLB实例将会导致无法访问API Server。 您可设置是否开放**使用EIP暴露API Server**。API Server提供了各类资源对象（Pod，Service等）的增删改查及Watch等HTTP Rest接口。 ■ 如果选择开放，ACK会创建一个EIP，并挂载到SLB上。此时，Master节点的6443端口（对应API Server）暴露出来，您可以在外网通过kubeconfig连接并操作集群。 ■ 如果选择不开放，则不会创建EIP，您只能在VPC内部用kubeconfig连接并操作集群。 **说明** 通常边缘节点需要通过公网和云端API Server交互，因此若不选中**使用EIP暴露API Server**，边缘节点将无法连接到云端集群，所创建集群也将无法在边缘场景下使用。
RDS白名单	设置RDS白名单。将节点IP添加到RDS实例的白名单中。 **说明** 允许白名单RDS访问Kubernetes集群，RDS必须在当前集群的VPC内。
安全组	支持选择**自动创建普通安全组**、**自动创建企业级安全组**、**选择已有安全组**。有关安全组的详细内容，请参见安全组概述。 **说明** ■ 只有白名单用户可以使用**选择已有安全组**功能。请提交工单申请。 ■ 指定已有安全组时，系统默认不会为安全组配置额外的访问规则，可能会导致访问异常，请自行管理安全组规则。关于如何管理安全组规则，请参见最小化集群访问规则。
集群删除保护	设置是否启用集群删除保护。为防止通过控制台或API误释放集群。
资源组	创建的集群将归属于选择的资源组。一个资源只能归属于一个资源组。根据不同的业务场景，您可以将资源组映射为项目、应用或组织等概念。更多信息，请参见资源组。

ii. 完成集群高级选项配置。

配置项	描述
kube-proxy代理模式	支持iptables和IPVS两种模式。 ■ iptables：成熟稳定的kube-proxy代理模式，Kubernetes Service的服务发现和负载均衡使用iptables规则配置，但性能一般，受规模影响较大，适用于集群存在少量的Service。 ■ IPVS：高性能的kube-proxy代理模式，Kubernetes Service的服务发现和负载均衡使用Linux ipvs模块进行配置，适用于集群存在大量的service，对负载均衡有高性能要求的场景。
标签	为集群绑定标签。输入键和对应的值，单击**添加**。 **说明** ■ 键是必需的，而*值*是可选的，可以不填写。 ■ 键不能是aliyun、http://、https://开头的字符串，不区分大小写，最多64个字符。 ■ *值*不能是http:// 或https://，可以为空，不区分大小写，最多128个字符。 ■ 同一个资源，标签键不能重复，相同标签键（Key）的标签会被覆盖。 ■ 如果一个资源已经绑定了20个标签，已有标签和新建标签会失效，您需要解绑部分标签后才能再绑定新的标签。
Secret落盘加密	在ACK@Edge Pro版集群中，选中**选择KMS密钥**可以使用在阿里云密钥管理服务KMS（Key Management Service）中创建的密钥加密Kubernetes Secret密钥。设置Secret落盘加密的详情，请参见使用阿里云KMS进行Secret的落盘加密。

5. 单击**下一步：Worker配置**，完成Worker节点配置。

说明 创建ACK@Edge Pro版集群，至少需要配置1个Worker节点，用于部署云端的管控组件。

配置项	描述
实例规格	支持选择多个实例规格。详情请参见实例规格族。 **说明** ACK@Edge Pro版集群的日志、监控、反向通道等一些增强特性需要在云端部署组件，因此默认创建至少一个ECS实例作为Worker节点。
已选规格	呈现选中的规格。

配置项	描述
数量	新增Worker实例（ECS实例）的数量。
系统盘	支持**ESSD云盘**、**SSD云盘**和**高效云盘**。 **说明** ○ 支持选中**开启云盘备份**以备份云盘数据。 ○ **ESSD云盘**支持自定义**性能级别**。 ESSD云盘容量越大，可供选择的性能级别越高（460 GiB容量以上可选PL2，1260 GiB以上可选PL3）。更多信息，请参见容量范围与性能级别的关系。
挂载数据盘	支持**ESSD云盘**、**SSD云盘**和**高效云盘**。挂载数据盘时，支持云盘**加密**和**开启云盘备份**。节点的数据盘加密时只能使用默认的CMK。
登录方式	○ 设置密钥。 ■ **密钥对**：如您已经创建密钥对，在下拉列表中选择目标密钥对。 ■ **新建密钥对**：此项用于您还未创建密钥对。创建密钥对，请参见创建SSH密钥对。密钥对创建完毕后，设置该密钥对作为登录集群的凭据。 ○ 设置密码。 ■ **登录密码**：设置节点的登录密码。 ■ **确认密码**：确认设置的节点登录密码。 **说明** 密码为8~30个字符，且必须同时包含三项（大写字母、小写字母、数字和特殊符号），其中特殊字符不包括下划线（_）。 **说明** 当您勾选**云监控插件**或**日志服务**时，需要给ECS设置登录方式。

6. 单击**下一步：组件配置**，完成组件配置。

配置项	描述
云监控插件	设置是否启用云监控插件。您可以选中**在ECS节点上安装云监控插件**，从而在云监控控制台查看所创建ECS实例的监控信息。

配置项	描述
日志服务	设置是否启用日志服务，您可使用已有Project或新建一个Project。默认选中**使用日志服务**。创建应用时，您可通过简单配置，快速使用日志服务，详情参见通过日志服务采集Kubernetes容器日志。

7. 单击**下一步：确认配置**。
8. 阅读并选中**服务协议**，然后单击**创建集群**。

> **说明** 一个包含多节点的ACK@Edge Pro版集群的创建时间一般约为十分钟。

执行结果

- 集群创建成功后，您可以在容器服务管理控制台的**集群列表**页面查看所创建的集群。
- 您可以单击目标集群右侧**操作**列下的**查看日志**，在**集群日志信息**页面查看集群的日志信息。 您也可以在集群日志信息页面中，单击**资源栈事件**查看更详细的信息。
- 单击目标集群右侧**操作**列下的**详情**，然后单击**基本信息**和**连接信息**页签，查看集群的基本信息和连接信息。其中：
  - **API Server公网连接端点**：Kubernetes的API Server对公网提供服务的地址和端口，可以通过此服务在用户终端使用kubectl等工具管理集群。
  - **API Service内网连接端点**：Kubernetes的API server对集群内部提供服务的地址和端口，此IP为负载均衡的地址。
  - **测试域名**：为集群中的服务提供测试用的访问域名。服务访问域名后缀是*<cluster_id>.<region_id>.alicontainer.com*。

> **说明** 在**基本信息**页签的**测试域名**右侧，单击**重新绑定域名**，您可以重新绑定访问域名。

5.3. 资源调度

5.3.1. Gang scheduling

ACK基于新版的Kube-scheduler框架实现Gang scheduling的能力，解决原生调度器对于All-or-Nothing作业调度的问题。本文举例说明如何使用Gang scheduling能力。

前提条件

- 您已创建ACK Pro版集群。具体步骤，请参见创建ACK Pro版集群。

> **注意** 目前Gang scheduling仅支持ACK Pro托管版集群。如果您需要专有版集群，请提交工单申请白名单。

- 系统组件版本要求具体如下表所示。

组件	版本要求
Kubernetes	1.16及以上

组件	版本要求
Helm版本	3.0及以上版本
Docker版本	19.03.5
操作系统	CentOS 7.6、CentOS 7.7、Ubuntu 16.04、Ubuntu 18.04、Alibaba Cloud Linux 2

背景信息

Gang scheduling是在并发系统中将多个相关联的进程调度到不同处理器上同时运行的策略，其最主要的原则是保证所有相关联的进程能够同时启动，防止部分进程的异常，导致整个关联进程组的阻塞。例如，您提交一个批量Job，这个批量Job包含多个任务，要么这多个任务全部调度成功，要么一个都调度不成功。这种All-or-Nothing调度场景，就被称作Gang scheduling。

Kubernetes目前已经广泛的应用于在线服务编排，为了提升集群的利用率和运行效率，ACK希望将Kubernetes作为一个统一的管理平台来管理在线服务和离线作业。但是由于调度器的限制，使得一些离线的工作负载无法迁移到Kubernetes。例如对于有All-or-Nothing特点的作业，它要求所有的任务在同一时间被调度，如果只是部分任务启动的话，启动的任务将持续等待剩余的任务被调度。在最坏的情况下，所有作业都处于挂起状态，从而导致死锁。为了解决这个问题，需要对调度器支持Gang scheduling。

功能介绍

ACK将一组需要同时调度的Pod称为PodGroup。您如果在提交All-or-Nothing作业时，可以通过设置labels字段的形式指定所属PodGroup的名称以及保证作业正常运行Task的最少运行个数。调度器会根据您指定的最少运行个数进行调度，只有当集群资源满足该Task最少运行个数时，才会统一调度，否则作业将一直处于Pending状态。

使用方式

使用Gang scheduling时，在创建的Pod处通过设置labels的形式配置min-available和name。

```
labels:
  pod-group.scheduling.sigs.k8s.io/name: tf-smoke-gpu
  pod-group.scheduling.sigs.k8s.io/min-available: "3"
```

- name：用于表示PodGroup的名称。
- min-available：用于表示当前集群资源至少满足最少可运行Pod数时，才能统一调度创建。

> **说明** 要求属于同一个PodGroup的Pod必须保持相同的优先级。

示例

文本通过运行Tensorflow的分布式作业来展示Gang scheduling的效果。当前测试集群有4个GPU卡。

1. 执行以下命令通过Kubeflow的Arena工具在已有Kubernetes集群中部署Tensorflow作业运行环境。

   > **说明** Arena是基于Kubernetes的机器学习系统开源社区Kubeflow中的子项目之一。Arena用命令行和SDK的形式支持了机器学习任务的主要生命周期管理（包括环境安装、数据准备，到模型开发、模型训练、模型预测等），有效提升了数据科学家工作效率。

```
git clone https://github.com/kubeflow/arena.git
kubectl create ns arena-system
kubectl create -f arena/kubernetes-artifacts/jobmon/jobmon-role.yaml
kubectl create -f arena/kubernetes-artifacts/tf-operator/tf-crd.yaml
kubectl create -f arena/kubernetes-artifacts/tf-operator/tf-operator.yaml
```

执行以下命令检查Tensorflow作业运行环境是否成功部署。当出现Running状态时，说明成功部署。

```
kubectl  get pods -n arena-system
```

```
NAME                READY  STATUS  RESTARTS  AGE
tf-job-dashboard-56cf48874f-gwlhv  1/1   Running  0     54s
tf-job-operator-66494d88fd-snm9m  1/1   Running  0     54s
```

2. 使用以下模板向集群中提交Tensorflow分布式作业，含有1个PS和4个Worker，每个Worker类型的Pod需要2个GPU。

```
apiVersion: "kubeflow.org/v1"
kind: "TFJob"
metadata:
 name: "tf-smoke-gpu"
spec:
 tfReplicaSpecs:
  PS:
   replicas: 1
   template:
    metadata:
     creationTimestamp: null
     labels:
      pod-group.scheduling.sigs.k8s.io/name: tf-smoke-gpu
      pod-group.scheduling.sigs.k8s.io/min-available: "5"
    spec:
     containers:
     - args:
      - python
      - tf_cnn_benchmarks.py
      - --batch_size=32
      - --model=resnet50
      - --variable_update=parameter_server
      - --flush_stdout=true
      - --num_gpus=1
      - --local_parameter_device=cpu
      - --device=cpu
      - --data_format=NHWC
      image: registry.cn-hangzhou.aliyuncs.com/kubeflow-images-public/tf-benchmarks-cpu:v20171202-bdab599-dirty-284af3
      name: tensorflow
      ports:
      - containerPort: 2222
       name: tfjob-port
      resources:
       limits:
        cpu: '1'
      workingDir: /opt/tf-benchmarks/scripts/tf_cnn_benchmarks
     restartPolicy: OnFailure
```

```
    restartPolicy: OnFailure
  Worker:
   replicas: 4
   template:
    metadata:
     creationTimestamp: null
     labels:
      pod-group.scheduling.sigs.k8s.io/name: tf-smoke-gpu
      pod-group.scheduling.sigs.k8s.io/min-available: "5"
    spec:
     containers:
     - args:
      - python
      - tf_cnn_benchmarks.py
      - --batch_size=32
      - --model=resnet50
      - --variable_update=parameter_server
      - --flush_stdout=true
      - --num_gpus=1
      - --local_parameter_device=cpu
      - --device=gpu
      - --data_format=NHWC
      image: registry.cn-hangzhou.aliyuncs.com/kubeflow-images-public/tf-benchmarks-gpu:v20171202-bdab599-dirty-284af3
      name: tensorflow
      ports:
      - containerPort: 2222
       name: tfjob-port
      resources:
       limits:
        nvidia.com/gpu: 2
      workingDir: /opt/tf-benchmarks/scripts/tf_cnn_benchmarks
     restartPolicy: OnFailure
```

- 不使用Gang scheduling功能

  执行以下命令查看Pod状态。集群中只用2个Worker类型的Pod在运行，剩余2个处于Pending状态。

  ```
  kubectl get pods
  ```

  ```
  NAME          READY  STATUS   RESTARTS  AGE
  tf-smoke-gpu-ps-0    1/1   Running  0     6m43s
  tf-smoke-gpu-worker-0  1/1   Running  0     6m43s
  tf-smoke-gpu-worker-1  1/1   Running  0     6m43s
  tf-smoke-gpu-worker-2  0/1   Pending  0     6m43s
  tf-smoke-gpu-worker-3  0/1   Pending  0     6m43s
  ```

  执行以下命令查看其中正在运行的Worker类型Pod的日志。处于等待剩余两个Worker类型Pod启动的状态，GPU占用却没有使用。

  ```
  kubectl logs -f tf-smoke-gpu-worker-0
  ```

```
INFO|2020-05-19T07:02:18|/opt/launcher.py|27| 2020-05-19 07:02:18.199696: I tensorflow/core/distributed_runtime/master.cc:221] CreateSession still waiting for response from worker: /job:worker/replica:0/task:3
INFO|2020-05-19T07:02:28|/opt/launcher.py|27| 2020-05-19 07:02:28.199798: I tensorflow/core/distributed_runtime/master.cc:221] CreateSession still waiting for response from worker: /job:worker/replica:0/task:2
```

- 使用Gang scheduling功能时

  执行以下命令查看Pod状态。因为集群的资源无法满足设定的最小数要求，则PodGroup无法正常调度，所有的Pod一直处于Pending状态。

  ```
  kubectl get pods
  ```

  ```
  NAME                    READY  STATUS   RESTARTS  AGE
  tf-smoke-gpu-ps-0       0/1    Pending  0         43s
  tf-smoke-gpu-worker-0   0/1    Pending  0         43s
  tf-smoke-gpu-worker-1   0/1    Pending  0         43s
  tf-smoke-gpu-worker-2   0/1    Pending  0         43s
  tf-smoke-gpu-worker-3   0/1    Pending  0         43s
  ```

  当集群中新增4个GPU卡的资源，则当前集群的资源满足设定的最小数要求，则PodGroup正常调度，4个Worker类型Pod开始运行。执行以下命令查看Pod状态。

  ```
  kubectl get pods
  ```

  ```
  NAME                    READY  STATUS   RESTARTS  AGE
  tf-smoke-gpu-ps-0       1/1    Running  0         3m16s
  tf-smoke-gpu-worker-0   1/1    Running  0         3m16s
  tf-smoke-gpu-worker-1   1/1    Running  0         3m16s
  tf-smoke-gpu-worker-2   1/1    Running  0         3m16s
  tf-smoke-gpu-worker-3   1/1    Running  0         3m16s
  ```

  执行以下命令查看其中一个Worker类型Pod的日志，显示训练任务已经开始。

  ```
  kubectl logs -f tf-smoke-gpu-worker-0
  ```

  ```
  INFO|2020-05-19T07:15:24|/opt/launcher.py|27| Running warm up
  INFO|2020-05-19T07:21:04|/opt/launcher.py|27| Done warm up
  INFO|2020-05-19T07:21:04|/opt/launcher.py|27| Step  Img/sec loss
  INFO|2020-05-19T07:21:05|/opt/launcher.py|27| 1 images/sec: 31.6 +/- 0.0 (jitter = 0.0) 8.318
  INFO|2020-05-19T07:21:15|/opt/launcher.py|27| 10  images/sec: 31.1 +/- 0.4 (jitter = 0.7) 8.343
  INFO|2020-05-19T07:21:25|/opt/launcher.py|27| 20  images/sec: 31.5 +/- 0.3 (jitter = 0.7) 8.142
  ```

5.4. 使用阿里云KMS进行Secret的落盘加密

在ACK@Edge Pro版集群中，您可以使用在阿里云密钥管理服务KMS（Key Management Service）中创建的密钥加密Kubernetes Secret密钥。本文主要介绍如何使用阿里云密钥管理服务（KMS）中管理的密钥对ACK@Edge Pro版集群中的Kubernetes Secret密钥数据进行落盘加密。

前提条件

- 在KMS控制台已创建用户主密钥，详细介绍请参见管理密钥。

> **说明** 当前开启边缘Pro版集群的Secret落盘加密只支持使用Aliyun_AES_256类型的主密钥，同时不支持开启自动轮转周期。

- 主账号需要授权容器服务账号使用AliyunCSManagedSecurityRole系统角色的权限。如果您使用的账号未授权，在创建边缘Pro版集群或修改已有边缘Pro版集群过程中开启Secret落盘加密时，系统会提示您进行安全系统角色授权。
- 如果当前登录账号是子账号，请确保该子账号有AliyunKMSCryptoAdminAccess系统权限，授权流程请参考为RAM用户授权。

背景信息

在Kubernetes集群中，我们通常使用Secrets密钥模型存储和管理业务应用涉及的敏感信息，比如应用密码、TLS证书、Docker镜像下载凭据等敏感信息。Kubernetes会将所有的这些Secrets密钥对象数据存储在集群对应的etcd中。更多关于密钥的信息，请参见Secrets。

在ACK@Edge Pro版集群中，您可以使用在密钥管理服务（KMS）中创建的密钥加密Kubernetes Secret密钥，加密过程基于Kubernetes提供的KMS Encryption Provider机制，使用信封加密的方式对存储在etcd中的Kubernetes Secret密钥进行自动加密和解密，信封加密的详细介绍请参见什么是信封加密，以下介绍Kubernetes Secret密钥进行加密和解密的过程：

- 当一个业务密钥需要通过Kubernetes Secret API存储时，数据会首先被API Server生成的一个随机的数据加密密钥进行加密，然后该数据密钥会被指定的阿里云KMS密钥加密为一个密文密钥存储在etcd中。
- 解密Kubernetes Secret密钥时，系统会首先调用阿里云KMS服务的解密OpenAPI进行密文密钥的解密，然后使用解密后的明文密钥对Secret数据解密并最终返回给用户。

在新建的ACK@Edge Pro版集群中开启Secret落盘加密

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面中，单击页面右上角的**创建集群**。
4. 单击**ACK边缘托管版**页签。
5. 在**ACK边缘托管版**页签找到**Secret落盘加密**，选中**选择KMS密钥**，在下拉框中选择KMS密钥ID。关于创建ACK@Edge Pro版集群的其他配置信息，请参见创建ACK@Edge Pro版集群。

登录操作审计控制台，在左侧导航栏单击**事件查询**，在**事件查询**页面有使用aliyuncsmanagedsecurityrole系统角色的加密和解密事件日志，则说明该集群后台已成功开启Secret落盘加密特性。

在已创建的ACK@Edge Pro版集群中开启Secret落盘加密

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面单击目标边缘Pro版集群名称。
4. 在集群详情页面单击**基本信息**页签，在**基本信息**区域中打开**Secret落盘加密**开关。

> **说明** 如果当前登录用户为子账号，请确保该子账号对该集群有RBAC的管理员或运维人员权限，授权流程请参考配置RAM用户RBAC权限。

当集群状态由**更新中**变为**运行中**时，说明该集群Secret落盘加密的特性已变更完成。

在已创建的ACK@Edge Pro版集群中关闭Secret落盘加密

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面单击目标边缘Pro版集群名称。
4. 在集群详情页面单击**基本信息**页签，在**基本信息**区域中关闭**Secret落盘加密**开关。

> **说明** 如果当前登录用户为子账号，请确保该子账号对该集群有RBAC的管理员或运维人员权限，授权流程请参考配置RAM用户RBAC权限。

当集群状态由**更新中**变为**运行中**时，说明该集群Secret落盘加密的特性已变更完成。

5.5. 自定义ACK@Edge Pro版集群的管控面参数

为了满足生产环境调整管控面参数的需求，ACK@Edge Pro版集群为您提供了管控面参数自定义功能。您可以根据需要修改托管组件Kube API Server和Kube Controller Manager（KCM）的参数。本文介绍如何自定义ACK@Edge Pro版集群的管控面参数。

注意事项

为了确保顺利完成管控面的参数修改，请仔细阅读以下注意事项：

- 修改参数之后，管控面会进行重启，建议您选择业务低峰进行参数修改操作。
- 修改参数之后，您输入的参数会覆盖ACK@Edge Pro提供的默认参数，从而导致默认参数失效。
- 为了保证管控面的稳定性，当前ACK@Edge Pro仅支持自定义部分参数。
- 请确保您输入的参数正确且完整，错误参数可能会导致管控面启动失败。关于参数设置的详细信息，请参见kube-apiserver和kube-controller-manager。

自定义ACK@Edge Pro版集群的管控面参数

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
4. 在集群管理页左侧导航栏中，选择**运维管理 > 组件管理**。

   下文以修改Kube API Server组件为例，说明如何修改组件参数。
5. 在**核心组件**区域，单击目标组件右下方的图标。

6. 在**kube-apiserver参数配置**对话框中，输入您的自定义参数，然后单击**确定**。

> **说明** 请确保您输入的参数完整性和正确性。目前ACK@Edge Pro版集群只支持修改Kube API Server和KCM组件的参数。关于Kube API Server和KCM参数的具体格式和合法值，请参见kube-apiserver和kube-controller-manager（需选择对应的Kubernetes版本）。

默认参数列表

当您进入组件的参数配置操作之后，组件原有的默认参数会被覆盖。您可以参考以下表格将参数恢复为默认值。

Kubernetes版本	组件名	参数	默认值
1.16	kube-apiserver	ServiceNodePortRange	30000-32767
		EnableAdmissionPlugins	• 如果开启了PodSecurityPolicy：默认值为 `NodeRestriction，PodSecurityPolicy` • 如果关闭了PodSecurityPolicy：默认值为 `NodeRestriction`
	kube-controller-manager	HorizontalPodAutoscalerSyncPeriod	15s

6.边缘单元化管理

6.1. 边缘单元化管理概述

在边缘计算场景下，不同分组的节点间往往存在网络不互通、资源不共享、资源异构和应用独立等明显的隔离属性，边缘容器服务ACK@Edge针对这种场景提出了边缘单元化管理概念。本文为您简单介绍如何实现边缘单元化管理。

传统边缘管理

- 在边缘计算场景下，边缘节点通常具备很强的区域性、地域性、或者其他逻辑上的分组特性，比如具有相同的CPU架构、运营商或云提供商。
- 相同的应用和镜像，可能需要部署到不同的节点池中。
- 原生Kubernetes Service的后端端点扁平分布在集群中任意节点。因此，跨跃不同分组节点的Service流量，会大概率出现访问不可达、或者访问效率低下的问题。

边缘单元化管理

针对以上场景，ACK@Edge提出了如下图所示解决方案。

- 节点单元化：以节点池视角对不同边缘区域下的主机进行统一管理和运维。
- 应用的单元化：使用新的单元化部署模型将用户的工作负载部署在不同的节点池中，业务的实例数、版本都可以按照节点池的维度进行统一管理。
- 流量的单元化：通过配置Service拓扑来限制Service后端Endpoint的被访问范围，例如边缘节点应用只能由相同节点池的节点访问，或者只能由本节点访问。

6.2. 边缘节点池管理

6.2.1. 边缘节点池概述

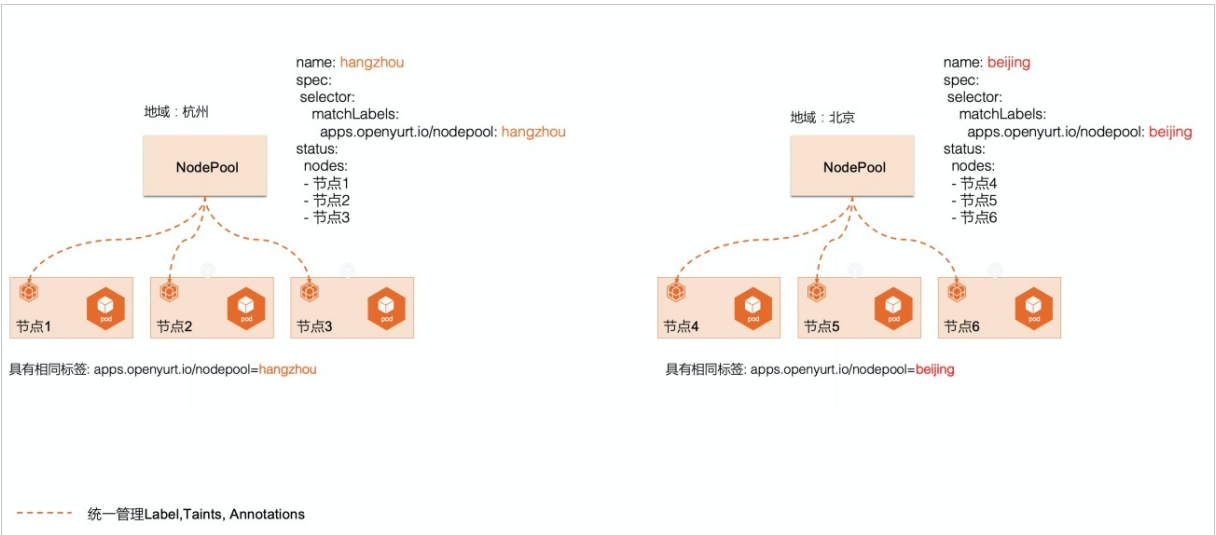
在边缘计算场景下，边缘容器服务ACK@Edge的**yurt-app-manager**组件提供了边缘节点池（NodePool）控制器功能，将节点按照特定属性抽象成节点池概念，以节点池的维度对不同边缘区域下的节点进行统一管理和运维。本文简单介绍边缘节点池的概念和工作原理。

传统节点管理

在边缘计算场景下，计算节点通常具有很强的特定属性，比如具有相同的CPU架构、运营商或云提供商等。为了便于管理，传统的做法是用Kubernetes打标签的方式来对节点进行分类管理，但是随着节点规模和标签数量的增加，对节点分类运维会变得越来越复杂，具体如下图所示。

边缘节点池

边缘节点池（NodePool）以节点池的维度对节点划分做了更高维度的抽象，您可以从节点池视角对不同边缘地域下的节点进行统一管理和运维，具体如下图所示。



更多信息，请参见边缘单元化管理概述。

相关文档

- 创建边缘节点池
- 向边缘节点池添加节点
- 创建增强型网络边缘节点池

6.2.2. 创建边缘节点池

您可以通过边缘节点池管理集群中的一组节点资源，例如在边缘节点池中统一管理节点的标签和污点。本文介绍如何创建边缘节点池。

前提条件

- 您已创建一个ACK边缘托管版集群，具体操作，请参见创建边缘托管版集群。
- 请确保您集群的Kubernetes版本大于等于1.16。

操作步骤

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
4. 在集群管理页左侧导航栏中，选择**节点管理 > 节点池**。
5. 在**节点池**页面中，单击页面右上角的**创建边缘节点池（Beta）**。
6. 在**创建边缘节点池（Beta）**页面完成创建节点池配置后，单击**提交**。

配置项	描述
节点池名称	节点池名称。
云边协同网络	可以选择基础型和增强型。更多信息，请参见创建增强型网络边缘节点池。
最大节点数量	节点池内可添加的最大节点数量。
节点标签	您可以为节点池内节点添加标签。
污点	您可以为节点池内节点添加污点。

节点池创建完成后，您可以在节点池列表中查看已创建的节点池信息。

相关文档

- 边缘节点池概述
- 创建边缘托管版集群
- 向边缘节点池添加节点

6.2.3. 向边缘节点池添加节点

您可以向已经创建的边缘节点池中添加工作负载节点，且需保证工作负载节点与Kubernetes API server的网络连通。本文介绍如何向边缘节点池添加节点。

前提条件

- 您已创建一个边缘节点池。具体操作，请参见创建边缘节点池。
- 请确保您集群的Kubernetes版本大于等于1.16。

> **注意** 在您使用集群前，请注意以下限制：
> - 仅支持添加操作系统为CentOS 7.4或7.6以及Ubuntu 18.04的版本。
> - 自动添加ENS节点，仅支持资源配置2核4 GB以上，操作系统为CentOS 7.4或7.6，且状态为Running的节点。
> - 集群的Kubernetes版本1.14.8-aliyunedge.1以上的版本开始支持ARM或ARM64架构的节点接入，且ARM架构支持的操作系统为CentOS 7.4，ARM64架构支持的操作系统为Ubuntu 18.04。

向边缘节点池添加节点

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
4. 在集群管理页左侧导航栏中，选择**节点管理 > 节点池**。
5. 在**节点池**页面中，单击目标节点池右侧**操作**列下的**添加已有节点**。

   后续操作步骤与向边缘集群添加节点的操作步骤相同，包括自动添加边缘节点服务ENS（Edge Node Service）节点到该节点池和手动添加ENS节点。具体操作，请参见添加边缘节点。

   节点添加成功后，您可以在目标边缘节点池右侧**操作**列的**详情**中查看已添加的节点列表。

相关文档

- 添加边缘节点

6.2.4. 创建增强型网络边缘节点池

增强型云边网络的功能基于ACK@Edge的SDN解决方案实现，它通过阿里云全球的接入点网格就近接入阿里云内网（CCN），同时通过CEN与云端VPC绑定，形成边缘和云端内网的通路。本文主要介绍增强型网络边缘节点池的实现原理和创建方法。

前提条件

- 有可用的云企业网实例和云连接网实例。具体操作，请参见创建CEN实例和创建云连接网。
- 规划网络，保证边缘节点的网段与云端VPC网段不发生冲突。

背景信息

边缘节点池支持**普通型**和**增强型**两种云边协同网络类型：

- **普通型**云边协同网络的边缘和云端通过公网互通，边缘节点池应用不能直接访问云端VPC内网。
- **增强型**基于ACK@Edge的SDN解决方案，提供安全、快速的云边协同网络，边缘节点池应用可直接通过VPC内网访问云端，并且相比于**普通型**网络，具有更好的云边网络质量和安全性保障。

描述	普通型	增强型
云边网络方案	公网	阿里云连接网。
是否能访问云端VPC内网	否	是。
云边网络质量	低	就近接入，网络质量高。

描述	普通型	增强型
安全性	低	加密，安全性高。
成本	低	高。 更多信息，请参见ACK@Edge计费说明。
适用场景	边缘业务对云端依赖不高	适用场景： • 业务对边缘、云端互相访问有强烈的需求。 • 对云边网络时延敏感，且要求保证网络质量。 • 对云边网络安全性有要求。

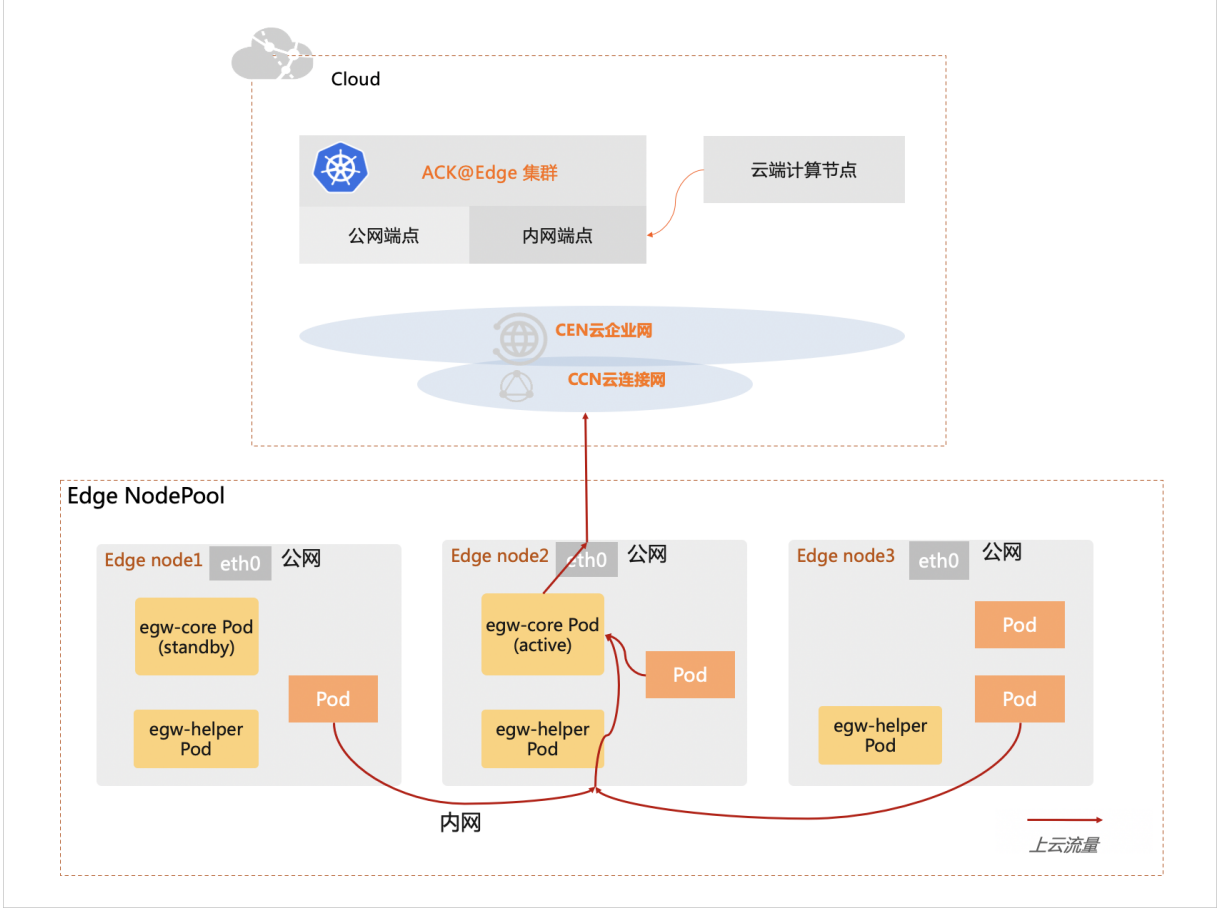
使用限制

- 目前增强型网络提供边缘Pod和云上Pod互通、边缘Pod和云上VPC互通、边缘节点单向访问云上VPC、边缘节点单向访问云上Pod的能力。暂不提供云上VPC、云上Pod访问边缘节点的能力，VPC内访问边缘节点仍需要通过EIP。
- 增强型网络节点池内需要保证至少接入2个以上AMD64架构的节点。
- 增强型网络节点池的网关组件以Pod部署在边缘节点上，目前只支持Flannel容器网络，不支持宿主机网络。
- 创建普通型或增强型网络边缘非默认节点池时，需指定节点池内最大接入节点数量，该值将被记录在NodePool资源对象的Annotation上，且不能修改，请做好节点池规模规划。
- 增强型网络节点池相关元信息会记录在K8s NodePool资源对象的Annotation上，您无需关注这些Annotation的管理，但请务必不能修改或删除这些Annotation，否则可能导致增强型网络不工作。更多关于Annotation的详细信息，请参见边缘节点池Annotation说明。
- 节点通过标签 `openyurt.io/desired-nodepool` 标识该节点归属的节点池，增强型网络节点池不支持通过修改节点标签完成在节点池之间迁移。如您需要迁移节点，请先下线节点再重新接入，并指定新的节点池，否则该节点将无法使用增强型网络模式。具体操作，请参见移除边缘节点。

增强型云边网络节点池实现原理

增强型云边网络基于ACK@Edge的SDN解决方案实现，依托于阿里云的全球网络基础设施，提供可靠、安全的云边通信能力。当您创建增强型网络边缘节点池，并完成节点接入后，您的边缘节点上将会自动部署增强型网关Pod。增强型网关通过阿里云全球的接入点网格就近接入阿里云内网（CCN），同时通过CEN与云端VPC绑定，形成边缘和云端内网的通路。该模式下，云边网络流量会经过加密，并且几乎全程在阿里云内网传输，能保证通信的质量和安全性。同时，边缘网络和云端VPC内网直接打通，可直接访问VPC内网服务。

当您创建增强型网络边缘节点池时，管控组件会在您边缘节点池上部署**edge-gateway-core(egw-core)**和**edge-gateway-helper(egw-helper)**组件。其中，**edge-gateway-core**为增强型网关核心组件，以Deployment方式部署，每个节点池有主、从两个实例，分布在不同节点上以保证高可用。**edge-gateway-helper**为节点路由同步组件，以Daemonset方式部署在节点池内的每个节点上，用来配置节点上的路由信息。



创建增强型网络边缘节点池

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
4. 在集群管理页左侧导航栏中，选择**节点管理 > 节点池**。
5. 在**节点池**页面中，单击页面右上角的**创建边缘节点池（Beta）**。
6. 在**创建边缘节点池（Beta）**页面，完成创建节点池剩余配置项。更多信息，请参见创建边缘节点池。
   - **云边协同网络**选择为**增强型**。
   - **云企业网**：
     - 如果使用您账号下的云企业网，请先选中**使用此账号云企业网**，然后选择您账号下相应的云企业网实例。
     - 如果使用其他账号下的云企业网，请先将当前集群的VPC和云连接网实例，跨账号授权并绑定到其他账号的云企业网实例下。具体操作，请参见跨账号网络实例授权和加载网络实例。然后选中**使用其他账号云企业网**，填写相应账号UID及云企业网实例ID。
   - **云连接网**选择为您账号下已创建的云连接网实例。
7. 单击**提交**。
8. 节点池创建完成后，需要向该节点池内至少添加2个节点。具体操作，请参见向边缘节点池添加节点。

名词解释

- 云企业网CEN（Cloud Enterprise Network），支持不同地域VPC间、VPC与本地数据中心间搭建私网通信通道，实现全网资源的互通。
- 云连接网CCN（Cloud Connect Network），分布式接入网关组成的设备接入矩阵，可以将云连接网绑定到云企业网，实现云下和云上全连接。

边缘节点池注解说明

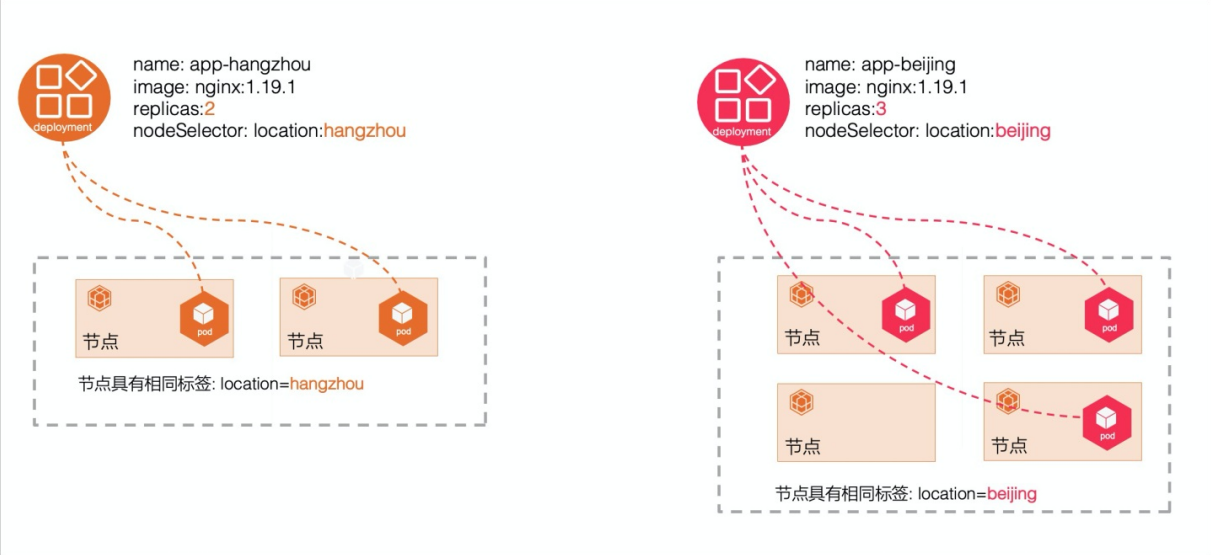
Annotation	说明
nodepool.openyurt.io/max-nodes	指定NodePool内最大接入Node数，只在边缘非默认节点池上存在。
nodepool.openyurt.io/pod-cidrs	预分配给NodePool的PodCIDR集合，只在边缘非默认节点池上存在。
nodepool.openyurt.io/cen-id	使用增强型网络NodePool的CEN ID。
nodepool.openyurt.io/ccn-id	使用增强型网络NodePool的CCN ID。
nodepool.openyurt.io/ccn-region	使用增强型网络NodePool的CCN Region，国内为cn-shanghai。
nodepool.openyurt.io/is-default	表示NodePool是否为边缘默认节点池。

6.3. 使用单元化部署应用模型

在边缘计算场景下，您可以以单元化部署模型将工作负载部署在不同的节点池中，业务的实例数、版本都可以按照节点池的维度进行统一管理。本文介绍如何使用单元化部署应用模型。

背景信息

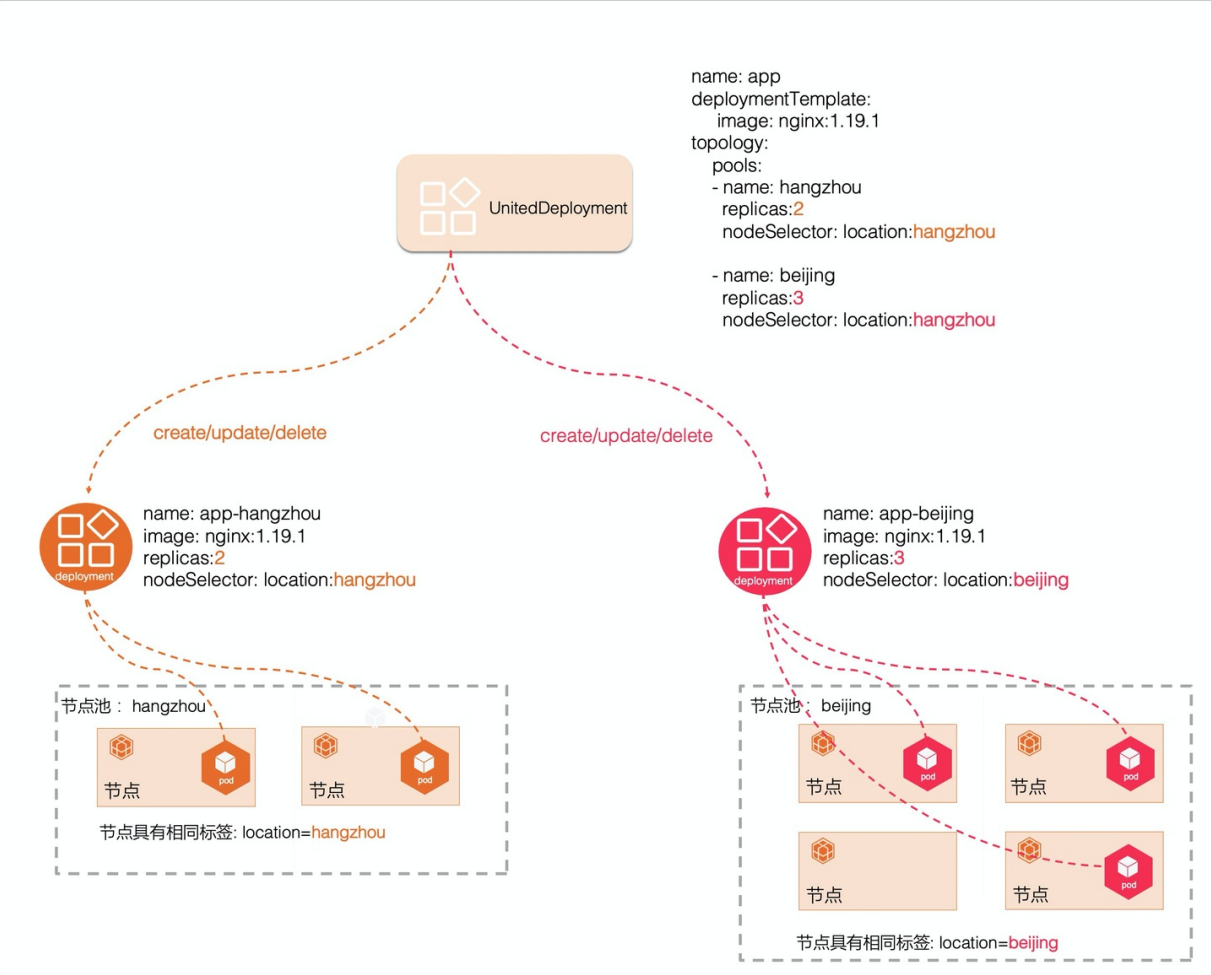
在边缘计算场景下，计算节点具有很明显的地域分布属性，相同的应用可能需要部署在不同地域下的计算节点上。以Deployment为例，传统的做法是先将相同地域的计算节点设置成相同的标签，然后创建多个Deployment，不同Deployment通过NodeSelectors选定不同的标签，从而实现将相同的应用部署到不同地域的需求。



随着地域分布越来越多，以及不同地域对应用的差异化需求，使得运维变得越来越复杂，具体表现在以下几个方面：

- 当镜像版本升级，需要修改每个Deployment的镜像版本配置。
- 需要自定义Deployment的命名规范来表明相同的应用。
- 相同应用的多个Deployment，除了Name、NodeSelectors和Replicas这些特性外，其他的差异化配置比较小。

单元化部署（UnitedDeployment）是边缘容器服务ACK@Edge提供的功能，通过更上层次的抽象，对多个Deployment进行统一管理，比如创建、更新和删除等操作。



单元化部署提供一个模板来定义应用，将多个Workload部署到不同的区域， 每个区域定义为一个节点池。目前单元化部署支持两种类型的Workload：StatefulSet和Deployment。控制器会根据单元化部署中节点池的配置创建子的Workload资源对象，每个资源对象都有一个期望的Replicas Pod数量。通过一个单元化部署实例就可以自动维护多个Deployment或者Statefulset资源，同时还能实现Name、NodeSelectors和Replicas等的差异化配置。

创建单元化部署实例

创建一个Workload模板为Deployment的UnitedDeployment单元化部署实例。

完整的YAML示例模板如下：

```
apiVersion: apps.openyurt.io/v1alpha1
kind: UnitedDeployment
metadata:
```

```
  name: example
  namespace: default
spec:
  revisionHistoryLimit: 5
  selector:
    matchLabels:
      app: example
  workloadTemplate:
    deploymentTemplate:
      metadata:
        creationTimestamp: null
        labels:
          app: example
      spec:
        selector:
          matchLabels:
            app: example
        template:
          metadata:
            creationTimestamp: null
            labels:
              app: example
          spec:
            containers:
            - image: nginx:1.19.3
              imagePullPolicy: Always
              name: nginx
            dnsPolicy: ClusterFirst
            restartPolicy: Always
  topology:
    pools:
    - name: cloud
      nodeSelectorTerm:
        matchExpressions:
        - key: apps.openyurt.io/nodepool
          operator: In
          values:
          - cloud
      replicas: 2
    - name: edge
      nodeSelectorTerm:
        matchExpressions:
        - key: apps.openyurt.io/nodepool
          operator: In
          values:
          - edge
      replicas: 2
      tolerations:
      - effect: NoSchedule
        key: apps.openyurt.io/taints
        operator: Exists
```

相关字段的解释如下表所示：

字段	含义
spec.workloadTemplate	代表支持的Workload模板，目前节点池支持 deploymentTemplate/statefulSetTemplate 两种模板。
spec.topology.pools	指定多个节点池。
spec.topology.pools[*].name	节点池的名称。
spec.topology.pools[*].nodeSelectorTerm	节点池的主机亲和性配置若需与节点池NodePool相对应，Key使用 apps.openyurt.io/nodepool ，Value使用节点池名字。
spec.topology.pools[*].tolerations	节点池的主机容忍性配置。
spec.topology.pools[*].replicas	每个节点池下Pod的实例数。

使用单元化部署应用模型实例管理Pod

- 升级Pod：通过修改 spec.template.workloadTemplate.deploymentTemplate 下面的字段触发升级流程，控制器把新的模板更新到各个节点池下的Workload里触发节点池控制器升级Pod。
- 扩容多个节点池下Pod的Replicas数目：通过修改 spec.topology.pools 下不同节点池的Repicas配置，触发相应节点池下应用Pod的扩缩容操作。

6.4. 配置Service流量拓扑

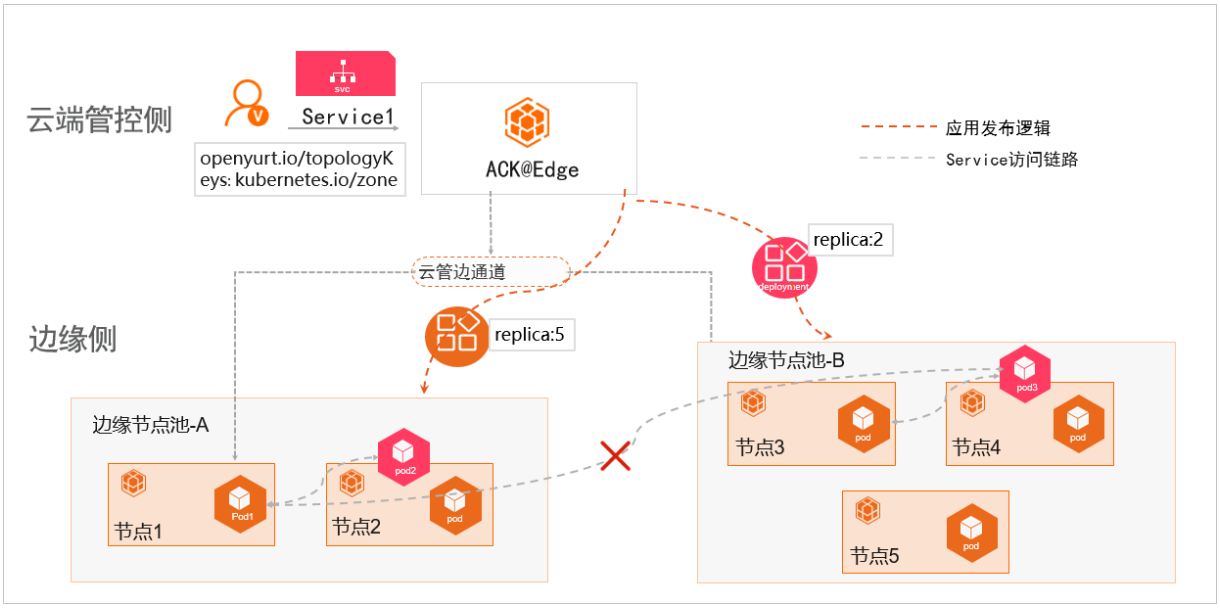
原生Kubernetes Service的后端端点扁平分布在集群中任意节点。因此，跨跃不同分组节点的Service流量，会大概率出现访问不可达、或者访问效率低下的问题。Service流量拓扑支持边缘节点应用只能由相同节点池的节点访问，或者只能由本节点访问。本文为您介绍Service流量拓扑管理功能和配置Service流量拓扑方法。

背景信息

在边缘计算场景下，边缘节点通常具备很强的区域性、地域性、或者其他逻辑上的分组特性（比如具有相同的CPU架构、运营商或云提供商），不同分组的节点间往往存在网络不互通、资源不共享、资源异构、应用独立等明显的隔离属性。

Service流量拓扑管理实现原理

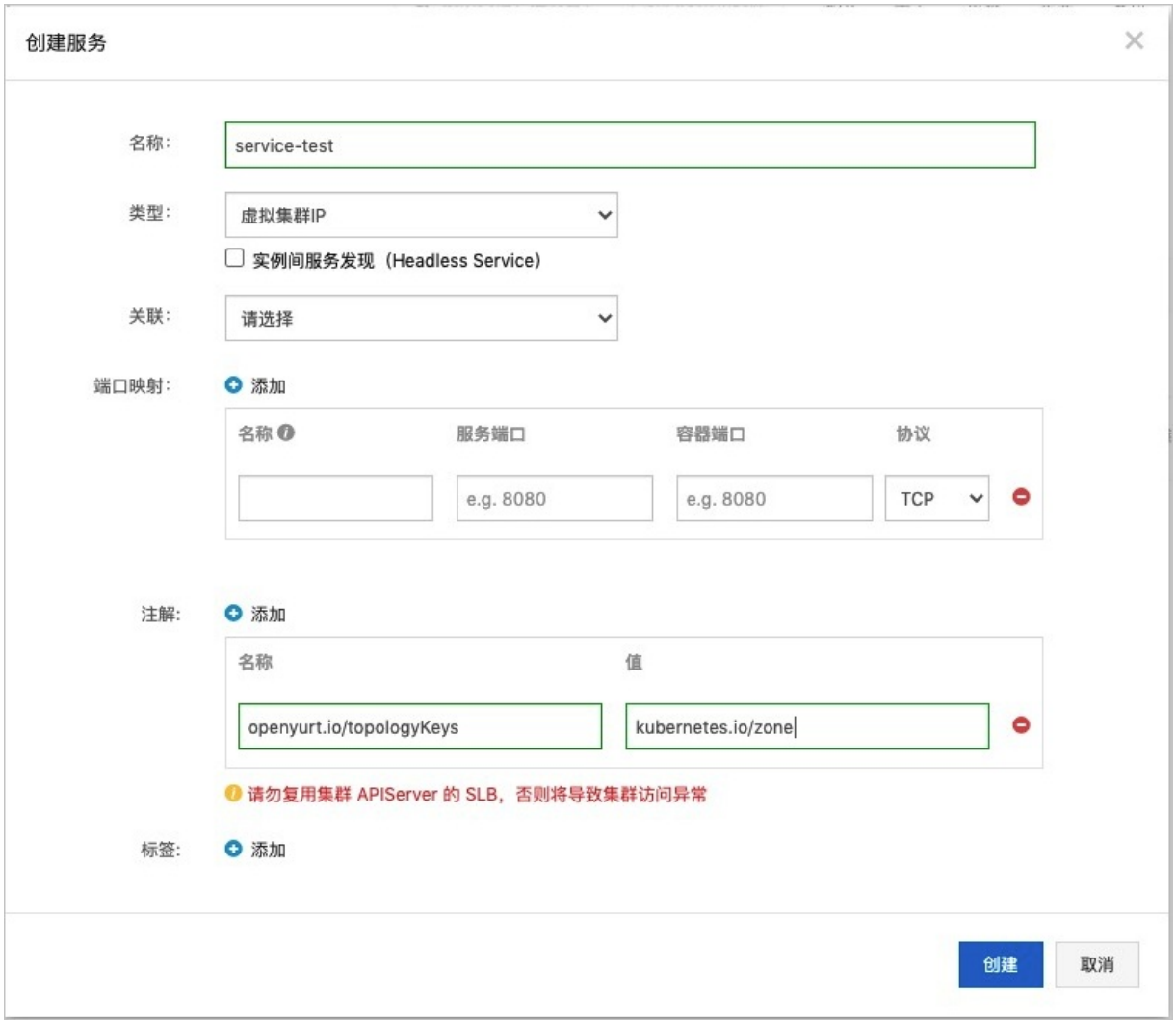
为了解决上述问题，ACK@Edge在原生的Service之上，增加了Endpoint的拓扑管理功能，即通过简单配置来限制Service后端Endpoint的访问范围。例如边缘节点应用只能由相同节点池的节点访问，或者只能由本节点访问。具体实现原理如下图所示。



- Service1关联后端Pod2和Pod3两个实例，且Service1通过`annotation："openyurt.io/topologyKeys: kubernetes.io/zone"`配置了其拓扑节点池范围。
- Pod2所在的节点2和Pod3所在的节点3分别属于两个不同的节点池A和节点池B。
- 因为Pod3和Pod1不在一个节点池，当Pod1访问Service1时，流量只会转发到Pod2上，访问Po3的流量被限制。

方式一：通过控制台配置Service拓扑

若您需要创建一个限制在本节点池内被访问的Service，只需要给Service添加注解即可。例如将**名称**配置为`openyurt.io/topologyKeys`，**值**配置为`kubernetes.io/zone`。关于创建服务的更多信息，请参见管理服务。



方式二：通过命令行配置Service拓扑

通过命令行配置Service拓扑有以下两种方式：

- 新建一个使用节点池拓扑域的Service，YAML样例如下。

```
apiVersion: v1
kind: Service
metadata:
 annotations:
  openyurt.io/topologyKeys: kubernetes.io/zone
 name: my-service-nodepool
 namespace: default
spec:
 ports:
 - port: 80
  protocol: TCP
  targetPort: 8080
 selector:
  app: nginx
 sessionAffinity: None
 type: ClusterIP
```

- 通过执行以下命令行配置Service拓扑，该Service使用的是节点池拓扑域。

```
kubectl annotate service xxx openyurt.io/topologyKeys='kubernetes.io/zone'
```

注解说明

通过在原生的Service上添加Annotation实现流量的拓扑配置，相关参数如下所示。

annotation Key	annotation Value	说明
openyurt.io/topologyKeys	kubernetes.io/hostname	限制Service只能被本节点访问。
openyurt.io/topologyKeys	kubernetes.io/zone或 openyurt.io/nodepool	限制Service只能被本节点池的节点访问可。边缘集群版本如果大于等于1.18，推荐您使用openyurt.io/nodepool。
-	-	Service不做任何拓扑限制。

7.边缘节点管理

7.1. 添加边缘节点

您可以向已经创建的边缘托管集群中添加工作负载节点，工作负载节点需要能够保证和Kubernetes Apiserver的网络联通。边缘托管集群支持接入云上ECS节点、云上边缘节点服务ENS（Edge Node Service）节点、非云节点等。

前提条件

- 如果之前没有创建过边缘托管集群，您需要先创建边缘Kubernetes集群。具体操作，请参见创建边缘托管版集群。
- 如果需要自动添加ENS节点，您需要先创建边缘服务。具体操作，请参见创建边缘服务。

使用限制

- 边缘集群托管服务在公测期间，每个集群中最多可包含40个节点。如果您需要添加更多节点，请提交工单申请。
- 自动添加ENS节点，仅支持资源配置2核4 GB以上，操作系统为CentOS 7.4或7.6，且状态为Running的节点。
- 当您选择手动接入节点时，支持接入的节点操作系统列表如下。

系统架构	系统版本	系统内核版本	边缘Kubernetes集群版本
AMD64	CentOS 7.4	3.10.X	≥1.12.6-aliyunedge.1
AMD64	CentOS 7.6	3.10.X	≥1.12.6-aliyunedge.1
AMD64	CentOS 8.0	4.18.X	≥1.18.8-aliyunedge.1
AMD64	Ubuntu 18.04	4.15.X	≥1.12.6-aliyunedge.1
AMD64	Ubuntu 18.04	5.4.X	≥1.16.9-aliyunedge.1
AMD64	Ubuntu 20.04	5.4.X	≥1.18.8-aliyunedge.1
ARM64	CentOS 8.0	4.19.X	≥1.14.8-aliyunedge.1
ARM64	Ubuntu 18.04	4.9.X	≥1.14.8-aliyunedge.1
ARM64	Ubuntu 18.04	4.19.X	≥1.14.8-aliyunedge.1
ARM	CentOS 7.7	4.19.X	≥1.14.8-aliyunedge.1

添加节点

1. 登录容器服务管理控制台。
2. 添加已有节点。您可以通过以下两个入口进行操作。
   - 入口一：

a. 在控制台左侧导航栏中，单击**集群**。
b. 选择目标集群并单击右侧**操作**列下的**更多 > 添加已有节点**。
c. 在**节点池**页面，选择目标节点池右侧**操作**列的**更多 > 添加已有节点**。

- 入口二：
  a. 在控制台左侧导航栏中，单击**集群**。
  b. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
  c. 在集群管理页左侧导航栏中，选择**节点管理 > 节点池**。
  d. 在**节点池**页面，选择目标节点池右侧**操作**列的**更多 > 添加已有节点**。

3. 进入添加节点页面，您可以选择**自动添加**或**手动添加**的方式，添加现有实例。
   - 您可选择自动添加的方式，您可以一次性添加多个ENS实例。

     > **说明** 目前自动添加的方式仅支持添加云上ENS节点。

     a. 选择**自动添加**，在已有ENS实例的列表中，选择所需的ENS实例，然后单击**下一步**。

     b. 确认实例信息无误后，单击**下一步**。
     c. 在弹出的对话框中，单击**确定**，进入**添加完成**页面。

     单击**去集群列表查看**，您可以在集群列表中看到ENS实例已添加到该集群中。

     > **说明** ENS实例成功加入集群需要大概2分钟。

   - 选择手动添加的方式。

     > **说明** 目前手动添加的方式支持添加云上ECS节点，云上ENS节点和非云节点。

     a. 选择**手动添加**，然后单击**下一步**。

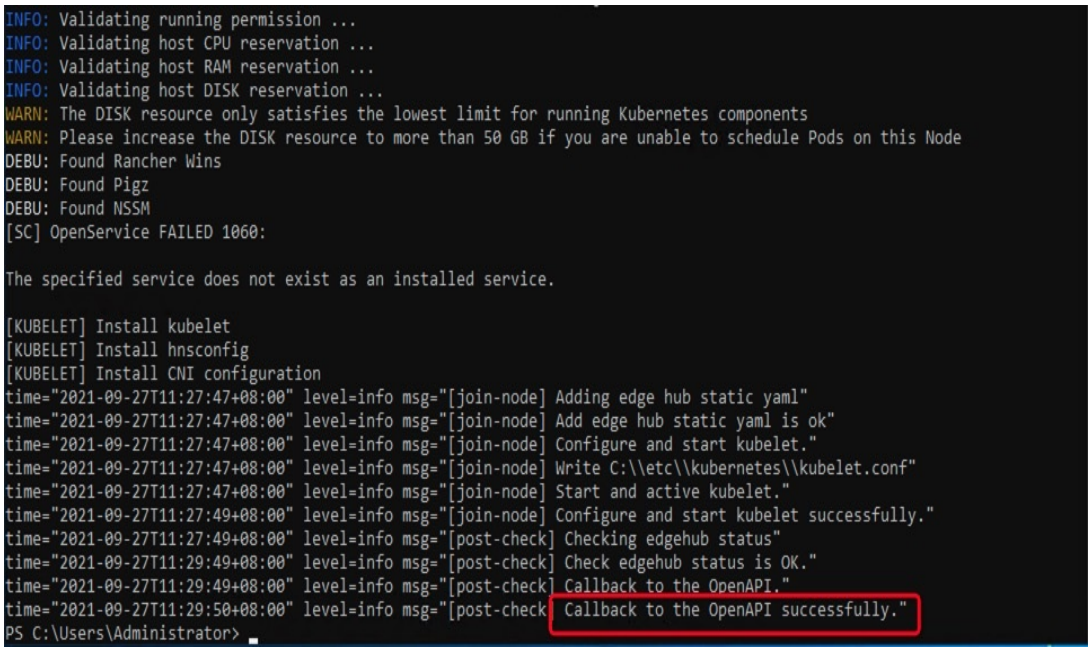
     b. 进入实例信息页面，您可以填写节点接入配置，具体的配置参数，请参见参数列表。

     > **说明** **脚本有效时间**的默认值是1小时，如果您需要长时间使用同一个脚本做批量添加，可以适当增加脚本的有效时间。当**脚本有效时间**配置为0小时时，表示脚本永久有效。

     c. 配置完成后单击**下一步**。

d. 进入**添加完成**页面，单击**复制**后，到您的边缘节点上粘贴并执行该脚本。

添加边缘节点成功的结果如下图所示。



参数列表

参数	参数说明	默认值
flannelIface	Flannel使用的网卡名。	节点默认路由的网卡名
enableIptables	是否开启 `iptables` 。	`false`
quiet	假设所有的问题回答自动回复 `yes` 。	`false`
manageRuntime	是否由接入工具安装并检测Runtime。	`false`
nodeNameOverride	设置节点名。	• ""（默认值，表示使用主机名） • "*"（表示随机生成6位的字符串） • "*.XXX"（表示随机生成6位字符串+XXX后缀）
allowedClusterAddons	需要安装的组件列表。默认为空，不安装。普通节点需要配置为["kube-proxy","flannel","coredns"]。	[]
gpuVersion	表示要接入的节点是否为GPU节点，默认为空。当前支持的GPU版本是Nvidia_Tesla_T4，Nvidia_Tesla_P4，Nvidia_Tesla_P100。	""（默认值，表示不作为GPU节点接入）
inDedicatedNetwork	表示是否通过专线接入边缘托管集群。	`false`

参数	参数说明	默认值
labels	表示接入时节点要加的标签。	{}
annotations	表示接入时给节点加的注解。	{}

nodeIface	该参数有两个作用：
	• kubelet从指定的网络接口获取节点IP信息。如果没有指定这个参数，kubelet将按如下顺序获取节点IP。
	  ◦ 从*/etc/hosts*中寻找与主机名同名的记录。
	  ◦ 默认路由所在的网络接口的IP地址。
	• 表示Flannel使用的网卡名，这里与参数flannelIface同义，后续flannelIface会用这个参数替代。
		""

7.2. 设置节点自治

本文主要为您介绍如何为边缘节点设置节点自治属性。节点自治可以保证在边缘和云端网络断连状态下，边缘节点上的业务应用可以持续稳定的运行，而不会被驱逐或者迁移到其他边缘节点。

前提条件

- 创建边缘托管版集群
- 添加边缘节点

背景信息

您可以通过ACK控制台设置节点自治属性，包括设置节点自治和节点非自治两种配置：

- 当边缘节点被设置为自治状态时，如果边缘节点和云端管控断连，此时不但系统能够保证节点上应用不会被驱逐，而且节点上的应用也会自动恢复。设置节点自治适用于边缘计算的弱网络连接场景。
- 当节点被设置为非自治状态时，如果边缘节点和云端管控断连，节点因不能正常的将心跳上报至管控端，而会被设置为**不可用（not ready）**状态，且节点上的应用在到达容忍时间之后将会被驱逐。

操作步骤

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面选择目标集群，单击集群名称或者**操作**列下的**详情**。
4. 在集群管理页左侧导航栏中，选择**节点管理 > 节点**。
5. 在**节点**页面，选择目标节点**操作**列的**更多 > 节点自治设置**。

   > **说明** 仅当前节点是边缘节点才会有**节点自治设置**按钮。

6. 在弹出的**节点自治设置**对话框中，单击**确定**。

> **说明** 边缘节点初始接入集群后，默认为非自治状态。您可以参照上述步骤开启或者关闭节点自治属性。

相关文档

- ACK@Edge概述
- 添加边缘节点

7.3. 移除边缘节点

您可以从已经创建的边缘托管集群中移除不需要的工作负载节点。本文主要介绍如何移除边缘节点。

前提条件

- 创建边缘托管版集群
- 通过kubectl工具连接集群

背景信息

- 移除节点会涉及Pod迁移，可能会影响业务，请在业务低峰期操作。
- 操作过程中可能存在非预期风险，请提前做好相关的数据备份。
- 操作过程中，后台会把当前节点设置为不可调度状态。
- 移除节点仅移除Worker节点，不会移除Master节点。
- 边缘集群存在云端节点和边缘节点两种类型的节点，两种类型的可以同时移除。
- 边缘集群至少需要保留一个云端节点。
- 移除节点请通过控制台进行操作，如果通过执行 `kubectl delete node` 命令行方式手动移除节点，则：
  - 对于云端节点：
    - 移除后的节点无法再添加到其他集群上。
    - 删除集群时，该节点所在的ECS实例会被释放。
  - 对于边缘节点：需要使用接入工具Edgeadm的Reset子命令重置节点之后才能接入其它集群。

操作步骤

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
4. 在集群管理页左侧导航栏中，选择**节点管理 > 节点**。
5. 在**节点**页面的目标节点右侧**操作**列，选择**更多 > 移除**。

   > **说明** 如果需要同时移除多个节点，可在**节点**页面，同时选中要移除的节点，单击**批量移除**。

6. （可选）在**移除节点**对话框中，如果上个步骤中所选的被移除节点全部是云端节点时，您可选中**同时释放ECS**和**自动排空节点（drain）**，单击**确定**。

> **说明** 如果上个步骤中所选的被移除节点包含边缘节点或者全部是边缘节点时，在**移除节点**对话框中不支持**同时释放ECS**和**自动排空节点（drain）**这两个选项。

- 同时释放ECS：
  - 释放ECS实例仅释放按量付费的ECS实例。
  - 对于预付费ECS实例，计费周期到期后，ECS实例会自动释放。
  - 您也可以在ECS实例到期前：
    - 申请退款，提前释放实例。具体操作，请参见退款规则及退款流程。
    - 将计费方式转为按量付费后释放实例。具体操作，请参见包年包月转按量付费。
  - 若不选择同时释放ECS，该节点所在的ECS实例会继续计费。
- 自动排空节点（drain）：把待移除节点上的Pod转移到其他节点。请确保集群其他节点的资源充足。您还可以通过执行 `kubectl drain node-name` 命令的方式把待移除节点上的Pod转移到其他节点。

> **说明** *node-name*格式为*your-region-name.node-id*。例如*cn-hangzhou.i-xxx* 。
>
> - *your-region-name*为您集群所在的地域名称。
> - *node-id*为待移除节点所在的ECS实例ID。

8.边缘扩展功能

8.1. 边缘网络自治

边缘网络自治可以保证在异常状态下应用恢复后，应用间的网络通信自动恢复。本文主要为您介绍边缘节点上的网络自治功能。

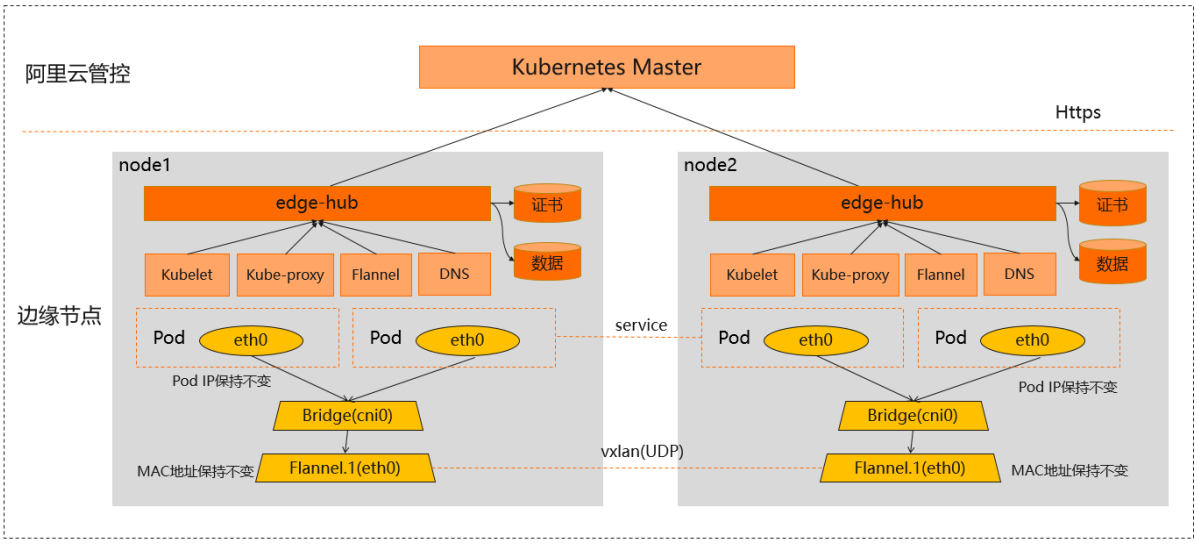
背景信息

边缘节点接入云端集群后，默认具备边缘网络自治能力。

- 具备网络自治能力的节点，节点上应用Pod IP与Pod Name保持绑定关系。无论是应用重启还是节点重启，Pod IP都将保持不变。同时容器网络VTEP（flannel.1虚拟网卡）的MAC地址跟Node Name保持绑定关系。无论是Flannel容器重启还是节点重启，VTEP的MAC地址都保持不变。
- 具备网络自治能力的节点，即使在边缘节点和云端管控网络断连等异常情况下，业务重启还是节点重启后，节点内或者跨节点间的业务应用通信都将自动恢复。适用于边缘计算的弱网络连接状态下应用跨节点通信的场景。

功能说明

- 无论业务应用选择主机网络模式还是非主机网络模式部署，业务应用都默认具备边缘网络自治能力。
- 边缘网络自治功能，保证异常状态下应用恢复后，跨节点应用间的网络通信自动恢复。具体如下所示。



> **说明** 当Pod删除重建或者迁移到其他节点后，Pod IP将发生变化。

8.2. 边缘运维通道

为了给您提供完整的Kubernetes集群使用体验，集群创建完成后，系统会默认部署edge-tunnel-server/edge-tunnel-agent组件来创建云端和边缘之间的运维通道，为您提供云端访问边缘端的能力。本文主要为您介绍边缘托管集群中的边缘运维通道关联组件和功能，以及如何扩展边缘监控能力。

背景信息

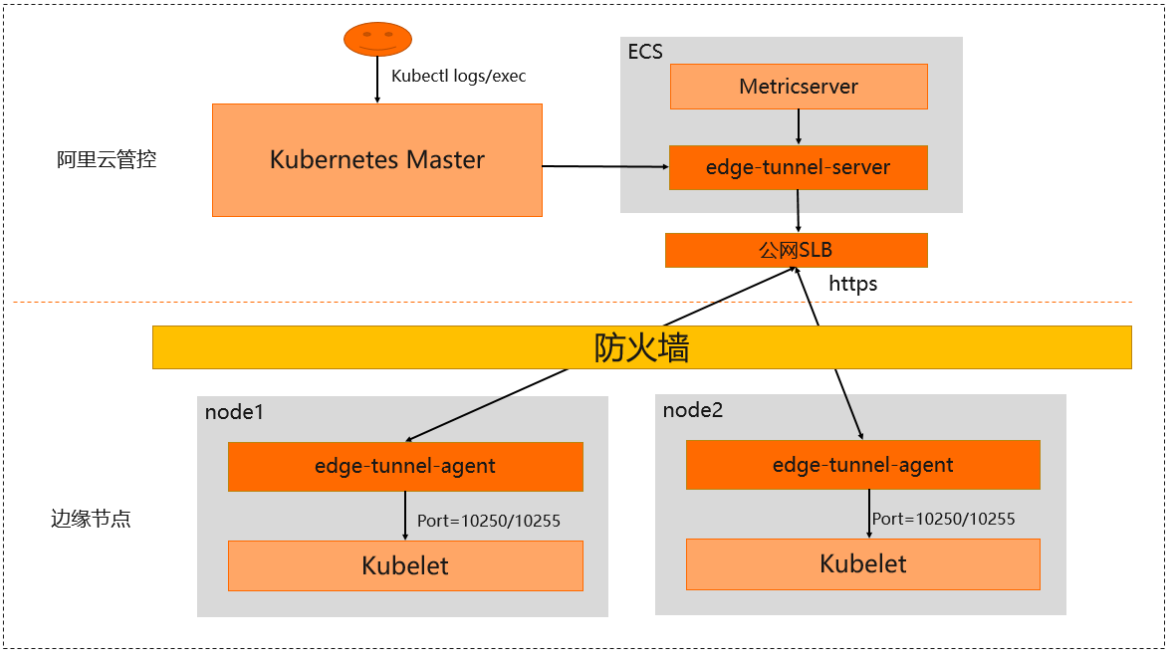
- 原生Kubernetes集群中，云端管控组件需要直接访问边缘节点的Kubelet来执行运维命令，或者云端运维

监控组件metrics-server需要从云端拉取边缘的监控指标数据。在边缘托管集群的场景下，当您的边缘节点部署在内网时，云端无法直接访问边缘节点。

- **edge-tunnel-server**采用Deployment模型部署在云端节点上。**edge-tunnel-agent**采用Daemonset模型部署在边缘节点上。
- 无论是Kubernetes原生运维命令（例如：kubectl logs或kubectl exec），还是metrics-server组件，都是通过访问边缘节点的kubelet组件的10250和10255端口来下发运维命令的。

功能说明

- 创建集群时，您需要选择购买至少1台云端ECS节点，用于部署边缘运维通道组件**edge-tunnel-server**。
- 为创建安全加密的公网运维通道，系统会为**edge-tunnel-server**组件创建的服务对象购买一个SLB，边缘节点上的**edge-tunnel-agent**将通过该SLB与edge-tunnel-server建立安全加密的运维通道。
- 当云端组件（例如：kube-apiserver、metrics-server）访问边缘节点10250和10255端口时，边缘集群默认会将访问请求自动导流到edge-tunnel-server组件，云端组件无需做任何修改。
- 本文具体实现原理如下图所示。



> **说明**
>
> - 当边缘节点和云端网络断连或者弱连接状态下，边缘运维通道可能无法正常工作。
> - 当您无意中删除或者停止了运维通道使用的SLB实例，边缘运维通道将无法正常工作。
> - 低版本集群（例如：v1.16.9-aliyunedge.1）的云端组件（例如：metrics-server）和edge-tunnel-server需要部署在同一个ECS节点，组件才能正常工作。从v1.18.8-aliyunedge.1集群版本开始，支持云端组件（例如：metrics-server）和edge-tunnel-server部署在不同ECS节点。

配置监控边缘节点的非默认端口

- 业务上云过程中需要迁移原系统中的运维监控方案到云上，实现云下系统到云上的无缝迁移，需要访问边缘节点的非默认端口（即非10250和10255端口）来收集监控数据，例如：访问边缘节点9051端口的监控数据。
- 边缘容器服务提供以下方式来配置云端组件访问边缘节点的非默认端口（即非10250和10255端口）：

- 更新 kube-system/edge-tunnel-server-cfg configmap 中的 dnat-ports-pair 字段。
- dnat-ports-pair 字段格式为： 访问端口1=10264，访问端口2=10264 。

通过云端访问边缘节点9051和9052端口的监控数据， 您需要做以下配置：

```
cat <<EOF | kubectl apply -f
apiVersion: v1
data:
 dnat-ports-pair: '9051=10264,9052=10264'
kind: ConfigMap
metadata:
 name: edge-tunnel-server-cfg
 namespace: kube-system
EOF
```

> **说明** 边缘节点的非默认端口（即非10250和10255端口）的监控数据只支持通过HTTP协议访问。

8.3. 使用LVM本地存储

ACK@Edge Pro版集群支持LVM（Logical Volume Manager）本地存储，提供自动化的逻辑卷生命周期管理能力，且能根据节点LVM本地存储容量进行调度。您只需定义节点本地盘的拓扑关系，即可通过原生PVC/PV方式使用LVM本地存储。本文介绍边缘Pro版集群如何使用LVM本地存储。

前提条件

集群的节点有可用的本地数据盘。

安装node-resource-manager和csi-local-plugin组件

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面，选择目标集群，并在目标集群右侧**操作**列下，选择**更多 > 系统组件管理**。
4. 在**组件管理**页面的存储区域，找到**node-resource-manager**和**csi-local-plugin**组件，单击**安装**。
5. 在**提示**对话框中单击**确定**。

配置节点VolumeGroup

> **说明** 为了保障数据安全，组件不会对VolumeGroup和Physical Volume进行删除。如果需要重新定义VolumeGroup，您需要先自行清理。

1. 使用如下YAML文件样例配置ConfigMap，指定节点VolumeGroup拓扑配置。

```
apiVersion: v1
kind: ConfigMap
metadata:
 name: node-resource-topo
 namespace: kube-system
data:
 volumegroup: |-
  volumegroup:
  - name: volumegroup1
    key: kubernetes.io/storagetype
    operator: In
    value: lvm
    topology:
     type: device
     devices:
     - /dev/sdb1
     - /dev/sdb2
     - /dev/sdc
```

参数解释如下：

参数	说明
name	VolumeGroup的名字。
key	匹配集群节点标签中的key的值。
operator	集群定义的 `Labels selector operator` ，主要包含如下四种操作符： ○ In：只有 `value` 的值与集群节点标签中的key对应的 `value` 值相同时才会匹配。 ○ NotIn：只有 `value` 的值与集群节点标签中的key对应的 `value` 值不相同时才会匹配。 ○ Exists：集群节点标签存在key就会匹配。 ○ DoesNotExist：集群节点标签不存在key就会匹配。
value	匹配Kubernetes Node Labels的key对应的 `value` 的值。
topology	节点上设备拓扑关系，其中 `topology.devices` 指定节点上的本地盘路径，该设备将被加到VolumeGroup中。

2. 给节点打标。
   - 按步骤1中的标签规则给相应存储节点添加对应自定义标签，以指定符合相应拓扑的节点类型。如步骤1中对应的标签为： `kubernetes.io/storagetype=lvm` 。
   - 给存储节点添加固定标签： `alibabacloud.com/edge-enable-localstorage='true'` ，使本地存储管理组件能调度到该节点。

   节点上的node-resource-manager组件将根据以上配置，自动创建对应的Physical Volume，并将其加入到VolumeGroup中。

使用LVM本地存储

使用以下YAML文件样例创建PVC指定StorageClass，并执行 `kubectl apply -f ****.yaml` 命令。一个PVC对应节点上一块逻辑卷，Pod创建成功后将挂载该逻辑卷。

> **说明** 集群默认的 `storageClassName` 为 `csi-local-lvm` 。

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: lvm-pvc-test
spec:
 accessModes:
 - ReadWriteOnce
 resources:
  requests:
   storage: 50Mi
 storageClassName: csi-local-lvm
---
apiVersion: apps/v1
kind: Deployment
metadata:
 labels:
  k8s-app: local-test
 name: local-test
 namespace: default
spec:
 replicas: 1
 selector:
  matchLabels:
   k8s-app: local-test
 template:
  metadata:
   labels:
    k8s-app: local-test
  spec:
   hostNetwork: true
   containers:
   - image: nginx:1.15.7-alpine
    imagePullPolicy: IfNotPresent
    name: nginx
    resources: {}
    volumeMounts:
     - name: local-pvc
       mountPath: /data
   dnsPolicy: ClusterFirst
   restartPolicy: Always
   tolerations:
   - operator: Exists
   nodeSelector:
    alibabacloud.com/is-edge-worker: "true"
   volumes:
   - name: local-pvc
    persistentVolumeClaim:
     claimName: lvm-pvc-test
```

执行以下命令，查看逻辑卷是否挂载成功。

```
kubectl exec -it local-test-564dfcf6dc-qhfsf sh
/ # ls /data
```

预期输出：

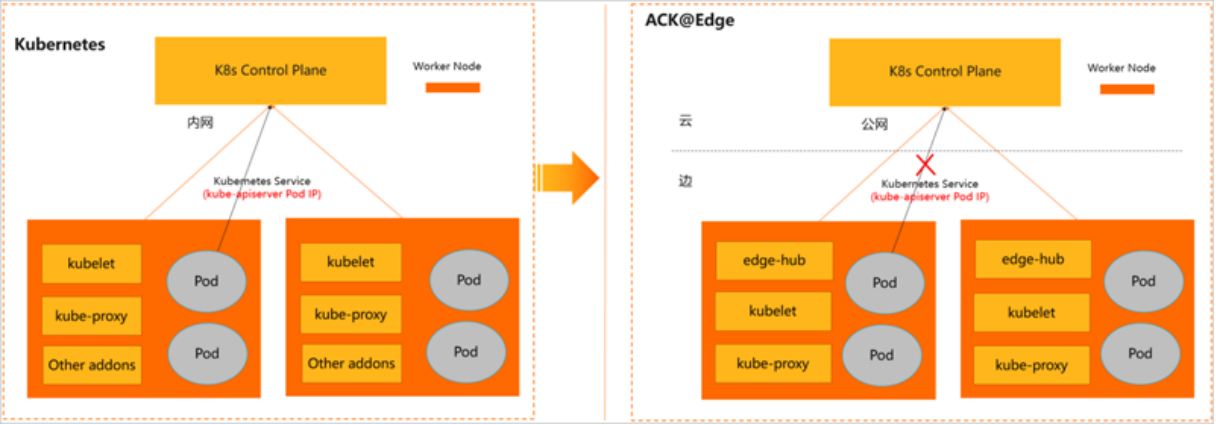
```
lost+found
```

从上述输出信息，可以知道逻辑卷已成功挂载在Pod上。

8.4. 在边缘场景无缝运行使用InClusterConfig的业务Pod

ACK@Edge是业界首个非侵入的边缘计算云原生产品服务，为了给您提供云边一体化的使用体验，通过InClusterConfig访问kube-apiserver的业务Pod，无需任何修改，可以直接被部署到边缘环境。本文介绍如何在边缘场景无缝运行使用InClusterConfig的业务Pod。

背景信息



当需要把原生Kubernetes中，通过InClusterConfig（即Kubernetes Service）访问kube-apiserver的业务Pod部署到边缘环境中，会出现以下问题：

- 问题一：Pod通过InClusterConfig地址访问kube-apiserver，节点上默认网络规则（iptables/ipvs）将会把请求转发到kube-apiserver的Pod IP，同时云端与边缘位于不同网络平面，边缘是无法访问到云端的Pod IP。所以边缘业务Pod无法通过InClusterConfig访问到kube-apiserver。
- 问题二：在解决问题一后，如果云边网络断开时业务Pod容器出现重启等状况，边缘Pod将无法从kube-apiserver获取到业务配置，这会影响到业务Pod的重启运行。

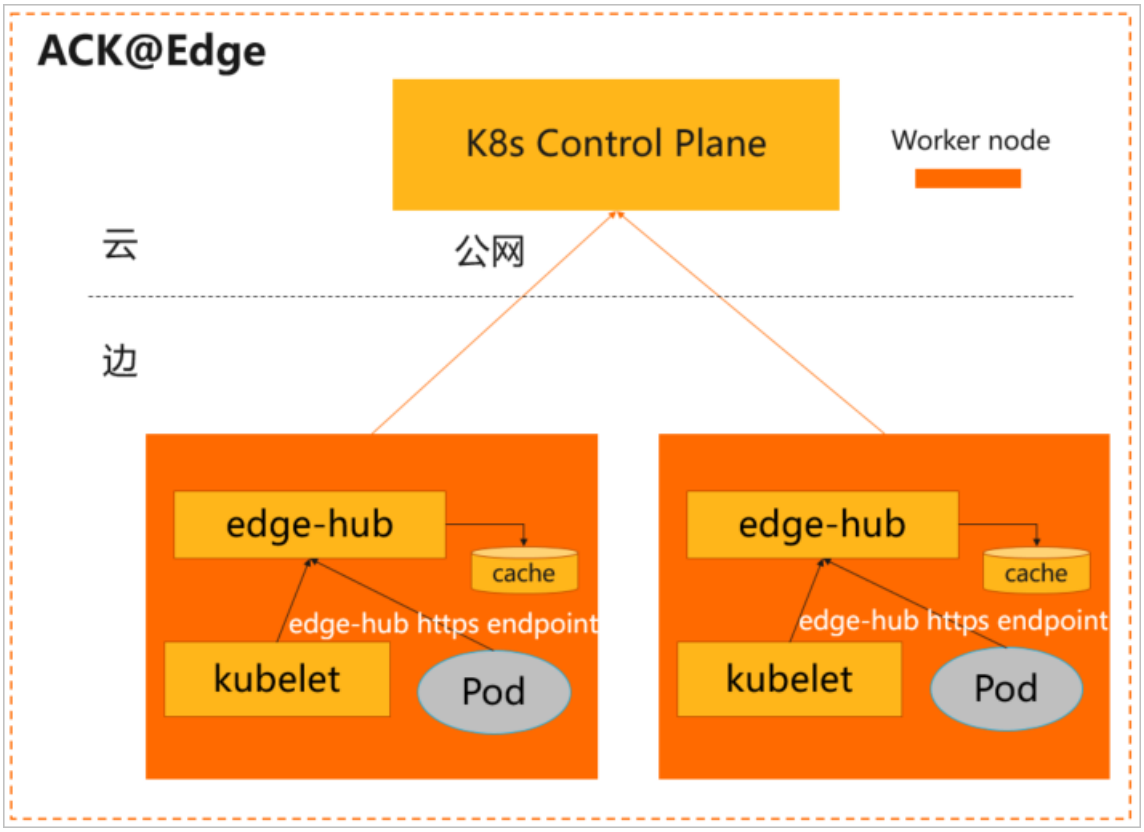
关于如何从Pod访问API的更多信息，请参见从Pod中访问API。

解决方案

通过边缘节点上的edge-hub非侵入的ACK@Edge服务可以解决上述问题，让使用InClusterConfig的业务Pod可以无需修改，直接运行在边缘场景。具体说明如下：

- 在业务Pod无感知状态下，边缘Pod的访问地址（即KUBERNETES_SERVICE_HOST和KUBERNETES_SERVICE_PORT环境变量）会被默认修改为edge-hub的HTTPS Endpoint（即`KUBERNETES_SERVICE_HOST=169.254.2.1`，`KUBERNETES_SERVICE_PORT=10268`），因此业务Pod的InClusterConfig会通过edge-hub来访问kube-apiserver，从而解决上述的问题一。

- 您需要手动开启edge-hub的数据缓存能力，这样即使云边断网，业务Pod重启时可以从edge-hub中获取到本地缓存数据，从而解决上述的问题二。



关于如何开启edge-hub的数据缓存能力，请参见开启edge-hub的数据缓存能力。

开启edge-hub的数据缓存能力

> 说明
> - 因为数据缓存在本地磁盘，所以不推荐为有大量list/watch请求的Pod开启数据缓存。
> - 开启edge-hub的数据缓存能力后，必须重启对应的业务Pod。

1. 获取User-Agent信息。

   User-Agent一般为业务容器的启动命令。

```
apiVersion: v1
kind: Pod
metadata:
 name: edge-app-pod
spec:
 containers:
  - name: "edge-app"
    image: "xxx/edge-app-amd64:1.18.8"
    command:
     - /bin/sh
     - -ec
     - |
       # User-Agent即为启动命令: edge-app。
       /usr/local/bin/edge-app --v=2
```

也可通过edge-hub的日志确认User-Agent信息，找到类似 `{User-Agent} watch {resource}` 的日志，如下所示：

```
I0820 07:50:18.899015     1 util.go:221] edge-app get services: /api/v1/services/xxx with status code 200, spent 21.035061152ms
```

2. 开启edge-hub缓存能力。

   在edge-hub-cfg ConfigMap的cache_agents字段中增加业务Pod请求的User-Agent Header来开启数据缓存能力。

   根据以下YAML示例，开启edge-hub缓存能力：

```
apiVersion: v1
kind: ConfigMap
metadata:
 name: edge-hub-cfg
 namespace: kube-system
data:
 # 缓存边缘edge-app pod(User-Agent header为edge-app)访问kube-apiserver的数据。
 # 开启缓存后，记得重启对应的业务Pod。
 cache_agents: "edge-app"  #添加多个组件请用半角逗号（,）分隔。
```

3. 确认业务Pod的云端返回数据缓存是否生效。

   在业务Pod运行的节点上，查看*/etc/kubernetes/cache/{User-Agent}*目录下是否有对应的数据。

9.边缘Windows容器

9.1. 将Windows节点接入ACK@Edge集群

您可以通过节点池管理集群中的一组节点资源，例如在节点池中统一管理节点的标签和污点。本文介绍在ACK@Edge集群中如何添加已有的Windows节点。

前提条件

已创建边缘Kubernetes集群。具体操作，请参见创建边缘托管版集群。

使用限制

- 边缘集群托管服务每个集群中最多可包含40个节点。如果您需要添加更多节点，请提交工单申请。
- Windows系统目前只支持Windows Server 2019。
- 支持在ACK@Edge集群中同时接入Windows节点和Linux节点。关于如何将Linux节点接入ACK@Edge集群，请参见添加边缘节点。

开启Containers特性

在Windows节点打开Windows PowerShell控制台，执行以下命令开启Containers特性。关于如何打开Windows PowerShell控制台，请参见安装Windows PowerShell。

```
Install-WindowsFeature Containers
```

预期输出：

```
Success   Restart Needed   Exit Code      Feature Result
True      Yes              SuccessRest...  {Containers}
WARNING: You must restart this server to finish the installation process.
```

根据预期输出所示，您需要手动重启Windows服务器。

添加Windows节点

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
4. 在集群管理页左侧导航栏中，选择**节点管理 > 节点**。
5. 在**节点**页面，单击右上角的**添加已有节点**。
6. 在**选择**配置向导页面，选择**添加方式**为**手动添加**，在已有云服务器的列表中，选择所需的ECS云服务器。
7. 单击**下一步**，配置**实例信息**。

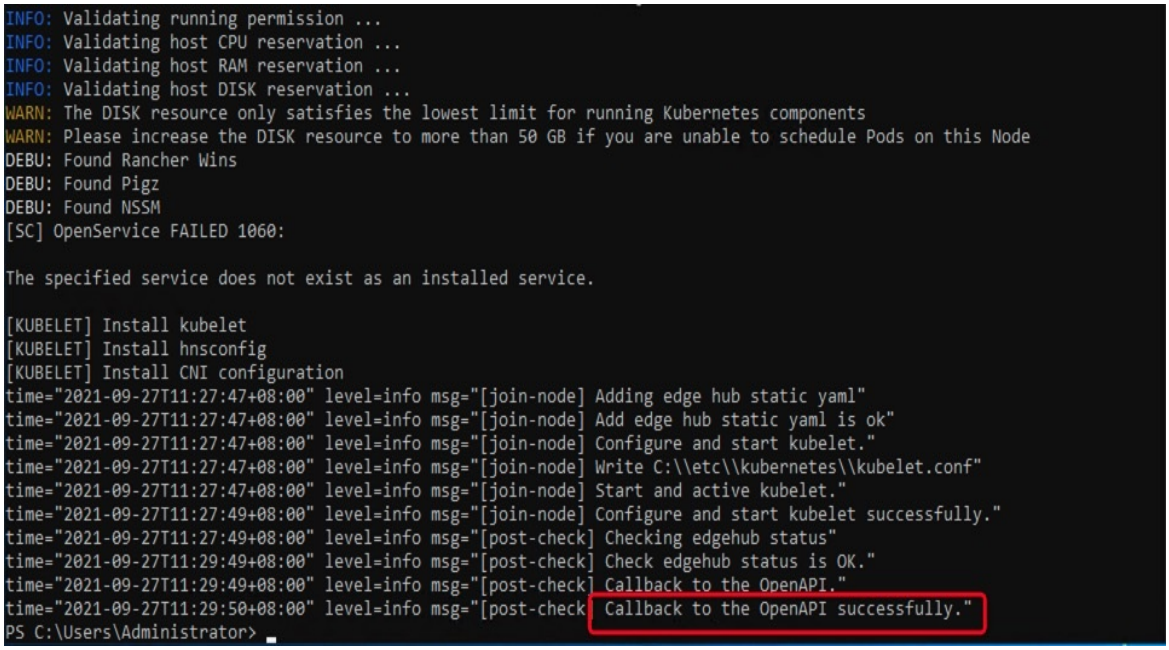
配置项	说明	示例值
集群ID/名称	当前要添加的集群信息，已默认配置。	c593a437a5e754c65876c3f47a8bd**** / testcluster

配置项	说明	示例值
脚本有效时间	**脚本有效时间**默认为1小时，如果您需要长时间使用同一个脚本做批量添加，可以适当增加脚本的有效时间。当**脚本有效时间**配置为0时，表示脚本永久有效。	1
架构	接入节点支持的CPU架构，Windows节点请选择AMD64。	AMD64

配置	接入节点的相关配置。Windows节点请使用示例配置。关于参数的详细描述，请参见参数列表。	{
		"quiet": true,
		"manageRuntime": true,
		"platform": "Windows"
		}

8. 单击**下一步**，在**添加完成**页面，单击**复制**后，在您的Windows节点上，打开PowerShell控制台，执行脚本。

添加Windows节点成功的结果如下图所示。



9. 在**添加完成**页面，单击**完成**。

9.2. 在Windows节点中创建应用

本文介绍如何在Windows节点中通过编排模板创建Web应用，该Web应用通过Deployment类型的工作负载创建Pod资源对象。

前提条件

边缘集群中已经存在Windows节点。关于Windows节点如何接入边缘集群，请参见将Windows节点接入ACK@Edge集群。

背景信息

在容器服务Kubernetes模板编排中，您需要自己定义一个应用运行所需的资源对象，通过标签选择器等机制，将资源对象组合成一个完整的应用。

使用限制

Windows节点上目前只能部署HostNetwork网络模式的工作负载。

操作步骤

1. 登录容器服务管理控制台。
2. 在控制台左侧导航栏中，单击**集群**。
3. 在**集群列表**页面中，单击目标集群名称或者目标集群右侧**操作**列下的**详情**。
4. 在集群管理页左侧导航栏中，选择**工作负载 > 无状态**。
5. 在**无状态**页面，单击右上角的**使用YAML创建资源**。
6. 对模板进行相关配置，完成配置后单击**创建**。
   - **命名空间**：在**创建**页面的顶部选择资源对象所属的命名空间，默认是Default。除了节点、持久化存储卷等底层计算资源以外，大多数资源对象需要作用于命名空间。
   - **示例模板**：阿里云容器服务提供了多种资源类型的Kubernetes YAML示例模板，让您快速部署资源对象。您可以根据Kubernetes YAML编排的格式要求自主编写，来描述您想定义的资源类型。
   - **添加工作负载**：您可通过此功能快速定义一个YAML模板。
   - **使用已有模板**：您可将已有编排模板导入到模板配置页面。
   - **保存模板**：您可以保存设置好的编排模板。

   下面是一个部署在Windows节点上的Web应用的示例编排。通过以下编排模板，即可快速创建一个Web应用。

```
apiVersion: apps/v1
kind: Deployment
metadata:
 labels:
  app: web-windows
 name: web-windows
spec:
 selector:
  matchLabels:
   app: web-windows
 template:
  metadata:
   labels:
    app: web-windows
  spec:
   restartPolicy: Always
   hostNetwork: true
   terminationGracePeriodSeconds: 30
   tolerations:
   - key: os
    value: windows
   affinity:
    nodeAffinity:
     requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
      - matchExpressions:
       - key: kubernetes.io/os
         operator: In
         values:
         - windows
   containers:
    - image: registry.cn-hangzhou.aliyuncs.com/acs/sample-web-windows:v1.0.1
      name: windows
      ports:
      - containerPort: 80
       protocol: TCP
```

7. 单击**创建**后，会提示部署状态信息。应用部署成功后，返回**无状态**页面可以查看创建的Web应用。
8. 在Windows节点打开Windows PowerShell控制台，执行以下命令访问Web应用。

   关于如何打开Windows PowerShell控制台，请参见安装Windows PowerShell。

   ```
   curl 127.0.0.1
   ```

   预期输出：

```
PS C:\Users\Administrator> curl 127.0.0.1


StatusCode        : 200
StatusDescription : OK
Content           : <!DOCTYPE html>
                    <html>
                    <head>
                    <title>Welcome to OpenResty!</title>
                    <style>
                        body {
                            width: 35em;
                            margin: 0 auto;
                            font-family: Tahoma, Verdana, Arial, sans-serif;
                        ...
RawContent        : HTTP/1.1 200 OK
                    Connection: keep-alive
                    Accept-Ranges: bytes
                    Content-Length: 674
                    Content-Type: text/html
                    Date: Thu, 19 Aug 2021 12:46:41 GMT
                    ETag: "5d73ccf0-2a2"
                    Last-Modified: Sat, 07 Sep 2019 ...
Forms             : {}
Headers           : {[Connection, keep-alive], [Accept-Ranges, bytes], [Content-Length, 674], [Content-Type, text/html]...}
Images            : {}
InputFields       : {}
Links             : {@{innerHTML=openresty.org; innerText=openresty.org; outerHTML=<A href="https://openresty.org/">openresty.org</A>; outerText=openresty.org; tagName=A;
                    href=https://openresty.org/}, @{innerHTML=openresty.com; innerText=openresty.com; outerHTML=<A href="https://openresty.com/">openresty.com</A>;
                    outerText=openresty.com; tagName=A; href=https://openresty.com/}}
ParsedHtml        : System.__ComObject
RawContentLength  : 674
```