云容器实例(CCI) 2.0

用户指南

文档版本 01

发布日期 2025-08-26

华为云计算技术有限公司

版权所有 © 华为云计算技术有限公司 2025。 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


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。

目 录

 版权所有 © 华为云计算技术有限公司

 版权所有 © 华为云计算技术有限公司

 版权所有 © 华为云计算技术有限公司

1 权限管理

1.1 CCI 权限说明

CCI当前认证鉴权是在统一身份认证服务（IAM）的能力基础上，提供的基于IAM的细粒度权限控制和IAM Token认证，同时支持命名空间级别及命名空间以下资源的权限控制，帮助用户便捷灵活地对租户下的IAM用户、用户组设定不同的操作权限。

- **CCI权限：**是基于IAM的细粒度授权。通过命名空间级别权限设置可以控制用户操作Namespace（如创建、删除Namespace等）。

1.2 创建用户并授权使用 CCI

如果您需要对您所拥有的云容器实例 CCI2.0进行精细的权限管理，您可以使用**统一身份认证服务**（Identity and Access Management，简称IAM），通过IAM，您可以：

- 根据企业的业务组织，在您的云账号中，给企业中不同职能部门的员工创建IAM用户，让员工拥有唯一安全凭证，并使用CCI资源。
- 根据企业用户的职能，设置不同的访问权限，以达到用户之间的权限隔离。
- 将CCI资源委托给更专业、高效的其他云账号或者云服务，这些账号或者云服务可以根据权限进行代运维。

如果云账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用CCI服务的其它功能。

本章节为您介绍对用户授权的方法，操作流程如**图1-1**所示。

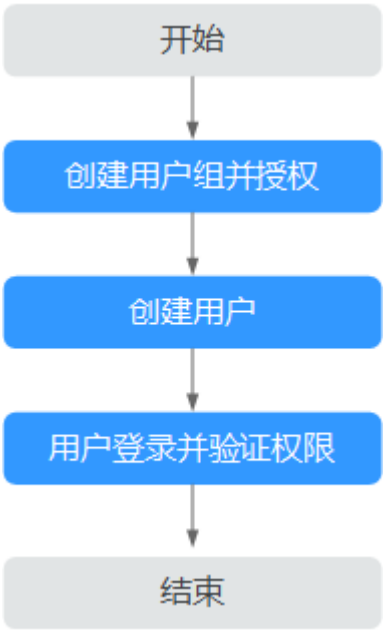
前提条件

给用户组授权之前，请您了解用户组可以添加的CCI权限，并结合实际需求进行选择，CCI支持的系统策略

版权所有 © 华为云计算技术有限公司

示例流程

图 1-1 给用户授权 CCI 权限流程



1. **创建用户组并授权**

   在IAM控制台创建用户组（例如开发人员组），并授予云容器实例 CCI2.0普通用户权限“CCI CommonOperations”。因为CCI 2.0为项目级服务，所以在给用户授予CCI 2.0相关系统策略权限时，还需要给用户授予IAM ReadOnlyAccess。

2. **创建用户并加入用户组**

   在IAM控制台创建用户（例如James），并将其加入**1**中创建的用户组。

3. **用户登录**并验证权限

   新创建的用户登录控制台，切换至授权区域，验证权限：

   - 在“服务列表”中选择云容器实例 CCI2.0，进入CCI主界面，左侧导航栏中选择“负载管理 > 无状态负载”，单击“创建工作负载”，如果可以正常创建工作负载，表示“CCI CommonOperations”已生效。
   - 在“服务列表”中选择云容器实例 CCI2.0，进入CCI主界面，左侧导航栏中选择“命名空间”，在右侧页面单击“创建命名空间”，如果无法创建命名空间，表示“CCI CommonOperations”已生效。

1.3 CCI 自定义策略

如果系统预置的CCI权限，不满足您的授权要求，可以创建自定义策略。

目前支持以下两种方式创建自定义策略：

- 可视化视图创建自定义策略：无需了解策略语法，按可视化视图导航栏选择云服务、操作、资源、条件等策略内容，可自动生成策略。
- JSON视图创建自定义策略：可以在选择策略模板后，根据具体需求编辑策略内容；也可以直接在编辑框内编写JSON格式的策略内容。

具体创建步骤请参见：**创建自定义策略**。本章为您介绍常用的CCI自定义策略样例。

版权所有 © 华为云计算技术有限公司

CCI 自定义策略样例

- 示例1：更新命名空间

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cci:namespace:update"
            ]
        }
    ]
}
```

- 示例2：拒绝用户删除命名空间

  拒绝策略需要同时配合其他策略使用，否则没有实际作用。用户被授予的策略中，一个授权项的作用如果同时存在Allow和Deny，则遵循**Deny优先原则**。

  如果您给用户授予CCIFullAccess的系统策略，但不希望用户拥有CCIFullAccess中定义的删除命名空间权限（cci:namespace:delete），您可以创建一条相同Action的自定义策略，并将自定义策略的Effect设置为Deny，然后同时将CCIFullAccess和拒绝策略授予用户，根据Deny优先原则，则用户可以对CCI执行除了删除命名空间外的所有操作。拒绝策略示例如下：

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": [
                "cci:namespace:delete"
            ],
            "Effect": "Deny"
        }
    ]
}
```

- 示例3：多个授权项策略

  一个自定义策略中可以包含多个授权项，且除了可以包含本服务的授权项外，还可以包含其他服务的授权项，可以包含的其他服务必须跟本服务同属性，即都是项目级服务或都是全局级服务。多个授权语句策略描述如下：

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": [
                "ecs:cloudServers:resize",
                "ecs:cloudServers:delete",
                "ims:images:list",
                "ims:serverImages:create"
            ],
            "Effect": "Allow"
        }
    ]
}
```

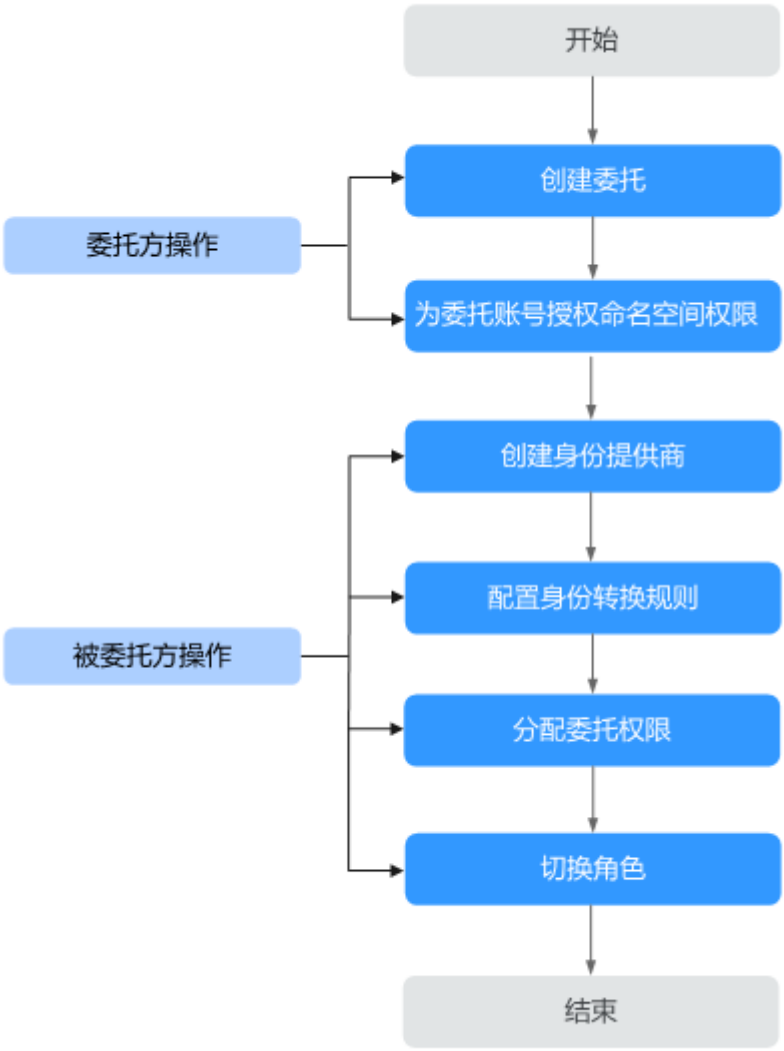
1.4 委托联邦用户管理资源

如果您需要将账号A的资源委托给账号B的联邦用户进行管理。首先您可以登录账号A，为账号B创建委托并授予命名空间权限。接下来登录账号B，与账号B做联邦身份认证，认证完成后，账号B将委托权限分配给联邦用户，使得联邦用户可以切换为账号A的委托。最后以联邦用户的身份登录到华为云，切换角色之后，就可以管理账号A中的资源。

版权所有 © 华为云计算技术有限公司

本章节为您介绍委托联邦用户管理资源的方法，操作流程如图1-2所示。

图 1-2 委托联邦用户管理资源流程



操作步骤

账号A委托账号B以联邦用户的身份管理账号A中的资源，需要完成以下步骤：

步骤1 **创建委托（委托方操作）**

登录委托方（账号A）IAM控制台创建委托，需要填写被委托方（账号B）的账号名称，并授予被委托方（账号B）云容器实例 CCI2.0所有权限“CCIFullAccess”，拥有该权限的用户可以执行云容器实例 CCI2.0所有资源的创建、删除、查询、更新操作。

步骤2 **联邦身份认证（被委托方操作）**

登录被委托方（账号B），在被委托方（账号B）中进行联邦身份认证操作。

在委托联邦用户管理资源之前，需对被委托方进行联邦身份认证，联邦身份认证过程主要分为两步：建立互信关系并创建身份提供商和在华为云配置身份转换规则。

版权所有 © 华为云计算技术有限公司

说明

创建身份提供商时，会创建出默认的身份转换规则，用户需要单击“编辑规则”将默认的身份转换规则更新掉，或者将默认的身份转换规则删除，重新创建新的规则。如果默认的身份转换规则在没有删掉的情况下直接添加新规则，可能会匹配上这条默认规则，添加的新规则就会不生效。

步骤3 分配委托权限（被委托方操作）

如果被委托方（账号B）下的子用户想要切换委托，就必须由被委托方（账号B）分配委托权限。因此，为了使得联邦用户拥有管理委托方（账号A）资源的权限，就需要被委托方（账号B）授予联邦用户所在用户组（federation_group）自定义策略“federation_agency”。“federation_group”用户组为联邦用户所在的用户组，由用户自定义，也是配置身份转换规则时，写入规则中的联邦用户组。

步骤4 切换角色（被委托方操作）

账号B以及分配了委托权限的联邦用户，可以切换角色至委托方账号A中，根据权限管理委托方的资源。

----结束

版权所有 © 华为云计算技术有限公司

2 环境设置

购买云服务 VPCEP

最终租户VPC内访问华为云100网段后端服务，需要借助云服务的VPCEP完成。

- 根据当前SWR企业仓库实现，需要购买OBS VPCEP，才能完成镜像拉取动作。
- 根据当前SWR公共镜像仓库的实现，需要在负载所关联的VPC内同时购买SWR VPCEP和OBS VPCEP，才能完成镜像拉取的动作。

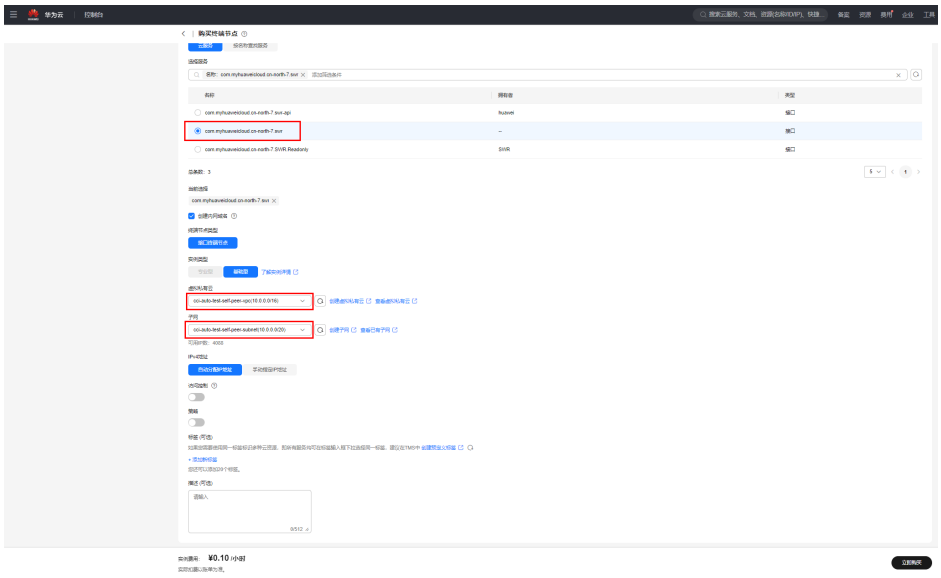
步骤1 进入**终端节点列表页**。

步骤2 在“终端节点”列表页，单击“购买终端节点”。

进入“购买终端节点”页面。

步骤3 根据界面提示配置参数。SWR VPCEP、OBS VPCEP均以VPC为单位进行购买。

图 2-1 购买 SWR VPCEP



版权所有 © 华为云计算技术有限公司

图 2-2 购买 OBS VPCEP

表 2-1 终端节点配置参数

参数	示例	说明
区域	**华北-北京四**	终端节点所在区域。 不同区域的资源之间内网不互通。请选择靠近您的区域，可以降低网络时延、提高访问速度。
计费方式	**按需计费**	按需计费是后付费模式，按终端节点的实际使用时长计费，可以随时开通/删除终端节点。 仅支持按需计费。

版权所有 © 华为云计算技术有限公司

参数	示例	说明
服务类别	**云服务**	可选择“云服务”或“按名称查找服务”。 ● 云服务：当您要连接的终端节点服务为云服务时，需要选择“云服务”。 ● 按名称查找服务：当您要连接的终端节点服务为用户私有服务时，需要选择“按名称查找服务”。 **注意** ● 购买SWR VPCEP时选择“云服务”。 ● 购买OBS VPCEP时选择“按名称查找服务”。
选择服务	-	若“服务类别”选择“云服务”，则会出现该参数。 终端节点服务实例已由运维人员预先创建完成，您可以直接使用。 **说明** OBS VPCEP选择时“按名称查找服务”，服务名称请提交工单获取。
虚拟私有云	-	选择终端节点所属的虚拟私有云。
子网	-	选择终端节点所属的子网。
路由表	-	当创建连接“网关”类型终端节点服务的终端节点时，则会出现该参数。 **说明** 该参数仅在开放区域可见。 建议选择所有路由表，否则可能导致网络无法访问。 根据实际需求选择终端节点所属的虚拟私有云的路由表。 添加路由的详细操作请参考《虚拟私有云用户指南》中的“**添加自定义路由**”。
策略	-	终端节点策略。 终端节点策略是一种基于资源的策略，您可以附加到VPC终端节点，以控制哪些主体可以使用该终端节点访问终端节点服务。
标签	**example_key1** **example_value1**	您按需为终端节点绑定标签，标签可用来分类和标识资源。 终端节点购买连接完成后支持修改该参数。
描述	-	终端节点描述内容。

版权所有 © 华为云计算技术有限公司

表 2-2 终端节点标签命名规则

参数	规则
键	• 不能为空。 • 对于同一资源键值唯一。 • 长度不超过128个字符。 • 取值为不包含“=”、“*”、“<”、“>”、“\”、“,”、“|”、“/”的所有Unicode字符，且首尾字符不能为空格。 • 取值可以包含任意语种字母、数字、空格和 _ . : = + - @，且首尾不能含有空格，不能以_sys_开头。
值	• 可以为空。 • 长度不超过255个字符。 • 取值为不包含“=”、“*”、“<”、“>”、“\”、“,”、“|”、“/”的所有Unicode字符，且首尾字符不能为空格。 • 取值可以包含任意语种字母、数字、空格和 _ . : / = + - @，且首尾不能含有空格。

步骤4 参数配置完成，单击“立即购买”，进行规格确认。

- 规格确认无误，单击“提交”，任务提交成功。
- 参数信息配置有误，需要修改，单击“上一步”，修改参数，然后单击“提交”。

步骤5 任务提交成功，返回终端节点列表。

步骤6 单击终端节点ID，即可查看终端节点的详细信息。

----结束

登录云容器实例 CCI2.0 控制台

登录云容器实例 CCI2.0控制台，并授予云容器实例访问其他云服务的权限。

步骤1 登录管理控制台。

步骤2 单击管理控制台左上角的选择区域。

云容器实例 CCI2.0当前仅支持在“华东-上海一”、“华东-上海二”、“华北-北京四”、“华南-广州”、“西南-贵阳一”和“华北-乌兰察布一”区域使用。

说明

云容器实例 CCI2.0不支持在子项目创建资源请求。

步骤3 在首页“服务列表”中，选择“容器 > 云容器实例 CCI2.0”。

进入云容器实例 CCI2.0控制台。

 版权所有 © 华为云计算技术有限公司

步骤4 首次登录云容器实例 CCI2.0控制台，需要授信云容器实例访问其他云服务资源的权限，单击“同意授权”。

授信成功后，将会创建一个委托，委托名称为“cci_admin_trust”，您可以在IAM服务控制台上查看。

----结束

（可选）上传镜像

云平台提供了容器镜像服务，您可以上传容器镜像到容器镜像仓库中，创建负载时使用该镜像，具体使用方法请参见**客户端上传镜像**。

须知

- 如果您开通了**企业管理**，使用账号登录云容器实例 CCI2.0控制台时，您需要在账号下的容器镜像服务中给IAM用户添加权限，IAM用户才能使用账号下的私有镜像。
- 给IAM用户添加权限有如下两种方法：
  - 在镜像详情中为IAM用户添加授权，授权完成后，IAM用户享有读取/编辑/管理该镜像的权限，具体请参见**在镜像详情中添加授权**。
  - 在组织中为IAM用户添加授权，使IAM用户对组织内所有镜像享有读取/编辑/管理的权限，具体请参见**在组织中添加授权**。

版权所有 © 华为云计算技术有限公司

3 命名空间

3.1 创建命名空间

命名空间（namespace）是一种在多个用户之间划分资源的方法。适用于用户中存在多个团队或项目的情况。

当前云容器实例 CCI2.0提供“通用计算型”资源，支持创建含CPU资源的容器实例，适用于通用计算场景。

说明

- 一个账号在一个区域，目前只能使用5个命名空间。
- 通用计算型支持X86镜像。

命名空间与网络的关系

从网络角度，命名空间对应一个虚拟私有云（VPC）中一个子网，如图3-1所示，在创建命名空间时会关联已有VPC或创建一个新的VPC，并在VPC下创建一个子网。后续在该命名空间下创建的容器及其他资源都会在对应的VPC及子网之内。

通常情况下，如果您在同一个VPC下还会使用其他服务的资源，您需要考虑您的网络规划，如子网网段划分、IP数量规划等，确保有可用的网络资源。

图 3-1 命名空间与 VPC 子网的关系

版权所有 © 华为云计算技术有限公司

哪些情况下适合使用多个命名空间

因为namespace可以实现部分的环境隔离。当你的项目和人员众多的时候可以考虑根据项目属性，例如生产、测试、开发划分不同的namespace。

创建命名空间

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“命名空间”。

步骤3 进入命名空间页面，单击右侧“创建命名空间”。

步骤4 填写命名空间名称。

说明

- 命名空间名称在云容器实例中需全局唯一。
- 请输入1到63个字符的字符串，可以包含小写英文字母、数字和中划线（-），并以小写英文字母或数字开头，小写英文字母或数字结尾。

步骤5 监控配置(可选)。

参数	说明
对接AOM(可选)	开启后可选择对接AOM实例。

步骤6 网络平面配置。

表 3-1 网络平面配置

参数	说明
启用IPv6	开启后支持ipv4/ipv6双栈协议网络。
虚拟私有云	选择云容器实例所在的虚拟私有云VPC，如果没有可选项可以单击右侧“新建虚拟私有云”创建。创建后不可修改。 建议使用网段：10.0.0.0/8~22，172.16.0.0/12~22，192.168.0.0/16~22。 **须知** • 此处VPC和子网的网段不能为10.247.0.0/16，10.247.0.0/16是云容器实例预留给负载访问的网段。如果您使用此网段，后续可能会造成IP冲突，导致负载无法创建或服务不可用；如果您不需要通过负载访问，而是直接访问Pod，则可以使用此网段。 • 命名空间创建完成后，在“命名空间 > 子网”中可查看到VPC和子网信息。

版权所有 © 华为云计算技术有限公司

参数	说明
子网	选择子网，如果没有可选项可以单击右侧“新建子网”创建。创建后子网不可修改。 • 创建的namespace会在设置的子网中预热部分IP，默认个数为10个。 • 在**高级设置**中可以设置预热的个数。 • 创建namespace后，由于预热了部分IP，会影响设置的subnet和VPC的删除，需要删除namespace之后才能正常删除对应的subnet和VPC。 **说明** 您需要关注子网的可用IP数，确保有足够数量的可用IP，如果没有可用IP，则会导致负载创建失败。
安全组	选择已有安全组作为该命名空间下安全组，如果没有可选项可以单击右侧“创建安全组”创建。创建后不可修改。

步骤7 高级设置（可选）。

每个命名空间下都提供了一个IP池，申请IP需要一段时间，如果需要快速创建负载，减少IP的申请时间，可通过自定义资源池大小来实现。

例如，某业务线日常的负载数为200，当达到流量高峰时，IP资源池会自动扩容，瞬间将IP资源池扩容到65535（IP资源池大小），同时会在回收间隔23h（IP资源池回收间隔）之后，进行回收超过资源池大小的部分即（65535-200=65335）个。

表 3-2 高级设置（**可选**）

参数	说明
预热 IP 资源池大小（个）	• 为每个命名空间预热一个 IP 池，用来加速容器组创建。 • 预热IP资源池的大小不能超过65535个。 • 使用通用型pod规格时，建议根据业务配置合理的预热IP资源池大小，以加快负载启动速度。 • 请合理配置预热IP数量，如配置IP数量过大，可能出现预热IP资源池大于子网可用IP数，导致子网IP被耗尽，进而影响用户使用其他服务。
预热 IP 资源池回收间隔（h）	IP 资源池弹性扩容出来的空闲 IP 资源，在一定时间内可进行回收。 **说明** 网卡回收机制： • 回收时间要求：根据network上配置的 yangtse.io/warm-pool-recycle-interval（单位小时）时间，控制该网卡是否能够完成回收。如 yangtse.io/warm-pool-recycle-interval 配置为 24，则回收的网卡必须已经申请了超过24小时后才会被回收。 • 回收速率：内部维持一个合理的速率，按照每次最多50个网卡的速率进行回收，避免回收过快或频繁回收导致网卡的重复创建和删除动作。

版权所有 © 华为云计算技术有限公司

步骤8 单击“确定”。

创建完成后，可以在命名空间详情中看到VPC、子网等信息。

----结束

删除命名空间

须知

删除命名空间将会删除该命名空间相关的所有数据资源（工作负载、服务、容器组、ConfigMap、Secret等）。

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“命名空间”，进入命名空间页面。

步骤3 选择需要删除的命名空间，在右侧操作列单击“删除”，并输入DELETE，然后单击“确认”。

说明

如需删除VPC、Subnet请前往**虚拟私有云**。

----结束

3.2 使用共享 VPC 创建 CCI 命名空间

共享VPC是通过资源访问管理服务（RAM）将本账号的VPC资源共享给其他账号使用。例如，租户A可以将自己账号下创建的VPC和子网共享给租户B。在租户B接受共享以后，租户B账号下可以查看到该共享子网及其所属的共享VPC，并可以使用该共享子网和共享VPC创建资源，如CCI 命名空间。详情请参见**共享VPC概述**。

使用场景

企业按企业的组织结构或业务形态，将账号有序组织集中管理。统一资源管理并与其他成员共享，节省资源重复配置。统一安全运维管理，便于企业集中配置安全策略，利于审计跟踪。

例如，资源所有者为企业IT账号，创建VPC及子网，并将多个子网分别共享给其他账号：

- 账号A为企业业务账号，使用子网1创建资源。
- 账号B为企业业务账号，使用子网2创建资源。

约束与限制

- 使用共享VPC创建的集群不支持使用共享ELB功能。
- 如果当前共享VPC下已创建CCI 命名空间，则共享VPC的所有者不应删除该共享，否则将会导致CCI 命名空间功能异常。

版权所有 © 华为云计算技术有限公司

操作步骤

账号A将VPC共享给账号B后，账号B即可在创建CCI 命名空间时选择共享VPC及其共享子网。

1. （账号A操作）使用资源访问管理服务（RAM）创建共享VPC，并指定资源使用者为账号B，详情请参见**创建共享**。

   共享创建完成后，RAM会向指定的使用者发送共享邀请，账号B需接受共享邀请后，才可以访问和使用被共享的资源。

2. （账号B操作）登录**云容器实例 CCI2.0**控制台，创建一个CCI 命名空间。

   在网络平面配置中，请选择由账号A共享的VPC和子网。其余配置可参考**创建命名空间**。

图 3-2 选择共享 VPC

版权所有 © 华为云计算技术有限公司

4 通过 CCI 控制台使用 CCI

4.1 负载管理

4.1.1 无状态负载

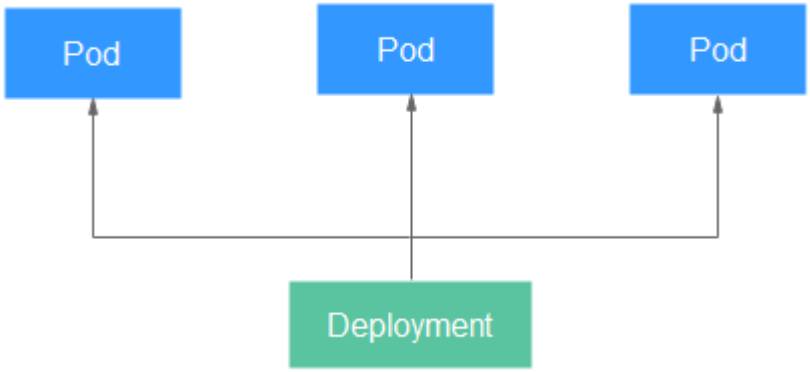
无状态负载与Kubernetes中Deployment Workloads的定义方式相同，是对Pod的服务化封装。一个无状态负载可以包含一个或多个Pod，每个Pod的角色相同，所以系统会自动为无状态负载的多个Pod分发请求。同一无状态负载的所有Pod共享存储卷。

在**容器组**章节介绍了Pod，Pod是创建或部署的最小单位。云容器实例提供了Controller（控制器）来管理Pod，Controller可以创建和管理多个Pod，提供副本管理、滚动升级和自愈能力，其中最为常用的就是Deployment。

一个Deployment可以包含一个或多个Pod副本，每个Pod副本的角色相同，所以系统会自动为Deployment的多个Pod副本分发请求。

Deployment集成了上线部署、滚动升级、创建副本，恢复上线任务，在某种程度上，Deployment可以帮用户实现无人值守的上线，大大降低了上线过程的复杂沟通、操作风险。

图 4-1 无状态负载



版权所有 © 华为云计算技术有限公司

通过控制台界面创建无状态负载

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“负载管理 > 无状态负载”，在页面单击“创建工作负载”。

步骤3 添加基础配置。

参数	说明
负载名称	请输入以小写字母或数字开头，小写字母、数字、中划线（-）、点（.）组成（其中两点不能相连，点不能与中划线相连），小写字母或数字结尾的1到63字符的字符串。负载名称不支持修改，如需修改名称，需要重新创建。
命名空间	选择命名空间，如果还未创建命名空间，请参考**命名空间**创建。
CPU架构	X86计算或鲲鹏。
实例类型	通用型或通用型（轻享）。 **说明** 仅X86计算CPU架构支持通用型（轻享）实例类型。
vCPUs	选择vCPU大小，0.25核-64核之间。
内存	根据选择的vCPU大小，选择内存大小。
数据存储（可选）	为容器提供存储，目前支持临时目录、配置项、密钥，请先为容器组添加数据存储后再挂载到指定容器中。 单击“添加数据存储”，选择存储卷类型，并输入存储卷名称。 • 临时目录：输入CCI容器组默认提供30GiB的免费存储空间，临时目录与系统盘共享。 • 配置项：选择**已创建的配置项**。 • 密钥：选择**已创建的密钥**。
实例数量	负载可以有一个或多个Pod，您可以设置具体Pod个数。每个负载Pod都由相同的容器部署而成。设置多个Pod主要用于实现高可靠性，当某个Pod故障时，负载还能正常运行。

步骤4 添加容器配置。一个Pod可以包含一个或多个运行不同镜像的容器，通常情况下一个Pod中只有一个容器，如果您的应用程序需要多个容器，请添加容器。

1. 请单击“添加容器”。
2. 填写基本信息。

表 4-1 基本信息

参数	说明
容器名称	请输入1到63个字符的字符串，可以包含小写英文字母、数字和中划线（-），并以小写英文字母或数字开头，小写英文字母或数字结尾。

版权所有 © 华为云计算技术有限公司

参数	说明
镜像	选择镜像。 **须知** 同一个Pod实例中的不同容器如果监听了相同的端口，则会导致端口冲突，Pod可能会启动失败。例如在Pod中添加了一个nginx镜像容器，启动了80端口，如果该Pod中另一个http服务的镜像也启动80端口，那么这个Pod就会出现端口冲突。 – 我的镜像：展示了您上传到容器镜像服务的镜像。 **说明** ■ 如您是IAM用户，您需要参考**（可选）上传镜像**进行权限设置后才可使用账号的私有镜像。 ■ CCI当前暂不支持对接第三方镜像仓库。 ■ 镜像单层解压后的实际大小不能超过20G。 – 共享镜像：展示了容器镜像服务中他人共享的镜像。 – 镜像中心：展示了镜像中心的公共镜像。
镜像版本	选择镜像的版本。
vCPUs	输入值不能小于0或大于剩余规格额度。
内存	输入值不能小于0或大于剩余规格额度。

3. 高级设置（可选）：

- 生命周期：生命周期脚本定义，在容器的生命周期的特定阶段执行调用。详细步骤请参见**4.1.4**。
- 健康检查：健康检查是指容器运行过程中，根据您需要，定时检查容器健康状况。详细步骤请参见**4.1.5**。
- 环境变量：在容器中设置环境变量，支持手动输入和引用变量。环境变量为应用提供极大的灵活性，您可以在应用程序中使用环境变量，在创建容器时为环境变量赋值，容器运行时读取环境变量的值，从而做到灵活地配置，而不是每次都重新编写应用程序制作镜像。

  手动输入只需要直接输入变量名称和变量值。

  变量引用支持引用PodIP（Pod的IP地址）、PodName（Pod的名称）以及Secret，输入变量名称，选择引用类型、引用值。其中Secret引用的创建请参见**4.5.2**。
- 数据存储：支持挂载数据存储到容器中，以实现容器内的数据访问。
- 安全设置：支持配置容器的运行用户。

步骤5 选择升级策略。

- 升级策略：升级方式支持“滚动升级”和“替换升级”。
  – 滚动升级：滚动升级将逐步用新版本的实例替换旧版本的实例，升级的过程中，业务流量会同时负载均衡分布到新老的实例上，因此业务不会中断。

    最大无效实例数：每次滚动升级允许的最大无效实例数，如果等于实例数有断服风险（最小存活实例数 = 实例数 - 最大无效实例数）。
  – 替换升级：先删除旧实例，再创建新实例。升级过程中业务会中断。

版权所有 © 华为云计算技术有限公司

步骤6 配置完成后，单击“创建工作负载”。

在负载列表中，待负载状态为“运行中”，负载创建成功。您可以单击负载名进入负载详情界面，刷新负载状态。

----结束

通过 YAML 创建无状态工作负载

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“负载管理 > 无状态负载”，在页面单击“YAML创建”。

步骤3 导入或者添无状态负载YAML，单击“确定”，创建无状态负载。

Deployment文件格式说明

- deployment.yaml资源描述

```
apiVersion: cci/v2
kind: Deployment
metadata:
  annotations:
    description: ''
  labels: {}
  name: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      annotations:
        vm.cci.io/pod-size-specs: 2.00_4.0
        resource.cci.io/pod-size-specs: 2.00_4.0
        metrics.alpha.kubernetes.io/custom-endpoints: '[{api:"",path:"",port:"",names:""}]'
        log.stdoutcollection.kubernetes.io: '{"collectionContainers": ["container-0"]}'
      labels:
        app: nginx
    spec:
      containers:
        - image: library/nginx:stable-alpine-perl
          name: container-0
          resources:
            limits:
              cpu: 2000m
              memory: 4096Mi
            requests:
              cpu: 2000m
              memory: 4096Mi
          command: []
          lifecycle: {}
      dnsPolicy: ''
      imagePullSecrets:
        - name: imagepull-secret
      dnsConfig: {}
  minReadySeconds: 0
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 0
      maxUnavailable: 1
```

- deployment.json资源描述

```
{
    "apiVersion": "cci/v2",
    "kind": "Deployment",
```

 版权所有 © 华为云计算技术有限公司

```
  "metadata": {
    "annotations": {
      "description": ""
    },
    "labels": {},
    "name": "nginx"
  },
  "spec": {
    "replicas": 2,
    "selector": {
      "matchLabels": {
        "app": "nginx"
      }
    },
    "template": {
      "metadata": {
        "annotations": {
          "vm.cci.io/pod-size-specs": "2.00_4.0",
          "resource.cci.io/pod-size-specs": "2.00_4.0",
          "metrics.alpha.kubernetes.io/custom-endpoints": "[{api:'',path:'',port:'',names:''}]",
          "log.stdoutcollection.kubernetes.io": "{\"collectionContainers\": [\"container-0\"]}"
        },
        "labels": {
          "app": "nginx"
        }
      },
      "spec": {
        "containers": [
          {
            "image": "library/nginx:stable-alpine-perl",
            "name": "container-0",
            "resources": {
              "limits": {
                "cpu": "2000m",
                "memory": "4096Mi"
              },
              "requests": {
                "cpu": "2000m",
                "memory": "4096Mi"
              }
            },
            "command": [],
            "lifecycle": {}
          }
        ],
        "dnsPolicy": "",
        "imagePullSecrets": [
          {
            "name": "imagepull-secret"
          }
        ],
        "dnsConfig": {}
      }
    },
    "minReadySeconds": 0,
    "strategy": {
      "type": "RollingUpdate",
      "rollingUpdate": {
        "maxSurge": 0,
        "maxUnavailable": 1
      }
    }
  }
}
```

----结束

版权所有 © 华为云计算技术有限公司

使用 ccictl 创建负载

云容器实例 CCI2.0支持使用ccictl工具，相比从控制台创建负载，使用ccictl的体验更接近Kubernetes。

约束与限制

已配置ccictl连接到CCI2.0。

操作步骤

步骤1 使用命令“ccictl create namespace cci-test”创建命名空间。

步骤2 使用命令“ccictl create -f network.yaml”创建网络，yaml示例如下：

```
apiVersion: yangtse/v2
kind: Network
metadata:
  annotations:
    yangtse.io/domain-id: ${domain-id}
    yangtse.io/project-id: ${project-id}
  name: cci-network
  namespace: cci-test
spec:
  networkType: underlay_neutron
  securityGroups:
    - ${security-group-id}
  subnets:
    - subnetID: ${subnet-id}
```

步骤3 使用命令“ccictl apply -f deployment.yaml”创建负载，yaml示例如下：

```
kind: Deployment
apiVersion: cci/v2
metadata:
  name: nginx
  namespace: cci-test
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
          - containerPort: 80
          resources:
            limits:
              cpu: 500m
              memory: 1Gi
            requests:
              cpu: 500m
              memory: 1Gi
      dnsPolicy: Default
```

步骤4 使用命令“ccictl apply -f service.yaml”创建Service，yaml示例如下：

```
kind: Service
apiVersion: cci/v2
metadata:
  name: service-nginx
  namespace: cci-test
  annotations:
    kubernetes.io/elb.class: elb
```

 版权所有 © 华为云计算技术有限公司

```
    kubernetes.io/elb.id: '${elb_id}'
spec:
  ports:
    - name: service-nginx-port
      protocol: TCP
      port: 80
      targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
```

步骤5 使用elb的公网ip和端口，访问nginx服务。

----结束

更新无状态负载

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“负载管理 > 无状态负载”，选择需要更新的deployment右侧操作中“YAML编辑”。

步骤3 编辑修改无状态负载yaml，单击确定，更新无状态负载。

----结束

删除 Pod

负载创建完后，可以对Pod进行手动删除操作，由于Pod是有控制器在控制，单击删除按钮后会立即重新创建一个新的Pod。手动删除Pod在某些场景下非常有用，比如升级到一半出现失败时、想重启业务进程时。

删除无状态负载

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“负载管理 -> 无状态负载”，选择需要删除的无状态负载，单击操作栏中“删除”，删除负载。

----结束

4.1.2 容器组

4.1.2.1 容器组概述

什么是 Pod 容器组

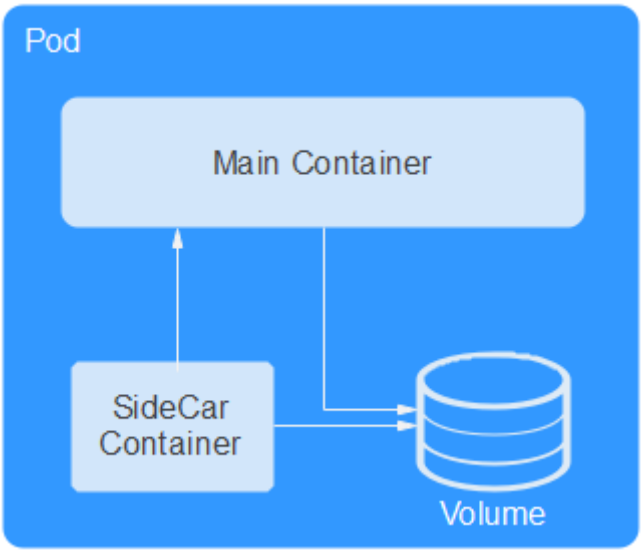
Pod（容器组）是Kubernetes创建或部署的最小单位。一个Pod（容器组）封装一个或多个容器（container）、存储资源（volume）、一个独立的网络IP以及管理控制容器运行方式的策略选项。

Pod使用主要分为两种方式：

版权所有 © 华为云计算技术有限公司

- Pod中运行一个容器。这是Kubernetes最常见的用法，您可以将Pod视为单个封装的容器，但是Kubernetes是直接管理Pod而不是容器。
- Pod中运行多个需要耦合在一起工作、需要共享资源的容器。通常这种场景下是应用包含一个主容器和几个辅助容器（SideCar Container），如图4-2所示，例如主容器为一个web服务器，从一个固定目录下对外提供文件服务，而辅助的容器周期性的从外部下载文件存到这个固定目录下。

图 4-2 Pod

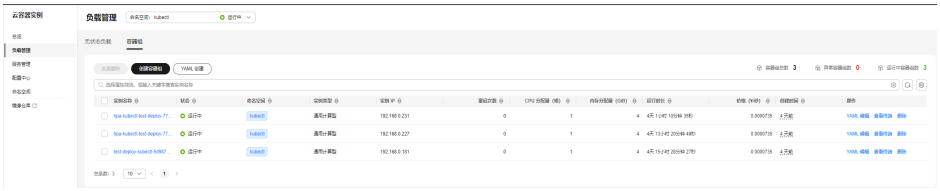


4.1.2.2 创建容器组

操作背景

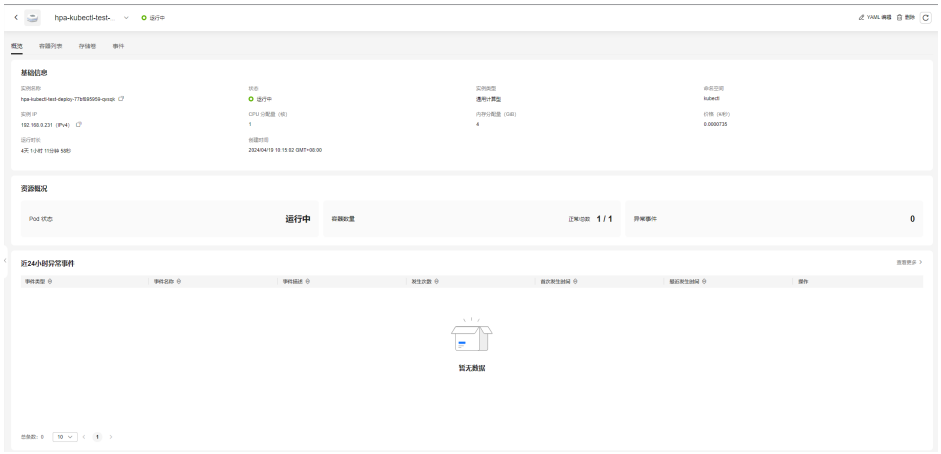
您可以通过云容器实例非常方便快速地创建容器，实现业务部署。在云容器实例CCI2.0控制台可以查看到所有Pod详情，包括基础信息、容器列表、存储卷、事件，以及在容器列表中使用远程终端访问Pod。您还可以对Pod进行删除操作。

图 4-3 选择 Pod 来源



版权所有 © 华为云计算技术有限公司

图 4-4 Pod 详情



前提条件

- 已创建命名空间，详情请参见**创建命名空间**。
- 已上传镜像，详情请参见**客户端上传镜像**。

约束与限制

在CCI 2.0，每个容器组所在环境都存在辅助运行的系统组件，它们将会占用底层部分CPU或内存等资源，可能会导致容器组的资源使用达不到预期上限。您可参考预留系统开销说明避免此类情形。

操作步骤

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 在左侧导航栏中选择“负载管理>容器组”，单击“创建容器组”。

步骤3 在“创建容器组”页面，填写容器组基本信息。

参数	说明
容器组名称	• 请输入容器组名称，同一命名空间下容器组不可重名。 • 请输入1到204个字符的字符串，容器组名称和命名空间名称字符之和不超过217，可以包含小写英文字母、数字、中划线（-）和点（.），并以小写英文字母或数字开头，小写英文字母或数字结尾。
命名空间	选择容器组所在的命名空间。
描述（可选）	描述容器组信息，少于等于250个字符。
CPU架构	X86计算或鲲鹏。
实例类型	通用型或通用型（轻享）。 **说明** 仅X86计算CPU架构支持通用型（轻享）实例类型。
vCPUs	选择vCPU大小，0.25核-64核之间。

版权所有 © 华为云计算技术有限公司

参数	说明
内存	根据选择的vCPU大小，选择内存大小。
数据存储（可选）	为容器提供存储，目前支持临时目录、配置项、密钥，请先为容器组添加数据存储后再挂载到指定容器中。 单击“添加数据存储”，选择存储卷类型，并输入存储卷名称。 • 临时目录：输入CCI容器组默认提供30GiB的免费存储空间，临时目录与系统盘共享。 • 配置项：选择**已创建的配置项**。 • 密钥：选择**已创建的密钥**。

步骤4 容器配置。

1. 添加容器基本信息。容器配置中各项资源总和不可超过容器组规格。

 版权所有 © 华为云计算技术有限公司

表 4-2 容器基本信息

参数	说明
容器名称	– 容器名称需全局唯一。 – 请输入1到63个字符的字符串，可以包含小写英文字母、数字和中划线（ - ），并以小写英文字母或数字开头，小写英文字母或数字结尾。
镜像	选择容器镜像。 **注意** 负载创建镜像暂不支持使用SWR企业版自定义域名镜像。
镜像版本	选择容器镜像版本。
vCPUs	选择vCPU大小，不可超出规格额度。
内存	选择内存大小，不可超出规格额度。

2. 添加容器高级设置（可选）。

版权所有 © 华为云计算技术有限公司

表 4-3 容器高级配置

参数	说明
生命周期	云容器实例提供了**容器生命周期钩子**，在容器的生命周期的特定阶段执行调用，比如容器在停止前希望执行某项操作，就可以注册相应的钩子函数，详情请参见**4.1.4**章节。目前提供的生命周期钩子函数如下所示： – **启动命令：**启动命令对于docker的ENTRYPOINT启动命令。 – **启动后处理：**容器启动后处理将在应用启动后触发。 – **停止前处理：**容器启动后处理将在应用启动后触发。
健康检查	健康检查是指容器运行过程中，根据需要，定时检查容器中应用健康状况。详情请参见**4.1.5**。 云容器实例基于Kubernetes，提供了三种健康检查的方式： – **存活探针：**通过检测容器是否正常来决定是否重启。 – **就绪探针：**用来确定容器是否已经就绪可以接受请求。 – **启动探针：**检测容器内应用是否已经启动。
环境变量	容器运行环境中设定的一个变量。可以在应用部署后修改，为应用提供极大的灵活性。
数据存储	支持挂载存储卷到容器中，以实现数据文件的读取或者持久化存储。若容器组内未添加您所需存储卷，请先添加存储卷。
安全设置	容器以当前用户权限运行，如 以root权限运行则填写root用户ID。

步骤5 镜像仓库访问凭证(可选)。选择镜像仓库访问凭证。

版权所有 © 华为云计算技术有限公司

步骤6 单击“立即创建”即可。

----结束

4.1.2.3 容器组规格

4.1.2.3.1 规格规整计算规则说明

对于通过Console或者API在CCI 2.0创建的pod，服务端会根据pod的cpu和内存规格，对pod的资源规格做出规整，以达到合理利用资源并满足CCI资源规格的目的。本章节将详细介绍如何进行资源规格的规整。

表 4-4 CCI 支持 Pod 规格表

容器实例 核/vCPU	容器实例 内存/GiB
0.25	0.5，1，2
0.5	0.5，1，2，3，4
1	1~8，1GiB为步长
2	2~16，1GiB为步长
4	4~32，1GiB为步长
8	8~64，4GiB为步长
16	16~128，8GiB为步长
32	32，64，128，256

版权所有 © 华为云计算技术有限公司

容器实例 核/vCPU	容器实例 内存/GiB
48	96，192，384
64	128，256，512

设置Pod规格有两种方式分别为**指定pod的内存和CPU规格**和**不指定pod的规格**。

方式一：指定 pod 的内存和 CPU 规格

- 通过pod annotation "resource.cci.io/pod-size-specs" 设置pod的资源规格。指定规格必须等于**表4-4**中支持的pod规格。否则创建pod接口会返回错误信息并拒绝创建pod。"resource.cci.io/pod-size-specs"值可设置格式为"${CPU}_${MEM}"，示例："resource.cci.io/pod-size-specs": "2.00_4.0"。
- 汇总pod内container的资源规格不能超过通过"resource.cci.io/pod-size-specs"设置的pod规格。否则创建pod接口会返回错误信息并拒绝创建pod。

方式二：不指定 pod 的规格

- 不指定pod规格时，则会通过汇总pod内container的资源规格来计算整个pod需要的内存和CPU规格。若计算所得规格匹配上**表4-4**中规格，则设置pod规格为匹配上规格值；若未匹配上，则基于**表1**自动向上规整，设置pod规格为规整后规格值。假设某一pod计算后规格为6core15Gi，自动规整后规格值为8core16Gi。
- 汇总pod内所有container的CPU和内存资源的计算方式如下：

  pod的规格=max(max(sum(spec.Containers.Request), container.Limit)，max(spec.initContainer的资源值))

  sum(spec.Containers.Request) ： 所有container.Reqeust的CPU和MEM的总和

  max(sum(spec.Containers.Request), container.Limit)：上一步计算的request总和与每个container的limit对比，取最大值

  max(spec.initContainer)：对所有initContainer的request和limit，CPU资源计算取最大值，内存资源计算取最大值，合并CPU和内存作为最大值

  max(max(sum(spec.Containers.Request), container.Limit)，max(spec.initContainer的资源值))取containers和initContainers指定资源量的最大值
- 向上规整规则如下：

  规整时将向最接近的Pod规格进行调整，必须满足设置的cpu和内存规格**小于等于****表4-4**支持的cpu和内存规格。若无法向上规整，则pod创建失败

  规整后的规格显示在pod annotation中，格式为resource.cci.io/size=${cpuCeil}_${memoryCeil}

  大于32core CPU的情况，不再进行自动向上规整。若您配置的pod内的container的CPU资源计算之后未匹配上**表4-4**中规格，则pod创建失败；若需要对32vCPU以上pod执行自动规整，需在pod annotation中声明"resource.cci.io/resizing-large-size-instance-greater-than-32-cores": "true"，允许大规格实例自动规整

版权所有 © 华为云计算技术有限公司

⚠ 注意

- 若pod未设置"resource.cci.io/pod-size-specs"，且pod内所有的container均未设置request和limit时，创建pod会返回错误信息并创建失败。
- 每个container的CPU和内存需要满足k8s原生的validate要求：当两者存在值时，0<=request<=limit。
- 每个pod最多创建10个容器。若您存在特殊需求，可以向CCI服务申请白名单方式放开此限制。

自动规整举例说明

表 4-5 自动规整举例

场景描述	Container配置	实际申请资源量	规整后规格	说明
单容器，仅配置request	containers: - resources: requests: cpu: '1.5' memory: 2Gi	1.5c2Gi	2c4Gi	只有request，看request值计算资源申请量
单容器，配置request和limit	containers: - resources: limits: cpu: '3.5' requests: cpu: '1.5' memory: 4.5Gi	3.5c4.5Gi	4c5Gi	配置了request和limit，取request和与limit的最大值
多容器，配置request	containers: - resources: requests: cpu: '1.5' memory: 2Gi - resources: requests: cpu: '1.5' memory: 2Gi	3c4Gi	4c4Gi	多容器，取多容器request和
多容器，配置limit	containers: - resources: limits: cpu: '1.5' memory: 2Gi - resources: limits: cpu: '1.5' memory: 2Gi	1.5c2Gi	2c4Gi	多容器，取多容器中最大limit

版权所有 © 华为云计算技术有限公司

场景描述	Container配置	实际申请资源量	规整后规格	说明
多容器，配置request和limit	containers: - resources: limits: cpu: '5' memory: 8Gi requests: cpu: '1.5' memory: 2Gi - resources: requests: cpu: '1.5' memory: 2Gi	5c8Gi	8c8Gi	多容器，取多容器Request和与limit的最大值
容器+init容器，容器资源量大	initContainers: - resources: requests: cpu: '1.5' memory: 2Gi containers: - resources: requests: cpu: '3' memory: 4Gi	3c4Gi	4c4Gi	取init容器和容器中资源最大值
容器+init容器，init容器资源量大	initContainers: - resources: requests: cpu: '5' memory: 8Gi - resources: requests: cpu: '6' memory: 9Gi containers: - resources: requests: cpu: '3' memory: 4Gi	6c9Gi	8c12Gi	取init容器中和容器中资源最大值
资源规整后规格与系统支持规格一致	containers: - resources: requests: cpu: '48' memory: 96Gi	48c96Gi	48c96Gi	规格匹配成功，无需规整
资源规整后到32c	containers: - resources: requests: cpu: '31' memory: 32Gi	31c32Gi	32c32Gi	32c以下规格默认规整
规整后资源大于32c	containers: - resources: requests: cpu: '33' memory: 96Gi	33c96Gi	失败	默认不对32c以上实例规整

版权所有 © 华为云计算技术有限公司

场景描述	Container配置	实际申请资源量	规整后规格	说明
规整后资源大于32c，并指定了允许对大规格实例进行规整	annotations: "resource.cci.io/resizing-large-size-instance-greater-than-32-cores": "true" containers: - resources: requests: cpu: '33' memory: 96Gi	33c96Gi	48c96Gi	用户指定允许对大规格实例规整后，对大规格实例进行规整

⚠ 注意

从CCE弹性到CCI的Pod，在Pod有多容器，且存在容器仅设置limit的场景下，CCE的pod创建时会遵循原生k8s default_pod规则，规则如下：

在检测到Pod容器未设置request资源但设置了limit资源时，会自动补充request资源且等于limit资源值。

示例：在CCE创建1个无状态工作负载deployment，deployment的spec.template里存在2个containers，且每个container仅设置limit资源。pod创建后，每个容器的request资源会被设置，且等于limit资源。

此种场景下，可能出现pod的规整后规格大于预期的情况，您可以通过手动指定容器的request资源进行避免。

4.1.2.3.2 预留系统开销说明

Pod需要运行一些必要的系统组件，会占用一些系统资源， 因此，您的Pod内存规格与Pod资源可分配量之间会存在差异。CCI对用户Pod可分配的资源计算法则如下：

- Pod内存规格 <= 2Gi：

  Pod资源可分配量 = **Pod内存规格**
- pod内存规格 > 2Gi：

  Pod资源可分配量 = **Pod内存规格** - **CCI组件资源预留** - **OS侧资源预留**

  ⚠ 注意

  Pod内存规格是指Pod的付费内存规格，即显示在pod annotation中，格式为resource.cci.io/size=${cpuCeil}_${memoryCeil}的参数中的${memoryCeil}值。

CCI 对 Pod 内存的预留规则

CCI对Pod内存的预留模型的总预留值等于**OS侧预留值**与**CCI管理Pod所需预留值**之和。

其中OS侧预留包括基础预留和随Pod内存规格变动的浮动预留；CCI侧预留固定预留250Mi内存。

版权所有 © 华为云计算技术有限公司

表 4-6 Pod 内存预留规则

场景	预留类型	基础/浮动	预留公式	预留对象
Pod内存规格 <= 2Gi	无	\	\	\
Pod内存规格 > 2Gi	OS侧预留	基础预留	固定150Mi	操作系统服务组件占用
		浮动预留（随Pod内存规格变化）	20Mi/Gi	操作系统内核占用
	CCI侧预留	基础预留	固定250Mi	云容器实例组件占用

预留系统开销举例说明

表 4-7 资源预留举例

Pod内存规格	Pod内存可用资源	计算说明	备注
0.5Gi	0.5Gi	/	Pod内存规格 <=2Gi
1Gi	1Gi	/	
2Gi	2Gi	/	
3Gi	2.55Gi	3Gi-250Mi-(150Mi +3*20Mi)= 2.55Gi	Pod内存规格 >2Gi
4Gi	3.53Gi	4Gi-250Mi-(150Mi +4*20Mi)= 3.53Gi	
8Gi	7.45Gi	8Gi-250Mi-(150Mi +8*20Mi)= 7.45Gi	
16Gi	15.29Gi	16Gi-250Mi-(150Mi +16*20Mi)= 15.29Gi	
32Gi	30.98Gi	32Gi-250Mi-(150Mi +32*20Mi)= 30.98Gi	
64Gi	62.35Gi	64Gi-250Mi-(150Mi +64*20Mi)= 62.35Gi	
128Gi	125.1Gi	128Gi-250Mi-(150Mi +128*20Mi)= 125.1Gi	
256Gi	250.6Gi	256Gi-250Mi-(150Mi +256*20Mi)= 250.6Gi	

版权所有 © 华为云计算技术有限公司

4.1.2.3.3 增加预留系统开销

在CCI 2.0运行的pod，辅助pod容器运行的系统组件会占用少量底层资源，因此在某些场景下，pod会出现实际内存使用无法达到pod规格内存的情况。因此针对此类场景，提供2个开关用以预留系统组件开销。

pod的annotation "resource.cci.io/memory-reservation" ： 是否开启预留系统开销。默认为"false"，开启为"true"。开启后，CCI 2.0会为pod实例的系统组件预留1Gi的内存，且如果超出当前pod规格，会自动向上规整到最近的规格。以规整后的规格进行计费。

pod的annotation "resource.cci.io/memory-burst-size" ： 是否扩大容器的内存资源隔离上限。默认为"false"，开启为"true"。开启后，会调整pod的所有容器的limit.memory资源到最大(即pod规格的内存大小)，以此扩大宿主环境的cgroup内存上限；可以让开启预留系统开销开关后，且因自动规格后升阶的pod，得以将被cgroup限制的内存释放出来，供给应用容器使用。

表 4-8 开启预留系统开销举例

场景描述	Pod配置	实际申请资源量	规整后规格	在CCI运行Pod的最终规格	说明
单容器，仅配置request，开启预留系统开销，开启扩大容器的内存资源隔离上限	annotations: "resource.cci.io/memory-reservation": "true" "resource.cci.io/memory-burst-size": "true" ... containers: - resources: requests: cpu: '1.5' memory: 2Gi	1.5c3Gi	2c4Gi	containers: - resources: limits: memory: 4Gi requests: cpu: '1.5' memory: 2Gi	计算流程如下： 1. 只有request，看request值计算资源申请量（1.5c2Gi） 2. 开启了预留系统开销，追加1Gi内存（1.5c3Gi） 3. 按照**4.3.1节规整规则**，自动向上规整后（2c4Gi） 4. 开启扩大容器的内存资源隔离上限，设置所有容器的limit资源到4Gi

版权所有 © 华为云计算技术有限公司

4.1.3 查看资源使用率

云容器实例提供了查看CPU和内存的界面，若您在创建命名空间时，开启了对接AOM选项，创建完工作负载后，您可以登录云容器实例控制台查看每个Pod的资源利用率。

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 单击无状态负载或者容器组的“监控”，即可查看资源使用率，如**图4-5**所示。

图 4-5 查看监控信息

----结束

4.1.4 生命周期

4.1.4.1 启动命令

启动容器就是启动主进程，但有些时候，启动主进程前，需要一些准备工作。比如MySQL类的数据库，可能需要一些数据库配置、初始化的工作，这些工作要在最终的MySQL服务器运行之前解决。这些操作，可以在制作镜像时通过在Dockerfile文件中设置ENTRYPOINT或CMD来完成，如下所示的Dockerfile中设置了**ENTRYPOINT ["top", "-b"]**命令，其将会在容器启动时执行。

版权所有 © 华为云计算技术有限公司

```
FROM ubuntu
ENTRYPOINT ["top", "-b"]
```

须知

启动命令必须为容器镜像支持的命令，否则会导致容器启动失败。

在云容器实例中同样可以设置容器的启动命令，例如上面Dockerfile中的命令，只要在创建容器组时配置容器的高级设置，选择生命周期启动命令，先单击“添加运行命令”，输入“top”命令，再单击“添加运行参数”，输入参数“-b”，如下图所示。

图 4-6 启动命令

由于容器引擎运行时只支持一条ENTRYPOINT命令，云容器实例中设置的启动命令会覆盖掉制作镜像时Dockerfile中设置的ENTRYPOINT和CMD命令，其规则如下表所示。

镜像Entrypoint	镜像CMD	容器运行命令	容器运行参数	最终执行
[touch]	[/root/test]	未设置	未设置	[touch /root/test]
[touch]	[/root/test]	[mkdir]	未设置	[mkdir]

版权所有 © 华为云计算技术有限公司

镜像Entrypoint	镜像CMD	容器运行命令	容器运行参数	最终执行
[touch]	[/root/test]	未设置	[/opt/test]	[touch /opt/test]
[touch]	[/root/test]	[mkdir]	[/opt/test]	[mkdir /opt/test]

4.1.4.2 启动后/停止前处理

设置容器生命周期

云容器实例提供了**容器生命周期钩子**，在容器的生命周期的特定阶段执行调用，比如容器在停止前希望执行某项操作，就可以注册相应的钩子函数。目前提供的生命周期钩子函数如下所示。

- 启动后处理（PostStart）：容器启动后触发。
- 停止前处理（PreStop）：容器停止前触发。

说明

当前云容器实例仅支持命令行类型（Exec）钩子函数。

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 在创建容器组配置生命周期过程中，选择“启动后处理”或者“停止前处理”页签。

例如需要在容器中执行“/postStart.sh all”命令，则在界面上做如下配置即可，第一行是命令行脚本名称，第二行是参数。

版权所有 © 华为云计算技术有限公司

图 4-7 命令行脚本

----结束

4.1.5 健康检查

健康检查是指容器运行过程中，根据需要，定时检查容器中应用健康状况。

云容器实例提供了以下几种健康检查的方式：

- **存活探针（liveness probe），探测应用是否已经启动：**该检查方式用于检测容器是否存活，类似于执行ps命令检查进程是否存在。如果容器的存活检查的结果为失败，云容器实例会对该容器执行重启操作；若容器的存活检查成功则不执行任何操作。
- **就绪探针（readiness probe），探测应用业务是否已经就绪：**该检查方式用于检测容器是否准备好开始处理用户请求。一些程序的启动时间可能很长，比如要加载磁盘数据或者要依赖外部的某个模块启动完成才能提供服务。这时候程序进程在，但是并不能对外提供服务。这种场景下该检查方式就非常有用。
- **启动探针（startup probe），探测应用是否已经启动**：该检查方式用于检测容器是否已经启动成功。它会在容器启动时进行检查，以确保应用程序能够正常启动并运行。

健康检查方式

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 在创建容器组配置健康检查过程，选择开启存活探针、就绪探针或启动探针。

版权所有 © 华为云计算技术有限公司

- **HTTP请求检查**

  探针往容器发送HTTP请求，如果探针收到2xx或3xx的返回状态码，说明容器是健康。
- **TCP端口检查**

  探针往容器发送TCP端口检查请求，如果探针收到2xx或3xx的返回状态码，说明容器是健康。
- **执行命令检查**

  探针执行容器中的命令并检查命令退出的状态码，如果状态码为0则说明健康。

  例如，您若希望使用“cat /tmp/healthy”命令检查/tmp/healthy目录是否存在，则可以如下图配置。

版权所有 © 华为云计算技术有限公司

图 4-8 检查

----结束

公共参数说明

表 4-9 健康检查参数说明

参数	参数说明
检测周期	执行探测的时间间隔，单位为秒。例如，设置为10，表示健康检查为每10秒探测一次。
延迟时间	容器启动后启动探针的延迟时间，单位为秒。例如，设置为10，表示从容器启动后10秒开始探测。

版权所有 © 华为云计算技术有限公司

参数	参数说明
超时时间	探针超时后等待的超时时间，单位为秒。例如，设置为10，表明执行健康检查的超时等待时间为10秒，如果超过这个时间，本次健康检查就被视为失败。若设置为0或不设置，默认超时等待时间为1秒。 **说明** 执行命令检查时，如果执行的命令在镜像中会生成子进程，可能会导致timeoutSeconds无法生效。
成功阈值	探针在失败后，被视为成功的最小连接次数。例如，设置为1，表明探针失败后连续成功探测1次，认为健康检查成功。
最大失败次数	被任务健康检查失败的探测连续失败次数。例如，设置为3，表明探针连续3次探测失败后，认为健康检查失败。此时，容器被视为不健康，触发重启操作。

4.1.6 远程终端

远程终端（web-terminal）提供连接容器功能，帮助您快速调试容器。

约束与限制

- web-terminal 默认是以sh shell登录容器，要求容器支持sh shell。
- 只有状态为“运行中”的容器可终端登录。
- 退出时，请在web-terminal中输入“exit”，否则会导致sh进程残留。

通过远程终端连接容器

步骤1 登录**云容器实例 CCI2.0**控制台。

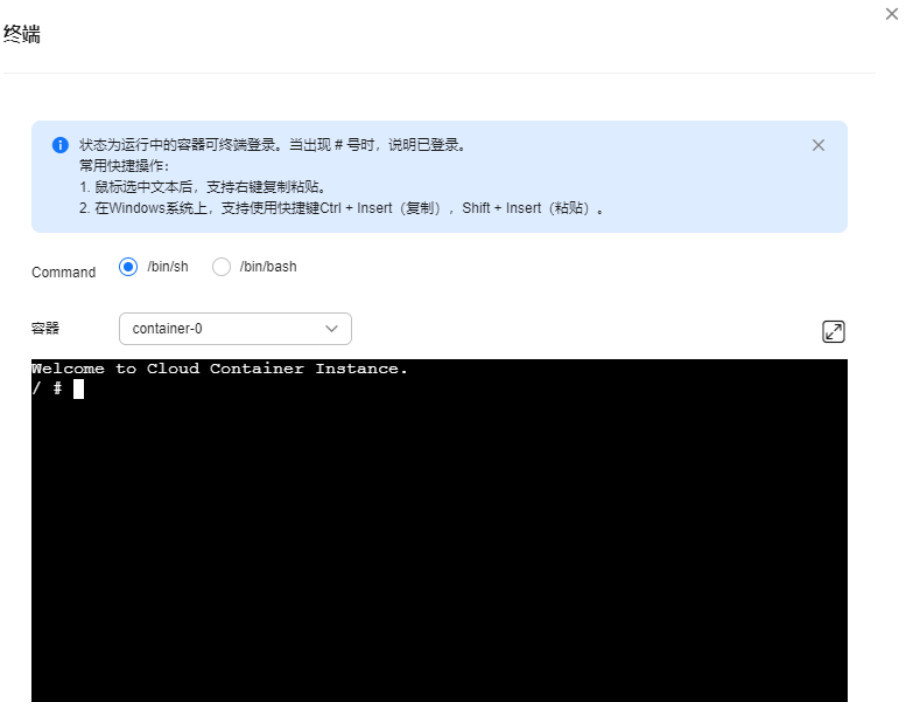
步骤2 在左侧导航栏中选择“负载管理>容器组”。

步骤3 在Pod实例右侧操作列选择“查看终端”页签。

当出现“#”号时，说明已登录。

版权所有 © 华为云计算技术有限公司

图 4-9 容器终端



----结束

4.1.7 升级负载

负载创建成功后，可以对负载更新和升级。当前支持“滚动升级”和“替换升级”两种方式。

- 滚动升级：将逐步用新版本的实例替换旧版本的实例，升级的过程中，业务流量会同时负载均衡分布到新老的实例上，因此业务不会中断。
- 替换升级：将先把您工作负载的老版本实例删除，再安装指定的新版本，升级过程中业务会中断。

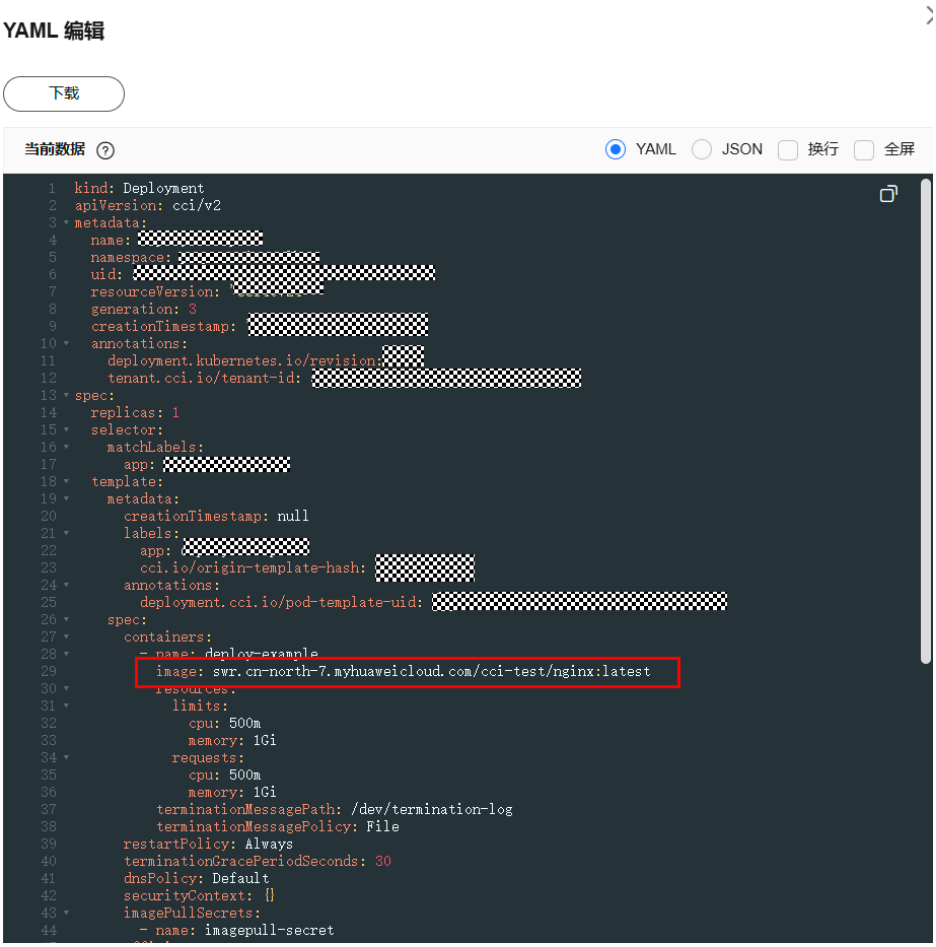
升级负载

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“负载管理 > 无状态负载”，单击YAML编辑。

步骤3 修改YAML中的字段以升级负载。以更新镜像为例，修改如下字段：

版权所有 © 华为云计算技术有限公司



步骤4 单击“确定”，升级负载。

----结束

4.1.8 伸缩负载

本节主要讲解工作负载弹性伸缩和手动伸缩的配置方式。请根据实际业务选择。

- 弹性伸缩：支持告警策略。配置完成后可基于资源变化自动触发实例的增减。
- 手动伸缩：配置完成后立即触发实例的增减。

约束与限制

- 当前仅支持无状态负载弹性伸缩。
- CCI 2.0当前仅支持在华南-广州和华东-上海一局点使用弹性伸缩。

版权所有 © 华为云计算技术有限公司

弹性伸缩

说明

- 当配置spec.metrics.resource.target.type为Utilization时，您需要在创建工作负载时指定资源的requests值。
- 当配置spec.metrics.resource.target.type为Utilization时，其数值的计算规则为：资源使用率 = 资源使用量 / 用户实际可使用的容器组规格。您可参考**容器组规格的计算方法**来确认您的容器组实际规格。

您可以根据业务需求自行定义伸缩策略，包括弹性指标、阈值以及步长等，以降低人为反复调整资源以应对业务变化和高峰压力的工作量，帮助您节约资源和人力成本。当前支持一种弹性伸缩策略：

告警策略：支持根据CPU/内存的使用率，进行工作负载的自动伸缩。工作负载创建完成后即可设置，在CPU/内存超过或少于一定值时，自动增减实例。

- **告警策略**：支持根据CPU/内存的使用率，进行工作负载的自动伸缩。
  a. 登录**云容器实例 CCI2.0**控制台。
  b. 在左侧导航栏中选择“负载管理 > 无状态负载”，单击工作负载名称进入负载详情。
  c. 在“负载详情”中，单击“弹性伸缩”，单击“YAML创建”配置弹性伸缩策略。

  弹性伸缩策略文件格式说明

  - hpa.yaml资源描述

```
kind: HorizontalPodAutoscaler
apiVersion: cci/v2
metadata:
  name: nginx              # HPA的名称
  namespace: test          # HPA所在的命名空间
spec:
  scaleTargetRef:         # 引用进行自动伸缩的目标资源
    kind: Deployment      # 目标资源类型，例Deployment
    name: nginx           # 目标资源名称
    apiVersion: cci/v2    # 目标资源版本
  minReplicas: 1          # HPA伸缩的最小副本数
  maxReplicas: 5          # HPA伸缩的最大副本数
  metrics:
    - type: Resource                # 使用资源指标
      resource:
        name: memory                # 资源名称，例如cpu或memory
        target:
          type: Utilization         # 指标类型，可以是Utilization（百分比）或AverageValue（绝对值）
          averageUtilization: 50    # 资源使用率，例当CPU使用率达到50%时触发扩容
  behavior:
    scaleUp:
      stabilizationWindowSeconds: 30  # 扩容后稳定时间，单位为秒
      policies:
      - type: Pods           #指定扩缩容的Pods数量
        value: 1
        periodSeconds: 30  # 每 30 秒检查一次
    scaleDown:
      stabilizationWindowSeconds: 30  # 缩容后稳定时间，单位为秒
      policies:
      - type: Percent      #按照现有Pods的百分比来扩缩容
        value: 50
        periodSeconds: 30  # 每 30 秒检查一次
```

  - hpa.json资源描述

```
{
  "kind": "HorizontalPodAutoscaler",
```


版权所有 © 华为云计算技术有限公司

```
{
  "apiVersion": "cci/v2",
  "metadata": {
    "name": "nginx",              # HPA的名称
    "namespace": "test"           # HPA所在的命名空间
  },
  "spec": {
    "scaleTargetRef": {         # 引用进行自动伸缩的目标资源
      "kind": "Deployment",       # 目标资源类型，例Deployment
      "name": "nginx",            # 目标资源名称
      "apiVersion": "cci/v2"      # 目标资源版本
    },
    "minReplicas": 1,             # HPA伸缩的最小副本数
    "maxReplicas": 5,             # HPA伸缩的最大副本数
    "metrics": [
      {
        "type": "Resource",           # 使用资源指标
        "resource": {
          "name": "memory",       # 资源名称，例如cpu或memory
          "target": {
            "type": "Utilization",          # 指标类型，可以是Utilization（百分比）或AverageValue（绝对值）
            "averageUtilization": 50        # 资源使用率，例当CPU使用率达到50%时触发扩容
          }
        }
      }
    ],
    "behavior": {
      "scaleUp": {
        "stabilizationWindowSeconds": 30,
        "policies": [
          {
            "type": "Pods",
            "value": 1,
            "periodSeconds": 30
          }
        ]
      },
      "scaleDown": {
        "stabilizationWindowSeconds": 30,
        "policies": [
          {
            "type": "Percent",
            "value": 50,
            "periodSeconds": 30
          }
        ]
      }
    }
  }
}
```

d. 单击“确定”。

可看到弹性伸缩策略。

图 4-10 弹性伸缩策略

待到触发条件发生时，弹性伸缩策略会自动启动。

版权所有 © 华为云计算技术有限公司

手动伸缩

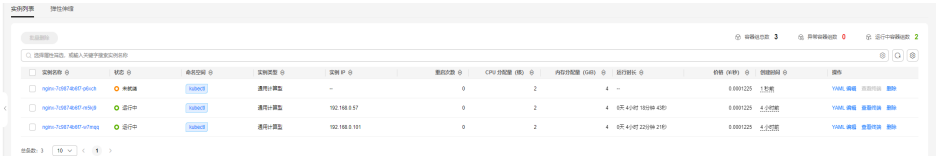
步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 在左侧导航栏中选择“负载管理 > 无状态负载”，单击工作负载名称右侧“YAML编辑”。

步骤3 修改YAML文档中spec.replicas数量，例如修改为“3”，单击“确定”后实例伸缩操作即可生效。

步骤4 在“Pod列表”处，可查看到新的实例在创建中，待状态为运行中时，表示已成功完成实例伸缩操作。

图 4-11 手动伸缩



----结束

4.2 网络管理

4.2.1 服务（Service）

4.2.1.1 服务概述

服务（Service）是用于定义Pod网络访问接口的资源，它允许以一种稳定的方式访问一组Pod，而无需关心这些Pod的具体位置和数量的变化。云容器实例支持LoadBalancer类型的服务。

负载访问可以分为如下几种场景：

- **内网访问**：创建Loadbalancer类型的Service，配置ELB（私网），通过ELB私网IP访问负载，并且可以配置IP地址黑白名单访问。该方式适合CCI中同一个命名空间中的负载相互访问，云服务内部资源（云容器实例以外的资源，如ECS等）且与负载在同一个VPC内互相访问，以及在同一个VPC不同命名空间的负载也可以选择此种方式。通过ELB的“私网IP:Port”为内网提供服务。
- **公网访问**：创建Loadbalancer类型的Service，配置ELB（公网）。通过ELB的[公网IP:port]从公网访问负载，并且可以配置黑白名单限制IP地址访问。

约束与限制

- CCI 2.0服务当前仅支持创建LoadBalancer类型的Service，支持配置独享型ELB，不支持配置共享型ELB。使用LoadBalancer类型的Service，CCI2.0当前仅支持配置后端IP地址为IPV4类型，不支持IPV6类型。
- 如果使用CCE集群中的Bursting插件对接CCI 2.0服务，支持配置独享型ELB的Ingress和Service。CCE突发弹性引擎（对接 CCI）插件1.5.5以下版本不支持配置ELB类型的Service。

版权所有 © 华为云计算技术有限公司

4.2.1.2 内网访问

概述

使用Service访问创建LoadBalancer类型的Service，配置ELB（私网），通过ELB私网IP访问负载。该方式适合CCI中同一个命名空间中的负载相互访问，云服务内部资源（云容器实例以外的资源，如ECS等）且与负载在同一个VPC内互相访问，以及在同一个VPC不同命名空间的负载也可以选择此种方式。通过ELB的“IP:Port”为内网提供服务。

负载中最小的资源单位就是Pod，访问负载就是访问负载中的Pod。

约束与限制

- 此处创建的ELB需要与负载所在命名空间在同一个VPC内。
- CCI暂时不支持共享型负载均衡，建议您创建独享型ELB实例。

配置内网访问-工作负载创建完成后设置

在工作负载创建完成后对Service进行配置，此配置对工作负载状态无影响，且实时生效。具体操作如下：

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“服务管理 ”，在右侧页面单击“YAML创建”。

步骤3 导入或者添加Service yaml。

Service文件格式说明

- Service.yaml资源描述

```
apiVersion: cci/v2
kind: Service
metadata:
  name: kubectl-test
  namespace: kubectl
  annotations:
    kubernetes.io/elb.class: elb
    kubernetes.io/elb.id: 1234567890 #关联的elb id，只支持独享型elb，不支持共享型elb。
spec:
  selector:
    app: kubectl-test #关联的负载label。
  ports:
    - name: service-0
      targetPort: 80   #容器开放的端口。
      port: 12222      #访问端口。
      protocol: TCP    #访问负载的协议。
  type: LoadBalancer
```

- Service.json资源描述

```
{
    "apiVersion": "cci/v2",
    "kind": "Service",
    "metadata": {
        "name": "kubectl-test",
        "namespace": "kubectl",
        "annotations": {
                    "kubernetes.io/elb.class": "elb",
            "kubernetes.io/elb.id": "1234567890"  #关联的elb id，只支持独享型elb，不支持共享型elb。
        }
    },
    "spec": {
        "selector": {
```

 版权所有 © 华为云计算技术有限公司

```
            "app": "kubectl-test" #关联的负载label。
        },
        "ports": [
            {
                "name": "service-0",
                "targetPort": 80,     #容器开放的端口。
                "port": 12222,        #访问端口。
                "protocol": "TCP",    #访问负载的协议。
                "type": "LoadBalancer"
            }
        ]
    }
}
```

步骤4 单击“确定”，创建Service。查看服务管理页面，根据访问地址私网IP和端口映射中的访问端口[私网IP:访问端口]，即可通过内网访问负载。

----结束

更新 Service

您可以在添加完Service后，更新此Service的访问端口，操作步骤如下：

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“服务管理 ”，在服务管理页面中，选择对应的命名空间，单击需要更新端口配置的Service后的“YAML编辑”。

步骤3 仅支持更新Service内访问端口参数修改。

端口配置：spec.ports[i].port：访问端口，端口范围为1-65535，可任意指定。

步骤4 单击“确定”，工作负载已更新Service。

----结束

4.2.1.3 公网访问

概述

公网访问是指使用外部网络访问负载，您可以给负载添加LoadBalancer类型的Service，配置ELB包含公网IP（ELB必须与负载在同一个VPC内），通过ELB实例访问负载。

约束与限制

- 创建的ELB需要与负载所在命名空间在同一个VPC内。
- 在使用弹性公网IP（Elastic IP Address，简称EIP）前，请先了解EIP的使用限制。详细信息，请参见**弹性公网IP**。
- CCI 2.0暂时不支持共享型负载均衡，建议您创建独享型ELB实例，且ELB实例已绑定公网IP。

配置公网访问-工作负载创建完成后设置

在工作负载创建完成后对Service进行配置，此配置对工作负载状态无影响，且实时生效。具体操作如下：

 版权所有 © 华为云计算技术有限公司

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“服务管理”，在右侧页面单击“YAML创建”。

步骤3 导入或者添加Service yaml。

Service文件格式说明

- Service.yaml资源描述

```
apiVersion: cci/v2
kind: Service
metadata:
  name: kubectl-test
  namespace: kubectl
  annotations:
    kubernetes.io/elb.class: elb
    kubernetes.io/elb.id: 1234567890 #关联的elb id，只支持独享型elb，不支持共享型elb。
spec:
  selector:
    app: kubectl-test #关联的负载label。
  ports:
    - name: service-0
      targetPort: 80   #容器开放的端口。
      port: 12222      #访问端口。
      protocol: TCP    #访问负载的协议。
  type: LoadBalancer
```

- Service.json资源描述

```
{
    "apiVersion": "cci/v2",
    "kind": "Service",
    "metadata": {
        "name": "kubectl-test",
        "namespace": "kubectl",
        "annotations": {
                    "kubernetes.io/elb.class": "elb",
            "kubernetes.io/elb.id": "1234567890"  #关联的elb id，只支持独享型elb，不支持共享型elb。
        }
    },
    "spec": {
        "selector": {
            "app": "kubectl-test" #关联的负载label。
        },
        "ports": [
            {
                "name": "service-0",
                "targetPort": 80,     #容器开放的端口。
                "port": 12222,        #访问端口。
                "protocol": "TCP",    #访问负载的协议。
                "type": "LoadBalancer"
            }
        ]
    }
}
```

步骤4 单击“确定”，创建Service。查看服务管理页面，根据访问地址公网IP和端口映射中的访问端口[公网IP:访问端口]，即可通过公网访问负载。

----结束

公网无法访问如何处理

- 公网能正常访问的前提是负载已处于运行中状态，如果您的负载处于未就绪或异常状态，公网访问将无法正常使用。
- 从负载开始创建到公网可以正常访问可需要1分钟到3分钟的时间，在此时间内网络路由尚未完成配置，请稍作等待。

 版权所有 © 华为云计算技术有限公司

- 负载创建3分钟以后仍然无法访问。可能原因是用户配置的容器端口实际上没有相应进程在监听，目前云容器实例服务无法检测出该类使用异常，需要您排查镜像是否有监听该容器端口。如果容器端口监听正确，此时无法访问的原因可能为ELB实例本身有问题，请排查ELB实例状态。

更新 Service

您可以在添加完Service后，更新此Service的访问端口。操作如下：

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“服务管理 ”，在服务管理页面中，选择对应的命名空间，单击需要更新端口配置的Service后的“YAML编辑 ”。

步骤3 仅支持更新Service内访问端口参数修改：

端口配置：spec.ports[i].port：访问端口，端口范围为1-65535，可任意指定。

步骤4 单击“确定 ”。工作负载已更新Service。

----结束

4.2.1.4 为负载均衡类型的 Service 配置 HTTP/HTTPS 协议

4.2.1.4.1 为负载均衡类型的 Service 配置 HTTP 协议

配置 HTTP 协议 Service

在工作负载创建完成后对Service进行配置，此配置对工作负载状态无影响，且实时生效。具体操作如下：

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“服务管理”，在右侧页面单击“YAML创建”。

步骤3 导入或者添加Service yaml。

Service文件格式说明

- Service.yaml资源描述

```
apiVersion: cci/v2
kind: Service
metadata:
  name: kubectl-test
  namespace: kubectl
  annotations:
    kubernetes.io/elb.class: elb
    kubernetes.io/elb.id: 1234567890 #关联的elb id，只支持独享型elb，不支持共享型elb。
    kubernetes.io/elb.protocol-port: "http:80"  # HTTP协议及端口号，需要与spec.ports中的端口号对应
spec:
  selector:
    app: kubectl-test #关联的负载label。
  ports:
    - name: service-0
      targetPort: 80   #容器开放的端口。
      port: 80         #访问端口。
      protocol: TCP    #协议需指定为TCP。
  type: LoadBalancer
```

- Service.json资源描述

```
{
  "apiVersion": "cci/v2",
```


版权所有 © 华为云计算技术有限公司

```
  "kind": "Service",
  "metadata": {
     "name": "kubectl-test",
     "namespace": "kubectl",
     "annotations": {
                 "kubernetes.io/elb.class": "elb",
        "kubernetes.io/elb.id": "1234567890"  #关联的elb id，只支持独享型elb，不支持共享型elb。
                 "kubernetes.io/elb.protocol-port": "http:80"  #HTTP协议及端口号，需要与spec.ports中
的端口号对应
     }
  },
  "spec": {
     "selector": {
        "app": "kubectl-test" #关联的负载label。
     },
     "ports": [
        {
           "name": "service-0",
           "targetPort": 80,     #容器开放的端口。
           "port": 80,           #访问端口。
           "protocol": "TCP",    #协议需指定为TCP。
           "type": "LoadBalancer"
        }
     ]
  }
}
```

步骤4 单击“确定”，创建Service。查看服务管理页面，根据访问地址IP和端口映射中的访问端口[IP:访问端口]，即可访问负载。

----结束

更新 Service

您可以在添加完Service后，更新此Service的访问端口。操作如下：

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“服务管理 ”，在服务管理页面中，选择对应的命名空间，单击需要更新端口配置的Service后的“YAML编辑”。

步骤3 仅支持更新Service内访问端口参数修改：

端口配置：spec.ports[i].port：访问端口，端口范围为1-65535，可任意指定。需同步修改 "kubernetes.io/elb.protocol-port": "http:80" 端口。

步骤4 单击“确定”。工作负载已更新Service。

----结束

HTTP 协议 Service 参数配置

您可以参考下表添加annotations字段配置HTTP协议类型Service的监听器。

 版权所有 © 华为云计算技术有限公司

参数	参数说明
kubernetes.io/elb.http2-enable	表示HTTP/2功能的开启状态。开启后，可提升客户端与LB间的访问性能，但LB与后端服务器间仍采用HTTP1.X协议。 取值范围： ● true：开启HTTP/2功能； ● false：关闭HTTP/2功能（默认为关闭状态）。 **注意** **仅监听器的协议为HTTPS时，支持开启或关闭HTTP/2功能。**当监听器的协议为HTTP时，该字段无效，默认将其设置为false。
kubernetes.io/elb.tls-certificate-ids	ELB服务中的SNI证书ID列表（SNI证书中必须带有域名），不同ID间使用英文逗号隔开。 获取方法：在ELB控制台，进入“弹性负载均衡 > 证书管理”获取。
kubernetes.io/elb.security-pool-protocol	监听器前端协议为HTTPS时，可以开启后端安全协议，开启后后端协议为HTTPS。**不支持更新已有监听器的后端安全协议，更新后仅对新创建的监听器生效**。 ● true：表示开启 ● false：表示关闭
kubernetes.io/elb.tls-ciphers-policy	监听器使用的安全策略。 取值：tls-1-0-inherit,tls-1-0, tls-1-1, tls-1-2,tls-1-2-strict，tls-1-2-fs，tls-1-0-with-1-3, tls-1-2-fs-with-1-3, hybrid-policy-1-0，默认：tls-1-0。 仅对HTTPS协议类型的监听器且关联LB为独享型时有效。
kubernetes.io/elb.x-forwarded-port	开启后可以将ELB实例的监听端口从报文的HTTP头中带到后端云服务器。 ● true：表示开启 ● false：表示关闭 仅独享型ELB的HTTP/HTTPS类型监听器支持配置。
kubernetes.io/elb.x-forwarded-for-port	开启后可以将客户端的源端口从报文的HTTP头中带到后端云服务器。 ● true：表示开启 ● false：表示关闭 仅独享型ELB的HTTP/HTTPS类型监听器支持配置。
kubernetes.io/elb.x-forwarded-host	开启后将以客户端请求头的Host重写X-Forwarded-Host传递到后端云服务器。 ● true：表示开启 ● false：表示关闭 仅独享型ELB的HTTP/HTTPS类型监听器支持配置。

版权所有 © 华为云计算技术有限公司

参数	参数说明
kubernetes.io/elb.gzip-enabled	数据压缩。 • true：开启，将对特定文件类型进行压缩。 • false：关闭，不会对任何文件类型进行压缩。在默认情况下数据压缩为关闭。 支持的压缩类型如下： • Brotli支持压缩所有类型。 • Gzip支持压缩的类型包括：text/xml text/plain text/css application/javascript application/x-javascript application/rss+xml application/atom+xml application/xml application/json。 仅独享型ELB的HTTP/HTTPS类型监听器支持配置。

4.2.1.4.2 为负载均衡类型的 Service 配置 HTTPS 协议

配置 HTTPS 协议 Service

在工作负载创建完成后对Service进行配置，此配置对工作负载状态无影响，且实时生效。具体操作如下：

步骤1 登录ELB服务**创建证书**。

步骤2 登录**云容器实例 CCI2.0**控制台。

步骤3 左侧导航栏中选择“服务管理”，在右侧页面单击“YAML创建”。

步骤4 导入或者添加Service yaml。

Service文件格式说明

- Service.yaml资源描述

```
apiVersion: cci/v2
kind: Service
metadata:
  name: kubectl-test
  namespace: kubectl
  annotations:
    kubernetes.io/elb.class: elb
    kubernetes.io/elb.id: 1234567890 #关联的elb id，只支持独享型elb，不支持共享型elb。
    kubernetes.io/elb.protocol-port: "https:443"  # HTTPS协议及端口号，需要与spec.ports中的端口号对应。
    kubernetes.io/elb.cert-id: "17e3b4f4bc40471c86741dc3aa211379"  # ELB服务中的证书ID。
spec:
  selector:
    app: kubectl-test #关联的负载label。
  ports:
    - name: service-0
      targetPort: 80   #容器开放的端口。
      port: 443        #访问端口。
      protocol: TCP    #协议需指定为TCP。
  type: LoadBalancer
```

- Service.json资源描述

```
{
    "apiVersion": "cci/v2",
    "kind": "Service",
```


版权所有 © 华为云计算技术有限公司

```
  "metadata": {
     "name": "kubectl-test",
     "namespace": "kubectl",
     "annotations": {
                  "kubernetes.io/elb.class": "elb",
        "kubernetes.io/elb.id": "1234567890"  #关联的elb id，只支持独享型elb，不支持共享型elb。
                  "kubernetes.io/elb.protocol-port": "https:443"  #HTTPS协议及端口号，需要与spec.ports
中的端口号对应。
                  "kubernetes.io/elb.cert-id": "17e3b4f4bc40471c86741dc3aa211379" # ELB服务中的证
书ID。
     }
  },
  "spec": {
     "selector": {
        "app": "kubectl-test" #关联的负载label。
     },
     "ports": [
        {
           "name": "service-0",
           "targetPort": 80,     #容器开放的端口。
           "port": 443,          #访问端口。
           "protocol": "TCP",    #协议需指定为TCP。
           "type": "LoadBalancer"
        }
     ]
  }
}
```

步骤5 单击“确定”，创建Service。查看服务管理页面，根据访问地址IP和端口映射中的访问端口[IP:访问端口]，即可访问负载。

----结束

更新 Service

您可以在添加完Service后，更新此Service的访问端口。操作如下：

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“服务管理 ”，在服务管理页面中，选择对应的命名空间，单击需要更新端口配置的Service后的“YAML编辑”。

步骤3 仅支持更新Service内访问端口参数修改：

端口配置：spec.ports[i].port：访问端口，端口范围为1-65535，可任意指定。需同步修改 "kubernetes.io/elb.protocol-port": "https:443" 端口。

步骤4 单击“确定”。工作负载已更新Service。

----结束

HTTPS 协议 Service 参数配置

您可以参考下表添加annotations字段配置HTTPS协议类型Service的监听器。

 版权所有 © 华为云计算技术有限公司

参数	参数说明
kubernetes.io/elb.http2-enable	表示HTTP/2功能的开启状态。开启后，可提升客户端与LB间的访问性能，但LB与后端服务器间仍采用HTTP1.X协议。 取值范围： ● true：开启HTTP/2功能； ● false：关闭HTTP/2功能（默认为关闭状态）。 **注意** **仅监听器的协议为HTTPS时，支持开启或关闭HTTP/2功能。**当监听器的协议为HTTP时，该字段无效，默认将其设置为false。
kubernetes.io/elb.tls-certificate-ids	ELB服务中的SNI证书ID列表（SNI证书中必须带有域名），不同ID间使用英文逗号隔开。 获取方法：在ELB控制台，进入“弹性负载均衡 > 证书管理”获取。
kubernetes.io/elb.security-pool-protocol	监听器前端协议为HTTPS时，可以开启后端安全协议，开启后后端协议为HTTPS。**不支持更新已有监听器的后端安全协议，更新后仅对新创建的监听器生效**。 ● true：表示开启 ● false：表示关闭
kubernetes.io/elb.tls-ciphers-policy	监听器使用的安全策略。 取值：tls-1-0-inherit,tls-1-0, tls-1-1, tls-1-2,tls-1-2-strict，tls-1-2-fs，tls-1-0-with-1-3, tls-1-2-fs-with-1-3, hybrid-policy-1-0，默认：tls-1-0。 仅对HTTPS协议类型的监听器且关联LB为独享型时有效。
kubernetes.io/elb.x-forwarded-port	开启后可以将ELB实例的监听端口从报文的HTTP头中带到后端云服务器。 ● true：表示开启 ● false：表示关闭 仅独享型ELB的HTTP/HTTPS类型监听器支持配置。
kubernetes.io/elb.x-forwarded-for-port	开启后可以将客户端的源端口从报文的HTTP头中带到后端云服务器。 ● true：表示开启 ● false：表示关闭 仅独享型ELB的HTTP/HTTPS类型监听器支持配置。
kubernetes.io/elb.x-forwarded-host	开启后将以客户端请求头的Host重写X-Forwarded-Host传递到后端云服务器。 ● true：表示开启 ● false：表示关闭 仅独享型ELB的HTTP/HTTPS类型监听器支持配置。

版权所有 © 华为云计算技术有限公司

参数	参数说明
kubernetes.io/elb.gzip-enabled	数据压缩。 • true：开启，将对特定文件类型进行压缩。 • false：关闭，不会对任何文件类型进行压缩。在默认情况下数据压缩为关闭。 支持的压缩类型如下： • Brotli支持压缩所有类型。 • Gzip支持压缩的类型包括：text/xml text/plain text/css application/javascript application/x-javascript application/rss+xml application/atom+xml application/xml application/json。 仅独享型ELB的HTTP/HTTPS类型监听器支持配置。

4.2.1.5 为 Service 配置黑名单/白名单访问策略

概述

使用Service服务时，您可以通过添加白名单和黑名单的方式控制访问负载均衡监听器的IP。

- 白名单：指定的IP允许访问，而其它IP拒绝访问。
- 黑名单：指定的IP拒绝访问，而其它IP允许访问。

前置条件

已在ELB控制台创建IP地址组，详情请参见**创建IP地址组**。

配置黑名单/白名单访问策略

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“服务管理”，在右侧页面单击“YAML创建”。

步骤3 导入或者添加Service yaml，配置参数见**表1**。

Service文件格式说明

- Service.yaml资源描述

```
apiVersion: cci/v2
kind: Service
metadata:
  name: kubectl-test
  namespace: kubectl
  annotations:
    kubernetes.io/elb.class: elb
    kubernetes.io/elb.id: 1234567890 #关联的elb id，只支持独享型elb，不支持共享型elb。
    kubernetes.io/elb.acl-id: <your_acl_id>              # ELB的IP地址组ID
    kubernetes.io/elb.acl-type: 'white'                  # 白名单控制
spec:
  selector:
    app: kubectl-test
  ports:
```


版权所有 © 华为云计算技术有限公司

```
  - name: service-0
    targetPort: 80   #容器开放的端口。
    port: 12222      #访问端口。
    protocol: TCP    #访问负载的协议。
type: LoadBalancer
```

- Service.json资源描述

```
{
   "apiVersion": "cci/v2",
   "kind": "Service",
   "metadata": {
      "name": "kubectl-test",
      "namespace": "kubectl",
      "annotations": {
                  "kubernetes.io/elb.class": "elb"
         "kubernetes.io/elb.id": "1234567890"  #关联的elb id，只支持独享型elb，不支持共享型elb。
                  "kubernetes.io/elb.acl-id": "<your_acl_id>"            # ELB的IP地址组ID
                  "kubernetes.io/elb.acl-type": "white"                  # 白名单控制
      }
   },
   "spec": {
      "selector": {
         "app": "kubectl-test"
      },
      "ports": [
         {
            "name": "service-0",
            "targetPort": 80,     #容器开放的端口。
            "port": 12222,        #访问端口。
            "protocol": "TCP",    #访问负载的协议。
            "type": "LoadBalancer"
         }
      ]
   }
}
```

步骤4 单击“确定”，创建Service。查看服务管理页面，根据访问地址IP和端口映射中的访问端口[IP地址:访问端口]，即可访问负载。

- 如果配置策略为白名单，仅白名单中IP地址可以访问，其余IP地址拒绝访问。
- 如果配置策略为黑名单，则黑名单中IP地址拒绝访问，其余IP地址均可访问。

----结束

表 4-10 ELB 访问控制注解

参数	类型	描述
kubernetes.io/elb.acl-id	String	• 不填写该参数时：表示CCI不对ELB侧访问控制进行修改。 • 参数值填写为ELB的IP地址组ID时：表示开启访问控制，为ELB设置IP地址黑名单或白名单。 • 参数可以填写最多5个IP地址组ID，用“,”隔开。 • IP地址组ID获取方法： 登录控制台后，单击顶部菜单右侧的“网络 > 弹性负载均衡ELB”，在网络控制台上单击“弹性负载均衡 > IP地址组”，复制目标IP地址组的“ID”即可。详情请参见**IP地址组**。

版权所有 © 华为云计算技术有限公司

参数	类型	描述
kubernetes.io/elb.acl-type	String	为ELB设置IP地址黑名单或白名单时需填写，取值如下： ● black：表示黑名单，仅所选IP地址组无法访问ELB地址。 ● white：表示白名单，仅所选IP地址组可以访问ELB地址。 如果填写了kubernetes.io/elb.acl-id，且kubernetes.io/elb.acl-type字段没有填写时，默认策略为白名单访问。

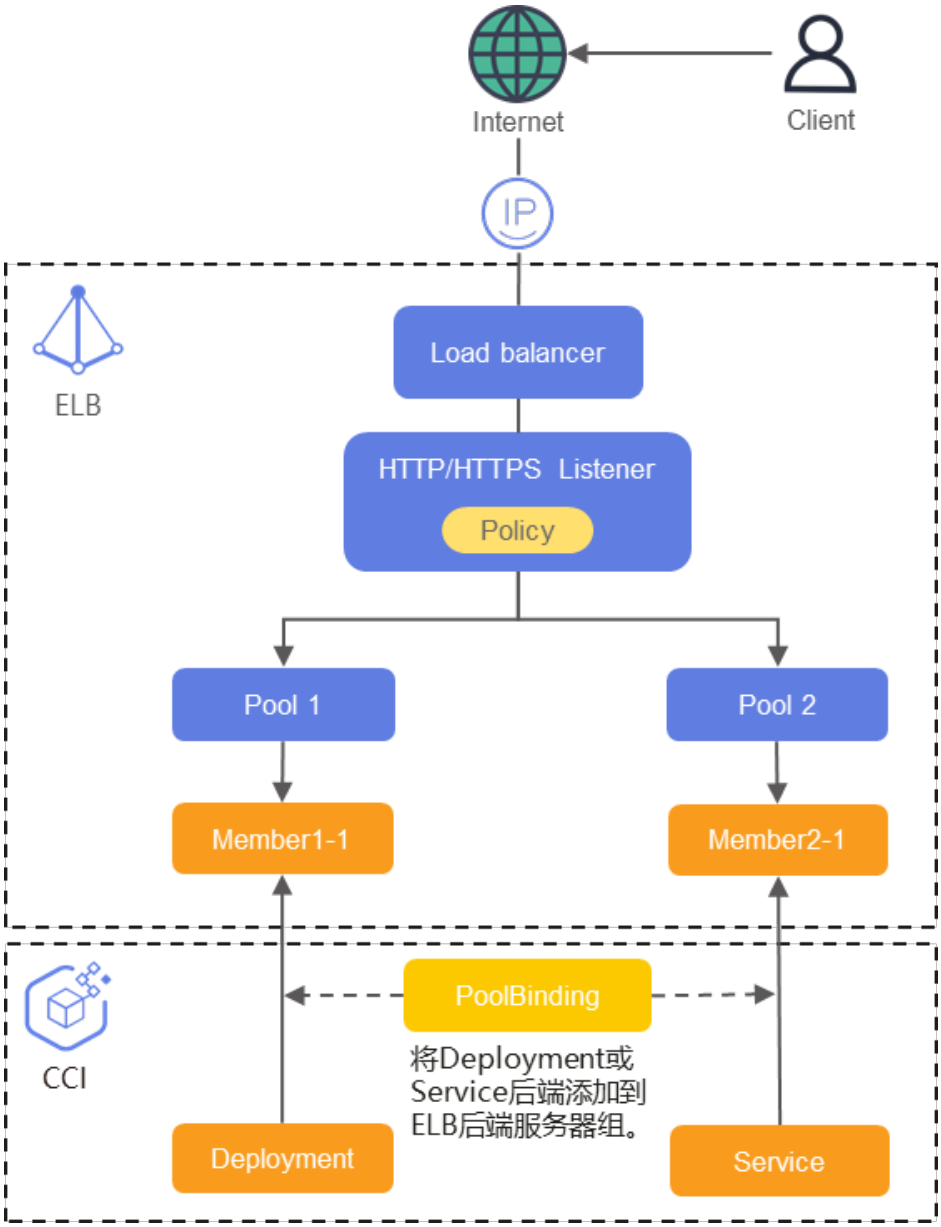
4.2.2 网关（Gateway）

4.2.2.1 后端服务器组关联配置（PoolBinding）

4.2.2.1.1 PoolBinding 概述

您在CCI 2.0服务部署工作负载后，可以通过弹性负载均衡（ELB）服务实现公网访问负载的目的。针对大量HTTP/HTTPS访问的需求场景，CCI 2.0提供了PoolBinding对象。PoolBinding可以将与之关联的Deployment或Service后端实例IP自动更新至弹性负载均衡（ELB）关联的后端服务器组（Pool）后端。您可以在弹性负载均衡（ELB）侧自定义负载均衡器和监听器配置，以及配置高级转发策略等，实现七层负载均衡的能力。

版权所有 © 华为云计算技术有限公司



约束与限制

- PoolBinding仅支持关联混合类型，转发模式为负载均衡的后端服务器组（Pool），CCI 2.0当前仅支持配置后端IP地址为IPV4类型，不支持IPV6类型。
- 一个后端服务组（Pool）只能与一个PoolBinding对象关联，后端服务器组支持与独享型ELB关联，不支持共享型ELB。创建的ELB和Pool所在VPC需与PoolBinding所在Namespace的VPC相同。
- PoolBinding支持配置的后端对象包括无状态负载（Deployment）和服务（Service）。
- 目前CCI 2.0 控制台界面暂不支持PoolBinding对象创建，用户可以通过curl命令，或者ccictl工具创建，ccictl配置步骤请参考**ccictl配置指南**。当前支持的PoolBinding对象操作有创建、删除和查询，不支持更新。
- 弹性负载均衡配置的约束与限制，请参考**弹性负载均衡约束与限制**。

版权所有 © 华为云计算技术有限公司

4.2.2.1.2 PoolBinding 后端关联 Deployment

PoolBinding支持配置一个后端对象，后端对象可以为Deployment。创建PoolBinding对象后，会自动将Deployment后端关联的Pod的Endpoint刷新至关联的后端服务器组（Pool）后端。

前提条件

- 在弹性负载均衡（ELB）已创建后端服务器组。
- 在云容器实例 CCI 2.0 console页完成工作负载（Deployment）创建。

方法一：通过 ccictl 命令管理 PoolBinding 对象

- poolbinding.yaml示例

```
apiVersion: loadbalancer.networking.openvessel.io/v1
kind: PoolBinding
metadata:
  name: test-c1        //创建的PoolBinding对象名称。
  namespace: test-ns   //创建的PoolBinding对象所属的命名空间，需与关联的Deployment在同一命名空间下。
spec:
  poolRef:
    id: e08*****-****-****-****-********29c1  //关联的后端服务器组ID，即创建的后端服务器组ID。
  targetRef:
    kind: Deployment   //关联的后端对象类型，示例为Deployment。
    group: cci/v2      //关联的后端对象group。
    name: test-name    //关联的后端对象名称，即创建的Deployment名称。
    port: 80           //关联的后端对象目标端口号，若后端对象为Deployment，即开放的容器端口。
```

- ccictl常用命令

```
// 创建PoolBinding
ccictl create -f poolbinding.yaml
// 查询namespace下所有的PoolBinding的详细信息。
ccictl get plb -n $namespace -oyaml
// 查询namespace下单个PoolBinding的详细信息。
ccictl get plb -n $namespace $name -oyaml
// 删除PoolBinding。
ccictl delete plb -n $namespace $name
```

表 4-11 主要参数说明

参数	参数说明
$namespace-name	创建的PoolBinding对象所属的命名空间，需与关联的Deployment在同一命名空间下。
$name	PoolBinding对象名称。

方法二：通过 curl 命令管理 PoolBinding 对象

- poolbinding.json示例

```
{
   "apiVersion": "loadbalancer.networking.openvessel.io/v1",
   "kind": "PoolBinding",
   "metadata": {
      "name": "test-c1",       //创建的PoolBinding对象名称。
      "namespace": "test-ns"   //创建的PoolBinding对象所属的命名空间，需与关联的Deployment在同一命名空间下。
   },
   "spec": {
      "poolRef": {
```

 版权所有 © 华为云计算技术有限公司

```
            "id": "e08*****-****-****-****-********29c1"  //关联的后端服务器组ID，即创建的后端服务器组ID。
        },
        "targetRef": {
            "kind": "Deployment",    //关联的后端对象类型，示例为Deployment。
            "group": "cci/v2",       //关联的后端对象group。
            "name": "test-name",     //关联的后端对象名称，即创建的Deployment名称。
            "port": 80               //关联的后端对象目标端口号，若后端对象为Deployment，即开放的容器端口。
        }
    }
}
```

- curl常用命令

```
// 创建PoolBinding。
curl -k -H "Content-Type: application/json" -H "X-Auth-Token: $token" "https://$endpoint/apis/loadbalancer.networking.openvessel.io/v1/namespaces/$namespace-name/poolbindings" -X POST -d@poolbinding.json
// 查询namespace下所有的PoolBinding的详细信息。
curl -k -H "Content-Type: application/json" -H "X-Auth-Token: $token" "https://$endpoint/apis/loadbalancer.networking.openvessel.io/v1/namespaces/$namespace-name/poolbindings"
// 查询namespace下单个PoolBinding的详细信息。
curl -k -H "Content-Type: application/json" -H "X-Auth-Token: $token" "https://$endpoint/apis/loadbalancer.networking.openvessel.io/v1/namespaces/$namespace-name/poolbindings/$name"
// 删除PoolBinding。
curl -k -H "Content-Type: application/json" -H "X-Auth-Token: $token" "https://$endpoint/apis/loadbalancer.networking.openvessel.io/v1/namespaces/$namespace-name/poolbindings/$name" -X DELETE
```

表 4-12 主要参数说明

参数	参数说明
$token	通过Token认证获取请求。
$endpoint	CCI 2.0的endpoint。 终端节点（Endpoint）即调用API的**请求地址**，不同服务不同区域的终端节点不同，您可以从**地区和终端节点**查询服务的终端节点。
$namespace-name	创建的PoolBinding对象所属的命名空间，需与关联的Deployment在同一命名空间下。
$name	PoolBinding对象名称。

4.2.2.1.3 PoolBinding 后端关联 Service

PoolBinding支持配置一个后端对象，后端对象可以为Service。创建PoolBinding对象后，会自动将Service后端关联的Pod的Endpoint刷新至关联的后端服务器组（Pool）后端。

约束与限制

不支持PooBinding关联的后端服务器组为Service自动生成的后端服务器组。

前提条件

- 在弹性负载均衡（ELB）已创建后端服务器组。
- 在云容器实例 CCI 2.0 console页完成工作负载（Deployment）创建。

版权所有 © 华为云计算技术有限公司

- 在云容器实例 CCI 2.0 console页完成服务（Service）创建。

方法一：通过 ccictl 命令管理 PoolBinding 对象

- poolbinding.yaml示例

```
apiVersion: loadbalancer.networking.openvessel.io/v1
kind: PoolBinding
metadata:
  name: test-c1        //创建的PoolBinding对象名称。
  namespace: test-ns   //创建的PoolBinding对象所属的命名空间，需与关联的Service在同一命名空间下。
spec:
  poolRef:
    id: e08*****-****-****-****-********29c1  //关联的后端服务器组ID，即创建的后端服务器组ID。
  targetRef:
    kind: Service     //关联的后端对象类型，示例为Service。
    group: cci/v2     //关联的后端对象group。
    name: test-name   //关联的后端对象名称，即创建的Service名称。
    port: 123         //关联的后端对象目标端口号，若后端对象为Service，即Serivce中配置的访问端口。
```

- ccictl常用命令

```
// 创建PoolBinding
ccictl create -f poolbinding.yaml
// 查询namespace下所有的PoolBinding的详细信息。
ccictl get poolbindings -n $namespace -oyaml
// 查询namespace下单个PoolBinding的详细信息。
ccictl get poolbindings -n $namespace $name -oyaml
// 删除PoolBinding。
ccictl delete poolbindings -n $namespace $name
```

表 4-13 主要参数说明

参数	参数说明
$namespace-name	创建的PoolBinding对象所属的命名空间，需与关联的Deployment在同一命名空间下。
$name	PoolBinding对象名称。

方法二：通过 curl 命令管理 PoolBinding 对象

- poolbinding.json示例

```
{
    "apiVersion": "loadbalancer.networking.openvessel.io/v1",
    "kind": "PoolBinding",
    "metadata": {
        "name": "test-c1",       //创建的PoolBinding对象名称。
        "namespace": "test-ns"   //创建的PoolBinding对象所属的命名空间，需与关联的Service在同一命名空间下。
    },
    "spec": {
        "poolRef": {
            "id": "e08*****-****-****-****-********29c1"  //关联的后端服务器组ID，即创建的后端服务器组ID。
        },
        "targetRef": {
            "kind": "Service",    //关联的后端对象类型，示例为Service。
            "group": "cci/v2",    //关联的后端对象group。
            "name": "test-name",  //关联的后端对象名称，即创建的Service名称。
            "port": 123           //关联的后端对象目标端口号，若后端对象为Service，即Serivce中配置的访问端口。
        }
    }
}
```


版权所有 © 华为云计算技术有限公司

- curl常用命令

```
// 创建PoolBinding。
curl -k -H "Content-Type: application/json" -H "X-Auth-Token: $token" "https://$endpoint/apis/loadbalancer.networking.openvessel.io/v1/namespaces/$namespace-name/poolbindings" -X POST -d@poolbinding.json
// 查询namespace下所有的PoolBinding的详细信息。
curl -k -H "Content-Type: application/json" -H "X-Auth-Token: $token" "https://$endpoint/apis/loadbalancer.networking.openvessel.io/v1/namespaces/$namespace-name/poolbindings"
// 查询namespace下单个PoolBinding的详细信息。
curl -k -H "Content-Type: application/json" -H "X-Auth-Token: $token" "https://$endpoint/apis/loadbalancer.networking.openvessel.io/v1/namespaces/$namespace-name/poolbindings/$name"
// 删除PoolBinding。
curl -k -H "Content-Type: application/json" -H "X-Auth-Token: $token" "https://$endpoint/apis/loadbalancer.networking.openvessel.io/v1/namespaces/$namespace-name/poolbindings/$name" -X DELETE
```

表 4-14 主要参数说明

参数	参数说明
$token	通过Token认证获取请求。
$endpoint	CCI 2.0的endpoint。 终端节点（Endpoint）即调用API的**请求地址**，不同服务不同区域的终端节点不同，您可以从**地区和终端节点**查询服务的终端节点。
$namespace-name	创建的PoolBinding对象所属的命名空间，需与关联的Deployment在同一命名空间下。
$name	PoolBinding对象名称。

4.2.3 Pod 指定子网配置

操作场景

如果Pod的网络配置包含多个子网时，支持通过Annotation（yangtse.io/subnets）配置Pod创建后使用的子网。

约束与限制

- Pod指定的子网最多不能超过20个。
- Pod指定的子网必须包含在 Pod网络配置（network）资源内，不能随意指定。
- 可以支持为Pod配置单个或者多个子网，多个子网使用逗号 (,) 分隔。
- 必须保证指定的子网下存在空闲的IP地址，否则会因为IP不足导致Pod创建失败。

通过 ccictl 命令行设置

您可以通过对工作负载添加annotations控制Pod使用的子网，如下所示。

```
apiVersion: cci/v2
kind: Deployment
metadata:
  annotations:
    description: ''
  labels: {}
```


版权所有 © 华为云计算技术有限公司

```
  name: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      annotations:
        vm.cci.io/pod-size-specs: 2.00_4.0
        resource.cci.io/pod-size-specs: 2.00_4.0
        metrics.alpha.kubernetes.io/custom-endpoints: '[{api:"",path:"",port:"",names:""}]'
        yangtse.io/subnets: ${subnetID1},${subnetID2}   # Pod指定子网
        log.stdoutcollection.kubernetes.io: '{"collectionContainers": ["container-0"]}'
      labels:
        app: nginx
    spec:
      containers:
        - image: library/nginx:stable-alpine-perl
          name: container-0
          resources:
            limits:
              cpu: 2000m
              memory: 4096Mi
            requests:
              cpu: 2000m
              memory: 4096Mi
          command: []
          lifecycle: {}
      dnsPolicy: ''
      imagePullSecrets:
        - name: imagepull-secret
      dnsConfig: {}
  minReadySeconds: 0
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 0
      maxUnavailable: 1
```

说明

yangtse.io/subnets：Pod需要配置的子网ID。

4.3 弹性伸缩

4.3.1 水平扩缩容

水平扩缩容（Horizontal Pod Autoscaler, HPA）是一种资源对象，能够根据某些指标对无状态负载中的容器组数量进行动态伸缩，使运行在上面的服务对指标的变化有一定的自适应能力。HPA会根据租户配置的策略自动增加或减少容器组数量以满足租户需求。

约束与限制

在CCI 2.0，每个容器组所在环境都存在辅助运行的系统组件，它们将会占用底层部分CPU或内存等资源，可能会导致容器组的资源使用达不到预期上限，从而达不到您设置的HPA规则的门槛。您可参考预留系统开销说明避免此类情形。

版权所有 © 华为云计算技术有限公司

弹性伸缩

说明

- 当配置spec.metrics.resource.target.type为Utilization时，您需要在创建工作负载时指定资源的requests值。
- 当配置spec.metrics.resource.target.type为Utilization时，其数值的计算规则为：资源使用率 = 资源使用量 / 用户实际可使用的容器组规格。您可参考**容器组规格的计算方法**来确认您的容器组实际规格。

您可以根据业务需求自行定义伸缩策略，包括弹性指标、阈值以及步长等，以降低人为反复调整资源以应对业务变化和高峰压力的工作量，帮助您节约资源和人力成本。当前支持一种弹性伸缩策略：

告警策略：支持根据CPU/内存的使用率，进行工作负载的自动伸缩。工作负载创建完成后即可设置，在CPU/内存超过或少于一定值时，自动增减实例。

- **告警策略**：支持根据CPU/内存的使用率，进行工作负载的自动伸缩。
  a. 登录**云容器实例 CCI2.0**控制台。
  b. 在左侧导航栏中选择“负载管理 > 无状态负载”，单击工作负载名称进入负载详情。
  c. 在“负载详情”中，单击“弹性伸缩”，单击“YAML创建”配置弹性伸缩策略。

  弹性伸缩策略文件格式说明

  - hpa.yaml资源描述

```
kind: HorizontalPodAutoscaler
apiVersion: cci/v2
metadata:
  name: nginx              # HPA的名称
  namespace: test          # HPA所在的命名空间
spec:
  scaleTargetRef:         # 引用进行自动伸缩的目标资源
    kind: Deployment      # 目标资源类型，例Deployment
    name: nginx           # 目标资源名称
    apiVersion: cci/v2    # 目标资源版本
  minReplicas: 1          # HPA伸缩的最小副本数
  maxReplicas: 5          # HPA伸缩的最大副本数
  metrics:
    - type: Resource                # 使用资源指标
      resource:
        name: memory                # 资源名称，例如cpu或memory
        target:
          type: Utilization         # 指标类型，可以是Utilization（百分比）或AverageValue（绝对值）
          averageUtilization: 50    # 资源使用率，例当CPU使用率达到50%时触发扩容
  behavior:
    scaleUp:
      stabilizationWindowSeconds: 30  # 扩容后稳定时间，单位为秒
      policies:
      - type: Pods           #指定扩缩容的Pods数量
        value: 1
        periodSeconds: 30  # 每 30 秒检查一次
    scaleDown:
      stabilizationWindowSeconds: 30  # 缩容后稳定时间，单位为秒
      policies:
      - type: Percent      #按照现有Pods的百分比来扩缩容
        value: 50
        periodSeconds: 30  # 每 30 秒检查一次
```

  - hpa.json资源描述

```
{
  "kind": "HorizontalPodAutoscaler",
```

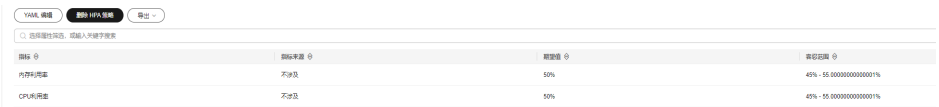

版权所有 © 华为云计算技术有限公司

```
  "apiVersion": "cci/v2",
  "metadata": {
     "name": "nginx",              # HPA的名称
     "namespace": "test"           # HPA所在的命名空间
  },
  "spec": {
     "scaleTargetRef": {         # 引用进行自动伸缩的目标资源
        "kind": "Deployment",        # 目标资源类型，例Deployment
        "name": "nginx",             # 目标资源名称
        "apiVersion": "cci/v2"       # 目标资源版本
     },
     "minReplicas": 1,             # HPA伸缩的最小副本数
     "maxReplicas": 5,             # HPA伸缩的最大副本数
     "metrics": [
        {
           "type": "Resource",             # 使用资源指标
           "resource": {
              "name": "memory",       # 资源名称，例如cpu或memory
              "target": {
                 "type": "Utilization",          # 指标类型，可以是Utilization（百分比）或
AverageValue（绝对值）
                 "averageUtilization": 50        # 资源使用率，例当CPU使用率达到50%时触
发扩容
              }
           }
        }
     ],
     "behavior": {
        "scaleUp": {
           "stabilizationWindowSeconds": 30,
           "policies": [
              {
                 "type": "Pods",
                 "value": 1,
                 "periodSeconds": 30
              }
           ]
        },
        "scaleDown": {
           "stabilizationWindowSeconds": 30,
           "policies": [
              {
                 "type": "Percent",
                 "value": 50,
                 "periodSeconds": 30
              }
           ]
        }
     }
  }
}
```

d. 单击“确定”。

可看到弹性伸缩策略。

图 4-12 弹性伸缩策略



待到触发条件发生时，弹性伸缩策略会自动启动。

版权所有 © 华为云计算技术有限公司

4.4 存储管理

4.4.1 存储概述

云容器实例支持临时存储空间扩容和持久化存储，以满足您的存储需求。创建工作负载时，可以使用以下类型的存储。

临时存储空间扩容

CCI默认为pod分配30GiB的临时存储空间，该存储空间与pod的生命周期保持一致，默认分配的30GiB存储空间免费。当需要使用更大的临时存储空间时，用户可以自定义进行扩容。详情请参见**4.4.2**。

极速文件存储卷（SFS Turbo）

云容器实例支持创建SFS Turbo极速文件存储卷并挂载到容器的某一路径下，极速文件存储卷具有按需申请，快速供给，弹性扩展，方便灵活等特点，适用于DevOps、容器微服务、企业办公等应用场景。详情请参见**4.4.3**。

并行文件系统（OBS）

云容器实例支持创建并行文件系统，并行文件系统（Parallel File System）是对象存储服务（Object Storage Service，OBS）提供的一种经过优化的高性能文件语义系统，旨在为基于对象存储服务作为统一数据湖存储的大数据场景提供解决方案。并行文件系统提供毫秒级别访问时延、TB/s级别带宽和百万级别的IOPS、高兼容性、高性能、高可扩展性、高可靠性的能力。详情请参见**4.4.4**。

4.4.2 临时存储空间扩容

使用场景

CCI默认为pod分配30GiB的临时存储空间，该存储空间与pod的生命周期保持一致，默认分配的30GiB存储空间免费。当pod需要写入大量数据至rootfs或emptyDir以及镜像大于30GiB时，您可以参考本章节扩容临时存储空间。

使用须知

- 临时存储空间扩容大小限制为[0,994]。
- 扩展临时存储空间时，由于扩容需要时间，负载创建会慢于常规负载。

使用 YAML 扩展临时存储空间

- 创建容器组时，扩容10GiB临时存储空间YAML定义如下：

```
apiVersion: cci/v2
kind: Pod
metadata:
  name: nginx
  namespace: ns
spec:
  extraEphemeralStorage:
    sizeInGiB: 10
  containers:
```


版权所有 © 华为云计算技术有限公司

```
  - name: container-1
    image: nginx:latest
```

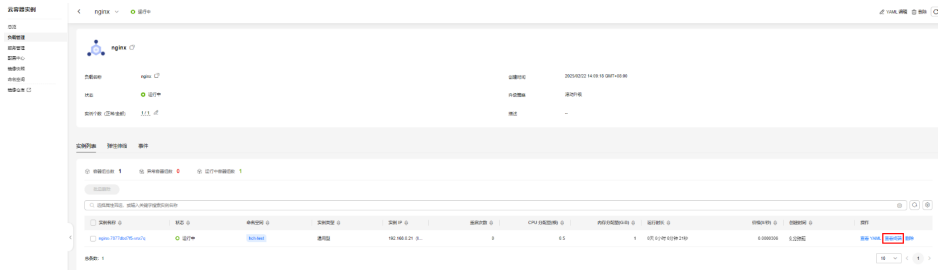
- 创建无状态负载时，扩容10GiB临时存储空间YAML定义如下：

```
apiVersion: cci/v2
kind: Deployment
metadata:
  name: nginx
  namespace: ns
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      extraEphemeralStorage:
        sizeInGiB: 10
      containers:
      - image: nginx:latest
        name: container-0
```

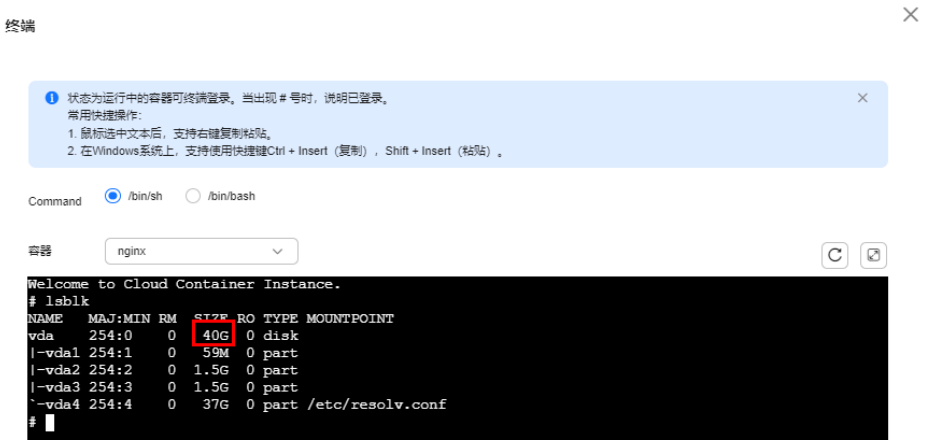
查看是否扩容成功

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 选择“负载管理”，单击负载进入负载详情，单击“查看终端”。



步骤3 输入lsblk并回车，查看扩容后的系统盘大小（默认为30GiB）。



----结束

版权所有 © 华为云计算技术有限公司

4.4.3 极速文件存储卷（SFS Turbo）

云容器实例支持创建**弹性文件存储**SFS Turbo（极速文件存储）并挂载到容器的某一路径下，极速文件存储具有按需申请，快速供给，弹性扩展，方便灵活等特点，适用于DevOps、容器微服务、企业办公等应用场景。

使用限制

- 待挂载的极速文件存储必须是按需付费。更多信息，请参见**极速文件存储计费**。
- 极速文件存储和负载必须在同一个VPC下，否则负载无法使用极速文件存储卷。
- 使用极速文件存储期间，不能修改极速文件存储关联的VPC配置信息，否则CCI中容器无法访问极速文件存储。
- 请谨慎执行极速文件存储的删除操作，以避免造成CCI中容器不可用。
- SFS Turbo文件系统无可用区概念，SFS Turbo PV不支持配置可用区亲和。
- CCI 2.0当前仅支持在西南-贵阳一和华东-上海一局点使用SFS Turbo。

导入极速文件存储

云容器实例目前仅支持通过PVC静态绑定的方式使用极速文件存储。

步骤1 创建SFS Turbo文件系统，详见**创建SFS Turbo文件系统**。

步骤2 创建PV。

1. 获取IAM token

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -X POST -d '{"auth": {"identity":{"methods": ["password"],"password":{"user": {"name": "$username","password": "$password","domain": {"name": "$domain"}}}},"scope": {"project":{"name": "$project"}}}}' https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens | grep "X-Subject-Token"|awk -F ": " '{print $2}'| sed "s/\r//"
```

2. 调用接口创建PV

```
curl -H "X-Auth-Token:$token" -H 'Content-Type: application/json' -X POST -d @pv.json  -k -v https://cci.cn-north-4.myhuaweicloud.com/apis/cci/v2/persistentvolumes
```

其中$token为**步骤2.1**中获取的IAM token，pv.json为您要创建的PV信息，示例json如下：

```
{
    "apiVersion": "cci/v2",
    "kind": "PersistentVolume",
    "metadata": {
        "annotations": {
        },
        "name": "pv-sfs-test"
    },
    "spec": {
        "accessModes": [
            "ReadWriteMany"
        ],
        "capacity": {
            "storage": "500Gi"
```


版权所有 © 华为云计算技术有限公司

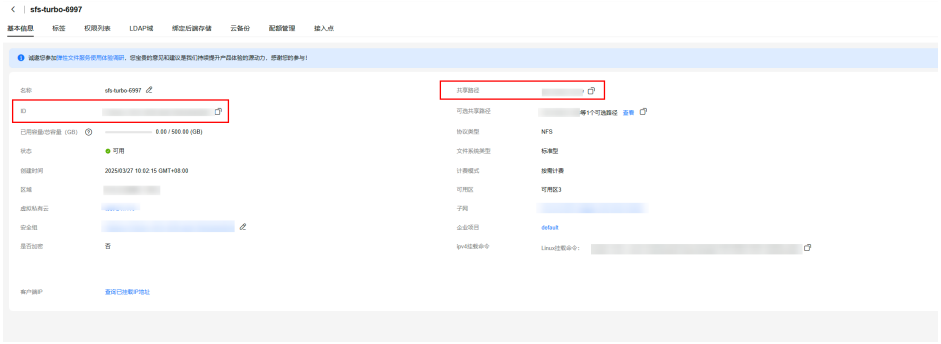
```
        },
        "csi": {
            "driver": "sfsturbo.csi.everest.io",
            "fsType": "nfs",
            "volumeHandle": "**************",
            "volumeAttributes": {
                "everest.io/share-export-location": "*******"
            }
        },
        "persistentVolumeReclaimPolicy": "Retain",
        "storageClassName": "csi-sfsturbo",
        "mountOptions": [
        ]
    }
}
```

表 4-15 关键参数说明

参数	是否必选	参数类型	描述
accessModes	是	List	参数解释：存储访问模式。 约束限制：极速文件存储卷必须为ReadWriteMany。
driver	是	String	参数解释：挂载依赖的存储驱动。 约束限制：极速文件存储卷必须为sfsturbo.csi.everest.io。
fsType	是	String	参数解释：存储实例类型。 约束限制：支持“nfs”，nfs为文件存储卷。
volumeHandle	是	String	参数解释：极速文件存储的ID。 约束限制：必须为已创建的极速文件存储的ID。
persistentVolumeReclaimPolicy	是	String	参数解释：PV卷回收策略。 约束限制：目前只支持Retain回收策略。 **Retain**：删除PVC，PV资源与底层存储资源均不会被删除，需要手动删除回收。PVC删除后PV资源状态为“已释放（Released）”，不能直接再次被PVC绑定使用。
storage	是	String	参数解释：存储容量，单位为Gi。 约束限制：设置为极速文件存储的大小。
storageClassName	是	String	参数解释：极速文件存储卷对应的存储类名称。 约束限制：极速文件存储卷对应的存储类名称为csi-sfsturbo。

版权所有 © 华为云计算技术有限公司

其中volumeHandle为您在**步骤1**创建的SFS Turbo的ID，everest.io/share-export-location为您在**步骤1**创建的SFS Turbo的共享路径。



创建PV请求响应如下。

```
{
  "kind": "PersistentVolume",
  "apiVersion": "cci/v2",
  "metadata": {
    "name": "pv-sfs-test",
    "uid": "bc3698e4-8923-488b-82e5-bab676b8eaa7",
    "resourceVersion": "462672983",
    "creationTimestamp": "2025-03-27T02:28:30Z",
    "labels": {
      "tenant.cci.io/tenant-id": "a                                    "
    },
    "annotations": {
      "tenant.cci.io/tenant-id": "a                                    "
    },
    "finalizers": [
      "kubernetes.io/pv-protection"
    ]
  },
  "spec": {
    "capacity": {
      "storage": "500Gi"
    },
    "csi": {
      "driver": "sfsturbo.csi.everest.io",
      "volumeHandle": "                                                ",
      "fsType": "nfs",
      "volumeAttributes": {
        "everest.io/share-export-location": "                    "
      }
    },
    "accessModes": [
      "ReadWriteMany"
    ],
    "persistentVolumeReclaimPolicy": "Retain",
    "storageClassName": "csi-sfsturbo",
    "volumeMode": "Filesystem"
  },
  "status": {
    "phase": "Pending"
  }
```

步骤3 创建PVC。

1. 获取IAM token，参见**步骤2.1**。
2. 调用接口创建PVC绑定创建的PV。

   ```
   curl -H "X-Auth-Token:$token" -H 'Content-Type: application/json' X POST -d @pvc.json  -k -v https://cci.cn-north-4.myhuaweicloud.com/apis/cci/v2/namespaces/${namespace}/persistentvolumeclaims
   ```

   其中pvc.json为您要创建的PVC信息，示例json如下：

   ```
   {
       "apiVersion": "cci/v2",
       "kind": "PersistentVolumeClaim",
       "metadata": {
           "name": "pvc-sfs",
           "annotations": {
           }
       },
       "spec": {
           "accessModes": [
   ```

 版权所有 © 华为云计算技术有限公司

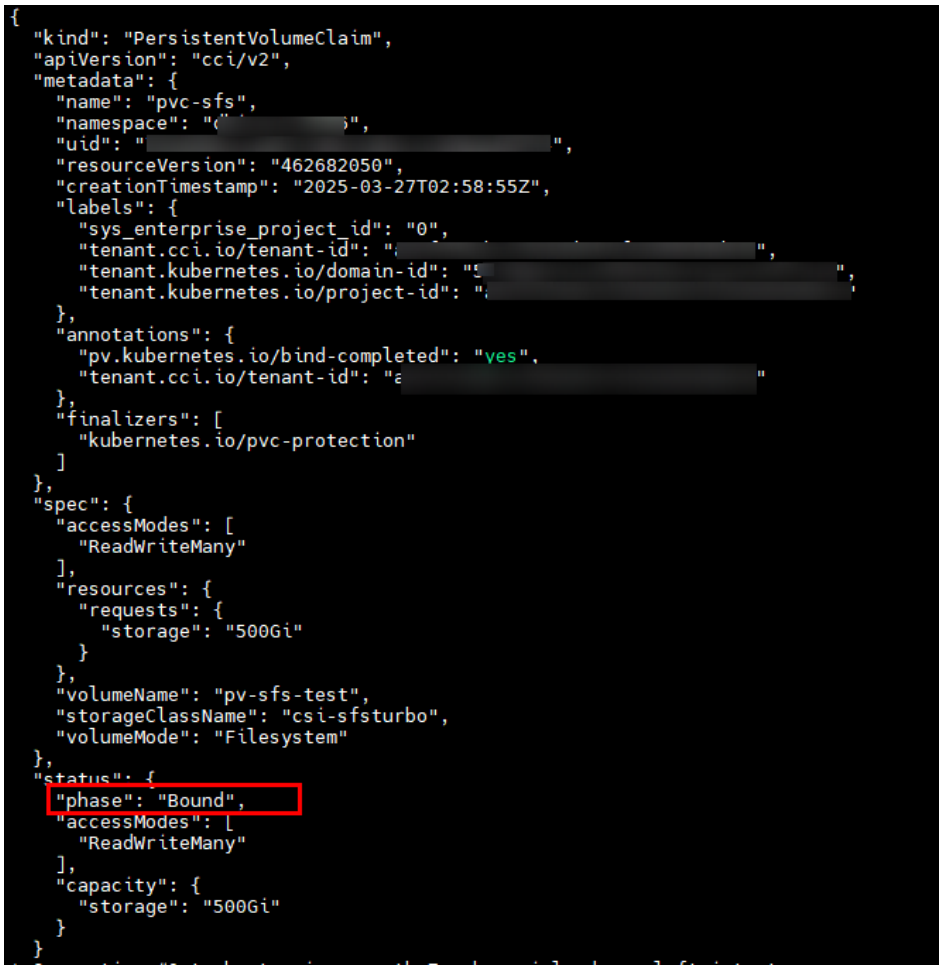
```
            "ReadWriteMany"
        ],
        "resources": {
            "requests": {
                "storage": "500Gi"
            }
        },
        "storageClassName": "csi-sfsturbo",
        "volumeName": "pv-sfs-test"
    }
}
```

表 4-16 关键参数说明

参数	是否必填	参数类型	描述
storage	是	String	参数解释：PVC申请容量，单位为Gi。 约束限制： – 设置为极速文件存储的大小。 – 与表4-15中PV设置的容量值相同。
storageClassName	是	String	参数解释：存储类名称。 约束限制：必须与表4-15中PV的存储类一致，对应极速文件存储卷的存储类名称为csi-sfsturbo。
volumeName	是	String	参数解释：PV的名称。 约束限制：必须与表4-15中PV名称一致。

其中volumeName为您在步骤3创建的PV名称。
创建PVC请求响应如下，可以看到PVC已经绑定到PV。

版权所有 © 华为云计算技术有限公司



----结束

使用极速文件存储卷

请参考**4.1-无状态负载**。在负载YAML中增加volume相关配置。

```
kind: Deployment
apiVersion: cci/v2
metadata:
  name: nginx
  namespace: test-ns
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      volumes:
        - name: my-storage
          persistentVolumeClaim:
            claimName: PVC名称
      containers:
        - name: nginx
          image: nginx:latest
          resources:
            limits:
```


版权所有 © 华为云计算技术有限公司

```
            cpu: 500m
            memory: 1Gi
          requests:
            cpu: 500m
            memory: 1Gi
        volumeMounts:
          - name: my-storage
            mountPath: 容器内挂载路径
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
    restartPolicy: Always
    terminationGracePeriodSeconds: 30
    dnsPolicy: Default
    securityContext: {}
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxUnavailable: 25%
    maxSurge: 25%
```

说明

- 创建极速文件存储过程中需要创建单独的虚拟机，耗时较长。因此当前仅支持使用已有的极速文件存储卷。
- 挂载子路径为极速文件存储根路径下的子路径，如果不存在会自动在文件存储中创建。该路径必须为相对路径。
- 使用极速文件存储卷，只支持通过YAML或者ccictl方式创建负载。

4.4.4 并行文件系统（OBS）

4.4.4.1 并行文件系统概述

并行文件系统介绍

并行文件系统（Parallel File System）是对象存储服务（Object Storage Service，OBS）提供的一种经过优化的高性能文件语义系统，旨在为基于对象存储服务作为统一数据湖存储的大数据场景提供解决方案。并行文件系统提供毫秒级别访问时延、TB/s级别带宽和百万级别的IOPS、高兼容性、高性能、高可扩展性、高可靠性的能力。

与对象桶的结构不同，并行文件系统访问路径中的每级目录都是一个独立的目录对象，例如“/dir01/dir02/example.txt”中“/dir01/”和“/dir01/dir02/”是目录对象，“/dir01/dir02/example.txt”是目录中的文件对象。在分层目录结构下，修改目录名仅需重命名单个目录对象，无需列举并修改指定目录前缀的全量对象。这种分层结构使并行文件系统的数据组织方式与HDFS基本一致，使用HDFS作为数据访问层的大数据分析框架可以通过OBSFileSystem插件（OBSA-HDFS）访问并行文件系统数据。详细介绍请参见**什么是并行文件系统**。

性能说明

容器负载挂载并行文件系统时，每挂载一个并行文件系统卷，后端会产生一个常驻进程。当负载使用并行文件系统数过多或大量读写并行文件系统文件时，常驻进程会占用大量内存，部分场景下内存消耗量参考**表1 单个并行文件系统常驻进程内存消耗**，为保证负载稳定运行，建议负载使用的并行文件系统卷数量不超过其申请的内存GiB数量，如负载的申请的内存规格为4GiB，则建议其使用的并行文件系统数**不超过**4个。

版权所有 © 华为云计算技术有限公司

表 4-17 单个并行文件系统常驻进程内存消耗

测试项目	内存消耗
长稳运行	约50M
2并发写10M文件	约110M
4并发写10M文件	约220M
单写100G文件	约300M

前提条件

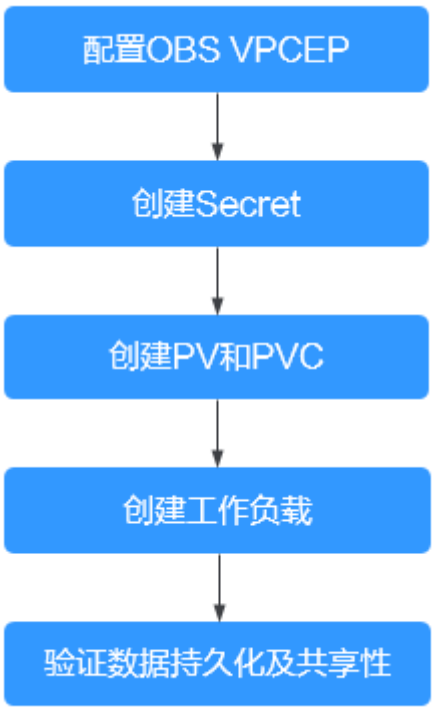
在使用并行文件系统之前，您需要先配置正确的OBS VPCEP，VPCEP详情请参见**什么是VPCEP**，否则可能导致无法挂载并行文件系统，推荐一次性创建所有可能涉及的OBS VPCEP，避免新增并行文件系统等场景时重复操作。请参考**购买云服务VPCEP**创建OBS VPCEP，其中OBS终端节点服务名称请提交工单或联系OBS服务运维人员获取。

使用场景

并行文件系统只支持**通过静态存储卷使用已有并行文件系统挂载**方式：即静态创建的方式，需要先使用已有的并行文件系统创建PV，然后通过PVC在工作负载中挂载存储。

使用流程图

图 4-13 使用并行文件系统流程图



版权所有 © 华为云计算技术有限公司

计费说明

使用已有并行文件系统，计费标准请参考**对象存储服务OBS计费**。

4.4.4.2 并行文件系统卷挂载设置访问密钥（AK/SK）

操作背景

使用Pod挂载并行文件系统卷前您需要先设置访问密钥（AK/SK），IAM用户可以使用自己的访问密钥挂载并行文件系统卷，从而可以对OBS进行访问权限控制（具体请参见**OBS不同权限控制方式的区别**）。

前提条件

您需要通过命令行创建，需要使用ccictl连接到CCI 2.0，详情请参见**ccictl配置指南**。

约束与限制

并行文件系统卷使用访问密钥（AK/SK）时，对应的AK/SK不允许删除或禁用，否则业务容器将无法访问已挂载的并行文件系统。

获取访问密钥

步骤1 登录控制台。

步骤2 鼠标指向界面右上角的登录用户名，在下拉列表中单击“我的凭证”。

步骤3 在左侧导航栏单击“访问密钥”。

步骤4 单击“新增访问密钥”，进入“新增访问密钥”页面。

步骤5 单击“确定”，下载访问密钥。

----结束

使用访问密钥创建 Secret

步骤1 获取访问密钥。

步骤2 对访问密钥进行base64编码（假设上文获取到的ak为“xxx”，sk为“yyy”），例如：在linux系统上可以执行以下命令：

```
echo -n xxx|base64
echo -n yyy|base64
```

记录编码后的AK和SK。

步骤3 新建一个secret的yaml，如secret-obs.yaml。

```
apiVersion: cci/v2
data:
  access.key: WE5WWVhVNU*****
  secret.key: Nnk4emJyZ0*****
kind: Secret
metadata:
  name: secret-obs
  namespace: test-obs-v1
  labels:
    secret.kubernetes.io/used-by: csi
type: cci/secure-opaque
```


版权所有 © 华为云计算技术有限公司

其中：

参数	是否必选	参数类型	描述
access.key	是	String	参数解释：base64编码后的AK。
secret.key	是	String	参数解释：base64编码后的SK。
name	是	String	参数解释：Secret的名称
namespace	是	String	参数解释：Secret的命名空间
secret.kubernetes.io/used-by	是	String	参数解释：csi类型存储使用的Secret标记 约束限制：值必须为csi
type	是	String	参数解释：密钥类型 约束限制：该值必须为cci/secure-opaque，使用该类型，用户输入的数据会自动加密。

步骤4 创建Secret。

```
ccictl create -f secret-obs.yaml
```

----结束

后续步骤

设置访问密钥（AK/SK）后，您可以**通过静态存储卷使用已有并行文件系统**。

4.4.4.3 通过静态存储卷使用已有并行文件系统

本文介绍如何使用已有的并行文件系统静态创建PV和PVC，并在工作负载中实现数据持久化与共享性。

前提条件

- 已设置**4.4.4.2**。
- 您需要通过命令行创建，需要使用ccictl连接到CCI 2.0，详情请参见**ccictl配置指南**。

约束与限制

- 使用并行文件系统时，挂载点不支持修改属组和权限。
- 使用PVC挂载并行文件系统时，负载每挂载一个并行文件系统卷，后端会产生一个常驻进程。当负载使用并行文件系统数过多或大量读写并行文件系统文件时，常驻进程会占用大量内存，为保证负载稳定运行，建议负载使用的并行文件系统卷数量不超过其申请的内存GiB数量，如负载的申请的内存规格为4GiB，则建议其使用的并行文件系统数**不超过**4个。
- 使用并行文件系统，不支持只读挂载，如果需要指定并行文件系统的权限，请参考**并行文件系统权限配置**章节配置并行文件系统权限。

版权所有 © 华为云计算技术有限公司

- 支持多个PV挂载同一个并行文件系统，但有如下限制：
  - 多个不同的PVC/PV使用同一个底层并行文件系统卷时，如果挂载至同一Pod使用，会因为PV的volumeHandle参数值相同导致无法挂载，请避免该使用场景。
  - PV中persistentVolumeReclaimPolicy参数必须设置为Retain，否则可能存在一个PV删除时，级联删除底层卷，其他关联这个底层卷的PV会由于底层存储被删除导致使用出现异常。
  - 重复用底层存储时，数据一致性由您自行维护。建议在应用层做好多读多写的隔离保护，合理规划文件使用时间，避免出现多个客户端写同一个文件的情况，防止产生数据覆盖和丢失。
- CCI 2.0当前支持在西南-贵阳一使用并行文件系统。

通过 ccictl 命令行使用已有对象存储

步骤1 使用ccictl连接CCI2.0。

步骤2 创建PV。

1. 创建pv-obs.yaml文件。

```
apiVersion: cci/v2
kind: PersistentVolume
metadata:
  name: pv-obs                          # PV的名称
spec:
  accessModes:
  - ReadWriteMany                       # 访问模式，并行文件系统必须为ReadWriteMany
  capacity:
    storage: 1Gi                        # 存储容量大小，此处仅为校验需要（不能为空和0），设置的大小不起作用
  csi:
    driver: obs.csi.everest.io          # 挂载依赖的存储驱动
    fsType: obsfs                       # 实例类型
    volumeHandle: <your_file_system_name>  # 并行文件系统的名称
    nodePublishSecretRef:               # 设置并行文件系统的密钥
      name: <your_secret_name>          # Secret的名称
      namespace: <your_namespace>       # Secret的命名空间
  persistentVolumeReclaimPolicy: Retain # 回收策略
  storageClassName: csi-obs             # 存储类名称
  mountOptions: []                      # 挂载参数
```

表 4-18 关键参数说明

参数	是否必选	参数类型	描述
accessModes	是	List	参数解释：存储访问模式 约束限制：并行文件系统必须为ReadWriteMany
driver	是	String	参数解释：挂载依赖的存储驱动 约束限制：并行文件系统必须为obs.csi.everest.io
fsType	是	String	参数解释：存储实例类型 约束限制：支持“obsfs”，即并行文件系统

版权所有 © 华为云计算技术有限公司

参数	是否必选	参数类型	描述
volumeHandle	是	String	参数解释：并行文件系统的名称 约束限制：必须为已创建的并行文件系统的名称
nodePublishSecretRef	是	Object	参数解释：并行文件系统卷挂载支持设置访问密钥（AK/SK），您可以使用AK/SK创建一个Secret，然后挂载到PV。详细说明请参见4.4.4.2 示例如下： nodePublishSecretRef: name: secret-demo namespace: default
mountOptions	否	List	参数解释：挂载参数，具体请参见4.4.4.5
persistentVolumeReclaimPolicy	是	String	参数解释：PV卷回收策略 约束限制：目前只支持Retain回收策略 **Retain**：删除PVC，PV资源与底层存储资源均不会被删除，需要手动删除回收。PVC删除后PV资源状态为“已释放（Released）”，不能直接再次被PVC绑定使用。
storage	是	String	参数解释：存储容量，单位为Gi 约束限制：对于并行文件系统来说，此处仅为校验需要（不能为空和0），设置的大小不起作用，此处设定为固定值1Gi
storageClassName	是	String	参数解释：并行文件系统对应的存储类名称 约束限制：并行文件系统对应的存储类名称为csi-obs

2. 执行以下命令，创建PV。

```
ccictl apply -f pv-obs.yaml
```

步骤3 创建PVC。

1. 创建pvc-obs.yaml文件。

```
apiVersion: cci/v2
kind: PersistentVolumeClaim
metadata:
  name: pvc-obs
  namespace: test-obs-v1
  annotations:
    csi.storage.k8s.io/fstype: obsfs
    csi.storage.k8s.io/node-publish-secret-name: <your_secret_name>          # Secret的名称
    csi.storage.k8s.io/node-publish-secret-namespace: <your_namespace>       # Secret的命名空间
spec:
  accessModes:
  - ReadWriteMany                                                  # 并行文件系统必须为ReadWriteMany
  resources:
    requests:
```


版权所有 © 华为云计算技术有限公司

```
    storage: 1Gi
storageClassName: csi-obs                    # 存储类名称，必须与PV的存储类一致。
volumeName: pv-obs                           # PV的名称
```

表 4-19 关键参数说明

参数	是否必填	参数类型	描述
fsType	是	String	参数解释：存储实例类型 约束限制：支持“obsfs”，即并行文件系统
csi.storage.k8s.io/node-publish-secret-name	是	String	参数解释：PV中指定Secret的名称
csi.storage.k8s.io/node-publish-secret-namespace	是	String	参数解释：PV中指定Secret的命名空间
storage	是	String	参数解释：PVC申请容量，单位为Gi 约束限制： – 对于并行文件系统来说，此处仅为校验需要（不能为空和0），设置的大小不起作用，此处设定为固定值1Gi – 与表4-18中PV设置的容量值相同。
storageClassName	是	String	参数解释：存储类名称 约束限制：必须与表4-18中PV的存储类一致，并行文件系统对应的存储类名称为csi-obs
volumeName	是	String	参数解释：PV的名称 约束限制：必须与表4-18中PV名称一致

2. 执行以下命令，创建PVC。

```
ccictl apply -f pvc-obs.yaml
```

步骤4 创建应用。

1. 创建web-demo.yaml文件，本示例中将并行文件系统挂载至/data路径。

```
apiVersion: cci/v2
kind: Deployment
metadata:
  name: web-demo
  namespace: test-obs-v1
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web-demo
  template:
    metadata:
      labels:
        app: web-demo
    spec:
      containers:
```

 版权所有 © 华为云计算技术有限公司

```
  - name: container-1
    image: nginx:latest
    resources:
      limits:
        cpu: 500m
        memory: 1Gi
      requests:
        cpu: 500m
        memory: 1Gi
    volumeMounts:
    - name: pvc-obs-volume    #卷名称，需与volumes字段中的卷名称对应
      mountPath: /data  #存储卷挂载的位置
  imagePullSecrets:
    - name: imagepull-secret
  volumes:
    - name: pvc-obs-volume    #卷名称，可自定义
      persistentVolumeClaim:
        claimName: pvc-obs    #已创建的PVC名称
```

2. 执行以下命令，创建一个挂载并行文件系统的应用。

```
ccictl apply -f web-demo.yaml
```

工作负载创建成功后，您可以尝试**验证数据持久化及共享性**。

----结束

验证数据持久化及共享性

步骤1 查看部署的应用及文件。

1. 执行以下命令，查看已创建的Pod。

```
ccictl get pod -n test-obs-v1| grep web-demo
```

预期输出如下：

```
web-demo-7864446874-6d4lp   1/1     Running   0          52s
web-demo-7864446874-xx6qh   1/1     Running   0          52s
```

2. 依次执行以下命令，查看Pod的/data路径下的文件。

```
ccictl exec web-demo-7864446874-6d4lp -n test-obs-v1 -- ls /data
ccictl exec web-demo-7864446874-xx6qh -n test-obs-v1 -- ls /data
```

两个Pod均无返回结果，说明/data路径下无文件。

步骤2 执行以下命令，在/data路径下创建static文件。

```
ccictl exec web-demo-7864446874-6d4lp -n test-obs-v1 --  touch /data/static
```

步骤3 执行以下命令，查看/data路径下的文件。

```
ccictl exec web-demo-7864446874-6d4lp -n test-obs-v1 -- ls /data
```

预期输出如下：

```
static
```

步骤4 验证数据持久化

1. 执行以下命令，删除名称为web-demo-7864446874-6d4lp的Pod。

```
ccictl delete pod web-demo-7864446874-6d4lp -n test-obs-v1
```

预期输出如下：

```
pod "web-demo-7864446874-6d4lp" deleted
```

删除后，Deployment控制器会自动重新创建一个副本。

2. 执行以下命令，查看已创建的Pod。

```
ccictl get pod -n test-obs-v1 | grep web-demo
```

预期输出如下，web-demo-7864446874-84slz为新建的Pod：

```
web-demo-7864446874-84slz   1/1     Running   0          110s
web-demo-7864446874-xx6qh   1/1     Running   0          8m47s
```

 版权所有 © 华为云计算技术有限公司

3. 执行以下命令，验证新建的Pod中/data路径下的文件是否更改。
```
ccictl exec web-demo-7864446874-84slz -n test-obs-v1 -- ls /data
```
预期输出如下：
```
static
```
static文件仍然存在，则说明数据可持久化保存。

步骤5 验证数据共享性

1. 执行以下命令，查看已创建的Pod。
```
ccictl get pod -n test-obs-v1 | grep web-demo
```
预期输出如下：
```
web-demo-7864446874-84slz   1/1     Running   0          6m3s
web-demo-7864446874-xx6qh   1/1     Running   0          13m
```
2. 执行以下命令，在任意一个Pod的/data路径下创建share文件。本例中选择名为web-demo-7864446874-84slz的Pod。
```
ccictl exec web-demo-7864446874-84slz -n test-obs-v1 --  touch /data/share
```
并查看该Pod中/data路径下的文件。
```
ccictl exec web-demo-7864446874-84slz -n test-obs-v1 -- ls /data
```
预期输出如下：
```
share
static
```
3. 由于写入share文件的操作未在名为web-demo-7864446874-xx6qh的Pod中执行，在该Pod中查看/data路径下是否存在文件以验证数据共享性。
```
ccictl exec web-demo-7864446874-xx6qh -n test-obs-v1 -- ls /data
```
预期输出如下：
```
share
static
```
如果在任意一个Pod中的/data路径下创建文件，其他Pod下的/data路径下均存在此文件，则说明两个Pod共享一个存储卷。

----结束

4.4.4.4 更新并行文件系统卷访问密钥（AK/SK）后自动生效

在您自建的集群中，如果业务容器依赖于并行文件系统服务来存储和读取数据，每次更改存储卷的访问密钥后，都必须手动重启所有相关的业务容器，以使新密钥生效，这可能会对业务的连续性产生一定影响。

CCI支持更新并行文件系统卷访问密钥后自动生效的功能。当您更新了并行文件系统卷的访问密钥后，系统能够自动地检测到这一变化，并将所有受影响的工作负载配置为新的访问密钥，而无需手动重启工作负载，保证密钥更新期间业务也可以平稳运行。

更新并行文件系统卷访问密钥（AK/SK）

步骤1 登录CCI控制台。

步骤2 在左侧导航栏选择“配置中心”，在右侧选择“密钥(Secret)”页签，单击待修改密钥“操作”列的“YAML编辑”按钮。

步骤3 修改（AK/SK）后，单击“确定”。

版权所有 © 华为云计算技术有限公司

⚠ 注意

请确保更新后的密钥合法有效，且具备访问对应的并行文件系统卷的权限，否则将导致工作负载无法访问已挂载的并行文件系统。

----结束

4.4.4.5 设置并行文件系统挂载参数

本章节主要介绍如何设置并行文件系统的挂载参数。您可以在PV中设置挂载参数，然后通过PVC绑定PV。

并行文件系统挂载参数

CCI 2.0在挂载并行文件系统时默认设置了**表4-20**和**表4-21**的参数，其中**表4-20**中的参数不可取消。除了这些参数外，您还可以设置其他的并行文件系统挂载参数，具体请参见**挂载并行文件系统**。

表 4-20 默认使用且不可取消的挂载参数

参数	是否必选	参数类型	描述
use_ino	是	无需填写参数值	参数解释：使用该选项，由obsfs分配inode编号。读写模式下自动开启。
big_writes	是	无需填写参数值	参数解释：配置后可更改写缓存最大值大小。
nonempty	是	无需填写参数值	参数解释：允许挂载目录非空。
allow_other	是	无需填写参数值	参数解释：允许其他用户访问挂载目录。
no_check_certificate	是	无需填写参数值	参数解释：不校验服务端证书。

版权所有 © 华为云计算技术有限公司

表 4-21 默认使用且可修改的挂载参数

参数	是否必选	参数类型	描述
max_write	否	Int	参数解释：仅配置big_writes的情况下才生效，推荐使用128KB。 默认取值：131072
ssl_verify_hostname	否	Int	参数解释：不根据主机名验证SSL证书。 默认取值：0
max_background	否	Int	参数解释：可配置后台最大等待请求数，并行文件系统自动使用。 默认取值：100
umask	否	Int	参数解释：配置文件权限的掩码。例如，如果umask值为022，而目录最大权限为777，则设置umask后该目录权限为777 - 022 = 755，即rwxr-xr-x。 默认取值：0

在 PV 中设置挂载参数

在PV中设置挂载参数可以通过mountOptions字段实现，如下所示，mountOptions支持挂载的字段请参见**并行文件系统挂载参数**。

步骤1 使用ccictl连接CCI2.0。

步骤2 在PV中设置挂载参数，示例如下：

```
apiVersion: cci/v2
kind: PersistentVolume
metadata:
  name: pv-obs                          # PV的名称
spec:
  accessModes:
  - ReadWriteMany                       # 访问模式，并行文件系统必须为ReadWriteMany
  capacity:
    storage: 1Gi                        # 存储容量大小，此处仅为校验需要（不能为空和0），设置的大小不起作用
  csi:
    driver: obs.csi.everest.io          # 挂载依赖的存储驱动
    fsType: obsfs                       # 实例类型
    volumeHandle: <your_file_system_name>  # 并行文件系统的名称
    nodePublishSecretRef:               # 设置并行文件系统的密钥
      name: <your_secret_name>          # Secret的名称
      namespace: <your_namespace>       # Secret的命名空间
  persistentVolumeReclaimPolicy: Retain # 回收策略
  storageClassName: csi-obs             # 存储类名称
  mountOptions:                         # 挂载参数
  - umask=027
```

步骤3 PV创建后，可以创建PVC关联PV，然后在工作负载的容器中挂载，具体操作步骤请参见**4.4.4.3**。

版权所有 © 华为云计算技术有限公司

步骤4 验证挂载参数是否生效。

本例中将PVC挂载至使用nginx:latest镜像的工作负载，可以登录到运行挂载并行文件系统卷的Pod所在节点，执行以下命令观察进程详情：

```
ps -ef | grep obsfs
```

回显如下所示：

```
root      1470     1  0 Jul03 ?        00:01:23 /bin/obsfs {your_obs_name} /mnt/paas/kubernetes/kubelet/pods/{pod_uid}/volumes/kubernetes.io~csi/{your_pv_name}/mount -o url=https://{endpoint}:443 -o endpoint={region} -o passwd_file=/opt/everest-host-connector/obstmpcred/{pod_uid}/{your_obs_name} -o allow_other -o nonempty -o big_writes -o use_ino -o no_check_certificate -o ssl_verify_hostname=0 -o max_background=100 -o umask=027 -o max_write=131072
```

----结束

4.5 配置中心

4.5.1 使用 ConfigMap

ConfigMap是一种用于存储应用所需配置信息的资源类型。资源创建完成后，可在容器应用中作为文件使用。

创建 ConfigMap

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 单击左侧导航栏的“配置中心”，进入配置中心页面。

步骤3 选择“配置项（ConfigMap）”，在配置中心页面中选择命名空间。

步骤4 在左上角单击“YAML创建”，输入ConfigMap的YAML定义内容，YAML定义可以参考**YAML格式**。

说明

云容器实例支持JSON或YAML格式，且文件大小不得超过1MiB。

步骤5 单击“确定”即可。

----结束

ConfigMap 的使用

配置项创建完成后，可以在创建容器组的过程中以存储卷形式挂载到容器中，如下图所示，将名为system-preset-aeskey的配置项挂载到容器中，存储卷名称为volume1。

 版权所有 © 华为云计算技术有限公司

图 4-14 使用 ConfigMap

ConfigMap 文件格式要求

ConfigMap资源文件支持YAML和JSON两种格式，且数据值大小不得超过1MiB。

- YAML格式

  文件名称为configmap.yaml，配置示例如下：

```
apiVersion: cci/v2
kind: ConfigMap
metadata:
  name: configmap-example
data:
  key1: value1
  key2: value2
```

- JSON格式

  文件名称为configmap.json，配置示例如下：

```
{
    "apiVersion": "cci/v2",
    "kind": "ConfigMap",
    "metadata": {
        "name": "configmap-example"
    },
    "data": {
        "key1": "value1",
        "key2": "value2"
    }
}
```

4.5.2 使用 Secret

Secret是Kubernetes中一种加密存储的资源对象，您可以将认证信息、证书、私钥等保存在密钥中，在容器启动时以环境变量加载到容器中，或以文件方式挂载到容器中。

版权所有 © 华为云计算技术有限公司

说明

建议用户对上传的Secret进行加密处理。

创建 Secret

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 单击左侧导航栏的“配置中心”，进入配置中心页面。

步骤3 选择“密钥（Secret）”，在配置中心页面中选择命名空间。

步骤4 在左上角单击“YAML创建”，输入Secret的YAML定义内容，YAML定义可以参考**yaml格式**。

说明

云容器实例支持JSON或YAML格式，且文件大小不得超过1MiB。

步骤5 单击“确定”即可。

Secret列表中会出现新创建的Secret。

----结束

Secret 的使用

Secret创建完后，可以在创建容器组的过程中以存储卷的形式挂载到容器中。如下图所示，将名为aksk-secret的密钥挂载到容器中，存储卷名称为volume2。

图 4-15 使用 Secret

Secret 文件格式说明

- secret.yaml资源描述文件

版权所有 © 华为云计算技术有限公司

例如现在有一个应用需要获取下面的key-value并加密，可以通过Secret来实现：

key1: value1

key2: value2

定义的Secret文件secret.yaml内容如下。其中Value需要进行Base64编码，Base64编码方法请参见**如何进行Base64编码**。

```
apiVersion: cci/v2
kind: Secret
metadata:
  name: mysecret           # secret的名称
data:
  key1: dmFsdWUx    #需要用Base64编码
  key2: dmFsdWUy    #需要用Base64编码
type: Opaque         # 必须为Opaque
```

- secret.json资源描述文件

  定义的Secret文件secret.json内容如下。

```
{
    "apiVersion": "cci/v2",
    "kind": "Secret",
    "metadata": {
        "name": "mysecret"
    },
    "data": {
        "key1": "dmFsdWUx",
        "key2": "dmFsdWUy"
    },
    "type": "Opaque"
}
```

如何进行 Base64 编码

对字符串进行Base64加密，可以直接使用“echo -n 要编码的内容 | base64”命令即可，示例如下：

```
root@ubuntu:~# echo -n "3306" | base64
MzMwNg==
```

4.6 镜像

4.6.1 镜像快照

4.6.1.1 镜像快照概述

用户通过CCI镜像快照功能，能够实现从SWR镜像仓库、开源镜像仓库、自建镜像仓库拉取镜像制作成相应的镜像快照。在创建负载过程中，使用预先创建的镜像快照，可以跳过镜像拉取动作，提升负载的启动速度。

用户在创建Pod时使用镜像快照（ ImageSnapshot ），可以避免镜像下载，减少Pod启动时间。本文将介绍镜像快照的基本功能，创建和使用方式等。

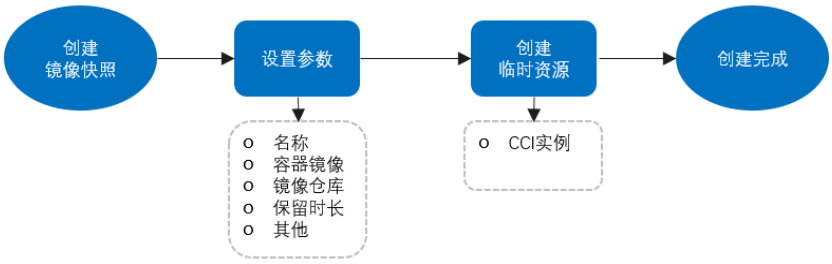
约束与限制

- CCI 2.0当前仅支持在华东-上海一和华北-北京四局点使用镜像快照。
- 单个镜像快照最多包含10个镜像。

版权所有 © 华为云计算技术有限公司

- 支持私有镜像仓库，但需要提供私有镜像仓库的访问凭证，包括地址和用于认证的信息，如~/.docker/config.json中的auth信息。
- 如果镜像需要通过公网拉取，则需要事先配置公网访问或者指定公网配置。
- 如果镜像快照中只有部分镜像与Pod中镜像相符，则不相符的镜像仍需通过下载拉取。
- CCI实例运行中镜像信息发生变动，且新镜像与匹配的镜像快照中镜像无相符的场景，则新镜像仍需通过下载拉取。
- 镜像快照默认使用 0.5核1G 规格的CCI实例进行制作，收取费用为制作过程中产生的费用。

镜像快照创建过程如下图所示：



使用方式

使用镜像快照创建Pod，支持以下两种方式：

- 自动匹配

自动匹配将从用户创建的所有可用镜像快照中选择最优的镜像。按以下顺序进行匹配：

a. 镜像匹配度：优先选择匹配度高的镜像快照，匹配度指的是Pod和镜像快照两者在镜像上的匹配情况。

b. 创建时间：优先选择创建时间更新的镜像快照。

- 明确指定

明确指定使用的镜像快照。

4.6.1.2 创建镜像快照

操作场景

本章介绍了如何创建镜像快照。要了解镜像快照的工作原理，请参阅4.6.1.1。

操作步骤

运行容器需要先拉取指定的容器镜像，但因网络和容器镜像大小等因素，镜像拉取耗时往往成了Pod启动的主要耗时。通过镜像快照功能，可以事先将要使用的镜像制作成快照，基于该快照创建Pod，从而避免镜像下载，提升Pod的启动速度。

以下示例创建一个名为my-imagesnapshot的镜像快照。

版权所有 © 华为云计算技术有限公司

```
apiVersion: cci/v2
kind: ImageSnapshot
metadata:
  name: 'my-imagesnapshot'
spec:
  buildingConfig:
    namespace: test-namespace
    eipID: xxxxxxxx
  imageSnapshotSize: 30
  ttlDaysAfterCreated: 7
  images:
    - image: 'nginx:stable-alpine-perl'
  registries:
    - imagePullSecret: imagepull-secret
      server: xxxxx.myhuaweicloud.com
      plainHTTP: true
```

表 4-22 参数说明

字段	类型	必填	示例值	描述
.metadata.name	string	是	my-imagesnapshot	镜像快照名称。
.spec.images.image	string	是	nginx:latest	用于制作镜像快照的镜像。
.spec.registries.server	string	是	serverA.com	不带 http:// 或 https:// 前缀的镜像仓库地址。
.spec.registries.imagePullSecret	string	否	imagepull-secret	访问镜像仓库使用的凭证Secret。
.spec.registries.plainHTTP	boolean	否	true	如果是使用HTTP协议的自建镜像仓库地址，需要设置为true，否则会因协议不同而导致镜像拉取失败。默认值为false。
.spec.registries.insecureSkipVerify	boolean	否	true	如果是使用自签发证书的自建镜像仓库地址，需要设置为true来跳过证书认证，否则会因证书认证失败而导致镜像拉取失败。默认值为false。
.spec.buildingConfig.namespace	string	是	my-namespace-a	用户命名空间。镜像快照制过程需要在用户命名空间创建CCI实例。
.spec.buildingConfig.eipID	string	否	3cxxxxe0-xxxx-xxxx-xxxx-8xxxxf3xxxx4	用于访问拉取公网镜像的EIP

版权所有 © 华为云计算技术有限公司

字段	类型	必填	示例值	描述
.spec.buildingConfig.autoCreateEIP	boolean	否	true	在制作镜像快照时，是否自动为创建镜像快照实例创建弹性公网IP，当指定了eipId的情况下，autoCreateEIP配置将被忽略。当autoCreateEIP为true时，需通过autoCreateEIPAttribute指定弹性公网IP的配置。
.spec.buildingConfig.autoCreateEIPAttribute.bandwidthChargeMode	string	否	bandwidth	按流量计费还是按带宽计费。 取值范围：bandwidth，traffic，不填或者为空时默认是bandwidth。其中IPv6国外默认是bandwidth，国内默认是traffic。
.spec.buildingConfig.autoCreateEIPAttribute.bandwidthSize	int	否	1000	带宽大小。 取值范围：默认1Mbit/s~2000Mbit/s
.spec.buildingConfig.autoCreateEIPAttribute.type	string	否	5_bgp	EIP的类型。 取值范围：5_bgp（全动态BGP），5_sbgp（静态BGP），5_youxuanbgp（优选BGP）。
.spec.buildingConfig.autoCreateEIPAttribute.ipVersion	int	否	4	弹性公网IP的版本。 取值范围：4或者6。 ● 4表示IPv4，不填或空字符串时，默认创建IPv4。 ● 6表示IPv6，开启NAT64能力。
.spec.buildingConfig.TimeoutMinutes	int	否	1440	镜像快照制作完成的超时时间，单位分钟。 取值范围：[30, 10080] 的整数，表示30分钟~1周。默认值为1440，表示1天。

版权所有 © 华为云计算技术有限公司

字段	类型	必填	示例值	描述
.spec.ttlDaysAfterCreated	integer	否	10	镜像快照保留时间，单位为天，过期将会被清理。默认值为0，即永不过期。 每当快照被负载/容器组资源使用时，其过期时间将被重置为当前时间加上快照保留时间。 **说明** 快照过期后，仍会占用配额，需定期审核过期镜像快照后删除。
.spec.imageSnapshotSize	integer	否	20	镜像快照大小，单位为GiB，默认值为20。

4.6.1.3 使用镜像快照

使用镜像快照创建Pod，支持以下两种方式：

- 自动匹配

  自动匹配将从所有用户创建的可用的镜像快照中选择最优的镜像。按以下顺序进行匹配：

  a. 镜像匹配度：优先选择匹配度高的镜像快照，匹配度指的是Pod和镜像快照两者在镜像上的匹配情况。

  b. 创建时间：优先选择创建时间更新的镜像快照。

- 明确指定

  明确指定使用的镜像快照。

说明

如果同时使用明确指定和自动匹配两种镜像快照方式创建Pod时，如果两者冲突，会返回400报错。

自动匹配

创建CCI实例时，可以通过注解中添加以下键值来开启自动匹配镜像快照。

名称	示例值	描述
cci.io/image-snapshot-auto-match	"true"	设置是否开启自动匹配镜像快照。

版权所有 © 华为云计算技术有限公司

名称	示例值	描述
cci.io/image-snapshot-usage-strategy	"size"	设置镜像快照自动匹配策略，默认为“size”。 ● "size"：选择匹配镜像规模之和最大的镜像快照 ● "quantity"：选择匹配镜像数量最多的镜像快照

以创建Deployment为例：

```
apiVersion: cci/v2
kind: Deployment
metadata:
  name: deployment-test
  namespace: ns-test
spec:
  replicas: 1
  selector:
    matchLabels:
      app: redis
  template:
    metadata:
      labels:
        app: redis
      annotations:
        cci.io/image-snapshot-auto-match: "true"
    spec:
      containers:
        - image: redis
          name: container-0
          resources:
            limits:
              cpu: 500m
              memory: 1024Mi
            requests:
              cpu: 500m
              memory: 1024Mi
      imagePullSecrets:
        - name: imagepull-secret
```

明确指定

创建CCI实例时，可以通过注解中添加以下键值来明确指定镜像快照和拦截策略。

名称	示例值	描述
cci.io/image-snapshot-specified-name	"my-imagesnapshot"	指定的镜像快照名称。
cci.io/image-snapshot-reject-if-not-available	"true"	如果指定的镜像快照不存在或状态不可用，默认会从镜像仓库下载镜像。如果需要拦截实例创建，则需配置此注解为 "true"。

版权所有 © 华为云计算技术有限公司

以创建Deployment为例：

```
apiVersion: cci/v2
kind: Deployment
metadata:
  name: deployment-test
  namespace: ns-test
spec:
  replicas: 1
  selector:
    matchLabels:
      app: redis
  template:
    metadata:
      labels:
        app: redis
      annotations:
        cci.io/image-snapshot-specified-name: "my-imagesnapshot"
        cci.io/image-snapshot-reject-if-not-available: "true"
    spec:
      containers:
        - image: redis
          name: container-0
          resources:
            limits:
              cpu: 500m
              memory: 1024Mi
            requests:
              cpu: 500m
              memory: 1024Mi
      imagePullSecrets:
        - name: imagepull-secret
```

4.6.1.4 管理镜像快照

操作场景

本章介绍了如何通过控制台对镜像快照执行查询、删除操作。要了解镜像快照的工作原理，请参阅**4.6.1.1**。

查看镜像快照

当您的镜像快照创建完成后，可以通过如下步骤查看详情。

步骤1 登录**云容器实例 CCI2.0**控制台。

步骤2 左侧导航栏中选择“镜像快照”，进入镜像快照列表页。

步骤3 查看镜像快照的名称、状态、快照剩余有效期、镜像快照大小等信息。查看事件详细信息。

当镜像快照状态为可用时，可以使用该镜像快照。

当镜像快照状态为异常时，您可单击“事件”查看事件详细信息。

----结束

删除镜像快照

当您的镜像快照不再使用时，可以通过如下步骤执行删除操作。

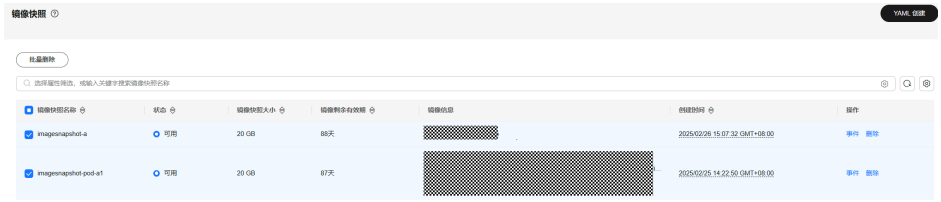
步骤1 登录**云容器实例 CCI2.0**控制台。

版权所有 © 华为云计算技术有限公司

步骤2 左侧导航栏中选择“镜像快照”，进入镜像快照列表页。

步骤3 勾选待删除的镜像快照，单击左上方“批量删除”按钮。

或直接单击镜像快照操作列下的“删除”。



----结束

4.6.2 拉取自建镜像仓库的镜像

拉取自建镜像仓库中的镜像时，可能会碰到因协议不同或者证书认证失败而导致镜像拉取失败的问题。本文介绍在自建镜像仓库采用HTTP协议和使用自签发证书的情况下，如何拉取自建镜像仓库中的镜像来创建CCI Pod。

配置说明

表 4-23 配置说明

annotation	示例值	配置说明
cci.io/http-registries	"harbor.***.com,192.168.XX.XX:5000,100.95.XX.XX,http://harbor.***.com"	拉取采用HTTP协议的自建镜像仓库中的镜像时，需配置该Annotation，使用逗号分隔多个地址。允许存在http://前缀，端口，相对路径，允许使用vpc ip地址、域名、公网ip地址，不允许填空地址，最多填入10个地址。
cci.io/insecure-registries	"harbor.***.com,192.168.XX.XX:5000,100.95.XX.XX,https://harbor.***.com"	拉取使用自签发证书的自建镜像仓库中的镜像时，需配置该Annotation来跳过证书认证，使用逗号分隔多个地址。允许存在https://前缀，端口，相对路径，允许使用vpc ip地址、域名、公网ip地址，不允许填空地址，最多填入10个地址。

版权所有 © 华为云计算技术有限公司

说明

- 如果镜像仓库地址有端口号，则需要带上端口号，例如：镜像地址为192.168.XX.XX:5000/nginx:latest，则cci.io/http-registries可设置为"192.168.XX.XX:5000"。
- 使用HTTP协议存在数据传输不加密、易受中间人攻击和缺乏身份验证等风险，可能会导致数据泄露、业务受损等，推荐使用https协议拉取镜像。

配置示例一：自建镜像仓库采用 HTTP 协议

- 创建2U4G Deployment

```
apiVersion: cci/v2
kind: Deployment
metadata:
  labels:
    app: http
  name: http
spec:
  replicas: 1
  selector:
    matchLabels:
      app: http
  template:
    metadata:
      labels:
        app: http
      annotations:
        resource.cci.io/pod-size-specs: 2.00_4.0
        cci.io/http-registries: 192.168.XX.XX
    spec:
      containers:
        - image: 192.168.XX.XX/harbor/nginx:latest
          name: container-0
      imagePullSecrets:
        - name: harbor-secret-new
```

- 创建2U4G Pod

```
apiVersion: cci/v2
kind: Pod
metadata:
  annotations:
    resource.cci.io/pod-size-specs: 2.00_4.0
    cci.io/http-registries: 192.168.XX.XX
  name: http
spec:
  containers:
    - image: '192.168.XX.XX/harbor/nginx:latest'
      imagePullPolicy: IfNotPresent
      name: container-1
  imagePullSecrets:
    - name: harbor-secret
```

配置示例二：自建镜像仓库使用自签发证书

- 创建2U4G Deployment

```
apiVersion: cci/v2
kind: Deployment
metadata:
  labels:
    app: insecure
  name: insecure
spec:
  replicas: 1
  selector:
    matchLabels:
      app: insecure
  template:
```


版权所有 © 华为云计算技术有限公司

```
  metadata:
    labels:
      app: insecure
    annotations:
      resource.cci.io/pod-size-specs: 2.00_4.0
      cci.io/insecure-registries: 192.168.XX.XX
  spec:
    containers:
      - image: 192.168.XX.XX/harbor/nginx:latest
        name: container-0
    imagePullSecrets:
      - name: harbor-secret-new
```

- 创建2U4G Pod

```
apiVersion: cci/v2
kind: Pod
metadata:
  annotations:
    resource.cci.io/pod-size-specs: 2.00_4.0
    cci.io/insecure-registries: 192.168.XX.XX
  name: insecure
spec:
  containers:
    - image: '192.168.XX.XX/harbor/nginx:latest'
      imagePullPolicy: IfNotPresent
      name: container-1
  imagePullSecrets:
    - name: harbor-secret
```

4.7 监控管理

使用云服务Prometheus监控，可以监控CCI 2.0云服务的多种指标。详情请参见**AOM Prometheus监控概述**。

约束与限制

一个企业项目下仅可以创建一个云服务类型的Prometheus实例。

CCI 2.0当前仅支持在华南-广州和华东-上海一局点使用监控管理。

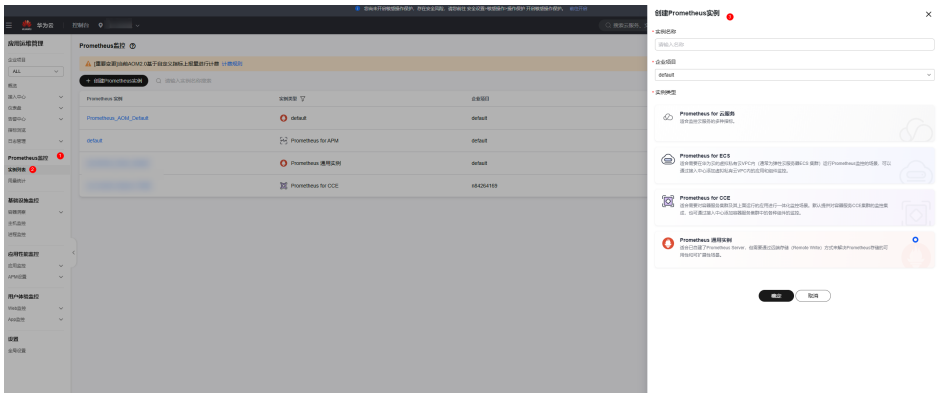
步骤一 创建 Prometheus 实例

1. 登录华为云AOM 2.0控制台。
2. 在左侧导航栏选择“Prometheus监控 -> 实例列表”，单击“创建Prometheus实例”。
3. 设置实例名称、企业项目和实例类型信息。

说明

如果已使用“CCE 突发弹性引擎 (对接 CCI)”插件或需要使用“CCE 突发弹性引擎 (对接 CCI)”插件，Prometheus实例同时需要给CCE集群使用，实例类型需选Prometheus for CCE。

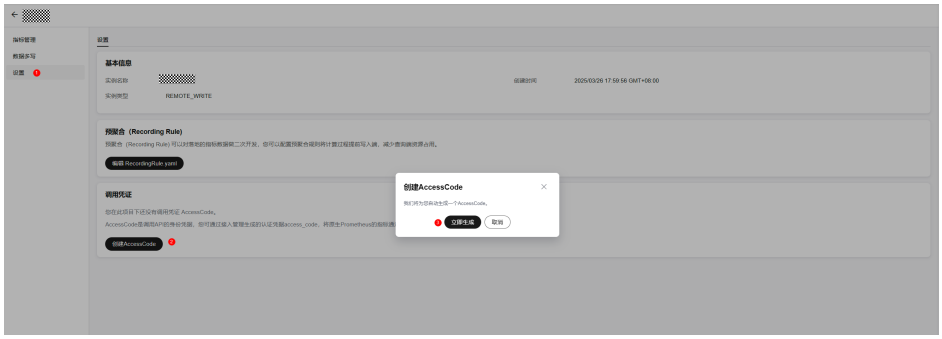
版权所有 © 华为云计算技术有限公司



4. 完成后单击“确定”。

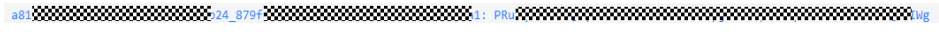
步骤二 获取密钥

1. 单击所创建的Prometheus实例名称，进入详情页。
2. 单击“设置”，在调用凭证栏单击“创建AccessCode”。
3. 单击“立即生成”。



4. 在服务地址栏获取<project_id>、<aom_id>、<aom_secret>。

5. 将以上3个信息拼接为字符串，格式为：<project_id>_<aom_id>:<aom_secret>。



⚠ 注意

冒号后需包含一个空格。

版权所有 © 华为云计算技术有限公司

6. 将其生成base64编码。

   YTgxZ[redacted]XU3SVdn

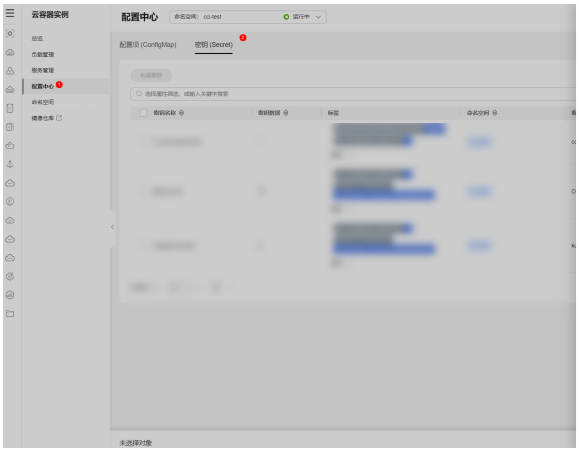
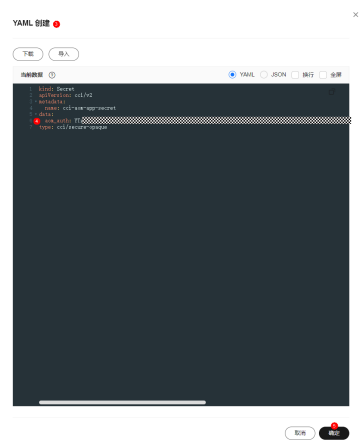
   您可参考以下shell命令生成base64编码：

   ```
   project_id=<project_id>
   aom_id=<aom_id>
   aom_secret=<aom_secret>
   echo -n "${project_id}_${aom_id}: ${aom_secret}" |base64 -w 0
   ```

7. 将生成的base64编码字符串填充到以下模板中，替换{AOMAuthBase64}，并复制此代码。

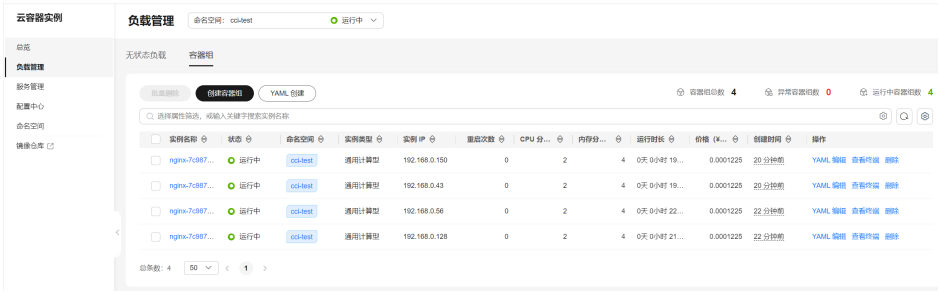
   ```
   kind: Secret
   apiVersion: cci/v2
   metadata:
     name: cci-aom-app-secret
   data:
     aom_auth: {AOMAuthBase64}
   type: cci/secure-opaque
   ```

8. 登录**云容器实例 CCI2.0**控制台。
9. 在左侧导航栏单击“配置中心”。
10. 在配置中心页面选择“密钥（Secret）”。
11. 单击“YAML创建”，将代码替换到CCI界面创建AOM APP Secret。

步骤三 创建容器组

1. 登录**云容器实例 CCI2.0**控制台。
2. 在左侧导航栏单击“负载管理”，选择“容器组”页签。
3. 单击“创建容器组”，并填写相关信息，可参考**容器组操作指导**。



4. 完成后单击“立即创建”。

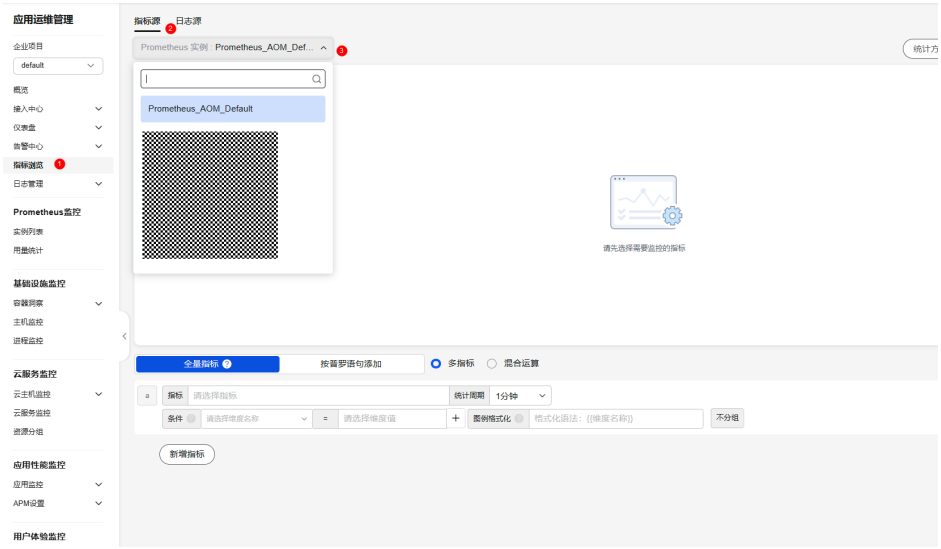
版权所有 © 华为云计算技术有限公司

步骤四 监控容器组

有两种查询方式可监控容器组。分别为“全量指标”和“按普罗语句添加”。

方式一 全量指标

1. 登录华为云AOM 2.0控制台。
2. 在左侧导航栏选择“指标浏览”。
3. 在指标源页面选择**步骤一 创建Prometheus实例**。



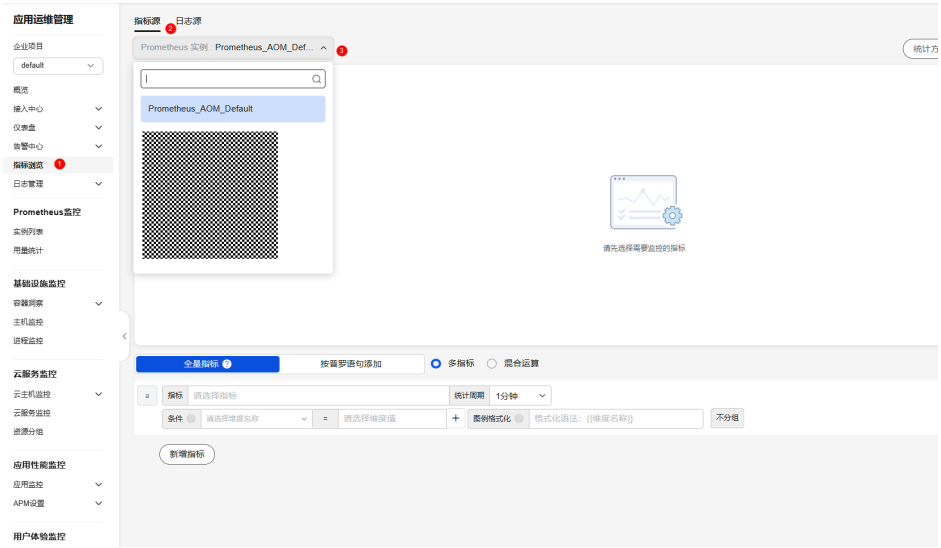
4. 在“全量指标”页签，单击“指标”。
5. 勾选“CCI”,选择相关常用指标。

6. 即可查看指标监控相关数据。

版权所有 © 华为云计算技术有限公司

方式二 按普罗语句添加

1. 登录华为云AOM 2.0控制台。
2. 在左侧导航栏选择“指标浏览”。
3. 在指标源页面选择**步骤一 创建Prometheus实例**。



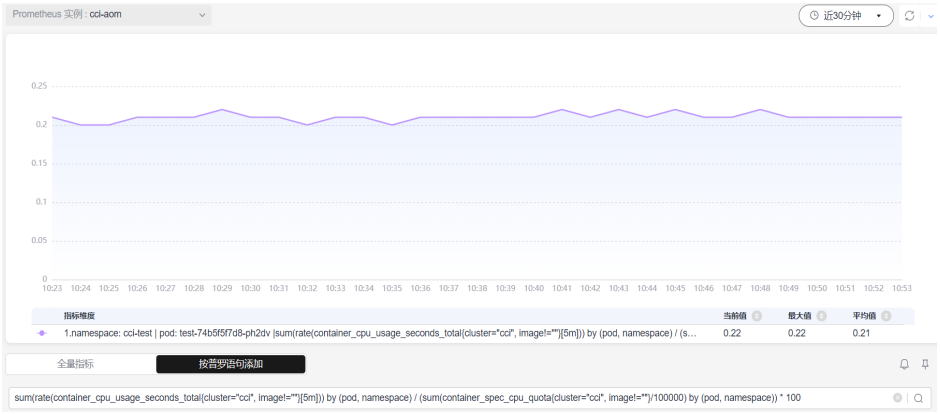
4. 在“按普罗语句添加”页签，在搜索框输入普罗语句后单击 即可。

常用普罗语句示例如下：

a. 查询所有Pod CPU使用率。

```
sum(rate(container_cpu_usage_seconds_total{cluster="cci", image!=""}[5m])) by (pod, namespace) / (sum(container_spec_cpu_quota{cluster="cci", image!=""}/100000) by (pod, namespace)) * 100
```

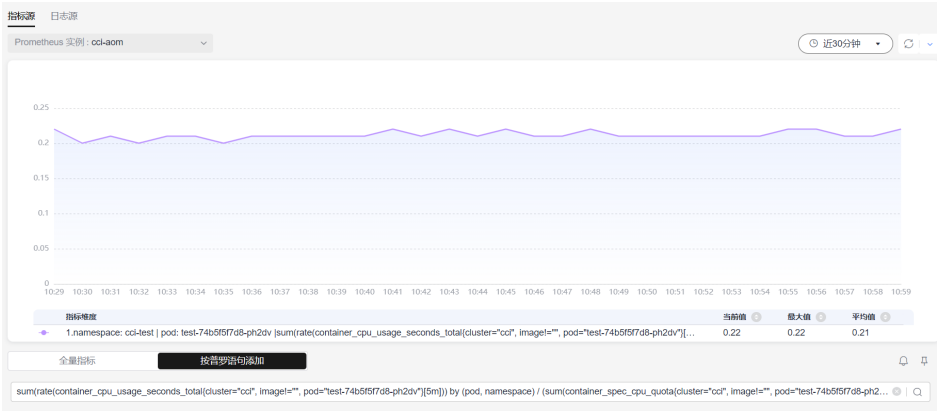

版权所有 © 华为云计算技术有限公司



b. 查询单个Pod CPU使用率。

将如下promQL中的<pod_name>替换为pod名。

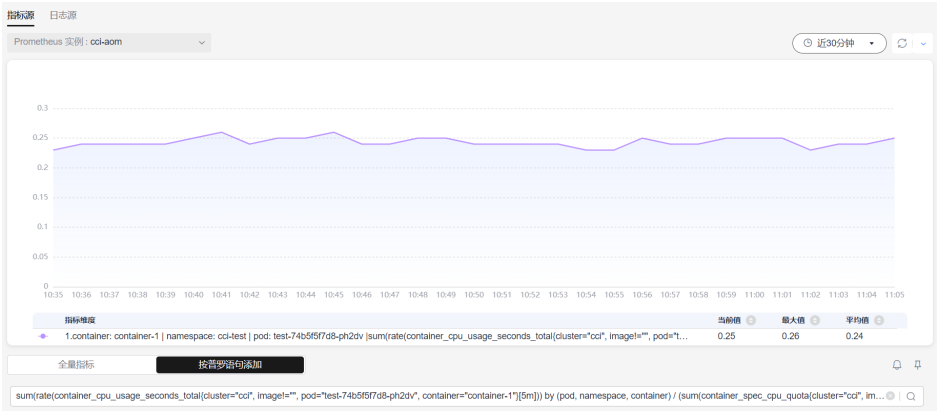
```
sum(rate(container_cpu_usage_seconds_total{cluster="cci", image!="", pod="<pod_name>"}[5m])) by (pod, namespace) / (sum(container_spec_cpu_quota{cluster="cci", image!="", pod="<pod_name>"}/100000) by (pod, namespace)) * 100
```



c. 查询单个Container CPU使用率。

将如下promQL中的<pod_name>替换为pod名，<container_name>替换为container名。

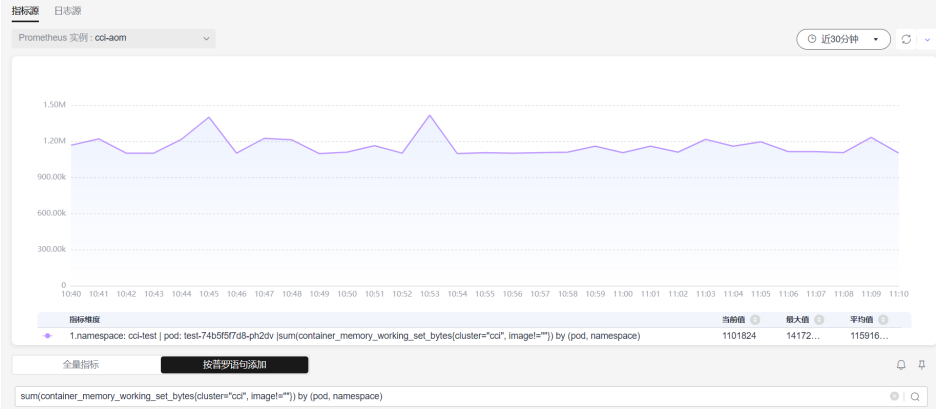
```
sum(rate(container_cpu_usage_seconds_total{cluster="cci", image!="", pod="<pod_name>", container="<container_name>"}[5m])) by (pod, namespace, container) / (sum(container_spec_cpu_quota{cluster="cci", image!="", pod="<pod_name>", container="<container_name>"}/100000) by (pod, namespace, container)) * 100
```



d. 查询所有Pod内存使用量。

```
sum(container_memory_working_set_bytes{cluster="cci", image!=""}) by (pod, namespace)
```

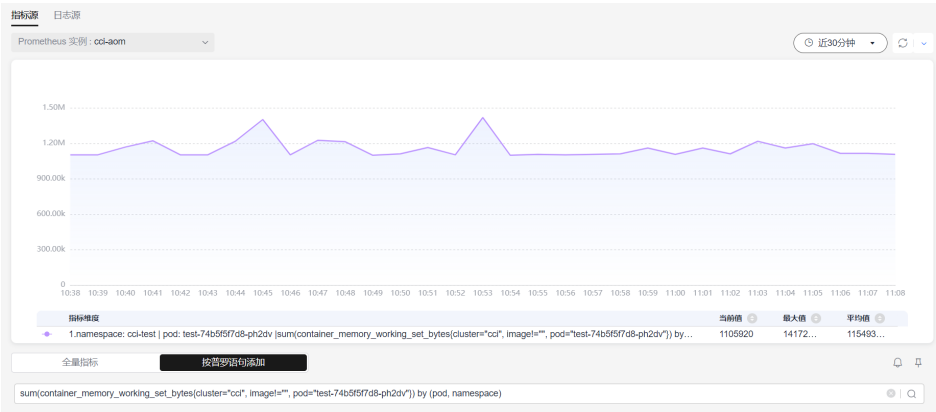

版权所有 © 华为云计算技术有限公司



e. 查询单个Pod内存使用量。

将如下promQL中的<pod_name>替换为pod名。

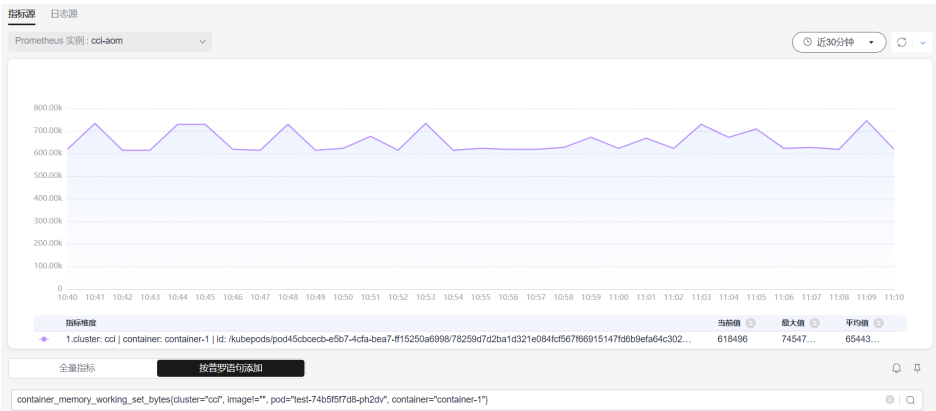
```
sum(container_memory_working_set_bytes{cluster="cci", image!="", pod="<pod_name>"}) by (pod, namespace)
```



f. 查询单个Container内存使用量。

将如下promQL中的<pod_name>替换为pod名，<container_name>替换为container名。

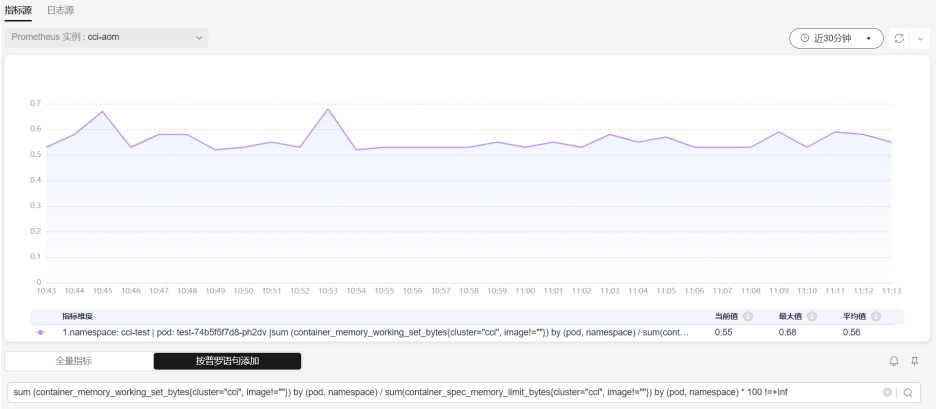
```
container_memory_working_set_bytes{cluster="cci", image!="", pod="<pod_name>", container="<container_name>"}
```



g. 查询所有Pod内存使用率。

```
sum (container_memory_working_set_bytes{cluster="cci", image!=""}) by (pod, namespace) / sum(container_spec_memory_limit_bytes{cluster="cci", image!=""}) by (pod, namespace) * 100 != +Inf
```

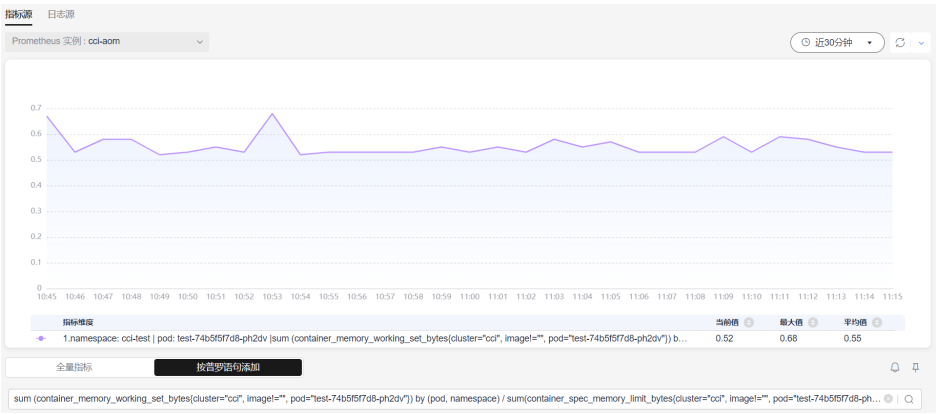

版权所有 © 华为云计算技术有限公司



h. 查询单个Pod内存使用率。

将如下promQL中的<pod_name>替换为pod名。

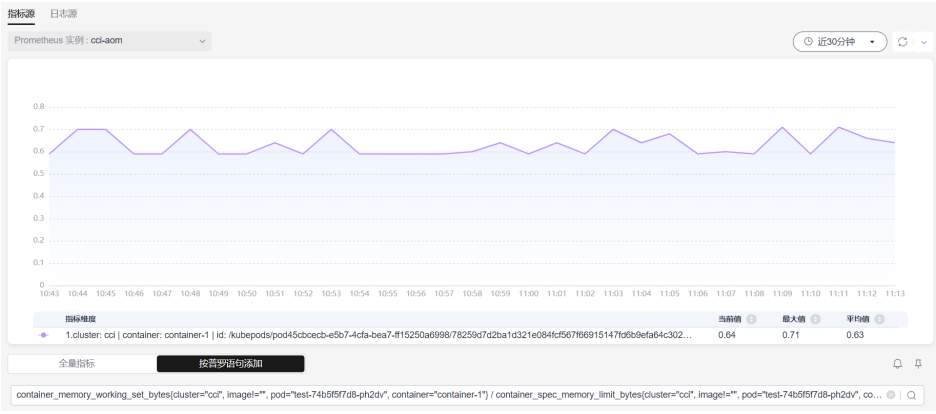
```
sum (container_memory_working_set_bytes{cluster="cci", image!="", pod="<pod_name>"}) by (pod, namespace) / sum(container_spec_memory_limit_bytes{cluster="cci", image!="", pod="<pod_name>"}) by (pod, namespace) * 100 !=+Inf
```



i. 查询单个Container内存使用率。

将如下promQL中的<pod_name>替换为pod名，<container_name>替换为container名。

```
container_memory_working_set_bytes{cluster="cci", image!="", pod="<pod_name>", container="<container_name>"} / container_spec_memory_limit_bytes{cluster="cci", image!="", pod="<pod_name>", container="<container_name>"} * 100 !=+Inf
```



仪表盘等更多监控运维方法请参考**创建仪表盘**。

版权所有 © 华为云计算技术有限公司

4.8 事件管理

4.8.1 开启 CCI 2.0 事件上报功能

CCI 2.0使用LTS服务为用户采集事件，并允许用户在LTS侧配置告警，及时感知负载异常状态。

在使用CCI 2.0过程中，会针对各种资源产生事件，包含Deployment、Service、HPA、Network、Pod等几种资源。

约束与限制

CCI 2.0当前仅支持在西南-贵阳一、华北-北京四、华东-上海一局点使用事件上报功能。

事件格式

CCI 2.0产生的事件格式遵从**cloudevent**格式标准，事件内容以json类型存储，示例及各字段解释如下：

```
{
   "data": {
      "metadata": {
         "name": "service-test-event",
         "namespace": "test",
         "uid": "1******3-2**1-4**a-9**8-2******0",
         "resourceVersion": "61788694",
         "creationTimestamp": "2024-11-30T07: 55: 40Z",
         "annotations": {
            "tenant.cci.io/tenant-id": "a81*******24"
         }
      },
      "involvedObject": {
         "kind": "Service",
         "namespace": "test",
         "name": "service-test",
         "uid": "f******3-d**b-4**e-a**c-9f******e3",
         "apiVersion": "cci/v2",
         "resourceVersion": "7669479"
      },
      "reason": "EnsuringLoadBalancer",
      "message": "Ensuring load balancer",
      "source": {
         "component": "service-controller"
      },
      "firstTimestamp": "2024-11-30T07: 52: 45Z",
      "lastTimestamp": "2024-12-03T02: 30: 02Z",
      "count": 4,
      "type": "Normal",
      "eventTime": null,
      "reportingComponent": "",
      "reportingInstance": ""
   },
   "deprecatedeventreason": "EnsuringLoadBalancer",
   "eventclass": "Normal",
```


版权所有 © 华为云计算技术有限公司

```
  "datacontenttype": "application/json",
  "time": "2024-04-05T17:31:00Z",
  "id": "b*****9-5737-4**1-8**a-3*******bc",
  "specversion": "1.0",
  "source": "cci:a8*********48b24:service:test:service-test:f******3-d**b-4**e-a**c-4******3",
  "type": "cloudservice.cci.service.publish.processing"
}
```


表 4-24 字段列表

字段	字段类型	描述
type	string	标识资源对象当前操作状态，格式为cloudservice.cci.${资源类型}.${资源操作}.${操作结果}.${附加信息}，其中附加信息片段仅存在于复杂场景
source	string	标识资源对象上下文，格式为cci:${projectid}:${资源类型}:${命名空间}:${资源名称}:${资源唯一id}
specversion	string	cloudevent版本标准，当前为1.0
id	string	事件对象的唯一标识，uuid随机生成
time	Timestamp	事件发生时间
eventclass	string	事件级别，仅包含Normal和Warning两类
deprecatedeventreason	string	触发事件的reason字段，复用k8s event reason(https://kubernetes.io/zh-cn/docs/reference/kubernetes-api/cluster-resources/event-v1/)字段内容，逐步弃用
datacontenttype	string	“data”字段存储类型，当前为application/json格式
data	k8s event object	event详细内容，可按照“datacontenttype”字段定义的存储类型解析

开启 CCI 2.0 事件上报功能

CCI 2.0采用按需开放的形式开启事件上报功能，需要用户主动调用CCI 2.0 API进行开启，功能默认关闭。

须知

- 新客户使用该API前需要先在CCI服务授权委托。
- LTS创建日志组日志流免费，并每月赠送每个账号一定量存储额度，超过部分将会产生费用详情请参见**计费说明**。您可在LTS服务“配置中心”，通过“超额继续采集日志”开关控制日志超额采集情况，详情请参见**设置LTS日志采集配额和使用量预警**。开启后表示当日超过免费赠送的额度时，继续采集日志，超出的部分将按需收费。如果您关闭了此开关，当日志超过每月免费赠送的额度时，将暂停采集日志，会导致后续CCI服务上报的事件不可见。

版权所有 © 华为云计算技术有限公司

表 4-25 API URI

API URI	描述
GET /v1/observabilityconfiguration	查询当前可观测性配置
PUT /v1/observabilityconfiguration	修改可观测性配置

功能开启后，CCI 2.0服务会在LTS服务侧创建一组日志组日志流，日志组命名格式为cci-event-${projectid}，日志组命名格式为event-${projectid}，存储时间默认为1天，请勿删除。

如需修改存储时间，您可前往云日志控制台，在“日志管理”页面日志组列表中选择相关日志组，在右侧操作列单击“修改”LTS侧日志组日志流中进行配置。

图 4-16 修改日志组

您可在“日志管理”页面日志组列表中单击相关日志组，在日志流中查看CCI 2.0产生的各种资源对象事件。

4.8.2 通过 LTS 配置告警规则

事件上报功能开关开启后，资源事件将上报到LTS日志组日志流中，可在LTS侧配置告警规则，在事件产生并上报到LTS侧后，会自动通知在SMN所配置的联系人。

版权所有 © 华为云计算技术有限公司

配置结构化解析

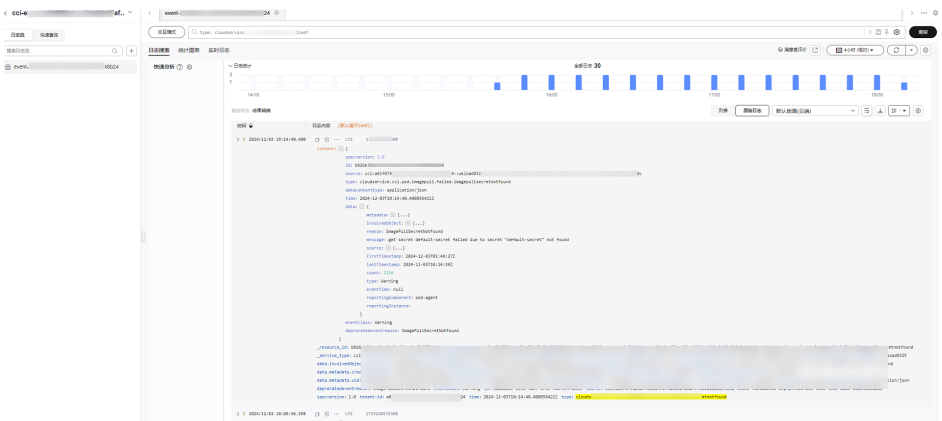
1. 登录云日志控制台，在“日志管理”页面日志组列表中单击相关日志组，进入日志组详情页。
2. 单击右上角 设置按钮，进入设置页面。
3. 选择“云端结构化解析 > JSON”，在“步骤一 选择示例日志”中输入LTS日志文本内容。
4. 单击“智能提取 > 保存”，配置成功后，后续上报的事件将按照结构化解析规则进行解析。

图 4-17 配置结构化解析

创建告警规则

1. 登录云日志控制台，在“日志管理”页面日志组列表中单击相关日志组，进入日志组详情页。
2. 根据日志搜索页面提供的相关标签，选择需要使用的过滤标签，配置过滤条件，确保能搜索目标事件。您也可参考**配置日志告警规则**帮助文档进行操作。

图 4-18 选择过滤标签



版权所有 © 华为云计算技术有限公司

⚠ 注意

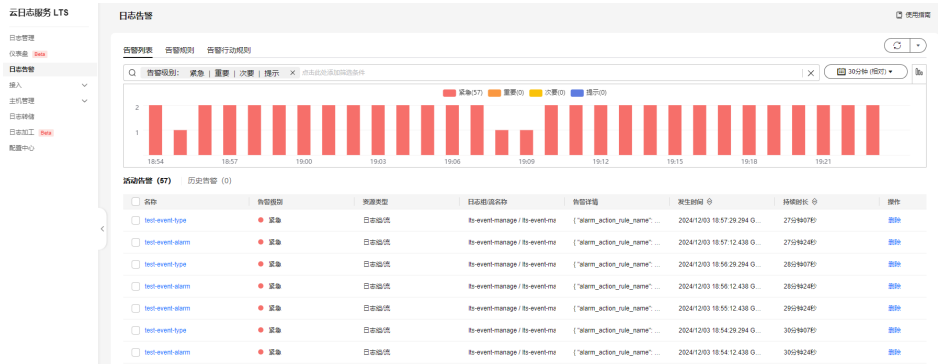
根据type字段进行过滤时，建议您使用模糊匹配方式，避免错过需要关注的事件。

3. 登录云日志控制台，在“日志告警”页面，选择“告警规则”。
4. 单击“创建”。
5. 填写基本信息并选择日志组日志流，在“关键词”输入过滤条件。
6. 开启告警行动规则，单击创建告警行动规则并选择。

图 4-19 高级设置

7. 单击“确定”即可。
8. 特定事件上报后触发告警，您可在云日志控制台，在“日志告警 > 告警列表”中查看

图 4-20 特定事件上报后触发告警



版权所有 © 华为云计算技术有限公司

4.8.3 事件列表

CCI 2.0在多种资源操作过程中生成事件，包含Deployment、Replicaset、Service、HPA、Network、Pod几种资源，每种资源的事件分为以下几类，以type字段进行划分，每种资源分别对应着几种固定的操作类型和操作结果，附加信息为备注信息，仅在某些复杂场景中提供。

表 4-26 事件列表

资源对象	操作类型	操作结果	附加信息	描述信息	类比k8s事件reason举例
Deployment	scale	succeeded	-	deployment扩缩容成功	ScalingReplicaSet等
	scale	failed	-	deployment扩缩容失败	ReplicaSetCreateError等
	delete	processing	-	deployment删除中	DeleteDeployment等
Replicaset	scale	succeeded	-	replicaset扩缩容成功	SuccessfulCreate、SuccessfulDelete等
	scale	failed	-	replicaset扩缩容失败	FailedCreate、FailedDelete等
Service	publish	failed	-	Service服务发现失败	SyncLoadBalancerFailed等
	publish	succeeded	-	Service服务发现成功	EnsuredLoadBalancer等
	publish	processing	-	Service服务发现中	EnsuringLoadBalancer等
	delete	failed	-	Service服务删除失败	DeleteLoadBalancerFailed等
	delete	succeeded	-	Service服务删除成功	DeletedLoadBalancer等
	delete	processing	-	Service服务删除中	DeletingLoadBalancer等
Horizontalpodautoscaler	rescale	failed	-	重扩缩容失败	FailedComputeMetricsReplicas、FailedRescale等
	rescale	failed	computemetricsfailed	因metrics计算失败导致的重扩缩容失败	FailedComputeMetricsReplicas等

版权所有 © 华为云计算技术有限公司

资源对象	操作类型	操作结果	附加信息	描述信息	类比k8s事件reason举例
	rescale	succeeded	-	重扩缩容成功	SuccessfulRescale等
	metricscrape	failed	-	指标获取失败	InvalidMetricSourceType、FailedGetResourceMetric等
	metricscrape	failed	invalidmetricsourcetype	因非法metrics类型导致指标获取失败	InvalidMetricSourceType等
	scaleobjectparse	failed	invalidscaleobject	因非法对象导致扩缩容对象解析失败	InvalidSelector、SelectorRequired等
	scaleobjectparse	failed	ambiguousshpa	因HPA匹配混乱导致扩缩容对象解析失败	AmbiguousSelector等
Network	networkprepare	failed	subneterror	因子网同步异常导致网络准备失败	FailedSyncNetwork等
	networkprepare	processing	-	网络准备中	ExternalDependenciesSyncd等
	networkassign	failed	noavailableip	因无可用IP导致网络分配失败	NetworkNoIPAvailable等
	networkrelease	failed	-	网络释放失败	NetworkFailed等
Pod	networkassign	failed	-	网络分配失败	FailedAssignENI、FailedAssignEIP等
	networkrelease	failed	-	网络释放失败	FailedReleaseENI等
	networkprobe	failed	-	网络探测失败	warningNetworkNotReady等
	schedule	failed	-	调度失败	FailedScheduling等
	schedule	succeeded	-	调度成功	Scheduled等
	logsetup	failed	-	日志配置失败	SetupLogFailed等

版权所有 © 华为云计算技术有限公司

资源对象	操作类型	操作结果	附加信息	描述信息	类比k8s事件reason举例
	imagepull	failed	-	镜像拉取失败	ImagePullSecretNotFound、FailedPullImage、BackOffPullImage等
	imagepull	failed	imagepullsecretnotfound	因imagepullsecret异常导致镜像拉取失败	ImagePullSecretNotFound等
	imagepull	processing	-	镜像拉取中	Pulling等
	imagepull	succeeded	-	镜像拉取成功	Pulled等
	containerlifecyclehook	failed	poststartfailed	poststart失败	FailedPostStartHook等
	containerlifecyclehook	failed	prestopfailed	prestop失败	FailedPreStopHook等
	containerlifecycleprobe	failed	-	健康检查失败	Unhealthy等
	containerlifecycleprobe	succeeded	-	健康检查成功	Healthy等
	podcreate	failed	-	pod创建失败	FailedCreate、FailedStart、BackOffStart等
	podcreate	processing	-	pod创建中	SandboxChanged等
	podcreate	succeeded	-	pod创建成功	SuccessfulCreate、Started等
	poddelete	failed	-	pod删除失败	FailedKillPod、FailedDelete等
	poddelete	processing	-	pod删除成功	Killing等
	metricsetup	succeed	-	指标设置成功	DefaultMetricsPort等

版权所有 © 华为云计算技术有限公司

资源对象	操作类型	操作结果	附加信息	描述信息	类比k8s事件reason举例
	volumemount	failed	-	volume挂载失败	FailedAttachVolume、FailedMount等
	volumemount	succeed	-	volume挂载成功	SuccessfulAttachVolume、SuccessfulMountVolume等

⚠ 注意

事件源上报事件存在流控，上报事件速度过快，会触发流控，可能会引发事件丢失。默认每个事件源事件上报数量上限为25条，5分钟恢复上报一条。

4.9 日志管理

4.9.1 日志采集

CCI 2.0利用LTS日志服务采集应用日志并上报LTS，可以协助您排查和诊断问题。

日志采集可靠性说明

日志系统的核心功能在于记录业务组件的全生命周期状态数据（包括启动初始化、退出、运行时信息及异常事件等），主要服务于组件运行状态查看与故障根因分析等运维场景。

请注意标准输出流（stdout/stderr）及本地日志文件采用非持久化存储机制，其数据完整性受制于以下风险因素：

- 日志轮转压缩机制可能触发历史文件清除。
- Kubernetes Pod实例终止导致的临时存储卷回收。
- 节点存储空间限制触发的操作系统自动清理。

尽管云原生日志采集插件通过多级缓冲、优先级队列、断点续传等机制优化采集可靠性，但在以下场景仍存在日志采集丢失的可能：

- 业务日志吞吐量超过采集端处理能力。
- 业务Pod终止并立即被容器引擎回收。
- 日志采集器Pod运行异常。

以下是基于云原生日志管理的最佳实践建议，请您认真考虑并采纳：

- 请通过专用高可靠性通道记录并持久化关键业务数据（如金融交易）。
- 请勿在日志中进行记录客户信息、支付凭证、会话令牌等敏感数据。

版权所有 © 华为云计算技术有限公司

约束与限制

- CCI 2.0当前仅支持在西南-贵阳一、华东-上海一、华北-北京四局点使用日志管理。
- 不支持指定系统、设备、cgroup、tmpfs等挂载目录的日志采集。
- 对于单行过长的日志（当前配置为250KB）会直接跳过采集。
- 只支持完整路径和文件名正则匹配，不支持路径正则匹配。
- 同一个容器中所有需要采集的日志文件名不能重复，若重复只会采集到其中一个日志文件的日志。
- Pod启动后不支持配置更新，即配置更新后关联的Pod需重启后配置才能生效。
- 文件所在目录需要容器启动前存在，不支持启动后创建的目录及下层日志文件的日志上报。
- 日志文件的文件名，最大长度为190字符，超过长度限制的日志文件将不会被采集。
- 当前暂不支持initcontainer日志采集。
- 鲲鹏实例暂不支持日志采集。

步骤一 创建日志组

步骤1 登录管理控制台，选择“管理与部署 > 云日志服务 LTS”，进入“日志管理”页面。

步骤2 登录**云日志服务控制台**。

步骤3 进入“日志管理”页面，单击“创建日志组”。

步骤4 在“创建日志组”页面中，参考**表4-27**填写日志组相关信息。

表 4-27 日志组参数说明

参数	说明
日志组名称	日志组是云日志服务进行日志管理的基本单位，用于对日志流进行分类。如果日志较多，需要分门别类，建议您为日志组做好命名，方便后续快速查找日志。 LTS会自动生成一个默认的日志组名称，建议您根据业务自定义日志组的名称，创建成功后支持修改日志组名称。命名规范如下： • 日志组名称只支持输入英文、数字、中文、中划线、下划线及小数点，且不能以小数点、下划线开头或以小数点结尾。长度为1~64个字符。 • 日志组名称不能重复。
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。 默认选择default，建议根据业务选择企业项目，也可单击“查看企业项目”，在企业项目管理页面查看全部企业项目。 • 企业项目需要开通后才能使用，请参考**如何开通企业项目**。 • 支持将该企业项目资源迁出，详细请参考**迁出企业项目资源**。

版权所有 © 华为云计算技术有限公司

参数	说明
日志存储时间(天)	日志组的存储时间，即日志上报到LTS后日志存储的时间。 日志数据默认存储30天，可以在1~365天之间设置。 云日志服务LTS根据配置的日志存储时间定时清理日志内容，例如日志存储时间为30天，上报到LTS的日志只保存30天，30天后开始删除日志内容。 **说明** 目前白名单用户的日志存储时间支持1095天，如有需要，请提工单申请。详细操作请参考**提交工单** • 日志数据默认存储7天，可以在1~7天之间设置。超出存储时间的日志将会被自动删除，您可以按需将日志数据转储至OBS桶中长期存储。 • 上报到云日志服务的原始日志在达到日志存储时间后的次日凌晨就会被删除。
标签	按照业务需求对不同的日志组添加对应的标签。单击“添加”，分别填写标签键key和标签值value，开启应用到日志流后，日志组下的所有日志流即可自动加上该标签。如需添加多个标签可重复该步骤，最多支持添加20个标签。 **标签键key限制条件：** • 标签键可以包含任意语种的字母、数字和空格，以及_.:=+-@字符，但首尾不能包含空格，且不能以_sys_开头。 • 标签键长度不能超过128个字符。 • 标签键名称不可重复。 **标签值value限制条件：** • 标签值可以包含任意语种的字母、数字和空格，以及_.:=+-@字符。 • 标签值长度不能超过255个字符。 **标签策略：** 若您的组织已经设定云日志服务的相关标签策略，则需按照标签策略规则为日志组、日志流、日志接入、主机组、告警规则添加标签。标签如果不符合标签策略的规则，则可能会导致日志组、日志流、日志接入、主机组、告警规则创建失败，请联系组织管理员了解标签策略详情。标签策略详细介绍请参考**标签策略概述**，标签管理详细介绍请参考**标签管理**。 **删除标签：** 如需删除标签，请单击标签操作列的“删除”。 **警告** 删除标签无法恢复，请谨慎操作。 如果配置转储时使用了该标签，删除标签后，请同步修改转储配置信息。
备注	自定义填写备注信息，字符长度0-1024个字符。

步骤5 单击“确定”，日志组创建成功，即可在日志组列表下方生成一条日志组信息。

版权所有 © 华为云计算技术有限公司

图 4-21 日志组

- 在日志组列表中，可以查看日志组名称、标签、日志流数量等信息。
- 单击日志组名称，可跳转到日志详情页面。

----结束

步骤二 创建日志流

1. 在云日志服务管理控制台，单击日志组名称对应的 按钮。
2. 单击展开页面左上角的“创建日志流”，输入日志流名称，名称需要满足如下要求：
   - 只支持输入英文、数字、中文、中划线、下划线及小数点，且不能以小数点、下划线开头或以小数点结尾。
   - 长度为1-64个字符。

   说明

   日志采集后，以日志流为单位，将多条日志数据发往云日志服务。如果日志较多，需要分门别类，建议您创建多个日志流，并给日志流做好命名，方便后续快速查找日志。
3. 在“企业项目”处选择业务需要的企业项目，也可单击“查看企业项目”，在企业项目管理页面查看全部企业项目。
4. 开启日志存储时间，使用日志流的日志存储时间，关闭日志存储时间，使用日志组的日志存储时间。
5. 开启日志存储时间后，根据需要是否开启智能冷存储。
6. 匿名写入默认关闭，适用于安卓/IOS/小程序/浏览器端上报日志，打开匿名写入则表示该日志流打开匿名写入权限，不会经过有效鉴权，可能产生脏数据。
7. 自定义设置标签值，按照“标签键=标签值”的形式填写，例如a=b。
8. 填写“备注”，字符长度0-1024个字符。
9. 单击“确定”，完成日志流的创建。在日志流列表中，可以查看日志流名称、操作等信息。

   说明

   - 支持查看日志流计费状态，日志计费请参考**价格计算器**。
   - 按日志流维度上报话单功能目前在友好用户内测中，您可以**提交工单**申请开通。

步骤三 获取日志组 ID 和日志流 ID

1. 在云日志服务管理控制台“日志管理”页面。
2. 选择**步骤一 创建日志组**所创建的日志组，在操作列选择“更多>详情”。
3. 获取并复制日志组ID。
4. 单击日志组名称对应的 按钮。

版权所有 © 华为云计算技术有限公司

5. 选择**步骤二 创建日志流**所创建的日志流，在操作列选择“详情”。
6. 获取并复制日志流ID。

步骤四 CCI 创建 Deployment

1. 登录**云容器实例 CCI2.0**控制台。
1. 在左侧导航栏单击“负载管理”，选择“无状态负载”页签。
2. 单击“YAML创建”，使用**YAML**或**JSON**创建无状态负载。
   - 方法一 使用YAML创建无状态负载示例：

```
apiVersion: cci/v2
kind: Deployment
metadata:
  annotations:
    description: ''
  labels: {}
  name: test
  namespace: default  # 命名空间
spec:
  replicas: 1
  selector:
    matchLabels:
      app: test
  template:
    metadata:
      annotations:
        logconf.k8s.io/fluent-bit-log-type: lts      # 必填项，使用TLS平台采集日志
        logconfigs.logging.openvessel.io: |
          {
            "default-config": {#支持设置多个容器的日志收集路径（stdout.log表示标准输出，/root/out.log表示rootfs（包含挂卷）下文本日志，/data/emptydir-xxx/*.log表示rootfs（包含挂卷）下文本目录
              "container_files": {
                "container-0": "stdout.log;/root/out.log;/data/emptydir-volume/*.log",
                "container-1": "/root/standard.log"
              },
              "regulation": "", #多行日志收集正则匹配规则，请参考正则匹配规则
              "lts-log-info": {    #仅支持设置单个日志组和日志流
                "<日志组ID>": "<日志流ID>"   #将日志组ID、日志流ID替换为在步骤三 获取日志组ID和日志流ID中获取的日志组ID和日志流ID
              }
            },
            "multi-config": {
              "container_files": {
                "container-0": "/root/multi.log",
                "container-1": "stdout.log;/root/out.log;/data/emptydir-memory-volume/*.log"
              },
              "regulation": "/(?<log>\\d+-\\d+-\\d+ \\d+:\\d+:\\d+.*)/",
              "lts-log-info": {    #仅支持设置单个日志组和日志流
                "<日志组ID>": "<日志流ID>"   #将日志组ID、日志流ID替换为在步骤三 获取日志组ID和日志流ID中获取的日志组ID和日志流ID
              }
            }
          }
        resource.cci.io/pod-size-specs: 2.00_8.0
      labels:
        app: test
        sys_enterprise_project_id: "0"
    spec:
      containers:
      - image: swr.***.com/paas_cci/vk-webhook:8.16.0
        imagePullPolicy: IfNotPresent
        command: ['sh', '-c', "while true; do echo hello; touch /root/out.log; echo hello >> /root/out.log; touch /data/emptydir-volume/emptydir.log; echo hello >> /data/emptydir-volume/emptydir.log; sleep 10; done"]
        lifecycle: {}
        volumeMounts:
```


版权所有 © 华为云计算技术有限公司

```
      - name: emptydir-volume
        mountPath: /data/emptydir-volume
      - name: emptydir-memory-volume
        mountPath: /data/emptydir-memory-volume
      name: container-0
      resources:
        limits:
          cpu: 100m
          memory: 100Mi
        requests:
          cpu: 100m
          memory: 100Mi
      terminationMessagePath: /dev/termination-log
      terminationMessagePolicy: File
    - image: swr.***.com/paas_cci/vk-webhook:8.16.0
      imagePullPolicy: IfNotPresent
      command: ['sh', '-c', "while true; do echo hello; touch /root/out.log; echo hello >> /root/out.log;
touch /data/emptydir-memory-volume/emptydir-memory.log; echo $(date +'%Y-%m-%d
%H:%M:%S.%3N') hello >> /data/emptydir-memory-volume/emptydir-memory.log; echo hello >> /
data/emptydir-memory-volume/emptydir-memory.log; sleep 10; done"]
      lifecycle: {}
      volumeMounts:
      - name: emptydir-volume
        mountPath: /data/emptydir-volume
      - name: emptydir-memory-volume
        mountPath: /data/emptydir-memory-volume
      name: container-1
      resources:
        limits:
          cpu: 100m
          memory: 100Mi
        requests:
          cpu: 100m
          memory: 100Mi
      terminationMessagePath: /dev/termination-log
      terminationMessagePolicy: File
    dnsPolicy: Default
    volumes:
    - name: emptydir-volume
      emptyDir: {}
    - name: emptydir-memory-volume
      emptyDir:
        sizeLimit: 1Gi
        medium: Memory
    enableServiceLinks: false
    restartPolicy: Always
    schedulerName: volcano
    securityContext: {}
    terminationGracePeriodSeconds: 30
  minReadySeconds: 0
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 0
      maxUnavailable: 1
```

表 4-28 关键参数说明

关键参数	是否必选	参数类型	描述
logconf.k8s.io/fluent-bit-log-type	是	String	• 参数解释：日志采集方式 • 约束限制：必填项 • 取值范围：lts

版权所有 © 华为云计算技术有限公司

关键参数	是否必选	参数类型	描述
logconfigs.logging.openvessel.io	是	String	• 参数解释：配置日志采集信息

说明

使用YAML示例时，“logconfigs.logging.openvessel.io”字段的相关参数注释需删除后再使用。

- 方法二 使用JSON创建无状态负载示例：

```
{
   "default-config": {
      "container_files": { // 支持设置多个容器的日志收集路径（stdout.log表示标准输出，/root/out.log表示rootfs（包含挂卷）下文本日志，/data/emptydir-xxx/*.log表示rootfs（包含挂卷）下文本目录）
         "container-0": "stdout.log;/root/out.log;/data/emptydir-volume/*.log",
         "container-1": "stdout.log"
      },
      "regulation": "", // 多行日志收集正则匹配规则，正则匹配规则参考https://docs.fluentbit.io/manual/pipeline/parsers/configuring-parser
      "lts-log-info": { // 仅支持设置单个日志组和日志流
        "日志组ID": "日志流ID"  // 将日志组ID、日志流ID替换为在步骤三 获取日志组ID和日志流ID中获取的日志组ID和日志流ID
      }
   },
   "multi-config": {
      "container_files": { // 支持设置多个容器的日志收集路径（stdout.log表示标准输出，/root/out.log表示rootfs（包含挂卷）下文本日志，/data/emptydir-xxx/*.log表示rootfs（包含挂卷）下文本目录）
         "container-0": "stdout.log;/root/out.log;/data/emptydir-memory-volume/*.log"
      },
      "regulation": "/(?<log>\\d+-\\d+-\\d+ \\d+:\\d+:\\d+.*)/",  // 多行日志收集正则匹配规则
      "lts-log-info": { // 仅支持设置单个日志组和日志流
        "日志组ID": "日志流ID"  // 将日志组ID、日志流ID替换为在步骤三 获取日志组ID和日志流ID中获取的日志组ID和日志流ID
      }
   }
}
```

注意

在YAML和JSON创建过程中需要将“lts-log-info”参数中添加**步骤三 获取日志组ID和日志流ID**。示例：\"lts-log-info\":{\"日志组ID\":\"日志流ID\"}

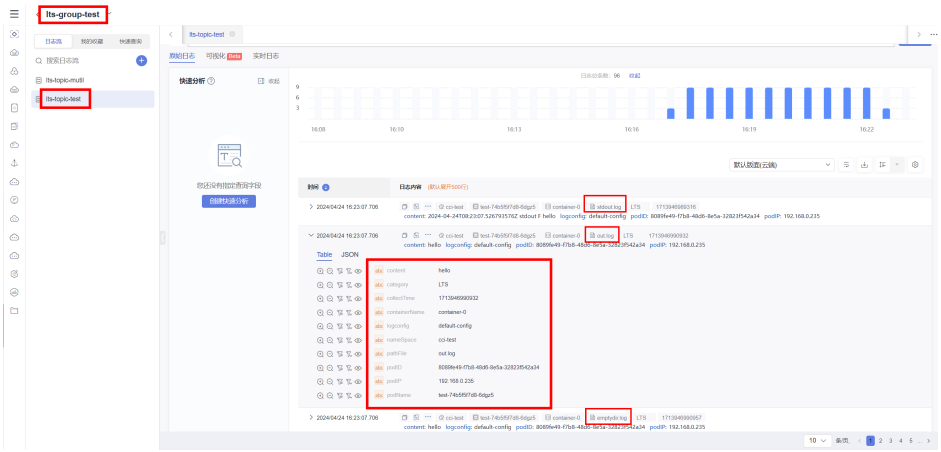
- 将日志组ID替换为**步骤三 获取日志组ID和日志流ID**。
- 将日志流ID替换为**步骤三 获取日志组ID和日志流ID**。

3. 完成后单击“确定”。
4. 单击创建的无状态负载，可查看负载状态。

步骤五 查看日志组、日志流上报情况

1. 在云日志服务管理控制台，单击日志组名称。
2. 进入日志详情页面查看日志。

版权所有 © 华为云计算技术有限公司



4.9.2 挂载容器标准输出日志

本节介绍如何将容器标准输出日志挂载到容器中，以支持从容器内部获取当前Pod各容器的标准输出日志。

约束与限制

“volume.cci.io/mount-stdlog-containers”和“volume.cci.io/mount-stdlog-containers-path”不支持同时配置。

通过 YAML 创建

已创建工作负载为例，配置示例如下，其中关键配置通过红色字体标出：

```
kind: Deployment
apiVersion: cci/v2
metadata:
  name: deploy-example
  namespace: namespace-example
spec:
  replicas: 1
  selector:
    matchLabels:
      app: deploy-example
  template:
    metadata:
      labels:
        app: deploy-example
      annotations:
        volume.cci.io/mount-stdlog-containers: sidecar  #指定需要挂载容器标准输出日志的容器名
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          resources:
            limits:
              cpu: '1'
              memory: 2Gi
            requests:
              cpu: '1'
              memory: 2Gi
        - name: sidecar
          image: sidecar:latest
          resources:
            limits:
              cpu: '0'
```


版权所有 © 华为云计算技术有限公司

```
          memory: '0'
        requests:
          cpu: '0'
          memory: '0'
  dnsPolicy: Default
  imagePullSecrets:
    - name: imagepull-secret
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxUnavailable: 0
    maxSurge: 100%
```

表 4-29 关键参数 Pod 的注解说明

Pod的注解	参数类型	描述	示例值
volume.cci.io/mount-stdlog-containers	String	1. 指定需要挂载容器标准输出日志的容器名，若有多个容器需要挂载则以逗号分隔。也可以配置"*"表示所有容器都进行挂载。如果配置了"*"则不可再配置其它容器名。 2. 匹配了该注解的容器，在启动后会将包含该Pod所有容器标准输出日志的目录挂载到容器内的/var/log/pods路径。	示例一： "container-0,container-1" 示例二： "*"
volume.cci.io/mount-stdlog-containers-path	String	1. 以json格式指定需要挂载容器标准输出日志的容器名和容器内挂载路径。容器名配置"*"表示所有容器都进行挂载。如果配置了"*"则不可再配置其它容器名。 2. 匹配了该注解的容器，在启动后会将包含该Pod所有容器标准输出日志的目录挂载到配置指定的路径。	示例一： "{\"container-0\":\"/var/log/pods\",\"container-1\":\"/tmp/log/pods\"}" 示例二： "{\"*\":\"/tmp/log/pods\"}"

版权所有 © 华为云计算技术有限公司

5 通过 CCE 使用 CCI

5.1 CCE 突发弹性引擎（对接 CCI）插件功能概览

本章节主要介绍CCE突发弹性引擎（对接 CCI）提供的服务功能概览、资源使用说明和自定义注解。

CCE突发弹性引擎（对接 CCI）作为一种虚拟的kubelet用来连接Kubernetes集群和其他平台的API。bursting插件的主要作用是将Kubernetes API扩展到无服务器的容器平台CCI。

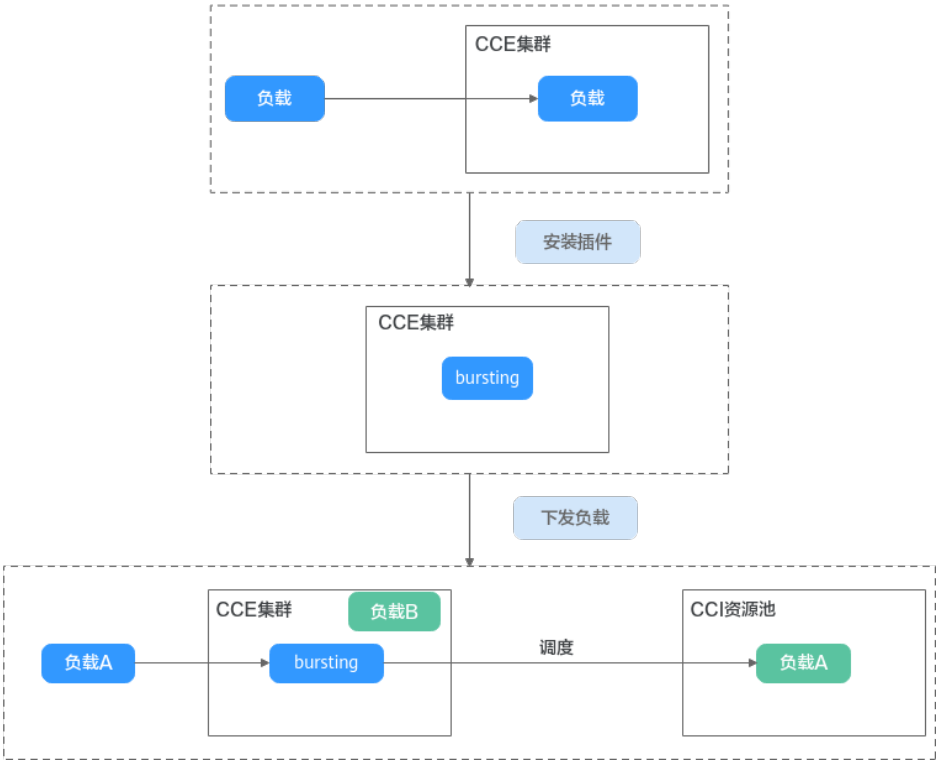
基于该插件，支持用户在短时高负载场景下，将部署在云容器引擎CCE上的无状态负载（Deployment）、有状态负载（StatefulSet）、普通任务（Job）、定时任务（CronJob）四种资源类型的容器实例（Pod），弹性创建到云容器实例 CCI2.0服务上，以减少集群扩容带来的消耗。

功能概览

使用CCE集群+bursting插件时，您需要重点关注**下发负载**和**调度**的流程。

 版权所有 © 华为云计算技术有限公司

图 5-1 突发弹性引擎流程图



注意

下发负载需要用户对负载进行配置，和用户自身的业务高度相关。

工作负载配置项	功能描述	功能规格	相关文档拓展
调度	用户可以通过多种方式来管理CCE集群的工作负载，通过合理的调度策略配置。可控制负载调度到CCI 2.0服务，进一步提升用户集群的资源利用率。	• 支持3种调度策略。 • 支持2种管理调度策略的方式。 • 支持多个虚拟节点调度。	5.3
资源配额	用户通过配置pod的cpu、memory等字段约束容器使用资源规格和上限。插件对资源规格进行规整，满足弹性CCI 2.0需求。	提供对pod资源配额进行规整的能力。	5.4
镜像	用户通过镜像配置自身业务镜像，将自己的业务容器运行在华为云CCE集群+CCI服务上。	• 支持修改镜像配置方式。 • 支持原地升级镜像。	5.5

版权所有 © 华为云计算技术有限公司

工作负载配置项	功能描述	功能规格	相关文档拓展
存储	用户通过存储相关的配置为工作负载外挂存储卷，以完成业务中数据持久存储的诉求。	• 支持多种存储类型。 • 支持替换工作负载hostpath配置方式。	5.6
网络	用户通过网络配置规划CCE集群和CCI之间的网络拓扑。	• 支持CCI侧负载通过Service发布。 • 支持CCE集群Pod与CCI中Pod通过Service互通。 • 支持指定cluster-dns。	5.10
监控	用户可以通过配置插件对接监控平台，提升弹性CCI工作负载的可观测性。	• CCI 2.0 console查看。	5.11

资源使用说明

CCE集群+bursting的使用场景涉及到华为云周边服务的搭配使用，如下表格详细描述了涉及到的周边服务。

涉及服务	资源说明	备注
CCI 2.0	插件会在CCI 2.0服务新建一个名为“bursting-”+“CCE集群ID”的命名空间。	• 不建议直接在CCI 2.0服务使用该命名空间，如需使用CCI 2.0服务，请另外新建命名空间。 • CCI 2.0服务命名空间不产生计费。
CCE	CCE侧的工作负载、Secret、Congfigmap、PV、PVC会同步到CCI，同时也占用CCE节点的资源。	• CCE集群资源规模1000 pod+1000 configmap建议申请2U4G节点规格。 • CCE集群资源规模2000 pod+2000 configmap建议申请4U8G节点规格。 • CCE集群资源规模4000 pod+4000 configmap建议申请8U16G节点规格。
ELB	开启“支持CCE集群pod与CCI集群pod通过service互通”功能bursting插件会自动创建ELB资源。	• 创建共享型性能保障规格的ELB，名称为cce-lb-xxx。 • 当卸载插件或关闭网络互通功能时，该ELB资源会被自动清除。

版权所有 © 华为云计算技术有限公司

涉及服务	资源说明	备注
VPC	弹性CCI的负载共享使用CCE集群所在VPC。	CCI命名空间Service网段为10.247.0.0/16，CCE集群的VPC子网网段请避开该网段。
SWR	创建弹性到CCI的工作负载选择镜像时，直接对接华为云SWR服务。	创建负载请确认镜像已被正确上传至您的SWR仓库。

Pod annotation

CCE集群+bursting的使用场景涉及到自定义注解，相关注解含义如下表所示。

annotation key	描述	相关文档
scheduling.cci.io/managed-by-profile	标记当前pod被哪个profile资源管理。	5.3
resource.cci.io/size	标记弹性后规整的cpu和memory资源。	5.4
bursting.cci.io/image-replacement	镜像地址前缀替换相关。	5.5

版本记录

表 5-1 CCE 突发弹性引擎（对接 CCI）插件版本记录

插件版本	支持的集群版本	更新特性
1.5.44	v1.21 v1.23 v1.25 v1.27 v1.28 v1.29 v1.30 v1.31 v1.32	• 支持CCE v1.32集群 • 支持自动注入Sidecar容器 • 适配ARM64节点部署

版权所有 © 华为云计算技术有限公司

插件版本	支持的集群版本	更新特性
1.5.29	v1.21 v1.23 v1.25 v1.27 v1.28 v1.29 v1.30 v1.31	支持Pod配置指定子网
1.5.28	v1.21 v1.23 v1.25 v1.27 v1.28 v1.29 v1.30 v1.31	功能优化
1.5.27	v1.21 v1.23 v1.25 v1.27 v1.28 v1.29 v1.30 v1.31	支持CCE v1.31集群
1.5.26	v1.21 v1.23 v1.25 v1.27 v1.28 v1.29 v1.30	问题修复

版权所有 © 华为云计算技术有限公司

插件版本	支持的集群版本	更新特性
1.5.24	v1.21 v1.23 v1.25 v1.27 v1.28 v1.29 v1.30	功能优化
1.5.16	v1.21 v1.23 v1.25 v1.27 v1.28 v1.29 v1.30	仅进行Pod级别CPU和Memory资源规整
1.5.8	v1.21 v1.23 v1.25 v1.27 v1.28 v1.29	适配CCE v1.29集群
1.5.2	v1.21 v1.23 v1.25 v1.27 v1.28	支持弹性至CCI 2.0
1.3.57	v1.21 v1.23 v1.25 v1.27 v1.28	适配CCE v1.28集群
1.3.54	v1.21 v1.23 v1.25 v1.27	修复部分问题

版权所有 © 华为云计算技术有限公司

插件版本	支持的集群版本	更新特性
1.3.48	v1.21 v1.23 v1.25 v1.27	• 支持v1.25、v1.27版本集群 • 支持JuiceFS类型的存储
1.3.44	v1.17 v1.19 v1.21 v1.23	支持Pod配置全域弹性公网IP
1.3.35	v1.17 v1.19 v1.21 v1.23	• 支持原地升级镜像 • 支持ReadinessGates
1.3.25	v1.17 v1.19 v1.21 v1.23	• 支持DownwardAPI Volume • 支持Projected Volume • 支持自定义StorageClass
1.3.19	v1.17 v1.19 v1.21 v1.23	支持schedule profile
1.3.7	v1.17 v1.19 v1.21 v1.23	支持v1.21、v1.23版本集群
1.2.12	v1.13 v1.15 v1.17 v1.19	• 新增了部分metrics指标 • 支持HPA与CustomedHPA • 支持将弹性到CCI的Pod中的hostPath转换为其它类型存储 • 修复Kubernetes Dashboard无法使用终端问题

版权所有 © 华为云计算技术有限公司

插件版本	支持的集群版本	更新特性
1.2.5	v1.13 v1.15 v1.17 v1.19	• 支持CCE Turbo集群 • 自动清理CCI中不再被Pod依赖的资源 • 支持配置Requests与Limits不相等，弹性到CCI时的资源申请量以Limits为准 • 修复CCI命名空间不存在时插件卸载失败问题 • 增加对Pod规格超过CCI限制的创建请求的拦截
1.2.0	v1.13 v1.15 v1.17 v1.19	• 支持v1.19版本集群 • 支持SFS、SFS Turbo类型存储 • 支持CronJob • 支持配置envFrom • 日志文件自动转储 • 屏蔽TCPSocket类型健康检查 • 支持配置资源标签（pod-tag） • 提升了性能和可靠性 • 修复了一些已知问题
1.0.5	v1.13 v1.15 v1.17	支持v1.17版本集群

5.2 快速使用

本章节主要指导用户在CCE集群上使用bursting插件，快速帮助用户将负载通过插件从CCE集群弹性到CCI 2.0上。

前提条件

- 使用插件前需要用户在CCI界面对CCI服务进行授权。
- 如果使用CCI 2.0服务对接CCE突发弹性引擎（对接 CCI）请先购买云服务VPCEP，具体操作步骤请参见**环境设置**。

约束与限制

- 仅支持VPC网络模式的CCE Standard集群和CCE Turbo集群。
- CCE突发弹性引擎（对接 CCI）插件1.5.37及以下版本不支持Arm集群。如果集群中包含Arm节点，插件实例将不会部署至Arm节点。
- 集群所在子网不能与10.247.0.0/16重叠，否则会与CCI命名空间下的Service网段冲突，导致无法使用。

版权所有 © 华为云计算技术有限公司

- Volcano调度器1.17.10及以下版本暂不支持使用Volcano调度器将挂载云存储卷的容器实例（Pod）弹性到CCI。
- 使用CCE集群中的Bursting插件对接CCI 2.0服务，支持配置独享型ELB的Ingress和Service。Bursting插件1.5.5以下版本不支持配置ELB类型的Service。
- 暂不支持守护进程集（DaemonSet）。
- 暂不支持动态资源分配（Dynamic Resource Allocation）特性，并在1.5.27插件版本对相关配置进行拦截。
- 安装bursting插件后会在CCI服务新建一个名为"bursting-"+集群ID的命名空间，该命名空间完全由bursting插件管理，不建议直接在CCI服务使用该命名空间，如需使用CCI服务，请另外新建命名空间。

安装插件

1. 登录CCE控制台。
2. 选择CCE集群，单击进入CCE集群总览页面。
3. 在导航栏左侧单击“插件中心”，进入插件中心首页。
4. 选择“CCE 突发弹性引擎 (对接 CCI)”插件，单击“安装”。
5. 选择插件版本，推荐您使用最新版本CCE 突发弹性引擎 (对接 CCI)插件。
6. 在安装插件页面，根据需求选择“规格配置”。
   - 选择“系统预置规格”时，系统会根据不同的预置规格配置插件的实例数及资源配额，具体配置值请以控制台显示为准。
   - 选择“自定义规格”时，您可根据需求调整插件实例数及资源配额。实例数为1时插件不具备高可用能力，当插件实例所在节点异常时可能导致插件功能无法正常使用，请谨慎选择。

     说明

     - CCE 突发弹性引擎 (对接 CCI) 插件在1.5.2及以上版本，将占用更多节点资源，请在升级CCE突发弹性引擎（对接 CCI）插件前预留节点可创建的Pod数量。节点可分配的Pod数量可参考：**节点可创建的最大Pod数量说明**。
     - 弹性到CCI的业务量不同时，插件的资源占用也不相同。其中proxy、resource-syncer、bursting-resource-syncer组件的资源配额设置与最大弹性Pod数量有关，virtual-kubelet、bursting-virtual-kubelet、profile-controller、webhook、bursting-webhook组件的资源配额设置与最大Pod创建或删除的并发数有关，各组件的资源申请值和限制值推荐计算公式请参见**表5-2**，其中P为最大弹性Pod数量，C为最大Pod创建或删除的并发数，建议用户评估自己的业务使用量，选择规格配置。

表 5-2 Bursting 组件规格计算公式

组件名称	CPU Request(m)	CPU Limit(m)	Memory Request(Mi)	Memory Limit(Mi)
virtual-kubelet、bursting-virtual-kubelet	(C+400)/2400*1000	(C+400)/600*1000	(C+400)/2400*1024	(C+400)/300*1024

版权所有 © 华为云计算技术有限公司

组件名称	CPU Request(m)	CPU Limit(m)	Memory Request(Mi)	Memory Limit(Mi)
profile-controller	(C+1000)/ 6000*1000	(C+400)/ 1200*1000	(C+1000)/ 6000*1024	(C+400)/ 1200*1024
proxy	(P+2000)/ 12000*1000	(P+800)/ 2400*1000	(P+2000)/ 12000*1024	(P+800)/ 2400*1024
resource-syncer、bursting-resource-syncer	(P+800)/ 4800*1000	(P+800)/ 1200*1000	(P+800)/ 4800*1024	(P+800)/ 600*1024
webhook、bursting-webhook	(C+400)/ 2400*1000	(C+400)/ 600*1000	(C+1000)/ 6000*1024	(C+400)/ 1200*1024

  - 网络互通（可选），开启后，支持CCE集群中的Pod与CCI集群中的Pod通过Kubernetes Service互通，并在插件安装时部署Proxy组件。详细功能介绍请参考**网络**。
7. 设置插件支持的参数配置。
  - 子网：选择弹到 CCI 的 Pod 实例会占用所选子网下的 IP，请合理规划网段，避免 IP 资源不足。
  - 企业项目：选择所在企业项目。
8. 完成以上配置后，单击“安装”。

工作负载下发

1. 登录CCE控制台。
2. 选择CCE集群，单击进入CCE集群总览页面。
3. 在导航栏左侧单击“工作负载”，进入工作负载首页。
4. 单击“创建工作负载”，具体操作步骤详情请参见**创建工作负载**。
5. 填写基本信息。“弹性至CCI”选择“强制调度”策略，CCI资源池选择“CCI 2.0资源池”。关于调度策略更多信息，请参考**调度负载到CCI**。
6. 进行容器配置。
7. 配置完成后，单击“创建工作负载”。
8. 在工作负载页面，选择工作负载名称，单击进入工作负载管理界面。
9. 工作负载所在节点为bursting-node，说明负载成功已调度到CCI。

版权所有 © 华为云计算技术有限公司

插件卸载

1. 登录CCE控制台。
2. 选择CCE集群，单击进入CCE集群总览页面。
3. 在导航栏左侧单击“插件中心”，进入插件中心首页。
4. 选择“CCE 突发弹性引擎 (对接 CCI)”插件，单击“卸载”。

表 5-3 特殊场景说明

特殊场景描述	场景现象	场景说明
CCE集群无节点，卸载插件。	插件卸载失败。	bursting插件卸载时会在集群中启动Job用于清理资源，卸载插件时请保证集群中至少有一个可以调度的节点。
用户直接删除集群，未卸载插件。	用户在CCI侧的命名空间中有资源残留，如果命名空间有计费资源，会造成额外计费。	由于直接删除集群，没有执行插件的资源清理Job，造成资源残留。用户可以手动清除残留命名空间及其下的计费资源来避免额外计费。

关于CCE突发弹性引擎（对接CCI）更多内容详情请参见：**CCE突发弹性引擎（对接CCI）**。

5.3 调度负载到 CCI 2.0

对于使用CCE集群和CCI的使用场景，用户可以按需将工作负载调度到CCE集群节点或者对接CCI的虚拟节点，本文详细介绍如何将CCE集群的工作负载调度到CCI 2.0上。

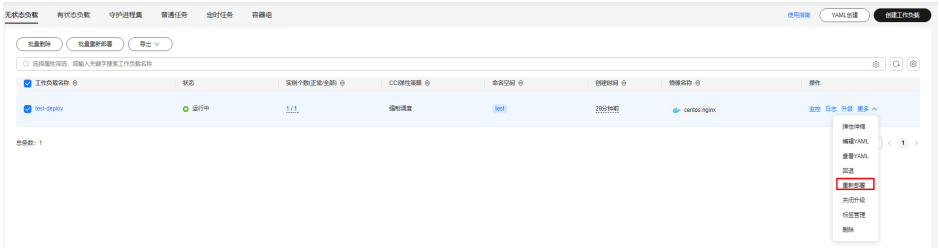
bursting插件当前提供了以下方式管理CCE集群的pod，使其能够调度到CCI 2.0：

- **方式一：通过配置labels控制pod调度到CCI**
- **方法二：通过配置profile控制pod调度到CCI**

版权所有 © 华为云计算技术有限公司

约束与限制

- 当前只有负载的原生标签能够被ScheduleProfile匹配，使负载能够被ScheduleProfile管理，将CCE集群的Pod调度到CCI 2.0。例如通过**ExtensionProfile**加到负载上的标签不能够被ScheduleProfile匹配，因此不会被ScheduleProfile管理调度功能。
- 通过配置labels控制pod调度到CCI 2.0的方式，需要在创建工作负载前添加label，若对已创建的工作负载进行追加label操作，存量工作负载对应的pod不会更新，可选择该工作负载，单击“更多 -> 重新部署”实现更新。



前提条件

- 插件版本：已安装CCE突发弹性引擎（对接 CCI）且插件，如果使用配置profile控制pod调度到CCI的方式请安装CCE突发弹性引擎（对接 CCI）插件版本为1.3.19及以上版本。
- 集群类型：如果使用配置profile控制pod调度到CCI方式仅支持VPC网络模式的CCE Standard集群和CCE Turbo集群。

调度策略

CCE集群工作负载弹性调度到CCI 2.0策略有以下几种：

调度策略	策略图解	适用场景
强制调度策略（enforce）		CCE工作负载强制弹性到CCI 2.0。
自动调度策略（auto）		根据用户集群调度器实际打分决定是否弹性到CCI 2.0。
本地优先调度策略（localPrefer）		工作负载优先调度到CCE集群，集群资源不足时工作负载弹性到CCI 2.0。
不开启（off）		工作负载不会弹性到CCI 2.0。

方法一：通过配置 labels 控制 pod 调度到 CCI

您可以通过控制台或YAML配置labels控制pod调度到CCI。

版权所有 © 华为云计算技术有限公司

通过控制台配置

步骤1 登录CCE控制台，单击集群名称，进入概览页面。

步骤2 选择“工作负载”，单击“创建工作负载”。

步骤3 在基本信息中“弹性至CCI”选择“不开启”以外的任意策略。

- 本地优先调度：Pod优先调度至CCE节点，当CCE节点资源不足时将Pod弹性至CCI。
- 强制调度：所有Pod均会弹性至CCI。

步骤4 勾选相关的CCI资源池。

- CCI 2.0资源池(bursting-node)：新一代Serverless容器资源池。
- CCI 1.0资源池(virtual-kubelet)：存量Serverless容器资源池，即将日落。

> **说明**
>
> 在CCE集群控制台（console）创建工作负载时，CCI弹性承载支持勾选“CCI 2.0资源池(bursting-node)”或者“CCI 1.0资源池(virtual-kubelet)”，如果使用CCI 1.0请勾选“virtual-kubelet”，如果使用CCI 2.0请勾选“bursting-node”。目前CCI 2.0仅对白名单用户开放，如需使用CCI 2.0服务，请提交工单申请开启CCI 2.0服务。

步骤5 CCE工作负载具体操作步骤详情请参见**创建工作负载**，完成后单击“确定”即可。

----结束

通过 YAML 文件配置

步骤1 登录CCE控制台，单击集群名称，进入概览页面。

步骤2 选择“工作负载”，单击“YAML创建”。

步骤3 在工作负载的yaml文件中添加virtual-kubelet.io/burst-to-cci标签。

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test
  namespace: default
  labels:
    bursting.cci.io/burst-to-cci: 'auto'    # 弹性到CCI
spec:
  replicas: 2
  selector:
    matchLabels:
      app: test
  template:
    metadata:
      labels:
        app: test
    spec:
      containers:
        - image: 'nginx:perl'
          name: container-0
          resources:
            requests:
              cpu: 250m
              memory: 512Mi
            limits:
              cpu: 250m
              memory: 512Mi
          volumeMounts: []
      imagePullSecrets:
        - name: default-secret
```

 版权所有 © 华为云计算技术有限公司

表 5-4 关键参数说明

参数	类型	描述
bursting.cci.io/burst-to-cci	String	表示CCE集群工作负载弹性调度到CCI策略，取值如下： • enforce，CCE工作负载强制弹性到CCI 2.0。 • auto，根据用户集群调度器实际打分决定是否弹性到CCI 2.0。 • localPrefer，工作负载优先调度到CCE集群，集群资源不足时工作负载弹性到CCI 2.0。 • off，工作负载不会弹性到CCI 2.0。

步骤4 完成后单击“确定”。

----结束

方法二：通过配置 profile 控制 pod 调度到 CCI

您可以通过控制台或YAML配置profile控制pod调度到CCI。

通过控制台配置

步骤1 登录CCE控制台，单击集群名称，进入概览页面。

步骤2 选择“策略 > CCI弹性承载策略”。

步骤3 单击“创建CCI弹性承载策略”，并填写相关信息。

表 5-5 创建 CCI 弹性承载策略

参数	参数说明
策略名称	输入策略名称。
命名空间	选择弹性策略生效的命名空间。您可选择命名空间或创建命名空间，创建命名空间操作步骤详情请参见**创建命名空间**。
关联负载	填写键值或引用负载标签。
调度策略	选择调度策略。 • 本地优先调度：Pod优先调度至CCE节点，当CCE节点资源不足时将Pod弹性至CCI。 • 强制调度：所有Pod均会弹性至CCI。

版权所有 © 华为云计算技术有限公司

参数	参数说明
分配策略	• 本地：设置本地CCE集群。 • CCI：设置CCI上运行的“最大实例数”。
最大实例数	设置“本地”或“CCI”运行的最大实例数。
本地缩容优先级	取值范围[-100, 100]，数值越大越先缩容。
CCI缩容优先级	取值范围[-100, 100]，数值越大越先缩容。
CCI资源池	• CCI 2.0资源池(bursting-node)：新一代Serverless容器资源池。 • CCI 1.0资源池(virtual-kubelet)：存量Serverless容器资源池，即将日落。

步骤4 单击“确定”。

步骤5 选择“工作负载”，单击并创建工作负载，其中“高级配置 > 标签与注解”中添加Pod标签，其中键值与**步骤3**创建关联负载中的键值需保持一致，其他操作步骤请参考**创建工作负载**。

步骤6 完成后单击“创建工作负载”即可。

----结束

通过 YAML 配置

步骤1 登录CCE控制台，单击集群名称，进入概览页面。

步骤2 选择“策略 > CCI弹性承载策略”。

步骤3 单击“YAML创建”，创建profile对象。

示例一：限制调度到CCE集群最大负载数，配置local maxNum和scaleDownPriority的profile模板如下：

```
apiVersion: scheduling.cci.io/v2
kind: ScheduleProfile
metadata:
  name: test-local-profile
  namespace: default
spec:
  objectLabels:
    matchLabels:
      app: nginx
  strategy: localPrefer
  virtualNodes:
    - type: bursting-node
  location:
    local:
      maxNum: 20 # 当前暂不支持local/cci同时配置maxNum
      scaleDownPriority: 2
    cci:
      scaleDownPriority: 10
```


版权所有 © 华为云计算技术有限公司

示例二：限制CCI集群最大负载数，配置cci maxNum和scaleDownPriority的profile的模板如下：

```
apiVersion: scheduling.cci.io/v2
kind: ScheduleProfile
metadata:
  name: test-cci-profile
  namespace: default
spec:
  objectLabels:
    matchLabels:
      app: nginx
  strategy: localPrefer
  virtualNodes:
    - type: bursting-node
  location:
    local: {}
    cci:
      maxNum: 20 # 当前暂不支持local/cci同时配置maxNum
      scaleDownPriority: 10
```

表 5-6 关键参数说明

参数	类型	描述
strategy	String	表示CCE集群工作负载弹性调度到CCI 2.0策略，取值如下： ● enforce，CCE工作负载强制弹性到CCI 2.0。 ● auto，根据用户集群调度器实际打分决定是否弹性到CCI 2.0。 ● localPrefer，工作负载优先调度到CCE集群，集群资源不足时工作负载弹性到CCI 2.0。
maxNum	int	表示pod最大数量配置。 取值范围[0~int32]。
scaleDownPriority	int	表示缩容优先级配置，数值越大所关联的pod越先缩容。 取值范围[-100, 100]。

说明

- location配置local字段（CCE侧）和cci字段（CCI侧）控制pod数量和缩容优先级。
- local字段和cci字段不可以同时设置maxNum。
- 缩容优先级为非必填项参数，如果不配置缩容优先级，默认值将为空。

步骤4 单击“确定”。

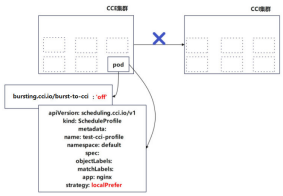
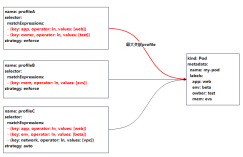
步骤5 创建无状态负载，使用selector方式选择含有“app：nginx”的pod，关联上文创建的profile。

```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: nginx
spec:
  replicas: 10
  selector:
```

 版权所有 © 华为云计算技术有限公司

```
  matchLabels:
    app: nginx
template:
  metadata:
    labels:
      app: nginx
  spec:
    containers:
      - name: container-1
        image: nginx:latest
        imagePullPolicy: IfNotPresent
        resources:
          requests:
            cpu: 250m
            memory: 512Mi
          limits:
            cpu: 250m
            memory: 512Mi
    imagePullSecrets:
      - name: default-secret
```

表 5-7 特殊场景使用说明

特殊使用场景	使用说明
pod同时使用label和profile控制调度CCI 2.0	使用label控制调度CCI 2.0的方式优先级高于使用profile控制调度CCI 2.0的方式。 例如label中策略为off，profile的策略为enforce，最终pod不会调度到CCI 2.0。 
pod同时被多个profile关联	pod有且仅有一个关联profile。当pod被多个profile关联，根据profile匹配pod的label的数量，选出最大关联profile，存在多个时，选择字典序小的profile关联。 pod最终被profileA关联 

步骤6 单击“确定”。

----结束

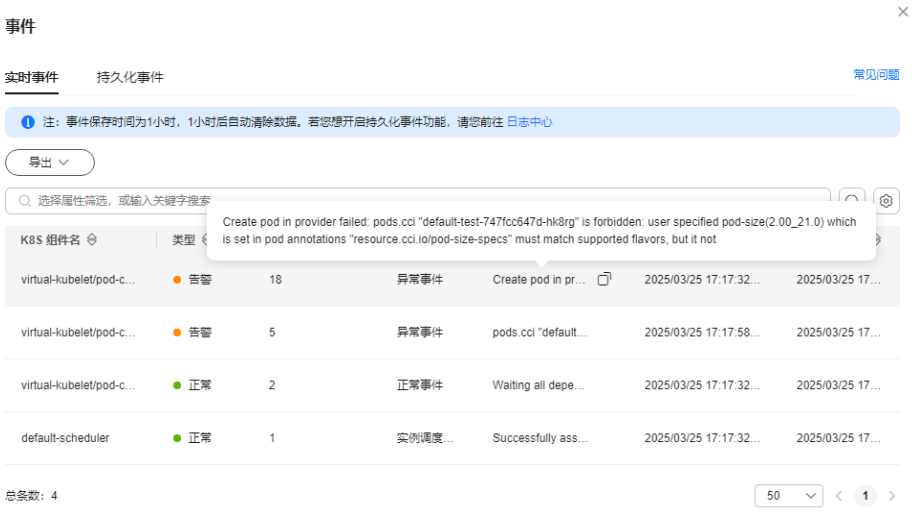
5.4 实例资源规整

对于CCE集群中弹性到CCI的pod，bursting插件会根据用户提供的CPU、内存规格，对pod的资源规格做出规整，以达到合理利用资源并满足CCI 2.0资源规格的目的。本章节将详细介绍如何进行资源规格的规整。

版权所有 © 华为云计算技术有限公司

约束与限制

- 弹性到CCI的pod需要至少有一个容器有指定CPU&Memory的limit或者request资源，或存在“resource.cci.io/pod-size-specs”的注解。
- 弹至CCI 2.0的pod需遵循CCI 2.0的规格约束，如果未命中CCI 2.0支持的规格，会在CCE侧的pod事件里产生如下类似报错，且对应pod在CCI侧不会被创建。规格详情可参考**4.1.2.3.1**。



Pod 资源规格算法

Pod需要运行一些必要的系统组件，会占用一些系统资源， 因此，您的Pod内存规格与Pod资源可分配量之间会存在差异。CCI对用户Pod可分配的资源计算法则如下：

- Pod内存规格 <= 2Gi：

  Pod资源可分配量 = **Pod内存规格**
- pod内存规格 > 2Gi：

  Pod资源可分配量 = **Pod内存规格** - **CCI组件资源预留** - **OS侧资源预留**

> **⚠ 注意**
>
> **Pod内存规格**是指Pod的付费内存规格，即显示在pod annotation中，格式为resource.cci.io/size=${cpuCeil}_${memoryCeil}的参数中的${memoryCeil}值。

详情请参见**规格规整计算规则说明**。

预留系统开销说明

从bursting弹性到CCI 2.0的Pod，可通过pod/podTemplate的annotation "resource.cci.io/memory-reservation"和"resource.cci.io/memory-burst-size"来开启预留系统开销。详细规则说明可参考：**预留系统开销说明**

版权所有 © 华为云计算技术有限公司

5.5 镜像

用户可以通过华为云镜像仓库服务SWR来管理业务镜像。本章节将介绍CCE+bursting插件场景中，涉及到镜像相关的使用场景及用法。通过阅读本章用户可以在CCE +bursting插件场景中：

- **使用SWR拉取用户业务镜像**
- **混合使用华为云SWR + 第三方镜像仓拉取用户业务镜像**

使用 SWR 拉取用户业务镜像

- **方式一：通过console选取SWR镜像**
  a. 用户镜像上传SWR。使用方式请参考：**SWR官方文档**。
  b. 在华为云CCE控制台创建负载选择镜像。

  c. 对接到SWR中的镜像。请确保您的SWR仓库中已正确上传了镜像，SWR服务使用详情请参见：**SWR官方文档**。
- **方式二：通过在CCE集群配置node选取SWR镜像**
  a. 查看SWR镜像仓库中的镜像地址，示例：swr.cn-north-4.myhuaweicloud.com/cci-test/nginx:1.0.0.x86_64_test
  其中region为cn-north-4，cci-test为swr的组织。
  b. 登录CCE集群节点。
  c. 配置工作负载yaml。

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test
  namespace: default
  labels:
    bursting.cci.io/burst-to-cci: enforce    # 弹性到CCI 2.0
spec:
  replicas: 2
  selector:
    matchLabels:
      app: test
  template:
    metadata:
      labels:
        app: test
    spec:
      containers:
        - image: swr.***.com/cci-test/nginx:1.0.0.x86_64_test
          name: container-0
```


版权所有 © 华为云计算技术有限公司

```
        resources:
          requests:
            cpu: 250m
            memory: 512Mi
          limits:
            cpu: 250m
            memory: 512Mi
        volumeMounts: []
      imagePullSecrets:
        - name: default-secret
```

d. 部署工作负载。

```
kubectl apply -f  test-deployment.yaml
```

混合使用华为云 SWR + 第三方镜像仓拉取用户业务镜像

操作场景：

在某些情况下，用户在CCE的工作负载拉取第三方镜像仓库的镜像，拉取的镜像可以同步到SWR服务，当发生业务波峰时，弹性到CCI的工作负载使用SWR镜像，可以极大提高镜像拉取效率，帮助用户更丝滑应对业务波峰。

操作指导：

为工作负载yaml配置annotation，示例值如下所示：

```
bursting.cci.io/image-replacement: '[
    {"repositoryPrefix":"harbor.domain","replaceWith":"swr.***.com/org1"},
    {"repositoryPrefix":"","replaceWith":"swr.***.com/org1"},
    {"repositoryPrefix":"harbor.domain/a/b/c/d","replaceWith":"swr.***.com/org2"}
]'
```

说明

- 配置策略执行无先后顺序。
- 替换策略可以配置多条，各个策略repositoryPrefix值不允许重复。

替换规则key	字段含义	字段说明
repositoryPrefix	用户希望被匹配并被替换的镜像前缀字段。	• 该字段为空，匹配所有image字段不含"/"的容器。 • 该字段不为空，匹配所有image字段最后一个"/"字符前缀相同的容器。 • 该字段字符校验规则与容器镜像名的规则一致，且不能以"/"字符结尾。
replaceWith	用户希望替换成的镜像前缀字段。	• 该字段字符不能与repositoryPrefix相同。 • 该字段字符校验规则与容器镜像名的规则一致，且不能以"/"字符结尾。

版权所有 © 华为云计算技术有限公司

表 5-8 annotation 说明

annotation	替换前	替换后	说明
bursting.cci.io/image-replacement: '[{"repositoryPrefix":"harbor.domain","replaceWith":"swr.***.com/org1"}]'	containers: - name: container-0 image: 'harbor.domain/ubuntu:latest'	containers: - name: container-0 image: 'swr.***.com/org1/ubuntu:latest'	repositoryPrefix匹配第三方仓库域名场景。
bursting.cci.io/image-replacement: '[{"repositoryPrefix":"","replaceWith":"swr.***.com/org1"}]'	containers: - name: container-1 image: 'nginx:latest'	containers: - name: container-1 image: 'swr.***.com/org1/nginx:latest'	repositoryPrefix为空场景。
bursting.cci.io/image-replacement: '[{"repositoryPrefix":"harbor.domain/a/b/c/d","replaceWith":"swr.***.com/org2"}]'	containers: - name: container-2 image: 'harbor.domain/a/b/c/d/redis:latest'	containers: - name: container-2 image: 'swr.***.com/org2/redis:latest'	repositoryPrefix匹配第三方仓库域名+组织目录场景。

5.6 存储

弹性到CCI的工作负载支持多种华为云存储配置，用于满足客户多样化的存储需求。通过阅读本章用户可以：

- 了解弹性CCI的负载支持的存储类型。
- 了解弹性CCI的负载Hostpath类型的典型场景以及如何使用。

约束与限制

调度到CCI的实例的存储类型支持ConfigMap、Secret、EmptyDir、DownwardAPI、Projected、PersistentVolumeClaims几种Volume类型，其中Projected和DownwardAPI类型仅bursting 1.3.25版本及以上支持。

- EmptyDir：不支持子路径。
- PersistentVolumeClaims：只支持SFS Turbo云存储类型，且只支持使用CSI类型的StorageClass。1.17.10及以下版本的Volcano调度器不支持调度所有云存储类型。
- Projected：如配置了serviceAccountToken类型的source，那么弹性到CCI后挂载的会是对应service-account-token secret中的token，该token为长期有效的token且没有预期受众，即expirationSeconds和audience两项配置不会生效。

版权所有 © 华为云计算技术有限公司

支持的存储类型

用户在配置负载存储类型时，CCE的console有如下选项。

弹性CCI的负载对存储类型的支持情况如下：

volume类型	是否支持	特殊场景说明
HostPath	否	• CCI是共享集群，不直接开放HostPath能力。 • 1.5.9及以上版本可支持配置path为/etc/localtime的HostPath存储，配置后CCI侧容器中挂载的时区将会与CCE节点的时区一致。
ConfigMap	是	-
Secret	是	-
EmptyDir	是	EmptyDir的sizeLimit仅对emptyDir.medium配置为“Memory”生效。
DownwardAPI	是	-
Projected	是	如配置了serviceAccountToken类型的source，弹性到CCI后会挂载对应service-account-token secret中的token，该token为长期有效，且token没有预期受众，即expirationSeconds和audience两项配置不会生效。
PersistentVolumeClaims	是	只支持SFS Turbo云存储类型，且只支持使用CSI类型的StorageClass。

负载 Hostpath 配置方式

操作场景

在使用CCE或者其他K8s集群时，用户使用HostPath类型存储。但CCI是共享集群，不开放HostPath能力，因此使用HostPath的Pod通过bursting弹到CCI时，会被拦截。如

版权所有 © 华为云计算技术有限公司

无法改变Pod spec.volumes中配置的HostPath，可通过配置Annotation的形式，允许让使用HostPath的Pod弹性到CCI上，bursting在校验时需要去掉Pod中的HostPath或者将HostPath替换为emptyDir。

约束与限制

- LocalDir和flexVolume当前暂不支持。

操作步骤

通过在Pod.Annotations中加入注解可以做到hostPath转emptydir。

- 单个hostPath替换为emptyDir配置方式：
  ```
  bursting.cci.io/hostpath-replacement: '[{"name":"source-hostpath-volume-3","policyType":"replaceByEmptyDir","emptyDir":{"sizeLimit":"10Gi"}}]'
  ```
- 单个hostPath忽略：
  ```
  bursting.cci.io/hostpath-replacement: '[{"name":"source-hostpath-volume-1","policyType":"remove"}]'
  ```
- 全部HostPath都忽略：
  ```
  bursting.cci.io/hostpath-replacement: '[{"name":"*","policyType":"remove"}]'
  ```
- 多个HostPath差异化替换策略：
  ```
  bursting.cci.io/hostpath-replacement: '[{"name":"source-hostpath-volume-1","policyType":"remove"},{"name":"source-hostpath-volume-3","policyType":"replaceByEmptyDir","emptyDir":{"sizeLimit":"10Gi"}}]'
  ```

  说明

  对于path为/etc/localtime的HostPath存储，会被单个HostPath替换的策略（策略name为具体的volume name）替换，不会被全部HostPath替换的策略（策略name为"*"）替换。

  参考deployment yaml示例：

  ```
  apiVersion: apps/v1
  kind: Deployment
  metadata:
    annotations:
      description: ''
    labels:
      bursting.cci.io/burst-to-cci: enforce
      appgroup: ''
      version: v1
    name: test
    namespace: default
  spec:
    replicas: 2
    selector:
      matchLabels:
        app: test
        version: v1
    template:
      metadata:
        labels:
          app: test
          version: v1
        annotations:
          bursting.cci.io/hostpath-replacement: '[{"name": "test-log2", "policyType": "remove"}, {"name": "test-log", "policyType": "replaceByEmptyDir", "emptyDir":{"sizeLimit":"10Gi"}}, {"name": "test-log1", "policyType": "remove" }]'
      spec:
        containers:
          - name: container-1
            image: nginx
            imagePullPolicy: IfNotPresent
            resources:
              requests:
                cpu: 250m
                memory: 512Mi
              limits:
  ```


版权所有 © 华为云计算技术有限公司

```
          cpu: 250m
          memory: 512Mi
      volumeMounts:
        - name: test-log
          mountPath: /tmp/log
        - name: test-log1
          mountPath: /tmp/log1
        - name: test-log2
          mountPath: /tmp/log2
  volumes:
  - hostPath:
      path: /var/paas/sys/log/virtual-kubelet
      type: ""
    name: test-log
  - hostPath:
      path: /var/paas/sys/log
      type: ""
    name: test-log1
  - hostPath:
      path: /var/paas/sys/log2
      type: ""
    name: test-log2
```

5.7 日志

5.7.1 通过 LTS 服务日志采集

CCI 2.0利用LTS日志服务采集应用日志并上报LTS，可以协助您排查和诊断问题。

日志采集可靠性说明

日志系统的核心功能在于记录业务组件的全生命周期状态数据（包括启动初始化、退出、运行时信息及异常事件等），主要服务于组件运行状态查看与故障根因分析等运维场景。

请注意标准输出流（stdout/stderr）及本地日志文件采用非持久化存储机制，其数据完整性受制于以下风险因素：

- 日志轮转压缩机制可能触发历史文件清除。
- Kubernetes Pod实例终止导致的临时存储卷回收。
- 节点存储空间限制触发的操作系统自动清理。

尽管云原生日志采集插件通过多级缓冲、优先级队列、断点续传等机制优化采集可靠性，但在以下场景仍存在日志采集丢失的可能：

- 业务日志吞吐量超过采集端处理能力。
- 业务Pod终止并立即被容器引擎回收。
- 日志采集器Pod运行异常。

以下是基于云原生日志管理的最佳实践建议，请您认真考虑并采纳：

- 请通过专用高可靠性通道记录并持久化关键业务数据（如金融交易）。
- 请勿在日志中进行记录客户信息、支付凭证、会话令牌等敏感数据。

版权所有 © 华为云计算技术有限公司

约束与限制

- CCI 2.0当前仅支持在西南-贵阳一、华东-上海一、华北-北京四局点使用日志管理。
- 不支持指定系统、设备、cgroup、tmpfs等挂载目录的日志采集。
- 对于单行过长的日志（当前配置为250KB）会直接跳过采集。
- 只支持完整路径和文件名正则匹配，不支持路径正则匹配。
- 同一个容器中所有需要采集的日志文件名不能重复，若重复只会采集到其中一个日志文件的日志。
- Pod启动后不支持配置更新，即配置更新后关联的Pod需重启后配置才能生效。
- 文件所在目录需要容器启动前存在，不支持启动后创建的目录及下层日志文件的日志上报。
- 日志文件的文件名，最大长度为190字符，超过长度限制的日志文件将不会被采集。
- 当前暂不支持initcontainer日志采集。
- 鲲鹏实例暂不支持日志采集。

步骤一 创建日志组

步骤1 登录管理控制台，选择“管理与部署 > 云日志服务 LTS”，进入“日志管理”页面。

步骤2 登录**云日志服务控制台**。

步骤3 进入“日志管理”页面，单击“创建日志组”。

步骤4 在“创建日志组”页面中，参考**表5-9**填写日志组相关信息。

表 5-9 日志组参数说明

参数	说明
日志组名称	日志组是云日志服务进行日志管理的基本单位，用于对日志流进行分类。如果日志较多，需要分门别类，建议您为日志组做好命名，方便后续快速查找日志。 LTS会自动生成一个默认的日志组名称，建议您根据业务自定义日志组的名称，创建成功后支持修改日志组名称。命名规范如下： • 日志组名称只支持输入英文、数字、中文、中划线、下划线及小数点，且不能以小数点、下划线开头或以小数点结尾。长度为1~64个字符。 • 日志组名称不能重复。
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。 默认选择default，建议根据业务选择企业项目，也可单击“查看企业项目”，在企业项目管理页面查看全部企业项目。 • 企业项目需要开通后才能使用，请参考**如何开通企业项目**。 • 支持将该企业项目资源迁出，详细请参考**迁出企业项目资源**。

版权所有 © 华为云计算技术有限公司

参数	说明
日志存储时间(天)	日志组的存储时间，即日志上报到LTS后日志存储的时间。 日志数据默认存储30天，可以在1~365天之间设置。 云日志服务LTS根据配置的日志存储时间定时清理日志内容，例如日志存储时间为30天，上报到LTS的日志只保存30天，30天后开始删除日志内容。 **说明** 目前白名单用户的日志存储时间支持1095天，如有需要，请提工单申请。详细操作请参考**提交工单** • 日志数据默认存储7天，可以在1~7天之间设置。超出存储时间的日志将会被自动删除，您可以按需将日志数据转储至OBS桶中长期存储。 • 上报到云日志服务的原始日志在达到日志存储时间后的次日凌晨就会被删除。
标签	按照业务需求对不同的日志组添加对应的标签。单击“添加”，分别填写标签键key和标签值value，开启应用到日志流后，日志组下的所有日志流即可自动加上该标签。如需添加多个标签可重复该步骤，最多支持添加20个标签。 **标签键key限制条件：** • 标签键可以包含任意语种的字母、数字和空格，以及_.:=+-@字符，但首尾不能包含空格，且不能以_sys_开头。 • 标签键长度不能超过128个字符。 • 标签键名称不可重复。 **标签值value限制条件：** • 标签值可以包含任意语种的字母、数字和空格，以及_.:=+-@字符。 • 标签值长度不能超过255个字符。 **标签策略：** 若您的组织已经设定云日志服务的相关标签策略，则需按照标签策略规则为日志组、日志流、日志接入、主机组、告警规则添加标签。标签如果不符合标签策略的规则，则可能会导致日志组、日志流、日志接入、主机组、告警规则创建失败，请联系组织管理员了解标签策略详情。标签策略详细介绍请参考**标签策略概述**，标签管理详细介绍请参考**标签管理**。 **删除标签：** 如需删除标签，请单击标签操作列的“删除”。 **警告** 删除标签无法恢复，请谨慎操作。 如果配置转储时使用了该标签，删除标签后，请同步修改转储配置信息。
备注	自定义填写备注信息，字符长度0-1024个字符。

步骤5 单击“确定”，日志组创建成功，即可在日志组列表下方生成一条日志组信息。

 版权所有 © 华为云计算技术有限公司

图 5-2 日志组

- 在日志组列表中，可以查看日志组名称、标签、日志流数量等信息。
- 单击日志组名称，可跳转到日志详情页面。

----结束

步骤二 创建日志流

1. 在云日志服务管理控制台，单击日志组名称对应的 按钮。
2. 单击展开页面左上角的“创建日志流”，输入日志流名称，名称需要满足如下要求：
   - 只支持输入英文、数字、中文、中划线、下划线及小数点，且不能以小数点、下划线开头或以小数点结尾。
   - 长度为1-64个字符。

   说明

   日志采集后，以日志流为单位，将多条日志数据发往云日志服务。如果日志较多，需要分门别类，建议您创建多个日志流，并给日志流做好命名，方便后续快速查找日志。
3. 在“企业项目”处选择业务需要的企业项目，也可单击“查看企业项目”，在企业项目管理页面查看全部企业项目。
4. 开启日志存储时间，使用日志流的日志存储时间，关闭日志存储时间，使用日志组的日志存储时间。
5. 开启日志存储时间后，根据需要是否开启智能冷存储。
6. 匿名写入默认关闭，适用于安卓/IOS/小程序/浏览器端上报日志，打开匿名写入则表示该日志流打开匿名写入权限，不会经过有效鉴权，可能产生脏数据。
7. 自定义设置标签值，按照“标签键=标签值”的形式填写，例如a=b。
8. 填写“备注”，字符长度0-1024个字符。
9. 单击“确定”，完成日志流的创建。在日志流列表中，可以查看日志流名称、操作等信息。

   说明

   - 支持查看日志流计费状态，日志计费请参考**价格计算器**。
   - 按日志流维度上报话单功能目前在友好用户内测中，您可以**提交工单**申请开通。

步骤三 获取日志组 ID 和日志流 ID

1. 在云日志服务管理控制台“日志管理”页面。
2. 选择**步骤一 创建日志组**所创建的日志组，在操作列选择“更多>详情”。
3. 获取并复制日志组ID。
4. 单击日志组名称对应的 按钮。

版权所有 © 华为云计算技术有限公司

5. 选择**步骤二 创建日志流**所创建的日志流，在操作列选择“详情”。
6. 获取并复制日志流ID。

步骤四 在 CCE 创建 Deployment

1. 登录**CCE控制台**。
2. 在左侧导航栏单击“负载管理”，选择“无状态负载”页签。
3. 单击“YAML创建”，使用**YAML**或**JSON**创建无状态负载。
   - 方法一 使用YAML创建无状态负载示例：

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test
  labels:
    bursting.cci.io/burst-to-cci: enforce
spec:
  replicas: 1
  selector:
    matchLabels:
      app: test
  template:
    metadata:
      annotations:
        logconf.k8s.io/fluent-bit-log-type: lts      # 必填项，使用TLS平台采集日志
        logconfigs.logging.openvessel.io: |
          {
            "default-config": {   #支持设置多个容器的日志收集路径（stdout.log表示标准输出，/root/out.log表示rootfs（包含挂卷）下文本日志，/data/emptydir-xxx/*.log表示rootfs（包含挂卷）下文本目录
              "container_files": {
                "container-0": "stdout.log;/root/out.log;/data/emptydir-volume/*.log"
              },
              "regulation": "", #多行日志收集正则匹配规则，请参考正则匹配规则
              "lts-log-info": {    #仅支持设置单个日志组和日志流
                "<日志组ID>": "<日志流ID>"   #将日志组ID、日志流ID替换为在步骤三 获取日志组ID和日志流ID中获取的日志组ID和日志流ID
              }
            }
          }
        resource.cci.io/pod-size-specs: 2.00_8.0
      labels:
        app: test
        sys_enterprise_project_id: "0"
    spec:
      containers:
      - image: busybox:latest
        imagePullPolicy: IfNotPresent
        command: ['sh', '-c', "while true; do echo hello; touch /root/out.log; echo hello >> /root/out.log; touch /data/emptydir-volume/emptydir.log; echo hello >> /data/emptydir-volume/emptydir.log; sleep 10; done"]
        lifecycle: {}
        volumeMounts:
        - name: emptydir-volume
          mountPath: /data/emptydir-volume
        - name: emptydir-memory-volume
          mountPath: /data/emptydir-memory-volume
        name: container-0
        resources:
          limits:
            cpu: 100m
            memory: 100Mi
          requests:
            cpu: 100m
            memory: 100Mi
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
```

 版权所有 © 华为云计算技术有限公司

```
    dnsPolicy: Default
    volumes:
    - name: emptydir-volume
      emptyDir: {}
    - name: emptydir-memory-volume
      emptyDir:
        sizeLimit: 1Gi
        medium: Memory
```

表 5-10 关键参数说明

关键参数	是否必选	参数类型	描述
logconf.k8s.io/fluent-bit-log-type	是	String	• 参数解释：日志采集方式 • 约束限制：必填项 • 取值范围：lts
logconfigs.logging.openvessel.io	是	String	• 参数解释：配置日志采集信息

说明

使用YAML示例时，“logconfigs.logging.openvessel.io”字段的相关参数注释需删除后再使用。

- 方法二 使用JSON创建无状态负载示例：

```
{
    "default-config": {
        "container_files": { // 支持设置多个容器的日志收集路径（stdout.log表示标准输出，/root/out.log表示rootfs（包含挂卷）下文本日志，/data/emptydir-xxx/*.log表示rootfs（包含挂卷）下文本目录）
            "container-0": "stdout.log;/root/out.log;/data/emptydir-volume/*.log",
            "container-1": "stdout.log"
        },
        "regulation": "", // 多行日志收集正则匹配规则，正则匹配规则参考https://docs.fluentbit.io/manual/pipeline/parsers/configuring-parser
        "lts-log-info": { // 仅支持设置单个日志组和日志流
            "日志组ID": "日志流ID"  // 将日志组ID、日志流ID替换为在步骤三 获取日志组ID和日志流ID中获取的日志组ID和日志流ID
        }
    },
    "multi-config": {
        "container_files": { // 支持设置多个容器的日志收集路径（stdout.log表示标准输出，/root/out.log表示rootfs（包含挂卷）下文本日志，/data/emptydir-xxx/*.log表示rootfs（包含挂卷）下文本目录）
            "container-0": "stdout.log;/root/out.log;/data/emptydir-memory-volume/*.log"
        },
        "regulation": "/(?<log>\\d+-\\d+-\\d+ \\d+:\\d+:\\d+.*)/",  // 多行日志收集正则匹配规则
        "lts-log-info": { // 仅支持设置单个日志组和日志流
            "日志组ID": "日志流ID"  // 将日志组ID、日志流ID替换为在步骤三 获取日志组ID和日志流ID中获取的日志组ID和日志流ID
        }
    }
}
```

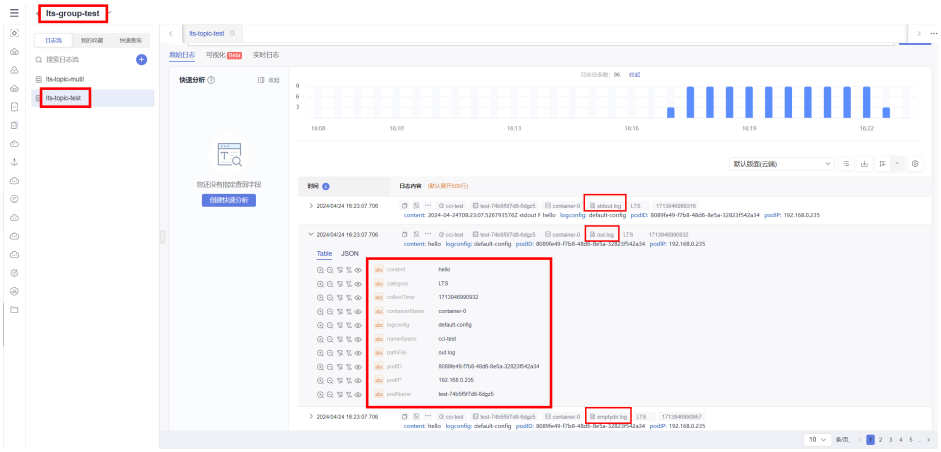

版权所有 © 华为云计算技术有限公司

> **注意**
>
> 在YAML和JSON创建过程中需要将“lts-log-info”参数中添加**步骤三 获取日志组ID和日志流ID**。示例：\"lts-log-info\":{\"日志组ID\":\"日志流ID\"}
>
> - 将日志组ID替换为**步骤三 获取日志组ID和日志流ID**。
> - 将日志流ID替换为**步骤三 获取日志组ID和日志流ID**。

4. 完成后单击“确定”。
5. 单击创建的无状态负载，可查看负载状态。

步骤五 查看日志组、日志流上报情况

1. 在云日志服务管理控制台，单击日志组名称。
2. 进入日志详情页面查看日志。



5.7.2 通过 Sidecar 采集日志

可通过hostpath-replacement将hostPath替换为CCI支持的emptyDir，通过pod-sidecars注入日志采集Sidecar，通过mount-stdlog-containers使日志采集Sidecar容器增加容器标准输出日志挂载。

前提条件

通过Sidecar采集日志前，您需要先参考**5.9.1**章节配置Sidecar工具容器。

约束与限制

“volume.cci.io/mount-stdlog-containers”和“volume.cci.io/mount-stdlog-containers-path”不支持同时配置。

通过 YAML 创建

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    version: v1
    bursting.cci.io/burst-to-cci: enforce
  name: deploy-example
```


版权所有 © 华为云计算技术有限公司

```
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: deploy-example
      version: v1
  template:
    metadata:
      annotations:
        bursting.cci.io/hostpath-replacement: '[{"name":"log","policyType":"replaceByEmptyDir","emptyDir":
{"sizeLimit":"10Gi"}}]'
        bursting.cci.io/pod-sidecars: '{"containers":[{"name":"log-sidecar","image":"fluent-
bit:latest","resources":{},"volumeMounts":[{"name":"log","mountPath":"/tmp/log"}],"position":"tail"}]}'
        volume.cci.io/mount-stdlog-containers: log-sidecar #指定需要挂载容器标准输出日志的容器名
      labels:
        app: deploy-example
        version: v1
    spec:
      containers:
      - image: nginx:latest
        name: nginx
        resources:
          limits:
            cpu: "1"
            memory: 2Gi
          requests:
            cpu: "1"
            memory: 2Gi
        volumeMounts:
        - mountPath: /tmp/log
          name: log
      imagePullSecrets:
        - name: default-secret
      volumes:
      - hostPath:
          path: /local/log
        name: log
```

通过以上方式配置，Sidecar工具容器可采集的日志可以分为以下两种：

- 标准输出。业务容器的标准输出日志会被挂载到/var/log/pods路径下，Sidecar可以以文件的方式收集业务容器的标准输出日志。
- 文件日志。业务容器将日志文件输出到和Sidecar共享的Volume中，Sidecar可以收集对应Volume内的文本文件。

版权所有 © 华为云计算技术有限公司

表 5-11 关键参数 Pod 的注解说明

Pod的注解	参数类型	描述	示例值
volume.cci.io/mount-stdlog-containers	String	1. 指定需要挂载容器标准输出日志的容器名，若有多个容器需要挂载则以逗号分隔。也可以配置"*"表示所有容器都进行挂载。如果配置了"*"则不可再配置其它容器名。 2. 匹配了该注解的容器，在启动后会将包含该Pod所有容器标准输出日志的目录挂载到容器内的/var/log/pods路径。	示例一： "container-0,container-1" 示例二： "*"
volume.cci.io/mount-stdlog-containers-path	String	1. 以json格式指定需要挂载容器标准输出日志的容器名和容器内挂载路径。容器名配置"*"表示所有容器都进行挂载。如果配置了"*"则不可再配置其它容器名。 2. 匹配了该注解的容器，在启动后会将包含该Pod所有容器标准输出日志的目录挂载到配置指定的路径。	示例一： "{\"container-0\":\"/var/log/pods\",\"container-1\":\"/tmp/log/pods\"}" 示例二： "{\"*\":\"/tmp/log/pods\"}"

5.8 配置 ClusterExtensionProfile 或 ExtensionProfile

对于将工作负载调度到CCI 2.0的场景，用户可以通过配置**ClusterExtensionProfile**或**ExtensionProfile**对象来调度到CCI 2.0的Pod，灵活使用*Cluster*级别的ClusterExtensionProfile，*Namespaced*级别的ExtensionProfile对调度到CCI 2.0的pod进行配置，从而减少对业务YAML的修改。

配置 ClusterExtensionProfile

步骤1 生成YAML文件。

YAML示例：

```
apiVersion: bursting.cci.io/v1
kind: ClusterExtensionProfile
metadata:
  name: test-cluster-profile
spec:
  actions:
    annotations:
      annotation1: value1
    hostPathReplacement:
    - emptyDir:
        sizeLimit: 10Gi
      name: volume1
```


版权所有 © 华为云计算技术有限公司

```
    policyType: replaceByEmptyDir
  labels:
    label1: value1
namespaceLabels:
  matchLabels:
    key2: value2
policy: override
```

步骤2 使用以下命令创建配置。

```
kubectl apply -f <test-cluster-profile>    #所创建的YAML文件名，替换为实际值
```

步骤3 （可选）使用以下命令查询是否配置成功。

```
kubectl get cextp
```

----结束

配置 ExtensionProfile

步骤1 生成YAML文件。

YAML示例：

```
apiVersion: bursting.cci.io/v1
kind: ExtensionProfile
metadata:
  name: test-profile
spec:
  actions:
    annotations:
      annotation1: value1
    hostPathReplacement:
    - emptyDir:
        sizeLimit: 10Gi
      name: volume1
      policyType: replaceByEmptyDir
    labels:
      label1: value1
  namespaceLabels:
    matchLabels:
      key2: value2
  policy: override
```

步骤2 使用以下命令创建配置。

```
kubectl apply -f <test-profile>    #所创建的YAML文件名，替换为实际值
```

步骤3 （可选）使用以下命令查询是否配置成功。

```
kubectl get extp
```

----结束

spec 配置说明

表 5-12 spec 参数说明

主要参数	参数说明
policy	支持配置策略为addOnly或override，如果某项配置与Pod原有定义冲突，配置为addOnly不会覆盖Pod原有定义，配置为override会覆盖Pod原有定义。默认配置为addOnly。
namespaceLabels字段	用于根据命名空间的Label筛选命名空间。

版权所有 © 华为云计算技术有限公司

主要参数	参数说明
objectLabels字段	用于根据Pod的Label筛选Pod。 **注意** matchLabels和matchExpressions的用法和匹配策略可参考K8S文档官方文档。

actions 参数说明：

表 5-13 actions 主要参数说明：

主要参数	参数说明
annotations	用于给Pod追加Annotations。
labels	用于给Pod追加Labels。
imageReplacement	用于给Pod添加**镜像地址转换策略**。
hostPathReplacement	用于给Pod添加**Hostpath卷转换策略**。
evictions	用于配置Pod磁盘压力驱逐策略。
podSidecars	用于给Pod添加**sidecar**。

profile 生效规则

- 仅会在Pod创建时对新建的Pod生效，不会修改已存在的Pod。
- 一个Pod最多仅会生效一个ClusterExtensionProfile和一个ExtensionProfile，若两种类型的profile都存在匹配上的，会先生效ClusterExtensionProfile，再生效ExtensionProfile。
- 若Pod所在Namespace的Labels能被多个ClusterExtensionProfile匹配，仅会生效一个最精确匹配的profile。
- 若Pod的Labels能被多个ExtensionProfile匹配，仅会生效一个最精确匹配的一个profile。
- 最精确匹配的定义：
  - profile的objectLabels或namespaceLabels中所有matchLabels和matchExpression的数量之和大的更精确。
  - 数量一样的选择profile name字母序最小的。

支持磁盘压力后 Pod 驱逐

在未使用**镜像快照**的场景下，当节点磁盘空间小于100MiB时，会为Pod上报NodeFsDiskPressure的Condition；在使用镜像快照的场景下，当节点临时存储磁盘空间小于100MiB时或容器镜像磁盘空间小于100MiB时，会分别上报NodeFsDiskPressure或ImageFsDiskPressure的Condition。当节点临时存储磁盘空大于100MiB时或容器镜像磁盘空间大于100MiB时，对应的Condition会被移除。

版权所有 © 华为云计算技术有限公司

```
status:
  conditions:
  - lastProbeTime: "2025-03-31T08:57:10Z"
    lastTransitionTime: "2025-03-31T08:57:10Z"
    message: 'The pod was low on resource: nodeFs ephemeral-storage. Threshold quantity:
      100Mi.'
    reason: PodHasNodeFsDiskPressure
    status: "True"
    type: NodeFsDiskPressure
  - lastProbeTime: null
    lastTransitionTime: "2025-03-31T08:45:21Z"
    status: "True"
    type: Initialized
  - lastProbeTime: null
    lastTransitionTime: "2025-03-31T08:45:39Z"
    status: "True"
    type: Ready
  - lastProbeTime: null
    lastTransitionTime: "2025-03-31T08:45:39Z"
    status: "True"
    type: ContainersReady
  - lastProbeTime: null
    lastTransitionTime: "2025-03-31T08:45:20Z"
    status: "True"
    type: PodScheduled
```

当上述磁盘空间不足的情况发生时，默认情况下，系统不会做任何处理，可能会影响业务运行。基于Kubernetes的驱逐和调度机制，通过对指定的ClusterExtensionProfile或ExtensionProfile配置驱逐策略，实现自动驱逐Pod。

一般情况下，业务通过Deployment的方式部署。如果被驱逐的Pod由Deployment管理，Deployment会自动创建新的Pod。

```
[root@test ~]# kubectl get pod -n default
NAME                          READY   STATUS              RESTARTS   AG
burst-disk-7b8c98b44-c25vr    1/1     Running             0          8m
burst-disk-7b8c98b44-t77l9    0/1     Evicted             0          16
```

在对应的ClusterExtensionProfile或ExtensionProfile增加内容如下，实现配置Pod驱逐策略：

```
actions:
  evictions:
    - type: "Hard"
      signal: "DiskPressure"
```

- 方式一：通过ClusterExtensionProfile配置指定命名空间标签的Pod驱逐策略

  ```
  apiVersion: bursting.cci.io/v1
  kind: ClusterExtensionProfile
  metadata:
    name: eviction-cluster-profile
  spec:
    actions:
      evictions:
        - type: "Hard"
          signal: "DiskPressure"
    namespaceLabels:
      matchLabels:
        key: value
  ```

- 方式二：通过ExtensionProfile配置在特定命名空间下的指定Label标识的Pod驱逐策略

  ```
  apiVersion: bursting.cci.io/v1
  kind: ExtensionProfile
  metadata:
    name: eviction-cluster-profile
    namespace: default
  spec:
  ```


版权所有 © 华为云计算技术有限公司

```
actions:
  evictions:
    - type: "Hard"
      signal: "DiskPressure"
objectLabels:
  matchExpressions:
  - key: key2
    operator: In
    values:
    - v2
    - v3
  - key: key3
    operator: Exists
  matchLabels:
    key1: value1
policy: addOnly
```

5.9 使用 Sidecar 工具容器

5.9.1 配置 Sidecar 工具容器

应用场景

使用CCE突发弹性引擎（对接CCI）插件弹性到CCI 2.0支持在Pod中自动注入客户的Sidecar工具容器的能力。

示例：如果客户需要使用CCE突发弹性引擎（对接CCI）插件弹性到CCI 2.0时将采集的日志上报至客户的日志中心，可以设置自动注入客户的日志采集工具容器策略。

您可参考以下几种方式配置Sidecar工具容器：

- **方式一 通过注解配置Sidecar**
- **方式二 通过ClusterExtensionProfile配置Sidecar**
- **方式三 通过ExtensionProfile配置Sidecar**

前提条件

如果ClusterExtensionProfile或ExtensionProfile方式配置Sidecar工具容器，您需要先配置ClusterExtensionProfile或ExtensionProfile，操作步骤请参考**5.8**。

约束与限制

- 支持注入多个InitContainer、Container，注入的Container不能与原有Pod中其他Container重名。
- 支持注入多个Volume，Volume仅能配置CCI支持的类型，注入的Volume不能与原有pod.spec.volumes中的重名。
- 通过ClusterExtensionProfile或者ExtensionProfile方式配置时：
  - 不支持同时在actions.podSidecars和actions.annotations配置键名为bursting.cci.io/pod-sidecars的注解。
  - profile匹配到的Pod，若已存在键为bursting.cci.io/pod-sidecars的注解，则根据profile的policy决定覆盖或忽略。

版权所有 © 华为云计算技术有限公司

方式一 通过注解配置 Sidecar

通过在Pod.Annotations中加入注解可以使弹至CCI的Pods添加Sidecar Container。

```
bursting.cci.io/pod-sidecars: '{
"containers":[{"name":"container-sidecar1","image":"busybox:latest","command":["sh","-c","sleep
3600"],"position":"head"}],
"initContainers":[{"command":["sh","-c","sleep
10"],"image":"busybox:latest","name":"initbusybox","position":"tail","resources":{},"volumeMounts":
[{"mountPath":"/tmp/auth","name":"mysecret"}]}],
"volumes":[{"name":"mysecret","secret":{"secretName":"testsecret"}}]}'
```

方式二 通过 ClusterExtensionProfile 配置 Sidecar

ClusterExtensionProfile的YAML模板如下：

```
apiVersion: bursting.cci.io/v1
kind: ClusterExtensionProfile
metadata:
  name: podsidecar-cluster-profile
spec:
  actions:
    podSidecars:
      initContainers:
      - command:
        - sh
        - -c
        - sleep 10
        image: busybox:latest
        name: initbusybox
        position: tail
        resources: {}
        volumeMounts:
        - mountPath: /tmp/auth
          name: mysecret
      volumes:
      - name: mysecret
        secret:
          secretName: testsecret
  namespaceLabels:
    matchLabels:
        key: value
```

方式三 通过 ExtensionProfile 配置 Sidecar

ExtensionProfile的YAML模板如下：

```
apiVersion: bursting.cci.io/v1
kind: ExtensionProfile
metadata:
  name: podsidecar-profile
  namespace: default
spec:
  actions:
    podSidecars:
      containers:
      - command:
        - sh
        - -c
        - sleep 3600
        image: busybox:latest
        name: busybox
        volumeMounts:
        - mountPath: /tmp/auth
          name: mysecret
      volumes:
      - name: mysecret
        secret:
          secretName: testsecret
```


版权所有 © 华为云计算技术有限公司

```
objectLabels:
  matchLabels:
    key1: value1
```

参数说明

表 5-14 关键参数说明

参数	参数说明
InitContainers	注入到CCI侧pod.spec.initContainers的initContainer列表。InitContainer对象的结构与Pod中initContainer的结构类似，不支持配置restartPolicy，并增加了position、envFromContainer、volumeMountFromContainer参数，详情请参见**表5-15**。
Containers	注入到CCI侧pod.spec.containers的Container列表。详情请参见**表5-15**。
Volumes	注入到CCI侧pod.spec.volumes的volume列表。

表 5-15 Container 参数描述

参数	是否必选	参数类型	描述
position	否	String	可通过为container配置position字段指定注入位置，配置为head注入在头部，配置为tail注入在尾部，默认注入在尾部，相同position的container注入的先后顺序与actions.podSidecars配置的顺序一致。

5.9.2 复制业务容器环境变量和卷挂载点

使用场景

在容器的日志收集与处理、配置管理、数据同步等场景中，可以通过环境变量传递配置文件的路径，或者通过共享卷挂载点将主业务容器的卷挂载到Sidecar容器中来实现特定功能。云容器实例提供了复制业务容器的环境变量（env）和卷挂载点（volumeMounts）的能力，从而简化了这些操作并实现特定功能。

您可参考以下几种方式复制业务容器环境变量和卷挂载点：

- **方式一 通过注解复制业务容器环境变量和卷挂载点**
- **方式二 通过ClusterExtensionProfile复制业务容器环境变量和卷挂载点**
- **方式三 通过ExtensionProfile复制业务容器环境变量和卷挂载点**

前提条件

如果使用ClusterExtensionProfile或ExtensionProfile方式复制业务容器环境变量和卷挂载点，您需要先配置Sidecar工具容器，操作步骤请参考**配置Sidecar工具容器**。

版权所有 © 华为云计算技术有限公司

功能规格

表 5-16 功能规格

功能描述	功能说明
注入Sidecar时支持复制业务容器env的能力	• 通过容器名和环境变量名来复制环境变量。 • 通过容器序号和环境变量名来复制环境变量。
注入Sidecar时支持复制业务容器volumeMount的能力	• 通过容器名和volumeMount名来复制卷挂载点。 • 通过容器序号和volumeMount名来复制卷挂载点。

约束与限制

表 5-17 功能约束

功能字段	字段约束
envFromContainer	• envFromContainer的name不能为空。 • envFromContainer的name不能重复或与sidecarContainer的env配置中的env名重复。 • containerFieldRef中的name和index至少配置其中一个。 • containerFieldRef的fieldPath须为"name"或以"env."开头。
volumeMountFromContainer	• volumeMountFromContainer的volumeMountName不能为空。 • volumeMountFromContainer的volumeMountName不能重复或与sidecarContainer的volumeMount配置中的name重复。 • containerRef中的name和index至少配置其中一个。

方式一 通过注解复制业务容器环境变量和卷挂载点

通过在Pod.Annotations中加入注解添加envFromContainer、volumeMountFromContainer复制业务容器的环境变量和卷挂载点。

```
bursting.cci.io/pod-sidecars: '{"containers":[{"name":"busybox","image":"busybox:latest","command":["sh","-c","sleep 3600"],"resources":{},"envFromContainer":[{"name":"POD_IP","containerFieldRef":{"index":0,"fieldPath":"env.POD_IP"}},{"name":"POD_NAME","containerFieldRef":{"name":"nginx","fieldPath":"env.POD_NAME"}},{"name":"CONTAINER_NAME","containerFieldRef":{"index":0,"fieldPath":"name"}}],"volumeMountFromContainer":[{"volumeMountName":"log-storage","containerRef":{"index":0}},{"volumeMountName":"log-storage1","containerRef":{"name":"nginx"}}]}]}'
```


版权所有 © 华为云计算技术有限公司

方式二 通过 ClusterExtensionProfile 复制业务容器环境变量和卷挂载点

步骤1 生成YAML文件。通过ClusterExtensionProfile来配置Sidecar容器的环境变量和卷挂载点。

YAML示例：

```
apiVersion: bursting.cci.io/v1
kind: ClusterExtensionProfile
metadata:
  name: podsidecar-cluster-profile
spec:
  actions:
    podSidecars:
      containers:
      - command:
        - sh
        - -c
        - sleep 3600
        image: busybox:latest
        name: busybox
        envFromContainer:
        - containerFieldRef:
            fieldPath: env.POD_IP
            index: 0
          name: POD_IP
        - containerFieldRef:
            fieldPath: env.POD_NAME
            name: nginx
          name: POD_NAME
        - containerFieldRef:
            fieldPath: name
            index: 0
          name: CONTAINER_NAME
        volumeMountFromContainer:
        - containerRef:
            index: 0
          volumeMountName: log-storage
        - containerRef:
            name: nginx
          volumeMountName: log-storage1
  namespaceLabels:
    matchLabels:
      key: value
```

步骤2 使用以下命令创建配置。

```
kubectl apply -f <podsidecar-cluster-profile>    #所创建的YAML文件名，替换为实际值
```

步骤3 （可选）查询是否配置成功。

```
kubectl get cextp
```

----结束

方式三 通过 ExtensionProfile 复制业务容器环境变量和卷挂载点

步骤1 生成YAML文件。通过ExtensionProfile来配置Sidecar容器的环境变量和卷挂载点。

YAML示例：

```
apiVersion: bursting.cci.io/v1
kind: ExtensionProfile
metadata:
  name: podsidecar-profile
spec:
  actions:
    podSidecars:
```


版权所有 © 华为云计算技术有限公司

```
    containers:
    - command:
      - sh
      - -c
      - sleep 3600
      image: busybox:latest
      name: busybox
      envFromContainer:
      - containerFieldRef:
          fieldPath: env.POD_IP
          index: 0
        name: POD_IP
      - containerFieldRef:
          fieldPath: env.POD_NAME
          name: nginx
        name: POD_NAME
      - containerFieldRef:
          fieldPath: name
          index: 0
        name: CONTAINER_NAME
      volumeMountFromContainer:
      - containerRef:
          index: 0
        volumeMountName: log-storage
      - containerRef:
          name: nginx
        volumeMountName: log-storage1
  objectLabels:
    matchLabels:
      key1: value1
```

步骤2 使用以下命令创建配置。

```
kubectl apply -f <podsidecar-profile>    #所创建的YAML文件名，替换为实际值
```

步骤3 （可选）使用以下命令查询是否配置成功。

```
kubectl get extp
```

----结束

参数说明

表 5-18 关键参数说明

参数	参数说明
InitContainers	注入到CCI侧pod.spec.initContainers的initContainer列表。InitContainer对象的结构与Pod中initContainer的结构类似，不支持配置restartPolicy，并增加了position、envFromContainer、volumeMountFromContainer参数，详情请参见**表5-19**。
Containers	注入到CCI侧pod.spec.containers的Container列表。详情请参见**表5-19**。
Volumes	注入到CCI侧pod.spec.volumes的volume列表。

版权所有 © 华为云计算技术有限公司

表 5-19 Container 参数描述

参数	是否必选	参数类型	描述
position	否	String	可通过为container配置position字段指定注入位置，配置为head注入在头部，配置为tail注入在尾部，默认注入在尾部，相同position的container注入的先后顺序与actions.podSidecars配置的顺序一致。

5.10 网络

5.10.1 配置网络互通能力

约束与限制

- 使用共享VPC的CCE集群不支持开启“网络互通”功能。
- 暂不支持HostNetwork网络模式。
- 跨CCE和CCI实例Service网络互通只支持集群内访问（ClusterIP）类型。
- 当前网络互通能力依赖Sidecar容器的启动，如需使用以下功能，插件升级到1.5.42及以上版本：
  - 业务容器需要使用postStart功能。
  - 在初始化容器中使用网络互通能力，需要开启**初始化容器网络互通开关**。
- 开启网络互通能力、开启初始化容器网络互通开关等功能，等待插件滚动升级完成，即插件再次处于运行中状态时，再下发负载，否则会导致功能异常。
- 使用CCE集群中的Bursting插件对接CCI 2.0服务，支持配置独享型ELB的Ingress和Service。Bursting插件1.5.5以下版本不支持配置ELB类型的Service。
- 跨CCE和CCI实例，在对接LoadBalancer类型的Service或Ingress时：
  - 禁止健康检查配置重新指定端口。在CCE集群下，由于CCI的容器与CCE的容器在ELB注册的后端使用端口不一致，指定健康检查端口会导致部分后端健康检查异常。
  - 跨CCE集群使用Service对接同一个ELB的监听器时，需确认健康检查方式，避免服务访问异常。
  - 跨CCE和CCI实例，在对接共享型LoadBalancer类型的Service时，需要放通节点安全组下100.125.0.0/16网段中容器使用的端口。

CCE 集群 Pod 与 CCI 集群中 Pod 通过 Service 互通的使用指导

步骤1 安装插件，勾选“网络互通”功能。

版权所有 © 华为云计算技术有限公司

步骤2 安装成功后租户账号下会自动创建负载均衡器，可以通过网络控制台进行查看。

步骤3 创建弹性CCI侧pod，并配置Service发布。

- 为了方便验证，镜像选择暴露容器80端口的nginx。

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: burst-nginx-service
  labels:
    app: burst-nginx-service
    bursting.cci.io/burst-to-cci: enforce
spec:
  replicas: 1
  selector:
    matchLabels:
      app: burst-nginx-service
  template:
    metadata:
      labels:
        app: burst-nginx-service
    spec:
      containers:
      - name: nginx
        image: [illegible]nginx
        ports:
        - containerPort: 80
        resources:
            limits:
              cpu: 250m
              memory: 512Mi
            requests:
              cpu: 250m
              memory: 512Mi
```

- 配置Service，引用工作负载的标签。推荐自动创建新的负载均衡器，以避免和插件自动创建的ELB产生冲突。

版权所有 © 华为云计算技术有限公司

步骤4 通过CCE控制台，获取该负载的访问方式。

步骤5 创建CCE侧Pod，并配置Service发布。方法请参考步骤**步骤3**，不添加弹性CCI标签(即无需在metadata.labels配置bursting.cci.io/burst-to-cci标签)。

步骤6 验证网络互访。

- 通过CCI侧进入该容器，访问CCE侧pod的Service地址，其形式为<service-name>.<namespace-name>.svc.cluster.local:port，观测CCI访问CCE Service网络打通。

 版权所有 © 华为云计算技术有限公司

图 5-3 访问 CCE pod 的 Service

```
Welcome to Cloud Container Instance.
sh-4.2# curl -kv testcce-ng.default.svc.cluster.local:30001
* About to connect() to testcce-ng.default.svc.cluster.local port 30001 (#0)
*   Trying 10          3...
^C
sh-4.2# curl -kv testcce-ng.default.svc.cluster.local:30002
* About to connect() to testcce-ng.default.svc.cluster.local port 30002 (#0)
*   Trying 10.247.90.173...
* Connected to testcce-ng.default.svc.cluster.local (10.247.90.173) port 30002 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.29.0
> Host: testcce-ng.default.svc.cluster.local:30002
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.16.0
< Date: Thu, 11 Jan 2024 06:37:40 GMT
< Content-Type: text/html
< Content-Length: 612
< Last-Modified: Tue, 23 Apr 2019 15:11:11 GMT
< Connection: keep-alive
< ETag: "5cbf2b0f-264"
```

- 同理进入CCE侧Pod。访问CCI侧Pod的Service地址。观测CCE访问CCI Service网络打通。

图 5-4 访问 CCI pod 的 Service

```
sh-4.2#
sh-4.2#
sh-4.2#
sh-4.2# curl -kv testcci-ng.default.svc.cluster.local:30001
* About to connect() to testcci-ng.default.svc.cluster.local port 30001 (#0)
*   Trying 10.247.174.50...
* Connected to testcci-ng.default.svc.cluster.local (10.247.174.50) port 30001 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.29.0
> Host: testcci-ng.default.svc.cluster.local:30001
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.16.0
< Date: Thu, 11 Jan 2024 06:39:17 GMT
< Content-Type: text/html
< Content-Length: 612
< Last-Modified: Tue, 23 Apr 2019 15:11:11 GMT
< Connection: keep-alive
< ETag: "5cbf2b0f-264"
```

----结束

初始化容器网络互通开关配置

初始化容器网络互通开关默认关闭，若要使用，请在开启网络互通功能开关的前提下，根据以下操作进行配置。

步骤1 登录CCE控制台。

步骤2 选择CCE集群，单击进入CCE集群总览页面。

步骤3 在导航栏左侧单击“插件中心”，进入插件中心首页。

步骤4 选择“CCE 突发弹性引擎 (对接 CCI)”插件，单击“编辑”。

版权所有 © 华为云计算技术有限公司

图 5-5 CCE 突发弹性引擎 (对接 CCI)插件

步骤5 在“网络互通（可选）”开启的条件下，单击“编辑YAML”。

图 5-6 编辑插件

步骤6 将“set_proxy_as_first_initcontainer”配置为“true”。

版权所有 © 华为云计算技术有限公司

图 5-7 修改参数

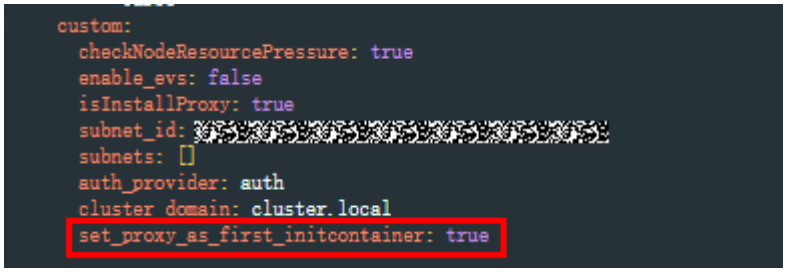


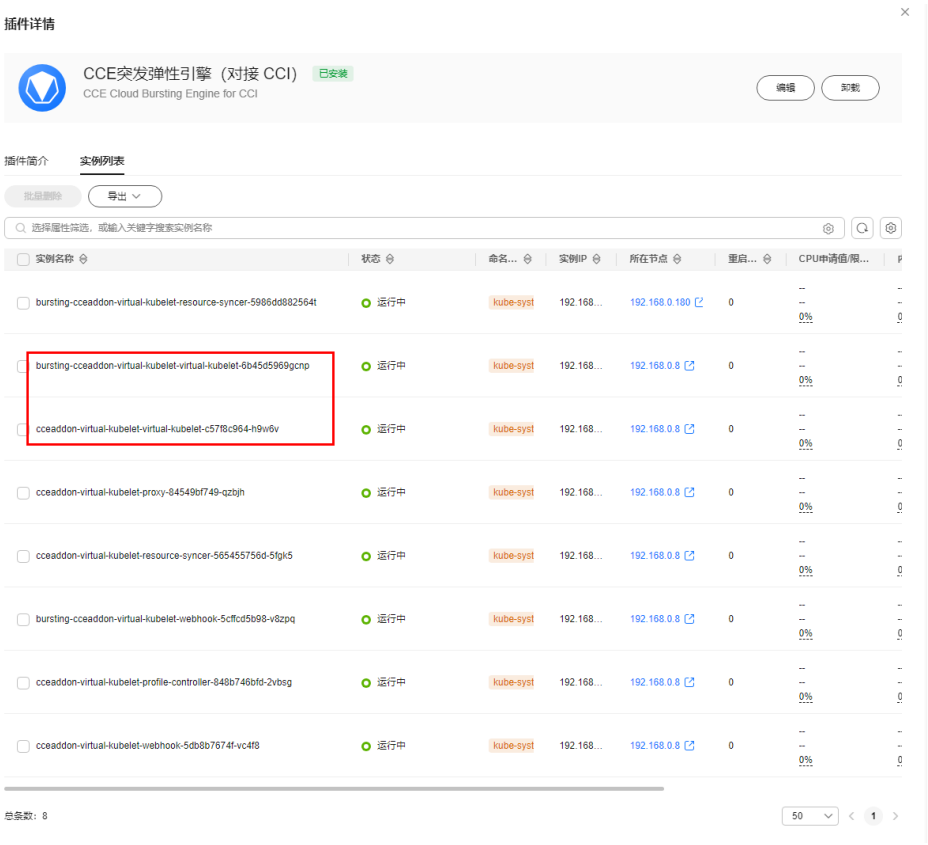
表 5-20 参数说明

参数	是否必选	取值类型	参数说明
set_proxy_as_first_initcontainer	否	Bool	参数解释：表示初始化容器网络互通功能是否开通。 取值范围：默认为false。 ● false：表示关闭初始化容器网络互通功能。 ● true：表示开启初始化容器网络互通功能。

步骤7 在“CCE 突发弹性引擎 (对接 CCI)”插件的实例列表中查看实例名称为“bursting-cceaddon-virtual-kubelet-virtual-kubelet-xxx”、“cceaddon-virtual-kubelet-virtual-kubelet-xxx”状态显示为“运行中”，表示重新部署完毕，该功能可正常运行。

版权所有 © 华为云计算技术有限公司

图 5-8 查看插件部署状态



----结束

5.10.2 配置默认使用的指定 DNS 服务器

如果用户需要弹性CCI的Pod可以使用指定的DNS服务器地址。bursting插件提供配置指定的DNS服务器地址的能力，无需在每个Pod上都配置dnsConfig，从而降低用户网络运维成本。

步骤1 登录CCE控制台。

步骤2 选择CCE集群，单击进入CCE集群总览页面。

步骤3 在导航栏左侧单击“插件中心”，进入插件中心首页。

步骤4 选择“CCE 突发弹性引擎 (对接 CCI)”插件，单击“编辑”。

步骤5 单击“编辑YAML”。

步骤6 登录CCE集群节点，编辑bursting负载的YAML。

```
kubectl edit deploy bursting-cceaddon-virtual-kubelet-virtual-kubelet -n kube-system
```

步骤7 启动参数中增加--cluster-dns=x.x.x.x参数，配置为dns服务器地址。

步骤8 保存配置修改，等待bursting-virtual-kubelet负载重启。

版权所有 © 华为云计算技术有限公司

```
[root@test ~]# kubectl get pod -n kube-system
NAME                                                      READY   STATUS    RESTARTS      AGE
bursting-cceaddon-virtual-kubelet-resource-syncer-566c9844tpgpm   1/1     Running   0             2m20s
bursting-cceaddon-virtual-kubelet-virtual-kubelet-697fd69fnrq8p   1/1     Running   0             35s
bursting-cceaddon-virtual-kubelet-webhook-5847c4f7c6-dftvz        1/1     Running   0             2m20s
cceaddon-virtual-kubelet-profile-controller-5858c6cfc5-ssh5x      1/1     Running   0             2m20s
cceaddon-virtual-kubelet-proxy-76675d6b6-t2s4v                    1/1     Running   0             2m20s
cceaddon-virtual-kubelet-resource-syncer-69fd8bb959-2c65k         1/1     Running   0             2m20s
cceaddon-virtual-kubelet-virtual-kubelet-7d6b44d9f-6sjs8          1/1     Running   0             2m20s
cceaddon-virtual-kubelet-webhook-d898d97dd-mnwfq                  1/1     Running   0             2m20s
```

步骤9 验证方式。通过exec进入弹性到CCI运行中的容器内，查看/etc/resolv.conf中首选nameserver是否为cluster-dns配置的地址。

表 5-21 使用场景约束限制

使用场景	约束限制
修改配置前存在弹性CCI的pod	• 修改配置后新创建弹性CCI的pod生效。 • 修改配置前的弹性的pod需要重建之后才生效。
cluster-dns配置上限	• k8s dnsConfig最多只支持配置3个nameservers。 • 保证cluster-dns配置的nameserver数量与Pod的spec.dnsConfig配置的nameserver数量之和不超过3个。

----结束

5.10.3 配置子网

约束与限制

- 配置子网最终为插件YAML中配置的spec.custom.subnet_id和spec.custom.subnets的合集。
- 如果需要删除子网，不能直接删除spec.custom.subnet_id，必须使用另外一个子网进行替换。

为命名空间配置扩展子网

Bursting插件关联的CCI侧命名空间默认使用插件创建时选择的子网，如果要使用其他子网，请根据操作进行配置。

步骤1 登录CCE控制台。

步骤2 选择CCE集群，单击进入CCE集群总览页面。

步骤3 在导航栏左侧单击“插件中心”，进入插件中心首页。

步骤4 选择“CCE 突发弹性引擎 (对接 CCI)”插件，单击“编辑”。

步骤5 单击“编辑YAML”。

步骤6 以如下格式配置spec.custom的subnets数组，subnetID为当前CCE集群VPC下其他子网的IPv4子网ID。

配置示例：

```
custom:
  subnet_id: 11c2a970-xxxx-xxxx-xxxx-dfxxxxxxxx7d
```


版权所有 © 华为云计算技术有限公司

```
subnets: [
{
    "subnetID": "efd70252-xxxx-xxxx-xxxx-72xxxxxxxxde"
},
{
    "subnetID": "55038f35-xxxx-xxxx-xxxx-ecxxxxxxxx49"
},
{
    "subnetID": "99b2a9f5-xxxx-xxxx-xxxx-64xxxxxxxx5a"
}
]
```

表 5-22 配置说明

参数	是否必选	取值类型	参数说明
subnet_id	是	String	表示安装插件时选择的默认子网的IPv4子网ID，且默认子网数量为1。
subnets	否	Array of subnet	表示命名空间的扩展子网数组

表 5-23 subnet 字段数据结构说明

参数	是否必选	参数类型	描述
subnetID	是	String	扩展子网的IPv4子网ID

步骤7 单击“提交”。在“CCE 突发弹性引擎 (对接 CCI)”插件的实例列表中查看实例名称状态均为“运行中”，表示部署完毕，该功能可正常运行。

----结束

5.11 监控

约束与限制

从CCE bursting弹到CCI 2.0的pod，请确保在CCE与CCI 2.0两个服务中创建的命名空间及Pod名称不完全相同，以避免CCI 2.0端出现Pod监控数据异常的情况。

在 CCE 控制台查看 Pod 监控

步骤1 登录CCE控制台。

步骤2 选择CCE集群，单击进入CCE集群总览页面。

步骤3 在导航栏左侧单击“监控中心”，进入监控中心首页。

步骤4 单击“立即开通”，详情可参见**开通监控中心**。

版权所有 © 华为云计算技术有限公司

步骤5 在监控中心”页面单击“Pod”页签。Pod列表页面呈现了所有Pod的综合信息，如需深入了解单个Pod的监控情况，可单击Pod名称，进入该Pod的“概览”页面，通过切换“容器列表”、“监控”页签查看相应内容，详情可参见**Pod监控**。

----结束

5.12 弹性伸缩

当使用bursting插件弹性pod到CCI 2.0上时，metrics-server插件无法采集这部分Pod的指标数据，可能会影响HPA工作。通过阅读本章节，您可以使用云原生监控插件替换metrics-server插件，以使HPA功能正常。

操作步骤

步骤1 安装“云原生监控插件”。

1. 登录CCE控制台。
2. 选择CCE集群，单击进入CCE集群总览页面。
3. 在导航栏左侧单击“插件中心”，进入插件中心首页。
4. 选择“云原生监控插件”，单击“安装”，跳转到安装详情界面。

图 5-9 安装云原生监控插件

5. 数据存储配置开启本地数据存储。
6. 单击安装，安装插件。

步骤2 通过云原生监控插件，提供系统资源指标。

- 如果CCE集群中未安装metrics-server插件，请开启Metric API，详情请参见**通过Metrics API提供资源指标**。配置完成后，可使用Prometheus采集系统资源指标。
- 如果CCE集群中已安装metrics-server插件，可以选择以下任意一种方式进行处理：
  - **方式一：请卸载“Kubernetes Metrics Server”插件后，开启Metric API。**
    i. 登录CCE控制台。
    ii. 选择CCE集群，单击进入CCE集群总览页面。
    iii. 在导航栏左侧单击“插件中心”，进入插件中心首页。
    iv. 选择“Kubernetes Metrics Server”插件，单击“卸载”。

版权所有 © 华为云计算技术有限公司

v. 开启Metric API，详情请参见**通过Metrics API提供资源指标**。配置完成后，可使用Prometheus采集系统资源指标。

- **方式二：修改APIService对象。**

  需要更新APIService对象“v1beta1.metrics.k8s.io”：

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

  可以将该对象保存为文件，命名为metrics-apiservice.yaml，然后执行以下命令：

```
kubectl apply -f metrics-apiservice.yaml
```

  执行kubectl top pod -n monitoring命令，如果显示如下，则表示Metrics API能正常访问：

```
# kubectl top pod -n monitoring
NAME                                          CPU(cores)   MEMORY(bytes)
......
custom-metrics-apiserver-d4f556ff9-l2j2m                38m          44Mi
......
```

  须知

  卸载插件时，需要执行以下kubectl命令，同时删除APIService对象，否则残留的APIService资源将导致metrics-server插件安装失败。

```
kubectl delete APIService v1beta1.metrics.k8s.io
```

步骤3 创建HPA策略

1. 登录CCE控制台。
2. 选择CCE集群，单击进入CCE集群总览页面。

版权所有 © 华为云计算技术有限公司

3. 在导航栏左侧单击“策略”。
4. 选择“弹性伸缩策略”。
5. 单击右侧的创建“HPA策略”。

6. 如配置cpu利用率规则，提高对应pod的cpu使用率，可看到HPA进行扩容操作。

----结束

5.13 多虚拟节点配置

功能说明

- 支持在集群中存在1个全局虚拟节点和多个AZ级虚拟节点，调度到全局节点上的Pod在CCI侧会在所有支持的可用区中随机调度，调度到AZ级虚拟节点上的Pod在CCI侧只会被调度到对应的一个AZ中。如果您不想要CCI侧随机调度，您需要手动将全局级别的虚拟节点（即bursting-node节点）设为不可调度状态。
- AZ级虚拟节点具有`failure-domain.beta.kubernetes.io/zone`和`topology.kubernetes.io/zone`两个Label，值为AZ名，可让Pod根据该节点Label来调度。
- 全局虚拟节点默认存在，不可删除，AZ级虚拟节点可以根据configMap配置动态增减。
- 支持配置节点的可调度CPU、Memory资源（capacity、allocatable）。
- 自插件1.5.29版本起，CCI服务会在AZ故障时将对应的虚拟节点设置为不可调度，租户可以查看虚拟节点的status.conditions信息感知对应AZ是否故障以及故障原因。当CCI将虚拟节点设置为不可调度时，表示租户的存量业务可能受损，新业务不应继续往对应的虚拟节点调度。

约束与限制

- 不带“.”的键，键的名称即为AZ名（不能配置为“global”，global意为全局节点），其值配置为"true"表示启用该AZ对应的虚拟节点。
- 带“.”的键，第一个“.”之前为AZ名，”.”之后为节点参数，当前支持“cpu”、“memory”，值与k8s resource规则一致（支持k、Mi、Gi等单位）。“cpu”、“memory”为可选配置，不配置则会为较大的默认值（cpu：200k，memory：1600000Gi）。
- 键只能为小写，且需要满足RFC 1123 subdomain格式（同节点名规则），AZ名最长支持47个字符。

版权所有 © 华为云计算技术有限公司

- AZ级虚拟节点的数量最多支持20个。
- Bursting检测AZ故障周期默认是5分钟，用户可以通过设置resource-syncer的启动参数'--az-status-sync-period'参数自行决定检测周期，最小为1min。

配置样例

在kube-system命名空间下配置如下ConfigMap

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: bursting-node
  namespace: kube-system
data:
  global.cpu: 2k          # global代表全局虚拟节点，全局虚拟节点默认创建，不可删除，可选配置节点的可用cpu、memory资源，不配置则为较大的默认值（cpu：200k，memory：1600000Gi）
  global.memory: 4000Gi
  region-a: "true"      # 如添加了键等于AZ名的配置项，且值为"true"，则会自动创建出对应AZ的虚拟节点
  region-a.cpu: 5k      # cpu、memory为可选配置，不配置则为较大的默认值（cpu：200k，memory：1600000Gi）
  region-a.memory: 8000Gi
  region-b: "true"
```

如上配置则集群中预期存在3个节点

- bursting-node
- bursting-node-region-a
- bursting-node-region-b

5.14 常见问题

问题一：弹性 CCI 功能不可用

问题原因：用户CCE集群所在子网与10.247.0.0/16重叠，与CCI命名空间下的Service网段冲突。

解决方案：重新规划CCE集群子网网段。

问题二：插件由 1.5.18 及以上版本回退至低于 1.5.18 后，Pod 通过 Service 访问出现异常

问题原因：插件升级到1.5.18及以上版本之后，新弹性到CCI的Pod中的sidecar无法兼容1.5.18以下版本的插件，因此插件回退版本后会导致这些Pod中的Service访问异常。插件在低于1.5.18版本时弹性到CCI的Pod不受影响。

解决方案：

- 方案一：将插件再升级到1.5.18及以上版本。
- 方案二：将Service访问异常的Pod删除重建，重建后弹性到CCI的Pod Service访问将恢复正常。

问题三：插件无法删除

问题场景：因为误修改swr_addr、swr_user导致的插件卸载失败。

版权所有 © 华为云计算技术有限公司

问题原因：插件卸载依赖gc-job执行，镜像拉取失败场景gc-job无法运行成功，导致卸载失败。

解决方案：再次进行卸载操作，依次删除gc-job。

1. 插件处于删除失败状态时，先登录至CCE集群配置有kubectl的节点后，再次单击“卸载”。
2. 在210秒内执行以下命令：
   a. 删除resource-gc-jobs。
   ```
   kubectl get job -nkube-system | grep "virtual-kubelet-.*-resource-gc-jobs"
   kubectl delete job -nkube-system xxx
   ```
   ```
   # kubectl get job -nkube-system | grep "virtual-kubelet-.*-resource-gc-jobs"
   virtual-kubelet-0psor2fajaltllvjfc2b-resource-gc-jobs   Running   0/1           41s        41s
   # kubectl delete job -nkube-system virtual-kubelet-0psor2fajaltllvjfc2b-resource-gc-jobs
   job.batch "virtual-kubelet-0psor2fajaltllvjfc2b-resource-gc-jobs" deleted
   ```
   b. 删除namespace-gc-jobs。
   ```
   kubectl get job -nkube-system | grep "virtual-kubelet-.*-namespace-gc-jobs"
   kubectl delete job -nkube-system yyy
   ```
   ```
   # kubectl get job -nkube-system | grep "virtual-kubelet-.*-namespace-gc-jobs"
   virtual-kubelet-u91uzeflrw7f6v9rrdfi-namespace-gc-jobs   Running   0/1           30s        30s
   # kubectl delete job -nkube-system virtual-kubelet-u91uzeflrw7f6v9rrdfi-namespace-gc-jobs
   job.batch "virtual-kubelet-u91uzeflrw7f6v9rrdfi-namespace-gc-jobs" deleted
   ```
3. 其余异常场景请提交工单协助处理。

问题四：依赖 Service 访问的业务容器启动失败

问题场景：业务容器启动过程或postStart配置依赖访问Service，出现第一次启动失败，待Sidecar容器启动后，重新启动又成功的现象。

问题原因：CCI侧pod访问Service依赖Sidecar容器，未完成Service同步时，业务容器访问Service会失败，同步完成后业务容器正常启动。

解决方案：请将插件升级至1.5.28及以上版本。

版权所有 © 华为云计算技术有限公司

6 运维管理

6.1 配置 coredump 能力

操作场景

创建/更新POD前，支持通过Annotation（ system.cci.io/coredump ）配置内核的core_pattern内容，配置后容器内业务进程故障可通过生成的coredump文件进而分析实例程序异常原因。

使用说明

创建pod时，请求的Annotation里增加core dump配置（ system.cci.io/coredump : ‘{“core_pattern”：“/var/cores/core.%h.%e.%p.%t”，“max_size”：2048}’ ），如果没有指定max size，默认值是1024，在对应pod的容器里使用core dump能力。

约束与限制

- 请求里的system.cci.io/core_pattern字段必须以“/var/”开头且文件路径里不能有“|”字符。
- 需要用户确保配置的路径在容器中存在，推荐使用挂卷的方式持久化保存coredump文件。

通过 ccictl 创建 pod

您可以通过对工作负载添加annotations控制是否为当前pod开启coredump能力，如下所示。

```
apiVersion: cci/v2
kind: Deployment
metadata:
  annotations:
    description: ''
  labels: {}
  name: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
```

 版权所有 © 华为云计算技术有限公司

```
    app: nginx
 template:
   metadata:
     annotations:
       vm.cci.io/pod-size-specs: 2.00_4.0
       resource.cci.io/pod-size-specs: 2.00_4.0
       metrics.alpha.kubernetes.io/custom-endpoints: '[{api:"",path:"",port:"",names:""}]'
       system.cci.io/coredump: "{\"core_pattern\": \"/var/cores/core.%h.%e.%p.%t\",\"max_size\":
1024}"    # 配置开启coredump能力
       log.stdoutcollection.kubernetes.io: '{"collectionContainers": ["container-0"]}'
     labels:
       app: nginx
   spec:
     containers:
       - image: library/nginx:stable-alpine-perl
         name: container-0
         resources:
           limits:
             cpu: 2000m
             memory: 4096Mi
           requests:
             cpu: 2000m
             memory: 4096Mi
         command: []
         lifecycle: {}
     dnsPolicy: ''
     imagePullSecrets:
       - name: imagepull-secret
     dnsConfig: {}
 minReadySeconds: 0
 strategy:
   type: RollingUpdate
   rollingUpdate:
     maxSurge: 0
     maxUnavailable: 1
```

说明

system.cci.io/coredump：Pod需要配置的coredump能力。

6.2 配置镜像 header 能力

操作场景

镜像下载时，通过Annotation（cci.io/registry-headers）配置registry和header信息，增加自定义的信息说明来源。

使用说明

创建pod时，请求的Annotation里增加registry-headers配置（cci.io/registry-headers: "[{\"registry\":\"swr.cn-north-4.myhuaweicloud.com\",\"headers\": {\"key1\":\"value1\",\"key2\":\"value2\"}},{\"registry\":\"swr.cn-north-8.myhuaweicloud.com\",\"headers\": {\"key1\":\"value1\",\"key2\":\"value2\"}}]"），在对应pod的容器里使用registry-headers能力。

约束与限制

- 请求里的cci.io/registry-headers字段要校验registry和headers。
- 请求的cci.io/registry-headers字段内容从annotation获取。

版权所有 © 华为云计算技术有限公司

通过 ccictl 创建 pod

您可以通过对工作负载添加annotations控制是否为当前pod开启registry-headers能力，如下所示。

```
apiVersion: cci/v2
kind: Deployment
metadata:
  annotations:
    description: ''
  labels: {}
  name: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      annotations:
        vm.cci.io/pod-size-specs: 2.00_4.0
        resource.cci.io/pod-size-specs: 2.00_4.0
        metrics.alpha.kubernetes.io/custom-endpoints: '[{api:"",path:"",port:"",names:""}]'
        cci.io/registry-headers: "[{\"registry\":\"swr.cn-north-4.myhuaweicloud.com\",\"headers\":
{\"key1\":\"value1\",\"key2\":\"value2\"}},{\"registry\":\"swr.cn-north-8.myhuaweicloud.com
\",\"headers\":{\"key1\":\"value1\",\"key2\":\"value2\"}}]"      # 配置开启registry-headers能力
        log.stdoutcollection.kubernetes.io: '{"collectionContainers": ["container-0"]}'
      labels:
        app: nginx
    spec:
      containers:
        - image: library/nginx:stable-alpine-perl
          name: container-0
          resources:
            limits:
              cpu: 2000m
              memory: 4096Mi
            requests:
              cpu: 2000m
              memory: 4096Mi
          command: []
          lifecycle: {}
      dnsPolicy: ''
      imagePullSecrets:
        - name: imagepull-secret
      dnsConfig: {}
  minReadySeconds: 0
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 0
      maxUnavailable: 1
```

说明

cci.io/registry-headers：Pod需要配置的registry-headers能力。

版权所有 © 华为云计算技术有限公司

7 成本标签管理

操作场景

云容器实例支持通过配置成本标签，方便您对Pod的成本进行分类和跟踪。

标签命名规则

- 每个标签由一对键值对（Key-Value）组成。示例：
  ```
  pod-tag.cci.io/<标签>:标签值
  ```
- 每个Pod最多可以添加20个标签。
- 对于每个Pod，每个标签键（Key）都必须是唯一的，每个标签键（Key）只能有一个值（Value）。
- 标签共由两部分组成：“标签键”和“标签值”，其中，“标签键”和“标签值”的命名规则如表7-1所示。

表 7-1 标签命名规则

参数	规则	样例
标签键	• 不能为空 • 对于同一个Pod，Key值唯一 • 长度不超过128个字符，一个中文代表一个字符 • 可用英文字母、中文字符、数字和空格，以及以下字符： _ . : = + - @ • 首尾字符不能为空格	Organization
标签值	• 长度不超过255个字符，一个中文代表一个字符 • 可用英文字母、中文字符、数字和空格，以及以下字符： _ . : = + - @ /	Apache

版权所有 © 华为云计算技术有限公司

操作步骤

步骤1 登录华为云管理控制台。

步骤2 在右上角的用户名下选择“标签管理”，进入标签管理服务页面。

图 7-1 标签管理

步骤3 创建预定义标签。在左侧导航栏选择“预定义标签”，单击“创建标签”，填写“标签键”和“标签值”，完成后单击“确定”。

步骤4 在控制台选择“费用 > 成本中心”，进入成本中心。

步骤5 选择“成本组织 > 成本标签”，在成本标签页面勾选**步骤3**创建的标签，单击“激活”，单击“确定”。

步骤6 等待所创建的成本标签激活状态显示“已激活”。

步骤7 在云容器实例服务创建Pod时在Podtemplate.metadata.labels中加入相关字段，YAML示例如下：

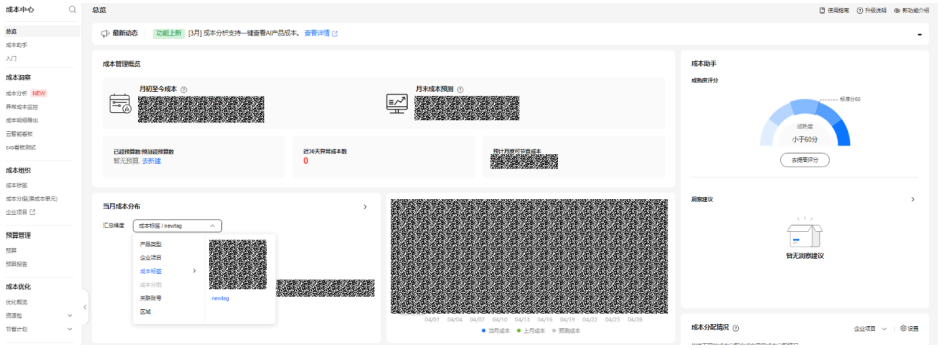
```
kind: Deployment
apiVersion: cci/v2
metadata:
  name: tag-example
  namespace: bursting-d0089910820250427-2031
spec:
  replicas: 3
  selector:
    matchLabels:
      app: tag-example
  template:
    metadata:
      labels:
        app: tag-example
        pod-tag.cci.io/newtag: tag-example-1   # pod-tag.cci.io/<自定义标签>: 自定义标签值
    spec:
      containers:
        - name: deploy-example
          image: swr.region-id.domain.com/org-name/image-name:tag
```


版权所有 © 华为云计算技术有限公司

```
        env:
          - name: ENV1
            value: 'false'
          - name: ENV2
            value: xxx
        resources:
          limits:
            cpu: 500m
            memory: 1Gi
          requests:
            cpu: 500m
            memory: 1Gi
      dnsPolicy: Default
      imagePullSecrets:
        - name: imagepull-secret
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxUnavailable: 0
    maxSurge: 100%
```

步骤8 在控制台选择“费用 > 成本中心”，进入成本中心。在成本中心总览页面当月成本分布模块，汇总维度选择成本标签，选择所创建的标签，查看相关信息。

图 7-2 当月成本分布



----结束

 版权所有 © 华为云计算技术有限公司

8 审计

8.1 云审计服务支持的 CCI 操作列表

CCI通过云审计服务（Cloud Trace Service，简称CTS）为您提供云服务资源的操作记录，记录内容包括您从云管理控制台或者开放API发起的云服务资源操作请求以及每次请求的结果，供您查询、审计和回溯使用。

表 8-1 云审计服务支持的 CCI 操作列表

操作名称	事件名称
创建一个Service对象	createService
删除一个Service对象	deleteService
替换指定的Service对象	replaceService
更新指定的Service对象	updateService
创建一个Deployment对象	createDeployment
删除一个Deployment对象	deleteDeployment
替换指定Namespace下的Deployment对象	replaceDeployment
更新指定Namespace下的Deployment对象	updateDeployment
创建一个Namespace	createNamespace
删除一个Namespace	deleteNamespace
创建一个Pod	createPod
更新指定Pod	updatePod
替换指定Pod	replacePod
删除一个Pod	deletePod

版权所有 © 华为云计算技术有限公司

操作名称	事件名称
替换可观测性配置	replaceObservabilityconfiguration
创建一个Configmap	createConfigmap
更新指定Configmap	updateConfigmap
替换指定Configmap	replaceConfigmap
删除一个Configmap	deleteConfigmap
创建一个Secret	createSecret
更新指定Secret	updateSecret
替换指定Secret	replaceSecret
删除指定Secret	deleteSecret
删除指定Network	deleteNetwork
创建一个Network	createNetwork
更新指定Network	updateNetwork
替换指定Network	replaceNetwork
创建PVC	createPersistentvolumeclaim
替换指定PVC	replacePersistentvolumeclaim
更新指定PVC	updatePersistentvolumeclaim
删除指定PVC	deletePersistentvolumeclaim
创建镜像快照	createImagesnapshot
删除指定镜像快照	deleteImagesnapshot
查询镜像快照	getImagesnapshot
查询指定Namespace下的所有镜像快照	listImagesnapshots
创建PV	createPV
更新指定PV	updatePV
替换指定PV	replacePV
删除指定PV	deletePV
查询PV	getPV
查询指定Namespace下的所有PV	listPVs
查询Replicaset	getReplicaset
查询指定Namespace下的所有Replicaset	listReplicasets
创建HPA	createHPA

 版权所有 © 华为云计算技术有限公司

操作名称	事件名称
删除指定HPA	deleteHPA
更新指定HPA	updateHPA
替换指定HPA	replaceHPA
查询HPA	getHPA
查询指定Namespace下的所有HPA	listHPAs
创建Poolbinding	createPoolbinding
删除指定Poolbinding	deletePoolbinding
查询Poolbinding	getPoolbinding
查询指定Namespace下的所有Poolbinding	listPoolbindings

8.2 查看云审计日志

操作场景

开启了云审计服务后，系统开始记录CCI资源的操作。云审计服务管理控制台保存最近7天的操作记录。

操作步骤

步骤1 登录管理控制台。

步骤2 单击管理控制台左上角的图标，选择区域。

步骤3 单击页面上方的“服务列表”，选择“管理与监管 > 云审计服务”，进入云审计服务信息页面。

步骤4 单击左侧导航树的“事件列表”，进入事件列表信息页面。

步骤5 事件列表支持通过筛选来查询对应的操作事件。当前事件列表支持四个维度的组合查询，详细信息如下：

- 事件类型、事件来源、资源类型和筛选类型。

  在下拉框中选择查询条件。其中，事件来源选择“CCI”。

  当筛选类型选择事件名称时，还需选择某个具体的事件名称。

  选择资源ID时，还需选择或者手动输入某个具体的资源ID。

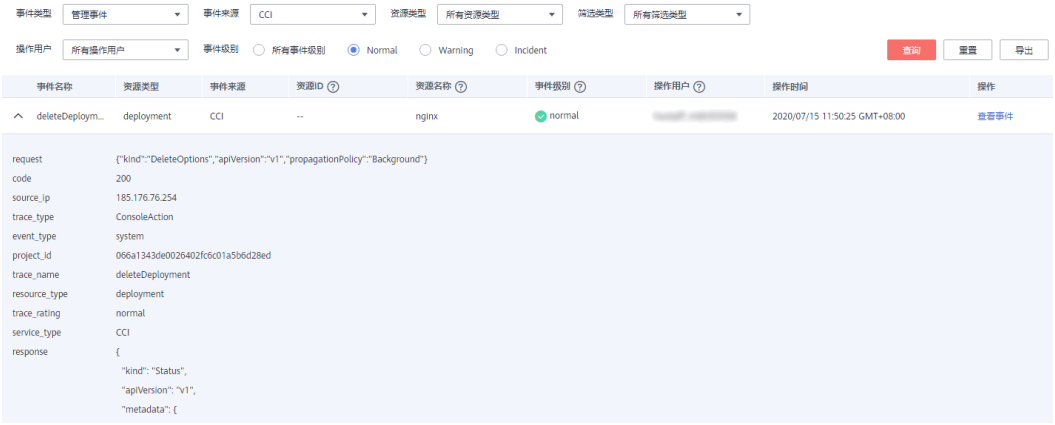
  选择资源名称时，还需选择或手动输入某个具体的资源名称。
- 操作用户：在下拉框中选择某一具体的操作用户，此操作用户指用户级别，而非租户级别。
- 事件级别：可选项为“所有事件级别”、“normal”、“warning”、“incident”，只可选择其中一项。

版权所有 © 华为云计算技术有限公司

- 起始时间、结束时间：可通过选择时间段查询操作事件。

步骤6 在需要查看的记录左侧，单击 展开该记录的详细信息，展开记录如图8-1所示。

图 8-1 展开记录



步骤7 在需要查看的记录右侧，单击“查看事件”，弹出一个窗口，如图8-2所示，显示了该操作事件结构的详细信息。

图 8-2 查看事件

----结束

 版权所有 © 华为云计算技术有限公司