

**ModelArts**

# 用户指南（ Lite Cluster ）

文档版本

01

发布日期

2026-01-08



**版权所有 © 华为云计算技术有限公司 2026。保留一切权利。**

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

# 目 录

<b>1 Lite Cluster 使用前必读.....</b>	<b>1</b>
1.1 Lite Cluster 使用流程.....	1
1.2 Lite Cluster 高危操作一览表.....	3
1.3 不同机型对应的软件配套版本.....	4
<b>2 Lite Cluster 资源开通.....</b>	<b>12</b>
<b>3 Lite Cluster 资源配置.....</b>	<b>26</b>
3.1 Lite Cluster 资源配置流程.....	26
3.2 配置 Lite Cluster 网络.....	37
3.3 配置 kubectl 工具.....	41
3.4 配置 Lite Cluster 存储.....	45
3.5 (可选) 配置驱动.....	46
3.6 (可选) 配置镜像预热.....	48
<b>4 Lite Cluster 资源使用.....</b>	<b>53</b>
4.1 在 Lite Cluster 资源池上使用 Snt9B 完成分布式训练任务.....	53
4.2 在 Lite Cluster 资源池上使用 ranktable 路由规划完成 PyTorch NPU 分布式训练.....	60
4.3 在 Lite Cluster 资源池上使用 Snt9B 完成推理任务.....	65
4.4 在 Lite Cluster 资源池上使用 Ascend FaultDiag 工具完成日志诊断.....	68
4.5 在 Lite Cluster 挂载 SFS Turbo.....	71
4.6 在 Lite Cluster 资源池设置并启用高可用冗余节点.....	81
4.7 在 Lite Cluster 跨区域访问其他服务.....	84
<b>5 Lite Cluster 资源管理.....</b>	<b>91</b>
5.1 Lite Cluster 资源管理介绍.....	91
5.2 管理 Lite Cluster 资源池.....	92
5.3 管理 Lite Cluster 节点池.....	94
5.4 管理 Lite Cluster 节点.....	99
5.5 扩缩容 Lite Cluster 资源池.....	106
5.6 升级 Lite Cluster 资源池驱动.....	110
5.7 升级 Lite Cluster 资源池单个节点驱动.....	113
5.8 监控 Lite Cluster 资源.....	114
5.8.1 使用 AOM 查看 Lite Cluster 监控指标.....	114
5.8.2 使用 Prometheus 查看 Lite Cluster 监控指标.....	151
5.9 释放 Lite Cluster 资源.....	155

<b>6 Lite Cluster 插件管理.....</b>	<b>156</b>
6.1 Lite Cluster 插件概述.....	156
6.2 节点故障检测(ModelArts Node Agent).....	161
6.3 指标监控插件(ModelArts Metrics Collector).....	162
6.4 AI 套件(ModelArts Device Plugin).....	164
6.5 Volcano 调度器.....	165
6.6 集群弹性引擎.....	167

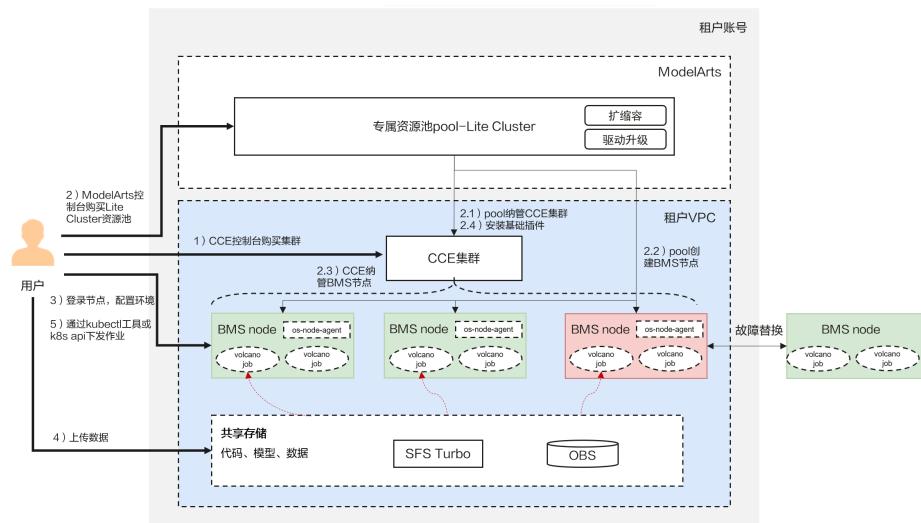
# 1

## Lite Cluster 使用前必读

### 1.1 Lite Cluster 使用流程

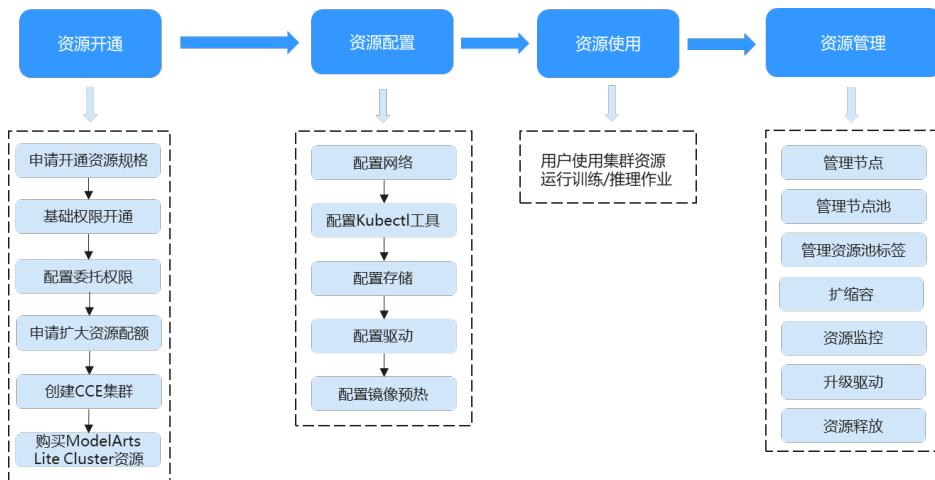
ModelArts Lite Cluster面向k8s资源型用户，提供托管式k8s集群，并预装主流AI开发插件以及自研的加速插件，以云原生方式直接向用户提供AI Native的资源、任务等能力，用户可以直接操作资源池中的节点和k8s集群。本文旨在帮助您了解Lite Cluster的基本使用流程，帮助您快速上手。

图 1-1 资源池架构图



如图所示为Lite Cluster架构图。Lite Cluster基于CCE服务实现对资源节点的管理，因此，用户首先需要购买一个CCE集群。在ModelArts控制台购买Lite Cluster集群时，ModelArts的资源池会先纳管这个CCE集群，然后根据用户设置的规格创建相应的计算节点（BMS/ECS）。随后，CCE会对这些节点进行纳管，并且ModelArts会在CCE集群中安装npuDriver、os-node-agent等插件。完成Cluster资源池的购买后，您即可对资源进行配置，并将数据上传至存储云服务中。当您需要使用集群资源时，可以使用kubectl工具或k8s API来下发作业。此外，ModelArts还提供了扩缩容、驱动升级等功能，方便您对集群资源进行管理。

图 1-2 使用流程



推荐您根据以下使用流程对Lite Cluster进行使用。

1. 资源开通：您需要开通资源后才可使用Lite Cluster，在开通资源前，请确保完成所有相关准备工作，包括申请开通所需的规格和进行权限配置。随后，在ModelArts控制台上购买Lite Cluster资源。请参考[2 Lite Cluster资源开通](#)。
2. 资源配置：完成资源购买后，需要对网络、存储、驱动进行相关配置。请参考[3 Lite Cluster资源配置](#)。
3. 资源使用：完成资源配置后，您可以使用集群资源运行训练和推理训练，具体案例可参考[4 Lite Cluster资源使用](#)。
4. 资源管理：Lite Cluster提供扩缩容、驱动升级等管理手段，您可在ModelArts控制台上对资源进行管理。请参考[5 Lite Cluster资源管理](#)。

表 1-1 相关名词解释

名词	含义
容器	容器技术起源于Linux，是一种内核虚拟化技术，提供轻量级的虚拟化，以便隔离进程和资源。尽管容器技术已经出现很久，却是随着Docker的出现而变得广为人知。Docker是第一个使容器能在不同机器之间移植的系统。它不仅简化了打包应用的流程，也简化了打包应用的库和依赖，甚至整个操作系统的文件系统能被打包成一个简单的可移植的包，这个包可以被用来在任何其他运行Docker的机器上使用。
Kubernetes	Kubernetes是一个开源的容器编排部署管理平台，用于管理云平台中多个主机上的容器化应用。Kubernetes的目标是让部署容器化的应用简单并且高效，Kubernetes提供了应用部署、规划、更新、维护的一种机制。使用Lite Cluster需要用户具备一定的Kubernetes知识背景，您可参考 <a href="#">Kubernetes基础知识</a> 。
CCE	云容器引擎（Cloud Container Engine，简称CCE）是一个企业级的Kubernetes集群托管服务，支持容器化应用的全生命周期管理，为您提供高度可扩展的、高性能的云原生应用部署和管理方案。

名词	含义
BMS	裸金属服务器（Bare Metal Server）是一款兼具虚拟机弹性和物理机性能的计算类服务，为您和您的企业提供专属的云上物理服务器，为核心数据库、关键应用系统、高性能计算、大数据等业务提供卓越的计算性能以及数据安全。
ECS	弹性云服务器（Elastic Cloud Server）是一种可随时自助获取、可弹性伸缩的云服务器，可帮助您打造可靠、安全、灵活、高效的应用环境，确保服务持久稳定运行，提升运维效率。
os-node-agent	ModelArts Lite k8s Cluster节点默认会安装os-node-agent插件，用于对节点进行管理，例如： <ul style="list-style-type: none"><li>驱动升级：通过os-node-agent插件下载驱动文件并进行驱动版本升级、回退。</li><li>故障检测：通过os-node-agent插件在系统内周期性巡检故障特征，及时发现节点故障。</li><li>指标采集：通过os-node-agent插件采集GPU/NPU利用率指标等重要的观测数据，上报到租户侧AOM。</li><li>节点运维：授权后，通过os-node-agent插件执行诊断脚本，进行故障定位定界。</li></ul>

## 1.2 Lite Cluster 高危操作一览表

当您在CCE、ECS或BMS服务控制台直接操作ModelArts Lite Cluster资源时，可能会导致资源池部分功能异常。下表可帮助您定位异常出现的原因，风险操作包括但不限于以下内容。

高危操作风险等级说明：

- 高：对于可能直接导致业务失败、数据丢失、系统不能维护、系统资源耗尽的高危操作。
- 中：对于可能导致安全风险及可靠性降低的高危操作。
- 低：高、中风险等级外的其他高危操作。

表 1-2 操作及其对应风险

操作对象	操作名称	风险描述	风险等级	应对措施
集群	升级、修改、休眠集群、删除集群等。	可能影响ModelArts侧基本功能，包括但不限于资源池管理、节点管理、扩缩容、驱动升级等。	高	不可恢复。
节点	退订、移除、关机、污点管理、切换/重装操作系统等。	可能影响ModelArts侧基本功能，包括但不限于节点管理、扩缩容、驱动升级、带本地盘机型的本地盘数据丢失等。	高	不可恢复。

操作对象	操作名称	风险描述	风险等级	应对措施
	修改网络安全组	可能影响ModelArts侧基本功能，包括但不限于节点管理、扩缩容、驱动升级等。	中	改回原有内容。
网络	修改/删除集群关联网段。	影响ModelArts侧基本功能，包括但不限于节点管理、扩缩容、驱动升级等。	高	不可恢复。
插件	升级、卸载GPU-beta插件。	可能导致GPU驱动使用异常。	中	回退版本、重装插件。
	升级、卸载huawei-npu插件。	可能导致NPU驱动使用异常。	中	回退版本、重装插件。
	升级、卸载volcano插件。	可能导致作业调度异常。	中	回退版本、重装插件。
	卸载ICAgent插件。	可能导致日志、监控功能异常。	中	回退版本、重装插件。
Helm	升级、回退、卸载os-node-agent。	导致驱动升级、故障检测、指标采集、节点运维功能异常。	高	联系华为云技术支持重装os-node-agent。
	升级、回退、卸载rdma-sriov-dev-plugin。	可能影响容器内使用RDMA网卡。	高	联系华为云技术支持重装rdma-sriov-dev-plugin。

## 1.3 不同机型对应的软件配套版本

由于弹性集群资源池可选择弹性裸金属或弹性云服务器作为节点资源，不同机型的节点对应的操作系统、适用的CCE集群版本等不相同，为了便于您制作镜像、升级软件等操作，本文对不同机型对应的软件配套版本做了详细介绍。

### CCE 集群维护策略说明

ModelArts Lite Cluster使用的CCE集群归属于用户，用户拥有对CCE集群的完全控制权。

- 如果您的Lite Cluster使用了EOS的CCE集群，应遵照CCE发布的生命周期策略，尽快升级到ModelArts推荐的CCE版本。  
关于如何升级CCE集群，请参见CCE[集群升级](#)指导。  
关于CCE集群版本策略，请参见CCE[集群版本公告](#)。
- 如果您在Lite Cluster场景遇到CCE集群相关的技术问题，请通过提交工单联系CCE技术支持进行问题的排查和解决。

## 裸金属服务器对应的软件配套版本

表 1-3 裸金属服务器

类型	卡类型	RDMA网络协议	操作系统	适用范围、约束	依赖插件
NP U	ascend-snt9b	RoCE	<ul style="list-style-type: none"><li>操作系统：EulerOS 2.10 64bit（推荐）</li><li>内核版本：4.19.90-vhulk2211.3.0.h 1543.eulerosv2r 10.aarch64</li><li>架构类型：aarch64</li></ul>	<ul style="list-style-type: none"><li>集群类型：CCE Standard</li><li>集群版本：v1.23（v1.23.5-r0及以上版本）  v1.25 v1.28 v1.31（推荐）</li><li>集群规模：50 200 1000 2000</li><li>集群网络模式：容器隧道网络 VPC</li><li>集群转发模式：iptables ipvs</li></ul>	<ul style="list-style-type: none"><li>huawei-npu</li><li>volcano</li></ul> 插件版本匹配关系请见 <a href="#">表1-5</a> 。
		RoCE	<ul style="list-style-type: none"><li>操作系统：Huawei Cloud EulerOS 2.0 64bit</li><li>内核版本：5.10.0-60.18.0.5 0.r865_35.hce2. aarch64</li><li>架构类型：aarch64</li></ul>	<ul style="list-style-type: none"><li>集群类型：CCE Turbo</li><li>集群版本：v1.23 v1.25 v1.28 v1.31（推荐）</li><li>集群规模：50 200 1000 2000</li><li>集群网络模式：ENI</li><li>集群转发模式：iptables ipvs</li></ul>	

类型	卡类型	RDMA网络协议	操作系统	适用范围、约束	依赖插件
	ascend-snt9	RoCE	<ul style="list-style-type: none"> <li>操作系统: EulerOS 2.8 64bit</li> <li>内核版本: 4.19.36-vhulk1907.1.0.h 619.eulerosv2r8.aarch64</li> <li>架构类型: aarch64</li> </ul>	<ul style="list-style-type: none"> <li>集群类型: CCE Standard Turbo</li> <li>集群版本: v1.23 ( v1.23.5-r0及以上版本 )   v1.25 v1.28 ( 推荐 )</li> <li>集群规模: 50 200 1000 2000</li> <li>集群网络模式: 容器隧道网络 VPC ENI</li> <li>集群转发模式: iptables ipvs</li> </ul>	
GPU	gpu-ant8	RoCE	<ul style="list-style-type: none"> <li>操作系统: EulerOS 2.10 64bit</li> <li>内核版本: 4.18.0-147.5.2.1 5.h1109.euleros v2r10.x86_64</li> <li>架构类型: x86</li> </ul>	<ul style="list-style-type: none"> <li>集群类型: CCE Standard</li> <li>集群版本: v1.23 v1.25 v1.28 v1.31 ( 推荐 )</li> <li>集群规模: 50 200 1000 2000</li> <li>集群网络模式: 容器隧道网络 分布式训练时仅支持容器隧道网络</li> <li>集群转发模式: iptables ipvs</li> </ul>	<ul style="list-style-type: none"> <li>gpu-beta</li> <li>rdma-sriov-dev-plugin</li> </ul> <p>插件版本匹配关系请见 <a href="#">表1-5</a>。</p>

类型	卡类型	RDMA网络协议	操作系统	适用范围、约束	依赖插件
	gpu-ant1	RoCE	<ul style="list-style-type: none"> <li>操作系统: EulerOS 2.10 64bit</li> <li>内核版本: 4.18.0-147.5.2.1 5.h1109.euleros v2r10.x86_64</li> <li>架构类型: x86</li> </ul>	<ul style="list-style-type: none"> <li>集群类型: CCE Standard</li> <li>集群版本: v1.23 v1.25 v1.28 v1.31 (推荐)</li> <li>集群规模: 50 200 1000 2000</li> <li>集群网络模式: 容器隧道网络 分布式训练时仅支持容器隧道网络</li> <li>集群转发模式: iptables ipvs</li> </ul>	
	gpu-vnt1	RoCE IB	<ul style="list-style-type: none"> <li>操作系统: EulerOS 2.9 64bit (推荐)</li> <li>内核版本: 4.18.0-147.5.1.6. h841.eulerosv2r 9.x86_64</li> <li>架构类型: x86</li> </ul>	<ul style="list-style-type: none"> <li>集群类型: CCE Standard</li> <li>集群版本: v1.23 v1.25 v1.28 v1.31 (推荐)</li> <li>集群规模: 50 200 1000 2000</li> <li>集群网络模式: 容器隧道网络 VPC 分布式训练时仅支持容器隧道网络</li> <li>集群转发模式: iptables ipvs</li> </ul>	
<ul style="list-style-type: none"> <li>RDMA: Remote Direct Memory Access ( RDMA ) 是一种直接内存访问技术，将数据直接从一台计算机的内存传输到另一台计算机。</li> <li>RoCE: RDMA over Converged Ethernet ( RoCE ) 是一种网络协议，允许应用通过以太网实现远程内存访问。</li> <li>IB: InfiniBand ( IB ) 是一种高性能计算机网络通信协议，专为高性能计算和数据中心互连设计。</li> </ul>					

## 弹性云服务器对应的软件配套版本

表 1-4 弹性云服务器

类型	卡类型	操作系统	适用范围	依赖插件
NPU	ascend-snt3p-300i	<ul style="list-style-type: none"><li>操作系统：Huawei Cloud EulerOS 2.0 64bit</li><li>架构类型：x86、arm</li></ul>	<ul style="list-style-type: none"><li>集群类型：CCE Standard</li><li>集群版本：v1.23 (v1.23.5-r0及以上版本)   v1.25   v1.28   v1.31 (推荐)</li><li>集群规模：50 200 1000 2000</li><li>集群网络模式：容器隧道网络 VPC</li><li>集群转发模式：iptables ipvs</li></ul>	<ul style="list-style-type: none"><li>huawei-npu</li><li>volcano</li></ul> 插件版本匹配关系请见 <a href="#">表1-5</a> 。
		<ul style="list-style-type: none"><li>操作系统：EulerOS 2.9</li><li>架构类型：x86</li></ul>	<ul style="list-style-type: none"><li>集群类型：CCE Standard、CCE Turbo</li><li>集群版本：v1.23 (v1.23.5-r0及以上版本)   v1.25   v1.28 (推荐)</li><li>集群规模：50 200 1000 2000</li><li>集群网络模式：容器隧道网络 VPC ENI</li><li>集群转发模式：iptables ipvs</li></ul>	
	ascend-snt3	<ul style="list-style-type: none"><li>操作系统：EulerOS 2.5</li><li>架构类型：x86</li></ul>	<ul style="list-style-type: none"><li>集群类型：CCE Standard</li><li>集群版本：v1.23   v1.25</li><li>集群规模：50 200 1000 2000</li><li>集群网络模式：容器隧道网络 VPC</li><li>集群转发模式：iptables ipvs</li></ul>	

类型	卡类型	操作系统	适用范围	依赖插件
	ascend-snt9b	<ul style="list-style-type: none"> <li>操作系统: EulerOS 2.8</li> <li>架构类型: arm</li> </ul>	<ul style="list-style-type: none"> <li>集群类型: CCE Standard</li> <li>集群版本: v1.23 v1.25</li> <li>集群规模: 50 200 1000 2000</li> <li>集群网络模式: 容器 隧道网络 VPC</li> <li>集群转发模式: iptables ipvs</li> </ul>	
		<ul style="list-style-type: none"> <li>操作系统: Huawei Cloud EulerOS 2.0 64bit</li> <li>架构类型: arm</li> </ul>	<ul style="list-style-type: none"> <li>集群类型: CCE Standard</li> <li>集群版本: v1.23 ( v1.23.5-r0及以上版本 )  v1.25 v1.28 v1.31 ( 推荐 )</li> <li>集群规模: 50 200 1000 2000</li> <li>集群网络模式: 容器 隧道网络 VPC</li> <li>集群转发模式: iptables ipvs</li> </ul>	
G P U	gpu-vnt1	<ul style="list-style-type: none"> <li>操作系统: EulerOS 2.9</li> <li>架构类型: x86</li> </ul>	<ul style="list-style-type: none"> <li>集群类型: CCE Standard</li> <li>集群版本: v1.23 v1.25 v1.28 v1.31 ( 推荐 )</li> <li>集群规模: 50 200 1000 2000</li> <li>集群网络模式: 容器 隧道网络 VPC</li> <li>集群转发模式: iptables ipvs</li> </ul>	<ul style="list-style-type: none"> <li>gpu-beta</li> <li>rdma-sriov-dev-plugin</li> </ul> <p>插件版本匹配关系请见<a href="#">表1-5</a>。</p>

类型	卡类型	操作系统	适用范围	依赖插件
	gpu-ant03	<ul style="list-style-type: none"><li>● 操作系统：EulerOS 2.9</li><li>● 架构类型：x86</li></ul>	<ul style="list-style-type: none"><li>● 集群类型：CCE Standard</li><li>● 集群版本：v1.23 v1.25 v1.28 v1.31（推荐）</li><li>● 集群规模：50 200 1000 2000</li><li>● 集群网络模式：容器隧道网络 VPC</li><li>● 集群转发模式：iptables ipvs</li></ul>	
	gpu-ant1-pcie40	<ul style="list-style-type: none"><li>● 操作系统：EulerOS 2.9</li><li>● 架构类型：x86</li></ul>	<ul style="list-style-type: none"><li>● 集群类型：CCE Standard</li><li>● 集群版本：v1.23 v1.25 v1.28 v1.31（推荐）</li><li>● 集群规模：50 200 1000 2000</li><li>● 集群网络模式：容器隧道网络 VPC</li><li>● 集群转发模式：iptables ipvs</li></ul>	
	gpu-tnt004	<ul style="list-style-type: none"><li>● 操作系统：EulerOS 2.9</li><li>● 架构类型：x86</li></ul>	<ul style="list-style-type: none"><li>● 集群类型：CCE Standard</li><li>● 集群版本：v1.23 v1.25 v1.28 v1.31（推荐）</li><li>● 集群规模：50 200 1000 2000</li><li>● 集群网络模式：容器隧道网络 VPC</li><li>● 集群转发模式：iptables ipvs</li></ul>	

## 驱动和插件版本与 CCE 集群版本适配关系

表 1-5 驱动与 CCE 集群版本适配关系

类别	驱动名称	驱动版本	适配CCE集群版本	适用范围、约束	插件功能描述
npuDriver	npu-driver	7.1.0.9.220-2 3.0.6 ( 推荐 )  7.1.0.7.220-2 3.0.5  7.1.0.5.220-2 3.0.3	无约束	NPU ( snt9b )	用于升级、回滚npu驱动。
gpuDriver	gpu-driver	515.65.01 ( 推荐 ) 510.47.03 470.182.03 470.57.02	无约束	GPU	用于升级、回滚gpu驱动，插件依赖gpu-beta版本。

表 1-6 插件版本与 CCE 集群版本适配关系

插件名称	插件版本	适配CCE集群版本	适用范围、约束	插件功能描述
gpu-beta	2.7.63 ( 推荐 )	v1.(28 31).*	GPU	支持在容器中使用GPU显卡的设备管理插件。
	2.6.4	v1.28.*		
	2.0.48	v1.(23 25).*		
huawei-npu	2.1.53 ( 推荐 )	v1.(23 25 28 31).*	NPU	支持容器里使用huawei NPU设备的管理插件。
	2.1.22	v1.(23 25 28).*		
volcano	1.16.8 ( 推荐 )	v1.(23 25 28 31).*	NPU	基于Kubernetes的批处理平台。
	1.15.8	v1.(23 25 28).*		
os-node-agent	7.0.0	无约束	无约束	OS插件，用于故障检测。
icagent	default	CCE默认安装当前适配版本	无约束	CCE基础组件，用于日志和监控。

# 2 Lite Cluster 资源开通

ModelArts Lite Cluster是华为云ModelArts平台中的一种专属资源池，面向k8s资源型用户，提供托管式k8s集群，并预装主流AI开发插件以及自研的加速插件，以云原生方式直接向用户提供AI Native的资源、任务等能力。用户可以直接操作资源池中的节点和k8s集群，适合需要直接使用云原生资源的场景。

ModelArts Lite Cluster与ModelArts Standard资源池区别：

- ModelArts Lite资源池：用户可以直接操作节点和k8s集群，适合需要直接使用云原生资源的场景。
- ModelArts Standard资源池：提供训练、推理、开发环境等上层能力，适合一站式开发需求。

本章节主要介绍开通Lite Cluster资源池的详细操作。

## 计费影响

- 在开通Lite Cluster资源后，会产生计算资源的计费。Lite Cluster资源池仅支持包年/包月计费模式，具体内容如表2-1所示。

表 2-1 计费项

计费项	计费项说明	适用的计费模式	计费公式
计算资源 专属资源池	使用计算资源的用量。 具体费用可参见 <a href="#">ModelArts价格详情</a> 。	包年/包月	规格单价 * 计算节点个数 * 购买时长

- 购买Cluster资源池时，需要选择CCE集群，具体费用请参考[CCE计费详情](#)。

## 集群资源开通流程

开通集群资源过程中用户侧需要完成的任务流程如下图所示。

图 2-1 用户侧任务流程

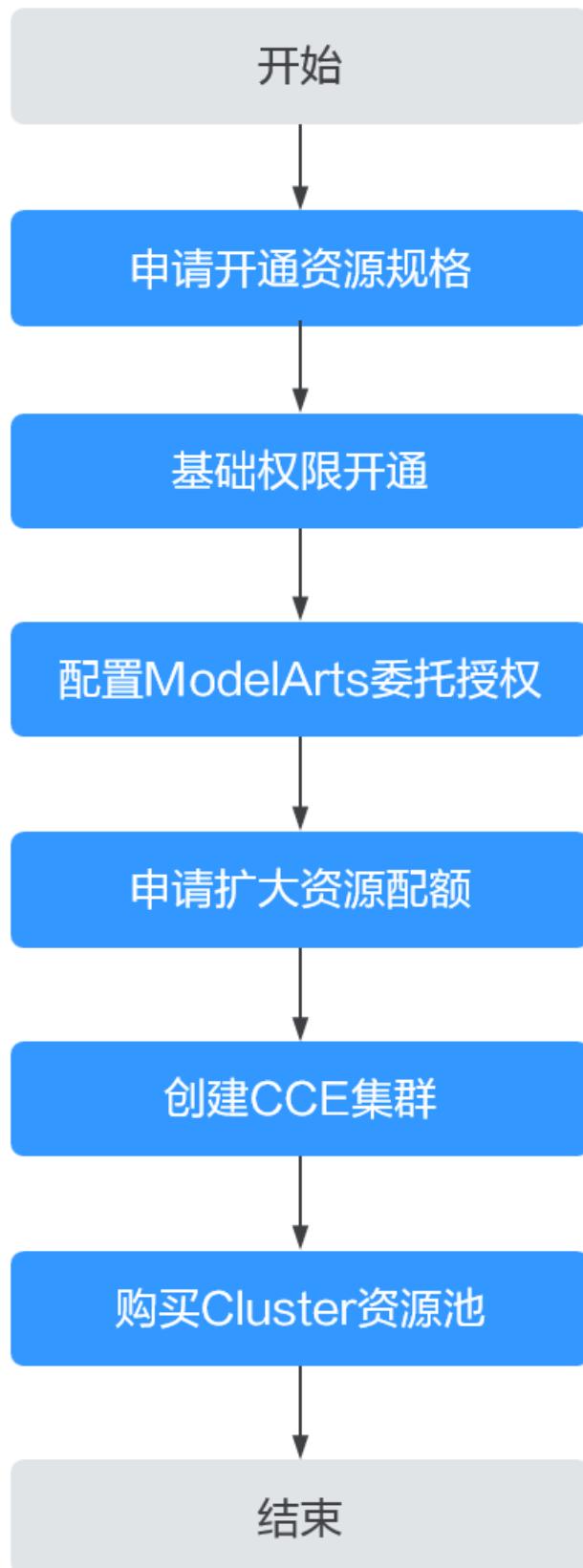


表 2-2 Cluster 资源开通流程

任务	说明
准备工作	在开通资源前，请确保完成所有相关准备工作，包括申请开通所需的规格和进行权限配置。
购买Lite Cluster 资源池	在ModelArts控制台上购买Cluster资源。

## 准备工作

在开通资源前，请确保完成所有相关准备工作，包括申请开通所需的规格和进行权限配置。

## Step1 申请开通资源规格

当前部分规格为受限购买（如modelarts.bm.npu.arm.8snt9b3.d），需要提前联系客户经理申请开通资源规格，预计1~3个工作日内开通（如果无客户经理可提交工单反馈）。

## Step2 基础权限开通

基础权限开通需要登录管理员账号，为子用户账号开通使用资源池所需的基础权限。

**步骤1** 登录[统一身份认证服务管理控制台](#)。

**步骤2** 单击目录左侧“用户组”，然后在页面右上角单击“创建用户组”。

**步骤3** 填写“用户组名称”并单击“确定”，完成用户组创建。

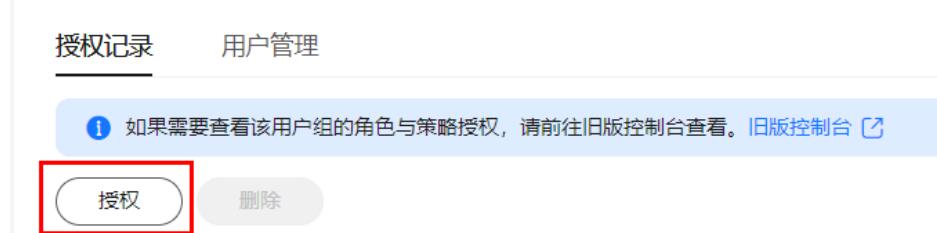
用户组名称只能包含中文、大小写字母、数字、空格或特殊字符(-\_)。

**步骤4** 在操作列单击“用户组管理”，将需要配置权限的用户加入用户组中。

**步骤5** 单击用户组名称，进入用户组详情页。

**步骤6** 在权限管理页签下，单击“授权”。

图 2-2 “配置权限”



**步骤7** 在搜索栏输入“ModelArtsFullAccessPolicy”，并勾选“ModelArtsFullAccessPolicy”。

图 2-3 ModelArtsFullAccessPolicy



以相同的方式，依次添加如下权限：

- ModelArts FullAccess
- CTS Administrator
- CCE Administrator
- BMS FullAccess
- IMS FullAccess
- DEW KeypairReadOnlyAccess
- VPC FullAccess
- ECS FullAccess
- SFS Turbo FullAccess
- OBS Administrator
- AOM FullAccess
- TMS FullAccess
- BSS Administrator

**步骤8** 单击“下一步”，授权范围方案选择“所有资源”。

**步骤9** 单击“确认”，完成基础权限开通。

设置权限完成后，单击用户组名称，进入用户组详情页，在授权记录页签下可以查看到已授予的权限。

----结束

### Step3 在 ModelArts 上创建委托授权

- 新建委托

第一次使用ModelArts时需要创建委托授权，授权允许ModelArts代表用户去访问其他云服务。

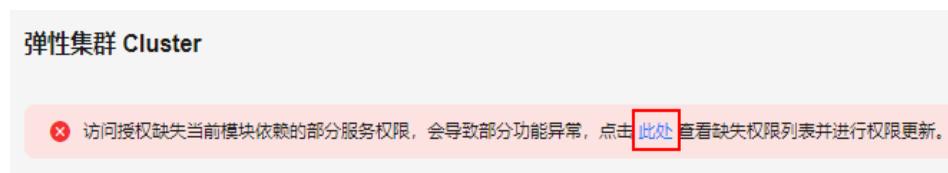
进入到**ModelArts管理控制台**的“权限管理”页面，单击“添加授权”，根据提示进行操作，详情请见[添加授权](#)。

- 更新委托

如果之前给ModelArts创过委托授权，此处可以更新授权。

- a. 进入到**ModelArts管理控制台**的“资源管理>轻量算力集群（Lite Cluster）”页面，查看是否存在授权缺失的提示。

图 2-4 轻量算力集群（Lite Cluster）权限缺失提示



- b. 如果有授权缺失，根据提示，单击“此处”更新委托。根据提示选择“追加至已有授权”，单击“确定”，系统会提示权限更新成功。

图 2-5 追加授权

### 访问授权权限不足

访问授权缺失当前模块依赖的如下服务权限，如需继续使用请添加权限至访问授权。[常见问题](#)



## Step4 申请扩大资源配置

集群所需的ECS实例数、内存大小、CPU核数和EVS硬盘大小等资源会超出华为云默认提供的资源配置，因此需要申请扩大配额。具体的配额方案请通过工单系统提交申请来联系客户经理获取。

配额需大于要开通的资源，且在购买开通前完成配额提升，否则会导致资源开通失败。

由于AI机型规格相对较大，资源池所需的ECS实例数、内存大小、CPU核数和EVS硬盘大小很可能会超出华为云默认提供的资源配置，因此需要申请扩大配额。请先联系客户经理确认资源配置提升具体方案，再参考本章节申请扩大配额。

**步骤1** 登录[华为云管理控制台](#)。

**步骤2** 在顶部导航栏单击“资源 > 我的配额”，进入服务配额页面。

图 2-6 我的配额



**步骤3** 在服务配额页面，单击右上角的“申请扩大配额”，填写申请材料后提交工单。

申请扩大配额主要是申请弹性云服务器ECS实例数、核心数（CPU核数）、RAM容量（内存大小）和云硬盘EVS磁盘容量这4个资源配额。具体的配额数量请先联系客户经理获取。

**图 2-7 ECS 资源类型**

### 服务配额 ②

服务	资源类型
frc	伸缩带宽策略
弹性云服务器 ECS	实例数 核心数 RAM容量(MB)

**图 2-8 云硬盘资源类型**

云硬盘	磁盘数 磁盘容量(GB) 快照数
-----	------------------------

----结束

## Step5 购买 CCE 集群

由于Lite Cluster资源池依赖于CCE集群来提供容器化的运行环境，并且CCE集群为Lite资源池提供必要的计算、存储和网络资源，所以购买Cluster资源池时，需要选择CCE集群。

如果您没有可用的CCE集群，可参考[购买Standard/Turbo集群](#)进行购买，集群配套版本请参考[1.3 不同机型对应的软件配套版本](#)。

当前仅支持CCE集群1.23&1.25&1.28&1.31版本。CCE 1.28集群版本支持通过控制台、API方式创建，CCE 1.23和CCE 1.25版本支持通过API方式创建，CCE 1.31集群版本支持通过控制台、API方式创建。不同版本的CCE集群创建方式请见[Kubernetes版本策略](#)。

如果您已有CCE集群，但CCE集群版本低于1.23，则可参考[升级集群的流程和方法](#)，建议将集群升级至1.28版本。

创建Cluster资源池时，请确保CCE集群为“运行中”状态。

## 购买 Lite Cluster 资源池

1. 登录[ModelArts管理控制台](#)，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”，进入“轻量算力集群（Lite Cluster）”页面。

- 在“轻量算力集群（Lite Cluster）”页面，单击“购买轻量算力集群”，进入购买轻量算力集群界面，参见下表填写参数。

表 2-3 Lite Cluster 资源池的参数说明

参数名称	子参数	说明
计费模式	包年/包月	包年/包月是预付费模式，按订单的购买周期计费，适用于可预估资源使用周期的场景，价格比按需计费模式更优惠。 Lite Cluster暂时只支持包年/包月计费模式。
集群规格	集群名称	系统默认提供一个名称，可以手动修改。 只能以小写字母开头，由小写字母、数字、中划线（-）组成，不能以中划线结尾。
	选择CCE集群	<p>在下拉列表中选择用户账户下已有的CCE集群。如果没有集群，单击右侧的“创建集群”，先去创建集群。集群配套版本请参考<a href="#">1.3 不同机型对应的软件配套版本</a>。</p> <p><b>警告</b> 用户在CCE创建用于ModelArts的CCE集群时，请勿勾选Volcano调度器，否则可能会导致Lite Cluster集群创建过程中节点驱动安装异常。 对于NPU类型，ModelArts Lite Cluster会在创建过程中自动安装Volcano调度器。</p> <p>创建Cluster资源池时，请确保CCE集群为“运行中”状态。</p> <p>当前仅支持CCE集群1.23&amp;1.25&amp;1.28&amp;1.31版本。CCE 1.28集群版本支持通过控制台、API方式创建，CCE 1.23和CCE 1.25版本支持通过API方式创建，CCE 1.31集群版本支持通过控制台、API方式创建。不同版本的CCE集群创建方式请见<a href="#">Kubernetes版本策略</a>。</p> <p>如果您已有CCE集群，但CCE集群版本低于1.23，则可参考<a href="#">升级集群的流程和方法</a>，建议将集群升级至1.28版本。</p>
节点池	节点池名称	为帮助您更好地管理Kubernetes集群内的节点，ModelArts支持通过节点池来管理节点。 新建节点池的名称，可自定义。 创建资源池后，暂不支持资源池中的存量节点池修改名称。
	节点类型	<ul style="list-style-type: none"><li>普通节点：单一物理主机或虚拟主机，提供基础的独立计算、存储和网络资源。</li><li>超节点：融合架构节点，提供大规模计算资源池，支持灵活调配和高密度部署。超节点专门用于支持大规模的模型推理任务。这些服务器通常配备有多个计算卡（如昇腾NPU），能够提供强大的计算能力，以满足高负载的推理需求。超节点资源仅支持西南-贵阳一、华北三、华北-乌兰察布一和华东二区域。</li><li>整柜节点：整合物理机独占资源，提供极高隔离性、性能和资源确定性。</li></ul>

参数名称	子参数	说明
	资源类型	<p>可以根据需要选择“裸金属服务器”或“弹性云服务器”。</p> <ul style="list-style-type: none"><li>• 裸金属服务器：是一款兼具弹性云服务器和物理机性能的计算类服务器，为您提供专属的云上物理服务器。</li><li>• 弹性云服务器：是一种可随时自助获取、可弹性伸缩的云服务器，可帮助您打造可靠、安全、灵活、高效的应用环境，确保服务持久稳定运行，提升运维效率。</li></ul>
	CPU架构	<p>CPU架构指的是中央处理器（CPU）的指令集和设计规范。支持X86和ARM64两种不同的CPU架构，同时支持X86和ARM64异构调度。请根据实际需要选择。</p> <ul style="list-style-type: none"><li>• X86：适用于大多数通用计算场景，支持广泛的软件生态。</li><li>• ARM64：适用于特定优化场景，如移动应用、嵌入式系统等，具有低功耗优势。</li></ul>
	实例规格类型	<p>支持CPU、GPU、Ascend三种芯片规格资源，根据实际需要选择。</p> <ul style="list-style-type: none"><li>• CPU：通用计算架构，适合通用任务，计算性能较低，适用于轻量级适合通用任务，计算性能较低。</li><li>• GPU：并行计算架构，适合并行任务，计算性能高，支持多卡分布式训练，适用于深度学习训练、图像处理等场景。</li><li>• Ascend：专用AI架构，适合AI任务，计算性能极高，支持多节点分布式部署，适用于AI模型训练、推理加速等场景。</li></ul>
	实例规格	<p>选择需要使用的规格。平台分配的资源规格包含了一定的系统损耗，实际可用的资源量小于规格标称的资源。实际可用的资源量可在资源池创建成功后，在详情页的“节点”页签中查看。</p>
	可用区	<p>根据实际情况选择“随机可用区”或“指定可用区”。可用区是在同一区域下，电力、网络隔离的物理区域。可用区之间内网互通，不同可用区之间物理隔离。</p> <ul style="list-style-type: none"><li>• 随机可用区：系统自动分配可用区。</li><li>• 指定可用区：指定资源池实例在哪个可用区域。考虑系统容灾时，推荐指定实例在同一个可用区。可设置可用区的实例数。</li></ul>

参数名称	子参数	说明
	实例数	<p>选择Lite Cluster资源池的实例个数（即节点个数），数量越多，计算性能越强。</p> <p>当“可用区”选择“指定可用区”时，实例数量会根据可用区的数据自动计算，此处无需再次设置。</p> <p>单次创建时，实例数建议不大于30，否则可能触发限流导致创建失败。</p> <p>部分区域的部分规格支持整柜购买，此时实例数会显示为“数量*整柜”，购买的实例总数为两者的乘积。整柜购买可实现不同任务间的物理隔离，避免通信冲突，在任务规模增大的同时保证计算性能线性度不下降。整柜下的实例生命周期需保持一致，需要一起创建、一起删除。</p> <p>超节点规格，即Snt9b23类型实例规格，支持自定义步长购买，此时实例数会显示为“数量*步长”，购买的实例总数为两者的乘积。步长为每次调整保障配额时的最小单位，在节点绑定场景下每个步长内的节点将作为一个整体，且属于同一批次。</p>

参数名称	子参数	说明
	存储配置	<p>支持设置如下存储配置。</p> <ul style="list-style-type: none"><li>系统盘：显示系统盘的磁盘类型和大小。系统盘的磁盘类型支持本地盘和云硬盘（包括通用SSD、高IO和超高IO）。部分规格的系统盘仅支持本地盘。</li><li>容器盘：显示容器盘的存储类型、大小和数量。部分规格的容器盘存储类型支持手动设置，可以选择本地盘或云硬盘。</li><li>容器盘高级配置：支持设置“指定磁盘空间”、“容器引擎空间大小”、写入模式。<ul style="list-style-type: none"><li>容器引擎空间大小：默认容器引擎空间大小为50GiB。可指定容器引擎空间大小和不限制空间大小。当选择“指定大小”时，默认值与最小值均为50GiB，不同规格的最大值不同，数值有效范围请参考界面提示。</li><li>写入模式：部分规格支持设置容器盘的写入模式为线性或条带化。线性逻辑卷是将一个或多个物理卷整合为一个逻辑卷，实际写入数据时会先往一个基本物理卷上写入，当存储空间占满时再往另一个基本物理卷写入。条带化是指创建逻辑卷时指定条带化，当实际写入数据时会将连续的数据分成大小相同的块，然后依次存储在多个物理卷上，实现数据的并发读写从而提高读写性能。条带化模式的存储池不支持扩容。</li></ul></li><li>数据盘：部分规格支持“添加普通数据盘”，挂载多个数据盘到资源池中。支持设置数据盘的“磁盘类型”、“大小”和“数量”。不同规格有不同的磁盘数量限制，比如：300I Duo节点仅支持挂载4个数据盘，具体请以界面提示为准。</li><li>数据盘高级配置：部分规格支持在数据盘高级配置参数中设置数据盘的挂载方式，具体如下：<ul style="list-style-type: none"><li>默认：仅是将云硬盘挂载到资源池上，未对挂载的云硬盘做任何处理，比如分区等。</li><li>挂载到指定目录：支持设置“数据盘挂载到的指定路径”和“写入模式”，包括线性和条带化。</li><li>以本地持久卷挂载：支持“持久卷写入模式”设置，包括线性和条带化，此处设置的是所有数据盘的写入模式。</li><li>以临时存储卷挂载：支持“临时卷写入模式”设置，包括线性和条带化，此处设置的是所有数据盘的写入模式。</li></ul></li></ul>
	网络配置	<p>勾选“网络配置”后，支持设置虚拟私有云为CCE集群网络。</p> <ul style="list-style-type: none"><li>虚拟私有云：默认为CCE集群所在VPC网络，不可修改。</li><li>节点子网：选择同一VPC网络下的子网作为节点子网，新创建的节点将会使用该子网资源。</li><li>关联安全组：用于指定节点池创建出来的节点使用的安全组。最多选择4个安全组。节点安全组需要放通一些端口以保障节点通信。如果不关联安全组将会使用集群中默认的节点安全组规则。</li></ul>

参数名称	子参数	说明
	环境配置	<p>勾选“环境配置”后，支持设置如下参数：</p> <ul style="list-style-type: none"><li>● 镜像配置：可以指定实例的操作系统。<ul style="list-style-type: none"><li>- 预置镜像：ModelArts提供，支持多种操作系统，并且内置了AI场景相关驱动和软件，为用户提供了一个完整的AI开发环境，方便用户直接进行开发和训练，而无需额外配置。</li><li>- 私有镜像：非ModelArts官方提供，请您在测试环境中自行完成充分的测试验证，再应用于生产环境。请确保您的镜像与CCE集群目标运行环境(如操作系统类型、内核版本、容器运行时、驱动等)兼容性、稳定性和安全性，不兼容的镜像可能导致驱动安装失败、AI业务部署失败或运行异常。</li></ul></li><li>● 容器引擎：容器引擎是Kubernetes最重要的组件之一，负责管理镜像和容器的生命周期。Kubelet通过Container Runtime Interface (CRI) 与容器引擎交互，以管理镜像和容器。此处支持选择Docker和Containerd。Containerd和Docker的详细差异对比请见<a href="#">容器引擎</a>。 您可以在创建资源池时选择容器引擎，也可在资源池创建完成后，在扩缩容界面修改。 如果CCE集群版本低于1.23，仅支持选择Docker作为容器引擎。如果CCE集群版本大于等于1.27，仅支持选择Containerd作为容器引擎。其余CCE集群版本，支持选择Containerd或Docker作为容器引擎。</li></ul>
	节点池标签管理	<p>单击添加节点池标签。</p> <ul style="list-style-type: none"><li>● 资源标签：通过为资源添加标签，可以对资源进行自定义标记，实现资源分类。也可在资源池创建完成后，在资源池详情页的“标签”页面修改。</li><li>● K8S标签：设置附加到Kubernetes对象（比如Pod）上的键值对。最多可以添加20条标签。使用该标签可区分不同节点，可结合工作负载的亲和能力实现容器Pod调度到指定节点的功能。</li><li>● 污点：默认为空。支持给节点加污点来设置反亲和性，每个节点最多配置20条污点。</li></ul>
	新增节点池	当您需要更多节点池时，可单击“新增节点池”创建多个节点池。根据业务设置各个节点池配置。

参数名称	子参数	说明
插件配置	选择插件	<p>ModelArts提供多种类型的插件，通过添加插件选择性扩展资源池功能，以满足业务需求。</p> <p>单击“选择插件”，在弹框中勾选需要配置的插件，单击“确定”。</p> <p>单击各插件的“查看详情”，可查看对应插件的功能介绍、版本更新特性等具体信息。</p> <p>默认添加如下插件：</p> <ul style="list-style-type: none"><li>● <b>节点故障检测(ModelArts Node Agent)</b>：ModelArts节点故障检测是一款监控集群节点异常事件的插件，以及对接第三方监控平台功能的组件。它是一个在每个节点上运行的守护程序，可从不同的守护进程中搜集节点问题。</li><li>● <b>6.3 指标监控插件(ModelArts Metrics Collector)</b>：默认内置插件，以节点守护程序运行，可采集节点及作业各类监控指标，并上报到AOM。</li><li>● <b>6.4 AI套件(ModelArts Device Plugin)</b>：支持容器里使用huawei NPU设备的管理插件。 仅实例规格类型选择“Ascend”时自动安装。</li><li>● <b>Volcano调度器(Volcano Scheduler)</b>：Volcano是一个基于Kubernetes的批处理平台，提供了机器学习、深度学习、生物信息学、基因组学及其他大数据应用所需要而Kubernetes当下缺失的一系列特性。</li></ul>
资源调度与切分	GPU驱动/NPU驱动	<p>打开“自定义驱动”开关，显示此参数，选择GPU/NPU驱动。如果规格类型为GPU则显示“GPU驱动”，如果规格类型为Ascend则显示“NPU驱动”。</p> <p>gpu-driver配套版本请参考<a href="#">1.3 不同机型对应的软件配套版本</a>。</p>
高级配置	集群描述	自定义集群描述信息。
	自定义节点前缀	<p>开启后，可为节点名称添加前缀。</p> <ul style="list-style-type: none"><li>● 添加前缀后，节点名称由前缀+随机数组成。</li><li>● 输入长度范围为1到64个字符。</li><li>● 前缀必须以小写字母开头，并由小写字母和数字组成，以“-”分隔。例如：node-com。</li></ul>
	标签	单击“添加新标签”，可以为Lite资源池配置标签信息，通过标签实现资源的分组管理。此处的标签信息可以同源标签管理服务TMS中预定义的标签信息。也可以在创建完成后的Lite资源池详情页面中通过“标签”页签设置标签信息。
管理	登录凭证	<p>集群登录方式，可以设置密码登录，也可以设置密钥对登录。</p> <ul style="list-style-type: none"><li>● 密码登录：默认用户名为root，用户自己设置密码。</li><li>● 密钥对（KeyPair）登录：可以选择已有的密钥对，或者单击右侧的“创建密钥对”，先去创建一个密钥对。</li></ul>

参数名称	子参数	说明
购买时长	-	选择购买时长。只有选择“包年/包月”计费模式时才需填写。自动续费默认关闭。勾选自动续费后，资源池到期后，会自动续期。自动续费时系统从可用余额扣款，详情请见 <a href="#">自动续费</a> 。如果购买时是按月购买，则按照1个月周期自动续费。如果购买时是按年购买，则自动续费周期为1年。

- 单击“立即购买”确认规格。产品规格和协议许可确认无误后，单击“提交”，即可创建Lite资源池。
  - 当资源池创建成功后，资源池的状态会变成“运行中”，当“节点个数”中的“可用节点”和“总数”值大于0时，资源池才能下发任务。
  - 可以将鼠标放在“创建中”字样上，查看当前创建过程详情。如果单击查看详情，可跳转到“操作记录”中。
  - 可以在Lite资源池列表右上角的“操作记录”中查看资源池的任务记录。

图 2-9 操作记录



图 2-10 查看操作记录

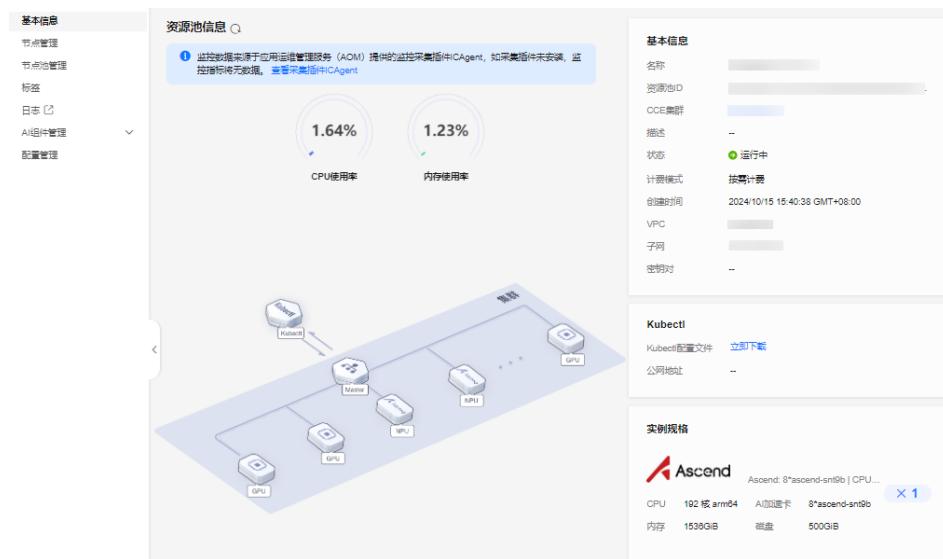


The screenshot shows a table with the following data:

名称ID	操作状态	操作类型	计费模式	创建时间
pool-f143xq pool- ...	等待中	新建	包年/包月	2024/09/23 11:59:08 GMT+08:00
订单ID	CS24 [redacted]	开始处理时间	--	
初始化规格	--	结束时间	--	
目标规格	1 * m4.0c [redacted] 11.8.64, 1 * m4.0s [redacted] 32	实际规格	--	

当资源池创建成功后，资源池的状态会变成“运行中”。单击集群资源名称，进入资源详情页。确认购买的规格是否正确。

图 2-11 查看资源详情



# 3 Lite Cluster 资源配置

## 3.1 Lite Cluster 资源配置流程

本章节介绍Lite Cluster环境配置详细流程，适用于加速卡环境配置。

### 前提条件

- 已完成集群资源购买和开通，具体请参见[2 Lite Cluster资源配置](#)。
- 集群的配置使用需要用户具备一定的知识背景，包括但不限于[Kubernetes基础知识](#)、网络知识、存储和镜像知识。

## 配置流程

图 3-1 Lite Cluster 资源配置流程图

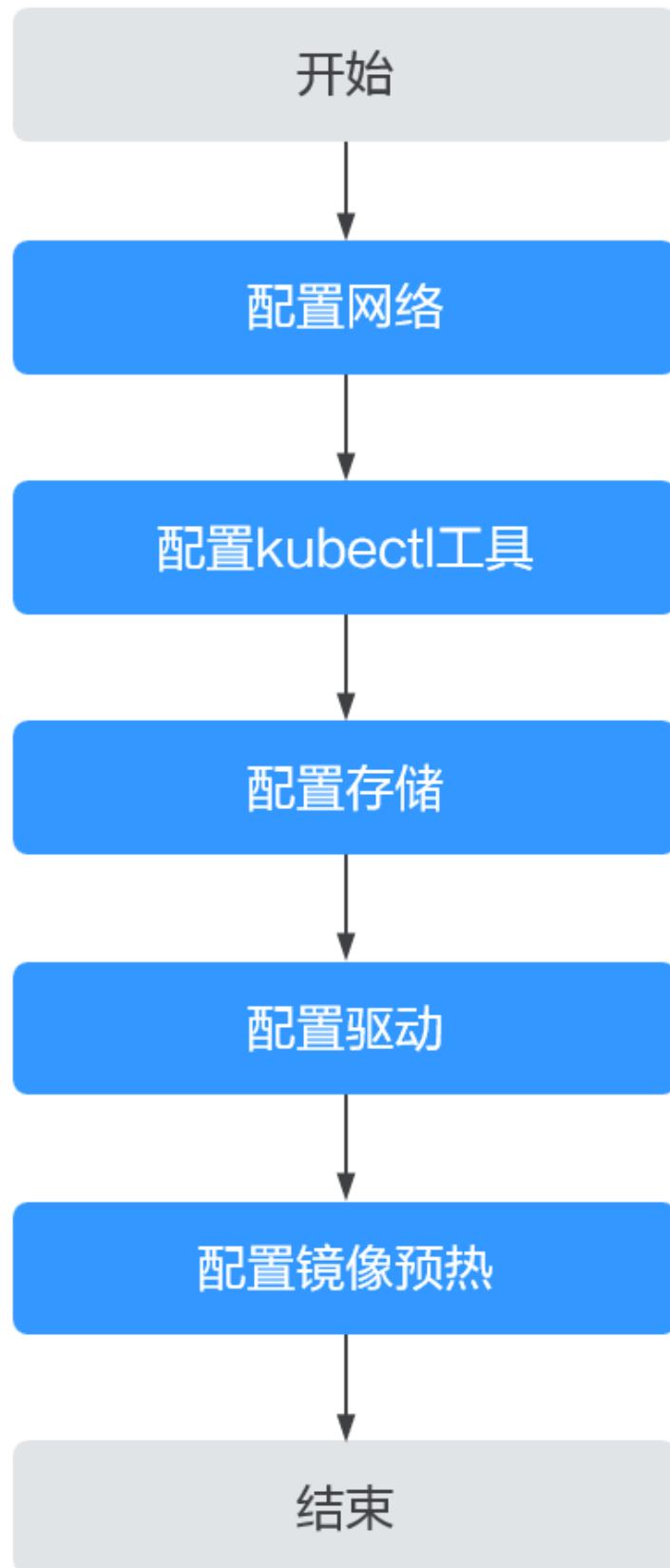


表 3-1 Cluster 资源配置流程

配置顺序	配置任务	场景说明
1	<a href="#">3.2 配置Lite Cluster网络</a>	购买资源池后，需要弹性公网IP并进行网络配置，配置网络后可通过公网访问集群资源。
2	<a href="#">3.3 配置kubectl工具</a>	kubectl是Kubernetes集群的命令行工具，配置kubectl后，您可通过kubectl命令操作Kubernetes集群。
3	<a href="#">3.4 配置Lite Cluster存储</a>	如果没有挂载任何外部存储，此时可用存储空间根据dockerBaseSize的配置来决定，可访问的存储空间比较小，因此建议通过挂载外部存储空间解决存储空间受限问题。容器中挂载存储有多种方式，不同的场景下推荐的存储方式不一样，您可根据业务实际情进行选择。
4	<a href="#">3.5（可选）配置驱动</a>	当专属资源池中的节点含有GPU/Ascend资源时，为确保GPU/Ascend资源能够正常使用，需要配置好对应的驱动。如果在购买资源池时，没配置自定义驱动，默认驱动不满足业务要求，可通过本章节将驱动升级到指定版本。
5	<a href="#">3.6（可选）配置镜像预热</a>	Lite Cluster资源池支持镜像预热功能，镜像预热可实现将镜像提前在资源池节点上拉取好，在推理及大规模分布式训练时有效缩短镜像拉取时间。

## 快速配置 Lite Cluster 资源案例

下文提供一个快速配置的案例，配置完成后您可登录到节点查看加速卡信息并完成一个训练任务。在运行此案例前，您需要购买资源，购买资源的步骤请参考[2 Lite Cluster资源开通](#)。

### 步骤1 登录节点。

#### （推荐）方式1：通过绑定公网IP的方式

客户可以为需要登录的节点绑定公网IP，然后可以通过Xshell、MobaXterm等bash工具登录节点。

1. 使用华为云账号登录CCE管理控制台。
2. 在CCE集群详情页面，单击“节点管理”页签，在“节点”页签中单击需要登录的节点名称，跳转至弹性云服务器页面。

图 3-2 节点管理

The screenshot shows the 'Nodes' tab of the 'Node Management' section in the ModelArts Lite Cluster interface. On the left sidebar, under 'Cluster', 'Node Management' is highlighted with a red circle labeled '1'. In the main area, there is a message about installing the NPD plugin. Below it are buttons for 'Export', 'Sync All Cloud Servers', 'Tag & Label Management', and 'More'. A search bar allows filtering by keyword. A table lists nodes: 'pool' (status: Running, Schedulable), 'dly-k' (status: Running, Schedulable), and another 'dly-k' (status: Running, Schedulable). A red circle labeled '2' is on the 'Nodes' tab, and a red circle labeled '3' is next to the first 'pool' entry.

## 3. 绑定弹性公网IP。

如果已有未绑定的弹性公网IP，直接选择即可。如果没有可用的弹性公网IP，需要先购买弹性公网IP。

图 3-3 弹性公网 IP

The screenshot shows the 'Elastic Public IP' tab of the node configuration interface. At the bottom, there are two buttons: 'Bind Elastic Public IP' and 'View Elastic Public IP'. The 'Bind Elastic Public IP' button is highlighted with a red rectangle.

单击“购买弹性公网IP”，进入购买页。

图 3-4 绑定弹性公网 IP



图 3-5 购买弹性公网 IP



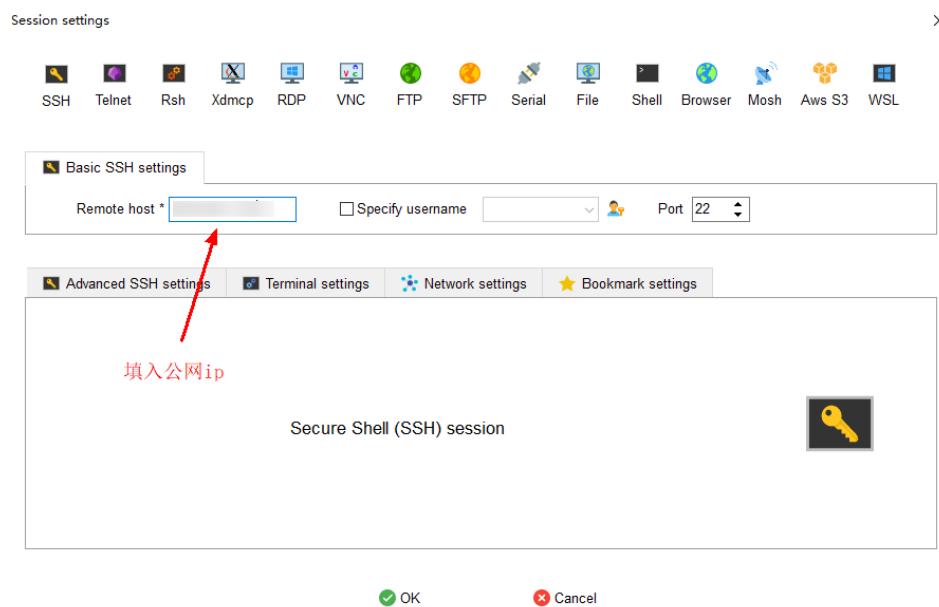
完成购买后，返回弹性云服务器页面，刷新列表。  
选择刚才创建的弹性公网IP，单击“确定”。

图 3-6 绑定弹性公网 IP



4. 绑定完成后，通过MobaXterm、Xshell登录。以MobaXterm为例，填入弹性公网IP，登录节点。

图 3-7 登录节点



### 方式2：通过华为云自带的远程登录功能

1. 使用华为云账号登录CCE管理控制台。
2. 在CCE集群详情页面，单击“节点管理”页签，在“节点”页签中单击需要登录的节点名称，跳转至弹性云服务器页面。

图 3-8 节点管理

The screenshot shows the 'pool' cluster's node management page. The left sidebar has '节点管理' (Node Management) highlighted with a red circle. The main area shows a table of nodes:

节点名称	状态	所属节点池
pool	运行中 可调度	pool
dly-k	运行中 可调度	pool
dly-k	运行中 可调度	pool

- 单击“远程登录”，在弹出的窗口中，单击“CloudShell登录”。

图 3-9 远程登录



- 在CloudShell中设置密码等参数后，单击“连接”即可登录节点，CloudShell介绍可参见[远程登录Linux弹性云服务器（CloudShell方式）](#)。

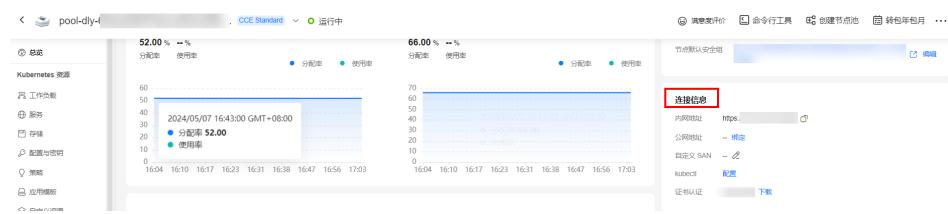
## 步骤2 配置kubectl工具。

登录[ModelArts管理控制台](#)，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”，进入“轻量算力集群（Lite Cluster）”页面。

单击创建的专属资源池，进入专属资源池详情页面，单击对应的CCE集群，进入CCE集群详情页面。

在CCE集群详情页面中，在集群信息中找到“连接信息”。

图 3-10 链接信息



使用kubectl工具。

- 如果通过内网使用kubectl工具，需要将kubectl工具安装在和集群在相同vpc下的某一台机器上。单击连接信息下kubectl后的“配置”按钮，根据界面提示使用kubectl工具。

图 3-11 通过内网使用 kubectl 工具

```
[root@ ~]# kubectl get node
The connection to the server localhost:8080 was refused - did you specify the right host or port?
[root@ ~]# cd /root/
[root@ ~]# mkdir .kube
[root@ ~]# cd .kube
[kube]# vi config
[kube]# kubectl config use-context internal
Switched to context "internal".
[kube]# kubectl get node
NAME      STATUS   ROLES     AGE      VERSION
.         Ready    <none>    14m     v1.23.9-r0-23.2.32
```

- 通过公网使用kubectl工具，可以将kubectl安装在任一台可以访问公网的机器。首先需要绑定公网地址，单击公网地址后的“绑定”按钮。

图 3-12 绑定公网地址



选择公网IP后单击“确定”，完成公网IP绑定。如果没有可选的公网IP，单击“创建弹性IP”跳至弹性公网IP页面进行创建。

绑定完成后，单击连接信息下kubectl后的“配置”按钮，根据界面提示使用kubectl工具。

### 步骤3 docker run方式启动任务。

Snt9B集群在纳管到CCE集群后，会安装容器运行时，下文以docker举例。仅做测试验证，可以不需要通过创建deployment或者volcano job的方式，直接启动容器进行测试。训练测试用例使用NLP的bert模型。

1. 拉取镜像。本测试镜像为bert\_pretrain\_mindspore:v1，已经把测试数据和代码打进镜像中。

```
docker pull swr.cn-southwest-2.myhuaweicloud.com/os-public-repo/bert_pretrain_mindspore:v1
docker tag swr.cn-southwest-2.myhuaweicloud.com/os-public-repo/bert_pretrain_mindspore:v1
bert_pretrain_mindspore:v1
```

2. 启动容器。

```
docker run -tid --privileged=true \
-u 0 \
-v /dev/shm:/dev/shm \
--device=/dev/davinci0 \
--device=/dev/davinci1 \
--device=/dev/davinci2 \
--device=/dev/davinci3 \
--device=/dev/davinci4 \
--device=/dev/davinci5 \
--device=/dev/davinci6 \
--device=/dev/davinci7 \
--device=/dev/davinci_manager \
--device=/dev/devmm_svm \
--device=/dev/hisi_hdc \
-v /usr/local/Ascend/driver:/usr/local/Ascend/driver \
-v /usr/local/sbin/npu-smi:/usr/local/sbin/npu-smi \
-v /etc/hccn.conf:/etc/hccn.conf \
bert_pretrain_mindspore:v1 \
bash
```

参数含义：

- --privileged=true //特权容器，允许访问连接到主机的所有设备
- -u 0 //root用户
- -v /dev/shm:/dev/shm //防止shm太小训练任务失败
- --device=/dev/davinci0 //npu卡设备
- --device=/dev/davinci1 //npu卡设备
- --device=/dev/davinci2 //npu卡设备

- --device=/dev/davinci3 //npu卡设备
- --device=/dev/davinci4 //npu卡设备
- --device=/dev/davinci5 //npu卡设备
- --device=/dev/davinci6 //npu卡设备
- --device=/dev/davinci7 //npu卡设备
- --device=/dev/davinci\_manager //davinci相关的设备管理的设备
- --device=/dev/devmm\_svm //管理设备
- --device=/dev/hisi\_hdc //管理设备
- -v /usr/local/Ascend/Driver:/usr/local/Ascend/Driver //npu卡驱动挂载
- -v /usr/local/sbin/npu-smi:/usr/local/sbin/npu-smi //npu-smi工具挂载
- -v /etc/hccn.conf:/etc/hccn.conf //hccn.conf配置挂载

3. 进入容器，并查看卡信息。

```
docker exec -it xxxxxxxx bash //进入容器，xxxxxxxx替换为容器id  
npu-smi info //查看卡信息
```

图 3-13 查看卡信息

npu-smi 23.0.rc2 Version: 23.0.rc2.2.b030					
NPU	Name	Health	Power(W)	Temp(C)	Hugepages-Usage(page)
Chip	Bus-Id	AICore(%)	Memory-Usage(MB)	HBM-Usage(MB)	
0	910B1	OK	93.1	46	0 / 0
0	0000:C1:00.0	0	0	0 / 0	4313 / 65536
1	910B1	OK	93.5	48	0 / 0
0	0000:01:00.0	0	0	0 / 0	4313 / 65536
2	910B1	OK	93.0	46	0 / 0
0	0000:C2:00.0	0	0	0 / 0	4314 / 65536
3	910B1	OK	93.1	47	0 / 0
0	0000:02:00.0	0	0	0 / 0	4339 / 65536
4	910B1	OK	93.3	48	0 / 0
0	0000:81:00.0	0	0	0 / 0	4313 / 65536
5	910B1	OK	94.8	48	0 / 0
0	0000:41:00.0	0	0	0 / 0	4181 / 65536
6	910B1	OK	93.3	49	0 / 0
0	0000:82:00.0	0	0	0 / 0	4180 / 65536
7	910B1	OK	93.2	48	0 / 0
0	0000:42:00.0	0	0	0 / 0	4180 / 65536
NPU	Chip	Process id	Process name	Process memory(MB)	
No running processes found in NPU 0					
No running processes found in NPU 1					
No running processes found in NPU 2					
No running processes found in NPU 3					
No running processes found in NPU 4					
No running processes found in NPU 5					
No running processes found in NPU 6					
No running processes found in NPU 7					

4. 执行下述命令启动训练任务。

```
cd /home/ma-user/modelarts/user-job-dir/code/bert/  
export MS_ENABLE_GE=1  
export MS_GE_TRAIN=1  
bash scripts/run_standalone_pretrain_ascend.sh 0 1 /home/ma-user/modelarts/user-job-dir/data/cn-news-128-1f-mind/
```

图 3-14 训练进程

```
[root@3c799939827b bert]# export MS_ENABLE_GE=1
[root@3c799939827b bert]# export MS_GE_TRAIN=1
[root@3c799939827b bert]# bash scripts/run_standalone_pretrain_ascend.sh 0 1 /home/ma-user/modelarts/user-job-dir/data/cn-news-128-if-mind/
=====
Please run the script as:
bash scripts/run_standalone_pretrain_ascend.sh DEVICE_ID EPOCH_SIZE DATA_DIR SCHEMA_DIR
for example: bash scripts/run_standalone_pretrain_ascend.sh 0 40 /path/zhi-wiki/ [/path/Schema.json][optional]
=====
[root@3c799939827b bert]# ps -ef
UID      PID  PPID  C STIME TTY      TIME CMD
root       1      0  0 15:55 pts/0    00:00:00 bash
root      22      0  0 15:55 pts/1    00:00:00 bash
root      61     22  1 99 15:56 pts/1   00:00:04 python /home/ma-user/modelarts/user-job-dir/code/bert/scripts/../run_pretrain.py --distribute=false --epoch_size=1
root     130     22  0 15:56 pts/1   00:00:00 ps -ef
```

查看卡占用情况，如图所示，此时0号卡被占用，说明进程正常启动。

npu-smi info //查看卡信息

图 3-15 查看卡信息

```
[root@3c799939827b bert]# npu-smi info
+-----+-----+-----+-----+-----+
| NPU  Name | Health | Power(W) | Temp(C) | Hugepages-Usage(page) |
| Chip      | Bus-Id | AICore(%) | Memory-Usage(MB) | HBM-Usage(MB) |
+-----+-----+-----+-----+-----+
| 0  910B1  | OK    | 102.4    | 47      | 0   / 0   |
| 0  0000:C1:00.0 | 0      | 0         | 0   / 0   | 19773 / 65536 |
+-----+-----+-----+-----+-----+
| 1  910B1  | OK    | 94.8     | 48      | 0   / 0   |
| 0  0000:01:00.0 | 0      | 0         | 0   / 0   | 4313 / 65536 |
+-----+-----+-----+-----+-----+
| 2  910B1  | OK    | 93.0     | 47      | 0   / 0   |
| 0  0000:C2:00.0 | 0      | 0         | 0   / 0   | 4314 / 65536 |
+-----+-----+-----+-----+-----+
| 3  910B1  | OK    | 93.1     | 47      | 0   / 0   |
| 0  0000:02:00.0 | 0      | 0         | 0   / 0   | 4338 / 65536 |
+-----+-----+-----+-----+-----+
| 4  910B1  | OK    | 93.2     | 48      | 0   / 0   |
| 0  0000:81:00.0 | 0      | 0         | 0   / 0   | 4312 / 65536 |
+-----+-----+-----+-----+-----+
| 5  910B1  | OK    | 95.6     | 48      | 0   / 0   |
| 0  0000:41:00.0 | 0      | 0         | 0   / 0   | 4180 / 65536 |
+-----+-----+-----+-----+-----+
| 6  910B1  | OK    | 93.6     | 48      | 0   / 0   |
| 0  0000:82:00.0 | 0      | 0         | 0   / 0   | 4180 / 65536 |
+-----+-----+-----+-----+-----+
| 7  910B1  | OK    | 93.7     | 49      | 0   / 0   |
| 0  0000:42:00.0 | 0      | 0         | 0   / 0   | 4180 / 65536 |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| NPU  Chip | Process id | Process name | Process memory(MB) |
+-----+-----+-----+-----+
| 0  0      | 2610117    |              | 15435          |
+-----+-----+-----+-----+
| No running processes found in NPU 1 |
+-----+-----+
| No running processes found in NPU 2 |
+-----+-----+
| No running processes found in NPU 3 |
+-----+-----+
| No running processes found in NPU 4 |
+-----+-----+
| No running processes found in NPU 5 |
+-----+-----+
| No running processes found in NPU 6 |
+-----+-----+
| No running processes found in NPU 7 |
+-----+-----+
```

训练任务大概会运行两小时左右，训练完成后自动停止。如果想停止训练任务，可执行下述命令关闭进程，查询进程后显示已无运行中python进程。

ps -ef  
pkill -9 python

图 3-16 关闭训练进程

```
[root@7890c1661df8 bert]# pkill -9 python
[root@7890c1661df8 bert]# ps -ef
UID      PID  PPID  C STIME TTY      TIME CMD
root       1      0  0 16:34 pts/0    00:00:00 bash
root      22      0  0 16:36 pts/1    00:00:00 bash
root     18252    22  0 16:43 pts/1   00:00:00 vim scripts/run_standalone_pretrain_ascend.sh
root     18255    22  0 16:54 pts/1   00:00:00 ps -ef
```

----结束

## 3.2 配置 Lite Cluster 网络

弹性公网IP（Elastic IP，简称EIP）提供独立的公网IP资源，包括公网IP地址与公网出口带宽服务。

购买Lite Cluster资源池后，需要使用弹性公网IP进行网络配置，配置网络后可通过公网访问Lite Cluster集群资源。

本章节介绍如何申请弹性公网IP并绑定到Lite Cluster集群。通过本文档，您可以实现弹性云服务器访问公网的目的。

### 计费影响

Lite Cluster绑定弹性公网IP后，可能产生带宽费用，详情请见[弹性公网IP计费说明](#)。

### 前提条件

- 已完成Lite Cluster集群资源购买和开通，具体请参见[2 Lite Cluster资源开通](#)。
- 已获取用于绑定的弹性公网IP，详情请见[申请弹性公网IP](#)。

### 通过绑定公网 IP 配置 Lite Cluster 网络

**步骤1** 登录[ModelArts管理控制台](#)，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”，进入“轻量算力集群（Lite Cluster）”页面。

**步骤2** 单击待配置网络的Lite Cluster名称，进入资源池详情页。

**步骤3** 在资源池详情页基本信息页签，单击CCE集群名称，进入CCE管理控制台集群管理页面。

图 3-17 Lite Cluster 资源池基本信息

基本信息	
名称	pool-
资源池ID	pool-
CCE集群	pool- <span style="border: 2px solid red; padding: 2px;">pool-</span>
描述	--
状态	<span style="color: green;">正在运行</span>
计费模式	按需计费
创建时间	2025/03/04 09:47:52 GMT+08:00
VPC	
子网	
密钥对	--

**步骤4** 找到购买Cluster资源时选择的CCE集群，单击名称进入CCE集群详情页面。

**步骤5** 左侧单击“节点管理”，切换至“节点”页签，单击需要登录的节点名称，跳转至弹性云服务器详情页面。

图 3-18 节点管理

节点池 节点 2

① 总览

Kubernetes 资源

- 工作负载
- 服务
- 存储
- 配置与密钥
- 策略
- 应用模板
- 自定义资源
- 命名空间

集群

- 节点管理 1
- 配置中心
- 集群升级 ●

② CCE Standard 运行中

③ 导出 同步全部云服务器 标签与污点管理 更多

默认按关键字搜索、过滤

节点名称	状态	所属节点...
pool	运行中 可调度	
dly-k	运行中 可调度	
dly-k	运行中 可调度	

步骤6 在弹性云服务器详情页单击“弹性公网IP”，切换至“弹性公网IP”页签。

步骤7 单击“绑定弹性公网IP”，选择未绑定的弹性公网IP，单击“确定”。

图 3-19 弹性公网 IP

pool-dly

基本信息 云硬盘 弹性网卡 安全组 弹性公网IP 监控 标签 云备份

绑定弹性公网IP 查看弹性公网IP

图 3-20 绑定弹性公网 IP



如果没有可用的弹性公网IP，需要先购买弹性公网IP，单击“购买弹性公网IP”，进入购买页。

完成购买后，返回弹性云服务器页面，刷新列表。选择刚才创建的弹性公网IP，单击“确定”。

购买弹性公网IP具体操作请参见[申请弹性公网IP](#)。

图 3-21 购买弹性公网 IP



图 3-22 绑定弹性公网 IP



**步骤8** 通过SSH方式远程访问集群资源包括两种方式，密码方式或密钥方式，二选一即可。

- 通过SSH密钥方式登录云服务器，具体操作请参见[SSH密钥登录方式](#)。
- 通过SSH密码方式登录云服务器，具体操作请参见[SSH密码登录方式](#)。

----结束

## 下一步操作

**3.3 配置kubectl工具**: kubectl是Kubernetes集群的命令行工具，配置kubectl后，您可通过kubectl命令操作Kubernetes集群。

## 3.3 配置kubectl工具

**kubectl**是Kubernetes提供的一种命令行工具，它用于与Kubernetes集群进行交互，帮助用户管理集群中的资源、查看集群状态、部署应用程序、进行调试等操作。通过kubectl，您可以方便地在命令行界面上执行集群管理任务。

配置kubectl后，您可连接到Lite Cluster资源池，通过kubectl命令操作Kubernetes集群，从而方便地管理Lite Cluster中的资源。

如果客户端需要通过kubectl连接到Lite Cluster的Kubernetes集群，可以选择两种访问方式：

- 内网访问：客户端通过内网IP地址与集群的API Server进行通信，数据流量不会经过互联网，安全性更强。
- 公网访问：集群的API Server会暴露一个公共接口，客户端可以通过互联网访问Kubernetes集群。

本文介绍如何为Lite Cluster的集群配置kubectl工具。

## 基本原理

kubectl通过kubeconfig配置文件获取集群信息，从而与Kubernetes集群的API服务器进行通信。**kubeconfig**文件是kubectl访问Kubernetes集群的身份凭证，包含集群连接信息（如API Server地址、CA证书）、用户认证凭证（如客户端证书、Token）、Context上下文配置（绑定集群、用户及默认命名空间的快捷关联）。通过这些配置信息，kubectl能够实现与Kubernetes集群的交互，并执行各种管理任务。

图 3-23 kubectl 连接集群



## 前提条件

- 已完成Lite Cluster集群资源购买和开通，具体请参见[2 Lite Cluster资源开通](#)。
- 如果通过VPC内网使用kubectl工具，请确保客户端与Lite Cluster集群在同一VPC内。

- 如果通过公网使用kubectl工具，需要提前获取用于绑定的弹性公网IP，详情请见[申请弹性公网IP](#)。

## Lite Cluster 配置 kubectl

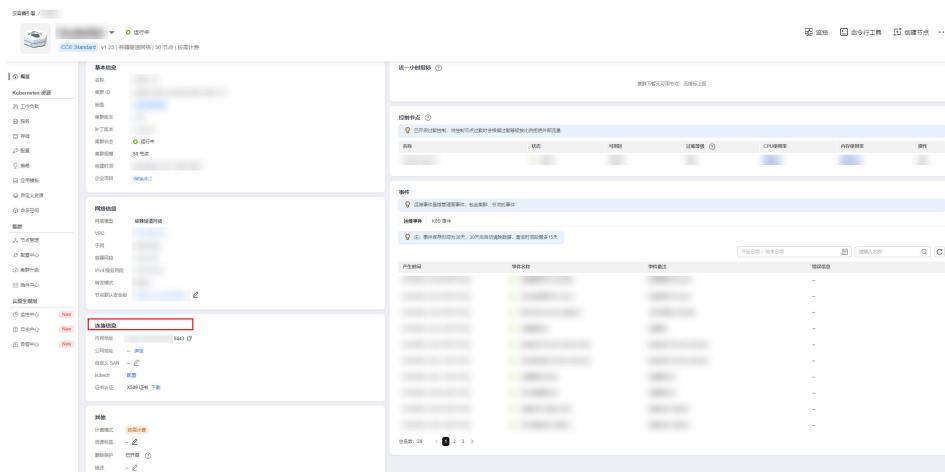
- 登录[ModelArts管理控制台](#)，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”，进入“轻量算力集群（Lite Cluster）”页面。
- 在“轻量算力集群（Lite Cluster）”页面中，单击创建的Lite Cluster专属资源池，进入资源池详情页面。

图 3-24 资源池详情-基本信息

基本信息	
名称	pool-[REDACTED]
资源池ID	pool-[REDACTED]
CCE集群	pool-[REDACTED] <span style="border: 2px solid red; padding: 2px;">pool-[REDACTED]</span>
描述	--
状态	<span style="color: green;">正在运行</span> 运行中
计费模式	按需计费
创建时间	2025/03/04 09:47:52 GMT+08:00
VPC	network-[REDACTED] [REDACTED]
子网	os-subnets-[REDACTED] [REDACTED]
密钥对	--

- 单击基本信息列中对应的“CCE集群”，进入CCE集群详情页面，在“集群信息”找到“连接信息”。

图 3-25 链接信息



#### 4. 配置kubectl工具。

- 如果通过内网使用kubectl工具，需要将kubectl工具安装在和集群在相同VPC下的某一台机器上。  
单击kubectl后的“配置”按钮，按照界面提示步骤操作即可，具体操作可参考[获取kubectl配置文件并配置kubectl](#)。

图 3-26 集群连接信息

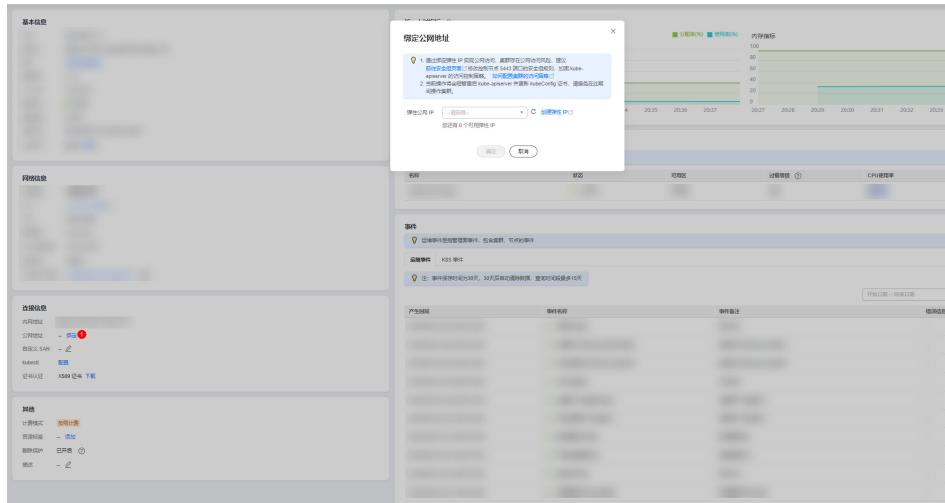


图 3-27 通过内网使用 kubectl 工具

```
[root@ ~]# kubectl get node
The connection to the server localhost:8080 was refused - did you specify the right host or port?
[root@ ~]# cd /root/
[root@ ~]# mkdir .kube
[root@ ~]# cd .kube
.kube[~]# vi config
.kube[~]# kubectl config use-context internal
Switched to context "internal".
.kube[~]# kubectl get node
NAME      STATUS    ROLES   AGE     VERSION
[redacted] Ready     <none>   14m    v1.23.9-r0-23.2.32
```

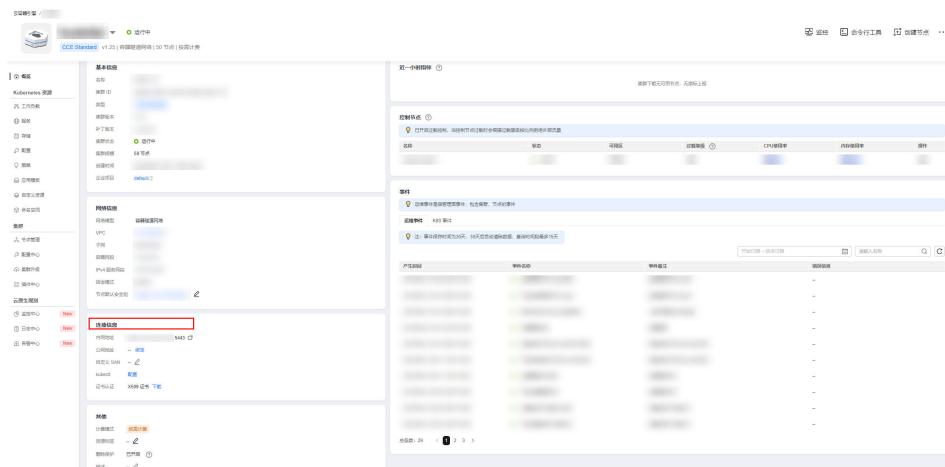
- 通过公网使用kubectl工具，可以将kubectl安装在任一台可以访问公网的客户端。首先需要绑定公网地址。
  - 单击公网地址后的“绑定”按钮。

图 3-28 绑定公网地址



- ii. 选择已有的公网IP，或者单击“创建弹性IP”跳至弹性公网IP控制台，创建新的弹性公网IP，详情请见[申请弹性公网IP](#)。
- iii. 完成公网地址绑定后，在“集群信息”找到“连接信息”，单击kubectl后的“配置”按钮。
- iv. 按照界面提示步骤操作即可，具体操作可参考[获取kubectl配置文件并配置kubectl](#)。

图 3-29 配置kubectl



5. 在安装了kubectl工具的客户端执行如下命令，显示集群节点即kubectl工具配置成功。  
`kubectl get node`

## 下一步操作

**3.4 配置Lite Cluster存储：**如果没有挂载任何外部存储，此时可用存储空间根据`dockerBaseSize`的配置来决定，可访问的存储空间比较小，因此建议通过挂载外部存储空间解决存储空间受限问题。容器中挂载存储有多种方式，不同的场景下推荐的存储方式不一样，您可根据业务实际情进行选择。

## 3.4 配置 Lite Cluster 存储

如果没有挂载任何外部存储，此时可用存储空间根据dockerBaseSize的配置来决定，可访问的存储空间比较小，因此建议通过挂载外部存储空间解决存储空间受限问题。

容器中挂载存储有多种方式，不同的场景下推荐的存储方式不一样，详情如[表3-2](#)所示。容器存储的基础知识了解请参见[存储基础知识](#)，有助您理解本章节内容。您可查看[数据盘空间分配说明](#)，了解节点数据盘空间分配的情况，以便您根据业务实际情况配置数据盘大小。

**表 3-2 容器挂载存储的方式及差异**

容器挂载存储的方式	使用场景	特点	挂载操作参考
EmptyDir	适用于训练缓存场景。	Kubernetes的临时存储卷，临时卷会遵从Pod的生命周期，与Pod一起创建和删除。	<a href="#">使用临时存储路径</a>
HostPath	适用于以下场景： 1. 容器工作负载程序生成的日志文件需要永久保存。 2. 需要访问宿主机上Docker引擎内部数据结构的容器工作负载。	节点存储。多个容器可能会共享这一个存储，会存在写冲突的问题。 Pod删除后，存储不会清理。	<a href="#">使用主机路径</a>
OBS	适用于训练数据集的存储。	对象存储。常用OBS SDK进行样本数据下载。存储量大，但是离节点比较远，直接训练速度会比较慢，通常会先将数据拉取到本地cache，然后再进行训练任务。	<ul style="list-style-type: none"><li><a href="#">静态挂载</a></li><li><a href="#">动态挂载</a></li></ul>
SFS Turbo	适用于海量小文件业务场景。	<ul style="list-style-type: none"><li>提供posix协议的文件系统；</li><li>需要和资源池在同一个VPC下或VPC互通；</li><li>价格较高。</li></ul>	<ul style="list-style-type: none"><li><a href="#">静态挂载</a></li><li>动态挂载：不支持</li></ul>

容器挂载存储的方式	使用场景	特点	挂载操作参考
SFS	适用于多读多写场景的持久化存储。	适用大容量扩展以及成本敏感型的业务场景，包括媒体处理、内容管理、大数据分析和分析工作负载程序等。 SFS容量型文件系统不适合海量小文件业务。	<ul style="list-style-type: none"><li>静态挂载</li><li>动态挂载</li></ul>
EVS	适用于需要持久化存储的场景。	每个云盘只能在单个节点挂载。 存储大小根据云硬盘的大小而定。	<ul style="list-style-type: none"><li>静态挂载</li><li>动态挂载</li></ul>

## 3.5（可选）配置驱动

当专属资源池中的节点含有GPU/Ascend资源时，为确保GPU/Ascend资源能够正常使用，需要配置好对应的驱动来满足业务需求。

Lite Cluster支持两种配置驱动的方式：

- 方式一：购买资源池时通过自定义驱动参数进行配置：**在购买资源池页面，部分GPU和Ascend规格资源池允许自定义安装驱动。开启自定义驱动开关并选择需要的驱动版本即可。
- 方式二：通过驱动升级功能对已有的资源池驱动版本进行升级：**如果在购买资源池时，没配置自定义驱动，默认驱动不满足业务要求，可通过驱动升级功能将驱动升级到指定版本。

### 方式一：购买资源池时通过自定义驱动参数进行配置

- 登录[ModelArts管理控制台](#)，在左侧菜单栏中选择“资源管理 > 标准算力集群(Standard Cluster)”。
- 在“标准算力集群(Standard Cluster)”页面，单击“购买标准算力集群”，进入购买标准算力集群界面填写参数。  
部分GPU和Ascend规格资源池允许自定义安装驱动。配置资源调度与切分时，打开“自定义驱动”开关，在“GPU驱动/Ascend驱动”选择对应GPU/Ascend驱动。gpu-driver配套版本请参考[1.3 不同机型对应的软件配套版本](#)。

图 3-30 GPU/Ascend 驱动



更多参数说明请参考[2 Lite Cluster资源开通](#)。

3. 单击“立即购买”确认规格。产品规格和协议许可确认无误后，单击“提交”，即可创建Lite Cluster资源池。

## 方式二：通过驱动升级功能对已有的资源池驱动版本进行升级

如果在购买资源池时，没配置自定义驱动，默认驱动不满足业务要求，可通过驱动升级功能将驱动升级到指定版本。

- Lite Cluster资源池状态处于运行中，且专属池中的节点需要含有GPU/Ascend资源。
- 升级需要重启节点，建议在低高峰期进行，以避免影响正在运行的任务，可前往资源池详情页“节点管理”页面查看节点资源占用情况。

### ⚠ 警告

升级驱动会重启节点。如果主机进行过差异化配置，重启节点可能会导致配置丢失，需谨慎考虑。

1. 登录[ModelArts管理控制台](#)，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”。在资源池列表中，选择需要进行驱动升级的资源池“... > 驱动升级”。  
或者在资源池列表单击资源池名称，进入资源池详情页，切换至“节点池管理”页签，单击节点池操作列“更多>驱动升级”。
2. 在“驱动升级”弹窗中，会显示当前Lite Cluster资源池的驱动类型、实例数、当前版本、目标版本、升级方式、升级范围和开启滚动开关。  
按[表5-7](#)设置驱动升级参数。
3. 设置完成后，单击“确定”开始升级驱动。

在资源池列表中，选择目标资源池，单击操作列中的“...”，然后选择“驱动升级”。在弹出的“驱动升级”页面中，查看当前版本和目标版本是否一致。如果一致，说明驱动已成功升级。

驱动升级更多介绍可参考[5.6 升级Lite Cluster资源池驱动](#)。

## 下一步操作

**3.6（可选）配置镜像预热：**Lite Cluster资源池支持镜像预热功能，镜像预热可实现将镜像提前在资源池节点上拉取好，在推理及大规模分布式训练时有效缩短镜像拉取时间。

## 3.6（可选）配置镜像预热

镜像预热是指在计算节点上提前加载所需的镜像，主要目的是提高镜像加载效率，减少训练作业启动时间。

Lite Cluster资源池支持镜像预热功能，提前在资源池节点上拉取镜像，在推理及大规模分布式训练时有效缩短镜像拉取时间。

本文将介绍如何在Lite Cluster配置镜像预热功能。

### 前提条件

- 已完成Lite Cluster集群资源购买和开通，具体请参见[2 Lite Cluster资源开通](#)。
- 镜像预热的镜像来源需要获取依赖服务SWR服务管理列表，需要将SWR服务操作权限委托给ModelArts服务，让ModelArts以您的身份使用依赖服务，代替您进行一些资源操作。详细操作参见[使用委托授权](#)。
- 镜像预热如果使用自定义镜像，需要将制作的自定义镜像需要上传至容器镜像服务（Software Repository for Container，SWR），详情请见[推送镜像到镜像仓库](#)。

### Lite Cluster 配置镜像预热

- 登录[ModelArts管理控制台](#)，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”，进入“轻量算力集群（Lite Cluster）”页面。
- 单击某个资源池名称，进入资源池详情。
- 单击左侧“配置管理”，进入资源池配置管理页面。

图 3-31 配置管理



4. 在镜像预热中单击编辑图标 ，填写镜像预热信息。

表 3-3 镜像预热参数

参数名称	说明
镜像来源	可选择“预置”或“自定义”的镜像。 <ul style="list-style-type: none"><li>• 预置：可选择SWR服务上自有的或他人共享的镜像。</li><li>• 自定义：可直接填写镜像地址。 需要提前将制作的自定义镜像需要上传至SWR服务，详情请见<a href="#">推送镜像到镜像仓库</a>。</li></ul>

参数名称	说明
添加镜像密钥	<p>如果本租户不具有预热镜像的权限（即非公开/非本租户私有/非他人共享的镜像），此时需要添加镜像密钥。在开启镜像密钥开关后，选择命名空间及对应密钥。创建密钥方法可参考<a href="#">创建密钥</a>，密钥类型须为kubernetes.io/dockerconfigjson类型。</p> <p>创建密钥所需的仓库地址、用户名、密码、可以参考对应租户的SWR登录指令。<a href="#">图3-35</a>中为临时登录指令，如果需长期有效登录指令，可单击图中的“如何获取长期有效指令”链接获取指导。</p> <p>如果需添加多个密钥，可以单击“+”新增密钥数。</p>
添加镜像预热配置	如果需添加多个镜像，可单击此按键。

图 3-32 预置镜像预热



图 3-33 预置镜像选择

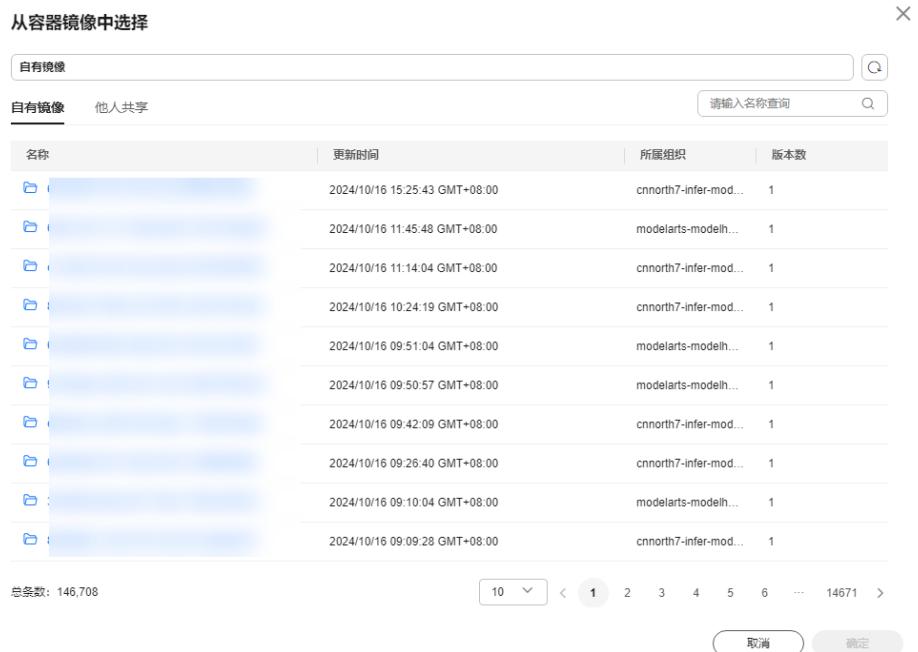


图 3-34 自定义镜像预热



图 3-35 登录指令



5. 单击“确定”后，在预热信息框中可以看到已成功预热的镜像信息。  
如果镜像预热失败，请检查镜像地址以及密钥是否正确。

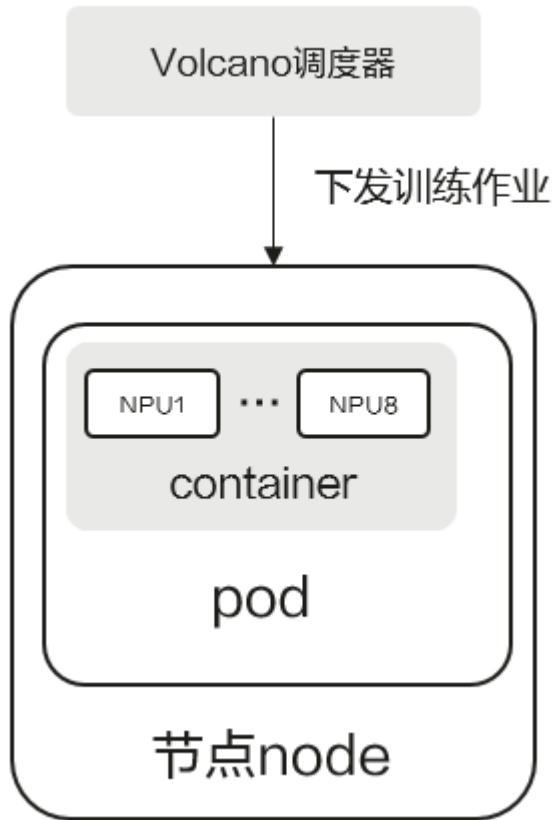
# 4 Lite Cluster 资源使用

## 4.1 在 Lite Cluster 资源池上使用 Snt9B 完成分布式训练任务

### 场景描述

本案例介绍如何在Snt9B上进行分布式训练任务，其中Cluster资源池已经默认安装volcano调度器，训练任务默认使用volcano job形式下发lite池集群。训练测试用例使用NLP的bert模型。

图 4-1 任务示意图



## 操作步骤

**步骤1** 拉取镜像。本测试镜像为bert\_pretrain\_mindspore:v1，已经把测试数据和代码打进镜像中。

```
docker pull swr.cn-southwest-2.myhuaweicloud.com/os-public-repo/bert_pretrain_mindspore:v1
docker tag swr.cn-southwest-2.myhuaweicloud.com/os-public-repo/bert_pretrain_mindspore:v1
bert_pretrain_mindspore:v1
```

**步骤2** 在主机上新建config.yaml文件。

config.yaml文件用于配置pod，本示例中使用sleep命令启动pod，便于进入pod调试。您也可以修改command为对应的任务启动命令（如“python train.py”），任务会在启动容器后执行。

config.yaml内容如下：

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: configmap1980-yourvcjobname  # 前缀使用“configmap1980-”不变，后接vcjob的名字
  namespace: default                # 命名空间自选，需要和下边的vcjob处在同一命名空间
  labels:
    ring-controller.cce: ascend-1980 # 保持不动
data:                                #data内容保持不动，初始化完成，会被volcano插件自动修改
  jobstart_hccl.json: |
    {
      "status": "initializing"
    }
---
```

```
apiVersion: batch.volcano.sh/v1alpha1 # The value cannot be changed. The volcano API must be used.
kind: Job # Only the job type is supported at present.
metadata:
  name: yourvcjobname # job名字, 需要和configmap中名字保持一致
  namespace: default # 和configmap保持一致
  labels:
    ring-controller.cce: ascend-1980 # 保持不动
    fault-scheduling: "force"
spec:
  minAvailable: 1 # The value of minAvailable is 1 in a single-node scenario and N in an N-
  node distributed scenario.
  schedulerName: volcano # 保持不动, Use the Volcano scheduler to schedule jobs.
  policies:
    - event: PodEvicted
      action: RestartJob
  plugins:
    configmap1980:
      - --rank-table-version=v2 # 保持不动, 生成v2版本ranktablefile
    env: []
    svc:
      - --publish-not-ready-addresses=true
  maxRetry: 3
  queue: default
  tasks:
    - name: "yourvcjobname-1"
      replicas: 1 # The value of replicas is 1 in a single-node scenario and N in an N-node
      scenario. The number of NPUs in the requests field is 8 in an N-node scenario.
      template:
        metadata:
          labels:
            app: mindspore
            ring-controller.cce: ascend-1980 # 保持不动, The value must be the same as the label in ConfigMap
        and cannot be changed.
        spec:
          affinity:
            podAntiAffinity:
              requiredDuringSchedulingIgnoredDuringExecution:
                - labelSelector:
                    matchExpressions:
                      - key: volcano.sh/job-name
                        operator: In
                        values:
                          - yourvcjobname
              topologyKey: kubernetes.io/hostname
            containers:
              - image: bert_pretrain_mindspore:v1 # 镜像地址, Training framework image, which can be
              modified.
                imagePullPolicy: IfNotPresent
                name: mindspore
                env:
                  - name: name # The value must be the same as that of Jobname.
                    valueFrom:
                      fieldRef:
                        fieldPath: metadata.name
                  - name: ip # IP address of the physical node, which is used to identify the
                  node where the pod is running
                    valueFrom:
                      fieldRef:
                        fieldPath: status.hostIP
                  - name: framework
                    value: "MindSpore"
                command:
                  - "sleep"
                  - "100000000000000000000000"
                resources:
                  requests:
                    huawei.com/ascend-1980: "1" # 需求卡数, key保持不变。Number of required NPUs.
                    The maximum value is 16. You can add lines below to configure resources such as memory and CPU.
                limits:
```

```
        huawei.com/ascend-1980: "1"          # 限制卡数, key保持不变。The value must be consistent
with that in requests.

        volumeMounts:
        - name: ascend-driver      #驱动挂载, 保持不动
          mountPath: /usr/local/Ascend/driver
        - name: ascend-add-ons     #驱动挂载, 保持不动
          mountPath: /usr/local/Ascend/add-ons
        - name: localtime
          mountPath: /etc/localtime
        - name: hccn                #驱动hccn配置, 保持不动
          mountPath: /etc/hccn.conf
        - name: npu-smi            #npu-smi
          mountPath: /usr/local/sbin/npu-smi

        nodeSelector:
        accelerator/huawei-npu: ascend-1980

        volumes:
        - name: ascend-driver
          hostPath:
            path: /usr/local/Ascend/driver
        - name: ascend-add-ons
          hostPath:
            path: /usr/local/Ascend/add-ons
        - name: localtime
          hostPath:
            path: /etc/localtime          # Configure the Docker time.
        - name: hccn
          hostPath:
            path: /etc/hccn.conf
        - name: npu-smi
          hostPath:
            path: /usr/local/sbin/npu-smi

        restartPolicy: OnFailure
```

**步骤3** 根据config.yaml创建pod。

```
kubectl apply -f config.yaml
```

**步骤4** 检查pod启动情况，执行下述命令。如果显示“1/1 running”状态代表启动成功。

```
kubectl get pod -A
```

**步骤5** 进入容器，{pod\_name}替换为您的pod名字（get pod中显示的名字），{namespace}替换为您的命名空间（默认为default）。

```
kubectl exec -it {pod_name} bash -n {namespace}
```

**步骤6** 查看卡信息，执行以下命令。

```
npu-smi info
```

kubernetes会根据config.yaml文件中配置的卡数分配资源给pod，如下图所示由于配置了1卡因此在容器中只会显示1卡，说明配置生效。

**图 4-2** 查看卡信息

```
[root@louleilei-louleilei-1-0 ma-user]# npu-smi info
+-----+-----+-----+-----+-----+
| npu-smi 23.0.rc2           Version: 23.0.rc2.2.b030 |
+-----+-----+-----+-----+-----+
| NPU   Name      | Health    | Power(W) | Temp(C) | Hugepages-Usage(page) |
| Chip   | Bus-Id    |          |          |          |          |
+-----+-----+-----+-----+-----+
| 0     910B1    | OK       | 93.1    | 48      | 0 / 0      |
| 0     | 0000:C1:00.0 | 0        | 0        | 4313 / 65536 |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
| NPU   Chip      | Process id | Process name          | Process memory(MB) |
+-----+-----+-----+-----+
| No running processes found in NPU 0 |
+-----+-----+-----+-----+
```

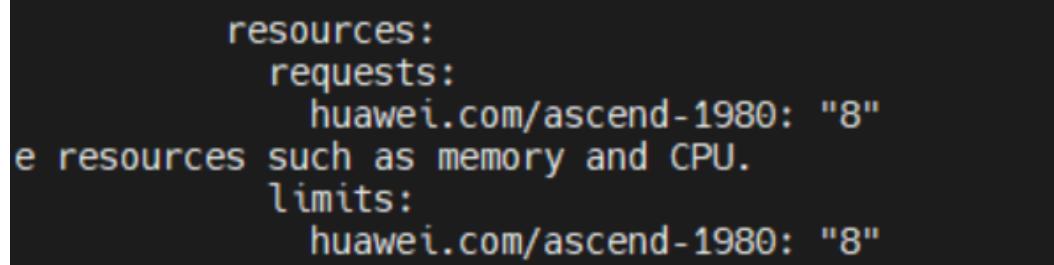
**步骤7** 修改pod的卡数。由于本案例中为分布式训练，因此所需卡数修改为8卡。

删除已创建的pod。

```
kubectl delete -f config.yaml
```

将config.yaml文件中“limit”和“request”改为8。  
vi config.yaml

图 4-3 修改卡数



重新创建pod。

```
kubectl apply -f config.yaml
```

进入容器并查看卡信息，{pod\_name}替换为您的pod名字，{namespace}替换为您的命名空间（默认为default）。

```
kubectl exec -it {pod_name} bash -n {namespace}  
npu-smi info
```

如图所示为8卡，pod配置成功。

图 4-4 查看卡信息

[root@os-node-created-ljknq ~]# kubectl get pod					
NAME	READY	STATUS	RESTARTS	AGE	
maos-node-agent-gqrvs	2/2	Running	32 (3d2h ago)	3d4h	
yourvcjobname-yourvcjobname-1-0 1/1 Running 0 52s					
[root@os-node-created-ljknq ~]# kubectl exec -it yourvcjobname-yourvcjobname-1-0 bash -n default					
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.					
[root@yourvcjobname-yourvcjobname-1-0 ma-user]# npu-smi info					
+-----+-----+-----+-----+-----+-----+					
NPU	23.0.rc2	Version:	23.0.rc2.2		
NPU	Name	Health	Power(W)	Temp(C)	Hugepages-Usage(page)
Chip		Bus-Id	AICore(%)	Memory-Usage(MB)	HBM-Usage(MB)
+-----+-----+-----+-----+-----+-----+					
0	910B4	OK	83.7	48	0 / 0
0	0000:C1:00.0	0	0	/ 0	3151 / 32768
+-----+-----+-----+-----+-----+-----+					
1	910B4	OK	83.8	48	0 / 0
0	0000:01:00.0	0	0	/ 0	3148 / 32768
+-----+-----+-----+-----+-----+-----+					
2	910B4	OK	83.8	45	0 / 0
0	0000:C2:00.0	0	0	/ 0	3149 / 32768
+-----+-----+-----+-----+-----+-----+					
3	910B4	OK	87.6	48	0 / 0
0	0000:02:00.0	0	0	/ 0	3147 / 32768
+-----+-----+-----+-----+-----+-----+					
4	910B4	OK	83.8	45	0 / 0
0	0000:81:00.0	0	0	/ 0	3148 / 32768
+-----+-----+-----+-----+-----+-----+					
5	910B4	OK	83.8	46	0 / 0
0	0000:41:00.0	0	0	/ 0	3148 / 32768
+-----+-----+-----+-----+-----+-----+					
6	910B4	OK	83.7	46	0 / 0
0	0000:82:00.0	0	0	/ 0	3147 / 32768
+-----+-----+-----+-----+-----+-----+					
7	910B4	OK	92.6	49	0 / 0
0	0000:42:00.0	0	0	/ 0	3148 / 32768
+-----+-----+-----+-----+-----+-----+					
NPU	Chip	Process Id	Process name	Process memory(MB)	
+-----+-----+-----+-----+-----+-----+					

### 步骤8 查看卡间通信配置文件，执行以下命令。

```
cat /user/config/jobstart_hccl.json
```

多卡训练时，需要依赖“rank\_table\_file”做卡间通信的配置文件，该文件自动生成，pod启动之后文件地址。为“/user/config/jobstart\_hccl.json”，“/user/config/jobstart\_hccl.json”配置文件生成需要一段时间，业务进程需要等待“/user/config/jobstart\_hccl.json”中“status”字段为“completed”状态，才能生成卡间通信信息。如下图所示。

**图 4-5 卡间通信配置文件**

```
[root@louleilei-louleilei-1-0 ma-user]# cat /user/config/jobstart_hccl.json
{"status": "completed", "version": "1.0", "server_count": "1", "server_list": [{"server_id": "192.168.229.117", "device": [{"device_id": "0", "device_ip": "29.20.124.238", "rank_id": "0"}, {"device_id": "1", "device_ip": "29.20.191.49", "rank_id": "1"}, {"device_id": "2", "device_ip": "29.20.176.195", "rank_id": "2"}, {"device_id": "3", "device_ip": "29.20.47.177", "rank_id": "3"}, {"device_id": "4", "device_ip": "29.20.152.143", "rank_id": "4"}, {"device_id": "5", "device_ip": "29.20.24.24", "rank_id": "5"}, {"device_id": "6", "device_ip": "29.20.141.103", "rank_id": "6"}, {"device_id": "7", "device_ip": "29.20.109.253", "rank_id": "7"}]}][root@louleilei-louleilei-1-0 ma-user]#
```

**步骤9 启动训练任务。**

```
cd /home/ma-user/modelarts/user-job-dir/code/bert/
export MS_ENABLE_GE=1
export MS_GE_TRAIN=1
python scripts/ascend_distributed_launcher/get_distribute_pretrain_cmd.py --run_script_dir ./scripts/
run_distributed_pretrain_ascend.sh --hyper_parameter_config_dir ./scripts/ascend_distributed_launcher/
hyper_parameter_config.ini --data_dir /home/ma-user/modelarts/user-job-dir/data/cn-news-128-1f-mind/ --
hccl_config /user/config/jobstart_hccl.json --cmd_file ./distributed_cmd.sh
bash scripts/run_distributed_pretrain_ascend.sh /home/ma-user/modelarts/user-job-dir/data/cn-
news-128-1f-mind/ /user/config/jobstart_hccl.json
```

**图 4-6 启动训练任务**

```
[root@yourvcjobname-yourvcjobname-1-0 bert]# export MS_ENABLE_GE=1
[root@yourvcjobname-yourvcjobname-1-0 bert]# export MS_GE_TRAIN=1
[root@yourvcjobname-yourvcjobname-1-0 bert]# python scripts/ascend_distributed_launcher/get_distribute_pretrain_cmd.py
--run_script_dir ./scripts/run_distributed_pretrain_ascend.sh --hyper_parameter_config_dir ./scripts/ascend_distributed_
launcher/hyper_parameter_config.ini --data_dir /home/ma-user/modelarts/user-job-dir/data/cn-news-128-1f-mind/ --hccl_c
onfig /user/config/jobstart_hccl.json --cmd_file ./distributed_cmd.sh
start scripts/ascend_distributed_launcher/get_distribute_pretrain_cmd.py
hccl_config_dir: /user/config/jobstart_hccl.json
hccl_time_out: 120
the number of logical core: 192
total rank size: 8
this server rank size: 8
avg_core_per_rank: 24

start training for rank 0, device 0:
rank_id: 0
device_id: 0
logic_id 0
core_nums: 0-23
epoch_size: 40
data_dir: /home/ma-user/modelarts/user-job-dir/data/cn-news-128-1f-mind/
log_file_dir: /home/ma-user/modelarts/user-job-dir/code/bert/L0G0/pretraining_log.txt

start training for rank 1, device 1:
rank_id: 1
device_id: 1
logic_id 1
core_nums: 24-47
epoch_size: 40
data_dir: /home/ma-user/modelarts/user-job-dir/data/cn-news-128-1f-mind/
log_file_dir: /home/ma-user/modelarts/user-job-dir/code/bert/L0G1/pretraining_log.txt

start training for rank 2, device 2:
rank_id: 2
device_id: 2
logic_id 2
core_nums: 48-71
epoch_size: 40
data_dir: /home/ma-user/modelarts/user-job-dir/data/cn-news-128-1f-mind/
log_file_dir: /home/ma-user/modelarts/user-job-dir/code/bert/L0G2/pretraining_log.txt
```

训练任务加载需要一定时间，在等待若干分钟后，可以执行下述命令查看卡信息。如下图可见，8张卡均被占用，说明训练任务在进行中

```
npu-smi info
```

图 4-7 查看卡信息

[root@yourvcjobname-yourvcjobname-1-0 bert]# npu-smi info						
npu-smi 23.0.rc2 Version: 23.0.rc2.2						
NPU	Name	Health	Power(W)	Temp(C)	Hugepages-Usage(page)	HBM-Usage(MB)
0	910B4	OK	220.1	55	0 / 0	
0		0000:c1:00.0	46	0 / 0	18763 / 32768	
1	910B4	OK	205.5	56	0 / 0	
0		0000:01:00.0	19	0 / 0	18761 / 32768	
2	910B4	OK	212.4	53	0 / 0	
0		0000:c2:00.0	36	0 / 0	18762 / 32768	
3	910B4	OK	233.6	55	0 / 0	
0		0000:02:00.0	48	0 / 0	18761 / 32768	
4	910B4	OK	221.7	51	0 / 0	
0		0000:01:00.0	47	0 / 0	18762 / 32768	
5	910B4	OK	200.9	55	0 / 0	
0		0000:41:00.0	13	0 / 0	18762 / 32768	
6	910B4	OK	219.5	53	0 / 0	
0		0000:02:00.0	33	0 / 0	18761 / 32768	
7	910B4	OK	220.7	58	0 / 0	
0		0000:42:00.0	47	0 / 0	18762 / 32768	
NPU	Chip	Process id	Process name	Process memory(MB)		
0	0	39	python	15453		
1	0	45	python	15453		
2	0	51	python	15453		
3	0	57	python	15453		
4	0	63	python	15453		
5	0	69	python	15453		
6	0	75	python	15452		
7	0	81	python	15453		

如果想停止训练任务，可执行下述命令关闭进程，查询进程后显示已无运行中python进程。

```
pkkill -9 python  
ps -ef
```

图 4-8 关闭训练进程

[root@7890c1661df8 bert]# pkill -9 python						
[root@7890c1661df8 bert]# ps -ef						
UID	PID	PPID	C STIME TTY	TIME	CMD	
root	1	0	0 16:34 pts/0	00:00:00	bash	
root	22	0	0 16:36 pts/1	00:00:00	bash	
root	18252	22	0 16:43 pts/1	00:00:00	vim scripts/run_standalone_pretrain_ascend.sh	
root	18255	22	0 16:54 pts/1	00:00:00	ps -ef	

## 说明

limit/request配置cpu和内存大小，已知单节点Snt9B机器为：8张Snt9B卡+192核1536GB，请合理规划，避免cpu和内存限制过小引起任务无法正常运行。

## 结束

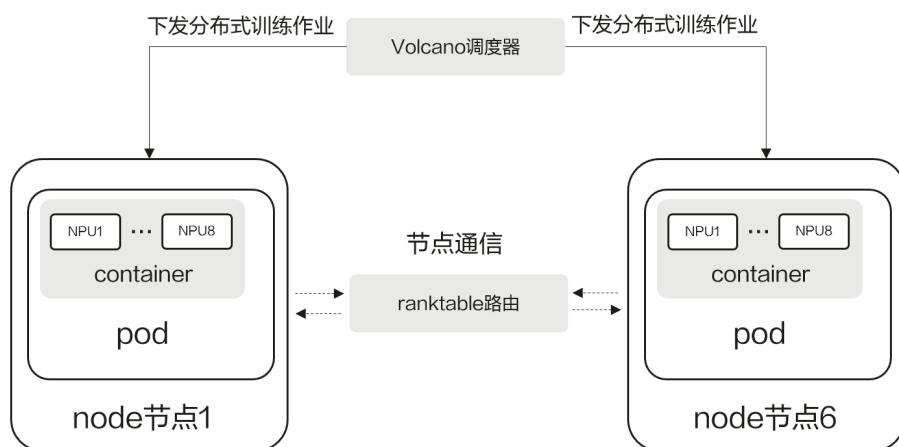
## 4.2 在 Lite Cluster 资源池上使用 ranktable 路由规划完成 PyTorch NPU 分布式训练

### 场景描述

ranktable路由规划是一种用于分布式并行训练中的通信优化能力，在使用NPU的场景下，支持对节点之间的通信路径根据交换机实际topo做网络路由亲和规划，进而提升节点之间的通信速度。

本案例介绍如何在ModelArts Lite场景下使用ranktable路由规划完成PyTorch NPU分布式训练任务，训练任务默认使用Volcano job形式下发到Lite资源池集群。

图 4-9 任务示意图



### 约束与限制

- 该功能只支持贵阳一区域，如果要在其他区域使用请联系技术支持。
- ModelArts Lite资源池对应的CCE集群需要安装1.10.12及以上版本的版Volcano插件。Volcano调度器的安装升级请参见[Volcano调度器](#)。仅华为云版Volcano插件支持开启路由加速特性。
- 训练使用的Python版本是3.7或3.9，否则无法实现ranktable路由加速。
- 训练作业的任务节点数要大于或等于3，否则会跳过ranktable路由加速。建议在大模型场景（512卡及以上）使用ranktable路由加速。
- 脚本执行目录不能是共享目录，否则ranktable路由加速会失败。
- 路由加速的原理是改变rank编号，所以代码中对rank的使用要统一，如果rank的使用不一致会导致训练异常。

### 操作步骤

#### 步骤1 开启ModelArts Lite资源池对应的CCE集群的cabinet插件。

- 在ModelArts Lite专属资源池列表，单击资源池名称，进入专属资源池详情页面。

2. 在基本信息页面单击CCE集群，跳转到CCE集群详情页面。
3. 在左侧导航栏选择“插件市场”，搜索“Volcano调度器”。
4. 单击“编辑”，查看高级配置的“plugins”参数下是否有`{"name":"cabinet"}`。
  - 是，则执行[步骤2](#)。
  - 否，则在高级配置的“plugins”参数下添加`{"name":"cabinet"}`，单击下方的“安装”使Volcano调度器更新配置，完成滚动重启。

### 步骤2 修改torch\_npu训练启动脚本。

#### 须知

脚本要使用`torch.distributed.launch/run`命令启动，不能使用`mp.spawn`命令启动，否则无法实现ranktable路由加速。

在使用PyTorch训练时，需要将“`RANK_AFTER_ACC`”环境变量赋值给“`NODE_RANK`”，使得ranktable路由规划生效。训练启动脚本（`xxxx_train.sh`）示例如下。其中“`MASTER_ADDR`”和“`NODE_RANK`”必须保持该赋值。

```
#!/bin/bash

# MASTER_ADDR
MASTER_ADDR="${MA_VJ_NAME}-${MA_TASK_NAME}-${MA_MASTER_INDEX:-0}.${MA_VJ_NAME}"
NODE_RANK="${RANK_AFTER_ACC:-$VC_TASK_INDEX}"
NNODES="$MA_NUM_HOSTS"
NGPUS_PER_NODE="$MA_NUM_GPUS"
# self-define, it can be changed to >=10000 port
MASTER_PORT="39888"

# replace ${MA_JOB_DIR}/code/torch_ddp.py to the actual training script
PYTHON_SCRIPT=${MA_JOB_DIR}/code/torch_ddp.py
PYTHON_ARGS=""

# set hccl timeout time in seconds
export HCCL_CONNECT_TIMEOUT=1800

# replace ${ANACONDA_DIR}/envs/${ENV_NAME}/bin/python to the actual python
CMD="${ANACONDA_DIR}/envs/${ENV_NAME}/bin/python -m torch.distributed.launch \
--nnode=$NNODES \
--node_rank=$NODE_RANK \
--nproc_per_node=$NGPUS_PER_NODE \
--master_addr $MASTER_ADDR \
--master_port=$MASTER_PORT \
$PYTHON_SCRIPT \
$PYTHON_ARGS
"
echo $CMD
$CMD
```

### 步骤3 在主机上新建“config.yaml”文件。

“`config.yaml`”文件用于配置pod，代码示例如下。代码中的“`xxxx_train.sh`”即为[步骤2](#)修改的训练启动脚本。

```
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
metadata:
  name: yourvcjobname      # job名字，根据实际场景修改
  namespace: default        # 命名空间，根据实际场景修改
  labels:
    ring-controller.cce: ascend-1980 # 保持不动
```

```
        fault-scheduling: "force"
spec:
  minAvailable: 6          # 节点数, 根据实际场景修改, 对应分布式训练使用的节点数
  schedulerName: volcano    # 保持不动
  policies:
    - event: PodEvicted
      action: RestartJob
  plugins:
    configmap1980:
      - --rank-table-version=v2      # 保持不动, 生成v2版本ranktablefile
  env: []
  svc:
    - --publish-not-ready-addresses=true # 保持不动, pod间互相通信使用及生成一些必要环境变量
  maxRetry: 1
  queue: default
  tasks:
    - name: "worker" # 保持不动
      replicas: 6           # 任务数, 对于pytorch而言就是节点数, 与minAvailable一致即可
      template:
        metadata:
          annotations:
            cabinet: "cabinet" # 保持不动, 开启tor-topo下发的开关
          labels:
            app: pytorch-npu # 标签, 根据实际场景修改
            ring-controller.cce: ascend-1980 # 保持不动
        spec:
          affinity:
            podAntiAffinity:
              requiredDuringSchedulingIgnoredDuringExecution:
                - labelSelector:
                    matchExpressions:
                      - key: volcano.sh/job-name
                        operator: In
                        values:
                          - yourvcjobname # job名字, 根据实际场景修改
              topologyKey: kubernetes.io/hostname
        containers:
          - image: swr.xxxxxx.com/xxxx/custom_pytorch_npu:v1      # 镜像地址, 根据实际场景修改
            imagePullPolicy: IfNotPresent
            name: pytorch-npu      # 容器名称, 根据实际场景修改
            env:
              - name: OPEN_SCRIPT_ADDRESS # 开放脚本地址, 其中region-id根据实际region修改, 例如cn-southwest-2
                value: "https://mtest-bucket.obs.{region-id}.myhuaweicloud.com/acc/rank"
              - name: NAME
                valueFrom:
                  fieldRef:
                    fieldPath: metadata.name
              - name: MA_CURRENT_HOST_IP          # 保持不动, 表示运行时当前pod所在节点的ip
                valueFrom:
                  fieldRef:
                    fieldPath: status.hostIP
              - name: MA_NUM_GPUS    # 每个pod使用的NPU卡数, 根据实际场景修改
                value: "8"
              - name: MA_NUM_HOSTS   # 参与分布式训练的节点数, 与minAvailable一致即可
                value: "6"
              - name: MA_VJ_NAME # volcano job名称
                valueFrom:
                  fieldRef:
                    fieldPath: metadata.annotations['volcano.sh/job-name']
              - name: MA_TASK_NAME #任务pod名称
                valueFrom:
                  fieldRef:
                    fieldPath: metadata.annotations['volcano.sh/task-spec']
            command:
              - /bin/bash
              - -c
              - "wget ${OPEN_SCRIPT_ADDRESS}/bootstrap.sh -q && bash bootstrap.sh; export RANK_AFTER_ACC=${VC_TASK_INDEX}; rank_acc=$(cat /tmp/RANK_AFTER_ACC 2>/dev/null); [ -n \"$"

```

```
{rank_acc}\]" ] && export RANK_AFTER_ACC=${rank_acc};export MA_MASTER_INDEX=$(cat /tmp/MASTER_INDEX 2>/dev/null || echo 0); bash xxxx_train.sh" # xxxx_train.sh换成实际训练脚本路径
resources:
  requests:
    huawei.com/ascend-1980: "8"          # 每个节点的需求卡数, key保持不变。与MA_NUM_GPUS一致
  limits:
    huawei.com/ascend-1980: "8"          # 每个节点的限制卡数, key保持不变。与MA_NUM_GPUS一致
  volumeMounts:
    - name: ascend-driver      #驱动挂载, 保持不动
      mountPath: /usr/local/Ascend/driver
    - name: ascend-add-ons     #驱动挂载, 保持不动
      mountPath: /usr/local/Ascend/add-ons
    - name: localtime
      mountPath: /etc/localtime
    - name: hccn                # 驱动hccn配置, 保持不动
      mountPath: /etc/hccn.conf
    - name: npu-smi
      mountPath: /usr/local/sbin/npu-smi
  nodeSelector:
    accelerator/huawei-npu: ascend-1980
  volumes:
    - name: ascend-driver
      hostPath:
        path: /usr/local/Ascend/driver
    - name: ascend-add-ons
      hostPath:
        path: /usr/local/Ascend/add-ons
    - name: localtime
      hostPath:
        path: /etc/localtime
    - name: hccn
      hostPath:
        path: /etc/hccn.conf
    - name: npu-smi
      hostPath:
        path: /usr/local/sbin/npu-smi
  restartPolicy: OnFailure
```

**步骤4** 执行如下命令，根据“config.yaml”创建并启动pod。容器启动后会自动执行训练作业。

```
kubectl apply -f config.yaml
```

**步骤5** 执行如下命令，检查pod启动情况。如果显示“1/1 running”状态代表启动成功。

```
kubectl get pod
```

**图 4-10 启动成功的回显**

```
| yourvcjobname1-worker-0 1/1 Running 0 110m |
```

**步骤6** 执行如下命令，查看日志。日志显示如图所示表示成功执行动态路由。

```
kubectl logs {pod-name}
```

其中{pod-name}替换为实际pod名称，可以在**步骤5**的返回信息中获取。

图 4-11 成功执行动态路由的回显

```
2024-01-30 19:45:21,397 INFO: Wait for Topo file ready
2024-01-30 19:45:21,401 INFO: Wait for Rank table file ready
2024-01-30 19:45:21,401 INFO: Rank table file [REDACTED] jobstart_hecl.json (K8S generated) is ready for read
2024-01-30 19:45:21,402 INFO: Rank table file [REDACTED] jobstart_hecl.json (K8S generated) is old format.convert it to new format start...
2024-01-30 19:45:21,402 INFO: Rank table file (V1) is generated
2024-01-30 19:45:21,402 INFO: Route plan begins. Current server 19 [REDACTED] 110
2024-01-30 19:45:21,410 INFO: Load in rank_file success. rank_file [REDACTED] jobstart_hecl.json
2024-01-30 19:45:21,410 INFO: route plan algorithm version 2
2024-01-30 19:45:21,414 INFO: save ranktable to file [REDACTED] jobstart_routeplan.json
2024-01-30 19:45:21,415 INFO: Route plan ends. Route plan acceleration True
2024-01-30 19:45:21,419 INFO: Route plan acc success, custom_dev is [[{"id": "16", "rank": 0}, {"id": "17", "rank": 1}, {"id": "18", "rank": 2}, {"id": "19", "rank": 3}, {"id": "20", "rank": 4}, {"id": "21", "rank": 5}, {"id": "22", "rank": 6}, {"id": "23", "rank": 7}], custom_id is 19 [REDACTED].2, current_node_rank is 2
```

## 说明

- 只有任务节点大于等于3的训练任务才能成功执行动态路由。
- 如果执行失败可以参考[故障排除：ranktable路由优化执行失败](#)处理。

----结束

## 故障排除：ranktable 路由优化执行失败

### 故障现象

容器日志有error信息。

### 可能原因

集群节点没有下发topo文件和ranktable文件。

### 操作步骤

- 在ModelArts Lite专属资源池列表，单击资源池名称，进入专属资源池详情页面。
- 在基本信息页面单击CCE集群，跳转到CCE集群详情页面。
- 在CCE集群详情页，选择左侧导航栏的“节点管理”，选择“节点”页签。
- 在节点列表，单击操作列的“更多 > 查看YAML”查看节点配置信息。
- 查看节点的yaml文件里“cce.kubectl.kubernetes.io/ascend-rank-table”字段是否有值。

如图所示，表示有值，节点已开启topo文件和ranktable文件的下发。否则，联系技术支持处理。

图 4-12 查看节点的 yaml 文件

查看YAML

当前数据

YAML JSON 换行 全屏

```
18:     modelarts.authoring.note
19:     node.cce.io/billing-mode
20:     node.kubernetes.io/baremetal
21:     node.kubernetes.io/controllable
22:     node.kubernetes.io/instance-type
23:     node.kubernetes.io/provider
24:     os.architecture: aarch64
25:     os.modelarts.node.os.name: "Ubuntu"
26:     os.modelarts.node.os.version: "4.19.90-0-musl"
27:     os.name: "Ubuntu 20.04.1"
28:     os.version: "4.19.90-0-musl"
29:     resource.category: ASCEND
30:     servertype: Ascend910B-2
31:     topology.kubernetes.io/r
32:     topology.kubernetes.io/z
33:   annotations:
34:     alibaba.kubernetes.io/provided-node-ip: "192.168.1.150"
35:     cce.kubectl.kubernetes.io/ascend-rank-table: "[{"device": [
36:       {"device_ip": "29.20.13.6", "device_id": "213.6"},  

37:       {"device_ip": "29.20.13.7", "device_id": "213.7"},  

38:       {"device_ip": "29.20.213.6", "device_id": "213.6"},  

39:       {"device_ip": "29.20.213.7", "device_id": "213.7"},  

40:       {"device_ip": "29.20.3.7", "device_id": "213.6"},  

41:       {"device_ip": "29.20.213.6", "device_id": "213.6"},  

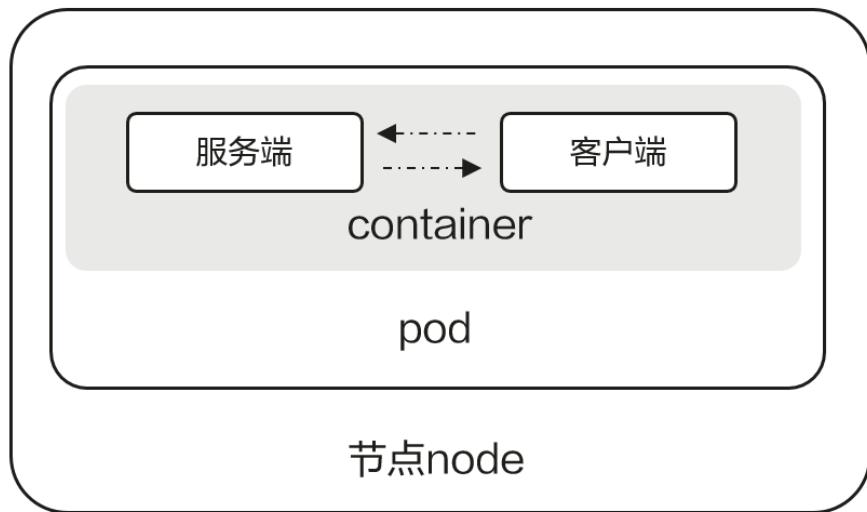
42:       {"device_ip": "29.20.213.7", "device_id": "213.7"}]}"
43:   csi.volume.kubernetes.io/nodeid: "[ disk.csi.evb64a5a-98e9-134a9c8ef1 ]"
44:
```

## 4.3 在 Lite Cluster 资源池上使用 Snt9B 完成推理任务

### 场景描述

本案例介绍如何在Snt9B环境中利用Deployment机制部署在线推理服务。首先创建一个Pod以承载服务，随后登录至该Pod容器内部署在线服务，并最终通过新建一个终端作为客户端来访问并测试该在线服务的功能。

图 4-13 任务示意图



### 操作步骤

**步骤1** 拉取镜像。本测试镜像为bert\_pretrain\_mindspore:v1，已经把测试数据和代码打进镜像中。

```
docker pull swr.cn-southwest-2.myhuaweicloud.com/os-public-repo/bert_pretrain_mindspore:v1
docker tag swr.cn-southwest-2.myhuaweicloud.com/os-public-repo/bert_pretrain_mindspore:v1
bert_pretrain_mindspore:v1
```

## 步骤2 在主机上新建config.yaml文件。

config.yaml文件用于配置pod，本示例中使用sleep命令启动pod，便于进入pod调试。您也可以修改command为对应的任务启动命令（如“python inference.py”），任务会在启动容器后执行。

config.yaml内容如下：

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: yourapp
  labels:
    app: infers
spec:
  replicas: 1
  selector:
    matchLabels:
      app: infers
  template:
    metadata:
      labels:
        app: infers
    spec:
      schedulerName: volcano
      nodeSelector:
        accelerator/huawei-npu: ascend-1980
      containers:
        - image: bert_pretrain_mindspore:v1          # Inference image name
          imagePullPolicy: IfNotPresent
          name: mindspore
          command:
            - "sleep"
            - "100000000000000000000000"
          resources:
            requests:
              huawei.com/ascend-1980: "1"      # 需求卡数，key保持不变。Number of required NPUs. The
maximum value is 16. You can add lines below to configure resources such as memory and CPU.
            limits:
              huawei.com/ascend-1980: "1"      # 限制卡数，key保持不变。The value must be consistent
with that in requests.
          volumeMounts:
            - name: ascend-driver           #驱动挂载，保持不动
              mountPath: /usr/local/Ascend/driver
            - name: ascend-add-ons         #驱动挂载，保持不动
              mountPath: /usr/local/Ascend/add-ons
            - name: hccn                   #驱动hccn配置，保持不动
              mountPath: /etc/hccn.conf
            - name: npu-smi                #npu-smi
              mountPath: /usr/local/sbin/npu-smi
            - name: localtime              #The container time must be the same as the host time.
              mountPath: /etc/localtime
          volumes:
            - name: ascend-driver
              hostPath:
                path: /usr/local/Ascend/driver
            - name: ascend-add-ons
              hostPath:
                path: /usr/local/Ascend/add-ons
            - name: hccn
              hostPath:
                path: /etc/hccn.conf
            - name: npu-smi
              hostPath:
                path: /usr/local/sbin/npu-smi
            - name: localtime
```

```
hostPath:  
  path: /etc/localtime
```

**步骤3** 根据config.yaml创建pod。

```
kubectl apply -f config.yaml
```

**步骤4** 检查pod启动情况，执行下述命令。如果显示“1/1 running”状态代表启动成功。

```
kubectl get pod -A
```

**步骤5** 进入容器，{pod\_name}替换为您的pod名字（get pod中显示的名字），{namespace}替换为您的命名空间（默认为default）。

```
kubectl exec -it {pod_name} bash -n {namespace}
```

**步骤6** 激活conda模式。

```
su - ma-user //切换用户身份  
conda activate MindSpore //激活 MindSpore环境
```

**步骤7** 创建测试代码test.py。

```
from flask import Flask, request  
import json  
app = Flask(__name__)  
  
@app.route('/greet', methods=['POST'])  
def say_hello_func():  
    print("----- in hello func -----")  
    data = json.loads(request.get_data(as_text=True))  
    print(data)  
    username = data['name']  
    rsp_msg = 'Hello, {}!'.format(username)  
    return json.dumps({"response":rsp_msg}, indent=4)  
  
@app.route('/goodbye', methods=['GET'])  
def say_goodbye_func():  
    print("----- in goodbye func -----")  
    return '\nGoodbye!\n'  
  
@app.route('/', methods=['POST'])  
def default_func():  
    print("----- in default func -----")  
    data = json.loads(request.get_data(as_text=True))  
    return '\n called default func !\n {} \n'.format(str(data))  
  
# host must be "0.0.0.0", port must be 8080  
if __name__ == '__main__':  
    app.run(host="0.0.0.0", port=8080)
```

执行代码，执行后如下图所示，会部署一个在线服务，该容器即为服务端。  
python test.py

图 4-14 部署在线服务

```
(MindSpore) [root@yourapp-664ddf9d49-qmc7s /]# python a.py  
* Serving Flask app 'a' (lazy loading)  
* Environment: production  
WARNING: This is a development server. Do not use it in a production deployment.  
Use a production WSGI server instead.  
* Debug mode: off  
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  
* Running on all addresses (0.0.0.0)  
* Running on http://127.0.0.1:8080  
* Running on http://172.16.0.45:8080  
Press CTRL+C to quit
```

**步骤8** 在XShell中新建一个终端，参考步骤5~7进入容器，该容器为客户端。执行以下命令验证自定义镜像的三个API接口功能。当显示如图所示时，即可调用服务成功。

```
curl -X POST -H "Content-Type: application/json" --data '{"name":"Tom"}' 127.0.0.1:8080/  
curl -X POST -H "Content-Type: application/json" --data '{"name":"Tom"}' 127.0.0.1:8080/greet  
curl -X GET 127.0.0.1:8080/goodbye
```

图 4-15 访问在线服务

```
[root@yourapp-664ddf9d49-qmc7s /]# curl -X POST -H "Content-Type: application/json" --data '{"name":"Tom"}' 127.0.0.1:8080/
called default func !
{'name': 'Tom'}
[root@yourapp-664ddf9d49-qmc7s /]# curl -X POST -H "Content-Type: application/json" --data '{"name":"Tom"}' 127.0.0.1:8080/greet
{
    "response": "Hello, Tom!"
}[root@yourapp-664ddf9d49-qmc7s /]# curl -XGET 127.0.0.1:8080/goodbye
Goodbye!
```

### 📖 说明

limit/request配置cpu和内存大小，已知单节点Snt9B机器为：8张Snt9B卡+192核1536GB，请合理规划，避免cpu和内存限制过小引起任务无法正常运行。

----结束

## 4.4 在 Lite Cluster 资源池上使用 Ascend FaultDiag 工具完成日志诊断

### 场景描述

本文档介绍了在ModelArts Lite环境下使用Ascend FaultDiag工具进行日志诊断的过程，包括日志采集、日志清洗、故障诊断三个步骤。

日志数据以节点为单位进行采集，在单节点日志目录下分别清洗，将清洗结果汇总后，进行故障诊断。例如，对于运行在8个节点共64卡集群上的任务，需要在8个节点上分别进行日志采集，收集的日志存储在worker-0 ~ worker-7这8个目录下。然后分别在8个目录下进行日志清洗，每一个目录下的日志清洗结果分别存储到output/worker-0 ~ output/worker-7下面。最后，在output目录下进行故障诊断，得到诊断结果。

Ascend FaultDiag工具下载地址请见[Ascend FaultDiag 故障诊断工具](#)。

### 步骤一：日志采集

共需要采集6类日志：用户训练打屏日志、主机侧操作系统日志（Host日志）、Device侧日志、CANN日志、主机侧资源信息、NPU网口资源信息。

- 用户训练打屏日志：指在训练过程中，通过设置环境变量将日志信息输出到标准输出（屏幕）的日志。
- 机侧操作系统日志（Host日志）：指在训练作业运行过程中，HOST侧用户进程产生的日志。
- Device侧日志：指在Host侧用户进程运行时，Device侧产生的AICPU、HCCP的日志，这些日志会被回传到Host侧。
- CANN日志：CANN日志是昇腾（Ascend）计算架构中用于记录CANN（Compute Architecture for Neural Networks）模块运行时信息的日志。在模型转换过程中，如果遇到“Convert graph to om failed”等错误，CANN日志可以帮助分析问题。
- 主机侧资源信息：指在主机（Host）侧运行的AI应用或服务所使用的资源统计信息。
- NPU网口资源信息：指在主机（Host）侧运行的AI应用或服务所使用的资源统计信息。

如果日志数据已经在训练时进行了输出和转储，例如存放在OBS，且符合[约束限制](#)中的文件名和路径约束，则跳过日志采集步骤，进入[步骤二：日志清洗](#)。

### ● 约束限制

- CANN日志采集后必须放在名为“process\_log”的文件夹下，示例：“worker-0/…/process\_log/”。
- Device侧日志采集后必须放在名为“device\_log”的文件夹下，示例：“worker-0/…/device\_log/”。
- 主机侧资源信息、NPU网口资源信息采集后必须放在名为“environment\_check”的文件夹下，示例：“worker-0/…/environment\_check/”。
- 单节点采集的日志，即单worker目录下，总文件大小应限制在5G以下，文件总数量不能超过一百万，否则将影响日志清洗效率。
- 用户训练打屏日志无大小限制，会默认只读最后100KB日志。
- CANN日志单个文件应限制在20MB以下。
- NPU状态监测指标文件、NPU网口统计监测指标文件、主机侧资源信息文件应限制在512MB以下。
- Host日志当前仅支持“/var/log”下的messages日志，且单个文件的转储大小上限应限制在512MB以下。

## 步骤二：日志清洗

采集的日志需要按照不同节点路径进行组织，如“worker-0”目录下存放从对应节点采集的所有日志。需要特别注意目录下“device\_log”、“process\_log”、“environment\_check”三个子目录是否存在，且命名正确。

### 1. 数据挂载

如果所采集日志的存储在OBS上，首先需要将OBS内的日志数据进行挂载。挂载方式建议使用[rclone工具](#)。

a. 下载安装rclone。

b. 首先配置访问OBS所需凭据：

```
# 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；  
# 本示例以ak和sk保存在环境变量中来实现身份验证为例，运行本示例前请先在本地环境中设置环境  
变量HUAWEICLOUD_SDK_AK和HUAWEICLOUD_SDK_SK。  
export AWS_ACCESS_KEY=${HUAWEICLOUD_SDK_AK}  
export AWS_SECRET_KEY=${HUAWEICLOUD_SDK_SK}  
export AWS_SESSION_TOKEN=${TOKEN}
```

c. 填写rclone配置文件rclone.conf：

```
[rclone]  
type = s3  
provider = huaweiOBS  
env_auth = true  
acl = private
```

d. 使用lsd命令查看日志路径下目录，验证是否配置成功：

```
rclone lsd rclone:${obs_bucket_name}/${path_to_logs} --config=${path_to_rclone.config} --s3-  
endpoint=${obs_endpoint} -no-check-certificate
```

e. 屏幕输出日志目录，示例（该任务只有worker-0一个节点）：

```
[root@test-7f56594b4-mvzb8 modelarts-ascend-brain]# /opt/rclone/rclone lsd rclone:${obs_bucket_input}/${obs_path_input}/\n> --config=/opt/rclone/rclone.conf --s3-endpoint=${s3_endpoint} -no-check-certificate\n          0 2000-01-01 00:00:00\n          -1 worker-0
```

f. 使用mount命令将日志目录挂载到本机：

```
rclone mount rclone:${obs_bucket_name}/${path_to_logs} /${path_to_local_dir} --config=${path_to_rclone.config} --s3-endpoint=${obs_endpoint} -no-check-certificate
```

## 2. 节点日志清洗

指定单节点日志路径为输入，指定该节点日志清洗存储路径为输出（输出路径需要为空），使用ascend-fd parse命令逐一对单个节点的日志进行清洗。

```
ascend-fd parse -i ${path_to_worker_logs} -o ${path_to_parse_output}
```

```
[root@node0 ~]# ascend-fd parse -i worker-0/ -o log-output/worker-0/
The parse job starts. Please wait.
These job ['NODE_ANOMALY', 'KNOWLEDGE_GRAPH', 'ROOT_CLUSTER', 'NET_CONGESTION'] succeeded.
The parse job is complete.
```

需要注意的是，清洗结果也与日志相似，**不同节点需要按照不同的worker目录分别进行存储**。

## 步骤三：故障诊断

因为Linux系统限制最大进程数（默认为1024），所以集群规格建议≤128台服务器（1024卡）。如果服务器数量超过此规格，需使用ulimit -n \${num}命令调整文件描述符上限，其中\${num}值大于卡数，如6k卡集群，则可设置为8192。

日志故障诊断需要指定所有节点清洗结果的所在路径，指定诊断结果存储路径为输出，且输出路径需要为空，使用ascend-fd diag命令进行日志故障诊断：

```
ascend-fd diag -i ${path_to_parse_outputs} -o ${path_to_diag_output}
```

诊断结果以两种形式进行呈现：

- 屏幕回显
- 在“\${path\_to\_diag\_output}/fault\_diag\_result”目录下生成的diag\_report.json文件。

```
root@192.168.1.11:~# cat result/fault_diag_result/diag_report.json
{
    "Version": "6.0.RC2",
    "Build_Time": "2024-05-09",
    "Root_Cluster": {
        "analyze_success": true,
        "fault_description": {
            "code": 101,
            "string": "所有节点的Plog都没有记录超时类错误日志。日志中有报错的节点为疑似根因节点，请排查。"
        },
        "root_cause_device": [
            "ALL Device"
        ],
        "device_link": [],
        "first_error_device": "worker-11 device-2: 2024-01-22-23:50:00.149859",
        "note": "根因节点分析检测出了多个的疑似故障根因节点，将优先排查这几个节点",
        "show_device_info": [
            {
                "device_type": "first_root_device",
                "device": "worker-11 device-2",
                "plog_file_path": "/mnt/wangdong/Logs/06641812-ffeb-4f15-a8b9-d677ac270a1f/log-output/worker-11/plog-p",
                "error_log": "[ERROR] RUNTIME(103942,python):2024-01-22-23:50:00.149.859 [engine.cc:1263]l128359 Report
428 [task.cc:92]l28359 PrintErrorInfo:Task execute failed, base info: device_id=2, stream_id=45, task_id=2, flip_
failed].\n[ERROR] RUNTIME(103942,python):2024-01-22-23:50:00.152.473 [task.cc:3210]l28359 PrintErrorInfo:model ex
back.cc:91]l28359 Notify:notify [HCCL] task fail start.notify taskid:2 streamid:45 retcode:507011\n[ERROR] RUNTIME
/thon):2024-01-22-23:50:00.154.237 [stream.cc:1041]l28359 GetError:Stream Synchronize failed, stream_id=3, retCode
99\n[ERROR] RUNTIME(103942,python):2024-01-22-23:50:00.154.255 [stream.cc:1044]l28359 GetError:Task execute faile
stream synchronize failed\n"
            }
        ],
        "Knowledge_Graph": {
            "analyze_success": true,
            "note": "",
            "fault": [
                {
                    "code": "NORMAL_OR_UNSUPPORTED",
                    "component": "",
                    "module": "",
                    "cause_zh": "故障根因设备诊断无异常",
                    "description_zh": "故障根因设备诊断无异常，可能情况为：a. 无相关故障发生；b. 存在未知故障。",
                    "suggestion_zh": "1. 若存在问题无法解决，请联系华为工程师定位排查",
                    "class": "",
                    "fault_source": [
                        "worker-0",
                        "worker-1",
                        "worker-2",
                        "worker-3",
                        "worker-4",
                        "worker-5",
                        "worker-6",
                        "worker-7",
                        "worker-8",
                        "worker-9",
                        "worker-10",
                        "worker-11",
                        "worker-12",
                        "worker-13",
                        "worker-14",
                        "worker-15"
                    ],
                    "fault_chains": []
                }
            ]
        }
    }
}
root@192.168.1.11:~#
```

## 4.5 在 Lite Cluster 挂载 SFS Turbo

### 场景描述

当本地服务器磁盘空间不足，无法满足业务增长需求时，将数据盘挂载到Lite Cluster资源池，可以实现存储资源的动态分配，满足存储资源的灵活调度、高效利用和安全访问需求。

华为云高性能弹性文件服务（Scalable File Service Turbo，SFS Turbo）提供按需扩展的高性能文件存储（NAS），能够满足大规模数据读写需求，且支持多个计算节点共享同一文件系统。在Lite Cluster挂载SFS Turbo，可以显著提升数据访问性能，可以在Lite Cluster的多个节点之间实现数据共享，能够提升资源利用率和任务协作效率，适用于高性能计算和分布式训练场景。

本文档以Ipv4为例介绍了在ModelArts Lite Cluster环境下挂载SFS Turbo文件系统的过 程，以及在主机及Kubernetes容器中挂载使用SFS Turbo文件系统的过 程。将Lite Cluster与待挂载的SFS Turbo文件系统建立在同一个VPC子网中。

## 注意事项

用户需合理规划资源池及SFS Turbo网段，避免ModelArts Lite Cluster节点与SFS Turbo文件系统网段冲突。具体规则：

当SFS Turbo文件系统选择192.168.x.x开头的VPC网段时，资源池节点需避免172.16.0.0/16的网段。

当SFS Turbo文件系统选择172.x.x.x开头的VPC网段时，资源池节点需避免192.168.0.0/16的网段。

当SFS Turbo文件系统选择10.x.x.x开头的VPC网段时，资源池节点需避免172.16.0.0/16的网段。

## 计费影响

- 在开通Lite Cluster资源后，会产生计算资源的计费。Lite Cluster资源池仅支持包年/包月计费模式，具体内容如[表4-1](#)所示。

表 4-1 计费项

计费项	计费项说明	适用的计费模式	计费公式
计算资源 专属资源池	使用计算资源的用量。 具体费用可参见 <a href="#">ModelArts价格详情</a> 。	包年/包月	规格单价 * 计算节点个数 * 购买时长

- 购买Cluster资源池时，需要选择CCE集群，具体费用请参考[CCE计费详情](#)。
- 挂载的SFS Turbo文件系统按购买时选择的存储容量和时长收费，详情请见[SFS 计费说明](#)。

## 步骤一：创建 VPC

在Lite Cluster挂载SFS Turbo，需要将Lite Cluster与待挂载的SFS Turbo文件系统建立在同一个VPC子网中，因此需要创建一个VPC子网。

- 进入[创建虚拟私有云页面](#)。
- 在“创建虚拟私有云”页面，根据界面提示配置VPC和子网的参数。

在本案例中部分参数说明请见[表4-2](#)，更多参数说明请见[创建虚拟私有云和子网](#)。

图 4-16 创建 VPC 和子网



表 4-2 虚拟私有云部分参数说明

参数	说明	样例
IPv4网段	<p>Lite Cluster挂载Sfs Turbo场景下，建议您使用<a href="#">RFC 1918</a>中指定的私有IPv4地址范围，作为VPC的网段，具体如下：</p> <ul style="list-style-type: none"><li>• 10.0.0.0/8-24：IP地址范围为10.0.0.0~10.255.255.255，掩码范围为8~24。</li><li>• 172.16.0.0/12-24：IP地址范围为172.16.0.0~172.31.255.255，掩码范围为12~24。</li><li>• 192.168.0.0/16-24：IP地址范围为192.168.0.0~192.168.255.255，掩码范围为16~24。</li></ul>	192.168.0.0/16

表 4-3 子网部分参数说明

参数	说明	取值样例
子网网段	在未开启IPv4/IPv6双栈的区域，显示此参数。 设置VPC子网的IPv4网段范围，参数填写说明请参见“子网IPv4网段”。	192.168.0.0/24

参数	说明	取值样例
子网IPv4网段	<p>在开启IPv4/IPv6双栈的区域，显示此参数。</p> <p>设置子网的IPv4网段范围，子网是VPC内的IP地址块，可以将VPC的网段分成若干块，建议您规划子网时，遵循以下原则：</p> <ul style="list-style-type: none"><li>• 子网内可用IP数量：子网创建成功后，不支持修改网段，请您结合业务所需的IP地址数量，提前合理规划好子网网段。<ul style="list-style-type: none"><li>- 子网网段不能太小，需要确保子网内可用IP地址数量可以满足业务需求。子网网段中第一个地址和后三个地址为系统预留地址，不能供实际业务使用，比如子网（10.0.0.0/24）中，10.0.0.1为网关地址、10.0.0.253为系统接口、10.0.0.254为DHCP使用、10.0.0.255为广播地址。</li><li>- 子网网段也不能太大，以免后续扩展新的业务时，VPC内可用网段不够，无法再创建新的子网。</li></ul></li><li>• 子网网段避免冲突：如果子网所在的VPC与其他VPC、或者VPC与云下数据中心需要通信时，则VPC子网网段和网络对端网段不能相同，否则无法正常通信。如果网络两端的子网网段已经相同，您可以创建新的子网，请</li></ul>	192.168.0.0/24

参数	说明	取值样例
	<p>参见<a href="#">为虚拟私有云创建新的子网</a>。</p> <p>子网的网段必须在VPC网段范围内，子网网段的掩码长度范围为“子网所在VPC的掩码~29”，比如VPC网段为10.0.0.0/16，掩码为16，则子网的掩码可在16~29范围内选择。</p> <p>关于VPC子网规划更详细的说明，请参见<a href="#">虚拟私有云和子网规划建议</a>。</p>	

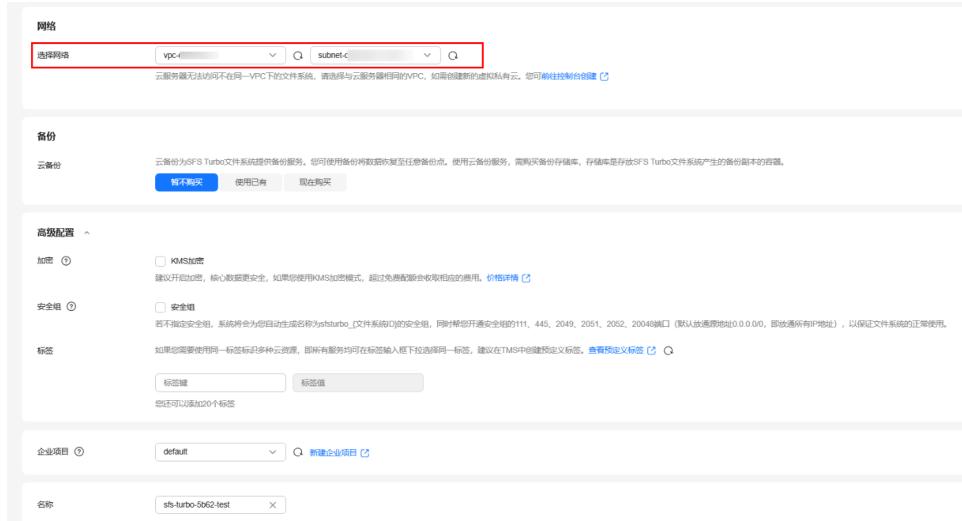
- 参数设置完成后，单击“立即创建”。
- 返回VPC列表，可以查看新创建的VPC。

## 步骤二：创建 SFS Turbo 文件系统

创建待挂载的SFS Turbo文件系统，网络使用[步骤一：创建VPC](#)中创建的VPC。

- 登录[弹性文件服务SFS控制台](#)，在左侧导航栏，选择“SFS Turbo”。在页面右上角单击“创建文件系统”。
- 如图4-17所示，根据界面提示配置参数。  
“选择网络”选择[步骤一：创建VPC](#)中创建的VPC及子网，更多参数说明请见[创建SFS Turbo文件系统](#)。

图 4-17 创建 SFS Turbo 文件系统



- 配置完成后，单击“立即创建”。
- 核对文件系统信息，确认无误后单击“提交”。

5. 根据页面提示，完成创建后，返回文件系统列表页面查看文件系统状态。

## 步骤三：创建 CCE

由于Lite Cluster资源池依赖于CCE集群来提供容器化的运行环境，并且CCE集群为Lite资源池提供必要的计算、存储和网络资源，所以购买Cluster资源池时，需要选择CCE集群。

如果您没有可用的CCE集群，可参考[购买Standard/Turbo集群](#)进行购买，集群配套版本请参考[1.3 不同机型对应的软件配套版本](#)。

### 1. 登录CCE控制台。

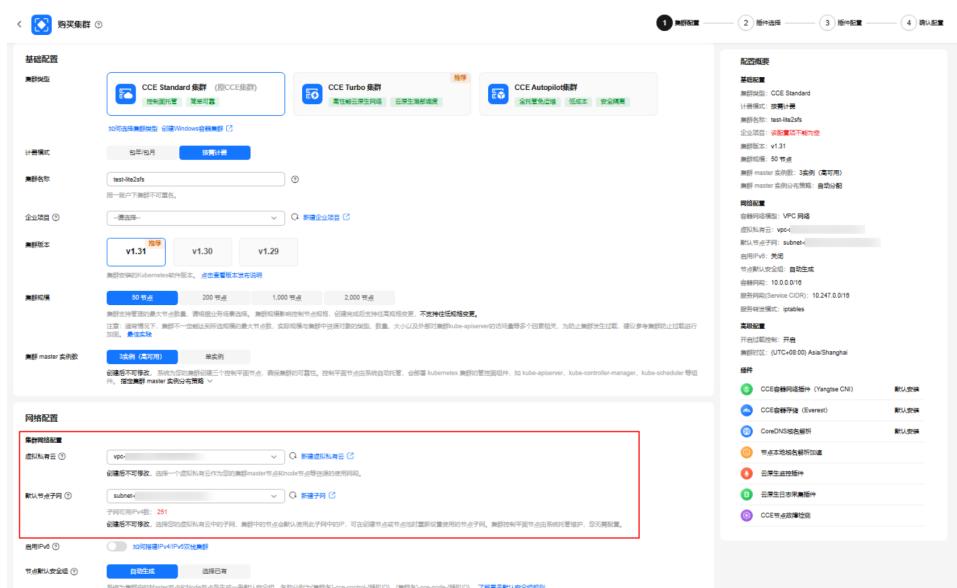
- 如果您的账号还未创建过集群，请在引导页面中单击页面上方的“[购买集群](#)”。
- 如果您的账号已经创建过集群，请在左侧菜单栏选择集群管理，单击右上角“[购买集群](#)”。

### 2. 对集群的基础信息进行配置。

- 虚拟私有云：选择[步骤一：创建VPC](#)创建的VPC。
- 默认节点子网：选择[步骤一：创建VPC](#)创建的子网。
- 启用IPv6：本文以IPv4为例，此处不启用IPv6。

更多参数说明请参见[购买Standard/Turbo集群](#)。

图 4-18 购买 CCE 集群



3. 单击“下一步：确认配置”，显示集群资源清单，确认无误后，单击“提交”。  
集群创建预计需要5-10分钟，您可以单击“返回集群管理”进行其他操作或单击“查看集群事件列表”后查看集群详情。

## 步骤四：创建 Lite Cluster 资源

使用[步骤三：创建CCE](#)已创建的CCE集群创建Lite Cluster资源。

1. 登录**ModelArts管理控制台**，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”，进入“轻量算力集群（Lite Cluster）”页面。
2. 在“轻量算力集群（Lite Cluster）”页面，单击“购买轻量算力集群”，进入购买轻量算力集群界面填写参数。  
CCE集群选择**步骤三：创建CCE**创建的CCE。更多参数说明请见**表2-3**。
3. 单击“立即购买”确认规格。产品规格和协议许可确认无误后，单击“提交”，即可创建Lite Cluster资源池。

当资源池创建成功后，资源池的状态会变成“运行中”。单击集群资源名称，进入资源详情页。确认购买的规格是否正确。

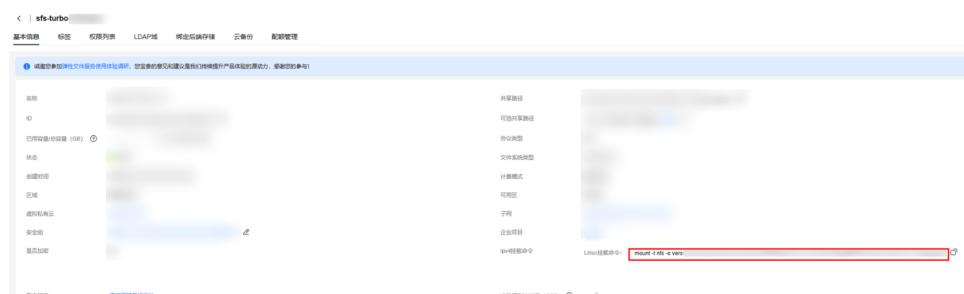
图 4-19 查看资源详情



## 步骤五：Lite Cluster 节点主机挂载 SFS Turbo 文件系统

1. 在弹性文件服务SFS控制台左侧导航栏选择“SFS Turbo”，单击**步骤二：创建SFS Turbo文件系统**创建的SFS Turbo文件系统名称，进入SFS Turbo文件系统详情界面，复制“Linux挂载命令”。

图 4-20 复制 Linux 挂载命令



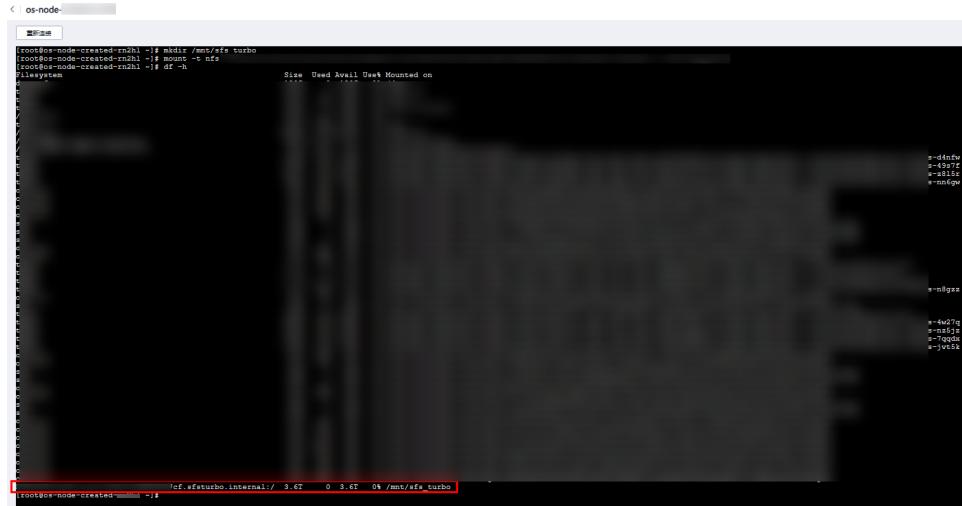
2. 通过Xshell、MobaXterm等bash工具登录Lite Cluster节点，执行命令创建待验证挂载的文件夹。  
`mkdir /mnt/sfs_turbo`
3. 执行前面步骤复制的Linux挂载命令。

```
mount -t nfs -o vers=3,nolock,proto=tcp,noresvport xxxx.sfsturbo.internal:/ /mnt/sfs_turbo
```

执行df -h，查看SFS文件系统挂载信息。

如图所示，此时SFS Turbo文件系统已成功挂载到节点主机的/mnt/sfs\_turbo目录。

图 4-21 查看挂载信息



## 步骤六：Lite Cluster k8s 集群工作负载挂载 SFS Turbo 文件系统

通过Kubernetes NFS可以将SFS Turbo文件系统挂载到工作负载中。

- 在弹性文件服务SFS控制台左侧导航栏选择“SFS Turbo”，单击[步骤二：创建SFS Turbo文件系统](#)创建的SFS Turbo文件系统名称，进入SFS Turbo文件系统详情界面，复制“共享路径”。

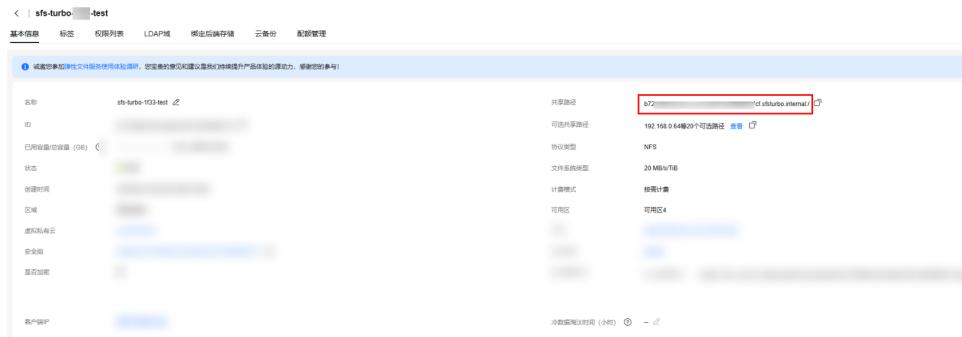
xxxxx.sfsturbo.internal:/

该路径以":"分割为k8s nfs挂载需要的两个参数，其中

“xxxxx.sfsturbo.internal”为工作负载nfs挂载参数“nfs.server”。

“/”为工作负载nfs挂载参数“nfs.path”。

图 4-22 复制共享路径



- 通过Xshell、MobaXterm等bash工具登录Lite Cluster节点，创建dep.yaml文件，内容如下。

```
kind: Deployment  
apiVersion: apps/v1
```

```
metadata:
  name: testlite2sfsturbo
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: testlite2sfsturbo
      version: v1
  template:
    metadata:
      labels:
        app: testlite2sfsturbo
        version: v1
    spec:
      volumes:
        - name: nfs0
          nfs:
            server: xxxxx.sfsturbo.internal ## 填写sfsturbo文件系统共享路径中的server信息
            path: /
      containers:
        - name: pod0
          image: swr.cn-southwest-2.myhuaweicloud.com/hwofficial/everest:2.4.134 ## 镜像地址
          command:
            - /bin/bash
            - '-c'
            - while true; do echo hello; sleep 10; done
          env:
            - name: PAAS_APP_NAME
              value: testlite2sfsturbo
            - name: PAAS_NAMESPACE
              value: default
            - name: PAAS_PROJECT_ID
              value: xxxx
          resources:
            limits:
              cpu: 250m
              memory: 2000Mi
            requests:
              cpu: 250m
              memory: 2000Mi
      volumeMounts:
        - name: nfs0
          mountPath: /mnt/sfsturbo/test
  imagePullPolicy: IfNotPresent
```

3. 执行kubectl apply -f dep.yaml创建工作负载。
4. 执行kubectl get pod 查看pod容器组启动成功。
5. 执行kubectl exec -it {pod\_name} -- bash 登录容器内。
6. 执行df -h  
如图所示，此时SFS Turbo文件系统已成功挂载到k8s容器内的/mnt/sfs\_turbo/test目录。

图 4-23 查看挂载信息

## 4.6 在 Lite Cluster 资源池设置并启用高可用冗余节点

## 场景描述

当业务在连续运行、高并发流量等场景下，一旦承载业务的节点出现故障，如果没有备份机制，可能导致业务中断等严重后果，造成巨大损失和影响。此时，需要一种保障机制来维持业务稳定运行。

高可用冗余节点是指在ModelArts平台中用于保障服务高可用性的备用节点。当业务节点发生故障时，高可用冗余节点可以快速接管服务，确保业务的连续性和稳定性，并且可根据业务流量动态分配资源，应对高并发场景。

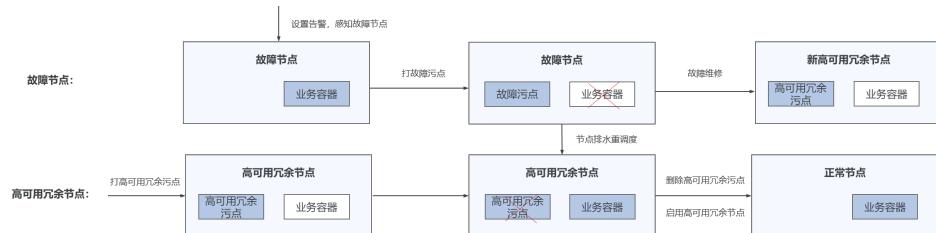
在ModelArts Standard资源池中，已经提供了高可用冗余节点的能力，详情请见[高可用冗余节点](#)。但是Lite Cluster暂无直接设置高可用冗余节点的能力，本文主要介绍在ModelArts Lite Cluster资源池中如何手动设置高可用冗余节点，当节点出现故障时，可以启用高可用冗余节点，快速恢复业务，而不用等待故障节点修复好。

图 4-24 高可用冗余节点



整体流程如下：

图 4-25 设置并启用高可用冗余节点流程



**步骤一：设置高可用冗余节点：**通过为节点打上特定污点的方式设置高可用冗余节点。

**步骤二：配置节点警报通知感知故障节点：**通过配置节点警报通知，感知节点故障。

**步骤三：高可用冗余节点替换故障节点：**为故障节点打上故障污点，并设置节点排水，排空故障节点的任务。同时，删除高可用冗余节点的特定污点，正式启用高可用冗余节点。

**步骤四：将故障节点转为新高可用冗余节点：**待华为云完成故障节点维修后，将故障节点转为新高可用冗余节点。

## 计费影响

高可用冗余节点的计费方式和普通节点相同，会产生计算资源的计费。Lite Cluster资源池仅支持包年/包月计费模式，具体内容如表4-4所示。

表 4-4 计费项

计费项		计费项说明	适用的计费模式	计费公式
计算资源	专属资源池	使用计算资源的用量。 具体费用可参见 <a href="#">ModelArts价格详情</a> 。	包年/包月	规格单价 * 计算节点个数 * 购买时长

## 前提条件

已创建Lite Cluster资源池，详情请见[Lite Cluster资源开通](#)。

## 步骤一：设置高可用冗余节点

在Lite Cluster资源池使用前或使用时，通过为节点打上特定污点的方式设置高可用冗余节点。

1. 登录ModelArts管理控制台，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”，进入“轻量算力集群（Lite Cluster）”页面。
2. 单击资源池名称，进入资源池详情页“基本信息”页签。
3. 单击CCE集群超链接，进入CCE集群节点管理页面。

图 4-26 资源池基本信息



4. 在选定的空闲节点右侧，单击“更多>污点管理”，进入污点管理。
5. 在污点管理弹框中，给该节点添加污点：key=backupNode，effect=NoSchedule，单击“确定”。

## 步骤二：配置节点告警通知感知故障节点

通过配置节点告警通知，感知节点故障。

节点故障指标(nt\_npg)默认会上报到AOM，您可以在AOM配置短信、邮件等通知方式。

同时，在节点故障后，您可以在ModelArts“资源管理>事件中心”，查看到该节点的计划事件，并授权华为云维修，详细请参考[事件中心页面授权运维](#)。

### □ 说明

以下步骤基于AOM1.0配置。

**步骤1 登录AOM控制台**

**步骤2 在左侧导航栏选择“告警中心 > 告警规则”，单击“创建告警规则”。**

**步骤3 设置告警规则（以NPU掉卡为例）。**

- 规则类型：选择指标告警规则。
- 配置方式：选择PromQL。
- 默认规则：选择自定义，命令行输入框：  
`sum(nt_npg{type="NT_NPU_CARD_LOSE"} != 2) by (cluster_name, node_ip,type)`
- 告警条件：选择触发条件为持续时间1分钟，产生重要告警。
- 告警通知（可选）：如果需要将告警通过邮件、手机方式通知您，可在告警通知处，为此告警规则配置行动规则。如果此处无行动规则，请新建告警行动规则。

----结束

## 步骤三：高可用冗余节点替换故障节点

为故障节点打上故障污点，并设置节点排水，排空故障节点的任务。同时，删除高可用冗余节点的特定污点，正式启用高可用冗余节点。

1. 给故障节点打上故障污点。

污点设置方式和**步骤一：设置高可用冗余节点**一致，设置污点key=faultyNode，effect=NoSchedule。

2. 在CCE集群节点管理页面，选择已按**步骤一：设置高可用冗余节点**设置污点的高可用冗余节点，单击列表项中的“污点管理”。

3. 在弹出的对话框中，找到“key”为“backupNode”的污点记录，单击“删除”，然后单击“确定”。

4. 在CCE集群节点管理页面，在该故障节点右侧，单击“更多>节点排水”。

5. 在节点排水界面，勾选“强制排水”，设置排水时，系统会自动将该节点设置为不可调度，同时自动打上key为“node.kubernetes.io/unschedulable”的污点。  
排空步骤1打上故障污点的故障节点上的任务，重新调度受影响的任务。

排水完成时，在CCE集群节点管理页面该节点状态处会显示排水成功。

6. 待排水完成后，在故障节点右侧，单击“更多>开启调度”，并单击“是”，取消自动设置的不可调度，同时会自动去除“key”为“node.kubernetes.io/unschedulable”的污点。

## 步骤四：将故障节点转为新高可用冗余节点

待华为云完成故障节点维修后，将故障节点转为新高可用冗余节点。

您可以在ModelArts“资源管理>事件中心”，查看到该节点的维修状态，事件状态显示为“已完成”时代表已维修完成，详细请参考[事件中心页面授权运维](#)。

参考[步骤一：设置高可用冗余节点](#)步骤，对已修复的故障节点添加污点key=backupNode, effect=NoSchedule，同时去除key=faultyNode的污点。

## 4.7 在 Lite Cluster 跨区域访问其他服务

### 场景描述

当您使用专属资源池创建作业时(如训练作业)，如果作业运行时有跨区域访问已搭建的站点服务或数据需求，可借助云连接实现跨区域的数据访问。

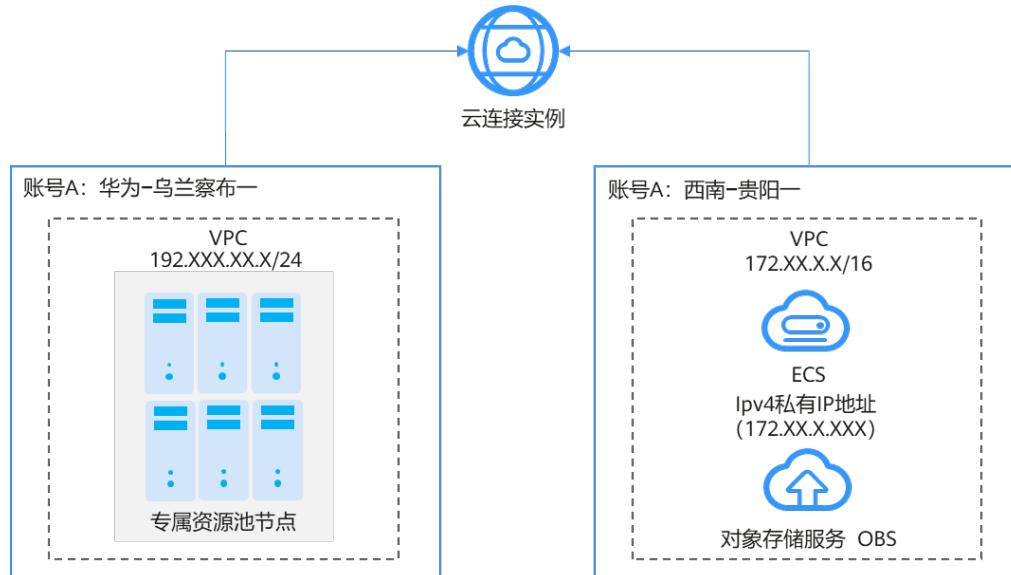
云连接实例支持区域，请参见[云连接实例支持区域](#)。

### 方案架构

企业在华为云账号A的华北-乌兰察布一创建了Lite Cluster资源池，在西南-贵阳一区域已搭建站点服务或数据，华北-乌兰察布一的Lite Cluster资源池需要访问西南-贵阳一区域的数据或服务。

创建一个云连接实例，将VPC接入云连接实例内，云连接实例内的VPC则可以实现网络互通。

图 4-27 资源池跨区域访问



组网规划说明：

- VPC网段（CIDR）不能重叠，重叠的VPC网段会导致路由冲突。需要确保VPC-B的网段和VPC-A的网段不重叠，还需要和资源池k8s集群的服务网段和容器网段不重叠。其中k8s集群的服务网段和容器网段可登录云容器引擎CCE页面，在集群详情的网络信息中查询。

- 安全组需要放通。本示例中，云服务器ECS-B01的安全组入方向需要放通VPC-A CIDR网段来的流量。

## 前提条件

- 已创建Lite Cluster资源池，详情请见[Lite Cluster资源开通](#)。
- 为账户充值。  
您需要确保账户有足够的金额，充值方式请参见[账户充值](#)。
- 已创建好资源池跨区域待连接的云服务器，并已设置好安全组规则，详情请见[购买ECS](#)。本示例中，云服务器ECS-B01的安全组入方向需要放通VPC-A CIDR网段来的流量。
- 创建VPC及子网，具体方法请参见[创建虚拟私有云和子网](#)。

## 创建云连接实例

本案例在账号A中创建一个云连接实例。

- 进入[云连接实例列表页面](#)。
- 单击页面右上方的“创建云连接”。
- 在弹出的对话框中根据[表4-5](#)填写对应参数。

**表 4-5 创建云连接实例参数**

参数	说明	取值样例
名称	云连接实例的名称。 长度为1~64个字符，中、英文字母，数字，下划线，中划线，点。	cc-test
企业项目	企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。	default
使用场景	选择虚拟私有云场景时，网络实例类型支持选择虚拟私有云（VPC）和虚拟网关（VGW）。	虚拟私有云
标签	云连接实例的标识，包括键和值。可以为云连接实例创建20个标签。 <b>说明</b> 如果已经通过TMS的预定义标签功能预先创建了标签，则可以直接选择对应的标签键和值。 预定义标签的详细内容，请参见 <a href="#">预定义标签简介</a> 。	-
描述	云连接实例的描述。 长度为0~255个字符。	-

- 单击“确定”，完成云连接实例的创建。

## 将网络实例加载至云连接实例

被授权用户根据规划的网络连通情况，将需要进行互通的VPC实例加载到创建的云连接实例中。

- 一个网络实例只可以加载到一个云连接实例中。
- VPC实例和与其关联的虚拟网关实例，不允许重复加载。

本示例中将VPC-A和VPC-B分别加载至云连接实例cc-test中，这里需要登录账号A操作，具体操作如下：

首先加载账号A的VPC-A。

- 进入[云连接实例列表页面](#)。
- 单击目标云连接实例名称，进入基本信息页面。
- 单击“网络实例”页签。
- 单击“加载网络实例”，在弹出的对话框中加载同账号网络实例。  
根据[表4-6](#)填写对应参数后，单击“确定”。

**表 4-6 加载同账号网络实例参数**

参数	说明	取值样例
账号	加载的网络实例的账号类型。	同账号
区域	需要连接的VPC所在区域。	华北-乌兰察布 —
实例类型	需要加载到云连接实例中实现互通的实例类型。 包括： <ul style="list-style-type: none"><li>虚拟私有云（VPC）</li><li>虚拟网关（VGW）</li></ul>	虚拟私有云 ( VPC )
VPC	需要加载到云连接实例中实现网络互通的VPC名称。  当实例类型参数选择虚拟私有云时，需要配置此参数。	VPC-A
VPC CIDRs	需要加载到云连接实例中实现网络互通的VPC内的网段路由。  当实例类型参数选择虚拟私有云时，需配置以下两个参数： <ul style="list-style-type: none"><li>子网</li><li>其他网段：其中包含自定义网段的配置</li></ul>	Subnet-A
备注	加载同账号网络实例备注信息。	-

然后加载账号A的VPC-B。

- 进入[云连接实例列表页面](#)。
- 单击目标云连接实例名称，进入基本信息页面。

3. 单击“网络实例”页签。
4. 单击“加载网络实例”，在弹出的对话框中选择跨账号加载。  
根据[表4-7](#)填写对应参数后，单击“确定”。

**表 4-7 加载同账号网络实例参数**

参数	说明	取值样例
账号	加载的网络实例的账号类型。	同账号
区域	需要连接的VPC所在区域。	西南-贵阳一
实例类型	需要加载到云连接实例中实现互通的实例类型。 包括： <ul style="list-style-type: none"><li>• 虚拟私有云（VPC）</li><li>• 虚拟网关（VGW）</li></ul>	虚拟私有云（VPC）
VPC	需要加载到云连接实例中实现网络互通的VPC名称。 当实例类型参数选择虚拟私有云时，需要配置此参数。	VPC-B
VPC CIDRs	需要加载到云连接实例中实现网络互通的VPC内的网段路由。 当实例类型参数选择虚拟私有云时，需配置以下两个参数： <ul style="list-style-type: none"><li>• 子网</li><li>• 其他网段：其中包含自定义网段的配置</li></ul>	Subnet-B
备注	加载同账号网络实例备注信息。	-

## 购买带宽包

云连接实例默认跨区域互通带宽为10kbps，仅用于测试连通性。为了实现相同大区不同区域或不同大区之间的互通，用户需要先购买带宽包，绑定到对应的云连接实例中，并配置域间带宽以保证业务正常使用。

一个云连接实例只能绑定一个相同规格的带宽包。

1. 进入[购买带宽包页面](#)。
2. 在购买带宽包页面中，根据[表4-8](#)填写对应参数，单击“立即购买”。

**表 4-8 购买带宽包参数**

参数	说明	取值样例
基础配置		

参数	说明	取值样例
计费模式	包年/包月。 用户根据需要选择购买时长，按照年或月为单位进行购买。	包年/包月
名称	带宽包的名称。 长度为1~64个字符，支持数字，英文字母，下划线，中划线，点。	bandwidthPackagetest
企业项目	企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。	default
标签	带宽包的标识，包括键和值。可以为带宽包创建20个标签。 <b>说明</b> 如果已经通过TMS的预定义标签功能预先创建了标签，则可以直接选择对应的标签键和值。 预定义标签的详细内容，请参见 <a href="#">预定义标签简介</a> 。	-
<b>带宽配置</b>		
计费方式	按带宽计费。	按带宽计费
互通类型	互通大区的类型。支持： <ul style="list-style-type: none"><li>大区内互通：指配置域间带宽的区域在同一个大区内。</li><li>跨大区互通：指配置域间带宽的区域在不同的大区内。</li></ul>	大区内互通
互通大区	需要实现互通的区域，即配置域间带宽时涉及的区域。	中国大陆
带宽	带宽是所有基于该带宽包配置的域间带宽总和，请根据网络情况提前做好规划。 单位Mbit/s。	10
购买时长	按照用户需求，选择对应的购买时间。 可支持自动续费。	1
云连接实例	选择需要绑定的云连接名称。支持： <ul style="list-style-type: none"><li>绑定</li><li>暂不绑定</li></ul>	暂不绑定

3. 在订单确认页面再次确认购买带宽包的信息，单击“提交”。

在带宽包列表中可查看带宽包信息，如果“状态”为“正常”，表示购买成功。

### 为带宽包绑定云连接实例

如果购买带宽包时没有绑定云连接实例，则需要将购买的带宽包和云连接实例绑定。

1. 进入[云连接实例列表页面](#)。
2. 单击目标云连接实例（cc-test）名称，进入基本信息页面。
3. 单击“带宽包”页签。
4. 单击“绑定带宽包”，在弹出的对话框中，选择已经购买的带宽包（bandwidthPackage-test）和云连接实例（cc-test）绑定。

## 配置域间带宽

云连接实例默认跨区域互通带宽为10kbps，仅用于测试连通性，需配置域间带宽以保证业务正常使用。

这里需要登录账号A操作。

1. 进入[云连接实例列表页面](#)。
2. 单击目标云连接实例名称，进入基本信息页面。
3. 单击“域间带宽”页签。
4. 单击“配置域间带宽”，根据[表4-9](#)填写对应参数。

**表 4-9 配置域间带宽参数**

参数	说明	取值样例
互通区域	需要实现互通的区域名称。 请选择两个需要互通的区域。	华北-乌兰察布一 西南-贵阳一
带宽包	云连接实例绑定的带宽包。	bandwidthPackage-test
带宽	两个区域实现互通的带宽。 所有基于该带宽包配置的域间带宽总和不超过带宽包的带宽，请预先做好规划。	10

5. 单击“确定”，完成配置。

配置完域间带宽后，配置了带宽的区域间就可以进行正常通信。

系统默认安全组规则是入方向访问受限，请确认区域内互访资源的安全组出方向、入方向规则配置正确，保证跨区域通信正常。

## （可选）配置跨区域 OBS 安全访问

1. 购买访问OBS桶的终端节点，配置通过VPCEP访问OBS桶。  
通过华为云工单获取对应桶的OBS 服务的VPCEP地址，例如在VPC-B中为西南-贵阳一桶创建OBS服务的VPCEP地址为：cn-southwest-2.com.myhuaweicloud.v4.obsrv2。
2. 资源池侧获取跨区域访问指定桶的IP地址。
  - a. 在OBS概览页面的域名信息中获取访问域名。
  - b. 在弹性云服务器页面远程登录资源池节点，通过dig命令行获取OBS桶的跨区域解析地址。格式为：dig {OBS桶访问域名}，其中OBS桶访问域名为上个步骤获取的访问域名。

以下示例中华北-乌兰察布一资源池VPC-A中的某台节点跨区域访问西南-贵阳一桶地址解析地址如下。

图 4-28 获取 OBS 桶跨区域解析地址

```
[root@test-k8s-cc-nodepool-81748-f5gal ~]# dig obs-pool-cc-test.obs.cn-southwest-2.myhuaweicloud.com
; <>> DiG 9.16.23 <>> obs-pool-cc-test.obs.cn-southwest-2.myhuaweicloud.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 28726
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags: udp: 4096
;; QUESTION SECTION:
;obs-pool-cc-test.obs.cn-southwest-2.myhuaweicloud.com. IN A
;;
;; ANSWER SECTION:
obs-pool-cc-test.obs.cn-southwest-2.myhuaweicloud.com. 120 IN CNAME obs.lz04.cn-southwest-2.myhuaweicloud.com.
obs.lz04.cn-southwest-2.myhuaweicloud.com. 600 IN A 116.63.191.7
obs.lz04.cn-southwest-2.myhuaweicloud.com. 600 IN A 116.63.191.2
obs.lz04.cn-southwest-2.myhuaweicloud.com. 600 IN A 116.63.191.6
obs.lz04.cn-southwest-2.myhuaweicloud.com. 600 IN A 116.63.191.5
;;
;; Query time: 92 msec
;; SERVER: 100.125.1.250#53(100.125.1.250)
;; WHEN: Tue Jun 06 08:40:31 CST 2025
;; MSG SIZE rcvd: 207
```

3. 在云连接中添加路由。

- 进入虚拟私有云页面，单击VPC-B名称进入详情页面。
- 在基本信息网络互通概览页面，单击VPC-B的路由表，找到匹配上个步骤中的VPCPE OBS地址网段。

在路由表目的地址包含OBS的行中，单击IP地址数，上述例子中116.63.191.7匹配的IP网段为116.63.191.0/28。

- 在云连接网络实例页面单击VPC-B，再单击“修改VPC CIDR”加入上述VPCPE OBS 地址网段，向云连接声明在VPC-B这侧还有一个网段用于访问OBS。

4. 跨区域访问OBS 验证。

- 在OBS控制台，单击步骤2西南-贵阳一的OBS桶，选择对象菜单，选中一个测试的对象后，在操作列更多中单击“复制对象URL”。
- 在弹性云服务器页面，选择一个华北-乌兰察布一资源池节点，单击“远程登录”，选择VNC方式登录。使用wget {OBS地址}。

以下示例中请求已经跨区域发送成功，只是因为测试方法没带鉴权返回403。

图 4-29 跨区域发送请求

```
[root@test-k8s-cc-nodepool-81748-f5gal ~]# wget https://obs-pool-cc-test.obs.cn-southwest-2.myhuaweicloud.com/cc-test.txt
--2025-06-06 10:03:31-- https://obs-pool-cc-test.obs.cn-southwest-2.myhuaweicloud.com/cc-test.txt
Resolving obs-pool-cc-test.obs.cn-southwest-2.myhuaweicloud.com (obs-pool-cc-test.obs.cn-southwest-2.myhuaweicloud.com)... 116.63.191.2, 116.63.191.6, 116.63.191.7
Connecting to obs-pool-cc-test.obs.cn-southwest-2.myhuaweicloud.com (obs-pool-cc-test.obs.cn-southwest-2.myhuaweicloud.com)|116.63.191.2|:443... connected.
HTTP request sent, awaiting response... 403 Forbidden
2025-06-06 10:03:31 ERROR 403: Forbidden.
[root@test-k8s-cc-nodepool-81748-f5gal ~]#
```

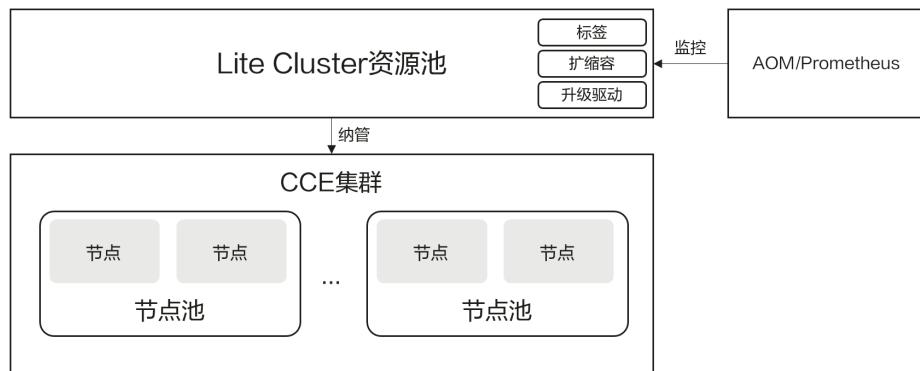
# 5 Lite Cluster 资源管理

## 5.1 Lite Cluster 资源管理介绍

在ModelArts控制台，您可以对已创建的资源进行管理。通过单击资源池名称，可以进入到资源池详情页，您可以在详情页进行下述操作。

- [5.2 管理Lite Cluster资源池](#): ModelArts支持对资源池进行管理，包括续费、开通/修改自动续费、扩容、升级驱动等操作。
- [5.3 管理Lite Cluster节点池](#): 为帮助您更好地管理Kubernetes集群内的节点，ModelArts支持通过节点池来管理节点。节点池是集群中具有相同配置的一组节点，一个节点池包含一个节点或多个节点，您可以创建、更新和删除节点池。
- [5.4 管理Lite Cluster节点](#): 节点是容器集群组成的基本元素，您可以对资源池内单节点进行替换、删除、重置等操作，也可以批量对节点进行删除、退订、续费等操作。
- [5.5 扩缩容Lite Cluster资源池](#): 当Cluster资源池创建完成，使用一段时间后，由于用户AI开发业务的变化，对于资源池资源量的需求可能会产生变化，面对这种场景，ModelArts提供了扩缩容功能，用户可以根据自己的需求动态调整。
- [5.6 升级Lite Cluster资源池驱动](#): 当资源池中的节点含有GPU/Ascend资源时，用户基于自己的业务，可能会有自定义GPU/Ascend驱动的需求，ModelArts面向此类客户提供了自助升级专属资源池GPU/Ascend驱动的能力。
- [5.8 监控Lite Cluster资源](#): ModelArts支持使用AOM和Prometheus对资源进行监控，方便您了解当前的资源使用情况。
- [5.9 释放Lite Cluster资源](#): 针对不再使用的Lite Cluster资源，您可以释放资源。

图 5-1 Lite Cluster 资源管理介绍



## 5.2 管理 Lite Cluster 资源池

### Lite Cluster 资源池续费管理

针对包年包月的Lite Cluster资源池，支持续费功能，还可以开通自动续费、修改自动续费。自动续费时系统从可用余额扣款，详情请见[自动续费](#)。

登录[ModelArts管理控制台](#)，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”，进入Lite资源池列表页中操作。

### 查看 Lite Cluster 资源池基本信息

在[ModelArts管理控制台](#)的左侧导航栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”，进入Lite资源池列表页中，单击Lite Cluster资源池名称，可以进入到Lite Cluster资源池详情页中查看更多信息。

图 5-2 查看 Lite Cluster 资源池基本信息



## 管理 Lite Cluster 资源池标签

通过给资源池添加标签，可以标识云资源，便于快速搜索资源池。

1. 登录[ModelArts管理控制台](#)，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”。
  2. 在Lite资源池列表中，单击资源池名称进入资源池详情页面。
  3. 在资源池详情页面，单击“标签”页签查看标签信息。
- 支持添加、修改、删除标签。标签详细用法请参见[ModelArts如何通过标签实现资源分组管理](#)。

图 5-3 标签



### 说明

最多支持添加20个标签。

## Lite Cluster 资源池配置管理

在资源池详情页面，单击“配置管理”，在配置管理页面，可以修改设置监控的命名空间、修改集群配置，配置镜像预热信息。

- 单击监控的图标，可以开启或关闭监控信息，并设置监控的命名空间。监控使用请参考[5.8.2 使用Prometheus查看Lite Cluster监控指标](#)。
- 单击集群配置的图标，可以设置绑核、Dropcache、大页内存参数。缺省值表示读取资源池镜像中的默认值。
  - 绑核：开启CPU绑核表示工作负载实例独占CPU，可以提升应用性能（比如训练作业、推理任务性能），减少应用的调度延迟，适用于对CPU缓存和调度延迟敏感的场景。关闭绑核表示关闭工作负载实例独占CPU的功能，优点是CPU共享池可分配的核数较多。也可关闭系统默认绑核后，在业务容器中用taskset等方式进行灵活绑核。
  - Dropcache：开启后表示启用Linux的缓存清理功能，是一种应用性能调优手段，在大部分场景下可以提升应用性能。但是清除缓存也可能会导致容器启动失败或系统性能暂时下降（因为系统需要重新从磁盘加载数据到内存中）。关闭表示不启用缓存清理功能。
  - 大页内存：开启表示配置使用透明大页功能。大页内存是一种内存管理机制，可以通过增大内存页的大小来提高系统性能。透明大页是动态分配大页内存的机制，可以简化大页内存的管理。开启大页内存也是一种应用调优手段，在大部分场景下可以提升应用性能，但是开启后也会引起soft lockup机制导致节点重启。关闭表示不使用大页内存功能。

- 单击镜像预热的图标，可以设置镜像来源、添加镜像密钥、添加镜像预热配置，具体操作请参见[3.6（可选）配置镜像预热](#)。

## 更多相关操作

其它更多操作如下：

- 节点池管理操作请参见[5.3 管理Lite Cluster节点池](#)
- 节点管理操作请参见[5.4 管理Lite Cluster节点](#)
- 扩缩容Lite Cluster资源池操作请参见[5.5 扩缩容Lite Cluster资源池](#)
- 升级Lite Cluster资源池驱动操作请参见[5.6 升级Lite Cluster资源池驱动](#)
- 升级Lite Cluster资源池单个节点驱动操作请参见[5.7 升级Lite Cluster资源池单个节点驱动](#)

## 5.3 管理 Lite Cluster 节点池

为帮助您更好地管理Kubernetes集群内的节点，ModelArts支持通过节点池来管理节点。一个节点池包含一个节点或多个节点，能通过节点池批量配置一组节点。

### 进入节点池管理页面

- 登录[ModelArts管理控制台](#)，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”。
- 在“轻量算力集群（Lite Cluster）”页面，单击Lite Cluster名称，进入资源详情页。
- 在资源池详情页，单击“节点池管理”页签，您可以创建、更新和删除节点池。

图 5-4 节点池管理



### 创建节点池

- 当您需要更多节点池时，可单击“创建节点池”新增节点池，参考[表5-1](#)填写参数。

华东二区域每个Lite Cluster集群最多可创建15个节点池。西南贵阳一每个Lite Cluster集群最多可创建50个节点池。其他区域每个Lite Cluster集群最多可创建10个节点池。

表 5-1 节点池参数说明

参数	说明
节点池名称	<p>新建节点池的名称，可自定义。</p> <p>只能以小写字母开头，由小写字母、数字、中划线（-）组成，不能以中划线（-）结尾，不能以-default结尾。</p>
实例规格	<p>支持CPU、GPU、Ascend三种芯片规格资源，根据实际需要选择。</p> <ul style="list-style-type: none"><li>• CPU：通用计算架构，适合通用任务，计算性能较低，适用于轻量级适合通用任务，计算性能较低。</li><li>• GPU：并行计算架构，适合并行任务，计算性能高，支持多卡分布式训练，适用于深度学习训练、图像处理等场景。</li><li>• Ascend：专用AI架构，适合AI任务，计算性能极高，支持多节点分布式部署，适用于AI模型训练、推理加速等场景。</li></ul>
驱动版本	当实例规格类型为Snt9b、D310P系列规格时，支持选择驱动版本。
操作系统	<p>可以指定实例的操作系统。</p> <ul style="list-style-type: none"><li>• 预置镜像：由华为云官方提供的镜像，覆盖华为自研的HCE OS、EulerOS镜像和第三方商业镜像，您可以根据实际需要选择。<ul style="list-style-type: none"><li>- Huawei Cloud EulerOS镜像：Huawei Cloud EulerOS（简称HCE）是基于openEuler构建的云上操作系统。HCE打造云原生、高性能、高安全、易迁移等能力，加速用户业务上云，提升用户的应用创新空间，可替代CentOS、EulerOS等公共镜像。</li><li>- 华为自研EulerOS镜像：EulerOS是基于开源技术的企业级Linux操作系统软件，具备高安全性、高可扩展性、高性能等技术特性，能够满足客户IT基础设施和云计算服务等多业务场景需求。</li></ul></li><li>• 说明<ul style="list-style-type: none"><li>EulerOS是基于开源操作系统openEuler进行开发的华为内部的操作系统。</li><li>- 第三方商业镜像：经华为云严格测试并制作发布，皆已正版授权，能够保证镜像安全、稳定。</li></ul></li><li>• 私有镜像：由用户创建或导入的个人镜像，仅用户自己可见。包含操作系统、预装的公共应用以及用户的私有应用。如果选择“私有镜像”，请提前在镜像服务IMS创建系统镜像，或者导入私有镜像到IMS，详情可参考<a href="#">镜像服务创建私有镜像</a>。</li></ul>

参数	说明
可用区	<p>根据实际情况选择“随机分配”或“指定可用区”。可用区是在同一区域下，电力、网络隔离的物理区域。可用区之间内网互通，不同可用区之间物理隔离。</p> <ul style="list-style-type: none"><li>随机分配：系统自动分配可用区。</li><li>指定可用区：指定资源池实例在哪个可用区域。考虑系统容灾时，推荐指定实例在同一个可用区。可设置可用区的实例数。</li></ul>
目标实例数	<p>选择节点池的节点个数，数量越多，计算性能越强。</p> <p>当“可用区”选择“指定可用区”时，实例数量会根据可用区的数据自动计算，此处无需再次设置。</p> <p>单次创建时，实例数建议不大于30，否则可能触发限流导致创建失败。</p> <p>目标总实例数不能超过节点池集群规模，如果节点池集群规模选择默认，目标总实例数不能超过50，具体请以控制台界面为准。</p> <p>部分区域的部分规格支持整柜购买，此时实例数会显示为“数量*整柜”，购买的实例总数为两者的乘积。整柜购买可实现不同任务间的物理隔离，避免通信冲突，在任务规模增大的同时保证计算性能线性度不下降。整柜下的实例生命周期需保持一致，需要一起创建、一起删除。</p> <p>超节点规格，即Snt9b23类型实例规格，支持自定义步长购买，此时实例数会显示为“数量*步长”，购买的实例总数为两者的乘积。步长为每次调整保障配额时的最小单位，在节点绑定场景下每个步长内的节点将作为一个整体，且属于同一批次。</p>
虚拟私有云	默認為CCE集群所在VPC网络，不可修改。
K8S标签	设置附加到Kubernetes对象（比如Pod）上的键值对。最多可以添加20条标签。使用该标签可区分不同节点，可结合工作负载的亲和能力实现容器Pod调度到指定节点的功能。
污点	默認為空。支持给节点加污点来设置反亲和性，每个节点最多配置20条污点。
容器引擎	<p>容器引擎是Kubernetes最重要的组件之一，负责管理镜像和容器的生命周期。Kubelet通过Container Runtime Interface (CRI) 与容器引擎交互，以管理镜像和容器。此处支持选择Docker和Containerd。Containerd和Docker的详细差异对比请见<a href="#">容器引擎</a>。</p> <p>如果CCE集群版本低于1.23，仅支持选择Docker作为容器引擎。如果CCE集群版本大于等于1.27，仅支持选择Containerd作为容器引擎。其余CCE集群版本，支持选择Containerd或Docker作为容器引擎。</p>
节点子网	选择同一VPC网络下的子网作为节点子网，新创建的节点池将会使用该子网资源。

参数	说明
关联安全组	用于指定节点池创建出来的节点使用的安全组。最多选择4个安全组。节点安全组需要放通一些端口以保障节点通信。如果不关联安全组将会使用集群中默认的节点安全组规则。
资源标签	通过为资源添加标签，可以对资源进行自定义标记，实现资源分类。
安装后执行脚本	请输入脚本命令，命令中不能包含中文字符，需传入Base64转码后的脚本，转码后的字符数不能超过2048。脚本将在Kubernetes软件安装后执行，不影响Kubernetes软件安装。 请不要在安装后执行脚本中使用reboot命令立即重启，如果需要重启，可以使用“shutdown -r 1”命令延迟1分钟重启。
节点计费模式	用户增加节点数量时，可以打开“节点计费模式”开关，为新创建的节点指定不同于资源池的计费模式或购买时长。 不选择时计费信息默认和资源池保持一致。例如用户可以在包周期的资源池中创建按需的节点。若用户不指定该参数，则新增的节点计费模式和资源池保持一致。 如果新创建的节点计费模式选择包周期，则需要选择勾选新增节点是否自动续费。勾选自动续费后，新增节点到期后会自动续期。 如果原节点池的计费模式为包周期，打开“节点计费模式”开关，修改新创建节点的计费说明时，如果计费模式仍为包周期，计费周期不能设置晚于原节点池的计费周期。例如原节点池的计费模式为包周期且6个月以后到期，增加节点数量时，新的节点计费说明选择包周期时，计费周期不能晚于6个月以后。

2. 确认配置信息，鼠标移至配置费用，可查看并确认费用明细，确认完成后，单击“确认”。
3. 在弹框中确认是否勾选新增节点自动续费，单击“确定”。  
创建完成可以在节点池管理页面查看已创建的节点池信息。

## 节点池配置弹性伸缩

根据Pod调度状态及资源使用情况对节点池的节点进行自动扩容缩容，同时支持多可用区、多实例规格、指标触发和周期触发等多种伸缩模式，满足不同的节点伸缩场景。

节点池使用弹性伸缩功能前，需要安装集群弹性引擎插件，更多详情请见[6.6 集群弹性引擎](#)。

## 查看节点列表

当您想查看某一节点池下的节点相关信息，可单击操作列的“节点列表”，可查询节点的名称、规格及可用区。

## 更新节点池

1. 当您想更新节点池配置时，可单击操作列的“更多>修改配置”，对配置进行更新操作。相关参数请参见[表5-1](#)。

需注意以下事项：

- 目标总实例数不能超过节点池集群规模，如果节点池集群规模选择默认，目标总实例数不能超过50，具体请以控制台界面为准。
- 更新节点池配置时，高级配置仅对新增的节点生效，其中“存量节点标签及污点”、“存量节点资源标签”支持对存量节点同步改动（勾选对应的复选框）。

节点池中更新的“资源标签”信息会同步到节点上。

图 5-5 更新节点池



2. 确认配置信息，鼠标移至配置费用，可查看并确认费用明细，确认完成后，单击“确认”。
3. 在弹框中确认是否勾选新增节点自动续费，单击“确定”。

更新完成可以在节点池管理页面查看已更新的节点池信息。

## 升级 Lite Cluster 资源池驱动

当Lite Cluster资源池中的节点含有GPU/Ascend资源时，资源池节点性能如果无法满足现有业务，升级驱动可以修复已知问题、提升性能或者支持新功能，确保资源池性能和兼容性得到优化。

可单击操作列的“更多>驱动升级”，升级Lite Cluster资源池GPU/Ascend驱动，详情请见[5.6 升级Lite Cluster资源池驱动](#)。

## 删除节点池

当有多个节点池时，支持删除节点池，此时在操作列会显示“删除”按钮，确认会影响的关联资源和关联作业，单击“删除”后输入“DELETE”并单击“确定”即可。

针对未退订或释放的包年/包月的节点，请单击“立即前往”，前往资源池详情页[删除/退订/释放节点](#)。

每个资源池至少需要有一个节点池，当只有一个节点池时不支持删除。

## 查看节点池的存储配置

在节点池管理的“更多>修改配置”页面，可以查看该节点池配置的系统盘、容器盘或数据盘的磁盘类型、大小、数量、写入模式、容器引擎空间大小等参数。

图 5-6 修改节点池配置



在Lite资源池的扩缩容页面，也可以查看节点池的存储配置信息。

## 查找搜索节点池

在节点池管理页面的搜索栏中，支持通过节点池名称、规格、容器引擎空间大小、可用区等关键字搜索节点池。

## 设置节点池列表显示信息

在节点池管理页面中，单击右上角的设置图标，支持对节点池列表中显示的信息进行自定义。

# 5.4 管理 Lite Cluster 节点

节点是容器集群组成的基本元素，在资源池详情页，单击“节点管理”页签，进行替换、删除、重置、续费等操作。当把鼠标放在节点名称上方时，会显示资源ID，资源ID可用于查询账单或者在费用中心查询包周期资源的计费信息。

## 删除/退订/释放节点

- 如果是“按需计费”的资源池，您可单击操作列的“删除”，在文本框中输入“DELETE”，单击“确定”，确认删除，即可实现对单个节点的资源释放。  
如果想批量删除节点，勾选待删除节点名称前的复选框，然后单击名称上方的“删除”，在文本框中输入“DELETE”，单击“确定”，确认删除，即可实现对多个节点的资源释放。
- 如果是“包年/包月”且资源未到期的资源池，您可单击操作列的“退订”，即可实现对节点的资源释放。支持批量退订节点。
- 如果是“包年/包月”且资源到期的资源池（处于宽限期），您可单击操作列的“释放”，即可实现对单个节点的资源释放。不支持批量释放处于宽限期的节点。  
部分“包年/包月”节点会出现“删除”按钮，原因是该节点为存量节点，单击“删除”即可实现节点的资源释放。

### □ 说明

- 删除/退订/释放节点可能导致该节点上运行的作业失败，请保证该节点无任务运行时再进行操作。
- 当资源池中存在异常节点时，可通过删除/退订/释放操作，将资源池中指定的异常节点移除，再通过扩容专属资源池获得和之前相同的总节点个数。
- 仅有一个节点时，无法进行删除/退订/释放操作。

## 开启/关闭删除锁

为了防止节点被误删除或退订，您可以根据业务对节点开启删除锁。开启删除锁的节点将无法正常使用删除/退订功能，需要关闭删除锁才可以进行删除/退订。

### □ 说明

- 仅支持对资源池中的节点开启删除锁功能进行节点保护，暂不支持对未纳管到资源池中的游离节点开启删除锁功能。
- 开启删除锁功能仅对节点删除/退订操作进行限制，节点替换、重启节点、重置节点等其他操作不受限制，删除包含开启删除锁节点的资源池操作也不受限制。
- 开启删除锁：单击操作列的“更多>开启删除锁”，在对话框中确认即将开启删除锁的节点信息，确认完后在文本框输入“YES”，单击“确定”，即可对节点开启删除锁。  
如果想批量对多个节点开启删除锁，勾选待开启删除锁的节点名称前的复选框，然后单击名称上方的“更多>开启删除锁”，即可实现对多个节点开启删除锁。
- 关闭删除锁：单击操作列的“更多>关闭删除锁”，在对话框中确认即将关闭删除锁的节点信息，确认完后在文本框输入“YES”，单击“确定”，即可对节点关闭删除锁。  
如果想批量对多个节点关闭删除锁，勾选待关闭删除锁的节点名称前的复选框，然后单击名称上方的“更多>关闭删除锁”，即可关闭多个节点的删除锁。

## 查询插件组件

在资源池详情页面的“节点管理”页签，可以查看当前节点的插件资源占用情况。

单击节点操作列的“更多>查询插件组件”，可在“组件列表”弹框中查看与插件相关的实例资源占用情况。

图 5-7 实例列表

组件名称	工作空间	部署方式	副本数 (个)	CPU配额	内存配额
modelarts-metric-col...	default	DaemonSet	--	申请 100m 限制 700m	申请 100Mi 限制 500Mi
modelarts-metric-col...	default	DaemonSet	--	申请 100m 限制 700m	申请 100Mi 限制 500Mi

总条数: 2

10 < 1 >

## 续费/开通自动续费/修改自动续费

对于包年/包月的节点，在“节点管理”页签中提供了续费、开通自动续费和修改自动续费功能，并支持对多个节点进行批量操作。

自动续费时系统从可用余额扣款，详情请见[自动续费](#)。

## 重置节点

“节点管理”页签中提供节点重置的功能。单击操作列的“重置”，可实现对单个节点的重置。勾选多个节点的复选框，单击节点列表上方的“更多>重置”按钮，可实现对多个节点的重置。

下发重置节点任务时需要填写以下参数。

表 5-2 重置参数说明

参数名称	说明
操作系统	选择下拉框中支持的操作系统。
配置方式	选择重置节点的配置方式。 <ul style="list-style-type: none"><li>按节点比例：重置任务包含多个节点时，可以设置同时被重置节点的最高比例。</li><li>按实例数量：重置任务包含多个节点时，可以设置同时被重置节点的最大个数。</li></ul>
驱动版本	可以在下拉框中指定重置节点的驱动版本。

单击“操作记录”可查看当前资源池重置节点的操作记录。重置中节点状态为“重置中”，重置成功后，节点状态变为“可用”）。重置节点操作不会收取费用。

## 说明书

- 重置节点将影响相关业务的运行，重置时本地盘会被清空、节点上的k8s标签会被清除，请谨慎操作。
- 节点状态为“可用”或“不可用”的节点才能进行重置。
- 同一时间单个节点只能处于一个重置任务中，无法对同一个节点同时下发多个重置任务。
- 当操作记录里有节点处于替换中时，该资源池无法进行重置节点操作。
- 当资源池处于驱动升级状态时，该资源池无法进行重置节点操作。
- GPU和NPU规格，重置节点完成后，节点可能会出现驱动升级的现象，请耐心等待。

## 节点排水

节点排水通常指在集群管理中，将某个节点上的工作负载（如Pods）安全地迁移到其他节点上，并将该节点标记为不可调度状态的过程。您可以通过控制台使用节点排水功能，安全地将节点上的Pod驱逐，后续新建的Pod都不会再调度到该节点。

在节点故障等场景下，节点排水功能可帮助您快速排空节点，将故障节点进行隔离，原节点上被驱逐的Pod将会由工作负载controller转移到其他正常可调度的节点上。

仅当节点状态为“可用-可调度”状态，才能执行节点排水操作。

### ⚠ 警告

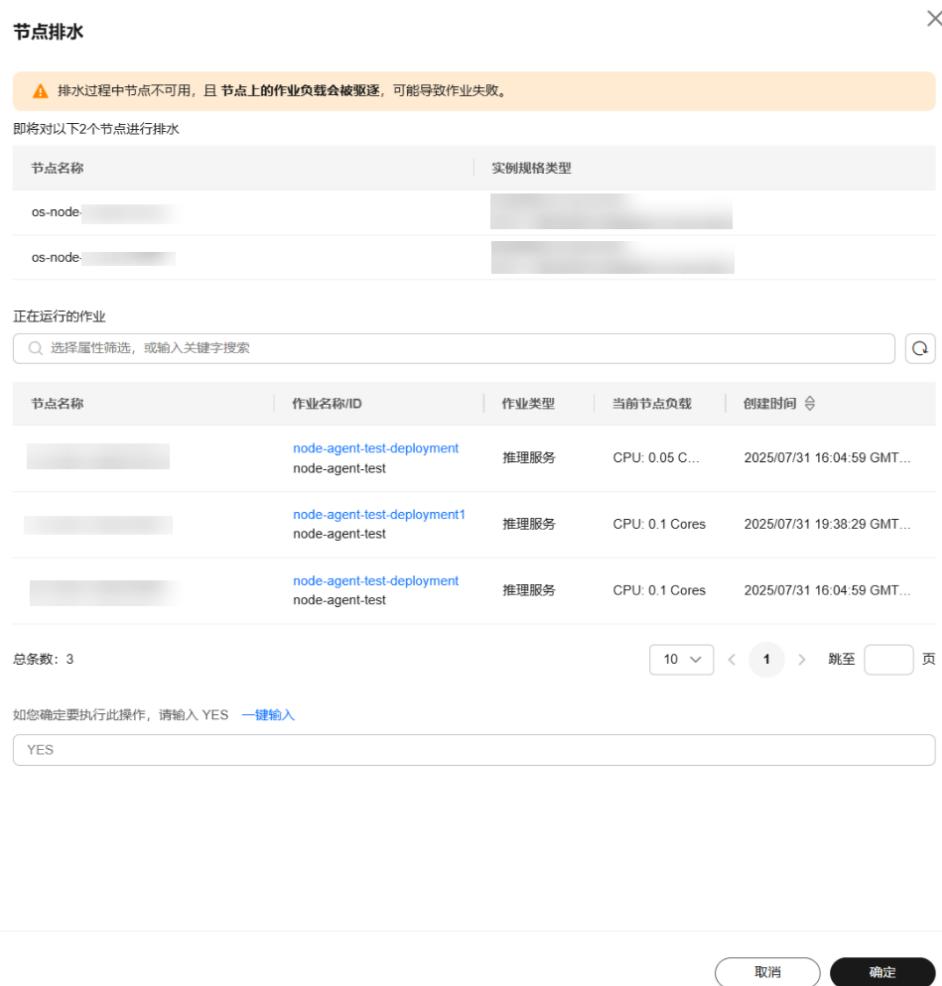
排水过程中节点不可用，且节点上的作业负载会被驱逐，可能导致作业失败。

- 登录[ModelArts管理控制台](#)，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”，进入“轻量算力集群（Lite Cluster）”页面。
- 在资源池列表中，单击某一资源池名称，进入资源池详情页。
- 在资源池详情页“节点管理”页签，根据业务选择节点排水。
  - 单节点排水：单击操作列的“更多>节点排水”。
  - 批量节点排水：勾选多个节点，单击节点列表上方的“更多>节点排水”。
- 在弹窗确认待排水的节点信息和节点正在运行的作业，单击“一键输入”，在输入框中输入YES，单击“确定”。

当节点状态为“可用-排水成功”时，表示排水作业已完成。

当节点状态为“可用-排水失败”时，可将鼠标悬停在节点状态，查看失败原因。

图 5-8 节点排水



## 事件中心页面授权运维

针对ModelArts运维平台告警的故障节点，控制台“资源管理>事件中心”页面记录故障节点的计划事件，包括故障节点的基本信息、事件类型、事件状态、事件描述等，并支持授权和重部署操作，授权华为技术支持对故障节点进行运维。

- **授权操作可执行条件**

故障节点可执行授权操作的事件类型和事件状态如**表5-3**所示。

**表 5-3 授权操作执行条件**

事件类型	事件状态	可执行授权操作
系统维护	待授权	授权、重部署
本地盘恢复	待授权	<p>授权、重部署 本地盘修复后，请通过<a href="#">重置节点</a>完成对分区的修复。</p> <p><b>警告</b> 授权后本地盘恢复操作将会导致本地盘数据丢失，授权前请先迁移业务和备份数据。</p>

事件类型	事件状态	可执行授权操作
节点重启	待授权	授权
运维授权	待授权	授权
超节点维护	待授权	授权
超节点重部署	待授权	重部署  超节点重部署需要在物理超节点内。在超节点满售时，不支持重部署，操作授权按钮为置灰状态。
超节点本地盘恢复	待授权	授权  <b>警告</b> 授权后本地盘恢复操作将会导致本地盘数据丢失，授权前请先迁移业务和备份数据。

- **授权操作**

当故障节点满足如[表5-3](#)所示的条件时，可通过授权操作授权华为技术支持对故障节点进行运维。

您可在控制台“资源管理>事件中心”页面，找到对应节点，在操作列单击“授权”，在弹出的提示框中单击“确认”即可完成授权。

如果计划事件不满足如[表5-3](#)所示的条件，操作授权按钮为置灰状态。

在完成运维操作后，华为云技术支持会主动关闭已获得授权，无需您额外操作。

- **重部署操作**

当故障节点满足如[表5-3](#)所示的重部署操作执行条件时，可通过重部署操作授权华为技术支持对故障节点进行重部署。

在完成运维操作后，华为云技术支持会主动关闭已获得授权，无需您额外操作。

---

**⚠ 警告**

重部署节点恢复更快，但本地盘数据将丢失，请谨慎操作。重部署前请先迁移业务和备份数据。

a. 在控制台“资源管理>事件中心”页面，找到对应节点，在操作列单击“重部署”。

如果计划事件不满足如[表5-3](#)所示的重部署操作执行条件，操作重部署按钮为置灰状态。

b. 确认是否勾选“强制重部署”，并在输入框中输入“YES”，单击“确认”即可完成授权。

由于重部署能力依赖节点的状态，当节点不可用时，无法完成重部署流程，如果勾选强制重部署，当节点不可用时，可通过强制重部署来将节点重部署。

### ⚠ 警告

强制重部署会在节点重部署完成后进行节点重置，会导致服务器的本地盘数据和云盘数据全部丢失，请谨慎操作。

## 重启节点

在节点的操作列，选择“更多>重启”，支持重启单个节点。也可以勾选节点名称，在节点列表上方单击“重启”，进行批量重启节点操作。重启节点将影响相关业务的运行，请谨慎操作。

## 添加/编辑/删除资源标签

资源标签用于方便管理资源的计费账单。

在节点的操作列，选择“更多>编辑资源标签”，支持编辑单个节点的资源标签。

也可以勾选节点名称，在节点列表上方单击“更多 > 添加/编辑资源标签”或者“删除资源标签”，批量操作节点资源标签。

图 5-9 添加/编辑/删除资源标签



## 导出节点数据

支持导出Lite资源池的节点信息到Excel表格中，方便查阅。

勾选节点名称，在节点列表上方单击“导出 > 导出全部数据到XLSX”或者“导出 > 导出已选中的数据到XLSX”，在浏览器的下载记录 中查看导出的Excel表格。

## 驱动升级

支持升级Lite资源池内单个节点驱动版本，或批量升级多个节点的驱动版本。详情请参见[5.7 升级Lite Cluster资源池单个节点驱动](#)章节。

## 查找搜索节点

在节点管理页面的搜索栏中，支持通过节点名称、状态、批次、驱动版本、驱动状态、IP地址、节点池、资源标签等关键字搜索节点。

## 设置节点列表显示信息

在节点管理页面中，单击右上角的设置图标 ，支持对节点列表中显示的信息进行自定义。

## 常见问题

### 重置节点后无法正常使用？

当ModelArts Lite的CCE集群在资源池上只有一个节点，且用户设置了volcano为默认调度器时，在ModelArts侧进行重置节点的操作后，节点无法正常使用，节点上的POD会调度失败。

具体原因分析和处理方法请见[重置节点后无法正常使用？](#)。

## 5.5 扩缩容 Lite Cluster 资源池

### 场景介绍

当Lite Cluster资源池创建完成，使用一段时间后，由于用户业务的变化，对于资源池资源量的需求可能会产生变化，面对这种场景，ModelArts Lite Cluster资源池提供了扩缩容功能，用户可以根据需求动态调整资源。

对已有规格实例数扩缩容，即增加或减少资源池已有规格的实例数量，增加实例数即扩容，减少实例数即缩容。扩缩容规格实例数适用于调整资源池的整体规模，减少资源池中的节点数量来优化资源使用；如果因为资源池节点异常或空闲需要移除特定的节点，请前往资源池详情页面[删除节点](#)。

#### 警告

- 缩容操作可能影响到正在运行的业务，建议用户在业务空窗期进行缩容，或进入资源池详情页面，在指定空闲的节点上进行删除来实现缩容。
- 缩容规格实例数时，资源池中如果包含已开启删除锁的节点，可能会导致开启删除锁的节点被删除，从而中断正在运行的业务，且该动作不可回退，因此建议不要对这些节点进行缩容。如果仍需缩容，请前往资源池详情页[删除节点](#)。

### 计费影响

在增加实例数量时，会产生计算资源的计费。具体费用可参见[ModelArts价格详情](#)。

可以在扩缩容时通过指定节点计费模式，为资源池新创建的节点设置不同于资源池的计费模式。例如用户可以在包周期的资源池中创建按需的节点，如果用户不指定该参数，创建的节点计费模式和资源池保持一致。具体内容如[表5-4](#)所示。

表 5-4 计费项

计费项		计费项说明	适用的计费模式	计费公式
计算资源	专属资源池	使用计算资源的用量。 具体费用可参见 <a href="#">ModelArts价格详情</a> 。	包年/包月	规格单价 * 计算节点个数 * 购买时长

## 前提条件

已经[开通Lite Cluster资源池](#)。

## 约束限制

- 只支持对状态为“运行中”的Lite Cluster资源池进行扩缩容。
- 缩容规格实例数时，当Lite Cluster资源池中只剩一个实例节点时，无法进行缩容操作。因此，缩容操作必须确保至少保留一个节点。
- 包年/包月的资源池仅支持扩容操作。

## 扩缩容 Lite Cluster 资源池

- 登录[ModelArts管理控制台](#)，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”。
- 单击某个资源池操作列的“扩缩容”对资源池进行扩缩容。对于为包周期资源池，此按钮为“扩容”，如果需要缩容，请进入到包周期资源池详情页对节点进行[退订节点](#)操作。
- 在“专属资源池扩缩容”页面，按[表5-5](#)设置扩缩容参数。

表 5-5 专属资源池扩缩容参数说明

参数	说明
实例规格类型	当前待扩缩容的Lite Cluster资源池实例规格类型。不可编辑。
规格内容	当前待扩缩容的Lite Cluster资源池规格内容。不可编辑。
当前实例数	当前待扩缩容的Lite Cluster资源池实例数。不可编辑。

参数	说明
可用区	<p>指定扩缩容完成后节点的可用区分布。可选择“随机分配”和“指定可用区”。</p> <ul style="list-style-type: none"><li>选择随机分配时，扩缩容完成后，节点的可用区分布由系统后台随机选择。</li><li>选择指定可用区时，可指定扩缩容完成后节点的可用区分布。可用区ID对应的实例数总和默认为“目标总实例数”。 比如：<ul style="list-style-type: none"><li>当前实例数为3，可用区ID对应的实例数总和为5，“目标总实例数”默认为5，表示扩容实例数至5。</li><li>当前实例数为3，可用区ID对应的实例数总和为2，“目标总实例数”默认为2，表示缩容实例数至2。</li></ul></li></ul>
容器引擎空间限制	<p>扩容资源池时，即“目标总实例数”大于“当前总实例数”时，可以设置新建节点的容器引擎空间大小，可指定“容器引擎空间”大小。</p> <p>此操作会导致资源池内该规格下节点的dockerBaseSize不一致，可能会使得部分任务在不同节点的运行情况不一致，请谨慎操作。<b>存量节点不支持修改容器引擎空间大小</b>。</p>
容器引擎空间大小	当“容器引擎空间限制”选择“指定大小”时，设置新建节点的容器引擎空间大小。
目标总实例数	<p>通过设置目标总实例数实现扩缩容。请用户根据本身业务诉求进行调整。</p> <ul style="list-style-type: none"><li>扩容：设置“目标总实例数”大于“当前总实例数”。</li><li>缩容：设置“目标总实例数”小于“当前总实例数”。</li></ul> <p>如果“可用区”选择“指定可用区”，不用另外设置“目标总实例数”。“目标总实例数”默认为可用区ID对应的实例数总和。</p> <p>如果购买资源池时，节点数量采用整柜方式购买（部分规格支持），则在扩缩容时为整柜方式扩缩容，目标实例总数等于“数量*整柜”。“整柜”参数为创建资源池时选择，扩缩容时不可修改。用户通过增减“数量”来改变“目标总实例数”。</p> <p>如果购买资源池时，实例规格为Snt9b23类型，即超节点规格，实例数量采用步长方式购买，则在扩缩容时为步长方式扩缩容，目标实例总数等于“数量*步长”。“步长”参数为创建资源池时选择，扩缩容时不可修改。用户通过增减“数量”来改变“目标总实例数”。</p>
节点池名称	当前待扩缩容的Lite Cluster资源池名称。不可编辑。

参数	说明
容器引擎	<p>容器引擎是Kubernetes最重要的组件之一，负责管理镜像和容器的生命周期。Kubelet通过Container Runtime Interface (CRI) 与容器引擎交互，以管理镜像和容器。其中Containerd调用链更短，组件更少，更稳定，占用节点资源更少，Containerd和Docker差异对比请见<a href="#">容器引擎</a>。</p> <p>如果CCE集群版本低于1.23，仅支持选择Docker作为容器引擎。如果CCE集群版本大于等于1.27，仅支持选择Containerd作为容器引擎。其余CCE集群版本，支持选择Containerd或Docker作为容器引擎。</p>
操作系统	<p>可以指定实例的操作系统。</p> <ul style="list-style-type: none"><li>预置镜像：由华为云官方提供的镜像，覆盖华为自研的HCE OS、EulerOS镜像和第三方商业镜像，您可以根据实际需要选择。<ul style="list-style-type: none"><li>- Huawei Cloud EulerOS镜像：Huawei Cloud EulerOS（简称HCE）是基于openEuler构建的云上操作系统。HCE打造云原生、高性能、高安全、易迁移等能力，加速用户业务上云，提升用户的应用创新空间，可替代CentOS、EulerOS等公共镜像。</li><li>- 华为自研EulerOS镜像：EulerOS是基于开源技术的企业级Linux操作系统软件，具备高安全性、高可扩展性、高性能等技术特性，能够满足客户IT基础设施和云计算服务等多业务场景需求。</li></ul></li><li><b>说明</b><p>EulerOS是基于开源操作系统openEuler进行开发的华为内部的操作系统。</p><ul style="list-style-type: none"><li>- 第三方商业镜像：经华为云严格测试并制作发布，皆已正版授权，能够保证镜像安全、稳定。</li></ul></li><li>私有镜像：由用户创建或导入的个人镜像，仅用户自己可见。包含操作系统、预装的公共应用以及用户的私有应用。如果选择“私有镜像”，请提前在镜像服务IMS创建系统镜像，或者导入私有镜像到IMS，详情可参考<a href="#">镜像服务创建私有镜像</a>。</li></ul>
驱动版本	当实例规格类型为Snt9b、D310P系列规格时，支持选择驱动版本。

4. 指定节点计费模式。用户增加节点数量时，可以打开“节点计费模式”开关，为资源池新扩容的节点设置不同于资源池的计费模式、购买时长和开启自动续费功能。例如用户可以在包周期的资源池中创建按需的节点。若用户不指定该参数，则新扩容的节点计费模式和资源池保持一致。
5. 设置完成后，单击“提交”，在弹出的确认框中单击“确定”完成扩缩容。  
在轻量算力集群（Lite Cluster）页面查看资源池的节点总数是否与设置的“目标总实例数”一致。

## 相关操作

- **5.4 管理Lite Cluster节点**: 因为资源池节点异常或空闲需要移除特定的节点，可前往资源池详情页面删除指定节点或批量删除节点。同时对资源池节点可进行替换、重置、续费等操作。
- **5.6 升级Lite Cluster资源池驱动**: 当专属资源池中的节点含有GPU/Ascend资源时，可基于自己的业务升级专属资源池GPU/Ascend驱动的能力。

## 5.6 升级 Lite Cluster 资源池驱动

### 场景介绍

当Lite Cluster资源池中的节点含有GPU/Ascend资源时，资源池节点性能如果无法满足现有业务，升级驱动可以修复已知问题、提升性能或者支持新功能，确保资源池性能和兼容性得到优化。

ModelArts提供升级Lite Cluster资源池GPU/Ascend驱动的功能，您可根据自身业务需要通过ModelArts控制台升级Lite Cluster资源池GPU/Ascend驱动。

### 安全升级与强制升级对比

驱动升级有两种升级方式：安全升级、强制升级，对比如下。

表 5-6 安全升级与强制升级对比

对比项	安全升级	强制升级
介绍	在节点空闲时进行驱动升级，不会影响正在运行的任务。升级过程平滑，减少对业务的影响。 开始升级后会先将节点进行隔离（不能再下发新的作业），待节点上的存量作业运行完成后进行升级，由于需要等待存量作业执行完毕，故升级周期可能比较长。	忽略节点上正在运行的任务，直接进行驱动升级。 升级速度快，无需等待节点空闲。
适用场景	非紧急情况，逐步升级。	紧急情况，快速完成升级。
注意事项	需等待节点空闲，升级周期较长。 在升级前，建议提前安排节点空闲时间，以减少对业务的影响。	可能导致正在运行的任务中断或失败。需谨慎使用，避免对业务造成影响。

### 约束限制

- Lite Cluster资源池状态处于运行中，且专属池中的节点需要含有GPU/Ascend资源。
- 升级需要重启节点，建议在低峰期进行，以避免影响正在运行的任务，可前往资源池详情页“节点管理”页面查看节点资源占用情况。

**⚠ 警告**

升级驱动会重启节点。如果主机进行过差异化配置，重启节点可能会导致配置丢失，需谨慎考虑。

## 升级 Lite Cluster 资源池 GPU/Ascend 驱动

1. 登录[ModelArts管理控制台](#)，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”。在资源池列表中，选择需要进行驱动升级的资源池“[...](#) > 驱动升级”。  
或者在资源池列表单击资源池名称，进入资源池详情页，切换至“节点池管理”页签，单击节点池操作列“更多>驱动升级”。
2. 在“驱动升级”弹窗中，会显示当前Lite Cluster资源池的驱动类型、实例数、当前版本、目标版本、升级方式、升级范围和开启滚动开关。按[表5-7](#)设置驱动升级参数。

**表 5-7 驱动升级参数说明**

参数	说明
目标版本	在目标版本下拉框中，选择当前驱动待升级的目标驱动版本。 对于资源池新增加的节点，可能会与资源池原有节点驱动不一致，为了保持驱动一致，目标版本可选择当前驱动版本，升级完成后所有节点驱动会升级为统一版本。
升级方式	可选择安全升级或强制升级，具体对比请见 <a href="#">安全升级与强制升级对比</a> 。 <ul style="list-style-type: none"><li>安全升级：待节点上没有作业时再升级，该方式升级周期可能比较长。</li><li>强制升级：忽略运行中作业，直接升级，可能会导致运行中作业失败。</li></ul>
开启滚动	开启开关后，支持滚动升级的方式升级驱动。 滚动升级是一种逐步替换实例的升级方式，适用于需要保持服务连续性的场景。通过分批次升级实例，确保在升级过程中始终有部分实例正常运行，从而减少停机时间。 滚动驱动升级时，驱动异常的节点对升级无影响，会和驱动正常的节点一起升级。

参数	说明
滚动方式	<p>当前支持“按节点比例”和“按实例数量”两种滚动方式。</p> <ul style="list-style-type: none"><li>按节点比例：每批次驱动升级的实例数量为“节点比例*资源池实例总数”。</li><li>按实例数量：每批次驱动升级的实例数量为设置的实例数量。</li></ul> <p>对于不同的升级方式，滚动升级选择节点的策略会不同：</p> <ul style="list-style-type: none"><li>如果“升级方式”为“安全升级”，则根据滚动实例数量选择无业务的节点，隔离节点并滚动升级。 无业务节点定义：在资源池详情“节点”页签下，如果GPU/Ascend的可用数等于总数，则为无业务节点。</li><li>如果“升级方式”为“强制升级”，则根据滚动实例数量随机选择节点，隔离节点并滚动升级。</li></ul>
节点比例	“滚动方式”选择“按节点比例”时，需要设置每批次驱动升级的实例数量比例，每批次驱动升级的实例数量为“节点比例*资源池实例总数”。
实例数	“滚动方式”选择“按实例数量”时，需要设置每批次驱动升级的实例数量。

图 5-10 驱动升级



3. 设置完成后，单击“确定”开始升级驱动。

在资源池列表中，选择目标资源池，单击操作列中的“...”，然后选择“驱动升级”。在弹出的“驱动升级”页面中，查看当前版本和目标版本是否一致。如果一致，说明驱动已成功升级。

## 5.7 升级 Lite Cluster 资源池单个节点驱动

### 场景介绍

当Lite Cluster资源池中的节点含有GPU/Ascend资源时，资源池节点性能如果无法满足现有业务，升级驱动可以修复已知问题、提升性能或者支持新功能，确保资源池性能和兼容性得到优化。

ModelArts提供升级Lite Cluster资源池GPU/Ascend驱动的功能，您可根据自身业务需要通过ModelArts控制台升级Lite Cluster资源池GPU/Ascend驱动。

## 约束限制

Lite Cluster资源池节点驱动状态处于“运行中”，且专属池中的节点需要含有GPU/Ascend资源。

## 节点驱动升级操作

1. 登录[ModelArts管理控制台](#)，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”。
2. 进入资源池详情页，在节点管理页面，选择需要进行驱动升级的节点，单击操作列的“更多 > 驱动升级”。
3. 在“驱动升级”弹窗中，会显示当前专属资源池节点的名称ID、规格和驱动版本号，选择节点待升级的“升级版本”。
4. 单击“确定”，开始升级单个节点的驱动。

在资源池详情页“节点管理”页面，选择目标资源池，单击操作列中的“更多”。如果“驱动升级”按钮置灰，说明驱动已成功升级。

## 5.8 监控 Lite Cluster 资源

### 5.8.1 使用 AOM 查看 Lite Cluster 监控指标

ModelArts Lite Cluster会定期收集资源池中各节点的关键资源（GPU、NPU、CPU、Memory等）的使用情况并上报到AOM，用户可直接在AOM上查看默认配置好的基础指标，也支持用户自定义一些指标项上报到AOM查看。

此外，还支持在ModelArts Lite Cluster上安装Prometheus开源监控工具，方便用户使用Prometheus工具在Lite Cluster集群内直接采集监控指标数据，具体参见[5.8.2 使用Prometheus查看Lite Cluster监控指标](#)章节。

本章节主要介绍如何在AOM上查看Lite Cluster监控指标。

#### AOM 上查看已有监控指标

1. 登录[控制台](#)，搜索AOM，进入“应用运维管理 AOM”控制台。
2. 单击“监控 > 指标浏览”，进入“指标浏览”“页面”，单击“添加指标查询”。

图 5-11 示例图片



3. 添加指标查询信息。
  - 添加方式：选择“按指标维度添加”。
  - 指标名称：在右侧下拉框中选择“全量指标”，然后选择想要查询的指标，参考[表5-8、表5-9](#)
  - 指标维度：填写过滤该指标的标签。
4. 单击确定，即可出现指标信息。

## 自定义监控指标上报到 AOM

用户有一些自定义的指标数据需要保存到AOM，ModelArts提供了命令方式将用户的自定义指标上报保存到AOM。

### 约束与限制

- ModelArts默认以30秒/次的频率调用自定义配置中提供的命令或http接口获取指标数据。
- 自定义配置中提供的命令或http接口返回的指标数据文本不能大于240KB。
- period采集周期（秒），默认30s采集一次。配置时候需配置30整数倍，最小采集频率30s。

### 命令方式采集自定义指标数据

用于创建自定义指标采集POD的YAML文件示例如下。

```
apiVersion: v1
kind: Pod
metadata:
  name: my-task
  annotations:
    ei.huaweicloud.com/metrics: '[{"customMetrics":[{"containerName":"my-task","exec":{"command":["cat","/metrics/task.prom"]}, "period":30}]}]' # ModelArts从哪个容器以及使用哪个命令获取指标数据，请根据实际情况替换containerName参数和command参数。period采集周期，单位秒，不配置默认30s采集一次。
spec:
  containers:
```

```
- name: my-task
  image: my-task-image:latest # 替换为实际使用的镜像
```

业务负载和自定义指标采集可以共用一个容器，也可以由SideCar容器采集指标数据，然后将自定义指标采集容器指定到SideCar容器，这样可以不占用业务负载容器的资源。

### 自定义指标数据格式

自定义指标数据的格式必须是符合open metrics规范的文本，即每个指标的格式应为：

```
<指标名称>{<标签名称>=<标签值>,...} <采样值> [毫秒时戳]
```

举例如下（#开头为注释，非必需）：

```
# HELP http_requests_total The total number of HTTP requests.
# TYPE http_requests_total gauge
http_requests_total{method="post",code="200"} 1656 1686660980680
http_requests_total{method="post",code="400"} 2 1686660980681
```

## 容器级别的监控指标介绍

表 5-8 容器级别的指标

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
CPU	CPU 使用率	ma_container_cpu_util	该指标用于统计测量对象的CPU使用率。	百分比 (Percent)	0 ~ 100 %	连续2个周期原始值 > 95%	建议	排查是否符合业务资源使用预期，如果业务无问题，无需处理。
	CPU 内核占用量	ma_container_cpu_used_core	该指标用于统计测量对象已经使用的CPU核个数	核 (Core)	≥0	NA	NA	NA
	CPU 内核总量	ma_container_cpu_limit_core	该指标用于统计测量对象申请的CPU核总量。	核 (Core)	≥1	NA	NA	NA
	CPU 显存使用率	ma_container_gpu_memory_util	该指标用于统计测量对象已使用的显存占显存容量的百分比。	百分比 (Percent)	0 ~ 100 %	连续2个周期原始值 > 95%	建议	排查是否符合业务资源使用预期，如果业务无问题，无需处理。

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
内存	内存总量	ma_container_memory_capacity_megabytes	该指标用于统计测量对象申请的物理内存总量。	兆字节 ( Megabytes )	≥0	NA	NA	NA
	物理内存使用率	ma_container_memory_util	该指标用于统计测量对象已使用内存占申请物理内存总量的百分比。	百分比 ( Percent )	0 ~ 100 %	连续2个周期原始值 > 95%	建议	排查是否符合业务资源使用预期，如果业务无问题，无需处理。
	物理内存使用量	ma_container_memory_used_megabytes	该指标用于统计测量对象实际已经使用的物理内存(对应 container_memory_working_set_bytes 当前内存工作集 ( working set ) 使用量。( 工作区内存使用量=活跃的匿名页和缓存，以及 file-baked 页 <= container_memory_usage_bytes ))	兆字节 ( Megabytes )	≥0	NA	NA	NA
存储	磁盘读取速率	ma_container_disk_read_kilobytes	该指标用于统计每秒从磁盘读出的数据量。	千字节/秒 ( Kilobytes/Second )	≥0	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
	磁盘写入速率	ma_container_disk_write_kilobytes	该指标用于统计每秒写入磁盘的数据量。	千字节/秒 ( Kilobytes/Second )	≥0	NA	NA	NA
GPU 显存	GPU显存容量	ma_container_gpu_memory_total_megabytes	该指标用于统计训练任务的显存容量。	兆字节 ( Megabytes )	>0	NA	NA	NA
	GPU显存使用率	ma_container_gpu_memory_util	该指标用于统计测量对象已使用的显存占显存容量的百分比。	百分比 ( Percent )	0 ~ 100 %	NA	NA	NA
	GPU显存使用量	ma_container_gpu_memory_used_megabytes	该指标用于统计测量对象已使用的显存。	兆字节 ( Megabytes )	≥0	NA	NA	NA
	GPU显存空闲容量	ma_container_gpu_memory_free_megabytes	该指标用于统计测量空闲的显存。	兆字节 ( Megabytes )	≥0	NA	NA	NA
	GPU	GPU使用率	ma_container_gpu_util	该指标用于统计测量对象的GPU使用率。	百分比 ( Percent )	0 ~ 100 %	连续2个周期原始值 > 95%	建议 排查是否符合业务资源使用预期，如果业务无问题，无需处理。

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
	GPU内存带宽利用率	ma_container_gpu_memory_copy_util	表示内存带宽利用率。以GPU Vnt1为例，其最大内存带宽为900 GB/sec，如果当前的内存带宽为450 GB/sec，则内存带宽利用率为50%。	百分比(Percent)	0~100%	NA	NA	NA
	GPU编码器利用率	ma_container_gpu_encode_util	表示编码器利用率	百分比(Percent)	%	NA	NA	NA
	GPU解码器利用率	ma_container_gpu_decode_util	表示解码器利用率	百分比(Percent)	%	NA	NA	NA
	GPU温度	DCGM_FI_DEV_GPU_TEMP	表示GPU温度。	摄氏度(°C)	自然数	NA	NA	NA
	GPU功率	DCGM_FI_DEV_POWER_USAGE	表示GPU功率。	瓦特(W)	>0	NA	NA	NA
	GPU显存温度	DCGM_FI_DEV_MEMORY_TEMP	表示显存温度。	摄氏度(°C)	自然数	NA	NA	NA
网络IO	下行速率	ma_container_network_receive_bytes	该指标用于统计测试对象的入方向网络流速。	字节/秒(Bytes/Second)	≥0	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
网络指标	接收包速率	ma_container_network_receive_packets	每秒网卡接收的数据包个数。	个/秒 ( Packets/Second )	≥0	NA	NA	NA
	下行错包率	ma_container_network_receive_error_packets	每秒网卡接收的错误包个数。	个/秒 ( Packets/Second )	≥0	连续2个周期原始值 > 1	紧急告警	网络丢包，建议提工单联系运维支持，排查网络问题。
	上行速率	ma_container_network_transmit_bytes	该指标用于统计测试对象的出方向网络流速。	字节/秒 ( Bytes/Second )	≥0	NA	NA	NA
	上行错包率	ma_container_network_transmit_error_packets	每秒网卡发送的错误包个数。	个/秒 ( Packets/Second )	≥0	连续2个周期原始值 > 1	紧急告警	网络丢包，建议提工单联系运维支持，排查网络问题。
	发送包速率	ma_container_network_transmit_packets	每秒网卡发送的数据包个数。	个/秒 ( Packets/Second )	≥0	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
NPU	NPU使用率	ma_contains_npu_util	该指标用于统计测量对象的NPU使用率。(即将废弃，替代指标为ma_contains_npu_ai_core_util)。	百分比(Percent)	0~100%	连续2个周期原始值>95%	建议	排查是否符合业务资源使用预期，如果业务无问题，无需处理。
	NPU显存使用率	ma_contains_npu_memory_util	该指标用于统计测量对象已使用的NPU显存占NPU存储容量的百分比。(即将废弃，snt3系列替代指标为ma_contains_npu_ddr_memory_util, snt9系列替代指标为ma_contains_npu_hbm_util)。	百分比(Percent)	0~100%	连续2个周期原始值>98%	建议	排查是否符合业务资源使用预期，如果业务无问题，无需处理。
	NPU显存使用量	ma_contains_npu_memory_used_megabytes	该指标用于统计测量对象已使用的NPU显存。(即将废弃，snt3系列替代指标为ma_contains_npu_ddr_memory_usage_bytes, snt9系列替代指标为ma_contains_npu_hbm_usage_bytes)。	≥0	兆字节(Megabytes)	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
NPU	NPU显存容量	ma_contains_npu_memory_total_megabytes	该指标用于统计测量对象的NPU显存容量。 ( 即将废弃, snt3系列替代指标为 ma_contains_npu_ddr_memory_bytes, snt9系列替代指标为 ma_contains_npu_hbm_bytes )。	>0	兆字节 ( Megabytes )	NA	NA	NA
	NPU整体利用率	ma_contains_npu_general_util	昇腾系列AI处理器NPU整体利用率，包括对AI Core和Vector Core的整体统计。( 驱动版本 24.1.RC2 及其以后支持)	百分比 ( Percent )	0 ~ 100 %	NA	NA	NA
	NPU算子重传成功次数	ma_contains_npu_operator_retry_success_cnt	该指标描述NPU算子重传成功次数 ( A3 Ascend HDK 24.1.RC3.3 版本及以上支持 )	数值	≥0	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
	NPU算子重传失败次数	ma_contains_npu_operator_retry_fail_cnt	该指标描述NPU算子重传失败次数(A3 Ascend HDK 24.1.RC3.3版本及以上支持)	数值	≥0	NA	NA	NA
	NPU借轨通信次数	ma_contains_npu_borrow_comms_cnt	该指标描述NPU借轨通信次数，借轨次数越多，传输效率越低(A3 Ascend HDK 24.1.RC3.3版本及以上支持)	数值	≥0	NA	NA	NA
AI处理器	AI处理器错误码	ma_contains_npu_ai_core_error_code	昇腾系列AI处理器错误码	-	-	连续3个周期原始值>0	紧急告警	卡异常，建议提工单联系运维支持。
	AI处理器健康状态	ma_contains_npu_ai_core_health_status	昇腾系列AI处理器健康状态	-	• 1 : 健康 • 0 : 不健康	连续2个周期原始值为0	紧急告警	卡异常，建议提工单联系运维支持。
	AI处理器功耗	ma_contains_npu_ai_core_power_usage_watts	昇腾系列AI处理器功耗(snt9和snt3为处理器功耗，snt3P为板卡功耗)	瓦特(W)	>0	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
	AI处理器温度	ma_contains_npu_ai_core_temperature_celsius	昇腾系列AI处理器温度	摄氏度 (°C)	自然数	NA	NA	NA
	AI处理器AI COR E利用率	ma_contains_npu_ai_core_util	昇腾系列AI处理器AI Core利用率，在NPU上执行cube算子的繁忙和空闲比	百分比 (Percent)	0~100%	连续2个周期原始值 > 95%	建议	排查是否符合业务资源使用预期，如果业务无问题，无需处理。
	AI处理器Vector COR E利用率	ma_contains_npu_vector_core_util	昇腾系列AI处理器Vector Core利用率，是在NPU上执行vector算子的繁忙和空闲比。	百分比 (Percent)	0~100%	连续2个周期原始值 > 95%	建议	排查是否符合业务资源使用预期，如果业务无问题，无需处理。
	NPU整体利用率	ma_contains_npu_general_util	昇腾系列AI处理器NPU整体利用率，包括对AI Core和Vector Core的整体统计。(驱动版本24.1.RC2及其以后支持)	百分比 (Percent)	0~100%	NA	NA	
	AI处理器AI COR E时钟频率	ma_contains_npu_ai_core_frequency_hertz	昇腾系列AI处理器AI Core时钟频率	赫兹 (Hz)	>0	NA	NA	NA
	AI处理器电压	ma_contains_npu_ai_core_voltage_volts	昇腾系列AI处理器电压	伏特 (V)	自然数	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
	AI处理器DDR内存总量	ma_contains_npu_ddr_memory_bytes	昇腾系列AI处理器DDR内存总量	字节( Byte )	>0	NA	NA	NA
	AI处理器DDR内存使用量	ma_contains_npu_ddr_memory_usage_bytes	昇腾系列AI处理器DDR内存使用量	字节( Byte )	>0	NA	NA	NA
	AI处理器DDR内存利用率	ma_contains_npu_ddr_memory_util	昇腾系列AI处理器DDR内存利用率	百分比( Percent )	0~100%	连续2个周期原始值>95%	建议	排查是否符合业务资源使用预期，如果业务无问题，无需处理。
	AI处理器HBM内存总量	ma_contains_npu_hbm_bytes	昇腾系列AI处理器HBM总内存(昇腾snt9 AI处理器专属)	字节( Byte )	>0	NA	NA	NA
	AI处理器HBM内存使用量	ma_contains_npu_hbm_usage_bytes	昇腾系列AI处理器HBM内存使用量(昇腾snt9 AI处理器专属)	字节( Byte )	>0	NA	NA	NA
	AI处理器HBM内存利用率	ma_contains_npu_hbm_util	昇腾系列AI处理器HBM内存利用率(昇腾snt9 AI处理器专属)	百分比( Percent )	0~100%	连续2个周期原始值>95%	建议	排查是否符合业务资源使用预期，如果业务无问题，无需处理。
	AI处理器HBM内存带宽利用率	ma_contains_npu_hbm_bandwidth_util	昇腾系列AI处理器HBM内存带宽利用率(昇腾snt9 AI处理器专属)	百分比( Percent )	0~100%	连续2个周期原始值>95%	建议	排查是否符合业务资源使用预期，如果业务无问题，无需处理。

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
AI处理器HBM内存时钟频率	AI处理器HBM内存时钟频率	ma_container_npu_hbm_frequency_hertz	昇腾系列AI处理器HBM内存时钟频率(昇腾snt9 AI处理器专属)	赫兹(Hz)	>0	NA	NA	NA
	AI处理器HBM内存温度	ma_container_npu_hbm_temperature_celsius	昇腾系列AI处理器HBM内存温度(昇腾snt9 AI处理器专属)	摄氏度(°C)	自然数	NA	NA	NA
	AI处理器AI CPU利用率	ma_container_npu_ai_cpu_util	昇腾系列AI处理器AI CPU利用率	百分比(Percent)	0~100%	NA	NA	NA
	AI处理器控制CPU利用率	ma_container_npu_ctrl_cpu_util	昇腾系列AI处理器控制CPU利用率	百分比(Percent)	0~100%	NA	NA	NA

## 节点级别的监控指标介绍

表 5-9 节点指标

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
CPU	CPU内核总量	ma_node_cpu_limit_core	该指标用于统计测量对象申请的CPU核总量。	核(Core)	≥1	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
CPU	CPU内核占用	ma_node_cpu_used_core	该指标用于统计测量对象已经使用的CPU核数。	核( Core )	≥0	NA	NA	NA
	CPU使用率	ma_node_cpu_util	该指标用于统计测量对象的CPU使用率。	百分比( Percent )	0~100%	连续2个周期原始值>95%	重要	排查是否符合业务资源使用预期，如果业务无问题，无需处理。
	CPU IO等待时间	ma_node_cpu_iowait_counter	从系统启动开始累计到当前时刻，硬盘IO等待时间	jiffies	≥0	NA	NA	NA
内存	物理内存使用率	ma_node_memory_util	该指标用于统计测量对象已使用内存占申请物理内存总量的百分比。	百分比( Percent )	0~100%	连续2个周期原始值>95%	重要	排查是否符合业务资源使用预期，如果业务无问题，无需处理。
	物理内存容量	ma_node_memory_total_mega_bytes	该指标用于统计测量申请的物理内存总量。	兆字节( Megabytes )	≥0	NA	NA	NA
网络IO	下行Bps	ma_node_network_receive_rate_bytes_seconds	该指标用于统计测试对象的入方向网络流速。	字节/秒( Bytes/Second )	≥0	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
	上行 Bps	ma_node_network_transmit_rate_bytes_seconds	该指标用于统计测试对象的出方向网络流速。	字节/秒 ( Bytes/Second )	≥0	NA	NA	NA
存储	磁盘读取速率	ma_node_disk_read_rate_kilobytes_seconds	该指标用于统计每秒从磁盘读出的数据量。只考虑被容器使用的数据盘。	千字节/秒 ( Kilobytes/Second )	≥0	NA	NA	NA
	磁盘写入速率	ma_node_disk_write_rate_kilobytes_seconds	该指标用于统计每秒写入磁盘的数据量。只考虑被容器使用的数据盘。	千字节/秒 ( Kilobytes/Second )	≥0	NA	NA	NA
	cache空间的总量	ma_node_cache_space_capacity_megabytes	该指标用于统计k8s空间的总容量。	兆字节 ( Megabytes )	≥0	NA	NA	NA
	cache空间的使用量	ma_node_cache_space_used_capacity_megabytes	该指标用于统计k8s空间的使用量。	兆字节 ( Megabytes )	≥0	NA	NA	NA
	cache空间的使用率	ma_node_cache_space_used_percent	该指标用于统计k8s空间的使用率	百分比 ( Percent )	≥0	连续2个周期原始值 > 90 %	紧急	请及时检查，防止磁盘写满影响业务。推荐清理计算节点无效数据。

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
容器空间的总量、使用量、使用率	容器空间的总量	ma_node_container_space_capacity_megabytes	该指标用于统计容器空间的总容量。	兆字节 ( Megabytes )	≥0	NA	NA	NA
	容器空间的使用量	ma_node_container_space_used_capacity_megabytes	该指标用于统计容器空间的使用量。	兆字节 ( Megabytes )	≥0	NA	NA	NA
	容器空间的使用率	ma_node_container_space_used_percent	该指标用于统计容器空间的使用率	百分比 ( Percent )	≥0	连续2个周期原始值 > 90 %	紧急	请及时检查，防止磁盘写满影响业务。推荐清理计算节点无效数据。
GPU	GPU使用率	ma_node_gpu_util	该指标用于统计测量对象的GPU使用率。	百分比 ( Percent )	0 ~ 100%	NA	NA	NA
	GPU显存容量	ma_node_gpu_mem_total_megabytes	该指标用于统计测量对象的显存容量。	兆字节 ( Megabytes )	>0	NA	NA	NA
	GPU显存使用率	ma_node_gpu_mem_util	该指标用于统计测量对象已使用的显存占显存容量的百分比。	百分比 ( Percent )	0 ~ 100%	连续2个周期原始值 > 97 %	提示	排查是否符合业务资源使用预期，如果业务无问题，无需处理。

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
GPU 显存使用量	GPU 显存使用量	ma_node_gpu_mem_used_megabytes	该指标用于统计测量对象已使用的显存。	兆字节 (Megabytes)	≥0	NA	NA	NA
	GPU 显存空闲容量	ma_node_gpu_mem_free_megabytes	该指标用于统计测量空闲的显存。	兆字节 (Megabytes)	>0	NA	NA	NA
	共享 GPU 任务运行数据	node_gpu_share_job_count	针对一个 GPU 卡，当前运行的共享资源使用的任务数量。	个	≥0	NA	NA	NA
	GPU 温度	DCGM_FL_DEV_GPU_TEMP	表示 GPU 温度。	摄氏度 (°C)	自然数	NA	NA	NA
	GPU 功率	DCGM_FL_DEV_POWER_USAGE	表示 GPU 功率。	瓦特 (W)	>0	NA	NA	NA
	GPU 显存温度	DCGM_FL_DEV_MEMORY_TEMPERATURE	表示显存温度。	摄氏度 (°C)	自然数	NA	NA	NA
NPU	NPU 使用率	ma_node_npu_util	该指标用于统计测量对象的 NPU 使用率。（即将废弃，替代指标为 ma_node_npu_ai_core_util）。	百分比 (Percent)	0~100%	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
NPU 整体 利用 率	NPU 整体 利用 率	ma_node_npu_general_util	昇腾系列AI处理器NPU整体利用率，包括对AI Core和Vector Core的整体统计。（驱动版本24.1.RC2及其以后支持）	百分比(Percent)	0~100%	NA	NA	NA
	NPU 显存 使用 率	ma_node_npu_memory_util	该指标用于统计测量对象已使用的NPU显存占NPU存储容量的百分比。（即将废弃，snt3系列替代指标为ma_node_npu_ddr_memory_util，snt9系列替代指标为ma_node_npu_hbm_util）。	百分比(Percent)	0~100%	连续2个周期原始值>97%	提示	排查是否符合业务资源使用预期，如果业务无问题，无需处理。
	NPU 显存 使用 量	ma_node_npu_memory_used_megabytes	该指标用于统计测量对象已使用的NPU显存。（即将废弃，snt3系列替代指标为ma_node_npu_ddr_memory_usage_bytes，snt9系列替代指标为ma_node_npu_hbm_usage_bytes）	兆字节(Megabytes)	≥0	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
AI处理器状态	NPU显存容量	ma_node_npu_memory_total_megabytes	该指标用于统计测量对象的NPU显存容量。 (即将废弃, snt3系列替代指标为ma_node_npu_ddr_memory_bytes, snt9系列替代指标为ma_node_npu_hbm_bytes)。	兆字节(Megabytes)	>0	NA	NA	NA
	AI处理器错误码	ma_node_npu_ai_core_error_code	昇腾系列AI处理器错误码	-	-	NA	NA	NA
	AI处理器健康状态	ma_node_npu_ai_core_health_status	昇腾系列AI处理器健康状态	-	• 1 : 健康 • 0 : 不健康	连续2周期值为0	紧急	提工单咨询。
	AI处理器功耗	ma_node_npu_ai_core_power_usage_watts	昇腾系列AI处理器功耗(snt9和snt3为处理器功耗, snt3P为板卡功耗)	瓦特(W)	>0	NA	NA	NA
	AI处理器温度	ma_node_npu_ai_core_temperature_celsius	昇腾系列AI处理器温度	摄氏度(°C)	自然数	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
AI处理器 风扇转速	AI处理器风扇转速	ma_node_npu_fan_speed_rpm	昇腾系列AI处理器的风扇转速	转/每分 ( RPM )	自然数	NA	NA	NA
	AI处理器AI CORE利用率	ma_node_npu_ai_core_util	昇腾系列AI处理器AI Core利用率，是在NPU上执行cube算子的繁忙和空闲比。	百分比 ( Percent )	0~100%	NA	NA	NA
	AI处理器Vector CORE利用率	ma_node_npu_vector_core_util	昇腾系列AI处理器Vector Core利用率，是在NPU上执行vector算子的繁忙和空闲比。	百分比 ( Percent )	0~100%	NA	NA	NA
	NPU整体利用率	ma_node_npu_general_util	昇腾系列AI处理器NPU整体利用率，包括对AI Core和Vector Core的整体统计。(驱动版本24.1.RC2及其以后支持)	百分比 ( Percent )	0~100%	NA	NA	NA
	AI处理器AI CORE时钟频率	ma_node_npu_ai_core_frequency_hertz	昇腾系列AI处理器AI Core时钟频率	赫兹 ( Hz )	>0	NA	NA	NA
	AI处理器电压	ma_node_npu_ai_core_voltage_volts	昇腾系列AI处理器电压	伏特 ( V )	自然数	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
AI处理器DDR内存	AI处理器DDR内存总量	ma_node_npu_ddr_memory_bytes	昇腾系列AI处理器DDR内存总量	字节( Byte )	>0	NA	NA	NA
	AI处理器DDR内存使用量	ma_node_npu_ddr_memory_usage_bytes	昇腾系列AI处理器DDR内存使用量	字节( Byte )	>0	NA	NA	NA
	AI处理器DDR内存利用率	ma_node_npu_ddr_memory_util	昇腾系列AI处理器DDR内存利用率	百分比( Percent )	0~100%	连续2个周期原始值>90%	提示	排查是否符合业务资源使用预期，如果业务无问题，无需处理。
	AI处理器HBM内存总量	ma_node_npu_hbm_bytes	昇腾系列AI处理器HBM总内存（昇腾snt9 AI处理器专属）	字节( Byte )	>0	NA	NA	NA
	AI处理器HBM内存使用量	ma_node_npu_hbm_usage_bytes	昇腾系列AI处理器HBM内存使用量（昇腾snt9 AI处理器专属）	字节( Byte )	>0	NA	NA	NA
	AI处理器HBM内存利用率	ma_node_npu_hbm_util	昇腾系列AI处理器HBM内存利用率（昇腾snt9 AI处理器专属）	百分比( Percent )	0~100%	连续2个周期原始值>97%	提示	排查是否符合业务资源使用预期，如果业务无问题，无需处理。

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
AI处理器HBM内存带宽利用率	ma_node_npu_hbm_bandwidth_util	昇腾系列AI处理器HBM内存带宽利用率（昇腾snt9 AI处理器专属）	百分比(Percent)	0~100%	NA	NA	NA	NA
AI处理器HBM内存时钟频率	ma_node_npu_hbm_frequency_hertz	昇腾系列AI处理器HBM内存时钟频率（昇腾snt9 AI处理器专属）	赫兹(Hz)	>0	NA	NA	NA	NA
AI处理器HBM内存温度	ma_node_npu_hbm_temperature_celsius	昇腾系列AI处理器HBM内存温度（昇腾snt9 AI处理器专属）	摄氏度(°C)	自然数	NA	NA	NA	NA
AI处理器AI CPU利用率	ma_node_npu_ai_cpu_util	昇腾系列AI处理器AI CPU利用率	百分比(Percent)	0~100%	NA	NA	NA	NA
AI处理器控制CPU利用率	ma_node_npu_ctrl_cpu_util	昇腾系列AI处理器控制CPU利用率	百分比(Percent)	0~100%	NA	NA	NA	NA
AI处理器Vector CORE利用率	ma_node_npu_vector_core_util	昇腾系列AI处理器Vector Core利用率，是在NPU上执行vector算子的繁忙和空闲比。	百分比(Percent)	0~100%	NA	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
NPU Macro相关指标	NPU Macro数据包重传次数	ma_node_npu_macr_o_retry_cnt	该指标描述NPU Macro在检测周期内(10s)数据包的重传次数 ( A3 24.1.RC2版本及以上支持 )	数值	≥0	NA	NA	NA
	NPU Macro接收报文数	ma_node_npu_macr_o_rx_cnt	该指标描述NPU Macro在检测周期内(10s)接收的报文数	数值	≥0	NA	NA	NA
	NPU Macro接收错误报文数	ma_node_npu_macr_o_crc_error_cnt	该指标描述NPU Macro在检测周期内(10s)接收的CRC错误报文数	数值	≥0	NA	NA	NA
	NPU Macro接收误码率	ma_node_npu_macr_o_crc_error_rate	该指标描述NPU Macro在检测周期内接收的CRC错误报文数占接收报文数的百分比	百分比 ( Percent )	0 ~ 100%	NA	NA	NA
	NPU 算子重传成功次数	ma_node_npu运营商_retry_success_cnt	该指标描述NPU 算子重传成功次数 ( A3 Ascend HDK 24.1.RC3.3 版本及以上支持 )	数值	≥0	NA	NA	NA
	NPU 算子重传失败次数	ma_node_npu运营商_retry_fail_cnt	该指标描述NPU 算子重传失败次数 ( A3 Ascend HDK 24.1.RC3.3 版本及以上支持 )	数值	≥0	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
	NPU借轨通信次数	ma_node_npu_borrow_comms_cnt	该指标描述NPU借轨通信次数，借轨次数越多，传输效率越低（A3Ascend HDK 24.1.RC3.3版本及以上支持）	数值	≥0	NA	NA	NA
节点算力卡分配率	节点算力卡总数量	ma_node_total_card	节点算力(gpu或者npu)卡数量	数值	≥0	NA	NA	NA
	节点算力卡已分配数量	ma_node_allocate_card	节点算力(gpu或者npu)卡,已分配到容器的卡数量	数值	≥0	NA	NA	NA
	节点算力卡分配率	ma_node_allocate_card_util	已分配到容器的卡数量和计算力卡总数量的比值	数值	≥0	NA	NA	NA
NPU HC CS 链路(A3规格)	NPU可用信用证数量	ma_npu_hccs_avail_credit	可用信用证数量,度量继续接收数据的能力(驱动版本25.1.RC1以上版本支持)	数值	≥0	NA	NA	NA
	HCCS链路发送总带宽	ma_node_npu_hccs_total_txbw_per_second	NPU发送数据总带宽(驱动版本24.1.RC3.5版本以上支持)	千兆/秒(GB/Second)	≥0	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
HCC S链路接收总带宽	HCC S链路接收总带宽	ma_node_npu_hccs_total_rxbw_per_second	NPU接收数据总带宽 ( 驱动版本 24.1.RC3.5 以上版本支持 )	千兆/秒 ( GB/Second )	≥0	NA	NA	NA
	HCC S链路发送带宽明细	ma_node_npu_hccs_txbw_per_second	NPU卡与各个交换机芯片发送带宽明细 ( 驱动版本 24.1.RC3.5 以上版本支持 )	千兆/秒 ( GB/Second )	≥0	NA	NA	NA
	HCC S链路接收带宽明细	ma_node_npu_hccs_rxbw_per_second	NPU卡与各个交换机芯片接收带宽明细 ( 驱动版本 24.1.RC3.5 以上版本支持 )	千兆/秒 ( GB/Second )	≥0	NA	NA	NA
infiniband 或 RoCE 网络	网卡接收数据总量	ma_node_infiniband_port_received_data_bytes_total	The total number of data octets, divided by 4, (counting in double words, 32 bits), received on all VLs from the port.	counting in double words, 32 bits	≥0	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
	网卡发送数据总量	ma_node_infiniband_port_transmitted_data_bytes_total	The total number of data octets, divided by 4, (counting in double words, 32 bits), transmitted on all VLs from the port.	counting in double words, 32 bits	≥0	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
NFS挂载状态	NFS检索文件属性操作拥塞时间	ma_node_mountstats_getattr_backlog_wait	Getattr is an NFS operation that retrieves the attributes of a file or directory, such as size, permissions, owner, etc. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performance and slow system response times.	ms	≥0	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
	NFS 检索文件属性操作往返时间	ma_node_mountstats_getattr_rt	Getattr is an NFS operation that retrieves the attributes of a file or directory, such as size, permissions, owner, etc. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply. RTT includes network transit time and server execution time. RTT is a good measurement for NFS latency. A high RTT can indicate network or server issues.	ms	≥0	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
NFS 检查 文件 权限 操作 拥塞 时间	ma_node_mountstats_access_backlog_wait	Access is an NFS operation that checks the access permissions of a file or directory for a given user. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performance and slow system response times.	ms	≥0	NA	NA	NA	

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
NFS 检查 文件 权限 操作 往返 时间	ma_node_mountstats_access_rt t	Access is an NFS operation that checks the access permissions of a file or directory for a given user. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply. RTT includes network transit time and server execution time. RTT is a good measurement for NFS latency. A high RTT can indicate network or server issues.	ms	≥0	NA	NA	NA	

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
	NFS 解析 文件 句柄 操作 拥塞 时间	ma_node_mountstats_lookup_backlog_wait	Lookup is an NFS operation that resolves a file name in a directory to a file handle. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performance and slow system response times.	ms	≥0	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
NFS 解析 文件 句柄 操作 往返 时间	ma_node_mountstats_lookup_rt	Lookup is an NFS operation that resolves a file name in a directory to a file handle. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply. RTT includes network transit time and server execution time. RTT is a good measurement for NFS latency. A high RTT can indicate network or server issues.	ms	≥0	NA	NA	NA	

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
	NFS 读文件操作拥塞时间	ma_node_mountstats_read_backlog_wait	Read is an NFS operation that reads data from a file. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performance and slow system response times.	ms	$\geq 0$	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
	NFS 读文件操作往返时间	ma_node_mountstats_read_rtt	Read is an NFS operation that reads data from a file. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply. RTT includes network transit time and server execution time. RTT is a good measurement for NFS latency. A high RTT can indicate network or server issues.	ms	$\geq 0$	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
	NFS写文件操作拥塞时间	ma_node_mountstats_write_backlog_wait	Write is an NFS operation that writes data to a file. Backlog wait is the time that the NFS requests have to wait in the backlog queue before being sent to the NFS server. It indicates the congestion on the NFS client side. A high backlog wait can cause poor NFS performance and slow system response times.	ms	≥0	NA	NA	NA

分类	名称	指标	指标含义	单位	取值范围	告警阈值	告警级别	处理建议
	NFS写文件操作往返时间	ma_node_mountstats_write_rtt	Write is an NFS operation that writes data to a file. RTT stands for Round Trip Time and it is the time from when the kernel RPC client sends the RPC request to the time it receives the reply. RTT includes network transit time and server execution time. RTT is a good measurement for NFS latency. A high RTT can indicate network or server issues.	ms	≥0	NA	NA	NA

## Label 指标介绍

表 5-10 Label 名字栏

指标对象	Label名字	Label描述
容器级别指标	pod_name	容器所属pod的名字。
	pod_id	容器所属pod的ID。
	node_ip	容器所属的节点IP值。

指标对象	Label名字	Label描述
集群级指标	container_id	容器ID。
	cluster_id	集群ID。
	cluster_name	集群名称。
	container_name	容器名称。
	namespace	是用户创建的POD所在的命名空间。
	app_kind	取自首个ownerReferences的kind字段。
	app_id	取自首个ownerReferences的uid字段。
	app_name	取自首个ownerReferences的name字段。
	npu_id	昇腾卡的ID信息，比如davinci0（即将废弃）。
	device_id	昇腾系列AI处理器的Physical ID。
	device_type	昇腾系列AI处理器类型。
	pool_id	物理专属池对应的资源池id。
	pool_name	物理专属池对应的资源池name。
	gpu_uuid	容器使用的GPU的UUID。
node级别指标	gpu_index	容器使用的GPU的索引。
	gpu_type	容器使用的GPU的型号。
	cluster_id	该node所属CCE集群的ID。
	node_ip	节点的IP。
	host_name	节点的主机名。
	pool_id	物理专属池对应的资源池ID。
	project_id	物理专属池的用户的project id。
	npu_id	昇腾卡的ID信息，比如davinci0（即将废弃）。
	device_id	昇腾系列AI处理器的Physical ID。
	device_type	昇腾系列AI处理器类型。
	gpu_uuid	节点上GPU的UUID。
	gpu_index	节点上GPU的索引。
	gpu_type	节点上GPU的型号。
	device_name	infiniband或RoCE网络网卡的设备名称。
	port	IB网卡的端口号。

指标对象	Label名字	Label描述
	physical_state	IB网卡每个端口的状态。
	firmware_version	IB网卡的固件版本。
	filesystem	NFS挂载的文件系统。
	mount_point	NFS的挂载点。
Diagnose	cluster_id	GPU所在节点所属的CCE集群ID。
	node_ip	GPU所在节点的IP。
	pool_id	物理专属池对应的资源池ID。
	project_id	物理专属池的用户的project id。
	gpu_uuid	GPU的UUID。
	gpu_index	节点上GPU的索引。
	gpu_type	节点上GPU的型号。
	device_name	infiniband或RoCE网络网卡的设备名称。
	port	IB网卡的端口号。
	physical_state	IB网卡每个端口的状态。
	firmware_version	IB网卡的固件版本。

## 5.8.2 使用 Prometheus 查看 Lite Cluster 监控指标

Prometheus是一款开源监控工具，ModelArts支持Exporter功能，方便用户使用Prometheus等第三方监控系统获取ModelArts采集到的指标数据。

本章节主要介绍如何通过Prometheus查看Lite Cluster监控指标。

### 约束限制

- 需要在ModelArts Lite Cluster资源池详情页的配置管理页面中先打开“监控”开关。
- 开通此功能后，兼容Prometheus指标格式的第三方组件可通过API `http://<节点IP>:<端口号>/metrics`获取ModelArts采集到的指标数据。
- 开通前需要确认使用的端口号，端口号可选取10120~10139范围内的任一端口号，请确认选取的端口号在各个节点上都没有被其他应用占用。

### Kubernetes 下 Prometheus 对接 ModelArts

- 使用kubectl连接集群，详细操作请参考[通过kubectl连接集群](#)。
- 配置Kubernetes的访问授权。

使用任意文本编辑器创建prometheus-rbac-setup.yml，YAML文件内容如下：

## 说明

该YAML用于定义Prometheus要用到的角色 ( ClusterRole )，为该角色赋予相应的访问权限。同时创建Prometheus所使用的账号 ( ServiceAccount )，将账号与角色进行绑定 ( ClusterRoleBinding )。

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: prometheus
rules:
- apiGroups: [""]
  resources:
  - pods
  verbs: ["get", "list", "watch"]
- nonResourceURLs: ["/metrics"]
  verbs: ["get"]
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: prometheus
  namespace: default
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: prometheus
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: prometheus
subjects:
- kind: ServiceAccount
  name: prometheus
  namespace: default
```

- 执行如下命令创建RBAC对应的各个资源。

```
$ kubectl create -f prometheus-rbac-setup.yml
clusterrole "prometheus" created
serviceaccount "prometheus" created
clusterrolebinding "prometheus" created
```

- 使用任意文本编辑器创建prometheus-config.yml，内容如下。该YAML用于管理Prometheus的配置，部署Prometheus时通过文件系统挂载的方式，容器可以使这些配置。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: prometheus-config
data:
  prometheus.yml: |
    global:
      scrape_interval: 10s
    scrape_configs:
    - job_name: 'modelarts'
      tls_config:
        ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
        bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
      kubernetes_sd_configs:
      - role: pod
        relabel_configs:
        - source_labels: [__meta_kubernetes_pod_name] # 指定从固定字符串开头的POD收集指标数据，ModelArts Node Agent插件版本低于7.2.0, pod前缀maos-node-agent-, 否则pod前缀是modelarts-metric-collector
          action: keep
          regex: ^(maos-node-agent-|modelarts-metric-collector).+
        - source_labels: [__address__] # 指定获取指标数据的地址和端口号为__address__:9390, __address__为POD的IP地址，也是节点IP地址
```

```
action: replace
regex: '(.*)'
target_label: __address__
replacement: "${1}:10120"
```

5. 执行如下命令创建ConfigMap资源。

```
$ kubectl create -f prometheus-config.yml
configmap "prometheus-config" created
```

6. 使用任意文本编辑器创建prometheus-deployment.yml，内容如下。

## 说明

该YAML用于部署Prometheus。将上面创建的账号（ServiceAccount）权限赋予了Prometheus，同时将上面创建的ConfigMap资源以文件系统的方式挂载到了prometheus容器的“/etc/prometheus”目录，并且通过--config.file=/etc/prometheus/prometheus.yml参数指定了“/bin/prometheus”使用该配置文件。

```
apiVersion: v1
kind: "Service"
metadata:
  name: prometheus
  labels:
    name: prometheus
spec:
  ports:
    - name: prometheus
      protocol: TCP
      port: 9090
      targetPort: 9090
  selector:
    app: prometheus
    type: NodePort
---
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    name: prometheus
  name: prometheus
spec:
  replicas: 1
  selector:
    matchLabels:
      app: prometheus
  template:
    metadata:
      labels:
        app: prometheus
    spec:
      hostNetwork: true
      serviceAccountName: prometheus
      serviceAccount: prometheus
      containers:
        - name: prometheus
          image: prom/prometheus:latest
          imagePullPolicy: IfNotPresent
          command:
            - "/bin/prometheus"
          args:
            - "--config.file=/etc/prometheus/prometheus.yml"
          ports:
            - containerPort: 9090
              protocol: TCP
            volumeMounts:
              - mountPath: "/etc/prometheus"
                name: prometheus-config
            volumes:
              - name: prometheus-config
```

```
configMap:  
  name: prometheus-config
```

7. 执行如下命令创建Prometheus实例，并查看创建情况：

```
$ kubectl create -f prometheus-deployment.yml  
service "prometheus" created  
deployment "prometheus" created  
  
$ kubectl get pods  
NAME           READY   STATUS    RESTARTS   AGE  
prometheus-55f655696d-wjqcl   1/1     Running   0          5s  
  
$ kubectl get svc  
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE  
kubernetes   ClusterIP   10.96.0.1   <none>        443/TCP      131d  
prometheus   NodePort    10.101.255.236  <none>        9090:32584/TCP  42s
```

## 查看 Prometheus 采集的指标数据

1. 在CCE页面为Prometheus所在节点绑定弹性公网IP，并打开节点的安全组配置，添加入方向规则，允许外部访问9090端口。

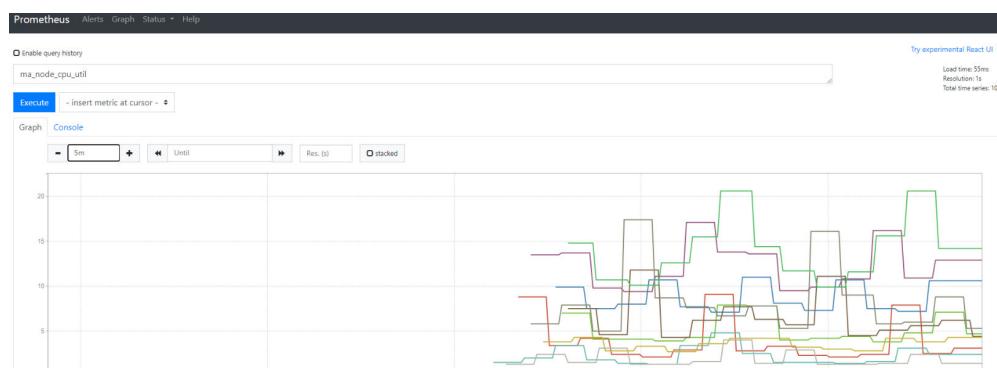
### 说明

如果使用Grafana对接Prometheus制作报表，可以将Grafana部署在集群内，这里不需要对Prometheus绑定公网IP和配置安全组，只需要对Grafana绑定公网IP和配置安全组即可。

图 5-12 添加入方向规则



2. 在浏览器地址栏输入http://<弹性公网IP>:9090，即可打开Prometheus监控浏览页面。单击Graph菜单，在输入框输入任意一个指标名称即可看到Prometheus收集到的指标数据：



## 5.9 释放 Lite Cluster 资源

针对不再使用的Lite Cluster资源，可以释放资源，停止计费相关介绍请见[停止计费](#)。

### ⚠ 警告

- 删除资源池，会同步删除资源池中按需计费的关联资源，且删除操作无法恢复，请谨慎操作。资源池中的包周期节点需要单独退订或者释放。
- 删除Lite Cluster资源池，会同步删除资源池中的磁盘，磁盘上的数据会被清除，不可恢复，请谨慎操作。
- 如果资源池中包含已开启删除锁的节点，删除资源池可能会导致开启删除锁的节点被删除，从而中断正在运行的业务且不可回退，请谨慎操作，确保没有关键业务受到影响。

### 退订包年/包月的 Lite Cluster 资源

- 登录[ModelArts管理控制台](#)，在左侧菜单栏中选择“资源管理 > 轻量算力集群（Lite Cluster）”。
- 在资源池列表中，单击操作列的“ $\cdots$  > 退订”，跳转至“退订资源”页面。
- 根据界面提示，确认需要退订的资源，并选择退订原因。
- 确认退订信息无误后，勾选“资源退订后……”提示信息。
- 单击“退订”，再次根据界面信息确认要退订的资源。
- 再次单击“退订”，完成包年/包月资源的退订操作。

# 6 Lite Cluster 插件管理

## 6.1 Lite Cluster 插件概述

ModelArts提供多种类型的插件，支持通过安装插件选择性扩展Lite Cluster资源池功能，以满足业务需求。

### 默认安装插件

在创建专属资源池时，已默认安装的插件。

 注意

资源池默认安装的插件不支持卸载。

表 6-1 默认安装插件简介

插件名称	插件简介
<a href="#">6.2 节点故障检测 (ModelArts Node Agent)</a>	ModelArts节点故障检测是一款监控集群节点异常事件的插件，以及对接第三方监控平台功能的组件。它是一个在每个节点上运行的守护程序，可从不同的守护进程中搜集节点问题。
<a href="#">6.4 AI套件 (ModelArts Device Plugin)</a>	CCE AI套件（Ascend NPU）是支持容器里使用Huawei NPU设备的管理插件。 <a href="#">开通Lite Cluster资源</a> 时，仅实例规格类型选择“Ascend”时自动安装。
<a href="#">6.5 Volcano调度器</a>	Volcano 是一个基于 Kubernetes 的批处理平台，提供了机器学习、深度学习、生物信息学、基因组学及其他大数据应用所需要而 Kubernetes 当下缺失的一系列特性。

### 手动安装插件

可根据业务需求，选择性安装插件用于扩展资源池功能。

表 6-2 默认安装插件简介

插件名称	插件简介
6.6 集群弹性引擎	集群弹性引擎是一个对集群中ModelArts资源池进行弹性伸缩的插件。集群弹性引擎可以根据用户配置的规则对各节点池进行扩容或者缩容。

## 插件生命周期

状态	状态属性	说明
安装中	中间状态	插件正处于部署状态。 如遇到插件配置错误或资源不足所有实例均无法调度等情况，系统会在10分钟后将该插件置为“不可用”状态。
运行中	稳定状态	插件正常运行状态，所有插件实例均正常部署，插件可正常使用。
升级中	中间状态	插件正处于更新状态。
不可用	稳定状态	不可用，表示插件状态异常，插件不可使用。可单击状态查看失败原因。
删除中	中间状态	插件处于正在被删除的状态。 如果长时间处于该状态，则说明出现异常。

## 在插件广场搜索查看插件

在[ModelArts管理控制台](#)插件广场页面展示了丰富的插件信息，在插件广场页面可搜索查看指定插件详情，并安装插件到指定资源池。

表 6-3 插件广场相关操作

操作	说明	操作步骤
搜索查看插件	进入插件广场搜索查看指定插件。	登录 <a href="#">ModelArts管理控制台</a> ，在控制台左侧导航栏中选择“插件广场”，进入“插件广场”页面。 在下拉框中可通过资源池类型过滤插件，也可在搜索框中输入关键词搜索相应的插件。
查看插件详情	在插件广场查看插件详情，包括插件简介、组件列表等信息。	<ol style="list-style-type: none"><li>1. 登录<a href="#">ModelArts管理控制台</a>，在控制台左侧导航栏中选择“插件广场”，进入“插件广场”页面。</li><li>2. 单击插件名称，可查看插件详情。</li></ol>
安装插件	部分插件支持手动安装指定插件。可在插件广场安装插件。	<ol style="list-style-type: none"><li>1. 登录<a href="#">ModelArts管理控制台</a>，在控制台左侧导航栏中选择“插件广场”，进入“插件广场”页面。</li><li>2. 在“安装插件”弹框中，选择待安装插件的资源类型。部分插件还需要选择插件版本。选择完成后单击“下一步”。<ul style="list-style-type: none"><li>• 专属集群：将插件安装至资源池，不同插件支持安装的资源池类型不同，请以界面为准。</li><li>• 专属节点：将插件安装至资源池中具体节点，请按照界面信息执行相关操作和命令。</li></ul></li><li>3. 配置插件相关参数。由于不同插件支持的配置参数不同，详细步骤请参见插件章节。</li></ol>

## 在资源池详情页查看 Lite Cluster 插件

在资源池详情页的“插件”页签，执行[表6-4](#)中的操作。

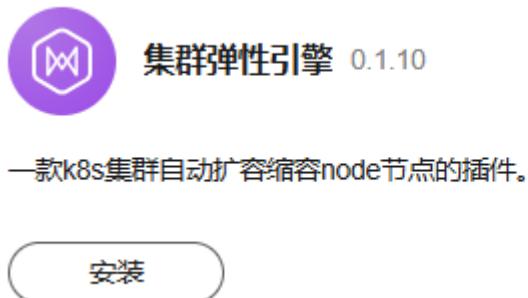
表 6-4 插件相关操作

操作	说明	操作步骤
查看插件列表	查看资源池所有插件列表。在此页面，可以查看插件详情、安装插件、升级插件、卸载插件，对插件集中管理。	<ol style="list-style-type: none"><li>登录<a href="#">ModelArts管理控制台</a>，在左侧菜单栏中选择“资源管理 &gt; 轻量算力集群（Lite Cluster）”。</li><li>单击资源池名称，进入资源池详情页。</li><li>单击“插件”，切换至“插件”页签。</li></ol>
查看插件详情	查看插件详情，包括插件简介、组件列表等信息。	<ol style="list-style-type: none"><li>登录<a href="#">ModelArts管理控制台</a>，在左侧菜单栏中选择“资源管理 &gt; 轻量算力集群（Lite Cluster）”。</li><li>单击资源池名称，进入资源池详情页。</li><li>单击“插件”，切换至“插件”页签。</li><li>单击插件名称，可查看插件详情。</li></ol>
默认安装插件	创建资源池时默认安装插件，无需手动操作。	<a href="#">2 Lite Cluster资源开通</a>
手动安装插件	在资源池中安装指定插件。	<p>方式一： <a href="#">2 Lite Cluster资源开通</a>时安装插件。</p> <p>方式二：</p> <ol style="list-style-type: none"><li>登录<a href="#">ModelArts管理控制台</a>，在左侧菜单栏中选择“资源管理 &gt; 轻量算力集群（Lite Cluster）”。</li><li>单击资源池名称，进入资源池详情页。</li><li>单击“插件”，切换至“插件”页签。</li><li>在未安装插件列表中，选择待安装的插件，单击“安装”。如<a href="#">图6-1</a>所示。</li><li>在“安装插件”弹框中，配置相关参数。 当前Lite Cluster支持手动安装集群弹性引擎插件，配置参数说明请见<a href="#">表6-11</a>。</li></ol>

操作	说明	操作步骤
编辑插件	编辑插件参数。	<ol style="list-style-type: none"><li>登录<a href="#">ModelArts管理控制台</a>，在左侧菜单栏中选择“资源管理 &gt; 轻量算力集群（Lite Cluster）”。</li><li>单击资源池名称，进入资源池详情页。</li><li>在资源池详情页，切换到“插件”页签。</li><li>在插件列表中，选择待编辑的插件，单击“编辑”。 由于不同插件支持的配置参数不同，详细步骤请参见插件章节。 当前仅如下插件版本支持编辑：<ul style="list-style-type: none"><li>节点故障检测(ModelArts Node Agent)插件7.2.0及以上版本</li><li>AI套件（Ascend NPU）2.1.53及以上版本</li><li>Volcano调度器插件1.17.11及以上版本</li><li>集群弹性引擎插件0.1.13及以上版本</li></ul></li><li>设置完插件参数后，单击“确定”。</li></ol>
升级插件	将插件升级至新版。	<ol style="list-style-type: none"><li>登录<a href="#">ModelArts管理控制台</a>，在左侧菜单栏中选择“资源管理 &gt; 轻量算力集群（Lite Cluster）”。</li><li>单击资源池名称，进入资源池详情页。</li><li>在资源池详情页，切换到“插件”页签。</li><li>在插件列表中，选择待升级的插件，单击“升级”。 当前Lite Cluster支持手动安装集群弹性引擎插件，配置参数说明请见<a href="#">表6-11</a>。 5. 设置完插件参数后，单击“确定”。</li></ol> <p><b>注意</b></p> <ul style="list-style-type: none"><li>插件基于 Helm 模板进行部署，修改或升级操作需通过ModelArts控制台插件列表执行或开放的插件管理 API 执行，切勿直接在 CCE 后台手动修改相关资源，以免引发异常或引入非预期问题，如升级后参数配置丢失或被覆盖等。</li><li>插件升级过程中可能影响资源池部分功能的使用，建议在升级前检查所有外部依赖项的状态及版本兼容性，并预留充足的时间窗口进行操作。具体影响内容可参考对应插件的说明章节。</li></ul>

操作	说明	操作步骤
卸载插件	将插件从资源池中卸载。卸载操作无法恢复，请谨慎操作。	<ol style="list-style-type: none"><li>登录<a href="#">ModelArts管理控制台</a>，在左侧菜单栏中选择“资源管理 &gt; 轻量算力集群（Lite Cluster）”。</li><li>单击资源池名称，进入资源池详情页。</li><li>在资源池详情页，切换到“插件”页签。</li><li>在插件列表中，选择待卸载的插件，单击“卸载”。</li><li>在弹出的确认窗口中一键输入“DELETE”，单击“确定”。</li></ol>

图 6-1 安装插件



## 常见问题

- 必选安装插件状态显示不可用时，或长时间处于安装中/删除中状态时，可单击资源池名称，查看基本信息，在基本信息的CCE集群里，单击进入该资源池的CCE集群。  
单击插件中心，找到对应的插件，单击插件详情，查看插件的实例列表，单击异常状态，查看具体的异常原因。
- 可选插件显示不可用时，或长时间处于安装中/删除中状态时，可先尝试卸载，重装插件。如果重装后插件状态仍显示不可用，可参考上一步骤定位插件异常详情。
- 经过以上操作，问题未得到解决，可联系MA技术人员。

## 6.2 节点故障检测(ModelArts Node Agent)

### 插件简介

ModelArts节点故障检测是一款监控集群节点异常事件的插件，以及对接第三方监控平台功能的组件。每个k8s的资源池默认都会安装，它是一个在每个节点上运行的守护程序，可从不同的守护进程中搜集节点问题。

## 安装插件

创建专属资源池时自动安装。当前不支持用户自助升级。

## 组件说明

表 6-5 节点故障检测插件组件说明

容器组件	说明	资源类型
maos-node-agent	支持监控集群节点异常，对接的第三方监控平台功能。	DeamonSet

## 版本记录

表 6-6 节点故障检测插件版本记录

插件版本	更新特性
7.3.0	支持ModelArts节点使用故障检测功能，监控集群节点异常事件。
7.2.2	支持ModelArts节点使用故障检测功能，监控集群节点异常事件。
7.2.0	支持ModelArts节点使用故障检测功能，监控集群节点异常事件。
6.8.0	支持ModelArts节点使用故障检测功能，监控集群节点异常事件。
6.7.0	支持ModelArts节点使用故障检测功能，监控集群节点异常事件。
6.6.0	支持ModelArts节点使用故障检测功能，监控集群节点异常事件。

## 6.3 指标监控插件(ModelArts Metrics Collector)

### 插件简介

指标监控插件 ( ModelArts Metrics Collector ) 是默认内置插件，以节点守护程序运行，可采集节点及各类作业监控指标，并上报到AOM。指标列表请见[使用AOM查看Lite Cluster监控指标](#)。

图 6-2 ModelArts 指标监控插件



## 约束与限制

- 创建资源池时自动安装。不支持卸载。
- 存量资源池，需要将节点故障检测（ModelArts Node Agent）插件版本升级到最新版本，自动安装该插件。
- 在插件升级期间，指标采集pod重启，可能存在短暂指标不上报，请谨慎执行升级操作。

## 组件说明

容器组件	说明	资源类型
modelarts-metric-collector	节点、容器指标采集	DaemonSet

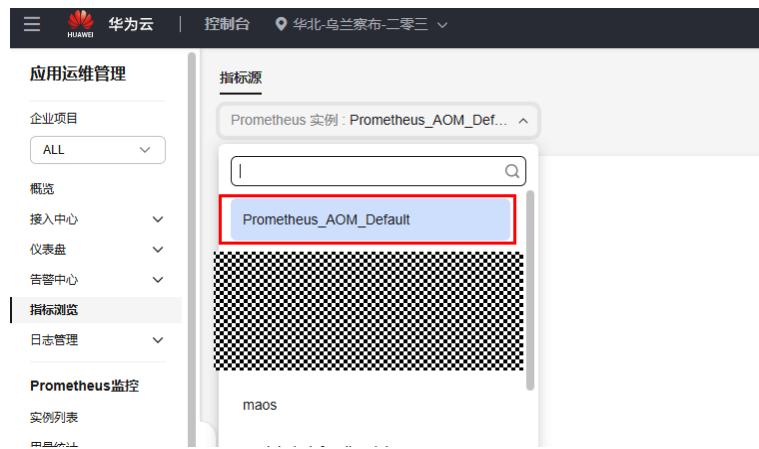
## 参数说明

参数	说明
备机上报	专属池备机是否上报指标，默认false不上报。
开启 exporter	支持使用Prometheus等第三方监控系统获取ModelArts采集到的指标数据。关闭后，将无法使用Prometheus等第三方监控系统采集指标。默认开启。 专属池：使用推理作业指标扩缩容，需开启。
上报至AOM 自定义 Prometheus 通用实例	指标默认上报到AOM平台的Prometheus_AOM_Default实例。 开启时，监控指标上报至自定义Prometheus通用实例，如图6-3所示。关闭后，则上报至Default默认Prometheus实例，即Prometheus_AOM_Default实例，如图图6-4所示。

图 6-3 自定义通用 Prometheus 实例



图 6-4 Prometheus\_AOM\_Default 实例



## 6.4 AI 套件(ModelArts Device Plugin)

### 插件简介

AI套件（ModelArts Device Plugin）是支持容器里使用huawei NPU设备的管理插件。

### 约束限制

[创建专属资源池](#)时，仅实例规格类型选择“Ascend”时自动安装。

### 组件说明

表 6-7 AI 套件 ( ModelArts Device Plugin ) 组件

容器组件	说明	资源类型
modelarts-device-plugin	支持容器里使用ModelArts Device Plugin设备的管理插件。	Daemon Set

## 版本记录

表 6-8 AI 套件 ( ModelArts Device Plugin ) 版本记录

插件版本	更新特性
7.3.0	支持Kubernetes v1.31
7.2.0	支持Kubernetes v1.31 支持A2单卡多pod
7.0.1	支持Kubernetes v1.31
7.0.0	支持Kubernetes v1.31
2.1.53	修复安全漏洞
2.1.46	支持Kubernetes v1.31
2.1.23	修复部分问题
2.1.22	<ul style="list-style-type: none"><li>修复了一些页面显示问题</li><li>支持查询超节点信息</li><li>支持上报显卡拓扑信息</li><li>修复了日志打印问题</li></ul>
2.1.5	<ul style="list-style-type: none"><li>适配CCE v1.29集群</li><li>新增静默故障码</li></ul>
1.2.14	支持NPU监控
1.2.5	支持NPU驱动自动安装

## 6.5 Volcano 调度器

### 插件简介

Volcano 是一个基于 Kubernetes 的批处理平台，提供了机器学习、深度学习、生物信息学、基因组学及其他大数据应用所需要而 Kubernetes 当下缺失的一系列特性。

Volcano提供了高性能任务调度引擎、高性能异构芯片管理、高性能任务运行管理等通用计算能力，通过接入AI、大数据、基因、渲染等诸多行业计算框架服务终端用户，最大支持1000 Pod/s的调度并发数，轻松应对各种规模的工作负载，大大提高调度效率和资源利用率。

Volcano针对计算型应用提供了作业调度、作业管理、队列管理等多项功能，主要特性包括：

- 丰富的计算框架支持：通过CRD提供了批量计算任务的通用API，通过提供丰富的插件及作业生命周期高级管理，支持TensorFlow, MPI, Spark等计算框架容器化运行在Kubernetes上。
- 高级调度：面向批量计算、高性能计算场景提供丰富的高级调度能力，包括成组调度，优先级抢占、装箱、资源预留、任务拓扑关系等。

- 队列管理：支持分队列调度，提供队列优先级、多级队列等复杂任务调度能力。

目前Volcano项目已经在Github开源，项目开源地址：<https://github.com/volcanosh/volcano>。

## 约束限制

升级插件时，谨慎将高版本升级至低版本。版本降级可能存在任务无法调度风险。

## 安装插件

2 Lite Cluster资源开通时，自动安装。

## 组件说明

表 6-9 Volcano 组件

容器组件	说明	资源类型
volcano-scheduler	负责Pod调度。	Deployment
volcano-controller	负责CRD资源的同步。	Deployment
volcano-admission	Webhook server端，负责Pod、Job等资源的校验和更改。	Deployment

## 版本记录

表 6-10 Volcano 调度器版本记录

插件版本	更新特性
1.18.3	支持NPU资源紧凑型缩容能力; 支持xGPU多卡抢占能力
1.17.11	优化机柜亲和与装箱能力; 昇腾NPU抢占能力优化; 支持Kubernetes v1.32; 支持昇腾高密机型拓扑亲和调度能力。
1.16.8	<ul style="list-style-type: none"><li>优化超节点资源调度能力</li><li>支持Kubernetes v1.31</li></ul>
1.15.8	支持昇腾NPU双DIE亲和调度能力
1.15.6	新增基于应用资源画像的超卖能力
1.13.5	<ul style="list-style-type: none"><li>支持自定义资源按照节点优先级缩容</li><li>优化抢占与节点扩容联动能力</li></ul>

插件版本	更新特性
1.12.18	<ul style="list-style-type: none"><li>适配CCE v1.29集群</li><li>默认开启抢占功能</li></ul>
1.12.1	应用弹性扩缩容性能优化
1.11.9	<ul style="list-style-type: none"><li>优化NPU芯片rank table排序能力</li><li>支持应用弹性伸缩场景下的优先级调度</li></ul>
1.10.10	修复本地持久卷插件未计算预绑定到节点的pod的问题
1.10.7	修复本地持久卷插件未计算预绑定到节点的pod的问题
1.7.1	Volcano支持v1.25集群

## 6.6 集群弹性引擎

### 插件简介

集群弹性引擎是一个对集群中ModelArts资源池进行弹性伸缩的插件。集群弹性引擎可以根据用户配置的规则对各节点池进行扩容或者缩容。

### 约束与限制

- 集群弹性引擎支持对集群中按需计费和包周期的Lite Cluster资源池节点进行扩容和缩容。
- 资源规格售罄和底层容量不足会导致扩容失败。
- 集群弹性引擎不支持对整柜购买的Lite Cluster资源池进行弹性伸缩。
- 集群弹性引擎插件使用用户全局委托的权限操作资源池，如果全局委托中涉及资源池操作相关的黑名单策略，需要先删除黑名单。
  - 在[ModelArts管理控制台](#)的“权限管理”页面获取当前“授权对象”对应的“授权内容”，即当前用户所授予的委托名称。

图 6-5 授权内容

授权对象	授权对象类型	授权类型	授权内容	创建时间	操作
所有用户	所有用户	委托	modelarts_agency	2024/07/26 15:52:46 GMT+08:00	<a href="#">查看权限</a> <a href="#">删除</a>
IAM子用户	委托	modelarts_agency		2022/03/23 14:44:51 GMT+08:00	<a href="#">查看权限</a> <a href="#">删除</a>

- 前往[IAM控制台](#)的委托页面，找到上一步骤获取的委托名称，单击操作列的“修改”。

图 6-6 IAM 委托

委托名称/ID	委托对象	委托时长	创建时间	描述	操作
...	云服务 ModelArts	永久	2023/12/28 09:31:52 GMT+...	Created by ModelArts service.	授权 修改 删除
...	云服务 ModelArts	永久	2022/03/14 19:24:51 GMT+...	Created by ModelArts service.	授权 修改 删除
modelarts_agency	云服务 ModelArts	永久	2022/03/14 19:24:33 GMT+...	Created by ModelArts service.	授权 <b>修改</b> 删除

- c. 单击“授权记录”，切换至授权记录页签。
- d. 单击ModelArts CommonOperations操作列的“删除”，在对话框中单击“确定”，删除ModelArts CommonOperations权限。

图 6-7 删除 ModelArts CommonOperations 权限

权限	权限描述	项目所属区域	授权主体	主体描述	主体类型	操作
DLI FullAccess	数据湖探索器所有权限	所有资源 [包含未来新增...]	modelarts_agency	Created by ModelArts ser...	委托	删除
VPC Administrator	虚拟私有云服务管理员	所有资源 [包含未来新增...]	modelarts_agency	Created by ModelArts ser...	委托	删除
EPS FullAccess	企业项目管理服务所有权限	所有资源 [包含未来新增...]	modelarts_agency	Created by ModelArts ser...	委托	删除
CTS Administrator	云审计服务 (CTS) 管理...	所有资源 [包含未来新增...]	modelarts_agency	Created by ModelArts ser...	委托	删除
ModelArts CommonOperations	ModelArts服务普通用户权...	所有资源 [包含未来新增...]	modelarts_agency	Created by ModelArts ser...	委托	<b>删除</b>
SFS ReadOnlyAccess	弹性文件服务只读权限	所有资源 [包含未来新增...]	modelarts_agency	Created by ModelArts ser...	委托	删除
OBS Administrator	对象存储服务管理员	所有资源 [包含未来新增...]	modelarts_agency	Created by ModelArts ser...	委托	删除
DWS Administrator	数据仓库服务 (DWS) 管...	所有资源 [包含未来新增...]	modelarts_agency	Created by ModelArts ser...	委托	删除
LTS FullAccess	云志服务所有权限	所有资源 [包含未来新增...]	modelarts_agency	Created by ModelArts ser...	委托	删除
CES ReadOnlyAccess	云监控服务只读权限	所有资源 [包含未来新增...]	modelarts_agency	Created by ModelArts ser...	委托	删除

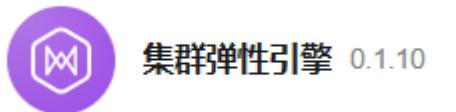
## 安装插件

1. 登录**ModelArts管理控制台**，在左侧菜单栏中选择“资源管理 > 轻量算力集群 ( Lite Cluster )”，进入“轻量算力集群 (Lite Cluster)”页面。
2. 单击资源池名称，进入资源池详情页。
3. 在资源池详情页，切换到“插件”页签。
4. 在未安装插件列表中，选择待安装的插件，单击“安装”。

### 说明

在新创建的资源池上，如果在未手动安装集群弹性引擎插件前，集群弹性引擎插件出现在已安装插件列表中，表示新创建的资源池选择的CCE集群曾经安装过集群弹性引擎插件，请先将集群弹性引擎插件卸载后重新安装。

图 6-8 安装插件



一款k8s集群自动扩容缩容node节点的插件。

安装

- 在“安装插件”弹框中，配置相关参数。

集群弹性引擎插件参数说明如下。

表 6-11 集群弹性引擎插件配置参数说明

参数	子参数	说明
规格配置	插件版本	指定部署的集群弹性引擎插件版本。
	插件规格	指定插件部署的规格。可选预置的规格或自定义规格。

- 阅读使用说明，勾选“我已阅读并知晓上述使用说明”。
- 单击“确定”。

## 配置节点池弹性伸缩策略

安装集群弹性引擎插件后，需要为节点池配置弹性伸缩策略。

弹性伸缩支持扩缩容按需计费节点。



自动缩容可能导致已配置的节点被删除且不可恢复，请谨慎操作。

- 在资源池详情页，切换到“节点池管理”页签。
- 单击操作列的“弹性伸缩配置”。
- 在弹性伸缩配置弹框中配置节点池伸缩策略。
  - 弹性扩容**  
开启后，支持节点池自动扩容。每个节点池支持最多添加6个扩容规则。

表 6-12 弹性扩容参数说明

参数	说明
自定义扩容规则	单击“添加规则”，在弹出的添加规则窗口中设置扩容规则的参数。 规则类型分为“周期触发”和“指标触发”。每个节点池支持最多添加6个扩容规则：5个周期触发类扩容规则，1个指标触发类扩容规则。不能添加相同的周期触发类扩容规则。不能添加多个指标触发规格。 扩容规则参数说明请见 <a href="#">表6-13</a> 。
节点池资源上限(个)	节点池中的总节点数达到配置的资源上限后将不再自动扩容。另外，如果当前节点数+期望扩容节点数>节点池上限，也不会触发扩容，这是为了保证扩容操作的原子性。

表 6-13 扩容规则参数说明

扩容规则类型	参数配置
周期触发	在特定时间段内自动扩容节点池下的节点数量，优化资源与需求的匹配，降低成本。 <ul style="list-style-type: none"><li>触发时间：可选择每天、每周、每月或每年的具体时间点。触发时间基于节点所在时区计算。</li><li>增加节点数(个)：弹性伸缩时节点池下的节点增加数量。</li></ul>
指标触发	基于NPU利用率动态扩容节点池下的节点数，提升任务执行效率。 <ul style="list-style-type: none"><li>触发条件：当前暂时仅支持NPU利用率触发扩容。当检测到节点的NPU使用率较低时，系统可能会将任务迁移到这些节点，或者调整节点数量以更好地匹配需求。 <math>NPU\text{利用率} = \text{节点池容器组(Pod)资源申请值} / \text{节点Pod可用资源值(Node Allocatable)}</math> 注意该百分比应大于autoscaler插件配置的缩容百分比</li><li>指定动作：<ul style="list-style-type: none"><li>自定义：自定义弹性伸缩时节点池下的节点增加数量。</li><li>自动计算：当达到触发条件时，将自动扩容节点，直至利用率恢复到触发条件以下。 <math>\text{增加节点数} = \text{节点池容器组(Pod)资源申请值} / (\text{单节点可用资源值} * \text{目标节点数}) - \text{当前节点数} + 1</math></li></ul></li></ul>

## - 弹性扩容

开启后，将综合判断整集群的资源情况，在满足负载迁移后能够正常调度运行的前提下，自动筛选节点来进行缩容。

表 6-14 弹性缩容参数说明

参数	说明
节点池资源下限（个）	节点池中的总节点数缩小至配置的资源下限后将不再自动缩容。 弹性缩容时，需要配置节点资源池下限(minCount)，否则该节点池自动缩容无法生效。
冷却时间（分钟）	触发弹性扩容后，再次启动缩容评估的冷却时间。

- 设置完成后，单击“确定”。

## 指标触发弹性伸缩配置

如果配置节点池弹性伸缩策略时，使用节点池卡分配率指标ma\_node\_pool\_allocate\_card\_util作为弹性伸缩策略，需要进行相关配置。

- 安装云原生插件，选择本地存储，并开启自定义指标采集功能，详情请见[创建使用自定义指标的HPA策略](#)。

- 创建external APIServices，并使用kubectl apply将配置应用到k8s集群。

- 登录[CCE控制台](#)，通过右上角【命令行工具】进入该集群的shell页面。
- 创建external.yaml文件，并将下面的yaml内容保存到该文件

- 执行kubectl apply -f external.yaml

```
apiVersion: apiregistration.k8s.io/v1
kind: APIService
metadata:
  labels:
    app: external-metrics-apiserver
    release: cceaddon-prometheus
    name: v1beta1.external.metrics.k8s.io
spec:
  group: external.metrics.k8s.io
  groupPriorityMinimum: 100
  insecureSkipTLSVerify: true
  service:
    name: custom-metrics-apiserver
    namespace: monitoring
    port: 443
    version: v1beta1
    versionPriority: 100
```

- 增加普罗插件自定义external指标。更多详情请见[修改配置文件](#)。

- 登录[CCE控制台](#)，单击集群名称进入集群。在集群控制台左侧导航栏中选择“配置与密钥”，切换至“monitoring”命名空间。
- 更新user-adapter-config配置项，通过修改user-adapter-config中rules字段将Prometheus暴露出的指标转换为HPA可关联的指标。

添加以下示例规则：

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-adapter-config
```

```
namespace: monitoring
data:
  config.yaml: |
    rules: []
    ...
#以下内容为新增内容
  externalRules:
    - seriesQuery: '{__name__="ma_node_allocate_card_util",pool_id!=""}'
      metricsQuery: avg(<<Series>>{<<.LabelMatchers>>}) by (pool_id,node_pool)
  resources:
    overrides:
      pool_id:
        resource: namespace
    name:
      as: ma_node_pool_allocate_card_util
```

4. 在CCE控制台，单击左侧“集群管理”，进入集群管理页面。
5. 单击集群名称进入集群，在左侧选择“工作负载”，默认进入“无状态负载”页签。切换至“monitoring”命名空间。选择custom-metrics-apiserver实例，单击工作负载后的“更多 > 重新部署”，执行重新部署操作。
6. 重部署完成后，可以通过CCE控制台的命令行工具查看当前指标的值。命令如下，其中pool\_id为资源池ID，node\_pool参数为节点池名；查询默认节点池时，该参数传空值。

```
kubectl get --raw /apis/external.metrics.k8s.io/v1beta1/namespaces/{{pool_id}}/ma_node_pool_allocate_card_util?labelSelector=node_pool={{node_pool_name}}
```

## 组件说明

表 6-15 集群弹性引擎 Nodescaler 组件

容器组件	说明	资源类型
nodescaler-controller-manager	负责资源池的弹性扩缩。	Deployment

## 相关操作

请见[在资源池详情页查看Lite Cluster插件](#)。

## 版本记录

表 6-16 集群弹性引擎插件版本记录

插件版本	更新特性
7.3.0	支持自定义部署规格。
0.1.20	支持自定义定时扩容、NPU分配率扩容和基于负载的空闲节点自动缩容。