阿里云

专有云企业版

容器服务Kubernetes版 开发指南

产品版本: v3.16.2

文档版本: 20220915

(一) 阿里云

容器服务Kubernetes版 开发指南·法律声明

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。

容器服务Kubernetes版 开发指南·<mark>通用约定</mark>

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。	
□ 注意	用于警示信息、补充说明等,是用户必须 了解的内容。	八)注意 权重设置为0,该服务器不会再接受新请求。
⑦ 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。	② 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 listinstanceid 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.SDK参考	80
1.1. 快速入门	08
1.1.1. 开发文档使用说明	08
1.1.2. API调用流程说明	09
1.2. 准备工作	11
1.2.1. 登录API与工具控制台	11
1.2.2. 查看API信息	12
1.2.3. 获取AccessKey	13
1.2.4. 获取STS AccessKey	14
1.2.5. 获取公共Header参数	17
1.2.6. 获取容器服务ACK的Endpoint	19
1.2.7. 获取容器服务ACK的SDK包	20
1.3. Java SDK调用示例	20
1.3.1. 安装Java SDK	20
1.3.2. 调用SDK	21
1.4. Python SDK调用示例	22
1.4.1. 安装Python SDK	22
1.4.2. 调用SDK	23
1.5. Go SDK调用示例	24
1.5.1. 安装Go SDK	24
1.5.2. 调用SDK	25
2.集群 API 列表	27
2.1. 集群 API 调用方式	27
2.1.1. 概述	27
2.1.2. 公共参数	27
2.1.3. 请求结构	29

29
30
33
33
43
44
46
46
54
60
63
64
65
66
68
72
72
74
75
76
76 76
76
76 78
76 78
76 78 79
76 78 79 80

2.3.4. 手动添加已有实例至集群	- 90
2.3.5. 节点池	- 93
2.3.5.1. 创建节点池	- 93
2.3.5.2. 扩容节点池	- 99
2.3.5.3. 修改节点池配置	100
2.3.5.4. 查询节点池列表	105
2.3.5.5. 查询节点池详情	115
2.3.5.6. 删除节点池	125
2.4. 模板	126
2.4.1. 创建编排模板	126
2.4.2. 删除编排模板	128
2.4.3. 查询所有编排模板	129
2.4.4. 查询编排模板详情	132
2.4.5. 更新编排模板	135
2.5. 组件	136
2.5.1. 安装集群组件	136
2.5.2. 查询集群组件详情	137
2.5.3. 查询集群组件版本	140
2.5.4. 查询集群组件升级状态	142
2.5.5. 升级集群组件	143
2.5.6. 卸载集群组件	144
3.附录	146
3.1. ASAPI网关调用流程说明	146
3.2. 通过ASAPI网关调用专有云API	148
3.2.1. 获取ASAPI的Endpoint	148
3.2.2. 获取ASAPI的SDK包	153
3.3. STS概要	154
3.3.1. 什么是STS	154

开发指南· <mark>目录</mark>	容器服务Kubernet es版
-----------------------	------------------

3.3	.2.	AssumeRole		15	54
-----	-----	------------	--	----	----

▶ 文档版本: 20220915

1.SDK参考

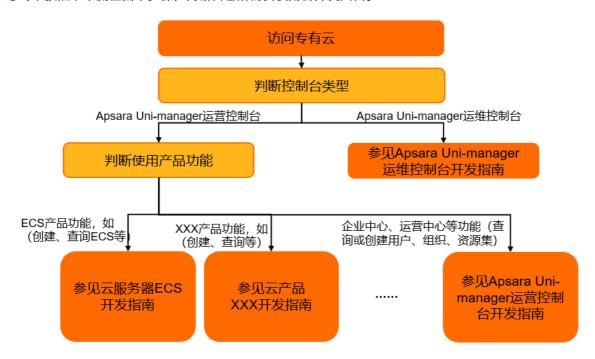
1.1. 快速入门

1.1.1. 开发文档使用说明

本文为您介绍在使用专有云API进行开发工作时,如何查看所需要的产品文档。

使用流程

您可以按照以下流程图中步骤,判断自己所需要使用的开发文档。



前提条件

通过运维人员或运营管理员获取专有云的版本。

操作步骤

您可以通过阿里云专有云文档中心,获取需要的产品文档。

- 1. 通过浏览器打开专有云文档中心。
- 2. 根据专有云的版本单击企业版、敏捷版或一体机标签。

3. 在下拉列表中,选择对应的专有云版本。



- 4. 在左侧导航栏中,选择目标产品名称。
- 5. 在单击目标文档区域的查看详情或下载PDF。

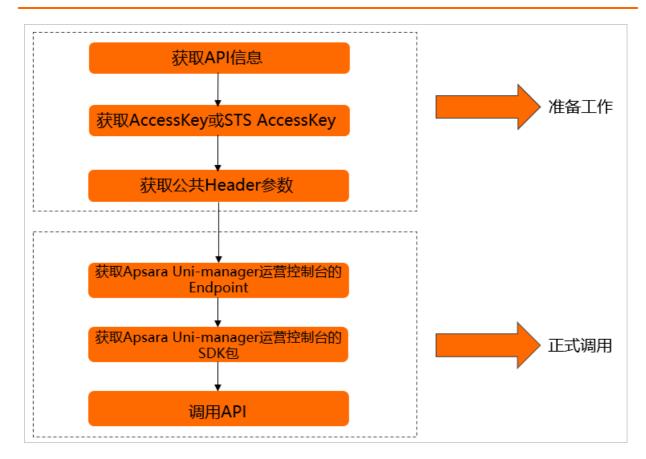
1.1.2. API调用流程说明

阿里云专有云支持通过POP网关或ASAPI网关调用专有云API,为了使您拥有与公有云一致的体验,自企业版V3.16.0开始,专有云API默认使用POP网关方式调用。本文介绍通过POP网关调用专有云API的流程。



历史版本使用ASAPI网关调用专有云API的方式仍然有效,详情请参见ASAPI<mark>网关调用流程说明、获取ASAPI的Endpoint、获取ASAPI的SDK包</mark>。

流程图



调用步骤

1. 获取AccessKey。

获取调用时所需要的AccessKey参数,请参见获取AccessKey或获取STS AccessKey。

? 说明

推荐使用个人AccessKey或STS AccessKey调用Apsara Uni-manager运营控制台及云产品接口。

2. 获取RegionID。

获取调用时所需要的RegionID参数,请参见获取公共Header参数。

3. 获取组织ID与资源集ID。

组织ID需要通过API进行获取,获取方式请参见*Apsara Uni-manager运营控制台开发指南*中的*Get Organi zationList章节*。

资源集ID需要通过API进行获取,获取方式请参见*Apsara Uni-manager运营控制台开发指南*中的*ListReso urceGroup章节*。



4. 获取对应的API参考文档。

您需要查看目标云产品的开发指南,了解API的功能与使用方式,开发指南的使用流程及获取方法请参见开发文档使用说明。

- 5. 获取云产品Endpoint。
- 6. 获取云产品SDK并调用云产品API。

1.2. 准备工作

1.2.1. 登录API与工具控制台

您可以通过Apsara Uni-manager运营控制台登录API与工具控制台。

前提条件

- 登录Apsara Uni-manager运营控制台前,确认您已从部署人员处获取Apsara Uni-manager运营控制台的服务域名地址。
- 推荐使用Chrome浏览器

操作步骤

- 1. 在浏览器地址栏中,输入Apsara Uni-manager运营控制台的服务域名地址,按回车键。
- 2. 输入正确的用户名及密码。

请向运营管理员获取登录控制台的用户名和密码。

- ② 说明 首次登录Apsara Uni-manager运营控制台时,需要修改登录用户名的密码,请按照提示完成密码修改。为提高安全性,密码长度必须为10~32位,且至少包含以下两种类型:
 - 英文大写或小写字母(A~Z、a~z)
 - 阿拉伯数字(0~9)
 - 特殊符号(感叹号(!)、at(@)、井号(#)、美元符号(\$)、百分号(%)等)
- 3. 单击登录。
- 4. 如果账号已激活MFA多因素认证,请根据以下两种情况进行操作:
 - 管理员强制开启MFA后的首次登录:
 - a. 在绑定虚拟MFA设备页面中,按页面提示步骤绑定MFA设备。

容器服务Kubernetes版 开发指南·SDK参考

- b. 按照步骤2重新输入账号和密码,单击登录。
- c. 输入6位MFA码后单击认证。
- 您已开启并绑定MFA:

输入6位MFA码后单击**认证**。

- ② 说明 绑定并开启MFA的操作请参见Apsara Uni-manager运营控制台用户指南中的绑定并开启虚拟MFA设备章节。
- 5. 在页面顶部的菜单栏中,选择产品 > 其他 > API与工具。

1.2.2. 查看API信息

您需要获取产品名称、API名称和API版本,使用SDK时需要填写这些信息。

背景信息

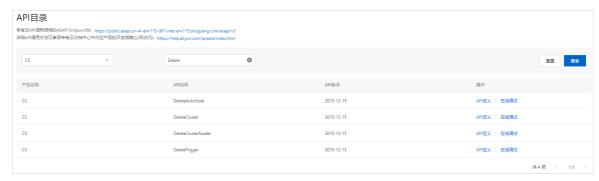
API目录中的API仅供调用参考,具体API的使用方式需要参见云产品的开发指南。如果发生不一致情况,以开发指南支持情况为准。

操作步骤

- 1. 在浏览器地址栏中,输入Apsara Uni-manager运营控制台的服务域名地址,按回车键。
- 2. 输入正确的用户名及密码。

请向运营管理员获取登录控制台的用户名和密码。

- ⑦ 说明 首次登录Apsara Uni-manager运营控制台时,需要修改登录用户名的密码,请按照提示完成密码修改。为提高安全性,密码长度必须为10~32位,且至少包含以下两种类型:
 - 英文大写或小写字母(A~Z、a~z)
 - 阿拉伯数字(0~9)
 - 特殊符号(感叹号(!)、at(@)、井号(#)、美元符号(\$)、百分号(%)等)
- 3. 单击账号登录。
- 4. 在页面顶部的菜单栏中,选择产品 > 其他 > API与工具。
- 5. 在左侧导航栏中选择API目录。
- 6. 在API目录页面,在下拉列表框中选择对应的产品,输入API名称的关键词进行搜索。



7. 查看并记录对应的产品名称、API名称和API版本。

1.2.3. 获取AccessKey

AccessKey支持RAM和STS两种授权模式,在发起调用时选择使用其中一种AccessKey即可。本文将为您介绍如何获取通过RAM授权的AccessKey。

背景信息

AccessKey通过使用AccessKey ID和AccessKey Secret对称加密,实现验证某个请求发送者的身份的功能。 其中AccessKey ID用于标示用户,AccessKey Secret是用户用于加密签名的字符串。

AccessKey的两种授权模式如下:

- 通过RAM授权给第三方请求者的AccessKey ID和AccessKey Secret。
- 通过STS授权给第三方请求者的AccessKey ID、AccessKey Secret和Security Token。获取方式请参见获取 STS AccessKey。

Apsara Uni-manager运营控制台提供了个人AccessKey和组织AccessKey(只有运营管理员和一级组织管理员可以获取组织AccessKey),推荐使用个人账号AccessKey调用Apsara Uni-manager运营控制台及云产品相关接口。如果使用个人账号AccessKey,需要在Header中添加以下限制性参数,否则可能提示权限不足:

● 警告 个人账号AccessKey是由Apsara Uni-manager运营控制台权限体系管控的受限AccessKey,组织AccessKey相对权限较大,需要管理员确认操作的安全性。

参数名称	描述	
x-acs-regionid	地域,如:cn-hangzhou-*。	
x-acs-organizationid	Apsara Uni-manager运营控制台中对应的组织ID。获取 方法可参见 <i>Apsara Uni-manager运营控制台开发指南</i> 中 的 <i>GetOrganizationList</i> 文档。	
x-acs-resourcegroupid	Apsara Uni-manager运营控制台中对应的资源集ID。获取方法可参见 <i>Apsara Uni-manager运营控制台开发指南</i> 中的 <i>ListResourceGroup</i> 文档。	
x-acs-instanceid	进行操作的实例ID,例如,ECS的实例ID可以从ECS列表中查询到。	

获取个人账号AccessKey

获取个人账号AccessKey的方法如下:

- 1. 登录Apsara Uni-manager运营控制台。
- 2. 在系统界面右上角,单击当前登录用户头像,单击个人信息。
- 3. 在阿里云AccessKey区域,您可以查看个人账户的AccessKey信息。



② 说明 Accesskey ID和AccessKey Secret是您访问云资源时的密钥,具有该账号完整的权限,请您妥善保管。

获取组织AccessKey

获取组织AccessKey的方法如下:

- 1. 管理员登录Apsara Uni-manager运营控制台。
- 2. 在顶部菜单栏,单击企业。
- 3. 在左侧导航栏中,选择资源管理 > 组织管理。
- 4. 在组织结构中,单击目标一级组织名称。
- 5. 单击管理Accesskey。
- 6. 在弹出的对话框中,查看组织AccessKey信息。

1.2.4. 获取STS AccessKey

本文为您介绍如何利用STS方式,调用专有云的产品API。

背景信息

阿里云临时安全令牌STS(Security Token Service)是阿里云提供的一种临时访问权限管理服务。您可以选择使用STS方式,调用专有云的产品API。

STS AccessKey由AccessKey ID、AccessKey Secret及SecurityToken三元组组成,使用STS方式调用云产品API时,三元组缺一不可。更多STS说明请参见什么是STS。

操作步骤

- 1. 获取用户角色的 RAM Role 。
 - i. 登录Apsara Uni-manager运营控制台。
 - ii. 单击页面右上角的用户头像,选择个人信息。
 - iii. 单击查看当前角色策略。

在当前选择角色策略对话框中查看 RAM Role 。

2. 获取STS Accesskey三元组。

? 说明

以下代码示例中可变参数均需替换为您实际的相关信息,尖括号(<>)在最终代码中不保留。 涉及替换的可变参数如下:

- <yourRegionID>, 请替换为您实际的地域ID。
- 。 <yourAccessKeyID>, 请替换为您实际的AccessKey ID。
- 。 <yourAccessKeySecret>, 请替换为您实际的AccessKey Secret。
- 。 <yourRoleSessionName>, 请替换为您当前实际的会话名称。
- <yourRoleArn>, 请替换为您实际的RAM Role。
- <sts.aliyuns.com>, 请替换为实际的STS Endpoint。

```
public static void main(String[] args) {
   // 创建DefaultAcsClient实例并初始化。
   // 请替换为您实际的地域ID、AccessKey ID、AccessKey Secret
   DefaultProfile profile = DefaultProfile.getProfile("<yourRegionID>", "<yourAccessKe</pre>
yID>", "<yourAccessKeySecret>");
   HttpClientConfig clientConfig = HttpClientConfig.getDefault();
   clientConfig.setIgnoreSSLCerts(true);
   clientConfig.setProtocolType(ProtocolType.HTTPS);
   profile.setHttpClientConfig(clientConfig);
   IAcsClient client = new DefaultAcsClient(profile);
   String assumeRole = callAssumeRole(client);
   if (StringUtils.isEmpty(assumeRole)) return;
   JSONObject assumeRoleJson = JSONObject.parseObject(assumeRole);
   JSONObject credentials = assumeRoleJson.getJSONObject("Credentials");
   // 获取到STS三元组
   String stsAK = credentials.getString("AccessKeyId");
   String stsSK = credentials.getString("AccessKeySecret");
   String stsToken = credentials.getString("SecurityToken");
private static String callAssumeRole(IAcsClient client) {
   CommonRequest request = new CommonRequest();
   // "sts.aliyuns.com" 请替换为实际的STS Endpoint, 一般格式为sts-vpc.${global:region}.${
global:internet-domain}
   request.setSysDomain("sts.aliyuns.com");
   request.setSysProduct("Sts");
   request.setSysAction("AssumeRole");
   request.setSysVersion("2015-04-01");
   request.setSysMethod(MethodType.POST);
   request.setHttpContentType(FormatType.FORM);
   // 业务参数,设置本次调用Session的别名,"<yourRoleSessionName>"请您替换为实际值
   request.putQueryParameter("RoleSessionName", "<yourRoleSessionName>");
   // 扮演的Role角色ARN,"<yourRoleArn>"请替换为上一步获取到的实际Ram Role ARN
   request.putQueryParameter("RoleArn", "<yourRoleArn>");
   try {
       CommonResponse response = client.getCommonResponse(request);
       return response.getData();
    } catch (ServerException e) {
       System.out.println(e.getErrCode());
       System.out.println(e.getErrMsg());
    } catch (ClientException e) {
       System.out.println(e.getErrCode());
       System.out.println(e.getErrMsg());
   return null;
```

3. 通过STS AccessKey三元组调用云产品的接口,下列代码以ACK的 DescribeUserQuota 接口为示例。

? 说明

以下代码示例中可变参数均需替换为您实际的相关信息,尖括号(<>)在最终代码中不保留。涉及替换的可变参数如下:

- <yourRegionID>, 请替换为您实际的地域ID。
- 。 <yourStsAccesskeyID>, 请替换为您实际的STS Accesskey ID。
- <yourStsAccessKeySecret>, 请替换为您实际的STS Accesskey Secret。
- <yourStsToken>, 请替换为实际STS三元组中的SecurityToken。
- <yourResourceGroupID>, 请替换为您实际的资源集ID。
- <yourOrganizationID>, 请替换为您实际的组织ID。

```
public class DescribeUserQuota {
   public static void main(String[] args) {
       /** use STS Token **/
       DefaultProfile profile = DefaultProfile.getProfile(
                                  // The region ID
           "<your-region-id>",
           "<your-access-key-id>", // The AccessKey ID of the RAM account
            "<your-access-key-secret>", // The AccessKey Secret of the RAM account
            "<your-sts-token>");
                                         // STS Token
       IAcsClient client = new DefaultAcsClient(profile);
       CommonRequest request = new CommonRequest();
       //request.setProtocol(ProtocolType.HTTPS);
       request.setMethod(MethodType.GET);
       request.setDomain("cs.cn-qingdao.aliyuncs.com");
       request.setVersion("2015-12-15");
       request.setUriPattern("/quota");
       request.putHeadParameter("Content-Type", "application/json");
       String requestBody = "" +
               "{}";
       request.setHttpContent(requestBody.getBytes(), "utf-8", FormatType.JSON);
           CommonResponse response = client.getCommonResponse(request);
           System.out.println(response.getData());
        } catch (ServerException e) {
           e.printStackTrace();
        } catch (ClientException e) {
           e.printStackTrace();
```

1.2.5. 获取公共Header参数

调用专有云API时需要提供多个Header参数。本文为您介绍这些参数的含义及其获取方式,并对调用后的一些返回值做出解释。

 容器服务Kubernetes版 开发指南·SDK参考

Header参数描述

名称	描述	
x-acs-regionid	环境Region,获取方式请参见 <mark>获取Region ID</mark> 。	
x-acs-organizationid	Apsara Uni-manager运营控制台上的组织ID,获取方式 请参见 <mark>获取组织ID</mark> 。	
x-acs-resourcegroupid	Apsara Uni-manager运营控制台上的资源集ID,获取方式请参见 <mark>获取资源集ID</mark> 。	
x-acs-instanceid	进行操作的实例ID,获取方式请参见 <mark>获取实例ID。</mark>	

获取Region ID

- 1. 管理员登录Apsara Uni-manager运营控制台。
- 2. 在顶部菜单栏中,单击企业。
- 3. 在左侧导航栏中,选择资源管理>地域管理。
- 4. 单击目标组织名称。
- 5. 在右侧Region列表中查看Region ID。

获取组织ID

组织ID需要通过API进行获取,获取方式请参见*Apsara Uni-manager运营控制台开发指南*中的 *Get Organizat ionList 章节*。

获取资源集ID

资源集ID需要通过API进行获取,获取方式请参见*Apsara Uni-manager运营控制台开发指南*中的*ListResourceGroup章节*。

获取集群ID

集群ID可以通过Apsara Uni-manager运营控制台中的云产品实例列表进行获取:

- 1. 登录Apsara Uni-manager运营控制台。
- 2. 在顶部的菜单栏中,选择产品>弹性计算>容器服务Kubernetes。
- 3. 在左侧导航栏中,单击集群。
- 4. 在集群列表页面单击目标集群的名称。
- 5. 在集群信息页面单击基本信息页签。 在基本信息页签下查看集群ID。

返回参数

对于列表类的云产品资源实例接口,调用后会默认增加如下返回值。

名称	描述
ResourceGroupName	资源集名称。
ResourceGroup	资源集ID。
Depart ment Name	组织名称。
Department	组织ID。

1.2.6. 获取容器服务ACK的Endpoint

本文介绍了如何获取容器服务Endpoint。

背景信息

通过POP网关调用专有云API时需要使用云产品业务域的Endpoint。您可以在阿里云驻场运维工程师处获取。您也可以通过如下方法自行获取Endpoint。

操作步骤

- 1. 在Apsara Uni-manager运维控制台的顶部菜单栏单击**产品运维**,然后在云平台运维区域单击**飞天基础** 运维平台。
- 2. 在飞天基础运维平台页面的左侧导航栏中,单击报表。
- 3. 在全部报表页面搜索 服务注册变量 。



4. 在服务注册变量页面,单击Service旁的 = 图标,输入 acs ,然后单击过滤。



5. 在 acs-acs control 服务的Service Registration列中,单击鼠标右键,然后单击显示更多。

在详情页面,查看的cs.openapi.endpoint值,即 Endpoint 。

1.2.7. 获取容器服务ACK的SDK包

通过POP网关调用专有云API时,需使用相应云产品的SDK。本文为您介绍如何获取阿里云开发工具包(SDK)。

软件开发工具包(SDK)包括代码以及示例,用户可以自行选择语言创建云应用。

目前SDK的语言支持: Java、Python、Go,您可以前往阿里云开发工具包(SDK)中进行选择。

? 说明

推荐使用 Java、Python、Go三种语言的SDK包,使用其他语言的SDK包可能发生不兼容情况。

SDK核心库推荐使用阿里云官网的SDK核心库,其内容所有云产品共用,且适用于专有云。

1.3. Java SDK调用示例

1.3.1. 安装Java SDK

本文介绍如何安装lava SDK。

环境要求

- 最低要求Java 6。
- 安装SDK核心库Core,可以通过POM文件引入的方式来安装。

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>4.5.21</version>
</dependency>
```

安装容器服务SDK

在Maven项目文件中添加以下依赖:

```
<dependency>
  <groupId>com.aliyun</groupId>
   <artifactId>aliyun-java-sdk-cs</artifactId>
   <version>4.8.1</version>
</dependency>
```

1.3.2. 调用SDK

本文介绍如何调用SDK。

操作步骤

1. 初始化配置对象com.aliyuncs.profile.Default Profile。

```
Profile对象包含 your-access-key-id 、 your-access-key-secret 和 your-region-id 。

IClientProfile profile = DefaultProfile.getProfile("<your-region-id>", "<your-access-ke y-id>", "<your-access-key-secret>");
```

2. 实例化一个客户端,从com.aliyuncs.IAcsClient类生成对象client。

```
IAcsClient client = new DefaultAcsClient(profile);
```

3. 创建对应API的request。

类的命名规则为API方法名加上request。

```
DescribeEventsRequest request = new DescribeEventsRequest();
```

4. 设置请求类request的参数。

通过request类的set方法设置必要的信息,即API中必须要提供的信息。

```
request.setcluster_id("test"); // 该参数值为假设值,请您根据实际情况进行填写。
request.settype("test"); // 该参数值为假设值,请您根据实际情况进行填写。
request.setpage_size("1"); // 该参数值为假设值,请您根据实际情况进行填写。
request.setpage_number("1"); // 该参数值为假设值,请您根据实际情况进行填写。
```

5. 通过client对象获得对应request响应response。

```
DescribeEventsResponse response = client.getAcsResponse(request);
System.out.println(JSON.toJSONString(response));
```

6. 调用response中对应的get方法获得返回的参数值。

```
System.out.println(response.getRequestId());
```

- 7. 使用 catch() 处理服务器报错和客户端报错。
 - 服务端报错

```
catch (ServerException e) {
   e.printStackTrace();
}
```

○ 客户端报错

```
catch (ClientException e) {
    System.out.println("ErrCode:" + e.getErrCode());
    System.out.println("ErrMsg:" + e.getErrMsg());
    System.out.println("RequestId:" + e.getRequestId());
}
```

完整代码示例

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.alibaba.fastjson.JSON;
import java.util.*;
import com.aliyuncs.cs.model.v20151215.*;
public class DescribeImages {
   public static void main(String[] args) {
       DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "LTAjVUwKznS*****
", "BNPO1zoNSi484oizGM9fzzwJJ*****");
       IAcsClient client = new DefaultAcsClient(profile);
       // 创建API请求并设置参数。
       DescribeEventsRequest request = new DescribeEventsRequest();
       request.setcluster id("test"); // 该参数值为假设值,请您根据实际情况进行填写。
       request.settype("test"); // 该参数值为假设值,请您根据实际情况进行填写。
       request.setpage size("1"); // 该参数值为假设值,请您根据实际情况进行填写。
       request.setpage number("1"); // 该参数值为假设值,请您根据实际情况进行填写。
       try {
           DescribeEventsResponse response = client.getAcsResponse(request);
           System.out.println(JSON.toJSONString(response));
           // 打印您需要的返回值,此处打印的是此次请求的RequestId。
           System.out.println(response.getRequestId());
       } catch (ServerException e) {
           e.printStackTrace();
       } catch (ClientException e) {
           // 打印错误码。
           System.out.println("ErrCode:" + e.getErrCode());
           System.out.println("ErrMsg:" + e.getErrMsg());
           System.out.println("RequestId:" + e.getRequestId());
```

1.4. Python SDK调用示例

1.4.1. 安装Python SDK

本文介绍如何安装Python SDK。

环境要求

- Python 2.7或3.x
- 安装SDK核心库OpenAPI,使用pip安装包依赖:

```
pip install alibabacloud_tea_openapi
```

安装容器服务SDK

PyPI PIP

```
pip install aliyun-python-sdk-cs==4.8.1
```

PyPI Anaconda

```
conde install aliyun-python-sdk-cs=4.8.1
```

1.4.2. 调用SDK

本文介绍如何调用SDK。

操作步骤

1. 实例化一个客户端,从aliyunsdkcore包中获取AcsClient类实例化对象client。

```
client对象包含 your-access-key-id 、 your-access-key-secret 和 your-region-id 。
```

```
from aliyunsdkcore.client import AcsClient
client = AcsClient(
    ak="<your-access-key-id>",
    secret="<your-access-key-secret>",
    region_id='<region-id>'
)
```

2. 创建对应API的request。

类的命名规则为API方法名加上request。

```
from aliyunsdkcs.request.v20151215.DescribeEventsRequest import DescribeEventsRequest
request = DescribeEventsRequest()
```

3. 设置请求类request的参数。

通过request类的set方法设置必要的信息,即API中必须要提供的信息。

```
// 该参数值为假设值,请您根据实际情况进行填写。
request.set_cluster_id("test")
// 该参数值为假设值,请您根据实际情况进行填写。
request.set_type("test")
// 该参数值为假设值,请您根据实际情况进行填写。
request.set_page_size("1")
// 该参数值为假设值,请您根据实际情况进行填写。
request.set_page_number("1")
```

4. 通过client对象获得对应request响应response。

```
response = client.do_action_with_exception(request)
print(response)
```

- 5. 使用try...except...处理服务器报错和客户端报错。
 - 服务端报错

```
try:
...
except ServerException as e:
   print(e)
```

。 客户端报错

```
try:
    ...
except ClientException as e:
    print(e)
```

完整代码示例

```
#!/usr/bin/env python
#coding=utf-8
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkcs.request.v20151215.DescribeEventsRequest import DescribeEventsRequest
client = AcsClient('<your-access-key-id>', '<your-access-key-secret>', 'cn-hangzhou')
// 创建API请求并设置参数。
request = DescribeEventsRequest()
// 该参数值为假设值,请您根据实际情况进行填写。
request.set cluster id("test")
// 该参数值为假设值,请您根据实际情况进行填写。
request.set_type("test")
// 该参数值为假设值,请您根据实际情况进行填写。
request.set page size("1")
// 该参数值为假设值,请您根据实际情况进行填写。
request.set_page_number("1")
response = client.do action with exception(request)
// 打印您需要的返回值。
# python2: print response
print(str(response, encoding='utf-8'))
```

1.5. Go SDK调用示例

1.5.1. 安装Go SDK

本文介绍如何安装Go SDK。

环境要求

- Go环境版本必须不低于1.10.x
- 安装SDK核心库Core

```
go get -u github.com/aliyun/alibaba-cloud-sdk-go/sdk
```

安装容器服务SDK

使用Go GET进行安装:

```
go get github.com/aliyun/alibaba-cloud-sdk-go/services/CS
```

1.5.2. 调用SDK

本文介绍如何调用SDK。

操作步骤

1. 在代码中引入依赖包。

```
import (
  cs "github.com/aliyun/alibaba-cloud-sdk-go/services/cs"
)
```

2. 实例化一个客户端,从cs.NewClientWithAccessKey类生成对象client。

```
NewClientWithAccessKey对象包含 your-access-key-id 、 your-access-key-secret 和 your-region-id 。

client, err := cs.NewClientWithAccessKey("<your-region-id>", "<your-access-key-id>", "<
```

3. 创建对应API的request。

your-access-key-secret>")

方法的命名规则为Create加上API方法名再加上Request。

```
request := cs.CreateDescribeEventsRequest()
```

4. 设置请求request的参数。

通过request设置必要的信息,即API中必须要提供的信息。

```
// 该参数值为假设值,请您根据实际情况进行填写。
request.cluster_id = "test"
// 该参数值为假设值,请您根据实际情况进行填写。
request.type = "test"
// 该参数值为假设值,请您根据实际情况进行填写。
request.page_size = "1"
// 该参数值为假设值,请您根据实际情况进行填写。
request.page_number = "1"
```

5. 通过client对象获得对应request响应response。

```
response, err := client.DescribeEvents(request)
```

6. 通过err可以处理请求报错。

```
if err != nil {
  fmt.Println(err)
  return
}
```

7. 调用response获得返回的参数值。

以下以requestId为例:

```
fmt.Println(response.RequestId)
```

完整代码示例

```
package main
import (
 "fmt"
 cs "github.com/aliyun/alibaba-cloud-sdk-go/services/cs"
func main() {
 client, err := cs.NewClientWithAccessKey("cn-hangzhou", "<yourAccessKeyID>", "<yourAccess</pre>
KeySecret>")
 // 创建API请求并设置参数。
 request := ecs.CreateDescribeEventsRequest()
 // 该参数值为假设值,请您根据实际情况进行填写。
 request.cluster id = "test"
 // 该参数值为假设值,请您根据实际情况进行填写。
 request.type = "test"
 // 该参数值为假设值,请您根据实际情况进行填写。
 request.page_size = "1"
 // 该参数值为假设值,请您根据实际情况进行填写。
 request.page_number = "1"
 response, err := client.DescribeEvents(request)
 if err != nil {
   fmt.Println(err)
   return
 // 打印您需要的返回值,此处打印的是此次请求的RequestId。
 fmt.Println(response.RequestId)
```

开发指南·集群 API 列表 容器服务Kubernetes版

2.集群 API 列表

2.1. 集群 API 调用方式

2.1.1. 概述

对容器服务 API 接口的调用是通过向容器服务 API 的服务端地址发送 HTTP 请求,并按照接口说明在请求中加入相应请求参数来完成的。根据请求的处理情况,系统会返回处理结果。

- 1. 公共参数
- 2. 请求结构
- 3. 返回参数
- 4. 签名机制

2.1.2. 公共参数

本文列举了容器服务Kubernetes版产品的公共请求头信息,通过ASAPI调用所需的公共参数信息,和公共返回头信息。

公共请求头部

公共请求参数是指每个接口都需要使用到的请求参数。

参数名称	说明	是否必须
Authorization	用于验证请求合法性的认证信息,采用 AccessKeyId:Signature 的形式。	是
Content-Length	RFC 2616中定义的HTTP请求内容长度。	是
Content-Type	RFC 2616中定义的HTTP请求内容类型。	是
Content-MD5	HTTP协议消息体的128-bit MD5散列值转换成BASE64编码的结果。为了防止所有请求被篡改,建议所有请求都附加该信息。	是
Date	请求的构造时间,目前只支持GMT格式。如果与MNS的服务器时间前后差 异超过15分钟将返回本次请求非 法。	是
Host	访问Host值,例 如: <i>diku.aliyuncs.com</i> 。	是
Accept	客户端需要的返回值类型,支 持 <i>application/json</i> 和 <i>application/x</i> <i>ml</i> 。	是

容器服务Kubernetes版 开发指南·集群 API 列表

参数名称	说明	是否必须
x-acs-version	API版本号。目前版本号为 <i>2015-12-</i> <i>15</i> 。	是
x-acs-region-id	地域(Region)指的是ECS实例所在的物理位置。	是
x-acs-signature-nonce	唯一随机数,用于防止网络重放攻 击。您在不同请求间要使用不同的随 机数值。	是
x-acs-signature-method	用户签名方式,目前只支持 <i>HMAC-</i> SHA1。	是

示例

GET /clusters HTTP/1.1 Host: cs.aliyuncs.com Accept: application/json

 $\label{local-agent:cs-sdk-python/0.0.1} $$ (Darwin/15.2.0/x86_64; 2.7.10) $$ x-acs-signature-nonce: f63659d4-10ac-483b-99da-ea8fde61eae3 $$$

Authorization: acs <yourAccessKeyId>:<yourSignature>

x-acs-signature-version: 1.0
Date: Wed, 16 Dec 2015 11:18:47 GMT
x-acs-signature-method: HMAC-SHA1

Content-Type: application/json; charset=utf-8

X-Acs-Region-Id: cn-beijing

Content-Length: 0

通过ASAPI调用所需的公共参数

通过ASAPI调用需要一些额外的公共参数,如下表所示。

参数名称	说明	是否必须
x-ascm-product-ak	由控制台提供的AK。	是
x-ascm-product-name	API产品名。容器服务为cs。	是
x-ascm-product-version	API版本号。目前版本号为2015-12- 15。	是
x-acs-regionid	地域ID。	是
x-acs-organizationid	组织ID。	是
x-acs-resourceGroupId	资源集ID。	是
x-acs-roleid	角色ID。	是

公共返回头部

开发指南·集群 API 列表 容器服务Kubernetes版

您发送的每次接口调用请求,无论成功与否,系统都会返回一个唯一识别码RequestId。

示例

XML示例:

JSON示例:

```
{
    "RequestId": "4C467B38-3910-447D-87BC-AC049166F216"
    /* 返回结果数据 */
}
```

2.1.3. 请求结构

本文主要介绍发起容器服务Open API请求时的结构。

服务地址

● 阿里云容器服务的Open API接入地址格式如下:

```
中心region: cs.${global:internet-domain}单元region: cs.${global:region}.${global:internet-domain}
```

● 通过ASAPI请求的API地址需要增加/roa前缀,如创建集群的地址为/clusters,则需请求/roa/clusters。

通信协议

支持通过HTTP或HTTPS通道进行请求通信。为了获得更高的安全性,推荐您使用HTTPS通道发送请求。

请求方法

使用HTTP的PUT、POST、GET、DELETE等HTTP Method发送不同的请求。

请求参数

每个请求都需要包含公共请求参数和指定操作所特有的请求参数。

请求编码

请求及返回结果都使用UTF-8字符集进行编码。

2.1.4. 返回参数

调用 API 服务后返回数据采用统一格式。返回的 HTTP 状态码为 2xx ,代表调用成功;返回的 HTTP 状态码为 4xx 或 5xx ,代表调用失败。调用成功返回的数据格式主要有 XML 和 JSON 两种,外部系统可以在请求时传入参数来制定返回的数据格式,默认为 XML 格式。

容器服务Kubernetes版 开发指南·集群 API 列表

为了便于您查看,本文档中的返回示例做了格式化处理,实际返回结果是没有换行、缩进等处理的。

2.1.5. 签名机制

本文介绍并演示如何通过签名机制保护容器服务不被非授权用户访问。

签名机制说明

Access Key ID和Access Key Secret 由阿里云官方颁发给访问者(可以通过阿里云官方网站申请和管理),其中Access Key ID用于标识访问者的身份;Access Key Secret是用于加密签名字符串和服务器端验证签名字符串的密钥,必须严格保密,只有阿里云和用户知道。

容器服务会对每个访问的请求进行验证,每个向容器服务提交的请求,都需要在请求中包含签名(Signature)信息。容器服务通过使用Access Key ID和Access Key Secret进行对称加密的方法来验证请求的发送者身份。如果计算出来的验证码和提供的一样即认为该请求是有效的;否则,容器服务将拒绝处理这次请求,并返回HTTP 403错误。

用户可以在HTTP请求中增加授权(Authorization)的Head来包含签名信息,表明这个消息已被授权。

容器服务要求将签名包含在HTTP Header中,格式为 Authorization: acs [Access Key Id]:

[Signature] .

Signature 的计算方法如下:

- VERB 表示HTTP的Method。比如示例中的 PUT 。
- ACCEPT 客户端需要的返回值类型,支持application/json和 application/xml。
- Content-MD5 表示请求内容数据的MD5值。
- Content-Type 表示请求内容的类型。
- Date 表示此次操作的时间,不能为空,目前只支持GMT格式。如果请求时间与CAS服务器时间相差超过15分钟,CAS会判定此请求不合法,并返回400错误。比如示例中的 Thu, 17 Mar 2012 18:49:58 GMT。
- CanonicalizedHeaders 表示HTTP中以 x-acs- 开始的字段组合。
- CanonicalizedResource 表示HTTP所请求资源的URI(统一资源标识符)。比如示例中的 /clusters?nam e=my-clusters&resource=new 。

CanonicalizedHeaders (即以 x-acs- 开头的header) 在签名验证前需要符合以下规范:

- 1. 将所有以 x-acs- 为前缀的HTTP请求头的名字转换成小写字母。比如将 X-ACS-Meta-Name: TaoBao 转换为 x-acs-meta-name: TaoBao 。阿里云规范请求头的名字是大小写不敏感的,建议全部使用小写。
- 2. 如果一个公共请求头的值部分过长,则需要处理其中的 \t 、 \n 、 \r 、 \f 分隔符,将其替换为英文半角的空格。
- 3. 将上一步得到的所有HTTP阿里云规范头按照字典序进行升序排列。

开发指南·集群 API 列表 容器服务Kubernetes版

4. 删除请求头和内容之间分隔符两端出现的任何空格。比如将 x-acs-meta-name: TaoBao, Alipay 转换 为 x-acs-meta-name: TaoBao, Alipay 。

5. 将所有的头和内容用 \n 分隔符分隔拼成最后的CanonicalizedHeaders。

② 说明 CanonicalizedResource 的格式规范: CanonicalizedResource 表示客户想要访问资源的规范描述,需要将子资源和qurey一同按照字典序,从小到大排列并以 为分隔符生成子资源字符串(? 后的所有参数)。

http://cs.aliyuncs.com/clusters?name=my-clusters&resource=new

CanonicalizedResource 应该为:

/clusters?name=my-clusters&resource=new

签名示例

示例概述

您可以通过该示例,了解加签的步骤。

示例使用的accessKeyId和accessKeySecret分别为 access_key_id 和 access_key_secret 。推荐您使用自己的OpenAPI调用程序,来计算下面这个示例的加签串,您自己的加签结果和示例结果。

请求的示例如下:

```
POST http://cs.aliyuncs.com/clusters?param1=value1&param2=value2 HTTP/1.1
Accept-Encoding: identity
Content-Length: 210
Content-MD5: 6U4ALMkKSj0PYbeQSHqgmA==
x-acs-version: 2015-12-15
Accept: application/json
User-Agent: cs-sdk-python/0.0.1 (Darwin/15.2.0/x86 64;2.7.10)
x-acs-signature-nonce: fbf6909a-93a5-45d3-8b1c-3e03a7916799
x-acs-signature-version: 1.0
Date: Wed, 16 Dec 2015 12:20:18 GMT
x-acs-signature-method: HMAC-SHA1
Content-Type: application/json; charset=utf-8
X-Acs-Region-Id: cn-beijing
Authorization: acs <yourAccessKeyId>:<yourSignature>
{"password": "Just$****","instance type": "ecs.m2.medium","name": "my-test-cluster-9708****
","size": 1,"network mode": "vpc","data disk category": "cloud","data disk size": 10,"ecs i
mage id": "m-2531****1"}
```

请求构造过程

计算 Content-Length 和 Content-MD5

Content-Length :body内容的长度。

⑦ 说明 示例body首位没有空格或换行符。

容器服务Kubernetes版 开发指南·集群 API 列表

```
body: {"password": "Just$****","instance_type": "ecs.m2.medium","name": "my-test-cluster-97
08****","size": 1,"network_mode": "vpc","data_disk_category": "cloud","data_disk_size": 10,
"ecs_image_id": "m-25311****"}
Content-Length: 210
```

Content-MD5 : MD5 的计算过程。

处理 CanonicalizedHeaders

```
# 将所有以`x-acs-'开头的头部列出来
x-acs-version: 2015-12-15
x-acs-signature-nonce: ca480402-7689-43ba-acc4-4d2013d9d8d4
x-acs-signature-version: 1.0
x-acs-signature-method: HMAC-SHA1
X-Acs-Region-Id: cn-beijing
# 将请求名字变成小写,去掉每一行首尾的空格,并按照字典序进行排序。删除请求头和内容之间分隔符两端出现的任何空格。
# 注意: 最后一行没有换行符。
x-acs-region-id:cn-beijing
x-acs-signature-method:HMAC-SHA1
x-acs-signature-method:HMAC-SHA1
x-acs-signature-version:1.0
x-acs-version:2015-12-15
```

计算 CanonicalizedResource

示例得到的 CanonicalizedResource , 长度应该为27。

⑦ 说明 第一行行尾有一个 \n 的换行符。

/clusters?param1=value1¶m2=value2

计算 Signature

组装 SignatureString 。示例中的加签字符串的长度为307。除最后一行外,每一行行尾均有一个 \n 的换行符。

开发指南·集群 API 列表 容器服务Kubernet es版

```
POST
application/json
6U4ALMkKSj0PYbeQSHqgmA==
application/json; charset=utf-8
Wed, 16 Dec 2015 12:20:18 GMT
x-acs-region-id:cn-beijing
x-acs-signature-method:HMAC-SHA1
x-acs-signature-nonce:fbf6909a-93a5-45d3-8b1c-3e03a7916799
x-acs-signature-version:1.0
x-acs-version:2015-12-15
/clusters?param1=value1&param2=value2
```

计算 Signature

- # 使用 accessKeySecret 来对加签字符串进行加密,其中示例使用的accessKeySecret是access_key_secret。hmac-sha1(SignatureString): fee03d405e421ebaf514adec881038c4b313584d
- # 类似于Content-MD5的计算方式,将得到的加密串转化成字节数组。
- # 将得到的字符数组做一个base64转换。得到最后的签名串。

base64(bytes(hmac-sha1(SignatureString))): ZmVlMDNkNDA1ZTQyMWViYWY1MTRhZGVjODgxMDM4YzRiMzEz
NTq0ZA==

Signature: ZmVlMDNkNDA1ZTQyMWViYWY1MTRhZGVjODgxMDM4YzRiMzEzNTg0ZA==

完成

经过以上的处理,添加一些其他头部信息,最终构成的HTTP请求如下所示。

```
POST http://cs.aliyuncs.com/clusters?param1=value1&param2=value2 HTTP/1.1
Accept-Encoding: identity
Content-Length: 210
Content-MD5: 6U4ALMkKSj0PYbeQSHqgmA==
x-acs-version: 2015-12-15
Accept: application/json
User-Agent: cs-sdk-python/0.0.1 (Darwin/15.2.0/x86 64;2.7.10)
x-acs-signature-nonce: fbf6909a-93a5-45d3-8b1c-3e03a7916799
x-acs-signature-version: 1.0
Date: Wed, 16 Dec 2015 12:20:18 GMT
x-acs-signature-method: HMAC-SHA1
Content-Type: application/json; charset=utf-8
X-Acs-Region-Id: cn-beijing
Authorization: acs <yourAccessKeyId>:<yourSignature>
","size": 1,"network mode": "vpc","data disk category": "cloud","data disk size": 10,"ecs i
mage_id": "m-25311****"}
```

2.2. 集群

2.2.1. 创建Kubernetes集群

调用CreateCluster创建一个标准Kubernetes专有版集群。

请求头

容器服务Kubernet es版 开发指南·集群 API 列表

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

POST /clusters HTTP/1.1

请求参数

名称	类型	位置	是否必选	示例值	描述
cluster_type	String	Body	否	Kubernetes	集群类型。取值: Kubernetes : Kubernetes专有版 或边缘专有版
name	String		否	cluster-demo	集群名称。 命名规则:由数字、汉字、英文字符或 短划线(-)组成,长度范围1~63个字 符,且不能以短划线(-)开头。
region_id	String	Body	否	cn-beijing	集群所在的地域ID。
disable_rollbac k	Boolea n	Body	否	true	集群创建失败是否回滚。取值: ● true : 当集群创建失败时,进行回滚操作。 ● false : 当集群创建失败时,不进行回滚操作。 默认值: true 。
timeout_mins	Long	Body	否	60	集群创建超时时间,单位:分钟。 默认值: 60 。
kubernetes_ver sion	String	Body	否	1.16.9-aliyun.1	集群版本,与Kubernetes社区基线版本保持一致。建议选择最新版本,若不指定,默认使用最新版本。目前您可以在ACK控制台创建两种最新版本的集群。您可以通过API创建其他Kubernetes版本集群。

开发指南·集群 API 列表 容器服务Kubernet es版

名称	类型	位置	是否必选	示例值	描述
snat_entry	Boolea n	Body	否	true	为专有网络配置SNAT。取值: true : 将为您创建NAT网关并自动配置SNAT规则,若您集群内的节点、应用等需要访问公网需要设置为 true 。 false : 不为您创建NAT网关及SNAT规则。这种模式下,集群内节点及应用将不能访问公网。 默认值: true 。
endpoint_publi c_access	Boolea n	Body	否	true	是否开启公网访问。通过EIP暴露API Server,实现集群公网访问。 true : 开启公网访问。 false : 不开启公网访问。选择不开放时,则无法通过外网访问集群API Server。 默认值: false 。
ssh_flags	Boolea n	Body	否	true	是否开放公网SSH登录。 • true : 表示开放。 • false : 表示不开放。 默认值: false 。
cloud_monitor_ flags	Boolea n	Body	否	true	集群是否安装云监控插件。取值: true : 安装云监控插件。 false : 不安装云监控插件。 默认值: false 。
deletion_protec tion	Boolea n	Body	否	true	集群删除保护,防止通过控制台或API 误删除集群。取值: true : 启用集群删除保护,将不能通过控制台或API删除集群。 false : 不启用集群删除保护,则能通过控制台或API删除集群。 默认值: false 。

容器服务Kubernet es版 开发指南·集群 API 列表

名称	类型	位置	是否必选	示例值	描述
node_cidr_mas k	String	Body	否	26	节点IP数量,通过指定网络的CIDR来确定IP的数量,只对于Flannel网络类型集群生效。 默认值:26。
user_ca	String	Body	否	BEGIN CERTIFICATE -****	自定义集群CA。
proxy_mode	String	Body	否	ipvs	kube-proxy代理模式 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,但是性能一般,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。 默认值: ipvs 。
tags.N.key	String	Body	否	key	标签key值。
tags.N.value	String	Body	否	value	标签value值。
addons.N.name	String	Body	否	nginx-ingress- controller	插件名称。
addons.N.confi g	String	Body	否	{\"IngressSlbNe tworkType\":\" internet\"}	插件配置参数。
addons.N.disab led	Boolea n	Body	否	false	是否禁止默认安装, 取值: ● true : 禁止默认安装 ● false : 不禁止默认安装

名称	类型	位置	是否必选	示例值	描述
os_type	String	Body	否	Linux	操作系统平台类型。取值: ● Windows ● Linux 默认值: Linux 。
platform	String	Body	否	Cent OS	操作系统发行版。取值: CentOS : Centos镜像 AliyunLinux : AliyunLinux镜像 WindowsCore : Windows镜像 CentOS 7.6 : CentOS 7.6镜像 ARM CentOS : CentOS 7.7镜像 NeoKylin : 麒麟操作系统 默认值: AliyunLinux 。
user_data	String	Body	否	lyEvdXNyL2Jpbi 9iYXNoCmVjaG 8glkhlbGxvIEFD* ***	自定义节点数据。更多详情,请参见云服务器ECS的实例自定义数据章节。
node_port_ran ge	String	Body	否	30000~32767	节点服务端口,可选端口范围: 30000~65535。 默认值: 30000~32767 。
login_password	String	Body	否	Hello@1234	SSH登录密码,和 key_pair 二选 一。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)。
cpu_policy	String	Body	否	none	节点CPU管理策略。当集群版本在 1.12.6及以上时支持以下两种策略: ● static : 允许为节点上具有某些资源特征Pod增强其CPU亲和性和独占性。 ● none :表示启用现有的默认CPU亲和性方案。 默认值: none 。

名称	类型	位置	是否必选	示例值	描述
taints.N.effect	String	Body	否	key	污点key值。
taints.N.key	String	Body	否	value	污点value值。
taints.N.value	String	Body	否	NoSchedule	污点生效策略。
master_count	Long	Body	否	3	Master节点数量,可选值 3 。 默认值: 3 。
master_system _disk_category	String	Body	否	cloud_ssd	Master节点系统盘类型,取值: ● cloud_efficiency : 高效云盘。 ● cloud_ssd : SSD云盘。 ● cloud_essd : ESSD云盘。
master_system _disk_size	Long	Body	否	120	Master节点系统盘类型,取值范围 40~500,单位: GiB。 默认值: 120 。
master_system _disk_snapshot _policy_id	String	Body	否	sp- 2zej1nogjvovnz 4z****	集群Master节点自动快照备份策略。
master_system _disk_performa nce_level	String	Body	否	PL1	Master节点系统盘磁盘性能,只针对 ESSD磁盘生效。
num_of_nodes	Long	Body	否	3	Worker节点数。范围是0~100。
worker_system _disk_category	String	Body	否	cloud_efficienc y	Worker节点系统盘类型,取值: ● cloud_efficiency : 高效云盘。 ● cloud_ssd: SSD 云盘。 默认值: cloud_ssd 。

名称	类型	位置	是否必选	示例值	描述
worker_system _disk_size	Long	Body	否	120	Worker节点系统盘大小,单位为GiB。 取值范围:40~500。 该参数的取值必须大于或者等于 max{40, ImageSize}。 默认值:120。
worker_system _disk_snapshot _policy_id	String	Body	否	sp- 2zej1nogjvovnz 4z****	集群Worker节点系统盘快照备份策 略。
worker_system _disk_performa nce_level	String	Body	否	PL1	集群Worker节点磁盘性能,只对ESSD 类型磁盘生效。
worker_data_di sks.N.auto_sna pshot_policy_id	String	Body	否	sp- 2zej1nogjvovnz 4z****	集群Worker节点数据盘自动备份策略 ID。
worker_data_di sks.N.category	String	Body	否	cloud_essd	集群Worker节点数据盘类型。
worker_data_di sks.N.encrypted	String	Body	否	true	集群Worker节点数据盘是否加密。
worker_data_di sks.N.performa nce_level	String	Body	否	PL1	集群Worker节点数据盘磁盘性能等 级,仅对ESSD磁盘生效。
worker_data_di sks.N.size	Long	Body	否	120	集群Worker节点数据盘大小。
vpcid	String	Body	否	vpc- 2zeik9h3ahvv2z z95****	集群使用的专有网络,创建集群时必须 为集群提供。

名称	类型	位置	是否必选	示例值	描述
					自动创建企业级安全组, 当 security_group_id 为空的时 生效。
is_enterprise_se curity_group	Boolea n	Body	否	true	② 说明 使用普通安全组时, 集群内节点与Terway Pod数量之 和不能超过2000。所以创建 Terway网络类型集群时,建议使 用企业安全组。
					◆ true : 创建并使用企业级安全组。◆ false : 不使用企业级安全组。默认值: true 。
	String Body 否 172.20.0.0/16		Pod网络地址段,不能和VPC网段冲突。当选择系统自动创建VPC时,默认使用172.16.0.0/16网段。		
container_cidr		String Body	否	172.20.0.0/16	② 说明
					当创建Flannel网络类型的集群时, 该字段为必填。当创建Trrway网络类型的集群时, 该字段不需要填。
					Service网络地址段,可选范围: 10.0.0.0/16~24,172.16~31.0.0/16~ 24,192.168.0.0/16~24
service_cidr	String	Body	否 1	172.21.0.0/20	不能与VPC网段10.1.0.0/21及VPC内已有Kubernetes集群使用的网段重复,创建成功后不能修改。 默认使用172.19.0.0/20网段。
resource_group _id	String	Body	否	rg- acfm3mkrure** **	资源组ID。
image_id	String	Body	否	m- bp16z7xko3vvv 8gt****	节点自定义镜像,默认使用系统镜像。 当选择自定义镜像时,将取代默认系统 镜像。

名称	类型	位置	是否必选	示例值	描述
cluster_domain	String	Body	否	cluster.local	集群本地域名。 命名规则:域名由半角句号(.)分隔的一个或多个部分构成,每个部分最长为63个字符,可以使用小写字母、数字和短划线(-),且首尾必须为小写字母或数字。
profile	String	Body	否	Default	集群标识。取值: Default : Kubernetes专有版 Edge : 边缘专有版
master_vswitch _ids	List <str ing></str 	Body	否	["vsw- 2ze3ds0mdip0 hdz8i****"]	虚拟交换机ID。
master_instanc e_types	List <str ing></str 	Body	否	["ecs.n4.xlarge"]	Master节点实例类型。关于ECS实例规格的具体介绍,请参见云服务器用户指南的实例规格章节。
worker_vswitch _ids	List <str ing></str 	Body	否	["vsw- 2ze3ds0mdip0 hdz8i****"]	虚拟交换机ID列表,取值范围: 1~20。为保证高可用性,建议选择不 同可用区的虚拟交换机。
					Worker节点实例规格,至少要指定一个实例规格。更多信息,请参见云服务器ECS指南的实例规格章节。
worker_instanc e_types	List <str ing></str 	Body	否 ["ecs.	["ecs.n4.large"]	② 说明 实例规格优先级随着 在数据中的位置增大依次降低。 当无法根据优先级较高的实例规 格创建出实例时,会自动选择下 一优先级的实例规格来创建实 例。
pod_vswitch_id s	List <str ing></str 	Body	否	["vsw- 2vc9mwkxbvud p25do****"]	创建Terway网络类型的集群时,需要 为Pod指定单独的虚拟交换机。

名称	类型	位置	是否必选	示例值	描述
runtime	Struct	Body	否	{"name": "docker", "version": "19.03.5"}	容器运行时,支持 containerd 、 docker 、 Sandboxed-Container.runv 三种 运行时,默认为 docker 。runtime 包括以下2个信息: Iname :容器运行时名称。 version :容器运行时版本。
name	String		否	docker	容器运行时名称
version	String		否	version 19.03.15	容器运行时版本

返回数据

名称	类型	示例值	描述
cluster_id	String	cb95aa626a47740af bf6aa099b650****	集群ID。
request_id	String	687C5BAA-D103- 4993-884B- C35E4314A1E1	请求ID。
task_id	String	T- 5a54309c80282e39e a00002f	任务ID。

示例

请求示例

```
POST /clusters HTTP/1.1
公共请求头
{
    "cluster\ type": "Kubernetes",
    "container\ cidr": "172.20.0.0/16",
   "instances": i-2ze4zxnm36vq00xn****,
   "login\ password": "Hello@1234",
    "master\_instance\_types": ["ecs.n4.xlarge"],
    "master\_system\_disk\_category": "cloud_ssd",
   "master\ system\ disk\ size": "120",
    "master\ vswitch\ ids": vsw-2ze3ds0mdip0hdz8i****,
    "name": "cluster-demo",
    "num\ of\ nodes": "3",
   "region\ id": "cn-beijing",
   "service\ cidr": "172.21.0.0/20",
    "vpcid": "vpc-2zeik9h3ahvv2zz95****",
    "worker\ instance\ types": ["ecs.n4.large"],
    "worker\ system\ disk\ category": "cloud efficiency",
    "worker\ system\ disk\ size": "120",
    "worker\_vswitch\_ids": vsw-2ze3ds0mdip0hdz8i****
```

正常返回示例

XML 格式

```
<cluster_id>cb95aa626a47740afbf6aa099b650****</cluster_id>
<task_id>T-5a54309c80282e39ea00002f</task_id>
<request_id>687C5BAA-D103-4993-884B-C35E4314A1E1</request_id>
```

JSON 格式

```
{
   "cluster_id": "cb95aa626a47740afbf6aa099b650****",
   "task_id": "T-5a54309c80282e39ea00002f",
   "request_id": "687C5BAA-D103-4993-884B-C35E4314A1E1"
}
```

2.2.2. 删除集群

调用DeleteCluster,根据集群ID删除集群实例,并释放集群所有节点资源。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

```
DELETE /clusters/[ClusterId] HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c905d1364c2dd 4b6284a3f4179 0c4****	集群ID。
retain_all_resou rces	Boolea n	Query	否	false	是否保留所有资源。如果设置该值为 true ,则会忽略 retain_resources。 true : 保留所有资源。 false : 不保留所有资源。 默认值: false 。
keep_slb	Boolea n	Query	否	false	是否保留SLB,取值: true : 保留创建的SLB资源。 false : 不保留创建的SLB资源。 默认值: false 。
retain_resource s	List <str ing></str 	Query	否	["sg- 2ze6ess9kho6f dn9****"]	资源名称。

示例

请求示例

正常返回示例

JSON 格式

HTTP/1.1 200 OK

2.2.3. 获取集群Kubeconfig接口

调用DescribeClusterUserKubeconfig返回包含当前登录用户身份信息的Kubernetes集群访问Kubeconfig。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

GET /k8s/[ClusterId]/user_config HTTP/1.1

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c5b5e80b0b64 a4bf6939d2d8f bbc5****	集群ID。
PrivatelpAddres s	Boolea n	Query	否	true	是否获取内网连接配置。取值: ture : 仅获取内网连接凭据。 false : 仅获取公网连接凭据。 默认值: false 。
TemporaryDura tionMinutes	Long	Query	否	15	临时kubeconfig有效期,单位:分钟。取值范围: 15 (15分钟)~4320 (3天)。 ● 当设置该参数时,将返回具有指定有效期限的集群访问配置,具体过期时间通过返回的 expiration 字段的值确定。 ● 当不设置该参数时,将返回长期有效的集群访问配置。

返回数据

名称	类型	示例值	描述
config	String	apiVersion: v1****	集群访问配置。
expiration	String	2024-03- 10T09:56:17Z	Kubeconfig的过期时间。格式:RFC3339格式的UT C时间。

示例

请求示例

```
GET /k8s/[ClusterId]/user_config HTTP/1.1
公共请求头
{
    "ClusterId": "c5b5e80b0b64a4bf6939d2d8fbbc5****"
}
```

正常返回示例

XML 格式

```
<expiration>2024-03-10T09:56:17Z</expiration>
<config>apiVersion: v1****</config>
```

JSON 格式

```
{
   "expiration": "2024-03-10T09:56:17Z",
   "config": "apiVersion: v1****"
}
```

2.2.4. 查询

2.2.4.1. 查询所有集群列表

调用DescribeClustersV1查看您在ACK中创建的所有的集群信息。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

```
GET /api/v1/clusters HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
name	String	Query	否	cluster-demo	集群名称。 命名规则:由数字、汉字、英文字符或 短划线(-)组成,长度范围1~63个字 符,且不能以短划线(-)开头。
cluster_type	String	Query	否	Kubernetes	集群类型,取值: Kubernetes : Kubernetes专有版 或边缘专有版。
page_size	Long	Query	否	10	每页显示的记录数。
page_number	Long	Query	否	3	总页数。

返回数据

名称	类型	示例值	描述
clusters	Array of cluster		集群信息列表。
Department	Integer	3	组织或部门ID,当前集群所属于的部组织或部门ID。
DepartmentNa me	String	acs-test	组织或部门名称,当前集群所属于的组织或部门ID。
RMRegionId	String	cn-wl-env115-d01	集群所在的地域ID。
RegionId	String	cn-beijing	集群所在地域ID。
ResourceGroup	Integer	58	资源集ID。
ResourceGroup Name	String	acr-test	资源集名称。
cluster_id	String	c3fb96524f9274b44 95df0f12a6b5****	集群ID。
cluster_type	String	Kubernetes	集群类型,取值: Kubernetes : Kubernetes专有版或边缘专有版。
created	String	2020-08- 20T10:51:29+08:00	集群创建时间。
current_version	String	1.16.9-aliyun.1	集群当前版本。
deletion_prote ction	Boolean	true	集群删除保护,防止通过控制台或API误删除集群。取值: true : 启用集群删除保护,将不能通过控制台或API删除集群。 false : 不启用集群删除保护,则能通过控制台或API删除集群。
docker_version	String	19.03.5	集群Docker版本。

名称	类型	示例值	描述
external_loadb alancer_id	String	lb- 2vcrbmlevo6kjpgch* ***	集群中Ingress SLB实例。 默认实例规格:性能保障型(slb.s1.small)。
init_version	String	1.16.9-aliyun.1	集群版本,与Kubernetes社区基线版本保持一致。建议选择最新版本,若不指定,默认使用最新版本。 目前ACK控制台提供两种最新Kubernetes版本的集群。您可以通过API创建其他Kubernetes版本集群。
maintenance_w indow	Struct		集群维护窗口。
duration	String	3h	维护时长。取值范围1~24,单位为小时。 默认值:3h。
enable	Boolean	false	是否开启维护窗口。默认值false。
maintenance_ti me	String	03:00:00Z	维护起始时间。Golang标准时间格 式"15:04:05Z"。
weekly_period	String	Monday,Thursday	维护周期。取值范围为:Monday~Sunday,多个值用半角逗号(,)分隔。 默认值: Thursday。
master_url	String	{\"api_server_endpoi nt\":\"\",\"intranet_ api_server_endpoint \":\"https://192.16 8.0.251:6443\"}	集群API Server访问地址,包含内网访问地址以 及公网访问地址。
meta_data	String	{\"api_server_endpoi nt\":\"\",\"intranet_ api_server_endpoint \":\"https://192.16 8.0.251:6443\"}	集群元数据信息。
name	String	cluster-demo	集群名称。 命名规则:由数字、汉字、英文字符或短划线 (-)组成,长度范围1~63个字符,且不能以短 划线(-)开头。

名称	类型	示例值	描述
network_mode	String	vpc	集群网络模式。取值: classic : 经典网络。 vpc : 专有网络。 overlay : 叠加网络。 calico : Calico网络。
next_version	String	1.18.8-aliyun.1	集群可升级版本。
private_zone	Boolean	false	集群Private Zone配置。取值: true : 启用Private Zone。 false : 不启用Private Zone。
profile	String	Default	集群标识,取值: ■ Edge : 边缘托管版集群。 ■ Default : 非边缘托管版集群。
region_id	String	cn-beijing	集群所在地域ID。
resource_group _id	String	rg- acfmyvw3wjm****	集群资源组ID。
security_group_ id	String	sg- 2vcgwsrwgt5mp0yi* ***	集群安全组ID。
size	Integer	5	集群当前节点数量,包含Master节点以及 Worker节点。
state	String	running	集群运行状态,取值: initial : 集群创建中。 failed : 集群创建失败。 running : 集群运行中。 updating : 集群升级中。 updating_failed : 集群升级失败。 scaling : 集群伸缩中。 stopped : 集群已经停止运行。 deleting : 集群删除中。 deleted : 集群已经被删除。

名称	类型	示例值	描述
subnet_cidr	String	172.21.0.0/16	Pod网络地址段,必须是有效的私有网段,即以下网段及其子网: 10.0.0.0/8 172.16-31.0.0/12-16 192.168.0.0/16 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。
tags	Array of tag		集群资源标签。
key	String	tier	标签key值。
value	String	backend	标签value值。
updated	String	2020-09- 16T11:09:55+08:00	集群更新时间。
vpc_id	String	vpc- 2vcg932hsxsxuqbgl* ***	集群专有网络ID,创建集群时必传。
vswitch_id	String	vsw- 2vc41xuumx5z2rdm a****,vsw- 2vc41xuumx5z2rdm a****	虚拟交换机ID,创建集群时可选择1~3个虚拟交换机。为保证集群高可用,建议选择不同可用区的虚拟交换机。
worker_ram_rol e_name	String	KubernetesWorkerR ole-ec87d15b-edca- 4302-933f- c8a16bf0****	Worker RAM角色名称,授权ECS实例为集群 Woker节点。
zone_id	String	cn-beijing-b	集群所属可用区ID。
page_info	Struct		分页信息。
page_number	Integer	3	分页数。
page_size	Integer	20	单页大小。
total_count	Integer	50	结果总数。

示例

请求示例

GET /api/v1/clusters HTTP/1.1

公共请求头

正常返回示例

XML 格式

```
<page info>
   <page number>3</page number>
   <total count>50</total count>
   <page size>20</page size>
</page info>
<clusters>
   <private zone>false</private zone>
   <resource group id>rg-acfmyvw3wjm****</resource group id>
   <vpc id>vpc-2vcg932hsxsxuqbgl****</vpc id>
   <network mode>vpc</network mode>
   <security group id>sg-2vcqwsrwqt5mp0yi****</security group id>
   <cluster type>Kubernetes</cluster type>
   <docker version>19.03.5</docker version>
   <next version>1.18.8-aliyun.1/next version>
   <zone id>cn-beijing-b</zone id>
   <Department>3</Department>
   <cluster id>c3fb96524f9274b4495df0f12a6b5****/cluster id>
   <external loadbalancer id>lb-2vcrbmlevo6kjpgch****</external loadbalancer id>
   <vswitch id>vsw-2vc41xuumx5z2rdma****,vsw-2vc41xuumx5z2rdma****/vswitch id>
   <meta data>{\"api server endpoint\":\"\",\"intranet api server endpoint\":\"https://192
.168.0.251:6443\"}</meta data>
   <RMRegionId>cn-wl-env115-d01/RMRegionId>
   <state>running</state>
   <ResourceGroup>58</ResourceGroup>
   <init version>1.16.9-aliyun.1</init version>
   <deletion protection>true</deletion protection>
   <created>2020-08-20T10:51:29+08:00</created>
   file>Default
   <subnet cidr>172.21.0.0/16/subnet cidr>
   <master url>{\"api server endpoint\":\"\",\"intranet api server endpoint\":\"https://19
2.168.0.251:6443\"}</master url>
   <region id>cn-beijing</region id>
   <current version>1.16.9-aliyun.1/current version>
   <size>5</size>
   <name>cluster-demo</name>
   <DepartmentName>acs-test/DepartmentName>
   <RegionId>cn-beijing</RegionId>
   <worker ram role name>KubernetesWorkerRole-ec87d15b-edca-4302-933f-c8a16bf0****</worker</pre>
_ram_role_name>
   <updated>2020-09-16T11:09:55+08:00</updated>
   <ResourceGroupName>acr-test</ResourceGroupName>
   <tags>
       <value>backend</value>
       <key>tier</key>
   <maintenance window>
       <duration>3h</duration>
       <weekly_period>Monday,Thursday</weekly_period>
       <enable>false</enable>
       <maintenance_time>03:00:00Z</maintenance_time>
   </maintenance window>
</clusters>
```

JSON 格式

```
"page info": {
        "page number": 3,
        "total count": 50,
        "page size": 20
    "clusters": {
        "private zone": false,
        "resource_group_id": "rg-acfmyvw3wjm****",
        "vpc id": "vpc-2vcg932hsxsxuqbgl****",
        "network mode": "vpc",
        "security group id": "sg-2vcgwsrwgt5mp0yi****",
        "cluster type": "Kubernetes",
        "docker version": "19.03.5",
        "next version": "1.18.8-aliyun.1",
        "zone_id": "cn-beijing-b",
        "Department": 3,
        "cluster id": "c3fb96524f9274b4495df0f12a6b5****",
        "external loadbalancer id": "lb-2vcrbmlevo6kjpgch****",
        "vswitch id": "vsw-2vc41xuumx5z2rdma****, vsw-2vc41xuumx5z2rdma****",
        "meta data": "\{\ server endpoint\":\\\",\\"intranet api server endpoin
t\\\":\\\"https://192.168.0.251:6443\\\"}",
        "RMRegionId": "cn-wl-env115-d01",
        "state": "running",
        "ResourceGroup": 58,
        "init version": "1.16.9-aliyun.1",
        "deletion protection": true,
        "created": "2020-08-20T10:51:29+08:00",
        "profile": "Default",
        "subnet cidr": "172.21.0.0/16",
        "master url": "{\\\"api server endpoint\\\":\\\"\\\"intranet api server endpoi
nt\\\":\\\"https://192.168.0.251:6443\\\"}",
        "region id": "cn-beijing",
        "current version": "1.16.9-aliyun.1",
        "size": 5,
        "name": "cluster-demo",
        "DepartmentName": "acs-test",
        "RegionId": "cn-beijing",
        "worker ram role name": "KubernetesWorkerRole-ec87d15b-edca-4302-933f-c8a16bf0****"
        "updated": "2020-09-16T11:09:55+08:00",
        "ResourceGroupName": "acr-test",
        "tags": {
           "value": "backend",
           "key": "tier"
        "maintenance window": {
           "duration": "3h",
            "weekly period": "Monday, Thursday",
            "enable": false,
           "maintenance time": "03:00:00Z"
```

}

2.2.4.2. 查询集群信息

调用DescribeClusterDetail,根据集群ID查看集群的详细信息。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

GET /clusters/[ClusterId] HTTP/1.1

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	cdde1f21ae22e 483ebcb068a6e b7f****	集群ID。

返回数据

名称	类型	示例值	描述
asapiSuccess	Boolean	true	接口数据是否从ASCM正确返回。
cluster_id	String	c82e6987e29614511 82edacd74faf****	集群ID。
cluster_type	String	Kubernetes	集群类型,取值: Kubernetes : Kubernetes专有版或边缘专有版集群。
created	String	2019-11- 25T15:50:20+08:00	集群创建时间。
current_version	String	1.16.6-aliyun.1	集群当前版本。

名称	类型	示例值	描述
deletion_prote ction	Boolean	true	集群删除保护,防止通过控制台或API误删除集群。取值: true : 启用集群删除保护,将不能通过控制台或API删除集群。 false : 不启用集群删除保护,则能通过控制台或API删除集群。
docker_version	String	19.03.5	集群中的Docker版本。
eagleEyeTraceI d	String	0a730954164127774 94927508d004e	Trace ID。
external_loadb alancer_id	String	lb- 2zehc05z3b8dwiifh* ***	集群Ingress LB实例ID。
init_version	String	1.16.6-aliyun.1	集群初始化版本。
maintenance_w indow	Struct		集群维护窗口。
duration	String	3h	维护时长。取值范围1~24,单位为小时。 默认值: 3h。
enable	Boolean	false	是否开启维护窗口。默认值false。
maintenance_ti me	String	03:00:00Z	维护起始时间。Golang标准时间格 式"15:04:05Z"。
weekly_period	String	Monday,Thursday	维护周期。取值范围为:Monday~Sunday,多个值用逗号分隔。 默认值: Thursday。
master_url	String	{\"intranet_api_serv er_endpoint\":\"htt ps://192.168.0.251: 6443\"***}	集群访问地址,包含内网访问地址及公网访问地址。
meta_data	String	\"Addons\":***	集群元数据信息。

名称	类型	示例值	描述
name	String	cluster-demo	集群名称。 命名规则:由数字、汉字、英文字符或短划线 (-)组成,长度范围1~63个字符,且不能以短 划线(-)开头。
network_mode	String	vpc	集群网络模式,取值: classic : 经典网络。 vpc : 专有网络。 overlay : 叠加网络。 calico : Calico网络。 默认值: vpc 。
next_version	String	1.18.8-aliyun.1	集群可升级版本。
private_zone	Boolean	false	集群Private Zone配置。 true : 启用Private Zone。 false : 不启用Private Zone。
profile	String	Default	面向场景时的集群类型: ● Default : Kubernetes专有版集群。 ● Edge : 边缘专有版集群。
region_id	String	cn-beijing	集群所在地域ID。
resource_group _id	String	rg- acfmyvw3wjm****	集群资源组ID。
security_group_ id	String	sg-25yq****	集群安全组ID。
size	Integer	5	集群节点数,包含Master节点及Worker节点。

名称	类型	示例值	描述
state	String	running	集群运行状态,取值: initial : 集群创建中。 failed : 集群创建失败。 running : 集群运行中。 updating : 集群升级中。 updating_failed : 集群升级失败。 scaling : 集群中缩中。 stopped : 集群已经停止运行。 deleting : 集群删除中。 deleted : 集群已经被删除。 delete_failed : 集群删除失败。
subnet_cidr	String	172.20.0.0/16	Pod网络地址段,必须是有效的私有网段,即以下网段及其子网: 10.0.0.0/8 172.16-31.0.0/12-16 192.168.0.0/16 不能与VPC及VPC内已有Kubernetes集群使用的网段重复,创建成功后不能被修改。
tags	Array of tag		集群资源标签。
key	String	key	标签key值。
value	String	value	标签value值。
updated	String	2020-01- 13T23:01:03+08:00	集群更新时间。
vpc_id	String	vpc- 2zecuu62b9zw7a7q n****	集群专有网络ID,创建集群时的必填参数。
vswitch_id	String	vsw- 2zete8s4qocqg0mf 6****,vsw- 2zete8s4qocqg0mf 6****	虚拟交换机ID,创建集群时可选择1~3个虚拟交换机。为保证集群高可用,建议选择不同可用区的虚拟交换机。

名称	类型	示例值	描述
worker_ram_rol e_name	String	KubernetesWorkerR ole-ec87d15b-edca- 4302-933f- c8a16bf0****	Worker RAM角色名称,授权ECS实例为集群 Woker节点。
zone_id	String	cn-beijing-a	集群所在可用区ID。

示例

请求示例

```
GET /clusters/[ClusterId] HTTP/1.1
公共请求头
{
    "ClusterId": "cdde1f21ae22e483ebcb068a6eb7f****"
}
```

正常返回示例

XML 格式

```
<private zone>false</private zone>
<resource group id>rg-acfmyvw3wjm****</resource group id>
<vpc id>vpc-2zecuu62b9zw7a7qn****/vpc id>
<network mode>vpc</network mode>
<maintenance window>
   <duration>3h</duration>
   <weekly period>Monday, Thursday</weekly period>
    <enable>false
    <maintenance time>03:00:00Z</maintenance time>
</maintenance_window>
<security_group_id>sg-25yq****</security group id>
<cluster type>Kubernetes</cluster type>
<docker version>19.03.5</docker version>
<next version>1.18.8-aliyun.1/next version>
<zone id>cn-beijing-a</zone id>
<cluster_id>c82e6987e2961451182edacd74faf****</cluster_id>
<external loadbalancer id>lb-2zehc05z3b8dwiifh****</external loadbalancer id>
<vswitch_id>vsw-2zete8s4qocqg0mf6****,vsw-2zete8s4qocqg0mf6***</vswitch_id>
<meta data>\"Addons\":***</meta data>
<state>running</state>
<eagleEyeTraceId>0a73095416412777494927508d004e/eagleEyeTraceId>
<init_version>1.16.6-aliyun.1</init_version>
<asapiSuccess>true</asapiSuccess>
<deletion protection>true</deletion protection>
<created>2019-11-25T15:50:20+08:00</created>
<subnet cidr>172.20.0.0/16</subnet cidr>
file>Default
<master_url>{\"intranet_api_server_endpoint\":\"https://192.168.0.251:6443\"***}</master_ur</pre>
<region id>cn-beijing</region id>
<current version>1.16.6-aliyun.1/current version>
<tags>
   <value>value</value>
   <key>key</key>
</tags>
<cluster spec>ack.standard</cluster spec>
<size>5</size>
<name>cluster-demo</name>
<worker ram role name>KubernetesWorkerRole-ec87d15b-edca-4302-933f-c8a16bf0****</worker ram</pre>
role name>
<updated>2020-01-13T23:01:03+08:00</updated>
```

JSON 格式

```
"private zone": false,
    "resource group id": "rg-acfmyvw3wjm****",
    "vpc id": "vpc-2zecuu62b9zw7a7qn****",
    "network mode": "vpc",
    "maintenance window": {
       "duration": "3h",
        "weekly period": "Monday, Thursday",
        "enable": false,
        "maintenance time": "03:00:00Z"
    },
    "security group id": "sg-25yq****",
    "cluster_type": "Kubernetes",
   "docker version": "19.03.5",
   "next version": "1.18.8-aliyun.1",
    "zone_id": "cn-beijing-a",
    "cluster id": "c82e6987e2961451182edacd74faf****",
    "external loadbalancer id": "lb-2zehc05z3b8dwiifh****",
    "vswitch id": "vsw-2zete8s4qocqq0mf6****, vsw-2zete8s4qocqq0mf6***",
    "meta data": "\\\"Addons\\\":***",
    "state": "running",
    "eagleEyeTraceId": "0a73095416412777494927508d004e",
    "init version": "1.16.6-aliyun.1",
   "asapiSuccess": true,
    "deletion protection": true,
    "created": "2019-11-25T15:50:20+08:00",
   "subnet cidr": "172.20.0.0/16",
    "profile": "Default",
    "master url": "{\\\"intranet api server endpoint\\\":\\\"https://192.168.0.251:6443\\\"
***}",
   "region id": "cn-beijing",
    "current_version": "1.16.6-aliyun.1",
    "tags": {
       "value": "value",
       "key": "key"
    "cluster spec": "ack.standard",
   "size": 5,
   "name": "cluster-demo",
   "worker ram role name": "KubernetesWorkerRole-ec87d15b-edca-4302-933f-c8a16bf0****",
    "updated": "2020-01-13T23:01:03+08:00"
}
```

2.2.4.3. 查询集群资源

调用DescribeClusterResources查询指定集群的所有资源。

请求头

该接口使用公共请求头, 无特殊请求头。请参见公共参数文档。

请求语法

GET /clusters/[ClusterId]/resources HTTP/1.1

请求参数

名称	类型	位置	是否必选	示例值	描述
Clusterid	String	Path	是	cb95aa626a477 40afbf6aa099b 65****	集群ID。

返回数据

名称	类型	示例值	描述
resources	Array of resource		资源对象列表。
auto_create	Integer	0	是否系统自动创建。 ① : 系统自动创建 ① : 非系统自动创建
cluster_id	String	cb95aa626a47740af bf6aa099b65****	集群ID。
created	String	2020-09- 11T10:11:54+08:00	资源创建时间。
dependencies	String	0	资源依赖项。
instance_id	String	lb- wz9poz4r0ymh8u0u f****	资源ID。
resource_info	String	{\"Id\":\"k8s_master _slb\",\"Name\":\"k 8s_master_slb\",\"T ype\":\"ALIYUN::SL B::LoadBalancer\",\ "Status\":\"CREATE_ COMPLETE\",\"Statu sReason\":\"state changed\",\"Update d\":\"2020-05- 21T13:25:02\",\"Ph ysicalld\":\"lb- wz9poz4r0ymh8u0u f****\"}	资源信息。

名称	类型	示例值	描述
resource_type	String	ALIYUN::SLB::LoadB alancer	资源类型。
			资源状态。可选值:
			● CREATE_COMPLETE : 成功创建资源。
			● CREATE_FAILED : 创建资源失败。
			● CREATE_IN_PROGRESS : 创建资源中。
state	String	CREAT E_COMPLET E	● DELETE_FAILED : 删除资源失败。
			● DELETE_IN_PROGRESS : 删除资源中。
			● ROLLBACK_COMPLETE : 成功回滚。
			● ROLLBACK_FAILED : 回滚失败。
			• ROLLBACK_IN_PROGRESS : 回滚中。

示例

请求示例

```
GET /clusters/[ClusterId]/resources HTTP/1.1
公共请求头
{
    "ClusterId": "cb95aa626a47740afbf6aa099b65****"
}
```

正常返回示例

XML 格式

JSON 格式

```
"resources": {
    "resource_info": "{\\\"Id\\\":\\\"k8s_master_slb\\\",\\\"Name\\\":\\\"k8s_master_sl
b\\\",\\\"Type\\\":\\\"ALIYUN::SLB::LoadBalancer\\\",\\\"2020-05-21T13:25:02\\\",\\\
"PhysicalId\\\":\\\"lb-wz9poz4r0ymh8u0uf****\\"]",
    "cluster_id": "cb95aa626a47740afbf6aa099b65****",
    "instance_id": "lb-wz9poz4r0ymh8u0uf****",
    "created": "2020-09-11T10:11:54+08:00",
    "resource_type": "ALIYUN::SLB::LoadBalancer",
    "state": "CREATE_COMPLETE",
    "auto_create": 0,
    "dependencies": "[]"
}
```

2.2.4.4. 查询集群命名空间

调用DescribeClusterNamespaces查询集群的命名空间列表。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

```
GET /k8s/[ClusterId]/namespaces HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c68069cade53b 4558924a924b4 15f****	集群ID。

返回数据

名称	类型	示例值	描述
namespaces	List	["kube-system"]	命名空间名称。

示例

请求示例

```
GET /k8s/[ClusterId]/namespaces HTTP/1.1
公共请求头
{
    "ClusterId": "c68069cade53b4558924a924b415f****"
}
```

正常返回示例

XML 格式

<namespaces>kube-system</namespaces>

JSON 格式

```
{
    "namespaces": ["kube-system"]
}
```

2.2.4.5. 查询集群代理

调用DescribeExternalAgent获取注册集群的代理配置。您可以将该配置部署到自己的集群用于访问API Server。

请求头

该接口使用公共请求头, 无特殊请求头。请参见公共参数文档。

请求语法

GET /k8s/[ClusterId]/external/agent/deployment HTTP/1.1

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c106f377e16f3 4eb1808d6b93 62c9****	集群ID。
Privat elpAddres s	String	Query	否	true	是否获取内网访问凭据。 • true: 仅获取内网连接凭据。 • false: 仅获取公网连接凭据。 默认值: false。

返回数据

名称	类型	示例值	描述
config	String	apiVersion: v1****	YAML格式的代理配置。

示例

请求示例

```
GET /k8s/[ClusterId]/external/agent/deployment HTTP/1.1
公共请求头
{
    "ClusterId": "c106f377e16f34eb1808d6b9362c9****"
}
```

正常返回示例

XML 格式

```
<config>apiVersion: v1****</config>
```

JSON 格式

```
{
    "config": "apiVersion: v1****"
}
```

2.2.4.6. 查询用户配额

调用DescribeUserQuota查询用户配额。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

```
GET /quota HTTP/1.1
```

请求参数

返回数据

名称	类型	示例值	描述
cluster_nodepo ol_quota	Long	10	单集群节点池配额。
cluster_quota	Long	50	用户总集群配额。

名称	类型	示例值	描述
node_quota	Long	100	单集群节点数配额。

示例

请求示例

```
GET /quota HTTP/1.1
公共请求头
```

正常返回示例

XML 格式

```
<amk_cluster_quota>20</amk_cluster_quota>
<cluster_nodepool_quota>10</cluster_nodepool_quota>
<cluster_quota>50</cluster_quota>
<node_quota>100</node_quota>
```

JSON 格式

```
"amk_cluster_quota": 20,
    "cluster_nodepool_quota": 10,
    "cluster_quota": 50,
    "node_quota": 100
}
```

2.2.4.7. 查询集群日志

调用DescribeClusterLogs查询指定集群日志。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

```
GET /clusters/[ClusterId]/logs HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c106f377e16f3 4eb1808d6b93 62c9****	集群ID。

返回数据

名称	类型	示例值	描述
data	Struct		日志列表
ID	Integer	590749245	日志ID。
cluster_id	String	c23421cfa74454bc8 b37163fd19af****	集群ID。
cluster_log	String	start to update cluster status to CREATE_COMPLETE	日志内容。
created	String	2020-09- 11T10:11:51+08:00	日志产生时间。
log_level	String	0004	日志级别。
updated	String	2020-09- 11T10:11:51+08:00	日志更新时间。

示例

请求示例

```
GET /clusters/[ClusterId]/logs HTTP/1.1
公共请求头
{
    "ClusterId": "c106f377e16f34eb1808d6b9362c9****"
}
```

正常返回示例

XML 格式

JSON 格式

```
"data": {
    "cluster_id": "c23421cfa74454bc8b37163fd19af***",
    "cluster_log": "start to update cluster status to CREATE_COMPLETE",
    "created": "2020-09-11T10:11:51+08:00",
    "log_level": 4,
    "ID": 590749245,
    "updated": "2020-09-11T10:11:51+08:00"
}
```

2.2.4.8. 查询Kubernetes版本详情

调用DescribeKubernetesVersionMetadata查询支持的Kubernetes版本的详细信息。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

```
GET /api/v1/metadata/versions HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterType	String	Query	是	Kubernetes	集群类型。
Region	String	Query	是	cn-beijing	地域。
KubernetesVers ion	String	Query	否	1.16.9-aliyun.1	Kubernetes版本,例如:1.16.9- aliyun.1。
Profile	String	Query	否	Default	边缘集群标识。● Default : Kubernetes专有版 集群。● Edge : 边缘专有版集群。
MultiAZ	Boolea n	Query	否	false	是否查询多可用区。

返回数据

名称	类型	示例值	描述	
data	Struct		版本详情列表。	
images	Array of image		ECS系统镜像列表。	
image_categor y	String	system	 镜像分类,部分取值: ● system : 公共镜像。 ● self : 您的自定义镜像。 ● others : 其他用户的公开镜像。 ● marketplace : 镜像市场镜像。 	
image_id	String	centos_7_7_x64_20G _alibase_20200426.v hd	镜像ID。	
image_name	String	Cent OS 7.7	镜像名称。	
image_type	String	Centos_7_64	镜像类型。	
os_type	String	Linux	操作系统平台类型。取值: Windows 、 Linux (默认)。	
os_version	String	7.7	操作系统发行版本号。	
platform	String	Cent OS	操作系统发行版。取值: CentOS : Centos镜像 AliyunLinux : AliyunLinux镜像 WindowsCore : Windows镜像 CentOS 7.6 : CentOS 7.6镜像 ARM CentOS : CentOS 7.7镜像 NeoKylin : 麒麟操作系统 默认值: AliyunLinux 。	
meta_data	Struct		Kubernetes版本元数据信息。	
Architecture	String	amd64	集群实例系统架构。	

名称	类型	示例值	描述	
AuditProjectNa me	String	k8s-log-c9bd1***	集群审计日志存储的SLS Project名称。	
DockerVersion	String	19.03.15	运行时版本。	
NodeNameMod e	String	true	自定义节点名称。	
Platform	String	"Cent OS"	操作系统发行版。取值: CentOS : Centos镜像 AliyunLinux : AliyunLinux镜像 WindowsCore : Windows镜像 CentOS 7.6 : CentOS 7.6镜像 ARM CentOS : CentOS 7.7镜像 NeoKylin : 麒麟操作系统 默认值: AliyunLinux 。	
PodVswitchId	String	vsw- f8zubs2z9x1cjnzk2* **	pod虚拟交换机,为terway网络类型的集群pod 分批ip	
ResourceGroupI d	String	rg- aekzsmulow2wngq	资源组ID。	
multi_az	Boolean	false	是否为多可用区。	
runtimes	Array of runtime		容器运行时配置。	
name	String	"docker"	运行时名称。	
version	String	"19.05.3"	运行时版本。	
version	String	"1.16.9-aliyun.1"	Kubernetes版本。	

示例

请求示例

```
GET /api/v1/metadata/versions HTTP/1.1
公共请求头
{
    "ClusterType": "Kubernetes",
    "Region": "cn-beijing"
}
```

正常返回示例

XML 格式

```
<data>
     <multi_az>false</multi_az>
     <version>"1.16.9-aliyun.1"
     <images>
           <image name>CentOS 7.7</image name>
           <image_category>system</image_category>
           <os version>7.7</os version>
           <os_type>Linux</os_type>
           <image_id>centos_7_7_x64_20G_alibase_20200426.vhd</image_id>
           <platform>CentOS</platform>
           <image type>Centos 7 64</image type>
     </images>
     <runtimes>
           <name>"docker"</name>
           <version>"19.05.3"</version>
     </runtimes>
      <meta data>
           <DockerVersion>19.03.15/DockerVersion>
           <Architecture>amd64</Architecture>
           <Platform>"CentOS"</Platform>
           <ResourceGroupId>rg-aekzsmulow2wngq </ResourceGroupId>
           <PodVswitchId>vsw-f8zubs2z9x1cjnzk2***</podVswitchId>
           <NodeNameMode>true</NodeNameMode>
           <AuditProjectName>k8s-log-c9bd1***</AuditProjectName>
     </meta data>
</data>
```

JSON 格式

```
"data": {
    "multi az": false,
    "version": "\"1.16.9-aliyun.1\"",
    "images": {
        "image_name": "CentOS 7.7",
        "image category": "system",
        "os_version": 7.7,
        "os_type": "Linux",
        "image_id": "centos_7_7_x64_20G_alibase_20200426.vhd",
        "platform": "CentOS",
        "image type": "Centos 7 64"
    },
    "runtimes": {
        "name": "\"docker\"",
        "version": "\"19.05.3\""
    },
    "meta_data": {
        "DockerVersion": "19.03.15",
        "Architecture": "amd64",
        "Platform": "\"CentOS\"",
        "ResourceGroupId": "rg-aekzsmulow2wngq",
        "PodVswitchId": "vsw-f8zubs2z9x1cjnzk2***",
        "NodeNameMode": true,
        "AuditProjectName": "k8s-log-c9bd1***"
}
```

2.2.5. 修改

2.2.5.1. 修改集群

调用ModifyCluster修改集群。

请求头

该接口使用公共请求头, 无特殊请求头。请参见公共参数文档。

请求语法

```
PUT /api/v2/clusters/[ClusterId] HTTP/1.1
```

请求参数

名称 类型 位置 是否必 选	示例值	描述
-------------------	-----	----

名称	类型	位置	是否必 选	示例值	描述
ClusterId	String	Path	是	cb95aa626a477 40afbf6aa099b 65****	集群ID。
api_server_eip	Boolea n	Body	否	true	集群是否绑定EIP,用于公网访问API Server。取值: ● true : 集群绑定EIP。 ● false : 集群不绑定EIP。
api_server_eip_i d	String	Body	否	eip- wz9fnasl6dsfh mvci****	集群API Server绑定的EIP实例ID,仅 当 api_server_eip 取值 为 true 时生效。
deletion_protec tion	Boolea n	Body	否	true	集群删除保护,防止通过控制台或API 误删除集群。取值: true : 启用集群删除保护,将 不能通过控制台或API删除集群。 false : 不启用集群删除保护, 则能通过控制台或API删除集群。 默认值: false 。
instance_deleti on_protection	Boolea n	Body	否	true	实例删除保护,防止通过控制台或API 误删除释放节点。取值: true: 不能通过控制台或API误 删除集群。 false: 能通过控制台或API误 删除集群。 默认值: false 。
ingress_domain _rebinding	String	Body	否	true	重新绑定集群测试域名。取值: ◆ true : 重新绑定集群测试域名。 ◆ false : 不重新绑定集群测试域名。 默认值: false 。
ingress_loadbal ancer_id	String	Body	否	lb- wz97kes8tnndk podw****	被修改集群的SLB实例ID。

名称	类型	位置	是否必选	示例值	描述
resource_group _id	String	Body	否	rg- acfmyvw3wjm* ***	集群资源组ID。

返回数据

名称	类型	示例值	描述
cluster_id	String	cb95aa626a47740af bf6aa09****	集群ID。
request_id	String	687C5BAA-D103- 4993-884B- C35E4314****	请求ID。
task_id	String	T- 5a54309c80282e39e a00****	任务ID。

示例

请求示例

```
PUT /api/v2/clusters/[ClusterId] HTTP/1.1
公共请求头
{
    "ClusterId": "cb95aa626a47740afbf6aa099b65****"
}
```

正常返回示例

XML 格式

```
<cluster_id>cb95aa626a47740afbf6aa09****</cluster_id>
<task_id>T-5a54309c80282e39ea00****</task_id>
<request_id>687C5BAA-D103-4993-884B-C35E4314****</request_id>
```

JSON 格式

```
{
   "cluster_id": "cb95aa626a47740afbf6aa09****",
   "task_id": "T-5a54309c80282e39ea00****",
   "request_id": "687C5BAA-D103-4993-884B-C35E4314****"
}
```

2.2.5.2. 修改集群标签

调用ModifyClusterTags修改当前Kubernetes集群的标签。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

```
POST /clusters/[ClusterId]/tags HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c106f377e16f3 4eb1808d6b93 62c9****	集群ID。
body	Array	Body	否		
tag	Struct		否		标签配置
key	String		否	env	标签 key 值。
value	String		否	prod	标签 value 值。

示例

请求示例

正常返回示例

```
XML 格式
```

```
HTTP/1.1 200 OK
```

JSON 格式

```
HTTP/1.1 200 OK
```

2.2.6. 升级

2.2.6.1. 升级集群

调用UpgradeCluster升级指定集群。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

```
POST /api/v2/clusters/[ClusterId]/upgrade HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c82e6987e2961 451182edacd74 faf****	集群ID。
body	Struct	body	否		
component_na me	String		否	k8s	组件名称,取值: k8s 。
version	String		否	1.14.8-aliyun.1	集群当前版本。
next_version	String		否	1.16.9-aliyun.1	集群可升级版本。

示例

请求示例

```
POST /api/v2/clusters/[ClusterId]/upgrade HTTP/1.1
公共请求头
{
    "ClusterId": ""
    "component_name": "k8s",
    "next_version": "1.16.9-aliyun.1",
    "version": "1.14.8-aliyun.1"
}
```

正常返回示例

```
JSON 格式
```

```
HTTP/1.1 200 OK
```

2.2.6.2. 查询集群升级状态

调用GetUpgradeStatus查询集群升级状态。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

GET /api/v2/clusters/[ClusterId]/upgrade/status HTTP/1.1

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c106f377e16f3 4eb1808d6b93 62c9****	集群ID。

返回数据

名称	类型	示例值	描述
error_message	String	subject to actual return	集群升级中的错误信息。
precheck_repor t_id	String	be4c8eb72de94d45 9ea7ace7c811d119	预检查返回ID。
status	String	running	集群目前升级的状态。
upgrade_step	String	prechecking	集群目前升级的阶段。
upgrade_task	Struct		升级任务详情。
message	String	subject to actual return	升级任务描述信息。
status	String	running	升级任务状态。

示例

请求示例

正常返回示例

XML 格式

JSON 格式

```
"error_message": "subject to actual return",

"upgrade_task": {
     "message": "subject to actual return",
     "status": "running"
},

"precheck_report_id": "be4c8eb72de94d459ea7ace7c811d119",
     "status": "running",
     "upgrade_step": "prechecking"
}
```

2.2.6.3. 暂停集群升级

调用PauseClusterUpgrade暂停用户集群升级。

请求头

该接口使用公共请求头, 无特殊请求头。请参见公共参数文档。

请求语法

```
POST /api/v2/clusters/[ClusterId]/upgrade/pause HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c82e6987e2961 451182edacd74 faf****	集群ID。

示例

请求示例

```
POST /api/v2/clusters/[ClusterId]/upgrade/pause HTTP/1.1
公共请求头
{
    "ClusterId": "c82e6987e2961451182edacd74faf****"
}
```

正常返回示例

JSON 格式

HTTP/1.1 200 OK

2.2.6.4. 取消集群升级

调用CancelClusterUpgrade取消集群升级。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

POST /api/v2/clusters/[ClusterId]/upgrade/cancel HTTP/1.1

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c106f377e16f3 4eb1808d6b93 62c9****	集群ID。

示例

请求示例

```
POST /api/v2/clusters/[ClusterId]/upgrade/cancel HTTP/1.1
公共请求头
{
    "ClusterId": "c106f377e16f34eb1808d6b9362c9****"
}
```

正常返回示例

```
JSON 格式
```

```
HTTP/1.1 200 OK
```

2.2.6.5. 恢复集群升级

调用ResumeUpgradeCluster重新开始升级暂停集群。

请求头

该接口使用公共请求头, 无特殊请求头。请参见公共参数文档。

请求语法

POST /api/v2/clusters/[ClusterId]/upgrade/resume HTTP/1.1

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c82e6987e2961 451182edacd74 faf****	集群ID。

示例

请求示例

```
POST /api/v2/clusters/[ClusterId]/upgrade/resume HTTP/1.1
公共请求头
{
    "ClusterId": "c82e6987e2961451182edacd74faf****"
}
```

正常返回示例

JSON 格式

HTTP/1.1 200 OK

2.3. 节点

2.3.1. 查询集群节点

调用DescribeClusterNodes查询集群节点。

请求头

该接口使用公共请求头, 无特殊请求头。请参见公共参数文档。

请求语法

GET /clusters/[ClusterId]/nodes HTTP/1.1

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c82e6987e2961 451182edacd74 faf****	集群ID。
instancelds	String	Query	否	i-node1,i- node2	节点实例ID,按照实例ID进行过滤。 节点实例ID不为空时会忽略此字段。多节点用半角逗号(,)分割。
nodepool_id	String	Query	否	np***	节点池ID。
state	String	Query	否	all	集群状态,按照集群运行状态进行过滤,取值:
pageSize	String	Query	否	3	分页大小。
pageNumber	String	Query	否	50	展示分页数。

返回数据

名称	类型	示例值	描述
nodes	Array of node		节点详情列表。
creation_time	String	"2020-08- 25T11:25:35+08:00	节点创建时间。
error_message	String	11***11	节点创建时错误信息。
expired_time	String	"2099-12- 31T23:59:00+08:00	节点到期时间。

名称	类型	示例值	描述	
host_name	String	"iZ2vcckdmxp7u0urj 2k****"	节点主机名。	
image_id	String	"aliyun_2_1903_x64_ 20G_alibase_202005 29.vhd"	节点使用的系统镜像ID。	
instance_charg e_type	String	"PostPaid"	节点付费类型。	
instance_id	String	"i- 2vcckdmxp7u0urj2k2 rw"	节点实例ID。	
instance_name	String	"worker-k8s-for-cs- c5cdf7e3938bc4f8e b0e44b21a80f****"	节点在集群中的名称。	
instance_role	String	"Worker"	节点角色类型: ● Master : 集群Master节点。 ● Worker : 集群Worker节点。	
instance_status	String	"Running"	节点运行状态。	
instance_type	String	"ecs.c5.xlarge"	节点类型。	
instance_type_f amily	String	"ecs.c5"	节点所属ECS族名称。	
ip_address	List	["192.168.44.247"]	节点IP地址。	
is_aliyun_node	Boolean	true	是否为阿里云实例。	
node_name	String	"cn- chengdu.192.168.0.3 6"	节点名称,集群内节点标识。	
node_status	String	"Ready"	节点是否就绪。	
nodepool_id	String	"np0794239424a84e b7a95327369d56*** *"	节点池ID。	
source	String	"ess_attach"	节点通过什么初始化出来的,例如:手动创建或 者ROS创建。	

名称	类型	示例值	描述
state	String	"running"	节点运行状态。
page	Struct		分页信息。
page_number	Integer	1	分页数。
page_size	Integer	10	单页大小。
total_count	Integer	5	结果总数。

示例

请求示例

正常返回示例

XML 格式

```
<creation time>"2020-08-25T11:25:35+08:00"</creation time>
   <error message>"***"</error message>
   <instance name>"worker-k8s-for-cs-c5cdf7e3938bc4f8eb0e44b21a80f****"</instance name>
   <node status>"Ready"</node status>
   <is aliyun node>true</is aliyun node>
   <node name>"cn-chengdu.192.168.0.36"</node name>
   <expired time>"2099-12-31T23:59:00+08:00"</expired time>
   <source>"ess attach"</source>
   <instance type family>"ecs.c5"</instance type family>
   <instance id>"i-2vcckdmxp7u0urj2k2rw"</instance id>
   <instance_charge_type>"PostPaid"</instance_charge_type>
   <instance role>"Worker"</instance role>
   <instance status>"Running"</instance status>
   <state>"running"</state>
   <image_id>"aliyun_2_1903_x64_20G_alibase_20200529.vhd"</image_id>
   <instance type>"ecs.c5.xlarge"</instance type>
   <nodepool id>"np0794239424a84eb7a95327369d56****"</nodepool id>
   <host name>"iZ2vcckdmxp7u0urj2k****"</host name>
   <ip address>["192.168.44.247"]</ip address>
</nodes>
<page>
   <page number>1</page number>
   <total count>5</total count>
   <page size>10</page size>
</page>
```

JSON 格式

```
"nodes": {
   "creation time": "\"2020-08-25T11:25:35+08:00\"",
   "error message": "\"***\"",
   "instance name": "\"worker-k8s-for-cs-c5cdf7e3938bc4f8eb0e44b21a80f****\"",
   "node status": "\"Ready\"",
   "is aliyun node": true,
   "node name": "\"cn-chengdu.192.168.0.36\"",
   "expired time": "\"2099-12-31T23:59:00+08:00\"",
   "source": "\"ess attach\"",
   "instance type family": "\"ecs.c5\"",
   "instance id": "\"i-2vcckdmxp7u0urj2k2rw\"",
    "instance_charge_type": "\"PostPaid\"",
   "instance role": "\"Worker\"",
   "instance status": "\"Running\"",
   "state": "\"running\"",
   "image id": "\"aliyun 2 1903 x64 20G alibase 20200529.vhd\"",
   "instance type": "\"ecs.c5.xlarge\"",
   "nodepool id": "\"np0794239424a84eb7a95327369d56****\"",
   "host name": "\"iZ2vcckdmxp7u0urj2k****\"",
   "ip address": "[\"192.168.44.247\"]"
},
"page": {
   "page number": 1,
   "total count": 5,
   "page_size": 10
```

2.3.2. 移除集群节点

调用RemoveClusterNodes移除指定集群额外节点。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

```
POST /api/v2/clusters/[ClusterId]/nodes/remove HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Patj	是	c850429a2287b 4d968e27e87a4 921****	集群ID。
body	Struct	Body	否		

名称	类型	位置	是否必选	示例值	描述
release_node	Boolea n		否	true	是否同时移除ECS。取值: true:同时移除ECS。 false:不同移除ECS。 默认值: false 。 ⑦ 说明 当节点为包年包月实例时,不支持同时移除ECS。
drain_node	Boolea n		否	true	是否自动排空节点上的Pod。取值: true : 自动排空节点上的 Pod。 false : 不自动排空节点上的 Pod。 默认值: false 。
nodes	String		否	["cn- chengdu.192.16 8.0.70"]	节点名称。

示例

请求示例

```
POST /api/v2/clusters/[ClusterId]/nodes/remove HTTP/1.1
公共请求头
{
    "ClusterId": "c82e6987e2961451182edacd74faf****"
    "release_node": false,
    "nodes": [
        "cn-shenzhen.192.168.0.70"
    ],
    "drain_node": false
}
```

正常返回示例

JSON 格式

HTTP/1.1 200 OK

2.3.3. 添加已有实例到集群

调用AttachInstances添加已有ECS实例到集群。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

POST /clusters/[ClusterId]/attach HTTP/1.1

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c106f377e16f3 4eb1808d6b93 62c9****	集群ID。
format_disk	Boolea n	Body	否	false	是否将容器数据和镜像存储在数据盘中。取值: true : 将容器数据和镜像存储在数据盘。 false : 不将容器数据和镜像存储在数据盘。 默认值: false 。 数据盘挂载规则: 如果ECS已挂载数据盘,且最后一块数据盘的文件系统未初始化,系统会自动将该数据盘格式化为EXT4,用来存放内容/var/lib/kubelet。 如果ECS未挂载数据盘,则不会挂载新的数据盘。 ① 说明 当选择将数据存储至数据盘并且ECS已挂载数据盘,数据盘内原有数据将丢失,请注意备份数据。
image_id	String	Body	否	aliyun_2_1903_ x64_20G_alibas e_20200529.vh d	自定义镜像ID,如果不传则使用默认系统镜像。 ② 说明 实例系统盘镜像将被替换为该镜像。

名称	类型	位置	是否必选	示例值	描述
keep_instance_ name	Boolea n	Body	否	true	是否保留原实例名称。取值: ● true : 保留实例名称。 ● false : 不保留实例名称。 默认值: true 。
user_data	String	Body	否	lyEvdXNyL2Jpbi 9iYXNoCmVjaG 8glkhlbGxvIEFDS yEi	节点自定义数据。更多详情,请参见云服务器ECS的实例自定义数据章节。
cpu_policy	String	Body	否	none	节点CPU管理策略。当集群版本在 1.12.6及以上时支持以下两种策略: ● static : 允许为节点上具有某些资源特征Pod增强其CPU亲和性和独占性。 ● none :表示启用现有的默认CPU亲和性方案。 默认值: none 。
runtime	Json	Body	否	{"name": "docker", "version": "19.03.5"}	容器运行时。
platform	String	Body	否	Cent OS	操作系统发行版。取值: CentOS : Centos镜像 AliyunLinux : AliyunLinux镜像 WindowsCore : Windows镜像 CentOS 7.6 : CentOS 7.6镜像 ARM CentOS : CentOS 7.7镜像 NeoKylin : 麒麟操作系统 默认值: AliyunLinux 。

名称	类型	位置	是否必选	示例值	描述
password	String	Body	否	Hello1234	待添加实例的SSH登录密码,和 key_pair 二选一。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),不支持反斜线(\)和半角双引号(")两个符号。 出于安全性考虑,密码传输过程中都是经过加密处理的。
instances	String	Body	否	["i- 2zed0sswuau6 o89b****"]	待添加的实例ID。
tags	Array	Body	否		节点标签。标签定义规则: • 标签由区分大小写的键值对组成,您最多可以设置20个标签。 • 标签键不可以重复,最长为64个字符;标签值可以为空,最长为128个字符。标签键和标签值都不能以 a liyun 、 acs: 、 https://或 http:// 开头。
key	String		否	key	key値。
value	String		否	value	value值。

返回数据

名称	类型	示例值	描述
list	Array of list		节点添加信息列表。
code	String	200	节点添加结果状态码。
instanceld	String	i- 2ze0lgm3y6iylcbt*** *	实例ID。

名称	类型	示例值	描述
message	String	successful	节点添加结果描述信息。
task_id	String	T- 5a544aff80282e39e a000039	任务ID。

示例

请求示例

正常返回示例

XML 格式

JSON 格式

```
"task_id": "T-5a544aff80282e39ea000039",

"list": {
    "code": 200,
    "instanceId": "i-2ze0lgm3y6iylcbt****",
    "message": "successful"
}
```

2.3.4. 手动添加已有实例至集群

调用DescribeClusterAttachScripts添加已有实例至ACK集群。该API返回唯一的可执行脚本。您获取脚本后,在已有节点上执行该脚本即可完成节点的接入。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

POST /clusters/[ClusterId]/attachscript HTTP/1.1

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	ca375a93a3047 4552ad2a0ebe 183e****	集群ID。
body	Struct	body	否		
keep_instance_ name	Boolea n		否	true	添加已有实到集群时,是否保留实例名称。如果不保留,则实例名称格式为worker-k8s-for-cs- <clusterid>。取值: true:保留实例名称。 false:不保留实例名称,会用系统规则进行替换。 默认值: true。</clusterid>
options	String		否	{\"enable ptabl es\": true,\"manage Runtime\": true,\"quiet\": true,\"allowed ClusterAddons\ ": [\"kube- proxy\",\"flann el\",\"coredns\ "]}	节点的接入配置参数。
arch	String		否	amd64	节点CPU架构。

名称	类型	位置	是否必选	示例值	描述
format_disk	Boolea n		否	false	手动添加已有实例到集群时,是否对该实例进行数据盘挂载,将容器和镜像存储在数据盘上。取值: true:对实例进行数据盘挂载,数据盘内原有数据将丢失,请注意备份数据。 false:不对实例进行数据盘挂载。 默认值:false。 数据挂载规则: 如果ECS已挂载数据盘,且最后一块数据盘的文件系统未初始化,系统会自动将该数据盘格式化为 ext4,用来存放内容/var/lib/docker、/var/lib/kubelet。 如果ECS未挂载数据盘,则不会挂载新的数据盘。
nodepool_id	String		否	np1c9229d9be 2d432c93f77a8 8fca0****	节点池ID,添加节点时可以将该节点添加到指定的节点池中。
rds_instances	String		否	["rm-xxx"]	RDS实例名称。

示例

请求示例

```
POST /clusters/[ClusterId]/attachscript HTTP/1.1
公共请求头

{
        "ClusterId": "ca375a93a30474552ad2a0ebe183e****"
        "nodepool_id": "np1c9229d9be2d432c93f77a88fca0****",
        "format_disk": false,
        "keep_instance_name": true,
        "rds_instances": ["rm-xxx"],
        "arch": "amd64",
        "options": "{\\\"enableIptables\\\": true,\\\"manageRuntime\\\": true,\\\"quiet\\\": true,\\\"allowedClusterA ddons\\\": [\\\"kube-proxy\\\",\\\"flannel\\\",\\\"coredns\\\"]}"
}
```

正常返回示例

JSON 格式

```
HTTP/1.1 200 OK
```

2.3.5. 节点池

2.3.5.1. 创建节点池

调用CreateClusterNodePool为集群创建节点池。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

POST /clusters/[ClusterId]/nodepools HTTP/1.1

请求参数

参数名称	类型	位置	是否必选	示例	说明
ClusterId	String	path	是	c61da77e8bf bc4c4c999af2 b51b65****	集群ID。
body	Struct	body	否		
count	Integer		否	1	节点池节点数量。
nodepool_inf o	Struct		否		节点池配置。
name	String		否	cluster-demo	节点池名称。
scaling_group	Struct		否		节点池扩容组配置。
system_disk_c ategory	String		否	cloud_efficien cy	节点系统盘类型,取值: ● cloud_efficiency : 高效 云盘。 ● cloud_ssd : SSD云盘。 ● cloud_essd : ESSD云盘。 默认 值: cloud_efficiency 。
system_disk_s ize	Integer		否	120	节点系统盘大小,单位为GiB。
system_disk_ performance_ level	String		否	PL1	节点系统盘磁盘性能,只针对ESSD 磁盘生效。

参数名称	类型	位置	是否必选	示例	说明
vpc_id	String		否	vpc- wz984yvbd6l ck22z3****	虚拟专有网络ID。
instance_char ge_type	String		否	PrePaid	节点池节点付费类型,取值: ● PrePaid : 预付费。 ● PostPaid : 按量付费。 默认值: PostPaid 。
internet_max_ bandwidth_o ut	Integer		否	5	公网IP带宽。
login_passwo rd	String		否	Hello1234	登录密码。
security_grou p_id	String		否	sg- wz9a8g2mt6 x5llu0****	安全组ID。
platform	String		否	Linux	操作系统发行版。
image_id	String		否	aliyun_2_1903 _x64_20G_alib ase_20200529 .vhd	自定义镜像ID,默认使用系统提供的镜像。
internet_char ge_type	String		否	PayByTraffic	公网IP收费类型。
scaling_policy	String		否	release	伸缩组模式,取值: release : 标准模式,根据申请资源值的使用量,通过创建、释放ECS的方式进行伸缩。 recycle : 极速模式,通过创建、停机、启动的方式进行伸缩,提高再次伸缩的速度(停机时计算资源不收费,只收取存储费用,本地盘机型除外)。 默认值: release 。
vswitch_ids	String		否	vsw- wz9mfnhmss ud6eicu****	虚拟交换机ID。
instance_type s	String		否	ecs.d1ne.2xla rge	实例规格。

参数名称	类型	位置	是否必选	示例	说明
tags	Array		否		仅为ECS实例添加标签。 标签键不可以重复,最大长度为 128个字符;标签键和标签值都不 能以"aliyun"、"acs:"开头, 或包含"https://"、 "http://"。
key	String		否	env	标签的名称。
value	String		否	prod	标签值。
data_disks	Array		否		数据盘配置。
category	String		否	cloud_ssd	数据盘类型。
size	Integer		否	40	数据盘大小,单位为GiB。
encrypted	String		否	true	是否对数据盘加密,取值: true : 对数据盘加密。 false : 不对数据盘加密。 默认值: false
auto_snapsho t_policy_id	String		否	sp- 2zej1nogjvov nz4z****	选择自动快照策略ID, 云盘会按照 快照策略自动备份。 默认值:空,不自动备份。
kubernetes_c onfig	Struct		否		集群相关配置。
cpu_policy	String		否	none	节点CPU管理策略。当集群版本在 1.12.6及以上时支持以下两种策略: static : 允许为节点上具有某些资源特征Pod增强其CPU亲和性和独占性。 none : 表示启用现有的默认CPU亲和性方案。 默认值: none 。

参数名称	类型	位置	是否必选	示例	说明
cms_enabled	Boolea n		否	true	是否在ECS节点上安装云监控,安装后可以在云监控控制台查看所创建ECS实例的监控信息,推荐开启。取值: true: 在ECS节点上安装云监控。 false: 不在ECS节点上安装云监控。 默认值: false。
unschedulabl e	Boolea n		否	false	节点扩容后设置为不可调度。勾选 该项后,新添加的节点注册到集群 时默认设置为不可调度,若想打开 调度选项,可以在节点列表中开 启。
user_data	String		否	dGhpcyBpcyB hIGV4YW1wb GU=	节点自定义数据。
runtime	String		否	docker	容器运行时。
runtime_versi on	String		否	19.03.5	容器运行时版本。
node_name_ mode	String		否	aliyun.com00 055test	节点名称由三部分组成:前缀 + 节点IP地址子串 + 后缀: ● 前缀和后缀均可由 "."分隔的一个或多个部分构成,每个部分可以使用小写字母、数字和 "-",节点名称首尾必须为小写字母和数字。 ● IP 地址段长度指截取节点 IP 地址末尾的位数,取值范围5~12。 例如: 节点IP地址为: 192.168.0.55,指定前缀为aliyun.com,IP 地址段长度为5,后缀为test,则节点名称为aliyun.com00055test。
labels	Array		否		节点标签,为Kubernetes集群节点 添加标签。
key	String		否	env	标签 key 值。
value	String		否	prod	标签 value 值。

参数名称	类型	位置	是否必选	示例	说明
taints	Array		否		污点
key	String		否	env	污点 key 值。
value	String		否	prod	污点 value 值。
effect	String		否	NoSchedule	调度策略。取值: ● NoSchedule :不能容忍,但仅影响调度过程,已被调度的Pod不受影响,仅对新增加的Pod生效。 ● NoExecute :不能容忍,当污点变动时,Pod对象会被驱逐。 ● PreferNoSchedule :柔性约束,节点现存Pod不受影响。尽量不去满足不合要求的Pod调度到节点上。 默认值: NoSchedule 。
tee_config	Struct		否		加密计算集群配置。
tee_enable	Boolea n		否	true	是否开启加密计算集群。
auto_scaling	Struct		否		自动伸缩配置。
enable	Boolea n		否	true	是否启用自动伸缩。取值: true : 开启节点池自动伸缩功能。 false : 不开启自动伸缩,当取值为false时,其他auto_scaling配置参数将不生效。 默认值: false 。
max_instance s	Integer		否	10	自动伸缩组最大实例数。
min_instances	Integer		否	1	自动伸缩组最小实例数。
type	String		否	сри	自动伸缩类型,按照自动伸缩实例 类型划分。

返回数据

名称	类型	示例值	描述
nodepool_id	String	np5faa48fb31b6b80 78d00****	节点池ID。

示例

请求示例

```
"nodepool info": {
   "name": "test"
},
"scaling_group": {
    "vswitch_ids": [
       "vsw-ow2maqt4wgnzsge7a****"
   ],
    "system disk category": "cloud efficiency",
    "system_disk_size": 120,
    "system_disk_performance_level": "",
    "data_disks": [],
    "instance_types": [
       "ecs.mn4.large"
    "vpc id": "vpc-ow2cm4rzhrclukhm6****",
    "tags": [],
    "instance_charge_type": "PostPaid",
    "internet_max_bandwidth_out": 0,
    "login_password": "Hello1234",
    "platform": "CentOS",
    "image_id": "centos_7_9_x64_20G_alibase_20210318.vhd",
    "rds_instances": []
},
"kubernetes_config": {
    "cpu_policy": "none",
    "cms enabled": false,
    "unschedulable": false,
    "labels": [],
    "taints": [],
    "user_data": "",
    "runtime": "docker",
    "runtime version": "19.03.15"
},
"auto_scaling": {
    "enable": false
},
"tee config": {
   "tee_enable": false
},
"count": 1
```

正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<nodepool_id>np31da1b38983f4511b490fc62108a****</nodepool_id>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
    "nodepool_id": "np31da1b38983f4511b490fc62108a****"
}
```

2.3.5.2. 扩容节点池

调用ScaleClusterNodePool扩容节点池节点。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

POST /clusters/[ClusterId]/nodepools/[NodepoolId] HTTP/1.1

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c23421cfa7445 4bc8b37163fd1 9af****	集群ID。
Nodepoolld	String	Path	是	np31da1b3898 3f4511b490fc6 2108a****	节点池ID。
body	Struct	body	否		
count	Long		否	1	扩容节点数量。受当前集群节点配额限制,单次操作最多扩容500个节点。

返回数据

名称	类型	示例值	描述
task_id	String	T- 5faa48fb31b6b8078 d00****	任务ID。

示例

请求示例

```
POST /clusters/[ClusterId]/nodepools/[NodepoolId] HTTP/1.1
公共请求头
{
    "ClusterId": "c23421cfa74454bc8b37163fd19af****",
    "NodepoolId": "np31da1b38983f4511b490fc62108a****"
    "count": 1
}
```

正常返回示例

XML 格式

```
<task_id>T-5faa48fb31b6b8078d00****</task_id>
```

```
JSON 格式
```

```
{
    "task_id": "T-5faa48fb31b6b8078d00****"
}
```

2.3.5.3. 修改节点池配置

调用ModifyClusterNodePool更新节点配置。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

```
PUT /clusters/[ClusterId]/nodepools/[NodepoolId] HTTP/1.1
```

请求参数

参数名称	类型	位置	是否必选	示例	说明
ClusterId	String	path	是	c23421cfa744 54bc8b37163 fd19af****	集群ID。
Nodepoolld	String	path	是	p31da1b3898 3f4511b490fc 62108a****	节点池ID。
update_node s	Boolea n	Body	否	true	同步更新节点标签及污点。

参数名称	类型	位置	是否必选	示例	说明
auto_scaling	Struct	Body	否		自动伸缩配置。
enable	Boolea n		否	true	是否启用自动伸缩,取值: true : 开启节点池自动伸缩功能。 false : 不开启自动伸缩,当取值为 false 时, auto scaling 内的其他配置参数将不生效。 默认值: false 。
max_instance s	Long		否	10	最大实例数。
min_instances	Long		否	2	最小实例数。
type	String		否	cpu	自动伸缩类型,按照自动伸缩实例 类型划分。取值: cpu :普通实例型。 gpu :GPU实例型。 gpushare :GPU共享型。 spot :抢占式实例型。 默认值:cpu。
kubernetes_c onfig	Struct	Body	否		集群相关配置。
labels	Array		否	[{"key": "String", "value": "String"}]	节点标签,为Kubernetes集群节点添加标签。标签定义规则: • 标签由区分大小写的键值对组成,您最多可以设置20个标签。 • 标签键不可以重复,最长为64个字符;标签值可以为空,最长为128个字符。标签键和标签值都不能以`aliyun`、`acs`:、`https://`或`http://`开头。
key	String		否	env	标签 key 值。
value	String		否	prod	标签 value 值。
taints	Array		否		节点污点配置。
key	String		否	env	污点 key 值。
value	String		否	prod	污点 value 值。

参数名称	类型	位置	是否必选	示例	说明
effect	String		否	NoSchedule	调度策略。取值: ● NoSchedule : 不能容忍,但仅影响调度过程,已被调度的Pod不受影响,仅对新增加的Pod生效。 ● NoExecute : 不能容忍,当污点变动时,Pod对象会被驱逐。 ● PreferNoSchedule : 柔性约束,节点现存Pod不受影响。尽量不去满足不合要求的Pod调度到节点上。 默认值: NoSchedule 。
cms_enabled	Boolea n		否	true	是否在ECS节点上安装云监控,安装后可以在云监控控制台查看所创建ECS实例的监控信息,推荐开启。取值: true : 在ECS节点上安装云监控。 false : 不在ECS节点上安装云监控。 默认值: false 。
runtime	String		否	docker	容器运行时名称。
runtime_versi on	String		否	19.03.5	容器运行时版本。
user_data	String		否	lyEvdXNyL2Jp bi9iYXNoCmVj aG8glkhlbGxvl EFDSyEi	节点池自定义数据。
unschedulabl e	Boolea n		否	false	节点扩容后设置为不可调度。勾选 该项后,新添加的节点注册到集群 时默认设置为不可调度,若想打开 调度选项,可以在节点列表中开 启。
scaling_group	Struct	Body	否		扩容组配置。
data_disks	Array		否		节点数据盘配置,取值范围 [0,10]。最多支持添加10块数据 盘。
category	String		否	cloud_ssd	数据盘大小,单位为GiB。

参数名称	类型	位置	是否必选	示例	说明
size	Integer		否	40	是否对数据盘加密,取值: true : 对数据盘加密。 false : 不对数据盘加密。 默认值: false
encrypted	String		否	true	选择自动快照策略ID, 云盘会按照 快照策略自动备份。 默认值:空,不自动备份。
tags	Array		否	testkey	仅为ECS实例添加标签。 标签键不可以重复,最大长度为 128个字符;标签键和标签值都不 能以"aliyun"、"acs:"开头, 或包含"https://"、 "http://"。
key	String		否	env	标签的名称。
value	String		否	prod	标签值。
instance_char ge_type	String		否	PrePaid	节点付费类型。
platform	String		否	AliyunLinux	操作系统发行版。取值: CentOS : Centos镜像 AliyunLinux : AliyunLinux镜像 WindowsCore : Windows 镜像 CentOS 7.6 : CentOS 7.6 镜像 ARM CentOS : CentOS 7.7 镜像 NeoKylin : 麒麟操作系统 默认值: AliyunLinux 。
image_id	String		否	aliyun_2_1903 _x64_20G_alib ase_20200904 .vhd	自定义镜像ID。可通过 DescribeKubernetesVersionMeta data查询系统支持的镜像,默认使 用系统最新镜像。
login_passwo rd	String		否	Hello1234	SSH登录密码。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)。

参数名称	类型	位置	是否必选	示例	说明
scaling_policy	String		否	release	伸缩组模式,取值: ● release : 标准模式,根据申请资源值的使用量,通过创建、释放ECS的方式进行伸缩。 ● recycle : 极速模式,通过创建、停机、启动的方式进行伸缩,提高再次伸缩的速度(停机时计算资源不收费,只收取存储费用,本地盘机型除外)。
system_disk_c ategory	String		否	cloud_efficien cy	节点系统盘类型,取值:● cloud_efficiency : 高效 云盘。● cloud_ssd : SSD云盘。默认值: cloud_ssd 。
system_disk_s ize	String		否	120	 节点系统盘大小,单位为GiB。 取值范围: [20,500]。 该参数的取值必须大于或者等于max{20, ImageSize}。 默认值: max{40,参数ImageId对应的镜像大小}。
internet_max_ bandwidth_o ut	Integer		否	10	节点公网IP出带宽最大值,单位为 Mbps(Mega bit per second), 取值范围:1~100。
multi_az_poli cy	String		否	BALANCE	多可用区伸缩组ECS实例扩缩容策 略。
internet_char ge_type	String		否	PayByBandwi dth	节点公网IP网络计费类型。
instance_type s	List <st ring></st 		否	["ecs.c6.large "]	实例规格。请参见 <i>云服务器用户指</i> <i>南</i> 的 <i>实例规格</i> 章节。
vswitch_ids	List <st ring></st 		否	["vsw- wz9uwxhaw mtzg7u9h**** "]	虚拟交换机ID。
tee_config	Struct	Body	否		加密计算集群配置。
tee_enable	Boolea n		否	false	是否开启加密计算集群,取值: ● true : 开启。 ● false : 不开启。 默认值: false 。

返回数据

名称	类型	示例值	描述
nodepool_id	String	np737c3ac1ac68470 3b9e10673aa2c****	节点池ID。
task_id	String	T- 5fd211e924e1d0078 7000293	任务ID。

示例

请求示例

```
PUT /clusters/[ClusterId]/nodepools/[NodepoolId] HTTP/1.1
公共请求头
{
    "ClusterId": "c23421cfa74454bc8b37163fd19af****",
    "NodepoolId": "p31da1b38983f4511b490fc62108a****"
}
```

正常返回示例

XML 格式

```
HTTP/1.1 200 OK

Content-Type:application/xml

<task_id>T-5fd211e924e1d00787000293</task_id>
<nodepool_id>np737c3ac1ac684703b9e10673aa2c****</nodepool_id>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
    "task_id": "T-5fd211e924e1d00787000293",
    "nodepool_id": "np737c3ac1ac684703b9e10673aa2c****"
}
```

2.3.5.4. 查询节点池列表

调用DescribeClusterNodePools查询集群内所有节点池详情。

请求头

该接口使用公共请求头, 无特殊请求头。请参见公共参数文档。

请求语法

```
GET /clusters/[ClusterId]/nodepools HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	cc0f87de0b8fb 403f86e10e204 f83****	集群ID。

返回数据

名称	类型	示例值	描述
nodepools	Array of nodepool		节点池实例列表。
auto_scaling	Struct		自动伸缩配置。
eip_bandwidth	Integer	5	EIP带宽峰值。
enable	Boolean	true	是否启用自动伸缩,取值: true : 开启节点池自动伸缩功能。 false : 不开启自动伸缩,当取值为 false 时, auto_scaling 内的其他配置 参数将不生效。
max_instances	Integer	10	最大实例数。
min_instances	Integer	2	最小实例数。
type	String	cpu	自动伸缩类型,按照自动伸缩实例类型划分。取值:
kubernetes_con fig	Struct		集群相关配置。

名称	类型	示例值	描述
cms_enabled	Boolean	true	是否在ECS节点上安装云监控,安装后可以在云监控控制台查看所创建ECS实例的监控信息,推荐开启。取值: true: 在ECS节点上安装云监控。 false: 不在ECS节点上安装云监控。
cpu_policy	String	none	节点CPU管理策略。当集群版本在1.12.6及以上时支持以下两种策略: ● static : 允许为节点上具有某些资源特征Pod增强其CPU亲和性和独占性。 ● none : 表示启用现有的默认CPU亲和性方案。
labels	Array of label		节点标签,为Kubernetes集群节点添加标签。标签定义规则: • 标签由区分大小写的键值对组成,您最多可以设置20个标签。 • 标签键不可以重复,最长为64个字符;标签值可以为空,最长为128个字符。标签键和标签值都不能以 aliyun 、 acs : 、 https:// 或 http:// 开头。
key	String	key	标签key值。
value	String	value	标签value值。
node_name_m ode	String	customized,test.,5,.	自定义节点名称。
overwrite_host name	Boolean	false	是否覆盖主机名。
runtime	String	docker	容器运行时名称。
runtime_version	String	19.03.5	容器运行时版本。
taints	Array of taint		节点污点信息。污点和容忍度(Toleration)相 互配合,可以用来避免Pod被分配到不合适的节 点上。

名称	类型	示例值	描述
effect	String	NoSchedule	调度策略。取值: NoSchedule : 不能容忍,但仅影响调度过程,已被调度的Pod不受影响,仅对新增加的Pod生效。 NoExecute : 不能容忍,当污点变动时,Pod对象会被驱逐。 PreferNoSchedule : 柔性约束,节点现存Pod不受影响。尽量不去满足不合要求的Pod调度到节点上。 默认值: NoSchedule 。
key	String	key	污点key值。
value	String	value	污点value值。
unschedulable	Boolean	false	节点扩容后设置为不可调度。选中该项后,新添加的节点注册到集群时默认设置为不可调度,若想打开调度选项,可以在节点列表中开启。
user_dat <i>a</i>	String	lyEvYmluL3NoCmVja G8glkhlbGxvIEFD****I yEvYmluL3NoCmVja G8glkhlbGxvIEFD****	节点池自定义数据。更多详情,请参见云服务器 ECS的实例自定义数据章节。
nodepool_info	Struct		节点池信息。
created	String	2020-09- 27T19:14:09.156823 496+08:00	节点池创建时间。
is_default	Boolean	true	是否为默认节点池,通常一个集群仅有一个默认节点池。取值: true : 设置为默认节点池。 false : 不设置为默认节点池。
name	String	default-nodepool	节点池名称。 命名规则:由数字、汉字、英文字符或短划线 (-)组成,长度范围1~63个字符,且不能以短 划线(-)开头。
nodepool_id	String	np615c0e096612421 6a0412e10afe0****	节点池ID。

名称	类型	示例值	描述
region_id	String	cn-beijing	地域ID。
resource_group _id	String	rg- acfmyvw3wjm****	资源组ID。
type	String	ess	节点池类型。
updated	String	2020-09- 27T20:37:46+08:00	节点池更新时间。
scaling_group	Struct		扩容组配置。
data_disks	String	[{ "category" : "String", "size" : Long, "encrypted" : "String", "auto_snapshot_pol icy_id" : "String" }]	数据盘配置。
data_disks	Array of data_disk		节点数据盘类型、大小等配置的组合。
auto_snapshot _policy_id	String	sp- 2zej1nogjvovnz4z*** *	开启云盘备份时的自动备份策略。
category	String	cloud_ssd	数据盘类型
encrypted	String	true	是否对数据盘加密。
size	Long	40	数据盘大小,取值范围: 40~32767
image_id	String	aliyun_2_1903_x64_2 0G_alibase_2020052 9.vhd	自定义镜像ID,可通 过 DescribeKubernetesVersionMetadata 查询系统支持的镜像。
instance_charg e_type	String	PostPaid	节点池节点付费类型,取值: ● PrePaid : 预付费。 ● PostPaid : 按量付费。
instance_types	List	["ecs.n4.large"]	节点实例规格。

名称	类型	示例值	描述
internet_charge _type	String	PayByBandwidth	节点公网IP网络计费类型。
internet_max_b andwidth_out	Integer	10	节点公网IP出带宽最大值,单位为Mbps(Mega bit per second),取值范围:1~100。
login_passwor d	String	*****	SSH登录密码。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)。 出于安全性考虑,密码查询结果经过加密处理。
multi_az_policy	String	BALANCE	多可用区伸缩组ECS实例扩缩容策略。
platform	String	AliyunLinux	操作系统发行版。取值: CentOS : Centos镜像 AliyunLinux : AliyunLinux镜像 WindowsCore : Windows镜像 CentOS 7.6 : CentOS 7.6镜像 ARM CentOS : CentOS 7.7镜像 NeoKylin : 麒麟操作系统 默认值: AliyunLinux 。
ram_policy	String	KubernetesWorkerR ole-021dc54f-929b- 437a-8ae0- 34c24d3e****	Worker RAM角色名称,授权ECS实例为集群 Woker节点。
scaling_group_i d	String	asg- 2ze8n5qw4atggut8* ***	伸缩组ID。
scaling_policy	String	release	伸缩组模式,取值: release : 标准模式,根据申请资源值的使用量,通过创建、释放ECS的方式进行伸缩。 recycle : 极速模式,通过创建、停机、启动的方式进行伸缩,提高再次伸缩的速度(停机时计算资源不收费,只收取存储费用,本地盘机型除外)。
security_group_ id	String	sg- 2ze1iuk12m2sb4c4* ***	安全组ID。

名称	类型	示例值	描述
system_disk_ca tegory	String	cloud_efficiency	节点系统盘类型,取值: ● cloud_efficiency : 高效云盘。 • cloud_ssd : SSD云盘。
system_disk_si ze	Integer	120	节点系统盘大小,单位为GiB。 取值范围:20~500
tags	Array of tag		仅为ECS实例添加标签。 标签键不可以重复,最大长度为128个字符;标 签键和标签值都不能以 aliyun 、 acs 开 头,或包含 https:// 、 http:// 。
key	String	key	标签key值。
value	String	value	标签value值。
vswitch_ids	List	["vsw- 2ze3ds0mdip0hdz8i ****"]	虚拟交换机ID。
status	Struct		节点池状态信息。
failed_nodes	Integer	0	失败实例数。
healthy_nodes	Integer	3	处于健康状态实例数。
initial_nodes	Integer	0	正在创建的节点数。
offline_nodes	Integer	0	离线节点数。
removing_node s	Integer	0	正在被移除的节点数。
serving_nodes	Integer	3	处于工作状态的节点数。

名称	类型	示例值	描述
state	String	active	节点池状态,取值: active : 已激活。 scaling : 伸缩中。 removing : 节点移除中。 deleting : 删除中。 updating : 更新中。
total_nodes	Integer	3	节点池内节点数量。
tee_config	Struct		加密计算配置。
tee_enable	Boolean	false	是否开启加密计算集群,取值: true : 开启。 false : 不开启。

示例

请求示例

```
GET /clusters/[ClusterId]/nodepools HTTP/1.1
公共请求头
{
    "ClusterId": "cc0f87de0b8fb403f86e10e204f83****"
}
```

正常返回示例

XML 格式

```
<nodepools>
     <tee_config>
           <tee_enable>false</tee_enable>
     </tee config>
      <scaling group>
            <scaling group id>asg-2ze8n5qw4atggut8****</scaling group id>
            <internet_charge_type>PayByBandwidth</internet_charge_type>
            <internet_max_bandwidth_out>10</internet_max_bandwidth_out>
            <security_group_id>sg-2zeliukl2m2sb4c4***</security_group_id>
            <multi_az_policy>BALANCE</multi_az_policy>
            <system disk size>120</system disk size>
            <platform>AliyunLinux</platform>
            <ram_policy>KubernetesWorkerRole-021dc54f-929b-437a-8ae0-34c24d3e****</ram_poli</pre>
cy>
            <data disks>
                  <auto_snapshot_policy_id>sp-2zej1nogjvovnz4z***</auto_snapshot_policy_id</pre>
```

```
<encrypted>true</encrypted>
                  <size>40</size>
                  <category>cloud ssd</category>
            </data disks>
            <instance_charge_type>PostPaid</instance charge type>
            <login password>*****</login password>
            <system disk category>cloud efficiency</system disk category>
            <image id>aliyun 2 1903 x64 20G alibase 20200529.vhd</image id>
            <scaling policy>release</scaling policy>
            <tags>
                 <value>value</value>
                 <key>key</key>
            <instance types>ecs.n4.large</instance types>
            <vswitch ids>vsw-2ze3ds0mdip0hdz8i****</vswitch ids>
     </scaling_group>
      <kubernetes config>
           <runtime version>19.03.5</runtime version>
            <cpu policy>none/cpu policy>
            <cms enabled>true</cms enabled>
            <overwrite hostname>false</overwrite hostname>
           <runtime>docker</runtime>
            <user data>IyEvYmluL3NoCmVjaG8qIkhlbGxvIEFD****IyEvYmluL3NoCmVjaG8qIkhlbGxvIEFD
****</user data>
           <unschedulable>false</unschedulable>
           <node_name_mode>customized,test.,5,.com</node_name_mode>
                 <value>value</value>
                  <key>key</key>
            </labels>
            <taints>
                  <effect>NoSchedule</effect>
                  <value>value</value>
                 <key>key</key>
            </taints>
     </kubernetes config>
      <auto scaling>
            <enable>true</enable>
           <min instances>2</min instances>
           <max instances>10</max instances>
            <eip bandwidth>5</eip bandwidth>
           <type>cpu</type>
     </auto scaling>
      <nodepool info>
            <resource group id>rg-acfmyvw3wjm****</resource group id>
            <created>2020-09-27T19:14:09.156823496+08:00</created>
           <name>default-nodepool</name>
           <region_id>cn-beijing</region_id>
            <type>ess</type>
            <is default>true</is default>
            <updated>2020-09-27T20:37:46+08:00</updated>
            <nodepool id>np615c0e0966124216a0412e10afe0****/nodepool id>
      </nodepool info>
     <status>
```

JSON 格式

```
"nodepools": {
        "tee config": {
            "tee enable": false
        "scaling group": {
            "scaling group id": "asg-2ze8n5qw4atggut8****",
            "internet charge type": "PayByBandwidth",
            "internet max bandwidth out": 10,
            "security group id": "sg-2zeliuk12m2sb4c4****",
            "multi az policy": "BALANCE",
            "system disk size": 120,
            "platform": "AliyunLinux",
            "ram policy": "KubernetesWorkerRole-021dc54f-929b-437a-8ae0-34c24d3e****",
            "data disks": {
                "auto_snapshot_policy_id": "sp-2zej1nogjvovnz4z****",
                "encrypted": true,
                "size": 40,
                "category": "cloud ssd"
            "instance charge type": "PostPaid",
            "login_password": "*****",
            "system disk category": "cloud efficiency",
            "image id": "aliyun 2 1903 x64 20G alibase 20200529.vhd",
            "scaling policy": "release",
            "tags": {
                "value": "value",
                "key": "key"
            },
            "instance types": ["ecs.n4.large"],
            "vswitch ids": ["vsw-2ze3ds0mdip0hdz8i****"]
        "kubernetes config": {
            "runtime version": "19.03.5",
            "cpu policy": "none",
            "cms enabled": true,
            "overwrite hostname": false,
            "runtime": "docker",
            "user data": "IyEvYmluL3NoCmVjaG8gIkhlbGxvIEFD****IyEvYmluL3NoCmVjaG8gIkhlbGxvI
EFD****",
```

```
"unschedulable": laise,
        "node name mode": "customized, test., 5, .com",
        "labels": {
             "value": "value",
            "key": "key"
        },
        "taints": {
            "effect": "NoSchedule",
            "value": "value",
            "key": "key"
    },
    "auto_scaling": {
        "enable": true,
        "min instances": 2,
        "max instances": 10,
        "eip bandwidth": 5,
        "type": "cpu"
    },
    "nodepool info": {
        "resource_group_id": "rg-acfmyvw3wjm****",
        "created": "2020-09-27T19:14:09.156823496+08:00",
        "name": "default-nodepool",
        "region_id": "cn-beijing",
        "type": "ess",
        "is default": true,
        "updated": "2020-09-27T20:37:46+08:00",
        "nodepool id": "np615c0e0966124216a0412e10afe0****"
    },
    "status": {
        "serving nodes": 3,
        "total_nodes": 3,
        "state": "active",
        "initial_nodes": 0,
        "removing nodes": 0,
        "offline_nodes": 0,
        "failed nodes": 0,
        "healthy nodes": 3
}
```

2.3.5.5. 查询节点池详情

调用DescribeClusterNodePoolDetail查询集群指定节点池详情。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

```
GET /clusters/[ClusterId]/nodepools/[NodepoolId] HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c23421cfa7445 4bc8b37163fd1 9af****	集群ID。
Nodepoolld	String	Path	是	np31da1b3898 3f4511b490fc6 2108a****	节点池ID。

返回数据

名称	类型	示例值	描述
auto_scaling	Struct		自动伸缩节点池配置。
enable	Boolean	true	是否启用自动伸缩。 true : 开启节点池自动伸缩功能。 false : 不开启自动伸缩,当取值为 false 时, auto_scaling 内的其他配置参数将不生效。
max_inst ances	Integer	10	最大实例数。
min_instances	Integer	1	最小实例数。
type	String	cpu	自动伸缩类型,按照自动伸缩实例类型划分。取值:
kubernetes_con fig	Struct		集群相关配置。
cms_enabled	Boolean	true	是否在ECS节点上安装云监控,安装后可以在云监控控制台查看所创建ECS实例的监控信息,推荐开启。取值: true : 在ECS节点上安装云监控。 false : 不在ECS节点上安装云监控。

名称	类型	示例值	描述
cpu_policy	String	none	节点CPU管理策略。当集群版本在1.12.6及以上时支持以下两种策略: static : 允许为节点上具有某些资源特征Pod增强其CPU亲和性和独占性。 none : 表示启用现有的默认CPU亲和性方案。
labels	Array of label		节点标签,为Kubernetes集群节点添加标签。标签定义规则: • 标签由区分大小写的键值对组成,您最多可以设置20个标签。 • 标签键不可以重复,最长为64个字符;标签值可以为空,最长为128个字符。标签键和标签值都不能以 aliyun 、 acs : 、 https:// 或 http:// 开头。
key	String	key	标签key值。
value	String	value	标签value值。
node_name_m ode	String	customized,test.,5,.	自定义节点名称。
overwrite_host name	Boolean	false	是否覆盖原主机名。
runtime	String	docker	容器运行时名称。
runtime_version	String	19.03.5	容器运行时版本。
taints	Array of taint		节点污点信息。污点和容忍度(Toleration)相 互配合,可以用来避免Pod被分配到不合适的节 点上。更多信息,请参见taint-and- toleration。

名称	类型	示例值	描述
effect	String	NoSchedule	调度策略。取值: NoSchedule : 不能容忍,但仅影响调度过程,已被调度的Pod不受影响,仅对新增加的Pod生效。 NoExecute : 不能容忍,当污点变动时,Pod对象会被驱逐。 PreferNoSchedule : 柔性约束,节点现存Pod不受影响。尽量不去满足不合要求的Pod调度到节点上。 默认值: NoSchedule 。
key	String	key	污点key值。
value	String	value	污点value值。
unschedulable	Boolean	false	设置节点不可调度 ● true : 节点设置为不可调度 ● false : 节点可调度
user_data	String	lyEvYmluL3NoCmVja G8glkhlbGxvlEFDSyEi	节点池自定义数据。更多详情,请参见_云服务器 ECS_的_实例自定义数据_章节。
nodepool_info	Struct		节点池配置。
created	String	2020-09- 27T19:14:09.156823 496+08:00	节点池创建时间。
is_default	Boolean	true	是否为默认节点池,通常一个集群仅有一个默认节点池。取值: true:设置为默认节点池。 false:不设置为默认节点池。
name	String	default-nodepool	节点池名称。 命名规则:由数字、汉字、英文字符或短划线 (-)组成,长度范围1~63个字符,且不能以短 划线(-)开头。
nodepool_id	String	np615c0e096612421 6a0412e10afe0****	节点池ID。

名称	类型	示例值	描述
region_id	String	cn-beijing	地域ID。
resource_group _id	String	rg- acfmyvw3wjmb****	资源组ID。
type	String	ess	节点池类型。
updated	String	2020-09- 27T20:37:46+08:00	节点池更新时间。
scaling_group	Struct		扩容组配置。
data_disks	Array of data_disk		节点数据盘类型、大小等配置的组合。
auto_snapshot _policy_id	String	sp- 2zej1nogjvovnz4z*** *	开启云盘备份时的自动备份策略。
category	String	cloud_ssd	数据盘类型
encrypted	String	true	是否对数据盘加密。
size	Long	40	数据盘大小,取值范围: 40~32767
image_id	String	aliyun_2_1903_x64_2 0G_alibase_2020052 9.vhd	自定义镜像ID,可通 过 DescribeKubernetesVersionMetadata 查询系统支持的镜像。
instance_charg e_type	String	PayByBandwidth	节点公网IP网络计费类型。
instance_types	List	["ecs.n4.large"]	节点实例规格。
internet_charge _type	String	PayByTraffic	公网IP收费类型。
internet_max_b andwidth_out	Integer	10	节点公网IP出带宽最大值,单位为Mbps(Mega bit per second),取值范围:1~100。

名称	类型	示例值	描述
login_passwor d	String	*****	SSH登录密码。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)。 出于安全性考虑,密码查询结果经过加密处理。
multi_az_policy	String	BALANCE	多可用区伸缩组ECS实例扩缩容策略。
platform	String	AliyunLinux	操作系统发行版。取值: CentOS : CentOs镜像 AliyunLinux : AliyunLinux镜像 WindowsCore : Windows镜像 CentOS 7.6 : CentOS 7.6镜像 ARM CentOS : CentOS 7.7镜像 NeoKylin : 麒麟操作系统 默认值: AliyunLinux 。
ram_policy	String	KubernetesWorkerR ole-021dc54f-929b- 437a-8ae0- 34c24d3e****	Worker RAM 角色名称,授权ECS实例为集群 Woker节点。
scaling_group_i d	String	asg- 2zeieod8giqmov7z* ***	伸缩组ID。
scaling_policy	String	release	伸缩组模式,取值: ● release : 标准模式,根据申请资源值的使用量,通过创建、释放ECS的方式进行伸缩。 ● recycle : 极速模式,通过创建、停机、启动的方式进行伸缩,提高再次伸缩的速度。
security_group_ id	String	sg- 2ze60ockeekspl3d** **	安全组ID。
system_disk_ca tegory	String	cloud_efficiency	节点系统盘类型,取值: ● cloud_efficiency : 高效云盘。 ● cloud_ssd : SSD云盘。

名称	类型	示例值	描述
system_disk_si ze	Integer	120	节点系统盘大小,单位为GiB。 取值范围:20~500
tags	Array of tag		仅为ECS实例添加标签。 标签键不可以重复,最大长度为128个字符;标 签键和标签值都不能以 aliyun 、 acs: 开 头,或包含 https:// 、 http:// 。
key	String	key	标签key值。
value	String	value	标签value值。
vswitch_ids	List	["vsw- 2ze3ds0mdip0hdz8i ****"]	虚拟交换机ID。
status	Struct		节点池状态配置。
failed_nodes	Integer	0	失败节点数。
healthy_nodes	Integer	3	健康节点数。
initial_nodes	Integer	0	正在创建节点数。
offline_nodes	Integer	0	离线节点数。
removing_node s	Integer	0	正在被移除节点数。
serving_nodes	Integer	3	工作中节点数。
state	String	active	节点池状态,取值: • active : 已激活。 • scaling : 伸缩中。 • removing : 节点移除中。 • deleting : 删除中。 • updating : 更新中。

名称	类型	示例值	描述
total_nodes	Integer	3	节点池内总节点数。
tee_config	Struct		加密计算节集群配置。
tee_enable	Boolean	false	是否开启加密计算集群,取值: true : 开启。 false : 不开启。

示例

请求示例

```
GET /clusters/[ClusterId]/nodepools/[NodepoolId] HTTP/1.1
公共请求头
{
    "ClusterId": "c23421cfa74454bc8b37163fd19af****",
    "NodepoolId": "np31da1b38983f4511b490fc62108a****"
}
```

正常返回示例

XML 格式

```
<tee config>
   <tee_enable>false</tee_enable>
</tee config>
<scaling group>
   <scaling group id>asg-2zeieod8giqmov7z****</scaling group id>
   <internet charge type>PayByTraffic</internet charge type>
   <ram policy>KubernetesWorkerRole-021dc54f-929b-437a-8ae0-34c24d3e****/ram policy>
   <instance_charge_type>PayByBandwidth</instance_charge_type>
   <login password>******/login password>
   <system disk category>cloud efficiency</system disk category>
   <internet max bandwidth out>10</internet max bandwidth out>
   <security group id>sg-2ze60ockeekspl3d****</security group id>
   <multi az policy>BALANCE</multi_az_policy>
   <image id>aliyun 2 1903 x64 20G alibase 20200529.vhd</image id>
   <system_disk_size>120</system_disk_size>
   <platform>AliyunLinux</platform>
   <scaling policy>release</scaling policy>
    <data disks>
       <auto snapshot_policy_id>sp-2zej1nogjvovnz4z****</auto_snapshot_policy_id>
       <encrypted>true</encrypted>
       <size>40</size>
       <category>cloud ssd</category>
    </data disks>
    <tags>
       <value>value</value>
```

```
<key>key</key>
    </tags>
    <instance types>ecs.n4.large</instance types>
    <vswitch ids>vsw-2ze3ds0mdip0hdz8i****</vswitch ids>
</scaling group>
<kubernetes config>
   <runtime_version>19.03.5</runtime_version>
   <cpu policy>none</cpu policy>
   <cms enabled>true</cms enabled>
    <overwrite hostname>false</overwrite hostname>
   <runtime>docker</runtime>
   <user data>IyEvYmluL3NoCmVjaG8gIkhlbGxvIEFDSyEi</user data>
   <unschedulable>false</unschedulable>
    <node name mode>customized,test.,5,.com</node name mode>
   <labels>
       <value>value</value>
       <key>key</key>
    </labels>
    <taints>
       <effect>NoSchedule</effect>
       <value>value</value>
        <key>key</key>
    </taints>
</kubernetes config>
<auto_scaling>
    <enable>true</enable>
   <min instances>1</min instances>
   <max instances>10</max instances>
   <type>cpu</type>
</auto scaling>
<nodepool info>
   <resource group id>rg-acfmyvw3wjmb****</resource group id>
   <created>2020-09-27T19:14:09.156823496+08:00</created>
    <name>default-nodepool</name>
   <region id>cn-beijing</region id>
   <type>ess</type>
   <is default>true</is default>
    <updated>2020-09-27T20:37:46+08:00</updated>
    <nodepool_id>np615c0e0966124216a0412e10afe0****</nodepool_id>
</nodepool info>
<status>
   <serving nodes>3</serving nodes>
   <total nodes>3</total nodes>
   <state>active</state>
   <initial nodes>0</initial nodes>
   <removing_nodes>0</removing_nodes>
   <offline nodes>0</offline nodes>
   <failed nodes>0</failed nodes>
   <healthy nodes>3</healthy nodes>
</status>
```

```
JSON 格式
```

```
{
```

```
"tee config": {
   "tee enable": false
"scaling_group": {
   "scaling group id": "asg-2zeieod8giqmov7z****",
   "internet charge type": "PayByTraffic",
   "ram policy": "KubernetesWorkerRole-021dc54f-929b-437a-8ae0-34c24d3e****",
   "instance_charge_type": "PayByBandwidth",
   "login password": "******",
   "system_disk_category": "cloud_efficiency",
   "internet max bandwidth out": 10,
   "security group id": "sg-2ze60ockeekspl3d****",
   "multi az policy": "BALANCE",
   "image id": "aliyun 2 1903 x64 20G alibase 20200529.vhd",
   "system disk size": 120,
   "platform": "AliyunLinux",
   "scaling_policy": "release",
   "data disks": {
       "auto snapshot policy id": "sp-2zej1nogjvovnz4z****",
       "encrypted": true,
       "size": 40,
       "category": "cloud ssd"
   },
   "tags": {
       "value": "value",
       "key": "key"
   "instance types": ["ecs.n4.large"],
   "vswitch ids": ["vsw-2ze3ds0mdip0hdz8i****"]
},
"kubernetes config": {
   "runtime version": "19.03.5",
   "cpu policy": "none",
   "cms enabled": true,
   "overwrite hostname": false,
   "runtime": "docker",
   "user data": "IyEvYmluL3NoCmVjaG8gIkhlbGxvIEFDSyEi",
   "unschedulable": false,
   "node name mode": "customized, test., 5, .com",
   "labels": {
       "value": "value",
       "key": "key"
   "taints": {
       "effect": "NoSchedule",
       "value": "value",
       "key": "key"
},
"auto_scaling": {
   "enable": true,
   "min instances": 1,
   "max instances": 10,
   "type": "cpu"
```

```
},
"nodepool info": {
    "resource_group_id": "rg-acfmyvw3wjmb****",
    "created": "2020-09-27T19:14:09.156823496+08:00",
    "name": "default-nodepool",
    "region id": "cn-beijing",
    "type": "ess",
    "is default": true,
    "updated": "2020-09-27T20:37:46+08:00",
    "nodepool id": "np615c0e0966124216a0412e10afe0****"
},
"status": {
    "serving nodes": 3,
   "total nodes": 3,
    "state": "active",
    "initial nodes": 0,
    "removing_nodes": 0,
    "offline_nodes": 0,
    "failed nodes": 0,
    "healthy nodes": 3
}
```

2.3.5.6. 删除节点池

调用DeleteClusterNodepool删除节点池。

请求头

该接口使用公共请求头, 无特殊请求头。请参见公共参数文档。

请求语法

```
DELETE /clusters/[ClusterId]/nodepools/[NodepoolId] HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c61da77e8bfbc 4c4c999af2b51 b65****	集群ID。
Nodepoolid	String	Path	是	np30db56bcac 7843dca90b999 c8928****	节点池ID。

返回数据

名称	类型	示例值	描述
request_id	String	7263C978-3DBD- 4E06-B319- 793B38A2****	请求ID。

示例

请求示例

```
DELETE /clusters/[ClusterId]/nodepools/[NodepoolId] HTTP/1.1
公共请求头
{
    "ClusterId": "c61da77e8bfbc4c4c999af2b51b65****",
    "NodepoolId": "np30db56bcac7843dca90b999c8928****"
}
```

正常返回示例

XML 格式

```
<request_id>7263C978-3DBD-4E06-B319-793B38A2****</request_id>
```

JSON 格式

```
{
    "request_id": "7263C978-3DBD-4E06-B319-793B38A2****"
}
```

2.4. 模板

2.4.1. 创建编排模板

调用CreateTemplate创建一个编排模板。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

```
POST|PUT /templates HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
body	Struct	Body	否		

名称	类型	位置	是否必选	示例值	描述
template	String		是	service- account- template	YAML格式的模板内容。
name	String		是	apiVersion: v1\nkind: ServiceAccount \nmetadata:\n name: test-sa	YAML格式的模板内容。
description	String		否	this is test	编排模板描述。
template_type	String		否	kubernetes	模板类型,取值可以是任意值。 • 当取值是 kubernetes 时将在控制台的编排模板页面展示该模板。 • 当取值是 compose 时将不在控制台展示。 推荐设置为 kubernetes 。 默认值: compose 。
tags	String		否	test	编排模板标签。

返回数据

名称	类型	示例值	描述
template_id	String	ba1fe77b-b01e- 4640-b77e- 8f1b80e3****	编排模板ID。

示例

请求示例

```
POST /templates HTTP/1.1
公共请求头
{
    "name": "service-account-template",
    "template": "apiVersion:v1\\nkind:ServiceAccount\\nmetadata:\\n name:test-sa",
    "tags": "test",
    "description": "this is test",
    "template_type": "kubernetes"
}
```

正常返回示例

XML 格式

```
<template_id>balfe77b-b01e-4640-b77e-8f1b80e3****/template_id>
```

```
JSON 格式
```

```
{
    "template_id": "balfe77b-b01e-4640-b77e-8f1b80e3****"
}
```

2.4.2. 删除编排模板

调用DeleteTemplate删除指定编排模板。

请求头

该接口使用公共请求头, 无特殊请求头。请参见公共参数文档。

请求语法

```
DELETE /templates/[TemplateId] HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
TemplateId	String	Path	是	20663627- 81ee-4cfe- 9b69- c96fcb84****	编排模板ID。

示例

请求示例

```
DELETE /templates/[TemplateId] HTTP/1.1
公共请求头
{
    "TemplateId": "20663627-81ee-4cfe-9b69-c96fcb84****"
}
```

正常返回示例

XML 格式

```
HTTP/1.1 200 OK
```

JSON 格式

```
HTTP/1.1 200 OK
```

2.4.3. 查询所有编排模板

调用DescribeTemplates查询编排模板。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

GET /templates HTTP/1.1

请求参数

名称	类型	位置	是否必选	示例值	描述
template_type	String	Query	是	kubernetes	模板类型,值可以是任意值。 • 当取值是 kubernetes 时将在控制台的编排模板页面展示该模板。 • 当取值是 compose 时将不在控制台的编排模板页面展示该模板。 默认值: kubernetes 。
page_num	Long	Query	否	1	对查询结果进行分页处理,指定返回第几页的数据。
page_size	Long	Query	否	10	对查询结果进行分页处理,指定每页包含的数据条数。 默认值:10。

返回数据

名称	类型	示例值	描述
page_info	Struct		分页信息。
page_number	Long	20	展示当前页数。
page_size	Long	3	单页最大数据条数。
total_count	Long	50	结果总数。

名称	类型	示例值	描述
templates	Array of template		模板列表。
acl	String	private	用户部署模板的访问权限,取值: private : 私有的。 public : 公共的。 shared : 可共享的。 默认值: private 。
created	String	2020-06- 10T16:30:16+08:00	部署模板创建时间。
description	String	a web server	部署模板描述信息。
id	String	874ec485-e7e6- 4373-8a3b- 47bde8ae****	部署模板ID。
name	String	webserver	部署模板名称。
tags	String	kubernetes	部署模板的标签,如果不显式指定,默认为模板名称。
template	String	apiVersion: apps/v1\nkind: Deployment\nmeta data:\n name: nginx-deployment- basic\n labels:\n app: nginx\nspec:\n replicas: 2\n selector:\n matchLabels:\n app: nginx\n template:\n metadata:\n labels:\n app: nginx\n spec:\n containers:\n - name: nginx\n image: busybox:latest\n ports:\n - containerPort: 80	YAML格式的模板内容。

名称	类型	示例值	描述
template_type	String	kubernetes	模板类型,值可以是任意值。 • 当取值是 kubernetes 时将在控制台的编排模板页面展示该模板。 • 当取值是 compose 时将在控制台Swarm集群页面显示该模板(已废弃)。
template_with_ hist_id	String	ad81d115-7c8b- 47e7-a222- 9c28d7f6e588	模板关联的父模板ID,用于实现模板多版本功能(同一模板的不同版本拥有相同的 template_with_hist_id 值)。
updated	String	2020-06- 10T16:30:16+08:00	部署模板更新时间。

示例

请求示例

```
GET /templates HTTP/1.1
公共请求头
    "template\_type": "kubernetes"
```

正常返回示例

XML 格式

```
<templates>
```

<template>apiVersion: apps/v1\nkind: Deployment\nmetadata:\n name: nginx-deployment-bas ic\n labels:\n app: nginx\nspec:\n replicas: 2\n selector:\n matchLabels:\n app: nginx\n te usybox:latest\n ports:\n - containerPort: 80</template>

```
mplate:\n metadata:\n labels:\n app: nginx\n spec:\n containers:\n - name: nginx\n image: b
    <template with hist id>ad81d115-7c8b-47e7-a222-9c28d7f6e588</template_with_hist_id>
    <created>2020-06-10T16:30:16+08:00</created>
    <name>webserver</name>
    <description>a web server</description>
    <template_type>kubernetes</template_type>
    <acl>private</acl>
    <id>874ec485-e7e6-4373-8a3b-47bde8ae****</id>
    <updated>2020-06-10T16:30:16+08:00</updated>
    <tags>kubernetes</tags>
</templates>
<page info>
    <page_number>20</page_number>
    <total count>50</total count>
   <page_size>3</page_size>
</page_info>
```

JSON 格式

```
"templates": {
        "template": "apiVersion: apps/v1\\nkind: Deployment\\nmetadata:\\n name: nginx-depl
oyment-basic\\n labels:\\n app: nginx\\nspec:\\n replicas: 2\\n selector:\\n matchLabels:\\
n app: nginx\n template:\n metadata:\n labels:\n app: <math>nginx\n spec:\n containers:\n
- name: nginx\\n image: busybox:latest\\n ports:\\n - containerPort: 80",
        "template_with_hist_id": "ad81d115-7c8b-47e7-a222-9c28d7f6e588",
        "created": "2020-06-10T16:30:16+08:00",
        "name": "webserver",
        "description": "a web server",
        "template type": "kubernetes",
        "acl": "private",
        "id": "874ec485-e7e6-4373-8a3b-47bde8ae****",
        "updated": "2020-06-10T16:30:16+08:00",
        "tags": "kubernetes"
    },
    "page info": {
        "page number": 20,
        "total count": 50,
        "page size": 3
```

2.4.4. 查询编排模板详情

调用DescribeTemplateAttribute查看指定模板的详细信息。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

```
GET /templates/[TemplateId] HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
TemplateId	String	Path	是	874ec485- e7e6-4373- 8a3b- 47bde8a****	模板ID。

名称	类型	位置	是否必选	示例值	描述
template_type	String	Query	否	kubernetes	模板类型,其值可以是任意值。 • 当取值是 kubernetes 时将在控制台的编排模板页面展示该模板。 • 当取值非 kubernetes 时在控制台的编排模板页面将不会展示该模板,推荐使用 kubernetes 。 默认值: kubernetes 。

返回数据

名称	类型	示例值	描述
body	Array of body		模板详情。
acl	String	private	编排模板访问权限。
created	String	2020-09- 16T19:21:29+08:00	编排模板创建时间。
description	String	test template	编排模板描述信息。
id	String	72d20cf8-a533- 4ea9-a10d- e7630d3d****	编排模板ID,每次变更都会有一个模板ID。
name	String	web	编排模板名称。
tags	String	sa	部署模板的标签。
template	String	apiVersion: V1\n***	编排模板YAML内容。
template_type	String	kubernetes	模板类型,其值可以是任意值。 • 当取值是 kubernetes 时将在控制台的编排模板页面展示该模板。 • 当取值非 kubernetes 时在控制台的编排模板页面将不会展示该模板,推荐使用 kubernetes 。 默认值: kubernetes 。

名称	类型	示例值	描述
template_with_ hist_id	String	874ec485-e7e6- 4373-8a3b- 47bde8ae****	编排模板唯一ID,不随模板更新而改变。
updated	String	2020-09- 16T19:21:29+08:00	编排模板更新时间。

示例

请求示例

```
GET /templates/[TemplateId] HTTP/1.1
公共请求头
{
    "TemplateId": "874ec485-e7e6-4373-8a3b-47bde8a****"
}
```

正常返回示例

XML 格式

JSON 格式

```
"body": {
    "template": "apiVersion: V1\\n***",
    "template_with_hist_id": "874ec485-e7e6-4373-8a3b-47bde8ae****",
    "created": "2020-09-16T19:21:29+08:00",
    "name": "web",
    "description": "test template",
    "template_type": "kubernetes",
    "acl": "private",
    "id": "72d20cf8-a533-4ea9-a10d-e7630d3d****",
    "updated": "2020-09-16T19:21:29+08:00",
    "tags": "sa"
}
```

2.4.5. 更新编排模板

调用UpdateTemplate更新一个编排模板。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

PUT /templates/[TemplateId] HTTP/1.1

请求参数

名称	类型	位置	是否必选	示例值	描述
Templateld	String	Path	是	72d20cf8- a533-4ea9- a10d- e7630d3d2708	模板ID。
body	Struct	Body	否		
template	String		否	service- account- template	YAML格式的模板内容。
name	String		否	apiVersion: v1\nkind: ServiceAccount \nmetadata:\n name: test-sa	YAML格式的模板内容。
description	String		否	this is test	编排模板描述。
template_type	String		否	kubernetes	模板类型,取值可以是任意值。 • 当取值是 kubernetes 时将在控制台的编排模板页面展示该模板。 • 当取值是 compose 时将不在控制台展示。 推荐设置为 kubernetes 。 默认值: compose 。
tags	String		否	test	编排模板标签。

示例

请求示例

```
PUT /templates/[TemplateId] HTTP/1.1
公共请求头

{
    "name": "service-account-template",
    "template": "apiVersion:v1\\nkind:ServiceAccount\\nmetadata:\\n name:test-sa",
    "tags": "test",
    "description": "this is test",
    "template_type": "kubernetes"
}
```

正常返回示例

JSON 格式

HTTP/1.1 200 OK

2.5. 组件

2.5.1. 安装集群组件

调用InstallClusterAddons为指定集群安装组件。

请求头

该接口使用公共请求头, 无特殊请求头。请参见公共参数文档。

请求语法

POST /clusters/[ClusterId]/components/install HTTP/1.1

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c82e6987e2961 451182edacd74 faf****	集群ID。
body	Array	Body	否		请求体参数。
config	String		否	{\"IngressDashb oardEnabled\": \"true\",\"sls_p roject_name\": \"your_sls_proj ect_name\"}"}	组件配置信息。
name	String		是	ags-metrics- collector	组件名称。

名称	类型	位置	是否必选	示例值	描述
version	String		是	v1.0.0.2- cc3b2d6-aliyun	组件版本。 ② 说明 可以通过调用 DescribeClusterAddonsVersion 查询版本号。更多信息,请参 见查询集群版本组件。

示例

请求示例

```
POST /clusters/[ClusterId]/components/install HTTP/1.1
公共请求头
{
    "ClusterId": "c82e6987e2961451182edacd74faf****"
}
```

正常返回示例

XML 格式

HTTP/1.1 200 OK

JSON 格式

HTTP/1.1 200 OK

2.5.2. 查询集群组件详情

调用DescribeAddons查询集群安装的组件详情。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

GET /clusters/components/metadata HTTP/1.1

请求参数

2 76		是否必选	示例值	描述
------	---------	------	-----	----

名称	类型	位置	是否必选	示例值	描述
region	String	Query	是	cn-beijing	集群所在地域ID。
cluster_type	String	Query	否	Kubernetes	集群类型,取值: Kubernetes : Kubernetes专有版 或边缘专有版集群

返回数据

名称	类型	示例值	描述
Component Gro ups	Array of Component Gro up		组件分组列表。
group_name	String	storage	组件分组名称。
items	Array of item		该分组内包含的组件名称。
name	String	flexvolume	组件名称。
StandardComp onents	Struct		标准组件。
key	Struct		标准组件的详细信息。
description	String	test	组件功能说明。
disabled	Boolean	false	是否禁止默认安装,集群创建时除了安装集群所需的组件外,还会额外安装一些与日志或者路由相关的组件(例如:Ingress),如果您不想默认安装,可通过设置该字段禁止默认安装。取值: true : 禁止默认安装。 false : 默认安装。
name	String	ack-arena	组件名称。

名称	类型	示例值	描述
required	String	false	是否为集群必需组件。取值: true : 必需组件,集群创建时必须要安装的组件。 false : 非必需组件,集群创建完成后,可通过组件管理进行安装。
version	String	0.5.0	组件版本。

示例

请求示例

```
GET /clusters/components/metadata HTTP/1.1
公共请求头
{
    "region": "cn-beijing"
}
```

正常返回示例

XML 格式

JSON 格式

2.5.3. 查询集群组件版本

调用DescribeClusterAddonsVersion查询指定集群安装的所有组件的信息。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

```
GET /clusters/[ClusterId]/components/version HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c82e6987e2961 451182edacd74 faf****	集群ID。

返回数据

名称	类型	示例值	描述
key	Struct		组件详情。
can_upgrade	Boolean	false	组件是否可升级。取值: ● true : 组件可升级。 ● false : 组件不可升级。

名称	类型	示例值	描述
changed	String	false	组件是否有更新。取值: ● true : 已更新。 ● false : 未更新。
component_na me	String	ack-node-problem- detector	组件名称。
exist	Boolean	true	组件是否已经安装。取值: ● true : 已经安装。 ● false : 未安装。
message	String	节点故障检测以及上 报至事件中心的插件	组件是否可升级的额外说明。
next_version	String	1.2.1	下一个可升级版本。
policy	String	overwrite	组件升级策略。 overwrite : 强制覆盖。 默认值: overwrite 。
ready_to_upgra de	String	false	组件是否满足升级条件。取值: ● true : 满足升级条件。 ● false : 不满足升级条件。
required	Boolean	false	组件是否为集群必需组件。取值: ● true : 为必需组件。 ● false : 非必需组件。
value	String	npd:\n # Custom monitor definitions to add to Node Problem Detector ****	组件安装时,用户自定义的参数值。
version	String	1.2.1	组件当前版本。

示例

请求示例

```
GET /clusters/[ClusterId]/components/version HTTP/1.1
公共请求头
{
    "ClusterId": "c82e6987e2961451182edacd74faf****"
}
```

正常返回示例

XML 格式

JSON 格式

```
"key": {
    "exist": true,
    "component_name": "ack-node-problem-detector",
    "can_upgrade": false,
    "ready_to_upgrade": false,
    "message": "节点故障检测以及上报至事件中心的插件",
    "value": "npd:\\n # Custom monitor definitions to add to Node Problem Detector ****
",
    "version": "1.2.1",
    "required": false,
    "policy": "overwrite",
    "changed": false,
    "next_version": "1.2.1"
}
```

2.5.4. 查询集群组件升级状态

调用DescribeClusterAddonsUpgradeStatus同时查询多个组件的升级状态。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共参数文档。

请求语法

```
GET /clusters/[ClusterId]/components/upgradestatus HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c23421cfa7445 4bc8b37163fd1 9af****	集群ID。
componentids	List <str ing></str 	Query	是	["metric- server"]	组件名称。

示例

请求示例

```
GET /clusters/[ClusterId]/components/upgradestatus HTTP/1.1
公共请求头
{
    "ClusterId": "c23421cfa74454bc8b37163fd19af****",
    "componentIds": "[metric-server"]
}
```

正常返回示例

```
XML 格式
```

```
HTTP/1.1 200 OK
```

JSON 格式

HTTP/1.1 200 OK

2.5.5. 升级集群组件

调用UpgradeClusterAddons将指定组件升级到指定版本。

请求头

该接口使用公共请求头, 无特殊请求头。请参见公共参数文档。

请求语法

```
POST /clusters/[ClusterId]/components/upgrade HTTP/1.1
```

请求参数

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	cf4299b79b3e3 4226abfdc80a4 bda****	集群ID。
body	Array	Body	否		请求参数。
component_na me	String		是	coredns	组件名称。
next_version	String		是	1.6.7	可升级版本。调 用 DescribeClusterAddonsVersi on 参数可以查看可以升级的版本。
version	String		否	v1.6.2	当前版本。

示例

请求示例

```
POST /clusters/[ClusterId]/components/upgrade HTTP/1.1
公共请求头
{
    "ClusterId": "cf4299b79b3e34226abfdc80a4bda****"
}
```

正常返回示例

JSON 格式

HTTP/1.1 200 OK

2.5.6. 卸载集群组件

调用UnInstallClusterAddons卸载指定集群的组件。

请求头

该接口使用公共请求头,无特殊请求头。请参见公共请求参数文档。

请求语法

POST /clusters/[ClusterId]/components/uninstall HTTP/1.1

请求参数

开发指南·集群 API 列表 容器服务Kubernet es版

名称	类型	位置	是否必选	示例值	描述
ClusterId	String	Path	是	c5b5e80b0b64 a4bf6939d2d8f bbc5****	集群ID。
addons	Array	Body	否		组件列表,结构为数组。
name	String		否	ack-node- problem- detector	组件名称。

示例

请求示例

```
POST /clusters/[ClusterId]/components/uninstall HTTP/1.1
公共请求头
{
    "ClusterId": "c5b5e80b0b64a4bf6939d2d8fbbc5****"
}
```

正常返回示例

JSON 格式

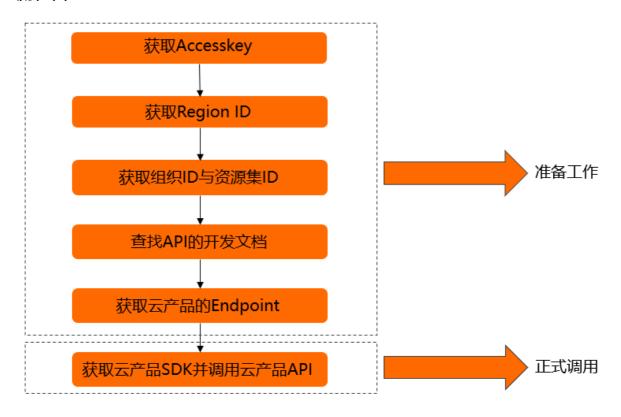
HTTP/1.1 200 OK

3.附录

3.1. ASAPI网关调用流程说明

专有云中绝大多数产品都可以通过ASAPI网关进行调用,本文以通过ASAPI网关调用专有云API为例,为您介绍调用流程及可以参考的文档资料。

流程图



调用步骤

- 1. 获取ASAPI的SDK。
 - 使用ASAPI调用API前, 您需要先获取ASAPI的SDK包, 请参见获取ASAPI的SDK包。
- 2. 获取ASAPI的Endpoint,请参见获取ASAPI的Endpoint。
- 3. 获取AccessKey。

获取调用时所需要的AccessKey参数。

? 说明

推荐使用个人AccessKey调用Apsara Uni-manager运营控制台及云产品接口。

- 4. 获取RegionID。
 - 获取调用时所需要的RegionID参数。
- 5. 获取组织ID与资源集ID。

组织ID需要通过API进行获取,获取方式请参见*Apsara Uni-manager运营控制台开发指南*中的章节*Get Or ganizat ionList*。

资源集ID需要通过API进行获取,获取方式请参见*Apsara Uni-manager运营控制台开发指南*中的章节*List ResourceGroup*。

? 说明

调用API时,通过Header参数组织ID(<x-acs-organizationid>)和资源集ID(<x-acs-resourcegroupid>)指定组织和资源集,实现实例资源的隔离查询。

- 组织ID (<x-acs-organizationid>) 不指定则默认获取当前用户所属的组织。
- 资源集ID(<x-acs-resourcegroupid>)不指定默认为空,如果指定资源集需要同时指定组织ID。

6. 查找API的开发文档。

您需要查看目标云产品的开发指南,了解API的功能与使用方式,查找开发指南的方式请参见开发文档 使用说明。

7. 通过SDK调用云产品API。

完成以上准备工作后,您可以使用SDK对API进行调用,调用的方式请根据实际使用的语言,在本手册中查看对应的调用文档。

以通过Java SDK调用ECS的DescribeInstances接口为例,调用的示例代码如下:

容器服务Kubernet es版 开发指南· <mark>附录</mark>

```
public static void main(String[] args) {
   ASClient client = new ASClient();
   //指定调用者的标示,仅用于记录日志
   client.setSdkSource("autoTest");
   // 准备请求参数
   Map<String, Object> parameters = new HashMap<String, Object>();;
   parameters.put("Product", "Ecs");//与对应API目录中的产品POPCode一致
   parameters.put("Action", "DescribeInstances"); //API名称,大小写需要与API目录中的一致
   parameters.put("Version", "2014-05-26"); //API版本
   //填写获取到的RegionID
   parameters.put("RegionId", "xxx");
   //填写获取到AccessKey ID和AccessKey Secret
   parameters.put("AccessKeyId", "xxx");
   parameters.put("AccessKeySecret", "xxx");
   //传入API文档中参数,请注意大小写敏感
   parameters.put("PageNumber",1);
   parameters.put("PageSize",10);
   Map<String, String> headers=new HashMap<>();
   // 输入当前操作的Apsara Uni-manager运营控制台组织ID
   headers.put("x-acs-organizationid", "3");
   // 输入当前操作的Apsara Uni-manager运营控制台资源集ID
   //headers.put("x-acs-resourcegroupid", "<your-resourcegroupid>");
   // 填写获取到ASAPI的Endpoint
   String endpoint="https://public.asapi.xxx.xxx/asapi/v3";
   // 发起请求
   String result = client.doPost(endpoint, headers, parameters);
   // 打印结果
   {\tt System.out.println(JSONObject.toJSONString(JSONObject.parseObject(result), Serializ}
erFeature.PrettyFormat));
```

3.2. 通过ASAPI网关调用专有云API

3.2.1. 获取ASAPI的Endpoint

本文将为您介绍三种获取Endpoint的方法及其校验方法。

获取方法

方法一: 通过服务注册变量获取Endpoint

- 1. 登录飞天基础运维平台。
 - i. 登录Apsara Uni-manager运维控制台。
 - ii. 在顶部菜单栏单击**产品运维**。
 - iii. 单击云平台运维下方的飞天基础运维平台。
- 2. 在左侧导航栏中单击报表。
- 3. 在全部报表页面搜索 服务注册变量 , 单击报表名称。



4. 在服务注册变量页面,单击Service旁的 = 图标,搜索 asapi 。



- 5. 在asapi的Service Registration列中,单击鼠标右键,选择显示更多。 在详情页面,查看的asapi.public.endpoint值。
- 6. 按照以下格式生成ASAPI的Endpoint。

https://<asapi.public.endpoint>/asapi/v3

方法二:通过集群资源生成Endpoint

- 1. 登录飞天基础运维平台。
- 2. 进入集群运维。
 - i. 在左侧导航栏中选择**运维>集群运维**。
 - ii. 在集群文本框中输入 ascm 。
 - iii. 单击操作列中的运维。
- 3. 获取domain值。
 - i. 单击集群资源页签。
 - ii. 在名称文本框中输入 asapi dns public 。

iii. 单击申请参数栏中的详情。

在申请参数页面查看domain值。

```
{
    "domain": "public.asapi.
    "ha_apply_type": "active=active",
    "name": "asapi_dns_public",
    "vip":
}
```

4. 按照以下格式生成ASAPI的Endpoint。

https://<domain>/asapi/v3

方法三: 自行拼装Endpoint

您还可以按照以下格式自行拼装Endpoint。

https://public.asapi.<region>.<internet-domain>/asapi/v3

? 说明

- 1. <region>: 替换为真实的地域ID值。
- 2. <internet-domain>:替换为专有云的根域名(特别提示是internet,不是intranet),可咨询专有云运维管理员。

假设region示例值为 cn-***-***-d01 , internet-domain示例值为 inter.***.***.com 。

完整的Endpoint示例值如下:

```
https://public.asapi.cn-***-***-d01.inter.***.com/asapi/v3
```

Endpoint (业务域名) 校验

输入以下代码, 若访问异常则表示业务域名不可用, 可使用历史业务域名。

curl https://public.asapi.\${global:region}.\${global:internet-domain}/api/version -k

? 说明

\${global:region} 和 \${global:internet-domain} 为变量,需要替换为您获取到的实际值。

访问正常示例:

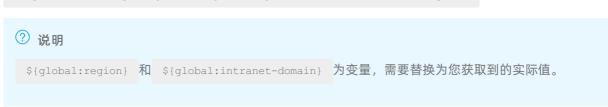
访问异常示例:

```
[root@color.org | DPS1_GDNX-CS #curl https://public.asapi.color.org | DPS1_GDNX-CS #curl https://public.asapi.color.org | DPS1_GDNX-CS | Curl: (35) Peer reports it experienced an internal error.
```

历史业务域名

仅支持HTTP协议,业务域名后缀为 intranet-domain ,业务域名结构如下:

http://server.asapi.\${global:region}.\${global:intranet-domain}/asapi/v3



操作步骤

获取变量\${global:region}、\${global:internet-domain}、\${global:intranet-domain}实际值的操作步骤如下:

1. 在地址栏中,输入Apsara Uni-manager运维控制台的访问地址*region-id*.aso.*intranet-domain-id*.com,按回车键。



? 说明

您可以单击页面右上角的下拉按钮来进行语言切换。

2. 输入正确的用户名及密码。

? 说明

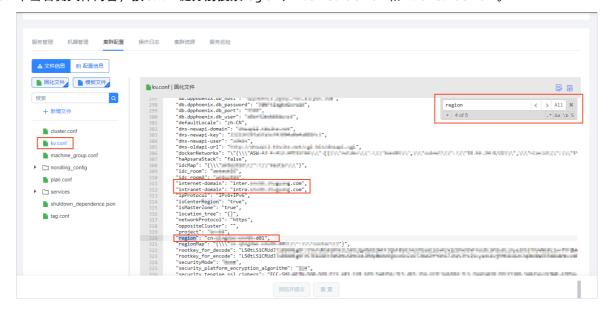
登录Apsara Uni-manager运维控制台的用户名和密码请从部署人员或管理员处获取。

首次登录Apsara Uni-manager运维控制台时,需要修改登录用户名的密码,请按照提示完成密码修改。 为提高安全性,密码必须满足以下要求:

- 。 英文大小写字母
- 阿拉伯数字 (0~9)
- 特殊符号,包括感叹号(!)、at(@)、井号(#)、美元符号(\$)、百分号(%)等
- 密码长度10~20个字符
- 3. 单击登录,进入Apsara Uni-manager运维控制台页面。
- 4. 在页面顶部导航栏中,选择**产品运维 > 云平台运维 > 飞天基础运维平台**,跳转到飞天基础运维平台 页面。
- 5. 在左侧菜单栏中,选择运维>集群运维。
- 6. 在**集群运维**页面搜索ascm-A, 并单击**集群**列的集群名称。



- 7. 单击集群配置页签。
- 8. 在集群配置页签的文件信息中,单击kv.conf。
- 9. 单击右侧文件内容,按Ctrl+F键分别搜索region、internet-domain和intranet-domain。



获取到的region值将替换
\${global:region}
, internet-domain值将替换

\${global:internet-domain}
, intranet-domain值将替换
\${global:intranet-domain}

② 说明
最终Endpoint (业务域名)中,
\${global:}
不保留。

3.2.2. 获取ASAPI的SDK包

专有云环境中使用ASAPI作为所有接入ASAPI产品的统一入口,并提供SDK进行产品侧接口的调用支持。

公网环境

公网环境下您可以进入专有云文档官网,在页面底部的其他资源中,下载对应版本ASAPI的SDK包。

非公网环境

若在非公网环境下使用ASAPI的SDK包,需要先在公网环境下进入专有云文档官网,在页面底部的**其他资源**中,下载对应版本ASAPI的SDK包和ASAPI的SDK依赖包。

ASAPI的SDK依赖包(JAVA)

greopld	artifactId	version
com.alibaba	alicrypto-java-aliyun	1.0.4
com.alibaba	fastjson	1.2.68
org.bouncycastle	bcprov-jdk15to18	1.66
org.bouncycastle	bcpkix-jdk15on	1.65
org.bouncycastle	bcprov-jdk15on	1.65
com.aliyun	gmsse	1.1.0
org.apache.commons	commons-lang3	3.0
commons-codec	commons-codec	1.11
commons-logging	commons-logging	1.2
org.apache.httpcomponents	httpcore	4.4.11
org.apache.httpcomponents	httpclient	4.5.7

ASAPI的SDK依赖包 (Python)

Package	Version
---------	---------

Package	Version
aliyun-python-sdk-core	2.13.35
certifi	2021.5.30
cffi	1.14.6
charset-normalizer	2.0.4
cryptography	3.4.8
idna	3.2
jmespath	0.10.0
pycparser	2.20
requests	2.26.0
urllib3	1.26.6

? 说明 Package版本支持使用更高或最新版本。

3.3. STS概要

3.3.1. 什么是STS

阿里云STS(Security Token Service)是阿里云提供的一种临时访问权限管理服务。RAM提供RAM用户和RAM角色两种身份。其中,RAM角色不具备永久身份凭证,而只能通过STS获取可以自定义时效和访问权限的临时身份凭证,即安全令牌(STS Token)。

使用RAM用户扮演角色时获取STS Token

有权限的RAM用户可以使用其访问密钥调用AssumeRole接口,以获取某个RAM角色的STS Token,从而使用STS Token访问阿里云资源。

产品优势

- 使用STS Token,减少长期访问密钥(Accesskey)泄露的风险。
- STS Token具有时效性,可以自定义有效期,到期后将自动失效,无需定期轮换。
- 可以为STS Token绑定自定义权限策略,提供更加灵活和精细的云资源授权。

3.3.2. AssumeRole

RAM用户调用AssumeRole接口获取一个扮演RAM角色的临时身份凭证STS Token。该接口中RAM用户扮演的是可信实体为阿里云账号类型的RAM角色。

请求参数

开发指南· <mark>附录</mark> 容器服务 Kubernet es版

名称	类型	是否必选	示例值	描述
Action	String	是	AssumeRole	要执行的操作。取值:AssumeRole
RoleArn	String	是	acs:ram::1234567 89012****:role/ad minrole	要扮演的RAM角色ARN。 该角色是可信实体为阿里云账号类型的RAM 角色。
RoleSessionNam e	String	是	alice	角色会话名称。 该参数为用户自定义参数。通常设置为调用 该API的用户身份,例如:用户名。在操作审 计日志中,即使是同一个RAM角色执行的操 作,也可以根据不同 的RoleSessionName来区分实际操作者, 以实现用户级别的访问审计。 长度为2~64个字符,可包含英文字母、数 字、半角句号(.)、at(@)、短划线(-) 和下划线(_)。
Policy	String	否	{"Statement": [{"Action": ["*"],"Effect": "Allow","Resource ": ["*"]}],"Version":"1	为STS Token额外添加的一个权限策略,进一步限制STS Token的权限。具体如下: 长度为1~1024个字符。

返回结果

名称	类型	示例值	描述
RequestId	String	6894B13B-6D71- 4EF5-88FA- F32781734A7F	请求ID。
Credentials	无	无	访问凭证。
AccessKeyId	String	STS.L4aBSCSJVMuKg5 U1****	访问密钥ID。

名称	类型	示例值	描述
AccessKeySecret	String	wyLTSmsyPGP1ohvv w8xYgB29dlGl8KMiH2 pK****	访问密钥。
SecurityToken	String	*****	安全令牌。
Expiration	String	2015-04- 09T11:52:19Z	失效时间。
AssumedRoleUser	无	无	角色扮演时的临时身份。
Arn	String	acs:ram::123456789 012****:role/adminr ole/alice	临时身份的ARN。
AssumedRoleId	String	34458433936495**** : alice	临时身份的ID。

示例

引入Pom依赖

操作需要依赖阿里云核心库来实现,在Pom中引入阿里云核心库包的操作如下:

```
<dependency>
  <groupId>com.aliyun</groupId>
   <artifactId>aliyun-java-sdk-core</artifactId>
   <version>[4.4.9,5.0.0)</version>
</dependency>
```

调用API获取STS AccessKey三元组

以下代码示例为通过调用 Sts::AssumeRole::2015-04-01 获取STS AccessKey三元组信息。

```
package com.aliyuns.test;

import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpClientConfig:
```

```
IMPOIL OOM • AII Y AIIOD • IIC CP • IIC CPOILICITOOTILIG,
import com.aliyuncs.http.MethodType;
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.profile.DefaultProfile;
import org.apache.commons.lang3.StringUtils;
public class STSAccessKeyDemo {
   public static void main(String[] args) {
       // 创建DefaultAcsClient实例并初始化。
       DefaultProfile profile = DefaultProfile.getProfile("<yourRegionID>", "<yourAccessKe</pre>
yID>", "<yourAccessKeySecret>");
       HttpClientConfig clientConfig = HttpClientConfig.getDefault();
       clientConfig.setIgnoreSSLCerts(true);
       clientConfig.setProtocolType(ProtocolType.HTTPS);
       profile.setHttpClientConfig(clientConfig);
       IAcsClient client = new DefaultAcsClient(profile);
       CommonRequest request = new CommonRequest();
        // "sts.aliyuns.com"要替换为实际的STS Endpoint, 一般格式为sts-vpc.${global:region}.${g
lobal:internet-domain}
       request.setSysDomain("sts.aliyuns.com");
       request.setSysProduct("Sts");
       request.setSysAction("AssumeRole");
       request.setSysVersion("2015-04-01");
       request.setSysMethod(MethodType.POST);
       request.setHttpContentType(FormatType.FORM);
        // 业务参数,设置本次调用Session的别名,"<yourRoleSessionName>"请替换为实际值
       request.putQueryParameter("RoleSessionName", "<yourRoleSessionName>");
       // 扮演的Role角色ARN, "<yourRoleArn>"请替换为上一步获取到的实际Ram Role ARN
       request.putQueryParameter("RoleArn", "<yourRoleArn>");
       try {
           CommonResponse response = client.getCommonResponse(request);
           String assumeRole = response.getData();
            if (StringUtils.isEmpty(assumeRole)) return;
           JSONObject assumeRoleJson = JSONObject.parseObject(assumeRole);
           JSONObject credentials = assumeRoleJson.getJSONObject("Credentials");
            // 获取到STS三元组
           String stsAK = credentials.getString("AccessKeyId");
           String stsSK = credentials.getString("AccessKeySecret");
           String stsToken = credentials.getString("SecurityToken");
            System.out.println("stsAk: " + stsAK +"\nstsSk: " + stsSK + "\nstsToken: " + st
sToken);
        } catch (ServerException e) {
           System.out.println(e.getErrCode());
            System.out.println(e.getErrMsq());
        } catch (ClientException e) {
           System.out.println(e.getErrCode());
           System.out.println(e.getErrMsg());
       }
   }
```

调用示例

下列代码以调用ECS的 DescribeZones 接口为示例。

```
package com.aliyuns.test;
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpClientConfig;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.profile.DefaultProfile;
public class DescribeZonesDemo {
   public static void main(String[] args) {
       // "<yourRegionID>"替换为真实的RegionID
       // "<yourStsAccesskey>", "<yourStsAccessKeySecret>"替换为上面获取到的STS三元组中stsAk、
stsSk
       DefaultProfile profile = DefaultProfile.getProfile("<yourRegionID>", "<yourStsAcces
sKey>", "<yourStsAccessKeySecret>");
       HttpClientConfig clientConfig = HttpClientConfig.getDefault();
       clientConfig.setIgnoreSSLCerts(true);
       clientConfig.setProtocolType(ProtocolType.HTTPS);
       profile.setHttpClientConfig(clientConfig);
       IAcsClient client = new DefaultAcsClient(profile);
       CommonRequest request = new CommonRequest();
       request.setSysDomain("ecs.aliyuns.com");
        request.setSysProduct("Ecs");
       request.setSysAction("DescribeZones");
       request.setSysVersion("2014-05-26");
       request.setSysMethod(MethodType.POST);
        request.setHttpContentType(FormatType.FORM);
       //"<yourRegionID>"替换为真实的RegionId
       request.putHeadParameter("x-acs-regionid", "<yourRegionID>");
       // "<yourStsToken>"替换为上面获取到的STS三元组中stsToken
       request.putQueryParameter("SecurityToken", "<yourStsToken>");
       // 非必填,资源集id
       // request.putHeadParameter("x-acs-resourcegroupid", "<yourResourceGroupID>");
       // 必填,组织id
       request.putHeadParameter("x-acs-organizationid", "<yourOrganizationID>");
       trv {
           CommonResponse response = client.getCommonResponse(request);
           System.out.println(response.getData());
        } catch (ServerException e) {
           System.out.println(e.getErrCode());
           System.out.println(e.getErrMsg());
        } catch (ClientException e) {
           System.out.println(e.getErrCode());
           System.out.println(e.getErrMsg());
```

 容器服务Kubernetes版 开发指南·附录

错误码

HttpCode	错误码	错误信息	描述
400	InvalidParameter	The parameter RoleArn is wrongly formed.	角色ARN格式错误。
400	InvalidParameter.RoleAr n	The parameter RoleArn is wrongly formed.	角色ARN格式错误。
400	InvalidParameter.RoleSe ssionName	The parameter RoleSessionName is wrongly formed.	RoleSessionName格式错误,支持输入2~64个字符,请输入至少2个字符,允许输入英文字母、数字、半角句号(.)、at(@)、短划线(-)和下划线(_)。
400	InvalidParameter.Durati onSeconds	The Min/Max value of DurationSeconds is 15min/1hr.	DurationSeconds参数设置错误,过期时间最小值为900秒,最大值为MaxSessionDuration设置的时间。
400	InvalidParameter.PolicyG rammar	The parameter Policy has not passed grammar check.	权限策略语法错误。
400	InvalidParameter.PolicyS ize	The size of Policy must be smaller than 1024 bytes.	权限策略长度超限,最大 不超过1024字符。
403	NoPermission	You are not authorized to do this action. You should be authorized by RAM.	STS Token没有权限。
404	EntityNotExist.Role	The specified Role not exists.	指定的RAM角色不存在。
500	InternalError	STS Server Internal Error happened.	服务器内部错误。