



# 弹性容器实例 Kubernetes对接ECI

文档版本: 20211111



# 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。

- 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。
- 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。
- 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。	
▲ 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。	會告 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。	大意 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 <i>,</i> 不是 用户必须了解的内容。	⑦ 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 <b>结果确认</b> 页面,单击 <b>确定</b> 。
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 listinstanceid
[] 或者 [alb]	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.对接ECI	05
1.1. 对接概述	05
1.2. ASK使用ECI	07
1.3. ACK使用ECI	08
1.4. 自建Kubernetes集群使用ECI	09
1.5. 线下Kubernetes集群使用ECI	16
1.6. 线下K8s集群部署VK的常见问题	23
1.7. 升级Virtual Kubelet	25
2.使用ECI	27
2.1. 使用概述	27
2.2. 调度Pod到ECI	28
2.3. 配置ECI Profile	32
2.4. ACK安装virtual-kubelet-autoscaler	38
2.5. ACK部署使用elastic-workload	38
2.6. ECI Pod Annotation	44
2.7. ECI自定义事件	58

# 1.对接ECI

# 1.1. 对接概述

ECI为Kubernetes提供了基础的容器Pod运行环境,但业务间的依赖、负载均衡、弹性伸缩、定期调度等能力依然需要Kubernetes来提供。本文为您介绍Kubernetes与ECI如何对接,使用ECI作为Pod的运行资源。

### 对接方式介绍

ECI为Kubernetes提供一种层次化的解决方案:即ECI负责底层Pod容器资源的调度和管理工作,Kubernetes 在ECI之上作为PaaS层来管理业务负载(例如:Deployment、Service、StatefulSet、CronJob)。

ECI在接管Pod容器底层基础设施的管理工作后,Kubernetes不再需要直接负责单个Pod的放置、启动等工作,也不再需要关心底层虚拟机的资源情况,通过ECI即可确保Pod需要的资源随时可用。

基于Kubernetes社区的Virtual Kubelet(简称VK)技术,ECI通过虚拟节点与Kubernetes实现无缝对接,使得Kubernetes集群可以轻松获得极大的弹性能力,而不必受限于集群的节点计算容量。更多信息,请参见Virtual Kubelet。

ECI与Kubernetes的对接分为以下两种方式:

对接方式	说明
完全基于ECI	部署Serverless Kubernetes集群,整个Kubernetes集群运行在ECI上,无需关注底层虚拟机的 运维和容量问题。
混合使用ECI和传统 服务器	ECI作为虚拟节点加入到Kubernetes集群中,可以提升集群的资源利用率和弹性效率,降低运 行成本。

### 完全基于ECI (Serverless Kubernetes)

依托ECI免运维的特性,Kubernetes可以完全依托ECI来构建,即所有Pod均运行在ECI上,Kubernetes仅需要 负责管理业务负载,以保证业务的稳定运行。

如果您正在进行Kubernetes集群的选型,强烈推荐您选用阿里云容器服务Serverless Kubernetes(ASK)。 ASK可以为您提供完全基于ECl运行的Kubernetes集群,为您的在线和离线业务、仿真环境、开发测试环境等 提供免运维、低成本的Kubernetes环境。

关于如何通过ASK使用ECI,请参见ASK使用ECI。

#### 混合使用ECI和传统服务器

如果您已经建立了Kubernetes集群,可以把ECI作为虚拟节点加入到集群中。对于长时间运行的业务负载,您可以将此类负载的弹性流量部分调度至ECI,缩短弹性扩容的时间,减少弹性部分的扩容成本,并尽可能充分利用已有资源。当业务流量下降后,Kubernetes集群可以快速释放部署在ECI上的Pod从而降低您的使用成本。

根据您已有Kubernetes集群的情况,对接ECI包括以下几种方式:

 如果您使用阿里云容器服务Kubernetes (ACK)部署集群,可以通过容器服务管理控制台直接添加虚拟节 点来使用ECI,更多信息,请参见ACK使用ECI。

- 如果您在阿里云ECS上自建了Kubernetes集群,需要部署Virtual Kubelet来使用ECI,更多信息,请参见自 建Kubernetes集群使用ECI。
- 如果您在线下IDC或者其他云上部署Kubernetes集群,需要部署Virtual Kubelet来使用阿里云的ECI,更多 信息,请参见线下Kubernetes集群使用ECI。如需进一步帮助,可咨询与您对接的阿里云解决方案架构师。

#### 管理工具介绍

您可以通过以下方式管理Kubernetes及ECI实例的运行情况:

• 弹性容器实例控制台

您可以通过弹性容器实例控制台查看ECI实例的运行情况。操作步骤如下:

- i. 登录弹性容器实例控制台。
- ii. 在顶部菜单栏左上角处选择地域。
- iii. 在容器组页面,您可以查看该地域下已经创建的ECI实例。
- 容器服务管理控制台

您可以通过容器服务管理控制台来操作ASK集群或ACK集群,并查看ECI实例的运行情况。查看ECI实例的操 作步骤如下:

- i. 登录容器服务管理控制台。
- ii. 在左侧导航栏单击集群。
- iii. 在集群列表中找到想要查看的集群,单击集群ID进入详情页面。
- iv. 在左侧导航栏, 选择工作负载>容器组。

v. 在容器组页面,选择命名空间,您可以查看该命名空间下的ECI实例。

• 阿里云CloudShell

您可以通过阿里云提供的CloudShell来管理Kubernetes集群。具体操作,请参见在CloudShell上通过kubectl 管理Kubernetes集群。

● kubectl客户端

您可以通过kubectl客户端在本地计算机来访问远端的Kubernetes集群。具体操作,请参见通过kubectl连接 Kubernetes集群。

#### 功能限制和说明

基于公有云的安全性和虚拟节点本身带来的限制,ECI目前还不支持Kubernetes中HostPath、DaemonSet等功能,具体如下表所示。

不支持的功能	说明	推荐替代方案
HostPath	挂载本地宿主机文件到容器中	使用emptyDir、云盘或者NAS文件 系统
HostNetwork	将宿主机端口映射到容器上	使用type=LoadBalancer的负载均 衡

不支持的功能	说明	推荐替代方案
DaemonSet	在容器所在宿主机上部署Static Pod	通过sidecar形式在Pod中部署多个 镜像
Privileged权限	容器拥有privileged权限	使用Security Context为Pod添加 Capability
type=NodePort的Service	将宿主机端口映射到容器上	使用type=LoadBalancer的负载均 衡

通过Serverless Kubernetes或者通过Kubernetes扩展虚拟节点的方式使用ECI时,请注意以下信息:

- Serverless Kubernetes集群和Kubernetes集群共享容器镜像仓库,您可以先将容器镜像上传到容器镜像仓库中,便于镜像拉取。推荐您使用阿里云容器镜像服务ACR,并使用专有网络的镜像地址(registry-vpc.xxx)。
- Serverless Kubernet es和Kubernet es扩展虚拟节点均支持Deployment、ReplicaSet、Job、Cronjob、 StatefulSet等常见controller,可以直接运行。
- Serverless Kubernet es和Kubernet es扩展虚拟节点均支持利用PrivateZone实现服务发现,建议您在创建 集群时开启PrivateZone。
- Serverless Kubernetes和Kubernetes+虚拟节点扩展均支持负载均衡,即配置type=LoadBalancer的 Service。

# 1.2. ASK使用ECI

本文介绍如何通过阿里云容器服务Serverless Kubernetes (ASK)使用ECI。ASK已经帮您整合好Kubernetes 集群、阿里云各类服务以及ECI,您可以通过一键创建ASK集群来直接使用ECI。

#### ASK简介

阿里云容器服务Serverless Kubernetes (ASK)是阿里云基于弹性计算架构推出的无服务器Kubernetes容器 服务。您无需购买节点,无需对集群进行节点维护和容量规划,即可直接部署容器应用,并且只需要为应用 配置的CPU和内存资源量进行按需付费。

ASK提供了完善的Kubernetes兼容能力,您可以直接使用 Kubernetes API或者命令行直接管理容器应用。同时,ASK集成阿里云各类服务,可以帮助您简化Kubernetes的开发,专注于应用构建而非基础架构运维。

在ASK集群中,您无需手动部署虚拟节点,可以直接创建Pod。Pod基于ECl运行在安全隔离的容器运行环境中,每个Pod对应一个ECl实例,如下图所示。



更多信息,请参见ASK产品概述。

#### 使用流程

您可以在容器服务管理控制台创建ASK集群,并在集群下直接部署应用。

#### ? 说明

在ASK集群上创建资源生成的Pod,与ECI实例一一对应,您也可以在ECI控制台查看相关信息。创建Pod时,如果想要使用ECI的相关功能,可以添加Annotation来实现。更多信息,请参见ECI Pod Annotation。

#### 相关使用流程如下:

- 1. 创建Serverless Kubernetes集群
- 2. 使用镜像创建应用
- 3. 创建服务
- 4. 查看容器

# 1.3. ACK使用ECI

本文介绍如何把EC通过虚拟节点方式接入到阿里云Kubernetes集群(ACK)。虚拟节点支持动态灵活地按需 创建Pod,免去容量规划的麻烦,可以帮助您降低成本,提升效率。

#### ACK简介

阿里云容器服务Kubernetes (ACK) 是全球首批通过Kubernetes一致性认证的服务平台,提供高性能的容器 应用管理服务。它整合了阿里云虚拟化、存储、网络和安全能力,简化集群的搭建和扩容等工作,让您专注 于容器化的应用的开发与管理。

您可以通过部署虚拟节点来将ECI接入到您的ACK集群中。有了虚拟节点后,当您的ACK集群需要扩容时,无 需规划节点的计算容量,可以直接在虚拟节点下按需创建ECI,ECI与集群中真实节点上的Pod之间网络互通。

在ACK集群中,您需要手动部署虚拟节点,才能在虚拟节点中创建Pod。Pod基于ECl运行在安全隔离的容器运行环境中,每个Pod对应一个ECl实例,如下图所示。



更多信息,请参见ACK产品概述。

#### 使用流程

如果您已经创建了ACK集群,可以在集群上部署虚拟节点来使用ECI。相关使用流程如下:

1. 部署虚拟节点。

您需要安装ack-virtual-node组件来部署虚拟节点。具体操作,请参见部署虚拟节点。

#### ? 说明

如果想要使用ECI相关的新功能,您需要升级ack-virtual-node组件。对于托管版ACK集群,系统将 自动升级。对于专有版ACK集群,您需要手动操作。更多信息,请参见升级Virtual Kubelet。

2. 调度Pod到虚拟节点。

在ACK集群中部署虚拟节点后,您可以将Pod调度到虚拟节点上运行。具体操作,请参见调度Pod到ECI。

⑦ 说明

在ACK集群的虚拟节点上运行的每个Pod,与ECI实例一一对应,您也可以在ECI控制台查看相关信息。创建Pod时,如果想要使用ECI的相关功能,可以添加Annotation来实现。更多信息,请参见ECI Pod Annotation。

更多关于如何使用ECI部署应用的操作,请参见:

- 运行Job任务
- 部署Ingress应用

# 1.4. 自建Kubernetes集群使用ECI

如果您在阿里云ECS上自建了Kubernetes集群,您需要在集群中部署虚拟节点(Virtual Kubelet)来使用 ECI。本文为您介绍在阿里云ECS上自建的集群如何对接ECI,包括将自建集群接入阿里云注册集群并部署虚拟 节点,或者在自建集群中直接部署Virtual Kubelet。

#### 背景信息

基于Kubernetes社区的Virtual Kubelet技术,ECI通过虚拟节点与Kubernetes实现无缝对接,使得 Kubernetes集群可以轻松获得极大的弹性能力,而不必受限于集群的节点计算容量。如果您的Kubernetes 集群是自建的,可以通过以下两种方式来使用ECI:

- 方式一: 将自建集群接入注册集群并部署虚拟节点
- 方式二:在自建集群中直接部署Virtual Kubelet

#### ? 说明

推荐您使用方式一,该方式便于您及时升级Virtual Kubelet的版本(即ack-virtual-node的版本),以使用相关的新功能。关于如何升级VK,请参见升级Virtual Kubelet。

#### 准备工作

操作前,您需要准备好以下信息:

参数	描述	获取方式

参数	描述	获取方式
ECI_ACCESS_KEY、 ECI_SECRET_KEY	AccessKey ID和对应的 AccessKey Secret。虚拟 节点操作ECI的凭证。	请参见 <mark>获取AccessKey</mark> 。
ALIYUN_CLUST ERID	集群ID,集群的唯一标 识。	<ul> <li>如果采用接入注册集群的方式,集群ID将由系统自动生成。</li> <li>如果采用部署Virtual Kubelet的方式,您需要自定义集群ID, ID由字母和数字组成,需确保设置的ID在本阿里云账号下是唯一的。</li> </ul>
ECI_REGION	地域ID。ECI实例将部署在 该地域下。	您可以通过 <mark>弹性容器实例控制台</mark> 或者调 用 <mark>DescribeRegions</mark> 获取ECI支持的地域信息。
ECI_VPC	专有网络VPC ID。ECl实例 将部署在该VPC下。	您可以在 <mark>专有网络控制台的专有网络</mark> 页面创建并查看 VPC。
ECI_VSWITCH	虚拟交换机ID。ECI实例将 部署在该交换机下。	您可以在 <mark>专有网络控制台的交换机</mark> 页面创建并查看交换 机,根据已选的VPC来选择对应的交换机。
ECI_SECURITY_GROUP	安全组ID。ECI实例将部署 在该安全组下。	您可以在 <mark>ECS管理控制台的安全组</mark> 页面创建并查看安全 组,根据已选的VPC来选择对应的安全组。

# 方式一: 接入注册集群并部署虚拟节点

您可以将自建集群通过接入注册集群的方式接入到阿里云容器服务管理平台,然后部署虚拟节点,以便使用 ECI。相关操作步骤如下:

- 1. 登录容器服务管理控制台。
- 2. 在左侧导航栏单击集群。
- 3. 创建注册集群。

i. 在集群列表页面, 单击创建集群。

ii. 选择**注册集群**页签,完成集群相关配置,然后单击**创建集群**。

需要注意的配置项如下表所示。更多信息,请参见创建注册集群。

参数	描述
地域、专有网络、虚拟 交换机	根据准备信息,选择所需的地域、专有网络和虚拟交换机。
API Server访问	默认为API Server创建一个内网SLB实例,您可以根据需要选择SLB实例规格。
绑定EIP	设置是否创建并绑定EIP,可用于建立集群链接。
安全组	自动创建一个安全组,用于划分安全域,统一控制网络流量。
日志服务	设置是否启用阿里云日志服务,可用于快速采集容器日志。
集群删除保护	设置是否启用集群删除保护,可防止通过控制台或API误释放集群。

- 4. 将自建集群接入到注册集群中。
  - i. 在集群列表页面,找到新创建的集群,单击集群名称。
  - ii. 在集群信息页面, 单击连接信息页签。
  - iii. 在自建集群中创建ConfigMap。

根据网络情况,选择**公网**页签或者**私网**页签,复制内容到YAML配置文件中(假设为

agent.yaml),然后在自建集群中执行 kubectl apply -f agent.yaml 命令创建ConfigMap。

iv. 在自建集群中执行以下命令查看接入情况。

kubectl -n kube-system get pod |grep ack-cluster-agent

预期返回:

ack-cluster-agent-5f7d568f6-6fc4k	1/1	Running 0	9s
ack-cluster-agent-5f7d568f6-tf6fp	1/1	Running 0	9s

- 5. 部署虚拟节点。
  - i. 在集群列表页面, 单击目标集群的名称。
  - ii. 在集群管理页的左侧导航栏中,选择运维管理>组件管理。
  - iii. 单击其他页签,找到ack-virtual-node组件,单击安装。

更多关于接入注册集群并部署虚拟节点的信息,请参见自建Kubernetes集群如何使用弹性容器实例ECI。

#### 方式二:直接部署Virtual Kubelet

<↓ 注意

您可以在集群内直接部署Virtual Kubelet(VK)来使用ECI。部署前,您需要先获取最新的VK版本。更多信息,请参见Virtual Kubelet版本记录。

部署Virtual Kubelet的操作步骤如下:

1. 准备部署Virtual Kubelet的配置文件vk.yaml。

配置文件的内容示例如下,您需要根据之前准备的信息,替换配置文件中对应的参数值,并去除注释信息。

Virtual Kubelet版本必须为v2.0.0.121-eff0e01c0-aliyun及以后版本。

```
apiVersion: v1
kind: Secret
metadata:
name: vk-accesskey
namespace: kube-system
type: Opaque
data:
#AK,需要Base64编码后填写到这里
accesskey.id: {{ ECI_ACCESS_KEY Base64 Encoding }}
#SK,需要Base64编码后填写到这里
accesskey.secret: {{ ECI_SECRET_KEY Base64 Encoding }}
---
apiVersion: v1
kind: ConfigMap
metadata:
name: eci-profile
namespace: kube-system
data:
# ECI无法连接到API Server时需要设置为"true"
# 例如:线下自建的K8s集群没有与云上vpc打通时,需要设置为"true";自建集群的API Server与ECI归属于不
同VPC时,需要设置为"true"
enableHybridMode: "false"
#需要使用K8s的Cluster IP特性时设置为"true"
enableClusterIp: "true"
#是否开启PrivateZone,需要开启时设置为"true"
enablePrivateZone: "false"
#资源组ID,需要指定时填写,不需要指定时设置为""即可
resourceGroupId: ""
#安全组ID,必须指定
securityGroupId: "sg-2ze********
#调度选择器,用于设置哪些Pod将自动调度到ECI上,具体请参见ECI Profile
selectors: ""
# 交换机ID,必须指定
vSwitchIds: "vsw-2zeq*********
# VPC ID, 必须指定
vpcld: "vpc-2ze0z**********
apiVersion: v1
kind: Service
metadata:
بعامطوا
```

100013. role: webhook name: vk-webhook namespace: kube-system spec: ports: - port: 443 targetPort: 443 selector: app: virtual-kubelet --apiVersion: v1 kind: ServiceAccount metadata: name: virtual-kubelet namespace: kube-system kind: ClusterRoleBinding apiVersion: rbac.authorization.k8s.io/v1beta1 metadata: name: virtual-kubelet subjects: - kind: ServiceAccount name: virtual-kubelet namespace: kube-system roleRef: kind: ClusterRole name: cluster-admin apiGroup: rbac.authorization.k8s.io ---apiVersion: apps/v1 kind: Deployment metadata: name: ack-virtual-node-controller namespace: kube-system labels: app: virtual-kubelet spec: replicas: 1 selector: matchLabels: app: virtual-kubelet template: metadata: labels: app: virtual-kubelet spec: affinity: nodeAffinity: requiredDuringSchedulingIgnoredDuringExecution: nodeSelectorTerms: - matchExpressions: - key: type operator: NotIn values:

```
- virtual-kubelet
podAntiAffinity:
 preferredDuringSchedulingIgnoredDuringExecution:
 - podAffinityTerm:
  labelSelector:
   matchExpressions:
   - key: app
    operator: In
    values:
    - virtual-node-eci
  topologyKey: kubernetes.io/hostname
  weight: 100
containers:
- name: vk
#按照实际情况替换region-id,在VK版本记录中查找最新的VK版本替换vk-tag
#注意:部署后应该关注vk版本发布,及时升级vk以获得阿里云的持续支持
image: registry-vpc.{{ region-id }}.aliyuncs.com/acs/virtual-nodes-eci:{{ vk-tag }}
imagePullPolicy: IfNotPresent
args:
---provider
- alibabacloud
- -- nodename
- virtual-kubelet
resources:
 requests:
  memory: 30M
  cpu: 100m
env:
#必须指定集群类型,基于阿里云ECS自建集群请设置为CLOUD,线下自建集群请设置为IDC
- name: CLUSTER_TYPE
 value: {{ cluster-type }}
- name: WEBHOOK
 value: "true"
- name: VKUBELET_TAINT_KEY
 value: "virtual-kubelet.io/provider"
- name: VKUBELET_TAINT_VALUE
 value: "alibabacloud"
- name: VKUBELET_TAINT_EFFECT
 value: "NoSchedule"
 #必须填写region-id,为实际的地域ID,如cn-beijing
- name: ECI_REGION
 value: {{ region-id }}
- name: ECI_QUOTA_CPU
 value: "1000000"
- name: ECI_QUOTA_MEMORY
 value: 6400Ti
- name: ECI_QUOTA_POD
 value: "10000"
- name: ECI_KUBE_PROXY
 value: "true"
- name: ECI_ACCESS_KEY
 valueFrom:
  secretKeyRef:
  name: vk-accesskey
```

key: accesskey.id - name: ECI\_SECRET\_KEY valueFrom: secretKeyRef: name: vk-accesskey key: accesskey.secret name: ALIYUN\_CLUSTERID # 自建K8s集群的集群ID,必须设置,账号下每个K8s集群的ID必须唯一 value: {{ cluster-id }} - name: KUBELET\_PORT value: "10250" - name: VKUBELET\_POD\_IP valueFrom: fieldRef: apiVersion: v1 fieldPath: status.podIP terminationGracePeriodSeconds: 60 serviceAccountName: virtual-kubelet

2. 部署Virtual Kubelet。

kubectl apply -f vk.yaml

3. 查看部署情况。

kubectl get deploy/virtual-node-controller -n kube-system

部署完成的返回示例如下:

NAME READY UP-TO-DATE AVAILABLE AGE virtual-node-controller 1/1 1 1 161m

4. 查看部署Virtual Kubelet后的节点信息。

kubectl get node -o wide

部署Virtual Kubelet后生成的虚拟节点名称为virtual-kubelet。如果能查询到virtual-kubelet节点,且状态为Ready,则表示部署成功。返回示例如下:

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-II	OS-IMAGE	KERNEL-
VERSION	CONTA	INER-RUI	NTIME	E				
k8s-master01	. Ready	<none></none>	50d	v1.14.2	192.168.*.*	<none></none>	CentOS Linux 7 (Core	3.10.0
-957.21.3.el7.	x86_64 d	ocker://1	8.9.6					
k8s-master02	Ready	<none></none>	50d	v1.14.2	192.168.*.*	<none></none>	CentOS Linux 7 (Core	3.10.0
-957.21.3.el7.	x86_64 d	ocker://1	8.9.6					
k8s-master03	Ready	<none></none>	50d	v1.14.2	192.168.*.*	<none></none>	CentOS Linux 7 (Core	3.10.0
-957.21.3.el7.	x86_64 d	ocker://1	8.9.6					
virtual-kubel	et Ready	/ agent	82n	n v1.11.2	172.30.*.*	<none></none>	<unknown> &lt;۱</unknown>	Inknow
n>	<unknow< td=""><td>/n&gt;</td><td></td><td></td><td></td><td></td><td></td><td></td></unknow<>	/n>						

#### 相关文档

接入ECI后,您可能需要调度Pod到ECI上运行,或者使用ECI相关功能。关于如何使用ECI,请参见使用概述。

# 1.5. 线下Kubernetes集群使用ECI

如果您在本地机房或者其他线下环境自建了Kubernet es集群,您需要在集群中部署虚拟节点(Virtual Kubelet)来使用ECI。本文为您介绍线下自建的集群如何对接ECI,包括将自建集群接入阿里云注册集群并部署虚拟节点,或者在自建集群中直接部署Virtual Kubelet。

#### 前提条件

已开通阿里云账号。具体操作,请参见阿里云账号注册流程。

#### 背景信息

基于Kubernetes社区的Virtual Kubelet技术,ECI通过虚拟节点与Kubernetes实现无缝对接,使得 Kubernetes集群可以轻松获得极大的弹性能力,而不必受限于集群的节点计算容量。如果您的Kubernetes 集群是自建的,可以通过以下两种方式来使用ECI:

- 方式一: 将自建集群接入注册集群并部署虚拟节点
- 方式二: 在自建集群中直接部署Virtual Kubelet

#### ? 说明

推荐您使用方式一,该方式便于您及时升级Virtual Kubelet的版本(即ack-virtual-node的版本),以使用相关的新功能。关于如何升级VK,请参见升级Virtual Kubelet。

#### 使用限制

在该混合云场景下,部分功能无法支持或有条件支持,不支持的功能如下:

- Annotations更新
- Labels更新
- Spec.ActiveDeadlineSeconds更新
- ConfigMap、Secret更新
- kubectl logs -f命令(支持logs查询,不支持follow模式)
- kubectl attach命令
- kubectl port-forward命令
- 部分DownwardAPI参数无法引用(如: status.hostIP)
- Clust erIP

需要通过阿里云高速通道或者智能接入网关SAG打通线下和云上网络,更多信息,请参见<mark>高速通道</mark>或者智能 接入网关SAG。

#### 准备工作

操作前,您需要准备好以下信息:

参数	描述	获取方式
ECI_ACCESS_KEY、 ECI_SECRET_KEY	AccessKey ID和对应的 AccessKey Secret。虚拟 节点操作ECI的凭证。	请参见 <mark>获取AccessKey</mark> 。
ALIYUN_CLUST ERID	集群ID,集群的唯一标 识。	<ul> <li>如果采用接入注册集群的方式,集群ID将由系统自动生成。</li> <li>如果采用部署Virtual Kubelet的方式,您需要自定义集群ID,ID由字母和数字组成,需确保设置的ID在本阿里云账号下是唯一的。</li> </ul>
ECI_REGION	地域ID。ECI实例将部署在 该地域下。	您可以通过 <mark>弹性容器实例控制台</mark> 或者调 用 <mark>DescribeRegions</mark> 获取ECI支持的地域信息。
ECI_VPC	专有网络VPC ID。ECI实例 将部署在该VPC下。	您可以在 <mark>专有网络控制台的专有网络</mark> 页面创建并查看 VPC。
ECI_VSWITCH	虚拟交换机ID。ECI实例将 部署在该交换机下。	您可以在 <mark>专有网络控制台的交换机</mark> 页面创建并查看交换 机,根据已选的VPC来选择对应的交换机。
ECI_SECURITY_GROUP	安全组ID。ECI实例将部署 在该安全组下。	您可以在ECS管理控制台的安全组页面创建并查看安全 组,根据已选的VPC来选择对应的安全组。

# 方式一: 接入注册集群并部署虚拟节点

您可以将自建集群通过接入注册集群的方式接入到阿里云容器服务管理平台,然后部署虚拟节点,以便使用 ECI。相关操作步骤如下:

- 1. 登录容器服务管理控制台。
- 2. 在左侧导航栏单击集群。
- 3. 创建注册集群。
  - i. 在集群列表页面, 单击创建集群。

ii. 选择**注册集群**页签,完成集群相关配置,然后单击**创建集群**。

需要注意的配置项如下表所示。更多信息,请参见创建注册集群。

参数	描述
地域、专有网络、虚拟 交换机	根据准备信息,选择所需的地域、专有网络和虚拟交换机。
API Server访问	默认为API Server创建一个内网SLB实例,您可以根据需要选择SLB实例规格。
绑定EIP	设置是否创建并绑定EIP,可用于建立集群链接。
安全组	自动创建一个安全组,用于划分安全域,统一控制网络流量。
日志服务	设置是否启用阿里云日志服务,可用于快速采集容器日志。
集群删除保护	设置是否启用集群删除保护,可防止通过控制台或API误释放集群。

- 4. 将自建集群接入到注册集群中。
  - i. 在**集群列表**页面,找到新创建的集群,单击集群名称。
  - ii. 在集群信息页面, 单击连接信息页签。
  - iii. 在自建集群中创建ConfigMap。

根据网络情况,选择**公网**页签或者**私网**页签,复制内容到YAML配置文件中(假设为

agent.yaml),然后在自建集群中执行 kubectl apply -f agent.yaml 命令创建ConfigMap。

iv. 在自建集群中执行以下命令查看接入情况。

kubectl -n kube-system get pod |grep ack-cluster-agent

预期返回:

ack-cluster-agent-5f7d568f6-6fc4k	1/1	Running 0	9s
ack-cluster-agent-5f7d568f6-tf6fp	1/1	Running 0	9s

- 5. 部署虚拟节点。
  - i. 在集群列表页面, 单击目标集群的名称。
  - ii. 在集群管理页的左侧导航栏中,选择运维管理>组件管理。
  - iii. 单击其他页签,找到ack-virtual-node组件,单击安装。

更多关于接入注册集群并部署虚拟节点的信息,请参见自建Kubernetes集群如何使用弹性容器实例ECI。

#### 方式二:直接部署Virtual Kubelet

您可以在集群内直接部署Virtual Kubelet(VK)来使用ECI。部署前,您需要先获取最新的VK版本。更多信息,请参见Virtual Kubelet版本记录。

部署Virtual Kubelet的操作步骤如下:

<↓ 注意

1. 准备部署Virtual Kubelet的配置文件vk.yaml。

配置文件的内容示例如下,您需要根据之前准备的信息,替换配置文件中对应的参数值,并去除注释信息。

```
Virtual Kubelet版本必须为v2.0.0.121-eff0e01c0-aliyun及以后版本。
```

```
apiVersion: v1
kind: Secret
metadata:
name: vk-accesskey
namespace: kube-system
type: Opaque
data:
#AK,需要Base64编码后填写到这里
accesskey.id: {{ ECI_ACCESS_KEY Base64 Encoding }}
#SK,需要Base64编码后填写到这里
accesskey.secret: {{ ECI_SECRET_KEY Base64 Encoding }}
---
apiVersion: v1
kind: ConfigMap
metadata:
name: eci-profile
namespace: kube-system
data:
# ECI无法连接到API Server时需要设置为"true"
# 例如:线下自建的K8s集群没有与云上vpc打通时,需要设置为"true";自建集群的API Server与ECI归属于不
同VPC时,需要设置为"true"
enableHybridMode: "false"
#需要使用K8s的Cluster IP特性时设置为"true"
enableClusterIp: "true"
#是否开启PrivateZone,需要开启时设置为"true"
enablePrivateZone: "false"
#资源组ID,需要指定时填写,不需要指定时设置为""即可
resourceGroupId: ""
#安全组ID,必须指定
securityGroupId: "sg-2ze********
#调度选择器,用于设置哪些Pod将自动调度到ECI上,具体请参见ECI Profile
selectors: ""
# 交换机ID,必须指定
vSwitchIds: "vsw-2zeq*********
# VPC ID, 必须指定
vpcld: "vpc-2ze0z**********
apiVersion: v1
kind: Service
metadata:
lahole
```

100013.

role: webhook name: vk-webhook namespace: kube-system spec: ports: - port: 443 targetPort: 443 selector: app: virtual-kubelet --apiVersion: v1 kind: ServiceAccount metadata: name: virtual-kubelet namespace: kube-system kind: ClusterRoleBinding apiVersion: rbac.authorization.k8s.io/v1beta1 metadata: name: virtual-kubelet subjects: - kind: ServiceAccount name: virtual-kubelet namespace: kube-system roleRef: kind: ClusterRole name: cluster-admin apiGroup: rbac.authorization.k8s.io --apiVersion: apps/v1 kind: Deployment metadata: name: ack-virtual-node-controller namespace: kube-system labels: app: virtual-kubelet spec: replicas: 1 selector: matchLabels: app: virtual-kubelet template: metadata: labels: app: virtual-kubelet spec: affinity: nodeAffinity: requiredDuringSchedulingIgnoredDuringExecution: nodeSelectorTerms: - matchExpressions: - key: type operator: NotIn values:

```
- virtual-kubelet
podAntiAffinity:
 preferredDuringSchedulingIgnoredDuringExecution:
 - podAffinityTerm:
  labelSelector:
   matchExpressions:
   - key: app
    operator: In
    values:
    - virtual-node-eci
  topologyKey: kubernetes.io/hostname
  weight: 100
containers:
- name: vk
#按照实际情况替换region-id,在VK版本记录中查找最新的VK版本替换vk-tag
#注意:部署后应该关注vk版本发布,及时升级vk以获得阿里云的持续支持
image: registry-vpc.{{ region-id }}.aliyuncs.com/acs/virtual-nodes-eci:{{ vk-tag }}
imagePullPolicy: IfNotPresent
args:
---provider
- alibabacloud
- -- nodename
- virtual-kubelet
resources:
 requests:
  memory: 30M
  cpu: 100m
env:
#必须指定集群类型,基于阿里云ECS自建集群请设置为CLOUD,线下自建集群请设置为IDC
- name: CLUSTER_TYPE
 value: {{ cluster-type }}
- name: WEBHOOK
 value: "true"
- name: VKUBELET_TAINT_KEY
 value: "virtual-kubelet.io/provider"
- name: VKUBELET_TAINT_VALUE
 value: "alibabacloud"
- name: VKUBELET_TAINT_EFFECT
 value: "NoSchedule"
 #必须填写region-id,为实际的地域ID,如cn-beijing
- name: ECI_REGION
 value: {{ region-id }}
- name: ECI_QUOTA_CPU
 value: "1000000"
- name: ECI_QUOTA_MEMORY
 value: 6400Ti
- name: ECI_QUOTA_POD
 value: "10000"
- name: ECI_KUBE_PROXY
 value: "true"
- name: ECI_ACCESS_KEY
 valueFrom:
  secretKeyRef:
  name: vk-accesskey
```

key: accesskey.id - name: ECI\_SECRET\_KEY valueFrom: secretKeyRef: name: vk-accesskey key: accesskey.secret - name: ALIYUN\_CLUSTERID # 自建K8s集群的集群ID,必须设置,账号下每个K8s集群的ID必须唯一 value: {{ cluster-id }} - name: KUBELET\_PORT value: "10250" - name: VKUBELET\_POD\_IP valueFrom: fieldRef: apiVersion: v1 fieldPath: status.podIP terminationGracePeriodSeconds: 60 serviceAccountName: virtual-kubelet

2. 部署Virtual Kubelet。

kubectl apply -f vk.yaml

3. 查看部署情况。

kubectl get deploy/virtual-node-controller -n kube-system

部署完成的返回示例如下:

NAME READY UP-TO-DATE AVAILABLE AGE virtual-node-controller 1/1 1 1 161m

4. 查看部署Virtual Kubelet后的节点信息。

kubectl get node -o wide

部署Virtual Kubelet后生成的虚拟节点名称为virtual-kubelet。如果能查询到virtual-kubelet节点,且状 态为Ready,则表示部署成功。返回示例如下:

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-II	OS-IMAGE	KERNEL-
VERSION	CONTA	INER-RUI	NTIME	E				
k8s-master01	. Ready	<none></none>	50d	v1.14.2	192.168.*.*	<none></none>	CentOS Linux 7 (Core	3.10.0
-957.21.3.el7.	x86_64 d	ocker://1	8.9.6					
k8s-master02	Ready	<none></none>	50d	v1.14.2	192.168.*.*	<none></none>	CentOS Linux 7 (Core	3.10.0
-957.21.3.el7.	x86_64 d	ocker://1	8.9.6					
k8s-master03	Ready	<none></none>	50d	v1.14.2	192.168.*.*	<none></none>	CentOS Linux 7 (Core	3.10.0
-957.21.3.el7.x86_64 docker://18.9.6								
virtual-kubel	et Ready	/ agent	82n	n v1.11.2	172.30.*.*	<none></none>	<unknown> &lt;۱</unknown>	Inknow
n>	<unknow< td=""><td>/n&gt;</td><td></td><td></td><td></td><td></td><td></td><td></td></unknow<>	/n>						

#### 相关文档

接入ECI后,您可能需要调度Pod到ECI上运行,或者使用ECI相关功能。关于如何使用ECI,请参见使用概述。

# 1.6. 线下K8s集群部署VK的常见问题

如果您在线下IDC自建了Kubernet es集群,您需要在K8s集群中需要部署VK(Virt ual Kubelet),即部署虚拟 节点后才能使用ECI。本文介绍线下K8s集群部署VK来使用ECI的常见问题。

- 网络相关问题
  - 。 云上服务如何访问线下IDC的Pod IP?
  - 。 线下IDC服务如何访问云上的Pod IP?
  - 内网域名无法解析怎么办?
  - 。 线下IDC服务如何访问云上服务?
- 镜像拉取相关问题
  - 使用自建的容器镜像仓库无法拉取镜像,怎么办?
- Pod调度相关问题
  - o 如何调度Pod到VK?
  - ◎ 为什么DaemonSet Pod调度到VK后一直是Pending状态?
  - 使用配置Pod Label的方式调度Pod到VK,为什么调度失败?

#### 云上服务如何访问线下IDC的Pod IP?

如果您使用高速通道(专线)来打通线下和云上网络,则线下IDC和云上服务之间可以通过边界路由协议BGP 相互学习路由规则,IDC可以将Pod IP的路由信息通过BGP广播给云上服务,实现云上服务可以访问线下IDC的 Pod IP。具体操作,请参见配置BGP。

#### 线下IDC服务如何访问云上的Pod IP?

如果您使用高速通道(专线)来打通线下和云上网络,配置BGP后,线下IDC和云上服务之间可以相互学习路 由规则,云上的Pod IP的路由信息可以部署CCM实现自动同步。更多关于CCM的信息,请参见Cloud Controller Manager。

自建K8s集群部署CCM后,可以将K8s的Pod IP路由信息同步到VPC路由表。部署CCM时,请注意以下事项:

• 需要将K8s集群节点的providerID修改为 <region-id>.<ecs-id> 的格式,例如

cn-shanghai.i-ankb8zjh2nzchf\*\*\*\*\*\*\* 。

• 确保Pod IP与节点的Pod CIDR相匹配,即该节点上的Pod IP地址均在该节点定义的Pod CIDR网段内。例

如: Calico IPAM需要配置为 host-local 才能使Pod IP与节点的Pod CIDR相匹配。

Pod CIDR网段可以在节点的spec信息中查看:

spec: podCIDR: 172.23.XX.0/26 podCIDRs: - 172.23.XX.0/26 providerID: cn-shanghai.i-ankb8zjh2nzchfxxxxxxx

#### 内网域名无法解析怎么办?

#### 问题描述

> 文档版本: 20211111

云上和线下服务由于无法解析内网域名,导致无法进行业务调用,包括:

- 云上服务无法解析IDC内网域名
- IDC服务无法解析云上PrivateZone域名

#### 解决方案

线下IDC和阿里云VPC分别属于两套网络环境,如果部署在线下IDC和云上的服务都需要通过DNS解析进行业务间调用,您可以配置PrivateZone云解析实现云上和线下服务通过内网域名方式实现相互访问。具体操作,请参见PrivateZone&VPN网关联动实现云上访问云下资源。

#### 线下IDC服务如何访问云上服务?

#### 问题描述

线下IDC服务通过专线无法访问阿里云服务,例如IDC通过专线无法访问云数据库RDS、对象存储OSS、日志服 务SLS等。

#### 解决方案

您可以通过以下两种方式解决问题,推荐您使用方案一。

● 方案一

在云上配置云服务域名,边界路由器VBR将通过边界路由协议BGP发布路由给线下IDC。具体操作,请参 见<mark>访问云服务</mark>。

• 方案二:配置IDC添加静态路由,将100.64.0.0/10路由到专线。

#### 使用自建的容器镜像仓库无法拉取镜像,怎么办?

#### 问题描述

使用自建的容器镜像仓库无法拉取镜像,报错如下:

```
Failed to pull image "harbor.example.com/default/nginx:latest": rpc error: code = Unknown
desc = failed to pull and unpack image "hat a ample.com/default/nginx:latest": failed
to resolve reference "harbor ple.com/default/nginx:latest": failed to do request: Head
"https://harbor ple.com/default/nginx/manifests/latest": x509: certificate is valid
for ingress.local, not harbor-beta.cm.service.163.org
```

#### 解决方案

上述问题可能是因为镜像仓库使用的自签发的证书,拉取镜像时无法通过证书认证导致。在创建Pod时,您可以添加以下Annotation来跳过证书认证:

"k8s.aliyun.com/insecure-registry": "<host-name>"

例如:假设私有镜像仓库中的Nginx镜像的链接为 test.example.com/test/nginx:apline ,则可以添加

"k8s.aliyun.com/insecure-registry": "test.example.com" 的Annotation来跳过证书认证。

#### 如何调度Pod到VK?

您可以根据业务需要选择合适的方式将Pod调度到VK,即在ECI上运行Pod。常用方式如下:

● 手动调度Pod

通过配置Pod Label、配置Namespace Label和配置ECI弹性调度等方式,可以手动将Pod调度到ECI上运行。更多信息,请参见调度Pod到ECI。

● 自动调度Pod

ECI Profile提供ECI Scheduler能力,基于Mut at ing Webhook机制实现了一种新的调度机制,即在ECI Profile中,您可以声明需要匹配的Namespace或者Pod的Label,对于Label能够匹配上的Pod,将被自动 调度到ECI。更多信息,请参见配置ECI Profile。

#### 为什么DaemonSet Pod调度到VK后一直是Pending状态?

由于VK对应的是虚拟节点,并不支持DaemonSet,因此在创建DaemonSet时,您需要配置反亲和性调度策略,禁止DaemonSet Pod调度到VK(即虚拟节点)上。配置示例如下:

spec: affinity: nodeAffinity: requiredDuringSchedulingIgnoredDuringExecution: nodeSelectorTerms: - matchExpressions: - key: type operator: NotIn values: - virtual-kubelet

#### 使用配置Pod Label的方式调度Pod到VK,为什么调度失败?

使用配置Pod Label的方式调度Pod到VK时,需要确保K8s版本为1.16及以上版本。

// MutatingAdmission not support ObjectSelector if Kubernetes version is less than 1.16

# 1.7. 升级Virtual Kubelet

基于Kubernetes社区的Virtual Kubelet技术, EC通过部署Virtual Kubelet的方式,以虚拟节点与Kubernetes 实现无缝对接。如果您想要使用ECI相关的新功能,需要及时更新Virtual Kubelet的版本。本文介绍如何升级 Virtual Kubelet。

#### 背景信息

对于不同的Kubernetes集群, Virtual Kubelet (VK)的升级支持情况如下:

- 阿里云ASK集群:自动升级VK。
- 阿里云ACK集群:分为托管版和专有版。托管版自动升级VK,专有版需要您手动升级VK。
- 自建集群:在ECS上或者线下自建的集群,需要您手动升级VK。

根据您集群的类型和VK部署方式,手动升级VK的方式分为以下两种:

• 在容器服务管理控制台直接升级组件

专有版ACK集群和通过接入注册集群方式接入的自建集群,可以在容器服务管理控制台进行操作,直接升级ack-virtual-node组件。

• 通过Kubelet命令修改VK的配置文件

通过直接部署VK方式接入的自建集群,可以修改VK对应的YAML配置文件,将VK镜像tag更新到需要的版本。关于VK的版本,请参见Virtual Kubelet版本记录。

#### 在容器服务管理控制台直接升级组件

- 1. 登录容器服务管理控制台。
- 2. 在左侧导航栏单击集群。
- 3. 在集群列表页面,单击目标集群的名称。
- 4. 在集群管理页的左侧导航栏中,选择运维管理>组件管理。
- 5. 找到ack-virtual-node组件,单击升级。

#### 通过Kubelet命令修改VK的配置文件

- 1. 通过Kubelet连接集群。
- 2. 执行以下命令修改VK的YAML配置文件。

kubectl edit deployment -n kube-system virtual-node-controller

示例如下:

```
apiVersion: apps/v1
kind: Deployment
metadata:
name: virtual-node-controller
namespace: kube-system
labels:
 app: virtual-node-controller
spec:
replicas: 1
selector:
 matchLabels:
  app: virtual-node-controller
template:
 metadata:
  labels:
   app: virtual-node-controller
 spec:
  containers:
  - name: virtual-node-controller
   image: registry.cn-beijing.aliyuncs.com/acs/virtual-nodes-eci:v2.*****-aliyun #将镜像tag改为最新版
本
env:
   - name: WEBHOOK
    value: "true"
```

# 2.使用ECI

# 2.1. 使用概述

通过Virtual Kubelet将ECI接入Kubernetes集群后,您可以将Pod调度到ECI上运行,并配置Annotation来使用ECI的相关功能。

### 调度Pod到ECI

对于Serverless Kubernet es集群,整个集群运行在ECI上,无需进行调度。对于混合使用ECI和普通节点的 Kubernet es集群,您可以根据需要将Pod调度到ECI上运行。主要场景如下:

● 手动调度Pod到ECI

通过配置Pod Label、配置Namespace Label和配置ECI弹性调度等方式,可以手动将Pod调度到ECI上运行。更多信息,请参见<mark>调度Pod到ECI</mark>。

● 自动调度Pod到ECI

ECI Profile提供ECI Scheduler能力,基于Mutating Webhook机制实现了一种新的调度机制,即在ECI Profile中,您可以声明需要匹配的Namespace或者Pod的Label,对于Label能够匹配上的Pod,将被自动 调度到ECI。更多信息,请参见配置ECI Profile。

● 自动将调度失败的Pod调度到ECI

在ACK集群中部署virtual-kubelet-autoscaler插件后,当ACK集群中Work Node的资源不足时,可以自动 将因为资源不足而调度失败的Pod二次调度到ECI上运行。更多信息,请参见ACK安装virtual-kubeletautoscaler。

● 自动将弹性负载的Pod调度到ECI

在ACK集群中部署elastic-workload插件后,可以控制一个负载(例如: Deployment、StatefulSet)在 Pod级别的细粒度分配,将弹性部分的Pod调度到ECI上运行,实现更为精细的调度策略。更多信息,请参 见ACK部署使用elastic-workload。

### 使用ECI功能

在Kubernetes集群中创建Pod到ECI时,为充分使用ECI提供的功能,在不改变Kubernetes语义的前提下,您可以根据需求为Pod添加Annotation。Annotation需添加到Pod级别的metadata中,常用的Annotation如下:

#### ? 说明

您可以在创建Pod时手动添加Annotation,也可以配置ECI Profile,通过ECI Profile的ECI Effect能力,实现自动追加Annotation到Label能够匹配上的Pod。关于如何配置ECI Effect,请参见配置ECI Profile。

参数	示例值	描述
k8s.aliyun.com/eci- vswitch	vsw- bp1xpiowfm5vo8o3c*** *	指定交换机ID,支持指定多个交换机实现多可用区功能。

参数	示例值	描述
k8s.aliyun.com/eci-use- specs	2-4Gi,4-8Gi,ecs.c6.xlarge	指定ECI实例规格,支持指定多规格,包括指定vCPU和内 存,或者ECS规格。
k8s.aliyun.com/eci- image-cache	"true"	设置是否自动匹配镜像缓存。
k8s.aliyun.com/eci- with-eip	"true"	设置是否自动创建并绑定EIP。

关于支持的Annotation列表以及配置示例,请参见ECI Pod Annotation。

? 说明

使用ECI功能的过程中,如果出现异常,您可以查看事件信息。更多信息,请参见ECI自定义事件。

# 2.2. 调度Pod到ECI

对于Serverless Kubernetes集群,整个集群运行在ECI上,无需进行调度。对于ACK集群或者自建的K8s集群,即混合使用ECI和普通节点的模式下,您可以根据需要将Pod调度到虚拟节点上的ECI上运行。本文介绍如何将Pod调度到ECI上运行。

#### 调度方式概述

在混合使用ECI和普通节点的模式下,一般可以通过以下几种方式将Pod调度到ECI:

? 说明

下述方式均需要对存量资源做一定的修改,无法做到零侵入,建议您配置ECI Profile,通过ECI Profile的 ECI Scheduler能力,在eci-profile配置文件中声明需要匹配的Pod或者Namespace的Label,对于Label 能够匹配上的Pod,将被自动调度到ECI。更多信息,请参见配置ECI Profile。

• 配置Pod Label

如果您有个别Pod需要调度到ECI上运行,可以直接为Pod添加特定的Label,则该Pod将运行在虚拟节点的ECI实例上。

• 配置Namespace Label

如果您有一类Pod需要调度到ECI上运行,可以创建一个Namespace并添加特定的Label,则该Namespace下的所有Pod将运行在虚拟节点的ECI实例上。

● 配置ECI弹性调度

ECI弹性调度是阿里云提供的一种弹性调度策略,在部署服务时,您可以添加Annotations来声明只使用普通节点的资源或者虚拟节点的ECI资源,或者在普通节点的资源不足时自动使用ECI资源,以满足不同场景下对弹性资源的不同需求。

● (不推荐)配置nodeSelector和tolerations、指定nodeName等其它方式

当集群中存在虚拟节点时,您可以通过配置nodeSelector和tolerations,或者指定nodeName的方式,将 Pod调度到虚拟节点上,Virtual Kubelet将会创建出相应的ECI实例来运行Pod。

### 方式一:配置Pod Label

您可以为Pod添加 alibabacloud.com/eci=true 的Label,将Pod调度到ECI上运行。

? 说明

```
仍支持添加 eci=true 的Label, 但不推荐。
```

#### 配置示例如下:

1. 创建Pod的YAML配置文件。

vim test-pod.yaml

test-pod.yaml的内容示例如下:

- apiVersion: v1 kind: Pod metadata: name: nginx1 labels: alibabacloud.com/eci: "true" #添加特定Label spec: containers: - image: nginx imagePullPolicy: Always name: nginx
- 2. 创建Pod。

kubectl apply -f test-pod.yaml

### 方式二:配置Namespace Label

您可以创建一个Namespace并添加 alibabacloud.com/eci=true 的Label,则该Namespace下的所有Pod将调 度到ECI上运行。

⑦ 说明

仍支持添加 virtual-node-affinity-injection=enabled 的Label,但不推荐。

#### 配置示例如下:

1. 创建Namespace。

kubectl create ns vk

2. 为Namespace添加Label。

kubectl label namespace vk alibabacloud.com/eci=true

3. 创建Pod的YAML配置文件。

vim test-pod.yaml

test-pod.yaml的内容示例如下:

```
apiVersion: v1
kind: Pod
metadata:
name: nginx
namespace: vk #指定添加了特定Label的Namespace
spec:
containers:
- image: nginx
imagePullPolicy: Always
name: nginx
```

4. 创建Pod。

kubectl apply -f test-pod.yaml

#### 方式三: 配置ECI弹性调度

您可以在Pod met adat a中添加Annot at ions来声明弹性资源的类型,对应的配置项为 alibabacloud.com/burst-resource,取值如下:

- eci: 当集群普通节点的资源不足时, 使用ECI。
- eci\_only: 只使用ECI。

配置示例如下:

1. 创建Deployment的YAML配置文件。

vim test-deployment.yaml

test-deploymen.yaml的内容示例如下:

apiVersion: apps/v1 kind: Deployment metadata: name: nginx labels: app: nginx spec: replicas: 4 selector: matchLabels: app: nginx template: metadata: name: nginx annotations: alibabacloud.com/burst-resource:eci #设置弹性调度的资源类型 labels: app: nginx spec: containers: - name: nginx image: nginx resources: limits: cpu: 2 requests: cpu: 2

2. 创建Deployment。

kubectl apply -f test-deployment.yaml

## (不推荐) 其它方式

您可以通过配置nodeSelector和tolerations、指定nodeName等方式将Pod调度到ECI上运行。

配置示例如下:

1. 创建Pod的YAML配置文件。

vim test-pod.yaml

test-pod.yaml的内容示例如下:

。 配置nodeSelector和tolerations

apiVersion: v1 kind: Pod metadata: name: nginx spec: containers: - image: nginx imagePullPolicy: Always name: nginx nodeSelector: type: virtual-kubelet tolerations: - key: virtual-kubelet.io/provider operator: Exists

○ 指定nodeName

apiVersion: v1
kind: Pod
metadata:
name: nginx
spec:
containers:
- image: nginx
imagePullPolicy: Always
name: nginx
nodeName: virtual-kubelet

2. 创建Pod。

kubectl apply -f test-pod.yaml

# 2.3. 配置ECI Profile

在Kubernetes场景中通过Virtual Kubelet使用ECI,如果遇到Pod无法无缝调度或者运行在ECI的问题,您可以 尝试通过配置ECI Profile的方式来解决。本文介绍什么是ECI Profile以及如何配置ECI Profile。

## ECI Profile介绍

在Kubernetes场景中通过Virtual Kubelet(简称VK)使用ECI时,可能会碰到Pod无法无缝调度到ECI,或者 Pod调度到ECI后,为了生效ECI的功能特性(如镜像缓存)需要添加Pod Annotation。这类问题一般由集群 管理员处理,但目前却需要研发人员通过调整Pod的配置文件来解决。

针对上述场景, ECI支持通过配置ECI Profile来解决。ECI Profile包括以下三个能力:

• ECI Scheduler

在混合使用ECI和普通节点的场景下,一般可以通过配置Pod Label、配置Namespace Label和配置ECI弹性 调度等方式将Pod调度到ECI,但这些方式均需要对存量资源做一定的修改,无法做到零侵入。

对于上述情况, ECI Scheduler基于Mut at ing Webhook机制实现了一种新的调度机制,即在ECI Profile中, 您可以声明需要匹配的Namespace或者Pod的Label,对于Label能够匹配上的Pod,将被自动调度到ECI。

• ECI Effect

对于ECI的一些功能特性,例如指定ECS实例规格,启用镜像缓存,设置NTP服务等,需要在Pod中追加 Annotation或者Label来实现。更多信息,请参见ECI Pod Annotation。

对于上述情况,ECI Effect实现了自动追加Annotation和Label的功能,即在ECI Profile中,您可以声明需要匹配的Namespace或者Pod的Label,以及需要追加的Annotation和Label,对于Label能够匹配上的Pod,将自动追加配置的Annotation和Label。

• 配置热更新

在ECI Profile中,包含了Cluster IP、混合云模式、PrivateZone的开关配置,以及资源组、安全组、VPC和 交换机等配置,您可以根据需要进行更新,更新后的配置可以即时生效(无需重启VK)。

#### 准备工作

使用ECI Profile时,您需要确保集群中的VK为最新版本,并且如果需要使用ECI Scheduler,则必须要开启 Webhook。

对于不同的Kubernetes集群,相应的准备工作如下:

• 阿里云ASK集群

自动升级VK到最新版本。

```
? 说明
```

ASK集群默认调度Pod到ECI, 无需使用ECI Scheduler。

- 阿里云ACK集群
  - VK托管: 自动升级VK到最新版本,并开启Webhook。
  - VK非托管: 您需要手动修改VK的配置文件,升级VK到最新版本,并开启Webhook。
- 其他集群

您需要手动修改VK的配置文件,升级VK到最新版本,并开启Webhook。

关于如何升级VK,请参见升级Virtual Kubelet。

#### 配置说明

ECI Profile会读取kube-system命名空间下的eci-profile配置文件,然后按照配置文件中的selectors去匹配 Pod,对于Label能够匹配上的Pod,将自动调度到ECI,或者追加Annotation和Label,以便生效ECI的功能特性。

您可以通过以下两种方式编辑eci-profile配置文件:

通过kubectledit命令

kubectl edit configmap eci-profile -n kube-system

- 通过容器服务管理控制台
  - i. 在容器服务管理控制台的集群页面,找到要配置的集群,单击集群名称进入集群详情页面。
  - ii. 在左侧导航栏选择配置管理>配置项。
  - iii. 选择命名空间为kube-system。
  - iv. 找到eci-profile进行编辑。

一个典型的eci-profile配置文件示例如下:

```
apiVersion: v1
kind: ConfigMap
metadata:
name: eci-profile
namespace: kube-system
data:
vpcld: "vpc-xxx"
securityGroupId: "sg-xxx"
vswitchIds: "vsw-111,vsw-222"
enableClusterIp: "true"
enableHybridMode: "false"
enablePrivateZone: "false"
selectors:
 [
   {
     "name":"default-selector-1",
     "objectSelector":{
      "matchLabels":{
        "alibabacloud.com/eci":"true"
      }
     }
   },
   ł
     "name":"default-selector-2",
     "objectSelector":{
      "matchLabels":{
        "eci":"true"
     }
   },
   {
     "name":"default-selector-3",
     "namespaceSelector":{
      "matchLabels":{
        "alibabacloud.com/eci":"true"
      }
     }
   },
   ł
     "name":"default-selector-4",
     "namespaceSelector":{
      "matchLabels":{
        "eci":"true"
      }
     }
   },
   {
     "name":"default-selector-5",
     "namespaceSelector":{
      "matchLabels":{
        "virtual-node-affinity-injection":"enabled"
      }
     },
```

```
"effect":{
    "annotations":{
        "k8s.aliyun.com/eci-image-cache": "true"
    },
    "labels":{
        "created-by-eci":"true"
    }
    }
  }
}
```

在上述配置文件中, data中包含了vpcld、vswitchlds、selectors等配置项。

其中, selectors包含了ECI Scheduler和ECI Effect的配置。在每个selector中, 您必须声明selector的名称, 可以根据需要声明以下信息:

- namespaceSelector: 要匹配的Namespace Label。
- objectSelector: 要匹配的Pod Label。
- effect:需要动态追加的Annotation和Label。

配置selectors后,您可以通过以下命令确认selectors是否生效。如果返回的YAML中已包含配置的 selectors,则表示配置成功;如果未包含selectors,请检查selectors的格式是否有误。

kubectl get mutatingwebhookconfigurations -o yaml vk-webhook

另外,vpcld、vswitchlds等配置对应VPC、交换机等信息,您可以根据需要进行更新。支持更新的配置如下 表所示。

配置项	示例值	说明
enableClusterIp	"true"	是否支持Cluster IP。
enableHybridMode	"false"	是否开启混合云模式。
enablePrivateZone	"false"	是否使用PrivateZone做域名解析。
resourceGroupId	rg-aek2z3elfs4qghy	资源组。
securityGroupId	sg-2ze0b9o8pjjzts4h****	安全组。
vSwitchlds	vsw-2zeet2ksvw7f14ryz****	交换机。可配置多个,用半角逗号间 隔。
vpcld	vpc-2zeghwzptn5zii0w7****	VPC.

## 配置ECI Scheduler示例

在eci-profile的selectors中,您可以以数组的形式声明需要匹配的Namespace或者Pod的Label。对于Label 能够匹配上的Pod,将会被自动调度到ECI。

配置示例如下:

```
○ 注意
实际部署时请去除注释信息。
selectors:
[
 {
  "name":"demo", #必填,不能为空
  "namespaceSelector":{ #选填, K8s的NamespaceLabel (namespaceSelector和objectSelector至少配置一
个)
   "matchLabels":{ #需要匹配的Label,如果填写多个,为与关系
     "department":"bigdata"
   }
  },
  "objectSelector":{ #选填, K8s的Pod Label (namespaceSelector和objectSelector至少配置一个)
    "matchLabels":{ #需要匹配的Label,如果填写多个,为与关系
     "type":"offline-task"
    }
  }
 }
]
↓ 注意
```

在一个selector中, 您至少要配置namespaceSelector和objectSelector中的一个, 如果两者同时配置了, 则Pod需要同时匹配两者。

上述selectors中,声明了一个名为demo的selector,可以实现以下功能:

如果Pod所属命名空间含有department=bigdata标签,并且Pod本身含有type=offline-task标签,则该Pod 将自动被调度到ECI。

### 配置ECI Effect示例

在eci-profile的selectors中,您可以以数组的形式声明需要匹配的Namespace或者Pod的Label,以及需要动态追加的Annotation和Label。对于Label能够匹配上的Pod,将自动追加配置的Annotation和Label。

配置示例如下:

↓ 注意实际部署时请去除注释信息。

```
selectors:
[
{
  "name":"demo", #必填, 不能为空
  "namespaceSelector":{ #选填, K8s的NamespaceLabel
   "matchLabels":{ #需要匹配的Label,如果填写多个,为与关系
     "department":"bigdata"
   }
  },
  "objectSelector":{ #选填, K8s的Pod Label
    "matchLabels":{ #需要匹配的Label,如果填写多个,为与关系
     "type":"offline-task"
   }
  },
  "effect":{ #需要动态追加的Annotation和Label
   "annotations":{
     "k8s.aliyun.com/eci-image-cache": "true"
   },
   "labels":{
     "created-by-eci":"true"
   }
  }
}
1
```

#### ○ 注意

- 在一个selector中,您可以根据需要选择配置namespaceSelector和objectSelector。如果两者 均未配置,则effect对所有调度到ECI的Pod均生效。
- 如果配置了多个selector,将按照顺序匹配selector。匹配成功后,将自动把effect中声明的 Annotation和Label追加到Pod中(只追加,不覆盖)。对于重复的Annotation和Label,Pod原 有声明的值优先级最高,其次是顺序匹配上的第一个selector中的effect,以此类推。

上述selectors中,声明了一个名为demo的selector,可以实现以下功能:

如果Pod所在的命名空间有department=bigdata标签,并且Pod本身含有type=offline-task标签,则该Pod 将自动调度到ECI,并且自动开启镜像缓存功能,增加created-by-eci=true标签。

#### 配置热更新示例

在eci-profile的data中,您可以修改vSwitchlds来更新创建ECl实例时可以使用的交换机,以实现多可用区功能。配置示例如下:

```
data:
enableClusterlp: "true"
enableHybridMode: "false"
enablePrivateZone: "false"
resourceGroupId: ""
securityGroupId: sg-2ze0b9o8pjjzts4h****
selectors: ""
vSwitchIds: vsw-2zeet2ksvw7f14ryz****,vsw-2ze94pjtfuj9vaymf**** #配置多个交换机实现多可用区功能
vpcId: vpc-2zeghwzptn5zii0w7****
```

# 2.4. ACK安装virtual-kubelet-autoscaler

virtual-kubelet-autoscaler是阿里云提供的Kubernetes Autoscaler插件,在Pod调度时,如果没有匹配的 node节点,会由virtual-kubelet-autoscaler进行接管,把调度失败的Pod重新调度到ECI上。

#### 前提条件

您已经成功创建了ACK集群并部署了虚拟节点。

- 关于如何创建ACK集群,请参见创建Kubernetes托管版集群。
- 关于如何部署虚拟节点,请参见部署虚拟节点。

#### 操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在左侧导航栏,选择市场>应用目录。
- 在阿里云应用页签下,找到ack-virtual-kubelet-autoscaler应用,然后单击该应用。
   阿里云应用页签下包含较多应用,您可以通过右上角的搜索框进行关键字检索。
- 4. 在ack-virtual-kubelet-autoscaler页面右侧的创建区域,选择目标集群,单击创建。

应用目录 - ad-virtual-kubelet-autoscaler	
Ack-virtual-kubelet-autoscaler incubator Install Virtual kuberet autoscaler in Allasta Coud Kubernetes cluster.	
90 68 ACK virtual-kubelet-autoscaler Helm Chart	65 <u>8</u>
Altobia Cloud ECI provider is an adapter to connect between kits and ECI service to implement pod from Klabernetes cluster on Altobia Cloud platform. You can dynamically create ECI container instance through an ECI provider.	(双支持 Kubernetes 版本 1.8.4 及以上的集群,对于 1.8.1 版本的集群, 您可以在集群列读中进行"集群升 级"操作。
virtual-kubelet-autoscaler need virtual-kubelet pre-installed. If you haven't installed virtual-kubelet,Piesse install the chart first.	<b>≴.tř</b>  ≠  k2 ▼
Charts Details	命名空间 Luite cutters
The chart will do the following:	发布各称
Deploy virtual-kubelet-autoscaler in Kubernetes cluster.	ack-virtual-kubelet-autoscaler
Prerequisites	2 682
A Kubernetes cluster of Alibaba Cloud container service has been created. Refer to guidance document.	

- 5. 查看部署结果。
  - i. 在左侧导航栏, 单击**集群**。
  - ii. 单击目标集群的ID。
  - iii. 在集群信息页面的左侧导航栏,选择应用>Helm。
  - iv. 查看ack-virtual-kubelet-autoscaler对应的状态是否为已部署。

< j: 2 🗸		所有集群 / 集群:2 / Helm									
自定义资源	•	Helm							应用目	录	刷新
▶ 服务与路由		发布名称	状态	命名空间	Chart 名称	Chart 版本	应用版本	更新时间			操作
▶ 配置管理		ack-kubernetes-elastic-workload	●已部署	kube-system	ack-kubernetes-elastic-workload	0.1.0	1.0	2021-01-18 16:54:55	详情	更新	删除
▶ 存储		ack-node-problem-detector	● 已部署	kube-system	ack-node-problem-detector	1.2.1	0.8.0	2021-01-13 19:49:11	详情	更新	删除
▼ 应用		ack-virtual-kubelet-autoscaler	●已部署	kube-system	ack-virtual-kubelet-autoscaler	0.0.1	0.0.1	2021-01-19 14:41:46	详情	更新	删除
Helm		ack-virtual-node	● 已部署	kube-system	ack-virtual-node	0.0.2	0.0.2	2021-01-13 19:58:21	详情	更新	删除

安装完成后,当ACK集群中已有work node资源不满足条件时,可以把Pod调度至虚拟节点的ECI上。更多信息,请参见通过virtual-kubelet-autoscaler将Pod自动调度到虚拟节点。

# 2.5. ACK部署使用elastic-workload

本文介绍如何通过阿里云Kubernetes集群(ACK)安装和使用elastic-workload(弹性负载)。

#### 前提条件

您已经成功创建了ACK集群并部署了虚拟节点。

- 关于如何创建ACK集群,请参见创建Kubernetes托管版集群。
- 关于如何部署虚拟节点,请参见部署虚拟节点。

#### 背景信息

在Kubernetes中,弹性包括调度层弹性(Pod)和资源层弹性(Node)两个层次,通常会通过HPA、 CronHPA、VPA等模型进行Pod的弹性伸缩,再通过cluster-autoscaler或者virtual-kubelet进行资源层的弹 性伸缩。两层之间通过Pod进行解耦,这样设计的好处是两层职责明确,坏处是解耦后相互结合的策略过于 简单,无法实现更精细的调度策略,在Kubernetes中最小的生命周期管理单元是一个Pod,而传统的 Kubernetes负载控制器(例如: Deployment、StatefulSet)管理的Pod是共享相同的调度策略的。因此, 如果您想要控制一个负载在不同资源上的细粒度分配时,您可以通过elastic-workload来实现。

#### 部署ack-kubernetes-elastic-workload

- 1. 登录容器服务管理控制台。
- 2. 在左侧导航栏,选择市场>应用目录。
- 3. 在**阿里云应用**页签下,找到ack-kubernetes-elastic-workload应用,然后单击该应用。 **阿里云应用**页签下包含较多应用,您可以通过右上角的搜索框进行关键字检索。
- 4. 在ack-kubernetes-elastic-workload页面右侧的创建区域,选择目标集群,单击创建。

应用目录 - ack-kubernetes-elastic-workload			
ack-kubernetes incubator ack-elastic is a kind of worklos	-elastic-workload		
说明参数			创建
Overview kubernetes-elastic-workload is a new kind of kubernetes meticulous scheduling such as fault recovery, multi unit d Architecture Design	vorkload especially for autoccaling. Traditional kubernetes workload is simple. The pods of k stribution and so on, kubernetex-elastic-workload would be a good choice.	subernetes workload are within a single schduling policy. If you want to do	(33년 Automote 왕조 134 20以上的意思, 37月 13.1 왕국的意思, 57月 以正義部列後十世子 南部中 왕 영웅, 28 28
	НРА		ALE
		•	ack-tubernetes-elastic-worklad

- 5. 查看部署结果。
  - i. 在左侧导航栏, 单击集群。
  - ii. 单击目标集群的ID。
  - iii. 在集群信息页面的左侧导航栏,选择应用>Helm。
  - iv. 查看ack-kubernetes-elastic-workload对应的状态是否为已部署。

< js ck2 →		所有集群 / 集群: js == k2 / Heli	m								
▶ 服务与路由	•	Helm							应用目	录	刷新
<ul> <li>配置管理</li> </ul>											
▶ 存储		发布名称	状态	命名空间	Chart 名称	Chart 版本	应用版本	更新时间			操作
▼ 应用		ack-kubernetes-elastic-workload	●已部署	kube-system	ack-kubernetes-elastic-workload	0.1.0	1.0	2021-01-18 16:54:55	详情	更新	删除
Helm		ack-node-problem-detector	●已部署	kube-system	ack-node-problem-detector	1.2.1	0.8.0	2021-01-13 19:49:11	详情	更新	删除
灰度发布 (公測)		ack-virtual-node	●已部署	kube-system	ack-virtual-node	0.0.2	0.0.2	2021-01-13 19:58:21	详情	更新	删除

#### 使用elastic-workload

假设:有一个应用经过容量规划,预计最多有4个副本运行在ECS上,平时低峰时保留2个副本,超过4个副本时弹性伸缩到虚拟节点,防止干扰其他正常容量规划的应用。

在Kubernetes中,任何一种负载都要解决两个问题,一个是调度问题,一个是生命周期管理问题。如果想要 实现上述场景,需要解决以下问题:

- 当副本数目达到某个数值后,如何控制调度策略的变化。
- 在生命周期管理时,如何优先处理某些Pod。

针对上述场景和问题,下文为您介绍elastic-workload的使用方法。

1. 创建一个简单的Deployment类型应用。

apiVersion: apps/v1 kind: Deployment metadata: name: nginx-deployment-basic labels: app: nginx spec: replicas: 2 selector: matchLabels: app: nginx template: metadata: labels: app: nginx spec: containers: - name: nginx image: nginx:1.7.9 ports: - containerPort: 80 resources: requests: #如果要配合使用HPA,必须要指定requests cpu: "500m" memory: "1024Mi"

2. 使用弹性负载表达。

apiVersion: autoscaling.alibabacloud.com/v1beta1 kind: ElasticWorkload metadata: name: elasticworkload-sample spec: sourceTarget: name: nginx-deployment-basic kind: Deployment apiVersion: apps/v1 min: 2 #最小副本数 max: 4 #最大副本数 replicas: 6 elasticUnit: - name: virtual-kubelet labels: alibabacloud.com/eci: "true"

弹性负载的使用方式就像HPA,它通过外部挂载的方式使用,并不影响原有业务。



弹性负载会监听原始负载,并根据弹性单元设定的调度策略,克隆并生成弹性单元的负载。根据弹性负载中副本的变化,动态地分配原始负载和弹性单元上的副本数目。

一个典型的弹性负载主要包括两个部分:

- sourceTarget部分主要定义原始负载的类型、副本数目可变化的范围。
- elasticUnit部分是一个数组,定义弹性单元的调度策略,如果有多个弹性单元,则按照模板的顺序定义。

在上述示例中:

- sourceTarget中定义了副本数目的上下限为2~4个,表示ElasticWorkload的replicas为2~4个副本时, 会调度到sourceTarget,当超过4个副本时,会调度到弹性单元,即虚拟节点virtual-kubelet。
- elast icUnit 中定义了弹性单元为虚拟节点virt ual-kubelet,对应的调度策略为
   labels: alibabacloud.com/eci=true 。
- 3. 查看部署结果。
  - 查看当前状态

kubectl describe ew elasticworkload-sample

返回示例如下,其中Status中的每个单元的Desired Replicas表示弹性负载的分配副本数目。

Name: elasticworkload-sample Namespace: default Labels: <none> Annotations: <none> API Version: autoscaling.alibabacloud.com/v1beta1

Kind: ElasticWorkload Metadata: Creation Timestamp: 2021-05-21T01:53:58Z Generation: 4 Managed Fields: API Version: autoscaling.alibabacloud.com/v1beta1 Fields Type: FieldsV1 fieldsV1: f:spec: .: f:elasticUnit: f:replicas: f:sourceTarget: .: f:apiVersion: f:kind: f:max: f:min: f:name: Manager: Apache-HttpClient **Operation: Update** Time: 2021-05-21T01:53:58Z API Version: autoscaling.alibabacloud.com/v1beta1 Fields Type: FieldsV1 fieldsV1: f:status: .: f:elasticUnitsStatus: f:replicas: f:selector: f:sourceTarget: .: f:apiVersion: f:desiredReplicas: f:kind: f:name: f:updateTimestamp: Manager: manager Operation: Update Time: 2021-05-21T01:56:45Z **Resource Version: 8727** /apis/autoscaling.alibabacloud.com/v1beta1/namespaces/default/elasticworkloads/el Self Link: asticworkload-sample c4a508aa-2702-4d17-ac25-e6a207c0761a UID: Spec: **Elastic Unit:** Labels: alibabacloud.com/eci: true Name: virtual-kubelet **Replicas:** 6 Source Target: API Version: apps/v1 Deployment Kind: Max

мах: 4 Min: 2 nginx-deployment-basic Name: Status: **Elastic Units Status: Desired Replicas: 2** Name: nginx-deployment-basic-unit-virtual-kubelet Update Timestamp: 2021-05-21T01:56:45Z **Replicas:** 6 Selector: app=nginx Source Target: API Version: apps/v1 **Desired Replicas: 4** Kind: Deployment nginx-deployment-basic Name: Update Timestamp: 2021-05-21T01:56:45Z Events: Type Reason Age From Message --------Normal SourceUpdate ElasticWorkload Source Target scale from 2 to 4 12m Normal UnitCreation ElasticWorkload ElasticWorkloadUnit nginx-deployment-basi 12m c-unit-virtual-kubelet created Normal ElasticWorkloadUpdate 9m27s (x9 over 12m) ElasticWorkload ElasticWorkload update Normal UnitUpdate 9m27s (x8 over 12m) ElasticWorkload ElasticWorkloadUnit virtual-kubelet has been updated

○ 查看Pod情况

kubectl get pod -o wide

返回示例如下,弹性负载已经克隆出新的Deployment与Pod,且Deployment的Pod副本数目是根据 设定的调度策略进行动态分配。

NAME	READY STATUS	RESTAR	TS AGE IP	NODE	NOMINA
TED NODE READINESS GATES					
nginx-deployment-basic-5bf87	7f5f59-22jnw	1/1	Running 0	16m 10.34.0.	131 cn-beiji
ng.172.16.0.1 <none></none>	<none></none>				
nginx-deployment-basic-5bf87	7f5f59-gfp24	1/1	Running 0	13m 10.34.0.1	.33 cn-beiji
ng.172.16.0.1 <none></none>	<none></none>				
nginx-deployment-basic-5bf87	f5f59-pw2zx	1/1	Running 0	13m 10.34.0.	134 cn-beiji
ng.172.16.0.1 <none></none>	<none></none>				
nginx-deployment-basic-5bf87	f5f59-qvh7m	1/1	Running 0	16m 10.34.0	.132 cn-beij
ing.172.16.0.1 <none></none>	<none></none>				
nginx-deployment-basic-unit-	virtual-kubelet-65f	b6f4cd7-	48ssb 1/1 Ru	nning 0 13	m 172.16.22.
157 virtual-kubelet-cn-beijing	g-e <none> <n< td=""><td>one&gt;</td><td></td><td></td><td></td></n<></none>	one>			
nginx-deployment-basic-unit-	virtual-kubelet-65f	b6f4cd7-	gjqhm 1/1 Ru	unning 0 1	3m 172.16.22
.158 virtual-kubelet-cn-beijing	g-e <none> <r< td=""><td>none&gt;</td><td></td><td></td><td></td></r<></none>	none>			

此外,弹性负载支持与HPA配合使用,可以将HPA作用在弹性负载上。弹性负载会根据HPA的状态动态调整 每个单元的副本分布,例如:如果当前是从6个副本缩容到4个副本,会优先将弹性单元的副本进行缩容。示 例如下: apiVersion: autoscaling/v2beta2 kind: HorizontalPodAutoscaler metadata: name: elastic-workload-demo namespace: default spec: scaleTargetRef: apiVersion: autoscaling.alibabacloud.com/v1beta1 kind: ElasticWorkload name: elasticworkload-sample minReplicas: 2 maxReplicas: 10 metrics: - type: Resource resource: name: cpu target: type: Utilization averageUtilization: 50

总的来说,弹性负载一方面通过克隆和覆写调度策略的方式生成多个Deployment,实现了调度策略的管理,另一方面通过上层的副本计算,调整原始负载和弹性单元的副本分配,实现了针对一部分Pod的优先处理。

# 2.6. ECI Pod Annotation

Kubernetes集群通过虚拟节点创建Pod到ECI时,为充分使用ECI提供的功能,在不改变Kubernetes语义的前提下,您可以根据需求为Pod添加Annotation。本文为您介绍ECI实例支持的Annotation及其配置示例。

ECI实例目前支持的Annotation如下表所示。

#### ? 说明

下表列举的Annotation仅适用于创建到虚拟节点上的Pod,即ECI实例,调度到ECS上的Pod不受这些Annotation影响。

参数	示例值	描述	相关文档
k8s.aliyun.com/eci -security-group	sg- bp1dktddjsg5nktv ****	安全组ID。	配置安全组
k8s.aliyun.com/eci -vswitch	vsw- bp1xpiowfm5vo8 o3c****	交换机ID,支持指定多个交换机实现多可用 区功能。	

参数	示例值	描述	多可用区创建实例 相关文档
k8s.aliyun.com/eci -schedule- strategy	VSwitchOrdered	多可用区调度策略。取值范围: • VSwitchOrdered:按顺序 • VSwitchRandom:随机	
k8s.aliyun.com/eci -ram-role-name	AliyunECIContainer GroupRole	AliyunEClContainer GroupRole RAM角色,赋予ECl访问阿里云产品的能力。	
k8s.aliyun.com/eci -use-specs	2-4Gi,4- 8Gi,ecs.c6.xlarge	ECI实例规格,支持指定多规格,包括指定 vCPU和内存,或者ECS规格。	多规格创建实例
k8s.aliyun.com/eci -spot-strategy	SpotAsPriceGo	抢占式实例策略。取值范围: <ul> <li>SpotAsPriceGo:系统自动出价,跟随当前市场实际价格。</li> <li>SpotWithPriceLimit:设置抢占实例价格上限。</li> </ul>	创建抢占式实例
k8s.aliyun.com/eci -spot-price-limit	0.5	抢占式实例价格。仅当k8s.aliyun.com/eci- spot-strategy设置为SpotWithPriceLimit时 有效。	
k8s.aliyun.com/eci -cpu-option-core	2	CPU物理核心数。	白之义の以选项
k8s.aliyun.com/eci -cpu-option-ht	1	每核线程数。	日正又いし近坝
k8s.aliyun.com/eci -reschedule- enable	"true"	是否开启ECI重调度。	无,详见下文。
k8s.aliyun.com/po d-fail-on-create- err	"true"	创建失败的ECI实例是否体现Failed状态。	无,详见下文。

#### Kubernetes对接ECI·使用ECI

参数	示例值	描述	相关文档
k8s.aliyun.com/eci -image-snapshot- id	imc- 2zebxkiif uyzzlhl*** *	指定镜像缓存ID。 ⑦ 说明 使用镜像缓存支持手动指定和自动匹配 两种方式,建议使用自动匹配方式。	使田錇俊绥友C₽D加
k8s.aliyun.com/eci -image-cache	"true"	自动匹配镜像缓存。 ⑦ 说明 使用镜像缓存支持手动指定和自动匹配 两种方式,建议使用自动匹配方式。	速创建Pod
k8s.aliyun.com/acr -instance-id	cri- j36zhodptmyq****	ACR企业版实例ID。	配置ACR企业版免密
k8s.aliyun.com/eci -eip-instanceid	eip- bp1q5n8cq4p7f6d zu****	EIP实例ID。	
k8s.aliyun.com/eci -with-eip	"true"	是否自动创建并绑定EIP。	
k8s.aliyun.com/eip -bandwidth	5	EIP带宽。	_
k8s.aliyun.com/eip -common- bandwidth- package-id	cbwp- 2zeukbj916scmj51 m****	共享带宽包ID。	
k8s.aliyun.com/eip -isp	BGP	<ul> <li>EIP线路类型,仅按量付费的EIP支持指定。取 值范围:</li> <li>BPG: BGP(多线)线路</li> <li>BGP_PRO: BGP(多线)精品线路</li> </ul>	为ECI实例绑定EIP

参数	示例值	示例值 描述		
k8s.aliyun.com/eip -internet-charge- type	PayByBandwidth	<ul><li>EIP的计量方式。取值范围:</li><li>PayByBandwidth:按带宽计费</li><li>PayByTraffic:按流量计费</li></ul>		
k8s.aliyun.com/eci -enable-ipv6	"true"	是否分配IPv6。	配置IPv6地址	
kubernetes.io/ingr ess-bandwidth	40M	入方向带宽。	ECI实例进行带宽限	
kubernetes.io/egr ess-bandwidth	20M	出方向带宽。	速	
k8s.aliyun.com/eci -extra-ephemeral- storage	50Gi	临时存储空间大小。	自定义临时存储空 间大小	
k8s.aliyun.com/eci -core-pattern	/pod/data/dump /core	Core dump文件保存目录。	查看Core dump文 件	
k8s.aliyun.com/eci -ntp-server	100.100.*.*	NTP Server。	为Pod配置NTP服务	

### 设置安全组

Virtual Kubelet启动时,将通过环境变量设置默认的安全组。所有创建在虚拟节点上的Pod默认使用Virtual Kubelet配置的。如果您有特殊需求,可以通过添加Annotation的方式,为Pod设置特定的安全组。

apiVersion: apps/v1	
kind: Deployment	
metadata:	
name: demo	
labels:	
app: nginx	
spec:	
replicas: 1	
selector:	
matchLabels:	
app: nginx	
template:	
metadata:	
annotations:	
k8s.aliyun.com/eci-security-group: "sg-bp1dktddjsg5nktv****"	#设置安全组。
labels:	
app: nginx	
spec:	
containers:	
- name: nginx	
image: nginx:latest	
nodeName: virtual-kubelet	

### 多可用区创建Pod

在创建Pod时,您可以指定多个交换机实现多可用区功能。系统将根据资源的库存情况,选择合适的可用区创建Pod。更多信息,请参见多可用区创建实例。

配置示例如下:

```
apiVersion: v1
kind: Pod
metadata:
annotations:
k8s.aliyun.com/eci-vswitch: "vsw-bp1xpiowfm5vo8o3c****,vsw-bp1rkyjgr1xwoho6k****" #指定多个交换
机ID。
k8s.aliyun.com/eci-schedule-strategy: "VSwitchOrdered" #设置多可用区调度策略。
name: nginx-test
spec:
containers:
- name: nginx
image: nginx:latest
```

### 设置RAM角色

您可以通过添加Annotation的方式为Pod设置RAM角色,授予Pod访问阿里云产品的能力。

○ 注意

请确保RAM角色的受信服务为云服务器。

apiVersion: apps/v1
kind: Deployment
metadata:
name: set-ram-role
labels:
app: vk
spec:
replicas: 1
selector:
matchLabels:
app: nginx
template:
metadata:
annotations:
k8s.aliyun.com/eci-ram-role-name:"AliyunECIContainerGroupRole" #设置RAM角色。
labels:
app: nginx
spec:
containers:
- name: nginx
image: nginx:latest
nodeName: virtual-kubelet

## 指定实例规格创建Pod

您可以通过添加Annotation的方式为Pod指定允许使用的实例规格。在创建Pod时,如果遇到库存不足的情况,将按照顺序遍历指定的实例规格,保证创建成功率。更多信息,请参见<mark>多规格创建实例</mark>。

#### ? 说明

实例规格支持直接指定vCPU和内存,或者指定具体的ECS规格。如果业务有特殊的规格需求,例如: GPU、高主频、本地盘等,请明确指定ECS规格。更多信息,请参见指定ECS规格创建实例。

apiVersion: apps/v1
kind: Deployment
metadata:
name: vk-cos-use
labels:
app: cos
spec:
replicas: 1
selector:
matchLabels:
app: cos
template:
metadata:
annotations:
"k8s.aliyun.com/eci-use-specs": "2-4Gi,4-8Gi,ecs.c6.xlarge" #支持指定多个规格,规格可以是vCPU和内存,
或者具体的ECS规格。
labels:
app: cos
spec:
containers:
- name: u1
image: "registry-vpc.cn-beijing.aliyuncs.com/lxx/cos-4g"
nodeName: virtual-kubelet

## 创建抢占式实例

对于无状态应用、Job任务等,使用抢占式实例运行,可以有效地节约实例使用成本。您可以通过添加 Annotation的方式,创建抢占式ECI示例。更多信息,请参见创建抢占式实例。

apiVersion: apps/v1 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
name: nginx-deployment-basic
labels:
app: nginx
spec:
replicas: 2
selector:
matchLabels:
app: nginx
template:
metadata:
labels:
app: nginx
annotations:
k8s.aliyun.com/eci-use-specs : "ecs.c5.large" #根据需要替换ECS实例规格。
k8s.aliyun.com/eci-spot-strategy:"SpotWithPriceLimit"#采用自定义设置价格上限的策略。
k8s.aliyun.com/eci-spot-price-limit: "0.250" #设置每小时价格上限。
spec:
# nodeSelector:
# env:test-team
containers:
- name: nginx
image: nginx:1.7.9 # replace it with your exactly <image_name:tags></image_name:tags>
ports:
- containerPort: 80

### 自定义设置CPU选项

对于一台ECI实例,CPU选项由CPU物理核心数和每核线程数决定。根据您创建ECI实例的方式,部分ECI实例支持自定义CPU选项。更多信息,请参见自定义CPU选项。

配置示例如下:

apiVersion: v1
kind: Pod
metadata:
annotations:
k8s.aliyun.com/eci-use-specs : "ecs.c6.2xlarge" #指定支持自定义CPU选项的ECS规格。
k8s.aliyun.com/eci-cpu-option-core: "2" #自定义设置CPU物理核心数为2。
k8s.aliyun.com/eci-cpu-option-ht: "1"    #自定义设置每核线程数为1,即关闭超线程。
name: nginx-test
spec:
containers:
- name: nginx
image: nginx:latest
restartpolicy: Always

#### 设置ECI重调度

调度Pod到虚拟节点时,可能会碰到调度失败的情况,您可以通过添加Annotation的方式为Pod开启重调度,即使异步调度失败了,仍会一直保持调度,不返回失败。

配置示例如下:

apiVersion: apps/v1 kind: Deployment metadata: name: set-eci labels: app: vk spec: replicas: 1 selector: matchLabels: app: nginx template: metadata: annotations: k8s.aliyun.com/eci-reschedule-enable: "true" #开启ECI重调度。 labels: app: nginx spec: containers: - name: nginx image: nginx:latest nodeName: virtual-kubelet

## Pod创建失败置为Failed

默认情况下,每个Pod在创建时,如果遇到错误,会重试一定次数,如果还是Failed,则Pod会处于Pending 状态。对于一些Job类型的任务,您可能希望直接体现Failed状态。此时,您可以通过添加Annotation的方 式,设置Pod遇到创建失败时,体现Failed状态。

apiVersion: apps/v1 kind: Deployment metadata: name: set-pod-fail-on-create-err labels: app: vk spec: replicas: 1 selector: matchLabels: app: nginx template: metadata: annotations: k8s.aliyun.com/pod-fail-on-create-err: "true" #设置Pod创建失败时,状态为Failed。 labels: app: nginx spec: containers: - name: nginx image: nginx:latest nodeName: virtual-kubelet

### 使用镜像缓存CRD加速创建Pod

使用镜像缓存技术可以加速创建Pod,您可以通过添加Annotation的方式,为Pod指定使用的镜像缓存,或 者开启自动匹配镜像缓存。更多信息,请参见使用镜像缓存CRD加速创建Pod。

配置示例如下:

指定镜像缓存

```
apiVersion: v1
kind: Pod
metadata:
annotations:
 k8s.aliyun.com/eci-image-snapshot-id: imc-2ze5tm5gehgtiiga**** #指定使用的镜像缓存。
name: nginx-imagecache-id
spec:
containers:
- image: nginx:1.7.9
 imagePullPolicy: IfNotPresent
 name: nginx
 resources:
  limits:
   cpu: 300m
   memory: 200Mi
  requests:
   cpu: 200m
   memory: 100Mi
nodeName: virtual-kubelet
```

• 自动匹配

spec:

apiVersion: v1 kind: Pod metadata: annotations: k8s.aliyun.com/eci-image-cache: "true" #开启自动匹配镜像缓存。 name: nginx-auto-match containers: - image: nginx:1.7.9 imagePullPolicy: IfNotPresent name: nginx resources: limits: cpu: 300m

```
memory: 200Mi
 requests:
  cpu: 200m
  memory: 100Mi
nodeName: virtual-kubelet
```

# 指定ACR企业版实例

阿里云容器服务ACR支持免密拉取,您可以通过添加Annotation的方式指定ACR企业版实例,从对应的镜像 仓库中拉取镜像。更多信息,请参见配置ACR企业版免密。

配置示例如下:

```
apiVersion: v1
kind: Pod
metadata:
annotations:
 k8s.aliyun.com/acr-instance-id: cri-j36zhodptmyq**** #指定ACR企业版实例。
name: cri-test
spec:
containers:
- image: test****-registry.cn-beijing.cr.aliyuncs.com/eci_test/nginx:1.0 #使用公网拉取镜像。
 imagePullPolicy: Always
 name: nginx
restartPolicy: Never
```

#### 绑定弹性公网IP

如果Pod有公网通信的需求,您可以为其绑定EIP。更多信息,请参见如何给Pod挂载弹性公网IP。

配置示例如下:

自动创建

apiVersion: v1 kind: Pod metadata: name: nginx annotations: k8s.aliyun.com/eci-with-eip:"true" #开启自动创建EIP。 k8s.aliyun.com/eip-bandwidth:"10" #设置带宽,默认为5,单位为Mbps。 spec: containers: - image: registry-vpc.cn-hangzhou.aliyuncs.com/jovi/nginx:alpine imagePullPolicy: Always name: nginx ports: - containerPort: 80 name: http protocol: TCP restartPolicy: OnFailure

● 指定EIP

apiVersion: v1 kind: Pod metadata: name: nginx annotations: k8s.aliyun.com/eci-eip-instanceid: "eip-bp1q5n8cq4p7f6dzu\*\*\*\*" #指定EIP。 spec: containers: - image: registry-vpc.cn-hangzhou.aliyuncs.com/jovi/nginx:alpine imagePullPolicy: Always name: nginx ports: - containerPort: 80 name: http protocol: TCP restartPolicy: OnFailure

#### 配置IPv6

相比IPv4,IPv6不仅可以解决网络地址资源有限的问题,还可以解决多种接入设备连入互联网障碍的问题。 您可以通过添加Annotation的方式为Pod分配IPv6地址。更多信息,请参见配置IPv6地址。

apiVersion: v1 kind: Pod metadata: name: nginx annotations: k8s.aliyun.com/eci-enable-ipv6: "true" #开启自动分配IPv6。 spec: containers: - name: nginx image: nginx nodeName: virtual-kubelet

### 设置出入方向带宽

CI支持配置流入和流出的网络带宽值。您可以通过添加Annotation的方式,为Pod指定出方向和入方向的带 宽值进行限速。更多信息,请参见ECI实例进行带宽限速。

配置示例如下:

```
apiVersion: v1
kind: Pod
metadata:
name: eci-qos
annotations:
kubernetes.io/ingress-bandwidth: 40M #设置入方向带宽。
kubernetes.io/egress-bandwidth: 10M #设置出方向带宽。
spec:
containers:
- name: nginx
image: nginx:latest
command: ["bash","-c","sleep 100000"]
```

### 设置临时存储空间大小

ECI实例默认提供20 GiB的免费存储空间,如果该存储空间大小无法满足您的需求,您可以通过添加 Annotation的方式,自定义增加临时存储空间大小。更多信息,请参见自定义临时存储空间大小。

```
apiVersion: v1
kind: Pod
metadata:
name: test
annotations:
k8s.aliyun.com/eci-extra-ephemeral-storage: "50Gi" #自定义设置临时存储空间大小。
spec:
containers:
- name: nginx
image: nginx:latest
image PullPolicy: lfNotPresent
restartPolicy: Always
nodeName: virtual-kubelet
```

# 设置Core dump文件的保存目录

容器进程异常退出会产生Core dump文件,默认情况下,Core dump文件的名称为core.pid,保存在当前目录下。您可以通过添加Annotation的方式,自定义设置Core dump文件保存目录。更多信息,请参见查看Core dump文件。

配置示例如下:

```
apiVersion: v1
kind: Pod
metadata:
name: test
annotations:
 k8s.aliyun.com/eci-core-pattern: "pod/data/dump/core" # 设置Core dump文件保存目录。
spec:
containers:
- image: nginx:latest
 name: test-container
 volumeMounts:
 - mountPath: /pod/data/dump/
  name: default-volume
volumes:
- name: nfs
 nfs:
  server: 143b24****-gfn3.cn-beijing.nas.aliyuncs.com
  path:/dump/
  readOnly: false
```

### 为Pod设置NTP服务

您可以为Pod添加k8s.aliyun.com/eci-ntp-server的Annotation来配置NTP服务。更多信息,请参见为Pod配置NTP服务。

apiVersion: apps/v1
kind: Deployment
metadata:
name: set-ngnix-ntp
labels:
app: vk
spec:
replicas: 1
selector:
matchLabels:
app: nginx
template:
metadata:
annotations:
k8s.aliyun.com/eci-ntp-server: 100.100.5.*,100.100.5.* #设置NTP服务器地址。
labels:
app: nginx
spec:
containers:
- name: nginx
image: nginx:latest
nodeName: virtual-kubelet

# 2.7. ECI自定义事件

本文为您介绍ECI自定义事件及对应的解决方案。

# ECI实例 (ContainerGroup) 相关事件

事件名称 (reason)	事件 类型	事件信息(message)	事件说明
StockExhaust	Warnin g	The stock of the specified zones will be used up. %s	当前可用区的ECI资源库存即将耗尽。建议更 换可用区。
NoStock	Warnin g	Create ECI failed because the specified instance is out of stock. %s	当前可用区的ECI资源库存不足。您可以使用 多可用区和多规格的方式创建ECI来提高创建 成功率。具体操作,请参见 <mark>多可用区创建实</mark> 例和多规格创建实例。
StockClose	Warnin g	Create ECI failed because current zone closed or limited without living instance. %s	创建ECI实例失败,无法使用当前可用区。请 更换可用区。
FailedScheduling	Warnin g	Unknown error occurred.	创建ECl实例失败,发生未知错误。请提交工 单。

事件名称 (reason)	事件 类型	事件信息(message)	事件说明
FailedScheduling	Warnin g	Schedule eci failed:%s	调度ECI实例失败。请重新尝试,或者提交工 单。
UnknownError	Warnin g	The ECI service is under heavy load while creating container group resources, please wait and try again later	ECI服务负载过大,请稍后再尝试创建ECI实 例。
SpotToBeRelease d	Warnin g	Spot ECI will be released in %s minutes	抢占式ECI实例即将过期,您需要在业务上进 行兼容处理。
DiskCapacityQuota Full	Warnin g	Your disk capacity quota is exceeded	磁盘容量已达到配额限制。请提交工单申请 提升配额。
AutoInstanceType Match	Normal	The most-matched instanceType for current eci instance is %s	ECI实例规格已进行规整。
UnknownError	Warnin g	An unknown error occurred for %s	发生未知错误。请提交工单。
CpuOptionsNotSu pported	Warnin g	The cpu options is not supported for your instanceType[%s] in current region	当前可用区下,您的ECI实例不支持CPU选 项。更多信息,请参见 <mark>自定义CPU选项</mark> 。
CpuOptionsNotVal id	Warnin g	The cpu options is not valid for your instanceType[%s]	您的ECI实例不支持CPU选项。更多信息,请 参见 <mark>自定义CPU选项</mark> 。
SystemFailureReb oot	Warnin g	The Specified ContainerGroup is rebooting	ECI实例正在重新启动。
Throttling	Warnin g	The request was denied due to system flow control, please wait and try again later	由于系统流量控制 <i>,</i> 请求被拒绝。请稍候再 尝试操作。

事件名称 (reason)	事件 类型	事件信息(message)	事件说明
MultiZoneRecomm endations	Normal	%s	建议使用多可用区功能。更多信息,请参 见 <mark>多可用区创建实例</mark> 。

# 镜像缓存(ImageCache)相关事件

事件名称 (reason)	事件 类型	事件信息(message)	事件说明
ImageCacheNotFo und	Warnin g	The specified image cache %s does not exist	指定的镜像缓存不存在。建议您先查询镜像 缓存,确保使用已存在的镜像缓存。
ImageCacheMissed	Warnin g	Missed image cache	没有匹配到镜像缓存,您需要先创建镜像缓 存。具体操作,请参见 <mark>创建镜像缓存</mark> 。
lmageCacheAutoCr eated	Normal	lmage cache %s is auto created	没有匹配到镜像缓存,系统已开始自动创建 镜像缓存。
ImageCacheAutoEl iminated	Warnin g	Image cache %s is auto eliminated	系统开始自动创建镜像缓存时,镜像缓存数 量已达到配额限制。系统将根据LRU策略自 动淘汰镜像缓存,您也可以提交工单申请提 升配额。
ImageCacheCapaci tyUtilization	Normal	Total capacity utilization of current CacheDisk is %sGB / %sGB	当前镜像缓存制作完成占用的空间使用量情 况。
lmageCacheAutoCr eateFailed	Warnin g	Image cache auto create failed for %s	系统自动创建镜像缓存失败。请重新尝试, 或者手动创建镜像缓存。
mageCacheFlashCr eatedFailed	Warnin g	Flash image cache is failed,standard image cache will be created	极速镜像缓存(对应本地快照)创建失败, 系统开始创建普通镜像缓存。
ImageCacheFlashFi nished	Normal	Flash image cache is created successfully, standard image cache will be created	极速镜像缓存(对应本地快照)创建成功, 系统开始创建普通镜像缓存。

事件名称 (reason)	事件 类型	事件信息(message)	事件说明
ImageCacheFlashU sedFailed	Warnin g	Flash image cache can only be used in zones with ESSD, but current zone is %s	极速镜像缓存(对应本地快照)需要使用 ESSD云盘,仅可以在支持ESSD云盘的可用区 下创建。
ImageCacheFlashR eplacedFailed	Warnin g	Standard image cache is failed, flash image cache %s will not be deleted. But flash image can only be used in zones with ESSD	极速镜像缓存(对应本地快照)创建成功, 但普通镜像缓存创建失败。系统将保留极速 镜像缓存对应的本地快照。
ImageCacheFlashR eplacedSuccess	Warnin g	Flash image cache [%s] has been replaced successfully	极速镜像缓存对应的本地快照已被正常删 除。
lmageCacheCreat e	Normal	Snapshot [%s] of image cache has been created, waiting for the completion of it	开始创建镜像缓存,请等待镜像缓存创建完 成后再使用。
lmageCacheFinishe d	Normal	Image cache [%s] has been made successfully	镜像缓存创建成功,您可以使用该镜像缓 存。
lmageCacheUpdat eFinished	Normal	Image cache [%s] has been updated successfully	镜像缓存更新成功,您可以使用该镜像缓 存。
lmageCacheUpdat eFailed	Warnin g	Image cache [%s] update failed, but the old resource is still available	镜像缓存更新失败,但原有镜像缓存仍可以 使用。请重新尝试更新操作。
lmageCacheUnkno wnError	Warnin g	An unknown error occurred while handling the image cache for %s	镜像缓存发生未知错误。请提交工单确认问 题。
SuccessfulHitImag eCache	Normal	Successfully hit image cache %s	成功匹配镜像缓存。
ImageCacheNotCo mplete	Normal	The production for specified image cache %s is not yet complete	镜像缓存还未创建完成。请确保镜像缓存已 创建完成(达到Ready状态),再进行使 用。

事件名称 (reason)	事件 类型	事件信息(message)	事件说明
ImageCacheElimin	Warnin	Current imageCache is	您使用的镜像缓存已被删除。请尝试使用其
ated	g	eliminated	它镜像缓存。

# 其它资源相关事件

资源 模块	事件 名称 (rea son)	事件 类型	事件信息(message)	事件说明
RamRo le	Unauth orized	Warnin g	No permission granted to eci service account for %s	账号没有操作ECI资源的权限。请确保账号已 有ECI服务关联角色。如果使用RAM用户,请 确保已正确授权RAM角色,具体操作,请参 见 <mark>为RAM用户授权</mark> 。
vSwitc h	Resour celnsuf ficient	Warnin g	The maximum number of IP address in the VSwitch %s is exceeded	交换机下可用的IP地址数量不足。请使用其 它交换机。
Securit yGroup	QuotaF ull	Warnin g	The maximum number of instances in the security group %s is exceeded	安全组内的实例数量已达到配额限制。请使 用其它安全组,或者提交工单申请提升配 额。
EIP	Create EipFaile d	Warnin g	%s	创建EIP失败。请重新尝试。
SLS	AliyunS lsQuot aExcee d	Warnin g	%s	SLS资源超出配额限制。请提交工单申请提升 配额。
SLS	AliyunS lsError	Warnin g	%s	SLS相关错误。
SLS	AliyunS lsProje ctInvali d	Warnin g	%s	指定的SLS Project是无效的。