

机器学习PAI PAI-DLC云原生深度学习训练平台

ALIBABA CLOUD

文档版本: 20210604



法律声明

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

- 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档,且仅能用 于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息,您应当严格 遵守保密义务;未经阿里云事先书面同意,您不得向任何第三方披露本手册内容或 提供给任何第三方使用。
- 未经阿里云事先书面许可,任何单位、公司或个人不得擅自摘抄、翻译、复制本文 档内容的部分或全部,不得以任何方式或途径进行传播和宣传。
- 由于产品版本升级、调整或其他原因,本文档内容有可能变更。阿里云保留在没有 任何通知或者提示下对本文档的内容进行修改的权利,并在阿里云授权通道中不时 发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠 道下载、获取最新版的用户文档。
- 4. 本文档仅作为用户使用阿里云产品及服务的参考性指引,阿里云以产品及服务的"现状"、"有缺陷"和"当前功能"的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引,但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的,阿里云不承担任何法律责任。在任何情况下,阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害,包括用户使用或信赖本文档而遭受的利润损失,承担责任(即使阿里云已被告知该等损失的可能性)。
- 5. 阿里云网站上所有内容,包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计,均由阿里云和/或其关联公司依法拥有其知识产权,包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意,任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外,未经阿里云事先书面同意,任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称(包括但不限于单独为或以组合形式包含"阿里云"、"Aliyun"、"万网"等阿里云和/或其关联公司品牌,上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司)。
- 6. 如若发现本文档存在任何错误,请与阿里云取得直接联系。

通用约定

格式	说明	样例
⚠ 危险	该类警示信息将导致系统重大变更甚至故 障,或者导致人身伤害等结果。	⚠ 危险 重置操作将丢失用户配置数据。
⚠ 警告	该类警示信息可能会导致系统重大变更甚 至故障,或者导致人身伤害等结果。	警告 重启操作将导致业务中断,恢复业务 时间约十分钟。
〔〕) 注意	用于警示信息、补充说明等,是用户必须 了解的内容。	大意 权重设置为0,该服务器不会再接受新 请求。
? 说明	用于补充说明、最佳实践、窍门等 <i>,</i> 不是 用户必须了解的内容。	⑦ 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在 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
[] 或者 [alb]	表示可选项,至多选择一个。	ipconfig [-all -t]
{} 或者 {a b}	表示必选项,至多选择一个。	switch {act ive st and}

目录

1.概述	06
2.授权	07
3.开始使用	09
3.1. 准备工作集群	09
3.2. 公共镜像列表	10
3.3. 公共资源组相关	35
3.3.1. 数据配置	35
3.3.2. 创建任务	38
3.3.3 管理任务	39
3.3.4. 通过Python SDK提交任务	40
3.3.5. 定时自动提交任务	42
3.3.6. 客户端工具	47
3.3.6.1. 命令列表	47
3.3.6.2. 准备工作	48
3.3.6.3. 创建命令	50
3.3.6.4. 删除命令	51
3.3.6.5. 提交命令	52
3.3.6.6. 停止命令	59
3.3.6.7. 查询命令	59
3.4. 专有资源组相关	65
3.4.1. 使用PAI-DLC Dashboard管理任务	65
3.4.2. 使用Arena管理任务(推荐)	68
3.4.3. ACK集群挂载NAS存储	71
3.4.4. ACK集群挂载OSS存储	75
3.4.5. ACK集群挂载CPFS存储	77
3.4.6. 开启公网Ingress服务	79

4.最佳实践	;	82
--------	---	----

机器学习PAI

1.概述

PAI-DLC (Deep Learning Containers) 是基于阿里巴巴容器服务ACK (Alibaba Cloud Container Service for Kubernetes)的深度学习训练平台,为您提供灵活、稳定、易用和极致性能的深度学习训练环境。本文介绍 PAI-DLC的功能亮点、计费规则及可用区。

功能简述

通过PAI-DLC和ACK,算法科学家们可以利用PAI团队深度优化的PAI-TensorFlow或PAI-Pytorch组件,方便快 速地提交深度学习任务、监控任务进度,并便捷地查看结果。同时,用户可以购买GPU服务器,以构建ACK 集群,通过Arena可以管理集群内的可用GPU资源,从而更好地规划训练任务。

功能亮点

- 支持训练按照任务实际使用时长付费
- 支持自定义运行环境
- 支持编程式离线调度
- 支持超大规模分布式训练

可用区

PAI-DLC支持华北2(北京)、华东2(上海)、华东1(杭州)及华南1(深圳)。

计费规则

计费规则请参见PAI-DLC计费说明。

2.授权

首次使用PAI-DLC,需要对服务关联角色进行授权。

前提条件

购买PAI, 详情请参见<mark>开通</mark>。

背景信息

为了确保PAI-DLC能够正常提供服务,您需要确认当前阿里云主账号拥有AliyunPAIDLCDefaultRole这一服务 角色。具体操作如下:

- 1. 登录RAM控制台。
- 2. 在左侧导航栏,单击RAM角色管理。
- 3. 在RAM角色管理页面的搜索中,输入AliyunPAIDLCDefaultRole,进行搜索:
 - 如果搜索到了该角色,则表示已经授权了PAI-DLC服务角色。
 - 如果没有搜索到该角色,则进行授权(授权)。

⑦ 说明 仅主账号可以进行授权, RAM用户无法授权。

操作步骤

- 1. 进入DLC云原生深度学习训练平台页面。
 - i. 登录PAI控制台。
 - ii. 在PA控制台首页,选择模型开发和训练 > DLC云原生深度学习训练平台。
- 2. 单击去授权,进入云资源访问授权页面。

机器学习PAI	机器学习PAI/DLC-云原生深度学习建模
概览	
数据准备 へ	尚未进行角色授权
数据集管理	使用此产品时,将会自动创建一个服务关联角色,以完成相应功能。 Aller PARTY CParty IPPale
数据标注	权限说明:具有拥有客户的阿里云kubernetes、ECS的操作权限
模型开发和训练 へ	去授权
Studio-可视化建模	
DSW-Notebook建模	
算法发布	
DLC-云原生深度学习建模	
模型管理へ	
模型列表	
Blade-模型优化 🔤	
模型部署 へ	
EAS-模型在线服务	

3. 单击同意授权。

NA NEW CALLER, MELINARCHET CE, EXCENSE, MERINE-MERINALISTICS,	
憲士な可能に行った。後に通びはの同	
al Kankan Juli A Lama Kanka	
NiyunPAIDLCDefaultRole	
#述: PAIDLC使用的审巴来访问您住具怎么产品中的资源。	

4. 授权成功后,可以在RAM访问控制页面查看授权的角色,详情请参见查看RAM角色基本信息。

RAM 访问控制	RAM 访问控制 / RAM角色管理			
概范	RAM角色管理			
人员管理へ	● 什么是 RAM 角色?			
用户组	KAM 知色可能制用如晶性的成体(约如:KAAM 用户、风个加田与中国云能的)进行提供 * 您无能户下的一个 RAM 用户(可能是代表一个移动 App 的启频服务); • 其他无能中中的 RAM 用户(管题进行预排中的资源内印);	因的一种安全方法。根据不同业用功衡,安信任约实体可能有如下一些的子:		
设置	 ECS 案例上运行的应用程序代码 (業要対云波運動行撮作); 支部何間五級条 (署要対応販沖中的)資源进行操作才能揭供服券); (回知意思)が開催。 			
SSO 管理	 正立即建位抽碎器 IPF,可以AFT用E 330。 RAM 角色派发组时有效的访问令牌(STS 令牌),使其成为一种更安全的授予访问权限 	的方法。		
収限管理 へ	特别说明:			
授权	RAM 角色不同于传统的教科书式角色(其含义是指一组权限集)。如果您要要使用教师	4书式角色的功能,请参考 RAM 权限策略(Policy)。		
权限策略管理	创建 RAM 角色 DLC Q.			G
RAM 角色管理	< RAM角色名称	辦注	创建时间	操作
CAuth 应用管理	AliyunPAIDLCDefaultRole	PAI DLC使用此角色未访问您在其他云产品中的资源。	2020年6月10日 14:13:56	活动现现 精神经经 删除

3.开始使用

3.1. 准备工作集群

基于不同的资源组,PAI-DLC支持公共资源组和专有资源组工作集群。本文为您介绍如何在PAI-DLC中准备适合自己的工作集群。

前提条件

- 如果您使用公共资源组,则需要为PAI-DLC服务关联角色授权,详情请参见授权。
- 如果您使用专有资源组,则需要准备如下工作:
 - 为ACK集群开启公网Ingress服务,详细请参见开启公网Ingress服务。
 - 为PAI-DLC服务关联角色授权,详情请参见授权。

背景信息

- 公共资源组按照任务实际时长计费,您无需自行维护计算资源集群,只需要明确任务需要使用的资源类型 即可。关于资源类型和定价,请参见PAI-DLC计费说明。
- 专有资源组不额外计费,您只需要支付底层ACK计算资源和相关网络组件的费用。关于如何创建ACK集群,详情请参见创建Kubernet es专有版集群。

准备公共资源组工作集群

- 1. 登录PAI控制台。
- 2. 在左侧导航栏,选择模型开发和训练 > 云原生深度学习训练(DLC)。
- 3. 在PAI-DLC管理控制台页面,如果看到系统自动创建的公共资源组集群(如下图所示),则表明集群准 备完毕。

机器学习PAI/DLC-云原生深度学习训练平台						帮助文档		
DLC–云原生深度学习训练平台								
PAI-DLC是PAI团队打造的	回基于阿里云容器服务的深度学习训	练平台,为您提供灵活、	稳定、易用和极致性	能的深度学习训练环境.				
添加集群 集群名称	、 ~ 请输入	Q						G
PAI-DLC名称	ACK集群名称	状态 ▽	类型	创建时间 小	描述	创建人	操作	
public-cluster	public-cluster	✓ 运行中	公共资源组	2021-01-14 15:00:00 (UTC+8)	公共资源组	系统默认	集群控制台	
					共有0条 , 每页显示:	10 🗸	〈 上一页 1	下一页 >

如果在PAI-DLC管理控制台页面,没有看到系统自动创建的公共资源组集群,请提工单工单系统。

准备专有资源组工作集群

- 1. 登录PAI控制台。
- 2. 在左侧导航栏,选择模型开发和训练 > 云原生深度学习训练(DLC)。
- 3. 在PAI-DLC管理控制台页面,单击添加集群。
- 4. 在添加集群面板, 配置参数。

参数	描述
集群名称	PAI-DLC集群名称。长度为1~30个字符,以小写字母、大写字母、数字或中文开头, 可以包含下划线(_)或短划线(-)。
描述内容	PAI-DLC添加的ACK集群信息,便于区分不同的集群。
ACK集群	 PAI-DLC支持以下方式添加ACK集群: 从ACK集群列表,选择已有的可绑定的ACK集群。 单击ACK集群后的新建集群,进入ACK管理控制台,新建ACK集群,详情请参见创建Kubernetes专有版集群。
组件版本	PAI-DLC组件的版本。

5. 单击确定。

新添加的集群会显示在PAI-DLC管理控制台页面的集群列表下。如果该集群的**状态从部署中**变为运行中,则可以开始深度学习训练。

3.2. 公共镜像列表

在PAI-DLC中创建深度学习任务时,您需要配置工作节点的镜像,系统支持公共镜像和自定义镜像。本文介 绍系统支持的公共镜像(社区镜像和官方镜像)及镜像版本详情,您可以根据实际情况选择合适的镜像。

背景信息

本文介绍如下镜像的详细版本信息,包括镜像操作系统、CUDA版本、Python版本及三方库列表。

类型	框架	镜像
		tensorflow-training:2.3-cpu-py36-ubuntu18.04
		tensorflow-training:2.3-gpu-py36-cu101-ubuntu18.04
计区结换	renson tow	tensorflow-training:1.15-cpu-py36-ubuntu18.04
社区況家		tensorflow-training:1.15-gpu-py36-cu100-ubuntu18.04
	DyTorch	pytorch-training:1.6.0-gpu-py37-cu101-ubuntu18.04
	Pyroleit	pytorch-training:1.7.1-gpu-py37-cu110-ubuntu18.04
		tensorflow-training:1.12.2PAI-cpu-py27-ubuntu16.04
		tensorflow-training:1.12.2PAI-mkl-cpu-py27-ubuntu16.04
		tensorflow-training:1.12.2PAI-gpu-py27-cu100-ubuntu16.04
		tensorflow-training:1.12.2PAI-cpu-py36-ubuntu16.04
	TensorFlow	tensorflow-training:1.12.2PAI-mkl-cpu-py36-ubuntu16.04
		tensorflow-training:1.12.2PAI-gpu-py36-cu100-ubuntu16.04

裏	框架	镜像
		tensorflow-training:1.15.0PAI-gpu-py27-cu100-ubuntu16.04
		tensorflow-training:1.15.0PAI-gpu-py36-cu100-ubuntu16.04
		pytorch-training:1.3.1PAI-gpu-py37-cu100-ubuntu16.04
	DuTorch	pytorch-training:1.4.0PAI-gpu-py37-cu100-ubuntu16.04
	Pyroich	pytorch-training:1.5.1PAI-gpu-py37-cu100-ubuntu16.04
		pytorch-training:1.6.0PAI-gpu-py37-cu100-ubuntu16.04

社区镜像(由社区提供的标准镜像)

社区镜像支持不同的资源类型,具体的镜像文件列表如下。每种镜像的操作系统、CUDA版本及三方库信息 请参见社区镜像版本详情。

registry.\${region}.aliyuncs.com/pai-dlc/pytorch-training:1.6.0-gpu-py37-cu101-ubuntu18.04 registry.\${region}.aliyuncs.com/pai-dlc/pytorch-training:1.7.1-gpu-py37-cu110-ubuntu18.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:2.3.0-cpu-py36-ubuntu18.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:2.3.0-gpu-py36-cu101-ubuntu18.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.15.4-cpu-py36-ubuntu18.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.15.4-cpu-py36-ubuntu18.04

其中 \${region} 需要替换为具体的地域,包括如下取值:

- cn-hangzhou
- cn-shanghai
- cn-qingdao
- cn-beijing
- cn-zhangjiakou
- cn-huhehaote
- cn-shenzhen
- cn-chengdu
- cn-hongkong
- ap-southeast-1

例如 \${region} 取值为 cn-hangzhou 时,社区提供的镜像如下表所示。

\${region}	框架	CPU/GP U	Python版本	镜像的URL
------------	----	-------------	----------	--------

PAI-DLC云原生深度学习训练平台·开 始使用

\${region}	框架	CPU/GP U	Python版本	镜像的URL
Cn- hangzhou	Tensorflow 2.3	CPU	3.6 (ру36)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:2.3.0-cpu-py36- ubuntu18.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:2.3.0-cpu-py36- ubuntu18.04
	Tensorflow 2.3	GPU	3.6 (py36)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:2.3.0-gpu-py36-cu101- ubuntu18.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:2.3.0-gpu-py36-cu101- ubuntu18.04
	Tensorflow 1.15	CPU	3.6 (py36)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.4-cpu-py36- ubuntu18.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.4-cpu-py36- ubuntu18.04
	Tensorflow 1.15	GPU	3.6 (py36)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.4-gpu-py36- cu100-ubuntu18.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.4-gpu-py36- cu100-ubuntu18.04
	PyTorch 1.6	GPU	3.7 (py37)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.6.0-gpu-py37-cu101- ubuntu18.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.6.0-gpu-py37-cu101- ubuntu18.04
	PyTorch 1.7	GPU	3.7 (py37)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.7.1-gpu-py37-cu110- ubuntu18.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.7.1-gpu-py37-cu110- ubuntu18.04

官方镜像(由PAI团队提供优化的镜像)

PAI-DLC提供了多种官方镜像,支持不同的资源类型、Python版本及深度学习框架TensorFlow和PyTorch。 官方公共镜像文件列表如下,关于每种镜像的操作系统、CUDA版本及三方库信息请参见官方镜像版本详 情。

registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.12.2PAI-cpu-py27-ubuntu16.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.12.2PAI-mkl-cpu-py27-ubuntu16.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.12.2PAI-gpu-py27-cu100-ubuntu16.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.12.2PAI-cpu-py36-ubuntu16.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.12.2PAI-mkl-cpu-py36-ubuntu16.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.12.2PAI-gpu-py36-cu100-ubuntu16.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.15.0PAI-gpu-py27-cu100-ubuntu16.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.15.0PAI-gpu-py36-cu100-ubuntu16.04 registry.\${region}.aliyuncs.com/pai-dlc/pytorch-training:1.3.1PAI-gpu-py37-cu100-ubuntu16.04 registry.\${region}.aliyuncs.com/pai-dlc/pytorch-training:1.4.0PAI-gpu-py37-cu100-ubuntu16.04 registry.\${region}.aliyuncs.com/pai-dlc/pytorch-training:1.5.1PAI-gpu-py37-cu100-ubuntu16.04 registry.\${region}.aliyuncs.com/pai-dlc/pytorch-training:1.6.0PAI-gpu-py37-cu100-ubuntu16.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.12.2PAI-cpu-py27-ubuntu18.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.12.2PAI-gpu-py27-cu101-ubuntu18.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.12.2PAI-cpu-py36-ubuntu18.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.12.2PAI-gpu-py36-cu101-ubuntu18.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.15.4PAI-cpu-py36-ubuntu18.04 registry.\${region}.aliyuncs.com/pai-dlc/tensorflow-training:1.15.4PAI-gpu-py36-cu101-ubuntu18.04

其中 \${region} 需要替换为具体的地域,包括如下取值:

- cn-hangzhou
- cn-shanghai
- cn-qingdao
- cn-beijing
- cn-zhangjiakou
- cn-huhehaote
- cn-shenzhen
- cn-chengdu
- cn-hongkong
- ap-southeast-1

例如 \${region} 取值为 cn-hangzhou 时, PAI-DLC所有的官方镜像如下表所示。

\${region}	框架	CPU/GP U	Python版本	镜像的URL
------------	----	-------------	----------	--------

PAI-DLC云原生深度学习训练平台·开 始使用

\${region}	框架	CPU/GP U	Python版本	镜像的URL
		CPU	2.7 (py27)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-cpu-py27- ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-cpu-py27- ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI2011-cpu- py27-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI2011-cpu- py27-ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI2011-cpu- py27-ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-cpu-py27- ubuntu18.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-cpu-py27- ubuntu18.04
		MKL- CPU	2.7 (py27)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-mkl-cpu- py27-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-mkl-cpu- py27-ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI2011-mkl- cpu-py27-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI2011-mkl- cpu-py27-ubuntu16.04

PAI-DLC云原生深度学习训练平台·<mark>开</mark> 始使用

\${region}	框架	CPU/GP U	Python版本	镜像的URL
	TensorFlow 1.12	GPU	2.7 (py27)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-gpu-py27- cu100-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-gpu-py27- cu100-ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI2011-gpu- py27-cu100-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-gpu-py27- cu100-ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-gpu-py27- cu100-ubuntu18.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-gpu-py27- cu101-ubuntu18.04
		CPU	3.6 (py36)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-cpu-py36- ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-cpu-py36- ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI2011-cpu- py36-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI2011-cpu- py36-ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI2011-cpu- py36-ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-cpu-py36- ubuntu18.04

PAI-DLC云原生深度学习训练平台·开 始使用

\${region}	框架	CPU/GP U	Python版本	镜像的URL
		MKL- CPU	3.6 (py36)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-mkl-cpu- py36-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-mkl-cpu- py36-ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI2011-mkl- cpu-py36-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI2011-mkl- cpu-py36-ubuntu16.04
cn- hangzhou		GPU	3.6 (py36)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-gpu-py36- cu100-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-gpu-py36- cu100-ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI2011-gpu- py36-cu100-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI2011-gpu- py36-cu100-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI2011-gpu- py36-cu100-ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-gpu-py36- cu101-ubuntu18.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.12.2PAI-gpu-py36- cu101-ubuntu18.04
		GPU	2.7 (py27)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.0PAI-gpu-py27- cu100-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.0PAI-gpu-py27- cu100-ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.0PAI2011-gpu- py27-cu100-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.0PAI2011-gpu- py27-cu100-ubuntu16.04

PAI-DLC云原生深度学习训练平台·<mark>开</mark> 始使用

\${region}	框架	CPU/GP U	Python版本	镜像的URL
	TensorFlow 1.15	 registry.cn-hangzhou dlc/tensorflow-train ubuntu18.04 registry-vpc.cn-hang dlc/tensorflow-train ubuntu18.04 		 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.4PAI-cpu-py36- ubuntu18.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.4PAI-cpu-py36- ubuntu18.04
		GPU	3.6 (py36)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.0PAI-gpu-py36- cu100-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.0PAI-gpu-py36- cu100-ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.0PAI2011-gpu- py36-cu100-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.0PAI2011-gpu- py36-cu100-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.0PAI2011-gpu- py36-cu100-ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.4PAI-gpu-py36- cu101-ubuntu18.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/tensorflow-training:1.15.4PAI-gpu-py36- cu101-ubuntu18.04
	PyTorch 1.3	GPU	3.7 (py37)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.3.1PAI-gpu-py37-cu100- ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.3.1PAI-gpu-py37-cu100- ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.3.1PAI2011-gpu-py37- cu100-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.3.1PAI2011-gpu-py37- cu100-ubuntu16.04

PAI-DLC云原生深度学习训练平台·开 始使用

\${region}	框架	CPU/GP U	Python版本	镜像的URL
	PyTorch 1.4	GPU	3.7 (py37)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.4.0PAI-gpu-py37-cu100- ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.4.0PAI-gpu-py37-cu100- ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.4.0PAI2011-gpu-py37- cu100-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.4.0PAI2011-gpu-py37- cu100-ubuntu16.04
	PyT orch 1.5	GPU	3.7 (py37)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.5.1PAI-gpu-py37-cu100- ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.5.1PAI-gpu-py37-cu100- ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.5.1PAI2011-gpu-py37- cu100-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.5.1PAI2011-gpu-py37- cu100-ubuntu16.04
	PyTorch 1.6	GPU	3.7 (py37)	 registry.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.6.0PAI-gpu-py37-cu100- ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.6.0PAI-gpu-py37-cu100- ubuntu16.04 registry.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.6.0PAI2011-gpu-py37- cu100-ubuntu16.04 registry-vpc.cn-hangzhou.aliyuncs.com/pai- dlc/pytorch-training:1.6.0PAI2011-gpu-py37- cu100-ubuntu16.04

社区镜像版本详情

每个社区镜像(由社区提供的标准镜像)支持的操作系统、Python版本及三方库列表如下:

- tensorflow-training:2.3-cpu-py36-ubuntu18.04
 - 操作系统: Ubuntu 18.04.5 LTS
 - Python版本: 3.6.9

三方库及版本			
absl-py 0.11.0	asn1crypto 0.24.0	astunparse 1.6.3	cachetools 4.2.0
certifi 2020.12.5	chardet 4.0.0	cryptography 2.1.4	gast 0.3.3
google-auth 1.24.0	google-auth-oauthlib 0.4.2	google-pasta 0.2.0	grpcio 1.34.0
h5py 2.10.0	idna 2.6	importlib-metadata 3.3.0	Keras-Preprocessing 1.1.2
keyring 10.6.0	keyrings.alt 3.0	Markdown 3.3.3	numpy 1.18.5
oauthlib 3.1.0	opt-einsum 3.3.0	pip 20.2.4	protobuf 3.14.0
pyasn1 0.4.8	pyasn1-modules 0.2.8	pycrypto 2.6.1	pygobject 3.26.1
pyxdg 0.25	requests 2.25.1	requests-oauthlib 1.3.0	rsa 4.6
SecretStorage 2.3.1	setuptools 51.1.1	six 1.15.0	tensorboard 2.4.0
tensorboard-plugin- wit 1.7.0	tensorflow 2.3.2	tensorflow-estimator 2.3.0	termcolor 1.1.0
typing-extensions 3.7.4.3	urllib3 1.26.2	werkzeug 1.0.1	wheel 0.30.0
wrapt 1.12.1	zipp 3.4.0		

- tensorflow-training:2.3-gpu-py36-cu101-ubuntu18.04
 - 。 操作系统: Ubuntu 18.04.5 LTS
 - 。 Python版本: 3.6.9
 - 。 CUDA版本: 10.1

三方库及版本			
absl-py 0.11.0	asn1crypto 0.24.0	astunparse 1.6.3	cachetools 4.2.0
certifi 2020.12.5	chardet 4.0.0	cryptography 2.1.4	gast 0.3.3
google-auth 1.24.0	google-auth-oauthlib 0.4.2	google-pasta 0.2.0	grpcio 1.34.0
h5py 2.10.0	idna 2.6	importlib-metadata 3.3.0	Keras-Preprocessing 1.1.2
keyring 10.6.0	keyrings.alt 3.0	Markdown 3.3.3	numpy 1.18.5
oauthlib 3.1.0	opt-einsum 3.3.0	pip 20.2.4	protobuf 3.14.0
pyasn1 0.4.8	pyasn1-modules 0.2.8	pycrypto 2.6.1	pygobject 3.26.1
pyxdg 0.25	requests 2.25.1	requests-oauthlib 1.3.0	rsa 4.6
SecretStorage 2.3.1	setuptools 51.1.1	six 1.15.0	tensorboard 2.4.0
tensorboard-plugin- wit 1.7.0	tensorflow-gpu 2.3.2	tensorflow-estimator 2.3.0	termcolor 1.1.0
typing-extensions 3.7.4.3	urllib3 1.26.2	werkzeug 1.0.1	wheel 0.30.0
wrapt 1.12.1	zipp 3.4.0	python-apt 1.6.5+ubuntu	10.5

- tensorflow-training:1.15-cpu-py36-ubuntu18.04
 - 。 操作系统: Ubuntu 18.04.5 LTS
 - Python版本: 3.6.9

三方库及版本			
absl-py 0.11.0	asn1crypto 0.24.0	astor 0.8.1	cryptography 2.1.4
gast 0.2.2	google-pasta 0.2.0	grpcio 1.34.0	h5py 2.10.0
idna 2.6	import lib-met adat a 3.3.0	Keras-Preprocessing 1.1.2	Keras-Applications 1.0.8
keyring 10.6.0	keyrings.alt 3.0	Markdown 3.3.3	numpy 1.18.5
opt-einsum 3.3.0	pip 20.3.3	protobuf 3.14.0	pycrypto 2.6.1
pygobject 3.26.1	pyxdg 0.25	SecretStorage 2.3.1	setuptools 51.1.1
six 1.11.0	tensorboard 1.15.0	tensorflow 1.15.5	tensorflow-estimator 1.15.1
termcolor 1.1.0	typing-extensions 3.7.4.3	werkzeug 1.0.1	wheel 0.30.0
wrapt 1.12.1	zipp 3.4.0		

- tensorflow-training:1.15-gpu-py36-cu100-ubuntu18.04
 - 操作系统: Ubuntu 18.04.5 LTS
 - Python版本: 3.6.9
 - 。 CUDA版本: 10.0
 - 三方库: 三方库列表及版本信息如下表所示。

三方库及版本			
absl-py 0.11.0	asn1crypto 0.24.0	astor 0.8.1	cryptography 2.1.4
gast 0.2.2	google-pasta 0.2.0	grpcio 1.34.0	h5py 2.10.0
idna 2.6	importlib-metadata 3.3.0	Keras-Preprocessing 1.1.2	Keras-Applications 1.0.8
keyring 10.6.0	keyrings.alt 3.0	Markdown 3.3.3	numpy 1.18.5
opt-einsum 3.3.0	pip 20.3.3	protobuf 3.14.0	pycrypto 2.6.1
pygobject 3.26.1	pyxdg 0.25	SecretStorage 2.3.1	setuptools 51.1.1
six 1.11.0	tensorboard 1.15.0	tensorflow-gpu 1.15.5	tensorflow-estimator 1.15.1
termcolor 1.1.0	typing-extensions 3.7.4.3	werkzeug 1.0.1	wheel 0.30.0
wrapt 1.12.1	zipp 3.4.0	python-apt 1.6.5+ubuntu	٥.5

- pytorch-training:1.6.0-gpu-py37-cu101-ubuntu18.04
 - 操作系统: Ubuntu 18.04.4 LTS
 - Python版本: 3.7.7
 - CUDA版本: 10.1
 - 三方库: 三方库列表及版本信息如下表所示。

三方库及版本			
backcall 0.2.0	beautifulsoup4 4.9.1	certifi 2020.6.20	cffi 1.14.0
chardet 3.0.4	conda 4.8.3	conda-build 3.18.11	conda-package- handling 1.7.0
cryptography 2.9.2	decorator 4.4.2	filelock 3.0.12	glob2 0.7
idna 2.9	ipython 7.16.1	ipython-genutils 0.2.0	jedi 0.17.1
Jinja2 2.11.2	libarchive-c 2.9	MarkupSafe 1.1.1	mkl-fft 1.1.0
mkl-random 1.1.1	mkl-service 2.3.0	numpy 1.18.5	olefile 0.46
parso 0.7.0	pexpect 4.8.0	pickleshare 0.7.5	Pillow 7.2.0
pip 20.0.2	pkginfo 1.5.0.1	prompt-toolkit 3.0.5	psutil 5.7.0
ptyprocess 0.6.0	pycosat 0.6.3	pycparser 2.20	Pygments 2.6.1
pyOpenSSL 19.1.0	PySocks 1.7.1	pytz 2020.1	PyYAML 5.3.1
requests 2.23.0	ruamel-yaml 0.15.87	setuptools 46.4.0.post20200518	six 1.14.0
soupsieve 2.0.1	torch 1.6.0	torchvision 0.7.0	tqdm 4.46.0
traitlets 4.3.3	urllib3 1.25.8	wcwidth 0.2.5	wheel 0.34.2

- pytorch-training:1.7.1-gpu-py37-cu110-ubuntu18.04
 - 操作系统: Ubuntu 18.04.5 LTS
 - Python版本: 3.8.5
 - 。 CUDA版本: 11.0

三方库及版本			
backcall 0.2.0	beautifulsoup4 4.9.3	brotlipy 0.7.0	certifi 2020.12.5
cffi 1.14.3	chardet 3.0.4	conda 4.9.2	conda-build 3.21.4
conda-package- handling 1.7.2	cryptography 3.2.1	decorator 4.4.2	dnspython 2.1.0
filelock 3.0.12	glob2 0.7	idna 2.10	ipython 7.19.0
ipython-genutils 0.2.0	jedi 0.17.2	Jinja2 2.11.2	libarchive-c 2.9
MarkupSafe 1.1.1	mkl-fft 1.2.0	mkl-random 1.1.1	mkl-service 2.3.0
numpy 1.19.2	olefile 0.46	parso 0.7.0	pexpect 4.8.0
pickleshare 0.7.5	Pillow 8.1.0	pip 20.2.4	pkginfo 1.7.0
prompt-toolkit 3.0.8	psutil 5.7.2	ptyprocess 0.7.0	pycosat 0.6.3
pycparser 2.20	Pygments 2.7.4	pyOpenSSL 19.1.0	PySocks 1.7.1
python-etcd 0.4.5	pytz 2020.5	PyYAML 5.3.1	requests 2.24.0
ruamel-yaml 0.15.87	setuptools 50.3.1.post20201107	six 1.15.0	soupsieve 2.1
torch 1.7.1	torchelastic 0.2.1	torchvision 0.8.2	tqdm 4.51.0
traitlets 5.0.5	typing-extensions 3.7.4.3	urllib3 1.25.11	wcwidth 0.2.5
wheel 0.35.1			

官方镜像版本详情

每个官方镜像(由PAI团队提供优化的镜像)支持的操作系统、Python版本及三方库列表如下:

- tensorflow-training:1.12.2PAI-cpu-py27-ubuntu16.04
 - 操作系统: Ubuntu 16.04.6 LTS
 - 。 Python版本: 2.7.18 Anaconda

三方库及版本			
absl-py 0.11.0	aliyun-python-sdk- core 2.13.15	aliyun-python-sdk-kms 2.14.0	astor 0.8.1
backports.weakref 1.0.post1	certifi 2020.6.20	crcmod 1.7	Cython 0.29.14
enum34 1.1.6	funcsigs 1.0.2	futures 3.3.0	gast 0.4.0
grpcio 1.27.2	h5py 2.10.0	jmespath 0.10.0	Keras-Applications 1.0.8
Keras-Preprocessing 1.1.2	Markdown 3.1.1	mkl-fft 1.0.15	mkl-random 1.1.0
mkl-service 2.3.0	mock 3.0.5	numpy 1.16.4	opencv-python 4.2.0.32
oss2 2.9.1	paiio 0.1.0	pip 9.0.1	protobuf 3.14.0
pycryptodome 3.9.7	pyodps 0.10.4	pypai 1.1.0+tensorflow.1.12. 2pai2011	requests 2.13.0
setuptools 36.4.0	six 1.15.0	tensorboard 1.12.2	tensorflow 1.12.2PAI2011
termcolor 1.1.0	toposort 1.5	Werkzeug 1.0.1	wheel 0.35.1

- tensorflow-training:1.12.2PAI-mkl-cpu-py27-ubuntu16.04
 - 操作系统: Ubuntu 16.04.6 LTS
 - 。 Python版本:2.7.18 Anaconda

三方库及版本			
absl-py 0.11.0	aliyun-python-sdk- core 2.13.15	aliyun-python-sdk-kms 2.14.0	astor 0.8.1
backports.weakref 1.0.post1	certifi 2020.6.20	crcmod 1.7	Cython 0.29.14
enum34 1.1.6	funcsigs 1.0.2	futures 3.3.0	gast 0.4.0
grpcio 1.27.2	h5py 2.10.0	jmespath 0.10.0	Keras-Applications 1.0.8
Keras-Preprocessing 1.1.2	Markdown 3.1.1	mkl-fft 1.0.15	mkl-random 1.1.0
mkl-service 2.3.0	mock 3.0.5	numpy 1.16.4	opencv-python 4.2.0.32
oss2 2.9.1	paiio 0.1.0	pip 9.0.1	protobuf 3.14.0
pycryptodome 3.9.7	pyodps 0.10.4	pypai 1.1.0+tensorflow.1.12. 2pai2011	requests 2.13.0
setuptools 36.4.0	six 1.15.0	tensorboard 1.12.2	tensorflow 1.12.2PAI2011
termcolor 1.1.0	toposort 1.5	Werkzeug 1.0.1	wheel 0.35.1

- tensorflow-training:1.12.2PAI-gpu-py27-cu100-ubuntu16.04
 - 操作系统: Ubuntu 16.04.6 LTS
 - 。 Python版本:2.7.18 Anaconda
 - 。 CUDA版本: 10.0

三方库及版本			
absl-py 0.11.0	aliyun-python-sdk- core 2.13.15	aliyun-python-sdk-kms 2.14.0	astor 0.8.1
backports.weakref 1.0.post1	certifi 2020.6.20	crcmod 1.7	Cython 0.29.14
enum34 1.1.6	funcsigs 1.0.2	futures 3.3.0	gast 0.4.0
grpcio 1.27.2	h5py 2.10.0	jmespath 0.10.0	Keras-Applications 1.0.8
Keras-Preprocessing 1.1.2	Markdown 3.1.1	mkl-fft 1.0.15	mkl-random 1.1.0
mkl-service 2.3.0	mock 3.0.5	numpy 1.16.4	opencv-python 4.2.0.32
oss2 2.9.1	paiio 0.1.0	pip 9.0.1	protobuf 3.14.0
pycryptodome 3.9.7	pyodps 0.10.4	pypai 1.1.0+tensorflow.gpu. 1.12.2pai2011	requests 2.13.0
setuptools 36.4.0	six 1.15.0	tensorboard 1.12.2	tensorflow-gpu 1.12.2PAI2011
termcolor 1.1.0	toposort 1.5	Werkzeug 1.0.1	wheel 0.35.1
subprocess32 3.5.4	tao-wrapper 0.1.1	whale 0.0.2	

- tensorflow-training:1.12.2PAI-cpu-py36-ubuntu16.04
 - 操作系统: Ubuntu 16.04.6 LTS
 - Python版本: 3.6.12 Anaconda

三方库及版本				
absl-py 0.11.0	aliyun-python-sdk- core 2.13.29	aliyun-python-sdk- core-v3 2.13.11	aliyun-python-sdk-kms 2.14.0	
astor 0.8.1	cached-property 1.5.2	certifi 2020.12.5	crcmod 1.7	
Cython 0.29.21	gast 0.4.0	grpcio 1.31.0	h5py 3.1.0	
importlib-metadata 3.4.0	jmespath 0.10.0	Keras-Applications 1.0.8	Keras-Preprocessing 1.1.2	
Markdown 3.3.3	mkl-fft 1.2.0	mkl-random 1.1.1	mkl-service 2.3.0	
numpy 1.16.4	opencv-python 4.2.0.32	oss2 2.12.1	paiio 0.1.0	
pip 20.2.4	protobuf 3.14.0	pycryptodome 3.9.9	pyodps 0.10.4	
pypai 1.1.0+tensorflow.1.12. 2pai2011	requests 2.13.0	setuptools 50.3.1.post20201107	six 1.15.0	
tensorboard 1.12.2	tensorflow 1.12.2PAI2011	termcolor 1.1.0	toposort 1.5	
typing-extensions 3.7.4.3	Werkzeug 1.0.1	wheel 0.35.1	zipp 3.4.0	

- tensorflow-training:1.12.2PAI-mkl-cpu-py36-ubuntu16.04
 - 操作系统: Ubuntu 16.04.6 LTS
 - 。 Python版本: 3.6.12 Anaconda

三方库及版本			
absl-py 0.11.0	aliyun-python-sdk- core 2.13.29	aliyun-python-sdk- core-v3 2.13.11	aliyun-python-sdk-kms 2.14.0
astor 0.8.1	cached-property 1.5.2	certifi 2020.12.5	crcmod 1.7
Cython 0.29.21	gast 0.4.0	grpcio 1.31.0	h5py 3.1.0
importlib-metadata 3.4.0	jmespath 0.10.0	Keras-Applications 1.0.8	Keras-Preprocessing 1.1.2
Markdown 3.3.3	mkl-fft 1.2.0	mkl-random 1.1.1	mkl-service 2.3.0
numpy 1.16.4	opencv-python 4.2.0.32	oss2 2.12.1	paiio 0.1.0
pip 20.2.4	protobuf 3.14.0	pycryptodome 3.9.9	pyodps 0.10.4
pypai 1.1.0+tensorflow.1.12. 2pai2011	requests 2.13.0	setuptools 50.3.1.post20201107	six 1.15.0
tensorboard 1.12.2	tensorflow 1.12.2PAI2011	termcolor 1.1.0	toposort 1.5
typing-extensions 3.7.4.3	Werkzeug 1.0.1	wheel 0.35.1	zipp 3.4.0

- tensorflow-training:1.12.2PAI-gpu-py36-cu100-ubuntu16.04
 - 操作系统: Ubuntu 16.04.6 LTS
 - Python版本: 3.6.12 Anaconda
 - 。 CUDA版本: 10.0

三方库及版本			
absl-py 0.11.0	aliyun-python-sdk- core 2.13.29	aliyun-python-sdk- core-v3 2.13.11	aliyun-python-sdk-kms 2.14.0
astor 0.8.1	cached-property 1.5.2	certifi 2020.12.5	crcmod 1.7
Cython 0.29.21	gast 0.4.0	grpcio 1.31.0	h5py 3.1.0
importlib-metadata 3.4.0	jmespath 0.10.0	Keras-Applications 1.0.8	Keras-Preprocessing 1.1.2
Markdown 3.3.3	mkl-fft 1.2.0	mkl-random 1.1.1	mkl-service 2.3.0
numpy 1.16.4	opencv-python 4.2.0.32	oss2 2.12.1	paiio 0.1.0
pip 20.2.4	protobuf 3.14.0	pycryptodome 3.9.9	pyodps 0.10.4
pypai 1.1.0+tensorflow.gpu. 1.12.2pai2011	requests 2.13.0	setuptools 50.3.1.post20201107	six 1.15.0
tensorboard 1.12.2	tensorflow-gpu 1.12.2PAI2011	termcolor 1.1.0	toposort 1.5
typing-extensions 3.7.4.3	Werkzeug 1.0.1	wheel 0.35.1	zipp 3.4.0
subprocess32 3.5.4	tao-wrapper 0.1.1	whale 0.0.2	

• tensorflow-training:1.15.0PAI-gpu-py27-cu100-ubuntu16.04

- 操作系统: Ubuntu 16.04.6 LTS
- Python版本: 2.7.18 Anaconda
- 。 CUDA版本: 10.0

三方库及版本			
absl-py 0.11.0	aliyun-python-sdk- core 2.13.15	aliyun-python-sdk-kms 2.14.0	astor 0.8.1
backports.weakref 1.0.post1	certifi 2020.6.20	crcmod 1.7	Cython 0.29.14
enum34 1.1.6	funcsigs 1.0.2	functools32 3.2.3.post2	futures 3.3.0
gast 0.2.2	google-pasta 0.2.0	opt-einsum 2.3.2	tensorflow-estimator 1.15.1
grpcio 1.27.2	h5py 2.10.0	jmespath 0.10.0	Keras-Applications 1.0.8
Keras-Preprocessing 1.1.2	Markdown 3.1.1	mkl-fft 1.0.15	mkl-random 1.1.0
mkl-service 2.3.0	mock 3.0.5	numpy 1.16.4	opencv-python 4.2.0.32
oss2 2.9.1	paiio 0.1.0	pip 9.0.1	protobuf 3.14.0
pycryptodome 3.9.7	pyodps 0.10.4	pypai 1.1.0+tensorflow.gpu. 1.15.0	requests 2.13.0
setuptools 44.1.1	six 1.15.0	tensorboard 1.15.0	tensorflow-gpu 1.15.0
termcolor 1.1.0	toposort 1.5	Werkzeug 1.0.1	wheel 0.35.1
subprocess32 3.5.4	tao-wrapper 0.1.1	whale 0.0.2	wrapt 1.12.1

- tensorflow-training:1.15.0PAI-gpu-py36-cu100-ubuntu16.04
 - 操作系统: Ubuntu 16.04.6 LTS
 - Python版本: 3.6.12 Anaconda
 - 。 CUDA版本: 10.0

三方库及版本			
absl-py 0.11.0	aliyun-python-sdk- core 2.13.29	aliyun-python-sdk- core-v3 2.13.11	aliyun-python-sdk-kms 2.14.0
astor 0.8.1	cached-property 1.5.2	certifi 2020.12.5	crcmod 1.7
Cython 0.29.21	gast 0.2.2	grpcio 1.31.0	h5py 3.1.0
importlib-metadata 3.4.0	jmespath 0.10.0	Keras-Applications 1.0.8	Keras-Preprocessing 1.1.2
Markdown 3.3.3	mkl-fft 1.2.0	mkl-random 1.1.1	mkl-service 2.3.0
numpy 1.16.4	opencv-python 4.2.0.32	oss2 2.12.1	paiio 0.1.0
pip 20.2.4	protobuf 3.14.0	pycryptodome 3.9.9	pyodps 0.10.4
pypai 1.1.0+tensorflow.gpu. 1.15.0	requests 2.13.0	setuptools 50.3.1.post20201107	six 1.15.0
tensorboard 1.15.0	tensorflow-gpu 1.15.0	termcolor 1.1.0	toposort 1.5
typing-extensions 3.7.4.3	Werkzeug 1.0.1	wheel 0.35.1	zipp 3.4.0
subprocess32 3.5.4	tao-wrapper 0.1.1	whale 0.0.2	google-pasta 0.2.0
opt-einsum 3.3.0	tensorflow-estimator 1.15.1	wrapt 1.12.1	

- pytorch-training:1.3.1PAI-gpu-py37-cu100-ubuntu16.04
 - 操作系统: Ubuntu 16.04.6 LTS
 - Python版本: 3.7.4
 - 。 CUDA版本: 10.0

三方库及版本			
absl-py 0.11.0	aiohttp 3.7.3	apex 0.1	asn1crypto 1.2.0
async-timeout 3.0.1	attrs 20.3.0	blinker 1.4	cachetools 4.2.0
certifi 2020.12.5	cffi 1.13.0	chardet 3.0.4	click 7.1.2
conda 4.9.2	conda-package- handling 1.6.0	cryptography 2.8	future 0.18.2
google-auth 1.24.0	google-auth-oauthlib 0.4.2	grpcio 1.31.0	idna 2.8
importlib-metadata 2.0.0	Markdown 3.3.3	mkl-fft 1.2.0	mkl-random 1.1.1
mkl-service 2.3.0	multidict 4.7.6	numpy 1.19.2	nvidia-dali 0.15.0
oauthlib 3.1.0	Pillow 8.1.0	pip 20.2.4	protobuf 3.13.0
pyasn1 0.4.8	pyasn1-modules 0.2.8	pycosat 0.6.3	pycparser 2.19
PyJWT 2.0.0	pyOpenSSL 19.0.0	PySocks 1.7.1	requests 2.22.0
requests-oauthlib 1.3.0	rsa 4.7	ruamel-yaml 0.15.46	sailfish 1.0.1
setuptools 50.3.1.post20201107	six 1.15.0	tensorboard 2.3.0	tensorboard-plugin- wit 1.6.0
torch 1.3.1+ali	torchsummary 1.5.1	torchvision 0.4.2	tqdm 4.36.1
typing-extensions 3.7.4.3	urllib3 1.24.2	Werkzeug 1.0.1	wheel 0.35.1
yarl 1.5.1	zipp 3.4.0		

- pytorch-training:1.4.0PAI-gpu-py37-cu100-ubuntu16.04
 - 操作系统: Ubuntu 16.04.6 LTS
 - Python版本: 3.7.4
 - 。 CUDA版本: 10.0

三方库及版本			
absl-py 0.11.0	aiohttp 3.7.3	apex 0.1	asn1crypto 1.2.0
async-timeout 3.0.1	attrs 20.3.0	blinker 1.4	cachetools 4.2.0
certifi 2020.12.5	cffi 1.13.0	chardet 3.0.4	click 7.1.2
conda 4.9.2	conda-package- handling 1.6.0	cryptography 2.8	future 0.18.2
google-auth 1.24.0	google-aut h-oaut hlib 0.4.2	grpcio 1.31.0	idna 2.8
importlib-metadata 2.0.0	Markdown 3.3.3	mkl-fft 1.2.0	mkl-random 1.1.1
mkl-service 2.3.0	multidict 4.7.6	numpy 1.19.2	nvidia-dali 0.15.0
oauthlib 3.1.0	Pillow 8.1.0	pip 20.2.4	protobuf 3.13.0
pyasn1 0.4.8	pyasn1-modules 0.2.8	pycosat 0.6.3	pycparser 2.19
PyJWT 2.0.0	pyOpenSSL 19.0.0	PySocks 1.7.1	requests 2.22.0
requests-oauthlib 1.3.0	rsa 4.7	ruamel-yaml 0.15.46	setuptools 50.3.1.post20201107
six 1.15.0	tensorboard 2.3.0	tensorboard-plugin- wit 1.6.0	torch 1.4.0+ali
torchsummary 1.5.1	torchvision 0.5.0	tqdm 4.36.1	typing-extensions 3.7.4.3
urllib3 1.24.2	Werkzeug 1.0.1	wheel 0.35.1	yarl 1.5.1
zipp 3.4.0			

- pytorch-training:1.5.1PAI-gpu-py37-cu100-ubuntu16.04
 - 操作系统: Ubuntu 16.04.6 LTS
 - Python版本: 3.7.4
 - 。 CUDA版本: 10.0

三方库及版本			
absl-py 0.11.0	aiohttp 3.7.3	apex 0.1	asn1crypto 1.2.0
async-timeout 3.0.1	attrs 20.3.0	blinker 1.4	cachetools 4.2.0
certifi 2020.12.5	cffi 1.13.0	chardet 3.0.4	click 7.1.2
conda 4.9.2	conda-package- handling 1.6.0	cryptography 2.8	future 0.18.2
google-auth 1.24.0	google-aut h-oaut hlib 0.4.2	grpcio 1.31.0	idna 2.8
importlib-metadata 2.0.0	Markdown 3.3.3	mkl-fft 1.2.0	mkl-random 1.1.1
mkl-service 2.3.0	multidict 4.7.6	numpy 1.19.2	nvidia-dali 0.15.0
oauthlib 3.1.0	Pillow 8.1.0	pip 20.2.4	protobuf 3.13.0
pyasn1 0.4.8	pyasn1-modules 0.2.8	pycosat 0.6.3	pycparser 2.19
PyJWT 2.0.0	pyOpenSSL 19.0.0	PySocks 1.7.1	requests 2.22.0
requests-oauthlib 1.3.0	rsa 4.7	ruamel-yaml 0.15.46	sailfish 1.0.1
setuptools 50.3.1.post20201107	six 1.15.0	tensorboard 2.3.0	tensorboard-plugin- wit 1.6.0
torch 1.5.1+ali	torchsummary 1.5.1	torchvision 0.6.1	tqdm 4.36.1
typing-extensions 3.7.4.3	urllib3 1.24.2	Werkzeug 1.0.1	wheel 0.35.1
yarl 1.5.1	zipp 3.4.0		

- pytorch-training:1.6.0PAI-gpu-py37-cu100-ubuntu16.04
 - 操作系统: Ubuntu 16.04.6 LTS
 - Python版本: 3.7.4
 - 。 CUDA版本: 10.0

三方库及版本			
absl-py 0.11.0	aiohttp 3.7.3	asn1crypto 1.2.0	async-timeout 3.0.1
attrs 20.3.0	blinker 1.4	cachetools 4.2.0	certifi 2020.12.5
cffi 1.13.0	chardet 3.0.4	click 7.1.2	conda 4.9.2
conda-package- handling 1.6.0	cryptography 2.8	future 0.18.2	google-auth 1.24.0
google-auth-oauthlib 0.4.2	grpcio 1.31.0	idna 2.8	importlib-metadata 2.0.0
Markdown 3.3.3	mkl-fft 1.2.0	mkl-random 1.1.1	mkl-service 2.3.0
multidict 4.7.6	numpy 1.19.2	nvidia-dali 0.15.0	oauthlib 3.1.0
Pillow 8.1.0	pip 20.2.4	protobuf 3.13.0	pyasn1 0.4.8
pyasn1-modules 0.2.8	pycosat 0.6.3	pycparser 2.19	PyJWT 2.0.0
pyOpenSSL 19.0.0	PySocks 1.7.1	requests 2.22.0	requests-oauthlib 1.3.0
rsa 4.7	ruamel-yaml 0.15.46	setuptools 50.3.1.post20201107	six 1.15.0
tensorboard 2.3.0	tensorboard-plugin- wit 1.6.0	torch 1.6.0+ali	torchsummary 1.5.1
torchvision 0.7.0	tqdm 4.36.1	typing-extensions 3.7.4.3	urllib3 1.24.2
Werkzeug 1.0.1	wheel 0.35.1	yarl 1.5.1	zipp 3.4.0

3.3. 公共资源组相关

3.3.1. 数据配置

为了方便用户在提交任务时指定所需的数据集和代码仓,PAI-DLC支持添加文件系统NAS或对象存储OSS的数据集以及Git代码仓。本文介绍如何在PAI-DLC Dashboard中新建数据集配置及代码配置。

前提条件

如果您使用OSS作为存储系统,请确保已经根据业务需求为服务关联角色授予了OSS访问权限。否则挂载 OSS后,进行数据访问时,可能产生I/O错误。关于为服务关联角色授予OSS访问权限的更多信息,请参见授权。

使用限制

由于OSS与NAS不同,并非一个真正的文件系统,而是一个分布式对象存储。因此使用OSS作为存储系统时,不支持文件系统的部分功能。例如,挂载OSS后,不支持对已经存在的文件追加写和覆盖写。

新建数据集配置

- 1. 进入PAI-DLC Dashboard。
 - i. 登录PA控制台。
 - ii. 在左侧导航栏,选择模型开发和训练 > 云原生深度学习训练(DLC)。
 - iii. 在PAI-DLC管理控制台页面,找到类型为公共资源组的工作集群,单击操作列下的集群控制台。
- 2. 在左侧导航栏,单击数据集配置。
- 3. 在数据集配置页面,单击新建数据集配置。
- 4. 在新建数据集配置页面, 配置如下参数。

参数	是否必填	描述
名称	是	数据名称。
描述	否	数据的描述信息,便于区分不同的数据。
存储系统类型	是	支持NAS和OSS。
存储配置	是	仅存储系统类型为NAS时,必须配置该参数。 该参数需要配置为NAS文件系统的ID。您可以登录NAS控制台,在对应的地域,查看NAS文件系统ID。 ② 说明 关于如何上传数据到NAS,请参见NFS文件系统数据的上传下载。
OSS路径	是	仅 存储系统类型 为OSS时,必须配置该参数。 该参数需要配置为同地域的OSS Bucket下的某路径。例 如, <i>oss://mybucket/path/to/dir</i> 。
本地存储目录	否	挂载数据时,所挂载的文件位置。

5. 单击提交。

新建代码配置

1. 进入PAI-DLC Dashboard。

i. 登录PA腔制台。

- ii. 在左侧导航栏,选择模型开发和训练 > 云原生深度学习训练 (DLC)。
- iii. 在PAI-DLC管理控制台页面,找到类型为公共资源组的工作集群,单击操作列下的集群控制台。
- 2. 在左侧导航栏,单击代码配置。
- 3. 在代码配置页面, 单击新建代码配置。
- 4. 在新建代码配置页面, 配置如下参数。
| 参数 | 是否必填 | 描述 |
|------------|------|--|
| 名称 | 是 | 代码仓名称。 |
| 描述 | 否 | 代码仓的描述信息,便于区分不同的代码仓。 |
| Git地址 | 是 | 代码仓的地址。 |
| 默认分支 | 是 | 代码分支,默认值为master。 |
| Git用户名 | 否 | 如果需要访问私有代码仓,则需要指定该参数。 |
| 代码源访问Token | 否 | 如果需要访问私有代码仓,则需要指定该参数。GitHub
Token的获取方式请参见下方的 <mark>获取GitHub账号的</mark>
Token。 |
| 本地存储目录 | 否 | 挂载数据时,所挂载的文件位置。 |

5. 单击**提交**。

获取GitHub账号的Token

- 1. 登录您的Git Hub账号。
- 2. 前往GitHub Developer setting。
- 3. 在Personal access tokens页面,单击右上方的Generate new token。
- 4. 在**Note**下方的文本框中输入Token名称,并在**Select scopes**区域选中权限复选框(建议选中所有权限)。

GitHub Apps	New personal access token						
OAuth Apps	Percenal access takens function like ordinary OAuth access takens. They can be used instead of a password for						
Personal access tokens	Git over HTTPS, or can be us	sed to authenticate to the API over Basic Authentication.					
	Note						
	one-token						
	What's this token for?						
	Select scopes						
	Scopes define the access for personal tokens. Read more about OAuth scopes.						
	🖾 геро	Full control of private repositories					
	repo:status	Access commit status					
	repo_deployment	Access deployment status					
	public_repo	Access public repositories					
	repo:invite	Access repository invitations					
	security_events	Read and write security events					
	workflow	Update GitHub Action workflows					
	write:packages	Upload packages to GitHub Package Registry					
	read:packages	Download packages from GitHub Package Registry					
	Malata maakamaa	Delate peakages from Citly & Deakage Degistry					

5. 单击Generate Token,即可生成Token。生成的Token如下图所示。

Settings / Developer settings										
GitHub Apps	Personal access tokens	Generate new token Revoke all								
OAuth Apps										
Personal access tokens	Make sure to copy your new personal access token now. You won't be able to see it again!									
	✓ 25ac									
Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.										
9 2021 GitHub, Inc. Terms Pr	racy Security Status Docs 🕥 Contact GitHub Pric	cing API Training Blog About								

3.3.2. 创建任务

本文介绍使用公共资源组工作集群时,如何创建深度学习任务。

背景信息

关于创建任务的最佳实践,请参见快速提交单机PyTorch迁移学习任务。

操作步骤

- 1. 进入PAI-DLC Dashboard。
 - i. 登录PA腔制台。
 - ii. 在左侧导航栏,选择模型开发和训练 > 云原生深度学习训练 (DLC)。
 - iii. 在PAI-DLC管理控制台页面,找到类型为公共资源组的工作集群,单击操作列下的集群控制台。
- 2. 在左侧导航栏,单击提交任务。
- 3. 在提交任务页面, 配置如下参数。

区域	参数	描述
	任务名称	深度学习训练的任务名称。长度为2~30个字符,以小写字母开头。
	任务类型	支持以下类型: TF-单机 TF-分布式 Pytorch单机 Pytorch分布式
	数据配置	根据数据存储位置进行配置,数据配置方式请参见数据配置。

区域	参数		描述					
<u>т</u> .н.п.к.			根据代码存储位置进行配置。您可以通过数据配置功能,配置对应的 地址和信息。					
	代码配置		⑦ 说明 由于PAI-DLC会将代码下载至指定工作路径,所以您 需要有代码仓库的访问权限。					
	工作目录		默认为/root路径,且不支持修改。					
	执行命令		支持任意Shell命令。例如,使用 python -c "print('Hello World')" 运行Python。					
	三方库配置		支持以下两种方式配置第三方库: • 三方库列表 :直接在下方文本框中输入三方库。 • requirements.txt文件目录 :将第三方库写入 requirements.txt文件中,在下方文本框中指定该 requirements.txt文件的路径。					
任务资源 配置	Worker 节点配置	节点镜像	工作节点的镜像,支持以下类型的镜像: • 官网镜像 官网镜像列表请参见公共镜像列表。 • 自定义镜像 需要配置公共可访问的Docker Registry Image URL。 TF-分布式 任务中的PS和Worker的镜像必须保持一致(TensorFlow 版本和Python版本必须一致,CPU和GPU资源可以不同)。					
		节点配置	计算节点的规格,详细的规格列表和费用请参见PAI-DLC计费说明。					
		节点数量	当前镜像和规格的节点数量。					

4. 单击提交任务。

3.3.3. 管理任务

PAI-DLC支持通过PAI-DLC Dashboard可视化地管理深度学习任务。本文介绍如何查看任务详情、停止任务及 克隆任务。

进入任务列表页面

- 1. 进入PAI-DLC Dashboard。
 - i. 登录PAI控制台。
 - ii. 在左侧导航栏,选择模型开发和训练 > 云原生深度学习训练 (DLC)。
 - iii. 在PAI-DLC管理控制台页面,找到类型为公共资源组的工作集群,单击操作列下的集群控制台。
- 2. 在左侧导航栏,单击任务列表。

查看任务详情

1. 进入任务列表页面。

2. (可选)在任务列表页面,您可以使用名称、时间区间及任务类型快速定位到待查询的任务。

3. 单击待查看任务的名称,即可查看任务详情(例如执行时长,执行命令及日志)。

停止任务

- 1. 进入任务列表页面。
- 2. (可选)在任务列表页面,您可以使用名称、时间区间及任务类型快速定位到待停止的任务。
- 3. 单击待停止任务操作列下的 ③图标, 即可停止该任务。

克隆任务

- 1. 进入任务列表页面。
- 2. (可选)在任务列表页面,您可以使用名称、时间区间及任务类型快速定位到待克隆的任务。
- 3. 单击待克隆任务操作列下的 图标。系统会克隆该任务,并跳转至提交任务页面,您可以修改相关参数后,再次提交任务。

3.3.4. 通过Python SDK提交任务

您可以通过界面或Python SDK的方式,提交PAI-DLC公共资源组任务。本文介绍如何通过Python SDK提交 PAI-DLC公共资源组任务,包括下载Python SDK、安装Python SDK及创建并提交任务。

前提条件

- 已开通PAI-DLC公共资源组,详情请参见授权。
- 已获取阿里云账户的AccessKey ID和AccessKey Secret , 详情请参见获取AccessKey。

背景信息

关于Python SDK更多接口的说明文档,请参见API参考。

如果您需要通过界面提交PAI-DLC公共资源组任务,请参见创建任务。此外,您也可以登录PAI-DSW探索者版,一键体验本文涉及的操作。

步骤一:下载Python SDK

不同版本的Python SDK下载地址如下:

- Python 2: 下载Python2 SDK
- Python 3: 下载Python3 SDK

步骤二:安装Python SDK

1. 下载Python SDK压缩包。

wget https://sdk-portal-cluster-prod.oss-cn-zhangjiakou.aliyuncs.com/downloads/u-3536038a-3de7-4f 2e-9379-0cb309d29355-python-pai-dlc.zip

其中 https://sdk-portal-cluster-prod.oss-cn-zhangjiakou.aliyuncs.com/downloads/u-3536038a-3de7-4f2e-9379-0cb309d29355-python-pai-dlc.zip 表示Pvthon3 SDK的下载地址, 您可以根据实际需求将其替换为 Pvthon2 SDK的下载地址 https://sdk-portal-cluster-prod.oss-cn-zhangjiakou.aliyuncs.com/downloads/u-46a0db30-e48a-4e12-8c87-b3a86321f3e2-python2-pai-dlc.zip 。

2. 对Python SDK的压缩包进行解压。

unzip u-3536038a-3de7-4f2e-9379-0cb309d29355-python-pai-dlc.zip

其中 u-3536038a-3de7-4f2e-9379-0cb309d29355-python-pai-dlc.zip 表示已经下载的Python SDK压缩包 名称,需要根据实际情况替换。

3. 安装Python SDK。

pip install ./pai-dlc-20201203

其中 pai-dlc-20201203 表示解压后的Python SDK安装包名称,需要根据实际情况替换。

步骤三: 创建并提交任务

创建任务时,需要使用当前阿里云账号对应的AccessKey信息。关于如何获取AccessKey,请参见获取 AccessKey。创建并提交任务的具体调用代码如下所示。

```
from alibabacloud_pai_dlc20201203.client import Client
from alibabacloud_tea_openapi.models import Config
from alibabacloud_pai_dlc20201203.models import CreateJobRequest, JobSpec
#初始化一个Client以访问DLC的API。
region = 'cn-beijing'
client = Client(
 Config(
   access_key_id='<替换成您自己的AccessKeyID>',
   access_key_secret='<替换成您自己的AccessKeySecret>',
   region_id=region,
   endpoint=f'pai-dlc.{region}.aliyuncs.com'
 )
)
#声明任务的资源配置,关于镜像选择可以参考文档中公共镜像列表,也可以传入自己的镜像地址。
spec = JobSpec(
 type='Worker',
 image='registry-vpc.cn-beijing.aliyuncs.com/pai-dlc/tensorflow-training:1.15-cpu-py36-ubuntu18.04',
 pod_count=1,
 ecs_spec='ecs.c6.large'
)
#声明任务的执行内容。
req = CreateJobRequest(
 display_name='sample-dlc-job',
 job_type='TFJob',
 job_specs=[spec],
 user_command='echo "Hello World"'
)
#提交任务。
response = client.create_job(req)
#获取任务ID。
job_id = response.body.job_id
#查询任务状态。
job = client.get_job(job_id).body
print('job status:', job.status)
#查看任务执行的命令。
job.user_command
```

3.3.5. 定时自动提交任务

本文为您介绍如何搭配使用PAI-DLC公共资源组和DataWorks,进行任务定时提交。

前提条件

- 购买DataWorks独享调度资源组,详情请参见DataWorks独享资源。
- 为PAI-DLC授权,详情请参见授权。

操作流程

您可以按照如下流程实现定时自动提交任务:

- 1. 步骤一: 创建独享调度资源组
- 2. 步骤二: 绑定归属的工作空间
- 3. 步骤三: 安装部署工具包dlc
- 4. 步骤四: 创建工作流
- 5. 步骤五: 提交任务测试
- 6. 步骤六:执行定时调度

步骤一: 创建独享调度资源组

- 1. 登录DataWorks控制台。
- 2. 在左侧导航栏,单击资源组列表。
- 3. 在独享资源组页签,单击创建调度资源组。
- 4. 在创建独享资源组面板, 配置参数。

参数	描述							
资源组类型	选择 独享调度资源组 。							
	资源组名称,必须在租户内唯一。							
资源组名称	⑦ 说明 租户即阿里云账号,一个租户(阿里云账号)下可以有多个RAM用户。							
资源组备注	资源组的简单描述,便于区分各资源组。							
订单号	选择已购买的独享资源组订单号。如果未购买,请单击 购买 。							

5. 单击确定。

⑦ 说明 独享资源组在20分钟内完成环境初始化,请耐心等待其状态更新为运行中。

步骤二: 绑定归属的工作空间

独享调度资源组需要绑定归属的工作空间,才可以在对应的工作空间下选择该资源组。

1. 在独享资源组页签,单击相应资源组后的修改归属工作空间。

2. 在修改归属工作空间对话框的分配到工作空间区域,单击目标工作空间操作列下的绑定。

步骤三:安装部署工具包dlc

安装dlc工具包,需要联系管理员授权。

- 1. 创建命令。
 - i. 在资源组列表的独享资源组页签,单击目标资源组操作列下的运维助手。
 - ii. 在运维助手页面, 单击**创建命令**。
 - iii. 在**创建命令**页面, 配置参数。

参数	描述							
命令名称	命令的名称。							
命令描述	命令的简要描述,便于区分各命令。							
命令类型	系统默认Shell类型,不支持修改。							
命令内容	输入如下命令。 wget -P /home/admin/usertools/tools/ http://dlc-tools.oss-cn- zhangjiakou.aliyuncs.com/release/linux/dlc chmod +x /home/admin/usertools/tools/dlc							
安装目录	安装至/home/admin/usertools/tools/目录。							
超时时间	命令执行的超时时间,单位为秒。如果命令执行超时,则系统强制结束命 令。建议配置为60秒。							

- iv. 单击创建。
- 2. 执行命令。
 - i. 在运维助手页面, 单击上一步已创建命令操作列下的运行命令。

DataWorks		DataWorks / 资源组列表 P								帮助文档
概览		← ywra								
工作空间列表						查看环境配置详情	环境基线命	令编排	创建命令	刷新
资源组列表		命令ID/名称	描述	命令类型	创建时间	是否基线命令		操作		
报警资源				Shell	2021-05-20 15:14:57	否	1	运行命令	删除	
开放平台 New		test								
计算引擎列表	^									
MaxCompute										

ii. 在运行命令面板,选择基线命令为是,运行成功后纳入环境基线,其他参数使用默认配置。

- iii. 单击运行。
- 3. 查看命令执行情况。

i. 在运维助手页面下方的列表, 单击相应命令后的查看结果。

aWorks	(DataWorks / 资源组列表									帮助文档
概览	•	 yuts,privat 	k_schedu	ler -							
工作空间列表							查看环境配置词	¥情 环境基线:	命令编排	创建命令	刷新
资源组列表		命令ID/名称	描述	命令类型	创建时间		是否基线命令		操作		
报警资源		Shell 2021-05-20 15:14:57				否运行			う命令 删除		
开放平台 New		lest									
计算引擎列表 个											
MaxCompute											
Graph Compute					1111111						
Hologres											刷新
		执行状态	命令执行ID		命令ID/名称	命令类型		命令内容		操作	
		✓ 成功			test	Shell				宣看结果	

ii. 在命令执行结果对话框, 查看命令执行情况。如果执行进度为100%, 则dlc工具包安装成功。

展开面板可以查看	旨命令的执行结果			
				刷新
> 执行成功				
12800K 12850K 12900K 12950K 13050K 13150K 13200K 13200K 13200K 13200K 2021-05-20 '/home/admin	15:15:04 (21.9 MB/ n/usertools/tools/d	96% 96% 97% 97% 97% 98% 98% 98% 98% 98% 99% 100% (s) - Ic' saved [136646	264M 0s 290M 0s 227M 0s 286M 0s 305M 0s 257M 0s 235M 0s 235M 0s 250M 0s 280M 0s 247M=0.6s	1

步骤四:创建工作流

- 1. 进入数据开发页面。
 - i. 登录DataWorks控制台。
 - ii. 在左侧导航栏, 单击工作空间列表。
 - iii. 选择工作空间所在地域后,单击相应工作空间后的进入数据开发。
- 2. 鼠标悬停至 图标,单击业务流程。



3. 在新建业务流程对话框, 输入业务名称和描述。

 ↓ 注意 业务名称必须是大小写字母、中文、数字、下划线(_)及小数点(.),且不能超过 128个字符。

- 4. 单击新建。
- 5. 在业务流程页面,拖拽通用 > Shell至右侧画布。



6. 在新建节点页面的节点名称文本框, 输入部署节点。

7. 单击提交。

步骤五:提交任务测试

因为定时提交任务基于原有任务节点,所以定时提交之前,需要进行冒烟测试,创建首个任务节点。如果已 经存在初始节点,则直接执行步骤六。

1. 编辑部署脚本。

- i. 在业务流程页面,双击已创建的Shell节点(部署节点)。
- ii. 在Shell节点页面, 输入如下命令。

```
#生成任务描述文件。
cat << EOF > jobfile
name=dataworks-job
workers=1
worker_spec=ecs.g6.large
worker_image=registry-vpc.cn-hangzhou.aliyuncs.com/pai-dlc/pytorch-training:1.7.1-gpu-py37-cu
110-ubuntu18.04
command=echo $(date)
EOF
#提交任务。
/home/admin/usertools/tools/dlc submit pytorchjob\
 --access_id=<access_id> \
 --access_key=<access_key>
 --endpoint=pai-dlc.cn-hangzhou.aliyuncs.com \
 --region=cn-hangzhou \
 --job_file=./jobfile \
 --interactive
```

jobfile是描述任务相关信息的文件,具体配置细节请参见提交命令。 endpoint参数与地域的对应关 系如下。

地域	Endpoint
华东2(上海)	pai-dlc.cn-shanghai.aliyuncs.com
华北2(北京)	pai-dlc.cn-beijing.aliyuncs.com
华东1(杭州)	pai-dlc.cn-hangzhou.aliyuncs.com
华南1(深圳)	pai-dlc.cn-shenzhen.aliyuncs.com

2. 运行脚本。

i. 在Shell节点页面,单击页面上方的 ③图标。

- ii. 在警告对话框, 单击继续运行。
- iii. 在运行参数页面,选择调度资源组为已创建的独享资源组。
- iv. 单击**确定**。运行完成后,即可生成一个任务。您可以前往PAI-DLC集群控制台的任务列表页面,查 看任务。

步骤六:执行定时调度

> 文档版本: 20210604

- 1. 执行调度任务。
 - i. 在Shell节点页面,单击页面右侧的调度配置。
 - ii. 在调度配置页面的时间属性区域,选择调度周期。
 - iii. 在调度依赖区域,单击依赖的上游节点后的使用工作空间根节点。
 - iv. 配置依赖关系,详情请参见配置调度依赖。
 - v. 单击Shell节点页面上方的凹图标,保存配置。
 - vi. 单击Shell节点页面上方的了图标, 提交调度任务。
- 2. 查看定时调度的运行实例。
 - i. 在Shell节点页面,单击右上方的运维中心。
 - ii. 在运维中心页面,选择周期任务运维 > 周期实例。
 - iii. 在实例详情页面,查看模型自动部署的定时时间。
 - iv. 选择操作列下的**更多 > 查看运行日志**, 查看每次定时提交任务的运行日志。

🚱 🔮 遠雄中心										e DetaSt		द 🔻 :
È 這堆大屏 Ś ś ś ś ś ś ś	- 114	3援家: 节点名称/节;	100 Q 业务日期 : 昨天 前天	全部 2020-06-30 · 2020-0	16-30 📋 1	■「病英型: 请选择节点关型 >	🛃 我的节点 🗌 我的2	出错节点 🗌 我的来完。	g节点 🗌 重跨过的节点	变慢节点		
Li 周期任务过端 周期任务 周期实例 2		基本信息		任务类型	责任人	优先级 扩	定时时间 11	11 第日表生	11 时间就开	结束时间 J1 Regi	操作	
补数据安例 测试实例		#7	07-01 10:00:13 ~ 10:02:14 (dur 2	SHELL	-	ī	2020-07-01 10:00:00	2020-06-30	2020-07-01 10:00:18	2020-07-01 10:02:14	DAG图 1 终。	止运行 重進 更多 🔻
 ○ 手助任务运输 ▲ 智能溢拉 		⊘ ₿₽₽⊅.£ #7	07-01 12:00:03 ~ 12:02:10 (dur 2	SHELL		1	2020-07-01 12:00:00	2020-06-30	2020-07-01 12:00:03	2020-07-01 12:02:10	DAGE 1 (S.	止退行 重路 更多 ▼
		Ø 254870,00 ⊄: 1498 m7s)	07-01 14:00:09 ~ 14:02:16 (dur 2	SHELL	-	1 3	2020-07-01 14:00:00	2020-06-30	2020-07-01 14:00:09	2020-07-01 14:02:16	DAG图 终.	止运行「重命」更多 マ
		⊘ 部署节点 # 1498 m9s)	07-01 16:00:11 ~ 16:02:20 (dur 2	SHELL	1000	1	2020-07-01 16:00:00	2020-06-30	2020-07-01 16:00:11	2020-07-01 16:02:20	DAGIN L 15.	止這行 重整 更多 - 重跑下游
		 ジ 部署予点 #7 1498 m59s) 	07-01 18:00:08 ~ 18:02:07 (dur 1	SHELL	-	1	2020-07-01 18:00:00	2020-06-30	2020-07-01 18:00:08	2020-07-01 18:02:07	DAGE	副成功 監停 (赤結) 院型 (解冻) 副者の接
		⊘ 84월17.4 4)498 m8s)	07-01 20:00:07 ~ 20:02:15 (dur 2	SHELL	-	1	2020-07-01 20:00.00	2020-06-30	2020-07-01 20:00:07	2020-07-01 20:02:15	DAGE	运行诊断 上下游 数看运行日志
		 ● 部署节点 #1 498 m2s) 	07-01 22:00:06 ~ 22:02:08 (dur 2	SHELL	104400	1	2020-07-01 22:00:00	2020-06-30	2020-07-01 22:00:06	2020-07-01 22:02:08	DAG盟 终.	止运行 重路 更多 ▼
		⊘ #7 681 s)	07-01 00:00:00 ~ 00:00:00 (dur 0	虚节点	-	1	2020-07-01 00:00:00	2020-06-30	2020-07-01 00:00:00	2020-07-01 00:00:00	DAG图 1 终。	止运行 重流 更多 🔻
=		终止运行 星路	置成功 图停 (冻结) 快发 (机	838)								

3.3.6. 客户端工具

3.3.6.1. 命令列表

您可以通过客户端工具管理PAI-DLC中的任务和数据。本文介绍客户端工具提供的常用命令。 客户端工具提供了用户认证、创建、删除、提交、停止及查询系列命令,命令列表如下。

类别	命令
<u> </u>	自动补全 (completion)
准電工1	用户认证(config)
	创建代码源(create code_source)
创建命令	创建数据源(create data_source)

类别	命令
删除会会	删除代码源(delete code_source)
א נוט זא ניונו 🗸	删除数据源
坦六今冬	提交TensorFlow训练任务(submit tfjob)
淀义叩マ	提交PyTorch训练任务(submit pytorchjob)
停止命令	停止训练任务(stop)
	查看任务日志(logs)
大均今久	查看仓库中的镜像(get images)
世世日イ	查看代码源(get code_source)
	查看数据源(get data_source)

3.3.6.2. 准备工作

您可以通过客户端工具管理PAI-DLC中的数据和任务。在使用客户端工具之前,您需要下载客户端并进行用 户认证。本文介绍下载客户端及进行用户认证的命令详情。

背景信息

手册使用客户端工具管理数据和任务之前,您需要按照如下流程完成准备工作:

- 1. 下载客户端工具。
- 2. 执行自动补全命令,查看客户端工具提供的命令集,详情请参见自动补全(completion)。
- 3. 用户认证 (config)。

下载客户端工具

您可以通过以下链接下载适合的客户端工具:

- 下载Linux 64版本的客户端
- 下载Mac版本的客户端

下载后的客户端工具无需安装,您只需要执行 chmod +x dlc 命令,为其添加可执行权限,即可在命令行中 调用客户端工具提供的命令。

自动补全 (completion)

功能

自动补全功能和社区的Kubectl、Arena命令行的自动补全功能使用方式相同。在命令行中,您可以通过连续按两次Tab键使用PAI-DLC客户端工具的自动补全功能,从而对尚未完成的命令进行提示。

格式

dlc completion <shelltype>

参数

> 文档版本: 20210604

<shelltype>表示待生成自动补全脚本的命令行类型。系统支持的命令行种类包括bash、fish、powershell及zsh。

示例

本文以bash类型为例,介绍自动补全的使用方法:

i. 执行如下命令,使自动补全命令生效。由于自动补全功能依赖于bash-completion包,因此如果执行 过程中报错,则表明您的环境中未安装bash-completion包。

source<(dlc completion bash)</pre>

ii. 如果没有安装bash-completion包,则使用如下命令安装,并通过 source 命令使之生效。然后再执行上一步命令。

如果已经安装了bash-completion包,即执行上一步命令时没有报错,则跳过该步骤。

Mac系统中使用如下命令。 brew install bash-completion && source /usr/local/etc/bash_completion # Linux系统中使用如下命令。 yum install bash-completion && source /etc/profile.d/bash_completion.sh

iii. 在如下 dlc 后连续按下两次Tab键,即可查看PAI-DLC客户端工具提供的命令。

dlc

系统返回的PAI-DLC客户端工具提供的命令如下所示。

completion config create delete get help logs stop submit

⑦ 说明 针对其他Shell类型,您可以通过 dlc completion <zsh | fish | powershell> --help 命令获取 详细的使用方法。

用户认证 (config)

功能

首次使用PAI-DLC客户端工具管理数据和任务时,需要先使用阿里云账号的AccessKey ID和AccessKey Secret进行身份认证。一次认证后,再次使用时无需认证。

格式

dlc config --access_id <yourAccessKeyId> --access_key <yourAccessKeySecret> [--endpoint <yourEndpoin
t>] [--region <yourRegion>]

参数

需要替换的参数	是否必选	描述	类型
<youraccesskeyid></youraccesskeyid>	是	阿里云账号的AccessKey ID。	STRNG
<youraccesskeysecret></youraccesskeysecret>	是	阿里云账号的AccessKey Secret。	STRNG
<yourendpoint></yourendpoint>	否	使用的PAI-DLC服务所在地域的Endpoint , 默认为pai-dlc.cn- shanghai.aliyuncs.com。	STRNG

需要替换的参数	是否必选	描述	类型
<yourregion></yourregion>	否	使用的PAI-DLC服务所在地域,默认为cn- shanghai。	STRNG

• 示例

进行用户认证后,系统默认会将配置内容保存至家目录的.dlc/config文件中。例如,执行以下类似用户命令。

dlc config -access_id <yourAccessKeyId> -access_key <yourAccessKeySecret>

系统返回如下类似内容。

Configuration saved to: ~/.dlc/config

3.3.6.3. 创建命令

您可以通过客户端工具创建代码源和数据源。本文介绍创建相关的命令详情,包括调用格式、参数解释及使 用示例。

创建代码源(create code_source)

• 功能

用于创建PAI-DLC训练任务中使用的代码源。

格式

dlc create code_source --display_name <yourCodeSourceName> [--description <yourCodeSourceDesc>] -code_repo_url <yourCodeRepoUrl> [--code_branch <yourCodeBranch>] [--mount_path <yourMountPath
>]

参数

需要替换的参数	是否必选	描述	类型
<yourcodesourcename ></yourcodesourcename 	是	创建的代码源名称。	STRING
<yourcodesourcedesc></yourcodesourcedesc>	否	对代码源的描述,默认值为空。	STRING
<yourcoderepourl></yourcoderepourl>	是	代码源对应的GitHub Repo URL。	STRING
<yourcodebranch></yourcodebranch>	否	创建代码源使用的分支,默认值为空。	STRING
<yourmountpath></yourmountpath>	否	在服务器上映射的路径,默认为 <i>/root/cod e/</i> 。	STRING

• 示例

您可以从GitHub的一个Repo创建代码源,示例如下。

dlc create code_source --display_name test_code_source_1 --code_repo_url https://github.com/tensorfl ow/examples

创建成功后,系统会返回已经创建好的代码源ID和创建代码源的请求ID(如下所示),以便后续答疑使用。

+		+	+
C	odeSourceId	RequestId	
+		+	+
code-	20210411xxxxxx-3	lm541xxxxxx D4Bxxxx	х-хххх-49FC-А600-хххххххххххх
+		+	+

创建数据源 (create data_source)

功能

用于创建PAI-DLC训练任务中使用的数据源。

格式

dlc create data_source [--data_source_type <yourDataSourceType>] --display_name <yourDataSourceNa me> [--description <yourDataSourceDesc>] --file_system_id <yourFileSystemId> [--mount_path <yourMo untPath>]

参数

需要替换的参数	是否必选	描述	类型
<yourdatasourcetype></yourdatasourcetype>	否	数据源类型,默认值为nas。目前仅支持 NAS。	STRING
<yourdatasourcename ></yourdatasourcename 	是	创建的数据源名称。	STRING
<yourdatasourcedesc></yourdatasourcedesc>	否	对数据源的描述,默认值为空。	STRING
<yourfilesystemid></yourfilesystemid>	是	NAS文件系统的ID。您可以登录 <mark>NAS控制</mark> 台,在对应的地域,查看NAS文件系统ID。	STRING
<yourmountpath></yourmountpath>	否	在服务器上映射的路径,默认为 <i>/root/dat</i> <i>a/</i> 。	STRING

• 示例

以从NAS数据创建数据源为例,需要指定file_system_id参数,命令如下。

dlc create data_source --display_name test_data_source_1 --file_system_id XXXXXXXXXXX

创建成功后,系统会返回已经创建好的数据源ID和创建数据源的请求ID(如下所示),以便后续答疑使用。

+		+		+
	DataSourceld	Ι	RequestId	1
+		+		+
dat	a-20210411xxxxx	3lm533	Sxxxxxx D4Bxxxxx	<-xxxx-49FC-A600-xxxxxxxxxxxxxx
+		+		+

3.3.6.4. 删除命令

您可以通过客户端工具删除代码源和数据源。本文介绍删除相关的命令详情,包括调用格式、参数解释及使 用示例。

删除代码源(delete code_source)

功能

用于删除已经创建的代码源。

格式

dlc delete code_source <yourCodeSourceId>[--force]

参数

参数	是否必选	描述	类型
<yourcodesourceid></yourcodesourceid>	是	待删除的代码源ID。	STRING
[force]	否	删除代码源时,如果使用了该参数,则无需 用户确认,即可删除代码源。反之,需要您 确认后才会执行删除命令。	不涉及

示例

dlc delete code_source code-20210411xxxxxx-3lm541xxxxxx --force

执行上述命令,系统会删除ID为code-20210411xxxxxx-3lm541xxxxx的代码源,且无需您确认。

删除数据源

功能

用于删除已经创建的数据源。

格式

dlc delete data_source <yourDataSourceId> [--force]

参数

参数	是否必选	描述	类型
<yourdatasourceld></yourdatasourceld>	是	待删除的数据源ID。	STRING
[force]	否	删除数据源时,如果使用了该参数,则您无 需确认,即可删除数据源。反之,需要您确 认后才会执行删除命令。	不涉及

• 示例

dlc delete data_source data-20210411xxxxxx-3lm533xxxxxx --force

执行上述命令,系统会删除ID为data-20210411xxxxxx-3lm533xxxxx的数据源,且无需您确认。

3.3.6.5. 提交命令

您可以通过客户端工具提交训练任务。本文介绍提交任务相关的命令详情,包括调用格式、参数解释及使用 示例。

提交TensorFlow训练任务(submit tfjob)

功能

用于提交TensorFlow训练任务。

● 格式

系统支持通过命令行参数或任务参数描述文件的方式提交TensorFlow任务。

```
#方式一:命令行参数的方式。
dlc submit tfjob --name=<yourTaskName> \
--command=<"yourStartCommand"> \
[--<parameterName>=<parameterValue>]
#方式二:任务参数描述文件的方式
dlc submit tfjob --job_file=<yourJobFileName>
```

参数

如果通过命令行参数的方式提交TensorFlow任务,则需要将命令中的如下参数替换为实际值。如果通过 任务参数描述文件的方式提交TensorFlow任务,则将任务参数描述文件中支持的参数以 <parameterName >=<parameterValue> 的形式写入文件中。

参数	是否必选	描述	类型	任务参数描述 文件中是否支 持该参数
name= <yourtaskname></yourtaskname>	是	任务的名称 <i>,</i> 多个任务名称可以 相同。	STRING	是
command= <"yourStartComm and">	是	各个节点的启动命令。	STRING	是
<parametername >= <parametervalue ></parametervalue </parametername 	否	<parametername>为提交任务 相关的可选参数名 称,<parametervalue>为参数 值。提交TensorFlow任务的相关 可选参数列表如提交TensorFlow 任务的相关可选参数所示。</parametervalue></parametername>	具体参见下表 中的每个可选 参数	具体参见下表 中的每个可选 参数

提交TensorFlow任务的相关可选参数

类别	参数名称	描述	类型	任务参数描述 文件中是否支 持该参数
	code_source	代码源,只能传入单个代码源,默认值 为空。	STRING	是
	data_sources	数据源,默认值为空。如果存在多个数 据源,则数据源之间以半角逗号(,) 分隔。	STRING	是

类别	参数名称	描述	类型	任务参数描述 文件中是否支 持该参数
通用参数	job_file	任务参数描述文件,默认值为空。文件 中的内容格式为 <parametername> = <parametervalue> , <parametern ame>与命令行参数名称保持一致。 如果指定了该参数,则系统优先使 用job_file中的参数。</parametern </parametervalue></parametername>	STRING	否
	thirdparty_lib s	Python三方库,默认值为空。如果存 在多个三方库,则三方库之间以半角逗 号(,)分隔。	STRING	是
	thirdparty_lib _dir	Python三方库安装时使用的 <i>requireme nts.txt</i> 文件所在的文件夹,默认值为 空。	STRING	否
	chief	 是否启用TensorFlow Chief节点,该 参数取值包括: false: 默认值,表示关闭 TensorFlow Chief节点。 true:表示开启TensorFlow Chief 节点。 	BOOL	是
	chief_image	TensorFlow Chief节点的镜像,默认 值为空。	STRING	是
	chief_spec	TensorFlow Chief节点使用的服务器 型号,默认值为空。	STRING	是
	master_imag e	TensorFlow Master节点的镜像,默认 值为空。	STRING	是
	master_spec	TensorFlow Master节点使用的服务器 型号。	STRING	是
TensorFlow特 有参数	masters	TensorFlow Master节点的数量,默认 值为0。	INT	是
	ps	TensorFlow Parameter Server节点的 数量,默认值为0。	INT	是
	ps_image	TensorFlow Parameter Server节点的 镜像,默认值为空。	STRING	是
	ps_spec	TensorFlow Parameter Server节点使 用的服务器型号,默认值为空。	STRING	是
	worker_imag e	TensorFlow Worker节点的镜像,默 认值为空。	STRING	是

类别	参数名称	描述	类型	任务参数描述 文件中是否支 持该参数
	worker_spec	TensorFlow Worker节点使用的服务 器型号,默认值为空。	STRING	是
	workers	TensorFlow Worker节点的数量,默 认值为0。	INT	是

示例

。 通过命令行参数提交一个2 Worker+ 1 PS的分布式作业,示例如下。

```
dlc submit tf --name=test_2021 --ps=1 \
```

--ps_spec=ecs.g6.8xlarge \

--ps_image=registry-vpc.cn-beijing.aliyuncs.com/pai-dlc/tensorflow-training:1.12.2PAI-cpu-py27-ubu ntu16.04 \

--workers=2 \

--worker_spec=ecs.g6.4xlarge \

--worker_image=registry-vpc.cn-beijing.aliyuncs.com/pai-dlc/tensorflow-training:1.12.2PAI-cpu-py27-ubuntu16.04 \

--command="python /root/data/dist_mnist/code/dist-main.py --max_steps=10000 --data_dir=/root/d ata/dist_mnist/data/" \

系统返回如下类似结果。

++	
Jobid Requestid	
++	
dlc-2021xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	xxx
++	
[Info] Job [dlc-2021xxxxxxxxxxxxxxxxxxxxxxxxxxxx] is [Running]	
[Info] Job [dlc-2021xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx] is [Running]	
[Info] Job [dlc-2021xxxxxxxxxxxxxxxxxxxxxxxxxxx] is [Running]	
[Info] Job [dlc-2021xxxxxxxxxxxxxxxxxxxxxxxxxxx] is [Running]	
-	

。 通过任务参数描述文件提交一个2 Worker+ 1 PS的分布式作业,示例如下。

dlc submit tfjob --job_file=job_file.dist_mnist.1ps2w

其中job_file.dist_mnist.1ps2w为任务参数描述文件,采用 <parameterName>=<parameterValue> 的格 式填写参数。job_file.dist_mnist.1ps2w的内容如下所示。

提交PyTorch训练任务(submit pytorchjob)

功能

用于提交PyTorch训练任务。

格式

系统支持通过命令行参数或任务参数描述文件的方式提交TensorFlow任务。

```
#方式一:命令行参数的方式。
dlc submit pytorchjob --name=<yourTaskName> \
--command=<"yourStartCommand"> \
[--<parameterName>=<parameterValue>]
#方式二:任务参数描述文件的方式。
dlc submit pytorchjob --job_file=<yourJobFileName>
```

参数

如果通过命令行参数的方式提交PyTorch任务,则需要将命令中的如下参数替换为实际值。如果通过任务参数描述文件的方式提交PyTorch任务,则将任务参数描述文件中支持的参数以 <parameterName>=<para meterValue> 的形式写入文件中。

参数	是否必选	描述	类型	任务参数描述文件 中是否支持该参数
name= <yourtaskname></yourtaskname>	是	任务的名称,多个 任务名称可以相 同,默认值为空。	STRING	是
command= <"yourStartComm and">	是	各个节点的启动命 令 <i>,</i> 默认值为空。	STRING	是

参数	是否必选	描述	类型	任务参数描述文件 中是否支持该参数
<parametername >= <parametervalue ></parametervalue </parametername 	否	<parametername >为提交任务相关的 可选参数名 称,<parameterv alue>为参数值。提 交PyTorch任务的 相关可选参数列表 如提交PyTorch任 务的相关可选参 数所示。</parameterv </parametername 	具体参见下表中的 每个可选参数	具体参见下表中的 每个可选参数

提交PyTorch任务的相关可选参数

类别	参数名称	描述	类型	任务参数描述文件 中是否支持该参数
	code_source	代码源,只能传入 单个代码源,默认 值为空。	STRING	是
	data_sources	数据源,默认值为 空。如果存在多个 数据源,则数据源 之间以半角逗号 (,)分隔。	STRING	是
通用参数	job_file	任务参数描述文 件,默认值为空。 文件的内容格式为 <narametername >=<narameterval ue>, <parameter rName>与命令行 参数名称保持一 致。 如果指定了该参 数,则系统优先使 用job_file中的参 数。</parameter </narameterval </narametername 	STRING	否
	thirdparty_libs	Python三方库,默 认值为空。如果存 在多个三方库,则 三方库之间以半角 逗号(,)分隔。	STRING	是

类别	参数名称	描述	类型	任务参数描述文件 中是否支持该参数
	thirdparty_lib_dir	Python三方库安装 时使用的 <i>requirem ents.txt</i> 文件所在 的文件夹 <i>,</i> 默认值 为空。	STRING	否
	master_image	PyTorch Master节 点的镜像 <i>,</i> 默认值 为空。	STRING	是
	master_spec	PyTorch Master节 点使用的服务器型 号,默认值为空。	STRING	是
ひょうたちちを新	masters	PyTorch Master节 点的数量 <i>,</i> 默认值 为0。	INT	是
PyTorcn符有参数	worker_image	PyTorch Worker节 点的镜像,默认值 为空。	STRING	是
	worker_spec	PyTorch Worker节 点使用的服务器型 号,默认值为空。	STRING	是
	workers	PyTorch Worker节 点的数量 <i>,</i> 默认值 为0。	INT	是

示例

通过命令行参数提交一个GPU的人脸识别模型训练任务,示例如下。

```
dlc submit pytorchjob --name=test_pt_face \
```

--workers=1 \

--worker_spec=ecs.gn6e-c12g1.3xlarge \

--worker_image=registry-vpc.cn-beijing.aliyuncs.com/pai-dlc/pytorch-training:1.7.1-gpu-py37-cu110-ub untu18.04 \

系统返回如下类似结果。

++
Jobid Requestid
++
dlc-2021xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
++
[Info] Job [dlc-2021xxxxxxxxxxxxxxxxxxxxxxxxxxxx] is [Running]
[Info] Job [dlc-2021xxxxxxxxxxxxxxxxxxxxxxxxxxxxx] is [Running]
[Info] Job [dlc-2021xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx] is [Running]
[Info] Job [dlc-2021xxxxxxxxxxxxxxxxxxxxxxxxxxxx] is [Running]

3.3.6.6. 停止命令

您可以通过客户端工具停止训练任务。本文介绍停止任务相关的命令详情,包括调用格式、参数解释及使用 示例。

停止训练任务 (stop)

功能

用于停止正在运行或正在创建的任务。如果停止已经处于停止或结束状态的任务,则系统返回告警信息。

格式

dlc stop <yourJobId> [--force]

● 参数

参数	是否必选	描述	类型
<yourjobid></yourjobid>	是	待停止任务的ID。	STRING
[force]	否	停止任务时,如果使用了该参数,则您无需确认,即 可停止任务。反之,需要您确认后才会执行停止命 令。	不涉及

- 示例
 - 停止一个已经处于停止状态的任务(假设ID为dlc-20210411xxxxxx-xxxxxxxxxx的任务处于停止状态)

dlc stop dlc-20210411xxxxxx-xxxxxxxxxxxxx

系统会返回如下类似的告警信息。

[WARN] The job you are trying to stop is not in creating/running status, can not be stopped.

○ 停止一个正在运行的任务(假设ID为dlc-20210411xxxxxx-xxxxxxxxxxx的任务处于运行状态)

dlc stop dlc-20210411xxxxxx-xxxxxxxxxxxxxx --force

系统返回如下类似结果。

[INFO] Job [dlc-20210411xxxxx-xxxxxxxxxx] has been stopped.

3.3.6.7. 查询命令

您可以通过客户端工具查看任务日志、镜像、代码源及数据源。本文介绍查询相关的命令详情,包括调用格 式、参数解释及使用示例。

背景信息

客户端工具提供以下查询相关命令:

- 查看任务日志 (logs)
- 查看仓库中的镜像 (get images)
- 查看代码源 (get code_source)
- 查看数据源 (get data_source)

查看任务日志 (logs)

功能

查看一个训练任务的日志详情。

格式

dlc logs <yourJobId> <yourPodId> [--max_events_num <yourMaxNum>] [--start_time <yourStartTime>] [-end_time <yourStartTime>]

 参数 	
------------------------	--

参数	是否必选	描述	类型
<yourjobld></yourjobld>	是	待查看训练任务的ID。	STRING
<yourpodid></yourpodid>	是	待查看日志的Pod ID。在分布式任务场景 下,存在多个Pod。	STRING
max_events_num <yourmaxnum></yourmaxnum>	否	返回的日志最大行数,默认值为2000。	INT
start_time <yourstarttime< td=""><td>否</td><td>日志查询的起始时间,默认值为7天前。例 如,start_time 2020-11-08T16:00:00Z</td><td>STRING</td></yourstarttime<>	否	日志查询的起始时间,默认值为7天前。例 如,start_time 2020-11-08T16:00:00Z	STRING
end_time <yourstarttime></yourstarttime>	否	日志查询的截止时间,默认值为当前时间。 例如,end_time 2020-11- 08T17:00:00Z。	STRING

示例

针对分布式训练任务的0号Worker节点,获取十行日志。

dlc logs dlc-20210411xxxxxx-xxxxxxxxx dlc-20210411xxxxxx-xxxxxxxxx-worker-0 --max_events_num 10

系统返回如下类似结果。

WARN: ./requirements.txt not found, skip installing requirements.

| PAI Tensorflow powered by Aliyun PAI Team. |

Network is under initialization...

Network successfully initialized.

[2021-04-16 12:27:56.368026] [INFO] [7#7] [tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 AVX512F FMA

[2021-04-16 12:27:56.375600] [INFO] [7#7] [tensorflow/core/distributed_runtime/master.cc:84] Disable AV X512.

[2021-04-16 12:27:56.375605] [INFO] [7#7] [tensorflow/core/distributed_runtime/master.cc:87] CPU Vendo r ID: GenuineIntel

查看仓库中的镜像 (get images)

功能

从PAI-DLC的镜像仓库中获取可用的镜像信息,您可以在提交训练任务时使用这些镜像。

格式

dlc get images [--order=<yourOrder>] [--framework=<yourFramework>] [--device=<yourDeviceType>] [--p rovider=<yourProvider>]

•	参	数
•	孧	剱

参数	是否必选	描述	类型
order= <yourorder></yourorder>	否	排序顺序,取值包括: • desc:默认值,表示降序。 • asc:表示升序。	STRING
framework= <yourframework></yourframework>	否	镜像包含的框架类型,默认值为空,表示任 意类型。该参数取值包括: 。 PyTorchJob 。 TFJob	STRING
device= <yourdevicetype ></yourdevicetype 	否	支持的设备类型,默认值为空,表示所有类型。该参数取值包括: 。 cpu 。 gpu	STRING
provider= <yourprovider></yourprovider>	否	镜像提供源,默认值为空,表示所有来源。 该参数取值包括: • Community:社区镜像。 • PAI: PAI提供的官方镜像。	STRING

• 示例

从仓库中获取社区提供的Pytorch GPU镜像, 命令如下。

dlc get images --framework=pytorch --device=gpu --provider=Community

系统返回如下类似结果。

+	+	+	+++	
 Prov	ImageUrl ider Accelerator Framework	l	ImageUrlVpc	Image
+		 +	+++	
regi pc.cr pu regi pc.cr pu	stry.cn-beijing.aliyuncs.com/pai-dlc/ h-beijing.aliyuncs.com/pai-dlc/pytorc pytorch stry.cn-beijing.aliyuncs.com/pai-dlc/ h-beijing.aliyuncs.com/pai-dlc/pytorc pytorch	pytorch-trai h-training:1. pytorch-trai h-training:1.	ning:1.6.0-gpu-py37-cu101-ubuntu1 6.0-gpu-py37-cu101-ubuntu18.04 (ning:1.7.1-gpu-py37-cu110-ubuntu1 7.1-gpu-py37-cu110-ubuntu18.04 (8.04 registry-v Community g .8.04 registry-v Community g
	+	+	+	

提交任务时,您可以从上述返回结果的ImageUrl和ImageUrlVpc字段,获取节点的镜像信息。

查看代码源 (get code_source)

功能

获取代码源的信息。如果没有指定代码源的ID,则系统返回所有的代码源。如果指定了代码源,则返回这 个特定代码源的信息。

格式

dlc get code_source [yourCodeSourceId] [--sort_by=<yourSortField>] [--order=<yourOrder>] [--display_n ame=<yourCodeSourceName>] [--page_num=<yourPageNum>] [--page_size=<yourPageSize>]

参数

参数	是否必选	描述	类型
yourCodeSourceld	否	代码源的ID,默认值为空。	STRING
sort_by= <yoursortfield></yoursortfield>	否	用于排序的字段,默认值 为GmtCreateTime。	STRING
order= <yourorder></yourorder>	否	排序顺序。取值包括: • desc: 默认值,表示降序。 • asc: 表示升序。	STRING
display_name= <yourcodesource Name></yourcodesource 	否	代码源的显示名称,支持模糊匹配,默认值 为空。	STRING
page_num= <yourpagenum></yourpagenum>	否	取第几页的数据,默认值为1。	INT
page_size= <yourpagesize></yourpagesize>	否	分页大小,默认值为10。	INT

● 示例

• 获取所有的代码源信息

dlc get code_source

系统返回如下类似结果。

+		++	+		+	+	
+	+	+	+	+	+		
1	CodeSourceId	DisplayName D	escription	CodeRepo	CodeBra	anch Code	Co
mmi	t CodeRepoUser	Name CodeRepoAc	cessToken	UserId GmtCre	eateTime	GmtModify	уТi
me	Í						
+		++	+		+	+	
+	+	++	+	+	+		
coc	le-2021041120595	52-3lm541ppXXXX h	hh h	ttp://github.com/	m	aster	
1	232113015	457841879 2021-04	-11T12:59:53Z	2021-04-11T12:59:	53Z		
co c	le-2021041022434	12-z9fnegctXXXX ne	w_cs new te	est https://github	.com	maste	r
	23	2113015457841879	2021-04-10T14	l:43:43Z 2021-04-1	0T14:43:43Z		
coc	le-2021041022433	39-t0pp0drcXXXX ne	ew_cs new to	est http://github	.com	maste	r
	23	2113015457841879	2021-04-10T14	l:43:39Z 2021-04-1	0T14:43:39Z		
coc	le-2021041022432	29-fie1f5o8XXXX nev	w_cs new te	st https://github	.com/NVIDIA/	FasterTran	sfor
mer	master	2	3211301545784	1879 2021-04-10T	14:43:29Z 20	21-04-10T1	.4:4
3:29	Z						
+		++	+		+	+	
+	+	++	+	+	+		
						Total	
4							
+		++	+		+	+	
+	++	++	+	+	+		

• 获取特定的代码源信息

dlc get code_source code-20210410224329-fie1f5o8XXXX

系统返回如下类似结果。

++			+	
FIELD	VALUE		+	
CodeSourceId	code-2021041	0224329-fi	e1f5o8XXX	x
DisplayName	new_cs			
Description n	ew test			
CodeRepo h	ttps://github.c	om/NVIDIA	/FasterTra	ansformer
CodeBranch I	master			
CodeCommit				
CodeRepoUserNar	ne			
CodeRepoAccessT	oken			
MountPath /	root/code/			
UserId 2321	130154578418	79		
GmtCreateTime	2021-04-10T	14:43:29Z		
GmtModifyTime	2021-04-10T	14:43:29Z		
++			+	

查看数据源 (get data_source)

功能

获取数据源的信息。如果没有指定数据源的ID,则系统返回所有的数据源。如果指定了数据源,则返回这 个特定数据源的信息。

● 格式

dlc get data_source [yourDataSourceId] [--sort_by=<yourSortField>] [--order=<yourOrder>] [--display_na me=<yourCodeSourceName>] [--page_num=<yourPageNum>] [--page_size=<yourPageSize>] [--data_sour ce_type=<yourCodeSourceType>]

参数

参数	是否必选	描述	类型
yourDataSourceld	否	数据源的ID,默认值为空。	STRING
sort_by= <yoursortfield></yoursortfield>	否	用于排序的字段,默认值 为GmtCreateTime。	STRING
order= <yourorder></yourorder>	否	排序顺序。取值包括: • desc: 默认值,表示降序。 • asc: 表示升序。	STRING
display_name= <yourcodesource Name></yourcodesource 	否	代码源的显示名称,支持模糊匹配,默认值 为空。	STRING
page_num= <yourpagenum></yourpagenum>	否	取第几页的数据,默认值为1。	INT
page_size= <yourpagesize></yourpagesize>	否	分页大小,默认值为10。	INT
data_source_type = <yourcodesource Type></yourcodesource 	否	数据源的类型,默认值为nas。当前仅支持 NAS。	STRING

• 示例

• 获取所有的数据源信息

dlc get data_source

系统返回如下类似结果。

++++++	++	+
DataSourceType DataSourceId DisplayName rId GmtCreateTime GmtModifyTime	Description F	ileSystemId Use
++++++	+	+
+ nas data-20210410224621-gml01wz0XXXX new_test_ 2113015457841879 2021-04-10T14:46:21Z 2021-04-10T14:40	1 test once 6:21Z	16a0b4b17a 23
nas data-20210323171833-w2hslsl1XXXX PyTorch-Fac a0b4b17a 232113015457841879 2021-03-23T09:18:34Z 20	ce-Nas Nas for PyTorc 21-03-23T09:18:34Z	h Face Detection 16
nas data-20210323171710-ap5jirtcXXXX PyTorch-Face a0b4b17a 232113015457841879 2021-03-23T09:17:11Z 20	e-Nas Nas for PyTorch 21-03-23T09:17:11Z	Face Detection 16
++++++	++++	+
+ 	Total 3	I.
+++++	+	+

• 获取特定的数据源信息

dlc get data_source data-20210410224621-gml01wz0XXXX

系统返回如下类似结果。

++
FIELD VALUE
TT DataSourceType nas
DataSourceId data-20210410224621-gml01wz0XXXX
DisplayName new_test_1
Description test once
FileSystemId 16a0b4b17a
MountPath /root/data/
UserId 232113015457841879
GmtCreateTime 2021-04-10T14:46:21Z
GmtModifyTime 2021-04-10T14:46:21Z
1 1 1

3.4. 专有资源组相关

3.4.1. 使用PAI-DLC Dashboard管理任务

PAI-DLC在ACK集群里部署了PAI-DLC Dashboard, 您可以通过可视化的方式管理深度学习任务。

前提条件

准备专有资源组工作集群,详情请参见准备专有资源组工作集群。

背景信息

目前, PAI-DLC Dashboard仅支持管理官方公共镜像的TensorFlow类型任务, 管理其他类型的任务需要通过 Arena方式, 详情请参见Arena官方文档。

提交任务

- 1. 进入PAI-DLC Dashboard页面。
 - i. 登录PA控制台。
 - ii. 在左侧导航栏,选择模型开发和训练 > DLC-云原生深度学习训练平台。
 - iii. 在PAI-DLC管理控制台页面,单击操作下的集群控制台。
- 2. 在PAI-DLC Dashboard页面的左侧导航栏,单击提交任务。
- 3. 在提交任务页面,配置必选参数。

区域	参数	描述				
甘大仁白	任务名称		深度学习训练的任务名称。长度为2~30个字符,以小写字母开头。			
本 平旧忌	任务类型		系统默认 TensorFlow ,不支持修改。			
			根据代码存储位置进行配置: 如果代码存储在仓库,则选中代码仓库,并配置仓库地址和分 支。 			
	代码配置		② 说明 由于PAI-DLC会将代码下载至工作路径/workspace,所以您需要有代码仓库的访问权限。			
			○ 如果代码存储在ACK集群挂载的存储卷,则选中云存储挂载。同时,在选择存储卷声明列表中,选择ACK集群挂载的存储卷。			
			支持Python命令,可以将训练数据地址(例如 <i>data_dir</i>)作为参数, 传入代码入口函数。			
	执行命令		* 代码配置: ① 代码仓库 云存储挂载 * 仓库地址 ③: https://code.aliyun.co ensorflow-sample-code.gtl 			
任务信息			代码分支: git 分支, 不语写就认为 master * 执行命令: python tensor(low:-sample::code/tljob/docker/mnist/main.pymax_steps=10000data_dir=tensor(low:: sample:code/data/			
		实例数量	配置执行任务的实例数量: 如果执行单机任务,可以默认使用Worker。 如果执行分布式任务,可以单击Worker后的添加任务类型,选择PS。 			
			⑦ 说明 PS类型节点不能使用GPU。			

区域	Worker 参数		描述
		镜像	选择官方公共镜像,且 PS 和 Worker 的镜像需要保持一致 (TensorFlow版本和Python版本需要保持一致,CPU和GPU资源可 以不同)。
		资源	配置CPU(核数)、内存及GPU(卡数)。
			⑦ 说明 GPU(卡数)不能大于ACK集群的GPU卡数。

4. (可选)在提交任务页面,配置可选参数。

区域	描述
环境变量	以键值对方式配置的环境变量,可以作为参数在代码中引用。
; ; ;	如果将训练数据存储在ACK集群挂载的存储卷中,您可以将读取数据的逻辑写到代码 中。 为了提高代码灵活性,PAI-DLC Dashboard支持在页面绑定ACK集群挂载的存 储卷,将训练数据存储地址作为参数,传入任务执行的入口函数。
存储配置	⑦ 说明 ACK集群挂载的存储卷需要在 pai-dlc-user namspace 下, 否则存储配置列表无法显示该存储卷。

5. 单击页面下方的提交任务。

查询任务

PAI-DLC Dashboard支持按照名称、时间区间及状态查询任务。

- 1. 进入PAI-DLC Dashboard页面。
 - i. 登录PAI控制台。
 - ii. 在左侧导航栏,选择模型开发和训练 > DLC-云原生深度学习训练平台。
 - iii. 在PAI-DLC管理控制台页面,单击操作下的集群控制台。
- 2. 在PAI-DLC Dashboard页面的左侧导航栏,单击任务查询。
- 3. 在任务查询页面,选择时间区间并单击查询。
- 4. 在任务查询页面的任务列表区域,单击任务名称。

PAI-DLC Dashboard	≡							8
四 任务查询 止 是交任务	Home / 任务宣询 任务查询							
	名称: 词输入		Bţj	町区 间: 2020-04-1	4 → 2020-05-14 笸	状态: 全部		童肉 重量
	任务列表							I X C 🕸
	名称	命名空间	任务类型	状态	〒 创建时间	结束时间	执行时长	操作
	tf-r	default	TFJob	• 执行成功	2020-05-13 11:06:33	2020-05-13 11:30:27	23m54s	删除
	tf-r	default	TFJob	• 执行成功	2020-05-13 01:28:17	2020-05-13 01:28:20	3s	删除
	tf-rB	default	TFJob	• 执行成功	2020-05-13 00:46:31	2020-05-13 00:46:35	4s	删除
	tf-r	default	TFJob	• 执行成功	2020-05-13 00:42:26	2020-05-13 00:42:30	4s	删除
	tf-r	default	TFJob	• 执行失败	2020-05-13 00:40:09	2020-05-13 00:40:12	3s	删除

5. 在任务详情页面,查看任务详细信息。

3.4.2. 使用Arena管理任务(推荐)

PAI-DLC支持ACK集群通过Arena的方式进行模型训练,您需要安装Arena客户端并配置KubeConfig,才能提 交深度学习任务。

背景信息

Arena是基于Kubernetes环境的一个适配AI的命令行工具,您可以通过Arena管理深度学习任务。

步骤一:安装客户端

- 1. 登录Arena官网,下载对应系统的安装包(Mac系统下载arena-installer-xxx-xxx-darwinamd64.tar.gz,Linux系统下载arena-installer-xxx-xxx-linux-amd64.tar.gz)。
- 2. 执行如下命令, 安装客户端。

tar -xvf arena-installer-xxx-xxx.tar.gz cd arena-installer sudo ./install.sh

arena-installer-xxx-xxx.tar.gz需要替换为实际的安装包名称。

3. 执行如下命令, 查看客户端安装情况。

arena version

步骤二:配置KubeConfig

本地指定远程的ACK集群提交任务,需要在\$HOME/.kube/config中配置ACK集群的Config信息。

- 1. 登录PAI控制台。
- 2. 在左侧导航栏,选择模型开发和训练 > DLC-云原生深度学习训练平台。
- 3. 在PAI-DLC管理控制台页面,单击待配置的ACK集群ID/名称。
- 4. 在集群基本信息页面的集群资源区域,单击复制。



5. 在本地计算机创建.kube/config文件,并将复制的信息粘贴至该文件。

vim \$HOME/.kube/config

步骤三:提交模型训练任务

- 提交TensorFlow模型训练任务:
 - i. 执行以下任何一种命令, 提交TensorFlow任务。

```
# 方法一: 启动一个TF训练任务。
arena submit tfjob paraname
# 方法二: 启动一个TF训练任务。
arena submit tf paraname
```

需要将paraname替换为实际的参数,其中必须配置的参数如下(您可以执行 arena submit tfjob --hel p 命令,查看任务的所有paraname及其解释):

- --name: 任务名称。
- --image: PAI-DLC支持的镜像,可以使用该镜像启动Pod进行深度学习训练。您可以根据PAI-DLC 集群的地域、框架、Python版本及资源类型选择官方公共镜像,或指定自定义镜像,详情请参 见公共镜像列表。
- --data: 数据源目录,格式为PVC名称:挂载目录。
- ii. 通过以下任何一种方式, 查看任务日志:
 - 执行如下命令, 查看任务日志。

arena logs yourTaskName

其中yourTaskName需要替换为实际的任务名称。

- 在PAI-DLC Dashboard页面,查看任务日志,详情请参见使用PAI-DLC Dashboard管理任务。
- 提交PyTorch模型训练任务。

您可以通过如下命令,提交PyTorch任务。

```
arena submit pytorch \
```

```
--namespace=pai-dlc-system --name=ddptest \
```

```
--gpus=8 --workers=2 \
```

- --image=registry-vpc.cn-hangzhou.aliyuncs.com/pai-dlc/pai-pytorch-training:1.5-gpu-py3 \
- --data=pai-hangzhou-cpfs-pvc:/mnt/luci-cpfs/\
- --working-dir=/mnt/luci-cpfs/luci-hangzhou/yanhao/centernet/\

"bash experiments/ctdet_coco_ddp.sh"

其中:

- --namespace: PAI-DLC命名空间。
- o --name: 任务名称。
- --gpus: 每个Worker申请的GPU数量。
- 。 --workers: Worker总数量。
- --image: 镜像名称,建议使用registry-vpc.cn-hangzhou.aliyuncs.com/pai-dlc/pai-pytorchtraining:1.5-gpu-py3。
- --data: 数据源目录,格式为PVC名称:挂载目录。
- --working-dir: 程序执行的工作目录。
- experiments/ctdet_coco_ddp.sh: 指待执行的脚步, 需要根据实际情况修改。

步骤四:管理任务

• 您可以通过如下命令,查看作业运行情况。

arena list -n pai-dlc-system

• 您可以通过如下命令,查看日志。

arena logs -f ddptest -n pai-dlc-system

• 您可以通过如下命令,删除作业。

arena delete ddptest -n pai-dlc-system

示例

单机任务

arena submit tf \

--name=pai-deeplearning-test-oss \

```
--image=registry.cn-shanghai.aliyuncs.com/pai-dlc/pai-tensorflow-training:1.12-cpu-py2 \
```

--data=pai-deeplearning-oss:/training_dir/\

"python /training_dir/code/main.py --max_steps=10000 --data_dir=/training_dir/data/"

- --name: 任务名称。提交任务后,可以执行 arena logs \${name} 命令, 查看任务日志。
- --image: 镜像。例如: registry.cn-shanghai.aliyuncs.com/pai-dlc/pai-tensorflow-training:1.12-cpu-py2。
- --data:数据源目录。例如: *pai-deeplearning-oss:/training_dir*,其中pai-deeplearning-oss表示 ACK集群创建的PVC, */training_dir*表示将该PVC挂载到训练Pod的*/training_dir*目录。
- python /training_dir/code/main.py --max_steps=10000 --data_dir=/training_dir/data/: Pod执行 的命令。其中/*training_dir/code*/表示OSS中的代码目录, --max_steps和--data_dir表示main.py的参 数(在main.py中,通过FALGS.max_steps和FLAGS.data_dir获取)。
- 分布式任务

```
arena submit tf \
```

--name=pai-deeplearning-dist-test-nas \

--workers=2 \

--worker-image=registry.cn-shanghai.aliyuncs.com/pai-dlc/pai-tensorflow-training:1.12-cpu-py2 \ --ps=1 \

--ps-image=registry.cn-shanghai.aliyuncs.com/pai-dlc/pai-tensorflow-training:1.12-cpu-py2 \

--data=pai-deeplearning-nas:/training_dir/\

"python /training_dir/code/dist-main.py --max_steps=10000 --data_dir=/training_dir/data/"

○ --name:任务名称。提交任务后,可以执行 arena logs \${name} 命令,查看任务日志。

- --workers: Worker数量。
- --worker-image: Worker镜像。
- --ps: PS数量。
- --ps-image: PS镜像。
- --data:数据源目录。例如: *pai-deeplearning-nas:/training_dir/*,其中pai-deeplearning-nas表示
 ACK集群创建的PVC, /*training_dir*表示将该PVC挂载到训练Pod的/*training_dir*目录。
- python /training_dir/code/dist-main.py --max_steps=10000 --data_dir=/training_dir/data/: Pod 执行的命令。其中 /training_dir/code/dist-main.py表示NAS中的代码目录, --max_steps和--data_dir 表示main.py的参数(在main.py中,通过FALGS.max_steps和FLAGS.data_dir获取)。

● GPU分布式任务

```
arena submit tf \
```

```
--name=pai-deeplearning-gpu-dist-test-oss \
```

- --gpus=1 \
- --workers= $2 \$

--worker-image=registry.cn-shanghai.aliyuncs.com/pai-dlc/pai-tensorflow-training:1.12-gpu-py2 \ --ps=1 \

--ps-image=registry.cn-shanghai.aliyuncs.com/pai-dlc/pai-tensorflow-training:1.12-cpu-py2 \ --data=pai-deeplearning-nas:/training_dir/ \

"python /training_dir/code/dist-main.py --max_steps=10000 --data_dir=/training_dir/data/"

○ --name:任务名称。提交任务后,可以执行 arena logs \${name} 命令,查看任务日志。

- --gpus: 单个Worker的GPU卡数。该参数取值不能大于单个ACK GPU节点的最大卡数。
- --workers: Worker数量。
- --worker-image: Worker的GPU镜像。
- --ps: PS数量。
- --ps-image: PS的CPU镜像。
- --data:数据源目录。例如: *pai-deeplearning-oss:/training_dir/*,其中pai-deeplearning-oss表示
 ACK集群创建的PVC, /*training_dir*表示将该PVC挂载到训练Pod的/*training_dir*目录。
- python /training_dir/code/dist-main.py --max_steps=10000 --data_dir=/training_dir/data/: Pod 执行的命令。其中 /training_dir/code/dist-main.py表示OSS中的代码目录, --max_steps和--data_dir 表示main.py的参数(在main.py中,通过FALGS.max_steps和FLAGS.data_dir获取)。

3.4.3. ACK集群挂载NAS存储

PAI-DLC支持文件存储NAS作为数据源。本文以性能型NAS为例,为您介绍如何为ACK集群挂载NAS存储。

步骤一:选择存储类型

NAS提供了多种文件存储类型,适用于不同的场景。

参数	容量型	性能型	极速型	并行文件系统CPFS			
时延	3 ms~10 ms	1 ms~2 ms	百微秒	1 ms~2 ms			
IOPS	15 KB	50 KB	● 标准型: 10 KB● 高级型: 30 KB	100 MB			
吞吐	10 GB/s	20 GB/s	● 标准型: 150 MB/s以上 ● 高级型: 300 MB/s以上	1 TB/s			
容量	10 PB	1 PB	32 T B	100 PB			
负载模型	通用	通用	随机小IO和容器服务	并行读写			
适用场景	网站文件和备份日 志	制造制作和共享桌 面	高性能网站容器共享存储	高性能计算和智能 分析			

在深度学习训练场景下,可以选择并行文件系统CPFS存储类型。目前,ACK集群不支持通过管理控制台创建

CPFS类型的NAS关联,需要手动创建CPFS类型的NAS关联,详情请参见如何在ACK集群中使用CPFS存储卷服务。

步骤二: 创建NAS存储空间

- 1. 创建文件系统,详情请参见创建文件系统。
- 2. 添加挂载点, 详情请参见<mark>管理挂载点</mark>。添加挂载点时, 选择**挂载点类型**为**专有网络**, 选择**权限组**为VPC 默认权限组。VPC网络和交换机需要与ACK集群保持一致。
- 3. 挂载文件系统。
 - i. 登录ACK管理控制台。
 - ii. 在左侧导航栏,选择集群 > 节点。
 - iii. 在节点列表页面,单击实例ID。

	N号全部资源 * 🧬 全球					Q (#	薪文档、控制台、API	、解决方案和资源	费用 工	单 音楽	企业	支持	官同 ⊾	Q. A	(
容器服务 - Kubernetes +	节点列表											RUN	标签管理	集群扩展	
概范	⑦ 按量付费转包年包月 ⑦ 节点失联急 ○	カ & 节点監控及告書 & 6	集 Kubernetes 诊断信息	<i>€</i> 移除节点											
▼ 集群	集群 PAI-DeepLearning \$ 标签过	28 -													
\$ #	□ 实例ID/名称/IP地址	角色/状态	配置	容器组 (已分配量)	CPU 请求/限制	CPU 使用量	内存 请求/限制	内存使用量		Kubelet 版本			更新时间		
节点 存储卷 命名空间	- Herbasse	Worker ●运行中 可调度	按量付费 ecs.g6.xlarge	12	25.00% 180.00%	6.83 %	6.47% 88.23%	18.98 %	v	1.14.8-aliyun	1	20	19-12-26 19:4	ð:46	
授权管理 ▼ 应用	-	Worker ●运行中 可调度	按量付费 ecs.g6.xlarge	5	5.00% 27.50%	2.73 %	0.98% 6.83%	17.13 %	v	1.14.8-aliyun.	1	20	19-12-26 19:4	∂:48	
无状态 有状态		Worker ●运行中 可调度	按量付费 ecs.g6.xlarge	6	5.00% 27.50%	3.05 %	0.98% 6.83%	86.51 %	v	1.14.8-aliyun.	1	20	19-12-26 19:4	ð:49	
守护进程集 任务	1 批量移除 节点维护	节点上线										共有3条,	每页显示: 2	5 0条 。	

iv. 在ECS管理控制台的实例详情页面,单击基本信息后的远程连接,即可通过Shell方式连接ECS。

<	o worker-
实例详情	
本实例磁盘	基本信息 远程连接 更多•
本实例快照	ID:
本实例弹性网卡	所在可用区: 华北 3 可用区 A
本实例安全组	名称:
本实例安全防护	描述: ESS
	地域: 华北3 (张家口)
	实例规格: ecs.g6.xlarge
	实例规格族: ecs.g6
	镜像ID:
	密钥对名称:
	RAM角色:

v. 登录NAS控制台, 在文件系统列表页面, 单击文件系统ID。
vi. 在挂载使用页面, 单击挂载命令下的 图标, 复制挂载命令。

(-) 阿里云					Q 提家文档、控制台、API、解决方面和资源	翡用 工单 铅索	企业 支持	ER E Q' 7	7 ⑦ 简体
IAS文件系统 / 文件系统	/								
0444646	50								
基本信息	挂载点								
挂载使用	满加挂载点								c
性能监控	挂载机英型	VPC	交换机	挂载点	挂歉命令	权限组	状态	操作	
	专有网络			ŵ	●勝字思知上的今週以下55日的公里記, 阿爾曼信包紙, 郑県 知知日用意能文件記, 也加爾爾伊用多姓 CS 同时编码一个文 件, 國際用 N544 物公理記, ●	VPC數认权限组 (全部允许)	✓ 可用	修改収積組(製用)日	注影響户演 影除
	挂载文件系统								
	挂戴文件系统到ECS	配置自动挂戴文件系统							

vii. 在Shell环境,执行**挂载命令**,详情请参见Linux系统挂载NFS文件系统。

viii. 挂载完成后,执行 df-h 命令,查看挂载的文件系统。

lesystem	Size	Used	Avail	Use% Mounte	d on	
ev/vdal	118G	28G	_86G	25% /		
rtmprs	7.76	ø	7.76	0% /dev		
0TS	1.76	2 04	1.76	U% /dev		
915 sfe	7 76	2.00	7 76	0% / FUI		
ofs	7.76	126	7.76	1% /vai		
ofs	7.76	126	7.76	1% /vai	white	
ofs	7.7G	12K	7.7G	1% /vai	-988g1	
ofs	7.7G		7.7G	1% /vai	-controller-token-	rtv5m
ofs	7.7G	12K	7.7G	1% /vai	wbfp	
ofs	7.7G	12K	7.7G	1% /vai	-s87hq	
ofs	7.76	12K	7.76	1% /vai	wbfp	
015	7.70	12K	7.70	1% /vai	-controller-token-	Ken E.
	1196	286	260	255 //21	edentiat-hetper-to	Ken-3w
i tay	64M	200	64M	8% /vai		
rlav	118G	28G	86G	25% /vai	DIST IN 12 WAT TAYS DUTT AT ANY 14 YO REPORT OF A DATE O	
	64M	0	64M	0% /vai	CORRECT AND COMPANY AND ADDRESS OF COMPANY AND ADDRESS OF COMPANY ADDRESS OF COMP	
ofs	7.7G		7.7G	1% /var	-hq66k	
ofs	7.7G	12K	7.7G	1% /vai	-hq66k	
ofs	7.7G	12K	7.7G	1% /var	ken-62wn4	
ertay	1186	286	86G	25% /vai		
a clav	1196	296	04M	25% /Val	and the set of the set	
r Lay	1186	286	866	25% /vai		
rlav	118G	28G	86G	25% /vai		
erlay	118G	28G	86G	25% /vai	CONTRACTOR OF CONTRACTOR CON	
	64M		64M	0% /vai	CONTRACTOR OF A DESCRIPTION OF A DESCRIP	
erlay	118G	28G	86G	25% /vai	Provide the first Proceeding of the Carlow State Sta	
•	64M	. 0	64M	0% /vai	CONTRACTOR AND A CONTRACT	
erlay	118G	28G	86G	25% /vai		
irlay	1186	286	866	25% /Vai		
	64M	å	64M	05 /vai		
rlav	1186	28G	866	25% /vai		
arlav	118G	28G	86G	25% /vai	CONTRACTOR CONTRACTOR CONTRACTOR CONTRACTOR	
	64M		64M	0% /vai	Control of the Control and an approximation of the Control and the Control of the	
erlay	118G	28G	86G	25% /vai	and the set of the set	
erlay	118G	28G	86G	25% /vai	CONTRACTOR AND A CONTRACTOR AND A DESCRIPTION OF A DESCRIPT	
	64M		64M	0% /vai	NUMBER OF STATES OF THE STATES OF	
riay	1186	286	866	25% /Vai		
	04M		64M	0% /Val		
	64M	ă	64M	0% /vai		
rlav	1186	28G	866	25% /vai		
rlav	118G	28G	86G	25% /vai	D III I C AND THE REPORT OF TH	
rlaý	118G	28G	86G	25% /vai	Control of a cold warding warding warding the control of the contr	
rlay	118G	28G	86G	25% /vai		
arlay	118G	28G	86G	25% /vai	A DESCRIPTION OF A DESC	
riay	118G	28G	86G	25% /vai	A DESCRIPTION OF A	
ir lay	118G	286	866	25% /vai		
arlav	1186	280	866	25% /vai		
	1.00					

步骤三: ACK集群挂载NAS

ACK集群支持以下方式挂载NAS存储:

- 静态NAS挂载存储
 - 通过Volume方式挂载
 - 通过PV/PVC方式挂载(PAI-DLC推荐)
- 动态NAS挂载存储

以PV/PVC方式为例,介绍挂载NAS存储的流程。其他的挂载方式详情请参见NAS存储卷使用说明。

- 1. 登录ACK管理控制台。
- 2. 在左侧导航栏,选择集群 > 存储卷。
- 3. 创建PV。

i. 在存储与存储说明页面,选择所需的集群并单击创建。



ii. 输入**数据卷名**,选择**挂载点域名为选择挂载点**。如果是容量型或性能型的NAS,可以直接从文件存储列表中选择挂载点。如果需要挂载NAS子目录,则单击**显示高级选项**,输入**子目录**。

创建存储着	3		\times
;	存储卷类型	○ 云盘 ● NAS ○ OSS	
	* 数据卷名	名称必须以小写字母开头,只能包含小写字母、数字、小数 点(.)和中划线(-)	
	存储驱动	 Flexvolume ● CSI 集群已部署 csi-plugin 	
	* 总量	20Gi	
	访问模式	ReadWriteMany ReadWriteOnce	
*:	挂载点域名	 选择挂载点 自定义 请选择 	
		▶ 显示高级选项	
	标签	● 添加标签	
		创建 取	消

iii. 单击创建。

4. 创建PVC。

i. 在存储与存储说明页面,单击存储说明页签。

ii. 选择所需的集群并单击创建。

iii. 单击已有存储卷后的选择已有存储卷,并根据提示选择已创建的PV。

创建存储声明	×
-+- XE	
木店	👻 使用已有存储者 🤍 使用存储类
存储声明类型	◎ 云盘 ● NAS ◎ OSS
名称	
	名称必须以小写字母开头,只能包含小写字母、数字、小数 点(.)和中划线(-)
分配模式	 已有存储卷
已有存储卷	选择已有存储卷
总量	20 Gi 🔻
	创建 取消

iv. 单击创建。

3.4.4. ACK集群挂载OSS存储

PAI-DLC支持OSS作为数据源。本文为您介绍如何为ACK集群挂载OSS存储。

前提条件

您已经创建OSS静态存储卷,请参见创建存储空间。

⑦ 说明 相同Region的OSS Bucket可以使用内网地址进行挂载。

创建PV

- 1. 登录容器服务管理控制台。
- 2. 在控制台左侧导航栏中,单击集群。
- 3. 在集群列表页面中,单击目标集群名称或者目标集群右侧操作列下的详情。
- 4. 在集群管理页左侧导航栏中,选择存储 > 存储卷。
- 5. 在存储卷页面单击右上角的创建。
- 6. 在创建存储卷对话框中配置参数。

参数	描述
存储卷类型	支持云盘/NAS/OSS三种云存储类型。本文中选择为OSS。
名称	创建的数据卷的名称。数据卷名在集群内必须唯一。本例为pv-oss。
存储驱动	支持Flexvolume和CSI。本文中选择为 CSI。

参数	描述
总量	所创建存储卷的容量。
访问模式	默认为ReadWriteMany。
访问证书	选择访问OSS所需的保密字典。 • 选择已有保密字典:选择命名空间及保密字典。 • 新建保密字典:配置命名空间、名称、AccessKey ID及AccessKey Secret。
可选参数	您可以为OSS存储卷输入定制化参数,格式为 -o *** -o *** 。
Bucket ID	您要使用的OSS bucket的名称。单击 选择 Bucket <i>,</i> 在弹出的对话框中选 择所需的bucket并单击 选择 。
访问域名	您可以参考以下情况选择需要的访问域名。 • 如果Bucket和ECS实例位于不同地域(Region),请选择 外网域名 。 • 如果Bucket和ECS实例位于相同地域,需要根据集群网络类型进行选择: ■ 若是VPC网络,请选择 VPC域名 。 ■ 若是经典网络,请选择 内网域名 。
标签	为该存储卷添加标签。

7. 参数配置完成后,单击创建。

创建PVC

- 1. 登录容器服务管理控制台。
- 2. 在控制台左侧导航栏中,单击集群。
- 3. 在集群列表页面中,单击目标集群名称或者目标集群右侧操作列下的详情。
- 4. 在集群管理页左侧导航栏中,选择存储>存储声明。
- 5. 在存储声明页面,单击右上角的创建。
- 6. 在创建存储声明页面中,填写界面参数。

参数	描述
存储声明类型	支持云盘、NAS、OSS三种云存储类型。本文中选择OSS。
名称	创建的数据卷的名称,数据卷名在集群内必须唯一。
	选择已有存储卷。
分配模式	⑦ 说明 若未创建存储卷,您可以设置分配模式为创建存储卷,配置创建 存储卷参数。具体操作,请参见步骤一:创建PV。

参数	描述
已有存储卷	单击 选择已有存储类 ,在目标存储卷右侧操作列单击 选择 ,选择存储卷。
	所创建存储卷的容量。
总量	⑦ 说明 所创建的存储卷容量不能超过磁盘容量。

7. 单击创建。创建成功后可以在列表中看到csi-oss-pvc,并且已绑定相应的存储卷。

3.4.5. ACK集群挂载CPFS存储

PAI-DLC支持CPFS作为数据源。本文为您介绍如何为ACK集群挂载CPFS存储。

背景信息

ACK集群支持通过PV/PVC的方式挂载CPFS存储。由于ACK管理控制台不支持直接创建CPFS PV/PVC,因此您需要通过YAML的方式手动创建。

操作步骤

1. 执行以下命令, 查看flexvolume-cpfs存储插件的部署情况。

\$ kubectl get pods -n kube-system	n gre	p flexvolume-o	pfs
flexvolume-cpfs-dcldf	1/1	Running 0	98m
flexvolume-cpfs-dkrt8	1/1	Running 0	98m

如果未安装flexvolume-cpfs,需要先安装此插件,详情请参见如何在ACK集群中使用CPFS存储卷服务。

- 2. 创建CPFS PV。
 - i. 创建pai-deeplearing-cpfs-pv.yaml文件,并拷贝以下模板至文件中。

```
apiVersion: v1
kind: PersistentVolume
metadata:
name: pai-deeplearning-cpfs
labels:
alicloud-pvname: pai-deeplearning-cpfs
spec:
capacity:
storage: 1000Gi
accessModes:
- ReadWriteMany
flexVolume:
driver: "alicloud/cpfs"
options:
 server: "cpfs-****-0th1.cn-shenzhen.cpfs.nas.aliyuncs.com@tcp:cpfs-****-mdm1.cn-shenzhen.cp
fs.nas.aliyuncs.com@tcp"
 fileSystem: "****"
 subPath: "/"
 options: "rw"
```

ii. 登录NAS控制台,在文件系统列表页面,单击文件系统ID。

NAS文件系统	NAS文件系统 / 文件系统列表										
极克	文件系统列表										
文件系统 ^	③ 请确保您的ECS内核版本符合要求。	并且挂载时使用了nores	wport參数,以规避NFS文	件系统卡住的风险。							
文件系统列表				-							
权限组	创建文件系统 文件系统类型: 全	部 > 文件系统	D 🗸 请输入	Q							
存储包	文件系统 ID/名称	文件系统类型	存储规格	使用量/总容量	可用区	计费方式	创建时间 4	状态	协议类型	挂载点数量	操作
数据服务 へ	+ cpts-	CPFS	标准型	515.00 MB / 2.00 TB	华南1可用区 A	按量付费	2020年4月22日 14:06:31	✓ 适行中	cpfs	1	监控 管理 更多
快照									每页显示	10 ~ 共19	・ く 上一页 1 下
文件迁移 🖸											

iii. 在基本信息页面的挂载点后,查看server和fileSystem。

NAS文件系统 / 文件系统 / c;	ofs-			
← cpfs-	Out .			
基本信息	基础信息			
挂载使用	文件系统 ID	cpts-	文件系统类型	CPFS
10/04000	文件系统名称	cpfs-	带宽	1 GB/s
121633590	最大容量	2.00 TB	使用量	515.00 MB
性能监控	任赎存储用量	0 MB	挂载点	cpts-t n.cps shenzhou wa
	网络类型	专有网络	专有网络	vpc-wz9fmse2vyafifc0hi4z3
	交换机	and the second se	是否加密	不加密
	KMSKeyld	*	地域	华南1 (深圳)
	可用区	华南 1 可用区 A	存储规格	标准型
	购买类型	按量付费	协议类型	cpfs
	状态	✓ 运行中	创建时间	2020年4月22日 14:06:31
	劉期时间		是否支持生命周期管理	不支持

- iv. 在 pai-deeplearing-cpfs-pv. yaml 文件中,更新server和 file System 的取值。
- v. 执行如下命令, 创建PV。

kubectl create -f pai-deeplearing-cpfs-pv.yaml

- 3. 创建CPFS PVC。
 - i. 创建pai-deeplearing-cpfs-pvc文件,并拷贝以下模板至文件中。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
name: pai-deeplearning-cpfs
spec:
accessModes:
- ReadWriteMany
resources:
requests:
storage: 1000Gi
selector:
matchLabels:
alicloud-pvname: pai-deeplearning-cpfs
```

② 说明 alicloud-pvname需要与*pai-deeplearing-cpfs-pv.yaml*文件中的alicloud-pvname保持一致。

ii. 执行如下命令, 创建PVC。

kubectl create -f pai-deeplearing-cpfs-pvc

4. 执行以下命令,检查PV/PVC的创建情况。

\$ kubectl get pv | grep pai-deeplearning-cpfs
pai-deeplearning-cpfs 1000Gi RWX Retain Bound default/pai-deeplearning-cpfs
152m
\$ kubectl get pvc | grep pai-deeplearning-cpfs
pai-deeplearning-cpfs Bound pai-deeplearning-cpfs 1000Gi RWX

5. 使用同一地域的ECS服务器, 挂载CPFS并上传数据, 详情请参见挂载文件系统。

3.4.6. 开启公网Ingress服务

PAI-DLC支持添加已有的ACK集群或新建的ACK集群,如果您需要PAI-DLC完整的功能(PAI-DLC Dashboard),则需要开启ACK集群的公网Ingress服务。

背景信息

如果没有ACK集群,需要新建ACK集群,详情请参见创建Kubernetes专有版集群。

查看是否开启公网Ingress服务

- 1. 进入集群详情页面。
 - i. 登录ACK集群管理控制台。
 - ii. 在集群列表, 单击集群名称。
- 2. 在集群基本信息页面的集群资源区域,查看Nginx Ingress SLB的配置情况。如果已配置链接,即开启了Ingress服务。
- 3. 确认**地址类型**为**公网**。
 - i. 在集群基本信息页面的集群资源区域,单击Nginx Ingress SLB后的链接。
 - ii. 在负载均衡SLB页面,选择实例 > 实例管理。
 - iii. 在**实例详情**页签的基本信息区域,确认地址类型为公网。

为新建的ACK集群开启公网Ingress服务

购买ACK集群时,在Ingress区域,选中安装Ingress组件,并选择负载均衡类型为公网,详情请参见创建 Kubernetes专有版集群。

Ingress	✓ 安装 Ingress 组件			
	负载均衡类型	公网		内网
	创建 Ingress Dasl	nboard		
存储插件	Flexvolume		CSI	
	如何选择 Kubernetes	集群的存储插作	ŧ	
テッ方标件	在 FCS 节点上安装	大学学校 化	与神光	
스페)エ)비(구	L COO PARLA		2 1世1手	
ᄶᄪᅚᆁᅚ	在节点上安装云监控指	GALLER 1997 1997 1997 1997 1997 1997 1997 199	2 JE17 至控控制台查和	f所创建ECS实例的监控信息
日志服务	在节点上安装云监控指 使用日志服务 🛆	在二面立面中 N 時, 可以在云幽 注意	2 推1 9 在控控制台查希	言所创建ECS实例的监控信息
日志服务	在节点上安装云监控指 使用日志服务 A 不开启日志服务时, 非	在二面过300円 100 許件,可以在云出 注意 行无法使用集群電	新拉拉制台宣和 新拉拉制台宣和	f所创建ECS实例的监控信息

为已有的ACK集群开启公网Ingress服务

- 1. 进入集群详情页面。
 - i. 登录ACK集群管理控制台。
 - ii. 在集群列表,单击集群名称。
- 2. 在左侧菜单栏,单击组件管理。
- 3. 在系统组件区域,单击Nginx Ingress Controller后的安装。

<	集群:						移引象行						
基本信息	提示:请务必通过 组件管理-升级功能 升级您的 Addon 组件,如采用其他方式升级可能会出现不可预知的问题,敬请知晓。												
节点列表	系统组件	组件	当前版本	可升级版本	状态		操作						
组件管理													
事件列表		alicloud-application-controller	v0.1.0.1-f832bed-aliyun	v0.1.0.1-f832bed-aliyun		无需升级 查看	YAML						
集群升级		alicloud-monitor-controller	v1.3.0	v1.3.0		无需升级 查看	YAML						
运行时升级		aliyun-acr-credential-helper	v20.03.16.0-36d5d7e-aliyun	v20.03.16.0-36d5d7e-aliyun		无需升级 查看	YAML						
集群审计 集群拓扑		Cloud Controller Manager ② 组件介绍 ③ 版本信息	v1.9.3.240-g40d97e1-aliyun	v1.9.3.240-g40d97e1-aliyun		无	需升级						
=		metrics-server	v0.2.1-bd3ae4f-aliyun	v0.2.1-bd3ae4f-aliyun		无需升级 查看	YAML						
-		Nginx Ingress Controller ② 组件介绍 ③ 版本信息		v0.30.0.1-5f89cb606-aliyun		C	安装						
		存储											
		alicloud-disk-controller		v1.14.8.44-c23b62c5-aliyun									
		csi-plugin <i>S</i> 组件介绍 <i>S</i> 版本信息	v1.14.8.39-0d749258-aliyun	v1.14.8.39-0d749258-aliyun		无需升级 查看	İYAML						
		csi-provisioner の 組件介绍 の 版本信息	v1.14.8.39-0d749258-aliyun	v1.14.8.39-0d749258-aliyun		无需升级 查看	YAML						

4. 在集群详情页面的集群信息区域,单击测试域名后的重新绑定域名。

PAI-DLC云原生深度学习训练平台·<mark>开</mark> 始使用

<	2 集群: 1 4													
基本信息		基本信息												
节点列表	Ľ	集群 ID: 虚拟专		有网络 ●运行中 地域:华北		地域: 华北2 (北京)	删除保护:							
组件管理														
事件列表	集群信息													
集群升级	API Server 公网连接端点 API Server 内网连接端点			绑定 EIP										
运行时升级				https://										
集群审计		Pod 网络 CIDR		10,06,0419										
集群拓扑	-	Service CIDR		10.01.0400										
		测试域名		• 重新研定域名										
		kube-proxy 代理模式		ipvs										
		节点 IP 数量		64										
		网络插件	Flannel											

4.最佳实践

本文为您介绍PAI-DLC的最佳实践。

- 快速提交单机PyTorch迁移学习任务
- 使用NAS提交单机PyTorch迁移学习任务
- 使用paiio读写MaxCompute表数据