天翼云•云容器引擎

用户使用指南

天翼云科技有限公司

目 录

1 什么是云容器引擎.....7
2 高危操作及解决方案.....9
3 集群管理.....13
3.1 集群概述.....13
3.2 集群生命周期.....15
3.3 购买混合集群.....16
3.4 Kubectl 访问集群.....27
3.4.1 Kubectl 使用指南.....27
3.4.2 通过 kubectl 操作 CCE 集群.....28
3.4.3 通过 kubectl 配置 kube-dns/CoreDNS 高可用.....29
3.4.4 Kubectl 常用命令.....31
3.5 集群弹性扩容.....37
3.6 集群升级.....40
3.6.1 集群版本升级说明.....40
3.6.2 升级集群.....42
3.6.3 集群跨版本业务迁移.....46
3.7 管理集群.....47
3.7.1 删除集群.....47
3.7.2 集群休眠与唤醒.....48
3.7.3 配置管理.....49
3.8 获取集群证书.....51
3.9 集群监控.....51
3.10 集群管理权限控制.....52
4 节点管理.....54
4.1 节点概述.....54
4.2 购买节点.....56
4.3 纳管已有节点到集群.....60
4.4 登录节点.....62
4.5 节点监控.....63
4.6 管理节点标签.....65
4.7 同步节点信息.....66
4.8 重置节点.....67
4.9 删除节点.....68
4.10 节点关机.....68
4.11 节点滚动升级.....69
4.12 节点预留资源计算公式.....73

4.13 为 docker 创建 Linux LVM 型磁盘分区 75
5 节点池管理 77
5.1 节点池概述 77
5.2 创建节点池 80
5.3 管理节点池 85
6 工作负载 89
6.1 工作负载概述 89
6.2 创建无状态负载(Deployment) 91
6.3 创建有状态负载(StatefulSet) 98
6.4 创建守护进程集(DaemonSet) 106
6.5 创建普通任务(Job) 111
6.6 创建定时任务(CronJob) 116
6.7 管理容器组(Pod) 121
6.8 管理工作负载和任务 122
6.9 工作负载弹性伸缩 128
6.10 容器设置 131
6.10.1 如何使用第三方镜像 131
6.10.2 设置容器规格 133
6.10.3 设置容器生命周期 135
6.10.4 设置容器启动命令 139
6.10.5 设置容器健康检查 143
6.10.6 设置环境变量 145
6.10.7 采集容器标准输出日志 147
6.10.8 采集容器内路径日志 148
6.10.9 对接 Prometheus 实现自定义指标监控 152
6.10.10 健康检查 UDP 协议安全组规则说明 153
6.10.11 Kubernetes 集群内置 DNS 配置说明 154
7 亲和/反亲和性调度 159
7.1 调度策略概述 159
7.2 自定义调度策略 161
7.2.1 节点亲和性 161
7.2.2 工作负载亲和性 162
7.2.3 工作负载反亲和性 164
7.3 简易调度策略 166
7.3.1 工作负载和可用区的亲和性 166
7.3.2 工作负载和可用区的反亲和性 167
7.3.3 工作负载和节点的亲和性 169
7.3.4 工作负载和节点的反亲和性 170

7.3.5 工作负载间的亲和性 172
7.3.6 工作负载间的反亲和性 173
8 网络管理 175
8.1 网络概述 175
8.2 Service 178
8.2.1 集群内访问(ClusterIP) 178
8.2.2 节点访问(NodePort) 185
8.2.3 负载均衡(LoadBalancer) 193
8.3 Ingress 206
8.3.1 Ingress 概述 206
8.3.2 基本功能操作 207
8.3.3 通过 Kubectl 命令行创建 Ingress 214
8.4 网络模型 226
8.4.1 网络模型概述 226
8.4.2 容器隧道网络 228
8.4.3 VPC 网络 229
8.5 NetworkPolicy 231
9 存储管理 233
9.1 存储概述 233
9.2 本地磁盘存储 234
9.3 云硬盘存储卷 239
9.3.1 云硬盘存储卷使用说明 239
9.3.2 使用云硬盘存储卷 240
9.3.3 使用 kubectl 自动创建云硬盘 245
9.3.4 使用 kubectl 对接已有云硬盘 247
9.3.5 使用 kubectl 部署带云硬盘存储卷的工作负载 258
9.4 文件存储卷 263
9.4.1 文件存储卷使用说明 263
9.4.2 使用文件存储卷 264
9.4.3 使用 kubectl 自动创建文件存储 268
9.4.4 使用 kubectl 对接已有文件存储 269
9.4.5 使用已有 PVC 创建有状态工作负载 275
9.4.6 使用 kubectl 部署带文件存储卷的工作负载 277
9.5 极速文件存储卷 282
9.5.1 极速文件存储卷使用说明 282
9.5.2 使用极速文件存储卷 283
9.5.3 使用 kubectl 对接已有极速文件存储卷 285
9.5.4 使用已有 PVC 创建有状态工作负载 289
9.5.5 使用 kubectl 挂载极速文件存储 291

9.6 快照与备份 292
10 命名空间 296
10.1 创建命名空间 296
10.2 管理命名空间 298
10.3 设置命名空间级的网络策略 299
10.4 设置资源配额及限制 300
11 配置中心 302
11.1 创建配置项 302
11.2 使用配置项 304
11.3 创建密钥 307
11.4 使用密钥 310
12 模板市场 313
12.1 我的模板 313
12.1.1 模板概述 313
12.1.2 准备模板包 313
12.1.3 上传模板包 316
12.1.4 通过模板创建工作负载 316
12.1.5 使用云硬盘 318
12.1.6 使用弹性负载均衡 319
13 插件管理 321
13.1 插件概述 321
13.2 CoreDNS（系统资源插件，必装） 322
13.3 Everest（系统资源插件，必装） 326
13.4 storage-driver（系统资源插件，必装） 328
13.5 autoscaler 329
13.6 metrics-server 335
13.7 cce-hpa-controller 336
13.8 prometheus 337
13.9 gpu-beta 340
14 弹性伸缩 344
14.1 弹性伸缩概述 344
14.2 工作负载弹性伸缩 346
14.2.1 工作负载伸缩原理 346
14.2.2 创建工作负载弹性伸缩（HPA） 349
14.2.3 创建工作负载弹性伸缩（CustomedHPA） 351
14.2.4 管理工作负载伸缩策略 353
14.3 集群/节点弹性伸缩 355
14.3.1 节点伸缩原理 355

14.3.2 创建节点伸缩策略 358
14.3.3 管理节点伸缩策略 361
14.3.4 节点伸缩常见问题 362
15 权限管理 365
15.1 CCE 权限概述 365
15.2 集群权限 368
15.3 命名空间权限 376
15.4 创建用户并授权使用 CCE 378
15.4.1 设置集群权限 378
15.4.2 设置命名空间权限 383
15.5 CCE 控制台的权限依赖 385
16 云监控服务 390
16.1 支持的监控指标 390
16.2 设置告警规则 391
16.3 查看监控指标 393
17 云审计服务 394
17.1 云审计服务支持的 CCE 操作列表 394
17.2 查看云审计日志 396
18 相关服务 399
18.1 容器镜像服务 399
18.2 应用运维管理 400

1 什么是云容器引擎

Kubernetes 是主流的开源容器编排平台。为了让用户可以方便地在云上使用 Kubernetes 管理容器应用，我们推出了基于原生 Kubernetes 的云容器引擎服务。

云容器引擎（Cloud Container Engine，简称 CCE）提供高度可扩展的、高性能的企业级 Kubernetes 集群，支持运行 Docker 容器。借助云容器引擎，您可以在云上轻松部署、管理和扩展容器化应用程序。

云容器引擎深度整合了高性能的计算（ECS/BMS）、网络（VPC/EIP/ELB）、存储（EVS/OBS/SFS）等服务，并支持 GPU、ARM、FPGA 等异构计算架构，支持多可用区（Available zone，简称 AZ）、多区域（Region）容灾等技术构建高可用 Kubernetes 集群，并提供高性能可伸缩的容器应用管理能力，简化集群的搭建和扩容等工作，让您专注于容器化应用的开发与管理。

名词解释

使用云容器引擎服务，会涉及到以下基本概念：

- **集群：**是指容器运行所需云资源的集合，包含了若干台云服务器、负载均衡器等云资源。
- **实例（Pod）：**由相关的一个或多个容器构成一个实例，这些容器共享相同的存储和网络空间。
- **工作负载：**Kubernetes 资源对象，用于管理 Pod 副本的创建、调度以及整个生命周期的自动控制。
- **Service：**由多个相同配置的实例（Pod）和访问这些实例（Pod）的规则组成的微服务。
- **Ingress：**Ingress 是用于将外部 HTTP（S）流量路由到服务（Service）的规则集合。
- **Helm 应用：**Helm 是管理 Kubernetes 应用程序的打包工具，提供了 Helm Chart 在指定集群内图形化的增删改查。
- **镜像仓库：**用于存放 Docker 镜像，Docker 镜像用于部署容器服务。

您在使用前可以了解更多 Kubernetes 相关知识，具体请参见 https://kubernetes.io/docs/concepts/。

主要功能

云容器引擎支持对容器应用的全生命周期管理，具有以下功能：

集群管理

- 通过控制台一键创建 Kubernetes 集群，支持跨可用区高可用。

一站式容器管理

- 容器应用全生命周期管理。
- 高性能容器隧道网络、VPC 网络等容器网络。
- 云硬盘 EVS、弹性文件存储 SFS、对象存储 OBS 等持久化存储支持。
- 资源、应用、容器多维度监控。
- 多样化的日志报表统计。
- 基于角色的权限管理和容器运行时安全。

应用市场内容

- 丰富的 Helm chart 组件。
- 对接开源镜像中心和容器镜像服务，支持自定义镜像和共享镜像。

开发者服务

- 提供 OpenAPI 和社区原生 API。
- 提供 Kubectl 插件和社区原生 Kubectl 工具。

开始使用

您可以通过多种方式使用云容器引擎服务：

- 您可以通过 CCE 控制台非常方便的执行创建集群、创建工作负载，以及配置网络、挂载存储、安装插件、弹性伸缩、权限管理等各种操作。
- 您也可以通过 Kubectl 命令行、API 参考执行操作，这需要您具备 Kubectl 相关开发技能，了解 Kubectl 相关操作，具体请参见 Kubernetes API、kubectl CLI。

2 高危操作及解决方案

业务部署或运行过程中，用户可能会触发不同层面的高危操作，导致不同程度上的业务故障。为了能够更好地帮助用户预估及避免操作风险，本节将从集群/节点、网络与负载均衡、日志、云硬盘多个维度出发，为用户展示哪些高危操作会导致怎样的后果，以及为用户提供相应的误操作解决方案。

集群/节点

表2-1 集群及节点高危操作

分类	高危操作	导致后果	误操作后解决方案
master 节点	修改集群内节点安全组	可能导致 master 节点无法使用 说明 命名规则：集群名称-cce-control-随机数	参照购买混合集群的安全组进行修复，放通安全组。
	节点到期或被销毁	该 master 节点不可用	不可恢复。
	重装操作系统	master 组件被删除	不可恢复。
	自行升级 master 或者 etcd 组件版本	可能导致集群无法使用	回退到原始版本。
	删除或格式化节点/etc/kubernetes 等核心目录数据	该 master 节点不可用	不可恢复。
	更改节点 IP	该 master 节点不可用	改回原 IP。
	自行修改核心组件（etcd、kube-apiserver、docker 等）参数	可能导致 master 节点不可用	按照推荐配置参数恢复，详情请参见 3.7.3 配置管理。

分类	高危操作	导致后果	误操作后解决方案
	自行更换 master 或 etcd 证书	可能导致集群不可用	不可恢复。
worker 节点	修改集群内节点安全组	可能导致节点无法使用 说明 命名规则：集群名称-cce-node-随机数	参照 3.3 购买混合集群的安全组进行修复，放通安全组。
	节点被删除	该节点不可用	不可恢复。
	重装操作系统	节点组件被删除，节点不可用	重置节点，具体请参见 4.8 重置节点。
	升级节点内核	可能导致节点无法使用或网络异常 说明 CCE 集群依赖系统内核版本，如非必要，请不要使用 yum update 更新或重装节点的操作系统内核（使用原镜像或其它镜像重装均属高危操作）	-
	更改节点 IP	节点不可用	改回原 IP。
	自行修改核心组件（kubelet、kube-proxy 等）参数	可能导致节点不可用、修改安全相关配置导致组件不安全等	按照推荐配置参数恢复，详情请参见操作场景。
	修改操作系统配置	可能导致节点不可用	尝试还原配置项或重置节点，具体请参见 4.8 重置节点。
	删除/opt、/var/paas 目录，删除数据盘	节点不可用	重置节点，具体请参见 4.8 重置节点。
	修改节点内目录权限、容器目录权限等	权限异常	不建议修改，请自行恢复。
	对节点进行磁盘格式化或分区	节点不可用	重置节点，具体请参见 4.8 重置节点。
	在节点上安装自己的其他软件	导致安装在节点上的 Kubernetes 组件异常，节点状态变成不可用，无法部署工作负载到此节点	卸载已安装软件，尝试恢复或重置节点，具体请参见 4.8 重置节点。

网络与负载均衡

表2-2 网络与负载均衡

高危操作	导致后果	误操作后解决方案
修改内核参数 net.ipv4.ip_forward=0	网络不通	修改内核参数为 net.ipv4.ip_forward=1
修改内核参数 net.ipv4.tcp_tw_recycle=1	导致 nat 异常	修改内核参数 net.ipv4.tcp_tw_recycle=0
节点安全组配置未放通容器 CIDR 的 53 端口 udp	集群内 DNS 无法正常工作	参照 3.3 购买混合集群的安全组进行修复，放通安全组。
通过 ELB 的控制台在 CCE 管理的 ELB 创建自定义的监听器	所做修改被 CCE 侧重置或 Ingress 故障	通过 service 的 yaml 来自动创建监听器。
通过 ELB 的控制台在 CCE 管理的 ELB 绑定自定义的后端		禁止手动绑定后端。
通过 ELB 的控制台修改 CCE 管理的 ELB 的证书		通过 ingress 的 yaml 来自动管理证书。
通过 ELB 的控制台修改 CCE 管理的 ELB 监听器名称		禁止修改 CCE 管理的 ELB 监听器名称。
通过 ELB 的控制台修改 CCE 管理的 ELB 实例、监听器、转发策略的描述		禁止修改 CCE 管理的 ELB 实例、监听器、转发策略的描述。

日志

表2-3 日志

高危操作	导致后果	误操作后解决方案
删除宿主机/tmp/ccs-log-collector/pos 目录	日志重复采集	无
删除宿主机/tmp/ccs-log-collector/buffer 目录	日志丢失	无

云硬盘

表2-4 云硬盘

高危操作	导致后果	误操作后解决方案	备注
控制台手动解挂 EVS	Pod 写入报 io error	删掉 node 上 mount 目录，重新调度 Pod	Pod 里面的文件记录了文件的采集位置
节点上 umount 磁盘挂载路径	Pod 写入本地磁盘	重新 mount 对应目录到 Pod 中	Buffer 里面是待消费的日志缓存文件
节点上直接操作 EVS	Pod 写入本地磁盘	无	无

3 集群管理

3.1 集群概述

随着应用程序开发向基于容器的方向发展，编排和管理资源的需求变得越来越重要。Kubernetes 是一个开源的、功能强大的容器编排系统，用于管理容器化应用和服务，它提供了应用部署、规划、更新、维护的一种机制，让部署容器化的应用更加简单并且高效。

云容器引擎（Cloud Container Engine，以下简称 CCE）是一种托管的 Kubernetes 服务，可进一步简化基于容器的应用程序部署和管理，您可以在 CCE 中方便的创建 Kubernetes 集群、部署您的容器化应用，以及方便的管理和维护。

集群优势

传统的应用部署方式是通过插件或脚本来安装应用。这样做的缺点是应用的运行、配置、管理、所有生存周期将与当前操作系统绑定，这样做并不利于应用的升级更新/回滚等操作，当然也可以通过创建虚拟机的方式来实现某些功能，但是虚拟机非常重，并不利于可移植性。

新的方式是通过部署容器方式实现，每个容器之间互相隔离，每个容器有自己的文件系统，容器之间进程不会相互影响，能区分计算资源。相对于虚拟机，容器能快速部署，由于容器与底层设施、机器文件系统解耦的，所以它能在不同云、不同版本操作系统间进行迁移。

容器占用资源少、部署快，每个应用可以被打包成一个容器镜像，每个应用与容器间成一对一关系也使容器有更大优势，使用容器可以在 build 或 release 的阶段，为应用创建容器镜像，因为每个应用不需要与其余的应用堆栈组合，也不依赖于生产环境基础结构，这使得从研发到测试、生产能提供一致环境。类似地，容器比虚拟机轻量、更“透明”，这更便于监控和管理。

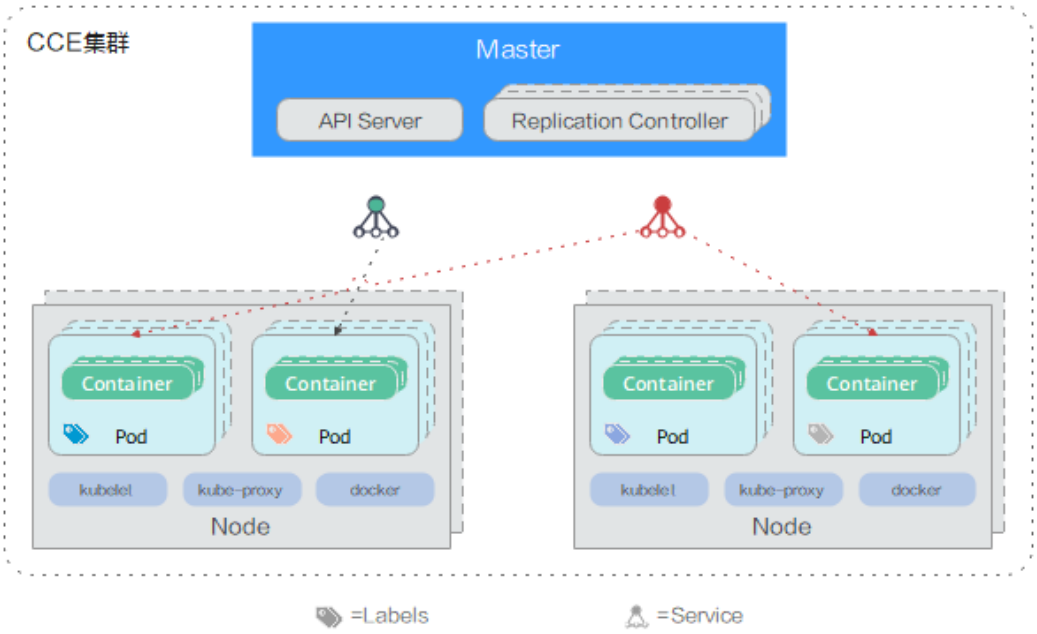
集群（Cluster）是容器运行所需云资源的集合，包含了若干云服务器节点（物理服务器或者虚拟机）、负载均衡、虚拟私有云等云资源，您可以在集群中运行您的应用程序。在 CCE 中，我们可以创建若干集群，每个集群可以创建多个容器，每个容器里面运行一个应用实例，然后通过内置的负载均衡策略，实现对这一组应用实例的管理、发现、访问，而这些细节都不需要运维人员去进行复杂的手工配置和处理。

集群架构

CCE 采用兼容标准的 Kubernetes 集群，Kubernetes 集群属于主从分布式架构，主要由 Master 和 Worker Node 组成，以及包括客户端命令行工具 kubectl 和其它附加项。因此，在 CCE 集群中至少包含一个 Master 和多个 Worker Node，这些 Worker Node 称为工作节点，这些节点都运行在 Kubernetes 集群编排系统中。

下图简要介绍了 CCE 集群的架构：

图3-1 集群架构图



- **Master：**管控集群的管理面节点，提供 Kubernetes 核心服务和应用程序工作负载的业务流程，对集群进行调度管理。Master Node 由 API Server、Scheduler、Cluster State Store 和 Controller-Manager Server 所组成。
- **Node：**业务运行的工作节点，用于运行应用程序的容器工作负载。Worker Node 包含 kubelet、kube proxy 和 Container Runtime。
- kubectl：用于通过命令行与 API Server 进行交互，而对 Kubernetes 进行操作，实现在集群中进行各种资源的增删改查等操作。
- replication Controller：副本控制器，用于伸缩副本数量。
- Pod：实例（Pod）是 Kubernetes 部署应用或服务的最小的基本单位。一个 Pod 封装多个应用容器（也可以只有一个容器）、存储资源、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。
- Container：容器。一个通过 Docker 镜像创建的运行实例，一个节点可运行多个容器。容器的实质是进程，但与直接在宿主执行的进程不同，容器进程运行于属于自己的独立的命名空间。
- Label：标签。其实就一对 key/value ，被关联到对象上，比如 Pod、标签的使用我们倾向于能够标示对象的特殊特点，并且对用户而言是有意义的，但是标签对内核系统是没有直接意义的。
- Service：服务，将运行在一组 Pods 上的应用程序公开为网络服务的抽象方法。

CCE 集群类型

云容器引擎 CCE 深度整合高性能的计算（ECS/BMS）、网络（VPC/EIP/ELB）、存储（EVS/OBS/SFS）等服务，支持多可用区（Available zone，简称 AZ）、多区域（Region）容灾等技术构建高可用 Kubernetes 集群。

CCE 已支持创建 Kubernetes 集群。

混合集群：支持虚拟机与裸金属服务器混合、支持 GPU 等异构节点的混合部署，基于高性能网络模型提供全方位、多场景、安全稳定的容器运行环境。

⚠ 注意

- 在创建、删除、扩容和缩容集群的操作中，请不要在统一身份认证服务（IAM）中执行权限变更或修改的操作，可能会导致创建、删除、扩容和缩容集群执行失败。
- CCE 节点的容器网络 canal 会使用一个网段作为容器网络的网段，该网段在创建集群时可配置，默认为 172.16.0.0/16；docker 服务也会默认创建一个 docker0 的网桥，docker0 的默认地址为 172.17.0.1。创建集群时，请确保集群中 VPC（虚拟私有云）所在的网段不能和上述两个网段重复。使用 VPC 的对等连接功能时，对端的 VPC 也不能和上述两个网段重复。

后续步骤

有关 CCE 概念和具体操作的详细信息，请参阅以下文章：

- 3.3 购买混合集群
- 3.4.2 通过 kubectl 操作 CCE 集群
- 6.2 创建无状态负载(Deployment)

3.2 集群生命周期

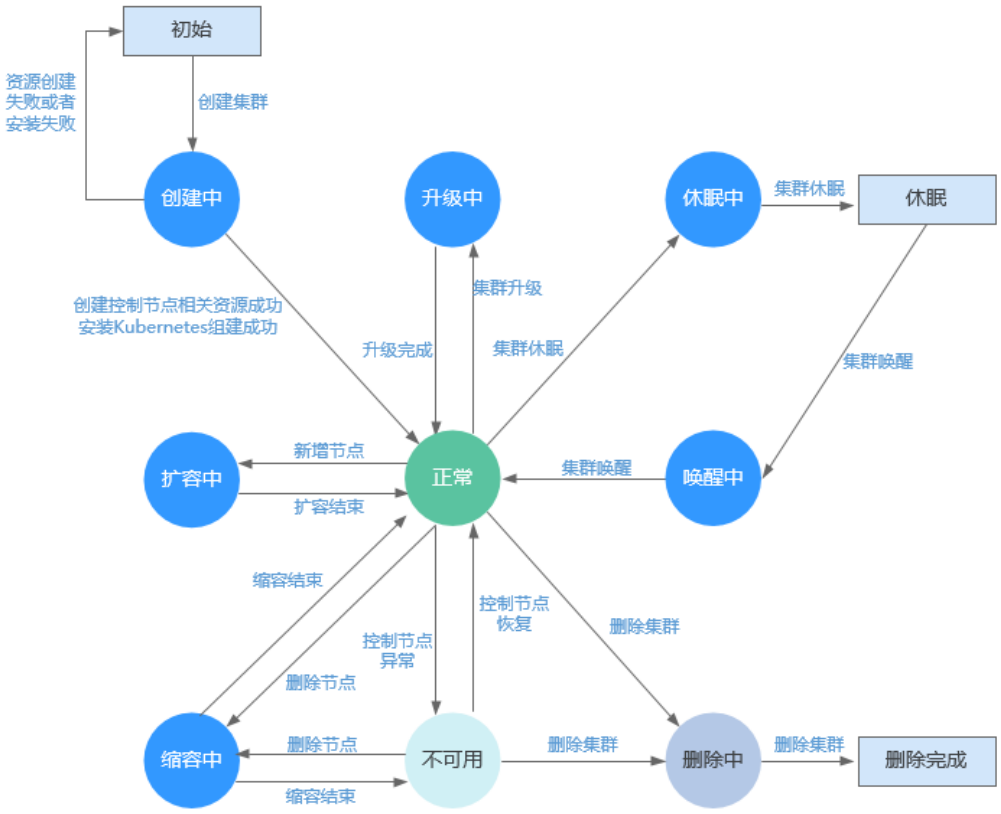
介绍集群各生命周期的状态，便于您更深入的理解集群处于不同阶段的运行情况。

表3-1 集群状态说明

状态	说明
创建中	集群正在创建，正在申请云资源
正常	集群正常运行
扩容中	集群正在增加节点
缩容中	集群正在删除节点
休眠中	集群正在休眠中

状态	说明
唤醒中	集群正在唤醒中
升级中	集群正在升级中
不可用	当前集群不可用
删除中	集群正在删除中

图3-2 集群状态流转



3.3 购买混合集群

您可以通过云容器引擎控制台非常方便快速的创建 Kubernetes 集群。Kubernetes 是大规模容器集群管理软件，一个集群可以管理一组节点资源。

CCE 集群支持虚拟机与裸金属服务器混合、支持 GPU 等异构节点的混合部署，基于高性能网络模型提供全方位、多场景、安全稳定的容器运行环境，您可以通过 4.2 购买节点实现多种场景的混合部署。

前提条件

- 创建首个集群前，您必须先确保已存在虚拟私有云，否则无法创建集群。若您已有虚拟私有云，可重复使用，无需重复创建。

  虚拟私有云为 CCE 集群提供一个隔离的、用户自主配置和管理的虚拟网络环境。
- 您需要新建一个密钥对，用于远程登录节点时的身份认证。

  若使用密码登录节点，请跳过此操作。
- 创建集群前，请提前规划好容器网段和服务网段。网段参数在集群创建后不可更改，需要重新创建集群才能调整，请谨慎选择。

创建集群须知

创建集群过程中会同步创建一些基础资源，列表如下：

表3-2 创建集群须知

资源名称	描述
控制节点及其相关资源	存在于云容器引擎资源租户下，用户不可见。
弹性云主机（可选创建）	集群节点，即用户的计算资源，对应“弹性云主机”中的ECS。 ECS 命名规则为：集群名称-随机数，可自定义，批量创建时会再加一串随机数。
安全组	集群会创建两个安全组，一个用于管理集群控制节点的安全组，一个用于管理集群工作节点的安全组。 **警告** **集群创建时自动创建的安全组以及安全组规则禁止删除，否则会导致集群异常。** 1. 控制节点安全组 命名规则：集群名称-cce-control-随机数 作用： • 出方向允许。 • 其他节点访问控制节点 kubernetes 相关服务。 2. 工作节点安全组 命名规则：集群名称-cce-node-随机数 作用： • 出方向允许。 • 开放 linux 或 windows 远程登录（22、3389）。 • kubernetes 组件之间访问使用（4789、10250）。 • kubernetes 用于对外开放的端口（30000-32767）。 • 相同安全组之间可以互相访问。
磁盘（可选创建）	分别给各个节点创建两个磁盘，一个是节点的系统盘，一

资源名称	描述
	个是给 docker 运行使用的数据盘。
弹性 IP（可选创建）	需用户选择，给节点绑定弹性 IP ，可以使节点访问外网。

操作步骤

步骤 1 登录 CCE 控制台，在总览页面单击“购买 Kubernetes 集群”，或在左侧导航栏中单击“资源管理 > 集群管理”，单击“混合集群”下的“购买”。

步骤 2 参照表 3-3 设置集群参数。

表3-3 创建集群参数配置

参数	参数说明
计费模式	• 包年/包月：预付费模式，按订单的购买周期计费，适用于可预估资源使用周期的场景，价格比按需计费模式更优惠。包年/包月集群创建后不能删除。 • 按需计费：后付费模式，按资源的实际使用时长计费，可以随时开通/删除资源。 本章以“按需计费”类型为例进行讲解。
区域	不同区域的云服务产品之间内网互不相通；请就近选择靠近您业务的区域，可减少网络时延，提高访问速度。
集群名称	新建集群的名称，创建后不可修改。 集群名称长度范围为 4-128 个字符，以小写字母开头，由小写字母、数字、中划线（-）组成，且不能以中划线（-）结尾。
版本	Kubernetes 社区基线版本，建议选择最新的版本。版本升级请参见 3.6.1 集群版本升级说明。 若有 **Beta** 版本时，您可以选择试用，但不建议您将该版本用于商用场景。
集群管理规模	集群管理规模是指当前集群的控制节点可以管理的最大工作节点规模，您可以选择 50 节点、200 节点、1000 节点、2000 节点（部分区域支持）管理规模，请根据您的业务需求选择，该规模在集群创建后不可更改，请慎重选择。 若选择“1000 节点”，表示当前集群的控制节点最多可管理 1000 个工作节点。由于不同管理规模的控制节点规格不同，因此配置费用会有差异。 任何一个集群中均包含“Master Node”和“Worker Node”，每一个 Node 对应一台云服务器。 • Master Node：集群的控制节点，在创建集群时会自动创建控制节点，负责整个集群的管理和调度。

参数	参数说明
	• Worker Node：集群的工作节点，即用户购买或纳管的节点。工作负载是由控制节点分配的，当某个工作节点宕机时，控制节点会将工作负载转移到其他工作节点上。
高可用	• 高可用模式开启后将创建多个控制节点，在单个控制节点发生故障后集群可以继续使用，不影响业务功能。 • 高可用模式开关在集群创建完成后不可变更 • 商用场景建议选择高可用模式集群。
虚拟私有云	新建集群所在的虚拟私有云，集群创建后不可更改。 虚拟私有云是通过逻辑方式进行网络隔离，提供安全、隔离的网络环境。 若没有虚拟私有云可选择，请单击“创建虚拟私有云”进行创建，完成创建后单击刷新按钮。
所在子网	节点虚拟机运行的子网环境，集群创建后不可更改。 通过子网提供与其他网络隔离的、可以独享的网络资源，以提高网络安全。 若没有子网可选择，请单击“创建子网”进行创建，完成创建后单击刷新按钮。虚拟私有云、子网、集群的关系请参见 3.1 集群概述。 **请确保子网下的 DNS 服务器可以解析 OBS 服务域名，否则无法创建节点。** **集群创建后子网无法修改，请谨慎选择。**
网络模型	集群创建成功后，网络模型不可更改，请谨慎选择。 **容器隧道网络** 容器隧道网络下只能添加同一类型的节点，即全部为虚拟机节点或全部为裸金属节点。 • 基于底层 VPC 网络，另构建了独立的 VXLAN 隧道化容器网络，适用于一般场景。 • VXLAN 是将以太网报文封装成 UDP 报文进行隧道传输。容器网络是承载于 VPC 网络之上的 Overlay 网络平面，具有付出少量隧道封装性能损耗，即可获得通用性强、互通性强、高级特性支持全面（例如 Network Policy 网络隔离）的优势，可以满足大多数应用需求。 **VPC 网络** VPC 网络模式下每个节点占用一条 VPC 路由规则，Console 界面中可显示当前局点支持的 VPC 路由规则条数，以及每个节点可供分配的容器 IP 个数（即可创建的 Pod 实例数目上限）， • 采用 VPC 路由方式与底层网络深度整合，适用于高性能场景，但每个节点占用一条 VPC 路由规则，节点数量受限于虚拟私有云 VPC 的路由配额。 • VPC 网络集群下的每个节点将会被分配固定大小的 IP 地址段，

参数	参数说明
	由于没有隧道封装的消耗，容器网络性能相对于容器隧道网络有一定优势。VPC 网络集群由于 VPC 路由中配置有容器网段与节点 IP 的路由，可以支持集群外直接访问容器实例等特殊场景。 说明 • VPC 网络模式下不支持扩展网段和 NetworkPolicy。 • **一个 VPC 下创建多个 VPC 网络模式的集群时，需要为每个集群选择一个不重叠的地址段，不仅不能和 VPC 地址重叠，也不和其他容器网段重叠。**
容器网段	请根据业务需求选择容器网段，确定容器网段后，容器实例将在规划的网段内分配 IP，集群创建后该网段不可更改。 • 未勾选“自动选择”：请手动选择网段。若与子网网段有冲突时将有红色文字提示，请重新选择。建议使用网段：10.0.0.0/8~18，172.16.0.0/16~18，192.168.0.0/16~18。 **不同集群使用相同的容器网段，会导致容器 IP 冲突，应用访问异常。** • 勾选“自动选择”：系统将自动分配与子网网段无冲突的网段。 容器网段要设置合理的掩码，掩码决定集群内可用节点数量。集群中容器网段掩码设置不合适，会导致集群实际可用的节点较少。设置掩码后，选项下方会有当前网段最多支持的实例估算值，请作参考。
服务网段	服务网段为 kubernetes service ip 网段，集群创建后该网段不可更改。服务网段与已创建的路由不能冲突，如果冲突，请重新选择。 • 使用默认网段：默认设置为 10.247.0.0/16 网段。 • 手动设置网段：请根据业务需求设置合理的网段和掩码，掩码决定集群内可用 service ip 数量。
认证方式	认证机制主要用于对集群下的资源做权限控制。例如 A 用户只能对某个命名空间下的应用有读写权限，B 用户对集群下的资源只有读权限等。角色权限控制的操作请参见 3.10 集群管理权限控制。 • 默认状态下不选定“认证能力增强”，此时默认开启 X509 认证模式，X509 是一种非常通用的证书格式。 • 若需要对集群进行权限控制，请勾选“认证能力增强”，选择“认证代理”。 单击“CA 根证书”后的“上传文件”，上传符合规范且合法的证书，并**勾选“我已确认上传的证书合法”**。 证书若不合法，集群将无法创建成功。请上传小于 1MB 的文件，上传格式支持.crt 或.cer 格式。
集群描述	选填，请输入新建容器集群相应的描述信息。

参数	参数说明
高级设置	单击“高级设置”后展开详细项目，支持的功能如下（当前可用区中不支持的功能将隐藏）： **服务转发模式：** • iptables：社区传统的 kube-proxy 模式，完全以 iptables 规则的方式来实现 service 负载均衡。该方式最主要的问题是在服务多的时候产生太多的 iptables 规则，非增量式更新会引入一定的时延，大规模情况下有明显的性能问题。 • ipvs：在社区获得广泛支持的 kube-proxy 模式，采用增量式更新，吞吐更高，速度更快，并可以保证 service 更新期间连接保持不断开，适用于大规模场景。 ipvs 模式下，ingress 和 service 使用相同的 ELB 实例时，无法在集群内的节点和容器中访问 ingress。 说明 • ipvs 为大型集群提供了更好的可扩展性和性能。 • ipvs 支持比 iptables 更复杂的负载平衡算法（最小负载，最少连接，位置，加权等）。 • ipvs 支持服务器健康检查和连接重试等。 **CPU 管理策略：** • 开启：支持给工作负载实例配置 CPU 独占，适用于对 CPU 缓存和调度延迟敏感的工作负载。 • 关闭：关闭工作负载实例独占 CPU 核的功能，优点是 CPU 共享池的可分配核数较多。 更多 CPU 管理策略内容请参见 Feature Highlight: CPU Manager。 **控制节点可用区：**多可用区模式支持集群管理面多可用区容灾，但是对于集群性能有所损耗。
购买时长	若选择创建“包年/包月”的集群，请设置购买时长。

步骤 3 单击“下一步：创建节点”，在“创建节点”步骤中，参照如下参数配置节点：

- **创建节点：**
  - 现在添加：创建集群的同时创建节点，当前仅支持虚拟机节点。如果节点创建失败集群会一起回滚。
  - 稍后添加：将不会创建节点，仅创建一个空集群，集群创建完成后可以添加虚拟机或裸金属节点。
- **计费模式：**支持“包年/包月”和“按需计费”两种计费类型。
  - 包年/包月：包年包月是预付费模式，按订单的购买周期计费，适用于可预估资源使用周期的场景，价格比按需计费模式更优惠。
  - 按需计费：按需计费是后付费模式，按资源的实际使用时长计费，可以随时开通/删除资源。

创建集群时节点的计费方式跟随集群的计费方式，如集群的计费模式选择“按需计费”，则创建过程中节点的计费模式只能为“按需计费”，“包年/包月”同理。创建方式请参考 4.2 购买节点。

包年/包月节点创建后不能删除，如需停止使用，请执行退订操作。

- **当前区域：**节点实例所在的物理位置。
- **可用区：**请根据业务需要进行选择。可用区是在同一区域下，电力、网络隔离的物理区域，可用区之间内网互通，不同可用区之间物理隔离。

  如果您需要提高工作负载的高可靠性，建议您在创建集群后将云服务器部署在不同的可用区，购买集群时节点只能部署在一个可用区。
- **节点类型：**选择“虚拟机节点”。
- **节点名称：**自定义节点名称。长度范围为 1-56 个字符，以小写字母开头，支持小写字母、数字、中划线(-)，不能以中划线(-)结尾。
- **节点规格：**请根据业务需求选择相应的节点规格。
  - 通用型：该类型实例提供均衡的计算、存储以及网络配置，适用于大多数的使用场景。通用型实例可用于 Web 服务器、开发测试环境以及小型数据库工作负载等场景。
  - 内存优化型：该类型实例提供内存比例更高的实例，可以用于对内存要求较高、数据量大的工作负载，例如关系数据库、NoSQL 等场景。
  - GPU 加速型：提供优秀的浮点计算能力，从容应对高实时、高并发的海量计算场景。P 系列适合于深度学习，科学计算，CAE 等；G 系列适合于 3D 动画渲染，CAD 等。
  - 高性能计算型：实例提供具有更稳定、超高性能计算性能的实例，可以用于超高性能计算能力、高吞吐量的工作负载场景，例如科学计算。
  - 通用计算增强型：该类型实例具有性能稳定且资源独享的特点，满足计算性能高且稳定的企业级工作负载诉求。

  为确保节点稳定性，系统会自动预留部分资源，用于运行必须的系统组件。详细请参见 4.12 节点预留资源计算公式。
- **操作系统：**请直接选择节点对应的操作系统。
- **系统盘：**设置工作节点的系统盘空间。您可以设置系统盘的规格为 40GB-1024GB 之间的数值，缺省值为 40GB。

  在默认情况下，系统盘可提供高 IO、超高 IO 两种基本的云硬盘类型。
- **数据盘：**设置工作节点的数据盘空间。您可以设置数据盘的规格为 100GB-32678GB 之间的数值，缺省值为 100GB。数据盘可提供的云硬盘类型与上方系统盘一致。

⚠ 注意

若数据盘卸载或损坏，会导致 docker 服务异常，最终导致节点不可用。建议不要删除该数据盘。

- 数据盘空间分配：单击后方的 更改配置，可以对数据盘中的“k8s 空间”和“用户空间”占比进行自定义设置，开启 LVM 管理的数据盘将按照设置的比例进行统一分配。部分集群版本不支持此功能，具体以界面为准。
  - k8s 空间：您可以自定义数据盘中 Docker 和 Kubelet 的资源占比。Docker 资源包含 Docker 工作目录、Docker 镜像数据以及镜像元数据；Kubelet 资源包含 Pod 配置文件、密钥以及临时存储 EmptyDir 等挂载数据。
  - 用户空间：定义本地盘中不分配给 kubernetes 使用的空间大小和用户空间挂载路径。

说明

请注意“挂载路径”不能设置为根目录“/”，否则将导致挂载失败。挂载路径一般设置为：

- /opt/xxxx（但不能为/opt/cloud）
- /mnt/xxxx（但不能为/mnt/paas）
- /tmp/xxx
- /var/xxx （但不能为/var/lib、/var/script、/var/paas 等关键目录）
- /xxxx（但不能和系统目录冲突，例如 bin、lib、home、root、boot、dev、etc、lost+found、mnt、proc、sbin、srv、tmp、var、media、opt、selinux、sys、usr 等）

注意不能设置为/home/paas、/var/paas、/var/lib、/var/script、/mnt/paas、/opt/cloud，否则会导致系统或节点安装失败。

- **虚拟私有云：**不可修改，仅用于展示当前集群所在的虚拟私有云。
- **所在子网：**通过子网提供与其他网络隔离的、可以独享的网络资源，以提高网络安全。可选择该集群虚拟私有云下的任意子网，集群节点支持跨子网。

  该参数仅在 v1.13.10-r0 及以上版本的集群中显示，请务必**确保子网下的 DNS 服务器可以解析 OBS 服务域名**，否则无法创建节点。

  已有集群添加节点时，如果子网对应的 VPC 新增了扩展网段且子网是扩展网段，要在控制节点安全组（即集群名称-cce-control-随机数）中添加如下三条安全组规则，以保证集群添加的节点功能可用（新建集群时如果 VPC 已经新增了扩展网段则不涉及此场景）：

- **弹性 IP**：独立申请的公网 IP 地址，若节点有互联网访问的需求，请选择“暂不使用”或“使用已有”。**集群开启 IPv6 时，不显示该参数。**

  弹性公网 IP 提供外网访问能力，可以灵活绑定及解绑，随时修改带宽。未绑定弹性公网 IP 的云服务器无法直接访问外网，无法直接对外进行互相通信。

  - 暂不使用：若新增节点未绑定弹性 IP，则在该节点上运行的工作负载将不能被外网访问，仅可作为私有网络中部署业务或者集群所需云服务器进行使用。
  - 使用已有：请选择已有的弹性 IP，将为当前节点分配已有弹性 IP。

  说明

  CCE 默认不启用 VPC 的 SNAT。若 VPC 启用了 SNAT，可以不使用 EIP 去访问外网。SNAT 具体请参见自定义策略。

- **共享带宽**：请选择“暂不使用”或“使用已有”。**仅在集群开启 IPv6 时，显示该参数。**

  弹性公网 IP 提供外网访问能力，可以灵活绑定及解绑，随时修改带宽。未绑定弹性公网 IP 的云服务器无法直接访问外网，无法直接对外进行互相通信。

- **登录方式**：支持密码和密钥对。
  - 选择“密码”：用户名默认为“root”，请输入登录节点的密码，并确认密码。

    登录节点时需要使用该密码，请妥善管理密码，系统无法获取您设置的密码内容。
  - 选择“密钥对”：选择用于登录本节点的密钥对，支持选择共享密钥。

    密钥对用于远程登录节点时的身份认证。若没有密钥对，可单击选项框右侧的“创建密钥对”来新建。

须知

如果子用户创建节点选择密钥对创建，这个密钥只对创建这个密钥的子用户有效，即使其他子用户在同一个组也无法选择，也无法使用。例如：A 用户创建的密钥，B 用户无法使用这个密钥对创建节点，并且 Console 也选不到。

- **云服务器高级设置：**（可选），单击^展开后可对节点进行如下高级功能配置：
  - **安装前执行脚本：**请输入脚本命令，大小限制为 0~1000 字符。

    脚本将在 Kubernetes 软件安装前执行，可能导致 Kubernetes 软件无法正常安装，需谨慎使用。常用于格式化数据盘等场景。
  - **安装后执行脚本：**请输入脚本命令，大小限制为 0~1000 字符。

    脚本将在 Kubernetes 软件安装后执行，不影响 Kubernetes 软件安装。常用于修改 Docker 配置参数等场景。
  - **新增数据盘：**单击“新增数据盘”，选择云硬盘类型并输入数据盘规格。
  - **子网 IP：**可选择“自动分配 IP 地址”和“手动分配 IP 地址”，推荐使用“自动分配 IP 地址”。
- **Kubernetes 高级设置：**（可选），单击^展开后可对集群进行如下高级功能配置：
  - **最大实例数：**节点最大允许创建的实例数(Pod)，该数量包含系统默认实例，取值范围为 16~128。

    该设置的目的为防止节点因管理过多实例而负载过重，请根据您的业务需要进行设置。
  - **自定义镜像仓库：**单击“新增自定义镜像仓库地址”输入镜像仓库地址。

    添加自定义镜像仓库地址（非 SSL 镜像源地址）到 docker 启动参数中，避免拉取个人镜像仓库的镜像失败，格式可为“IP 地址:端口或者域名”。安装后执行脚本与自定义镜像仓库不能同时使用。
  - **单容器可用数据空间：**该参数用于设置一个容器可用的数据空间大小，设置范围为 10G 到 80G。如果设置的参数超过数据盘中 Docker 可占用的实际数据空间（由数据盘设置项中的资源分配自定义参数指定，默认为数据盘大小的 90%），将以 Docker 的实际空间大小为主。该参数仅在 v1.13.10-r0 及以上版本的集群中显示。
- **节点购买数量：**此处设置的节点数不能超过集群管理的最大节点规模，请根据业务需求和界面提示进行选择，单击后方的(?)可查看影响能添加节点数的因素（取决于最小值）。如需申请更多配额，请单击。
- **购买时长：**若选择创建“包年/包月”的集群，请设置购买时长。

步骤 4 单击“下一步：安装插件”，在“安装插件”步骤中选择要安装的插件。

“系统资源插件”为必装插件，“高级功能插件”可根据实际需求进行选择性安装。

所有插件也可以在集群创建完成后，在左侧导航栏中单击“插件管理”进行安装或卸载，具体请参见 13 插件管理 。

步骤 5 单击“下一步：配置确认”，阅读“使用说明”并点选“我已知晓上述限制”，确认所设置的服务选型参数、规格和费用等信息。

步骤 6 确认规格和费用后，单击“提交”，集群开始创建。

若选择购买“包年包月”的集群，请单击“去支付”，根据界面提示进行付款操作。

集群创建预计需要 6-10 分钟，您可以单击“返回集群管理”进行其他操作或单击“查看集群事件列表”后查看集群详情。待集群状态为“正常”，表示集群创建成功。

----结束

相关操作

- 通过命令行工具连接集群：请参见 3.4.2 通过 kubectl 操作 CCE 集群。
- 添加节点：集群创建完成后，若您需要为集群添加更多节点，请参见 4.2 购买节点。
- 登录节点：请参见 4.4 登录节点。
- 创建命名空间：同个集群内可创建多个命名空间，形成逻辑上的不同分组，便于不同的分组在共享使用集群资源时还能被分别管理。若您需要为集群创建命名空间，请参见 10 命名空间。
- 创建工作负载：集群创建完成后，您可以使用镜像创建一个可公网访问的应用，请参见 6.2 创建无状态负载(Deployment)或 6.3 创建有状态负载(StatefulSet)。
- 单击已成功创建的集群名称，进入“集群详情”页可查看集群详情。

表3-4 已创建的集群详情

页签类别	说明
集群详情	可查看该集群的详情及运行状态等。
监控	可查看集群下全部节点的 CPU 和内存分配率（即分配量的最大值），以及控制节点的 CPU 和内存使用率、控制节点规格等信息。
事件	• 可以直接在“事件”页签下查看集群的事件。 • 可以设置查询条件，比如设置事件产生的时间段或搜索事件名称，查看相关事件。
弹性扩容	您可以根据实际业务需要对集群的工作节点进行扩容和缩容，详情请参见 3.5 集群弹性扩容。 v1.17 版本的集群不再支持 AOM 提供的弹性伸缩机制，请使用节点池功能进行弹性伸缩，详情请参见 5.1 节点池概述。
kubectl	若您需要从客户端计算机连接到 kubernetes 集群，请使用 kubernetes 命令行客户端 kubectl，详情请参见 3.4.2 通过 kubectl 操作 CCE 集群。

3.4 Kubectl 访问集群

3.4.1 Kubectl 使用指南

使用 Kubectl 命令时，您需要具备 Kubectl 相关开发技能，了解 Kubectl 相关操作。详细请参考 Kubernetes API、kubectl CLI。

请到 kubernetes 版本发布页面下载与集群版本对应的或者更新的 kubectl。

连接集群

3.4.2 通过 kubectl 操作 CCE 集群

CoreDNS 高可用

- 3.4.3 通过 kubectl 配置 kube-dns/CoreDNS 高可用

创建工作负载

- 通过 Kubectl 创建无状态工作负载
- 通过 kubectl 创建有状态工作负载

工作负载亲和/反亲和调度

- 工作负载与节点亲和-yaml 示例
- 工作负载与节点反亲和-yaml 示例
- 工作负载间亲和-yaml 示例
- 工作负载间反亲和-yaml 示例
- 工作负载与可用区亲和-yaml 示例
- 工作负载与可用区反亲和-yaml 示例

工作负载访问方式设置

- 通过 kubectl 命令实现集群内访问
- 通过 kubectl 命令实现节点访问
- 通过 kubectl 命令行实现四层负载均衡
- 8.3.3 通过 Kubectl 命令行创建 Ingress

工作负载高级设置

- 设置容器生命周期-Yaml 样例

任务管理

- 使用 kubectl 创建 Job
- 使用 kubectl 创建 CronJob

配置中心

- 使用 kubectl 创建配置项
- 使用 kubectl 创建密钥

存储管理

- 9.3.3 使用 kubectl 自动创建云硬盘
- 9.3.5 使用 kubectl 部署带云硬盘存储卷的工作负载

3.4.2 通过 kubectl 操作 CCE 集群

操作场景

本文将以 CCE 集群为例，介绍如何通过 kubectl 或 web-terminal 插件操作 CCE 集群。

使用 kubectl 操作集群

背景信息

若您需要从客户端计算机连接到 kubernetes 集群，请使用 kubernetes 命令行客户端 kubectl。

前提条件

CCE 支持“VPC 网络内访问”和“互联网访问”两种方式访问集群。

- VPC 网络内访问：您需要在 ECS 控制台购买一台云服务器，并确保和当前集群在同个 VPC 内。
- 互联网访问：您需要准备一台能连接公网的云服务器。

须知

通过“互联网访问”方式访问集群，集群的 kube-apiserver 将会暴露到互联网，存在被攻击的风险，建议对 kube-apiserver 所在节点的 EIP 配置 DDoS 高防服务。

下载 kubectl

您需要先下载 kubectl 以及配置文件，拷贝到您的客户端机器，完成配置后，即可以使用访问 kubernetes 集群。

请到 kubernetes 版本发布页面下载与集群版本对应的或者更新的 kubectl。

安装和配置 kubectl

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 集群管理”，单击待连接集群下的“命令行工具 > kubectl”。

步骤 2 在集群详情页中的“kubectl”页签下，请参照界面中的提示信息完成集群连接。

说明

- 您可以在“kubectl”页签下方便的**下载 kubectl 配置文件**（kubeconfig.json），CCE 支持帐号和 IAM 用户分别下载该配置文件，IAM 用户下载的配置文件只有 30 天的有效期，而帐号下载的配置文件会长期有效。该文件用于对接认证用户集群，请用户妥善保存该认证凭据，防止文件泄露后，集群有被攻击的风险。如果不幸泄露，可以通过证书更新的方式，替换认证凭据。
- 由于 EIP 无法固定，所以服务器端证书无法预置 EIP，若从互联网访问则无法开启客户端校验服务器端。如需开启客户端校验服务器端，请使用 VPC 访问方式。
- IAM 用户下载的配置文件所拥有的 Kubernetes 权限与 CCE 控制台上 IAM 用户所拥有的权限一致。

----结束

相关操作

连接集群后，您可以使用 Kubernetes 管理工作负载，具体请参见 3.4.1 Kubectl 使用指南。

3.4.3 通过 kubectl 配置 kube-dns/CoreDNS 高可用

操作场景

通过 kubernetes 命令行工具来配置 kube-dns/CoreDNS 的高可用。

kube-dns/CoreDNS 为集群提供了 DNS 服务。若集群中只部署了单个 kube-dns/CoreDNS，则存在 kube-dns/CoreDNS 故障后影响整个集群运作的风险，因此建议为集群配置多个 kube-dns/CoreDNS。

说明

- 在 CCE 中新建的 kubernetes 1.11 及以上版本集群默认会安装 CoreDNS 插件。
- 更多 DNS 相关信息请参见 13.2 CoreDNS（系统资源插件，必装）或 6.10.11 Kubernetes 集群内置 DNS 配置说明。

前提条件

集群能够被互联网访问，或集群与客户端机器在同一个 VPC 下。

操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 集群管理”，单击待连接集群下的“kubectl”。

步骤 2 根据界面提示的步骤设置集群 API 访问方式。

步骤 3 根据界面提示的步骤配置命令行工具。

命令行工具配置成功后，可使用命令行工具来手动配置 kube-dns/CoreDNS 高可用。

步骤 4 登录客户端机器。

步骤 5 编辑 kube-dns/CoreDNS 的 deployment 配置文件。

下面以 CoreDNS 为例，执行命令如下：

kubectl edit deployment coredns -n kube-system

修改 deployment 配置文件中 spec 字段下的 replicas 参数为所需的 CoreDNS 实例数。

示例：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
  creationTimestamp: 2019-02-11T09:36:04Z
  generation: 1
  labels:
    app: coredns
    kubernetes-app: coredns
    kubernetes.io/cluster-service: "true"
    kubernetes.io/name: CoreDNS
    release: cceaddon-coredns
  name: coredns
  namespace: kube-system
  resourceVersion: "1927"
  selfLink: /apis/extensions/v1beta1/namespaces/kube-system/deployments/coredns
  uid: 737b9296-2de0-11e9-b629-fa163e7fb882
spec:
  progressDeadlineSeconds: 600
  replicas: 2
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: coredns
      kubernetes-app: coredns
  strategy:
    rollingUpdate:
      maxSurge: 10%
      maxUnavailable: 0
    type: RollingUpdate
  template:
    metadata:
      annotations:
        checksum/config:
3095a9b4028195e7e0b8b22c550bf183d0b7a8a7eba20808b36081d0b39f8b81
```

----结束

3.4.4 Kubectl 常用命令

基础命令

get

get 命令用于获取集群的一个或一些 resource 信息。

该命令可以列出集群所有资源的详细信息，resource 包括集群节点、运行的 pod、Replication Controller、service 等。

须知

集群中可以创建多个 namespace，未指定 namespace 的情况下，所有操作都是针对--namespace=default。

例如：

获取所有 pod 的详细信息：

```
kubectl get po -o wide
```

获取所有 namespace 下的运行的所有 pod：

```
kubectl get po --all-namespaces
```

获取所有 namespace 下的运行的所有 pod 的标签：

```
kubectl get po --show-labels
```

获取该节点的所有命名空间：

```
kubectl get namespace
```

说明

查询其他节点需要加-s 指定节点，类似可以使用“kubectl get rc”，“kubectl get svc”，“kubectl get nodes”，“kubectl get deploy”等获取其他 resource 信息。

以 yaml 格式输出 pod 的详细信息：

```
kubectl get po <podname> -o yaml
```

以 json 格式输出 pod 的详细信息：

```
kubectl get po <podname> -o json
kubectl get po rc-nginx-2-btv4j -o=custom-columns=LABELS:.metadata.labels.app
```

说明

其中 LABELS 为显示的列标题，可以自己设置，“.metadata.labels.app”为查询的数据需要按照之前的 yaml 或 json 获取。

create

kubectl 命令用于根据文件或输入创建集群 resource。

如果已经定义了相应 resource 的 yaml 或 json 文件，直接 kubectl create -f filename 即可创建文件内定义的 resource。

```
kubectl create -f filename
```

expose

expose 将一个资源包括 pod、Replication Controller、service、deployment 等公开为一个新的 service。

```
kubectl expose deployment deployname --port=81 --type=NodePort --target-port=80 --
name=service-name
```

说明

给 deployname 发布一个服务，-port 为暴露出去的端口，-type 为服务类型，-target-port 为容器端口，port 通过 clusterip 加端口访问，target-port 通过节点 ip 加端口访问。

run

例如：

在集群中运行一个特定的镜像。

```
kubectl run deployname --image=nginx:latest
```

在创建时指定运行的命令：

```
kubectl run deployname -image=busybox --command -- ping baidu.com
```

set

在对象上设置特定功能。

例如：

将一个 deployname 的 image 改为镜像为 1.0 的 image：

```
kubectl set image deploy deployname containername=containername:1.0
```

edit

edit 提供了另一种更新 resource 源的操作。

例如：

使用 edit 直接更新 pod 的命令为：

```
kubectl edit po po-nginx-btv4j
```

上面命令的效果等效于：

```
kubectl get po po-nginx-btv4j -o yaml >> /tmp/nginx-tmp.yaml
vim /tmp/nginx-tmp.yaml
/*do some changes here */
kubectl replace -f /tmp/nginx-tmp.yaml
```

explain

查看文档或参考资料。

例如：

查看 pods/service 的相关文档：

```
kubectl explain pods,svc
```

delete

根据 resource 名或 label 删除 resource。

例如：

立刻删除该 pod：

```
kubectl delete po podname --now
kubectl delete -f nginx.yaml
kubectl delete deployment deployname
```

部署命令

rolling-update*

rolling-update 是一个非常重要的命令，对于已经部署并且正在运行的业务，rolling-update 提供了不中断业务的更新方式。rolling-update 每次起一个新的 pod，等新 pod 完全起来后删除一个旧的 pod，然后再起一个新的 pod 替换旧的 pod，直到替换掉所有的 pod。rolling-update 需要确保新的版本有不同的 name，Version 和 label，否则会报错 。

```
kubectl rolling-update poname -f newfilename
kubectl rolling-update poname -image=image:v2
```

如果在升级过程中，发现有问题可以中途停止 update，并回滚到前面版本。

```
kubectl rolling-update poname -rollback
```

rollout

管理资源的发布。

例如：

查看指定资源的部署状态：

```
kubectl rollout status deployment/deployname
```

查看指定资源的发布历史：

```
kubectl rollout history deployment/deployname
```

回滚指定资源，默认回滚至上一个版本：

```
kubectl rollout undo deployment/test-nginx
```

scale

scale 用于程序在负载加重或缩小时将副本进行扩容或缩小。

```
kubectl scale deployment deployname --replicas=newnumber
```

autoscale

autoscale 命令提供了自动根据 pod 负载对其副本进行扩缩的功能。autoscale 命令会给一个 rc 指定一个副本数的范围，在实际运行中根据 pod 中运行的程序的负载自动在指定的范围内对 pod 进行扩容或缩容。

```
kubectl autoscale deployment deployname --min=minnumber --max=maxnumber
```

集群管理命令

cordon、drain、uncordon*

有时候会遇到这样一个场景，一个 node 需要升级，但是在该 node 上又有许多运行的 pod，或者该 node 已经瘫痪，需要保证功能的完善，则需要使用这组命令，使用步骤如下：

步骤 1 使用 cordon 命令将一个 node 标记为不可调度。这意味着新的 pod 将不会被调度到该 node 上。

```
kubectl cordon nodename
```

步骤 2 使用 drain 命令，将运行在该 node 上运行的 pod 平滑的搬迁到其他节点上。

```
kubectl drain newnodename
```

步骤 3 对该节点进行一些节点维护的操作，如升级内核、升级 Docker 等。

步骤 4 节点维护完后，使用 uncordon 命令解锁该 node，使其重新变得可调度。

```
kubectl uncordon nodename
```

----结束

cluster-info

查看在集群中运行的插件：

```
kubectl cluster-info
```

查看详细信息：

```
kubectl cluster-info dump
```

top*

显示资源（CPU/Memory/Storage）使用。需要 Heapster 运行。

taint*

修改一个或多个节点上的 taint。

certificate*

修改证书资源。

故障诊断和调试命令

describe

describe 类似于 get，同样用于获取 resource 的相关信息。不同的是，get 获得的是更详细的 resource 个性的详细信息，describe 获得的是 resource 集群相关的信息。describe 命令同 get 类似，但是 describe 不支持-o 选项，对于同一类型 resource，describe 输出的信息格式，内容域相同。

说明

如果发现是查询某个 resource 的信息，使用 get 命令能够获取更加详尽的信息。但是如果想要查询某个 resource 的状态，如某个 pod 并不是在 running 状态，这时需要获取更详尽的状态信息时，就应该使用 describe 命令。

```
kubectl describe po <podname>
```

logs

logs 命令用于显示 pod 运行中，容器内程序输出到标准输出的内容。如果要获得 tail -f 的方式，需使用-f 选项。

```
kubectl logs -f podname
```

exec

与 docker 的 exec 用法相似，如果一个 pod 中，有多个容器，需要使用-c 选项指定容器。

```
kubectl exec -it podname bash
kubectl exec -it podname -c containername bash
```

port-forward*

转发一个或多个本地端口至一个 pod。

例如：

转发 pod 中的 6000 端口到本地的 5000 端口：

```
kubectl port -forward podname 5000:6000
```

proxy*

运行一个 proxy 到 kubernetes api server。

例如：

控制节点开启 HTTP Rest 接口：

```
kubectl proxy -accept-hosts='.*' -port=8001 -address='0.0.0.0'
```

cp

拷贝文件或目录到容器：

```
cp filename newfilename
```

auth*

检查授权。

attach*

attach 命令效果类似于 logs -f，退出查看使用 ctrl-c。如果一个 pod 中有多个容器，要查看具体的某个容器的输出，需要在 pod 名后使用-c containername 指定运行的容器。

```
kubectl attach podname -c containername
```

高级命令

replace

replace 命令用于对已有资源进行更新、替换。当我们需要更新 resource 的一些属性的时候，如果修改副本数量，增加、修改 label，更改 image 版本，修改端口等。都可以直接修改原 yaml 文件，然后执行 replace 命令。

```
kubectl replace -f filename
```

须知

名字不能被更新。另外，如果是更新 label，原有标签的 pod 将会与更新 label 后的 rc 断开联系，有新 label 的 rc 将会创建指定副本数的新的 pod，但是默认并不会删除原来的 pod。所以此时如果使用 get po 将会发现 pod 数翻倍，进一步 check 会发现原来的 pod 已经不会被新 rc 控制。

apply*

apply 命令提供了比 patch，edit 等更严格的更新 resource 的方式。通过 apply，用户可以将 resource 的 configuration 使用 source control 的方式维护在版本库中。每次有更新时，将配置文件 push 到 server，然后使用 kubectl apply 将更新应用到 resource。kubernetes 会在引用更新前将当前配置文件中的配置同已经应用的配置做比较，并只更新更改的部分，而不会主动更改任何用户未指定的部分。apply 命令的使用方式同 replace 相同，不同的是，apply 不会删除原有 resource，然后创建新的。apply 直接在原有 resource 的基础上进行更新。同时 kubectl apply 还会在 resource 中添加一条注释，标记当前的 apply，类似于 git 操作。

```
kubectl apply -f
```

patch

如果一个容器已经在运行，这时需要对一些容器属性进行修改，又不想删除容器，或不方便通过 replace 的方式进行更新。kubernetes 还提供了一种在容器运行时，直接对容器进行修改的方式，就是 patch 命令。 例如已存在一个 pod 的 label 为 app=nginx1，如果需要在运行过程中，将其修改为 app=nginx2。

```
kubectl patch pod podname -p '{"metadata":{"lables":{"app":"nginx1"}}}'
```

convent*

不同的 api 版本之间转换配置文件。

设置命令

label

更新资源上的标签：

```
kubectl label pods my-pod new-label=newlabel
```

annotate

更新资源上的注释：

```
kubectl label pods my-pod icon-url=http://……
```

completion

用于实现 kubectl 工具自动补全。

其他命令

api-versions

打印受支持的 api 版本：

```
kubectl api-versions
```

api-resources

打印支持的 api 资源：

```
kubectl api-resources
```

config*

修改 kubeconfig 文件：用于访问 api，比如配置认证信息。

help

所有命令帮助。

version

打印客户端和服务版本信息。

```
kubectl version
```

3.5 集群弹性扩容

操作场景

通过云容器引擎管理控制台，您可以根据实际业务需要对集群的工作节点进行扩容和缩容，当集群中出现由于资源不足而无法调度的工作负载时自动触发扩容，从而减少人力成本。

约束与限制

- 该功能仅支持通过**按需计费**方式购买的**虚拟机节点**，不支持“包年/包月”方式购买的节点。
- 目前不支持集群中 Master 节点的扩容和缩容。

- 如果您有集群自动扩缩容的需求，请通过 autoscaler 插件实现，具体请参见 13.5 autoscaler。
- v1.17 及以上版本的集群不再支持 AOM 提供的弹性伸缩机制，请使用节点池功能进行弹性伸缩，详情请参见 5.1 节点池概述。

集群自动扩容

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 集群管理”，单击待设置伸缩策略集群下的“更多 > 弹性扩容”。

步骤 2 在“扩容配置”页签，单击“编辑”，为集群的弹性扩容设置冷却时间、集群最大节点数和节点配置。

表3-5 扩容配置

参数	参数说明
冷却时间	扩容策略执行后停止继续匹配的时间，目的是等待扩容动作完成后在系统稳定且集群正常的情况下进行下一次策略匹配。 取值范围为：60 秒~3600 秒，默认为 900 秒。由于节点创建时间需要 2-10 分钟，冷却时间小于 900 秒可能无法达到预期。
单次扩容节点数上限	扩容策略执行时，集群下最大节点数。 例如集群最大节点数为 x，取值范围：1≤x<集群节点配额。 **说明** **集群节点配额受两处限制，一是集群的规模，即单集群的节点数量。二是帐户的节点配额。此处的集群节点配额数，取两处限制中配额较少的。**
节点配置	如果扩容策略执行后需要扩容，则系统会创建节点。 1. 单击“设置”，配置创建节点的各项参数。创建节点的参数配置请参见购买节点。此处计费方式只能选择按需计费。 2. 完成节点配置后，单击“提交”。

步骤 3 确认扩容配置及节点参数后，单击“确定”。

步骤 4 为集群设置扩容策略，选择“扩容策略”页签，单击“添加扩容策略”。

- 策略名称：自定义策略名称，例如 policy01。
- 选择“策略类型”，当前支持三种弹性扩容策略：
  - 表 3-6：支持根据集群 CPU/内存分配量的设置，进行集群的自动扩容。

表3-6 添加告警策略

参数	参数说明
* 指标	支持“CPU 分配量”指标和“内存分配量”指标。
* 触发条件	设置触发扩容策略的条件，即 CPU 或内存的分配量的平均值大于或小于某个百分比数，将触发扩容。
* 监控窗口	数据的汇聚窗口大小。单击下拉选项进行选择。 若设置为 15 分钟，表示每 15 分钟监控一次。
* 连续周期	指监控窗口内连续触发阈值的次数，计算周期固定一分钟。 若设置为 3，则表示指标数据连续三个监控窗口达到了设定的阈值，则触发策略动作。
* 执行操作	策略触发后执行的动作。

- 表 3-7：支持在特定时间点进行集群的自动扩容。

表3-7 添加定时策略

参数	参数说明
* 策略类型	选择定时策略。
* 触发时间	策略触发时间。
* 执行操作	策略触发后执行的动作。

- 表 3-8：支持以天、周、月为周期的扩容策略。

表3-8 添加周期策略

参数	参数说明
* 策略类型	选择周期策略。
* 选择时间	选择策略触发的时间。
* 执行操作	策略触发后执行的动作。

步骤 5 单击“确定”。

伸缩完成后，在左侧导航栏中选择“资源管理 > 节点管理”，在查看节点列表页面，可查看到扩容的工作节点。

----结束

3.6 集群升级

3.6.1 集群版本升级说明

当某个集群版本支持升级能力后，您可以随时进行升级操作。

您可以在 CCE 控制台中确认您的集群是否可以进行升级操作，**确认方法：**单击“资源管理 > 集群管理”，查看待升级集群右上角是否存在“可升级”提示，若存在则该集群支持升级，若不存在，则该集群不支持升级。

注意事项

- 当前仅支持虚拟机节点的 CCE 集群升级。
- 集群升级时若需要升级 CoreDNS 插件，请确保节点数大于等于 CoreDNS 的实例数，且 CoreDNS 的所有实例都处于运行状态，否则将导致升级失败。1.13 版本的集群若需升级到更高版本，需先将 CoreDNS 插件升级当前集群可用的最新版本。
- 1.13 版本内升级时，集群上的应用只会在升级网络组件时有短暂中断。

集群升级路径

如下表格详细介绍了各集群版本能够升级到的目标版本，以及升级方式和升级影响。

表3-9 集群升级路径和影响说明

源版本	目标版本	升级方式	升级影响
v1.13	v1.15	滚动升级 重置升级	• coredns 配置中 proxy 配置项不支持，需要替换成 forword。 • 存储插件从原来的 storage-driver 替换成了 Everest。

升级方式

集群升级页面会根据集群版本以及不同局点提供三种升级方式，三种方式的控制节点升级流程是一致的，用户节点的升级方式区别及优缺点如下：

表3-10 升级方式的区别和优缺点

升级名称	方式	优点	缺点
滚动升级	节点上只升级 Kubernetes 组件以及网络部分组件，存量节点将全部打上 SchedulingDisabled 标签，仅保证原本运行的应用不受影响，**升级完成后，用户还需手动新建节**	可以基本上保证业务不断。	/

升级名称	方式	优点	缺点
	点，并逐步释放老节点，将应用逐步迁移到新节点上，用户控制升级步骤。		
重置升级	对用户节点使用最新的 node 镜像进行操作系统重置。	时间最短，用户介入操作也较少。	如果节点上有数据或者配置，会丢失，同样会有一段时间业务受损。
平滑升级	采取传统软件升级方式，在用户节点按组件逐个进行升级，您可以通过提交工单申请开启。	用户介入的操作较少，等同于一键式升级。	跨版本由于 OS 会升级，节点会重启，会有一段时间业务受损。

须知

从 1.15 版本开始使用滚动升级达成不断服的升级能力，平滑升级的场景后续将由滚动升级覆盖。

大版本升级说明

表3-11 集群跨大版本升级说明

升级前版本	目标版本	说明
v1.13	v1.15	暂不支持在 Console 中将 v1.13 升级到 v1.15，您可以通过提交工单申请升级。 集群升级到新版本后，不支持回退到老版本。 • 社区 v1.13 与 v1.15 版本之间的 CHANGELOG v1.14 到 v1.15 的变化： https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.15.md v1.13 到 v1.14 的变化： https://github.com/kubernetes/kubernetes/blob/master/CHANGELOG/CHANGELOG-1.14.md • v1.13.11-r1 升级到 v1.15.11-r1 版本的集群后，原 v1.13 的 fuxi Flexvolume（即 storage-driver 插件）容器存储由 v1.15 的 CSI（即 Everest 插件，插件版本 v1.1.6 及以上）接管，原有功能保持不变，但请注意不要新建 fuxi Flexvolume 的存储，否则将导致部分存储功能异常。

3.6.2 升级集群

操作场景

您可以通过云容器引擎管理控制台快速升级到 Kubernetes 最新版本或者 bugfix 版本，以支持新特性的使用。

升级前，请先了解 CCE 各集群版本能够升级到的目标版本，以及升级方式和升级影响，详情请参见 3.6.1 集群版本升级说明。

注意事项

- **集群升级属于不可逆操作，升级后无法回滚到之前版本，请谨慎进行。**
- 请在升级集群前，请查看集群状态是否均为健康状态。若集群不正常，您可以自行修复，若仍有问题请提交工单联系我们协助您进行修复。
- 为了您的数据安全，强烈建议您先备份数据然后再升级，升级过程中不建议对集群进行任何操作。
- 升级前，请在 CCE 控制台中确认您的集群是否可以进行升级操作，**确认方法：**单击“资源管理 > 集群管理”，查看待升级集群右上角是否存在“可升级”提示，若存在则该集群支持升级，若不存在，则该集群不支持升级。更多详细信息请参见 3.6.1 集群版本升级说明。

前提条件

请在集群升级前检查集群和节点的健康状况，确保集群和节点正常可用。

检查方式一：控制台查看

在 CCE 控制台中，单击左侧栏目树的“资源管理”，分别单击“集群管理”和“节点管理”，查看集群和节点的状态是否正常。

检查方式二：Kubectl 命令查看

步骤 1 配置 Kubectl 命令，具体请参见 3.4.2 通过 kubectl 操作 CCE 集群。

步骤 2 执行如下命令，确保集群的所有模块都处于健康状态。

kubectl get cs

命令行终端显示如下信息：

```
NAME                 STATUS    MESSAGE             ERROR
scheduler            Healthy   ok
controller-manager   Healthy   ok
etcd-0               Healthy   {"health": "true"}
etcd-1               Healthy   {"health": "true"}
etcd-2               Healthy   {"health": "true"}
```

步骤 3 执行如下命令，确保所有节点都处于 Ready 状态。

kubectl get nodes

说明

所有节点只能 Ready 状态，不能包含其他状态。

```
NAME                STATUS    ROLES     AGE       VERSION
region01a-xxx   Ready     master    38d       v1.9.7-r1
region01a-xxx   Ready     <none>    38d       v1.9.7-r1
region01a-xxx   Ready     <none>    38d       v1.9.7-r1
region01a-xxx   Ready     <none>    38d       v1.9.7-r1
region01a-xxx   Ready     master    38d       v1.9.7-r1
region01a-xxx   Ready     master    38d       v1.9.7-r1
```

----结束

集群升级前排查项

在集群升级前，请根据如下 Checklist 进行排查，以便提前发现风险和问题。

表3-12 集群升级排查项

类别	检查项
集群	确认当前集群的 Node IP（包括 EIP），是否有作为其他的配置或者白名单等。
	执行升级前预检查功能。
工作负载	记录工作负载的数量、工作负载状态，便于升级后对比。
	针对您使用的数据库（例如云专线 Redis、Mongdb 等），要提前考虑白名单、路由或安全组策略变化等问题。
存储	记录存储状态，保证升级后存储不丢失。
网络	检查使用的负载均衡服务、Ingress，并做好备份。
	针对使用云专线的业务场景，需重点关注业务所在节点或实例 IP 发生变化时，应提前在云专线开通路由。
插件	社区 Kubernetes 1.9 版本升级 1.11 版本时，集群的 kube-dns 会被卸载并替换为 CoreDNS，请备份您配置在 kube-dns 中的 DNS 地址，以便在域名解析异常时重新在 CoreDNS 中进行配置。
运维	私有配置：在升级前的集群中检查是否在节点或容器中放置了数据面密码、证书、环境变量等配置，当容器重启（例如节点异常重新调度 pod），会导致配置丢失，业务异常。
	检查并备份内核参数或者系统配置。

操作步骤

本章节仅为示例。

步骤 1 登录 CCE 控制台，单击左侧导航栏的“资源管理 > 集群管理”，在集群列表页面查看您的集群版本。

步骤 2 单击待升级集群右下角的“更多 > 集群升级”。

说明

若您的集群当前已是最新版本，则“集群升级”按钮为灰色不可用状态。

步骤 3 在弹出的“升级前预检查”对话框中，单击“开始检查”。

步骤 4 升级前预检查开始启动，在此过程中集群状态将显示为“预检查中”，期间该集群不能进行新业务下发，运行中的业务不受影响，该过程将持续 3-5 分钟。

步骤 5 待升级前预检查的状态显示为“已完成”时，单击“去升级”。

步骤 6 进入集群升级页面，参照表 3-13 确认或配置基本信息。

表3-13 基本信息确认与配置

参数	参数说明
集群名称	请确认您要升级的集群名称。
当前版本	请确认待升级集群的版本。
升级后版本	请确认升级后的目标版本。
节点升级策略	**重置升级：**控制节点采用平滑升级方式，而用户节点采用重置安装方式，节点操作系统将会被重装，系统盘和数据盘的数据均会被清空，请谨慎使用。 说明 • 本集群的节点及工作负载生命周期管理功能暂不可用。 • API 访问功能暂不可用。 • 由于升级过程节点进行重置安装，用户已运行的工作负载业务将会中断。 • 用户节点的系统盘和数据盘将会被清空，升级前请事先备份重要数据。 • 用户节点上挂载的非 LVM 管理的数据盘，升级后需要重新挂载，盘中数据不会丢失。 • 云硬盘的配额需大于 0。 • 容器的 IP 地址会发生变化，但是不影响容器间的网络通信。 • 用户节点的自定义标签将不会保留。 • 集群升级时间约为 12 分钟。 **滚动升级：**控制节点采用平滑升级方式，用户节点采用节点池滚动升级，适用于集群下节点均采用节点池创建的场景。 说明 • 本集群的节点及工作负载生命周期管理功能暂不可用。

参数	参数说明
	• API 访问功能暂不可用。 • 用户已运行的工作负载业务不会中断。 • 集群升级时间约为 12 分钟。
登录方式	支持密码和密钥对。 • 选择“密码”：用户名默认为“root”，请输入登录节点的密码，并确认密码。 登录节点时需要使用该密码，请妥善管理密码，系统无法获取您设置的密码内容。 • 选择“密钥对”：选择用于登录本节点的密钥对，支持选择共享密钥。 密钥对用于远程登录节点时的身份认证。若没有密钥对，可单击选项框右侧的“创建密钥对”来新建。
节点升级优先级	可选择优先升级的节点。

步骤 7 完成后单击“下一步”，在弹出的“集群升级”对话框中单击“确定”。

根据您选择的“节点升级策略”，对话框中会有如下两种不同的提示：

- 重置升级：升级后的集群版本将使用更高版本操作系统，升级将会重启节点并升级操作系统版本，升级过程中运行的业务会暂时中断。
- 滚动升级：滚动升级后，需要用户重置节点（同时去除不可调度标签）或新建节点才能完成升级。

步骤 8 进入“升级插件”步骤，如有需要升级的插件会有红色圆点提示，请单击插件卡片左下角的“升级”按钮，完成后单击页面右下角的“升级”。

说明

- 集群将依次升级控制节点，然后并发升级用户节点。用户节点较多时，节点将分批升级。
- 请选择合适的时间段进行升级，以减少升级对业务的影响。
- 单击“升级”会立刻开始执行升级操作，并且无法撤销。升级过程中请勿对节点执行关机、重启等操作。

步骤 9 在弹出的“集群升级”对话框中阅读提示信息，确认后单击“确定”，注意集群升级后不可回退。

步骤 10 在集群列表页面中可以看到集群的状态为“升级中”，升级过程需要一定的时间，请耐心等待升级完成。

- 升级成功后，您可以在集群列表或集群详情页面查看升级后的集群状态和版本。
- 如果升级失败，请参考步骤 2~步骤 9 重新进行升级集群操作，若仍有问题请提交工单联系我们协助您进行修复。

----结束

3.6.3 集群跨版本业务迁移

适用场景

本章介绍在 CCE 中如何将老版本集群的业务迁移到新版本集群。

适用于需要大幅度跨版本集群升级的需求，可以接受新建新版本集群而进行业务迁移的升级方式。

前提条件

表3-14 迁移前 Checklist

类别	描述
集群相关	Nodeip 强相关：确认之前集群的节点 IP（包括 EIP），是否有作为其他的配置或者白名单之类的设置。
工作负载	记录工作负载数目，便于迁移后检查。
存储	1. 确认应用中存储，是否使用公有云，或者自己搭建存储。 2. 自动创建的存储需要在新集群中变成使用已有存储。
网络	1. 注意使用的负载均衡服务，以及 Ingress。 2. 老版本的集群只支持经典型负载均衡服务，迁移到新集群中需要改成共享型负载均衡服务，对应负载均衡服务将会重新建立。
运维	私有配置：确认在之前集群中，是否在节点上配置内核参数或者系统配置。

操作步骤

步骤 1 创建新集群

创建与老版本集群同规格同配置的集群，创建方法请参见 3.3 购买混合集群。

步骤 2 添加节点

添加同规格节点，并且在节点上配置之前的手动配置项，创建方法请参见 4.2 购买节点。

步骤 3 创建存储

在新集群中使用已有存储创建 PVC，PVC 名称不变，方法请参见 9.3.4 使用 kubectl 对接已有云硬盘。

说明

切流方案仅支持 OBS、SFS、共享云硬盘类型的存储。非共享云硬盘存储切流需要将老集群内的工作负载暂停，会导致断服。

步骤 4 创建工作负载

在新集群中创建工作负载，名称和规格参数保持不变，创建方法请参见 6.2 创建无状态负载(Deployment)或 6.3 创建有状态负载(StatefulSet)。在工作负载中挂载存储，方法请参见 9.3.5 使用 kubectl 部署带云硬盘存储卷的工作负载。

步骤 5 创建服务

在新集群中创建 Service，名称和规格参数保持不变，创建方法请参见 8.2.1 集群内访问(ClusterIP)、8.2.2 节点访问(NodePort)、8.2.3 负载均衡(LoadBalancer)。

步骤 6 调测功能

全部创建完成后，请自行调测业务，调测无问题后切换流量。

步骤 7 老集群退订或删除

新集群全部功能 ready，使用解关联功能卸载存储关联卷，方法请参见解关联云硬盘。退订或者删除老集群，删除集群方法请参见 3.7.1 删除集群。

----结束

3.7 管理集群

3.7.1 删除集群

操作场景

本章节介绍如何删除集群。

须知

删除集群会将集群内的节点以及运行的业务都销毁，请谨慎操作。

操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 集群管理”。

步骤 2 单击待删除集群后的“更多 > 删除集群”。

步骤 3 在弹出的“删除集群”窗口中，根据系统提示进行删除操作。

说明

- 删除该集群会将集群内的节点（纳管节点和包周期节点除外）以及运行的工作负载和服务都销毁。
- 删除集群需要花费 1~3 分钟，请耐心等候。
- 请在窗口下方的输入框中输入 **DELETE** 以确认删除此集群。

- 集群不可用时删除集群，存储会残留。
- 集群版本为 v1.13.10 及之前版本时，请勿在 ELB 服务手动修改监听器名称和后端服务器名称，否则删除集群会有资源残留。

步骤 4 单击“是”，开始执行删除集群操作。

----结束

3.7.2 集群休眠与唤醒

操作场景

暂时不需要使用集群时，为节省集群管理费用，建议您将集群设置为休眠状态。

集群休眠后，将无法在此集群上创建和管理工作负载等资源。

休眠中的集群可以快速唤醒，正常使用。

集群休眠操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 集群管理”。

步骤 2 单击待休眠集群后的“更多 > 集群休眠”。

步骤 3 在弹出的集群休眠提示框中，查看风险提示，单击“是”，等待集群完成休眠。

须知

- 按需付费集群休眠后，将暂停收取控制节点资源费用。
- 集群休眠后，集群所属的工作节点（ECS）、绑定的弹性 IP、带宽等资源仍将按各自的计费方式进行收费。如需关机节点，请在集群休眠提示框中勾选“关机集群下所有节点”或参见 4.10 节点关机。

步骤 4 当集群状态由“休眠中”变为“休眠”时，即完成休眠操作。

----结束

集群唤醒操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 集群管理”。

步骤 2 单击待唤醒集群后的“更多 > 集群唤醒”。

步骤 3 在弹出的集群唤醒提示框中，单击“是”，等待集群完成唤醒。

步骤 4 当集群状态由“唤醒中”变为“正常”时，即完成唤醒操作。

说明

集群唤醒后，将继续收取控制节点资源费用。

----结束

3.7.3 配置管理

操作场景

为方便对 CCE 集群中的 kubernetes 配置参数进行管理，开发了配置管理功能，通过该功能您可以对核心组件进行深度配置。

约束与限制

本功能仅支持在 **v1.15 及以上版本**的集群中使用，V1.15 以下版本不显示该功能。

操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 集群管理”。

步骤 2 单击集群后的“更多 > 配置管理”。

步骤 3 在侧边栏滑出的“配置管理”窗口中，根据业务需求修改 Kubernetes 的参数值：

表3-15 配置管理参数

组件	参数	详情	取值
docker	default-ulimit-nofile	限制容器内句柄数	-
kube-apiserver	default-not-ready-toleration-seconds	notReady 容忍时间，NoExecute that is added by default to every pod that does not already have such a toleration.	默认：300
	default-unreachable-toleration-seconds	unreachable 容忍时间，NoExecute that is added by default to every pod that does not already have such a toleration.	默认：300
	max-mutating-requests-inflight	在给定时间内的最大 mutating 请求数。 当服务器超过此值时，它会拒绝请求。 0 表示无限制。	默认：1000
	max-requests-inflight	在给定时间内的最大 non-mutating 请求数。 当服务器超过此值时，它会拒绝请	默认：2000

组件	参数	详情	取值
		求。 0 表示无限制。	
	service-node-port-range	nodeport 端口范围	默认：30000-32767 取值范围： min≥30000 max≤65535
kube-controller-manager	concurrent-deployment-syncs	deployment 的并发处理数	默认：5
	concurrent-endpoint-syncs	endpoint 的并发处理数	默认：5
	concurrent-gc-syncs	garbage collector 的并发数	默认：20
	concurrent-namespace-syncs	namespace 的并发处理数	默认：10
	concurrent-replicaset-syncs	replicaset 的并发处理数	默认：5
	concurrent-resource-quota-syncs	resource quota 的并发处理数	默认：5
	concurrent-service-syncs	service 的并发处理数	默认：10
	concurrent-serviceaccount-token-syncs	serviceaccount-token 的并发处理数	默认：5
	concurrent-ttl-after-finished-syncs	ttl-after-finished 的并发处理数	默认：5
	concurrent_rc_syncs	rc 的并发处理数	默认：5
	kube-api-qps	与 kube-apiserver 通信的 qps	默认：100
	kube-api-burst	与 kube-apiserver 通信的 burst	默认：100
kube-scheduler	kube-api-qps	与 kube-apiserver 通信的 qps	默认：100
	kube-api-burst	与 kube-apiserver 通信的 burst	默认：100

步骤 4 单击“确定”，完成配置操作。

----结束

参考链接

- kube-apiserver
- kube-controller-manager
- kube-scheduler

3.8 获取集群证书

操作场景

通过控制台获取集群证书，使用该证书可以访问 Kubernetes。

操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 集群管理”。

步骤 2 在集群列表页面中，单击对应集群后的“更多 > 证书获取”。

步骤 3 在弹出的“证书获取”窗口中，根据系统提示下载集群 X509 证书。后期访问 Kubernetes 时需要加载此证书。

须知

- 下载的证书包含 client.key、client.crt、ca.crt 三个文件，请妥善保管您的证书，不要泄露。
- 集群中容器之间互访不需要证书。

----结束

3.9 集群监控

操作场景

云容器引擎与监控服务（Cloud Eye）无缝集成，您可以实时查看每个集群控制节点的资源使用情况，了解 CCE 集群控制节点的监控指标，及时收到异常告警并做出反应，保证业务顺畅运行。

操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 集群管理”。

步骤 2 单击待监控的集群名称，进入集群详情。

步骤 3 单击“更多”右侧的图标，可以查看集群资源监控详情。

步骤 4 单击控制节点右侧的“监控”链接，可进入应用运维管理控制台，在“主机监控”中查看对应控制节点的基本监控信息，包括 CPU 使用率、物理内存使用率、磁盘使用率等指标。

----结束

3.10 集群管理权限控制

操作场景

若需要对集群中的资源进行权限控制，（例如 A 用户只能对某个命名空间下的应用有读写权限，B 用户只能对集群下的资源有读权限等），请参照本章节操作。

操作步骤

步骤 1 若需要对集群进行权限控制，请在创建集群时的“认证方式”参数后勾选“认证能力增强”，选择“认证代理”。单击“CA 根证书”后的“上传文件”，上传符合规范且合法的证书。详细请参见表 3-3。

步骤 2 通过 kubectl 创建角色。

下面的例子展示了如何创建“Role”R：允许该角色读取 default 空间下的所有 Pod。参数详细解释请参见 Kubernetes 官方文档。

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: default
  name: pod-reader
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["get", "watch", "list"]
```

步骤 3 通过 kubectl 创建角色绑定。

下面的例子给出 RoleBindings 赋予 default 命名空间下的 **pod-reader** 的角色给用户 **jane**。该策略允许用户 jane 可以读取 default 命名空间下的所有 pods。参数详细解释请参见 Kubernetes 官方文档。

```
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: read-pods
  namespace: default
subjects:
- kind: User
  name: jane   #的用户名
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: pod-reader    #创建的角色名
  apiGroup: rbac.authorization.k8s.io
```

步骤 4 创建角色并与用户绑定成功后，请在接口请求的 headers 中携带用户信息以及集群创建时上传的证书访问 kubernetes 接口。例如调取查询 pod 的接口时，执行命令如下：

curl -k -H "X-Remote-User: *jane*" --cacert /root/tls-ca.crt --key /root/tls.key --cert /root/tls.crt https://*192.168.23.5:5443*/api/v1/namespaces/default/pods

返回 200 表示访问成功，返回 403 表示没有权限访问。

说明

为避免命令执行失败，请提前把证书上传到/root 目录下。

参数解释如下：

- X-Remote-User: jane：请求头固定为 X-Remote-User，jane 为用户名。
- tls-ca.crt ：创建集群时上传的 CA 根证书。
- tls.crt：与集群创建时所上传的 CA 根证书配套的客户端证书。
- tls.key：与集群创建时所上传的 CA 根证书配套的客户端秘钥。
- 192.168.23.5:5443：为连接集群的地址，获取方式如下：

  登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 集群管理”，单击待连接集群的名称，在集群基本信息中获取“内网 apiserver 地址”后的 IP 地址和端口号。

说明

- CCE 支持帐号和 IAM 用户分别下载 config 文件（kubeconfig.json），IAM 用户下载的 config 文件只有 30 天的有效期，而帐号下载的 config 文件会长期有效。该文件用户对接认证用户集群，请用户妥善保存该认证凭据，防止文件泄露后，集群有被攻击的风险。如果不幸泄露，可以通过证书更新的方式，替换认证凭据。
- IAM 用户下载的 config 文件所拥有的 Kubernetes 权限与 CCE 控制台上 IAM 用户所拥有的权限一致。

另外，还支持 **X-Remote-Group** 的头域：用户组名，在做 RoleBinding 时，可以将 Role 与 Group 进行绑定，并在访问集群时携带用户组信息。

----结束

4 节点管理

4.1 节点概述

简介

节点是容器集群组成的基本元素。节点取决于业务，既可以是虚拟机，也可以是物理机。每个节点都包含运行 Pod 所需要的基本组件，包括 Kubelet、Kube-proxy 、Container Runtime 等。

说明

CCE 创建的 Kubernetes 集群包含 master 节点和 node 节点，本章讲述的节点特指 **node 节点**，node 节点是集群的计算节点，即运行容器化应用的节点。

在云容器引擎 CCE 中，主要采用高性能的弹性云主机作为节点来构建高可用的 Kubernetes 集群。

注意事项

- 为了保证节点的稳定性，CCE 集群节点上会根据节点的规格预留一部分资源给 Kubernetes 的相关组件（kubelet、kube-proxy 以及 docker 等）和 Kubernetes 系统资源，使该节点可作为您的集群的一部分。 因此，您的节点资源总量与节点在 Kubernetes 中的可分配资源之间会存在差异。节点的规格越大，在节点上部署的容器可能会越多，所以 Kubernetes 自身需预留更多的资源。
- 节点的网络（如虚机网络、容器网络等）均被 CCE 接管，不允许用户自行添加网卡或改变路由，若自行修改，CCE 不保证服务可用。
- 节点购买后如需对其进行升级或降低规格等操作，请将节点关机后进行变更，也可以重新购买节点后，将老节点删除。

节点生命周期

生命周期是指节点从创建到删除（或释放）历经的各种状态。

表4-1 节点生命周期状态说明

状态	状态属性	说明

状态	状态属性	说明
可用	稳定状态	节点正常运行状态，并连接上集群。 在这个状态的节点可以运行您的业务。
不可用	稳定状态	节点运行异常状态。 在这个状态下的实例，不能对外提供业务，需要4.8 重置节点或联系管理员进行处理。
创建中	中间状态	创建节点实例后，在节点状态进入运行中之前的状态。
安装中	中间状态	节点处于安装 Kubernetes 软件的过程中。
删除中	中间状态	节点处于正在被删除的状态。 如果长时间处于该状态，则说明出现异常，需要联系管理员处理。
关机	稳定状态	节点被正常停止。 在这个状态下的实例，不能对外提供业务，您可以在弹性云主机列表页对其进行开机操作。
错误	稳定状态	节点处于异常状态。 在这个状态下的实例，不能对外提供业务，需要4.8 重置节点或联系管理员进行处理。

相关操作

您可以登录 CCE 控制台并参考以下文档， 进行节点对应的操作：

- 4.2 购买节点
- 4.4 登录节点
- 4.5 节点监控
- 4.6 管理节点标签
- 4.7 同步节点信息
- 4.8 重置节点
- 4.9 删除节点
- 4.10 节点关机

4.2 购买节点

操作场景

节点是指接入到平台的计算资源，包括虚拟机、物理机等。用户需确保所在项目节点资源充足，若节点资源不足，会导致创建工作负载等操作失败。

前提条件

- 已创建至少一个集群，请参见 3.3 购买混合集群。
- 您需要新建一个密钥对，用于远程登录节点时的身份认证。

  若使用密码登录节点，请跳过此操作。

约束与限制

- 仅支持创建 KVM 虚拟化类型的节点，非 KVM 虚拟化类型的节点创建后无法正常使用。
- 集群中的节点一旦购买后不可变更可用区。
- 集群中通过“按需计费”模式购买的节点，在 CCE“节点管理”中进行删除操作后将会直接被删除；通过“包年/包月”模式购买的节点不能直接删除，请通过页面右上角用户名称下的“我的订单”执行资源退订操作。

操作步骤

步骤 1 登录 CCE 控制台，可通过如下两种方式进入“购买节点”页面：

- 在左侧导航栏中选择“资源管理 > 节点管理”，选择节点所在的集群后，在节点列表页面单击上方的“购买节点”。
- 在左侧导航栏中选择“资源管理 > 集群管理”，在集群卡片上，单击“购买节点”。

步骤 2 计费模式：支持“按需计费”和“包年/包月”类型。本章以“按需计费”类型为例进行讲解。

步骤 3 选择区域和可用区。

- **当前区域**：节点实例所在的物理位置。
- **可用区**：请根据业务需要进行选择。可用区是在同一区域下，电力、网络隔离的物理区域，可用区之间内网互通，不同可用区之间物理隔离。

  如果您需要提高工作负载的高可靠性，建议您在创建集群后将云服务器部署在不同的可用区，购买集群时节点只能部署在一个可用区。

步骤 4 **节点类型**：选择“虚拟机节点”。配置以下参数。

- **节点名称**：自定义节点名称。长度范围为 1-56 个字符，以小写字母开头，支持小写字母、数字、中划线(-)，不能以中划线(-)结尾。
- **节点规格**：请根据业务需求选择相应的节点规格。

- 通用型：该类型实例提供均衡的计算、存储以及网络配置，适用于大多数的使用场景。通用型实例可用于 Web 服务器、开发测试环境以及小型数据库工作负载等场景。
- GPU 加速型：提供优秀的浮点计算能力，从容应对高实时、高并发的海量计算场景。P 系列适合于深度学习，科学计算，CAE 等；G 系列适合于 3D 动画渲染，CAD 等。
- 高性能计算型：实例提供具有更稳定、超高性能计算性能的实例，可以用于超高性能计算能力、高吞吐量的工作负载场景，例如科学计算。
- 通用计算增强型：该类型实例具有性能稳定且资源独享的特点，满足计算性能高且稳定的企业级工作负载诉求。

为确保节点稳定性，系统会自动预留部分资源，用于运行必须的系统组件。详细请参见 4.12 节点预留资源计算公式。

- **操作系统：**请直接选择节点对应的操作系统。
- **系统盘：**设置工作节点的系统盘空间。您可以设置系统盘的规格为 40GB-1024GB 之间的数值，缺省值为 40GB。

  在默认情况下，系统盘可提供高 IO、超高 IO 两种基本的云硬盘类型。
- **数据盘：**设置工作节点的数据盘空间。您可以设置数据盘的规格为 100GB-32678GB 之间的数值，缺省值为 100GB。数据盘可提供的云硬盘类型与上方系统盘一致。

⚠ 注意

若数据盘卸载或损坏，会导致 docker 服务异常，最终导致节点不可用。建议不要删除该数据盘。

- 数据盘空间分配：单击后方的 更改配置，可以对数据盘中的“k8s 空间”和“用户空间”占比进行自定义设置，开启 LVM 管理的数据盘将按照设置的比例进行统一分配。部分集群版本不支持此功能，具体以界面为准。
  - k8s 空间：您可以自定义数据盘中 Docker 和 Kubelet 的资源占比。Docker 资源包含 Docker 工作目录、Docker 镜像数据以及镜像元数据；Kubelet 资源包含 Pod 配置文件、密钥以及临时存储 EmptyDir 等挂载数据。
  - 用户空间：定义本地盘中不分配给 kubernetes 使用的空间大小和用户空间挂载路径。

说明

请注意“挂载路径”不能设置为根目录“/”，否则将导致挂载失败。挂载路径一般设置为：

- /opt/xxxx（但不能为/opt/cloud）
- /mnt/xxxx（但不能为/mnt/paas）
- /tmp/xxx
- /var/xxx （但不能为/var/lib、/var/script、/var/paas 等关键目录）
- /xxxx（但不能和系统目录冲突，例如 bin、lib、home、root、boot、dev、etc、lost+found、mnt、proc、sbin、srv、tmp、var、media、opt、selinux、sys、usr 等）

注意不能设置为/home/paas、/var/paas、/var/lib、/var/script、/mnt/paas、/opt/cloud，否则会导致系统或节点安装失败。

- **虚拟私有云：**不可修改，仅用于展示当前集群所在的虚拟私有云。
- **所在子网：**通过子网提供与其他网络隔离的、可以独享的网络资源，以提高网络安全。可选择该集群虚拟私有云下的任意子网，集群节点支持跨子网。

  该参数仅在 v1.13.10-r0 及以上版本的集群中显示，请务必**确保子网下的 DNS 服务器可以解析 OBS 服务域名**，否则无法创建节点。

  已有集群添加节点时，如果子网对应的 VPC 新增了扩展网段且子网是扩展网段，要在控制节点安全组（即集群名称-cce-control-随机数）中添加如下三条安全组规则，以保证集群添加的节点功能可用（新建集群时如果 VPC 已经新增了扩展网段则不涉及此场景）：

- **弹性 IP：**独立申请的公网 IP 地址，若节点有互联网访问的需求，请选择“暂不使用”或“使用已有”。**集群开启 IPv6 时，不显示该参数。**

  弹性公网 IP 提供外网访问能力，可以灵活绑定及解绑，随时修改带宽。未绑定弹性公网 IP 的云服务器无法直接访问外网，无法直接对外进行互相通信。

  - 暂不使用：若新增节点未绑定弹性 IP，则在该节点上运行的工作负载将不能被外网访问，仅可作为私有网络中部署业务或者集群所需云服务器进行使用。
  - 使用已有：请选择已有的弹性 IP，将为当前节点分配已有弹性 IP。

说明

CCE 默认不启用 VPC 的 SNAT。若 VPC 启用了 SNAT，可以不使用 EIP 去访问外网。SNAT 具体请参见自定义策略。

- **共享带宽：**请选择“暂不使用”或“使用已有”。**仅在集群开启 IPv6 时，显示该参数。**

弹性公网 IP 提供外网访问能力，可以灵活绑定及解绑，随时修改带宽。未绑定弹性公网 IP 的云服务器无法直接访问外网，无法直接对外进行互相通信。

- **登录方式：**支持密码和密钥对。
  - 选择“密码”：用户名默认为“root”，请输入登录节点的密码，并确认密码。

    登录节点时需要使用该密码，请妥善管理密码，系统无法获取您设置的密码内容。
  - 选择“密钥对”：选择用于登录本节点的密钥对，支持选择共享密钥。

    密钥对用于远程登录节点时的身份认证。若没有密钥对，可单击选项框右侧的“创建密钥对”来新建。

须知

如果子用户创建节点选择密钥对创建，这个密钥只对创建这个密钥的子用户有效，即使其他子用户在同一个组也无法选择，也无法使用。例如：A 用户创建的密钥，B 用户无法使用这个密钥对创建节点，并且 Console 也选不到。

- **云服务器高级设置：**（可选），单击 展开后可对节点进行如下高级功能配置：
  - **安装前执行脚本：**请输入脚本命令，大小限制为 0~1000 字符。

    脚本将在 Kubernetes 软件安装前执行，可能导致 Kubernetes 软件无法正常安装，需谨慎使用。常用于格式化数据盘等场景。
  - **安装后执行脚本：**请输入脚本命令，大小限制为 0~1000 字符。

    脚本将在 Kubernetes 软件安装后执行，不影响 Kubernetes 软件安装。常用于修改 Docker 配置参数等场景。
  - **新增数据盘：**单击“新增数据盘”，选择云硬盘类型并输入数据盘规格。
  - **子网 IP：**可选择“自动分配 IP 地址”和“手动分配 IP 地址”，推荐使用“自动分配 IP 地址”。
- **Kubernetes 高级设置：**（可选），单击 展开后可对集群进行如下高级功能配置：
  - **最大实例数：**节点最大允许创建的实例数(Pod)，该数量包含系统默认实例，取值范围为 16~128。

    该设置的目的为防止节点因管理过多实例而负载过重，请根据您的业务需要进行设置。
  - **自定义镜像仓库：**单击“新增自定义镜像仓库地址”输入镜像仓库地址。

    添加自定义镜像仓库地址（非 SSL 镜像源地址）到 docker 启动参数中，避免拉取个人镜像仓库的镜像失败，格式可为“IP 地址:端口或者域名”。安装后执行脚本与自定义镜像仓库不能同时使用。
  - **单容器可用数据空间：**该参数用于设置一个容器可用的数据空间大小，设置范围为 10G 到 80G。如果设置的参数超过数据盘中 Docker 可占用的实际数据空间（由数据盘设置项中的资源分配自定义参数指定，默认为数据盘大小的 90%），将以 Docker 的实际空间大小为主。该参数仅在 v1.13.10-r0 及以上版本的集群中显示。

- **节点购买数量：**此处设置的节点数不能超过集群管理的最大节点规模，请根据业务需求和界面提示进行选择，单击后方的可查看影响能添加节点数的因素（取决于最小值）。如需申请更多配额，请单击。

步骤 5 **购买时长：**若选择“包年包月”的计费模式购买节点时，请设置购买时长。

步骤 6 单击“下一步：配置确认”，确认订单无误后，单击“提交”。

若计费模式为“包年包月”，在确认订单无误后，请单击“去支付”，请根据界面提示进行付款。

系统将自动跳转到节点列表页面，待节点状态为“可用”，表示节点添加成功。添加节点预计需要 6-10 分钟左右，请耐心等待。

说明

集群创建时自动创建的安全组以及安全组规则禁止删除，否则会导致集群异常。

步骤 7 单击“返回节点列表”，待状态为可用，表示节点创建成功。

说明

可分配资源：可分配量按照实例请求值(request)计算，表示实例在该节点上可请求的资源上限，不代表节点实际可用资源。

计算公式为：

- 可分配 CPU = CPU 总量 - 所有实例的 CPU 请求值 - 其他资源 CPU 预留值
- 可分配内存 = 内存总量 - 所有实例的内存请求值 - 其他资源内存预留值

----结束

4.3 纳管已有节点到集群

操作场景

CCE 集群支持两种添加节点的方式：4.2 购买节点和纳管节点，纳管节点是指将“已购买的弹性云主机（ECS）加入到 CCE 集群中”，所纳管节点的计费模式支持“按需计费”和“包年/包月”两种类型。

本节将指导您通过 CCE 控制台纳管已有虚拟机节点。

须知

- 纳管时，会将所选弹性云主机的操作系统重置为 CCE 提供的标准镜像，以确保节点的稳定性，请选择操作系统及重置后的登录方式。
- 所选弹性云主机挂载的系统盘、数据盘都会在纳管时被格式化，请确保信息已备份。
- 纳管过程中，请勿在弹性云主机控制台对所选虚拟机做任何操作。

约束与限制

- 目前仅支持在 **CCE 集群**中纳管同一 Region 下的虚拟机节点（含 GPU 加速型）。
- 集群版本需 V1.13 及以上。
- 集群开启 IPv6 后，只支持纳管所在的子网开启了 IPv6 功能的节点；集群未开启 IPv6，只支持纳管所在的子网未开启 IPv6 功能的节点。
- 原虚拟机节点创建时若已设置密码或密钥，需等待虚拟机节点可用 10 分钟后方可纳管。

前提条件

支持纳管符合如下条件的弹性云主机：

- 待纳管节点已购买成功，且必须状态为“运行中“，且未被其他集群所使用。
- 纳管节点需与 CCE 集群在同一虚拟私有云内（若集群版本低于 1.13.10，纳管节点还需要与 CCE 集群在同一子网内）。
- 需挂载数据盘，数据盘需满足有且仅有 1 块，且容量不少于 100GB，多余的数据盘请先卸载。
- 节点规格要求：CPU 必须 2 核及以上，内存必须 4GB 及以上，网卡有且仅能有一个。

操作步骤

步骤 1 登录 CCE 控制台，单击左侧导航栏的“资源管理 > 集群管理”，单击待纳管集群下的“更多 > 纳管节点”。

步骤 2 （可选）登录 CCE 控制台，单击左侧导航栏的“资源管理 > 节点管理”，选择节点所在的集群后，单击页面右上角的“纳管节点”。

步骤 3 在打开的界面中展示了当前符合要求的弹性云主机。如果不符合纳管要求时会显示相应的红色提示图标，列表下方会有被过滤的弹性云主机数量，鼠标移动到后方的可以查看具体原因。

步骤 4 勾选需要纳管的弹性云主机，单击“下一步”。

步骤 5 在“待纳管节点”页面中，按照界面提示选择“登录方式”，单击“高级配置”后的按钮，可以设置“最大实例数”、“自定义镜像仓库”和“资源分配”，然后单击“提交”。

步骤 6 在弹出的“信息确认”页面中确认信息无误后，单击“确定”。

系统将提示节点纳管请求提交成功。

----结束

移除节点

移除节点指从集群中移除“弹性云主机（简称 ECS）”，并不会删除 ECS 和卸载已安装的 CCE 相关组件。

如需清理 CCE 相关组件，请前往 ECS 控制台将机器关机后，点击“更多 > 镜像/磁盘 > 重装系统”操作。

仅纳管的节点支持移除操作。

说明

仅纳管的节点支持“移除”操作，集群中新增的节点如需删除请选择“删除”操作。

步骤 1 在 CCE 控制台中，选择导航栏的“资源管理 > 节点管理”，在节点列表中，单击待移除节点后方“操作”栏中的“更多 > 移除”。

步骤 2 在弹出的“移除纳管节点”弹框中，输入 **REMOVE**，确认移除该节点，单击“确定”，即可移除纳管的节点。

说明

以上步骤仅从集群中移除了节点（即弹性云主机 ECS），并不会删除节点和已安装的 CCE 相关组件。请根据界面中展示的步骤清理 CCE 相关资源。

----结束

4.4 登录节点

约束与限制

- SSH 方式登录时要求该节点（弹性云主机 ECS）已绑定弹性公网 IP。
- 只有运行中的弹性云主机才允许用户登录。
- Linux 操作系统用户名为 root。

登录方式概述

登录节点（弹性云云主机 ECS）的方式有如下两种：

- **管理控制台远程登录（VNC 方式）**

  未绑定弹性公网 IP 的弹性云主机可通过管理控制台提供的远程登录方式直接登录。

- **SSH 方式登录**

仅适用于 Linux 弹性云主机。您可以使用远程登录工具（例如 PuTTY、Xshell、SecureCRT 等）登录弹性云主机。如果普通远程连接软件无法使用，您可以使用云服务器 ECS 管理控制台的管理终端连接实例，查看云服务器操作界面当时的状态。

说明

- SSH 方式登录包括 **SSH 密钥**和 **SSH 密码**两种方式。
- 本地使用 Windows 操作系统登录 Linux 节点时，输入的镜像用户名（Auto-login username）为：root。

  CCE 控制台不提供针对节点的操作系统升级，也不建议您通过 yum 方式进行升级，如果您在节点上通过 yum update 升级了操作系统，会导致容器网络的组件不可用。

表4-2 Linux 云服务器登录方式一览

是否绑定 EIP	本地设备操作系统	连接方法
是	Windows	使用 PuTTY、Xshell 等远程登录工具： • 密码方式鉴权：SSH 密码方式登录的“客户端使用 Windows 系统”。 • 密钥方式鉴权：SSH 密钥方式登录的本地使用 Windows 操作系统。
是	Linux	使用命令连接： • 密码方式鉴权：SSH 密码方式登录的“客户端使用 Linux 系统”。 • 密钥方式鉴权：SSH 密钥方式登录的本地使用 Linux 操作系统。
是/否	Windows 或者 Linux	使用管理控制台远程登录方式：Linux 云服务器远程登录（VNC 方式）。

4.5 节点监控

操作场景

云容器引擎（CCE）与云监控服务（Cloud Eye）无缝集成，可以实时查看每个集群工作节点的资源使用情况，您可以使用该服务监控您的节点，执行自动实时监控、告警和通知操作，帮助您更好地了解节点的各项性能指标。

云监控服务（Cloud Eye）为您提供一个针对弹性云主机、带宽等资源的立体化监控平台。让您全面了解云上的资源使用情况、业务的运行状况，并及时收到异常告警做出反应，保证业务顺畅运行。具体请参见。

提供的云监控服务，可以对节点运行状态进行日常监控。您可以通过 CCE 控制台或云监控服务控制台，直观地查看节点的各项监控指标。

由于监控数据的获取与传输会花费一定时间，因此，云监控显示的是当前时间 5～10 分钟前的节点状态。如果您的节点刚刚创建完成，请等待 5～10 分钟后查看监控数据。

操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 节点管理”。

步骤 2 在列表右上角选择所需的集群，在该集群下选择所需的节点。

步骤 3 单击节点名称后的“监控”，在云监控服务中可查看监控信息。

步骤 4 在弹出节点监控窗口中，可查看该节点的基本监控信息，包括 CPU 使用率、内存使用率、磁盘使用率、CPU/内存/磁盘使用记录、上行速率、下行速率、网络监控等指标。

表4-3 支持的主要监控指标

监控指标	指标含义
CPU 使用率	该指标为从物理机层面采集的 CPU 使用率，数据准确性低于从弹性云主机内部采集的数据。
内存使用率	该指标用于统计节点的内存使用率。
磁盘使用率	节点的磁盘使用率，统计对象是使用率最大的磁盘，而非所有磁盘的平均使用率； 节点的磁盘剩余容量，统计对象是剩余容量最大的磁盘，而非所有磁盘的剩余容量之和。
CPU/内存/磁盘使用记录	可查看一小时内的 CPU 使用率、内存使用率和磁盘使用率的曲线。
下行速率	一般指从网络下载数据到节点的速度，单位 KB/S。
上行速率	一般指从节点上传网络的速度，单位 KB/S。
网络监控	可查看一小时内的下行速率、上行速率的曲线。

----结束

相关操作

- 16.1 支持的监控指标
- 16.2 设置告警规则

4.6 管理节点标签

节点标签可以给节点打上不同的标签，给节点定义不同的属性，通过这些标签可以快速的了解各个节点的特点。

节点标签使用场景

节点标签的主要使用场景有两类。

- 节点管理：通过节点标签管理节点，给节点分类。
- 工作负载与节点的亲和与反亲和：
  - 有的工作负载需要的 CPU 大，有的工作负载需要的内存大，有的工作负载需要 IO 大、可能会影响其他工作负载正常工作等等，此时建议给节点添加不同标签。在部署工作负载的时候，就可以选择相应标签的节点亲和部署，保证系统正常工作；反之，可以使用节点的反亲和部署。
  - 一个系统可以分为多个模块，每个模块由多个微服务组成，为保证后期运维的高效，可以将节点打上对应模块的标签，让各模块部署到各自的节点模块上，互不干扰、方便开发到各自节点上去维护。

节点固有标签

节点创建出来会存在一些固有的标签，并且是无法删除的，这些标签的含义请参见表4-4。

表4-4 节点固有标签

键	值
failure-domain.beta.kubernetes.io/is-baremetal	表示是否为裸金属节点。 例如：false，表示非裸金属节点
failure-domain.beta.kubernetes.io/region	表示节点当前所在区域。
failure-domain.beta.kubernetes.io/zone	表示节点所在区域的可用区。
node.kubernetes.io/subnetid	表示子网的 ID。
os.architecture	表示节点处理器架构。 例如：amd64，表示 AMD64 位的处理器
os.name	表示节点的操作系统名称。
os.version	表示节点内核版本。

添加节点标签

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 节点管理”。

步骤 2 在节点列表中，单击“操作”栏的“标签管理”。

步骤 3 在弹出的“标签管理”窗口中，单击标签列表下方的“添加标签”，填写需要增加标签的“键”和“值”，单击“确定”。

例如，填写的键为“deploy_qa”，值为“true”，就可以从逻辑概念表示该节点是用来部署 QA（测试）环境使用。

步骤 4 标签添加成功后，再次单击“标签管理”，可查看到已经添加的标签。

----结束

删除节点标签

删除标签只能够删除用户添加的标签，节点固有的标签不能删除。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 节点管理”。

步骤 2 在节点列表中，单击“操作”栏的“标签管理”。

步骤 3 单击“删除”，单击“确定”，删除标签。

可查看到“标签变更成功”。

----结束

4.7 同步节点信息

操作场景

集群中的每一个节点对应一台弹性云主机或物理机，集群节点创建成功后，您仍可以根据需求，修改云服务器的名称或变更规格。

CCE 节点的部分信息是独立于弹性云主机 ECS 维护的，当您在 ECS 控制台中修改云服务的名称、弹性公网 IP，以及变更计费方式或变更规格后，需要通过“同步节点信息”功能将信息同步到 CCE 控制台相应节点中，同步后信息将保持一致。

ECS 常见信息修改如下：

- 修改节点名称。
- 计费方式变更。
- 当您购买的节点规格无法满足业务需要时，变更规格，升级 vCPU、内存。

操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 节点管理”。

步骤 2 单击节点后的“更多 > 同步节点信息”。

说明

您也可以单击节点名称进入节点详情页，单击右上角“同步节点信息”。

同步完成后，页面右上角将会提示“同步成功”。

----结束

4.8 重置节点

操作场景

在 CCE 集群中重置节点会将该节点以及节点内运行的业务都销毁，重置前请确认您的正常业务运行不受影响，请谨慎操作。

注意事项

- 重置节点功能不会重置控制节点，仅能对工作节点进行重置操作，如果重置后节点仍然不可用，请删除该节点重新购买。
- **重置节点将对节点操作系统进行重置安装，节点上已运行的工作负载业务将会中断，请在业务低峰期操作。**
- **节点重置后操作系统版本不变，但工作节点的系统盘和 docker 数据盘将会被清空，重置前请事先备份重要数据。**
- **用户节点如果有自行挂载了数据盘，重置完后会清除挂载信息，请事先备份重要数据，重置完成后请重新执行挂载行为，数据不会丢失。**
- 节点上的工作负载实例的 IP 会发生变化，但是不影响容器网络通信。
- 云硬盘必须有剩余配额。
- 操作过程中，后台会把当前节点设置为不可调度状态。

操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 节点管理”，单击节点后的“更多 > 重置节点”。

步骤 2 在弹出的“重置节点”窗口中输入 **RESET**，同时重新配置密码或密钥登录方式。

步骤 3 单击“是”，等待完成节点重置。

重置节点后，原有节点上的工作负载实例会自动迁移至其他可用节点。

----结束

4.9 删除节点

操作场景

在 CCE 集群中删除节点会将该节点以及节点内运行的业务都销毁，删除前请确认您的正常业务运行不受影响，请谨慎操作。

约束与限制

- 删除 CCE 集群时，ECS 节点也将一起删除，暂不支持删除 CCE 集群而保留 ECS 节点的需求。
- 对于包周期（包年/包月）预付费的 ECS 节点不能直接删除，请通过页面右上角用户名下的“我的订单”，执行资源退订操作。

注意事项

- 删除节点会涉及 Pod 迁移，可能会影响业务，请在业务低峰期操作。
- 操作过程中可能存在非预期风险，请提前做好相关的数据备份。
- 操作过程中，后台会把当前节点设置为不可调度状态。
- 删除节点仅能移除工作节点，不会移除控制节点。

操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 节点管理”，单击节点后的“更多 > 删除”。

步骤 2 在弹出的“删除节点”窗口中输入 **DELETE** ，单击“是”，等待完成节点删除。

说明

- 删除节点后，原有节点上的工作负载实例会自动迁移至其他可用节点。
- 节点上绑定的磁盘和 EIP 如果属于重要资源请先解绑，否则会被级联删除。
- 释放 ECS 实例仅释放“按需计费”的 ECS 实例。

----结束

4.10 节点关机

操作场景

集群中的节点关机后，该节点以及节点内的业务将停止运行，节点关机前，请先确认您的正常业务运行将不受影响，请谨慎操作。节点关机后不再收费。

约束与限制

- 节点关机会涉及 Pod 迁移，可能会影响业务，请在业务低峰期操作。
- 操作过程中可能存在非预期风险，请提前做好相关的数据备份。

- 操作过程中，后台会把当前节点设置为不可调度状态。
- 节点关机仅能对工作节点关机，不会对控制节点关机。

操作方法

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 节点管理”。

步骤 2 在节点管理列表中，单击待关机节点的名称。

步骤 3 在打开的节点管理详情页中，单击节点名称。

步骤 4 页面跳转至弹性云主机详情页中，单击右上角的“关机”，在弹出的关机窗口中单击“是”，即可完成关机操作。

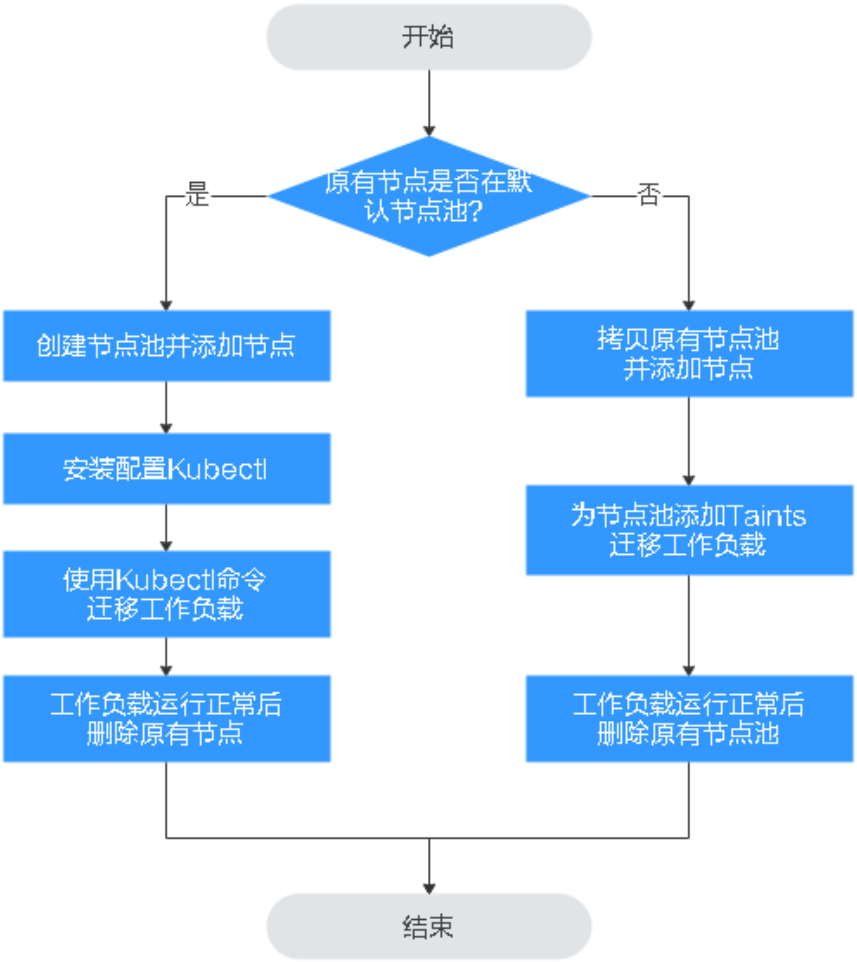
----结束

4.11 节点滚动升级

操作场景

用户在不删除现有节点的情况下，将工作负载迁移到新的节点上。迁移流程如图 4-1 所示。

图4-1 节点迁移流程



约束与限制

- 现有节点和工作负载待迁移的节点必须在同一集群。
- 当前仅支持在 Kubernetes v1.13.10 及以后集群版本执行此操作。
- 默认节点池 DefaultPool 不支持修改配置。

原有节点在默认节点池

步骤 1 创建新的节点。

1. 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 节点池管理”。
2. 在集群选择框中，选择现有节点所属的集群。
3. 单击“创建节点池”，设置节点池如下参数，其他参数根据需要进行修改，参数说明请参见 5.2 创建节点池。
   a. 节点池名称：新建节点池的名称，例如 nodepool-demo。
   b. 节点购买数量：本例添加一个节点。
   c. 节点规格：请根据业务需求选择相应的节点规格。

d. 操作系统：选择节点对应的操作系统。

e. 登录方式：支持密码和密钥对。

- 选择“密码”：用户名默认为“root”，请输入登录节点的密码，并确认密码。

  登录节点时需要使用该密码，请妥善管理密码，系统无法获取您设置的密码内容。

- 选择“密钥对”：在选项框中选择用于登录本节点的密钥对，并单击勾选确认信息。

  密钥对用于远程登录节点时的身份认证。若没有密钥对，可单击选项框右侧的“创建密钥对”来新建。

4. 单击“下一步：配置确认”。确认节点池配置后，单击“提交”。

   单击“返回节点池管理”返回节点池列表。在节点列表中可查看到新建节点池已创建，且状态为“正常”。

步骤 2 单击节点池名称，在节点列表中可查看到新建节点的 IP 地址。

步骤 3 安装配置 kubectl。

1. 在左侧导航栏中选择“资源管理 > 集群管理”，单击现有节点所在集群下的“命令行工具 > kubectl”。
2. 在集群详情页中的“kubectl”页签下，请参照界面中的提示信息完成集群连接。

步骤 4 迁移工作负载。

1. 给需要迁移工作负载的节点打上 Taint（污点）。

   kubectl taint node *[node]* **key=value:***[effect]*

   其中，*[node]*为待迁移工作负载所在节点的 IP；*[effect]*取值为 NoSchedule、PreferNoSchedule 或 NoExecute，此处必须设置为 NoSchedule。

   - NoSchedule：一定不能被调度。
   - PreferNoSchedule：尽量不要调度。
   - NoExecute：不仅不会调度，还会驱逐 Node 上已有的 Pod。

   说明

   若需要重新设置污点时，可执行 **kubectl taint node** *[node]* **key:[***effect]*-命令去除污点。

2. 安全驱逐节点上的工作负载。

   kubectl drain *[node]*

   其中，*[node]*为待转移工作负载所在节点的 IP。

3. 在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”。在工作负载列表中，待迁移工作负载的状态由“运行中”变为“未就绪”。工作负载状态再次变为“运行中”，表示迁移成功。

   说明

   迁移工作负载时，若工作负载配置了节点亲和性，则工作负载会一直提示“未就绪”等异常情况。请单击工作负载名称进入到负载详情页，在选择“调度策略”页签，删除原节点的亲和性配置，并单击“简易调度策略”配置新的节点亲和性和反亲和性策略，详情请参见 7.3 简易调度策略。

工作负载迁移成功后，在工作负载详情页的“实例列表”页签，可查看到工作负载状已迁移到步骤 1 中所创建的节点上。

步骤 5 删除原有节点。

工作负载迁移成功且运行正常后，即可在“资源管理 > 节点管理”中删除原有节点。

----结束

原有节点不在默认节点池

步骤 1 拷贝节点池并添加节点。

1. 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 节点池管理”。
2. 在集群选择框中，选择现有节点所属的集群。

   在节点池列表中，查找到原有节点所在的节点池。
3. 单击该节点池名称后的“更多 > 拷贝”，在“创建节点池”页面下设置如下参数，其他参数根据需要进行修改，参数说明请参见 5.2 创建节点池。
   - 节点池名称：新建节点池的名称，例如 nodepool-demo。
   - 节点购买数量：本例添加一个节点。
   - 节点规格：请根据业务需求选择相应的节点规格。
   - 操作系统：选择节点对应的操作系统。
   - 登录方式：支持密码和密钥对。
     - 选择“密码”：用户名默认为“root”，请输入登录节点的密码，并确认密码。

       登录节点时需要使用该密码，请妥善管理密码，系统无法获取您设置的密码内容。
     - 选择“密钥对”：在选项框中选择用于登录本节点的密钥对，并单击勾选确认信息。

       密钥对用于远程登录节点时的身份认证。若没有密钥对，可单击选项框右侧的“创建密钥对”来新建。
4. 单击“下一步：配置确认”。确认节点池配置后，单击“提交”。

   单击“返回节点池管理”返回节点池列表。在节点列表中可查看到新建节点池已创建，且状态为“正常”。

步骤 2 单击节点池名称，在节点列表中可查看到新建节点的 IP 地址。

步骤 3 迁移工作负载。

1. 单击节点池 nodepool-demo 后的“编辑”配置 Taints 参数。
2. 单击“Add Taint”，输入 Key 和 Value 值，Effect 选项有 NoSchedule、PreferNoSchedule 或 NoExecute，此处必须选择“NoExecute”。
   - NoSchedule：一定不能被调度。
   - PreferNoSchedule：尽量不要调度。
   - NoExecute：不仅不会调度，还会驱逐 Node 上已有的 Pod。

说明

若需要重新设置污点时，单击“删除”去除污点。

3. 单击“保存”。
4. 在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”。在工作负载列表中，待迁移工作负载的状态由“运行中”变为“未就绪”。工作负载状态再次变为“运行中”，表示迁移成功。

说明

迁移工作负载时，若工作负载配置了节点亲和性，则工作负载会一直提示“未就绪”等异常情况。请单击工作负载名称进入到负载详情页，在选择“调度策略”页签，删除原节点的亲和性配置，并单击“简易调度策略”配置新的节点亲和性和反亲和性策略，详情请参见 7.3 简易调度策略。

工作负载迁移成功后，在工作负载详情页的“Pods”页签，可查看到工作负载状已迁移到步骤 1 中所创建的节点上。

步骤 4 删除原有节点。

工作负载迁移成功且运行正常后，即可在“资源管理 > 节点池管理”中删除原有节点。

----结束

4.12 节点预留资源计算公式

节点的部分资源需要运行一些必要的 Kubernetes 系统组件和 Kubernetes 系统资源，使该节点可作为您的集群的一部分。 因此，您的节点资源总量与节点在 Kubernetes 中的可分配资源之间会存在差异。节点的规格越大，在节点上部署的容器可能会越多，所以 kubernetes 自身需预留更多的资源。

为了保证节点的稳定性，CCE 集群节点上会根据节点的规格预留一部分资源给 Kubernetes 的相关组件（kubelet，kube-proxy 以及 docker 等）。

CCE 对用户节点可分配的资源计算法则如下：

Allocatable = Capacity - Reserved - Eviction Threshold

即，**节点上可配置值=总量-预留值-驱逐阈值**。

CCE 对节点内存的预留规则

mem_reserved_total = reserved_value_for_system + reserved_value_for_pods

- **reserved_value_for_system：**
  - total_mem <= 8GB，reserved_value_for_system = 0MB
  - 8GB < total_mem <= 16GB, reserved_value_for_system = ((total_mem – 8GB)*1024*10%)MB
  - 16GB < total_mem <= 128GB, reserved_value_for_system = (8GB*1024*10% + (total_mem – 16GB)*1024*6%)MB

  - total_mem > 128GB, reserved_value_for_system = (8GB*1024*10% + 112GB*1024*6% + (total_mem – 128GB)*1024*2%)MB

- **reserved_value_for_pods：**

  total_mem <= 2GB：

  reserved_value_for_pods = total_mem *25%

  total_mem > 2GB：

  - 0 < pods <= 16, reserved_value_for_pods = 700 MB
  - 16 < pods <= 32, reserved_value_for_pods = (700 + (pods - 16)*18.75)MB
  - 32 < pods <= 64, reserved_value_for_pods = (1024 + (pods - 32)*6.25)MB
  - 64 < pods <= 128, reserved_value_for_pods = (1230 + (pods - 64)*7.80)MB
  - pods > 128, reserved_value_for_pods = (1740 + (pods - 128)*11.20)MB

其中：

- “total_mem”为内存总量。
- “pods”为节点的最大实例数。
- “mem_reserved_total”为总预留值。
- “reserved_value_for_system”为系统组件预留值。
- “reserved_value_for_pods”为 Kubelet 管理 Pod 所需预留值。

须知

对于小规格节点，需根据实际使用情况调整节点的最大实例数。或者在 CCE 控制台创建节点时，需考虑根据节点规格自适应调整节点的最大实例数参数。

CCE 对节点 CPU 的预留规则

- total_cpu <= 1core，reserved_value= total_cpu *6%
- 1core < total_cpu <= 2core, reserved_value= 1core*6% + (total_cpu – 1core)*1 %
- 2core < total_cpu <= 4core, reserved_value= 1core*6% + 1core*1% + (total_cpu – 2core)*0.5%
- total_cpu > 4core, reserved_value= 1core*6% + 1core*1% + 2core*0.5% + (total_cpu – 4core)*0.25%

其中：

- “total_cpu”为 CPU 总量。
- “reserved_value”为预留值。
- CCE 对节点预留了额外的 100Mi 给 kubelet 驱逐所用。

4.13 为 docker 创建 Linux LVM 型磁盘分区

操作场景

检查是否存在“可用裸盘”和“Linux LVM 型磁盘分区”，并提供创建 Linux LVM 型磁盘分区的方法。

前提条件

为了使系统更加稳定，请给 docker 单独挂载数据盘，使用 direct-lvm 模式。

操作步骤

步骤 1 检查当前节点是否存在可用的裸盘。

1. 以 root 用户登录目标节点。
2. 查看存在的裸盘设备。

   lsblk -l | grep disk

   若回显如下，表示节点存在名为 xvda 和 xvdb 的裸盘。

   ```
   xvda  202:0    0   40G  0 disk
   xvdb  202:16   0  100G  0 disk
   ```

3. 检查裸盘是否已经被使用。

   lsblk /dev/<*devicename*>

   devicename 为裸盘名称，例如上个步骤的 xvda 和 xvdb。

   分别执行 **lsblk /dev/xvda** 和 **lsblk /dev/xvdb**，回显如下，表示 xvda 已被分区并使用，而裸盘 xvdb 可用。若没有可用裸盘，请为该节点绑定云硬盘，建议不小于 80G。

   ```
   NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
   xvda    202:0    0   40G  0 disk
   ├─xvda1 202:1    0  100M  0 part /boot
   └─xvda2 202:2    0 39.9G  0 part /
   NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
   xvdb 202:16   0  100G  0 disk
   ```

步骤 2 检查是否存在可用区，当前仅支持 Linux LVM 型分区。

1. 以 root 用户登录目标节点。
2. 查看当前系统类型为 Linux LVM 的分区。

   sfdisk -l 2>>/dev/null| grep "Linux LVM"

   回显如下，表示当前系统有“/dev/nvme0n1p1”和“/dev/nvme0n1p2”两个 Linux LVM 分区。

   ```
   /dev/nvme0n1p1         1  204800  204800  209715200   8e  Linux LVM
   /dev/nvme0n1p2    204801  409600  204800  209715200   8e  Linux LVM
   ```

3. 检查分区是否被使用。

   lsblk <*partdevice*>

   <*partdevice*>为上个步骤中查到的 Linux LVM 分区。

本例中，分别执行 **lsblk /dev/nvme0n1p1** 和 **lsblk /dev/nvme0n1p2**，回显如下，表示 nvme0n1p1 分区已被使用，nvme0n1p2 可用。

```
NAME                      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme0n1p1                   259:3    0  200G  0 part
└─vgpaas-thinpool_tdata   251:8    0  360G  0 lvm
  └─vgpaas-thinpool       251:10   0  360G  0 lvm
NAME     MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme0n1p2 259:1    0  100G  0 part
```

若无可用区，请执行步骤 3 为 docker 创建可用区。

步骤 3　为 docker 创建 Linux LVM 型磁盘分区。

1. 执行如下命令，创建磁盘分区。其中 devicename 为可用的裸盘名，如步骤 1 中的 xvdb。

   fdisk /dev/*devicename*

2. 输入 **n** 创建新分区，输入 **p** 展示主分区号，输入 **4** 表示第 4 个主分区。

   图4-2 创建分区

   ```
   Command (m for help): n
   Partition type
      p   primary (0 primary, 0 extended, 4 free)
      e   extended (container for logical partitions)
   Select (default p): p
   Partition number (1-4, default 1): 4
   ```

3. 例如起始扇区和 Last 扇区配置如下：

   ```
   起始 扇区 (1048578048-4294967295，默认为 1048578048)：
   将使用默认值 1048578048
   Last 扇区，+扇区 or +size{K,M,G} (1048578048-4294967294，默认为 4294967294)：+100G
   ```

   表示分区 4 已设置为 Linux 类型，大小为 100GiB。

4. 输入 **t**，修改分区系统类型，并在提示输入时输入 Hex 代码 8e 将系统修改为 Linux LVM 型。

   ```
   命令(输入 m 获取帮助)：t
   分区号 (1-4，默认 4)：4
   Hex 代码(输入 L 列出所有代码)：8e
   已将分区"Linux"的类型更改为"Linux LVM"
   ```

5. 输入 **w**，保存新建分区修改。

   ```
   命令(输入 m 获取帮助)：w
   The partition table has been altered!
   ```

6. 执行 **partprobe** 命令，刷新磁盘分区。

----结束

5 节点池管理

5.1 节点池概述

简介

为帮助您更好地管理 Kubernetes 集群内的节点，云容器引擎 CCE 引入节点池概念。节点池是集群中具有相同配置的一组节点，一个节点池包含一个节点或多个节点。

您可以在 CCE 控制台中创建新的自定义节点池，借助节点池基本功能方便快捷地创建、管理和销毁节点，而不会影响整个集群。新节点池中所有节点的参数和类型都彼此相同，您无法在节点池中配置单个节点，任何配置更改都会影响节点池中的所有节点。

在云容器引擎中，创建集群时配置的节点参数和类型将成为默认节点池“DefaultPool”，该节点池不可编辑、删除或迁移，也不支持扩容、弹性伸缩。

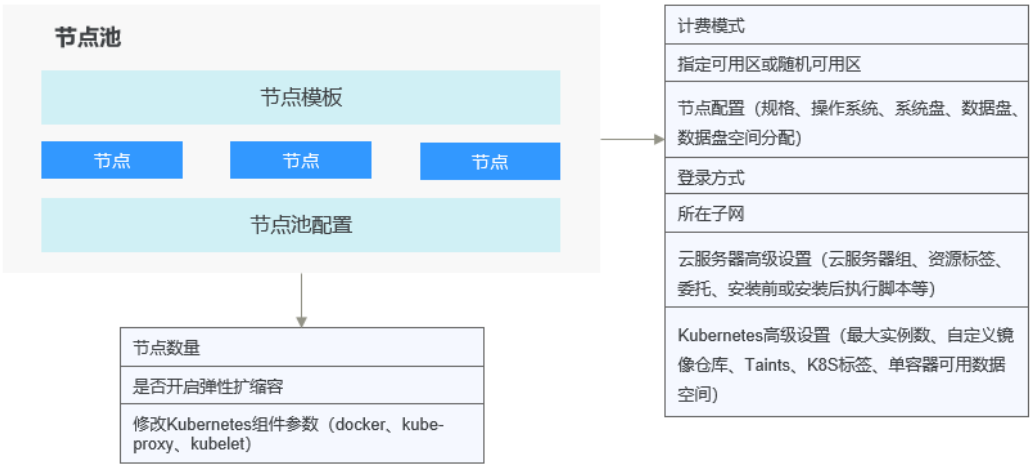
通过节点池功能您还可以实现节点的动态扩缩容：

- 当集群中出现因资源不足而无法调度的实例（Pod）时，自动触发扩容，为您减少人力成本。
- 当满足节点空闲等缩容条件时，自动触发缩容，为您节约资源成本。

本章节介绍节点池在云容器引擎（CCE）中的工作原理，以及如何创建和管理节点池。

节点池架构

图5-1 节点池整体架构图



通常情况下，节点池内的节点均具有如下相同属性：

- 节点操作系统。
- 节点 Kubernetes 组件启动参数。
- 节点自定义启动脚本。
- 节点“K8S 标签”及“Taints”设置。

此外，CCE 将同时围绕节点池扩展以下属性：

- 节点池级别操作系统。
- 节点池级别每节点的 Pod 数上限。

应用场景

当业务需要使用大规模集群时，推荐您使用节点池进行节点管理，以提高大规模集群易用性。

下表介绍了多种大规模集群管理场景，并分别展示节点池在每种场景下发挥的作用：

表5-1 节点池场景及作用

场景	作用
集群存在较多异构节点（机型配置不同）	通过节点池可规范节点分组管理。
集群需要频繁扩缩容节点	通过节点池可降低操作成本。
集群内应用程序调度规则复杂	通过节点池标签可快速指定业务调度规则。
集群内节点日常维护	通过节点池可便捷操作 Kubernetes 版本升级、

场景	作用
	Docker 版本升级。

功能点及注意事项

功能点	功能说明	注意事项
创建节点池	新增节点池， 暂不支持包年包月类型。	单个集群不建议超过 100 个节点池。
删除节点池	删除节点池时会先删除节点池中的节点，原有节点上的工作负载实例会自动迁移至其他节点池的可用节点。	如果工作负载实例具有特定的节点选择器，且如果集群中的其他节点均不符合标准，则工作负载实例可能仍处于无法安排的状态。
节点池开启弹性伸缩	开启弹性伸缩后，节点池将根据集群负载情况自动创建或删除节点池内的节点。	节点池中的节点建议不要放置重要数据，以防止节点被弹性缩容，数据无法恢复。
节点池关闭弹性伸缩	关闭弹性伸缩后，节点池内节点数量不随集群负载情况自动调整。	/
调整节点池大小	支持直接调整节点池内节点个数。若减小节点数量，将从现有节点池内随机缩容节点。	开启弹性伸缩后，不建议手动调整节点池大小。
调整节点池配置	可修改节点池名称、节点个数、K8S 标签、Taints 及资源标签。	修改 K8S 标签和 Taints 会对节点池内节点全部生效，可能会引起 Pod 重新调度，请谨慎变更。
添加已有节点	可添加不属于集群的节点到节点池。要求如下： • 待添加节点需与 CCE 集群在同一虚拟私有云和子网内。 • 实例未被其他集群使用且实例与节点池配置相同（机型、计费模式等）。	无特殊情况时，不建议添加已有节点，推荐直接新建节点池。
移出节点池内节点	可以将同一个集群下某个节点池中的节点迁移到默认节点池（defaultpool）中	暂不支持将默认节点池（defaultpool）中的节点迁移到其他节点池中，也不支持将自定义节点池中的节点迁移到其他自定义节点池。
拷贝节点池	可以方便的拷贝现有节点池的配置，从而创建新的节点池。	/

功能点	功能说明	注意事项
配置 kubernetes 参数	通过该功能您可以对核心组件进行深度配置。	• 本功能仅支持在 v1.15 及以上版本的集群中对节点池进行配置，V1.15 以下版本不显示该功能。 • 默认节点池 DefaultPool 不支持修改该类配置。

将工作负载部署到特定节点池

在定义工作负载时，您可以间接的控制将其部署在哪个节点池上。

例如，您可以通过 CCE 控制台中工作负载页面的“调度策略”设置工作负载与节点的亲和性，强制将该工作负载部署到特定节点池上，从而实现该工作负载仅在该节点池中的节点上运行的目的。如果您需要更好地控制工作负载实例的调度位置，您可以使用 7.1 调度策略概述章节中关于工作负载与节点的亲和或反亲和策略相关说明。

您也可以为容器指定资源请求，工作负载将仅在满足资源请求的节点上运行。

例如，如果工作负载定义了需要包含四个 CPU 的容器，则工作负载将不会选择在具有两个 CPU 的节点上运行。

相关操作

您可以登录 CCE 控制台并参考以下文档， 进行节点池对应的操作：

- 5.2 创建节点池
- 5.3 管理节点池
- 6.2 创建无状态负载(Deployment)
- 7.3.3 工作负载和节点的亲和性

5.2 创建节点池

操作场景

本章介绍了如何添加运行 CCE 集群的节点池以及对节点池执行操作。要了解节点池的工作原理，请参阅 5.1 节点池概述。

操作步骤

将节点池添加到现有集群。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 节点池管理”。

步骤 2 单击右上角的“创建节点池”。

步骤 3 在创建节点池页面中，参照如下说明设置节点池选型参数。

- **计费模式：**

  节点池仅支持“按需计费”的计费模式，该模式将根据实际使用的资源按小时计费。

  节点池创建后，自建的节点池里的资源无法转包周期，默认节点池里的资源可以转包周期。您可以把自建节点池里的资源迁移到默认的节点池里后再进行转包周期的操作。如何迁移请参见迁移节点。

- **当前区域：**指节点实例所在的物理位置。

  请就近选择靠近您业务的区域，可减少网络时延，提高访问速度；不同区域的云服务产品之间内网互不相通。

- **节点池名称：**新建节点池的名称，默认按“集群名-nodepool-随机数”生成名称，可自定义。

- **节点类型：**目前仅支持虚拟机节点。

- **节点购买数量：**该节点池下购买的节点数量，此处设置的节点数不能超过集群管理的最大节点规模，请根据业务需求和界面提示选择，如需更多配额，请单击提交工单申请扩大配额。

- **弹性扩缩容：**
  - 默认不开启。
  - 单击 开启后，节点池将根据集群负载情况自动创建或删除节点池内的节点，参数设置如下：
    - 节点数上限和节点数下限：您可设置节点数的上限和下限，保证节点数在合理的范围内伸缩。
    - 优先级：请根据业务需要设置相应数值，该数值表示节点池之间进行弹性扩缩容的优先级，数值越大优先级越高，如设置为 4 的节点池比设置为 1 的节点池优先启动弹性伸缩。若多个节点池的值设置相同，如都设置为 2，表示这几个节点池之间不分优先级，系统将按最小资源浪费原则进行伸缩。
    - 弹性缩容冷却时间：请设置时间，单位为分钟或小时。弹性缩容冷却时间是指当前节点池扩容出的节点多长时间不能被缩容。

  说明

  节点池中的节点建议不要放置重要数据，以防止节点被弹性缩容，数据无法恢复。

  为保证功能的正常使用，节点池开启弹性扩缩容功能后，请务必安装 13.5 autoscaler。

- **可用区：**可用区是在同一区域下，电力、网络隔离的物理区域，可用区之间内网互通，不同可用区之间物理隔离。

  请根据业务需要进行选择。**节点池创建之后不支持修改可用区属性，请谨慎选择。**

  如果您需要提高工作负载的高可靠性，建议您选择**“随机可用区”**，将节点随机均匀分布在不同可用区中。

- **节点规格：**请根据业务需求选择相应的节点规格。

- 通用型：该类型实例提供均衡的计算、存储以及网络配置，适用于大多数的使用场景。通用型实例可用于 Web 服务器、开发测试环境以及小型数据库工作负载等场景。
- 内存优化型：该类型实例提供内存比例更高的实例，可以用于对内存要求较高、数据量大的工作负载，例如关系数据库、NoSQL 等场景。
- 通用入门型：该类型实例提供均衡的计算、存储以及网络配置，利用 CPU 积分机制保证基准性能，适合平时不会持续高压力使用 CPU，但偶尔需要提高计算性能完成工作负载的场景，可用于轻量级 Web 服务器、开发、测试环境以及中低性能数据库等场景。
- 通用计算增强型：该类型实例具有性能稳定且资源独享的特点，满足计算性能高且稳定的企业级工作负载诉求。

须知

为确保节点稳定性，系统会自动预留部分资源，用于运行必须的系统组件，详细请参见 4.12 节点预留资源计算公式 。

- **操作系统：**请直接选择节点对应的操作系统。

须知

重装操作系统或修改操作系统配置将导致节点不可用，请务必谨慎操作，具体请参见 2 高危操作及解决方案。

- **虚拟私有云：**跟随集群，不可变更。
- **所在子网：**通过子网提供与其他网络隔离的、可以独享的网络资源，以提高网络安全。

  可选择该集群虚拟私有云下的任意子网，集群节点支持跨子网。

  请确保子网下的 DNS 服务器可以解析 OBS 服务域名，否则无法创建节点。
- **系统盘：**设置工作节点的系统盘空间。您可以设置系统盘的规格为 40GB-1024GB 之间的数值，缺省值为 40GB。

  在默认情况下，系统盘可提供高 IO、超高 IO 两种基本的云硬盘类型。
- **数据盘：**设置工作节点的数据盘空间。您可以设置数据盘的规格为 100GB-32768GB 之间的数值，缺省值为 100GB。数据盘可提供的云硬盘类型与系统盘一致，此处不再赘述，详情参见系统盘中的云硬盘类型介绍。

⚠ 注意

若数据盘卸载或损坏，会导致 docker 服务异常，最终导致节点不可用。建议不要删除该数据盘。

- **登录方式：**支持密码和密钥对。

- 选择“密码”：用户名默认为“root”，请输入登录节点的密码，并确认密码。

  登录节点时需要使用该密码，请妥善管理密码，系统无法获取您设置的密码内容。
- 选择“密钥对”：在选项框中选择用于登录本节点的密钥对，并单击勾选确认信息。

  密钥对用于远程登录节点时的身份认证。若没有密钥对，可单击选项框右侧的“创建密钥对”来新建，创建密钥对操作步骤请参见。

步骤 4 **云服务器高级设置**：（可选），单击^ 展开后可对节点进行如下高级功能配置：

- **安装前执行脚本**：请输入脚本命令，大小限制为 0~1000 字符。

  脚本将在 Kubernetes 软件安装前执行，可能导致 Kubernetes 软件无法正常安装，需谨慎使用。常用于格式化数据盘等场景。
- **安装后执行脚本**：请输入脚本命令，大小限制为 0~1000 字符。

  脚本将在 Kubernetes 软件安装后执行，不影响 Kubernetes 软件安装。常用于修改 Docker 配置参数等场景。
- **新增数据盘**：单击“新增数据盘”，选择云硬盘类型并输入数据盘规格。
- **子网 IP**：可选择“自动分配 IP 地址”和“手动分配 IP 地址”，推荐使用“自动分配 IP 地址”。

步骤 5 **Kubernetes 高级设置**：（可选），单击^ 展开后可对集群进行如下高级功能配置：

- **最大实例数**：节点最大允许创建的实例数(Pod)，该数量包含系统默认实例，取值范围为 16~128。

  该设置的目的为防止节点因管理过多实例而负载过重，请根据您的业务需要进行设置。
- **自定义镜像仓库**：单击“新增自定义镜像仓库地址”输入镜像仓库地址。

  添加自定义镜像仓库地址（非 SSL 镜像源地址）到 docker 启动参数中，避免拉取个人镜像仓库的镜像失败，格式可为“IP 地址:端口或者域名”。安装后执行脚本与自定义镜像仓库不能同时使用。
- **Taints**：默认为空。支持给该节点池扩容出来的节点加 Taints 来设置反亲和性，每个节点池最多配置 10 条 Taints，每条 Taints 包含以下 3 个参数：
  - Key：必须以字母或数字开头，可以包含字母、数字、连字符、下划线和点，最长 63 个字符；另外可以使用 DNS 子域作为前缀。
  - Value：必须以字符或数字开头，可以包含字母、数字、连字符、下划线和点，最长 63 个字符。
  - Effect：只可选 NoSchedule，PreferNoSchedule 或 NoExecute。

须知

- Taints 配置时需要配合 Pod 的 toleration 使用，否则可能导致扩容失败或者 Pod 无法调度到扩容节点。
- 节点池创建后可单击列表项的“编辑”修改配置，修改后将同步到节点池下的已有节点。

- **K8S 标签：**K8S 标签是附加到 Kubernetes 对象（比如 Pods）上的键值对，旨在用于指定对用户有意义且相关的对象的标识属性，但不直接对核心系统有语义含义。 详细请参见 Labels and Selectors。
- **单容器可用数据空间：**该参数用于设置一个容器可用的数据空间大小，设置范围为 10G 到 80G。如果设置的参数超过数据盘中 Docker 可占用的实际数据空间（由数据盘设置项中的资源分配自定义参数指定，默认为数据盘大小的 90%），将以 Docker 的实际空间大小为主。该参数仅在 v1.13.10-r0 及以上版本的集群中显示。

步骤 6 （可选）您可以单击左侧的 添加节点池 按钮添加多个节点池，在按钮下方可以查看您可用的节点池配额数量。

步骤 7 完成配置后，单击“下一步：配置确认”，确认所设置的服务选型参数、规格和费用等信息。

步骤 8 确认规格和费用后，单击“提交”，节点池开始创建。

节点池创建预计需要 6-10 分钟，您可以单击“返回节点池管理”进行其他操作或单击“查看节点池事件列表”后查看节点池详情。待节点池状态为“正常”，表示节点池创建成功。

----结束

查看创建的节点池

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 节点池管理”。

步骤 2 在节点池管理页面中，单击右上角的集群选择框，选择集群后可显示当前集群下所有的节点池，并可查看每个节点池的节点类型、节点规格、弹性扩缩容状态和操作系统等。

说明

- 节点池功能上线后，会在每个集群中创建一个默认节点池“DefaultPool”，该节点池不能被编辑、删除或迁移，集群中原有的节点及节点池外创建的节点均会显示在默认节点池“DefaultPool”中。
- 单击默认节点池“DefaultPool”中“节点”数据框，可查看 DefaultPool 中的节点列表。

步骤 3 单击右上角的 Autoscaler 状态选择框，可筛选全部、已启用、未启用 Autoscaler 功能的节点池。

步骤 4 在节点池列表中，单击节点池的名称，在节点池详情页面，可查看节点池的基本信息、ECS 高级设置、Kubernetes 高级设置、节点列表等信息。

----结束

5.3 管理节点池

操作场景

为方便对 CCE 集群中的 kubernetes 配置参数进行管理，开发了配置管理功能，通过该功能您可以对核心组件进行深度配置，更多信息请参见 kubelet。

本文将介绍集群创建后对节点池的“配置管理”功能。

约束与限制

- 仅支持在 **v1.15 及以上版本**的集群中对节点池进行配置，V1.15 以下版本不显示该功能。
- 默认节点池 DefaultPool 不支持修改该类配置。

操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 节点池管理”。

步骤 2 在集群选择框中，选择集群，可查看到当前集群下所有的节点池。

步骤 3 单击节点池名称后的“配置管理”。

步骤 4 在侧边栏滑出的“配置管理”窗口中，根据业务需求修改 Kubernetes 的参数值：

表5-2 配置管理参数

组件	参数	详情	默认	修改限制
docker	native-umask	`--exec-opt native.umask	normal	-
	docker-base-size	`--storage-opts dm.basesize	10G	-
kube-proxy	conntrack-min	sysctl -w net.nf_conntrack_max	131072	支持在节点池生命周期中修改
	conntrack-tcp-timeout-close-wait	sysctl -w net.netfilter.nf_conntrack_tcp_timeout_clouse_wait	1h0m0s	
kubelet	cpu-manager-policy	`--cpu-manager-policy	none	支持在节点池生命周期中修改
	kube-api-qps	与 kube-apiserver 通信的 qps	100	
	kube-api-burst	与 kube-apiserver 通信的 burst	100	
	max-pods	kubelet 管理的 pod 上限	110	
	with-local-dns	是否使用本地 IP 作为该	false	

组件	参数	详情	默认	修改限制
		节点的 ClusterDNS		
	allowed-unsafe-sysctls	允许使用的不安全系统配置	[]	

步骤 5 单击“确定”，完成配置操作。

----结束

编辑节点池

步骤 1 登录，在左侧导航栏中选择“资源管理 > 节点池管理”。

步骤 2 在集群选择框中，选择集群，可查看到当前集群下所有的节点池。

步骤 3 单击节点池名称后的“编辑”，在弹出的“编辑节点池”中，配置以下参数：

表5-3 配置节点池参数

参数	参数说明
节点池名称	自定义节点池名称。
节点个数	请根据业务需求调整节点个数。
弹性扩缩容	默认不开启。 单击 开启后，节点池将根据业务需求自动创建或删除节点池内的节点，参数设置如下： • 节点数上限和节点数下限：您可设置节点数的上限和下限，保证节点数在合理的范围内伸缩。 • 优先级：该数值表示节点池之间进行弹性扩缩容的优先级，数值越大优先级越高，如设置为 4 的节点池比设置为 1 的节点池优先启动弹性伸缩。若多个节点池的值设置相同，如都设置为 2，表示这几个节点池之间不分优先级，系统将按最小资源浪费原则进行伸缩。 • 弹性缩容冷却时间：请设置时间，单位为分钟。弹性缩容冷却时间是指当前节点池扩容出的节点多长时间不能被缩容。 为保证功能的正常使用，节点池开启弹性扩缩容功能后，请务必安装 13.5 autoscaler。
编辑密钥对 （仅新加坡 Region 可用）	默认不开启。 单击 开启后，可选择已有密钥对或创建密钥。密钥对用于远程登录节点时的身份认证。
Taints	• 默认为空。支持给该节点池扩容出来的节点加 Taints 来

参数	参数说明
	设置反亲和性，每个节点池最多配置 10 条 Taints，每条 Taints 包含以下 3 个参数： - Key：必须以字母或数字开头，可以包含字母、数字、连字符、下划线和点，最长 63 个字符；另外可以使用 DNS 子域作为前缀。 - Value：必须以字符或数字开头，可以包含字母、数字、连字符、下划线和点，最长 63 个字符。 - Effect：只可选 NoSchedule，PreferNoSchedule 或 NoExecute。 须知 Taints 配置时需要配合 Pod 的 toleration 使用，否则可能导致扩容失败或者 Pod 无法调度到扩容节点。
K8S 标签	K8S 标签是附加到 Kubernetes 对象（比如 Pods）上的键值对，旨在用于指定对用户有意义且相关的对象的标识属性，但不直接对核心系统有语义含义。 详细请参见 Labels and Selectors。
资源标签	如果您需要使用同一标签标识多种云资源，即所有服务均可在标签输入框下拉选择同一标签，建议在 TMS 中创建“预定义标签”。 预定义标签对所有支持标签功能的服务资源可见，通过使用预定义标签可以提升标签创建和迁移效率。具体请参见。 标签更改不会影响原有节点。

步骤 4 配置完成后，单击“保存”。

在节点池列表中可查看到节点池状态为“伸缩中”，待状态变为“完成”，节点池参数编辑成功，修改的配置后将同步到节点池下的已有节点。

----结束

删除节点池

删除节点池，会先删除节点池中的节点，节点删除后，原有节点上的工作负载实例会自动迁移至其他节点池的可用节点。如果工作负载实例具有特定的节点选择器，且如果集群中的其他节点均不符合标准，则工作负载实例可能仍处于无法安排的状态。

步骤 1 登录，在左侧导航栏中选择“资源管理 > 节点池管理”。

步骤 2 在集群选择框中，选择集群，可查看到当前集群下所有的节点池。

步骤 3 单击节点池名称后的“更多 > 删除”。

步骤 4 在弹出的“删除节点池”窗口中，请仔细阅读界面提示。

步骤 5 确定要对节点池进行删除操作后，请在输入框中输入 **DELETE** 并单击“是”，即可完成节点池的删除。

----结束

拷贝节点池

通过 CCE 控制台可以方便的拷贝现有节点池的配置，从而创建新的节点池。

步骤 1 登录，在左侧导航栏中选择“资源管理 > 节点池管理”。

步骤 2 在集群选择框中，选择集群，可查看到当前集群下所有的节点池。

步骤 3 单击节点池名称后的“更多 > 拷贝”。

步骤 4 在弹出的“创建节点池”窗口中，可以看到拷贝的节点池配置，您可以根据需要进行修改，确定配置后单击“下一步：配置确认”。

步骤 5 在“配置确认”步骤中再次确认规格并单击“提交”，即可完成节点池的拷贝并创建新的节点池。

----结束

迁移节点

您可以将同一个集群下某个节点池中的节点迁移到默认节点池（defaultpool）中，暂不支持将默认节点池（defaultpool）中的节点迁移到其他节点池中，也不支持将自定义节点池中的节点迁移到其他自定义节点池。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 节点池管理”。

步骤 2 在集群选择框中，选择集群，可查看到当前集群下所有的节点池。

步骤 3 单击待迁移的节点池名称后的“更多 > 迁移”。

步骤 4 在弹出的“节点迁移”窗口中，选择目标节点池和待迁移节点。

说明

迁移完成后，节点原有资源标签、K8S 标签、Taints 不受影响，并将为其打上未拥有的目标节点池的 K8S 标签、Taints。

步骤 5 单击“确定”，即可完成节点迁移。

----结束

6 工作负载

6.1 工作负载概述

工作负载即 Kubernetes 对一组 Pod 的抽象模型，用于描述业务的运行载体，包括 Deployment、Statefulset、Daemonset、Job、CronJob 等多种类型。

云容器引擎 CCE 提供基于 Kubernetes 原生类型的容器部署和管理能力，支持容器工作负载部署、配置、监控、扩容、升级、卸载、服务发现及负载均衡等生命周期管理。

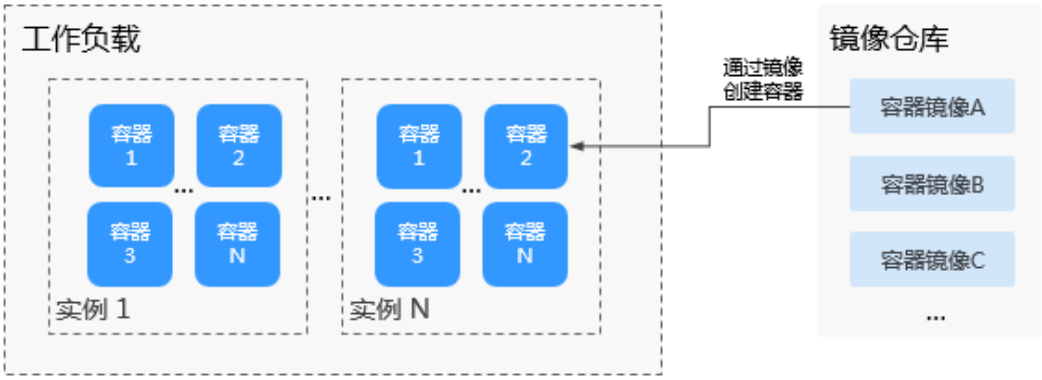
基本概念

- **无状态工作负载：**即 kubernetes 中的“Deployment”，无状态工作负载支持弹性伸缩与滚动升级，适用于实例完全独立、功能相同的场景，如：nginx、wordpress 等。创建无状态工作负载请参见 6.2 创建无状态负载(Deployment)。
- **有状态工作负载：**即 kubernetes 中的“StatefulSet”，有状态工作负载支持实例有序部署和删除，支持持久化存储，适用于实例间存在互访的场景，如 ETCD、mysql-HA 等。创建有状态工作负载请参见 6.3 创建有状态负载(StatefulSet)。
- **创建守护进程集：**即 kubernetes 中的“DaemonSet”，守护进程集确保全部（或者某些）节点都运行一个 Pod 实例，支持实例动态添加到新节点，适用于实例在每个节点上都需要运行的场景，如 ceph、fluentd、Prometheus Node Exporter 等。创建守护进程集请参见 6.4 创建守护进程集(DaemonSet)。
- **普通任务：**即 kubernetes 中的“Job”，普通任务是一次性运行的短任务，部署完成后即可执行。使用场景为在创建工作负载前，执行普通任务，将镜像上传至镜像仓库。创建普通任务请参见 6.5 创建普通任务(Job)。
- **定时任务：**即 kubernetes 中的“CronJob”，定时任务是按照指定时间周期运行的短任务。使用场景为在某个固定时间点，为所有运行中的节点做时间同步。创建定时任务请参见 6.6 创建定时任务(CronJob)。

工作负载与容器间的关系

如图 6-1 所示，一个工作负载由一个或多个实例（Pod）组成。一个实例由一个或多个容器组成，每个容器都对应一个容器镜像。对于无状态工作负载，实例都是完全相同的。

图6-1 工作负载与容器的关系



工作负载生命周期说明

表6-1 状态说明

状态	说明
运行中	所有实例都处于运行中才是运行中。
未就绪	容器处于异常、实例数为 0 或 pending 状态时显示此状态。
升级/回滚中	触发升级或回滚动作后，工作负载会处于升级/回滚中。
可用	当多实例无状态工作负载运行过程中部分实例异常，可用实例不为 0，工作负载会处于可用状态。
执行完成	任务执行完成，仅普通任务存在该状态。
已停止	触发停止操作后，工作负载会处于停止状态，实例数变为 0。v1.13 之前的版本存在此状态。
删除中	触发删除操作后，工作负载会处于删除中状态。
暂停中	触发暂停操作后，工作负载会处于暂停中状态。

相关操作

由于社区 pod 结构体中没有 ExtendPathMode，用户使用 client-go 调用创建 pod 或者 deployment 的 api 接口时，创建的 pod 中没有 ExtendPathMode。为了与社区的 client-go 兼容，CCE 提供了相关解决方案，具体请参见《常见问题》> Pod 接口 ExtendPathMode: PodUID 如何与社区 client-go 兼容。

6.2 创建无状态负载(Deployment)

操作场景

在运行中始终不保存任何数据或状态的工作负载称为“无状态负载 Deployment”，例如 nginx。您可以通过控制台或 kubectl 命令行创建无状态负载。

前提条件

- 在创建容器工作负载前，您需要存在一个可用集群。若没有可用集群 ，请参照 3.3 购买混合集群中内容创建。
- 若工作负载需要被外网访问，请确保集群中至少有一个节点已绑定弹性 IP，或已购买负载均衡实例。

说明

创建多个工作负载时，请确保容器使用的端口不冲突 ，否则部署会失败。

通过控制台创建

云容器引擎提供了多种创建工作负载的方式，您可以通过如下方式进行创建：

- 基于“开源镜像中心”创建工作负载，无需上传镜像。
- 基于“我的镜像”创建工作负载，用户首先需要将镜像上传至容器镜像服务。
- 基于“共享镜像”创建工作负载，即其它租户通过“容器镜像服务”共享给您的镜像。
- 您希望通过 YAML 方式创建工作负载，您可在“创建无状态工作负载”高级设置页面单击界面右侧的“YAML 创建”，通过 yaml 的方式创建工作负载。YAML 的说明请参见表 6-4。YAML 编写完成后，可单击“创建”，直接创建工作负载。

说明

YAML 文件是和界面保持同步的，您也可以通过界面和 YAML 互动完成工作负载的创建。例如：

- 界面中填写工作负载名称后，YAML 文件会自动关联该名称。
- 界面中添加完镜像后，YAML 中也会自动关联该镜像。

控制台界面右侧的“YAML 创建”不支持多个 YAML 混合，请分别创建，否则创建时将会报错。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”，单击“创建无状态工作负载”。参照表 6-2 设置工作负载基本信息，其中带“*”标志的参数为必填参数。

表6-2 工作负载基本信息

参数	参数说明
* 工作负载名称	新建工作负载的名称，命名必须唯一。 请输入 4 到 63 个字符的字符串，可以包含小写英文字母、数字

参数	参数说明
	和中划线（-），并以小写英文字母开头，小写英文字母或数字结尾。
* 集群名称	新建工作负载所在的集群。
* 命名空间	在单集群中，不同命名空间中的数据彼此隔离。使应用可以共享同个集群的服务，也能够互不干扰。若您不设置命名空间，系统会默认使用 default 命名空间。
* 实例数量	工作负载的实例数量。工作负载可以有一个或多个实例，用户可以设置具体实例个数，默认为 2，可自定义设置为 1。 每个工作负载实例都由相同的容器部署而成。设置多个实例主要用于实现高可靠性，当某个实例故障时，工作负载还能正常运行。若使用单实例，节点异常或实例异常会导致服务异常。
时区同步	单击 开启后，容器将和节点使用相同时区。 **须知** **时区同步功能开启后，在“数据存储 > 本地磁盘”中，将会自动添加 HostPath 类型的磁盘，请勿修改删除该磁盘。**
工作负载描述	工作负载描述信息。

步骤 2 单击“下一步：容器设置”，添加容器。

1. 单击“添加容器”，选择需要部署的镜像。
   - **我的镜像：**展示了您创建的所有镜像仓库。
   - **第三方镜像：**CCE 支持拉取第三方镜像仓库（即镜像仓库与开源镜像中心之外的镜像仓库）的镜像创建工作负载。使用第三方镜像时，请确保工作负载运行的节点可访问公网。第三方镜像的具体使用方法请参见 6.10.1 如何使用第三方镜像。
     - 若您的镜像仓库不需要认证，密钥认证请选择“否”，并输入“镜像名称”，单击“确定”。
     - 若您的镜像仓库都必须经过认证（帐号密码）才能访问，您需要先创建密钥再使用第三方镜像，具体操作请参见 6.10.1 如何使用第三方镜像。
   - **共享镜像：**其它租户通过“容器镜像服务”共享给您的镜像将在此处展示，您可以基于共享镜像创建工作负载。
2. 配置镜像基本信息。

   工作负载是 Kubernetes 对一组 Pod 的抽象模型，用于描述业务的运行载体，一个 Pod 可以封装 1 个或多个容器，您可以单击右上方的“添加容器”，添加多个容器镜像并分别进行设置。

表6-3 镜像参数说明

参数	说明

参数	说明
镜像名称	导入的镜像，您可单击“更换镜像”进行更换。
* 镜像版本	选择需要部署的镜像版本。
* 容器名称	容器的名称，可修改。
特权容器	特权容器是指容器里面的程序具有一定的特权。 若选中，容器将获得超级权限，例如可以操作宿主机上面的网络设备、修改内核参数等。
容器规格	**CPU 配额：** • 申请：容器需要使用的最小 CPU 值，默认 0.25Core。 • 限制：允许容器使用的 CPU 最大值。建议设容器配额的最高限额，避免容器资源超额导致系统故障。 **内存配额：** • 申请：容器需要使用的内存最小值，默认 512MiB。 • 限制：允许容器使用的内存最大值。如果超过，容器会被终止。 申请和限制的具体请参见 6.10.2 设置容器规格。 **GPU 配额：**当集群中包含 GPU 节点时，才能设置 GPU，无 GPU 节点不显示此选项。 容器需要使用的 GPU 百分比。勾选“使用”并设置百分比，例如设置为 10%，表示该容器需使用 GPU 资源的 10%。若不勾选“使用”，或设置为 0，则无法使用 GPU 资源。 **GPU 显卡：**工作负载实例将被调度到 GPU 显卡类型为指定显卡的节点上。 若勾选“不限制”，容器将会随机使用节点中的任一显卡。您也可以勾选某个显卡，容器将使用特定显卡。

3. **生命周期：**用于设置容器启动和运行时需要执行的命令。
   - 启动命令：设置容器启动时执行的命令，具体请参见 6.10.4 设置容器启动命令。
   - 启动后处理：设置容器成功运行后执行的命令，详细配置方法请参见 6.10.3 设置容器生命周期。
   - 停止前处理：设置容器结束前执行的命令，通常用于删除日志/临时文件等，详细配置方法请参见 6.10.3 设置容器生命周期。
4. **健康检查：**CCE 提供了存活与业务两种探针，用于判断容器和用户业务是否正常运行。详细配置方法请参见 6.10.5 设置容器健康检查。
   - 工作负载存活探针：检查容器是否正常，不正常则重启实例。
   - 工作负载业务探针：检查用户业务是否就绪，不就绪则不转发流量到当前实例。
5. **环境变量：**在容器中添加环境变量，一般用于通过环境变量设置参数。

在“环境变量”页签，单击“添加环境变量”，当前支持三种类型：

- 手动添加：输入变量名称、变量/变量引用。
- 密钥导入：输入变量名称，选择导入的密钥名称和数据。您需要提前创建密钥，具体请参见 11.3 创建密钥。
- 配置项导入：输入变量名称，选择导入的配置项名称和数据。您需要提前创建配置项，具体请参见 11.1 创建配置项。

说明

对于已设置的环境变量，单击环境变量后的“编辑”，可对该环境变量进行编辑。单击环境变量后的“删除”，可删除该环境变量。

6. **数据存储：**给容器挂载数据存储，支持本地磁盘和云存储，适用于需持久化存储、高磁盘 IO 等场景。具体请参见 9.2 本地磁盘存储、9.5 极速文件存储卷。
7. **安全设置：**对容器权限进行设置，保护系统和其他容器不受其影响。

   请输入用户 ID，容器将以当前用户权限运行。
8. **容器日志：**设置容器日志采集策略、配置日志目录。用于收集容器日志便于统一管理和分析。详细配置请参见 6.10.7 采集容器标准输出日志、6.10.8 采集容器内路径日志。

步骤 3 单击“下一步：工作负载访问设置”，单击“添加服务”，设置工作负载访问方式。

若工作负载需要和其它服务互访，或需要被公网访问，您需要添加服务，设置工作负载访问方式。

工作负载访问的方式决定了这个工作负载的网络属性，不同访问方式的工作负载可以提供不同网络能力，具体请参见 8.1 网络概述。

步骤 4 单击“下一步：高级设置”，配置更多高级策略。

- **升级策略：**您可以指定无状态工作负载的升级方式，包括逐步“滚动升级”和整体“替换升级”。
  - 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：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。
- **缩容策略：**为工作负载删除提供一个时间窗，预留给生命周期中 PreStop 阶段执行命令。若超过此时间窗，进程仍未停止，该工作负载将被强制删除。
  - 缩容时间窗 (s)：请输入时间，该时间为工作负载停止前命令的执行时间窗（0-9999 秒），默认 30 秒。
  - 缩容优先级：可根据业务需要选择“优先减少新实例”或“优先减少老实例”。
- **迁移策略：**当工作负载实例所在的节点不可用时，系统将实例重新调度到其它可用节点的时间窗。
  - 迁移时间窗 (s)：请输入时间，默认为 300 秒。

- **调度策略：**您可以根据需要自由组合静态的全局调度策略或动态的运行时调度策略来实现自己的需求。具体请参见 7.1 调度策略概述。
- **Pod 高级设置**
  - Pod 标签：内置 app 标签在工作负载创建时指定，主要用于设置亲和性与反亲和性调度，暂不支持修改。您可以单击下方的“添加标签”增加标签。

图6-2 Pod 高级设置

- **客户端 DNS 配置**：CCE 集群内置 DNS 插件 CoreDNS，为集群内的工作负载提供域名解析服务。详细使用方法请参见 6.10.11 Kubernetes 集群内置 DNS 配置说明。
  - DNS 策略：
    - 追加域名解析配置：选择该配置后，将保留默认配置，以下“IP 地址”和“搜索域”配置可能不生效。
    - 替换域名解析配置：选择该配置后，将仅使用以下“IP 地址”和“搜索域”配置进行域名解析。
    - 继承 Pod 所在节点域名解析配置：将继承 Pod 所在节点的域名解析配置。
  - IP 地址：您可对自定义的域名配置域名服务器，值为一个或一组 DNS IP 地址，如“1.2.3.4”。
  - 搜索域：定义域名的搜索域列表，当访问的域名不能被 DNS 解析时，会把该域名与搜索域列表中的域依次进行组合，并重新向 DNS 发起请求，直到域名被正确解析或者尝试完搜索域列表为止。
  - 超时时间（s）：查询超时时间，请自定义。
  - ndots：表示域名中必须出现的“.”的个数，如果域名中的“.”的个数不小于 ndots，则该域名为一个 FQDN，操作系统会直接查询；如果域名中的“.”的个数小于 ndots，操作系统会在搜索域中进行查询。
- **自定义指标监控：**是指监控系统提供的一种指标收集机制，该机制允许工作负载在部署时自定义需要上报的指标名称以及获取这些指标数据的接入点信息，在应用运行时由监控系统按固定的频率访问接入点进行指标的收集。详细请参见 6.10.9 对接 Prometheus 实现自定义指标监控。
- **性能管理配置：**的性能管理服务可协助您快速进行工作负载的问题定位与性能瓶颈分析。详细请参见 6.10.10 **错误!未找到引用源。**。

步骤 5 配置完成后，单击“创建”，在创建成功页面单击“返回工作负载列表”，查看工作负载状态。

在工作负载列表中，当工作负载状态为“运行中”时，表示工作负载创建成功。

工作负载状态不会实时更新，请单击右上角的 图标或按 F5 刷新页面查看。

步骤 6 在“无状态负载 Deployment”页面的工作负载列表中，复制“外部访问地址”，可在浏览器中访问工作负载。

说明

- 当工作负载访问方式设为“节点访问（NodePort）”并绑定弹性 IP 或设为“负载均衡（LoadBalancer）”时，才可以获取外部访问地址，可以访问外网。
- 工作负载列表页在超过 500 条以上时，将采用 Kubernetes 的分页机制进行分页。Kubernetes 的分页机制：仅支持回到第一页和查看下一页，不支持查看上一页，且在分页显示的情况下，资源总数显示的是批量查询出的数目而不是真实总数。

----结束

通过 kubectl 命令行创建

本节以 nginx 工作负载为例，说明 kubectl 命令创建工作负载的方法。

前提条件

请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令，使弹性云主机连接集群。

操作步骤

步骤 1 登录已配置好 kubectl 命令的弹性云主机。

步骤 2 创建一个名为 nginx-deployment.yaml 的描述文件。其中，nginx-deployment.yaml 为自定义名称，您可以随意命名。

vi nginx-deployment.yaml

描述文件内容如下。此处仅为示例，deployment 的详细说明请参见 kubernetes 官方文档。

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        imagePullPolicy: Always
```

```
      name: nginx
    imagePullSecrets:
    - name: default-secret
```

以上 yaml 字段解释如表 6-4。

表6-4 deployment 字段详解

字段名称	字段说明	必选/可选
apiVersion	表示 API 的版本号。 说明 集群版本为 1.9 之前的无状态应用 apiVersion 格式为 extensions/v1beta1，1.9 之后的集群兼容 extensions/v1beta1 和 apps/v1 两种格式 Version，请根据集群版本输入。	必选
kind	创建的对象类别。	必选
metadata	资源对象的元数据定义。	必选
name	deployment 的名称。	必选
Spec	用户对 deployment 的详细描述的主体部分都在 spec 中给出。	必选
replicas	实例数量。	必选
selector	定义 Deployment 可管理的容器实例。	必选
strategy	升级类型。当前支持两种升级方式，默认为滚动升级。 • RollingUpdate：滚动升级。 • ReplaceUpdate：替换升级。	可选
template	描述创建的容器实例详细信息。	必选
metadata	元数据。	必选
labels	metadata.labels 定义容器标签。	可选
spec: containers	• image（必选）：容器镜像名称。 • imagePullPolicy（可选）：获取镜像的策略，可选值包括 Always（每次都尝试重新下载镜像）、Never（仅使用本地镜像）、IfNotPresent（如果本地有该镜像，则使用本地镜像，本地不存在时下载镜像），默认为 Always。 • name（必选）：容器名称。	必选

字段名称	字段说明	必选/可选
imagePullSecrets	Pull 镜像时使用的 secret 名称。若使用私有镜像，该参数为必选。 • 需要 Pull SWR 容器镜像仓库的镜像时，参数值固定为 default-secret。 • 当 Pull 第三方镜像仓库的镜像时，需设置为创建的 secret 名称。	可选

步骤 3 创建 deployment。

kubectl create -f nginx-deployment.yaml

回显如下表示已开始创建 deployment。

```
deployment "nginx" created
```

步骤 4 查看 deployment 状态。

kubectl get pods

deployment 状态显示为 Running，表示 deployment 已创建成功。

```
NAME                     READY     STATUS    RESTARTS   AGE
icagent-m9dkt            0/0       Running   0          3d
nginx-1212400781-qv313   1/1       Running   0          3d
```

参数解析：

- NAME：pod 的名称
- READY：已经部署完毕的 pod 副本数
- STATUS：状态
- RESTARTS：重启次数
- AGE：已经运行的时间

步骤 5 若工作负载（即 deployment）需要被访问（集群内访问或节点访问），您需要设置访问方式，具体请参见 8 网络管理创建对应服务。

----结束

6.3 创建有状态负载(StatefulSet)

操作场景

在运行过程中会保存数据或状态的工作负载称为“有状态工作负载（statefulset）”。例如 Mysql，它需要存储产生的新数据。

因为容器可以在不同主机间迁移，所以在宿主机上并不会保存数据，这依赖于 CCE 提供的高可用存储卷，将存储卷挂载在容器上，从而实现有状态工作负载的数据持久化。

前提条件

- 在创建容器工作负载前，您需要存在一个可用集群。若没有请参照 3.3 购买混合集群中内容创建。
- 若工作负载需要被外网访问，请确保集群中至少有一个节点已绑定弹性 IP，或已购买负载均衡实例。

说明

创建多个工作负载时，请确保容器使用的端口不冲突 ，否则部署会失败。

通过控制台创建

云容器引擎提供了多种创建工作负载的方式，您可以通过如下方式进行创建：

1. 基于“开源镜像中心”创建工作负载，无需上传镜像。
2. 基于“我的镜像”创建工作负载，用户首先需要将镜像上传至容器镜像服务。
3. 基于“共享镜像”创建工作负载，即其它租户通过“容器镜像服务”共享给您的镜像。
4. 若您希望通过 YAML 方式创建工作负载，您可在“创建有状态工作负载”页面单击界面右侧的“YAML 创建”，通过 yaml 的方式创建工作负载。YAML 的说明请参见通过 kubectl 命令行创建。YAML 编写完成后，可单击“创建”，直接创建工作负载。

说明

YAML 文件是和界面保持同步的，您也可以通过界面和 YAML 互动完成工作负载的创建。例如：

- 界面中填写工作负载名称后，YAML 文件会自动关联该名称。
- 界面中添加完镜像后，YAML 中也会自动关联该镜像。

控制台界面右侧的“YAML 创建”不支持多个 YAML 混合，请分别创建，否则创建时将会报错。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 有状态负载 StatefulSet”，单击“创建有状态工作负载”。在打开的创建有状态工作负载页面中，参照表 6-5 设置工作负载基本信息，其中带“*”标志的参数为必填参数。

表6-5 工作负载基本信息

参数	参数说明
* 工作负载名称	新建工作负载的名称，命名必须唯一。 请输入 4 到 52 个字符的字符串，可以包含小写英文字母、数字和中划线（-），并以小写英文字母开头，小写英文字母或数字结尾。

参数	参数说明
* 集群名称	新建工作负载所在的集群。
* 命名空间	在单集群中，不同命名空间中的数据彼此隔离。使应用可以共享同个集群的服务，也能够互不干扰。若您不设置命名空间，系统会默认使用 default 命名空间。
* 实例数量	工作负载的实例数量。工作负载可以有一个或多个实例，用户可以设置具体实例个数，默认为 2，可自定义设置为 1。 每个工作负载实例都由相同的容器部署而成。设置多个实例主要用于实现高可靠性，当某个实例故障时，工作负载还能正常运行。若使用单实例，节点异常或实例异常会导致服务异常。
时区同步	单击 开启后，容器将和节点使用相同时区。 须知 时区同步功能开启后，在“数据存储 > 本地磁盘”中，将会自动添加 HostPath 类型的磁盘，请勿修改删除该磁盘。
工作负载描述	工作负载描述信息。

步骤 2 单击“下一步：容器设置”，添加容器。

1. 单击“添加容器”，选择需要部署的镜像。
   - **我的镜像**：展示了您创建的所有镜像仓库。
   - **开源镜像中心**：展示了开源镜像中心仓库中的官方镜像。
   - **第三方镜像**：CCE 支持拉取第三方镜像仓库（即镜像仓库与开源镜像中心之外的镜像仓库）的镜像创建工作负载。使用第三方镜像时，请确保工作负载运行的节点可访问公网。第三方镜像的具体使用方法请参见 6.10.1 如何使用第三方镜像。
     - 若您的镜像仓库不需要认证，密钥认证请选择“否”，并输入“镜像名称”，单击“确定”。
     - 若您的镜像仓库都必须经过认证（帐号密码）才能访问，您需要先创建密钥再使用第三方镜像，具体操作请参见 6.10.1 如何使用第三方镜像。
   - **共享镜像**：其它租户通过“容器镜像服务”共享给您的镜像将在此处展示，您可以基于共享镜像创建工作负载。
2. 配置镜像基本信息。

   工作负载是 Kubernetes 对一组 Pod 的抽象模型，用于描述业务的运行载体，一个 Pod 可以封装 1 个或多个容器，您可以单击右上方的“添加容器”，添加多个容器镜像并分别进行设置。

表6-6 镜像参数说明

参数	说明
镜像名称	导入的镜像，您可单击“更换镜像”进行更换。

参数	说明
* 镜像版本	选择需要部署的镜像版本。
* 容器名称	容器的名称，可修改。
特权容器	特权容器是指容器里面的程序具有一定的特权。 若选中，容器将获得超级权限，例如可以操作宿主机上面的网络设备、修改内核参数等。
容器规格	**CPU 配额：** • 申请：容器需要使用的最小 CPU 值，默认 0.25Core。 • 限制：允许容器使用的 CPU 最大值。建议设容器配额的最高限额，避免容器资源超额导致系统故障。 **内存配额：** • 申请：容器需要使用的内存最小值，默认 512MiB。 • 限制：允许容器使用的内存最大值。如果超过，容器会被终止。 申请和限制的具体请参见 6.10.2 设置容器规格。 **GPU 配额：**当集群中包含 GPU 节点时，才能设置 GPU，无 GPU 节点不显示此选项。 容器需要使用的 GPU 百分比。勾选“使用”并设置百分比，例如设置为 10%，表示该容器需使用 GPU 资源的 10%。若不勾选“使用”，或设置为 0，则无法使用 GPU 资源。 **GPU 显卡：**工作负载实例将被调度到 GPU 显卡类型为指定显卡的节点上。 若勾选“不限制”，容器将会随机使用节点中的任一显卡。您也可以勾选某个显卡，容器将使用特定显卡。

3. **生命周期：**用于设置容器启动和运行时需要执行的命令。
   - 启动命令：设置容器启动时执行的命令，具体请参见 6.10.4 设置容器启动命令。
   - 启动后处理：设置容器成功运行后执行的命令，详细配置方法请参见 6.10.3 设置容器生命周期。
   - 停止前处理：设置容器结束前执行的命令，通常用于删除日志/临时文件等，详细配置方法请参见 6.10.3 设置容器生命周期。
4. **健康检查：**CCE 提供了存活与业务两种探针，用于判断容器和用户业务是否正常运行。详细配置方法请参见 6.10.5 设置容器健康检查。
   - 工作负载存活探针：检查容器是否正常，不正常则重启实例。
   - 工作负载业务探针：检查用户业务是否就绪，不就绪则不转发流量到当前实例。
5. **环境变量：**在容器中添加环境变量，一般用于通过环境变量设置参数。

   在“环境变量”页签，单击“添加环境变量”，当前支持三种类型：

- 手动添加：输入变量名称、变量/变量引用。
- 密钥导入：输入变量名称，选择导入的密钥名称和数据。您需要提前创建密钥，具体请参见 11.3 创建密钥。
- 配置项导入：输入变量名称，选择导入的配置项名称和数据。您需要提前创建配置项，具体请参见 11.1 创建配置项。

说明

对于已设置的环境变量，单击环境变量后的“编辑”，可对该环境变量进行编辑。单击环境变量后的“删除”，可删除该环境变量。

6. **数据存储：**给容器挂载数据存储，支持本地磁盘和云存储，适用于需持久化存储、高磁盘 IO 等场景。具体请参见 9.2 本地磁盘存储、、9.5 极速文件存储卷。

说明

有状态工作负载只能在创建时添加数据存储，创建完成后无法再添加。

7. **安全设置：**对容器权限进行设置，保护系统和其他容器不受其影响。

   请输入用户 ID，容器将以当前用户权限运行。

8. **容器日志：**设置容器日志采集策略、配置日志目录。用于收集容器日志便于统一管理和分析。详细配置请参见 6.10.7 采集容器标准输出日志、6.10.8 采集容器内路径日志。

步骤 3 单击“下一步：工作负载访问设置”，设置“实例间发现服务”和工作负载访问方式。

设置“实例间发现服务”，如表 6-7。

表6-7 参数配置

参数	参数说明
服务名称	输入工作负载所对应的服务名称，用于集群内工作负载间的互相访问。该服务主要用于实例的内部发现，不需要有单独的 IP 地址，也不需要做负载均衡。
端口名称	端口名称用于给容器端口命名，通常以端口用途命名。
容器端口	输入容器的监听端口。

单击“添加服务”，设置工作负载访问方式。

若工作负载需要和其它服务互访，或需要被公网访问，您需要添加服务，设置工作负载访问方式。

工作负载访问的方式决定了这个工作负载的网络属性，不同访问方式的工作负载可以提供不同网络能力，具体请参见 8.1 网络概述。

步骤 4 单击“下一步：高级配置”，配置更多高级策略。

- 升级策略：仅支持“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。

- 实例管理策略：支持“有序策略”和“并行策略”两种。

  有序策略：默认策略，有状态负载会逐个的、按顺序的进行部署、删除、伸缩实例， 只有前一个实例部署 Ready 或删除完成后，有状态负载才会操作后一个实例。

  并行策略：支持有状态负载并行创建或删除所有的实例，有状态负载发生变更时立刻在实例上生效。

- **缩容策略**：为工作负载删除提供一个时间窗，预留给生命周期中 PreStop 阶段执行命令。若超过此时间窗，进程仍未停止，该工作负载将被强制删除。
  - 缩容时间窗 (s)：请输入时间，该时间为工作负载停止前命令的执行时间窗（0-9999 秒），默认 30 秒。
  - 缩容优先级：可根据业务需要选择“优先减少新实例”或“优先减少老实例”。
- **调度策略**：您可以根据需要自由组合静态的全局调度策略或动态的运行时调度策略来实现自己的需求。具体请参见 7.1 调度策略概述。
- **Pod 高级设置**
  - Pod 标签：内置 app 标签在工作负载创建时指定，主要用于设置亲和性与反亲和性调度，暂不支持修改。您可以单击下方的“添加标签”增加标签。

图6-3 Pod 高级设置

- **客户端 DNS 配置**：CCE 集群内置 DNS 插件 CoreDNS，为集群内的工作负载提供域名解析服务。详细使用方法请参见 6.10.11 Kubernetes 集群内置 DNS 配置说明。
  - DNS 策略：
    - 追加域名解析配置：选择该配置后，将保留默认配置，以下“IP 地址”和“搜索域”配置可能不生效。
    - 替换域名解析配置：选择该配置后，将仅使用以下“IP 地址”和“搜索域”配置进行域名解析。
    - 继承 Pod 所在节点域名解析配置：将继承 Pod 所在节点的域名解析配置。
  - IP 地址：您可对自定义的域名配置域名服务器，值为一个或一组 DNS IP 地址，如“1.2.3.4”。

- 搜索域：定义域名的搜索域列表，当访问的域名不能被 DNS 解析时，会把该域名与搜索域列表中的域依次进行组合，并重新向 DNS 发起请求，直到域名被正确解析或者尝试完搜索域列表为止。
- 超时时间（s）：查询超时时间，请自定义。
- ndots：表示域名中必须出现的“.”的个数，如果域名中的“.”的个数不小于 ndots，则该域名为一个 FQDN，操作系统会直接查询；如果域名中的“.”的个数小于 ndots，操作系统会在搜索域中进行查询。

- **自定义指标监控：**是指监控系统提供的一种指标收集机制，该机制允许工作负载在部署时自定义需要上报的指标名称以及获取这些指标数据的接入点信息，在应用运行时由监控系统按固定的频率访问接入点进行指标的收集。详细请参见 6.10.9 对接 Prometheus 实现自定义指标监控。
- **性能管理配置：**的性能管理服务可协助您快速进行工作负载的问题定位与性能瓶颈分析。详细请参见 6.10.10 **错误!未找到引用源。**。

步骤 5 配置完成后，单击“创建”，单击“返回工作负载列表”。在工作负载列表中，当工作负载状态为“运行中”时，表示工作负载创建成功。如果工作负载状态未实时更新，请单击右上角的 图标或按 F5 刷新页面查看。

说明

- 节点不可用时，pod 状态变为“未就绪”，此时需要手工删除有状态工作负载的 pod，pod 实例才会迁移到正常节点上。
- 工作负载列表页在超过 500 条以上时，将采用 Kubernetes 的分页机制进行分页。Kubernetes 的分页机制：仅支持回到第一页和查看下一页，不支持查看上一页，且在分页显示的情况下，资源总数显示的是批量查询出的数目而不是真实总数。

----结束

通过 kubectl 命令行创建

本节以 etcd 为例来进行说明。

前提条件

请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令，使弹性云主机连接集群。

操作步骤

步骤 1 登录已配置好 kubectl 命令的弹性云主机。

步骤 2 创建一个名为 etcd-statefulset.yaml 的文件。

其中，etcd-statefulset.yaml 为自定义名称，您可以随意命名。

vi etcd-statefulset.yaml

以下内容仅为示例，若需要了解 statefulset 的详细内容，请参考 kubernetes 官方文档。

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: etcd
```

```
spec:
  replicas: 2
  selector:
    matchLabels:
      app: etcd
  serviceName: etcd-svc
  template:
    metadata:
      labels:
        app: etcd
    spec:
      containers:
      - env:
        - name: PAAS_APP_NAME
          value: tesyhhj
        - name: PAAS_NAMESPACE
          value: default
        - name: PAAS_PROJECT_ID
          value: 9632fae707ce4416a0ab1e3e121fe555
        image: etcd
        imagePullPolicy: IfNotPresent
        name: container-0
  updateStrategy:
    type: RollingUpdate
```

vi etcd-headless.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: etcd
  name: etcd-svc
spec:
  clusterIP: None
  ports:
  - name: etcd-svc
    port: 3120
    protocol: TCP
    targetPort: 3120
  selector:
    app: etcd
  sessionAffinity: None
  type: ClusterIP
```

步骤 3 创建工作负载以及对应 headless 服务。

kubectl create -f etcd-statefulset.yaml

回显如下，表示有状态工作负载（stateful）已创建成功。

```
statefulset.apps/etcd created
```

kubectl create -f etcd-headless.yaml

回显如下，表示对应 headless 服务已创建成功。

```
service/etcd-svc created
```

步骤 4 若工作负载需要被访问（集群内访问或节点访问），您需要设置访问方式，具体请参见 8 网络管理创建对应服务。

----结束

6.4 创建守护进程集(DaemonSet)

操作场景

云容器引擎（CCE）提供多种类型的容器部署和管理能力，支持对容器工作负载的部署、配置、监控、扩容、升级、卸载、服务发现及负载均衡等特性。

其中守护进程集（DaemonSet）可以确保全部（或者某些）节点上仅运行一个 Pod 实例，当有节点加入集群时，也会为他们新增一个 Pod 。 当有节点从集群移除时，这些 Pod 也会被回收。删除 DaemonSet 将会删除它创建的所有 Pod。

使用 DaemonSet 的一些典型用法：

- 运行集群存储 daemon，例如在每个节点上运行 glusterd、ceph。
- 在每个节点上运行日志收集 daemon，例如 fluentd、logstash。
- 在每个节点上运行监控 daemon，例如 Prometheus Node Exporter、collectd、Datadog 代理、New Relic 代理，或 Ganglia gmond。

一种简单的用法是为每种类型的守护进程在所有的节点上都启动一个 DaemonSet。一个稍微复杂的用法是为同一种守护进程部署多个 DaemonSet；每个具有不同的标志，并且对不同硬件类型具有不同的内存、CPU 要求。

前提条件

在创建守护进程集前，您需要存在一个可用集群。若没有可用集群 ，请参照 3.3 购买混合集群中内容创建。

操作步骤

步骤 1 登录 CCE 管理控制台。

步骤 2 在左侧导航栏中选择“工作负载 > 守护进程集 DaemonSet”，单击页面右上角的“创建守护进程集”。参照表 6-8 设置基本信息，其中带“*”标志的参数为必填参数。

表6-8 工作负载基本信息

参数	参数说明
* 工作负载名称	新建工作负载的名称，命名必须唯一。 请输入 4 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（-），并以小写英文字母开头，小写英文字母或数字结尾。
* 集群名称	新建工作负载所在的集群。

参数	参数说明
* 命名空间	在单集群中，不同命名空间中的数据彼此隔离。使应用可以共享同个集群的服务，也能够互不干扰。若您不设置命名空间，系统会默认使用 default 命名空间。
时区同步	单击 开启后，容器将和节点使用相同时区。 须知 时区同步功能开启后，在“数据存储 > 本地磁盘”中，将会自动添加 HostPath 类型的磁盘，请勿修改删除该磁盘。
工作负载描述	工作负载描述信息。

步骤 3 单击“下一步：容器设置”，添加容器。

1. 单击“添加容器”，选择需要部署的镜像。
   - **我的镜像：**展示了您创建的所有镜像仓库。
   - **开源镜像中心：**展示了开源镜像中心仓库中的官方镜像。
   - **第三方镜像：**CCE 支持拉取第三方镜像仓库（即镜像仓库与开源镜像中心之外的镜像仓库）的镜像创建工作负载。使用第三方镜像时，请确保工作负载运行的节点可访问公网。第三方镜像的具体使用方法请参见 6.10.1 如何使用第三方镜像。
     - 若您的镜像仓库不需要认证，密钥认证请选择“否”，并输入“镜像名称”，单击“确定”。
     - 若您的镜像仓库都必须经过认证（帐号密码）才能访问，您需要先创建密钥再使用第三方镜像，具体操作请参见 6.10.1 如何使用第三方镜像。
   - **共享镜像：**其它租户通过“容器镜像服务”共享给您的镜像将在此处展示，您可以基于共享镜像创建工作负载。
2. 配置镜像基本信息。

   工作负载是 Kubernetes 对一组 Pod 的抽象模型，用于描述业务的运行载体，一个 Pod 可以封装 1 个或多个容器，您可以单击右上方的“添加容器”，添加多个容器镜像并分别进行设置。

表6-9 镜像参数说明

参数	说明
镜像名称	导入的镜像，您可单击“更换镜像”进行更换。
* 镜像版本	选择需要部署的镜像版本。
* 容器名称	容器的名称，可修改。
特权容器	特权容器是指容器里面的程序具有一定的特权。 若选中，容器将获得超级权限，例如可以操作宿主机上面的网络设备、修改内核参数等。

参数	说明
容器规格	**CPU 配额：** • 申请：容器需要使用的最小 CPU 值，默认 0.25Core。 • 限制：允许容器使用的 CPU 最大值。建议设容器配额的最高限额，避免容器资源超额导致系统故障。 **内存配额：** • 申请：容器需要使用的内存最小值，默认 512MiB。 • 限制：允许容器使用的内存最大值。如果超过，容器会被终止。 申请和限制的具体请参见 6.10.2 设置容器规格。 **GPU 配额：**当集群中包含 GPU 节点时，才能设置 GPU，无 GPU 节点不显示此选项。 容器需要使用的 GPU 百分比。勾选“使用”并设置百分比，例如设置为 10%，表示该容器需使用 GPU 资源的 10%。若不勾选“使用”，或设置为 0，则无法使用 GPU 资源。 **GPU 显卡：**工作负载实例将被调度到 GPU 显卡类型为指定显卡的节点上。 若勾选“不限制”，容器将会随机使用节点中的任一显卡。您也可以勾选某个显卡，容器将使用特定显卡。

3. **生命周期：**用于设置容器启动和运行时需要执行的命令。
   - 启动命令：设置容器启动时执行的命令，具体请参见 6.10.4 设置容器启动命令。
   - 启动后处理：设置容器成功运行后执行的命令，详细配置方法请参见 6.10.3 设置容器生命周期。
   - 停止前处理：设置容器结束前执行的命令，通常用于删除日志/临时文件等，详细配置方法请参见 6.10.3 设置容器生命周期。
4. **健康检查：**CCE 提供了存活与业务两种探针，用于判断容器和用户业务是否正常运行。详细配置方法请参见 6.10.5 设置容器健康检查。
   - 工作负载存活探针：检查容器是否正常，不正常则重启实例。
   - 工作负载业务探针：检查用户业务是否就绪，不就绪则不转发流量到当前实例。
5. **环境变量：**在容器中添加环境变量，一般用于通过环境变量设置参数。

   在“环境变量”页签，单击“添加环境变量”，当前支持三种类型：
   - 手动添加：输入变量名称、变量/变量引用。
   - 密钥导入：输入变量名称，选择导入的密钥名称和数据。您需要提前创建密钥，具体请参见 11.3 创建密钥。
   - 配置项导入：输入变量名称，选择导入的配置项名称和数据。您需要提前创建配置项，具体请参见 11.1 创建配置项。

说明

对于已设置的环境变量，单击环境变量后的“编辑”，可对该环境变量进行编辑。单击环境变量后的“删除”，可删除该环境变量。

6. **数据存储：**给容器挂载数据存储，支持本地磁盘和云存储，适用于需持久化存储、高磁盘 IO 等场景。具体请参见 9.2 本地磁盘存储、9.5 极速文件存储卷。
7. **安全设置：**对容器权限进行设置，保护系统和其他容器不受其影响。

   请输入用户 ID，容器将以当前用户权限运行。
8. **容器日志：**设置容器日志采集策略、配置日志目录。用于收集容器日志便于统一管理和分析。详细配置请参见 6.10.7 采集容器标准输出日志、6.10.8 采集容器内路径日志。

步骤 4 单击“下一步：工作负载访问设置”，单击“添加服务”，设置工作负载访问方式。

若工作负载需要和其它服务互访，或需要被公网访问，您需要添加服务，设置工作负载访问方式。

工作负载访问的方式决定了这个工作负载的网络属性，不同访问方式的工作负载可以提供不同网络能力，具体请参见 8.1 网络概述。

步骤 5 单击“下一步：高级设置”，配置更多高级策略。

- **升级策略：**
  - 升级方式：支持“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）。
- **缩容策略：**

  缩容时间窗：请输入时间。为工作负载删除提供一个时间窗，预留给生命周期中 PreStop 阶段执行命令。若超过此时间窗，进程仍未停止，该工作负载将被强制删除。
- **调度策略：**您可以根据需要自由组合静态的全局调度策略或动态的运行时调度策略来实现自己的需求。具体请参见 7.1 调度策略概述。
- **Pod 高级设置**
  - Pod 标签：内置 app 标签在工作负载创建时指定，主要用于设置亲和性与反亲和性调度，暂不支持修改。您可以单击下方的“添加标签”增加标签。

图6-4 Pod 高级设置

- **客户端 DNS 配置**：CCE 集群内置 DNS 插件 CoreDNS，为集群内的工作负载提供域名解析服务。详细使用方法请参见 6.10.11 Kubernetes 集群内置 DNS 配置说明。
  - DNS 策略：
    - 追加域名解析配置：选择该配置后，将保留默认配置，以下“IP 地址”和“搜索域”配置可能不生效。
    - 替换域名解析配置：选择该配置后，将仅使用以下“IP 地址”和“搜索域”配置进行域名解析。
    - 继承 Pod 所在节点域名解析配置：将继承 Pod 所在节点的域名解析配置。
  - IP 地址：您可对自定义的域名配置域名服务器，值为一个或一组 DNS IP 地址，如“1.2.3.4”。
  - 搜索域：定义域名的搜索域列表，当访问的域名不能被 DNS 解析时，会把该域名与搜索域列表中的域依次进行组合，并重新向 DNS 发起请求，直到域名被正确解析或者尝试完搜索域列表为止。
  - 超时时间（s）：查询超时时间，请自定义。
  - ndots：表示域名中必须出现的“.”的个数，如果域名中的“.”的个数不小于 ndots，则该域名为一个 FQDN，操作系统会直接查询；如果域名中的“.”的个数小于 ndots，操作系统会在搜索域中进行查询。
- **自定义指标监控**：是指监控系统提供的一种指标收集机制，该机制允许工作负载在部署时自定义需要上报的指标名称以及获取这些指标数据的接入点信息，在应用运行时由监控系统按固定的频率访问接入点进行指标的收集。详细请参见 6.10.9 对接 Prometheus 实现自定义指标监控。
- **性能管理配置**：的性能管理服务可协助您快速进行工作负载的问题定位与性能瓶颈分析。详细请参见 6.10.10 **错误!未找到引用源。**。

步骤 6 配置完成后，单击“创建”，在创建成功页面单击“返回工作负载列表”，查看工作负载状态。

在工作负载列表中，当工作负载状态为“运行中”时，表示工作负载创建成功。

工作负载状态不会实时更新，请单击右上角的 图标或按 F5 刷新页面查看。

----结束

6.5 创建普通任务(Job)

操作场景

普通任务是一次性运行的短任务，部署完成后即可执行。正常退出（exit 0）后，任务即执行完成。

普通任务是用来控制批处理型任务的资源对象。批处理业务与长期伺服业务（Deployment、Statefulset）的主要区别是：

批处理业务的运行有头有尾，而长期伺服业务在用户不停止的情况下永远运行。Job 管理的 Pod 根据用户的设置把任务成功完成就自动退出了。成功完成的标志根据不同的 spec.completions 策略而不同，即：

- 单 Pod 型任务有一个 Pod 成功就标志完成。
- 定数成功型任务保证有 N 个任务全部成功。
- 工作队列型任务根据应用确认的全局成功而标志成功。

前提条件

已创建资源，具体操作请参见 4.2 购买节点。若已有集群和节点资源，无需重复操作。

操作步骤

步骤 1 （可选）普通任务需要基于镜像创建，若选择私有镜像，用户首先需要将镜像上传至镜像仓库。

步骤 2 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 普通任务 Job”，单击“创建普通任务”。

步骤 3 参照表 6-10 配置任务基本信息，其中带“*”标志的参数为必填参数。

表6-10 任务基本信息

参数	参数说明
* 任务名称	新建任务的名称，命名必须唯一。 请输入 4 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（-），并以小写英文字母开头，小写英文字母或数字结尾。
* 容器集群	新建任务所在的集群。
* 命名空间	新建任务所属的命名空间，默认为 default。
* 实例数量	任务的实例数量。任务可以有一个或多个实例，用户可以设置具体实例个数，默认为 1。 每个任务实例都由相同的容器部署而成。设置多个实例主要用于实现高可靠性，当某个实例故障时，任务还能正常运行。

参数	参数说明
任务描述	任务描述信息。

步骤 4 单击“下一步：添加容器”，添加容器和镜像。

1. 单击“选择镜像”，选择需要部署的镜像。
   - 我的镜像：展示了您创建的所有镜像仓库。
   - 开源镜像中心：展示了开源镜像中心仓库中的官方镜像。
   - 第三方镜像：CCE 支持拉取第三方镜像仓库（即镜像仓库与开源镜像中心之外的镜像仓库）的镜像创建任务。使用第三方镜像时，请确保任务运行的节点可访问公网。第三方镜像的具体使用方法请参见 6.10.1 如何使用第三方镜像。
     - 若您的镜像仓库不需要认证，密钥认证请选择“否”，并输入“镜像地址”，单击“确定”。
     - 若您的镜像仓库都必须经过认证（帐号密码）才能访问，您需要先创建密钥再使用第三方镜像，具体操作请参见 6.10.1 如何使用第三方镜像。
   - 共享镜像：其它租户通过“容器镜像服务”共享给您的镜像将在此处展示，您可以基于共享镜像创建工作负载。
2. 配置镜像参数。

表6-11 添加容器参数说明

参数	说明
镜像	导入的镜像，您可单击“更换镜像”进行更换。
* 镜像版本	选择需要部署的镜像版本。
* 容器名称	容器的名称，可修改。
容器规格	**CPU 配额：** • 申请：容器需要使用的最小 CPU 值，默认 0.25Core。 • 限制：允许容器使用的 CPU 最大值。建议设容器配额的最高限额，避免容器资源超额导致系统故障。 **内存配额：** • 申请：容器需要使用的内存最小值，默认 0.5GiB。 • 限制：允许容器使用的内存最大值。如果超过，容器会被终止。 申请和限制的具体请参见 6.10.2 设置容器规格。 **GPU 配额：**当集群中包含 GPU 节点时，才能设置 GPU，无 GPU 节点不显示此选项。 容器需要使用的 GPU 百分比。勾选“使用”并设置百分比，例如设置为 10%，表示该容器需使用 GPU 资源的 10%。若不勾选“使用”，或设置为 0，则无法使用 GPU 资源。

参数	说明
	GPU 显卡：工作负载实例将被调度到 GPU 显卡类型为指定显卡的节点上。 若勾选“不限制”，容器将会随机使用节点中的任一显卡。您也可以勾选某个显卡，容器将使用特定显卡。

3. （可选）高级设置。

表6-12 高级设置

参数	参数说明
生命周期	生命周期脚本定义，主要针对容器类任务的生命周期事件采取的动作。 • 启动命令：输入容器启动命令，容器启动后会立即执行。详细步骤请参见 6.10 容器设置。 • 启动后处理：任务启动后触发。详细步骤请参见 6.10.3 设置容器生命周期。 • 停止前处理：任务停止前触发。详细步骤请参见 6.10.3 设置容器生命周期。
环境变量	在容器中添加环境变量，一般用于通过环境变量设置参数。在环境变量页签，单击“添加环境变量”。当前支持三种类型。 • 手动添加：输入变量名称、变量/变量引用。 • 密钥导入：输入变量名称，选择导入的密钥名称和数据。您需要提前创建密钥，具体请参见 11.3 创建密钥。 • 配置项导入：输入变量名称，选择导入的配置项名称和数据。您需要提前创建配置项，具体请参见 11.1 创建配置项。
数据存储	支持挂载本地磁盘或者云存储到容器中，以实现数据文件的持久化存储。 详细步骤请参见 9 存储管理。
容器日志	CCE 支持配置工作负载日志策略，便于日志收集分析，以及按周期防爆处理。详细步骤请参见 6.10.7 采集容器标准输出日志。

4. （可选）一个任务实例包含 1 个或多个相关容器。若您的任务包含多个容器，请单击“添加容器”，再执行添加容器的操作。

步骤 5 配置完成后，单击“创建”。

待状态为“执行中”，普通任务创建成功。

----结束

使用 kubectl 创建 Job

Job 的配置参数如下所示。

- spec.template 格式与 Pod 相同。
- RestartPolicy 仅支持 Never 或 OnFailure。
- 单个 Pod 时，默认 Pod 成功运行后 Job 即结束。
- .spec.completions 表示 Job 结束需要成功运行的 Pod 个数，默认为 1。
- .spec.parallelism 表示并行运行的 Pod 的个数，默认为 1。
- spec.backoffLimit 表示失败 Pod 的重试最大次数，超过这个次数不会继续重试。
- .spec.activeDeadlineSeconds 表示 Pod 运行时间，一旦达到这个时间，Job 即其所有的 Pod 都会停止。且 activeDeadlineSeconds 优先级高于 backoffLimit，即到达 activeDeadlineSeconds 的 Job 会忽略 backoffLimit 的设置。

根据.spec.completions 和.spec.Parallelism 的设置，可以将 Job 划分为以下几种类型。

表6-13 任务类型

Job 类型	说明	使用示例
一次性 Job	创建一个 Pod 直至其成功结束	数据库迁移
固定结束次数的 Job	依次创建一个 Pod 运行直至 completions 个成功结束	处理工作队列的 Pod
固定结束次数的并行 Job	依次创建多个 Pod 运行直至 completions 个成功结束	多个 Pod 同时处理工作队列
并行 Job	创建一个或多个 Pod 直至有一个成功结束	多个 Pod 同时处理工作队列

以下是一个 Job 配置示例，保存在 myjob.yaml 中，其计算 π 到 2000 位并打印输出。

```
apiVersion: batch/v1
kind: Job
metadata:
  name: pi-with-timeout
spec:
  completions: 50        # Job 结束需要运行 50 个 Pod，这个示例中就是打印 π 50 次
  parallelism: 5        # 并行 5 个 Pod
  backoffLimit: 5        # 最多重试 5 次
  template:
    spec:
      containers:
      - name: pi
        image: perl
        command: ["perl",  "-Mbignum=bpi", "-wle", "print bpi(2000)"]
      restartPolicy: Never
```

说明：

- apiVersion: batch/v1 是当前 job 的 Version
- kind: Job：指定当前资源的类型时 Job
- restartPolicy: Never：是指当前的重启策略。对于 Job，只能设置为 Never 或者 OnFailure。对于其他 controller（比如 Deployment）可以设置为 Always。

运行该任务，如下：

步骤 1 启动这个 job。

```
[root@k8s-master k8s]# kubectl apply -f myjob.yaml
job.batch/myjob created
```

步骤 2 查看这个 job。

kubectl get job

```
[root@k8s-master k8s]# kubectl get job
NAME    COMPLETIONS   DURATION   AGE
myjob   1/1           23s        3m45s
```

completions 为 1/1 表示成功运行了这个 job。

步骤 3 查看 pod 的状态。

kubectl get pod

```
[root@k8s-master k8s]# kubectl get pod
NAME          READY   STATUS      RESTARTS   AGE
myjob-29qlw   0/1     Completed   0          4m5s
```

状态为 Completed 表示这个 job 已经运行完成。

步骤 4 查看这个 pod 的日志。

kubectl logs

```
[root@k8s-master k8s]# kubectl logs myjob-29qlw
hello k8s job!
```

----结束

相关操作

普通任务创建完成后，您还可执行表 6-14 中操作。

表6-14 其他操作

操作	操作说明
查看 YAML	单击任务名称后的“查看 YAML”，可查看到当前任务对应的 YAML 文件。
删除普通任务	1. 选择待删除的任务，单击操作列的“删除”。 2. 单击“确定”。 任务删除后将无法恢复，请谨慎操作。

6.6 创建定时任务(CronJob)

操作场景

定时任务是按照指定时间周期运行的短任务。使用场景为在某个固定时间点，为所有运行中的节点做时间同步。

定时任务是基于时间的 Job，就类似于 Linux 系统的 crontab，在指定的时间周期运行指定的 Job，即：

- 在给定时间点只运行一次。
- 在给定时间点周期性地运行。

CronJob 的典型用法如下所示：

- 在给定的时间点调度 Job 运行。
- 创建周期性运行的 Job，例如数据库备份、发送邮件。

前提条件

已创建资源，具体操作请参见 4.2 购买节点。若已有集群和节点资源，无需重复操作。

操作步骤

步骤 1 （可选）定时任务需要基于镜像创建，若选择私有镜像，用户首先需要将镜像上传至镜像仓库。

步骤 2 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 定时任务 CronJob”，单击“创建定时任务”。

步骤 3 参照表 6-15 配置定时任务基本信息，其中带“*”标志的参数为必填参数。

表6-15 任务基本信息

参数	参数说明
* 任务名称	新建任务的名称，命名必须唯一。 请输入 4 到 52 个字符的字符串，可以包含小写英文字母、数字和中划线（-），并以小写英文字母开头，小写英文字母或数字结尾。
* 容器集群	新建任务所在的集群。
* 命名空间	新建任务所在的命名空间。若不选择，默认配置为 default。
任务描述	任务描述信息。

步骤 4 配置完成后，单击“下一步：定时规则”。

步骤 5 设置定时规则。

表6-16 定时规则参数

参数	参数说明
* 并发策略	支持如下三种模式： • Forbid：在前一个任务未完成时，不创建新任务。 • Allow：定时任务不断新建 Job，会抢占集群资源。 • Replace：已到新任务创建时间点，但前一个任务还未完成，新的任务会取代前一个任务。
* 定时规则	指定新建定时任务在何时执行。
任务记录	可以设置保留执行成功或执行失败的任务个数，设置为 0 表示不保留。

步骤 6 单击“下一步：添加容器”，添加容器。

1. 单击“选择镜像”，选择需要部署的镜像。
   - 我的镜像：展示了您创建的所有镜像仓库。
   - 开源镜像中心：展示了开源镜像中心仓库中的官方镜像。
   - 第三方镜像：CCE 支持拉取第三方镜像仓库（即镜像仓库与开源镜像中心之外的镜像仓库）的镜像创建任务。使用第三方镜像时，请确保任务运行的节点可访问公网。第三方镜像的具体使用方法请参见 6.10.1 如何使用第三方镜像。
     - 若您的镜像仓库不需要认证，密钥认证请选择“否”，并输入“镜像地址”，单击“确定”。
     - 若您的镜像仓库都必须经过认证（帐号密码）才能访问，您需要先创建密钥再使用第三方镜像，具体操作请参见 6.10.1 如何使用第三方镜像。
   - 共享镜像：其它租户通过“容器镜像服务”共享给您的镜像将在此处展示，您可以基于共享镜像创建工作负载。
2. 配置镜像参数。

表6-17 添加容器参数说明

参数	说明
镜像	导入的镜像，您可单击“更换镜像”进行更换。
* 镜像版本	选择需要部署的镜像版本。
* 容器名称	容器的名称，可修改。
容器规格	**CPU 配额：** • 申请：容器需要使用的最小 CPU 值，默认 0.25Core。

参数	说明
	• 限制：允许容器使用的 CPU 最大值。建议设容器配额的最高限额，避免容器资源超额导致系统故障。 **内存配额：** • 申请：容器需要使用的内存最小值，默认 0.5GiB。 • 限制：允许容器使用的内存最大值。如果超过，容器会被终止。 申请和限制的具体请参见 6.10.2 设置容器规格。 **GPU 配额：**当集群中包含 GPU 节点时，才能设置 GPU，无 GPU 节点不显示此选项。 容器需要使用的 GPU 百分比。勾选“使用”并设置百分比，例如设置为 10%，表示该容器需使用 GPU 资源的 10%。若不勾选“使用”，或设置为 0，则无法使用 GPU 资源。 **GPU 显卡：**工作负载实例将被调度到 GPU 显卡类型为指定显卡的节点上。 若勾选“不限制”，容器将会随机使用节点中的任一显卡。您也可以勾选某个显卡，容器将使用特定显卡。

3. （可选）高级设置。

表6-18 高级设置

参数	参数说明
生命周期	生命周期脚本定义，主要针对容器类任务的生命周期事件工作负载采取的动作。 • 启动命令：输入容器启动命令，容器启动后会立即执行。详细步骤请参见 6.10 容器设置。 • 启动后处理：任务启动后触发。详细步骤请参见 6.10.3 设置容器生命周期。 • 停止前处理：任务停止前触发。详细步骤请参见 6.10.3 设置容器生命周期。
环境变量	在容器中添加环境变量，一般用于通过环境变量设置参数。在“环境变量”页签，单击“添加环境变量”。当前支持三种类型。 • 手动添加：输入变量名称、变量/变量引用。 • 密钥导入：输入变量名称，选择导入的密钥名称和数据。您需要提前创建密钥，具体请参见 11.3 创建密钥。 • 配置项导入：输入变量名称，选择导入的配置项名称和数据。您需要提前创建配置项，具体请参见 11.1 创建配置项。

4. （可选）一个任务实例包含 1 个或多个相关容器。若您的任务包含多个容器，请单击“添加容器”，再执行添加容器的操作。

步骤 7 设置完成后，单击“创建”。

待状态为“已启动”，定时任务创建成功。

----结束

使用 kubectl 创建 CronJob

CronJob 的配置参数如下所示：

- .spec.schedule 指定任务运行时间与周期，格式与 Cron 相同，例如“0 * * * * ”或“@hourly”。
- .spec.jobTemplate 指定需要运行的任务，格式与使用 kubectl 创建 Job 相同。
- .spec.startingDeadlineSeconds 指定任务开始的截止期限。
- .spec.concurrencyPolicy 指定任务的并发策略，支持 Allow、Forbid 和 Replace 三个选项。
  - Allow（默认）：允许并发运行 Job。
  - Forbid：禁止并发运行，如果前一个还没有完成，则直接跳过下一个。
  - Replace：取消当前正在运行的 Job，用一个新的来替换。

下面是一个 CronJob 的示例，保存在 cronjob.yaml 文件中。

```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: hello
spec:
  schedule: "*/1 * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
          - name: hello
            image: busybox
            args:
            - /bin/sh
            - -c
            - date; echo Hello from the Kubernetes cluster
          restartPolicy: OnFailure
```

运行该任务，如下：

步骤 1 创建 CronJob。

kubectl create -f cronjob.yaml

命令行终端显示如下信息：

```
cronjob.batch/hello created
```

步骤 2 执行如下命令，查看执行情况。

kubectl get cronjob

```
NAME      SCHEDULE      SUSPEND   ACTIVE    LAST SCHEDULE   AGE
hello     */1 * * * *   False     0         <none>          9s
```

kubectl get jobs

```
NAME              COMPLETIONS   DURATION   AGE
hello-1597387980   1/1           27s        45s
```

kubectl get pod

```
NAME                           READY     STATUS      RESTARTS   AGE
hello-1597387980-tjv8f         0/1       Completed   0          114s
hello-1597388040-lckg9         0/1       Completed   0          39s
```

kubectl logs hello-1597387980-tjv8f

```
Fri Aug 14 06:56:31 UTC 2020
Hello from the Kubernetes cluster
```

kubectl delete cronjob hello

```
cronjob.batch "hello" deleted
```

须知

删除 CronJob 时，对应的普通任务及相关的 Pod 都会被删除。

----结束

相关操作

定时任务创建完成后，您还可执行表 6-19 中操作。

表6-19 其他操作

操作	操作说明
编辑 YAML	单击定时任务名称后的“编辑 YAML”，可修改当前任务对应的 YAML 文件。
停止定时任务	1. 选择待停止的任务，单击操作列的“停止”。 2. 单击“确定”。
删除定时任务	1. 选择待删除的任务，单击操作列的“删除”。 2. 单击“确定”。 任务删除后将无法恢复，请谨慎操作。

6.7 管理容器组(Pod)

操作场景

容器组（Pod）是 Kubernetes 中最小的可部署单元。一个 Pod（容器组）包含了一个应用程序容器（某些情况下是多个容器）、存储资源、一个唯一的网络 IP 地址、以及一些确定容器该如何运行的选项。Pod 容器组代表了 Kubernetes 中一个独立的应用程序运行实例，该实例可能由单个容器或者几个紧耦合在一起的容器组成。

Kubernetes 集群中的 Pod 存在如下两种使用途径：

- 一个 Pod 中只运行一个容器。"one-container-per-pod" 是 Kubernetes 中最常见的使用方式。此时，您可以认为 Pod 容器组是该容器的 wrapper，Kubernetes 通过 Pod 管理容器，而不是直接管理容器。
- 一个 Pod 中运行多个需要互相协作的容器。您可以将多个紧密耦合、共享资源且始终在一起运行的容器编排在同一个 Pod 中，可能的情况有：
  - Content management systems, file and data loaders, local cache managers 等
  - log and checkpoint backup, compression, rotation, snapshotting 等
  - data change watchers, log tailers, logging and monitoring adapters, event publishers 等
  - proxies, bridges, adapters 等
  - controllers, managers, configurators, and updaters

您可以在云容器引擎 CCE 中方便的管理容器组（Pod），如编辑 YAML、监控、删除等操作。

编辑 YAML

可通过在线 YAML 编辑窗对容器组（Pod）的 YAML 文件进行修改和下载。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 容器组 Pod”。

步骤 2 单击实例列表中后的“编辑 YAML”，在弹出的“编辑 YAML”窗中可对当前容器组（Pod）的 YAML 文件进行修改。

步骤 3 单击“修改”，在弹出的提示框中单击“确定”，完成修改。

说明

如果实例由其他工作负载创建，暂不支持在容器组（Pod）中单独修改。

步骤 4 （可选）在“编辑 YAML”窗中，单击“下载”，可下载该 YAML 文件。

----结束

实例监控

您可以通过 CCE 控制台查看工作负载实例的 CPU、内存使用情况以及网络上行和下行速率、磁盘的读写速率，以确定需要的资源规格。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 容器组 Pod”。

步骤 2 单击实例列表中后的“监控”，可查看相应实例的 CPU 使用率、内存使用率以及网络上行和下行速率、磁盘的读写速率。

说明

实例处于非运行状态时，无法查看实例的监控数据。

----结束

删除实例

若实例无需再使用，您可以将其删除。实例删除后，将无法恢复，请谨慎操作。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 容器组 Pod”。

步骤 2 单击待删除实例后的“删除”。

请仔细阅读系统提示，删除操作无法恢复，请谨慎操作。

步骤 3 单击“是”，即可删除该实例。

说明

- 若 Pod 所在节点不可用或者关机，负载无法删除时可以在详情页面实例列表选择强制删除。
- 请确保要删除的存储没有被其他负载使用，导入和存在快照的存储只做解关联操作。

----结束

相关链接

- The Distributed System Toolkit: Patterns for Composite Containers
- Container Design Patterns

6.8 管理工作负载和任务

操作场景

工作负载创建后，您可以对其执行伸缩、升级、编辑 YAML、日志、监控、回退、删除等操作。

日志

您可以通过“日志”功能查看无状态工作负载、有状态工作负载、守护进程集、普通任务的日志信息。本文以无状态工作负载为例说明如何查看日志。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”。

步骤 2 单击工作负载后的“日志”。

在弹出的“日志”窗口中可以查看最近 5 分钟、最近 30 分钟、最近 1 小时内的日志信息。

说明

云容器引擎服务对接了应用运维管理服务 AOM 提供日志查看、检索功能。2019 年 10 月 10 日对应用运维管理服务进行价格调整，每月赠送租户 500M 免费日志采集额度，超过免费额度部分将产生费用。

----结束

升级

基于 CCE，您可以通过更换镜像或镜像版本实现无状态工作负载、有状态工作负载、守护进程集的快速升级，业务无中断。本文以无状态工作负载为例说明如何进行升级。

若需要更换镜像或镜像版本，您需要提前将镜像上传到容器镜像服务。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”，单击待升级工作负载后的“升级”。

说明

有状态工作负载升级时，若升级类型为替换升级，需要用户手动删除实例后才能升级成功，否则界面会始终显示“升级中”。

步骤 2 请根据业务需求进行工作负载的升级。

- **镜像名称**：单击“更换镜像”，选择新的镜像。
- **镜像版本**：在镜像版本后的下拉框中选择对应版本。
- **容器名称**：单击容器名称后的 ，修改容器名称。
- **特权容器**：开启后，容器将可以访问主机上的所有设备。
- **容器规格**：可分别设置 CPU 配额、内存配额、GPU 配额和昇腾 310 配额。
- **高级设置**：
  - **生命周期**：用于设置容器启动和运行时需要执行的命令。
    - 启动命令：设置容器启动时执行的命令，具体请参见 6.10.4 设置容器启动命令。
    - 启动后处理：设置容器成功运行后执行的命令，详细配置方法请参见 6.10.3 设置容器生命周期。
    - 停止前处理：设置容器结束前执行的命令，通常用于删除日志/临时文件等，详细配置方法请参见 6.10.3 设置容器生命周期。
  - **健康检查**：CCE 提供了存活与业务两种探针，用于判断容器和用户业务是否正常运行。详细配置方法请参见 6.10.5 设置容器健康检查。
    - 工作负载存活探针：检查容器是否正常，不正常则重启实例。
    - 工作负载业务探针：检查用户业务是否就绪，不就绪则不转发流量到当前实例。
  - **环境变量**：在容器中添加环境变量，一般用于通过环境变量设置参数。

    在“环境变量”页签，单击“添加环境变量”，当前支持三种类型：

- 手动添加：输入变量名称、变量/变量引用。
- 密钥导入：输入变量名称，选择导入的密钥名称和数据。您需要提前创建密钥，具体请参见 11.3 创建密钥。
- 配置项导入：输入变量名称，选择导入的配置项名称和数据。您需要提前创建配置项，具体请参见 11.1 创建配置项。

说明

对于已设置的环境变量，单击环境变量后的“编辑”，可对该环境变量进行编辑。单击环境变量后的“删除”，可删除该环境变量。

- **数据存储：**给容器挂载数据存储，支持本地磁盘和云存储，适用于需持久化存储、高磁盘 IO 等场景。具体请参见 9.2 本地磁盘存储、9.5 极速文件存储卷。

说明

有状态工作负载只能在创建时添加数据存储，创建完成后无法再添加。

- **安全设置：**对容器权限进行设置，保护系统和其他容器不受其影响。

  请输入用户 ID，容器将以当前用户权限运行。
- **容器日志：**设置容器日志采集策略、配置日志目录。用于收集容器日志便于统一管理和分析。详细配置请参见 6.10.7 采集容器标准输出日志、6.10.8 采集容器内路径日志。

步骤 3 更新完成后，单击“提交”。

----结束

编辑 YAML

可通过在线 YAML 编辑窗对无状态工作负载、有状态工作负载、守护进程集和容器组的 YAML 文件进行修改和下载。普通任务和定时任务的 YAML 文件仅支持查看、复制和下载。本文以无状态工作负载为例说明如何在线编辑 YAML。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”。

步骤 2 单击工作负载后的“更多 > 编辑 YAML”，在弹出的“编辑 YAML”窗中可对当前工作负载的 YAML 文件进行修改。

步骤 3 单击“修改”，在弹出的提示框中单击“确定”，完成修改。

步骤 4 （可选）在“编辑 YAML”窗中，单击“下载”，可下载该 YAML 文件。

----结束

伸缩

您可以根据业务需求自行定义无状态工作负载、有状态工作负载和守护进程集的伸缩策略，降低因应对业务变化和高峰压力而人为反复调整资源的工作量，帮助您节约资源和人力成本。本文以无状态工作负载为例说明如何使用伸缩功能。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”。

步骤 2 单击待设置伸缩策略工作负载后的“更多 > 伸缩”，进入工作负载详情页面。

步骤 3 在“伸缩”页签，可设置“弹性伸缩”和“手动伸缩”策略，或“编辑伸缩规则”。

详细设置方法请参见 6.9 工作负载弹性伸缩。

----结束

监控

您可以通过 CCE 控制台查看无状态工作负载、守护进程集和容器组的 CPU 和内存占用情况，以确定需要的资源规格。本文以无状态工作负载为例说明如何使用监控功能。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”。

步骤 2 单击已创建工作负载的名称，进入工作负载详情页面。在监控页签，可查看工作负载的 CPU 利用率和物理内存使用率。

步骤 3 单击“实例列表”，单击某个实例名称前的 ，单击“监控”。

步骤 4 查看相应实例的 CPU 使用率、内存使用率。

- CPU 使用率。

  横坐标表示时间，纵坐标表示 CPU 使用率。绿色线条表示 CPU 使用率，红色线条表示 CPU 使用限额。

  说明

  CPU 使用量需要计算，故初次显示时，CPU 使用量会比内存使用量晚一分钟左右显示。

  只有实例处于运行状态时，才能查看 CPU 使用量。

- 物理内存使用率。

  横坐标表示时间，纵坐标表示内存使用量。绿色线条表示内存使用率，红色线条表示内存使用限额。

  说明

  实例处于非运行状态时，无法查看内存使用量。

----结束

回退（仅无状态工作负载可用）

所有无状态工作负载的发布历史记录都保留在系统中，您可以回退到指定的版本。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”。

步骤 2 单击待回退工作负载后的“更多 > 回退”。

步骤 3 在“回退到该版本”后，选择回退版本，单击“确定”。

----结束

暂停（仅无状态工作负载可用）

无状态工作负载可以进行暂停操作。暂停后，对负载进行的升级操作可以正常下发，但不会被应用到实例。

如果您正在滚动升级的过程中，滚动升级会在暂停命令下发后停止，出现新旧实例共存的状态。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”。

步骤 2 单击待暂停的工作负载后方操作栏中的“更多 > 暂停”。

步骤 3 在弹出的负载暂停信息提示框中，单击“确认”。

步骤 4 单击“确定”，可完成暂停操作。

须知

工作负载状态为暂停中时无法执行回退操作。

----结束

恢复（仅无状态工作负载可用）

暂停中的负载可以进行恢复操作。恢复后，负载可以正常升级和回退，负载下的实例会与负载当前的最新信息进行一次同步，如果有不一致的，则会自动按照负载的最新信息进行升级。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”。

步骤 2 单击待恢复的工作负载后方操作栏中的“更多 > 恢复”。

步骤 3 在弹出的负载恢复信息提示框中，单击“确认”。

----结束

标签管理

标签是以 key/value 键值对的形式附加在工作负载上的。添加标签后，可通过标签对工作负载进行管理和选择，主要用于设置亲和性与反亲和性调度。您可以给多个工作负载打标签，也可以给指定的某个工作负载打标签。

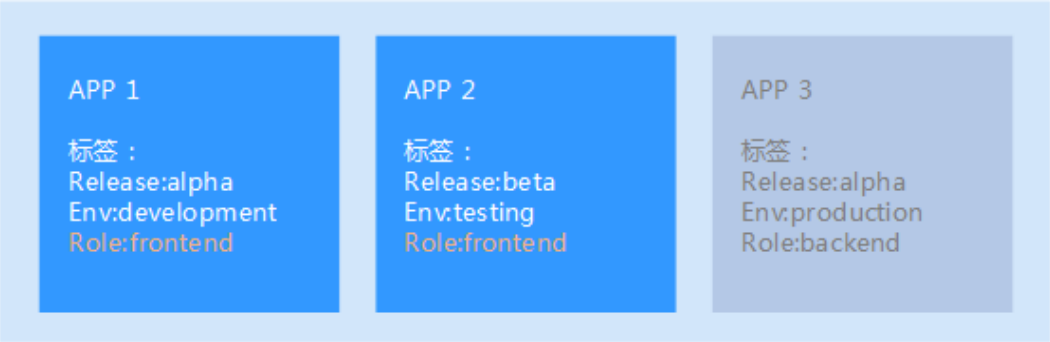
您可以根据业务需求对无状态工作负载、有状态工作负载和守护进程集的标签进行管理，本文以无状态工作负载为例说明如何使用标签管理功能。

如下图，假设为工作负载（例如名称为 APP1、APP2、APP3）定义了 3 个标签：release、env、role。不同工作负载定义了不同的取值，分别为：

- APP 1：[release:alpha;env:development;role:frontend]
- APP 2：[release:beta;env:testing;role:frontend]
- APP 3：[release:alpha;env:production;role:backend]

在使用调度或其他功能时，选择“key/value”值分别为“role/frontend”的工作负载，则会选择到“APP1 和 APP2”。

图6-5 标签案例



步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”。

步骤 2 单击待添加标签的工作负载，进入工作负载详情页面。

步骤 3 单击“标签管理”，单击“添加标签”，输入键和值，单击“确定”。

说明

标签格式要求如下：以字母和数字开头或结尾，由字母、数字、连接符（-）、下划线（_）、点号（.）组成且 63 字符以内。

----结束

删除工作负载/任务

若工作负载无需再使用，您可以将工作负载或任务删除。工作负载或任务删除后，将无法恢复，请谨慎操作。本文以无状态工作负载为例说明如何使用删除功能。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”。

步骤 2 单击待删除工作负载后的“更多 > 删除”，删除工作负载。

请仔细阅读系统提示，删除操作无法恢复，请谨慎操作。

步骤 3 单击“是”。

说明

- 若 Pod 所在节点不可用或者关机，负载无法删除时可以在详情页面实例列表选择强制删除。
- 请确保要删除的存储没有被其他负载使用，导入和存在快照的存储只做解关联操作。

----结束

6.9 工作负载弹性伸缩

伸缩策略定义后，可基于资源变化、固定时间、固定周期自动触发实例的增减，从而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。

- 弹性伸缩：支持告警、定时、周期三种策略。配置完成后可基于资源变化、固定时间、固定周期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。

说明

伸缩策略优先级：手动伸缩和自动伸缩同时配置的情况下，在不执行手动伸缩时，资源调度以自动伸缩为准，伸缩按照监控信息自动触发，如果使用手动伸缩，自动伸缩会暂时失效。

弹性伸缩-HPA/CustomedHPA

该弹性伸缩能力通过创建 HPA 或 CustomedHPA 策略实现，**您可以前往弹性伸缩查看所有策略，或进行更多操作。**

弹性伸缩-AOM

您可以根据业务需求自行定义伸缩策略，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。

该伸缩能力由应用运维管理 AOM 服务提供，v1.17 版本的集群不支持该伸缩能力。

当前支持三种弹性伸缩策略：

告警策略：支持根据 CPU/内存的设置，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。

定时策略：支持在特定时间点进行工作负载的自动伸缩。适用于秒杀周年庆等活动，例如在秒杀这个时间点增加一定数量的实例个数。

周期策略：支持以天、周、月为周期的伸缩策略。适用于周期性的流量变化。

- **告警策略**：支持根据 CPU/内存的设置，进行工作负载的自动伸缩。
  a. 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment/有状态负载 StatefulSet”，单击工作负载名称后的“更多 > 伸缩”。
  b. 在弹性伸缩下，单击“添加伸缩策略”。
  c. 参照表 6-20 设置伸缩策略参数：

表6-20 添加告警策略

参数	参数说明
策略名称	请输入伸缩策略的名称。 伸缩策略名称必须符合 1-64 字符，只能用字母、数字、下划线、中划线的组合，必须以字母开头。

参数	参数说明
策略类型	选择“告警策略”。 告警策略基于历史数据进行判断触发，以 **1 分钟**为周期去判断监控窗口时间内用户设定的指标是否满足触发条件，若连续 n 个周期都满足触发条件，将会执行操作。
指标	指标是对资源性能的数据描述或状态描述。 • CPU 使用率：该指标用于统计测量对象的 CPU 使用率。工作负载实际使用的与申请的 CPU 核数量比率。 • 数据发送速率：该指标用于统计测量对象每秒钟发送的数据量。 • 磁盘写入速率：该指标用于统计每秒写入磁盘的数据量。 • 物理内存使用率：该指标用于统计测量对象已使用内存占申请物理内存总量的百分比。 • 物理内存总量：该指标用于统计测量对象申请的物理内存总量。 • 数据接收速率：该指标用于统计测量对象每秒钟接收的数据量。 • CPU 内核总量：该指标用于统计测量对象申请的 CPU 核总量。 • 物理内存使用量：该指标用于统计测量对象实际已经使用的物理内存（Resident Set Size）。 • 磁盘读取速率：该指标用于统计每秒从磁盘读出的数据量。 • CPU 内核占用：该指标用于统计测量对象已经使用的 CPU 核个数。 • 容器错包个数：该指标用于统计测量对象收到错误包的数量。
触发条件	可选“>”或“<”，支持以上指标的使用量达到设定值时触发伸缩策略。 如：指标选择“CPU 使用率”，此处选择“>”70%，表示在 CPU 使用率超过 70%时触发伸缩策略。
监控窗口	指数据的汇聚窗口大小。 若设置为 60 秒，表示每 60 秒统计一次。
连续周期	指监控窗口内连续触发阈值的次数，计算周期固定一分钟。 若设置为 3，则表示指标数据连续三个统计周期达到了设定的阈值，则触发策略动作。
执行操作	策略触发后执行的动作，可增加或减少实例个数。

d. 单击“确定”。

e. 在弹性伸缩下，可看到策略已启动。

图6-6 查看告警策略

待到触发条件发生时，弹性伸缩策略会自动启动。

- **定时策略**：支持在特定时间点进行工作负载的自动伸缩。
  a. 在弹性伸缩下，单击“添加伸缩策略”，选择“定时策略”。

表6-21 添加定时策略

参数	参数说明
策略名称	请输入伸缩策略的名称。 伸缩策略名称必须符合 1-64 字符，只能用字母、数字、下划线、中划线的组合，必须以字母开头。
策略类型	选择定时策略。
触发时间	策略触发时间。
执行操作	策略触发后执行的动作，可增加、减少或设置实例个数。

  b. 单击“确定”。
  c. 在弹性伸缩下，可看到策略已启动。

图6-7 查看定时策略

待到触发时间时，可查看到实例列表页签中，弹性伸缩已生效。

- **周期策略**：支持以天、周、月为周期的伸缩策略。
  a. 在弹性伸缩下，单击“添加伸缩策略”，选择“周期策略”

表6-22 添加周期策略

参数	参数说明
策略名称	请输入伸缩策略的名称。 伸缩策略名称必须符合 1-64 字符，只能用字母、数字、下划线、中划线的组合，必须以字母开头。
策略类型	选择周期策略。
选择时间	选择策略触发的时间。

参数	参数说明
执行操作	策略触发后执行的动作，可增加、减少或设置实例个数。

b. 单击“确定”。

c. 在弹性伸缩下，可看到策略已启动。

图6-8 查看周期策略

待触发条件生效时，弹性伸缩策略会自动启动。

手动伸缩

步骤 1 登录，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment/有状态负载 StatefulSet”，单击工作负载名称后的“伸缩”。

步骤 2 在“手动伸缩”策略下，单击 修改实例数量，例如修改为“3”，单击“保存”后实例伸缩操作即可生效。

步骤 3 在“实例列表”页签下，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6-9 手动伸缩

----结束

6.10 容器设置

6.10.1 如何使用第三方镜像

操作场景

CCE 支持拉取第三方镜像仓库的镜像来创建工作负载。

通常第三方镜像仓库必须经过认证（帐号密码）才能访问，而 CCE 中容器拉取镜像是使用密钥认证方式，这就要求在拉取镜像前先创建镜像仓库的密钥。

前提条件

使用第三方镜像时，请确保工作负载运行的节点可访问公网。您可以通过 8.2.3 负载均衡(LoadBalancer)方式访问公网。

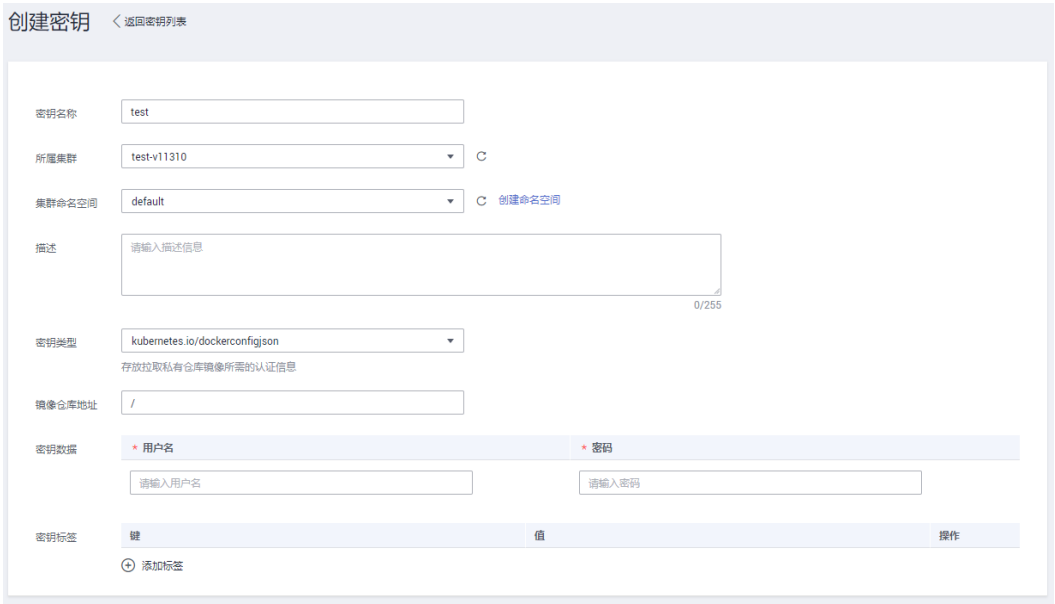
通过界面操作

步骤 1 创建第三方镜像仓库的密钥。

单击左侧导航栏的“配置中心 > 密钥 Secret”，单击“添加密钥”，密钥类型必须选择为 kubernetes.io/dockerconfigjson，如下图所示。详细操作请参见 11.3 创建密钥。

此处的“用户名”和“密码”请填写第三方镜像仓库的帐号密码。

图6-10 添加密钥



步骤 2 参照 6.2 创建无状态负载(Deployment)或 6.3 创建有状态负载(StatefulSet)，选择第三方镜像时，请执行如下操作。

1. 密钥认证：是。
2. 选择密钥：选择步骤 1 中创建的密钥。
3. 镜像地址：输入镜像地址。

步骤 3 单击“创建”。

----结束

使用 kubectl 创建第三方镜像仓库的密钥

步骤 1 请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令。

步骤 2 登录已配置好 kubectl 命令的弹性云主机。

步骤 3 通过 kubectl 创建认证密钥 ，该密钥类型为 dockercfg 类型。

```
kubectl create secret docker-registry myregistrykey --docker-
server=DOCKER_REGISTRY_SERVER --docker-username=DOCKER_USER --docker-
password=DOCKER_PASSWORD --docker-email=DOCKER_EMAIL
```

其中，myregistrykey 为密钥名称，其余参数如下所示。

- DOCKER_REGISTRY_SERVER：第三方镜像仓库的地址，如“www.3rdregistry.com”或“10.10.10.10:443”。
- DOCKER_USER：第三方镜像仓库的帐号。
- DOCKER_PASSWORD：第三方镜像仓库的密码。
- DOCKER_EMAIL：第三方镜像仓库的邮箱。

步骤 4 创建工作负载时使用第三方镜像，具体步骤请参见如下。

dockecfg 类型的密钥作为私有镜像获取的认证方式，以 Pod 为例，创建的 myregistrykey 作为镜像的认证方式。

```
apiVersion: v1
kind: Pod
metadata:
  name: foo
  namespace: default
spec:
  containers:
    - name: foo
      image: www.3rdregistry.com/janedoe/awesomeapp:v1
  imagePullSecrets:
    - name: myregistrykey              #使用上面创建的密钥
```

----结束

6.10.2 设置容器规格

操作场景

CCE 支持在创建工作负载时为添加的容器设置资源限制。可以对工作负载中每个实例所用的 CPU 配额、内存配额进行申请和限制，对每个实例所用的 GPU 和昇腾 310 配额设置使用或不使用。

配置含义

在 CPU 配额和内存配额设置中，**申请**与**限制**的含义如下：

- 勾选“申请”表示启动该配置，系统根据申请值调度该实例到满足条件的节点去部署工作负载。

- 不勾选“申请”表示系统调度实例到随机的一个节点去部署工作负载。
- 勾选“限制”表示启动该配置，根据设定的值，限制工作负载使用的资源。
- 不勾选“限制”表示实例使用的资源不做限制，但若实例使用的内存资源超过节点可分配内存时，可能会导致工作负载不可用或者节点不可用。

说明

创建工作负载时，建议设置 CPU 和内存的资源上下限。同一个节点上部署的工作负载，对于未设置资源上下限的工作负载，如果其异常资源泄露会导致其它工作负载分配不到资源而异常。未设置资源上下限的工作负载，工作负载监控信息也会不准确。

在 GPU 配额和昇腾 310 配额设置中，**使用**与**不限制**的含义如下：

- 勾选“使用”表示启动该配置，系统根据设定的值调度该实例到满足条件的节点去部署工作负载。
- “不限制”默认选中，不可取消。表示该项对实例使用的资源不做限制。

配置说明

- CPU 配额：

表6-23 CPU 配额说明

参数	说明
CPU 申请	容器使用的最小 CPU 需求，作为容器调度时资源分配的判断依赖。只有当节点上可分配 CPU 总量 ≥ 容器 CPU 申请数时，才允许将容器调度到该节点。
CPU 限制	容器能使用的 CPU 最大值。

建议配置方法：

节点的实际可用分配 CPU 量 ≥ 当前实例所有容器 CPU 限制值之和 ≥ 当前实例所有容器 CPU 申请值之和，节点的实际可用分配 CPU 量请在“资源管理 > 节点管理”中对应节点的“可分配资源”列下查看“CPU: ** Core”。

- 内存配额：

表6-24 内存配额说明

参数	说明
内存申请	容器使用的最小内存需求，作为容器调度时资源分配的判断依赖。只有当节点上可分配内存总量 ≥ 容器内存申请数时，才允许将容器调度到该节点。
内存限制	容器能使用的内存最大值。当内存使用率超出设置的内存限制值时，该实例可能会被重启进而影响工作负载的正常使用。

建议配置方法：

节点的实际可用分配内存量 ≥ 当前节点所有容器内存限制值之和 ≥ 当前节点所有容器内存申请值之和，节点的实际可用分配内存量请在“资源管理 > 节点管理”中对应节点的“可分配资源”列下查看“内存: ** GiB”。

说明

可分配资源：可分配量按照实例请求值(request)计算，表示实例在该节点上可请求的资源上限，不代表节点实际可用资源。 计算公式为：

- 可分配 CPU = CPU 总量 - 所有实例的 CPU 请求值 - 其他资源 CPU 预留值
- 可分配内存 = 内存总量 - 所有实例的内存请求值 - 其他资源内存预留值

使用示例

以集群包含一个资源为 4Core 8GB 的节点为例，已经部署一个包含两个实例的工作负载到该集群上，并设置两个实例（实例 1，实例 2）的资源为{CPU 申请，CPU 限制，内存申请，内存限制}={1Core，2Core，2GB，2GB}。

那么节点上 CPU 和内存的资源使用情况如下：

- 节点 CPU 可分配量=4Core-（实例 1 申请的 1Core+实例 2 申请的 1Core）=2Core
- 节点内存可分配量=8GB-（实例 1 申请的 2GB+实例 2 申请的 2GB）=4GB

因此节点还剩余 2Core 4GB 的资源可供下一个新增的实例使用。

6.10.3 设置容器生命周期

操作场景

CCE 提供了回调函数，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回调函数如下所示：

- **启动命令：**容器将会以该启动命令启动，请参见 6.10.4 设置容器启动命令。
- **启动后处理：**容器启动后触发，请参见启动后处理。
- **停止前处理：**容器停止前触发。设置停止前处理，确保升级或实例删除时可提前将实例中运行的业务排水。详细请参见停止前处理。

容器如何执行命令和参数

Docker 的镜像拥有存储镜像信息的相关元数据，如果不设置生命周期命令和参数，容器运行时将运行镜像制作时提供的默认的命令和参数，Docker 将这两个字段定义为“Entrypoint”和 "CMD"。

如果在创建工作负载时填写了容器的运行命令和参数，将会覆盖镜像构建时的默认命令 "Entrypoint"、"CMD"，规则如下：

表6-25 容器如何执行命令和参数

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]
[touch]	[/root/test]	未设置	[/opt/test]	[touch /opt/test]
[touch]	[/root/test]	[mkdir]	[/opt/test]	[mkdir /opt/test]

启动命令

在默认情况下，镜像启动时会运行默认命令，如果想运行特定命令或重写镜像默认值，需要进行相应设置。设置方法请参见 6.10.4 设置容器启动命令。

启动后处理

步骤 1 登录 CCE 控制台，在创建工作负载时，展开“生命周期”。

步骤 2 在“启动后处理”后，设置启动后处理的参数，如表 6-26。

表6-26 启动后处理-参数说明

参数	说明

命令行方式	在容器中执行指定的命令，配置为需要执行的命令。命令的格式为 Command Args[1] Args[2]…（Command 为系统命令或者用户自定义可执行程序，如果未指定路径则在默认路径下寻找可执行程序），如果需要执行多条命令，建议采用将命令写入脚本执行的方式。
	如需要执行的命令如下：
	`exec:`
	`  command:`
	`  - /install.sh`
	`  - install_agent`
	请在执行脚本中填写: /install install_agent。这条命令表示容器创建成功后将执行 install.sh。

HTTP 请求方式	发起一个 HTTP 调用请求。配置参数如下： • 路径：请求的 URL 路径，可选项。 • 端口：请求的端口，必选项。 • 主机地址：请求的 IP 地址，可选项，默认是容器所在的节点 IP。

----结束

停止前处理

步骤 1 登录 CCE 控制台，在创建工作负载配置生命周期过程中，选择“停止前处理”。

步骤 2 在“停止前处理”后，设置停止前处理的参数，如表 6-26。

表6-27 停止前处理-参数说明

参数	说明

命令行方式	在容器中执行指定的命令，配置为需要执行的命令。命令的格式为 Command Args[1] Args[2]…（Command 为系统命令或者用户自定义可执行程序，如果未指定路径则在默认路径下寻找可执行程序），如果需要执行多条命令，建议采用将命令写入脚本执行的方式。
	如需要执行的命令如下：
	`exec:`
	`  command:`
	`  - /uninstall.sh`
	`  - uninstall_agent`
	请在执行脚本中填写: /uninstall uninstall_agent。这条命令表示容器结束前将执行 uninstall.sh。

HTTP 请求方式	发起一个 HTTP 调用请求。配置参数如下： • 路径：请求的 URL 路径，可选项。 • 端口：请求的端口，必选项。 • 主机地址：请求的 IP 地址，可选项，默认是容器所在的节点 IP。

----结束

容器重启策略

Pod 通过 restartPolicy 字段指定重启策略，重启策略类型为：Always、OnFailure 和 Never，默认为 Always。

restartPolicy 仅指通过同一节点上的 kubelet 重新启动容器。

重启策略	说明
Always	当容器失效时，由 kubelet 自动重启该容器。
OnFailure	当容器终止运行且退出码不为 0 时，由 kubelet 自动重启该容器。
Never	不论容器运行状态如何，kubelet 都不会重启该容器。

说明

可以管理 Pod 的控制器有 ReplicaSet Controller，Job，DaemonSet，及 kubelet（静态 Pod）。

- RS 和 DaemonSet：必须设置为 Always，需要保证该容器持续运行。
- Job：OnFailure 或 Never，确保容器执行完后不再重启。
- kubelet：在 Pod 失效的时候重启它，不论 RestartPolicy 设置为什么值，并且不会对 Pod 进行健康检查。

设置容器生命周期-YAML 样例

本节以 nginx 为例，说明 kubectl 命令设置容器生命周期的方法。

前提条件

请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令，使弹性云主机连接集群。

操作步骤

步骤 1 登录已配置好 kubectl 命令的弹性云主机。

步骤 2 创建一个名为 nginx-deployment.yaml 的描述文件。其中，nginx-deployment.yaml 为自定义名称，您可以随意命名。

vi nginx-deployment.yaml

在以下配置文件中，您可以看到 postStart 命令在容器目录/bin/bash 下写了个 **install.sh** 命令。 preStop 执行 **uninstall.sh** 命令。

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: nginx
    spec:
      restartPolicy: Always              #重启策略
      containers:
      - image: nginx
        command:
        - sleep 3600                     #启动命令
        imagePullPolicy: Always
        lifecycle:
          postStart:
            exec:
```

```
            command:
            - /bin/bash
            - install.sh                    #启动后命令
          preStop:
            exec:
              command:
              - /bin/bash
              - uninstall.sh                #停止前命令
        name: nginx
      imagePullSecrets:
      - name: default-secret
```

----结束

6.10.4 设置容器启动命令

操作场景

创建工作负载或任务时，通常通过镜像指定容器中运行的进程。

在默认情况下，镜像会运行默认命令，如果想运行特定命令或重写镜像默认值，需要用到以下设置：

- **工作目录**：指定运行命令的工作目录。
  若镜像中未指定工作目录，且在界面中也未指定，默认是“/”。
- **运行命令**：控制镜像运行的实际命令。
- **运行参数**：传递给运行命令的参数。

须知

容器启动后，容器中的内容不应修改。如果修改配置项（例如将容器应用的密码、证书、环境变量配置到容器中），当容器重启（例如节点异常重新调度 pod）后，会导致配置丢失，业务异常。

配置信息应通过入参等方式导入容器中，以免重启后配置丢失。

容器如何执行命令和参数

Docker 的镜像拥有存储镜像信息的相关元数据，如果不设置生命周期命令和参数，容器运行时会运行镜像制作时提供的默认的命令和参数，Docker 原生定义这两个字段为“Entrypoint”和"CMD"。

如果在创建工作负载时填写了容器的运行命令和参数，将会覆盖镜像构建时的默认命令"Entrypoint"、"CMD"，规则如下：

表6-28 容器如何执行命令和参数

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]
[touch]	[/root/test]	未设置	[/opt/test]	[touch /opt/test]
[touch]	[/root/test]	[mkdir]	[/opt/test]	[mkdir /opt/test]

设置启动命令

步骤 1 登录 CCE 控制台，在创建工作负载或任务时，展开“生命周期”。

步骤 2 在启动命令后，输入运行命令和运行参数，如表 6-29。

说明

- 当前启动命令以字符串数组形式提供，对应于 Docker 的 Entrypoint 启动命令，格式为：["executable", "param1", "param2",..]。Kubernetes 的容器启动命令可参见这里。
- 容器的生命周期与启动命令的生命周期一致，即启动命令执行完成后容器的生命周期结束。

表6-29 容器启动命令

命令方式	操作步骤
运行命令	输入可执行的命令，例如“/run/server”。 若运行命令有多个，多个命令之间用空格进行分隔。若命令本身带空格，则需要加引号（""）。 说明 多命令时，运行命令建议用/bin/sh 或其他的 shell，其他全部命令作为参数来传入。
运行参数	输入控制容器运行命令参数，例如--port=8080。 若参数有多个，多个参数以换行分隔。

下面将以启动一个 nginx 为例，介绍容器启动命令典型的三个使用场景：

示例代码如下：

```
nginx -c nginx.conf
```

- **场景一：**容器的“运行命令”和“运行参数”均作设置，界面截图如下：

图6-11 运行命令和运行参数均设置

所生成的 YAML 样例如下：

```
        command:
          - nginx
        args:
          - '-c'
          - nginx.conf
```

- **场景二：**仅设置容器的“运行命令”，界面截图如下：

图6-12 仅设置运行命令

说明

运行命令中前后要加英文双引号""，若不加则会按照空格将命令拆分成多条执行。

所生成的 YAML 样例如下：

```
        command:
          - nginx -c nginx.conf
        args:
```

- **场景三：**仅设置容器的“运行参数”，界面截图如下：

图6-13 仅设置运行参数

说明

如果运行命令没有添加到系统路径中，可以使用/bin/sh 来执行命令，命令需要加英文双引号""。

所生成的 YAML 样例如下：

```
        command:
          - /bin/sh
        args:
          - '-c'
          - '"nginx -c nginx.conf"'
```

步骤 3 您可以通过如下方式检查或修改 YAML：

- 创建工作负载时，在“高级设置”步骤中，单击右侧的“YAML 创建”。
- 工作负载创建完成后，在工作负载列表中，单击工作负载名称后的“更多 > 编辑 YAML”。
- 工作负载创建完成后，进入工作负载详情页面，单击右上角的“编辑 YAML”。

----结束

设置容器启动命令-YAML 样例

本节以 nginx 为例，说明通过 kubectl 设置容器启动命令的方法。

前提条件

请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令，使弹性云主机连接集群。

操作步骤

参见通过 kubectl 命令行创建无状态工作负载或通过 kubectl 命令行创建有状态工作负载时，容器启动命令的参数设置如下所示，详细请参见 kubernetes 官方文档。

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        command:
        - sleep
        - '3600'                          #启动命令
        imagePullPolicy: Always
        lifecycle:
          postStart:
            exec:
              command:
              - /bin/bash
              - install.sh                #启动后命令
          preStop:
            exec:
              command:
              - /bin/bash
              - uninstall.sh              #停止前命令
        name: nginx
      imagePullSecrets:
      - name: default-secret
```

6.10.5 设置容器健康检查

操作场景

健康检查是指容器运行过程中，根据用户需要，定时检查容器健康状况。若不配置健康检查，如果服务出现业务异常，pod 将无法感知，也不会自动重启去恢复业务。最终导致虽然 pod 状态显示正常，但 pod 中的业务异常的情况。

CCE 提供了两种健康检查的探针：

- **工作负载存活探针：**用于检测容器是否正常，类似于我们执行 ps 命令检查进程是否存在。如果容器的存活检查失败，集群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。
- **工作负载业务探针：**用于检查用户业务是否就绪，不就绪则不转发流量到当前实例。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。如果容器的就绪检查失败，集群会屏蔽请求访问该容器；若检查成功，则会开放对该容器的访问。

检查方式

- **HTTP 请求检查**

  HTTP 请求方式针对的是提供 HTTP/HTTPS 服务的容器，集群周期性地对该容器发起 HTTP/HTTPS GET 请求，如果 HTTP/HTTPS response 返回码属于 200~399 范围，则证明探测成功，否则探测失败。使用 HTTP 请求探测必须指定容器监听的端口和 HTTP/HTTPS 的请求路径。

  例如：提供 HTTP 服务的容器，HTTP 检查路径为：/health-check；端口为：80；主机地址可不填，默认为容器实例 IP，此处以 172.16.0.186 为例。那么集群会周期性地对容器发起如下请求：GET http://172.16.0.186:80/health-check。

  图6-14 HTTP 请求检查

- **TCP 端口检查**

  对于提供 TCP 通信服务的容器，集群周期性地对该容器建立 TCP 连接，如果连接成功，则证明探测成功，否则探测失败。选择 TCP 端口探测方式，必须指定容器监听的端口。

  例如：我们有一个 nginx 容器，它的服务端口是 80，我们对该容器配置了 TCP 端口探测，指定探测端口为 80，那么集群会周期性地对该容器的 80 端口发起 TCP 连接，如果连接成功则证明检查成功，否则检查失败。

- **执行命令检查**

  命令检查是一种强大的检查方式，该方式要求用户指定一个容器内的可执行命令，集群会周期性地在容器内执行该命令，如果命令的返回结果是 0 则检查成功，否则检查失败。

  对于上面提到的 TCP 端口检查和 HTTP 请求检查，都可以通过执行命令检查的方式来替代：

  - 对于 TCP 端口探测，我们可以写一个程序来对容器的端口进行 connect，如果 connect 成功，脚本返回 0，否则返回-1。
  - 对于 HTTP 请求探测，我们可以写一个脚本来对容器进行 wget。

    wget http://127.0.0.1:80/health-check

    并检查 response 的返回码，如果返回码在 200~399 的范围，脚本返回 0，否则返回-1。

须知

- 必须把要执行的程序放在容器的镜像里面，否则会因找不到程序而执行失败。
- 如果执行的命令是一个 shell 脚本，由于集群在执行容器里的程序时，不在终端环境下，因此不能直接指定脚本为执行命令，需要加上脚本解决器。比如脚本是 **/data/scripts/health_check.sh**，那么我们使用执行命令检查时，指定的程序应该是 **sh /data/scripts/health_check.sh**。究其原因是集群在执行容器里的程序时，不在终端环境下。

公共参数说明

表6-30 公共参数说明

参数	参数说明
延迟时间	延迟检查时间，单位为秒，此设置与业务程序正常启动时间相关。 例如，设置为 30，表明容器启动后 30 秒才开始健康检查，该时间是预留给业务程序启动的时间。
超时时间	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。

6.10.6 设置环境变量

操作场景

环境变量是指容器运行环境中设定的一个变量，环境变量可以在工作负载部署后修改，为工作负载提供极大的灵活性。

CCE 中设置的环境变量与 Dockerfile 中的“ENV”效果相同。

在 CCE 中，您可以通过三种方式来添加环境变量：手动添加、密钥导入、配置项导入。

须知

容器启动后，容器中的内容不应修改。如果修改配置项（例如将容器应用的密码、证书、环境变量配置到容器中），当容器重启（例如节点异常重新调度 pod）后，会导致配置丢失，业务异常。

配置信息应通过入参等方式导入容器中，以免重启后配置丢失。

手动添加

步骤 1 创建工作负载时，添加容器镜像后，展开“环境变量”，单击“添加环境变量”。

步骤 2 根据实际需求，设置如下参数。

- 类型：手动添加。
- 变量名称：自定义填写变量名称，如 demo。
- 变量/变量引用：输入变量的值，如 value。

图6-15 手动添加环境变量

----结束

密钥导入

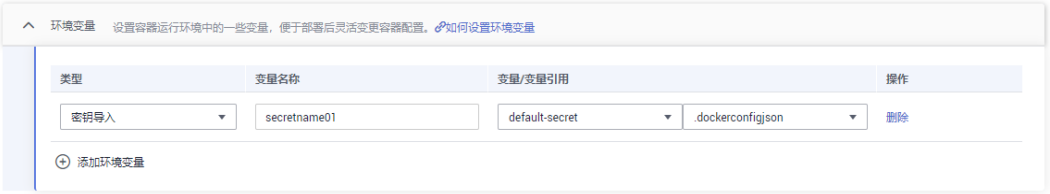
步骤 1 您需要先创建密钥，详细步骤请参见 11.3 创建密钥。

步骤 2 创建工作负载时，添加容器镜像后，展开“环境变量”，单击“添加环境变量”。

步骤 3 根据实际需求，设置如下参数。

- 类型：密钥导入。
- 变量名称：自定义填写变量名称。
- 变量/变量引用：请选择对应的密钥名称和键。

图6-16 密钥导入



----结束

配置项导入

步骤 1 您需要先创建配置项，详细步骤请参见 11.1 创建配置项。

步骤 2 创建工作负载时，添加容器镜像后，展开“环境变量”，单击“添加环境变量”。

步骤 3 根据实际需求，设置如下参数。

- 类型：配置项导入。
- 变量名称：自定义填写变量名称。
- 变量/变量引用：请选择对应的配置项名称和键。

图6-17 配置项导入

----结束

6.10.7 采集容器标准输出日志

操作场景

CCE 支持配置工作负载日志策略，便于日志的统一收集、管理和分析，以及按周期防爆处理。

本章节向您介绍如何采集容器标准输出日志。如果您需要通过配置日志策略的方式采集容器内路径日志，请参见 6.10.8 采集容器内路径日志。

操作步骤

步骤 1 在创建工作负载时，添加容器后，展开“容器日志”。

步骤 2 保留默认配置，完成工作负载的创建。

步骤 3 查看日志。

工作负载创建完成后，访问 nginx。进入工作负载详情页，单击右上角的“日志”按钮可查看日志详情。日志约需要等待 5 分钟查看。

说明

云容器引擎服务对接了应用运维管理服务 AOM 提能。2019 年 10 月 10 日对应用运维管理服务进行价格调整，每月赠送租户 500M 免费日志采集额度，超过免费额度部分将产生费用。

----结束

6.10.8 采集容器内路径日志

操作场景

CCE 支持配置工作负载日志策略，便于日志的统一收集、管理和分析，以及按周期防爆处理。

云容器引擎服务对接了应用运维管理服务 AOM，支持容器日志采集、查看、检索功能。使用该功能前首先要配置日志采集路径，本章节向您介绍如何采集容器内路径日志。

如果您需要通过不做任何配置的方式采集容器标准输出日志，请参见 6.10.7 采集容器标准输出日志。

注意事项

- ICAgent 只采集*.log、*.trace 和*.out 类型的文本日志文件。
- AOM 默认会采集容器标准输出日志，您不用做任何配置。

操作步骤

在 CCE 中添加日志策略

步骤 1 在 CCE 中创建 6.2 创建无状态负载(Deployment)时，添加容器后，展开“容器日志”配置区域。

步骤 2 单击“添加日志策略”，设置自定义日志参数，配置日志策略，以 nginx 为例，不同工作负载根据实际情况配置。

步骤 3 存储类型有“主机路径”和“容器路径”两种类型可供选择：

- 主机路径：可将主机上的路径挂载到指定的容器路径。日志策略配置参数如下：

表6-31 添加日志策略-主机路径

参数	参数说明
存储类型	设置为“主机路径”。将主机上的路径挂载到指定的容器路径。
添加容器挂载	
*主机路径	输入主机的路径，如：/var/paas/sys/log/nginx
挂载路径	输入数据卷挂载到容器上的路径，如：/tmp 须知 • 请不要挂载在系统目录下，如“/”、“**/var/run**”等，会导致容器异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响容器启动的文件，否则文件会被替换，导致容器启动异常，工作负载创建失败。 • 挂载高危目录的情况下 ，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。

参数	参数说明
	• AOM 只采集最近修改过的前 20 个日志文件，且默认采集两级子目录。 • AOM 只采集挂载路径下的“.log”、“.trace”、“.out”文本日志文件。 • 容器中挂载点的权限设置方法，请参见为 Pod 或容器配置安全性上下文。
主机扩展路径	通过实例的 ID 或者容器的名称扩展主机路径，实现同一个主机路径下区分来自不同容器的挂载。 会在原先的“卷目录/子目录”中增加一个三级目录。使用户更方便获取单个 Pod 输出的文件。 • None：不配置拓展路径。 • PodUID：Pod 的 ID。 • PodName：Pod 的名称。 • PodUID/ContainerName：Pod 的 ID/容器名称。 • PodName/ContainerName：Pod 名称/容器名称。
采集路径	设置采集路径可以更精确的指定采集内容，当前支持以下设置方式： • 不设置则默认采集当前路径下.log .trace .out 文件 • 设置**表示递归采集 5 层目录下的.log .trace .out 文件 • 设置*表示模糊匹配 例子： 采集路径为/tmp/**/test*.log 表示采集/tmp 目录及其 1-5 层子目录下的全部以 test 开头的.log 文件。 注意 使用采集路径功能请确认您的采集器 ICAgent 版本为 5.12.22 或以上版本。
日志转储	此处日志转储是指日志的本地绕接。 • 设置：AOM 每分钟扫描一次日志文件，当某个日志文件超过 50MB 时，会立即对其转储（转储时会在该日志文件所在的目录下生成一个新的 zip 文件。对于一个日志文件，AOM 只保留最近生成的 20 个 zip 文件，当 zip 文件超过 20 个时，时间较早的 zip 文件会被删除），转储完成后 AOM 会将该日志文件清空。 • 不设置：若您在下拉列表框中选择“不设置”，则 AOM 不会对日志文件进行转储。 说明 • AOM 的日志绕接能力是使用 copytruncate 方式实现的，如果选择了设置，请务必保证您写日志文件的方式是 append（追加模式），否则可能出现文件空洞问题。 • 当前主流的日志组件例如 Log4j、Logback 等均已经具备日志文件的绕接能力，如果您的日志文件已经实现了绕接能力，则无需设置。否则

参数	参数说明
	可能出现冲突。 • 建议您的业务自己实现绕接，可以更灵活的控制绕接文件的大小和个数。

- 容器路径：日志仅输出到容器路径，无需挂载主机路径。日志策略配置参数如下：

说明

此功能需要采集器 ICAgent 版本升级到 5.10.79 或以上版本。

表6-32 添加日志策略-容器路径

参数	参数说明
存储类型	设置为“容器路径”。 日志仅输出到容器路径，无需挂载主机路径。此功能需要采集器 ICAgent 版本升级到 5.10.79 或以上版本。
添加容器挂载	
挂载路径	输入数据卷挂载到容器上的路径，如：/tmp 须知 • 请不要挂载在系统目录下，如“/”、“**/var/run**”等，会导致容器异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响容器启动的文件，否则文件会被替换，导致容器启动异常，工作负载创建失败。 • 挂载高危目录的情况下 ，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。 • AOM 只采集最近修改过的前 20 个日志文件，且默认采集两级子目录。 • AOM 只采集挂载路径下的“.log”、“.trace”、“.out”文本日志文件。 • 容器中挂载点的权限设置方法，请参见为 Pod 或容器配置安全性上下文。
采集路径	设置采集路径可以更精确的指定采集内容，当前支持以下设置方式： • 不设置则默认采集当前路径下.log .trace .out 文件 • 设置**表示递归采集 5 层目录下的.log .trace .out 文件 • 设置*表示模糊匹配 例子： 采集路径为/tmp/**/test*.log 表示采集/tmp 目录及其 1-5 层子目录下的全部以 test 开头的.log 文件。 注意

参数	参数说明
	使用采集路径功能请确认您的采集器 ICAgent 版本为 5.12.22 或以上版本。
日志转储	此处日志转储是指日志的本地绕接。 • 设置：AOM 每分钟扫描一次日志文件，当某个日志文件超过 50MB 时，会立即对其转储（转储时会在该日志文件所在的目录下生成一个新的 zip 文件。对于一个日志文件，AOM 只保留最近生成的 20 个 zip 文件，当 zip 文件超过 20 个时，时间较早的 zip 文件会被删除），转储完成后 AOM 会将该日志文件清空。 • 不设置：若您在下拉列表框中选择“不设置”，则 AOM 不会对日志文件进行转储。 说明 • AOM 的日志绕接能力是使用 copytruncate 方式实现的，如果选择了设置，请务必保证您写日志文件的方式是 append（追加模式），否则可能出现文件空洞问题。 • 当前主流的日志组件例如 Log4j、Logback 等均已经具备日志文件的绕接能力，如果您的日志文件已经实现了绕接能力，则无需设置。否则可能出现冲突。 • 建议您的业务自己实现绕接，可以更灵活的控制绕接文件的大小和个数。

步骤 4 单击“确定”，并完成创建工作负载。

----结束

查看日志

日志采集路径配置和工作负载创建完成后，若已配置的路径下存在日志文件，则 ICAgent 会从已配置的路径中采集日志文件，采集大概需要 1 分钟，请您耐心等待。

待采集完成后，进入工作负载详情页，单击右上角的“日志”按钮查看日志详情并进行分析。

说明

云容器引擎服务对接了应用运维管理服务 AOM 提供日志查看、检索功能。2019 年 10 月 10 日对应用运维管理服务进行价格调整，每月赠送租户 500M 免费日志采集额度，超过免费额度部分将产生费用。

6.10.9 对接 Prometheus 实现自定义指标监控

背景知识

在 CCE 创建容器应用时，支持您通过对接普罗米修斯（prometheus）来上报自定义指标，并将其展示在应用运维管理（AOM）服务中的“监控 > 指标浏览”中。

指标是应用运维管理（AOM）服务中对资源性能的数据描述或状态描述，指标由命名空间、维度、指标名称和单位组成。指标分为系统指标和自定义指标。

- 系统指标：AOM 提供的基础指标，例如：CPU 使用率、CPU 内核占用等。
- 自定义指标：您自己定义的指标。可参考如下两种方式上报自定义指标。
  - 方式一：通过 AOM 提供的接口上报自定义指标。
  - 方式二：在 CCE 创建容器工作负载时，通过对接普罗米修斯（prometheus）上报自定义指标，详细操作请参见操作步骤。

前提条件

- 您已经成功创建一个 Kubernetes 集群，参见 3.3 购买混合集群。
- 您已连接到集群的控制节点，方便快速查看节点标签等信息，参见 3.4.2 通过 kubectl 操作 CCE 集群。
- 在设置自定义监控前，您需先了解和安装 prometheus，并在您的工作负载中提供获取自定义指标数据的 GET 接口，以便能够通过该接口获取您的自定义指标数据。

说明

当前仅支持获取 prometheus 的 Gauge 指标类型。

操作步骤

步骤 1 当您在创建工作负载时，在“高级设置”步骤中配置“自定义指标监控”。

步骤 2 您的 exporter 必须提供自定义指标的上报端口及上报路径，请参考图 6-18 设置。设置完成后，CCE 将通过“http://PodIP:上报端口/上报路径”GET 请求（例如，http://192.168.1.19:8080/metrics）获取自定义指标数据。

图6-18 设置自定义指标监控

表6-33 参数说明

参数	说明	是否必须设置
上报路径	exporter 提供的供 CCE 获取自定义指标数据的 URL。 由字母、数字、斜杠（/）和下划线（_）组成，且**必须以“/”开头**。例如，/metrics。	是
上报端口	exporter 提供的供 CCE 获取自定义指标数据的端口。 取值范围：1～65535。例如，8080。	是
监控维度	exporter 提供的自定义指标名称。 自定义指标名称为由字母、数字和下划线（_）组成的字符串，长度为 5～100 个字符。输入格式为：["自定义指标名称 1","自定义指标名称 2"]，多个自定义指标名称请以英文逗号（,）分隔。例如，["cpu_usage","mem_usage"]。 • 如果不设置，则 CCE 会获取所有的自定义指标数据。 • 如果设置，例如，设置为["cpu_usage","mem_usage"]，则 CCE 会对自定义指标进行过滤，只获取 cpu_usage、mem_usage 指标数据。	否

----结束

6.10.10 健康检查 UDP 协议安全组规则说明

操作场景

当负载均衡协议为 UDP 时，健康检查也采用的 UDP 协议，您需要打开其后端服务器的 ICMP 协议安全组规则。

操作步骤

步骤 1 登录弹性云主机控制台，找到 CCE 集群中的节点对应的弹性云主机，单击云服务器名称，进入详情页面。

步骤 2 在“安全组”页签下，单击“更改安全组规则”。

步骤 3 在打开的页面中单击“入方向规则”页签，单击“添加规则”，为云服务器添加入方向规则，单击“确定”。

说明

- 您只需为工作负载所在集群下的任意一个节点更改安全组规则，请添加规则即可，不要修改原有的安全组规则。
- 安全组需放通网段 100.125.0.0/16 流量。

----结束

6.10.11 Kubernetes 集群内置 DNS 配置说明

工作负载的 DNS 配置介绍

Kubernetes 集群内置 DNS 插件 Kube-DNS/CoreDNS，为集群内的工作负载提供域名解析服务。业务在高并发调用场景下，如果使用到域名解析服务，可能会触及到 Kube-DNS/CoreDNS 的性能瓶颈，导致 DNS 请求概率失败，影响用户业务正常运行。在 Kubernetes 使用的过程中，发现有些场景下工作负载的域名解析存在冗余的 DNS 查询，使得高并发场景更容易触及 DNS 的性能瓶颈。根据业务使用场景，对工作负载的 DNS 配置进行优化，能够在一定程度上减少 DNS 请求概率失败的问题。

更多 DNS 相关信息请参见 13.2 CoreDNS（系统资源插件，必装）或 3.4.3 通过 kubectl 配置 kube-dns/CoreDNS 高可用。

Linux 系统域名解析文件配置项的介绍

在 Linux 系统的节点或者容器里执行 cat /etc/resolv.conf 命令，能够查看到 DNS 配置，以 Kubernetes 集群的容器 DNS 配置为例：

```
nameserver 10.247.x.x
search default.svc.cluster.local svc.cluster.local cluster.local
options ndots:5
```

配置项说明：

- nameserver：容器解析域名时查询的 DNS 服务器的 IP 地址列表。如果设置为 10.247.x.x 说明 DNS 对接到 Kube-DNS/CoreDNS，如果是其他 IP 地址，则表示采用 DNS 或者用户自建的 DNS。
- search：定义域名的搜索域列表，当访问的域名不能被 DNS 解析时，会把该域名与搜索域列表中的域依次进行组合，并重新向 DNS 发起请求，直到域名被正确解析或者尝试完搜索域列表为止。对于 CCE 集群来说，容器的搜索域列表配置 3 个域，当解析一个不存在的域名时，会产生 8 次 DNS 查询，因为对于每个域名需要查询两次，分别是 IPv4 和 IPv6。
- options：定义域名解析配置文件的其他选项，常见的有 timeout、attempts 和 ndots 等等。Kubernetes 集群容器的域名解析文件设置为 options ndots:5，该参数的含义是当域名的“.”个数小于 ndots 的值，会先把域名与 search 搜索域列表进行组合后进行 DNS 查询，如果均没有被正确解析，再以域名本身去进行 DNS 查询。当域名的“.”个数大于或者等于 ndots 的值，会先对域名本身进行 DNS 查询，如果没有被正确解析，再把域名与 search 搜索域列表依次进行组合后进行 DNS 查询。如查询 www.huaweicloud.com 域名时，由于该域名的“.”个数为 2，小于 ndots 的值，所以 DNS 查询请求的顺序依次为：
  www.huaweicloud.com.default.svc.cluster.local、
  www.huaweicloud.com.svc.cluster.local、 www.huaweicloud.com.cluster.local 和
  www.huaweicloud.com，需要发起至少 7 次 DNS 查询请求才能解析出该域名的 IP。可以看出，这种配置在访问外部域名时，存在大量冗余的 DNS 查询，存在优化点。

说明

完整的 Linux 域名解析文件配置项说明可以参考文档：http://man7.org/linux/man-pages/man5/resolv.conf.5.html。

- **Kubernetes 集群应用的 DNS 相关配置项介绍**

  前面已经分析过，应用在某些场景下会出现冗余的 DNS 查询。Kubernetes 为应用提供了与 DNS 相关的配置选项，通过对应用进行 DNS 配置，能够在某些场景下有效地减少冗余的 DNS 查询，提升业务并发量。目前应用配置中与 DNS 相关的字段有 **dnsPolicy** 和 **dnsConfig**。

  dnsPolicy 字段说明：

  dnsPolicy 字段是应用设置的 DNS 策略，默认值为“ClusterFirst”。基于 dnsPolicy 策略生成的域名解析文件会与 dnsConfig 设置的 DNS 参数进行合并，合并规则将在“dnsConfig 字段说明”中说明，dnsPolicy 当前支持四种参数值：

  - ClusterFirst：应用对接 Kube-DNS/CoreDNS（CCE 集群的 Kube-DNS/CoreDNS 默认级联 DNS）。这种场景下，容器既能够解析 service 注册的集群内部域名，也能够解析发布到互联网上的外部域名。由于该配置下，域名解析文件设置了 search 搜索域列表和 ndots: 5，因此当访问外部域名和集群内部长域名（如 kubernetes.default.svc.cluster.local）时，大部分域名都会优先遍历 search 搜索域列表，导致至少有 6 次无效的 DNS 查询，只有访问集群内部短域名（如 kubernetes）时，才不存在无效的 DNS 查询。
  - ClusterFirstWithHostNet：对于配置主机网络的应用，默认配置 kubelet 的“--resolv-conf”参数指向的域名解析文件（CCE 集群在该配置下对接 DNS）。当 dnsPolicy 设置为“ClusterFirstWithHostNet”时，应用对接 Kube-DNS/CoreDNS。该配置下，容器的域名解析文件与“ClusterFirst”一致，也存在无效的 DNS 查询。
  - Default：容器的域名解析文件使用 kubelet 的“--resolv-conf”参数指向的域名解析文件（CCE 集群在该配置下对接 DNS），没有配置 search 搜索域列表和 options。该配置只能解析注册到互联网上的外部域名，无法解析集群内部域名，且不存在无效的 DNS 查询。
  - None：Kubernetes v1.9（Beta in v1.10）中引入的新选项值。设置为 None 之后，必须设置 dnsConfig，此时容器的域名解析文件将完全通过 dnsConfig 的配置来生成。

  dnsConfig 字段说明：

  dnsConfig 为应用设置 DNS 参数，设置的参数将合并到基于 dnsPolicy 策略生成的域名解析文件中。当 dnsPolicy 为“None”，应用的域名解析文件完全由 dnsConfig 指定；当 dnsPolicy 不为“None”时，会在基于 dnsPolicy 生成的域名解析文件的基础上，追加 dnsConfig 中配置的 dns 参数。

  - nameservers：DNS 的 IP 地址列表。当应用的 dnsPolicy 设置为“None”时，列表必须至少包含一个 IP 地址，否则此属性是可选的。列出的 DNS 的 IP 列表将合并到基于 dnsPolicy 生成的域名解析文件的 nameserver 字段中，并删除重复的地址。
  - searches：域名查询时的 DNS 搜索域列表，此属性是可选的。指定后，提供的搜索域列表将合并到基于 dnsPolicy 生成的域名解析文件的 search 字段中，并删除重复的域名。Kubernetes 最多允许 6 个搜索域。
  - options：DNS 的配置选项，其中每个对象可以具有 name 属性（必需）和 value 属性（可选）。该字段中的内容将合并到基于 dnsPolicy 生成的域名解析文件的 options 字段中，dnsConfig 的 options 的某些选项如果与基于 dnsPolicy 生成的域名解析文件的选项冲突，则会被 dnsConfig 所覆盖。

工作负载的 DNS 配置实践

前面介绍了 Linux 系统域名解析文件以及 Kubernetes 为应用提供的 DNS 相关配置项，下面将举例介绍应用如何进行 DNS 配置。

- **场景 1 对接 kubernetes 内置的 Kube-DNS/CoreDNS**

  场景说明：

  这种方式适用于应用中的域名解析只涉及集群内部域名，或者集群内部域名+外部域名两种方式，应用默认采用这种配置。

  示例：

```
apiVersion: v1
kind: Pod
metadata:
  namespace: default
  name: dns-example
spec:
  containers:
  - name: test
    image: nginx
  dnsPolicy: ClusterFirst
```

  该配置下容器的域名解析文件将如下所示：

```
nameserver 10.247.3.10
search default.svc.cluster.local svc.cluster.local cluster.local
options ndots:5
```

- **场景 2 直接对接 DNS**

  场景说明：

  这种方式适用于应用只访问注册到互联网的外部域名，该场景不能解析集群内部域名。

  示例：

```
apiVersion: v1
kind: Pod
metadata:
  namespace: default
  name: dns-example
spec:
  containers:
  - name: test
    image: nginx
  dnsPolicy: Default //使用 kubelet 的"--resolv-conf"参数指向的域名解析文件（CCE 集群在该配置下对接 DNS）
```

  该配置下容器的域名解析文件将如下所示：

```
nameserver 10.125.x.x
```

- **场景 3 主机网络模式的应用对接 Kube-DNS/CoreDNS**

  场景说明：

  对于配置主机网络模式的应用，默认对接 DNS，如果应用需要对接 Kube-DNS/CoreDNS，需将 dnsPolicy 设置为“ClusterFirstWithHostNet”。

  示例：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
spec:
  template:
    metadata:
      labels:
        app: nginx
    spec:
      hostNetwork: true
      dnsPolicy: ClusterFirstWithHostNet
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

该配置下容器的域名解析文件将如下所示：

```
nameserver 10.247.3.10
search default.svc.cluster.local svc.cluster.local cluster.local
options ndots:5
```

- **场景 4 自定义应用的域名配置**

  场景说明：

  用户可以完全自定义配置应用的域名解析文件，这种方式非常灵活，dnsPolicy 和 dnsConfig 配合使用，几乎能够满足所有使用场景，如对接用户自建 DNS 的场景、串联多个 DNS 的场景以及优化 DNS 配置选项的场景等等。

  示例 1：对接用户自建 DNS

  该配置下，dnsPolicy 为“None”，应用的域名解析文件完全根据 dnsConfig 配置生成。

```
apiVersion: v1
kind: Pod
metadata:
  namespace: default
  name: dns-example
spec:
  containers:
  - name: test
    image: nginx
  dnsPolicy: "None"
  dnsConfig:
    nameservers:
    - 10.2.3.4 //用户自建 DNS 的 IP 地址
    searches:
    - ns1.svc.cluster.local
    - my.dns.search.suffix
    options:
    - name: ndots
      value: "2"
    - name: timeout
      value: "3"
```

该配置下容器的域名解析文件将如下所示：

```
nameserver 10.2.3.4
search ns1.svc.cluster.local my.dns.search.suffix
options timeout:3 ndots:2
```

示例 2：修改域名解析文件的 ndots 选项，减少无效的 DNS 查询

该配置下，dnsPolicy 不为“None”，会在基于 dnsPolicy 生成的域名解析文件的基础上，追加 dnsConfig 中配置的 dns 参数。

```
apiVersion: v1
kind: Pod
metadata:
  namespace: default
  name: dns-example
spec:
  containers:
  - name: test
    image: nginx
  dnsPolicy: "ClusterFirst"
  dnsConfig:
    options:
    - name: ndots
      value: "2" //该配置会将基于 ClusterFirst 策略生成的域名解析文件的 ndots:5 参数改写为
ndots:2
```

该配置下容器的域名解析文件将如下所示：

```
nameserver 10.247.3.10
search default.svc.cluster.local svc.cluster.local cluster.local
options ndots:2
```

7 亲和/反亲和性调度

7.1 调度策略概述

CCE 支持“自定义调度策略”和“简易调度策略”。自定义调度策略开放节点亲和、工作负载亲和以及工作负载反亲和调度策略的配置，以满足用户的更高需求。简易调度策略提供简单便捷以及足够功能的调度方式。

自定义调度策略

在自定义调度策略中用户可以设置“节点亲和性”、“工作负载亲和性”和“工作负载反亲和性”，详情详见 Affinity and anti-affinity。

- 7.2.1 节点亲和性
- 7.2.2 工作负载亲和性
- 7.2.3 工作负载反亲和性

说明

亲和性调度中需要设置的标签依赖节点的标签和 Pod 的标签，用户可以使用内置的标签进行设置，或者可以根据需求自定义节点和 Pod 的标签并使用。

简易调度策略

简易调度策略提供工作负载和可用区的亲和性、工作负载和节点的亲和性以及工作负载间的亲和性调度，用户可根据业务需求进行相应的设置部署工作负载。

- **工作负载和可用区的亲和性：**基于可用区可以设置多条调度策略（包含亲和、反亲和），但只需满足其中一条就会进行调度。
  - **与可用区的亲和性：**即 7.3.1 工作负载和可用区的亲和性。
  - **与可用区的反亲和性：**即 7.3.2 工作负载和可用区的反亲和性。
- **工作负载和节点的亲和性：**基于节点可以设置多条调度策略（包含亲和、反亲和），但只需满足其中一条就会进行调度。例如集群下有节点 A、B、C，设置调度策略为亲和节点 A，反亲和节点 B，则工作负载可以调度到除 B 以外任意一个节点上
  - **与节点的亲和性：**即 7.3.3 工作负载和节点的亲和性。

  - **与节点的反亲和性**：即 7.3.4 工作负载和节点的反亲和性。

- **工作负载间的亲和性**：基于工作负载可以设置多条调度策略，但多条策略中设置的标签必须同时出现在一个工作负载中。
  - **与工作负载的亲和性**：即 7.3.5 工作负载间的亲和性。用户可根据业务需求进行工作负载的就近部署，容器间通信就近路由，减少网络消耗。

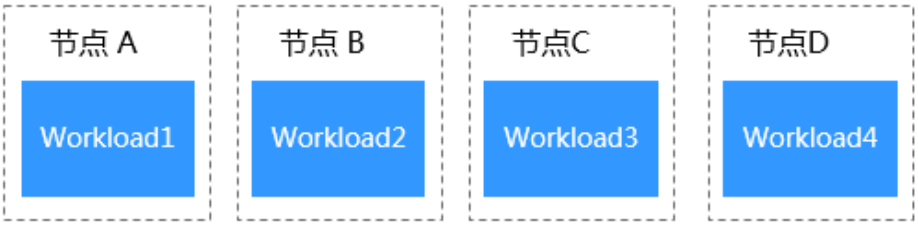
    如图 7-1，Workload1、Workload2、Workload3 和 Workload4 部署在相同节点上，为亲和性部署。

    图7-1 工作负载间亲和

  - **与工作负载的反亲和性**：即 7.3.6 工作负载间的反亲和性。同个工作负载的多个实例反亲和部署，减少宕机影响；互相干扰的工作负载反亲和部署，避免干扰。

    如图 7-2，Workload1、Workload2、Workload3 和 Workload4 分别部署在不同节点上，这四个工作负载为反亲和性部署。

    图7-2 工作负载间反亲和

须知

在设置“工作负载间的亲和性”和“工作负载和节点的亲和性”时，需确保不要出现互斥情况，否则工作负载会部署失败。

例如如下互斥情况，工作负载将会部署失败：

- Workload1、Workload2 设置了工作负载间的反亲和，例如 Workload1 部署在“节点 A”，Workload2 部署在“节点 B”。
- Workload3 部署上线时，既希望与 Workload2 亲和，又希望可以部署在不同节点如“节点 C”或**“节点 A”**上，这就造成了工作负载亲和节点亲和间的互斥，导致最终工作负载部署失败。

7.2 自定义调度策略

7.2.1 节点亲和性

通过控制台设置

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”或“工作负载 > 有状态负载 StatefulSet”。

步骤 2 在无状态工作负载或有状态工作负载列表中，单击工作负载名称进入详情页，在“调度策略”页签下，单击 “自定义调度策略”。

步骤 3 在节点亲和性设置中，依据节点中的标签进行业务需求的设置。

说明

节点亲和性调度支持必须满足和尽量满足（硬约束 Required/软约束 Preferred），以及可以设置相应的匹配关系（In, NotIn, Exists, DoesNotExist, Gt, and Lt）：

- **必须满足**：即硬约束，设置必须要满足的条件，对应于 requiredDuringSchedulingIgnoredDuringExecution，您可以添加多条必须满足的规则，多条规则间是一种“或”的关系，即只需要满足一条规则即会进行调度。
- **尽量满足**：即软约束，设置尽量满足的条件，对应于 preferredDuringSchedulingIgnoredDuringExecution，您可以添加多条尽量满足的规则，无论是满足其中一条或者是都不满足都会进行调度。另外可以为规则设置权重值，权重值越高会被优先调度。
- **选择器**：对应于 matchExpressions，您可以添加多条选择器，多条选择器之间是一种“与”的关系，即需要满足全部选择器才能依据此条规则进行调度。
- **标签名**：对应节点的标签，可以使用默认的标签也可以用户自定义标签。
- **匹配关系**：即操作符，可以设置六种匹配关系（In, NotIn, Exists, DoesNotExist. Gt, and Lt）。In 和 NotIn 操作符可以添加单个值或者多个 value 值（多值使用；进行划分），Exists 和 DoesNotExist 判断某个 label 是否存在，不需设置 value 值。Gt 和 Lt 判断 label 的值大于或者小于某个值（value 值要求整数）。

----结束

通过 kubectl 命令行设置

本节以 nginx 为例，创建节点的亲和性。

前提条件

已有使用 nginx 容器的工作负载和节点。

操作步骤

使用内置节点标签 kubernetes.io/hostname，并添加相应的节点，同时设置操作符为 In，最后单击“确定”提交。

设置后的工作负载节点亲和性所得的 yaml 如下：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      imagePullSecrets:
        - name: default-secret
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
              - matchExpressions:
                  - key: kubernetes.io/hostname
                    operator: In
                    values:
                      - 192.168.6.174
```

7.2.2 工作负载亲和性

通过控制台设置

决定工作负载的 Pod 和哪些工作负载的 Pod 部署在同一个拓扑域。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”或“工作负载 > 有状态负载 StatefulSet”。

步骤 2 在无状态工作负载或有状态工作负载列表中，单击工作负载名称进入详情页，在“调度策略”页签下，单击 “自定义调度策略”。

步骤 3 在工作负载亲和性设置中，依据工作负载的 Pod 标签，同时配置命名空间和拓扑域进行业务需求的设置。

说明

工作负载亲和性调度支持必须满足和尽量满足（硬约束 Required/软约束 Preferred），以及可以设置相应的匹配关系（In, NotIn, Exists, DoesNotExist）：

- **必须满足**：即硬约束，设置必须要满足的条件，对应于 requiredDuringSchedulingIgnoredDuringExecution，您可以添加多条必须满足的规则。在规则中还需要设置命名空间和拓扑域。
- **尽量满足**：即软约束，设置尽量满足的条件，对应于 preferredDuringSchedulingIgnoredDuringExecution，您可以添加多条尽量满足的规则，无论是满足其中一条或者是都不满足都会进行调度。另外可以为规则设置权重值，权重值越高会被优先调度。
- **命名空间**：即 namespaces，默认情况下使用和当前工作负载相同的命名空间，您可以设置其他的已有的命名空间。
- **拓扑域**：即 topologyKey，拓扑域通过设置工作节点的标签，包含默认和自定义标签，用于指定调度时作用域。
- **选择器**：对应于 matchExpressions，您可以添加多条选择器，多条选择器之间是一种“与”的关系，即需要满足全部选择器才能依据此条规则进行调度。
- **标签名**：对应工作负载的标签，您可以使用默认标签 app 或者使用自定义标签。
- **匹配关系**：即操作符，可以设置四种匹配关系（In, NotIn, Exists, DoesNotExist）。In 和 NotIn 操作符可以添加单个值或者多个 value 值（多值使用；进行划分），Exists 和 DoesNotExist 判断某个 label 是否存在，不需设置 value 值。

----结束

通过 kubectl 命令行设置

本节以 nginx 为例，创建工作负载节点的亲和性。

前提条件

已有使用 nginx 容器的工作负载和节点。

操作步骤

使用默认的命名空间 default，拓扑域使用内置的节点标签 kubernetes.io/hostname 用于表示以节点为区分范围，设置标签 app 和 type，标签值为 redis 和 database。同时设置操作符为 In，最后单击“确定”提交。

设置后的工作负载亲和性所得的 yaml 如下：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
  namespace: default
spec:
  replicas: 2
  selector:
```

```
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      imagePullSecrets:
        - name: default-secret
      affinity:
        podAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            - labelSelector:
                matchExpressions:
                  - key: app
                    operator: In
                    values:
                      - redis
                  - key: type
                    operator: In
                    values:
                      - database
                    namespaces:
                      - default
              topologyKey: kubernetes.io/hostname
```

须知

上例中只有当某个工作负载同时具有 app 和 type 两个标签的时候，工作负载 nginx 才能调度成功，会将工作负载 nginx 调度到此工作负载的节点上。

7.2.3 工作负载反亲和性

通过控制台设置

决定工作负载的 Pod 不和哪些工作负载的 Pod 部署在同一个拓扑域。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”或“工作负载 > 有状态负载 StatefulSet”。

步骤 2 在无状态工作负载或有状态工作负载列表中，单击工作负载名称进入详情页，在“调度策略”页签下，单击 “自定义调度策略”。

步骤 3 在工作负载反亲和性设置中，依据工作负载中标签同时还可以配置命名空间和拓扑域进行业务需求的设置。

说明

工作负载反亲和性调度支持必须满足和尽量满足（硬约束 Required/软约束 Preferred），以及可以设置相应的匹配关系（In、 NotIn、Exists、DoesNotExist）：

- **必须满足**：即硬约束，设置必须要满足的条件，对应于 requiredDuringSchedulingIgnoredDuringExecution，您可以添加多条必须满足的规则。在规则中还需要设置命名空间和拓扑域。
- **尽量满足**：即软约束，设置尽量满足的条件，对应于 preferredDuringSchedulingIgnoredDuringExecution，您可以添加多条尽量满足的规则，无论是满足其中一条或者是都不满足都会进行调度。另外可以为规则设置权重值，权重值越高会被优先调度。
- **命名空间**：即 namespaces，默认情况下使用和当前工作负载相同的命名空间，您可以设置其他的已有的命名空间。
- **拓扑域**：即 topologyKey，拓扑域通过设置工作节点的标签，包含默认和自定义标签，用于指定调度时作用域。
- **选择器**：对应于 matchExpressions，您可以添加多条选择器，多条选择器之间是一种“与”的关系，即需要满足全部选择器才能依据此条规则进行调度。
- **标签名**：对应工作负载的标签，您可以使用默认标签 app 或者使用自定义标签。
- **匹配关系**：即操作符，可以设置四种匹配关系（In, NotIn, Exists, DoesNotExist）。In 和 NotIn 操作符可以添加单个值或者多个 value 值（多值使用；进行划分），Exists 和 DoesNotExist 判断某个 label 是否存在，不需设置 value 值。

----结束

通过 kubectl 命令行设置

本节以 nginx 为例，创建工作负载的节点反亲和性。

前提条件

已有使用 nginx 容器的工作负载和节点。

操作步骤

使用默认的命名空间 default，拓扑域使用内置的节点标签 kubernetes.io/hostname 用于表示以节点为区分范围，设置标签 app，标签值为 redis。同时设置操作符为 In，最后单击“确定”提交。

设置后的工作负载反亲和性所得的 yaml 如下：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
```

```
      app: nginx
  spec:
    imagePullSecrets:
      - name: default-secret
    affinity:
      podAntiAffinity:
        requiredDuringSchedulingIgnoredDuringExecution:
          - labelSelector:
              matchExpressions:
                - key: app
                  operator: In
                  values:
                    - redis
            namespaces:
              - default
            topologyKey: kubernetes.io/hostname
```

7.3 简易调度策略

7.3.1 工作负载和可用区的亲和性

通过控制台设置

步骤 1 参照 6.2 创建无状态负载(Deployment)或 6.3 创建有状态负载(StatefulSet)，在“高级设置”的“调度策略”下，单击“工作负载和可用区的亲和性 > 与可用区的亲和性”前的 。

步骤 2 单击工作负载希望部署到的可用区。

当前创建的工作负载会部署到已选择的可用区中。

----结束

通过 kubectl 命令行设置

本节以 nginx 为例，说明 kubectl 命令创建工作负载的方法。

前提条件

请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令，使弹性云主机连接集群。

操作步骤

参见通过 kubectl 命令行创建无状态工作负载或通过 kubectl 命令行创建有状态工作负载，工作负载和可用区亲和性的 yaml 示例如下：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: az-in-deployment
spec:
  replicas: 1
```

```
  selector:
    matchLabels:
      app: az-in-deployment
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: az-in-deployment
    spec:
      containers:
      - image: nginx
        imagePullPolicy: Always
        name: nginx
      imagePullSecrets:
      - name: default-secret
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: failure-domain.beta.kubernetes.io/zone #node 中 lable 的 key
                operator: In
                values:
                - az1                                #node 中对应 key 的 value
```

工作负载创建完成后设置

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”或“工作负载 > 有状态负载 StatefulSet”。

步骤 2 单击工作负载名称进入详情页，单击“调度策略 > 简易调度策略 > 添加亲和对象”。

步骤 3 对象类型选为“可用区”，单击工作负载希望部署到的可用区，当前创建的工作负载会部署到已选择的可用区中。

说明

该方法可新增、编辑和删除调度策略。

----结束

7.3.2 工作负载和可用区的反亲和性

通过控制台设置

步骤 1 参照 6.2 创建无状态负载(Deployment)或 6.3 创建有状态负载(StatefulSet)，在“高级设置”的“调度策略”下，单击“工作负载和可用区的亲和性 > 与可用区的反亲和性”前的 。

步骤 2 单击工作负载不希望部署到的可用区。

当前创建的工作负载不会部署到已选择的可用区中。

----结束

通过 kubectl 命令行设置

本节以 nginx 为例，说明 kubectl 命令创建工作负载的方法。

前提条件

请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令，使弹性云主机连接集群。

操作步骤

参见通过 kubectl 命令行创建无状态工作负载或通过 kubectl 命令行创建有状态工作负载，工作负载和可用区反亲和性的 yaml 示例如下：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        imagePullPolicy: Always
        name: nginx
      imagePullSecrets:
      - name: default-secret
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: failure-domain.beta.kubernetes.io/zone       #node 中 label 的 key
                operator: NotIn
                values:
                - az1                                    #node 中对应 key 的 value
```

工作负载创建完成后设置

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”或“工作负载 > 有状态负载 StatefulSet”。

步骤 2 单击工作负载名称进入详情页，单击“调度策略 > 简易调度策略 > 添加反亲和对象”。

步骤 3 对象类型选为“可用区”，单击工作负载不希望部署到的可用区，当前创建的工作负载不会部署到已选择的可用区中。

说明

该方法可新增、编辑和删除调度策略。

----结束

7.3.3 工作负载和节点的亲和性

通过控制台设置

步骤 1 参照 6.2 创建无状态负载(Deployment)或 6.3 创建有状态负载(StatefulSet)，在“高级设置”的“调度策略”下，单击“工作负载和节点的亲和性 > 与节点的亲和性”下的“添加”。

步骤 2 勾选工作负载需要部署到的节点，单击“确定”。

若勾选多个节点，部署工作负载时会在这些勾选的节点中自动指定。

----结束

通过 kubectl 命令行设置

本节以 nginx 为例，说明 kubectl 命令创建工作负载的方法。

前提条件

请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令，使弹性云主机连接集群。

操作步骤

参见通过 kubectl 命令行创建无状态工作负载或通过 kubectl 命令行创建有状态工作负载，工作负载和节点亲和性的 yaml 示例如下：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: nginx
```

```
  spec:
    containers:
    - image: nginx
      imagePullPolicy: Always
      name: nginx
    imagePullSecrets:
    - name: default-secret
    affinity:
      nodeAffinity:
        requiredDuringSchedulingIgnoredDuringExecution:
          nodeSelectorTerms:
          - matchExpressions:
            - key: nodeName          #node 中 lable 的 key
              operator: In
              values:
              - test-node-1          #node 中对应 key 的 value
```

工作负载创建完成后设置

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”或“工作负载 > 有状态负载 StatefulSet”。

步骤 2 单击工作负载名称进入详情页，单击“调度策略 > 简易调度策略 > 添加亲和对象”。

步骤 3 对象类型选为“节点”，勾选工作负载希望部署到的节点，设置完成后当前工作负载会部署到已选择的节点上。

说明

该方法可新增、编辑和删除调度策略。

----结束

7.3.4 工作负载和节点的反亲和性

通过控制台设置

步骤 1 参照 6.2 创建无状态负载(Deployment)或 6.3 创建有状态负载(StatefulSet)，在“高级设置”的“调度策略”下，单击“工作负载和节点的亲和性 > 与节点的反亲和性”下的“添加”。

步骤 2 勾选工作负载不希望部署到的节点，单击“确定”。

若勾选多个节点，工作负载将不会部署到这些节点上。

----结束

通过 kubectl 命令行设置

本节以 nginx 为例，说明 kubectl 命令创建工作负载的方法。

前提条件

请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令，使弹性云主机连接集群。

操作步骤

参见通过 kubectl 命令行创建无状态工作负载或通过 kubectl 命令行创建有状态工作负载，工作负载和节点反亲和性的 yaml 示例如下：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        imagePullPolicy: Always
        name: nginx
      imagePullSecrets:
      - name: default-secret
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: nodeName          #node 中 lable 的 key
                operator: NotIn        #notin 说明不部署
                values:
                - test-node-1          #node 中对应 key 的 value
```

工作负载创建完成后设置

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”或“工作负载 > 有状态负载 StatefulSet”。

步骤 2 单击工作负载名称进入详情页，单击“调度策略 > 简易调度策略 > 添加反亲和对象”。

步骤 3 对象类型选择“节点”，勾选工作负载不希望部署到的节点，设置完成后当前工作负载不会部署到已选择的节点上。

图7-3 添加反亲和对象-节点

说明

该方法可新增、编辑和删除调度策略。

----结束

7.3.5 工作负载间的亲和性

通过控制台设置

步骤 1 参照 6.2 创建无状态负载(Deployment)或 6.3 创建有状态负载(StatefulSet)，在“高级设置”的“调度策略”下，单击“工作负载间的亲和性 > 与工作负载的亲和性”下的“添加”。

步骤 2 勾选希望部署到相同节点的工作负载，单击“确定”。

当前创建的工作负载会和已勾选的工作负载部署在相同节点上。

----结束

通过 kubectl 命令行设置

本节以 nginx 为例，说明 kubectl 命令创建工作负载的方法。

前提条件

请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令，使弹性云主机连接集群。

操作步骤

参见通过 kubectl 命令行创建无状态工作负载或通过 kubectl 命令行创建有状态工作负载，工作负载间亲和性的 yaml 示例如下：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        imagePullPolicy: Always
        name: nginx
      imagePullSecrets:
      - name: default-secret
      affinity:
        podAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
```

```
          nodeSelectorTerms:
          - matchExpressions:
            - key: app          #工作负载中 lable 的 key
              operator: In
              values:
              - test     #工作负载中对应 lable 的 value
```

工作负载创建完成后设置

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”或“工作负载 > 有状态负载 StatefulSet”。

步骤 2 单击工作负载名称进入详情页，单击“调度策略 > 简易调度策略 > 添加亲和对象”。

步骤 3 对象类型选择“工作负载”，勾选希望部署到相同节点上的工作负载，设置完成后当前工作负载会和已勾选的工作负载部署在相同节点上。

说明

该方法可新增、编辑和删除调度策略。

----结束

7.3.6 工作负载间的反亲和性

通过控制台设置

步骤 1 参照 6.2 创建无状态负载(Deployment)或 6.3 创建有状态负载(StatefulSet)，在“高级设置”的“调度策略”下，单击“工作负载间的亲和性 > 与工作负载的反亲和性”下的“添加”。

步骤 2 勾选不希望部署到相同节点的工作负载，单击“确定”。

当前创建的工作负载会和已勾选的工作负载部署在不同节点上。

----结束

通过 kubectl 命令行设置

本节以 nginx 为例，说明 kubectl 命令创建工作负载的方法。

前提条件

请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令，使弹性云主机连接集群。

操作步骤

参见通过 kubectl 命令行创建无状态工作负载或通过 kubectl 命令行创建有状态工作负载，工作负载间反亲和性的 yaml 示例如下：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
```

```
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        imagePullPolicy: Always
        name: nginx
      imagePullSecrets:
      - name: default-secret
      affinity:
        podAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: app          #工作负载中 lable 的 key
                operator: NotIn
                values:
                - test     #工作负载中对应 lable 的 value
```

工作负载创建完成后设置

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”或“工作负载 > 有状态负载 StatefulSet”。

步骤 2 单击工作负载名称进入详情页，单击“调度策略 > 简易调度策略 > 添加反亲和对象”。

步骤 3 对象类型选择“工作负载”，勾选希望部署到不同节点上的工作负载，设置完成后当前的工作负载会和已勾选的工作负载部署在不同节点上。

说明

该方法可新增、编辑和删除调度策略。

----结束

8 网络管理

8.1 网络概述

云容器引擎通过将 Kubernetes 网络和 VPC 深度集成，提供了稳定高性能的容器网络，能够满足多种复杂场景下工作负载间的互相访问。

约束与限制

- 每个命名空间下，创建的服务数量不能超过 6000 个。此处的服务对应 kubernetes 的 service 资源，即工作负载所添加的服务。
- 容器开启 hostPort 或 hostNetwork 网络模式后，若需对外提供服务，则需要给容器所在节点开启对应端口的安全组(containerPort)。

Service 基本概念

Kubernetes 中每一个工作负载会有一个或多个实例（Pod），每个实例（Pod）的 IP 地址由网络插件动态随机分配（Pod 重启后 IP 地址会改变）。为屏蔽这些后端实例的动态变化和对多实例的负载均衡，引入了 Service 这个资源对象。

Service 是一种资源，提供了我们访问单个或多个容器应用的能力。每个服务在其生命周期内，都拥有一个固定的 IP 地址和端口。每个服务对应了后台的一个或多个 Pod，通过这种方式，客户端就不需要关心 Pod 所在的位置，方便后端进行方便的 Pod 扩容、缩容等操作。

用户在 Kubernetes 中可以部署各种容器，其中一部分是通过 HTTP、HTTPS 协议对外提供七层网络服务，另一部分是通过 TCP、UDP 协议提供四层网络服务。而 Kubernetes 定义的 Service 资源就是用来管理集群中四层网络的服务访问。

根据创建 Service 的 type 类型不同，可分成如下模式：

- **ClusterIP**：默认类型，为服务分配集群虚拟 IP，此时集群内部的 pod 可以通过服务名称寻址到服务的集群虚拟 IP 地址，集群外无效。
- **NodePort**：在每个节点上为服务分配静态端口号，此时如果在集群外部访问任何一个节点的 IP 地址加指定的端口号，kube-proxy 会将流量转发到服务的集群虚拟 IP，再由虚拟 IP 寻址到 Pod。
- **LoadBalancer**：通过云服务供应商提供的 load balancer 向外部暴露服务，由指定的 load balancer 负责对 NodePort 与 ClusterIP 服务的路由。

ClusterIP 和 NodePort 类型的 Service，在不同云服务商或是自建集群中的行为表现通常情况下相同。而 LoadBalancer 类型的 Service，由于使用了云服务商的负载均衡进行服务暴露，云服务商会围绕其负载均衡的能力提供不同的额外功能。例如，控制负载均衡的网络类型，后端绑定的权重调节等，详情请参见本章相关文档。

服务访问方式

在 CCE 中，支持以下类型的互联互通：

- **集群内访问（ClusterIP）**

  工作负载暴露给同一集群内其他工作负载访问的方式，可以通过“集群内部域名”访问。集群内部域名格式为“<自定义的访问方式名称>.<工作负载所在命名空间>.svc.cluster.local”，例如“nginx.default.svc.cluster.local”。详细请参见 8.2.1 集群内访问(ClusterIP)。

- **节点访问（NodePort）**

  节点访问（NodePort）是指在每个节点的 IP 上开放一个静态端口，通过静态端口对外暴露服务。节点访问（NodePort）会路由到 ClusterIP 服务，这个 ClusterIP 服务会自动创建。通过请求 <NodeIP>:<NodePort>，可以从集群的外部访问一个 NodePort 服务。详细请参见 8.2.2 节点访问(NodePort)。

- **负载均衡（LoadBalancer）**

  通过弹性负载均衡从公网访问工作负载，与弹性 IP 方式相比提供了高可靠的保障，一般用于系统中需要暴露到公网的服务。访问方式由公网弹性负载均衡 ELB 服务地址以及设置的访问端口组成，例如“10.117.117.117:80”。详细请参见 8.2.3 负载均衡(LoadBalancer)。

七层负载均衡（Ingress）

通常情况下，service 和 pod 的 IP 仅可在集群内部访问。集群外部的请求需要通过负载均衡转发到 service 在 Node 上暴露的 NodePort 上，然后再由 kube-proxy 通过边缘路由器 (edge router) 将其转发给相关的 Pod 或者丢弃，而 Ingress 就是为进入集群的请求提供路由规则的集合。

Ingress 可以给 service 提供集群外部访问的 URL、负载均衡、SSL 终止、HTTP 路由等。为了配置这些 Ingress 规则，集群管理员需要部署一个 Ingress controller，它监听 Ingress 和 service 的变化，并根据规则配置负载均衡并提供访问入口。

Ingress 的组成部分：

- Nginx：实现负载均衡到 pod 的集合。
- Ingress Controller：从集群 api 获取 services 对应 pod 的 ip 到 nginx 配置文件中。
- Ingress：为 nginx 创建虚拟主机。

Ingress 的创建与管理请参见 8.3.1 Ingress 概述。

网络策略（NetworkPolicy）

网络策略（NetworkPolicy）是一种关于 pod 间及 pod 与其他网络端点间所允许的通信规则的规范。

NetworkPolicy 资源使用标签选择 pod，并定义选定 pod 所允许的通信规则，详细请参见 8.5 NetworkPolicy。

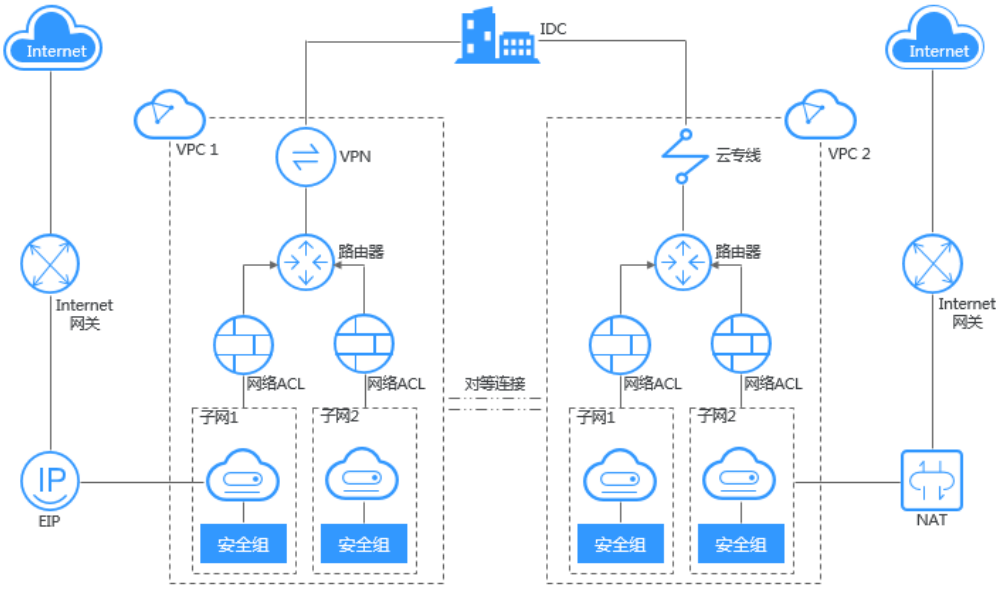
网络底层基础设施

- **VPC**

  虚拟私有云（Virtual Private Cloud，以下简称 VPC），是基于创建的自定义私有网络，不同的专有网络之间彻底逻辑隔离。可以为云服务器、云数据库和负载均衡等构建隔离的、用户自主配置和管理的虚拟网络环境，提升用户云上资源的安全性，简化用户的网络部署。

  您可以在 VPC 中定义安全组、VPN、IP 地址段、带宽等网络特性。用户可以通过 VPC 方便地管理、配置内部网络，进行安全、快捷的网络变更。同时，用户可以自定义安全组内与组间弹性云主机的访问规则，加强弹性云主机的安全保护。

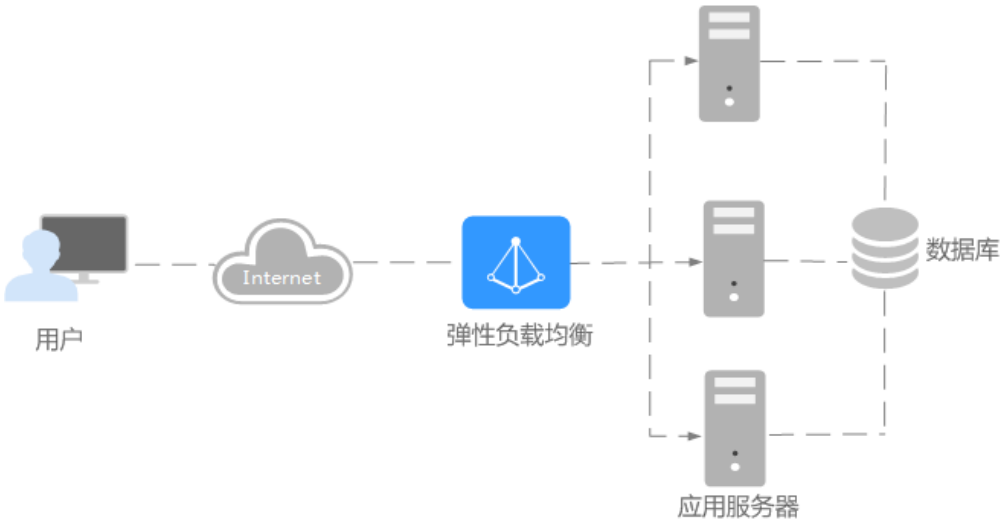
  图8-1 VPC 组成



- **ELB**

  弹性负载均衡（Elastic Load Balance，简称 ELB）是将访问流量根据转发策略分发到后端多台服务器的流量分发控制服务。弹性负载均衡可以通过流量分发扩展应用系统对外的服务能力，通过消除单点故障提升应用系统的可用性。

图8-2 弹性负载均衡



弹性负载均衡支持经典型、共享型两种负载均衡：

- 经典型负载均衡：适用于访问量较小，应用模型简单的 web 业务。
- 共享型负载均衡：适用于访问量较大的 web 业务，提供基于域名和 URL 的路由均衡能力，实现更加灵活的业务转发。

8.2 Service

8.2.1 集群内访问(ClusterIP)

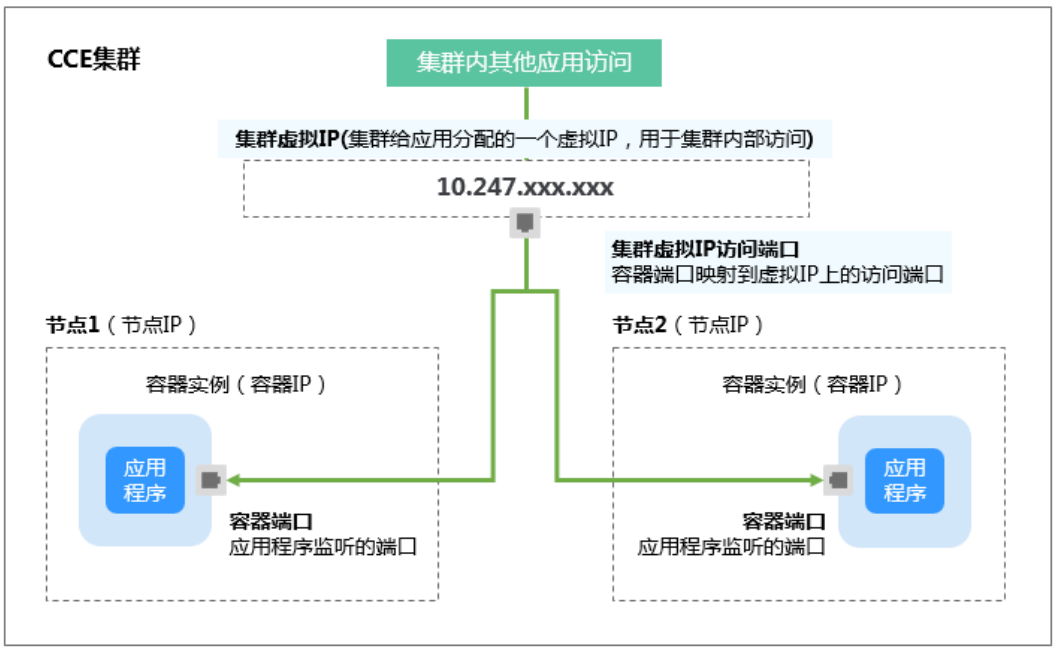
操作场景

集群内访问表示工作负载暴露给同一集群内其他工作负载访问的方式，可以通过“集群内部域名”访问。

集群内部域名格式为“<自定义的服务名称>.<工作负载所在命名空间>.svc.cluster.local:<端口号>”，例如“nginx.default.svc.cluster.local:80”。

访问通道、容器端口与访问端口映射如图 8-3 所示。

图8-3 集群内访问



工作负载创建时设置

您可以在创建工作负载时通过 CCE 控制台设置 Service 访问方式，如下：

步骤 1 参考 6.2 创建无状态负载(Deployment)、6.3 创建有状态负载(StatefulSet)或 6.4 创建守护进程集(DaemonSet)，在“工作负载访问设置”步骤，单击“添加服务”。

- **访问类型**：选择“集群内访问（ClusterIP）”。
- **Service 名称**：自定义服务名称，可与工作负载名称保持一致。
- **端口配置**：
  - 协议：请根据业务的协议类型选择。
  - 容器端口：工作负载程序实际监听的端口，需用户确定。nginx 程序实际监听的端口为 80。
  - 访问端口：容器端口映射到集群虚拟 IP 上的端口，用虚拟 IP 访问工作负载时使用，端口范围为 1-65535，可任意指定。

步骤 2 单击“下一步”进入“高级设置”页面，直接单击“创建”。

步骤 3 单击“查看工作负载详情”，在“访问方式”页签下获取访问地址，例如 10.247.74.100:8080。

----结束

工作负载创建完成后设置

您可以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 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”，在工作负载列表页单击要设置 Service 的工作负载名称。

步骤 2 在“Service”页签，单击“添加 Service”。

步骤 3 在“添加 Service”页面，访问类型选择“集群内访问（ClusterIP）”。

步骤 4 设置集群内访问参数。

- **Service 名称：**自定义服务名称，可与工作负载名称保持一致。
- **集群名称：**工作负载所在集群的名称，此处不可修改。
- **命名空间：**工作负载所在命名空间，此处不可修改。
- **关联工作负载：**要添加 Service 的工作负载，此处不可修改。
- **端口配置：**
  - 协议：请根据业务的协议类型选择。
  - 容器端口：工作负载程序实际监听的端口，需用户确定。nginx 程序实际监听的端口为 80。
  - 访问端口：容器端口映射到集群虚拟 IP 上的端口，用虚拟 IP 访问工作负载时使用，端口范围为 1-65535，可任意指定。

步骤 5 单击“创建”。工作负载已添加“集群内访问（ClusterIP）”的服务。

----结束

验证访问方式

步骤 1 在管理控制台首页，单击“计算 > 弹性云主机”。

步骤 2 在弹性云主机页面，找到同一 VPC 内任意一台云服务器，并确认连接到访问地址中 IP 与端口的安全组是开放的。

步骤 3 登录工作负载所在集群的任意节点。

步骤 4 使用 curl 命令访问工作负载验证工作负载是否可以正常访问。您可以通过 IP 或者域名的方式来验证。

方式一：通过 IP 地址验证。

curl *10.247.74.100:8080*

其中 10.247.74.100:8080 为步骤 3 中获取的访问地址。

回显如下表示工作负载可正常访问。

```
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
```

```
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

方式二：进入容器，在容器内通过域名验证。

curl *nginx.default.svc.cluster.local:8080*

其中 *nginx.default.svc.cluster.local* 为步骤 3 中获取的域名访问地址。

回显如下表示工作负载可正常访问。

```
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

----结束

更新 Service

您可以在添加完 Service 后，更新此 Service 的端口配置，操作步骤如下：

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 网络管理”，在 **Service** 页签下，选择对应的集群和命名空间，单击需要更新端口配置的 Service 后的“更新”。

步骤 2 在“更新 Service”页面，访问类型选择“集群内访问（ClusterIP）”。

步骤 3 更新集群内访问参数。

- **Service 名称：**您创建的 Service 名称，此处不可修改。
- **集群名称：**工作负载所在集群的名称，此处不可修改。
- **命名空间：**工作负载所在命名空间，此处不可修改。
- **关联工作负载：**要添加 Service 的工作负载，此处不可修改。
- **端口配置：**
  - 协议：请根据业务的协议类型选择。
  - 容器端口：工作负载程序实际监听的端口，需用户确定。nginx 程序实际监听的端口为 80。
  - 访问端口：容器端口映射到集群虚拟 IP 上的端口，用虚拟 IP 访问工作负载时使用，端口范围为 1-65535，可任意指定。

步骤 4 单击“更新”，工作负载已更新 Service。

----结束

通过 kubectl 命令行创建

您可以通过 kubectl 命令行设置 Service 访问方式。本节以 nginx 为例，说明 kubectl 命令实现集群内访问的方法。

前提条件

请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令，使弹性云主机连接集群。

操作步骤

步骤 1 登录已配置好 kubectl 命令的弹性云主机。

步骤 2 创建并编辑 nginx-deployment.yaml 和 nginx-clusterip-svc.yaml 文件。

其中，nginx-deployment.yaml 和 nginx-clusterip-svc.yaml 为自定义名称，您可以随意命名。

vi nginx-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
```

```
      app: nginx
  spec:
    containers:
    - image: nginx
      imagePullPolicy: Always
      name: nginx
    imagePullSecrets:
    - name: default-secret
```

vi nginx-ClusterIp-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: nginx
  name: nginx-clusterip
spec:
  ports:
  - name: service0
    port: 8080
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  type: ClusterIP
```

表8-1 关键参数说明

参数	是否必填	参数类型	描述
port	是	Integer	将由此服务公开的端口，对应界面上的访问端口。
targetPort	是	Integer	对应界面上的容器端口。
type	否	String	对应界面上的访问类型，必须是： • ClusterIP • NodePort • LoadBalancer 默认值：ClusterIP，表示“集群虚拟IP”。

步骤 3 创建工作负载。

kubectl create -f nginx-deployment.yaml

回显如下，表示工作负载已开始创建。

```
deployment "nginx" created
```

kubectl get po

回显如下，工作负载状态为 Running，表示工作负载已处于运行中状态。

```
NAME                    READY     STATUS             RESTARTS   AGE
etcd-0                  0/1       ImagePullBackOff   0          27m
icagent-m9dkt           0/0       Running            0          3d
nginx-2601814895-znhbr   1/1       Running           0          15s
```

步骤 4 创建服务。

kubectl create -f nginx-ClusterIp-svc.yaml

回显如下，表示服务已开始创建。

```
service "nginx-clusterip" created
```

kubectl get svc

回显如下，表示服务已创建成功，CLUSTER-IP 已生成。

```
NAME             TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
etcd-svc         ClusterIP   None             <none>        3120/TCP   30m
kubernetes       ClusterIP   10.247.0.1       <none>        443/TCP    3d
nginx-clusterip   ClusterIP   10.247.200.134   <none>        80/TCP     20s
```

步骤 5 登录工作负载所在集群的任意节点。

步骤 6 采用 curl 命令访问工作负载验证工作负载是否可以正常访问。您可以通过 IP 或者域名的方式来验证。

方式一：通过 IP 地址验证。

curl *10.247.200.134:80*

回显如下表示工作负载可正常访问。

```
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

方式二：通过域名验证。

curl *nginx*-**clusterip.default.svc.cluster.local:***8080*

回显如下表示工作负载可正常访问。

```
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

----结束

8.2.2 节点访问(NodePort)

操作场景

节点访问（NodePort）是指在每个节点的 IP 上开放一个静态端口，通过静态端口对外暴露服务。节点访问（NodePort）会路由到 ClusterIP 服务，这个 ClusterIP 服务会自动创建。通过请求 <NodeIP>:<NodePort>，可以从集群的外部访问一个 NodePort 服务。

约束与限制

- “节点访问（NodePort）”默认为 VPC 内网访问，如果需要使用弹性 IP 通过公网访问该服务，请提前在集群的节点上绑定弹性 IP。
- 创建 service 后，如果服务亲和从集群级别切换为节点级别，连接跟踪表将不会被清理，建议用户创建 service 后不要修改服务亲和属性，如需修改请重新创建 servcie。
- 同一个节点内的容器不支持访问 **externalTrafficPolicy** 为 local 的 service。

工作负载创建时设置

您可以在创建工作负载时通过控制台设置 Service 访问方式，本节以 nginx 为例进行说明。

步骤 1 参考 6.2 创建无状态负载(Deployment)、6.3 创建有状态负载(StatefulSet)或 6.4 创建守护进程集(DaemonSet)，在“工作负载访问设置”步骤，单击“添加服务”。

- **访问类型**：选择“节点访问（NodePort）”。

  说明

  如果需要使用弹性 IP 通过公网访问该服务，请提前在集群的节点上绑定弹性 IP。

- **Service 名称**：自定义服务名称，可与工作负载名称保持一致。
- **服务亲和**：
  - 集群级别：集群下所有节点的 IP+访问端口均可以访问到此服务关联的负载，服务访问会因路由跳转导致一定性能损失，且无法获取到客户端源 IP。
  - 节点级别：只有通过负载所在节点的 IP+访问端口才可以访问此服务关联的负载，服务访问没有因路由跳转导致的性能损失，且可以获取到客户端源 IP。
- **端口配置**：
  - 协议：请根据业务的协议类型选择。
  - 容器端口：容器镜像中工作负载实际监听的端口，取值范围为 1-65535。
  - 访问端口：容器端口映射到节点私有 IP 上的端口，建议选择“自动生成”。
    - 自动生成：系统会自动分配端口号。
    - 指定端口：指定固定的节点端口，默认取值范围为 30000-32767。若指定端口时，请确保同个集群内的端口唯一性。

步骤 2 完成配置后，单击“确定”。

步骤 3 单击“下一步：高级设置”进入高级设置页面，直接单击“创建”。

步骤 4 单击“查看工作负载详情”，在访问方式页签下获取访问地址，例如“192.168.0.160:30358”。

----结束

工作负载创建完成后设置

您可以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 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”，在工作负载列表页单击要设置 Service 的工作负载名称。

说明

如果当前 Service 被关联到 Ingress，则更新 Service 的端口信息后 Ingress 将不可用，需要删除重建。

步骤 2 在“Service”页签，单击“添加 Service”。

步骤 3 在“添加 Service”页面，访问类型选择“节点访问（NodePort）”。

说明

如果需要使用弹性 IP 通过公网访问该服务，请提前在集群的节点上绑定弹性 IP。

步骤 4 设置节点访问参数：

- **Service 名称：**自定义服务名称，可与工作负载名称保持一致。
- **集群名称：**工作负载所在集群的名称，此处不可修改。
- **命名空间：**工作负载所在命名空间，此处不可修改。
- **关联工作负载：**要添加 Service 的工作负载，此处不可修改。
- **服务亲和：**
  - 集群级别：集群下所有节点的 IP+访问端口均可以访问到此服务关联的负载，服务访问会因路由跳转导致一定性能损失，且无法获取到客户端源 IP。
  - 节点级别：只有通过负载所在节点的 IP+访问端口才可以访问此服务关联的负载，服务访问没有因路由跳转导致的性能损失，且可以获取到客户端源 IP。
- **端口配置：**
  - 协议：请根据业务的协议类型选择。
  - 容器端口：容器镜像中工作负载程序实际监听的端口，需用户确定。nginx 程序实际监听的端口为 80。
  - 访问端口：容器端口映射到节点私有 IP 上的端口，建议选择“自动生成”。
    - 自动生成：系统会自动分配端口号。
    - 指定端口：指定固定的节点端口，默认取值范围为 30000-32767。若指定端口时，请确保同个集群内的端口唯一性。

步骤 5 单击“创建”。工作负载已添加“节点访问（NodePort）”的服务。

----结束

验证访问方式

步骤 1 在管理控制台首页，单击“计算 > 弹性云主机”。

步骤 2 在弹性云主机页面，找到同一 VPC 内任意一台云服务器，并确认连接到访问地址中 IP 与端口的安全组是开放的。

步骤 3 单击“远程登录”，弹出登录页面，输入用户密码登录。

步骤 4 使用 curl 命令访问工作负载验证工作负载是否可以正常访问。

说明

- 节点访问(NodePort)会在集群内节点上分配一个虚拟 IP，即可以在集群内部通过虚拟 IP 的验证方式验证。其中，虚拟 IP 访问端口默认与容器端口一致。
- 如果需要使用弹性 IP 通过公网访问该服务，请提前在集群的节点上绑定弹性 IP。

curl *192.168.0.160:*30358

其中“192.168.0.160:30358”为步骤 4 中获取到的访问地址，即节点虚拟 IP+访问端口。

回显如下表示访问成功。

```
<html>
<head>
```

```
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

----结束

更新 Service

您可以在添加完 Service 后，更新此 Service 的端口配置，操作步骤如下：

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 网络管理”，在 Service 页签下，选择对应的集群和命名空间，单击需要更新端口配置的 Service 后的“更新”。

步骤 2 在更新 Service 页面，访问类型选择“节点访问（NodePort）”。

步骤 3 更新节点访问参数：

- **Service 名称：**您创建的 Service 名称，此处不可修改。
- **集群名称：**工作负载所在集群的名称，此处不可修改。
- **命名空间：**工作负载所在命名空间，此处不可修改。
- **关联工作负载：**要添加 Service 的工作负载，此处不可修改。
- **服务亲和：**
  - 集群级别：集群下所有节点的 IP+访问端口均可以访问到此服务关联的负载，服务访问会因路由跳转导致一定性能损失，且无法获取到客户端源 IP。
  - 节点级别：只有通过负载所在节点的 IP+访问端口才可以访问此服务关联的负载，服务访问没有因路由跳转导致的性能损失，且可以获取到客户端源 IP。
- **端口配置：**
  - 协议：请根据业务的协议类型选择。
  - 容器端口：容器镜像中工作负载程序实际监听的端口，需用户确定。nginx 程序实际监听的端口为 80。
  - 访问端口：容器端口映射到节点私有 IP 上的端口，建议选择“自动生成”。

- 自动生成：系统会自动分配端口号。
- 指定端口：指定固定的节点端口，默认取值范围为 30000-32767。若指定端口时，请确保同个集群内的端口唯一性。

步骤 4 单击“更新”。工作负载已更新 Service。

----结束

kubectl 命令行创建

您可以通过 kubectl 命令行设置 Service 访问方式。本节以 nginx 为例，说明 kubectl 命令实现节点访问的方法。

前提条件

请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令，使弹性云主机连接集群。

操作步骤

步骤 1 登录已配置好 kubectl 命令的弹性云主机。

步骤 2 创建并编辑 nginx-deployment.yaml 以及 nginx-nodeport-svc.yaml 文件。

其中，nginx-deployment.yaml 和 nginx-nodeport-svc.yaml 为自定义名称，您可以随意命名。

vi nginx-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        imagePullPolicy: Always
        name: nginx
      imagePullSecrets:
      - name: default-secret
```

vi nginx-nodeport-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
```

```
    app: nginx
  name: nginx-nodeport
spec:
  ports:
  - name: service
    nodePort: 30000
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  type: NodePort
```

表8-2 关键参数说明

参数	是否必填	参数类型	描述
nodePort	否	Integer	对应界面上的访问端口，取值范围为 30000 ~ 32767，不填写表示自动生成。
port	是	Integer	集群虚拟 IP 的访问端口，取值范围为 1 ~ 65535。
protocol	否	String	该端口的 IP 协议，支持“TCP”和“UDP”。 默认值：TCP
targetPort	是	String	对应界面上的容器端口，取值范围为 1 ~ 65535。
type	是	String	对应界面上的访问类型，NodePort 表示“节点私有 IP”。

表8-3 metadata.annotations 相关参数（对接 Ingress 需开启健康检查和会话保持，如下为其所需字段及说明）

参数	是否必填	参数类型	描述
kubernetes.io/elb.session-affinity-mode	否	String	负载均衡监听是基于 IP 地址的会话保持，即来自同一 IP 地址的访问请求转发到同一台后端服务器上。 • 不启用：不填写该参数。 • 开启会话保持：需增加该参数，取值“SOURCE_IP”，表示基于源 IP 地址。

参数	是否必填	参数类型	描述
kubernetes.io/elb.session-affinity-option	否	表 8-4 Object	七层负载均衡会话保持配置选项。
kubernetes.io/elb.lb-algorithm	否	String	默认值：“ROUND_ROBIN”，为后端云服务器组的负载均衡算法。 取值范围： • ROUND_ROBIN：加权轮询算法。 • LEAST_CONNECTIONS：加权最少连接算法。 • SOURCE_IP：源 IP 算法。 当该字段的取值为 SOURCE_IP 时，后端云服务器组绑定的后端云服务器的 weight 字段无效。
kubernetes.io/elb.health-check-flag	否	String	默认开启，为是否开启 ELB 健康检查功能。 • 开启：“”（空值）或“on” • 关闭：“off”
kubernetes.io/elb.health-check-option	否	表 8-5 Object	ELB 健康检查配置选项。

表8-4 elb.session-affinity-option 字段数据结构说明

参数	是否必填	参数类型	描述
persistence_timeout	否	String	当 elb.session-affinity-mode 是“HTTP_COOKIE”时生效，设置会话保持的超时时间（秒）。 默认值为：1440，取值范围：1-1440。
app_cookie_name	是	String	当 elb.session-affinity-mode 是“APP_COOKIE”时生效，设置会话保持的超时时间（秒）。 取值范围：1-10000 字符。

表8-5 elb.health-check-option 字段数据结构说明

参数	是否必填	参数类型	描述
delay	是	String	开始健康检查的初始等待时间（秒），可选 默认值：5，取值范围：1-50
timeout	是	String	健康检查的超时时间（秒），可选 默认值：10，取值范围 1-50
max_retries	是	String	健康检查的最大重试次数，可选 默认值：3，取值范围 1-10
protocol	是	String	健康检查的协议，可选 默认值：取关联服务的协议 取值范围：“TCP”或者“HTTP”
path	是	String	健康检查的 URL，协议是“HTTP”时配置，可选 默认值：“/” 取值范围：1-10000 字符

步骤 3 创建工作负载。

kubectl create -f nginx-deployment.yaml

回显如下，表示工作负载已开始创建。

```
deployment "nginx" created
```

kubectl get po

回显如下，工作负载状态为 Running，表示工作负载已处于运行状态。

```
NAME                     READY     STATUS             RESTARTS   AGE
etcd-0                   0/1       ImagePullBackOff   0          48m
icagent-m9dkt            0/0       Running            0          3d
nginx-2601814895-qhxqv   1/1       Running            0          9s
```

步骤 4 创建服务。

kubectl create -f nginx-nodeport-svc.yaml

回显如下，表示服务开始创建。

```
service "nginx-nodeport" created
```

kubectl get svc

回显如下，表示服务已创建完成。

```
NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
etcd-svc         ClusterIP   None           <none>       3120/TCP       49m
kubernetes       ClusterIP   10.247.0.1     <none>        443/TCP        3d
nginx-nodeport   NodePort    10.247.4.225   <none>        80:30000/TCP   7s
```

步骤 5 采用 curl 命令访问工作负载验证工作负载是否可以正常访问。

curl *192.168.2.240:30000*

其中 192.168.2.240 为集群中任意一个节点的 IP 地址，30000 为节点开放的端口号。

回显如下，表示可正常访问。

```
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

----结束

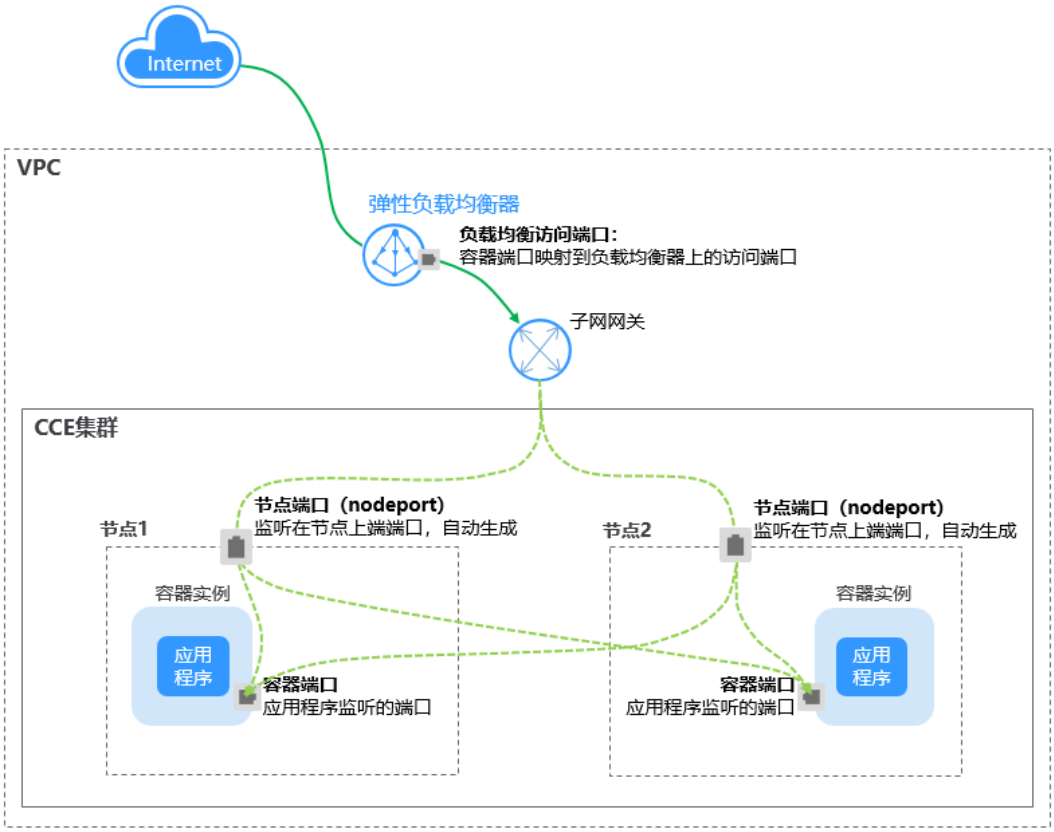
8.2.3 负载均衡(LoadBalancer)

操作场景

负载均衡(LoadBalancer)可以通过弹性负载均衡从公网访问到工作负载，与弹性 IP 方式相比提供了高可靠的保障，一般用于系统中需要暴露到公网的服务。

负载均衡访问方式由公网弹性负载均衡 ELB 服务地址以及设置的访问端口组成，例如“10.117.117.117:80”。

图8-4 负载均衡(LoadBalancer)



约束与限制

- CCE 中的负载均衡（LoadBalancer）访问类型使用弹性负载均衡 ELB 提供网络访问，存在如下产品约束：
  - 自动创建的 ELB 实例建议不要被其他资源使用，否则会在删除时被占用，导致资源残留。
  - 正在使用的 ELB 实例请不要修改监听器名称，否则可能导致无法正常访问。
- 创建 service 后，如果服务亲和从集群级别切换为节点级别，连接跟踪表将不会被清理，建议用户创建 service 后不要修改服务亲和属性，如需修改请重新创建 servcie。
- 容器内不支持访问 **externalTrafficPolicy** 为 local 的 service。

工作负载创建时设置

可以在创建工作负载时通过 CCE 控制台设置 Service 访问方式，本节以 nginx 为例进行说明。

步骤 1 参考 6.2 创建无状态负载(Deployment)、6.3 创建有状态负载(StatefulSet)或 6.4 创建守护进程集(DaemonSet)，在“工作负载访问设置”步骤，单击“添加服务”。

- **访问类型**：选择“负载均衡（LoadBalancer）”。
- **Service 名称**：自定义服务名称，可与工作负载名称保持一致。

- **服务亲和：**
  - 集群级别：集群下所有节点的 IP+访问端口均可以访问到此服务关联的负载，服务访问会因路由跳转导致一定性能损失，且无法获取到客户端源 IP。
  - 节点级别：只有通过负载所在节点的 IP+访问端口才可以访问此服务关联的负载，服务访问没有因路由跳转导致的性能损失，且可以获取到客户端源 IP。

负载均衡配置：

- **负载均衡：**请根据业务需求选择。

  根据业务需求选择“公网”或“私网”，详情请参见。
  - 公网：支持自动创建和使用已有负载均衡实例两种方式。
  - 私网：支持自动创建和使用已有负载均衡实例两种方式。

  负载均衡实例需与当前集群处于相同 VPC 且为相同公网或私网类型。
  - 规格配置：选择“公网 > 自动创建”时显示此配置项，单击

可修改负载均衡实例的名称、规格、计费模式和带宽。
  - 健康检查：默认开启。请根据界面提示进行配置，部分参数可参考 6.10.5 设置容器健康检查。
- **端口配置：**
  - 协议：请根据业务的协议类型选择。
  - 容器端口：容器镜像中工作负载实际监听端口，需用户确定。nginx 程序实际监听的端口为 80。
  - 访问端口：容器端口最终映射到负载均衡服务地址的端口，用负载均衡服务地址访问工作负载时使用，端口范围为 1-65535，可任意指定。

步骤 2 完成配置后，直接单击“确定”。

步骤 3 单击“下一步：高级配置”进入高级设置页面，直接单击“创建”。

步骤 4 创建成功后，单击“工作负载 > 无状态负载 Deployment”或“工作负载 > 有状态负载 StatefulSet”，在工作负载列表页面，单击“工作负载名称”进入工作负载详情页，在“访问方式”页签下，即可获取方式地址。

步骤 5 单击访问地址，即可跳转到访问页面。

----结束

工作负载创建完成后设置

您可以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 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”或“工作负载 > 有状态负载 StatefulSet”，在工作负载列表页单击要设置 Service 的工作负载名称。

步骤 2 在“Service”页签，单击“添加 Service”。

步骤 3 在“添加 service”页面，访问类型选择“**负载均衡（LoadBalancer）**”。

步骤 4 设置负载均衡参数。

- **Service 名称：**自定义服务名称，可与工作负载名称保持一致。
- **集群名称：**工作负载所在集群的名称，此处不可修改。
- **命名空间：**工作负载所在命名空间，此处不可修改。
- **关联工作负载：**要添加 Service 的工作负载，此处不可修改。
- **服务亲和：**
  - 集群级别：集群下所有节点的 IP+访问端口均可以访问到此服务关联的负载，服务访问会因路由跳转导致一定性能损失，且无法获取到客户端源 IP。
  - 节点级别：只有通过负载所在节点的 IP+访问端口才可以访问此服务关联的负载，服务访问没有因路由跳转导致的性能损失，且可以获取到客户端源 IP。

负载均衡配置：

- **负载均衡：**根据业务需求选择“公网”或“私网”。
  - 公网：支持自动创建和使用已有负载均衡实例两种方式。
  - 私网：支持自动创建和使用已有负载均衡实例两种方式。

  负载均衡实例需与当前集群处于相同 VPC 且为相同公网或私网类型。
  - 规格配置：选择“公网 > 自动创建”时显示此配置项，单击

 可修改负载均衡实例的名称、规格、计费模式和带宽。
  - 健康检查：默认开启。请根据界面提示进行配置，部分参数可参考 6.10.5 设置容器健康检查。
- **端口配置：**
  - 协议：请根据业务的协议类型选择。
  - 容器端口：容器镜像中工作负载程序实际监听的端口，需用户确定。nginx 程序实际监听的端口为 80。
  - 访问端口：容器端口最终映射到负载均衡服务地址的端口，用负载均衡服务地址访问工作负载时使用，端口范围为 1-65535，可任意指定。

步骤 5 单击“创建”。工作负载已添加“负载均衡（LoadBalancer）”的服务。

----结束

更新 Service

您可以在添加完 Service 后，更新此 Service 的端口配置。操作如下：

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 网络管理”，在 **Service** 页签下，选择对应的集群和命名空间，单击需要更新端口配置的 Service 的“更新”。

步骤 2 在“更新 Service”页面，访问类型选择“负载均衡（LoadBalancer）”。

步骤 3 更新负载均衡参数：

- **Service 名称：**您创建的 Service 名称，此处不可修改。
- **集群名称：**工作负载所在集群的名称，更新时此处不可修改。
- **命名空间：**工作负载所在命名空间，更新时此处不可修改。

- **关联工作负载：**要添加 Service 的工作负载，更新时此处不可修改。
- **服务亲和：**
  - 集群级别：集群下所有节点的 IP+访问端口均可以访问到此服务关联的负载，服务访问会因路由跳转导致一定性能损失，且无法获取到客户端源 IP。
  - 节点级别：只有通过负载所在节点的 IP+访问端口才可以访问此服务关联的负载，服务访问没有因路由跳转导致的性能损失，且可以获取到客户端源 IP。

负载均衡配置：

- **负载均衡：**更新时公网/私网在此处不可修改，但可以选择集群所在 VPC 下的其他负载均衡实例。

  负载均衡实例需与当前集群处于相同 VPC 且为相同公网或私网类型。
  - 健康检查：默认开启。请根据界面提示进行配置，部分参数请参考 6.10.5 设置容器健康检查。
- **端口配置：**
  - 协议：请根据业务的协议类型选择。
  - 容器端口：容器镜像中工作负载程序实际监听的端口，需用户确定。nginx 程序实际监听的端口为 80。
  - 访问端口：容器端口最终映射到负载均衡服务地址的端口，用负载均衡服务地址访问工作负载时使用，端口范围为 1-65535，可任意指定。

步骤 4 单击“更新”。工作负载已更新 Service。

----结束

通过 kubectl 命令行创建-使用已有共享型 ELB

您可以在创建工作负载时通过 kubectl 命令行设置 Service 访问方式。本节以 nginx 为例，说明 kubectl 命令实现负载均衡 (LoadBalancer)访问的方法。

前提条件

请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令，使弹性云主机连接集群。

操作步骤

步骤 1 登录已配置好 kubectl 命令的弹性云主机。

步骤 2 创建并编辑 nginx-deployment.yaml 以及 nginx-elb-svc.yaml 文件。

其中，nginx-deployment.yaml 和 nginx-elb-svc.yaml 为自定义名称，您可以随意命名。

vi nginx-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
```

```
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        imagePullPolicy: Always
        name: nginx
      imagePullSecrets:
      - name: default-secret
```

vi nginx-elb-svc.yaml

说明

若需要开启会话保持，需要满足如下条件：

- 工作负载协议为 TCP。
- 工作负载的各实例已设置反亲和部署，即所有的实例都部署在不同节点上。具体请参见 7.3.4 工作负载和节点的反亲和性。

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubernetes.io/elb.class: union
    kubernetes.io/session-affinity-mode: SOURCE_IP
    kubernetes.io/elb.id: 3c7caa5a-a641-4bff-801a-feace27424b6
    kubernetes.io/elb.subnet-id: 5083f225-9bf8-48fa-9c8b-67bd9693c4c0
  labels:
    app: nginx
  name: nginx
spec:
  loadBalancerIP: 10.78.42.242
  externalTrafficPolicy: Local
  ports:
  - name: service0
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
```

表8-6 关键参数说明

参数	是否必填	参数类型	描述
kubernetes.io/elb.class	否	String	请根据不同的应用场景和功能需求选择合适的负载均衡器类型。

参数	是否必填	参数类型	描述
			关键取值如下： • elasticity：经典型负载均衡。 • union：共享型负载均衡。 默认值：union
kubernetes.io/session-affinity-mode	否	String	负载均衡监听是基于 IP 地址的会话保持，即来自同一 IP 地址的访问请求转发到同一台后端服务器上。 • 不启用：不填写该参数。 • 开启会话保持：需增加该参数，取值“SOURCE_IP”，表示基于源 IP 地址。
kubernetes.io/elb.session-affinity-option	否	表 8-7 Object	ELB 会话保持配置选项，可设置会话保持的超时时间。
kubernetes.io/elb.id	是	String	为共享型负载均衡实例的 ID，取值范围：1-100 字符。 在关联已有 ELB 时：必填。 **获取方法：** 在控制台的“服务列表”中，单击“网络 > 弹性负载均衡 ELB”，单击 ELB 的名称，在 ELB 详情页的“基本信息”页签下找到“ID”字段复制即可。
kubernetes.io/elb.subnet-id	-	String	为子网的 ID，取值范围：1-100 字符。 • Kubernetes v1.11.7-r0 及以下版本的集群自动创建时：必填， • Kubernetes v1.11.7-r0 以上版本的集群：可不填。
kubernetes.io/elb.eip-id	-	String	为弹性公网 IP 地址的 ID，取值范围：1-100 字符。 非自动创建 elb 时：可选。 **获取方法：** 在控制台的“服务列表”中，单击“网络 > 弹性公网 IP EIP”，单击 EIP 的名称，在 EIP 详情页的“基本信息”中找到“ID”字段复制即可。
kubernetes.io/elb.enterpriseID	否	String	**v1.15 及以上版本的集群支持此字段，v1.15 以下版本默认创建到 default 项目下。** 为 ELB 企业项目 ID，选择后可以直接

参数	是否必填	参数类型	描述
			创建在具体的 ELB 企业项目下。 取值范围：1-100 字符。 **获取方法：** 登录控制台后，单击顶部菜单右侧的“企业 > 项目管理”，在打开的企业项目列表中单击要加入的企业项目名称，进入企业项目详情页，找到“ID”字段复制即可。
kubernetes.io/elb.lb-algorithm	否	String	默认值：“ROUND_ROBIN”，为后端云服务器组的负载均衡算法。 取值范围： • ROUND_ROBIN：加权轮询算法。 • LEAST_CONNECTIONS：加权最少连接算法。 • SOURCE_IP：源 IP 算法。 当该字段的取值为 SOURCE_IP 时，后端云服务器组绑定的后端云服务器的 weight 字段无效。
kubernetes.io/elb.health-check-flag	否	String	默认开启，为是否开启 ELB 健康检查功能。 • 开启：“（空值）”或“on” • 关闭：“off”
kubernetes.io/elb.health-check-option	否	表 8-8 Object	ELB 健康检查配置选项。
kubernetes.io/hws-hostNetwork	否	String	为标记工作负载服务是否使用主机网络模式。 默认是未使用主机网络，取值范围：“true”或者“false”
loadBalancerIP	否	String	配置为 ELB 的 IP 地址，私网 ELB 配置私有 IP，公网 ELB 配置为公网 IP。
externalTrafficPolicy	否	String	若需要开启会话保持，需增加该参数，表示请求转到固定节点；若某个服务发布成 ELB 服务且为 Local 模式，客户端如果在集群内，那么必须与服务端在同一个节点上才能正常访问。
port	是	Integer	集群虚拟 IP 的访问端口，也是注册到负载均衡上的端口。
targetPort	是	String	对应界面上的容器端口。

表8-7 elb.session-affinity-option 字段数据结构说明

参数	是否必填	参数类型	描述
persistence_timeout	是	String	当 elb.session-affinity-mode 是“SOURCE_IP”时生效，设置会话保持的超时时间（秒）。 默认值为：60，取值范围：1-60。

表8-8 elb.health-check-option 字段数据结构说明

参数	是否必填	参数类型	描述
delay	否	String	开始健康检查的初始等待时间（秒） 默认值：5，取值范围：1-50
timeout	否	String	健康检查的超时时间（秒） 默认值：10，取值范围 1-50
max_retries	否	String	健康检查的最大重试次数 默认值：3，取值范围 1-10
protocol	否	String	健康检查的协议 默认值：取关联服务的协议 取值范围：“TCP”、“UDP_CONNECT”或者“HTTP”
path	否	String	健康检查的 URL，协议是“HTTP”时配置 默认值：“/” 取值范围：1-10000 字符

步骤 3 创建工作负载。

kubectl create -f nginx-deployment.yaml

回显如下，表示工作负载已开始创建。

```
deployment "nginx" created
```

kubectl get po

回显如下，工作负载状态为 Running 状态，表示工作负载已运行中。

```
NAME                     READY     STATUS             RESTARTS   AGE
etcd-0                   0/1       ImagePullBackOff   0          1h
icagent-m9dkt            0/0       Running            0          3d
nginx-2601814895-c1xhw   1/1       Running            0          6s
```

步骤 4 创建服务。

kubectl create -f nginx-elb-svc.yaml

回显如下，表示服务已创建。

```
service "nginx" created
```

kubectl get svc

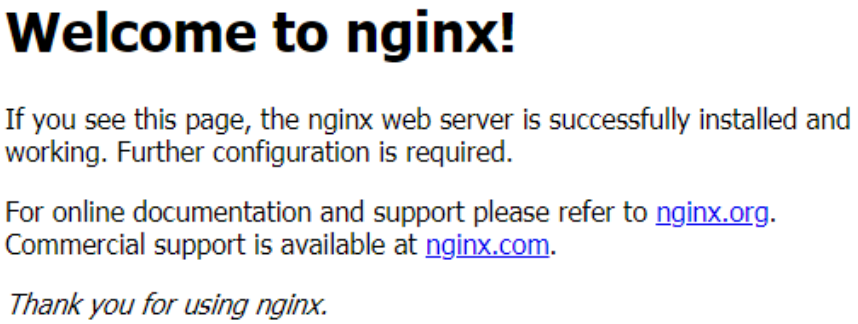
回显如下，表示工作负载访问方式已设置成功，工作负载可访问。

```
NAME         TYPE           CLUSTER-IP       EXTERNAL-IP    PORT(S)        AGE
etcd-svc     ClusterIP      None             <none>         3120/TCP       1h
kubernetes   ClusterIP      10.247.0.1       <none>         443/TCP        3d
nginx        LoadBalancer   10.247.130.196   10.78.42.242   80:31540/TCP   51s
```

步骤 5 在浏览器中输入访问地址，例如输入 10.78.42.242:80。10.78.42.242 为负载均衡实例 IP 地址，80 为对应界面上的访问端口。

可成功访问 nginx。

图8-5 通过负载均衡访问 nginx



----结束

通过 kubectl 命令行创建-自动创建共享型 ELB

您可以在创建工作负载时通过 kubectl 命令行设置 Service 访问方式。本节以 nginx 为例，说明 kubectl 命令实现负载均衡（LoadBalancer）访问的方法。

前提条件

请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令，使弹性云主机连接集群。

操作步骤

步骤 1 登录已配置好 kubectl 命令的弹性云主机。

步骤 2 创建并编辑 nginx-deployment.yaml 以及 nginx-elb-svc.yaml 文件。

其中，nginx-deployment.yaml 和 nginx-elb-svc.yaml 为自定义名称，您可以随意命名。

vi nginx-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        imagePullPolicy: Always
        name: nginx
      imagePullSecrets:
      - name: default-secret
```

vi nginx-elb-svc.yaml

说明

若需要开启会话保持，需要满足如下条件：

- 工作负载协议为 TCP。
- 工作负载的各实例已设置反亲和部署，即所有的实例都部署在不同节点上。具体请参见 7.3.4 工作负载和节点的反亲和性。

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubernetes.io/elb.class: union
    kubernetes.io/session-affinity-mode: SOURCE_IP
    kubernetes.io/elb.subnet-id: 5083f225-9bf8-48fa-9c8b-67bd9693c4c0
    kubernetes.io/elb.enterpriseID: debb7ae2-6d2f-4e6c-a0aa-1ccafd92b8eb
    kubernetes.io/elb.autocreate: '{"type":"public","bandwidth_name":"cce-bandwidth-
1551163379627","bandwidth_chargemode":"bandwidth","bandwidth_size":5,"bandwidth_sha
retype":"PER","eip_type":"5_bgp","name":"james"}'
  labels:
    app: nginx
  name: nginx
spec:
  externalTrafficPolicy: Local
  ports:
  - name: service0
```

```
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
```

表8-9 关键参数说明

参数	是否必填	参数类型	描述
kubernetes.io/elb.autocreate	是	表 8-10 object	自动创建 service 关联的 ELB **示例：** • 公网自动创建： 值为 '{"type":"public","bandwidth_name":"cce-bandwidth-1551163379627","bandwidth_chargemode":"bandwidth","bandwidth_size":5,"bandwidth_sharetype":"PER","eip_type":"5_bgp","name":"james"}' • 私网自动创建： 值为 '{"type":"inner", "name": "A-location-d-test"}'

表8-10 elb.autocreate 字段数据结构说明

参数	是否必填	参数类型	描述
name	否	String	自动创建的负载均衡的名称。 取值范围：1-64 个字符，小写字母，数字，下划线，小写字母开头，小写字母或者数字结尾。 默认名称：cce-lb+service.UID
type	否	String	负载均衡实例网络类型，公网或者私网。 • public：公网型负载均衡 • inner：私网型负载均衡 默认类型：inner
bandwidth_name	是	String	带宽的名称，默认值为：cce-bandwidth-******。 取值范围：1-64 个字符，小写字母，数字，下划线，小写字母开头，小写字

参数	是否必填	参数类型	描述
			母或者数字结尾。
bandwidth_chargemode	是	String	带宽付费模式。 • bandwidth：按带宽计费 • traffic：按流量计费
bandwidth_size	是	Integer	带宽大小，请根据 Region 带宽支持范围设置。
bandwidth_sharetype	是	String	带宽共享方式。 • PER：独享带宽 • WHOLE：共享带宽
eip_type	是	String	弹性公网 IP 类型，请参考 ELB 支持的弹性公网 IP 类型。
available_zone	是	Array of strings	负载均衡所在可用区。
l4_flavor_name	是	String	四层负载均衡实例名称。
l7_flavor_name	是	String	七层负载均衡实例名称。
ElbVirSubnetIDs	是	Array of strings	负载均衡后端所在子网，不填默认集群子网。不同实例规格将占用不同数量子网 IP，不建议使用其他资源（如集群，节点等）的子网网段。

步骤 3 创建工作负载。

kubectl create -f nginx-deployment.yaml

回显如下，表示工作负载已开始创建。

```
deployment "nginx" created
```

kubectl get po

回显如下，工作负载状态为 Running 状态，表示工作负载已运行中。

```
NAME                     READY     STATUS             RESTARTS   AGE
etcd-0                   0/1       ImagePullBackOff   0          1h
icagent-m9dkt            0/0       Running            0          3d
nginx-2601814895-c1xhw   1/1       Running            0          6s
```

步骤 4 创建服务。

kubectl create -f nginx-elb-svc.yaml

回显如下，表示服务已创建。

```
service "nginx" created
```

kubectl get svc

回显如下，表示工作负载访问方式已设置成功，工作负载可访问。

```
NAME         TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
etcd-svc     ClusterIP      None             <none>        3120/TCP       1h
kubernetes   ClusterIP      10.247.0.1       <none>        443/TCP        3d
nginx        LoadBalancer   10.247.130.196   10.78.42.242  80:31540/TCP   51s
```

步骤 5 在浏览器中输入访问地址，例如输入 10.78.42.242:80。10.78.42.242 为负载均衡实例 IP 地址，80 为对应界面上的访问端口。

可成功访问 nginx。

图8-6 通过负载均衡访问 nginx

----结束

8.3 Ingress

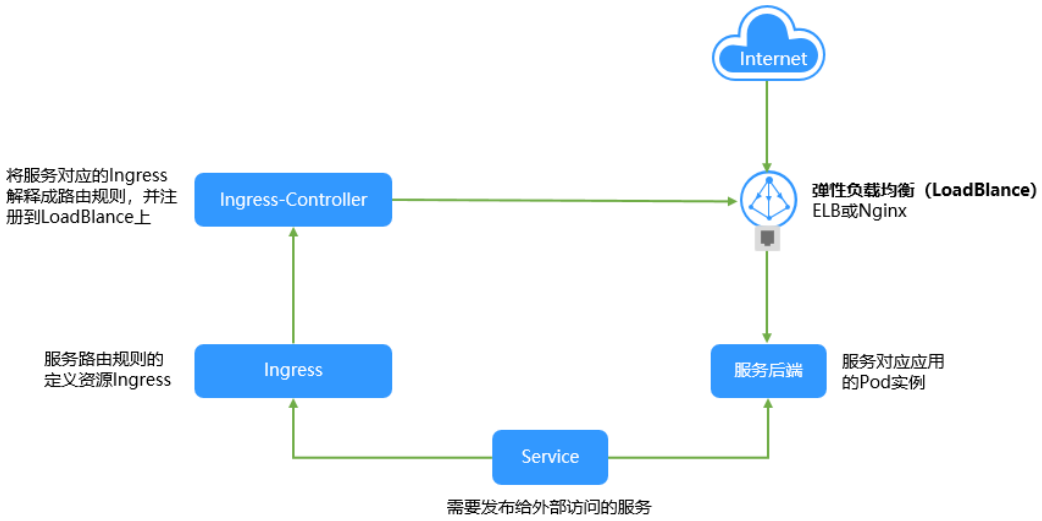
8.3.1 Ingress 概述

七层负载均衡（Ingress）是采用了共享型弹性负载均衡和独享型弹性负载均衡，在四层负载均衡访问方式的基础上支持了 URI 配置，通过对应的 URI 将访问流量分发到对应的服务。同时，服务根据不同 URI 实现不同的功能。

七层负载均衡访问方式由弹性负载均衡 ELB 服务地址、设置的访问端口、定义的 URI 组成，例如：10.117.117.117:80/helloworld。

通过配置公网类型和私网类型的负载均衡实例可以实现公网的七层路由转发和内网（同一 VPC 内）的七层路由转发。

图8-7 七层负载均衡(Ingress)



8.3.2 基本功能操作

前提条件

已在管理控制台中创建“弹性负载均衡”实例。

1. 登录管理控制台首页，在服务列表中选择“网络 > 弹性负载均衡”。
2. 单击右上角的“创建弹性负载均衡”。

说明

CCE 中的负载均衡（LoadBalancer）访问类型使用弹性负载均衡 ELB 提供网络访问，存在如下产品约束：

- 自动创建的 ELB 实例建议不要被其他资源使用，否则会在删除时被占用，导致资源残留。

创建 Ingress

您可以在创建工作负载时通过 CCE 控制台设置访问方式，本节以创建一个 nginx 工作负载并添加 Ingress 类型的 Service 为例进行说明。

步骤 1 创建工作负载，详细步骤请参见 6.2 创建无状态负载(Deployment)、6.3 创建有状态负载(StatefulSet)或 6.4 创建守护进程集(DaemonSet)。

- 若创建工作负载时，部署了“节点访问(NodePort)”类型的 Service，请直接执行步骤 3。
- 若创建工作负载时，未部署“节点访问(NodePort)”类型的 Service，请先执行步骤 2。

步骤 2 （可选）若创建工作负载时，未配置“节点访问（NodePort）”，请执行如下操作。

1. 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 网络管理”。

2. 在 Service 页签下，单击“添加 Service”。选择类型为“节点访问 (NodePort)”。
   - **Service 名称：**自定义服务名称，可与工作负载名称保持一致。
   - **集群名称：**选择需要添加 Service 的集群。
   - **命名空间：**选择需要添加 Service 的命名空间。
   - **关联工作负载：**单击“选择工作负载”，选择需要配置节点访问 (NodePort) 的工作负载名称，单击“确定”。
   - **服务亲和：**
     - 集群级别：集群下所有节点的 IP+访问端口均可以访问到此服务关联的负载，服务访问会因路由跳转导致一定性能损失，且无法获取到客户端源 IP。
     - 节点级别：只有通过负载所在节点的 IP+访问端口才可以访问此服务关联的负载，服务访问没有因路由跳转导致的性能损失，且可以获取到客户端源 IP。
   - **端口配置：**
     - 协议：请根据业务的协议类型选择。
     - 容器端口：容器镜像中工作负载实际监听的端口，需用户确定。nginx 程序实际监听的端口为 80。
     - 访问端口：容器端口映射到节点私有 IP 上的端口，用私有 IP 访问工作负载时使用，端口范围为 30000-32767，建议选择“自动生成”。
       - 自动生成：系统会自动分配端口号。
       - 指定端口：指定固定的节点端口，默认取值范围为 30000-32767。若指定端口时，请确保同个集群内的端口唯一性。
3. 单击“创建”，节点访问方式设置成功。

步骤 3 添加 Ingress 类型的 Service。

1. 单击 CCE 左侧导航栏的“资源管理 > 网络管理”。
2. 在 Ingress 页签下，单击“添加 Ingress”。
   - **Ingress 名称：**自定义 Ingress 名称，例如 ingress-demo。
   - **集群名称：**选择需要添加 Ingress 的集群。
   - **命名空间：**选择需要添加 Ingress 的命名空间。
   - **负载均衡配置：**Ingress 使用弹性负载均衡服务（ELB）的负载均衡器提供七层网络访问。

须知

1. ingress 创建后请在 CCE 页面对所选 ELB 实例进行配置升级和维护，否则可能导致 ingress 服务异常。
2. 自动创建的 ELB 实例尽可能不要被其他 ingress 或者 service 使用，否则可能导致资源残留。

负载均衡：负载均衡实例需与当前集群处于相同 VPC 且为相同公网或私网类型。

请根据业务需求选择“公网”或“私网”。

- 公网：支持自动创建和使用已有负载均衡实例两种方式。
  - 更改配置：选择“公网 > 自动创建”时，单击规格配置下的“更改配置”，可修改待创建的负载均衡实例的名称、规格、计费模式和带宽。
- 私网：支持自动创建和使用已有负载均衡实例两种方式。

- **监听器配置：**Ingress 为负载均衡器配置监听器，监听器对负载均衡器上的请求进行监听，并分发流量。

  开启“对接 Nginx”后将不显示该配置。

  - 对外协议：支持 HTTP 和 HTTPS。
  - 对外端口：开放在负载均衡服务地址的端口，可任意指定。

    若选择启用“对接 Nginx”后，将默认开启 80 和 443 端口，此处“对外端口”参数项将不显示。
  - 服务器证书：请选择密钥证书。

说明

- 选择 HTTPS 协议时，才需要创建密钥证书 ingress-test-secret.yaml。创建密钥的方法请参见 11.3 创建密钥。
- 同一个 ELB 实例的同一个端口配置 HTTPS 时，一个监听器只支持配置一个密钥证书。若使用两个不同的密钥证书将两个 Ingress 添加到同一个 ELB 下的同一个监听器，ELB 侧实际只生效最初的证书。

- **转发策略配置：**请求的访问地址与转发规则匹配时（转发规则由域名、URL 组成），此请求将被转发到对应的目标 Service 处理。

须知

服务负载均衡配置中可更改 ELB 实例的分配策略类型、会话保持、健康检查等内容，请勿在 ELB 服务修改相关信息。

- 域名：实际访问的域名地址。请确保所填写的域名已注册并备案，在 Ingress 创建完成后，将域名与自动创建的负载均衡实例的 IP（即 Ingress 访问地址的 IP 部分）绑定。一旦配置了域名规则，则必须使用域名访问。

- URL 匹配规则：
  - 前缀匹配：例如映射 URL 为/healthz，只要符合此前缀的 URL 均可访问。例如/healthz/v1，/healthz/v2。
  - 精确匹配：表示精准匹配，只有完全匹配上才能生效。例如映射 URL 为/healthz，则必须为此 URL 才能访问。
  - 正则匹配：可设定映射 URL 规范，例如规范为**/[A-Za-z0-9_.-]+/test**。只要符合此规则的 URL 均可访问，例如/abcA9/test，/v1-Ab/test。正则匹配规则支持 POSIX 与 Perl 两种标准。
- URL：需要注册的访问路径，例如：/healthz。
- 目标 Service：请选择已有 Service 或新建 Service。页面列表中仅支持选择“节点访问（NodePort）” 类型的 Service，该查询结果已自动过滤。
- Service 访问端口：可选择目标 Service 的访问端口。
- 服务负载均衡配置：该配置是基于服务的配置，若有多条路由使用当前服务，这些路由将使用相同的服务负载均衡配置。
  - 分配策略类型：可选择加权轮询算法、加权最少连接或源 IP 算法。

    加权轮询算法：根据后端服务器的权重，按顺序依次将请求分发给不同的服务器。它用相应的权重表示服务器的处理性能，按照权重的高低以及轮询方式将请求分配给各服务器，相同权重的服务器处理相同数目的连接数。常用于短连接服务，例如 HTTP 等服务。

    加权最少连接：最少连接是通过当前活跃的连接数来估计服务器负载情况的一种动态调度算法。加权最少连接就是在最少连接数的基础上，根据服务器的不同处理能力，给每个服务器分配不同的权重，使其能够接受相应权值数的服务请求。常用于长连接服务，例如数据库连接等服务。

    源 IP 算法：将请求的源 IP 地址进行 Hash 运算，得到一个具体的数值，同时对后端服务器进行编号，按照运算结果将请求分发到对应编号的服务器上。这可以使得对不同源 IP 的访问进行负载分发，同时使得同一个客户端 IP 的请求始终被派发至某特定的服务器。该方式适合负载均衡无 cookie 功能的 TCP 协议。
  - 会话保持：默认不启用。启用后需选择会话保持类型，设置会话保持时间。

    HTTP cookie：负载均衡器会根据客户端第一个请求生成一个 cookie，后续所有包含这个 cookie 值的请求都会由同一个后端服务器处理。

    应用程序 cookie：该选项依赖于后端应用。后端应用生成一个 cookie 值，后续所有包含这个 cookie 值的请求都会由同一个后端服务器处理。共享型负载均衡支持该会话保持类型，**独享型负载均衡暂不支持**。
  - 健康检查：默认不启用。请根据界面提示进行配置。
- 操作：可单击“删除”按钮删除该配置。

单击“添加转发策略”按钮可添加多条转发策略。

步骤 4 配置完成后，单击“创建”。

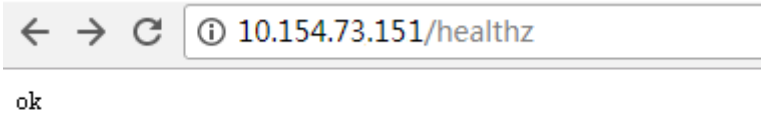
创建完成后，在 Ingress 列表可查看到已创建成功的 Ingress。

步骤 5 访问工作负载（例如名称为 defaultbackend）的“/healthz”接口。

方式一：负载均衡 IP 访问（需负载均衡访问的服务不能配置域名）

1. 获取 defaultbackend“/healthz”接口的访问地址，访问地址有负载均衡实例、对外端口、映射 URL 组成，例如：10.154.73.151:80/healthz。
   a. 在左侧导航栏选择“资源管理 > 网络管理”。
   b. 在“Ingress”页签下，单击 Ingress 名称，可查看访问地址。
2. 在浏览器中输入“/healthz”接口的访问地址，如：http://10.154.73.151:80/healthz，即可成功访问工作负载，如图 8-8。

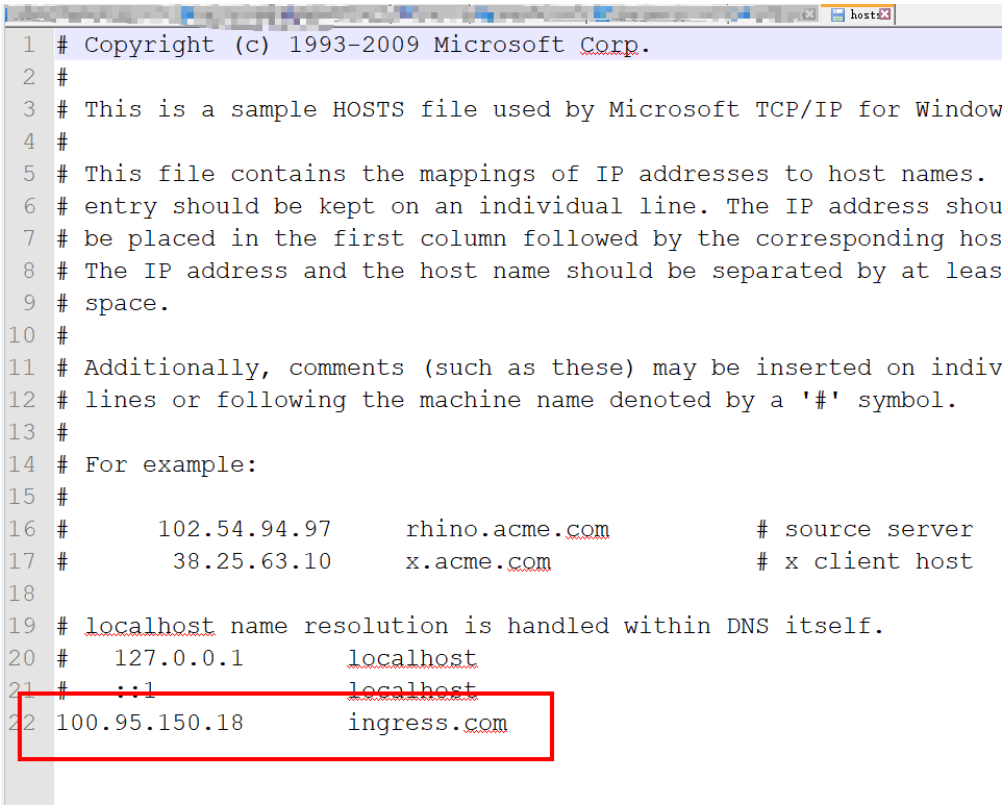
图8-8 访问 defaultbackend“/healthz”接口



方式二：域名访问

以 ingress 中已配置域名 ingress.com 为例。

1. 获取 ingress-demo 的域名与访问地址的 IP、URL 与端口。
   a. 在左侧导航栏选择“资源管理 > 网络管理”。
   b. 在“Ingress”页签下，可查看访问地址。
   c. 单击 Ingress 名称，可查看域名。
2. 在本地主机的 C:\Windows\System32\drivers\etc\hosts 中配置访问地址的 IP 和域名，如下图：



3. 在浏览器中输入 http://域名:访问地址端口/映射 url，如：http://ingress.com:81/james。

----结束

更新 Ingress

您可以在添加完 Ingress 后，更新此 Ingress 的端口、域名和路由配置。操作如下：

说明

- Ingress 创建后请避免在 ELB 服务的控制台中操作 Ingress 对接的 ELB 资源，如需修改请在 CCE 控制台中对所选 ELB 实例进行配置升级和维护，否则可能导致 Ingress 服务异常。
- 在 CCE 控制台中更新 Ingress 时，负载均衡配置中可更改 ELB 实例的分配策略类型、会话保持、健康检查等内容，请勿在 ELB 服务修改相关信息。
- 自动创建的 ELB 实例尽可能不要被其他 Ingress 或者 Service 使用，否则可能导致资源残留。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 网络管理”。在 **Ingress** 页签下，选择对应的集群和命名空间，单击待更新 Ingress 后方的“更新”。

步骤 2 在“更新 Ingress”页面，更新如下参数：

监听器配置

- **对外端口**：开放在负载均衡服务地址的端口，可任意指定。

转发策略配置

- **服务负载均衡配置**：单击 更改配置 展开后，可对如下参数进行设置：

- 分配策略类型：可选择加权轮询算法、加权最少连接或源 IP 算法。
  - 加权轮询算法：根据后端服务器的权重，按顺序依次将请求分发给不同的服务器。它用相应的权重表示服务器的处理性能，按照权重的高低以及轮询方式将请求分配给各服务器，相同权重的服务器处理相同数目的连接数。常用于短连接服务，例如 HTTP 等服务。
  - 加权最少连接：最少连接是通过当前活跃的连接数来估计服务器负载情况的一种动态调度算法。加权最少连接就是在最少连接数的基础上，根据服务器的不同处理能力，给每个服务器分配不同的权重，使其能够接受相应权值数的服务请求。常用于长连接服务，例如数据库连接等服务。
  - 源 IP 算法：将请求的源 IP 地址进行 Hash 运算，得到一个具体的数值，同时对后端服务器进行编号，按照运算结果将请求分发到对应编号的服务器上。这可以使得对不同源 IP 的访问进行负载分发，同时使得同一个客户端 IP 的请求始终被派发至某特定的服务器。该方式适合负载均衡无 cookie 功能的 TCP 协议。
- 会话保持：默认不启用。启用后需选择会话保持类型，设置会话保持时间。
  - HTTP cookie：负载均衡器会根据客户端第一个请求生成一个 cookie，后续所有包含这个 cookie 值的请求都会由同一个后端服务器处理。
  - 应用程序 cookie：该选项依赖于后端应用。后端应用生成一个 cookie 值，后续所有包含这个 cookie 值的请求都会由同一个后端服务器处理。
- 健康检查：默认开启。请根据界面提示进行配置。

- 单击“添加转发策略”按钮可添加多条转发策略。
  - 域名：实际访问的域名地址。请确保所填写的域名已注册并备案，在 Ingress 创建完成后，将域名与自动创建的负载均衡实例的 IP（即 Ingress 访问地址的 IP 部分）绑定。一旦配置了域名规则，则必须使用域名访问。
  - URL 匹配规则：
    - 前缀匹配：例如映射 URL 为/healthz，只要符合此前缀的 URL 均可访问。例如/healthz/v1，/healthz/v2。
    - 精确匹配：表示精准匹配，只有完全匹配上才能生效。例如映射 URL 为/healthz，则必须为此 URL 才能访问。
    - 正则匹配：可设定映射 URL 规范，例如规范为**/[A-Za-z0-9_.-]+/test**。只要符合此规则的 URL 均可访问，例如/abcA9/test，/v1-Ab/test。正则匹配规则支持 POSIX 与 Perl 两种标准。
  - URL：需要注册的访问路径，例如：/healthz。
  - 目标 Service：请选择已有 Service 或新建 Service。页面列表中仅支持选择“节点访问（NodePort）” 类型的 Service，该查询结果已自动过滤。Service 访问端口：可选择目标 Service 的访问端口。
  - 服务负载均衡配置：该配置是基于服务的配置，若有多条路由使用当前服务，这些路由将使用相同的服务负载均衡配置。详细配置请参见服务负载均衡配置。
  - 操作：可单击“删除”按钮删除该配置。

步骤 3 单击“提交”。工作负载已更新 Ingress。

----结束

相关操作

由于社区 Ingress 结构体中没有 property 属性，用户使用 client-go 调用创建 ingress 的 api 接口时，创建的 Ingress 中没有 property 属性。为了与社区的 client-go 兼容，CCE 提供了相关解决方案，具体请参见《常见问题》中的 Ingress 中的 property 字段如何实现与社区 client-go 兼容。

8.3.3 通过 Kubectl 命令行创建 Ingress

操作场景

本节以 nginx 为例，说明 kubectl 命令实现 ingress 访问的方法。

前提条件

请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令，使弹性云主机连接集群。

创建 Ingress-对接已有 ELB

步骤 1 登录已配置好 kubectl 命令的弹性云主机。

步骤 2 创建 ingress-test-deployment.yaml、ingress-test-svc.yaml、ingress-test-ingress.yaml 以及 ingress-test-secret.yaml 文件。

ingress-test-deployment.yaml、ingress-test-svc.yaml、ingress-test-ingress.yaml、ingress-test-secret.yaml 为自定义名称，您可以随意命名。

说明

- 选择 HTTPS 协议时，才需要创建密钥证书 ingress-test-secret.yaml。创建密钥的方法请参见 11.3 创建密钥。
- 同一个 ELB 实例的同一个端口配置 HTTPS 时，选择的证书需要是一样的。

vi ingress-test-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ingress-test-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: ingress-test-deployment
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: ingress-test-deployment
    spec:
      containers:
        #第三方公开镜像，可以参见描述获取地址，也可以使用自己的镜像
```

```
      - image: nginx
        imagePullPolicy: Always
        name: nginx
      imagePullSecrets:
      - name: default-secret
```

vi ingress-test-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: ingress-test-svc
  name: ingress-test-svc
spec:
  ports:
  - name: service0
    port: 8888
    protocol: TCP
    targetPort: 8888       #若需要设置多个端口，可依次填写，如下展示
  - name: service1
    port: 8081
    protocol: TCP
    targetPort: 8081
  selector:
    app: ingress-test-deployment
  type:  NodePort
```

表8-11 关键参数说明

参数	是否必填	参数类型	描述
port	是	Integer	集群虚拟 IP 的访问端口，取值范围为 1 ~ 65535。
protocol	是	String	该端口的 IP 协议，支持“TCP”和“UDP”。
targetPort	是	String	对应界面上的容器端口，应用程序实际监听的端口，取值范围为 1 ~ 65535。
type	是	String	采用 Nodeport 的访问类型连接负载均衡，NodePort 表示“节点私有 IP”。

vi ingress-test-ingress.yaml

- 1.15 及以上集群版本中的 apiVersion 为: networking.k8s.io/v1beta1
- 1.13 及以下集群版本中的 apiVersion 为: extensions/v1beta1

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/elb.id: f7891f9a-49f2-4ee2-b1ae-f019cd84eb4f
```

```
    kubernetes.io/elb.ip: 192.168.0.39
    kubernetes.io/elb.subnet-id: 29a0567e-96f1-4227-91cc-64f54d0b064d
    kubernetes.io/elb.port: "80"
    kubernetes.io/ingress.class: cce
    kubernetes.io/elb.tls-ciphers-policy: tls-1-2
  name: ingress-test-ingress
spec:
  tls:
  - secretName: ingress-test-secret
  rules:
  - http:
      paths:
      - backend:
          serviceName: ingress-test-svc
          servicePort: 8888
        property:
          ingress.beta.kubernetes.io/url-match-mode: EQUAL_TO
        path: "/healthz"
    host: ingress.com
```

说明

安全策略选择（kubernetes.io/elb.tls-ciphers-policy）仅在 1.17.11 及以上版本的集群中支持。

表8-12 关键参数说明

参数	是否必填	参数类型	描述
kubernetes.io/elb.id	是	String	为共享型负载均衡实例的 ID，取值范围：1-100 字符。 **获取方法：** 在控制台的“服务列表”中，单击“网络 > 弹性负载均衡 ELB”，单击 ELB 的名称，在 ELB 详情页的“基本信息”页签下找到“ID”字段复制即可。
kubernetes.io/elb.ip	是	String	为共享型负载均衡实例的服务地址，公网 ELB 配置为公网 IP，私网 ELB 配置为私网 IP。
kubernetes.io/elb.subnet-id	-	String	为子网的 ID，取值范围：1-100 字符。 • Kubernetes v1.11.7-r0 及以下版本的集群自动创建时：必填， • Kubernetes v1.11.7-r0 以上版本的集群：可不填。
kubernetes.io/elb.eip-id	是	String	为弹性公网的 ID，取值范围：1-100 字符。 **获取方法：** 在控制台的“服务列表”中，单击“网络 > 弹性公网 IP EIP”，单击 EIP 的

参数	是否必填	参数类型	描述
			名称，在 EIP 详情页的“基本信息”中找到“ID”字段复制即可。
kubernetes.io/elb.enterpriseID	否	String	**v1.15 及以上版本的集群支持此字段，v1.15 以下版本默认创建到 default 项目下。** ELB 企业项目 ID，选择后可以直接创建在具体的 ELB 企业项目下。 取值范围：1-100 字符。 **获取方法：** 登录控制台后，单击顶部菜单右侧的“企业 > 项目管理”，在打开的企业项目列表中单击要加入的企业项目名称，进入企业项目详情页，找到“ID”字段复制即可。
kubernetes.io/elb.tls-ciphers-policy	否	String	默认值：“tls-1-2”，为监听器使用的安全策略，仅在 HTTPS 协议下生效。 取值范围： • tls-1-0-inherit • tls-1-0 • tls-1- • tls-1-2 • tls-1-2-strict 各安全策略使用的加密套件列表详细参见表 8-13。
kubernetes.io/elb.port	是	Integer	界面上的对外端口，为注册到负载均衡服务地址上的端口。 取值范围：1-65535。
kubernetes.io/ingress.class	是	String	是否开启 nginx-ingress 插件。 • cce：表示不开启 nginx-ingress 插件，将默认使用共享型负载均衡实例。 • nginx：表示开启 nginx-ingress 插件。 通过 API 接口创建 Ingress 时必须增加该参数。
tls	否	Array of strings	HTTPS 协议时，需添加此参数。
secretName	否	String	HTTPS 协议时添加，配置为创建的密钥证书名称。

参数	是否必填	参数类型	描述
serviceName	是	String	为 ingress-test-svc.yaml 的服务名称。
servicePort	是	Integer	为 ingress-test-svc.yaml 的 Port。
ingress.beta.kubernetes.io/url-match-mode	否	String	路由匹配策略。 默认值：STARTS_WITH(前缀匹配)。 取值范围： • EQUAL_TO：精确匹配 • STARTS_WITH：前缀匹配 • REGEX：正则匹配
path	是	String	为路由路径，用户自定义设置。
host	否	String	为域名配置，默认为空，域名全匹配。

表8-13 tls_ciphers_policy 取值说明

安全策略	支持的 TLS 版本类型	使用的加密套件列表
tls-1-0-inherit	TLSv1.2 TLSv1.1 TLSv1	ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:AES128-GCM-SHA256:AES256-GCM-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:AES128-SHA256:AES256-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES128-SHA:DHE-RSA-AES128-SHA:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:AES128-SHA:AES256-SHA:DHE-DSS-AES128-SHA:CAMELLIA128-SHA:EDH-RSA-DES-CBC3-SHA:DES-CBC3-SHA:ECDHE-RSA-RC4-SHA:RC4-SHA:DHE-RSA-AES256-SHA:DHE-DSS-AES256-SHA:DHE-RSA-CAMELLIA256-SHA:DHE-DSS-CAMELLIA256-SHA:CAMELLIA256-SHA:EDH-DSS-DES-CBC3-SHA:DHE-RSA-CAMELLIA128-SHA:DHE-DSS-CAMELLIA128-SHA
tls-1-0	TLS1v1.2 TLSv1.1 TLSv1	ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:AES128-GCM-SHA256:AES256-GCM-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-
tls1-1	TLSv1.2 TLSv1.1	
tls-1-2	TLSv1.2	

安全策略	支持的 TLS 版本类型	使用的加密套件列表
		SHA256:AES128-SHA256:AES256-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES128-SHA:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:AES128-SHA:AES256-SHA
tls-1-2-strict	TLSv1.2	ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:AES128-GCM-SHA256:AES256-GCM-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:AES128-SHA256:AES256-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384

vi ingress-test-secret.yaml

```
apiVersion: v1
data:
  tls.crt: LS0******tLS0tCg==
  tls.key: LS0tL******0tLS0K
kind: Secret
metadata:
  annotations:
    description: test for ingressTLS secrets
  name: ingress-test-secret
  namespace: default
type: IngressTLS
```

说明

此处 tls.crt 和 tls.key 为示例，请获取真实密钥进行替换。tls.crt 和 tls.key 的值为 Base64 加密后的内容。

步骤 3 创建工作负载。

kubectl create -f ingress-test-deployment.yaml

回显如下，表明工作负载已创建。

```
deployment/ingress-test-deployment created
```

kubectl get po

回显如下，表明工作负载创建成功。

```
NAME                                READY   STATUS    RESTARTS   AGE
ingress-test-1627801589-r64pk   1/1     Running   0          6s
```

步骤 4 创建密钥。

kubectl create -f ingress-test-secret.yaml

回显如下，表明密钥已创建。

```
secret/ingress-test-secret created
```

kubectl get secrets

回显如下，表明密钥创建成功。

```
NAME                       TYPE                                  DATA      AGE
dash-dashboard             Opaque                                0         7d
dash-dashboard-token-f2nbk   kubernetes.io/service-account-token   3         7d
default-secret             kubernetes.io/dockerconfigjson        1         8d
default-token-wfn4l          kubernetes.io/service-account-token   3         8d
paas.elb                   cfe/secure-opaque                     2         8d
ingress-test-secret          IngressTLS                            2         13s
```

步骤 5 创建服务。

kubectl create -f ingress-test-svc.yaml

回显如下，表示服务已创建。

```
service/ingress-test-svc created
```

kubectl get svc

回显如下，表示服务创建成功。

```
NAME           TYPE         CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
ingress-test    NodePort     10.247.189.207    <none>       8080:30532/TCP   5s
kubernetes     ClusterIP    10.247.0.1       <none>        443/TCP          3d
```

kubectl create -f ingress-test-ingress.yaml

回显如下，表示 ingress 服务已创建。

```
ingress/ingress-test-ingress created
```

kubectl get ingress

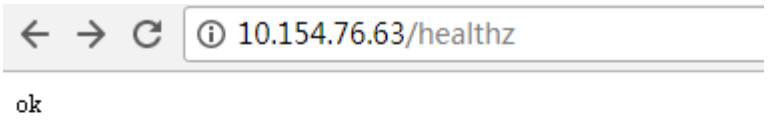
回显如下，表示 ingress 服务创建成功，工作负载可访问。

```
NAME           HOSTS     ADDRESS         PORTS   AGE
ingress-test     *        10.154.76.63     80      10s
```

步骤 6 在浏览器中输入访问地址 http://10.154.76.63/healthz。

其中，10.154.76.63 为统一负载均衡实例的 IP 地址。

图8-9 访问 healthz



----结束

创建 Ingress-自动创建 ELB

步骤 1 登录已配置好 kubectl 命令的弹性云主机。

步骤 2 创建 ingress-test-deployment.yaml、ingress-test-svc.yaml、ingress-test-ingress.yaml 以及 ingress-test-secret.yaml 文件。

ingress-test-deployment.yaml、ingress-test-svc.yaml、ingress-test-ingress.yaml、ingress-test-secret.yaml 为自定义名称，您可以随意命名。

说明

- 选择 HTTPS 协议时，才需要创建密钥证书 ingress-test-secret.yaml。创建密钥的方法请参见 11.3 创建密钥。
- 同一个 ELB 实例的同一个端口配置 HTTPS 时，选择的证书需要是一样的。

vi ingress-test-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ingress-test-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: ingress-test-deployment
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: ingress-test-deployment
    spec:
      containers:
        #第三方公开镜像，可以参见描述获取地址，也可以使用自己的镜像
      - image: nginx
        imagePullPolicy: Always
```

```
      name: nginx
    imagePullSecrets:
    - name: default-secret
```

vi ingress-test-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: ingress-test-svc
  name: ingress-test-svc
spec:
  ports:
  - name: service0
    port: 8888
    protocol: TCP
    targetPort: 8888      #若需要设置多个端口，可依次填写，如下展示
  - name: service1
    port: 8081
    protocol: TCP
    targetPort: 8081
  selector:
    app: ingress-test-deployment
  type:  NodePort
```

表8-14 关键参数说明

参数	是否必填	参数类型	描述
port	是	Integer	集群虚拟 IP 的访问端口，取值范围为 1 ~ 65535。
protocol	是	String	该端口的 IP 协议，支持“TCP”和“UDP”。
targetPort	是	String	对应界面上的容器端口，应用程序实际监听的端口，取值范围为 1 ~ 65535。
type	是	String	采用 Nodeport 的访问类型连接负载均衡，NodePort 表示“节点私有 IP”。

vi ingress-test-ingress.yaml

- 1.15 及以上集群版本中的 apiVersion 为: networking.k8s.io/v1beta1
- 1.13 及以下集群版本中的 apiVersion 为: extensions/v1beta1

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/elb.subnet-id: 29a0567e-96f1-4227-91cc-64f54d0b064d
    kubernetes.io/elb.enterpriseID: f6b9fffc-a9b7-4a50-a25e-364f587abe44
    kubernetes.io/elb.autocreate: '{"type":"public","bandwidth_name":"cce-bandwidth-
```

```
1551163379627","bandwidth_chargemode":"bandwidth","bandwidth_size":5,"bandwidth_sha
retype":"PER","eip_type":"5_bgp","name":"james"}'
    kubernetes.io/elb.tls-ciphers-policy: tls-1-2
    kubernetes.io/elb.port: "80"
    kubernetes.io/ingress.class: cce
  name: ingress-test-ingress
spec:
  tls:
  - secretName: ingress-test-secret
  rules:
  - http:
      paths:
      - backend:
          serviceName: ingress-test-svc
          servicePort: 8888
        property:
          ingress.beta.kubernetes.io/url-match-mode: EQUAL_TO
        path: "/healthz"
    host: ingress.com
```

表8-15 关键参数说明

参数	是否必填	参数类型	描述
kubernetes.io/elb.autocreate	是	表 8-16 object	自动创建 Ingress 关联的 ELB **示例：** • 公网自动创建： 值为 '{"type":"public","bandwidth_name":"cce-bandwidth-1551163379627","bandwidth_chargemode":"bandwidth","bandwidth_size":5,"bandwidth_sharetype":"PER","eip_type":"5_bgp","name":"james"}' • 私网自动创建： 值为 '{"type":"inner", "name": "A-location-d-test"}'

表8-16 elb.autocreate 字段数据结构说明

参数	是否必填	参数类型	描述
name	否	String	自动创建的负载均衡的名称。 取值范围：1-64 个字符，小写字母，数字，下划线，小写字母开头，小写字母或者数字结尾。 默认值：cce-lb+ingress.UID

参数	是否必填	参数类型	描述
type	否	String	负载均衡实例网络类型，公网或者私网。 • public：公网型负载均衡 • inner：私网型负载均衡 默认值：inner
bandwidth_name	是	String	带宽的名称，默认值为：cce-bandwidth-******。 取值范围：1-64 个字符，小写字母，数字，下划线，小写字母开头，小写字母或者数字结尾。
bandwidth_chargemode	是	String	带宽付费模式。 • bandwidth：按带宽计费 • traffic：按流量计费
bandwidth_size	是	Integer	带宽大小，请根据 Region 带宽支持范围设置。
bandwidth_sharetype	是	String	带宽类型，标识是否是共享带宽。 取值范围： • WHOLE：共享带宽 • PER：独享带宽
eip_type	是	String	弹性公网 IP 类型，请参考 ELB 支持的弹性公网 IP 类型。
available_zone	是	Array of strings	负载均衡所在可用区，必填项。
l4_flavor_name	否	String	四层负载均衡实例名称。
l7_flavor_name	是	String	七层负载均衡实例名称，必填项。
ElbVirSubnetIDs	否	Array of strings	负载均衡后端所在子网，不填默认集群子网。不同实例规格将占用不同数量子网 IP，不建议使用其他资源（如集群，节点等）的子网网段。 默认值：集群所在子网

vi ingress-test-secret.yaml

```
apiVersion: v1
data:
  tls.crt: LS0******tLS0tCg==
  tls.key: LS0tL******0tLS0K
kind: Secret
```

```
metadata:
  annotations:
    description: test for ingressTLS secrets
  name: ingress-test-secret
  namespace: default
type: IngressTLS
```

说明

此处 tls.crt 和 tls.key 为示例，请获取真实密钥进行替换。tls.crt 和 tls.key 的值为 Base64 加密后的内容。

步骤 3 创建工作负载。

kubectl create -f ingress-test-deployment.yaml

回显如下，表明工作负载已创建。

```
deployment/ingress-test-deployment created
```

kubectl get po

回显如下，表明工作负载创建成功。

```
NAME                              READY     STATUS            RESTARTS   AGE
ingress-test-1627801589-r64pk   1/1       Running           0          6s
```

步骤 4 创建密钥。

kubectl create -f ingress-test-secret.yaml

回显如下，表明密钥已创建。

```
secret/ingress-test-secret created
```

kubectl get secrets

回显如下，表明密钥创建成功。

```
NAME                       TYPE                                  DATA      AGE
dash-dashboard             Opaque                                0         7d
dash-dashboard-token-f2nbk   kubernetes.io/service-account-token   3         7d
default-secret             kubernetes.io/dockerconfigjson        1         8d
default-token-wfn4l        kubernetes.io/service-account-token   3         8d
paas.elb                   cfe/secure-opaque                     2         8d
ingress-test-secret        IngressTLS                            2         13s
```

步骤 5 创建服务。

kubectl create -f ingress-test-svc.yaml

回显如下，表示服务已创建。

```
service/ingress-test-svc created
```

kubectl get svc

回显如下，表示服务创建成功。

```
NAME           TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
ingress-test   NodePort    10.247.189.207   <none>        8080:30532/TCP   5s
kubernetes     ClusterIP   10.247.0.1       <none>        443/TCP          3d
```

kubectl create -f ingress-test-ingress.yaml

回显如下，表示 ingress 服务已创建。

```
ingress/ingress-test-ingress created
```

kubectl get ingress

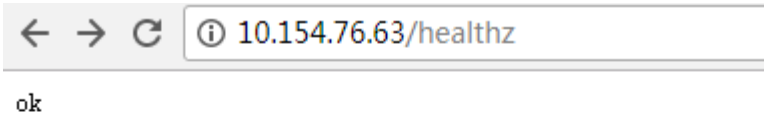
回显如下，表示 ingress 服务创建成功，工作负载可访问。

```
NAME           HOSTS     ADDRESS         PORTS   AGE
ingress-test     *        10.154.76.63     80      10s
```

步骤 6 在浏览器中输入访问地址 http://10.154.76.63/healthz。

其中，10.154.76.63 为统一负载均衡实例的 IP 地址。

图8-10 访问 healthz



----结束

8.4 网络模型

8.4.1 网络模型概述

网络模型为集群内 Pod 分配 IP 地址并提供网络服务，CCE 支持容器隧道网络、VPC 网络，你可在创建集群时进行选择。

关于网络模型的详细介绍，请参见：

- 8.4.2 容器隧道网络
- 8.4.3 VPC 网络

选择网络模型

表 8-17 主要介绍 CCE 所支持的网络模型，您可根据实际业务需求进行选择。

⚠ 注意

集群创建成功后，网络模型不可更改，请谨慎选择。

表8-17 网络模型对比

对比维度	容器隧道网络	VPC 网络
数据面依赖	OVS	IPVlan，VPC 路由
适用集群	CCE 集群 虚机集群	CCE 集群 虚机集群
是否支持网络策略 （networkpolicy）	是	否
是否支持 ENI	否	是，容器网络与 VPC 网络深度整合，Pod 内直接使用 VPC 弹性网卡。
IP 地址管理	IP 地址可迁移	• 每个节点分配一个小子网。 • 在 VPC Router 上添加静态路由，下一跳为节点 IP。
网络性能	基于 vxlan 隧道封装，有性能损耗。	• 无隧道封装，性能好，媲美主机网络。 • 跨节点通过 VPC Router 转发。
组网规模	最大可支持 2000 节点	受限 VPC 路由表能力。
外部依赖	无	依赖 VPC Router 静态路由表能力。
适用场景	• 一般容器业务场景。 • 对网络时延、带宽要求不是特别高的场景。	• 对网络时延、带宽要求高。 • 容器与虚机 IP 互通，使用了微服务注册框架的，如 Dubbo、CSE 等。

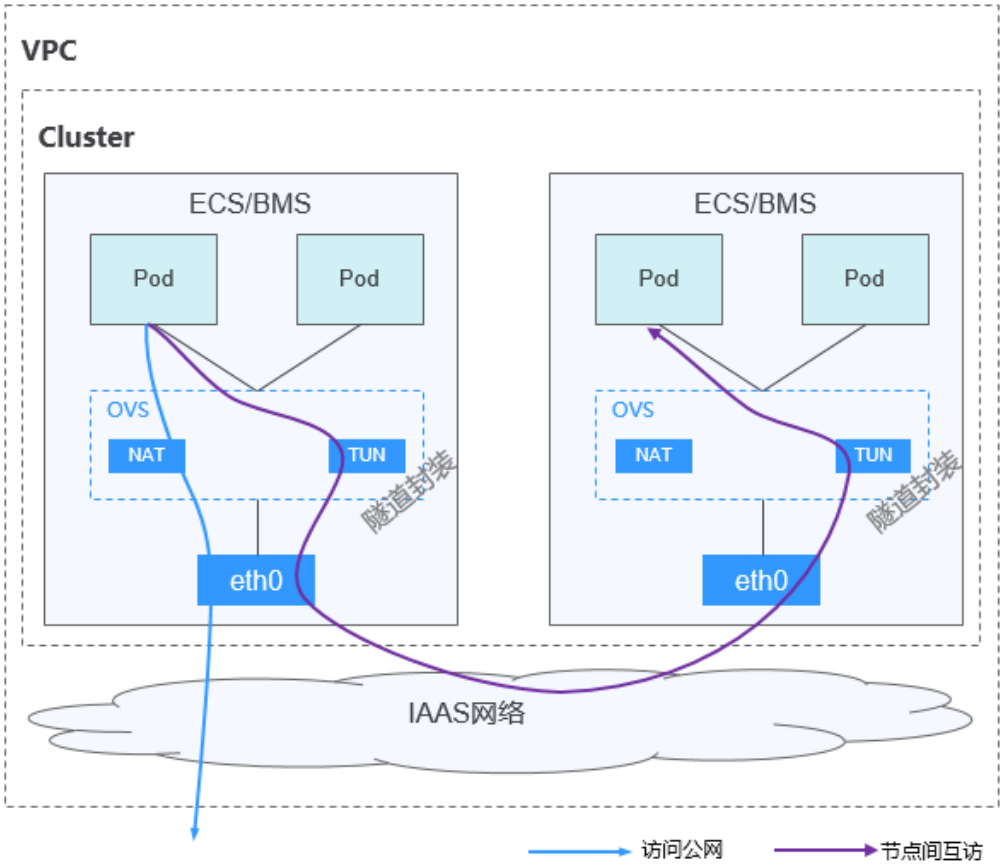
须知

1. VPC 网络集群实际支持规模受限于 VPC 的路由表路由条目配额，创建前请提前评估集群规模。
2. VPC 网络模型默认支持容器与同一 VPC 的虚拟机直接互访，与其他 VPC 的主机在配置对等连接策略后可以支持直接互访。此外，云专线/VPN 等混合组网场景在合理规划后可以支持对端直接与容器互访。

8.4.2 容器隧道网络

容器隧道网络（Overlay）基于底层 VPC 网络，另构建了独立的 VXLAN 隧道化容器网络，适用于一般场景。VXLAN 是将以太网报文封装成 UDP 报文进行隧道传输。容器网络是承载于 VPC 网络之上的 Overlay 网络平面，具有付出少量隧道封装性能损耗，即可获得通用性强、互通性强、高级特性支持全面（例如 Network Policy 网络隔离）的优势，可以满足大多数应用需求。

图8-11 容器隧道网络



说明如下：

Overlay L2 指的是容器跨节点以 Overlay 通信，所有节点上的容器 IP 都是 2 层可达的。Canal 支持的 Overlay 类型为 vxlan，采用的 vswitch 为 openvswitch。

Overlay L2 特性支持的功能包括：

- 支持多平面通信和隔离。
- 支持 Pod 访问 Service IP。
- 支持 Pod 应用和 Host 应用的访问。
- IP 地址为以节点为单位的预分配。

约束与限制

- Overlay L2 通信的上行链路网卡设备需要预配置 IP 地址。
- Overlay L2 目前不支持与其他网络模式的混用。
- ServiceIP 不支持多平面，只支持默认平面。
- Overaly L2 逻辑网络掩码（大子网掩码）目前不支持配置为大于 24 的值。
- Overlay L2 逻辑网络分配给节点的小子网掩码默认为 26，支持配置为 24~30。另外当小子网掩码为配置为 24 时，大子网掩码不支持配置为 24。
- Overlay L2 由于在每个节点上都是小子网，容器内网络的网关都是由 Canal 自动分配，不支持指定网关。

优缺点

优点：不受 VPC 配额规格、响应速度限制（路由条目数、弹性网卡数、创建速度限制）。

缺点：封装的额外开销，网络复杂性较高、性能较低，无法直接利用 VPC 提供的负载均衡、安全组等能力。

应用场景

适用于对网络时延、带宽要求不是特别高的一般容器业务场景。

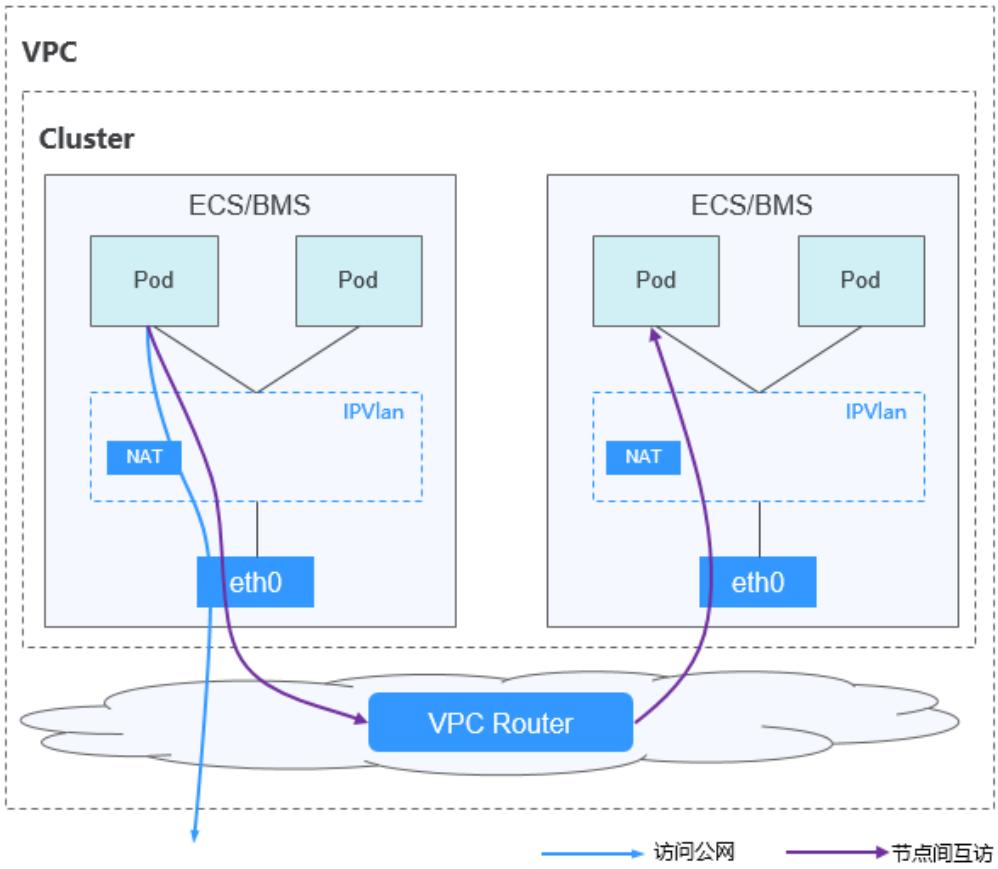
适用集群

- CCE 集群
- 虚机集群

8.4.3 VPC 网络

VPC 网络采用 VPC 路由方式与底层网络深度整合，适用于高性能场景，节点数量受限于虚拟私有云 VPC 的路由配额。每个节点将会被分配固定大小的 IP 地址段。VPC 网络由于没有隧道封装的消耗，容器网络性能相对于容器隧道网络有一定优势。VPC 网络集群由于 VPC 路由中配置有容器网段与节点 IP 的路由，可以支持集群外直接访问容器实例等特殊场景。

图8-12 VPC 网络



说明如下：

- 节点内容器间通信：通过 ipvlan 子接口分配给各个的容器，同节点的容器间通信直接通过 ipvlan 直接转发。
- 跨节点容器间通信：借助 VPC 的路由器的转发能力，所有跨节点容器间的通信全部默认路由到默认网关，借助 VPC 的路由能力，转发到对端节点上。

优缺点

优点：可利用 VPC 原生的网络功能，性能较高、功能丰富。

缺点：节点数量受限于虚拟私有云 VPC 的路由配额。每个节点将会被分配固定大小的 IP 地址段。

应用场景

适用于对网络时延、带宽要求高的高性能场景。

适用集群

- CCE 集群
- 虚机集群

8.5 NetworkPolicy

随着业务逻辑的复杂化，越来越多的应用需要大量模块之间的网络调用。传统的单一外部防火墙，或依照应用分层的防火墙已渐渐无法满足需求。在一个大的集群里面，各模块、业务逻辑层，或各职能团队之间的网络策略需求越来越强。

CCE 基于 Kubernetes 的网络策略功能进行了加强，通过配置网络策略，允许在同个集群内实现网络的隔离，也就是可以在某些实例（Pod）之间架起防火墙。

使用场景例如：某个用户有支付系统，且严格要求只能某几个组件能访问该支付系统，否则有被攻破的安全风险，通过配置网络策略可免除该风险。

约束与限制

- VPC 网络模型暂不支持网络策略（NetworkPolicy）。
- 网络策略（NetworkPolicy）暂不支持设置 egress 路由规则。

使用说明

若工作负载（例如名称为 workload1）未配置网络策略，那“当前集群内的其它工作负载”都可以访问“名为 workload1 的工作负载”。

设置网络策略

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 网络管理”。在 NetworkPolicy 页签，单击“添加 NetworkPolicy”。

- NetworkPolicy 名称：自定义输入 NetworkPolicy 名称。
- 集群名称：选择网络策略所在集群。
- 命名空间：选择网络策略所在命名空间。
- 关联工作负载：

  单击“选择工作负载”，选择“需要设置网络策略的工作负载”，例如工作负载名称为 **workload-1**，单击“确定”。
- 规则：单击“添加规则”，参数设置请参见表 8-18。

表8-18 添加规则

参数	参数说明
方向	当前仅支持入方向。即“其它工作负载”访问“当前的工作负载（即当前案例中的 workload-1）”。
协议	请选择对应的协议类型，目前支持 TCP 和 UDP 协议，不支持 ICMP 协议。
目的容器端口	容器镜像中应用程序实际监听端口，需用户确定。nginx 程序实际监听的端口为 80。 若不填写容器端口，默认所有端口都可被访问。
远端	选择可访问“当前工作负载”的“其它工作负载”。通过目的容器端口

参数	参数说明
	来访问。 • 命名空间：若选择某个命名空间，则该命名空间下的所有工作负载都会加入白名单，即都可访问 **workload-1**。 • 工作负载：若选择某个工作负载，即该工作负载可以访问 **workload-1**。仅支持选择与 **workload-1** 同个命名空间下的“其它工作负载”。

步骤 2 设置完成后，单击“创建”。

步骤 3 若需要为当前工作负载添加更多网络策略，例如其它端口需要被某个工作负载访问，可单击“添加 NetworkPolicy”，继续添加更多策略。

创建成功后，“仅远端中配置好的命名空间或工作负载”可以访问“当前工作负载”。

----结束

9 存储管理

9.1 存储概述

容器存储是为容器工作负载提供存储的组件，支持多种类型的存储，同一个工作负载（pod)可以使用任意数量的存储。

说明

Kubernetes1.13 版本之前的 CCE 集群不支持端到端容器存储扩容功能，PVC 容量与存储容量不一致。

存储类型选择

创建工作负载时，可以使用以下类型的存储。建议将工作负载 pod 数据存储在云存储上。若存储在本地磁盘上，节点异常无法恢复时，本地磁盘中的数据也将无法恢复。

- 本地硬盘：将容器所在宿主机的文件目录挂载到容器的指定路径中（对应 Kubernetes 的 HostPath），也可以不填写源路径（对应 Kubernetes 的 EmptyDir），不填写时将分配主机的临时目录挂载到容器的挂载点，指定源路径的本地硬盘数据卷适用于将数据持久化存储到容器所在宿主机，EmptyDir（不填写源路径）适用于容器的临时存储。配置项（ConfigMap）是一种用于存储工作负载所需配置信息的资源类型，内容由用户决定。密钥（Secret）是一种用于存储工作负载所需要认证信息、密钥的敏感信息等的资源类型，内容由用户决定。详情参见 9.2 本地磁盘存储。
- 云硬盘存储卷：CCE 支持将 EVS 创建的云硬盘挂载到容器的某一路径下。当容器迁移时，挂载的云硬盘将一同迁移。这种存储方式适用于需要永久化保存的数据。详情参见 9.3 云硬盘存储卷。
- 极速文件存储卷：CCE 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。详情参见 9.5 极速文件存储卷。
- 快照与备份：CCE 通过云硬盘 EVS 为您提供快照功能，云硬盘快照指的是云硬盘数据在某个时刻的完整拷贝或镜像，是一种重要的数据容灾手段，当数据丢失时，可通过快照将数据完整的恢复到快照时间点。详情参见 9.6 快照与备份。

9.2 本地磁盘存储

通过本地磁盘存储将容器所在宿主机的文件目录挂载到容器的指定路径中，也可以不填写源路径。

本地磁盘使用场景

使用本地硬盘有四种形式：

- **主机路径挂载**：将容器所在宿主机的文件目录挂载到容器指定的挂载点中，如容器需要访问/etc/hosts 则可以使用 HostPath 映射/etc/hosts 等场景。
- **临时路径挂载**：用于临时存储，生命周期与容器实例相同。容器实例消亡时，EmptyDir 会被删除，数据会永久丢失。
- **配置项挂载**：将 ConfigMap 配置项中的 key 映射到容器中，可以用于挂载配置文件到指定容器目录。ConfigMap 的创建请参见 11.1 创建配置项，具体使用请参见 11.2 使用配置项。
- **密钥挂载**：将密钥中的数据挂载到容器的某一路径中。密钥是一种用于存储工作负载所需要认证信息、密钥的敏感信息等的资源类型，内容由用户决定。Secret 的创建请参见 11.3 创建密钥，具体使用请参见 11.4 使用密钥。

下面分别介绍如何通过这四种形式进行挂载。

主机路径(HostPath)挂载

主机路径(HostPath)挂载表示将主机上的路径挂载到指定的容器路径。通常用于：“容器工作负载程序生成的日志文件需要永久保存”或者“需要访问宿主机上 Docker 引擎内部数据结构的容器工作负载”。

步骤 1 参照 6.2 创建无状态负载(Deployment)、6.3 创建有状态负载(StatefulSet)或 6.4 创建守护进程集(DaemonSet)，在“容器设置”步骤中，展开“数据存储”，单击“添加本地磁盘”。

步骤 2 设置添加本地磁盘参数，如表 9-1。

表9-1 卷类型选择主机路径挂载

参数	参数说明
存储类型	主机路径(HostPath)。
主机路径	输入主机路径，如/etc/hosts。 说明 请注意“主机路径”不能设置为根目录“/”，否则将导致挂载失败。挂载路径一般设置为： • /opt/xxxx（但不能为/opt/cloud） • /mnt/xxxx（但不能为/mnt/paas） • /tmp/xxx • /var/xxx （但不能为/var/lib、/var/script、/var/paas 等关键目录）

参数	参数说明
	• /xxxx（但不能和系统目录冲突，例如 bin、lib、home、root、boot、dev、etc、lost+found、mnt、proc、sbin、srv、tmp、var、media、opt、selinux、sys、usr 等） 注意不能设置为/home/paas、/var/paas、/var/lib、/var/script、/mnt/paas、/opt/cloud，否则会导致系统或节点安装失败。
添加容器挂载	配置如下参数： 1. 子路径：请输入子路径，如：tmp。 使用子路径挂载本地磁盘，实现在单一 Pod 中重复使用同一个 Volume。不填写时默认为根。 2. 挂载路径：请输入挂载路径，如：/tmp。 数据存储挂载到容器上的路径。请不要挂载在系统目录下，如“/”、“**/var/run**”等，会导致容器异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响容器启动的文件，否则文件会被替换，导致容器启动异常，工作负载创建失败。 须知 挂载高危目录的情况下 ，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。 3. 权限： • 只读：只能读容器路径中的数据卷。 • 读写：可修改容器路径中的数据卷，容器迁移时新写入的数据不会随之迁移，会造成数据丢失。 单击“添加容器挂载”可增加多条设置，单击“确定”完成配置。

----结束

临时路径(EmptyDir)挂载

临时路径(EmptyDir)挂载适用于临时存储、灾难恢复、共享运行时数据等场景，工作负载实例的删除或者迁移会导致临时路径被删除。

步骤 1 参照 6.2 创建无状态负载(Deployment)、6.3 创建有状态负载(StatefulSet)或 6.4 创建守护进程集(DaemonSet)，在添加容器后，展开“数据存储”，单击“添加本地磁盘”。

步骤 2 选择本地磁盘类型为“临时路径挂载”，设置添加本地磁盘参数，如表 9-2。

表9-2 卷类型选择临时路径挂载

参数	参数说明
存储类型	临时路径(EmptyDir)。

参数	参数说明
磁盘介质	• 若勾选“内存”，可以提高运行速度，但存储容量受内存大小限制。适用于数据量少，读写效率要求高的场景。 说明 • Memory 的 EmptyDir 使用的是内存，注意内存大小，用超过了容易 oom。 • Memory 的 EmptyDir 的大小为实例规格的 50%，暂时无法更改。 • 不使用 Memory 的 EmptyDir 不会占用系统内存。 • 若不勾选“内存”，即存储在硬盘上，适用于数据量大，读写效率要求低的场景。
添加容器挂载	配置如下参数： 1. 子路径：请输入子路径，如：tmp。 使用子路径挂载本地磁盘，实现在单一 Pod 中重复使用同一个 Volume。不填写时默认为根。 2. 挂载路径：请输入挂载路径，如：/tmp。 数据存储挂载到容器上的路径。请不要挂载在系统目录下，如“/”、“**/var/run**”等，会导致容器异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响容器启动的文件，否则文件会被替换，导致容器启动异常，工作负载创建失败。 须知 挂载高危目录的情况下 ，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。 3. 权限： • 只读：只能读容器路径中的数据卷。 • 读写：可修改容器路径中的数据卷，容器迁移时新写入的数据不会随之迁移，会造成数据丢失。 单击“添加容器挂载”可增加多条设置，单击“确定”完成配置。

----结束

配置项(ConfigMap)挂载

配置项(ConfigMap)挂载是将配置项中的数据挂载到指定的容器路径。平台提供工作负载代码和配置文件的分离，“配置项挂载”用于处理工作负载配置参数。用户需要提前创建工作负载配置，操作步骤请参见 11.1 创建配置项。

步骤 1 参照 6.2 创建无状态负载(Deployment)、6.3 创建有状态负载(StatefulSet)或 6.4 创建守护进程集(DaemonSet)，在添加容器后，展开“数据存储”，单击“添加本地磁盘”。

步骤 2 选择本地磁盘类型为“配置项挂载”，设置添加本地磁盘参数，如表 9-3。

表9-3 卷类型选择配置项挂载

参数	参数说明
存储类型	配置项(ConfigMap)。
配置项	选择对应的配置项名称。 配置项需要提前创建，具体请参见 11.1 创建配置项。
添加容器挂载	配置如下参数： 1. 子路径：请输入子路径，如：tmp。 使用子路径挂载本地磁盘，实现在单一 Pod 中重复使用同一个 Volume。不填写时默认为根。 2. 挂载路径：请输入挂载路径，如：/tmp。 数据存储挂载到容器上的路径。请不要挂载在系统目录下，如“/”、“**/var/run**”等，会导致容器异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响容器启动的文件，否则文件会被替换，导致容器启动异常，工作负载创建失败。 须知 挂载高危目录的情况下 ，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。 3. 设置权限：只读。只能读容器路径中的数据卷。 单击“添加容器挂载”可增加多条设置，单击“确定”完成配置。

----结束

密钥(Secret)挂载

密钥(Secret)挂载将密钥中的数据挂载到指定的容器路径，密钥内容由用户决定。用户需要提前创建密钥，操作步骤请参见 11.3 创建密钥。

步骤 1 参照 6.2 创建无状态负载(Deployment)、6.3 创建有状态负载(StatefulSet)或 6.4 创建守护进程集(DaemonSet)，在添加容器后，展开“数据存储”，单击“添加本地磁盘”。

步骤 2 选择本地磁盘类型为“密钥挂载”，设置添加本地磁盘参数，如表 9-4。

表9-4 卷类型选择密钥挂载

参数	参数说明
存储类型	密钥(Secret)。
密钥	选择对应的密钥名称。 密钥需要提前创建，具体请参见 11.3 创建密钥。

参数	参数说明
添加容器挂载	配置如下参数： 1. 子路径：请输入子路径，如：tmp。 使用子路径挂载本地磁盘，实现在单一 Pod 中重复使用同一个 Volume。不填写时默认为根。 2. 挂载路径：请输入挂载路径，如：/tmp。 数据存储挂载到容器上的路径。请不要挂载在系统目录下，如“/”、“**/var/run**”等，会导致容器异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响容器启动的文件，否则文件会被替换，导致容器启动异常，工作负载创建失败。 须知 挂载高危目录的情况下 ，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。 3. 设置权限：只读。只能读容器路径中的数据卷。 单击“添加容器挂载”可增加多条设置，单击“确定”完成配置。

----结束

使用 kubectl 挂载主机路径

CCE 支持使用 kubectl 将容器所在宿主机的文件目录挂载到容器指定的挂载点中。

步骤 1 请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令。

步骤 2 执行如下命令，配置名为“hostPath-pod-example.yaml”的创建 Pod 的 yaml 文件。

touch hostPath-pod-example.yaml

vi hostPath-pod-example.yaml

在无状态工作负载中挂载主机路径，示例如下：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: hostpath-pod-example
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hostpath-pod-example
  template:
    metadata:
      labels:
        app: hostpath-pod-example
```

```
  spec:
    containers:
    - image: nginx
      name: container-0
      volumeMounts:
      - mountPath: /tmp
        name: hostpath-example
    restartPolicy: Always
    volumes:
    - name: hostpath-example
      hostPath:
        path: /tmp
```

表9-5 本地磁盘储存依赖参数说明

参数	描述
mountPath	容器内挂载路径，示例中挂载到“/tmp”路径。
hostPath	主机路径，示例中主机路径为“/tmp”。

说明

“spec.template.spec.containers.volumeMounts.name ”和 “spec.template.spec.volumes.name”有映射关系，必须保持一致。

步骤 3 执行如下命令创建 Pod。

kubectl create -f hostPath-pod-example.yaml

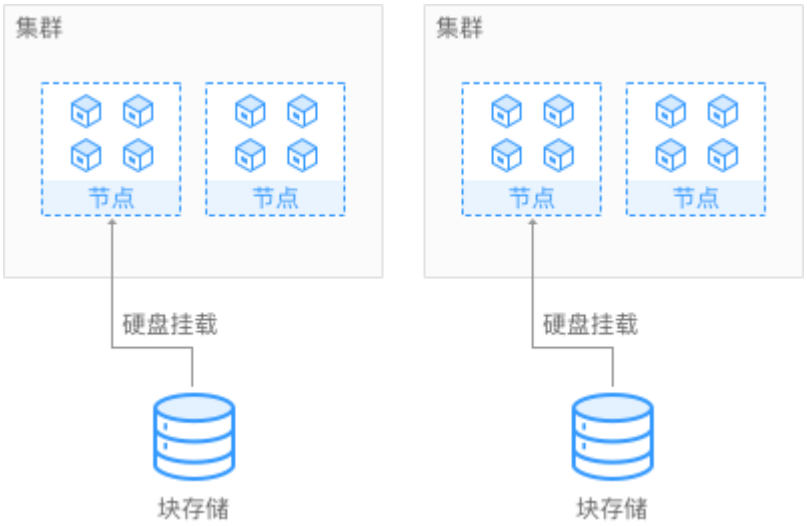
----结束

9.3 云硬盘存储卷

9.3.1 云硬盘存储卷使用说明

为满足数据的持久化需求，CCE 支持将云硬盘（EVS）挂载到容器中。通过云硬盘，可以将存储系统的远端文件目录挂载到容器中，数据卷中的数据将被永久保存，即使删除了容器，数据卷中的数据依然保存在存储系统中。

图9-1 CCE 挂载云硬盘存储卷



云硬盘存储说明

- 您可以像使用传统服务器硬盘一样，对挂载到服务器上的块存储（硬盘）做格式化、创建文件系统等操作。
- 数据不共享：每台服务器使用独立的块存储（硬盘），多服务器之间数据隔离。
- 私有网络：数据访问必须在数据中心内部网络中。
- 单卷容量有限（TB 级），但性能极佳（IO 读写时延 ms 级），主要面向数据库，企业办公应用等场景。
- 适用于供单实例部署的无状态负载（Deployment）和普通任务（Job），以及有状态工作负载（StatefulSet）的每个实例独占式使用。

9.3.2 使用云硬盘存储卷

约束与限制

- CCE 默认创建计费模式为“按需计费”的云硬盘。
- 云硬盘不支持跨可用区挂载，且暂时不支持被多个工作负载、同一个工作负载的多个实例或多个任务使用。
- 由于 CCE 集群各节点之间暂不支持共享盘的数据共享功能，多个节点挂载使用同一个云硬盘可能会出现读写冲突、数据缓存冲突等问题，所以创建无状态工作负载时，若使用了 EVS 云硬盘，建议工作负载只选择一个实例。
- 创建有状态工作负载并添加云存储时，云硬盘暂不支持使用已有存储。
- 不支持导入分区过或者具有非 ext4 文件系统的云硬盘。
- CCE 集群中的容器存储目前已支持加密（Kubernetes 1.13 版本及以上），当前仅在部分区域（Region）提供端到端支持。
- 存储不支持选择企业项目，新创建的存储卷默认创建到 default 企业项目下。

购买云硬盘存储卷

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 存储管理”，单击“购买云硬盘存储卷”。

步骤 2 参照表 9-6 配置磁盘基本信息。

表9-6 创建磁盘基本信息配置

参数	参数说明
* PVC 名称	新建 PVC 的名称，PVC 创建时将自动创建存储卷，一个 PVC 对应一个存储卷。存储卷名称在 PVC 创建时自动生成。
集群名称	云硬盘部署所在的集群。
命名空间	选择云硬盘所部署的命名空间，若无需选择，保持默认 default 即可。
磁盘容量	新建存储的容量。
访问模式	访问模式（AccessModes）是用来对 PV 进行访问模式的设置，用于描述用户应用对存储资源的访问权限。 • ReadWriteOnce (RWO)：基于 EVS 非共享卷提供容器负载单 Pod 单读单写块存储的功能，但是该卷只能被单个节点挂载。v1.13.10-r1 开始支持 RWO 模式的 EVS 卷。 • ReadWriteMany (RWX)：基于 EVS 共享卷提供容器负载访问块存储的功能。由于容器侧不支持跨节点的 EVS 共享卷读写一致性能力，在老集群（<v1.15.6）下受限使用（需要确保单负载单实例单读单写），v1.15 开始只支持创建和使用 RWO 模式的 EVS 非共享卷
可用区	选择云硬盘所属可用区。
类型	新建云磁盘类型，默认为“普通 I/O”。 • 普通 I/O：是指由 SATA 存储提供资源的磁盘类型。 • 高 I/O：是指由 SAS 存储提供资源的磁盘类型。 • 超高 I/O：是指由 SSD 存储提供资源的磁盘类型。
存储格式	默认为 CSI，暂不支持更改。
加密	“KMS 加密”默认不勾选。 勾选“KMS 加密”后，将为云硬盘提供易用、高安全性的密钥管理服务（Key Management Service，KMS）。若未创建委托请单击“创建委托”，并配置如下参数： • 委托名称：委托表示委托方通过创建信任关系，给被委托方授予访问其资源的权限。当“委托名称”为“EVSAccessKMS”时，表示已经成功授权 EVS 访问 KMS，授权成功后，EVS 可以获取 KMS 密钥用来加解密云硬盘系统。

参数	参数说明
	• 密钥名称：密钥资源创建完成后，可在容器应用中加载使用。 • 密钥 ID：默认生成。 该功能仅在部分 Region 的 1.13.10 及以上版本的集群中支持。

步骤 3 单击“立即购买”。确认订单详情无误后，单击“提交”，等待创建成功。

创建成功后，存储管理列表中会出现已创建的存储，待状态为“正常”，表示存储已创建成功。

步骤 4 单击存储名称，进入存储详情页，可查看到存储的卷名称、存储容量等信息。

----结束

添加云硬盘存储卷

步骤 1 参照 6.2 创建无状态负载(Deployment)、6.3 创建有状态负载(StatefulSet)或 6.5 创建普通任务(Job)创建工作负载或普通任务，在添加容器后，展开“数据存储”。在“云存储”页签，单击“添加云存储”。

步骤 2 选择存储类型为“云硬盘”。

表9-7 挂载云硬盘参数配置

参数	参数说明
云存储类型	云硬盘：云硬盘的使用方式与传统服务器硬盘完全一致。同时，云硬盘具有更高的数据可靠性，更高的 I/O 吞吐能力和更加简单易用等特点，适用于文件系统、数据库或者其他需要块存储设备的系统软件或工作负载。 注意 • **如需挂载云硬盘，创建工作负载时的实例数量必须选择为 1 个实例，即单实例，选择多实例后挂载云硬盘的选项将置灰，无法挂载。** • 创建有状态工作负载并添加云存储时，云硬盘暂不支持使用已有存储。 • 云硬盘不支持跨可用区挂载，且暂时不支持被多个工作负载、同一个工作负载的多个实例或多个任务使用。
分配方式	
使用已有存储	选择已创建的存储，您需要提前创建好存储。 针对同一集群和命名空间，创建无状态工作负载时可以选择“使用已有存储”。 创建有状态工作负载时暂不支持选择“使用已有存储”，只能使用“自动分配存储”。
自动分配存储	选择自动分配存储后，需要配置如下选项：

参数	参数说明
	1. 访问模式：是用来对 PV 进行访问模式的设置，用于描述用户应用对存储资源的访问权限。 • ReadWriteOnce (RWO)：基于 EVS 非共享卷提供容器负载单 Pod 单独单写块存储的功能，但是该卷只能被单个节点挂载。v1.13.10-r1 开始支持 RWO 模式的 EVS 卷。 • ReadWriteMany (RWX)：基于 EVS 共享卷提供容器负载访问块存储的功能。由于容器侧不支持跨节点的 EVS 共享卷读写一致性能力，在老集群（<1.15.6）下受限使用（需要确保单负载单实例单独单写），1.15 开始只支持创建和使用 RWO 模式的 EVS 非共享卷 2. 可用区：存储所在的可用区，自动分配存储仅支持 Node 节点所在可用区。 3. 子类型：选择存储的子类型。 • 高 I/O：指由 SAS 存储介质构成的云硬盘。 • 普通 I/O：指由 SATA 存储介质构成的云硬盘。 • 超高 I/O：指由 SSD 存储介质构成的云硬盘。 4. 存储容量：输入存储容量，单位为 GB。请不要超过存储容量配额，否则会创建失败。 5. 存储格式：默认选择 CSI。 CSI，即 Container Storage Interface，容器存储接口机制，用于在 Kubernetes 和外部存储系统之间建立一套标准的存储管理接口，通过该接口为容器提供存储服务。 6. 加密：勾选“KMS 加密”后，将为云硬盘提供易用、高安全性的密钥管理服务（Key Management Service，KMS），该功能仅在部分 Region 的 1.13.10 及以上版本的集群中支持。若未创建委托请单击“创建委托”，并配置如下参数： • 委托名称：委托表示委托方通过创建信任关系，给被委托方授予访问其资源的权限。当“委托名称”为“EVSAccessKMS”时，表示已经成功授权 EVS 访问 KMS，授权成功后，EVS 可以获取 KMS 密钥用来加解密云硬盘系统。 • 密钥名称：密钥资源创建完成后，可在容器应用中加载使用。 • 密钥 ID：默认生成。
添加容器挂载	1. 单击“添加容器挂载”。 2. 挂载路径：输入数据卷挂载到容器上的路径。 须知 • 请不要挂载在系统目录下，如“/”、“**/var/run**”等，会导致容器异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响容器启动的文件，否则文件会被替换，导致容器启动异常，工作负载创建失败。 • 挂载高危目录的情况下 ，建议使用低权限帐号启动，否则可能会造成

参数	参数说明
	宿主机高危文件被破坏。 3. 设置权限。 • 只读：只能读容器路径中的数据卷。 • 读写：可修改容器路径中的数据卷，容器迁移时新写入的数据不会随之迁移，会造成数据丢失。

步骤 3 单击“确定”。

----结束

导入云硬盘存储卷

CCE 支持导入已有的云硬盘存储。

说明

一块云硬盘只能导入一个命名空间，不能同时导入多个命名空间。如果一块云硬盘已经被导入到一个命名空间，则在其他命名空间下不可见，不能再次导入。**如果需要导入已经格式化文件系统（ext4）的 EVS 磁盘，需要确保磁盘没有设置分区，否则可能存在数据丢失的情况。**

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 存储管理”，在“云硬盘存储卷”页签，单击“导入”。

步骤 2 从列表里选择要导入的云硬盘存储，单击“确定”。

----结束

云硬盘包周期

- 如需使用包周期的云硬盘，请参考如下转包周期指导：

  方法一：

  a. 使用 CCE 存储管理创建所需容量的云硬盘，该云硬盘默认计费模式为按需计费。

  b. 在 CCE 云硬盘列表单击该云硬盘的转包周期链接，对该云硬盘进行计费模式转换为包周期的操作，该操作将耗费一定时间。

  方法二：

  a. 在 EVS 服务侧创建用户所需容量的包周期云硬盘。

  b. 在 CCE 存储管理单击导入，选择步骤一创建的云硬盘，单击确定，将该云硬盘导入到 CCE 中使用。

> **⚠ 注意**
>
> 在 CCE 侧删除包周期云硬盘时并不会真正删除该云硬盘，CCE 只做解关联操作。若用户想彻底删除该云硬盘，请先在 CCE 侧进行解关联操作，然后至费用中心执行退订操作。

解关联云硬盘

创建或导入云硬盘成功后，云硬盘会自动和当前集群绑定，其它集群无法使用该存储。解关联后，将和此集群解绑，其它集群均可以导入使用该云硬盘。

若云硬盘已被工作负载挂载，将无法和集群解关联。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 存储管理”，在云硬盘存储卷列表中，单击云硬盘后的“解关联”。

步骤 2 确认解关联后，单击“确定”。

----结束

相关操作

云硬盘存储创建完成后，您还可以执行表 9-8 中的操作。

表9-8 其他操作

操作	说明
删除云硬盘存储	1. 选择需要删除的云硬盘，单击操作列的“删除”。 2. 根据界面提示完成删除操作。

9.3.3 使用 kubectl 自动创建云硬盘

操作场景

CCE 支持使用 PersistentVolumeClaim（PVC）的形式创建云硬盘。

操作步骤

步骤 1 请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令。

步骤 2 执行如下命令，配置名为“pvc-evs-auto-example.yaml”的创建 PVC 的 yaml 文件。

touch pvc-evs-auto-example.yaml

vi pvc-evs-auto-example.yaml

- **1.15 及以上版本的集群，yaml 文件配置示例如下：**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-evs-auto-example
  namespace: default
  annotations:
    everest.io/disk-volume-type: SAS
  labels:
    failure-domain.beta.kubernetes.io/region: region01
    failure-domain.beta.kubernetes.io/zone: AZ01
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: csi-disk
```

表9-9 关键参数说明

参数	描述
everest.io/disk-volume-type	云硬盘类型，全大写。 当前支持高 I/O（SAS）、超高 I/O（SSD）和普通 I/O（SATA）。
failure-domain.beta.kubernetes.io/region	集群所在的 region。
failure-domain.beta.kubernetes.io/zone	创建云硬盘所在的可用区，必须和工作负载规划的可用区保持一致。
storage	存储容量，单位为 Gi。
storageClassName	存储卷动态供应关联的 k8s storage class 名称。 v1.15 集群使用的 csi 关联的 storage class 名称：csi-disk。
accessModes	指定读写模式，显示 volume 实际具有的访问模式。 1.15 集群版本只支持非共享卷，此字段设置为“ReadWriteOnce”。

- **1.15 之前的 1.9、1.11、1.13 版本集群，yaml 文件配置示例如下：**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-evs-auto-example
  namespace: default
  annotations:
    volume.beta.kubernetes.io/storage-class: SAS
```

```
  labels:
    failure-domain.beta.kubernetes.io/region: region01
    failure-domain.beta.kubernetes.io/zone: AZ01
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
```

表9-10 关键参数说明

参数	描述
volume.beta.kubernetes.io/storage-class	云硬盘类型，全大写。 当前支持高 I/O（SAS）、超高 I/O（SSD）和普通 I/O（SATA）。
failure-domain.beta.kubernetes.io/region	集群所在的 region。
failure-domain.beta.kubernetes.io/zone	创建云硬盘所在的可用区，必须和工作负载规划的可用区保持一致。
storage	存储容量，单位为 Gi。
accessModes	指定读写模式，显示 volume 实际具有的访问模式。 支持配置“ReadWriteMany”（共享卷）与“ReadWriteOnly”（非共享卷）

步骤 3 执行如下命令创建 PVC。

kubectl create -f pvc-evs-auto-example.yaml

命令执行完成后，会在集群所在分区创建 EVS 云硬盘，您可以在“存储管理 > 云硬盘存储卷”中查看该云硬盘，也可以在 EVS 的控制台中根据卷名称查看该硬盘。

----结束

9.3.4 使用 kubectl 对接已有云硬盘

操作场景

CCE 支持使用已有的 EVS 云硬盘创建 PersistentVolume。创建成功后，通过创建相应的 PersistentVolumeClaim 绑定当前 PersistentVolume 使用。

操作步骤

步骤 1 登录 EVS 控制台，创建一个 EVS 云硬盘，记录云硬盘的 VolumeID、容量和磁盘类型。

步骤 2 请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令。

步骤 3 新建两个 yaml 文件，用于创建 PersistentVolume（PV）、PersistentVolumeClaim（PVC），假设文件名分别为 **pv-evs-example.yaml**、**pvc-evs-example.yaml**。

touch pv-evs-example.yaml pvc-evs-example.yaml

K8S 集群版本（K8S version）	说明	yaml 示例
1.15+	1.15 及以上版本的集群	请参见 1.15+ yaml 文件配置示例
1.11.7 ≤ K8S version ≤ 1.13	1.11.7 以上及 1.13 版本集群	请参见 1.11.7~1.13 yaml 文件配置示例
1.11 ≤ K8S version < 1.11.7	1.11.7 之前的 1.11 版本集群	请参见 1.11~1.11.7 yaml 文件配置示例
K8S version = 1.9	1.9 版本集群	请参见 1.9 yaml 文件配置示例

1.15+（1.15 及以上版本的集群）

- **vi pv-evs-example.yaml**

  PV yaml 文件配置示例如下：

```
apiVersion: v1
kind: PersistentVolume
metadata:
  labels:
    failure-domain.beta.kubernetes.io/region: region01
    failure-domain.beta.kubernetes.io/zone: AZ01
  annotations:
    pv.kubernetes.io/provisioned-by: everest-csi-provisioner
  name: pv-evs-example
spec:
  accessModes:
  - ReadWriteOnce
  capacity:
    storage: 10Gi
  csi:
    driver: disk.csi.everest.io
    fsType: ext4
    volumeAttributes:
      everest.io/disk-mode: SCSI
      everest.io/disk-volume-type: SAS
      storage.kubernetes.io/csiProvisionerIdentity: everest-csi-provisioner
    volumeHandle: 0992dbda-6340-470e-a74e-4f0db288ed82
  persistentVolumeReclaimPolicy: Delete
  storageClassName: csi-disk
```

表9-11 关键参数说明

参数	描述
everest.io/disk-volume-type	云硬盘类型，全大写。 当前支持高 I/O（SAS）、超高 I/O（SSD）和普通 I/O（SATA）。
failure-domain.beta.kubernetes.io/region	集群所在的 region。
failure-domain.beta.kubernetes.io/zone	创建云硬盘所在的可用区，必须和工作负载规划的可用区保持一致。
storage	云硬盘的容量，单位为 Gi。
storageClassName	存储卷动态供应关联的 k8s storage class 名称。 云硬盘需使用“csi-disk”。
accessModes	指定读写模式，显示 volume 实际具有的访问模式。 1.15 集群版本只支持非共享卷，此字段设置为“ReadWriteOnce”。
driver	挂载依赖的存储驱动。 EVS 云硬盘配置为“disk.csi.everest.io”。
volumeHandle	云硬盘的 volumeID。 **获取方法：**在云服务器控制台中，单击左侧栏目树中的“云硬盘 > 磁盘”，单击要对接的云硬盘名称进入详情页，在“概览信息”页签下单击“ID”后的复制图标即可获取云硬盘的 volumeID。
everest.io/disk-mode	云硬盘磁盘模式，取值支持 SCSI。

- **vi pv-evs-example.yaml**

  PVC yaml 文件配置示例如下：

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  labels:
    failure-domain.beta.kubernetes.io/region: region01
    failure-domain.beta.kubernetes.io/zone: AZ01
  annotations:
    everest.io/disk-volume-type: SAS
    volume.beta.kubernetes.io/storage-provisioner: everest-csi-provisioner
  name: pvc-evs-example
  namespace: default
spec:
```

```
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  volumeName:  pv-evs-example
  storageClassName: csi-disk
```

表9-12 关键参数说明

参数	描述
everest.io/disk-volume-type	云硬盘类型，全大写。 当前支持高 I/O（SAS）、超高 I/O（SSD）和普通 I/O（SATA），必须和已有 PV 保持一致。
failure-domain.beta.kubernetes.io/region	集群所在的 region。
failure-domain.beta.kubernetes.io/zone	EVS 云硬盘所在可用区，必须和工作负载规划的可用区保持一致。u 欧
storage	PVC 申请容量，单位为 Gi。 必须和已有 PV 的 storage 大小保持一致。
storageClassName	指定 k8s storage class 名称。 云硬盘需使用“csi-disk”。
volumeName	PV 的名称。

步骤 4 创建 PV。

kubectl create -f pv-evs-example.yaml

步骤 5 创建 PVC。

kubectl create -f pvc-evs-example.yaml

执行成功后，可以在“资源管理 > 存储管理”的云硬盘存储中查看创建的 PVC，也可以在 EVS 页面根据名称查看 EVS 云硬盘。

步骤 6 （可选）增加集群关联的 metadata，确保在删除节点或集群时避免删除已挂载的静态 PV 关联的 EVS 盘。

⚠ 注意

若不执行本步骤或创建静态 PV/PVC 时没有执行过本步骤，请务必确保删除节点前，提前将静态 PV 关联的云硬盘从节点上卸载。

1. 获取租户 Token。

2. 获取 EVS 访问地址 EVS_ENDPOINT。
3. 给 EVS 静态 PV 关联的 EVS 盘补充集群关联的 metadata。

```
curl -X POST ${EVS_ENDPOINT}/v2/${project_id}/volumes/${volume_id}/metadata --
insecure \
    -d '{"metadata":{"cluster_id": "${cluster_id}", "namespace":
"${pvc_namespace}"}}' \
    -H 'Accept:application/json' -H 'Content-
Type:application/json;charset=utf8' \
    -H 'X-Auth-Token:${TOKEN}'
```

表9-13 关键参数说明

参数	描述
EVS_ENDPOINT	EVS 访问地址，配置为 b 中获取的值。
project_id	项目 ID。
volume_id	关联 EVS 盘的 ID，配置为步骤 2 待创建静态 PV 中的 volume_id，也可在 EVS 控制台，单击待导入的云硬盘名称，在磁盘详情界面的“概览信息”中获取 ID 的值。
cluster_id	待创建 EVS PV 的集群 ID。在 CCE 控制台中单击“资源管理 > 集群管理”。单击待关联集群的名称，在集群详情页面，即可获取集群 ID。
pvc_namespace	待绑定 PVC 的 namespace 名称。
TOKEN	用户 Token，配置为 a 中获取的值。

例如，执行如下命令：

```
curl -X POST
https://evs.region01.myhuaweicloud.com:443/v2/060576866680d5762f52c0150e726aa7/
volumes/69c9619d-174c-4c41-837e-31b892604e14/metadata --insecure \
    -d '{"metadata":{"cluster_id": "71e8277e-80c7-11ea-925c-0255ac100442",
"namespace": "default"}}' \
    -H 'Accept:application/json' -H 'Content-
Type:application/json;charset=utf8' \
    -H 'X-Auth-Token:MIIPe******IsIm1ldG
```

请求执行完成后，执行如下命令，可查看 EVS 盘是否已关联集群的 metadata。

```
curl -X GET ${EVS_ENDPOINT}/v2/${project_id}/volumes/${volume_id}/metadata --
insecure \
    -H 'X-Auth-Token:${TOKEN}'
```

例如，执行如下命令：

```
curl -X GET
https://evs.region01.myhuaweicloud.com/v2/060576866680d5762f52c0150e726aa7/volu
mes/69c9619d-174c-4c41-837e-31b892604e14/metadata --insecure \
    -H 'X-Auth-Token:MIIPeAYJ***9t1c31ASaQ=='
```

在回显中就可以看到该 EVS 盘当前的 metadata。

```
{
    "metadata": {
        "namespace": "default",
        "cluster_id": "71e8277e-80c7-11ea-925c-0255ac100442",
        "hw:passthrough": "true"
    }
}
```

----结束

历史版本示例

1.11.7 ≤ K8S version ≤ 1.13（1.11.7 以上及 1.13 版本集群）

- **PV yaml 文件配置示例如下：**

```
apiVersion: v1
kind: PersistentVolume
metadata:
  labels:
    failure-domain.beta.kubernetes.io/region: region01
    failure-domain.beta.kubernetes.io/zone:  AZ01
  annotations:
    pv.kubernetes.io/provisioned-by: flexvolume-huawei.com/fuxivol
  name: pv-evs-example
spec:
  accessModes:
  - ReadWriteMany
  capacity:
    storage: 10Gi
  flexVolume:
    driver: huawei.com/fuxivol
    fsType: ext4
    options:
      disk-mode: SCSI
      fsType: ext4
      volumeID: 0992dbda-6340-470e-a74e-4f0db288ed82
  persistentVolumeReclaimPolicy: Delete
  storageClassName: sas
```

表9-14 关键参数说明

参数	描述
failure-domain.beta.kubernetes.io/region	集群所在的 region。
failure-domain.beta.kubernetes.io/zone	创建云硬盘所在的可用区，必须和工作负载规划的可用区保持一致。
storage	云硬盘的容量，单位为 Gi。
storageClassName	云硬盘类型，当前支持高 I/O（sas）、超高 I/O（ssd）和普通 I/O（sata）。

参数	描述
driver	挂载依赖的存储驱动。 EVS 云硬盘配置为“huawei.com/fuxivol”。
volumeID	云硬盘的 volumeID。 **获取方法：**在 CCE 控制台中，单击左侧栏目树中的“资源管理-存储管理”，在“云硬盘存储卷”页签下单击 PVC 的名称，在 PVC 详情页中复制“PVC UID”后的内容即可。
disk-mode	云硬盘磁盘模式，取值可以是 VBD 和 SCSI。 v1.11.7 之前的 CCE 集群，该字段无需填写，默认都是 VBD。 v1.11.7+以及 v1.13 的 Linux x86 CCE 集群要求该字段值必须存在，且基于 PVC 触发动态创建的都是 EVS SCSI 模式的卷，因此这里静态 PV 形式优先选用 SCSI 模式的云硬盘；同时支持升级后的老集群中 VBD 卷能够继续正常使用。

- **PVC yaml 文件配置示例如下：**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: sas
    volume.beta.kubernetes.io/storage-provisioner: flexvolume-
huawei.com/fuxivol
  labels:
    failure-domain.beta.kubernetes.io/region: region01
    failure-domain.beta.kubernetes.io/zone: AZ01
  name: pvc-evs-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  volumeName: pv-evs-example
```

表9-15 关键参数说明

参数	描述
volume.beta.kubernetes.io/storage-class	存储类型，支持 sas，ssd 和 sata。必须和已有 PV 保持一致。

参数	描述
volume.beta.kubernetes.io/storage-provisioner	必须使用 flexvolume-huawei.com/fuxivol。
failure-domain.beta.kubernetes.io/region	集群所在的 region。
failure-domain.beta.kubernetes.io/zone	EVS 云硬盘所在可用区，必须和工作负载规划的可用区保持一致。
storage	PVC 申请容量，单位为 Gi。 必须和已有 PV 的 storage 大小保持一致。
volumeName	PV 的名称。

1.11 ≤ K8S version < 1.11.7（1.11.7 之前的 1.11 版本集群）

- **PV yaml 文件配置示例如下：**

```
apiVersion: v1
kind: PersistentVolume
metadata:
  labels:
    failure-domain.beta.kubernetes.io/region: region01
    failure-domain.beta.kubernetes.io/zone:  AZ01
  name: pv-evs-example
spec:
  accessModes:
  - ReadWriteMany
  capacity:
    storage: 10Gi
  flexVolume:
    driver: huawei.com/fuxivol
    fsType: ext4
    options:
      fsType: ext4
      volumeID: 0992dbda-6340-470e-a74e-4f0db288ed82
  persistentVolumeReclaimPolicy: Delete
  storageClassName: sas
```

表9-16 关键参数说明

参数	描述
failure-domain.beta.kubernetes.io/region	集群所在的 region。 Region 对应的值请参见。
failure-domain.beta.kubernetes.io/zone	创建云硬盘所在的可用区，必须和工作负载规划的可用区保持一致。 zone 对应的值请参见。
storage	云硬盘的容量，单位为 Gi。

参数	描述
storageClassName	云硬盘类型，当前支持高 I/O（sas）、超高 I/O（ssd）和普通 I/O（sata）。
driver	挂载依赖的存储驱动。 EVS 云硬盘配置为“huawei.com/fuxivol”。
volumeID	云硬盘的 volumeID。 **获取方法：**在 CCE 控制台中，单击左侧栏目树中的“资源管理-存储管理”，在“云硬盘存储卷”页签下单击 PVC 的名称，在 PVC 详情页中复制“PVC UID”后的内容即可。
disk-mode	云硬盘磁盘模式，取值可以是 VBD 和 SCSI。 v1.11.7 之前的 CCE 集群，该字段无需填写，默认都是 VBD。 v1.11.7+以及 v1.13 的 Linux x86 CCE 集群要求该字段值必须存在，且基于 PVC 触发动态创建的都是 EVS SCSI 模式的卷，因此这里静态 PV 形式优先选用 SCSI 模式的云硬盘；同时支持升级后的老集群中 VBD 卷能够继续正常使用。

- **PVC yaml 文件配置示例如下：**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: sas
    volume.beta.kubernetes.io/storage-provisioner: flexvolume-
huawei.com/fuxivol
  labels:
    failure-domain.beta.kubernetes.io/region: region01
    failure-domain.beta.kubernetes.io/zone: AZ01
  name: pvc-evs-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  volumeName: pv-evs-example
```

表9-17 关键参数说明

参数	描述
volume.beta.kubernetes.io/storage-	存储类型，支持 sas，ssd 和 sata。必须和已有

参数	描述
class	PV 保持一致。
volume.beta.kubernetes.io/storage-provisioner	必须使用 flexvolume-huawei.com/fuxivol。
failure-domain.beta.kubernetes.io/region	集群所在的 region。
failure-domain.beta.kubernetes.io/zone	EVS 云硬盘所在可用区，必须和工作负载规划的可用区保持一致。
storage	PVC 申请容量，单位为 Gi。 必须和已有 PV 的 storage 大小保持一致。
volumeName	PV 的名称。

K8S version = 1.9（1.9 版本集群）

- **PV yaml 文件配置示例如下：**

```
apiVersion: v1
kind: PersistentVolume
metadata:
  labels:
    failure-domain.beta.kubernetes.io/region: region01
    failure-domain.beta.kubernetes.io/zone:  AZ01
  name: pv-evs-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  capacity:
    storage: 10Gi
  flexVolume:
    driver: huawei.com/fuxivol
    fsType: ext4
    options:
      fsType: ext4
      kubernetes.io/namespace: default
      volumeID: 0992dbda-6340-470e-a74e-4f0db288ed82
  persistentVolumeReclaimPolicy: Delete
  storageClassName: sas
```

表9-18 关键参数说明

参数	描述
failure-domain.beta.kubernetes.io/region	集群所在的 region。
failure-domain.beta.kubernetes.io/zone	创建云硬盘所在的可用区，必须和工作负载规划的可用区保持一致。

参数	描述
storage	云硬盘的容量，单位为 Gi。
storageClassName	云硬盘类型，当前支持高 I/O（sas）、超高 I/O（ssd）和普通 I/O（sata）。
driver	挂载依赖的存储驱动。 EVS 云硬盘配置为“huawei.com/fuxivol”。
volumeID	云硬盘的 volumeID。 **获取方法：**在 CCE 控制台中，单击左侧栏目树中的“资源管理-存储管理”，在“云硬盘存储卷”页签下单击 PVC 的名称，在 PVC 详情页中复制“PVC UID”后的内容即可。
disk-mode	云硬盘磁盘模式，取值可以是 VBD 和 SCSI。 v1.11.7 之前的 CCE 集群，该字段无需填写，默认都是 VBD。 v1.11.7+以及 v1.13 的 Linux x86 CCE 集群要求该字段值必须存在，且基于 PVC 触发动态创建的都是 EVS SCSI 模式的卷，因此这里静态 PV 形式优先选用 SCSI 模式的云硬盘；同时支持升级后的老集群中 VBD 卷能够继续正常使用。

- **PVC yaml 文件配置示例如下：**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: sas
    volume.beta.kubernetes.io/storage-provisioner: flexvolume-
huawei.com/fuxivol
  labels:
    failure-domain.beta.kubernetes.io/region: region01
    failure-domain.beta.kubernetes.io/zone: AZ01
  name: pvc-evs-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  volumeName: pv-evs-example
  volumeNamespace: default
```

表9-19 关键参数说明

参数	描述
volume.beta.kubernetes.io/storage-class	存储类型，支持 sas，ssd 和 sata。必须和已有 PV 保持一致。
volume.beta.kubernetes.io/storage-provisioner	必须使用 flexvolume-huawei.com/fuxivol。
failure-domain.beta.kubernetes.io/region	集群所在的 region。
failure-domain.beta.kubernetes.io/zone	EVS 云硬盘所在可用区，必须和工作负载规划的可用区保持一致。
storage	PVC 申请容量，单位为 Gi。 必须和已有 PV 的 storage 大小保持一致。
volumeName	PV 的名称。

9.3.5 使用 kubectl 部署带云硬盘存储卷的工作负载

操作场景

云硬盘创建或导入 CCE 后，可以在工作负载中挂载云硬盘。

须知

云硬盘不支持跨可用区挂载。在挂载前，您可以使用 kubectl get pvc 命令查询当前集群所在分区下可用 PVC。

操作步骤

步骤 1 执行如下命令，配置名为“evs-pod-example.yaml”的创建 Pod 的 yaml 文件。

touch evs-pod-example.yaml

vi evs-pod-example.yaml

在无状态工作负载中基于 pvc 共享式使用云硬盘存储示例：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: evs-pod-example
  namespace: default
spec:
  replicas: 1
  selector:
```

```
    matchLabels:
      app: evs-pod-example
  template:
    metadata:
      labels:
        app: evs-pod-example
    spec:
      containers:
      - image: nginx
        name: container-0
        volumeMounts:
        - mountPath: /tmp
          name: pvc-evs-example
      restartPolicy: Always
      volumes:
      - name: pvc-evs-example
        persistentVolumeClaim:
          claimName: pvc-evs-auto-example
```

表9-20 关键参数说明

前置路径	参数	描述
spec.template.spec.containers.volumeMounts	name	容器内挂载卷的名称。
spec.template.spec.containers.volumeMounts	mountPath	容器内挂载路径，示例中挂载到“/tmp”路径。
spec.template.spec.volumes	name	卷的名称。
spec.template.spec.volumes.persistentVolumeClaim	claimName	已有 PVC 名称。

说明

“spec.template.spec.containers.volumeMounts.name ”和 “spec.template.spec.volumes.name”有映射关系，必须保持一致。

在有状态工作负载中基于 PVCTemplate 独占式使用云硬盘存储示例：

- **1.15 及以上版本的集群，yaml 示例如下：**

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: deploy-evs-sata-in
  namespace: default
  generation: 1
  labels:
    appgroup: ''
  annotations:
    container.io/container-0: ***/swr/dockerimage/nginx.png
    description: ''
spec:
```

```
  replicas: 1
  selector:
    matchLabels:
      app: deploy-evs-sata-in
      failure-domain.beta.kubernetes.io/region: region01
      failure-domain.beta.kubernetes.io/zone: AZ01
  template:
    metadata:
      labels:
        app: deploy-evs-sata-in
        failure-domain.beta.kubernetes.io/region: region01
        failure-domain.beta.kubernetes.io/zone: AZ01
      annotations:
        metrics.alpha.kubernetes.io/custom-endpoints:
'[{"api":"","path":"","port":"","names":""}]'
        pod.alpha.kubernetes.io/initialized: 'true'
    spec:
      containers:
        - name: container-0
          image: 'nginx:1.12-alpine-perl'
          env:
            - name: PAAS_APP_NAME
              value: deploy-evs-sata-in
            - name: PAAS_NAMESPACE
              value: default
            - name: PAAS_PROJECT_ID
              value: a2cd8e998dca42e98a41f596c636dbda
          resources: {}
          volumeMounts:
            - name: bs-sata-mountoptionpvc
              mountPath: /tmp
          terminationMessagePath: /dev/termination-log
          terminationMessagePolicy: File
          imagePullPolicy: IfNotPresent
      restartPolicy: Always
      terminationGracePeriodSeconds: 30
      dnsPolicy: ClusterFirst
      securityContext: {}
      imagePullSecrets:
        - name: default-secret
      affinity: {}
      schedulerName: default-scheduler
  volumeClaimTemplates:
    - metadata:
        name: bs-sata-mountoptionpvc
        namespace: default
        annotations:
          everest.io/disk-volume-type: SATA
      spec:
        accessModes:
          - ReadWriteOnce
        resources:
          requests:
            storage: 10Gi
        storageClassName: csi-disk
```

```
  serviceName: wwww
  podManagementPolicy: OrderedReady
  updateStrategy:
    type: RollingUpdate
  revisionHistoryLimit: 10
```

表9-21 关键参数说明

前置路径	参数	描述
metadata	name	创建的工作负载名称。
spec.template.spec.containers	image	工作负载的镜像。
spec.template.spec.containers.volumeMount	mountPath	容器内挂载路径，示例中挂载到“/tmp”路径。
spec	serviceName	工作负载对应的服务，服务创建过程请参见6.3 创建有状态负载(StatefulSet)。

说明

“spec.template.spec.containers.volumeMounts.name ”和

“spec.volumeClaimTemplates.metadata.name”有映射关系，必须保持一致。

- **1.13 及之前版本，yaml 示例如下：**

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: deploy-evs-sata-in
  namespace: default
  generation: 1
  labels:
    appgroup: ''
  annotations:
    container.io/container-0: ***/swr/dockerimage/nginx.png
    description: ''
spec:
  replicas: 1
  selector:
    matchLabels:
      app: deploy-evs-sata-in
      failure-domain.beta.kubernetes.io/region: region01
      failure-domain.beta.kubernetes.io/zone: AZ01
  template:
    metadata:
      labels:
        app: deploy-evs-sata-in
        failure-domain.beta.kubernetes.io/region: region01
        failure-domain.beta.kubernetes.io/zone: AZ01
      annotations:
        metrics.alpha.kubernetes.io/custom-endpoints:
'[{"api":"","path":"","port":"","names":""}]'
```

```
      pod.alpha.kubernetes.io/initialized: 'true'
  spec:
    containers:
      - name: container-0
        image: 'nginx:1.12-alpine-perl'
        env:
          - name: PAAS_APP_NAME
            value: deploy-evs-sata-in
          - name: PAAS_NAMESPACE
            value: default
          - name: PAAS_PROJECT_ID
            value: a2cd8e998dca42e98a41f596c636dbda
        resources: {}
        volumeMounts:
          - name: bs-sata-mountoptionpvc
            mountPath: /tmp
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
        imagePullPolicy: IfNotPresent
    restartPolicy: Always
    terminationGracePeriodSeconds: 30
    dnsPolicy: ClusterFirst
    securityContext: {}
    imagePullSecrets:
      - name: default-secret
    affinity: {}
    schedulerName: default-scheduler
  volumeClaimTemplates:
    - metadata:
      name: bs-sata-mountoptionpvc
      annotations:
        volume.beta.kubernetes.io/storage-class: sata
        volume.beta.kubernetes.io/storage-provisioner: flexvolume-
huawei.com/fuxivol

    spec:
      accessModes:
        - ReadWriteMany
      resources:
        requests:
          storage: 10Gi
  serviceName: wwww
  podManagementPolicy: OrderedReady
  updateStrategy:
    type: RollingUpdate
  revisionHistoryLimit: 10
```

表9-22 关键参数说明

前置路径	参数	描述
metadata	name	创建的工作负载名称。
spec.template.spec.containers	image	工作负载的镜像。

前置路径	参数	描述
spec.template.spec.containers.volumeMount	mountPath	容器内挂载路径，示例中挂载到“/tmp”路径。
spec	serviceName	工作负载对应的服务，服务创建过程请参见6.3 创建有状态负载(StatefulSet)。

说明

“spec.template.spec.containers.volumeMounts.name ”和
“spec.volumeClaimTemplates.metadata.name”有映射关系，必须保持一致。

步骤 2 执行如下命令创建 Pod。

kubectl create -f evs-pod-example.yaml

创建完成后，登录 CCE 控制台，在左侧导航栏中选择“存储管理 > 云硬盘存储卷”。单击 PVC 名称，在 PVC 详情页面可查看云硬盘和 PVC 的绑定关系。

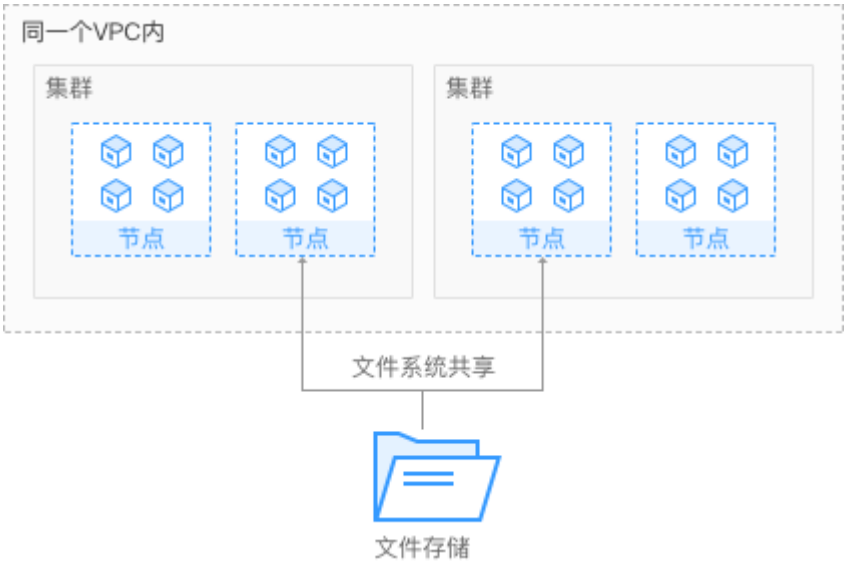
----结束

9.4 文件存储卷

9.4.1 文件存储卷使用说明

文件存储（SFS）适用于多种使用场景，如媒体处理、内容管理、大数据、分析工作负载程序等。

图9-2 CCE 挂载文件存储卷



文件存储说明

- 符合标准文件协议：用户可以将文件系统挂载给服务器，像使用本地文件目录一样。
- 数据共享：多台服务器可挂载相同的文件系统，数据可以共享操作和访问。
- 私有网络：数据访问必须在数据中心内部网络中。
- 单文件系统容量较高（PB 级），性能极佳（IO 读写时延 ms 级），主要面向媒体编辑、HPC、文件共享等场景。
- 适用于多读多写（ReadWriteMany）场景下的各种工作负载（Deployment/StatefulSet）和普通任务（Job）使用。

9.4.2 使用文件存储卷

约束与限制

- CCE v1.7.3-r2 前的集群版本不支持创建文件存储服务，请参照界面要求创建新版本集群，再使用文件存储服务。
- CCE 集群中的容器存储目前已支持加密（Kubernetes 1.13 版本及以上），当前仅在部分区域（Region）提供端到端支持。
- 存储不支持选择企业项目，新创建的存储卷默认创建到 default 企业项目下。

创建文件存储卷

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 存储管理”。

步骤 2 在“文件存储卷”页签，单击“创建文件存储卷”。

步骤 3 配置如表 9-23 基本信息。

表9-23 创建文件存储基本信息配置

参数	参数说明
* PVC 名称	新建 PVC 的名称，该 PVC 名称不同于卷名称，实际的卷名称在 PVC 创建时自动生成。
集群名称	文件存储部署所在的集群。
命名空间	选择对应的命名空间。
总容量	总容量是创建的单个卷的容量，使用费用按照实际使用的容量进行计费。
访问模式	访问模式（AccessModes）是用来对 PV 进行访问模式的设置，用于描述用户应用对存储资源的访问权限。 • ReadWriteOnce (RWO)：文件存储卷只能以读写模式被单个节点同时加载。 • ReadWriteMany (RWX)：文件存储卷能够以读写模式被多个节点同时加载。

参数	参数说明
存储格式	默认为 CSI，暂不支持更改。
加密	“KMS 加密”默认不勾选。 勾选“KMS 加密”后，将为文件存储提供易用、高安全性的密钥管理服务（Key Management Service，KMS）。若未创建委托请单击“创建委托”，并配置如下参数： • 委托名称：委托表示委托方通过创建信任关系，给被委托方授予访问其资源的权限。当“委托名称”为“SFSAccessKMS”时，表示已经成功授权 SFS 访问 KMS，授权成功后，SFS 可以获取 KMS 密钥用来加解密文件系统。 • 密钥名称：密钥资源创建完成后，可在容器应用中加载使用。 • 密钥 ID：默认生成。 该功能仅在部分 Region 的 1.13.10 及以上版本的集群中支持。

步骤 4 单击“立即创建”，等待创建成功。

创建成功后，存储管理列表中会出现已创建的存储，待 PVC 状态为“正常”，表示存储已创建成功。

步骤 5 单击存储名称，进入存储详情页，可查看到存储的卷名称、存储容量等信息。

----结束

添加文件存储卷

步骤 1 参照 6.2 创建无状态负载(Deployment)、6.3 创建有状态负载(StatefulSet)、6.4 创建守护进程集(DaemonSet)或 6.5 创建普通任务(Job)创建工作负载或普通任务，在添加容器后，展开“数据存储”。在“云存储”页签，单击“添加云存储”。

步骤 2 选择存储类型为“文件存储”。

表9-24 挂载文件存储参数配置

参数	参数说明
云存储类型	文件存储：文件存储适用于多种使用场景，包括媒体处理、内容管理、大数据和分析应用程序等。
分配方式	
使用已有存储	• 云存储名称：选择已创建的存储，您需要提前创建存储，创建存储步骤请参见创建文件存储卷。 • 子类型：已创建的文件存储子类型。 • 存储容量：该容量值为 PVC 的属性值，若在 IaaS 侧实施过存

参数	参数说明
	储扩容后，容量值不一致是正常的。CCE1.13 集群开始支持端到端容器存储扩容功能后，PVC 容量才会和存储实体容量一致
自动分配存储	自动创建存储，需要输入存储的容量。 • 子类型：文件存储子类型为 NFS。 • 存储容量：单位为 GB。请不要超过存储容量配额，否则会创建失败。 • 存储格式：默认选择 CSI。 CSI，即 Container Storage Interface，容器存储接口机制，用于在 Kubernetes 和外部存储系统之间建立一套标准的存储管理接口，通过该接口为容器提供存储服务。 • 加密：勾选“KMS 加密”后，将为文件存储提供易用、高安全性的密钥管理服务（Key Management Service，KMS），该功能仅在部分 Region 的 1.13.10 及以上版本的集群中支持。若未创建委托请单击“创建委托”，并配置如下参数： – 委托名称：委托表示委托方通过创建信任关系，给被委托方授予访问其资源的权限。当“委托名称”为“SFSAccessKMS”时，表示已经成功授权 SFS 访问 KMS，授权成功后，SFS 可以获取 KMS 密钥用来加解密文件系统。 – 密钥名称：密钥资源创建完成后，可在容器应用中加载使用。 – 密钥 ID：默认生成。
添加容器挂载	配置如下参数： 1. 子路径：输入文件存储的子路径，如：tmp。 Kubernetes 中数据卷挂载的 subpath，指引用卷内的子路径而不是其根，不填写时默认为根。现只支持文件存储，必须是相对路径，且不能以“/”或“../”开头。 2. 挂载路径：输入挂载路径，如：/tmp。 数据存储挂载到容器上的路径，请不要挂载在系统目录下，如“/”、“**/var/run**”等，会导致容器异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响容器启动的文件，否则文件会被替换，导致容器启动异常，工作负载创建失败。 须知 挂载高危目录的情况下 ，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。 3. 设置权限。 • 只读：只能读容器路径中的数据卷。 • 读写：可修改容器路径中的数据卷，容器迁移时新写入的数据不会随之迁移，会造成数据丢失。 单击“添加容器挂载”可增加多条设置，单击“确定”完成配

参数	参数说明
	置。

步骤 3 单击“确定”。

----结束

导入文件存储卷

CCE 支持导入已有的文件存储。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 存储管理”，在“文件存储卷”页签，单击“导入”。

步骤 2 从列表里选择要导入的文件存储。

步骤 3 选择需要导入文件存储的集群名称和命名空间名称，单击“确定”。

----结束

解关联文件存储卷

创建或导入文件存储卷成功后，文件存储卷会自动和当前集群绑定，其它集群也可以使用该存储。解关联后，将和此集群解绑，其它集群仍然可以导入使用该文件存储卷。

若文件存储卷已被工作负载挂载，将无法和集群解关联。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 存储管理”，在文件存储卷列表中，单击文件存储卷后的“解关联”。

步骤 2 确认解关联后，单击“确定”。

----结束

相关操作

文件存储创建完成后，您还可以执行表 9-25 中的操作。

表9-25 其他操作

操作	说明
删除文件存储	1. 选择需要删除的存储名称，单击操作列的“删除”。 2. 根据界面提示完成删除操作。
导入文件存储	CCE 支持导入已有的文件存储。 1. 在“文件存储卷”页签下，单击“导入”。 2. 从列表里选择要导入的文件存储。 3. 选择需要导入文件存储的集群名称和命名空间名称。

操作	说明
	4. 单击“确定”。

9.4.3 使用 kubectl 自动创建文件存储

操作场景

CCE 支持使用 PersistentVolumeClaim（PVC）的形式创建文件存储。

操作步骤

步骤 1 请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令。

步骤 2 执行如下命令，配置名为“pvc-sfs-auto-example.yaml”的创建 PVC 的 yaml 文件。

touch pvc-sfs-auto-example.yaml

vi pvc-sfs-auto-example.yaml

- **1.15 及以上版本的集群，yaml 文件配置示例如下：**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name:  pvc-sfs-auto-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  storageClassName: csi-nas
```

表9-26 关键参数说明

参数	描述
storageClassName	k8s storage class 名称；使用“csi-nas”。
name	创建的 PVC 名称。
accessModes	只支持 ReadWriteMany，不支持 ReadWriteOnly。
storage	存储容量，单位为 Gi。

- **1.13 以及之前版本集群，yaml 文件配置示例如下：**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
```

```
    volume.beta.kubernetes.io/storage-class: nfs-rw
  name: pvc-sfs-auto-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
```

表9-27 关键参数说明

参数	描述
volume.beta.kubernetes.io/storage-class	文件存储类型，当前支持标准文件协议类型（nfs-rw）。
name	创建的 PVC 名称。
accessModes	只支持 ReadWriteMany，不支持 ReadWriteOnly。
storage	存储容量，单位为 Gi。

步骤 3 执行如下命令创建 PVC。

kubectl create -f pvc-sfs-auto-example.yaml

命令执行完成后会在集群所在 VPC 内创建一个文件存储，您可以在“存储管理 > 文件存储卷”中查看该文件系统，也可以在 SFS 的控制台中查看该文件系统。

----结束

9.4.4 使用 kubectl 对接已有文件存储

操作场景

CCE 支持使用已有的文件存储来创建 PersistentVolume，创建成功后，通过创建相应的 PersistentVolumeClaim 来绑定当前的 PersistentVolume 使用。

操作步骤

步骤 1 登录 SFS 控制台，创建一个文件存储，记录文件存储的 ID、共享路径和容量。

步骤 2 请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令。

步骤 3 新建两个 yaml 文件，用于创建 PersistentVolume（PV）、PersistentVolumeClaim（PVC）。假设文件名分别为 **pv-sfs-example.yaml**、**pvc-sfs-example.yaml**。

touch pv-sfs-example.yaml pvc-sfs-example.yaml

K8S 集群版本（K8S version）	说明	yaml 示例

K8S 集群版本（K8S version）	说明	yaml 示例
1.15+	1.15 及以上版本的集群	请参见 1.15+ yaml 文件配置示例
1.11 ≤ K8S version < 1.13	1.11 以上及 1.13 版本集群	请参见 1.11~1.13 yaml 文件配置示例
K8S version = 1.9	1.9 版本集群	请参见 1.9 yaml 文件配置示例

1.15+（1.15 及以上版本的集群）

- **vi pv-sfs-example.yaml**

  PV yaml 文件配置示例：

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-sfs-example
  annotations:
    pv.kubernetes.io/provisioned-by: everest-csi-provisioner
spec:
  accessModes:
  - ReadWriteMany
  capacity:
    storage: 10Gi
  claimRef:
    apiVersion: v1
    kind: PersistentVolumeClaim
    name: pvc-sfs-example
    namespace: default
  csi:
    driver: nas.csi.everest.io
    fsType: nfs
    volumeAttributes:
      everest.io/share-export-location: sfs-nas01.cn-north-
4.myhuaweicloud.com:/share-436304e8
      storage.kubernetes.io/csiProvisionerIdentity: everest-csi-provisioner
    volumeHandle: 682f00bb-ace0-41d8-9b3e-913c9aa6b695
  persistentVolumeReclaimPolicy: Delete
  storageClassName: csi-nas
```

表9-28 关键参数说明

参数	描述
driver	挂载依赖的存储驱动，文件存储配置为“nas.csi.everest.io”。
everest.io/share-export-location	文件存储的共享路径。 获取方法：在 CCE 控制台中，单击顶部的“服务列

参数	描述
	表 > 存储 > 弹性文件服务”，在弹性文件服务列表中可以看到“挂载地址”列，即为文件存储的共享路径。
volumeHandle	文件存储的 ID。 获取方法：在 CCE 控制台中，单击顶部的“服务列表 > 存储 > 弹性文件服务”，在弹性文件服务列表中单击对应的弹性文件服务名称，在详情页中复制“ID”后的内容即可。
storage	文件存储的大小。
storageClassName	k8s storage class 名称；需配置为"csi-nas"。
spec.claimRef.apiVersion	固定值"v1"。
spec.claimRef.kind	固定值"PersistentVolumeClaim"。
spec.claimRef.name	pvc 名称；与下一步创建的 pvc 的 name 一致。
spec.claimRef.namespace	pvc 的 namespace；与下一步创建的 pvc 的 namespace 一致。

- **vi pvc-sfs-example.yaml**

  PVC yaml 文件配置示例：

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-provisioner: everest-csi-provisioner
  name: pvc-sfs-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  storageClassName: csi-nas
  volumeName: pv-sfs-example
```

表9-29 关键参数说明

参数	描述
storageClassName	需配置为"csi-nas"。必须和已有 PV 保持一致。
storage	存储容量，单位 Gi，必须和已有 pv 的 storage 大小保持一致。
volumeName	PV 的名称。

说明

文件存储所在 VPC 必须与工作负载规划部署的 ECS 虚拟机的 VPC 保持一致。

步骤 4 创建 PV。

kubectl create -f pv-sfs-example.yaml

步骤 5 创建 PVC。

kubectl create -f pvc-sfs-example.yaml

----结束

历史版本示例

1.11 ≤ K8S version ≤ 1.13（1.11 以上及 1.13 版本集群）

- **PV yaml 文件配置示例：**

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-sfs-example
  annotations:
    pv.kubernetes.io/provisioned-by: flexvolume-huawei.com/fuxinfs
spec:
  accessModes:
  - ReadWriteMany
  capacity:
    storage: 10Gi
  claimRef:
    apiVersion: v1
    kind: PersistentVolumeClaim
    name: pvc-sfs-example
    namespace: default
  flexVolume:
    driver: huawei.com/fuxinfs
    fsType: nfs
    options:
      deviceMountPath: sfs-nas1.southchina.huaweicloud.com:/share-73bdfafd
      fsType: nfs
      volumeID: f6976f9e-2493-419b-97ca-d7816008d91c
  persistentVolumeReclaimPolicy: Delete
  storageClassName: nfs-rw
```

表9-30 关键参数说明

参数	描述
driver	挂载依赖的存储驱动，文件存储配置为“huawei.com/fuxinfs”。
deviceMountPath	文件存储的共享路径。

参数	描述
	获取方法：在 CCE 控制台中，单击顶部的“服务列表 > 存储 > 弹性文件服务”，在弹性文件服务列表中可以看到“挂载地址”列，即为文件存储的共享路径。
volumeID	文件存储的 ID。 获取方法：在 CCE 控制台中，单击左侧栏目树中的“资源管理-存储管理”，在“文件存储卷”页签下单击 PVC 的名称，在 PVC 详情页中复制“PVC UID”后的内容即可。
storage	文件存储的大小。
storageClassName	文件存储支持的读写方式，当前支持 nfs-rw、nfs-ro。
spec.claimRef.apiVersion	固定值"v1"。
spec.claimRef.kind	固定值"PersistentVolumeClaim"。
spec.claimRef.name	与下一步创建的 pvc 的 name 一致。
spec.claimRef.namespace	pvc 的 namespace；是下一步创建的 pvc 的 namespace 一致。

- **PVC yaml 文件配置示例：**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: nfs-rw
    volume.beta.kubernetes.io/storage-provisioner: flexvolume-
huawei.com/fuxinfs
  name: pvc-sfs-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  volumeName: pv-sfs-example
```

表9-31 关键参数说明

参数	描述
volume.beta.kubernetes.io/storage-class	文件存储支持的读写方式，支持 nfs-rw 、nfs-ro。必须和已有 PV 保持一致。
volume.beta.kubernetes.io/storage-provisioner	必须使用 flexvolume-huawei.com/fuxinfs。

参数	描述
storage	存储容量，单位 Gi，必须和已有 pv 的 storage 大小保持一致。
volumeName	PV 的名称。

K8S version = 1.9（1.9 版本集群）

- **PV yaml 文件配置示例：**

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-sfs-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  capacity:
    storage: 10Gi
  flexVolume:
    driver: huawei.com/fuxinfs
    fsType: nfs
    options:
      deviceMountPath: sfs-nas1.southchina.huaweicloud.com:/share-73bdfafd
      fsType: nfs
      kubernetes.io/namespace: default
      volumeID: f6976f9e-2493-419b-97ca-d7816008d91c
  persistentVolumeReclaimPolicy: Delete
  storageClassName: nfs-rw
```

表9-32 关键参数说明

参数	描述
driver	挂载依赖的存储驱动，文件存储配置为“huawei.com/fuxinfs”。
deviceMountPath	文件存储的共享路径。 获取方法：在 CCE 控制台中，单击顶部的“服务列表 > 存储 > 弹性文件服务”，在弹性文件服务列表中可以看到“挂载地址”列，即为文件存储的共享路径。
volumeID	文件存储的 ID。 获取方法：在 CCE 控制台中，单击左侧栏目树中的“资源管理-存储管理”，在“文件存储卷”页签下单击 PVC 的名称，在 PVC 详情页中复制“PVC UID”后的内容即可。
storage	文件存储的大小。
storageClassName	文件存储支持的读写方式，当前支持 nfs-rw、nfs-ro。

- **PVC yaml 文件配置示例：**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: nfs-rw
    volume.beta.kubernetes.io/storage-provisioner: flexvolume-
huawei.com/fuxinfs
  name: pvc-sfs-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  volumeName: pv-sfs-example
  volumeNamespace: default
```

表9-33 关键参数说明

参数	描述
volume.beta.kubernetes.io/storage-class	文件存储支持的读写方式，支持 nfs-rw 、nfs-ro。必须和已有 PV 保持一致。
volume.beta.kubernetes.io/storage-provisioner	必须使用 flexvolume-huawei.com/fuxinfs。
storage	存储容量，单位 Gi，必须和已有 pv 的 storage 大小保持一致。
volumeName	PV 的名称。

9.4.5 使用已有 PVC 创建有状态工作负载

操作场景

CCE 支持使用已有的文件存储（PersistentVolumeClaim），创建有状态工作负载（StatefulSet）。

操作步骤

步骤 1 参照创建文件存储卷中操作创建文件存储卷，记录文件存储卷名称。

步骤 2 请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令。

步骤 3 新建一个 PVC 文件，用于创建工作负载。假设文件名为 **sfs-statefulset-example.yaml**。

touch sfs-statefulset-example.yaml

vi sfs-statefulset-example.yaml

配置示例：

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: sfs-statefulset-example
  namespace: default
spec:
  podManagementPolicy: OrderedReady
  replicas: 2
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: sfs-statefulset-example
  serviceName: qwqq
  template:
    metadata:
      annotations:
        metrics.alpha.kubernetes.io/custom-endpoints:
'[{"api":"","path":"","port":"","names":""}]'
        pod.alpha.kubernetes.io/initialized: "true"
      creationTimestamp: null
      labels:
        app: sfs-statefulset-example
    spec:
      affinity: {}
      containers:
      - env:
        - name: PAAS_APP_NAME
          value: sfs-statefulset-example
        - name: PAAS_NAMESPACE
          value: default
        - name: PAAS_PROJECT_ID
          value: b7bb7d77a2974a8fa8985cbfb63f23c0
        image: nginx:latest
        imagePullPolicy: Always
        name: container-0
        resources: {}
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
        volumeMounts:
        - mountPath: /tmp
          name: pvc-sfs-example
      dnsPolicy: ClusterFirst
      imagePullSecrets:
      - name: default-secret
      restartPolicy: Always
      schedulerName: default-scheduler
      securityContext: {}
      terminationGracePeriodSeconds: 30
      volumes:
        - name: pvc-sfs-example
          persistentVolumeClaim:
            claimName: cce-sfs-demo
```

```
      tolerations:
      - effect: NoExecute
        key: node.kubernetes.io/not-ready
        operator: Exists
        tolerationSeconds: 300
      - effect: NoExecute
        key: node.kubernetes.io/unreachable
        operator: Exists
        tolerationSeconds: 300
  updateStrategy:
    type: RollingUpdate
```

表9-34 关键参数说明

前置路径	参数	描述
spec	replicas	实例数。
metadata	name	新建工作负载的名称。
spec.template.spec.containers	image	新建工作负载使用的镜像。
spec.template.spec.containers.volumeMounts	mountPath	容器内挂载路径。
spec	serviceName	工作负载对应的服务，服务创建过程请参见 6.3 创建有状态负载(StatefulSet)。
spec.template.spec.volumes.persistentVolumeClaim	claimName	已有 PVC 名称。

说明

spec.template.spec.containers.volumeMounts.name 和 spec.template.spec.volumes.name 有映射关系，必须保持一致。

步骤 4 创建有状态工作负载。

kubectl create -f sfs-statefulset-example .yaml

----结束

9.4.6 使用 kubectl 部署带文件存储卷的工作负载

操作场景

文件存储卷创建或导入 CCE 后，可以在工作负载中挂载文件存储卷。

操作步骤

步骤 1 执行如下命令，配置名为“sfs-pod-example.yaml”的创建 Pod 的 yaml 文件。

touch sfs-pod-example.yaml

vi sfs-pod-example.yaml

在无状态工作负载中基于 pvc 共享式使用文件存储示例：

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: sfs-pod-example                                 # 工作负载名称
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: sfs-pod-example
  template:
    metadata:
      labels:
        app: sfs-pod-example
    spec:
      containers:
      - image: nginx
        name: container-0
        volumeMounts:
        - mountPath: /tmp                               # 挂载路径
          name: pvc-sfs-example
      restartPolicy: Always
      volumes:
      - name: pvc-sfs-example
        persistentVolumeClaim:
          claimName: pvc-sfs-auto-example               # 挂载 PVC
```

表9-35 关键参数说明

前置路径	参数	描述
metadata	name	创建的 Pod 名称。
spec.template.spec.containers.volumeMounts	mountPath	容器内挂载路径，此处示例中为“/tmp”。
spec.template.spec.volumes.persistentVolumeClaim	claimName	已有 PVC 名称。

说明

“spec.template.spec.containers.volumeMounts.name” 和 “spec.template.spec.volumes.name”有映射关系，必须保持一致。

在有状态工作负载中基于 PVCTemplate 独占式使用文件存储：

1.15 及以上版本的集群，yaml 配置示例如下：

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: deploy-sfs-nfs-rw-in
  namespace: default
  generation: 1
  labels:
    appgroup: ''
  annotations:
    container.io/container-0: 'https:///swr/dockerimage/nginx.png'
    description: ''
spec:
  replicas: 2
  selector:
    matchLabels:
      app: deploy-sfs-nfs-rw-in
  template:
    metadata:
      labels:
        app: deploy-sfs-nfs-rw-in
      annotations:
        metrics.alpha.kubernetes.io/custom-endpoints:
'[{"api":"","path":"","port":"","names":""}]'
        pod.alpha.kubernetes.io/initialized: 'true'
    spec:
      containers:
        - name: container-0
          image: 'nginx:1.12-alpine-perl'
          env:
            - name: PAAS_APP_NAME
              value: deploy-sfs-nfs-rw-in
            - name: PAAS_NAMESPACE
              value: default
            - name: PAAS_PROJECT_ID
              value: 8190a2a1692c46f284585c56fc0e2fb9
          resources: {}
          volumeMounts:
            - name: bs-nfs-rw-mountoptionpvc
              mountPath: /aaa
          terminationMessagePath: /dev/termination-log
          terminationMessagePolicy: File
          imagePullPolicy: IfNotPresent
      restartPolicy: Always
      terminationGracePeriodSeconds: 30
      dnsPolicy: ClusterFirst
      securityContext: {}
      imagePullSecrets:
        - name: default-secret
      affinity: {}
      schedulerName: default-scheduler
  volumeClaimTemplates:
    - metadata:
        name: bs-nfs-rw-mountoptionpvc
        namespace: default
        annotations: {}
```

```
      enable: true
    spec:
      accessModes:
        - ReadWriteMany
      resources:
        requests:
          storage: 11Gi
      storageClassName: csi-nas
  serviceName: wwww
  podManagementPolicy: OrderedReady
  updateStrategy:
    type: RollingUpdate
  revisionHistoryLimit: 10
```

表9-36 关键参数说明

前置路径	参数	描述
metadata	name	创建的工作负载名称。
spec.template.spec.containers	image	工作负载的镜像。
spec.template.spec.containers.volumeMount	mountPath	容器内挂载路径，此处示例中为“/tmp”。
spec	serviceName	工作负载对应的服务，服务创建过程请参见 6.3 创建有状态负载(StatefulSet)。

说明

“spec.template.spec.containers.volumeMounts.name ”和“spec.volumeClaimTemplates.metadata.name”有映射关系，必须保持一致。

1.13 及之前集群版本示例：

```
apiVersion: apps/v1
kind: StatefulSet

metadata:
  name: deploy-sfs-nfs-rw-in
  namespace: default
  generation: 1
  labels:
    appgroup: ''
  annotations:
    container.io/container-0: 'https://***/swr/dockerimage/nginx.png'
    description: ''
spec:
  replicas: 2
  selector:
    matchLabels:
      app: deploy-sfs-nfs-rw-in
  template:
```

```
    metadata:
      labels:
        app: deploy-sfs-nfs-rw-in
      annotations:
        metrics.alpha.kubernetes.io/custom-endpoints:
'[{"api":"","path":"","port":"","names":""}]'
        pod.alpha.kubernetes.io/initialized: 'true'
    spec:
      containers:
        - name: container-0
          image: 'nginx:1.12-alpine-perl'
          env:
            - name: PAAS_APP_NAME
              value: deploy-sfs-nfs-rw-in
            - name: PAAS_NAMESPACE
              value: default
            - name: PAAS_PROJECT_ID
              value: 8190a2a1692c46f284585c56fc0e2fb9
          resources: {}
          volumeMounts:
            - name: bs-nfs-rw-mountoptionpvc
              mountPath: /aaa
          terminationMessagePath: /dev/termination-log
          terminationMessagePolicy: File
          imagePullPolicy: IfNotPresent
      restartPolicy: Always
      terminationGracePeriodSeconds: 30
      dnsPolicy: ClusterFirst
      securityContext: {}
      imagePullSecrets:
        - name: default-secret
      affinity: {}
      schedulerName: default-scheduler
  volumeClaimTemplates:
    - metadata:
        name: bs-nfs-rw-mountoptionpvc
        creationTimestamp: null
        annotations:
          volume.beta.kubernetes.io/storage-class: nfs-rw
          volume.beta.kubernetes.io/storage-provisioner: flexvolume-
huawei.com/fuxinfs
      spec:
        accessModes:
          - ReadWriteMany
        resources:
          requests:
            storage: 11Gi
  serviceName: wwww
  podManagementPolicy: OrderedReady
  updateStrategy:
    type: RollingUpdate
  revisionHistoryLimit: 10
```

表9-37 关键参数说明

前置路径	参数	描述
metadata	name	创建的工作负载名称。
spec.template.spec.containers	image	工作负载的镜像。
spec.template.spec.containers.volumeMount	mountPath	容器内挂载路径，此处示例中为“/tmp”。
spec	serviceName	工作负载对应的服务，服务创建过程请参见 6.3 创建有状态负载(StatefulSet)。

说明

“spec.template.spec.containers.volumeMounts.name ”和
“spec.volumeClaimTemplates.metadata.name”有映射关系，必须保持一致。

步骤 2 执行如下命令创建 Pod。

kubectl create -f sfs-pod-example.yaml

创建完成后，登录 CCE 控制台，在左侧导航栏中选择“存储管理 > 文件存储卷”。单击 PVC 名称，在 PVC 详情页面可查看文件存储服务和 PVC 的绑定关系。

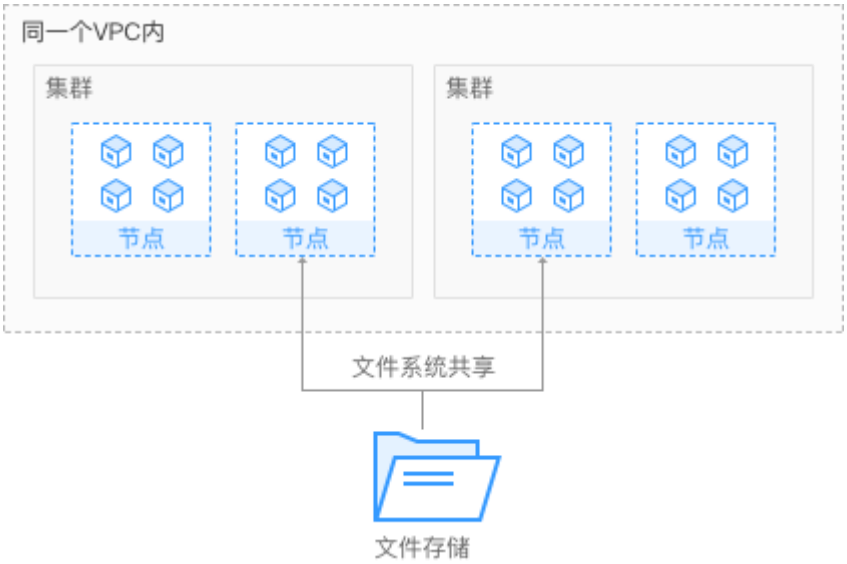
----结束

9.5 极速文件存储卷

9.5.1 极速文件存储卷使用说明

极速文件存储（SFS Turbo）具有按需申请，快速供给，弹性扩展，方便灵活等特点，适用于多种使用场景，主要面向 DevOps、容器微服务、企业办公等场景。

图9-3 CCE 挂载极速文件存储卷



极速文件存储说明

- 符合标准文件协议：用户可以将文件系统挂载给服务器，像使用本地文件目录一样。
- 数据共享：多台服务器可挂载相同的文件系统，数据可以共享操作和访问。
- 私有网络：数据访问必须在数据中心内部网络中。
- 直接使用云上现有 IAAS 服务构建独享的云文件存储，为租户提供数据隔离保护和 IOPS 性能保障，主要面向 DevOps、容器微服务、企业办公等场景。
- 适用于多读多写（ReadWriteMany）场景下的各种工作负载（Deployment/StatefulSet）和普通任务（Job）使用。

9.5.2 使用极速文件存储卷

约束与限制

暂不支持直接创建极速文件存储卷，您可以参照界面要求前往 SFS Turbo 控制台创建后，再使用极速文件存储服务。

导入极速文件存储卷

CCE 支持导入已有的极速文件存储。

步骤 1 登录 CCE 控制台，单击左侧导航栏的“资源管理 > 存储管理”，在“极速文件存储卷”页签下，单击“导入”。

步骤 2 从列表里选择要导入的极速文件存储。

步骤 3 选择需要导入极速文件存储的集群名称和命名空间名称。

步骤 4 单击“确定”。导入成功后，存储管理列表中会出现已导入的存储，待 PVC 状态为“正常”，表示存储已导入成功。

----结束

添加极速文件存储卷

步骤 1 参照 6.2 创建无状态负载(Deployment)、6.3 创建有状态负载(StatefulSet)、6.4 创建守护进程集(DaemonSet)或 6.5 创建普通任务(Job)创建工作负载或普通任务，在添加容器后，展开“数据存储”，在“云存储”页签，单击“添加云存储”。

步骤 2 选择存储类型为“极速文件存储”。

表9-38 极速文件存储参数说明

参数	参数说明
云存储类型	极速文件存储：适用于多种使用场景，主要面向 DevOps、容器微服务、企业办公等场景。
分配方式	
使用已有存储	云存储名称：选择已创建的存储。
添加容器挂载	配置如下参数： 1. 子路径：输入文件存储的子路径，如：tmp。 Kubernetes 中数据卷挂载的 subpath，指引用卷内的子路径而不是其根，不填写时默认为根。现只支持文件存储，必须是相对路径，且不能以“/”或“../”开头。 2. 挂载路径：输入挂载路径，如：/tmp。 数据存储挂载到容器上的路径，请不要挂载在系统目录下，如“/”、“**/var/run**”等，会导致容器异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响容器启动的文件，否则文件会被替换，导致容器启动异常，工作负载创建失败。 须知 挂载高危目录的情况下 ，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。 3. 设置权限。 • 只读：只能读容器路径中的数据卷。 • 读写：可修改容器路径中的数据卷，容器迁移时新写入的数据不会随之迁移，会造成数据丢失。 单击“添加容器挂载”可增加多条设置，单击“确定”完成配置。

步骤 3 单击“确定”。

----结束

解关联极速文件存储卷

导入极速文件存储卷成功后，极速文件存储卷会自动和当前集群绑定，其它集群也可以导入该存储。解关联后，将和此集群解绑，其它集群仍然可以导入使用该文件存储卷。

若极速文件存储卷已被工作负载挂载，将无法和集群解关联。

步骤 1 登录 CCE 控制台，单击左侧导航栏的“资源管理 > 存储管理”，在极速文件存储卷列表中，单击极速文件存储卷后的“解关联”。

步骤 2 查看系统提示，单击“确定”。

----结束

9.5.3 使用 kubectl 对接已有极速文件存储卷

操作场景

CCE 支持使用已有的极速文件存储来创建 PersistentVolume，创建成功后，通过创建相应的 PersistentVolumeClaim 来绑定当前的 PersistentVolume 使用。

操作步骤

步骤 1 登录 SFS 控制台，创建一个文件存储，记录文件存储的 ID、共享路径和容量。

步骤 2 请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令。

步骤 3 新建两个 yaml 文件，用于创建 PersistentVolume（PV）、PersistentVolumeClaim（PVC）。假设文件名分别为 **pv-efs-example.yaml**、**pvc-efs-example.yaml**。

touch pv-efs-example.yaml pvc-efs-example.yaml

K8S 集群版本（K8S version）	说明	yaml 示例
1.15+	1.15 及以上版本的集群	请参见 1.15+ yaml 文件配置示例
K8S version ≤ 1.13	1.13 及之前版本的集群	请参见 1.13- yaml 文件配置示例

1.15+（1.15 及以上版本的集群）

- **vi pv-efs-example.yaml**

  PV yaml 文件配置示例如下：

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-efs-example
  annotations:
```

```
    pv.kubernetes.io/provisioned-by: everest-csi-provisioner
spec:
  accessModes:
  - ReadWriteMany
  capacity:
    storage: 10Gi
  csi:
    driver: sfsturbo.csi.everest.io
    fsType: nfs
    volumeAttributes:
      everest.io/share-export-location: 192.168.0.169:/
      storage.kubernetes.io/csiProvisionerIdentity: everest-csi-provisioner
    volumeHandle: 8962a2a2-a583-4b7f-bb74-fe76712d8414
  persistentVolumeReclaimPolicy: Delete
  storageClassName: csi-sfsturbo
```

表9-39 关键参数说明

参数	描述
driver	挂载依赖的存储驱动，极速文件存储配置为“sfsturbo.csi.everest.io”。
everest.io/share-export-location	极速文件存储的共享路径。
volumeHandle	极速文件存储的 ID。 获取方法：在 CCE 控制台中，单击左侧栏目树中的“资源管理-存储管理”，在“极速文件存储卷”页签下单击 PVC 的名称，在 PVC 详情页中复制“PVC UID”后的内容即可。
storage	文件存储的大小。
storageClassName	指定 k8s storage class 名称；极速文件存储卷需配置为"csi-sfsturbo”。

- **vi pvc-efs-example.yaml**

  PVC yaml 文件配置示例如下：

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-provisioner: everest-csi-provisioner
  name: pvc-efs-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
```

```
  storageClassName: csi-sfsturbo
  volumeName: pv-efs-example
```

表9-40 关键参数说明

参数	描述
storageClassName	指定 k8s storage class 名称；需配置为"csi-sfsturbo”。
storage	存储容量，单位 Gi，必须和已有 pv 的 storage 大小保持一致。
volumeName	PV 的名称。

说明

极速文件存储所在 VPC，子网必须与工作负载规划部署的 ECS 虚拟机的 VPC 保持一致，安全组开放入方向端口(111、445、2049、2051、20048)。

步骤 4 创建 PV。

kubectl create -f pv-efs-example.yaml

步骤 5 创建 PVC。

kubectl create -f pvc-efs-example.yaml

----结束

历史版本示例

K8S version ≤ 1.13（1.13 及之前版本集群）

- **PV yaml 文件配置示例如下：**

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-efs-example
  annotations:
    pv.kubernetes.io/provisioned-by: flexvolume-huawei.com/fuxiefs
spec:
  accessModes:
  - ReadWriteMany
  capacity:
    storage: 100Gi
  flexVolume:
    driver: huawei.com/fuxiefs
    fsType: efs
    options:
      deviceMountPath: 192.168.0.169:/
      fsType: efs
      volumeID: 8962a2a2-a583-4b7f-bb74-fe76712d8414
  persistentVolumeReclaimPolicy: Delete
  storageClassName: efs-standard
```

表9-41 关键参数说明

参数	描述
driver	挂载依赖的存储驱动，极速文件存储配置为“huawei.com/fuxiefs”。
deviceMountPath	极速文件存储的共享路径。
volumeID	极速文件存储的 ID。 获取方法：在 CCE 控制台中，单击左侧栏目树中的“资源管理-存储管理”，在“极速文件存储卷”页签下单击 PVC 的名称，在 PVC 详情页中复制“PVC UID”后的内容即可。
storage	文件存储的大小。
storageClassName	极速文件存储支持的卷类型，当前支持 efs-standard、efs-performance（目前 SFS Turbo 不支持动态创建，所以此参数后续没有使用）。

- **PVC yaml 文件配置示例如下：**

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  annotations:
    volume.beta.kubernetes.io/storage-class: efs-standard
    volume.beta.kubernetes.io/storage-provisioner: flexvolume-
huawei.com/fuxiefs
  name: pvc-efs-example
  namespace: default
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 100Gi
  volumeName: pv-efs-example
```

表9-42 关键参数说明

参数	描述
volume.beta.kubernetes.io/storage-class	文件存储支持的读写方式，支持 efs-standard、efs-performance。必须和已有 PV 保持一致。
volume.beta.kubernetes.io/storage-provisioner	必须使用 flexvolume-huawei.com/fuxiefs。
storage	存储容量，单位 Gi，必须和已有 pv 的 storage 大小保持一致。

参数	描述
volumeName	PV 的名称。

9.5.4 使用已有 PVC 创建有状态工作负载

操作场景

CCE 支持使用已有的极速文件存储（SFS Turbo），创建有状态工作负载（StatefulSet）。

操作步骤

步骤 1 参照创建文件存储卷中操作创建极速文件存储卷，记录极速文件存储卷名称。

步骤 2 请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令。

步骤 3 新建一个文件，用于创建工作负载。假设文件名为 **efs-statefulset-example.yaml**。

touch efs-statefulset-example.yaml

vi efs-statefulset-example.yaml

配置示例：

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: efs-statefulset-example
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: efs-statefulset-example
  template:
    metadata:
      annotations:
        metrics.alpha.kubernetes.io/custom-endpoints:
'[{"api":"","path":"","port":"","names":""}]'
        pod.alpha.kubernetes.io/initialized: 'true'
      labels:
        app: efs-statefulset-example
    spec:
      containers:
        - image: 'nginx:1.0.0'
          name: container-0
          resources:
            requests: {}
            limits: {}
          env:
            - name: PAAS_APP_NAME
              value: efs-statefulset-example
```

```
          - name: PAAS_NAMESPACE
            value: default
          - name: PAAS_PROJECT_ID
            value: b18296881cc34f929baa8b9e95abf88b
        volumeMounts:
          - name: efs-statefulset-example
            mountPath: /tmp
            readOnly: false
            subPath: ''
    imagePullSecrets:
      - name: default-secret
    terminationGracePeriodSeconds: 30
    volumes:
      - persistentVolumeClaim:
          claimName: cce-efs-import-jnr481gm-3y5o
        name: efs-statefulset-example
    affinity: {}
    tolerations:
      - key: node.kubernetes.io/not-ready
        operator: Exists
        effect: NoExecute
        tolerationSeconds: 300
      - key: node.kubernetes.io/unreachable
        operator: Exists
        effect: NoExecute
        tolerationSeconds: 300
  podManagementPolicy: OrderedReady
  serviceName: test
  updateStrategy:
    type: RollingUpdate
```

表9-43 关键参数说明

参数	描述
replicas	实例数。
name	新建工作负载的名称。
image	新建工作负载使用的镜像。
mountPath	容器内挂载路径。
serviceName	工作负载对应的服务，服务创建过程请参见 6.3 创建有状态负载(StatefulSet)。
claimName	已有 PVC 名称。

说明

spec.template.spec.containers.volumeMounts.name 和 spec.template.spec.volumes.name 有映射关系，必须保持一致。

步骤 4 创建有状态工作负载。

kubectl create -f efs-statefulset-example.yaml

----结束

9.5.5 使用 kubectl 挂载极速文件存储

操作场景

极速文件存储创建或导入 CCE 后，可以在工作负载中挂载极速文件存储。

操作步骤

步骤 1 执行如下命令，配置名为“efs-pod-example.yaml”的创建 Pod 的 yaml 文件。

touch efs-pod-example.yaml

vi efs-pod-example.yaml

在无状态工作负载中基于 pvc 共享式使用极速文件存储示例：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: efs-pod-example                                 # 工作负载名称
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: efs-pod-example
  template:
    metadata:
      labels:
        app: efs-pod-example
    spec:
      containers:
      - image: nginx
        name: container-0
        volumeMounts:
        - mountPath: /tmp                               # 挂载路径
          name: pvc-efs-example
      restartPolicy: Always
      volumes:
      - name: pvc-efs-example
        persistentVolumeClaim:
          claimName: pvc-sfs-auto-example               # 挂载 PVC
```

表9-44 关键参数说明

参数	描述
name	为创建的 Pod 名称。
app	为 Pod 工作负载名称。

参数	描述
mountPath	为容器内挂载路径，此处示例中为“/tmp”。

说明

“spec.template.spec.containers.volumeMounts.name” 和 “spec.template.spec.volumes.name”有映射关系，必须保持一致。

步骤 2 执行如下命令创建 Pod。

kubectl create -f efs-pod-example.yaml

创建完成后，在 CCE 界面“存储管理 > 极速文件存储卷”中单击 PVC 名称，在 PVC 详情页面可查看极速文件存储服务和 PVC 的绑定关系。

----结束

9.6 快照与备份

CCE 通过云硬盘 EVS 服务为您提供快照功能，云硬盘快照简称快照，指云硬盘数据在某个时刻的完整拷贝或镜像，是一种重要的数据容灾手段，当数据丢失时，可通过快照将数据完整的恢复到快照时间点。

您可以创建快照，从而快速保存指定时刻云硬盘的数据。同时，您还可以通过快照创建新的云硬盘，这样云硬盘在初始状态就具有快照中的数据。

使用须知

- 快照功能**仅支持 v1.15 及以上版本**的集群，且需要安装基于 CSI 的 Everest 插件才可以使用。
- 基于快照创建的云硬盘，其子类型（普通 IO/高 IO/超高 IO）、是否加密、磁盘模式（VBD/SCSI）、共享性(非共享/共享)、容量等都要与快照关联母盘保持一致，这些属性查询和设置出来后不能够修改。

使用场景

快照功能可以帮助您实现以下需求：

- **日常备份数据**

  通过对云硬盘定期创建快照，实现数据的日常备份，可以应对由于误操作、病毒以及黑客攻击等导致数据丢失或不一致的情况。

- **快速恢复数据**

  更换操作系统、应用软件升级或业务数据迁移等重大操作前，您可以创建一份或多份快照，一旦升级或迁移过程中出现问题，可以通过快照及时将业务恢复到快照创建点的数据状态。

  例如，当由于云服务器 A 的系统盘 A 发生故障而无法正常开机时，由于系统盘 A 已经故障，因此也无法将快照数据回滚至系统盘 A。此时您可以使用系统盘 A

已有的快照新创建一块云硬盘 B 并挂载至正常运行的云服务器 B 上，从而云服务器 B 能够通过云硬盘 B 读取原系统盘 A 的数据。

说明

当前 CCE 提供的快照能力与 K8S 社区 CSI 快照功能一致：只支持基于快照创建新云硬盘，不支持将快照回滚到源云硬盘。

- **快速部署多个业务**

  通过同一个快照可以快速创建出多个具有相同数据的云硬盘，从而可以同时为多种业务提供数据资源。例如数据挖掘、报表查询和开发测试等业务。这种方式既保护了原始数据，又能通过快照创建的新云硬盘快速部署其他业务，满足企业对业务数据的多元化需求。

操作概览

1. 您可以创建快照，从而快速保存指定时刻云硬盘的数据。
2. 您可以从快照创建云硬盘，这样云硬盘在初始状态就具有快照中的数据，当数据丢失时，可通过快照将云硬盘数据完整的恢复到快照时间点。
3. 当快照不再使用时，可以删除快照以释放虚拟资源。

创建快照

通过云硬盘可以创建快照，从而保存指定时刻的云硬盘数据。

说明

只有可用或正在使用状态的云硬盘才能创建快照，单个云硬盘最多支持创建 7 个快照。云硬盘创建请参见 9.3 云硬盘存储卷。

加密磁盘的快照数据以加密方式存放，非加密磁盘的快照数据以非加密方式存放。

操作步骤：

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 存储管理”。

步骤 2 在“快照与备份”页签，单击“创建快照”。

步骤 3 在打开的“创建快照”页面中，配置如表 9-45 基本信息。

表9-45 创建快照基本信息配置

参数	参数说明
快照名称	新建快照的名称，默认以“cce-disksnap-”开头，样例：cce-disksnap-01
集群名称	快照部署所在的集群。
命名空间	选择对应的命名空间。
选择存储	选择创建快照的云硬盘，即已创建的云硬盘存储卷。 只有可用或正在使用状态的磁盘才能创建快照。磁盘创建请参见 9.3 云硬盘存储卷。

步骤 4 单击“立即创建”。

步骤 5 单击“返回快照管理”，当快照状态为“正常”时，表示创建成功。

单击快照名称前的^，可查看快照的状态和详细信息。

----结束

从快照创建云硬盘

您可以在快照列表页面，选择快照用来创建云硬盘数据卷。

通过快照创建云硬盘数据卷时，有以下约束：

- 通过快照创建云硬盘数据卷时，磁盘类型、磁盘模式、加密属性需和快照源云硬盘保持一致。
- 一个快照最多支持创建 128 个云硬盘数据卷。
- 从快照创建云硬盘数据卷时，不支持批量创建，数量只能为“1”。

操作步骤：

步骤 1 登录，在左侧导航栏中选择“资源管理 > 存储管理”。

步骤 2 单击“快照与备份”页签，在快照列表中，找到指定快照并单击快照所在行的“操作”列下的“创建数据卷”。

步骤 3 设置云硬盘数据卷的各项参数。具体请参见 9.3 云硬盘存储卷。

通过快照创建云硬盘数据卷时，容量大小不能低于快照大小。当您未指定云硬盘数据卷的容量时，当快照大小容量低于 10GB，默认容量为 10GB，当快照大小高于 10GB，默认容量和快照大小保持一致。

步骤 4 单击“立即购买”。

步骤 5 根据界面提示付款，单击“提交”。

步骤 6 在“任务提交成功”页面，单击“返回存储卷列表”，在“云硬盘存储卷”页签下，查看云硬盘存储卷的状态。

待云硬盘存储卷状态变为“正常”时，表示创建成功。

----结束

删除快照

当快照不再使用时，您可以删除快照以释放虚拟资源，删除快照有以下约束：

- 当快照状态为“正常”或者“错误”时，才可以删除快照。
- 如果将创建快照的云硬盘存储卷删除，那么对应的快照也会被同时删除。

操作步骤：

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 存储管理”。

步骤 2 单击“快照与备份”页签，在快照列表中，找到指定快照并单击快照所在行的“操作”列下的“删除”。

步骤 3 (可选) 如果需要删除多个快照，可勾选 选中多个快照，单击快照列表左上方的“删除”按钮。

步骤 4 在弹出对话框中，确认删除信息并输入 **DELETE** 后，单击“是”进行删除。

----结束

参考文档

- Volume Snapshots

10 命名空间

10.1 创建命名空间

操作场景

命名空间（Namespace）是对一组资源和对象的抽象整合。在同一个集群内可创建不同的命名空间，不同命名空间中的数据彼此隔离。使得它们既可以共享同一个集群的服务，也能够互不干扰。

例如可以将开发环境、测试环境的业务分别放在不同的命名空间。

前提条件

至少已创建一个集群，请参见 3.3 购买混合集群。

约束与限制

每个命名空间下，创建的服务数量不能超过 6000 个。此处的服务对应 kubernetes 的 service 资源，即工作负载所添加的服务。

命名空间类别

命名空间按创建类型分为两大类：集群默认创建的、用户创建的。

- 集群默认创建的：集群在启动时会默认创建 default、kube-public、kube-system、kube-node-lease 命名空间。
  - default：所有未指定 Namespace 的对象都会被分配在 default 命名空间。
  - kube-public：此命名空间下的资源可以被所有人访问（包括未认证用户），用来部署公共插件、容器模板等。
  - kube-system：所有由 Kubernetes 系统创建的资源都处于这个命名空间。
  - kube-node-lease：每个节点在该命名空间中都有一个关联的“Lease”对象，该对象由节点定期更新。NodeStatus 和 NodeLease 都被视为来自节点的心跳，在 v1.13 之前的版本中，节点的心跳只有 NodeStatus，NodeLease 特性从 v1.13 开始引入。NodeLease 比 NodeStatus 更轻量级，该特性在集群规模扩展性和性能上有明显提升。

- 用户创建的：用户可以按照需要创建命名空间，例如开发环境、联调环境和测试环境分别创建对应的命名空间。或者按照不同的业务创建对应的命名空间，例如系统若分为登录和游戏服务，可以分别创建对应命名空间。

创建命名空间

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 命名空间”。单击“创建命名空间”。

步骤 2 参照表 10-1 设置命名空间参数，其中带“*”标志的参数为必填参数。

表10-1 命名空间基本信息

参数	参数说明
* 命名空间	新建命名空间的名称，命名必须唯一。
* 集群	新建命名空间属于哪个集群。
节点亲和	开启后，当前命名空间下所创建的工作负载只能调度到拥有特定标签的节点上。您可以在节点管理中通过标签管理为节点添加标签。 该参数仅在 v1.13.10-r0 及以上版本的集群中显示。
命名空间描述	输入对命名空间的描述信息。
资源配额设置	资源配额可以限制命名空间下的资源使用，进而支持以命名空间为粒度的资源划分。 须知 **建议根据需要在命名空间中设置资源配额，避免因资源过载导致集群或节点异常。** 例如：在集群中每个节点可以创建的实例（Pod）数默认为 110 个，如果您创建的是 50 节点规格的集群，则最多可以创建 5500 个实例。因此，您可以在命名空间中自行设置资源配额以确保所有命名空间内的实例总数不超过 5500 个，以避免资源过载。 可设置配额的资源类型如下： • CPU（Core） • 内存（MiB） • 有状态工作负载（StatefulSet） • 无状态工作负载（Deployment） • 普通任务（Job） • 定时任务（CronJob） • 实例（Pod） • 服务（Service） 请输入整型数值，"0"表示不限制该资源的使用。 若您需要限制 CPU 或内存的配额，则创建工作负载时必须

参数	参数说明
	指定 CPU 或内存请求值。

步骤 3 配置完成后，单击“确定”。

----结束

10.2 管理命名空间

使用命名空间

- 创建工作负载时，您可以选择对应的命名空间，实现资源或租户的隔离。
- 查询工作负载时，选择对应的命名空间，查看对应命名空间下的所有工作负载。

命名空间使用实践

- **按照不同环境划分命名空间**

  一般情况下，工作负载发布会经历开发环境、联调环境、测试环境，最后到生产环境的过程。这个过程中不同环境部署的工作负载相同，只是在逻辑上进行了定义。分为两种做法：

  - 分别创建不同集群。

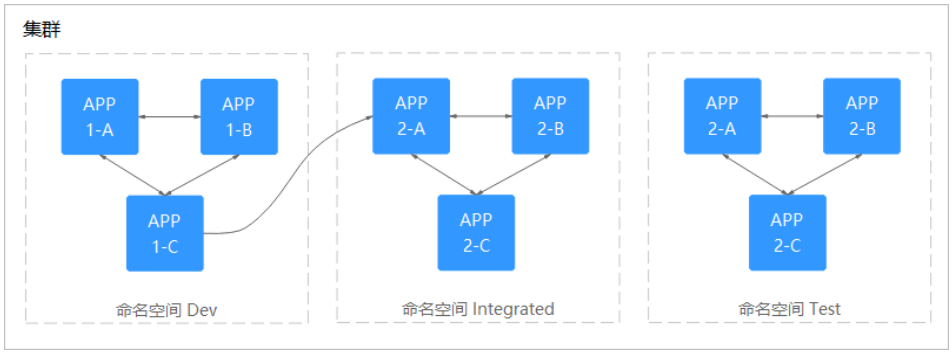
    不同集群之间，资源不能共享。同时，不同环境中的服务互访需要通过负载均衡才能实现。

  - 不同环境创建对应命名空间。

    同个命名空间下，通过服务名称（Service name）可直接访问。跨命名空间的可以通过服务名称、命名空间名称访问。

    例如下图，开发环境/联调环境/测试环境分别创建了命名空间。

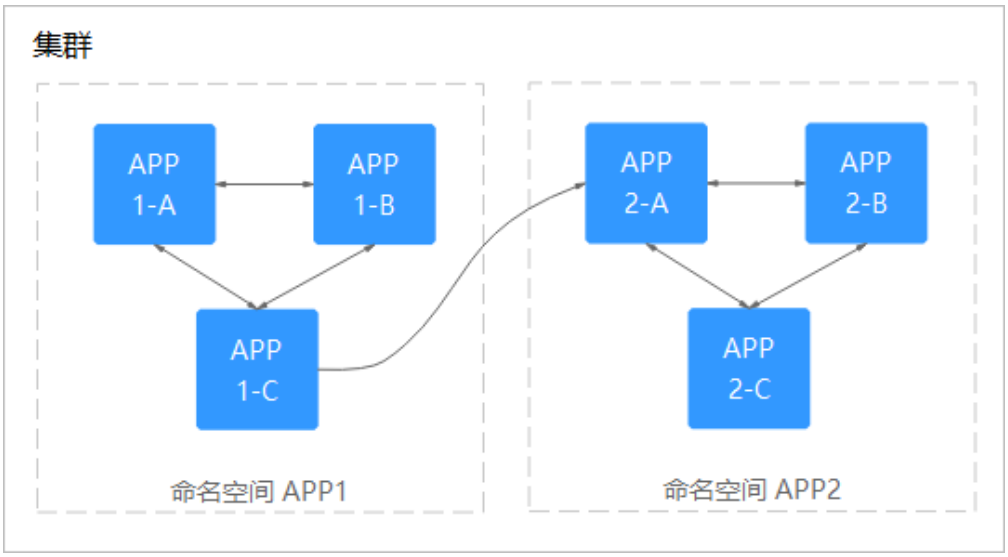
    图10-1 不同环境创建对应命名空间

- **按照应用划分命名空间**

对于同个环境中，应用数量较多的情况，建议进一步按照工作负载类型划分命名空间。例如下图中，按照 APP1 和 APP2 划分不同命名空间，将不同工作负载在逻辑上当做一个工作负载组进行管理。且同一个命名空间内的工作负载可以通过服务名称访问，不同命名空间下的通过服务名称、命名空间名称访问。

图10-2 按照工作负载划分命名空间



删除命名空间

删除命名空间会删除该命名空间下所有的资源（如工作负载，短任务、配置项等），请谨慎操作。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“ 资源管理 > 命名空间”。

步骤 2 在“集群”下拉框中，选择命名空间所在的集群。

步骤 3 选中待删除的命名空间，单击“删除”。

根据系统提示进行删除操作。系统内置的命名空间不支持删除。

----结束

10.3 设置命名空间级的网络策略

您可以通过网络隔离开关，设置命名空间层面的网络策略。

例如命名空间 default，网络隔离的默认状态为“隔离状态未开启”，表示“当前集群下的所有工作负载”都可以访问“命名空间 default 下的工作负载”。

若您需要设置其它工作负载不可以访问“命名空间 default 下的工作负载”，请参照以下步骤设置：

须知

网络模型为“VPC 网络”时不支持命名空间（namespace）的网络隔离，仅在“容器隧道网络”模式下支持。

前提条件

- 您已成功创建一个 Kubernetes 集群，参见 3.3 购买混合集群。
- 您已成功创建一个命名空间，参见 10.1 创建命名空间。

操作步骤

步骤 1 登录为 CCE 控制台，在左侧导航栏中选择“资源管理 > 命名空间”。

步骤 2 在“集群”下拉框中，选择命名空间所在的集群。

步骤 3 在待设置命名空间（例如 default）后，将网络隔离一栏的“隔离状态未开启”改为“隔离状态已开启”，开启后，当前集群内的其他工作负载都不能访问此命名空间下的工作负载。

设置完成后，当前集群内其它命名空间下的工作负载都不能访问 default 下的工作负载。

----结束

10.4 设置资源配额及限制

通过设置命名空间级别的资源配额，实现多团队或多用户在共享集群资源的情况下限制团队、用户可以使用的资源总量，包括限制命名空间下创建某一类型对象的数量以及对象消耗计算资源（CPU、内存）的总量。

前提条件

- 您已成功创建一个 Kubernetes 集群，参见 3.3 购买混合集群。
- 您已成功创建一个命名空间，参见 10.1 创建命名空间。

背景信息

默认情况下，运行中的 Pod 可以无限制的使用 Node 节点上的 CPU 和内存，这意味着任意一个 Pod 都可以无节制地使用集群的计算资源，某个命名空间的 Pod 可能会耗尽集群的所有资源。

kubernetes 在一个物理集群上提供了多个虚拟集群，这些虚拟集群被称为命名空间。命名空间可用于多种工作用途，满足多用户的使用需求，通过为每个命名空间配置资源额度可以有效限制资源滥用，从而保证集群的可靠性。

您可为命名空间配置包括 CPU、内存、Pod 数量等资源的额度，更多信息请参见 Resource Quotas。

其中，不同的集群规模对应的 Pod 数量推荐值如下：

维度	集群规模	Pod 数量推荐值
规格	50 节点	2500 Pod 实例
	200 节点	1W Pod 实例
	1000 节点	3W Pod 实例
	2000 节点	5W Pod 实例

操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 命名空间”。

步骤 2 在“集群”下拉框中，选择命名空间所在的集群。

步骤 3 单击对应命名空间后的“配额管理”，单击“编辑配额”。

系统级别的命名空间 kube-system、kube-public 默认不支持设置资源配额。

步骤 4 设置资源配额，单击“确定”。

- CPU（Core）：限制命名空间下工作负载实例（Pod）能申请 CPU 资源的最大值。单位为“核”。
- 内存（MiB）：限制命名空间下工作负载实例能申请内存资源的最大值。单位为 MiB。
- 有状态工作负载（StatefulSet）：限制命名空间下能创建有状态负载的最大数量。
- 无状态工作负载（Deployment）：限制命名空间下能创建无状态负载的最大数量。
- 普通任务（Job）：限制命名空间下能创建普通任务的最大数量。
- 定时任务（CronJob）：限制命名空间下能创建定时任务的最大数量。
- 实例（Pod）：限制命名空间下能创建实例的最大数量。
- 服务（Service）：限制命名空间下能创建服务的最大数量。

须知

- 命名空间设置了 CPU 或内存资源配额后，创建工作负载时，必须指定 CPU 或内存的请求值（request）和约束值（limit），否则 CCE 将拒绝创建实例。若设置资源配额值为 0，则不限制该资源的使用。
- 配额累计使用量包含 CCE 系统默认创建的资源，如 default 命名空间下系统默认创建的 kubernetes 服务（该服务可通过后端 kubectl 工具查看）等，故建议命名空间下的资源配额略大于实际期望值以去除系统默认创建资源的影响。

----结束

11 配置中心

11.1 创建配置项

操作场景

配置项（ConfigMap）是一种用于存储工作负载所需配置信息的资源类型，内容由用户决定。配置项创建完成后，可在容器工作负载中作为文件或者环境变量使用。

配置项允许您将配置文件从容器镜像中解耦，从而增强容器工作负载的可移植性。

配置项价值如下：

- 使用配置项功能可以帮您管理不同环境、不同业务的配置。
- 方便您部署相同工作负载的不同环境，配置文件支持多版本，方便您进行更新和回滚工作负载。
- 方便您快速将您的配置以文件的形式导入到容器中。

前提条件

已创建集群和节点资源，具体操作请参见 3.3 购买混合集群。若已有集群和节点资源，无需重复操作。

操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“配置中心 > 配置项（ConfigMap）”，单击“创建配置项”。

步骤 2 参照表 11-1 设置新增配置参数。

表11-1 新建配置参数说明

参数	参数说明
配置名称	新建的配置名称，同一个命名空间里命名必须唯一。
所属集群	使用新建配置的集群。

参数	参数说明
集群命名空间	新建配置所在的命名空间。若不选择，默认配置为 default。
描述	配置项的描述信息。
配置数据	工作负载配置的数据可以在容器中使用，或被用来存储配置数据。其中，“键”代表文件名；“值”代表文件中的内容。 1. 单击“添加更多配置数据” 。 2. 输入键、值。
配置标签	标签以 Key/value 键值对的形式附加到各种对象上（如工作负载、节点、服务等）。 标签定义了这些对象的可识别属性，用来对它们进行管理和选择。 1. 单击“添加标签” 。 2. 输入键、值。

步骤 3 配置完成后，单击“创建”。

工作负载配置列表中会出现新创建的工作负载配置。

----结束

使用 kubectl 创建配置项

步骤 1 请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令。

步骤 2 创建并编辑 cce-configmap.yaml 文件。

vi cce-configmap.yaml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cce-configmap
data:
  SPECIAL_LEVEL: Hello
  SPECIAL_TYPE: CCE
```

步骤 3 创建配置项。

kubectl create -f cce-configmap.yaml

kubectl get cm

```
NAME             DATA          AGE
cce-configmap      3             3h
cce-configmap1     3             7m
```

----结束

相关操作

配置项创建完成后，您还可以执行表 11-2 中的操作。

表11-2 其他操作

操作	说明
查看 YAML	单击配置项名称后的“查看 YAML”，可查看到当前配置项的 YAML 文件。
更新配置	1. 选择需要更新的配置项名称，单击“更新”。 2. 根据表 11-1 更改信息。 3. 单击“更新”。
删除配置	选择要删除的配置项，单击“删除”。 根据系统提示删除配置。

11.2 使用配置项

配置项创建后，可在工作负载环境变量、命令行参数和数据卷三个场景使用。

- 通过配置项设置工作负载环境变量
- 通过配置项设置命令行参数
- 使用配置项挂载到工作负载数据卷

本节以下面这个 ConfigMap 为例，具体介绍 ConfigMap 的用法。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cce-configmap
data:
  SPECIAL_LEVEL: Hello
  SPECIAL_TYPE: CCE
```

须知

在 Pod 里使用 ConfigMap 时，需要 Pod 和 ConfigMap 处于同一集群和命名空间中。

通过配置项设置工作负载环境变量

您可以在创建工作负载时将配置项设置为环境变量，使用 valueFrom 参数引用 ConfigMap 中的 Key/Value。

```
apiVersion: v1
kind: Pod
```

```
metadata:
  name: configmap-pod-1
spec:
  containers:
    - name: test-container
      image: busybox
      command: [ "/bin/sh", "-c", "env" ]
      env:
        - name: SPECIAL_LEVEL_KEY
          valueFrom:                              ##使用 valueFrom 来指定 env 引用配置项的 value 值
            configMapKeyRef:
              name: cce-configmap                 ##引用的配置文件名称
              key: SPECIAL_LEVEL                  ##引用的配置项 key
  restartPolicy: Never
```

如果您需要将多个配置项的 value 值定义为 pod 的环境变量值，您只需要在 pod 中添加多个环境变量参数即可。

```
env:
- name: SPECIAL_LEVEL_KEY
  valueFrom:
    configMapKeyRef:
         name: cce-configmap
         key: SPECIAL_LEVEL
- name: SPECIAL_TYPE_KEY
  valueFrom:
    configMapKeyRef:
         name: cce-configmap
         key: SPECIAL_TYPE
```

如果要将一个配置项中所有数据都添加到环境变量中，可以使用 envFrom 参数，配置项中的 key 会成为 Pod 中的环境变量名称。

```
apiVersion: v1
kind: Pod
metadata:
  name: configmap-pod-2
spec:
  containers:
    - name: test-container
      image: busybox
      command: [ "/bin/sh", "-c", "env" ]
      envFrom:
      - configMapRef:
          name: cce-configmap
  restartPolicy: Never
```

通过配置项设置命令行参数

您可以使用配置项设置容器中的命令或者参数值，使用环境变量替换语法 $(VAR_NAME)来进行。如下面的编排示例所示。

```
apiVersion: v1
kind: Pod
metadata:
  name: configmap-pod-3
```

```
spec:
  containers:
    - name: test-container
      image: busybox
      command: [ "/bin/sh", "-c", "echo $(SPECIAL_LEVEL_KEY) $(SPECIAL_TYPE_KEY)" ]
      env:
        - name: SPECIAL_LEVEL_KEY
          valueFrom:
            configMapKeyRef:
              name: cce-configmap
              key: SPECIAL_LEVEL
        - name: SPECIAL_TYPE_KEY
          valueFrom:
            configMapKeyRef:
              name: cce-configmap
              key: SPECIAL_TYPE
  restartPolicy: Never
```

这个 Pod 运行后，输出如下内容。

```
Hello CCE
```

使用配置项挂载到工作负载数据卷

配置项也可以在数据卷里面使用，只需在创建工作负载时将配置项挂载到工作负载中即可。挂载完成后，最终生成以 key 为文件名， value 为文件内容的配置文件。

```
apiVersion: v1
kind: Pod
metadata:
  name: configmap-pod-4
spec:
  containers:
    - name: test-container
      image: busybox
      command: [ "/bin/sh", "-c", "ls /etc/config/" ]   ##列出该目录下的文件名称
      volumeMounts:
      - name: config-volume
        mountPath: /etc/config                          ##挂载到/etc/config 目录下
  volumes:
    - name: config-volume
      configMap:
        name: cce-configmap
  restartPolicy: Never
```

这个 Pod 运行后，在/etc/config 目录下会生成 SPECIAL_LEVEL 和 SPECIAL_TYPE 两个文件，文件的内容为 Hello 和 CCE。且运行时会输出如下内容，即 cce-configmap 中 SPECIAL_LEVEL 和 SPECIAL_TYPE 的名称。

```
SPECIAL_TYPE
SPECIAL_LEVEL
```

挂载 ConfigMap 到数据卷还可以在界面上进行操作，在创建工作负载时，设置容器的高级设置，选择数据存储，添加本地磁盘，选择 ConfigMap 即可配置。具体操作请参见配置项(ConfigMap)挂载。

11.3 创建密钥

操作场景

密钥（Secret）是一种用于存储工作负载所需要认证信息、密钥的敏感信息等的资源类型，内容由用户决定。资源创建完成后，可在容器工作负载中作为文件或者环境变量使用。

前提条件

已创建集群和节点资源，具体操作请参见 3.3 购买混合集群。若已有集群和节点资源，无需重复操作。

操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“配置中心 > 密钥（Secret）”，单击“添加密钥”。

步骤 2 您可以直接创建密钥或基于 YAML 来创建。若希望通过 YAML 创建，请跳转至步骤 4。

步骤 3 方式一：直接创建密钥。

参照表 11-3 设置基本信息。

表11-3 基本信息说明

参数	参数说明
密钥名称	新建的密钥的名称，同一个命名空间内命名必须唯一。
所属集群	使用新建密钥的集群。
集群命名空间	新建密钥所在的命名空间，默认为 default。
描述	密钥的描述信息。
密钥类型	新建的密钥类型。 • Opaque：一般密钥类型。 • kubernetes.io/dockerconfigjson：存放拉取私有仓库镜像所需的认证信息。 • IngressTLS：存放 7 层负载均衡服务所需的证书。 • 其他：若需要创建其他类型的密钥，请手动输入密钥类型。
密钥数据	工作负载密钥的数据可以在容器中使用。 • 当密钥为 Opaque 类型时。 1. 单击“添加更多密钥数据” 。 2. 输入键、值。其中“值”必须使用 Base64 编码，Base64 编码方法请参见如何进行 Base64 编码。

参数	参数说明
	• 当密钥为 kubernetes.io/dockerconfigjson 类型时，输入私有镜像仓库的帐号和密码。 • 当密钥为 IngressTLS 类型时，上传证书文件和私钥文件。 说明 • 证书是自签名或 CA 签名过的凭据，用来进行身份认证。 • 证书请求是对签名的请求，需要使用私钥进行签名。
密钥标签	标签以 Key/value 键值对的形式附加到各种对象上（如工作负载、节点、服务等）。 标签定义了这些对象的可识别属性，用来对它们进行管理和选择。 1. 单击“添加标签” 。 2. 输入键、值。

步骤 4 方式二：基于 YAML 文件创建密钥。

说明

若需要通过上传文件的方式创建资源，请确保资源描述文件已创建。CCE 支持 json 或 yaml 格式，详细请参见 Secret 资源文件配置说明。

您可以导入或直接编写文件内容，格式为 YAML 或 JSON。

- 方式一：导入编排文件。

  单击“添加文件”，导入格式为 YAML 或 JSON 的文件。编排内容中可直接展示编排内容。
- 方式二：直接编排内容。

  在编排内容区域框中，输入 YAML 或 JSON 文件内容。

步骤 5 配置完成后，单击“创建”。

密钥列表中会出现新创建的密钥。

----结束

Secret 资源文件配置说明

本章节主要介绍 Secret 类型的资源描述文件的配置示例。

例如现在有一个工作负载需要获取帐号密码，可以通过 Secret 来实现：

- yaml 文件格式

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

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret          # secret 的名称
  namespace: default       #命名空间，默认为 default
data:
  username: my-username  #用户名
  password: ******  #需要用 Base64 编码
type: Opaque     # type 建议不要做修改
```

- json 文件格式

  定义的 Secret 文件 secret.json 内容如下。

```
{
  "apiVersion": "v1",
  "kind": "Secret",
  "metadata": {
    "name": "mysecret",
    "namespace": "default"
  },
  "data": {
    "username": "my-username",
    "password": "******"
  },
  "type": "Opaque"
}
```

使用 kubectl 创建密钥

步骤 1 请参见 3.4.2 通过 kubectl 操作 CCE 集群配置 kubectl 命令。

步骤 2 通过 Base64 编码，创建并编辑 cce-secrets.yaml 文件。

```
# echo -n "待编码内容" | base64
******
```

vi cce-secret.yaml

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
data:
  username: admin
  password: ******
```

步骤 3 创建密钥。

kubectl create -f cce-secret.yaml

创建完成后可以查询到密钥。

kubectl get secret

----结束

相关操作

密钥创建完成后，您还可以执行表 11-4 中的操作。

说明

密钥列表中包含系统密钥资源，系统密钥资源不可更新，也不能删除，只能查看。

表11-4 其他操作

操作	说明
查看 YAML	单击密钥名称后的“查看 YAML”，可查看到当前密钥的 YAML 文件。
更新密钥	1. 选择需要更新的密钥名称，单击“更新”。 2. 根据表 11-3 更改信息。 3. 单击“更新”。
删除密钥	选择要删除的密钥，单击“删除”。 根据系统提示删除密钥。
批量删除密钥	1. 勾选需要删除的密钥名称。 2. 单击页面左上角的“删除”，删除选中的密钥。 3. 根据系统提示删除密钥。

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 "待编码内容" | base64
******
```

11.4 使用密钥

密钥创建后，可在工作负载环境变量和数据卷两个场景使用。

须知

如下密钥为 CCE 系统使用的，请勿对其做任何操作。

- 不要操作 kube-system 下的 secrets。
- 不要操作任何命名空间下的 default-secret、paas.elb。其中，default-secret 用于 SWR 的私有镜像拉取，paas.elb 用于该命名空间下的服务对接 ELB。

- 使用密钥配置 Pod 的数据卷

- 使用密钥设置 Pod 的环境变量

本节以下面这个所 Secret 为例，具体介绍 Secret 的用法。

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
data:
  username: my-username #用户名
  password: ******  #需要用 Base64 编码
```

须知

在 Pod 里使用密钥时，需要 Pod 和密钥处于同一集群和命名空间中。

使用密钥配置 Pod 的数据卷

密钥可以在 Pod 中作为文件使用。如下面的 Pod 示例所示，mysecret 密钥的 username 和 password 以文件方式保存在/etc/foo 目录下。

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
  - name: mypod
    image: redis
    volumeMounts:
    - name: foo
      mountPath: "/etc/foo"
      readOnly: true
  volumes:
  - name: foo
    secret:
      secretName: mysecret
```

另外，还可以指定密钥的目录路径和权限，username 存放在容器中的/etc/foo/my-group/my-username 目录下。

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
  - name: mypod
    image: redis
    volumeMounts:
    - name: foo
      mountPath: "/etc/foo"
```

```
  volumes:
  - name: foo
    secret:
      secretName: mysecret
      items:
      - key: username
        path: my-group/my-username
        mode: 511
```

挂载 Secret 到数据卷还可以在界面上进行操作，在创建工作负载时，设置容器的高级设置，选择数据存储，添加本地磁盘，选择 Secret 即可配置。具体请参见密钥(Secret)挂载。

使用密钥设置 Pod 的环境变量

密钥可以在 Pod 中设置为环境变量。如下面的 Pod 示例所示，mysecret 密钥的 username 和 password 配置为 Pod 的环境变量。

```
apiVersion: v1
kind: Pod
metadata:
  name: secret-env-pod
spec:
  containers:
  - name: mycontainer
    image: redis
    env:
      - name: SECRET_USERNAME
        valueFrom:
          secretKeyRef:
            name: mysecret
            key: username
      - name: SECRET_PASSWORD
        valueFrom:
          secretKeyRef:
            name: mysecret
            key: password
  restartPolicy: Never
```

12 模板市场

12.1 我的模板

12.1.1 模板概述

模板市场是云容器引擎基于 Kubernetes Helm 标准的模板提供统一的资源管理与调度，高效地实现了模板的快速部署与后期管理，大幅简化了 Kubernetes 资源的安装管理过程。

说明

Helm 是管理 Kubernetes 应用程序的打包工具。更多详情请查看 Helm 官网文档。

我的模板是通过自定义 Helm 模板来简化工作负载部署的服务。

本章节指导您如何通过自定义模板创建工作负载，通过 CCE 控制台，您可通过多种方式创建编排模板。

约束与限制

- 单个用户可以上传模板的个数有限制，请以各个 Region 控制台界面中提示的实际值为准。
- 模板若存在多个版本，则消耗对应数量的模板配额。
- 由于模板的操作权限同时具有较高的集群操作权限，因此租户应当谨慎授予用户对于模板生命周期管理的权限，包括上传模板的权限，以及创建、删除和更新模板实例的权限。

12.1.2 准备模板包

您可以通过如下两种方法准备模板包：

- 自定义模板包
- 使用 Kubernetes 官方模板包

须知

若创建的工作负载需要使用云硬盘（EVS）功能和弹性负载均衡（ELB）功能，则需要对模板包进行修改，具体使用方法请参见 12.1.5 使用云硬盘和 12.1.6 使用弹性负载均衡。

自定义模板包

步骤 1 根据需求自定义设置模板包内容。

制作模板包的方法，请参见 https://github.com/helm/helm/blob/master/cmd/helm/chart.go。

步骤 2 按照模板包规范的要求设置模板包目录结构和命名模板包。

----结束

使用 Kubernetes 官方模板包

步骤 1 访问 https://github.com/helm/charts，获取需要的社区模板包。

步骤 2 登录 Linux 机器。

步骤 3 上传步骤 1 中获取到的模板包。

步骤 4 执行如下命令，压缩模板包。

- 若 Linux 机器没有安装 Helm 客户端，则执行如下命令。

  tar pzcf {name}-{version}.tgz {name}/

  其中，

  {name}替换为实际的模板包名。

  {version}实际的模板包版本号。

须知

{name}和{version}必须与模板包中 Chart.yaml 中所写的 name 和 version 相同。

- 若 Linux 机器已安装 Helm 客户端，则执行如下命令。

  helm package {name}/

  其中，将{name}替换为实际的模板包名。

步骤 5 按照模板包规范的要求设置模板包目录结构和命名模板包。

----结束

模板包规范

以下以 redis 为例，在准备 redis 模板包时根据模板包规范制作模板包。

- **命名要求**

  模板包命名格式为：工作负载名称-主版本号.次版本号.修订号.tgz，如 redis-0.4.2.tgz。

  说明

  模板包的版本号需遵循语义化版本规则。

  - 主版本号、次版本号为必选，修订号为可选。
  - 版本号的长度不能超过 64 个字符。
  - 主版本号、次版本号的数值为整数，均需要≥0，且≤99。
  - 修订号由数字、大小写英文字母即连接符组成，即[0-9A-Za-z-]。

- **目录结构**

  模板包的目录结构如下所示：

```
redis/
  templates/
  values.yaml
  README.md
  Chart.yaml
  .helmignore
```

  目录说明如表 12-1 所示，带*的为必选项：

表12-1 模板包目录说明

参数	参数说明
* templates	用于存放所有的 template（模板）文件。
* values.yaml	用于描述 template 文件所需的配置参数。 须知 定义 template 文件配置参数时，请注意此处定义的“镜像地址”务必和容器镜像仓库中对应的镜像地址保持一致。否则创建工作负载会异常，提示镜像拉取失败。 镜像地址获取方法如下：在 CCE 控制台，单击左侧导航栏的“镜像仓库”，进入容器镜像服务控制台。在“我的镜像 > 自有镜像”中，单击已上传镜像的名称，在“镜像版本 ”页签的“下载指令”栏中即可获取镜像地址，单击 按钮即可复制该指令。
README.md	一个 markdown 文件，包括： • 描述 Chart 提供的工作负载或服务。 • 运行 Chart 的前提。 • 解释 values.yaml 文件中的配置。 • 安装和配置 Chart 的相关信息。
* Chart.yaml	模板的基本信息说明。
.helmignore	设定在工作负载安装时不需要读取 templates 的某些文件或数据。

12.1.3 上传模板包

操作场景

上传模板到“模板市场 > 我的模板”中，为后期创建工作负载准备。

操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“模板市场 > 我的模板”，单击“上传模板”。

步骤 2 单击“添加文件”，选中待上传的工作负载包后，单击“上传”。

说明

由于上传模板时创建 OBS 桶的命名规则由 cce-charts-{region}-**{domain_name}**变为 cce-charts-{region}-**{domain_id}**，其中旧命名规则中的 domain_name 系统会做 base64 转化并取前 63 位，如果您在现有命名规则的 OBS 桶中找不到模板，请在旧命名规则的桶中进行查找。

----结束

相关操作

模板创建完成后，在“我的模板”页面您还可以执行表 12-2 中的操作。

表12-2 相关操作

操作	说明
安装模板	单击“安装”，安装该模板用于创建工作负载，具体请参见 12.1.4 通过模板创建工作负载。
更新模板	更新模板是针对同名同版本的模板，即只更新内容不更新版本，操作同“上传模板”。
下载模板	单击“更多 > 下载”，将模板下载到本地。
删除模板	单击“更多 > 删除”，删除已创建的模板。 模板删除后不能恢复，请谨慎操作。

12.1.4 通过模板创建工作负载

创建模板实例

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“模板市场 > 我的模板”。

步骤 2 在已上传的模板中，单击“安装”。

步骤 3 参照表 12-3 设置安装工作负载参数，其中带“*”标志的参数为必填参数。

表12-3 安装工作负载参数说明

参数	参数说明
* 模板实例名称	新建模板实例名称，命名必须唯一。
* 模板版本	默认为模板的版本。
* 部署集群	选择部署工作负载的集群。
* 命名空间	指定部署的命名空间。
高级配置	用户可以导入 values.yaml 文件，导入后可替换模板包中的 values.yaml 文件；也可直接在配置框中在线编辑模板参数。 **说明** **此处导入的 values.yaml 文件需符合 yaml 规范，即 KEY:VALUE 格式。对于文件中的字段不做任何限制。** **导入的 value.yaml 的 key 值必须与所选的模板包的 values.yaml 保持一致，否则不会生效。即 key 不能修改。** 1. 单击“导入配置文件”。 2. 选择对应的 values.yaml 文件，单击“打开”。

步骤 4 配置完成后，单击“自定义安装”。

步骤 5 确认订单后，单击“提交”。

步骤 6 单击“返回模板实例列表”查看模板工作负载的运行状态，或单击“查看模板实例详情”查看模板工作负载的详情。

----结束

升级模板工作负载

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“模板市场 > 我的模板 > 模板实例”。

步骤 2 单击待升级工作负载后的“升级”，设置升级模板工作负载的参数。

步骤 3 选择对应的模板版本。

步骤 4 参照界面提示修改模板参数。单击“升级”，再单击“提交”。

步骤 5 单击“返回模板实例列表”，模板状态为“升级成功”时，表明工作负载升级成功。

----结束

回退模板工作负载

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“模板市场 > 我的模板 > 模板实例”。

步骤 2 单击待回退工作负载后的“更多 > 回退”，选择要回退的工作负载版本，单击“回退”。

模板工作负载列表中，状态为“回退成功”时，表明工作负载回退成功。

----结束

卸载模板工作负载

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“模板市场 > 我的模板 > 模板实例”。

步骤 2 单击待卸载模板实例后的“更多 > 卸载”，确认待卸载模板实例后，单击“是”。模板实例卸载后不能恢复，请谨慎操作。

----结束

12.1.5 使用云硬盘

CCE 支持使用自有插件对接云硬盘，以支持用户可持久化存储的需求。

以下样例展示了如何在模板中定义使用自有云硬盘，创建该模板工作负载时容器会动态创建一块 10Gi 大小的云硬盘并将其挂载到容器中。

须知

CCE 当前只支持动态创建云硬盘。

```
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: helm-test-slave
spec:
  updateStrategy:
    type: "RollingUpdate"
  serviceName: helm-test-slave-headless
  replicas: 1
  template:
    metadata:
      labels:
        app: helm-test-slave
        type: slave
        release: "helm-test"
        failure-domain.beta.kubernetes.io/region: region01
        failure-domain.beta.kubernetes.io/zone: AZ01
    spec:
      containers:
        - name: helm-test-slave
          image: nginx:alpine-per1
          volumeMounts:
          - mountPath: /redis-data
            name: helm-test-slave
          - mountPath: /opt/rancher/
```

```
          name: utility
        - mountPath: /etc/redis/
          name: redis-conf
        ports:
          - containerPort: 6379
  volumeClaimTemplates:
  - metadata:
      labels:
        app: helm-test-slave
        type: slave
        release: "helm-test"
      name: helm-test-slave
      annotations:
        "volume.beta.kubernetes.io/storage-class": sas
        "volume.beta.kubernetes.io/storage-provisioner": flexvolume-
huawei.com/fuxivol
    spec:
      accessModes: [ "ReadWriteMany" ]
      resources:
        requests:
          storage: 10Gi
```

表12-4 关键参数说明

参数	参数说明
* annotations	用于 console 展示，其中 volume.beta.kubernetes.io/storage-class 代表云硬盘类型：sas/sata/ssd，具体参考 EVS 服务定义。volume.beta.kubernetes.io/storage-provisioner 固定为 flexvolume-huawei.com/fuxivol。
* accessModes	云硬盘访问方式，取值有如下三种： • ReadWriteOnce • ReadOnlyMany • ReadWriteMany
* resource.request.storage	云硬盘大小，单位为 Gi，最小值为 10Gi。
* failure-domain.beta.kubernetes.io/region	云硬盘所在区域。
* failure-domain.beta.kubernetes.io/zone	云硬盘所在分区。

12.1.6 使用弹性负载均衡

模板支持使用弹性负载均衡的服务类型，其定义方法与社区一致。

若要在 CCE 界面上显示 ELB 服务类型的信息，需要在对应的资源类型模板中添加如下 annotation，仅用于展示。

```
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: {{ .Release.Name }}-master
  annotations:
    "service.protal.kubernetes.io/access-ip": "10.4.4.14:8888"
    "service.protal.kubernetes.io/type": LoadBalancer
spec:
  ……
```

表12-5 关键参数说明

参数	参数说明
*annotations	用于 console 展示，其中 service.protal.kubernetes.io/access-ip 代表 ELB 实例的 IP 及暴露的端口号。service.protal.kubernetes.io/type 固定为 LoadBalancer。

13 插件管理

13.1 插件概述

CCE 提供了多种类型的插件，用于管理集群的扩展功能，以支持选择性扩展满足特性需求的功能。

表13-1 插件列表

插件名称	插件简介
13.2 CoreDNS（系统资源插件，必装）	CoreDNS 插件是一款通过链式插件的方式为 Kubernetes 提供域名解析服务的 DNS 服务器。
13.3 Everest（系统资源插件，必装）	Everest 是一个云原生容器存储系统，基于 CSI 为 Kubernetes v1.15.6 及以上版本集群对接云硬盘服务 EVS、极速文件存储 SFS Turbo 等存储服务的能力。
13.4 storage-driver（系统资源插件，必装）	storage-driver 插件是用于对接块存储、文件存储等 Iaas 存储服务的 FlexVolume 驱动。
13.5 autoscaler	集群自动扩缩容插件 autoscaler，是根据 pod 调度状态及资源使用情况对集群的工作节点进行自动扩容缩容的插件。
13.6 metrics-server	Metrics-Server 是集群核心资源监控数据的聚合器。
13.7 cce-hpa-controller	cce-hpa-controller 插件是一款 CCE 自研的插件，能够基于 CPU 利用率、内存利用率等指标，对无状态工作负载进行弹性扩缩容。
13.8 prometheus	Prometheus 是一套开源的系统监控报警框架。在云容器引擎 CCE 中，支持以插件的方式快捷安装 Prometheus。
13.9 gpu-beta	gpu-beta 插件是支持在容器中使用 GPU 显卡的设备管理插件，仅支持 Nvidia 驱动。

13.2 CoreDNS（系统资源插件，必装）

插件简介

CoreDNS 插件是一款通过链式插件的方式为 Kubernetes 提供域名解析服务的 DNS 服务器。

CoreDNS 是由 CNCF 孵化的开源软件，用于 Cloud-Native 环境下的 DNS 服务器和服务发现解决方案。CoreDNS 实现了插件链式架构，能够按需组合插件，运行效率高、配置灵活。在 kubernetes 集群中使用 CoreDNS 能够自动发现集群内的服务，并为这些服务提供域名解析。同时，通过级联 DNS 服务器，还能够为集群内的工作负载提供外部域名的解析服务。目前 CoreDNS 已经成为社区 kubernetes 1.11 及以上版本集群推荐的 DNS 服务器解决方案。

该插件为系统资源插件，kubernetes 1.11 及以上版本的集群在创建时默认安装。

CoreDNS 官网：https://coredns.io/

开源社区地址：https://github.com/coredns/coredns

说明

- 若需要升级 CoreDNS 插件，请确保集群中的节点数大于等于 CoreDNS 的实例数，且 CoreDNS 的所有实例都处于运行状态，否则将导致升级失败。
- DNS 详细使用方法请参见 6.10.11 Kubernetes 集群内置 DNS 配置说明或 3.4.3 通过 kubectl 配置 kube-dns/CoreDNS 高可用。

约束与限制

CoreDNS 正常运行需要集群中至少有两个节点，在升级 CoreDNS 插件时，请确保 CoreDNS 的所有实例都处于运行状态。

安装插件

本插件为系统默认安装，若因特殊情况卸载后，可参照如下步骤重新安装。

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“插件管理”，在“插件市场”页签下，单击 **coredns** 插件下的“安装插件”。

步骤 2 在安装插件页面，选择安装的集群和插件版本，单击“下一步：规格配置”。

步骤 3 在“规格配置”步骤中，可配置如下参数：

表13-2 cordns 插件参数配置

参数	参数说明
插件规格	并发域名解析能力，请根据业务需求选择插件规格。
实例数	选择上方插件规格后，显示插件中的实例数，此处仅作显示。
容器	选择插件规格后，显示插件容器的 CPU 和内存配额，此处仅作

参数	参数说明
	显示。
提示	关于本插件的注意事项，请仔细阅读。
存根域	单击“添加”，您可对自定义的域名配置域名服务器，格式为一个键值对，键为 DNS 后缀域名，值为一个或一组 DNS IP 地址，如"acme.local -- 1.2.3.4,6.7.8.9"。

步骤 4 完成以上配置后，单击“安装”。

待插件安装完成后，单击“返回”，在“插件实例”页签下，选择对应的集群，可查看到运行中的实例，这表明该插件已在当前集群的各节点中安装。

----结束

为 CoreDNS 配置存根域

集群管理员可以修改 CoreDNS Corefile 的 ConfigMap 以更改服务发现的工作方式。使用插件 proxy 可对 CoreDNS 的存根域进行配置。

若集群管理员有一个位于 10.150.0.1 的 Consul 域名解析服务器，并且所有 Consul 的域名都带有.consul.local 的后缀。

执行以下命令，在 CoreDNS 的 ConfigMap 中添加如下信息即可将该域名服务器配置在 CoreDNS 中：

kubectl edit configmap coredns -n kube-system

```
consul.local:5353 {
       errors
       cache 30
       proxy . 10.150.0.1
    }
```

v1.15.11 及之后的集群版本，修改后最终的 ConfigMap 如下所示：

```
apiVersion: v1
data:
  Corefile: |-
    .:5353 {
       cache 30
       errors
       health
       kubernetes cluster.local in-addr.arpa ip6.arpa {
         pods insecure
         upstream /etc/resolv.conf
         fallthrough in-addr.arpa ip6.arpa
       }
       loadbalance round_robin
       prometheus 0.0.0.0:9153
       forward . /etc/resolv.conf
       reload
```

```
    }

    consul.local:5353 {
        errors
        cache 30
        forward . 10.150.0.1
    }
kind: ConfigMap
metadata:
  name: coredns
  namespace: kube-system
```

1.15.11 之前的集群版本，修改后最终的 ConfigMap 如下所示：

```
apiVersion: v1
data:
  Corefile: |-
    .:5353 {
        cache 30
        errors
        health
        kubernetes cluster.local in-addr.arpa ip6.arpa {
          pods insecure
          upstream /etc/resolv.conf
          fallthrough in-addr.arpa ip6.arpa
        }
        loadbalance round_robin
        prometheus 0.0.0.0:9153
        proxy . /etc/resolv.conf
        reload
    }

    consul.local:5353 {
        errors
        cache 30
        proxy . 10.150.0.1
    }
kind: ConfigMap
metadata:
  name: coredns
  namespace: kube-system
```

kubernetes 中的域名解析逻辑

DNS 策略可以在每个 pod 基础上进行设置，目前，Kubernetes 支持 **Default、ClusterFirst、ClusterFirstWithHostNet 和 None** 四种 DNS 策略，具体请参见 https://kubernetes.io/docs/concepts/services-networking/dns-pod-service/。这些策略在 pod-specific 的 **dnsPolicy** 字段中指定。

- **“Default”**：如果 dnsPolicy 被设置为“Default”，则名称解析配置将从 pod 运行的节点继承。 自定义上游域名服务器和存根域不能够与这个策略一起使用。
- **“ClusterFirst”**：如果 dnsPolicy 被设置为“ClusterFirst”，任何与配置的集群域后缀不匹配的 DNS 查询（例如，www.kubernetes.io）将转发到从该节点继承的上游名称服务器。集群管理员可能配置了额外的存根域和上游 DNS 服务器。

- **“ClusterFirstWithHostNet”：**对于使用 hostNetwork 运行的 Pod，您应该明确设置其 DNS 策略“ClusterFirstWithHostNet”。
- **“None”：**它允许 Pod 忽略 Kubernetes 环境中的 DNS 设置。应使用 dnsConfigPod 规范中的字段提供所有 DNS 设置 。

说明

- Kubernetes 1.10 及以上版本，支持 Default、ClusterFirst、ClusterFirstWithHostNet 和 None 四种策略；低于 Kubernetes 1.10 版本，仅支持 default、ClusterFirst 和 ClusterFirstWithHostNet 三种。
- “Default”不是默认的 DNS 策略。如果 dnsPolicy 的 Flag 没有特别指明，则默认使用“**ClusterFirst**”。

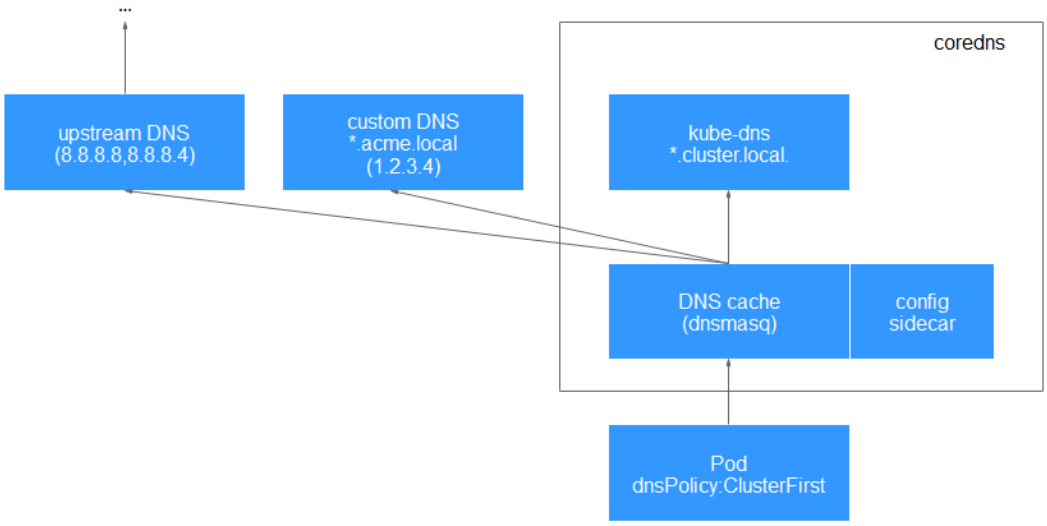
路由请求流程：

未配置存根域：没有匹配上配置的集群域名后缀的任何请求，例如“www.kubernetes.io”，将会被转发到继承自节点的上游域名服务器。

已配置存根域：如果配置了存根域和上游 DNS 服务器，DNS 查询将基于下面的流程对请求进行路由：

1. 查询首先被发送到 coredns 中的 DNS 缓存层。
2. 从缓存层，检查请求的后缀，并根据下面的情况转发到对应的 DNS 上：
   - 具有集群后缀的名字（例如“.cluster.local”）：请求被发送到 coredns。
   - 具有存根域后缀的名字（例如“.acme.local”）：请求被发送到配置的自定义 DNS 解析器（例如：监听在 1.2.3.4）。
   - 未能匹配上后缀的名字（例如“widget.com”）：请求被转发到上游 DNS。

图13-1 路由请求流程



升级插件

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“插件管理”，在“插件实例”页签下，选择对应的集群，单击 **coredns** 下的“升级”。

说明

- 如果升级按钮处于冻结状态，则说明当前插件版本是最新的版本，不需要进行升级操作。
- 升级 coredns 插件时，会替换原先节点上的旧版本的 coredns 插件，安装最新版本的 coredns 插件以实现功能的快速升级。如果升级出现异常，请卸载插件后重新安装和配置。

步骤 2 在基本信息页面选择插件版本，单击“下一步”。

步骤 3 参照表 13-3 配置插件安装参数。配置完成后，单击“升级”即可升级 coredns 插件。

表13-3 安装插件说明

参数	参数说明
插件规格	并发域名解析能力。请根据业务需求选择插件规格。
存根域	用户可对自定义的域名配置域名服务器，格式为一个键值对，键为 DNS 后缀域名，值为一个或一组 DNS IP 地址，如"acme.local --- 1.2.3.4,6.7.8.9"。

----结束

卸载插件

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“插件管理”，在“插件实例”页签下，选择对应的集群，单击 **coredns** 下的“卸载”。

步骤 2 在弹出的窗口中，单击“是”，可卸载该插件。

----结束

13.3 Everest（系统资源插件，必装）

插件简介

Everest 是一个云原生容器存储系统，基于 CSI 为 Kubernetes v1.15.6 及以上版本集群对接云硬盘服务 EVS、对象存储服务 OBS、弹性文件服务 SFS、极速文件存储 SFS Turbo 等存储服务的能力。

该插件为系统资源插件，kubernetes 1.15 及以上版本的集群在创建时默认安装。

说明

- v1.13.11-r1 升级到 v1.15.11-r1 集群后，原本 v1.13 的 Flexvolume Fuxi 容器存储由 v1.15 的 Everest 接管（插件版本 v1.1.6 及以上），原有功能保持不变。
- Everest 插件在 **1.2.0 版本**优化了使用 OBS 存储时的**秘钥认证功能**，请在 Everest 插件升级完成后（从低于 1.2.0 的版本升级到 1.2.0 及以上版本），重启集群中使用 obs 的全部工作负载，否则工作负载使用 OBS 存储能力将受影响！

约束与限制

- 仅支持 **v1.15 及以上版本**的 CCE 集群安装本插件。
- v1.13 及以下版本集群创建时默认必装 13.4 storage-driver（系统资源插件，必装）插件。

安装插件

本插件为系统默认安装，若因特殊情况卸载后，可参照如下步骤重新安装。

步骤 1 在 CCE 控制台中，单击左侧导航栏的“插件管理”，在“插件市场”页签下，单击“everest”插件下的“安装插件”。

步骤 2 在安装插件页面，选择安装的集群和插件版本，单击“下一步：规格配置”。

步骤 3 该插件可配置“单实例”或“高可用”规格，选择后单击“安装”。

待插件安装完成后，单击“返回”，在“插件实例”页签下，选择对应的集群，可查看到运行中的实例，这表明该插件已在当前集群的各节点中安装。

----结束

升级插件

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“插件管理”，在“插件实例”页签下，选择对应的集群，单击“everest”下的“ 升级”。

说明

- 如果升级按钮处于冻结状态，则说明当前插件版本是最新的版本，不需要进行升级操作。
- 升级 everest 插件时，会替换原先节点上的旧版本的 everest 插件，安装最新版本的 everest 插件以实现功能的快速升级。

步骤 2 在基本信息页面选择插件版本，单击“下一步”。

步骤 3 该插件可配置“单实例”或“高可用”规格，选择后单击“升级”即可升级“everest”插件。

----结束

卸载插件

步骤 1 在 CCE 控制台中，单击左侧导航栏的“插件管理”，在“插件实例”页签下，选择对应的集群，单击“everest”下的“卸载”。

步骤 2 在弹出的窗口中，单击“是”，可卸载该插件。

----结束

13.4 storage-driver（系统资源插件，必装）

插件简介

storage-driver 插件是用于对接块存储、极速文件存储等 Iaas 存储服务的 FlexVolume 驱动。

storage-driver 是一款云存储驱动插件，北向遵循标准容器平台存储驱动接口。实现 Kubernetes Flex Volume 标准接口，提供容器使用 EVS 块存储、极速文件存储等 IAAS 存储的能力。通过安装升级云存储插件可以实现云存储功能的快速安装和更新升级。

该插件为系统资源插件，kubernetes 1.13 及以下版本的集群在创建时默认安装。

说明

- v1.15.11-r1 是 Flexvolume fuxi（即 storage-driver）和 CSI（即 13.3 Everest（系统资源插件，必装）插件）兼容接管的过渡版本，在 v1.17 集群中已经去除了对 Flexvolume fuxi 的支持，请您将 Flexvolume fuxi 的使用切换 CSI Everest 上。
- CSI Everest 兼容接管 Flexvolume fuxi 后，原有功能保持不变，但请注意不要新建 fuxi Flexvolume 的存储，否则将导致部分存储功能异常。
- 当存储功能有升级或者 BUG 修复时，用户无需升级集群或新建集群来升级存储功能，仅需安装或升级 storage-driver 插件。

约束与限制

本插件仅支持在 **v1.13 及以下版本**的集群中安装，v1.15 及以上版本集群创建时默认必装 13.3 Everest（系统资源插件，必装）插件。

安装插件

本插件为系统默认安装，若因特殊情况卸载后，可参照如下步骤重新安装。

在云容器引擎 CCE 中，kubernetes 1.11 及以上版本的新建集群，将会默认安装 storage-driver 插件。

未安装 storage-driver 插件的集群，可参考如下步骤进行安装：

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“插件管理”，在“插件市场”页签下，单击 **storage-driver** 插件下的“安装插件”。

步骤 2 在安装插件页面，选择安装的集群和插件版本，单击“下一步：规格配置”。

步骤 3 云存储插件暂未开放可配置参数，直接单击“安装”。

待插件安装完成后，单击“返回”，在“插件实例”页签下，选择对应的集群，可查看到运行中的实例，这表明该插件已在当前集群的各节点中安装。

----结束

升级插件

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“插件管理”，在“插件实例”页签下，选择对应的集群，单击 **storage-driver** 下的“升级”。

说明

- 如果升级按钮处于冻结状态，则说明当前插件版本是最新的版本，不需要进行升级操作。
- 升级 storage-driver 插件时，会替换原先节点上的旧版本的 storage-driver 插件，安装最新版本的 storage-driver 插件以实现功能的快速升级。

步骤 2 在基本信息页面选择插件版本，单击“下一步”。

步骤 3 云存储插件暂未开放可配置参数，直接单击“升级”即可升级 storage-driver 插件。

----结束

卸载插件

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“插件管理”，在“插件实例”页签下，选择对应的集群，单击 **storage-driver** 下的“卸载”。

步骤 2 在弹出的窗口中，单击“是”，可卸载该插件。

----结束

13.5 autoscaler

插件简介

autoscaler 是 Kubernetes 中非常重要的一个 Controller，它提供了微服务的弹性能力，并且和 Serverless 密切相关。

弹性伸缩是很好理解的一个概念，当微服务负载高（CPU/内存使用率过高）时水平扩容，增加 pod 的数量以降低负载，当负载降低时减少 pod 的数量，减少资源的消耗，通过这种方式使得微服务始终稳定在一个理想的状态。

云容器引擎简化了 Kubernetes 集群的创建、升级和手动扩缩容，而集群中应用的负载本身是会随着时间动态变化的，为了更好的平衡资源使用率以及性能，kubernetes 引入了 autoscaler 插件，它可以根据部署的应用所请求的资源量自动的动态的伸缩集群，详情请了解 zh-cn_topic_0000001092643222.xml。

开源社区地址：https://github.com/kubernetes/autoscaler

插件说明

autoscaler 可分成扩容和缩容两个方面：

- **自动扩容**

  当集群中的 Pod 由于工作节点资源不足而无法调度时，会触发集群扩容，扩容节点与所在分组资源配额一致。详情请参见 zh-cn_topic_0000001092643222.xml。

  当前该插件使用的是最小浪费策略，即若 pod 创建需要 3 核，此时有 4 核、8 核两种规格，优先创建规格为 4 核的节点。

  说明

  自动扩容策略在满足如下条件时才会执行：

  - 节点上的资源不足。
  - Pod 的调度配置中不能包含节点亲和的策略（即 Pod 若已经设置亲和某个节点，则不会自动扩容节点），节点亲和策略设置方法请参见 7.2.1 节点亲和性。

- **自动缩容**

  当集群节点处于一段时间空闲状态时（默认 10min），会触发集群缩容操作（即节点会被自动删除）。当节点存在以下几种状态的 pod 时，不可缩容：

  - pod 有设置 PodDisruptionBudget，当移除 pod 不满足对应条件时，节点不会缩容。
  - pod 由于一些限制，如亲和、反亲和等，无法调度到其他节点，节点不会缩容。
  - pod 拥有 cluster-autoscaler.kubernetes.io/safe-to-evict: 'false'这个 annotations 时，节点不缩容。
  - 节点上存在 kube-system namespace 下的 Pod（除 kube-system daemonset 创建的 Pod），节点不缩容。
  - 节点上有非 controller（deployment/replica set/job/stateful set）创建的 Pod，节点不缩容。

约束与限制

- 集群为 1.9.7-r1 及以上版本时，才支持此功能。
- 安装时请确保有足够的资源安装本插件。
- 该插件功能仅支持通过**按需计费**方式购买的**虚拟机节点**。

安装插件

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“插件管理”，在“插件市场”页签下，单击 **autoscaler** 插件下的“安装插件”。

步骤 2 在安装插件页面，选择您要安装插件的集群和插件版本，单击“下一步：规格配置”。

步骤 3 参照表 13-4 配置插件安装参数。

表13-4 基本配置

参数	支持的插件版本	参数说明
插件规格	所有版本	插件部署可选择“单实例”和“高可用”两种规格。 • 单实例：以单实例部署插件。 • 高可用：以多实例部署插件，具有高可用能力，需占用更多的计算资源。
实例数	所有版本	选择上方插件规格后，显示插件中的实例数，此处仅作显示。
容器	所有版本	选择插件规格后，显示插件容器的 CPU 和内存配额，此处仅作显示。
登录方式	部分版本无此参数	为自动扩容工作节点选择登录方式，当前支持“密码”和“密钥对”两种方式，您可根据需要进行选择。 **选择“密码”方式：** • 密码：自动扩容工作节点密码。通过该密码登录工作节点执行相关操作，用户名为 root。 • 确认密码：再次输入自动扩容工作节点密码。 **选择“密钥对”方式：** 密钥对：密钥对用于远程登录节点时的身份认证。
自动缩容	所有版本	默认不开启：将不开启自动缩容，只进行自动扩容。 开启：开启自动缩容，且自建节点和插件扩容所创建的节点均可缩容。 • 空置时间（min）：当集群节点处于一段时间的空闲状态时，会触发集群缩容操作，删除节点，默认 10min。 • 百分比：当集群节点资源低于多少百分比时，进行集群缩容扫描（默认 0.5，即 50%，cpu 和 mem 都要满足的条件下才会缩容）。 • 扩容后缩容冷却时间：扩容执行后能再次启动缩容评估的时间间隔，默认 10min。 • 节点删除后缩容冷却时间：删除节点后能再次启动缩容评估的时间间隔，默认 10min。 • 缩容失败后缩容冷却时间：缩容失败后能再次启动缩容评估的时间间隔，默认 3min。 • 空节点缩容最大并发数：默认 10。 • 不可移除节点的重检查时间：节点被判定不可移除后能再次启动检查的时间间隔，默认

参数	支持的插件版本	参数说明
		5min。 说明 当节点处于以下几种状态时，不可缩容： • pod 设置为 PodDisruptionBudget，当不满足条件时，节点不会缩容。 • pod 设置本地存储时，节点不会缩容。 • pod 由于一些限制，如亲和、反亲和等，无法调度到其他节点，节点不会缩容。 • pod 拥有 cluster-autoscaler.kubernetes.io/safe-to-evict: 'false'这个 annotations 时，节点不缩容。 • 节点上存在 kube-system 命名空间下的 Pod（除 kube-system daemonset 创建的 Pod）。 • 节点上有非 controller（deployment/replica set/job/stateful set）创建的 Pod。
预置节点池配置	部分版本无此参数	此为默认资源组。集群扩容时，若没有其他可用分组，将使用默认分组的资源规格进行扩容节点。 **单击“添加预置节点池配置”进行参数设置：** • 可用区：选择一个可用区。可用区是指在同一区域下，电力、网络隔离的物理区域。 • 操作系统：选择节点对应的操作系统。 • Taints：可选项，默认为空。 支持给该节点池扩容出来的节点加 Taints 来设置反亲和性，每个节点池最多配置 10 条 Taints，每条 Taints 包含以下 3 个参数： – Key：必须以字母或数字开头，可以包含字母、数字、连字符、下划线和点，最长 63 个字符；另外可以使用 DNS 子域作为前缀。 – Value：必须以字母或数字开头，可以包含字母、数字、连字符、下划线和点，最长 63 个字符。 – Effect：只可选 NoSchedule，PreferNoSchedule 或 NoExecute。 须知 • Taints 配置时需要配合 Pod 的 toleration 使用，否则可能导致扩容失败或者 Pod 无法调度到扩容节点。 • 配置后无法修改，请您谨慎配置，错误的配置可能会导致扩容失败或 pod 无法调度。 • 标签管理：通过为资源添加标签，可以对资源

参数	支持的插件版本	参数说明
		进行自定义标记，实现资源的分类。 说明 您可以在 TMS 中创建“预定义标签”，预定义标签对所有支持标签功能的服务资源可见，通过使用预定义标签可以提升标签创建和迁移效率。 节点规格：节点的 CPU 配置和内存配置。

单击下方的“高级设置”，可以配置更多插件参数：

表13-5 高级设置

参数	插件版本	参数说明
节点总数	所有版本	可扩容的最大节点总数。
cpu 总数（核）	所有版本	可扩容的最大 cpu 总数（核）。
内存总数（GB）	所有版本	可扩容的最大内存总数（GB）。
自动扩容	部分版本无此参数	**未调度实例扩容：**默认选中。 **启用集群使用率扩容：**可选，扩容能力增强。 • cpu 扩容使用率：当节点池 CPU 达到所设置的使用上限，将扩容当前预置节点池的节点数。 • 内存扩容使用率：当节点池内存达到所设置的使用上限，将扩容当前预置节点池的节点数。
磁盘	部分版本无此参数	系统盘和数据盘：设置节点磁盘空间。 • 系统盘：规格为[40,1024]GB，用户可以配置，缺省值为 40GB。 • 数据盘：规格为[100,32678]GB，用户可以配置，缺省值为 100GB。 勾选“资源分配自定义”后，您可以对数据盘中的 Docker 和 Kubelet 资源占比进行自定义设置。 须知 • 磁盘使用 direct-lvm 模式，移除将使用 loop-lvm 模式，有影响系统稳定性的风险。 • Docker 资源包含 Docker 镜像数据以及镜像元数据，Kubelet 资源包含 Pod 配置文件、密钥以及临时存储 EmptyDir 等挂载数据。

参数	插件版本	参数说明
		• 数据盘：单击“新增数据盘”，您可以增加一块数据盘。 系统盘和数据盘均可提供以下性能规格的云硬盘： • 普通 IO：是指由 SATA 存储提供资源的磁盘类型。提供可靠的块存储，单个云硬盘的最大 IOPS 可达到 1000，可运行关键应用程序。 • 高 IO：是指由 SAS 存储提供资源的磁盘类型。提供可达到 3000 的高 IO 和低至 1 ms 的读写延时，支持 NoSQL/关系型数据库，数据仓库，文件系统等应用。 • 超高 IO：是指由 SSD 存储提供资源的磁盘类型。提供可达到 20000 的超高 IO 和低至 1 ms 超低读写时延，支持 NoSQL/关系型数据库，数据仓库等应用。
命令行注入	部分版本无此参数	请输入脚本命令。 **说明** 脚本命令大小限制：0~1000 字符。

步骤 4 配置完成后，单击“安装”。

待插件安装完成后，单击“返回”，在“插件实例”页签下，选择对应的集群，可查看到运行中的实例，这表明该插件已在当前集群的各节点中安装。

----结束

升级插件

步骤 1 在 CCE 控制台中，单击左侧导航栏的“插件管理”，在“插件实例”页签下，选择对应的集群，单击 autoscaler 下的“ 升级”。

说明

- 如果升级按钮处于冻结状态，则说明当前插件版本是最新的版本，不需要进行升级操作；
- 若升级按钮可单击，则单击升级按钮即可升级 autoscaler 插件。
- 升级 autoscaler 插件时，会替换原先节点上的旧版本的 autoscaler 插件，安装最新版本的 autoscaler 插件以实现功能的快速升级。

步骤 2 在弹出的窗口中，配置参数后，升级该插件。配置参数可参考安装插件中的参数说明。

----结束

卸载插件

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“插件管理”，在“插件实例”页签下，选择对应的集群，单击 **autoscaler** 下的“卸载”。

步骤 2 在弹出的窗口中，单击“是”，可卸载该插件。

----结束

13.6 metrics-server

从 Kubernetes 1.8 开始，Kubernetes 通过 Metrics API 提供资源使用指标，例如容器 CPU 和内存使用率。这些度量可以由用户直接访问（例如，通过使用 kubectl top 命令），或者由集群中的控制器（例如，Horizontal Pod Autoscaler）使用来进行决策，具体的组件为 Metrics-Server，用来替换之前的 heapster，heapster 从 1.11 开始逐渐被废弃。

Metrics Server 是集群核心资源监控数据的聚合器，您可以在 CCE 控制台中快速安装本插件。

社区官方项目及文档：https://github.com/kubernetes-sigs/metrics-server。

约束与限制

仅支持 v1.13 及以上版本的 CCE 集群安装本插件。

安装插件

安装本插件后，可在“弹性伸缩”页面的“工作负载伸缩”页签下，创建 HPA 策略，具体请参见 zh-cn_topic_0000001092291708.xml。

步骤 1 在 CCE 控制台中，单击左侧导航栏的“插件管理”，在“插件市场”页签下，单击“metrics-server”下的“安装插件”按钮。

步骤 2 在安装插件页面，选择安装的集群和插件版本，单击“下一步：规格配置”。

步骤 3 该插件可配置“单实例”或“高可用”规格，选择后单击“安装”。

待插件安装完成后，单击“返回”，在“插件实例”页签下，选择对应的集群，可查看到运行中的实例，这表明该插件已在当前集群的各节点中安装。

----结束

升级插件

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“插件管理”，在“插件实例”页签下，选择对应的集群，单击“metrics-server”下的“ 升级”。

说明

- 如果升级按钮处于冻结状态，则说明当前插件版本是最新的版本，不需要进行升级操作。

- 升级“metrics-server”插件时，会替换原先节点上的旧版本的“metrics-server”插件，安装最新版本的“metrics-server”插件以实现功能的快速升级。

步骤 2 在基本信息页面选择插件版本，单击“下一步”。

步骤 3 参考安装插件中参数说明配置参数后，单击“升级”即可升级“metrics-server”插件。

----结束

卸载插件

步骤 1 在 CCE 控制台中，单击左侧导航栏的“插件管理”，在“插件实例”页签下，选择对应的集群，单击“Metrics Server”下的“卸载”。

步骤 2 在弹出的窗口中，单击“是”，可卸载该插件。

----结束

13.7 cce-hpa-controller

cce-hpa-controller 插件是一款 CCE 自研的插件，能够基于 CPU 利用率、内存利用率等指标，对无状态工作负载进行弹性扩缩容。

主要功能

- 支持按照当前实例数的百分比进行扩缩容。
- 支持设置一次扩缩容的最小步长。
- 支持按照实际指标值执行不同的扩缩容动作。

约束与限制

- 仅支持 v1.15 及以上版本的 CCE 集群安装本插件。
- 安装本插件后，还需要安装 13.8 prometheus（1.1.0 及以上版本），用于指标的采集。

安装插件

安装本插件后，可在“弹性伸缩”页面的“工作负载伸缩”页签下，创建 CustomedHPA 策略，具体请参见 zh-cn_topic_0000001092291708.xml。

步骤 1 在 CCE 控制台中，单击左侧导航栏的“插件管理”，在“插件市场”页签下，单击“cce-hpa-controller”下的“安装插件”按钮。

步骤 2 在安装插件页面，选择安装的集群和插件版本，单击“下一步：规格配置”。

步骤 3 该插件可配置“单实例”或“高可用”规格，选择后单击“安装”。

待插件安装完成后，单击“返回”，在“插件实例”页签下，选择对应的集群，可查看到运行中的实例，这表明该插件已在当前集群的各节点中安装。

----结束

升级插件

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“插件管理”，在“插件实例”页签下，选择对应的集群，单击“cce-hpa-controller”下的“ 升级”。

说明

- 如果升级按钮处于冻结状态，则说明当前插件版本是最新的版本，不需要进行升级操作。
- 升级“cce-hpa-controller”插件时，会替换原先节点上的旧版本的“cce-hpa-controller”插件，安装最新版本的“cce-hpa-controller”插件以实现功能的快速升级。

步骤 2 在基本信息页面选择插件版本，单击“下一步”。

步骤 3 参考安装插件中参数说明配置参数后，单击“升级”即可升级“cce-hpa-controller”插件。

----结束

卸载插件

步骤 1 在 CCE 控制台中，单击左侧导航栏的“插件管理”，在“插件实例”页签下，选择对应的集群，单击“cce-hpa-controller”下的“卸载”。

步骤 2 在弹出的窗口中，单击“是”，可卸载该插件。

----结束

13.8 prometheus

插件简介

Prometheus 是一套开源的系统监控报警框架。它启发于 Google 的 borgmon 监控系统，由工作在 SoundCloud 的 Google 前员工在 2012 年创建，作为社区开源项目进行开发，并于 2015 年正式发布。2016 年，Prometheus 正式加入 Cloud Native Computing Foundation，成为受欢迎度仅次于 Kubernetes 的项目。

在云容器引擎 CCE 中，支持以插件的方式快捷安装 Prometheus。

插件官网：https://prometheus.io/

开源社区地址：https://github.com/prometheus/prometheus

插件特点

作为新一代的监控框架，Prometheus 具有以下特点：

- 强大的多维度数据模型：
  a. 时间序列数据通过 metric 名和键值对来区分。
  b. 所有的 metrics 都可以设置任意的多维标签。
  c. 数据模型更随意，不需要刻意设置为以点分隔的字符串。

d. 可以对数据模型进行聚合，切割和切片操作。

e. 支持双精度浮点类型，标签可以设为全 unicode。

- 灵活而强大的查询语句（PromQL）：在同一个查询语句，可以对多个 metrics 进行乘法、加法、连接、取分数位等操作。
- 易于管理：Prometheus server 是一个单独的二进制文件，可直接在本地工作，不依赖于分布式存储。
- 高效：平均每个采样点仅占 3.5 bytes，且一个 Prometheus server 可以处理数百万的 metrics。
- 使用 pull 模式采集时间序列数据，这样不仅有利于本机测试而且可以避免有问题的服务器推送坏的 metrics。
- 可以采用 push gateway 的方式把时间序列数据推送至 Prometheus server 端。
- 可以通过服务发现或者静态配置去获取监控的 targets。
- 有多种可视化图形界面。
- 易于伸缩。

需要指出的是，由于数据采集可能会有丢失，所以 Prometheus 不适用对采集数据要 100%准确的情形。但如果用于记录时间序列数据，Prometheus 具有很大的查询优势，此外，Prometheus 适用于微服务的体系架构。

约束与限制

1.11 及以上版本的 CCE 集群支持此插件功能。

安装插件

步骤 1 在 CCE 控制台中，单击左侧导航栏的“插件管理”，在“插件市场”页签下，单击 **Prometheus** 下的“安装插件”按钮。

步骤 2 在安装插件页面，选择安装的集群和插件版本，单击“下一步：规格配置”。

步骤 3 在“规格配置”步骤中，配置以下参数：

表13-6 Prometheus 配置参数说明

参数	参数说明
插件规格	根据业务需求，选择插件的规格，包含如下选项： • 演示规格（100 容器以内）：适用于体验和功能演示环境，该规模下 prometheus 占用资源较少，但处理能力有限。建议在集群内容器数目不超过 100 时使用。 • 小规格（2000 容器以内）：建议在集群中的容器数目不超过 2000 时使用。 • 中规格（5000 容器以内）：建议在集群中的容器数目不超过 5000 时使用。 • 大规格（超过 5000 容器）：建议集群中容器数目超过 5000 时使用此规格。

参数	参数说明
实例数	选择上方插件规格后，显示插件中的实例数，此处仅作显示。
容器	选择插件规格后，显示插件容器的 CPU 和内存配额，此处仅作显示。
监控数据保留期	自定义监控数据需要保留的天数，默认为 15 天。
存储	按照界面提示配置如下参数： • 类型：支持云硬盘。 • 可用区：请根据业务需要进行选择。可用区是在同一区域下，电力、网络隔离的物理区域，可用区之间内网互通，不同可用区之间物理隔离。 • 子类型：支持普通 IO、高 IO 和超高 IO 三种类型。 • 容量：请根据业务需要输入存储容量，默认 10G。 说明 若命名空间 monitoring 下已存在 pvc，将使用此存储作为存储源。

步骤 4 单击“安装”。

待插件安装完成后，单击“返回”，在“插件实例”页签下，选择对应的集群，可查看到运行中的实例，这表明该插件已在当前集群的各节点中安装。

步骤 5 在 CCE 控制台中，单击左侧导航栏的“插件管理”，在“插件实例”中，单击“Prometheus”进入详情页，可以查看插件实例的详细情况。

----结束

升级插件

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“插件管理”，在“插件实例”页签下，选择对应的集群，单击“Prometheus”下的“ 升级”。

说明

- 如果升级按钮处于冻结状态，则说明当前插件版本是最新的版本，不需要进行升级操作。
- 升级“Prometheus”插件时，会替换原先节点上的旧版本的“Prometheus”插件，安装最新版本的“Prometheus”插件以实现功能的快速升级。

步骤 2 在基本信息页面选择插件版本，单击“下一步”。

步骤 3 参考安装插件中参数说明配置参数后，单击“升级”即可升级“Prometheus”插件。

----结束

卸载插件

步骤 1 在 CCE 控制台中，单击左侧导航栏的“插件管理”，在“插件实例”页签下，选择对应的集群，单击 **Prometheus** 下的“卸载”。

步骤 2 在弹出的窗口中，单击“是”，可卸载该插件。

----结束

参考资源

- Prometheus 概念及详细配置请参阅 Prometheus 官方文档
- Node exporter 安装请参考 node_exporter github 仓库
- Slack 信息发送请参考 Incoming Webhooks

13.9 gpu-beta

插件简介

gpu-beta 插件是支持在容器中使用 GPU 显卡的设备管理插件，仅支持 Nvidia 驱动。

约束与限制

- 安装 gpu-beta 插件的集群中必须包含 GPU 节点。
- 下载的驱动必须是后缀为“.run”的文件。

须知

- 若下载链接为公网地址，如 Nvidia 官网，各 GPU 节点均需要绑定 EIP。
- 若下载链接为 OBS 上的链接，无需绑定 EIP 。
- 请确保 Nvidia 驱动版本与 GPU 节点适配。
- 更改驱动版本后，需要重启节点才能生效。

获取驱动链接-公网地址

步骤 1 登录到 *https://www.nvidia.cn/Download/Find.aspx?lang=cn* 网站。

步骤 2 如图 13-2 所示，在“NVIDIA 驱动程序下载”框内选择对应的驱动信息。其中“操作系统”必须选 **Linux 64-bit**。

图13-2 参数选择

步骤 3 驱动信息确认完毕，单击“搜索”按钮，会跳转到驱动信息展示页面，该页面会显示驱动的版本信息如图 13-3，单击“下载”到下载页面。

图13-3 驱动信息

步骤 4 获取驱动软件链接方式分两种：

- 方式一：如图 13-4，在浏览器的链接中找到路径为 *url=/tesla/396.37/NVIDIA-Linux-x86_64-396.37.run* 的路径，补齐全路径 https://us.download.nvidia.com/tesla/396.37/NVIDIA-Linux-x86_64-396.37.run 该方式节点需要绑定 EIP 。
- 方式二：如图 13-4，单击“下载”按钮下载驱动，然后上传到 OBS，获取软件的链接，该方式节点不需要绑定 EIP。

图13-4 获取链接

----结束

获取驱动链接-OBS 地址

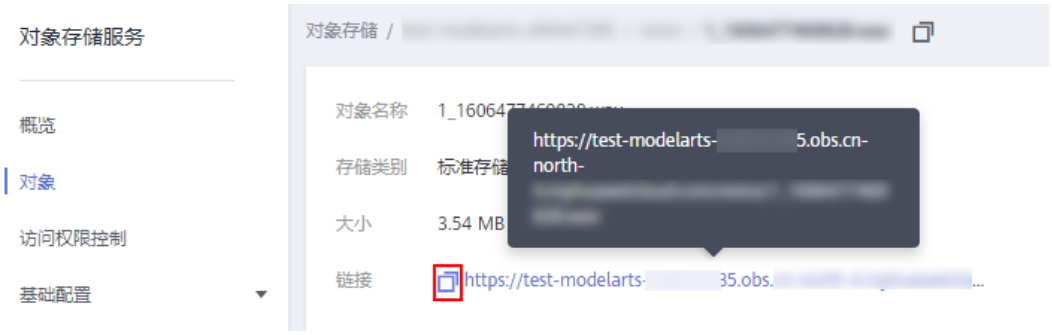
步骤 1 将驱动上传到对象存储服务 OBS 中。

步骤 2 在 OBS 管理控制台左侧导航栏选择“对象存储”。

步骤 3 在桶列表单击待操作的桶，进入“概览”页面。

步骤 4 在左侧导航栏，单击“对象”。

步骤 5 选中目标对象，在对象详情页复制驱动链接。



----结束

安装插件

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“插件管理”。在“插件市场”页签下，单击 gpu-beta 插件下的“安装插件”。

步骤 2 在安装插件页面，选择安装的集群和插件版本，单击“下一步：规格配置”。

步骤 3 在规格配置页面，配置驱动链接地址。如何获取驱动链接，请参考获取驱动链接-公网地址。

例如，nvidia396.37 版本驱动链接：*https://us.download.nvidia.com/tesla/396.37/NVIDIA-Linux-x86_64-396.37.run*。

步骤 4 单击“安装”。

安装完成后，在当前集群的各 GPU 节点上会安装该插件实例。

----结束

升级插件

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“插件管理”，在“插件实例”页签下，选择对应的集群，单击“gpu-beta”下的“升级”。

说明

- 如果升级按钮处于冻结状态，则说明当前插件版本是最新的版本，不需要进行升级操作。
- 升级“gpu-beta”插件时，会替换原先节点上的旧版本的“gpu-beta”插件，安装最新版本的“gpu-beta”插件以实现功能的快速升级。

步骤 2 在基本信息页面选择插件版本，单击“下一步”。

步骤 3 参考安装插件中参数说明配置参数后，单击“升级”即可升级“gpu-beta”插件。

步骤 4 重启节点（必须）。

在 ECS 控制台重启节点。登录官网，选择“控制台”，选择弹性云主机所在区域，选择“计算 > 弹性云主机 ECS”，在弹性云主机列表中找到目标节点，并单击“操作”列下的“更多 > 重启”。

----结束

卸载插件

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“插件管理”。在“插件实例”页签下，选择对应的集群，单击“gpu-beta”下的“卸载”。

步骤 2 在弹出的窗口中，单击“是”，可卸载该插件。

说明

卸载 gpu-beta 插件不会卸载驱动。若重新安装了驱动，需重启所有 GPU 节点。

----结束

14 弹性伸缩

14.1 弹性伸缩概述

弹性伸缩是根据业务需求和策略，经济地自动调整弹性计算资源的管理服务。

背景介绍

随着 Kubernetes 已经成为云原生应用编排、管理的事实标准，越来越多的应用选择向 Kubernetes 迁移，用户也越来越关心在 Kubernetes 上应用如何快速扩容面对业务高峰，以及如何在业务低谷时快速缩容节约资源与成本。

在 Kubernetes 的集群中，“弹性伸缩”一般涉及到扩缩容 Pod 个数以及 Node 个数。Pod 代表应用的实例数（每个 Pod 包含一个或多个容器），当业务高峰的时候需要扩容应用的实例个数。所有的 Pod 都是运行在某一个节点（虚机或裸机）上，当集群中没有足够多的节点来调度新扩容的 Pod，那么就需要为集群增加节点，从而保证业务能够正常提供服务。

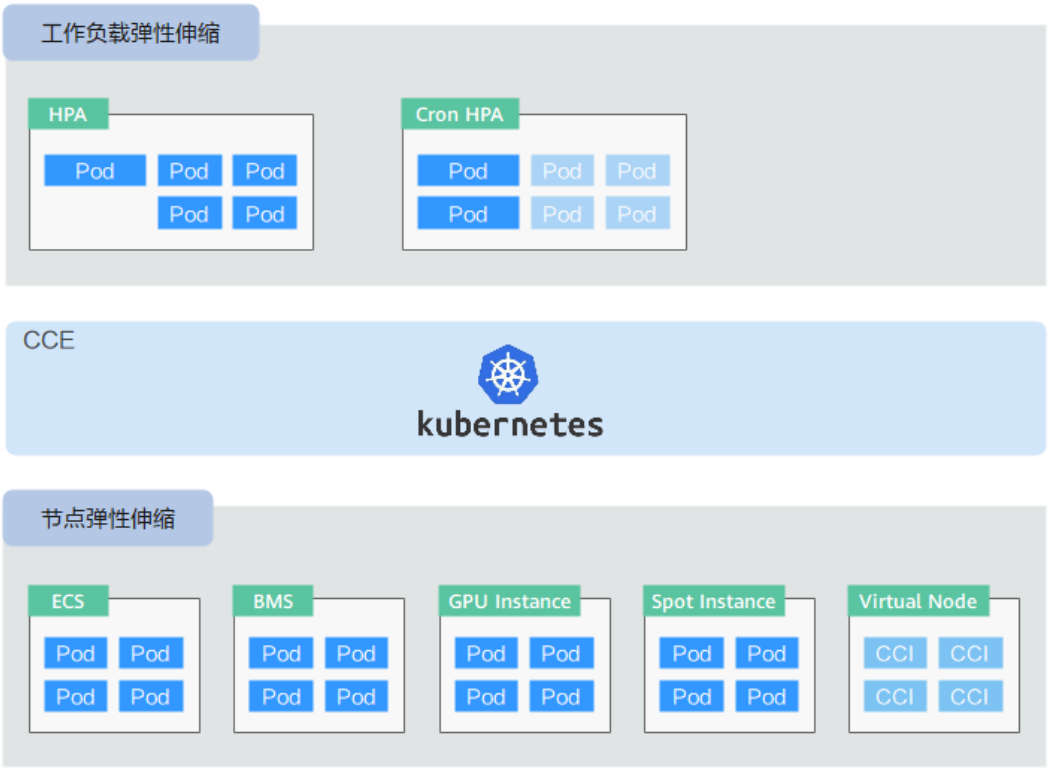
弹性伸缩在 CCE 上的使用场景非常广泛，典型的场景包含在线业务弹性、大规模计算训练、深度学习 GPU 或共享 GPU 的训练与推理、定时周期性负载变化等。

弹性伸缩分为如下两个维度：

- **14.2** 工作负载弹性伸缩：即调度层弹性，主要是负责修改负载的调度容量变化。例如，HPA 是典型的调度层弹性组件，通过 HPA 可以调整应用的副本数，调整的副本数会改变当前负载占用的调度容量，从而实现调度层的伸缩。
- **14.3** 集群/节点弹性伸缩：即资源层弹性，主要是集群的容量规划不能满足集群调度容量时，会通过弹出 ECS 或 CCI 等资源的方式进行调度容量的补充。

两个维度的弹性组件与能力可以分开使用，也可以结合在一起使用，并且两者之间可以通过调度层面的容量状态进行解耦。

CCE 弹性伸缩组件介绍



CCE 支持的两种弹性伸缩对比

表14-1 CCE 支持的两种弹性伸缩对比

弹性伸缩	基于 Kubernetes 插件实现（推荐）	基于 AOM 服务实现
实现方式	基于 CCE 提供的插件 13.7 cce-hpa-controller 实现工作负载弹性伸缩 基于 CCE 提供的插件 13.5 autoscaler 实现节点弹性伸缩	基于 AOM 服务实现弹性伸缩，伸缩指标和触发伸缩均由 AOM 服务提供
使用入口	**工作负载伸缩** 1. 在 CCE 左侧导航栏中选择“弹性伸缩”，进入“工作负载伸缩”。 2. 在左侧导航栏中选择“工作负载 > 无状态负载 Deployment/有状态负载 StatefulSet”，进入工作负载详情页，在“伸缩”页签下“弹性伸缩 - HPA”。 **节点伸缩**	**工作负载伸缩** 在 CCE 左侧导航栏中选择“工作负载 > 无状态负载 Deployment/有状态负载 StatefulSet”，单击工作负载名称进入工作负载详情页，在“伸缩”页签下选择“弹性伸缩 - AOM”。详情请参见弹性伸缩-AOM。 **节点扩容** 在 CCE 左侧导航栏中选择“资源管理 > 集群管理”，进入集群详情页，选择“弹性扩容”页签。详情请

弹性伸缩	基于 Kubernetes 插件实现（推荐）	基于 AOM 服务实现
	在 CCE 左侧导航栏中选择“弹性伸缩”，进入“节点伸缩”。	参见 3.5 集群弹性扩容。
约束与限制	• 请在使用的集群中安装对应的插件，详情请参见 14.2.2 创建工作负载弹性伸缩（HPA）、14.3.2 创建节点伸缩策略。 • 节点伸缩仅针对节点池下的节点，使用弹性伸缩的节点池需开启弹性扩缩容功能（在创建/编辑节点池时指定）。	当前区域的 AOM 服务支持伸缩功能，并且 CCE 集群版本低于 v1.17。
使用须知	• 支持监控指标伸缩和定时伸缩两种策略的负载伸缩。 • 支持调度扩容（在负载实例无法调度时自动从节点池中扩容节点）、监控指标扩容、定时扩容三种节点扩容策略。 • 节点缩容仅支持资源分配率缩容机制，当集群下 CPU 和内存分配率低于用户设置的门限（在安装/编辑 13.5 autoscaler 时设置）时将对节点池下节点启动缩容（该功能可以关闭）。 • defaultpool 仅用于兼容非节点池创建的节点。	• 支持监控指标伸缩和定时伸缩两种策略的负载伸缩。 • 支持监控指标和定时两种策略的节点扩容，不支持节点缩容。 • 节点仅支持设置一个节点配置（在集群详情页的“弹性扩容”页签下进行设置），且节点模板支持的节点可配参数较少。

14.2 工作负载弹性伸缩

14.2.1 工作负载伸缩原理

CCE 支持 HPA 策略和 CustomedHPA 策略两种工作负载伸缩方式。两种策略的对比如下：

表14-2 HPA 和 CustomedHPA 策略对比

伸缩策略	HPA 策略	CustomedHPA 策略
实现方式	Kubernetes 中实现 POD 水平自动伸缩的功能，即 Horizontal Pod Autoscaling	自研的弹性伸缩增强能力

伸缩策略	HPA 策略	CustomedHPA 策略
策略规则	基于指标（CPU 利用率、内存利用率），对无状态工作负载进行弹性扩缩容。	基于指标（CPU 利用率、内存利用率）或周期（每天、每周、每月或每年的具体时间点），对无状态工作负载进行弹性扩缩容。
主要功能	在 kubernetes 社区 HPA 功能的基础上，增加了应用级别的冷却时间窗和扩缩容阈值等功能。	指标触发 • 支持按照当前实例数的百分比进行扩缩容。 • 支持设置一次扩缩容的最小步长。 • 支持按照实际指标值执行不同的扩缩容动作。 周期触发 支持选择天、周、月或年的具体时间点或周期作为触发时间

说明

伸缩策略优先级：手动伸缩和自动伸缩同时配置的情况下，在不执行手动伸缩时，资源调度以自动伸缩为准，伸缩按照监控信息自动触发，如果使用手动伸缩，自动伸缩会暂时失效。

HPA 工作原理

HPA（Horizontal Pod Autoscaler）是用来控制 Pod 水平伸缩的控制器，HPA 周期性检查 Pod 的度量数据，计算满足 HPA 资源所配置的目标数值所需的副本数量，进而调整目标资源（如 Deployment）的 replicas 字段。

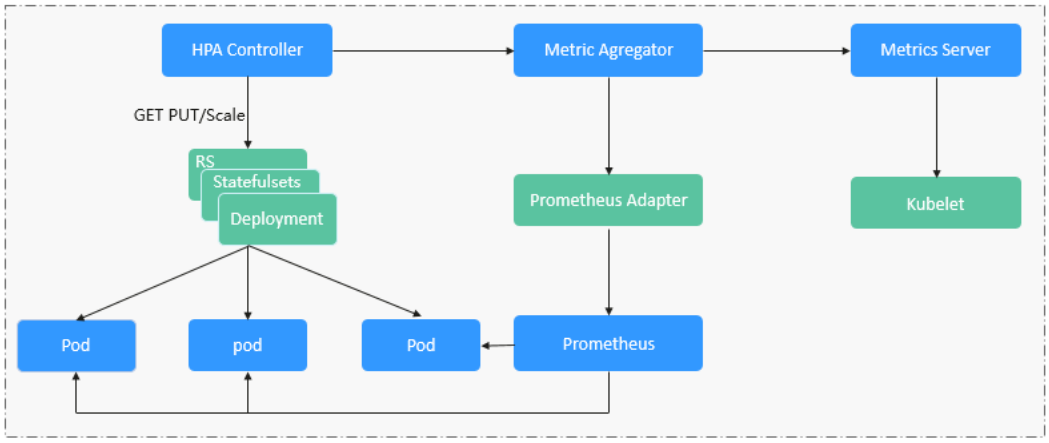
想要做到自动弹性伸缩，先决条件就是能感知到各种运行数据，例如集群节点、Pod、容器的 CPU、内存使用率等等。而这些数据的监控能力 Kubernetes 也没有自己实现，而是通过其他项目来扩展 Kubernetes 的能力，CCE 提供如下两个插件来实现该能力：

- Prometheus 是一套开源的系统监控报警框架，能够采集丰富的 Metrics（度量数据），目前已经基本是 Kubernetes 的标准监控方案。
- Metrics Server 是 Kubernetes 集群范围资源使用数据的聚合器。Metrics Server 从 kubelet 公开的 Summary API 中采集度量数据，能够收集包括了 Pod、Node、容器、Service 等主要 Kubernetes 核心资源的度量数据，且对外提供一套标准的 API。

使用 HPA（Horizontal Pod Autoscaler）配合 Metrics Server 可以实现基于 CPU 和内存的自动弹性伸缩，再配合 Prometheus 还可以实现自定义监控指标的自动弹性伸缩。

HPA 主要流程如图 14-1 所示。

图14-1 HPA 流程图



HPA 的核心有如下 2 个部分：

- 监控数据来源

  最早社区只提供基于 CPU 和 Mem 的 HPA，随着应用越来越多搬迁到 k8s 上以及 Prometheus 的发展，开发者已经不满足于 CPU 和 Memory，开发者需要应用自身的业务指标，或者是一些接入层的监控信息，例如：Load Balancer 的 QPS、网站的实时在线人数等。社区经过思考之后，定义了一套标准的 Metrics API，通过聚合 API 对外提供服务。

  - metrics.k8s.io： 主要提供 Pod 和 Node 的 CPU 和 Memory 相关的监控指标。
  - custom.metrics.k8s.io： 主要提供 Kubernetes Object 相关的自定义监控指标。
  - external.metrics.k8s.io：指标来源外部，与任何的 Kubernetes 资源的指标无关。

- 扩缩容决策算法

  HPA controller 跟据当前指标和期望指标来计算缩放比例，计算公式如下：

  desiredReplicas = ceil[currentReplicas * (currentMetricValue / desiredMetricValue)]

  例如当前的指标值是 200m，目标值是 100m，那么按照公式计算期望的实例数就会翻倍。那么在实际过程中，可能会遇到实例数值反复伸缩，导致集群震荡。为了保证稳定性，HPA controller 从以下几个方面进行优化：

  - 冷却时间：在 1.11 版本以及之前的版本，社区引入了 horizontal-pod-autoscaler-downscale-stabilization-window 和 horizontal-pod-autoScaler-upscale-stabilization-window 这两个启动参数代表扩容冷却时间和缩容冷却时间，这样保证在冷却时间内，跳过扩缩容。1.14 版本之后引入延迟队列，保存一段时间内每一次检测的决策建议，然后根据当前所有有效的决策建议来进行决策，从而保证期望的副本数尽量小的发生变更，保证稳定性。
  - 忍受度：可以看成一个缓冲区，当实例变化范围在忍受范围之内的话，保持原有的实例数不变。

    首先定义 ratio = currentMetricValue / desiredMetricValue

    当|ratio － 1.0| <= tolerance 时，则会忽略，跳过 scale。

    当|ratio － 1.0| > tolerance 时， 就会根据之前的公式计算期望值。

当前社区版本中默认值为 0.1

由于当前冷却时间窗，忍受度都是全局的参数，对于用户来说，可能应用所需的忍受度和冷却时间窗都不一致，所以当前 CCE 在社区的基础上，提供应用级别的忍受度和冷却时间窗。

HPA 是基于指标阈值进行伸缩的，常见的指标主要是 CPU、内存，当然也可以通过自定义指标，例如 QPS、连接数等进行伸缩。但是存在一个问题：基于指标的伸缩存在一定的时延，这个时延主要包含：采集时延(分钟级) + 判断时延(分钟级) + 伸缩时延(分钟级)。这个分钟级的时延，可能会导致应用 CPU 飚高，相应时间变慢。为了解决这个问题，CCE 提供了定时策略，对于一些有周期性变化的应用，提前扩容资源，而业务低谷时，定时回收资源。

14.2.2 创建工作负载弹性伸缩（HPA）

HPA 策略即 Horizontal Pod Autoscaling，是 Kubernetes 中实现 POD 水平自动伸缩的功能。该策略在 kubernetes 社区 HPA 功能的基础上，增加了应用级别的冷却时间窗和扩缩容阈值等功能。

前提条件

若使用系统指标，则需要安装 13.6 metrics-server 和 13.8 prometheus 插件：

- metrics-server：负责采集 kubernetes 集群中 kubelet 的公开指标项，包含 CPU 利用率、内存利用率。HPA 弹性策略如基于 CPU/MEM 利用率，必须安装此插件。
- prometheus：负责采集 kubernetes 集群中 kubelet 的公开指标项（CPU 利用率、内存利用率）。

约束与限制

- HPA 策略：仅支持 1.13 及以上版本的集群创建。
- 每个工作负载只能创建一个策略，即如果您创建了一个 HPA 策略，则不能再对其创建 14.2.3 创建工作负载弹性伸缩（CustomedHPA）或其他 HPA 策略，您可以删除该 HPA 策略后再创建。

操作步骤

步骤 1 在 CCE 控制台中，单击左侧导航栏的“弹性伸缩”，在“工作负载伸缩”页签下，单击“创建 HPA 策略”。

步骤 2 进入创建工作负载 HPA 策略页面，在“插件检测”步骤中：

- 若插件名称后方显示

，请单击插件后方的“现在安装”，根据业务需求配置插件参数后单击“立即安装”，等待插件安装完成。
- 若插件名称后方显示

，则说明插件已安装成功。

步骤 3 确认插件已安装成功后，单击“下一步：策略配置”。

说明

如果插件已提前安装成功，单击“创建 HPA 策略”后，在“插件检测”步骤中经过短暂检测后将直接进入“策略配置”步骤。

步骤 4 在“策略配置”步骤中，参照表 14-3 设置策略参数。

表14-3 HPA 策略参数配置

参数	参数说明
策略名称	新建策略的名称，请自定义。
集群名称	请选择工作负载所在的集群。
命名空间	请选择工作负载所在的命名空间。
关联工作负载	请选择要设置 HPA 策略的工作负载。
实例范围	请输入最小实例数和最大实例数。 策略触发时，工作负载实例将在此范围内伸缩。
冷却时间	请输入缩容和扩容的冷却时间，单位为分钟，**缩容扩容冷却时间不能小于 1 分钟**。 **该设置仅在 1.15 及以上版本的集群中显示，1.13 版本的集群不支持该设置。** 策略成功触发后，在此缩容/扩容冷却时间内，不会再次触发缩容/扩容，目的是等待伸缩动作完成后在系统稳定且集群正常的情况下进行下一次策略匹配。
策略规则	策略规则可基于系统指标。 **系统指标** • 指标：可选择“CPU 利用率”或“内存利用率”。 说明 利用率 = 工作负载容器组（Pod）的实际使用量 / 申请量。 • 期望值：请输入期望资源平均利用率。 期望值表示所选指标的期望值，通过向上取整(当前指标值 / 期望值 × 当前实例数)来计算需要伸缩的实例数。 • 阈值：请输入缩容和扩容阈值。 当指标值大于缩容阈值且小于扩容阈值时，不会触发扩容或缩容。**阈值仅在 1.15 及以上版本的集群中支持**。 您可以单击“添加策略规则”，设置多条伸缩策略。 说明 HPA 在计算扩容、缩容实例数时，会选择最近 5 分钟内的最大值。

步骤 5 设置完成后，单击“创建”，在“完成”步骤中若显示“创建工作负载策略***提交成功”，可单击“返回工作负载伸缩策略”。

步骤 6 在“工作负载伸缩”页签下，可以看到刚刚创建的 HPA 策略。

----结束

14.2.3 创建工作负载弹性伸缩（CustomedHPA）

CustomedHPA 策略是自研的弹性伸缩增强能力，能够基于指标（CPU 利用率、内存利用率）或周期（每天、每周、每月或每年的具体时间点），对无状态工作负载进行弹性扩缩容。

主要功能如下：

- 支持按照当前实例数的百分比进行扩缩容。
- 支持设置一次扩缩容的最小步长。
- 支持按照实际指标值执行不同的扩缩容动作。

前提条件

若使用 CustomedHPA 策略，则必须安装 13.7 cce-hpa-controller 和 13.8 prometheus 插件，且两个插件版本均需为 1.1.0 及以上：

- cce-hpa-controller：支持按照当前实例数的百分比进行扩缩容；支持设置一次扩缩容的最小步长；支持按照实际指标值执行不同的扩缩容动作。创建 CustomedHPA 策略必须安装此插件。
- prometheus：负责采集 kubernetes 集群中 kubelet 的公开指标项（CPU 利用率、内存利用率）。创建 CustomedHPA 策略必须安装此插件。

约束与限制

- CustomedHPA 策略：仅支持 1.15 及以上版本的集群创建。
- CustomedHPA 策略不支持鲲鹏集群。
- 每个工作负载只能创建一个策略，即如果您创建了一个 14.2.2 创建工作负载弹性伸缩（HPA），则不能再对其创建 CustomedHPA 策略或其他 HPA 策略，您可以删除该 HPA 策略后再创建。

操作步骤

步骤 1 在 CCE 控制台中，单击左侧导航栏的“弹性伸缩”，在“工作负载伸缩”页签下，单击“创建 CustomedHPA 策略”按钮。

步骤 2 进入创建工作负载 CustomedHPA 策略页面，在“插件检测”步骤中：

- 若插件名称后方显示 ，请单击插件后方的“现在安装”按钮，根据业务需求配置插件参数后单击“立即安装”，等待插件安装完成。
- 若插件名称后方显示 ，则说明插件已安装成功。

步骤 3 确认插件已安装成功后，单击“下一步：策略配置”。

说明

如果插件已提前安装成功，单击“创建 CustomedHPA 策略”按钮后，在“插件检测”步骤中经过短暂检测后将直接进入“策略配置”步骤。

步骤 4 在“策略配置”步骤中，参照表 14-4 设置策略参数。

表14-4 CustomedHPA 策略参数配置

参数	参数说明
策略名称	新建策略的名称，请自定义。
集群名称	请选择工作负载所在的集群。
命名空间	请选择工作负载所在的命名空间。
关联工作负载	请选择要设置 CustomedHPA 策略的工作负载。
实例范围	请输入最小实例数和最大实例数。 策略触发时，工作负载实例将在此范围内伸缩。
冷却时间	请输入冷却时间值，单位为分钟。 策略成功触发后，在此冷却时间内，不会再次触发缩容/扩容，目的是等待伸缩动作完成后在系统稳定且集群正常的情况下进行下一次策略匹配。
策略规则	单击“添加策略规则”在弹出的窗口中设置伸缩策略参数： • 规则名称：请输入规则名称，可自定义。 • 类型：可选择“指标触发”或“周期触发”。 **指标触发** • 触发条件：请选择“CPU 利用率”或“内存利用率”，选择“>”或“<”，并输入百分比的值。 • 执行动作：与上述“触发条件”相对应，达到触发条件值后所要执行的动作，可添加多个执行动作。 • 是否启用：可单击启用或关闭该策略规则。 **周期触发** • 触发时间：可选择每天、每周、每月或每年的具体时间点。 • 执行动作：与上述“触发时间”相对应，达到触发时间值后所要执行的动作。 • 是否启用：可单击启用或关闭该策略规则。 单击确定后，您可以在“策略规则”列表中查看添加的规则，并可执行开启关闭、编辑、删除等操作。 单击“策略规则”列表下方的“添加策略规则”，可设置多条策略。

步骤 5 设置完成后，单击“创建”，在“完成”步骤中若显示“创建工作负载策略***提交成功”，可单击“返回工作负载伸缩策略”。

步骤 6 在“工作负载伸缩”页签下，可以看到刚刚创建的 CustomedHPA 策略。

----结束

14.2.4 管理工作负载伸缩策略

操作场景

HPA 或 CustomedHPA 策略创建完成后，可对创建的策略进行更新、克隆、编辑 YAML 以及删除等操作。

查看 HPA 策略

您可以查看 HPA 策略的规则、状态和事件，参照界面中的报错提示有针对性的解决异常事件。

步骤 1 登录 CCE 控制台，在左侧导航栏中单击“弹性伸缩”，在“工作负载伸缩”页签下，单击要查看的 HPA 策略前方的 。

步骤 2 在展开的区域中，可以看到规则、状态和事件页签，若策略异常，请参照界面中的报错提示进行定位处理。

说明

您还可以在工作负载详情页中查看已创建的 HPA 策略，登录 CCE 控制台，在左侧导航栏中单击“工作负载 > 无状态负载 Deployment”，单击工作负载后方“操作”中的“更多 > 伸缩”，在该工作负载详情页的“伸缩”页签下可以看到“弹性伸缩 - HPA / CustomedHPA”，您在“弹性伸缩”页面配置的 HPA 策略也会在这里显示。

表14-5 事件类型及名称

事件类型	事件名称	描述
正常	SuccessfulRescale	扩缩容成功
异常	InvalidTargetRange	无效的 TargetRange
	InvalidSelector	无效选择器
	FailedGetObjectMetric	获取对象失败数
	FailedGetPodsMetric	获取 Pod 列表失败指标
	FailedGetResourceMetric	获取资源失败数
	FailedGetExternalMetric	获取外部指标失败
	InvalidMetricSourceType	无效的指标来源类型
	FailedConvertHPA	转换 HPA 失败

事件类型	事件名称	描述
	FailedGetScale	获取比例尺失败
	FailedComputeMetricsReplicas	计算指标副本数失败
	FailedGetScaleWindow	获取 ScaleWindow 失败
	FailedRescale	扩缩容失败

----结束

查看 CustomedHPA 策略

您可以查看 CustomedHPA 策略的规则和最新状态，参照界面中的报错提示有针对性的解决异常事件。

步骤 1 登录 CCE 控制台，在左侧导航栏中单击“弹性伸缩”，在“工作负载伸缩”页签下，单击要查看的 CustomedHPA 策略前方的。

步骤 2 在展开的区域中，可以看到规则页签，若策略异常，请单击“最新状态”栏中的“详情”，参照展开的详细信息进行定位处理。

说明

您还可以在工作负载详情页中查看已创建的 CustomedHPA 策略，登录 CCE 控制台，在左侧导航栏中单击“工作负载 > 无状态负载 Deployment”，单击工作负载后方“操作”中的“更多 > 伸缩”，在该工作负载详情页的“伸缩”页签下可以看到“弹性伸缩 - HPA / CustomedHPA”，您在“弹性伸缩”页面配置的 CustomedHPA 策略也会在这里显示。

----结束

更新 HPA 或 CustomedHPA 策略

以 HPA 策略为例。

步骤 1 登录 CCE 控制台，在左侧导航栏中单击“弹性伸缩”，在“工作负载伸缩”页签下，单击策略后方“操作”栏中的“更新”。

步骤 2 在打开的“更新工作负载 HPA 策略”页面中，更新策略参数。

步骤 3 单击“更新”完成策略更新。

----结束

克隆 HPA 或 CustomedHPA 策略

步骤 1 登录 CCE 控制台，在左侧导航栏中单击“弹性伸缩”，在“工作负载伸缩”页签下，单击策略后方“操作”栏中的“克隆”。

步骤 2 在打开的“创建工作负载 HPA 策略”页面中，可以看到部分参数（如实例范围、冷却时间和策略规则）已经克隆过来，请按照业务需求补充或修改其他策略参数。

步骤 3 单击“创建”完成策略克隆，在“工作负载伸缩”页签下的策略列表中可以看到新克隆的策略。

----结束

编辑 YAML（HPA 策略）

步骤 1 登录 CCE 控制台，在左侧导航栏中单击“弹性伸缩”，在“工作负载伸缩”页签下，单击 HPA 策略后方“操作”栏中的“更多 > 编辑 YAML”。

步骤 2 在弹出的“编辑 YAML”窗口中，可以对 YAML 进行修改和下载。

步骤 3 在右上角关闭窗口完成操作。

----结束

查看 YAML（CustomedHPA 策略）

步骤 1 登录 CCE 控制台，在左侧导航栏中单击“弹性伸缩”，在“工作负载伸缩”页签下，单击 CustomedHPA 策略后方“操作”栏中的“更多 > 查看 YAML”。

步骤 2 在弹出的“查看 YAML”窗口中，可以对 YAML 进行复制和下载，不能对其修改。

步骤 3 在右上角关闭窗口完成操作。

----结束

删除 HPA 或 CustomedHPA 策略

步骤 1 登录 CCE 控制台，在左侧导航栏中单击“弹性伸缩”，在“工作负载伸缩”页签下，单击策略后方“操作”栏中的“更多 > 删除”。

步骤 2 在弹出的“删除 HPA 策略”窗口中，确认是否删除。

步骤 3 单击“是”完成删除操作。

----结束

14.3 集群/节点弹性伸缩

14.3.1 节点伸缩原理

HPA 是针对 Pod 级别的，但是如果集群的资源不够了，那就只能对节点进行扩容了。集群节点的弹性伸缩本来是一件非常麻烦的事情，但是好在现在的集群大多都是构建在云上，云上可以直接调用接口添加删除节点，这就使得集群节点弹性伸缩变得非常方便。

Cluster Autoscaler 是 Kubernetes 提供的集群节点弹性伸缩组件，根据 Pod 调度状态及资源使用情况对集群的节点进行自动扩容缩容。

CCE 的集群节点弹性伸缩请参见 14.3.2 创建节点伸缩策略。

Cluster Autoscaler 工作原理

Cluster Autoscaler（简称 CA）的主要流程包括两部分：

- ScaleUp 流程： CA 会每隔 15s 检查一次所有不可调度的 Pod，根据用户设置的策略，选择出一个符合要求的节点组进行扩容。
- ScaleDown 流程：CA 每隔 10s 会扫描一次所有的 Node，如果该 Node 上所有的 Pod Requests 少于用户定义的缩容百分比时，CA 会模拟将该节点上的 Pod 是否能迁移到其他节点，如果可以的话，当满足不被需要的时间窗以后，该节点就会被移除。

如上所述，当集群节点处于一段时间空闲状态时（默认 10min），会触发集群缩容操作。

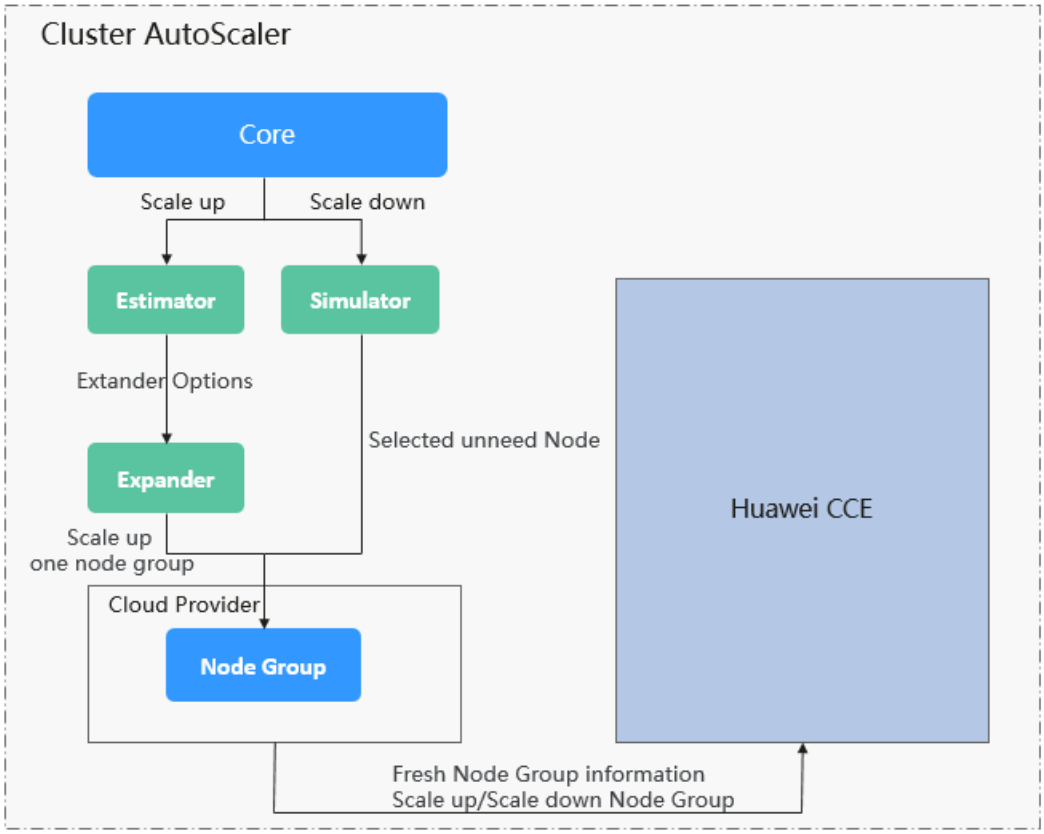
当节点存在以下几种状态的 pod 时，不可缩容：

1. 节点上的 pod 有设置 PodDisruptionBudget，当移除 pod 不满足对应条件时，节点不会缩容。
2. 节点上的 pod 设置本地存储时，节点不会缩容。
3. 节点上的 pod 由于一些限制，如亲和、反亲和或者 taints 等，无法调度到其他节点，节点不会缩容。
4. 节点上的 pod 拥有"cluster-autoscaler.kubernetes.io/safe-to-evict": "false"这个 annotations 时，节点不缩容。
5. 节点上存在 kube-system namespace 下的 Pod（除 kube-system daemonset 创建的 Pod），节点不缩容。
6. 节点上有非 controller（deployment/replica set/job/stateful set）创建的 Pod，节点不缩容。

Cluster AutoScaler 架构

Cluster AutoScaler 架构如图 14-2 所示，主要由以下几个核心模块组成：

图14-2 Cluster AutoScaler 架构图



说明如下：

- Estimator： 负责扩容场景下，评估满足当前不可调度 Pod 时，每个节点池需要扩容的节点数量。
- Simulator： 负责缩容场景下，找到满足缩容条件的节点。
- Expander： 负责在扩容场景下，根据用户设置的不同的策略来，从 Estimator 选出的节点池中，选出一个最佳的选择。当前 Expander 有多种策略：
  - Random： 随机选择一个节点池，如果用户没有设置的话，默认是 Random。
  - most-Pods： 选择此次扩容后能满足调度最多 Pod 的节点池，如果有相同的，再随机选择一个。
  - least-waste： 选择此次扩容完成后，具有最小浪费的 CPU 或者 Mem 资源的节点池。
  - price： 选择此次扩容所需节点金额最小的节点池。
  - priority： 根据用户自定义的权重，选择权重最高的节点池。

当前 CCE 提供除 Price 以外的所有的策略，当前 CCE 插件默认使用 least-waste 策略。

14.3.2 创建节点伸缩策略

CCE 的自动伸缩能力是通过节点自动伸缩组件 autoscaler 实现的，可以按需弹出节点实例，支持多可用区、多实例规格、多种伸缩模式，满足不同的节点伸缩场景。

前提条件

使用节点伸缩功能前，需要安装 13.5 autoscaler 插件，插件版本要求 1.13.8 及以上。

缩容说明

节点伸缩策略中不能直接设置缩容策略，你可以在安装 13.5 autoscaler 时进行设置。

节点缩容仅支持资源分配率缩容机制，当集群下 CPU 和内存分配率低于您设置的门限（在安装/编辑 autoscaler 插件时设置）时将对节点池下节点启动缩容（该功能可以关闭）。

操作步骤

步骤 1 在 CCE 控制台中，单击左侧导航栏的“弹性伸缩”，在“节点伸缩”页签下，单击“创建节点伸缩策略”按钮。

步骤 2 进入创建节点伸缩策略页面，在“插件检测”步骤中：

- 若插件名称后方显示 ，请单击插件后方的“现在安装”按钮，根据业务需求配置插件参数后单击“立即安装”，等待插件安装完成。
- 若插件名称后方显示 ，则说明插件已安装成功。

步骤 3 确认插件已安装成功后，单击“下一步：策略配置”。

说明

如果插件已提前安装成功，单击“创建节点伸缩策略”按钮后，在“插件检测”步骤中经过短暂检测后将直接进入“策略配置”步骤。

步骤 4 在打开的“创建节点伸缩策略”页面中，参照表 14-6 设置策略参数。

表14-6 节点伸缩策略参数配置

参数	参数说明
策略名称	新建策略的名称，请自定义。
关联节点池	请单击“添加节点池”后选择要操作的节点池。您可以关联多个节点池，以使用相同的伸缩策略。
执行规则	单击“添加规则”，在弹出的添加规则窗口中设置如下参数： **规则名称：**请输入规则名称，可自定义。 **规则类型：**可选择“指标触发”或“周期触发”，两种类型区别如下：

参数	参数说明
	• **指标触发：** – 触发条件：请选择“CPU 分配率”或“内存分配率”，输入百分比的值。该百分比应大于 autoscaler 插件中配置的缩容百分比。 **说明** **分配率 = 节点池容器组（Pod）资源申请量 / 节点池 Pod 可用资源量 (Node Allocatable) 。** – 执行动作：与上述“触发条件”相对应，达到触发条件值后所要执行的动作。 • **周期触发：** – 触发时间：可选择每天、每周、每月或每年的具体时间点。 – 执行动作：与上述“触发时间”相对应，达到触发时间值后所要执行的动作。 您可以单击“添加规则”，设置多条节点伸缩策略。您最多可以添加 1 条 CPU 使用率指标规则、1 条内存使用率指标规则，且规则总数小于等于 10 条。

步骤 5 设置完成后，单击“创建”，在“完成”步骤中若显示“创建节点伸缩策略***任务提交成功”，可单击“返回节点策略”。

步骤 6 在“节点伸缩”页签下，可以看到刚创建的节点伸缩策略。

----结束

Yaml 样例

节点伸缩策略 Yaml 样例如下：

```
apiVersion: autoscaling.cce.io/v1alpha1
kind: HorizontalNodeAutoscaler
metadata:
  creationTimestamp: "2020-02-13T12:47:49Z"
  generation: 1
  name: xxxx
  namespace: kube-system
  resourceVersion: "11433270"
  selfLink: /apis/autoscaling.cce.io/v1alpha1/namespaces/kube-
system/horizontalnodeautoscalers/xxxx
  uid: c2bd1e1d-60aa-47b5-938c-6bf3fadbe91f
spec:
  disable: false
  rules:
  - action:
      type: ScaleUp
      unit: Node
      value: 1
    cronTrigger:
```

```
    schedule: 47 20 * * *
  disable: false
  ruleName: cronrule
  type: Cron
- action:
    type: ScaleUp
    unit: Node
    value: 2
  disable: false
  metricTrigger:
    metricName: Cpu
    metricOperation: '>'
    metricValue: "40"
    unit: Percent
  ruleName: metricrule
  type: Metric
targetNodepoolIds:
- 7d48eca7-3419-11ea-bc29-0255ac1001a8
```

表14-7 关键参数说明

参数	参数类型	描述
spec.disable	Bool	伸缩策略开关，会对策略中的所有规则生效
spec.rules	Array	伸缩策略中的所有规则
spec.rules[x].ruleName	String	规则名称
spec.rules[x].type	String	规则类型，当前支持“Cron”和“Metric”两种类型
spec.rules[x].disable	Bool	规则开关，当前仅支持“false”
spec.rules[x].action.type	String	规则操作类型，当前仅支持“ScaleUp”
spec.rules[x].action.unit	String	规则操作单位，当前仅支持“Node”
spec.rules[x].action.value	Integer	规则操作数值
spec.rules[x].cronTrigger	/	可选，仅在周期规则中有效
spec.rules[x].cronTrigger.schedule	String	周期规则的 cron 表达式
spec.rules[x].metricTrigger	/	可选，仅在指标规则中有效
spec.rules[x].metricTrigger.metricName	String	指标规则对应的指标，当前支持“Cpu”和“Memory”两种类型
spec.rules[x].metricTrigger.metricOperation	String	指标规则的比较符，当前仅支持“>”

参数	参数类型	描述
spec.rules[x].metricTrigger.metricValue	String	指标规则的阈值，支持 1-100 之间的所有整数，需以字符串表示
spec.rules[x].metricTrigger.Unit	String	指标规则阈值的单位，当前仅支持“%”
spec.targetNodepoolIds	Array	伸缩策略关联的所有节点池
spec.targetNodepoolIds[x]	String	伸缩策略关联节点池的 uid

14.3.3 管理节点伸缩策略

操作场景

节点伸缩策略创建完成后，可对创建的策略进行删除、编辑、停用、启用、克隆等操作。

查看节点伸缩策略

您可以查看节点伸缩策略的关联节点池、执行规则和伸缩历史，参照界面中的提示有针对性的解决异常问题。

步骤 1 登录 CCE 控制台，在左侧导航栏中单击“弹性伸缩”，在“节点伸缩”页签下，单击要查看的策略前方的。

步骤 2 在展开的区域中，可以看到该策略的关联节点池、执行规则和伸缩历史页签，若策略异常，请参照界面中的报错提示进行定位处理。

说明

您还可以在节点池管理中关闭或开启弹性扩缩容，登录 CCE 控制台，在左侧导航栏中单击“资源管理 > 节点池管理”，单击要操作的节点池右上角的“编辑”，在弹出的“编辑节点池”窗口中的可以看到“弹性扩缩容”按钮，并可设置节点数上下限和弹性缩容冷却时间。

----结束

删除节点伸缩策略

步骤 1 登录 CCE 控制台，在左侧导航栏中单击“弹性伸缩”，在“节点伸缩”页签下，单击要删除的策略后方“操作”栏中的“删除”。

步骤 2 在弹出的“删除节点策略”窗口中，确认是否删除。

步骤 3 确认后，在输入框中输入 **DELETE**。

步骤 4 单击“确认”按钮即完成删除操作。

----结束

编辑节点伸缩策略

步骤 1 登录 CCE 控制台，在左侧导航栏中单击“弹性伸缩”，在“节点伸缩”页签下，单击要更新的策略后方“操作”栏中的“编辑”。

步骤 2 在打开的“编辑节点伸缩策略”页面中，更新策略参数。

步骤 3 完成设置后，单击“确定”按钮完成编辑操作。

----结束

克隆节点伸缩策略

步骤 1 登录 CCE 控制台，在左侧导航栏中单击“弹性伸缩”，在“节点伸缩”页签下，单击要克隆的策略后方“操作”栏中的“更多 > 克隆”。

步骤 2 在打开的“创建节点伸缩策略”页面中，可以看到部分参数已经克隆过来，请按照业务需求补充或修改其他策略参数。

步骤 3 单击“立即创建”按钮完成策略克隆，在“节点伸缩”页签下的策略列表中可以看到新克隆的策略。

----结束

停用/启用节点伸缩策略

步骤 1 登录 CCE 控制台，在左侧导航栏中单击“弹性伸缩”，在“工作负载伸缩”页签下，单击策略后方“操作”栏中的“更多 > 停用”，若策略为停用状态时，则单击“更多 > 启用”。

步骤 2 在弹出的“停用节点策略”或“启用节点策略”窗口中，确认是否进行停用或启用操作。

步骤 3 单击“确定”完成操作，在节点伸缩的列表中可以看到该策略的状态。

----结束

14.3.4 节点伸缩常见问题

如果多条规则同时满足条件，会如何执行？

该问题有如下两种情况：

- 如果同时配置了“CPU 分配率”和“内存分配率”的规则，两种或多种规则同时满足扩容条件时，执行扩容节点数更多的规则。
- 如果同时配置了“CPU 分配率”和“周期触发”的规则，当达到“周期触发”的时间值时 CPU 也满足扩容条件时，较早执行的 A 规则会将节点池状态置为伸缩中状态，导致 B 规则无法正常执行。待 A 规则执行完毕，节点池状态恢复正常后，B 规则也不会执行。

配置了 CPU 或者内存规则后，策略的检测周期是多久？是否只要一次检测出满足条件就会触发扩容节点？

周期不固定，会因 autoscaler 每次循环的处理逻辑而变动。只要一次检测出满足条件就会触发扩容（当然还要满足冷却时间、节点池状态等约束条件）。

节点伸缩和 autoscaler 插件的本身功能如何配合？

节点伸缩中创建的策略和 autoscaler 插件中的配置同时生效时，比如不可调度和指标规则同时满足时，优先执行不可调度扩容。

- 若不可调度执行成功，跳过指标规则逻辑，进入下一次循环。
- 若不可调度执行失败，才执行指标规则逻辑。

节点池弹性扩容时，CCE 是按什么策略扩容的？

节点池新增了优先级等功能，弹性扩容时 CCE 将按照如下策略来选择节点池进行扩容：

1. 通过预判算法判断节点池是否能满足让 Pending 的 Pod 正常调度的条件，包括节点资源大于 Pod 的 request 值、nodeSelect、nodeAffinity 和 taints 等是否满足 Pod 正常调度的条件；另外还会过滤掉扩容失败（因为资源不足等原因）还处于 15min 冷却时间的节点池。
2. 有多个节点池满足条件时，判断节点池设置的优先级（优先级默认值为 0，取值范围为 0-100，其中 100 为最高，0 为最低），选择优先级最高的节点池扩容。
3. 如果有多个节点池处于相同的优先级，或者都没有配置优先级时，通过最小浪费原则，根据节点池里设置的虚拟机规格，计算刚好能满足 Pending 的 Pod 正常调度，且浪费资源最少的节点池。
4. 如果还是有多个节点池的虚拟机规格都一样，只是 AZ 不同，那么会随机选择其中一个节点池触发扩容。

节点池中配置的缩容冷却时间和 autoscaler 插件中配置的缩容冷却时间是如何影响的？

节点池配置的缩容冷却时间

弹性缩容冷却时间：当前节点池扩容出的节点多长时间不能被缩容，作用范围为节点池级别。

autoscaler 插件配置的缩容冷却时间

扩容后缩容冷却时间：autoscaler 触发扩容后（不可调度、指标、周期策略）整个集群多长时间内不能被缩容，作用范围为集群级别。

节点删除后缩容冷却时间：autoscaler 触发缩容后整个集群多长时间内不能继续缩容，作用范围为集群级别。

缩容失败后缩容冷却时间：autoscaler 触发缩容失败后整个集群多长时间内不能继续缩容，作用范围为集群级别。

如果出现优先选择的节点池资源不足，会不会自动选择下一个节点池？

会。

15 权限管理

15.1 CCE 权限概述

CCE 权限管理是在统一身份认证服务（IAM）与 Kubernetes 的角色访问控制（RBAC）的能力基础上，打造的细粒度权限管理功能，支持基于 IAM 的细粒度权限控制和 IAM Token 认证，支持集群级别、命名空间级别的权限控制，帮助用户便捷灵活的对租户下的 IAM 用户、用户组设定不同的操作权限。

如果您需要对上购买的 CCE 集群及相关资源进行精细的权限管理，例如限制不同部门的员工拥有部门内资源的细粒度权限，您可以使用 CCE 权限管理提供的增强能力进行多维度的权限管理。

本章节将介绍 CCE 权限管理机制及其涉及到的基本概念。如果当前帐号已经能满足您的要求，您可以跳过本章节，不影响您使用 CCE 服务的其它功能。

CCE 支持的权限管理能力

CCE 的权限管理包括“集群权限”和“命名空间权限”两种能力，能够从集群和命名空间层面对用户组或用户进行细粒度授权，具体解释如下：

- **15.2** 集群权限：是基于 IAM 系统策略的授权，可以通过用户组功能实现 IAM 用户的授权。用户组是用户的集合，通过集群权限设置可以让某些用户组操作集群（如创建/删除集群、节点、节点池、模板、插件等），而让某些用户组仅能查看集群。

  集群权限涉及 CCE 非 Kubernetes API，支持 IAM 细粒度策略、企业项目管理相关能力。

- **15.3** 命名空间权限：是基于 Kubernetes RBAC 能力的授权，通过权限设置可以让不同的用户或用户组拥有操作不同 Kubernetes 资源的权限。同时 CCE 基于开源能力进行了增强，可以支持基于 IAM 用户或用户组粒度进行 RBAC 授权、IAM token 直接访问 API 进行 RBAC 认证鉴权。

  命名空间权限涉及 CCE Kubernetes API，基于 Kubernetes RBAC 能力进行增强，支持对接 IAM 用户/用户组进行授权和认证鉴权，但与 IAM 细粒度策略独立。

⚠ 注意

“集群权限”仅针对与集群相关的资源（如创建/删除集群、节点、节点池、模板、插件等）有效，您必须确保同时配置了“命名空间权限”，才能有操作 Kubernetes 资源（如工作负载、Service 等）的权限。

统一身份认证服务（IAM）

统一身份认证（Identity and Access Management，简称 IAM）是提供权限管理的基础服务，可以帮助您安全地控制服务和资源的访问权限。

以下为 IAM 的优势：

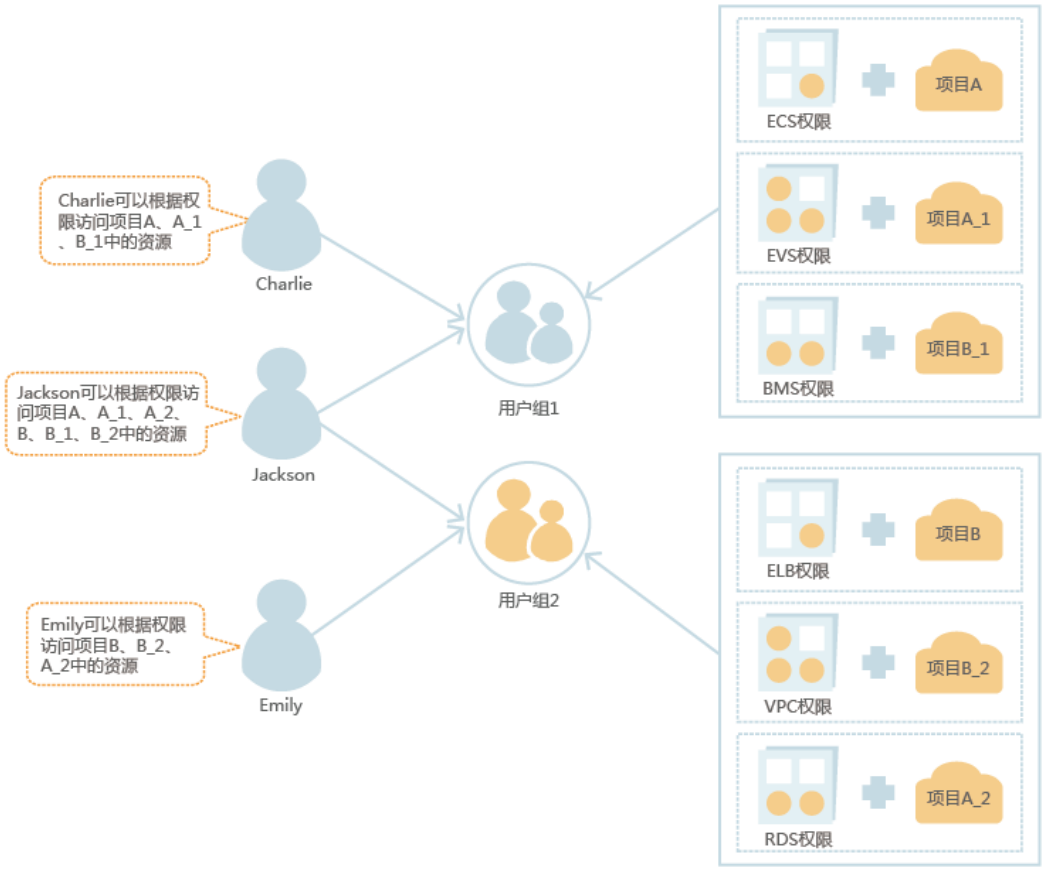
对的资源进行精细访问控制

您注册后，系统自动创建帐号，帐号是资源的归属以及使用计费的主体，对其所拥有的资源具有完全控制权限，可以访问所有的云服务。

如果您在购买了多种资源，例如弹性云主机、云硬盘、裸金属服务器等，您的团队或应用程序需要使用您在中的资源，您可以使用 IAM 的用户管理功能，给员工或应用程序创建 IAM 用户，并授予 IAM 用户刚好能完成工作所需的权限，新创建的 IAM 用户可以使用自己单独的用户名和密码登录。IAM 用户的作用是多用户协同操作同一帐号时，避免分享帐号的密码。

除了 IAM 外，还有企业管理服务同样可以进行资源权限管理，相对于 IAM，企业管理对资源的控制粒度更为精细，同时还支持企业项目费用的管理，建议结合企业需求选

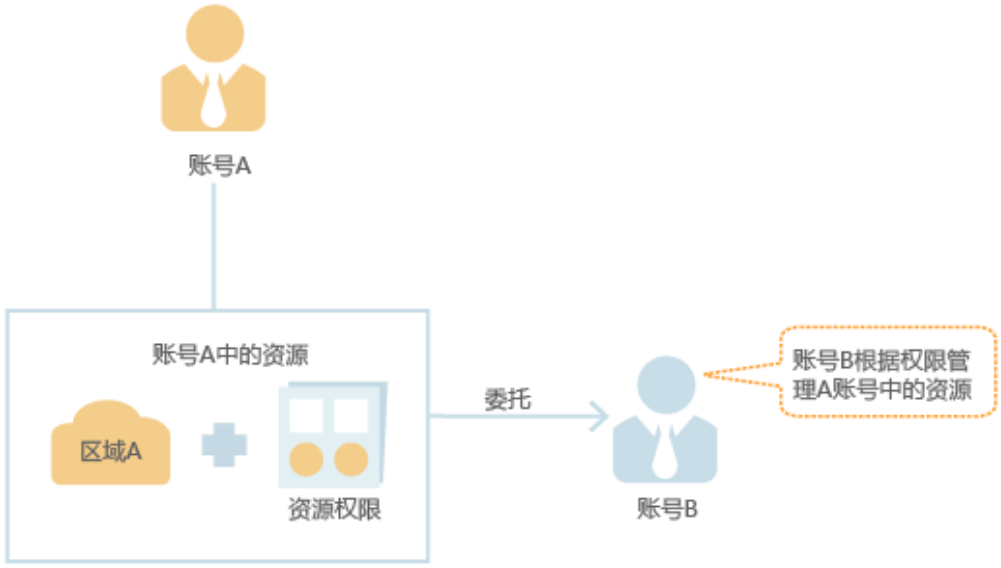
择 IAM 或是企业管理进行资源权限管理。



跨帐号的资源操作与授权

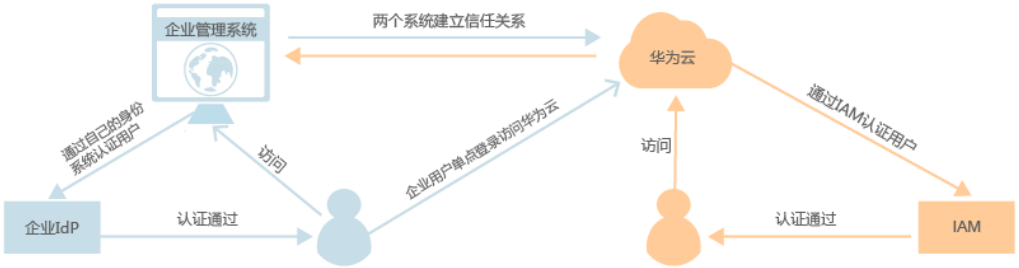
如果您在购买了多种资源，其中一种资源希望由其它帐号管理，您可以使用 IAM 提供的委托功能。

例如您在上购买的部分资源，希望委托给一家专业的代运维公司来运维，通过 IAM 的委托功能，代运维公司可以使用自己的帐号对您委托的资源进行运维。当委托关系发生变化时，您可以随时修改或撤消对代运维公司的授权。下图中帐号 A 即为委托方，帐号 B 为被委托方。



使用企业已有帐号登录

当您希望本企业员工可以使用企业内部的认证系统登录，而不需要在中重新创建对应的 IAM 用户，您可以使用 IAM 的身份提供商功能，建立您所在企业与的信任关系，通过联合认证使员工使用企业已有帐号直接登录，实现单点登录。



Kubernetes 的角色访问控制（RBAC）

Kubernetes 基于角色的访问控制（Role-Based Access Control，即”RBAC”）使用”rbac.authorization.k8s.io” API Group 实现授权决策，允许管理员通过 Kubernetes API 动态配置策略。详情请参见 15.3 命名空间权限。

15.2 集群权限

CCE 集群权限是基于 IAM **系统策略**和**自定义策略**的授权，可以通过用户组功能实现 IAM 用户的授权，如何授权请参见 15.4.1 设置集群权限。

> **⚠ 注意**
>
> 集群权限仅针对与集群相关的资源（如集群、节点等）有效，您必须确保同时配置了15.4.2 设置命名空间权限，才能有操作 Kubernetes 资源（如工作负载、Service 等）的权限。

系统策略

IAM 中预置的 CCE 系统策略当前包含 **CCE FullAccess** 和 **CCE ReadOnlyAccess** 两种策略，分别支持的细粒度操作（Action）权限如下：

表15-1 CCE FullAccess 策略主要权限

操作（Action）	Action 详情	说明
cce:*:*	cce:cluster:create	创建集群
	cce:cluster:delete	删除集群
	cce:cluster:update	更新集群，如后续允许集群支持 RBAC，调度参数更新等
	cce:cluster:upgrade	升级集群
	cce:cluster:start	唤醒集群
	cce:cluster:stop	休眠集群
	cce:cluster:list	查询集群列表
	cce:cluster:get	查询集群详情
	cce:node:create	添加节点
	cce:node:delete	删除节点/批量删除节点
	cce:node:update	更新节点，如更新节点名称
	cce:node:get	查询节点详情
	cce:node:list	查询节点列表
	cce:nodepool:create	创建节点池
	cce:nodepool:delete	删除节点池
	cce:nodepool:update	更新节点池信息
	cce:nodepool:get	获取节点池
	cce:nodepool:list	列出集群的所有节点池
	cce:release:create	创建模板实例

操作（Action）	Action 详情	说明
	cce:release:delete	删除模板实例
	cce:release:update	更新升级模板实例
	cce:job:list	查询任务列表（集群层面的 job）
	cce:job:delete	删除任务/批量删除任务（集群层面的 job）
	cce:job:get	查询任务详情（集群层面的 job）
	cce:storage:create	创建存储
	cce:storage:delete	删除存储
	cce:storage:list	列出所有磁盘
	cce:addonInstance:create	创建插件实例
	cce:addonInstance:delete	删除插件实例
	cce:addonInstance:update	更新升级插件实例
	cce:addonInstance:get	获取插件实例
	cce:addonTemplate:get	获取插件模板
	cce:addonInstance:list	列出所有插件实例
	cce:addonTemplate:list	列出所有插件模板
	cce:chart:list	列出所有模板
	cce:chart:delete	删除摸板
	cce:chart:update	更新模板
	cce:chart:upload	上传模板
	cce:chart:get	获取模板信息
	cce:release:get	获取模板实例信息
	cce:release:list	列出所有模板实例
	cce:userAuthorization:get	获取 CCE 用户授权
	cce:userAuthorization:create	创建 CCE 用户授权
ecs:*:*	-	ECS（弹性云主机）服务的所有权限。

操作（Action）	Action 详情	说明
evs:*:*	-	EVS（云硬盘）的所有权限。 可以将云硬盘挂载到云服务器，并可以随时扩容云硬盘容量
vpc:*:*	-	VPC（虚拟私有云，包含二代 ELB）的所有权限。 创建的集群需要运行在虚拟私有云中，创建命名空间时，需要创建或关联 VPC，创建在命名空间的容器都运行在 VPC 之内。
sfs:*:get*	-	SFS（弹性文件存储服务）资源详情的查看权限。
sfs:shares:ShareAction	-	SFS（弹性文件存储服务）资源的扩容共享。
aom:*:get	-	AOM（应用运维管理）资源详情的查看权限。
aom:*:list	-	AOM（应用运维管理）资源列表的查看权限。
aom:autoScalingRule:*	-	AOM（应用运维管理）自动扩缩容规则的所有操作权限。
apm:icmgr:*	-	APM（应用性能管理服务）操作 ICAgent 权限。

表15-2 CCE ReadOnlyAccess 策略主要权限

操作（Action）	操作（Action）	说明
cce:*:get	cce:cluster:get	查询集群详情
	cce:node:get	查询节点详情
	cce:job:get	查询任务详情（集群层面的 job）
	cce:addonInstance:get	获取插件实例
	cce:addonTemplate:get	获取插件模板
	cce:chart:get	获取模板信息
	cce:nodepool:get	获取节点池
	cce:release:get	获取模板实例信息

操作（Action）	操作（Action）	说明
	cce:userAuthorization:get	获取 CCE 用户授权
cce:*:list	cce:cluster:list	查询集群列表
	cce:node:list	查询节点列表
	cce:job:list	查询任务列表（集群层面的 job）
	cce:addonInstance:list	列出所有插件实例
	cce:addonTemplate:list	列出所有插件模板
	cce:chart:list	列出所有模板
	cce:nodepool:list	列出集群的所有节点池
	cce:release:list	列出所有模板实例
	cce:storage:list	列出所有磁盘
cce:kubernetes:*	-	操作所有 kubernetes 资源，具体权限请在 15.3 命名空间权限中配置。
ecs:*:get	-	ECS（弹性云主机）所有资源详情的查看权限。 CCE 中的一个节点就是具有多个云硬盘的一台弹性云主机
ecs:*:list	-	ECS（弹性云主机）所有资源列表的查看权限。
evs:*:get	-	EVS（云硬盘）所有资源详情的查看权限。可以将云硬盘挂载到云服务器，并可以随时扩容云硬盘容量
evs:*:list	-	EVS（云硬盘）所有资源列表的查看权限。
evs:*:count	-	-
vpc:*:get	-	VPC（虚拟私有云，包含二代 ELB）所有资源详情的查看权限。 创建的集群需要运行在虚拟私有云中，创建命名空间时，需要创建或关联 VPC，创建在命名空间的容器都运行在 VPC 之内
vpc:*:list	-	VPC（虚拟私有云，包含二代 ELB）所有资源列表的查看权限。
sfs:*:get*	-	SFS（弹性文件服务）服务所有资源详情的查看权限。

操作（Action）	操作（Action）	说明
sfs:shares:ShareAction	-	SFS（弹性文件服务）资源的扩容共享。
aom:*:get	-	AOM（应用运维管理）服务所有资源详情的查看权限。
aom:*:list	-	AOM（应用运维管理）服务所有资源列表的查看权限。
aom:autoScalingRule:*	-	AOM（应用运维管理）服务自动扩缩容规则的所有操作权限。

自定义策略

如果系统预置的 CCE 策略，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。

本章为您介绍常用的 CCE 自定义策略样例。

说明

job 类型的操作不支持自定义资源路径，如 cce:job:get、cce:job:list、cce:job:delete。

CCE 自定义策略样例：

- 示例 1：创建一个名称为“test”的集群

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cce:cluster:create"
            ]
        }
    ]
}
```

- 示例 2：拒绝用户删除节点

  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 CCE FullAccess 的系统策略，但不希望用户拥有 CCE FullAccess 中定义的删除节点权限（cce:node: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 CCE FullAccess 和拒绝策略授予用户，根据 Deny 优先原则，则用户可以对 CCE 执行除了删除节点外的所有操作。拒绝策略示例如下：

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "cce:node:delete"
            ]
        }
    ]
}
```

- 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",
                "ecs:cloudServers:delete",
                "ims:images:list",
                "ims:serverImages:create"
            ],
            "Effect": "Allow"
        }
    ]
}
```

CCE 集群权限与 IAM RBAC

CCE 兼容 IAM 传统的系统角色进行权限管理，建议您切换使用 IAM 的细粒度策略，避免设置过于复杂或不必要的权限管理场景。

CCE 当前支持的角色如下：

- IAM 的基础角色：
  - te_admin（Tenant Administrator）：可以调用除 IAM 外所有服务的所有 API。
  - readonly（Tenant Guest）：可以调用除 IAM 外所有服务的只读权限的 API。
- CCE 的自定义管理员角色：CCE Administrator。
- 由于历史原因，CCE 的 API 中保留了对应用管理与运维平台（ServiceStage）的三个系统角色（SvcStg Administrator、SvcStg Developer、SvcStg Operator）的兼容，当前 CCE 和 ServiceStage 已经全面适配了 IAM 的细粒度策略进行权限管理，**不建议您继续使用该系统角色进行权限管理**。其在 CCE 侧的具体权限如下：

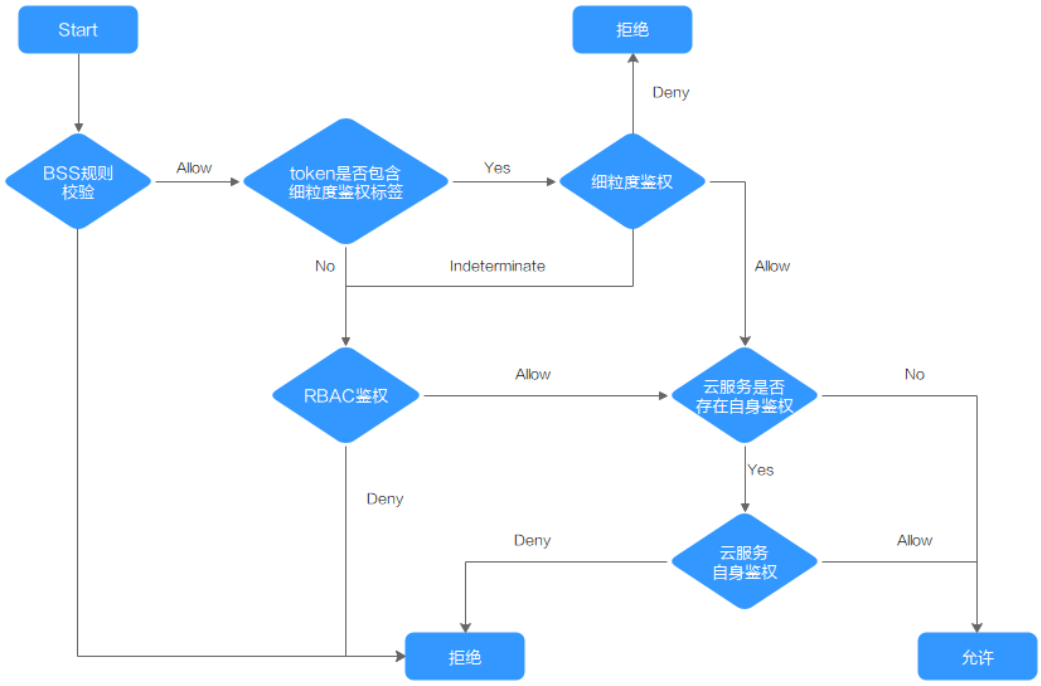
- SvcStg Administrator：拥有 CCE Administrator 相同权限，例外：拥有此角色的用户不默认具有命名空间的权限（Kubernetes RBAC）。
- SvcStg Developer：拥有 CCE Administrator 相同权限，例外：拥有此角色的用户不默认具有命名空间的权限（Kubernetes RBAC）。
- SvcStg Operator：拥有 CCE 的只读权限，但不默认具有命名空间的权限。

说明

- Tenant Administrator、Tenant Guest 是特殊的 IAM 系统角色，当配置任意系统或自定义策略后，Tenant Administrator、Tenant Guest 将以系统策略形式生效，用于兼容 IAM RBAC 和 ABAC 场景。
- 如果用户有 Tenant Administrator 或者 CCE Administrator 的系统角色，则此用户拥有 Kubernetes RBAC 的 cluster-admin 权限，在集群创建后不可移除。

  如果用户为集群创建者，则默认被授权 Kubernetes RBAC 的 cluster-admin 权限，此项权限可以在集群创建后被手动移除:
- 方式 1：权限管理 - 命名空间权限 - 移除 cluster-creator。
- 方式 2：通过 API 或者 kubectl 删除资源，ClusterRoleBinding：cluster-creator。

RBAC 与 IAM 策略共存时，CCE 开放 API 或 Console 操作的后端鉴权逻辑如下：



注意

CCE 部分接口由于涉及命名空间权限或关键操作，需要特殊权限：

clusterCert 获取集群 k8s kubeconfig: cceadm/teadmin

15.3 命名空间权限

命名空间权限是基于 Kubernetes RBAC 能力的授权，通过权限设置可以让不同的用户或用户组拥有操作不同 Kubernetes 资源的权限。Kubernetes RBAC API 定义了四种类型：Role、ClusterRole、RoleBinding 与 ClusterRoleBinding，这四种类型之间的关系和简要说明如下：

- **Role 和 ClusterRole：**描述角色和权限的关系。在 Kubernetes 的 RBAC API 中，一个角色定义了一组特定权限的规则。命名空间范围内的角色由 Role 对象定义，而整个 Kubernetes 集群范围内有效的角色则通过 ClusterRole 对象实现。
- **RoleBinding 和 ClusterRoleBinding：**描述 subjects （包含 users, groups, service accounts）和角色的关系。角色绑定将一个角色中定义的各种权限授予一个或者一组用户，则该用户或用户组则具有对应绑定的 Role 或 ClusterRole 定义的权限。

表15-3 RBAC API 所定义的四种类型

类型名称	说明
Role	Role 对象只能用于授予对某一 namespace 中资源的访问权限。
ClusterRole	ClusterRole 对象可以授予整个集群范围内资源访问权限，也可以对以下几种资源的授予访问权限： • 集群范围资源（例如节点，即 node）。 • 非资源类型 endpoint（例如”/healthz”）。 • 跨所有 namespaces 的范围资源（例如 pod，需要运行命令 kubectl get pods --all-namespaces 来查询集群中所有的 pod)。
RoleBinding	RoleBinding 可以将同一 namespace 中的 subject（用户）绑定到某个具有特定权限的 Role 下，则此 subject 即具有该 Role 定义的权限。
ClusterRoleBinding	ClusterRoleBinding 在整个集群级别和所有 namespaces 将特定的 subject 与 ClusterRole 绑定，授予权限。

CCE 命名空间权限

CCE 中的命名空间权限是基于 Kubernetes RBAC 能力的授权，通过权限设置可以让不同的用户或用户组拥有操作不同 Kubernetes 资源的权限。

CCE 的 kubernetes 资源通过命名空间进行权限设置，目前包含 **cluster-admin**、**admin**、**edit**、**view** 四种角色，详见表 15-4。

表15-4 用户/用户组角色说明

默认 ClusterRole	描述

默认 ClusterRole	描述
cluster-admin	允许超级用户在平台上的任何资源的所有操作。允许对集群中以及所有命名空间中的全部资源进行完全控制。
admin	允许管理员访问权限。允许对命名空间中大多数资源进行读/写操作，包含创建角色和角色绑定（RoleBinding）的能力。但不允许对资源配额或者命名空间本身进行写操作。
edit	允许对命名空间内的大多数对象进行读/写操作，不允许查看或修改角色（Roles）或者角色绑定（RoleBinding）。
view	允许对命名空间的大多数对象进行只读操作，但不允许查看角色（Roles）或者角色绑定（RoleBinding），不允许查看 Secrets，因为这类操作属于越权。

说明

- CCE 集群已支持基于 SAML 2.0 的联合身份验证，目前暂不支持 ADFS 联邦用户查看命名空间和工作负载。

  目前已支持基于 SAML 2.0 的单点登录功能，如果您已经有自己的身份认证系统，同时您的用户需要使用您帐号内的资源，您可以使用的身份提供商功能，实现用户使用企业身份提供商（Identity Provider ，简称 IdP）帐号单点登录，这一过程称之为联邦身份认证。

  目前支持两种形式的联邦身份认证：

- **浏览器页面单点登录（Web SSO）：**浏览器作为通讯媒介，适用于普通用户通过浏览器访问。
- **调用 API 接口：**开发工具/应用程序作为通讯媒介，例如 OpenStack Client、ShibbolethECP Client，适用企业或用户通过 API 调用方式访问。
- 除默认 ClusterRole（cluster-admin、admin、edit、view）以外的其他角色为系统组件所使用的角色，请根据您的业务需要进行选择。

更多 Kubernetes RBAC 授权的内容可以参考 Kubernetes RBAC 官方文档。

CCE 命名空间权限的增强

支持绑定 IAM 用户或用户组到 Kubernetes 的 User 和 Group，以支持对应用户的 IAM Token 直接访问 Kubernetes API 并支持 Kubernetes RBAC 权限控制。

API

```
curl https://{master-IP}:5443/api/v1/namespaces/default/pods -H
"Authorization:Bearer $token" -k
```

kubectl

```
// 使用用户信息配置 kubeconfig 后即可正常使用 kubectl
{
    "apiVersion": "v1",
    "clusters": [
```

```
        {
            "cluster": {
                "server": "https://192.168.0.57:5443",
                "insecure-skip-tls-verify": true
            },
            "name": "myCluster"
        }
    ],
    "contexts": [
        {
            "context": {
                "cluster": "myCluster",
                "user": "user"
            },
            "name": "myContext"
        }
    ],
    "current-context": "myContext",
    "kind": "Config",
    "preferences": {},
    "users": [
        {
            "name": "user",
            "user": {
                "token": "$token"
            }
        }
    ]
}
```

说明

通过 Console、APIGW 访问 Kubernetes API，CCE 针对 list namespace,permission 进行了特殊处理，将支持获取当前子用户有权限的 NameSpace 列表，而非 Kubernetes 原生的 403，提供更好的前端显示体验。

15.4 创建用户并授权使用 CCE

15.4.1 设置集群权限

本章节通过简单的用户组授权方法，使 IAM 用户“James”拥有对应集群的只读权限，操作流程如示例流程所示。

须知

- 集群权限仅针对与集群相关的资源（如集群、节点等）有效，您必须确保同时配置了 15.4.2 设置命名空间权限，才能有操作 Kubernetes 资源（如工作负载、Service 等）的权限。
- CCE 提供了 CCE FullAccess 和 CCE ReadOnlyAccess 两种策略，建议您通过默认的系统策略实现权限管理。
- 如果您需要使用自定义策略，请仔细阅读自定义策略中可以添加的授权项（Action），如遇问题提交工单处理。

前提条件

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

配置说明

- 您需要拥有一个帐号，仅帐号或授予了 CCE Administrator 权限且全局设置了 Security Administrator 角色的 IAM 用户，才有权限进入 CCE“权限管理”界面对其他 IAM 用户进行授权操作。
- 如果目标 IAM 用户涉及读写权限变更，只能由帐号授权。
- 由于 IAM 的安全限制，当您通过云容器引擎控制台的授权配置涉及到帐号 IAM 授权的修改时，需要您按照页面上给出的参考策略内容和操作说明，在 IAM 控制台进行目标 IAM 用户的手动授权。
- 权限设置完成后，需退出重新登录才能生效。

示例流程

用户组是用户的集合，CCE 通过 IAM 的用户组功能实现用户的授权。您在 IAM 中创建的 IAM 用户，需要加入特定用户组后，IAM 用户才具备用户组所拥有的权限。

关于创建用户组并给用户组授权的方法，可以参考如下操作：

- 步骤一：创建用户组
- 步骤二：为用户组授权
- 步骤三：创建 IAM 用户并加入用户组
- 步骤四：使用 IAM 用户登录并验证权限

步骤一：创建用户组

在“统一身份认证 IAM”服务控制台创建用户组，并授予具有云容器引擎只读权限的“CCE ReadOnlyAccess”策略。

用户组是用户的集合，IAM 通过用户组功能实现用户的授权。您在 IAM 中创建的用户，需要加入特定用户组后，用户才具备用户组所拥有的权限。关于创建用户组并给用户组授权的方法，可以参考如下操作。

步骤 1 使用注册的帐号登录，登录时请选择“帐号登录”。

步骤 2 在控制台首页菜单中单击“云容器引擎 CCE”，进入 CCE 控制台。

步骤 3 单击 CCE 控制台左侧导航栏中的“权限管理”，单击“集群权限”页签下的“创建用户组”。

步骤 4 进入“统一身份认证服务”控制台的“创建用户组”界面，输入用户组名称（本例以“开发人员组”为例）。

单击“确定”，用户组创建完成，界面自动返回用户组列表，列表中显示新建的用户组。

说明

您最多可以创建 20 个用户组，如果当前资源配额无法满足业务需要，您可以申请扩大配额。

----结束

步骤二：为用户组授权

步骤 1 在用户组列表中，单击新建用户组右侧的“权限配置”。

在用户组权限页签下，单击列表左上方的“配置权限”。

步骤 2 选择作用范围。此处选择“区域级项目”，且需要在下拉框中选择需要授权的区域。

CCE 服务为区域级项目，不是全局服务，两者的区别如下：

- 全局服务：服务部署时不区分物理区域，在全局区域中授权。全局服务包括对象存储服务（OBS）、内容分发网络（CDN）、标签管理服务（TMS）等。
- 区域级项目：基于区域（如华北-北京四、华南-广州等）进行部署的服务，授权后只在授权区域生效，如果需要所有区域都生效，则所有区域都需要进行授权操作。

步骤 3 在搜索框中搜索“CCE”，勾选需要授予用户组的权限，本例此处选择“CCE ReadOnlyAccess”。

步骤 4 单击“确定”，完成用户组授权。

----结束

步骤三：创建 IAM 用户并加入用户组

IAM 用户与企业中的实际员工或是应用程序相对应，有唯一的安全凭证，可以通过加入一个或多个用户组来获得用户组的权限。关于 IAM 用户的创建方式请参见如下步骤。

1. 在统一身份认证服务，左侧导航窗格中，单击“用户”>“创建用户”。
2. 在“创建用户”页面填写“用户信息”。如需一次创建多个用户，可以单击“添加用户”进行批量创建，每次最多可创建 10 个用户。

表15-5 用户信息

用户信息	说明
用户名	必填。IAM 用户登录的用户名，此处以“James”和“Alice”为例。
邮箱	“访问方式”选择“首次登录时设置”时必填，选择其他时选填。IAM 用户绑定的邮箱，可作为子帐户的登录凭证，也可由 IAM 用户自己绑定。
手机号	选填。IAM 用户绑定的手机号，可作为子帐户的登录凭证，也可由 IAM 用户自己绑定。
描述	选填。记录 IAM 用户相关信息。

3. 在“创建用户”页面选择“访问方式”，完成后单击“下一步”。

表15-6 访问方式

访问方式	配置信息		说明
管理控制台访问	控制台登录密码设置方式	首次登录时设置	如果您不是用户 James 的使用主体，建议您选择该方式，输入用户的邮箱和手机，用户 James 通过邮件中的一次性链接登录，自行设置密码。
		自动生成	仅在创建单个用户时适用，如果您本次创建 2 个及以上的用户，则不支持此方式。
		自定义	如果您是用户 James 的使用主体，建议您选择该方式，设置自己的登录密码。
	登录保护	开启登录保护	开启登录保护后，IAM 用户登录时，除了在登录页面输入用户名和密码外（第一次身份验证），还需要在登录验证页面输入验证码（第二次身份验证），该功能是一种安全实践，建议开启登录保护，多次身份认证可以提高帐号安全性。 您可以选择通过手机、邮箱、虚拟 MFA 进行登录验证。
		不开启	-
编程访问	--	--	创建用户完成后即可下载本次创建的所有用户的管理访问密钥。一个用户最多拥有两个访问密钥。这些 AK/SK 可以对进行编程调用，例如，通过 API 调用方式访问时，您可能需要使用访问密

访问方式	配置信息		说明
			钥。

说明

- 用户可以使用此处设置的用户名、邮箱或手机号码任意一种方式登录。
- 当用户忘记密码时，可以通过此处绑定的邮箱或手机自行重置密码，如果用户没有绑定邮箱或手机号码，只能由管理员重置密码。
- 用户登录系统时，输入用户名和初始密码后，将进入“首次登录修改密码”，需要创建一个新密码，该功能可以保证用户的密码是由使用者本人所设置，防止密码泄露。

4. 单击“下一步”，将用户加入到用户组（可选）。
   - 选择新创建的用户组“开发人员组”。将用户加入用户组，用户将具备用户组的权限，这一过程即给该用户授权。其中“admin”为系统缺省提供的用户组，具有管理人员以及所有云服务资源的操作权限。
   - 如需创建新的用户组，可单击“创建用户组”，填写用户组名称和描述（可选），创建成功后即可将用户加入到新创建的用户组中。
5. 单击“下一步”，IAM 用户创建成功，用户列表中显示新创建的 IAM 用户。如果在访问方式中勾选了“编程访问”，可在此页面下载访问密钥。

步骤四：使用 IAM 用户登录并验证权限

IAM 用户创建完成后，可以使用新用户的用户名及身份凭证登录验证权限，即“CCE ReadOnlyAccess”权限。

步骤 1 在登录页面，单击右下角的“IAM 用户登录”。

图15-1 IAM 用户登录

步骤 2 在“IAM 用户登录”页面，输入帐号名、用户名及用户密码，使用新创建的 IAM 用户登录。

- 帐号名为该 IAM 用户所属帐号的名称。
- 用户名和密码为创建 IAM 用户“James”时输入的用户名和密码，首次登录时需要重置密码。

如果登录失败，您可以联系您的帐号主体，确认用户名及密码是否正确，或是重置用户名及密码。

步骤 3 使用 IAM 用户“James”登录成功后，进入控制台，请先切换至授权区域。

步骤 4 在“服务列表”中选择“云容器引擎 CCE”，返回 CCE 控制台主界面，对 IAM 用户 James 的集群权限进行验证。

验证方式（参考）：您可以尝试创建一个集群或休眠一个已创建的集群，此时如果提示“您的权限不足。”，则表明您为 James 用户设置的“CCE ReadOnlyAccess”只读权限策略已生效。

----结束

15.4.2 设置命名空间权限

本章节通过简单的命名空间授权方法，将 CCE 服务的用户和用户组授予操作不同命名空间资源的权限，从而使用户和用户组在拥有对应的 CCE 集群标准权限的同时，又可以拥有对集群中命名空间的操作权限。设置流程如示例流程所示。

配置说明

- 您需要拥有一个帐号，有一个或若干个用户组和 IAM 用户。
- 本例将对用户和用户组授予操作不同命名空间资源的权限，在您的实际业务中，您可根据业务需求仅对用户或用户组授予不同的权限。
- 本例仅用于给用户或用户组在未授权过的命名空间下新增权限，已授权的用户或用户组的权限可以在“权限管理 > 命名空间权限”的列表“操作”栏中单击“编辑权限”进行修改。
- 当给用户或用户组添加多个权限时，多个权限会同时生效（取并集）；为用户组设置的权限将作用于用户组下的全部用户。

约束与限制

kubernetes RBAC 的授权能力支持 1.13 及以上版本集群。

在 v1.11.7-r2 版本的集群中默认开启 RBAC 功能，若需使用 RBAC 功能请将集群升级至 v1.11.7-r2 或以上版本。升级方法请参见 3.6.2 升级集群。

表15-7 各类型用户在不同版本集群中的权限对比

用户类型	v1.11.7-r2 以下版本的集群	v1.11.7-r2 及以上版本的集群
帐号	所有权限	所有权限
赋予 CCE Administrator 权限的 IAM 用户	所有权限	所有权限
赋予 CCE FullAccess 或者 CCE ReadOnlyAccess 权限的 IAM 用户	所有权限	按照权限管理界面设置的命名空间权限处理
赋予 Tenant Guest 权限的 IAM 用户（未申请开启细粒度公测的帐号）	只读权限	只读权限
赋予 Tenant Guest 权限的 IAM 用户（已申请开启细粒度公测的帐号）	所有权限 说明 开启细粒度公测后，行为有变，从只读变为所有权限	按照权限管理界面设置的命名空间权限处理

示例流程

命名空间是对一组资源和对象的抽象整合。在同一个集群内可创建不同的命名空间，不同命名空间中的数据彼此隔离，使得它们既可以共享同一个集群的服务，也能够互不干扰。命名空间的一个重要的作用是充当一个虚拟的集群，用于多种工作用途，满足多用户的使用需求。

本章节将沿用 15.4 创建用户并授权使用 CCE 中创建的 IAM 用户“James”和用户组“开发人员组”进行示例，为 IAM 用户“James”和用户组“开发人员组”添加命名空间权限，可以参考如下操作：

- 步骤一：为 IAM 用户/用户组添加命名空间权限
- 步骤二：用户登录并验证权限

步骤一：为 IAM 用户/用户组添加命名空间权限

本步骤将在 CCE 控制台中为 IAM 用户“James”以及用户组“开发人员组”授予某个集群命名空间下资源的操作权限，设置如下：

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“权限管理”，进入权限管理页面。

步骤 2 单击“命名空间权限”页签，在命名空间权限列表右上方选择要添加授权的集群，单击“添加授权”按钮，进入添加授权页面。

步骤 3 在添加授权页面，确认集群名称，选择该集群下要授权使用的命名空间，此处选择“default”。

步骤 4 单击“添加用户权限”，为“开发人员组”增加“admin”权限，在展开的选项中进行如下配置：

- 用户/用户组：选择“用户组”，并在二级选项中选择“开发人员组”。
- 权限：选择“admin”。

单击“添加用户权限”可继续增加其他用户或用户组，并赋予相应的权限。

步骤 5 单击下方的“添加命名空间权限”，为用户“James”增加在另一个命名空间“monitoring”的权限，在展开的选项中进行如下配置：

- 命名空间：选择“monitoring”。
- 用户/用户组：选择“用户”，并在二级选项中选择“James”增加。
- 权限：选择“view”。

步骤 6 单击“创建”，完成以上用户和用户组在命名空间中的相应权限设置。

说明

经过以上操作，授权结果如下：

- 由于“开发人员组”包含 IAM 用户“James”，因此 IAM 用户“James”也同时获得了在命名空间“default”的“admin”权限。
- 为 IAM 用户“James”增加了在命名空间“monitoring”的“view”权限。

----结束

步骤二：用户登录并验证权限

使用 IAM 用户“James”登录云容器引擎控制台，验证授予的命名空间权限，验证步骤如下：

步骤 1 在登录页面，单击右下角的“IAM 用户登录”。

步骤 2 在“IAM 用户登录”页面，输入帐号名、用户名及用户密码，使用新创建的 IAM 用户登录。

- 帐号名为该 IAM 用户所属帐号的名称。
- 用户名和密码为创建 IAM 用户 James 时输入的用户名和密码，首次登录时需要重置密码。

如果登录失败，您可以联系您的帐号主体，确认用户名及密码是否正确，或是重置用户名及密码。

步骤 3 登录成功后，进入控制台，请先切换至授权区域。

步骤 4 在“服务列表”中选择“云容器引擎 CCE”，进入 CCE 控制台，对 IAM 用户 James 的命名空间权限进行验证。

----结束

15.5 CCE 控制台的权限依赖

CCE 对其他云服务有诸多依赖关系，因此在您开启 IAM 系统策略授权后，在 CCE Console 控制台中的各项功能需要配置相应的服务权限后才能正常查看或使用，详细说明如下：

- 依赖服务的权限配置均基于您已设置了 IAM 系统策略授权的 CCE FullAccess 或 CCE ReadOnlyAccess 策略权限，详细设置方法请参见 15.4.1 设置集群权限。
- Tenant Guest 权限在您开启 IAM 系统策略授权之后将不再生效，也不会再有全局只读的效果。
- 1.11.7-r2 及以上版本的集群，显示情况依赖于命名空间权限的设置情况，如果没有设置命名空间权限，则无法查看集群下的资源。
  - 如果您设置了全部命名空间的 view 权限，则可以查看到对应集群的全部命名空间下的资源，但密钥（Secret）除外，密钥（Secret）需要在命名空间权限下设置 admin 或者 edit 权限才能查看。
  - CustomedHPA 与 HPA 策略需要配置命名空间 cluster-admin 权限下才能生效。
  - 如果您设置的是单一命名空间的 view 权限，则看到的只能是指定命名空间下的资源。

依赖服务的权限设置

如果 IAM 用户需要在 CCE Console 控制台中拥有相应功能的查看或使用权限，请确认已经对该用户所在的用户组设置了 CCE Administrator、CCE FullAccess 或 CCE ReadOnlyAccess 策略的集群权限，再按如下表 15-8 增加依赖服务的角色或策略。

表15-8 CCE Console 中依赖服务的角色或策略

Console 控制台功能	依赖服务	需配置角色/策略
总览	应用运维管理 AOM	• IAM 用户设置了 CCE Administrator 权限后，需要增加 AOM FullAccess 权限后才能访问总览中的数据图表。 • 支持设置了 IAM ReadOnlyAccess 和 CCE FullAccess 或 CCE ReadOnlyAccess 权限的 IAM 用户直接访问总览中的数据图表。
工作负载	弹性负载均衡 ELB 应用性能管理 APM 应用运维管理 AOM NAT 网关 NAT 对象存储服务 OBS 弹性文件服务 SFS	正常创建工作负载时不依赖其他服务的权限。 • 如果需要创建 ELB 类型的服务，需要设置 ELB FullAccess 或者 ELB Administrator 权限，以及 VPC Administrator 权限。 • 如果需要使用 Java 探针，需要设置 AOM FullAccess 和 APM FullAccess 权限。 • 如果需要 NAT 网关类型的服务，需要设置 NAT Gateway Administrator 权限。 • 如果使用对象存储，需要全局设置 OBS Administrator 权限。 说明 由于缓存的存在，对用户、用户组以及企业项目授予 OBS 相关的 RBAC 策略后，大概需要等待 13 分钟 RBAC 策略才能生效；授予 OBS 相关的系统策略后，大概需要等待 5 分钟系统策略能生效。 • 如果使用文件存储，需要设置 SFS FullAccess 权限。
集群管理	应用运维管理 AOM 费用中心 BSS	• 如果需要弹性扩容权限，需要设置 AOM FullAccess 权限。 • 如果需要转包周期，需要设置 BSS Administrator 权限。
节点管理	弹性云主机 ECS	当 IAM 用户权限为 CCE Administrator 时，如果创建和删除节点，需要配置 ECS FullAccess 或 ECS Administrator 权限，以及 VPC Administrator 权限。
网络管理	弹性负载均衡 ELB	正常创建时不依赖其他服务的权限。 • 如果需要创建 ELB 类型的服务，需

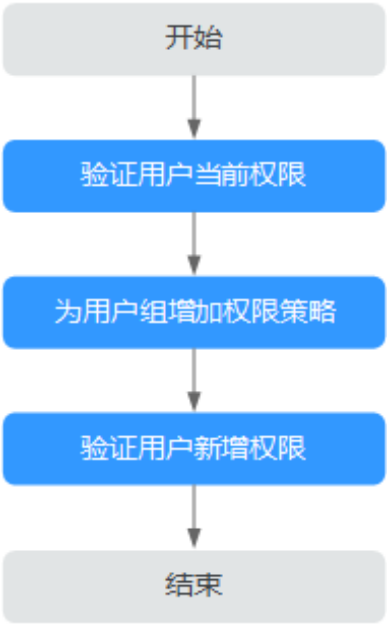
Console 控制台功能	依赖服务	需配置角色/策略
	NAT 网关 NAT	要设置 ELB FullAccess 或者 ELB Administrator 权限，以及 VPC Administrator 权限。 • 如果需要 NAT 网关类型的服务，需要设置 NAT Administrator 权限。
存储管理	极速文件存储 EFS（SFS Turbo）	使用极速文件存储，需要设置 SFS Turbo Admin 权限
命名空间	/	无需其他依赖权限。
模板市场	/	当前仅支持帐号、设置了 CCE Administrator 权限的 IAM 用户访问。
插件管理	/	支持帐号、设置了 CCE Administrator、CCE FullAccess 或 CCE ReadOnlyAccess 等权限的 IAM 用户访问本功能。
权限管理	/	• 支持帐号访问。 • 支持设置了 CCE Administrator 和 Security Administrator（全局级策略）权限的 IAM 用户访问。 • 支持设置了 IAM ReadOnlyAccess 和 CCE FullAccess 或 CCE ReadOnlyAccess 权限的 IAM 用户访问。
配置中心	/	• 配置项（ConfigMap）无需其他依赖权限。 • 密钥（Secret）需要在命名空间权限下设置 cluster-admin、admin 或者 edit 权限才能查看，依赖服务需要添加 DEW KeypairFullAccess 或者 DEW KeypairReadOnlyAccess 权限。
帮助中心	/	无需其他依赖权限。
其他服务跳转	容器镜像服务 SWR 应用运维管理 AOM 云监控服务 Cloud Eye（存储管理与节点管理的“监控”跳转）	为便于您快速进入 CCE 相关服务的控制台，在 CCE 控制台中增加了其他服务的跳转链接，CCE 默认没有这些服务的全部权限，如果 IAM 用户需要查看或使用其功能，请按照该服务的权限策略说明设置相应的权限策略。

示例流程

本章节通过为 IAM 用户“James”增加 CCE **“总览”**页面监控信息的查看权限为例进行演示。

IAM 用户“James”在用户组“开发人员组”，“开发人员组”仅有 CCE Administrator 权限。而参照表 15-8 可知，CCE 总览页面的监控信息还需要增加“AOM FullAccess”策略才能正常查看。

图15-2 依赖服务授权流程



步骤一：验证用户当前权限

步骤 1 使用 IAM 用户“James”登录 CCE 控制台。

步骤 2 单击左侧导航栏中的“总览”，进入总览页面，可以看到该用户无权查看监控相关的模块信息。

----结束

步骤二：为用户组增加系统策略

步骤 1 IAM 用户“James”退出登录，使用帐号登录 CCE 控制台。

步骤 2 在 CCE 控制台中，单击左侧导航栏中的“权限管理”，在“集群权限”页签下单击“开发人员组”后的“设置权限”。

步骤 3 在弹出的“设置权限”窗口中，单击“去设置”。

步骤 4 进入统一身份认证服务的“用户组 > 开发人员组”页面，在“用户组权限”页签下，单击“配置权限”。

步骤 5 在弹出的“配置权限”窗口中，选择作用范围。此处选择“区域级项目”，并在下拉框中选择需要授权的区域。

步骤 6 在权限列表上方的搜索框中搜索“AOM”，单击“AOM FullAccess”策略并选中，单击“确定”。

步骤 7 在“权限管理”页签下可以看到已经增加“AOM FullAccess”策略。

步骤 8 返回 CCE 控制台，在“权限管理”的“集群权限”页签下单击右上角的刷新图标 ，可以看到“AOM FullAccess”权限策略已添加成功。

----结束

步骤三：验证用户新增权限

步骤 1 使用 IAM 用户“James”登录 CCE 控制台。

步骤 2 单击左侧导航栏中的“总览”，进入总览页面，可以看到该用户已有权查看监控相关的模块信息。

步骤 3 为 IAM 用户“James”增加查看总览页监控相关的依赖服务权限完成，验证可正常查看。

----结束

16 云监控服务

16.1 支持的监控指标

CCE 通过云监控服务（Cloud Eye）为您提供节点的监控，每个节点对应一台弹性云主机。

监控是保持 CCE 节点可靠性、可用性和性能的重要部分，通过监控，用户可以观察节点资源。您可以使用该服务监控您的节点，执行自动实时监控、告警和通知操作，帮助您更好地了解节点的各项性能指标。

表16-1 支持的主要监控指标

监控指标	指标含义
CPU 使用率	统计测量对象的 CPU 使用率，以百分比为单位。
磁盘读速率	统计每秒从测量对象读出数据量，以字节/秒为单位。
磁盘读操作速率	该指标用于统计每秒从弹性云主机读取数据的请求次数，以请求/秒为单位。
磁盘写速率	该指标用于统计每秒写到弹性云主机的数据量，以字节/秒为单位。
磁盘写操作速率	该指标用于统计每秒从弹性云主机写数据的请求次数，以请求/秒为单位。
带外网络流入速率	该指标用于在虚拟化层统计每秒流入弹性云主机的网络流量，以字节/秒为单位。
带外网络流出速率	该指标用于在虚拟化层统计每秒流出弹性云主机的网络流量，以字节/秒为单位。

相关操作

云监控服务（Cloud Eye）为您提供一个针对弹性云主机、带宽等资源的立体化监控平台。使您全面了解云上的资源使用情况、业务的运行状况，并及时收到异常告警做出反应，保证业务顺畅运行。

16.2 设置告警规则

操作场景

通过设置节点告警规则，用户可自定义监控目标与通知策略，及时了解节点运行状况，从而起到预警作用。

设置节点的告警规则包括设置告警规则名称、监控对象、监控指标、告警阈值、监控周期和是否发送通知等参数。本节介绍了设置告警规则的具体方法。

操作步骤

步骤 1 登录管理控制台。

步骤 2 单击“管理与部署 > 云监控服务”。

步骤 3 选择“告警 > 告警规则”。

步骤 4 单击“创建告警规则”。

步骤 5 在“创建告警规则”界面，根据界面提示配置参数。

1. 根据界面提示，配置规则信息参数。

图16-1 配置规则信息

表16-2 配置规则信息

参数	参数说明
名称	系统会随机产生一个名称，用户也可以进行修改。 取值样例：alarm-b6al
描述	告警规则描述（此参数非必填项）。

2. 选择监控对象，配置告警内容参数。

表16-3 配置告警内容

参数	参数说明	取值样例
资源类型	配置告警规则监控的服务名称。	弹性云主机
维度	用于指定告警规则对应指标的维度名称	云服务器
监控范围	告警规则适用的资源范围，可选择资源分组或指定资源。 说明 • 当选择资源分组时，该分组下任何资源满足告警策略时，都会触发告警通知。 • 选择指定资源时，勾选具体的监控对象，单击 将监控对象同步到右侧对话框。	指定资源
选择分组	当监控范围为资源分组时需配置此参数。	-
选择类型	根据需要可选择从模板导入或自定义创建。 说明 当监控范围为指定资源时可选择从模板导入。	自定义创建
模板	选择需要导入的模板。	-
告警策略	触发告警规则的告警策略。 当资源类型选择站点监控、日志监控、自定义监控、具体的云服务时，告警策略为一个周期性的动作。当资源类型选择事件监控时，具体的事件为一个瞬间的操作动作，而不是周期性动作。 例如：CPU 使用率，监控周期为 5 分钟，连续三个周期平均值≥80%	-
挂载点	当监控指标为细颗粒度的磁盘类监控指标时需配置该参数。 Windows 系统请输入对应的驱动器号，比如 C、D 或者 E 等，Linux 系统请输入对应的挂载点，比如 /dev、/opt 等。	/dev
告警级别	根据告警的严重程度不同等级，可选择紧急、重要、次要、提示。	重要

3. 根据界面提示，配置告警通知参数。

表16-4 配置告警通知

参数	参数说明
发送通知	配置是否发送邮件、短信、HTTP 和 HTTPS 通知用户。
生效时间	该告警规则仅在生效时间内发送通知消息。 如生效时间为 08:00-20:00，则该告警规则仅在 08:00-20:00 发送通知消息。
主题通知	需要发送告警通知的主题名称。 当发送通知选择“是”时，需要选择已有的主题名称，若此处没有需要的主题则需先创建主题，该功能会调用消息通知服务（SMN），创建主题请参见《消息通知服务用户指南》。
触发条件	可以选择“出现告警”、“恢复正常”两种状态，作为触发告警通知的条件。

4. 配置完成后，单击“立即创建”，完成告警规则的创建。

----结束

告警规则添加完成后，当监控指标触发设定的阈值时，云监控服务会在第一时间通过消息通知服务实时告知您云上资源异常，以免因此造成业务损失。

16.3 查看监控指标

操作场景

云平台提供的云监控，可以对节点运行状态进行日常监控。您可以通过管理控制台，直观地查看节点的各项监控指标。

由于监控数据的获取与传输会花费一定时间，因此，云监控显示的是当前时间 5～10 分钟前的节点状态。如果您的节点刚刚创建完成，请等待 5～10 分钟后查看监控数据。

操作步骤

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“资源管理 > 节点管理”。

步骤 2 单击待监控节点后的“监控”，系统会链接到云监控控制台，查看节点监控信息。

----结束

17 云审计服务

17.1 云审计服务支持的 CCE 操作列表

CCE 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17-1 云审计服务支持的 CCE 操作列表

操作名称	资源类型	事件名称
创建集群	cce	createCluster
更新集群	cce	updateCluster
删除集群	cce	deleteCluster/claimCluster
创建节点	cce	createNode
添加静态节点	cce	addStaticNode
更新节点	cce	updateNode
删除一个主机	cce	deleteOneHost
删除所有主机	cce	deleteAllHosts
挂起用户资源	cce	suspendUserResource
创建 ConfigMap	cce	createConfigmaps
创建 DaemonSet	cce	createDaemonsets
创建 Deployment	cce	createDeployments
创建 Event	cce	createEvents
创建 Ingress	cce	createIngresses
创建 Job	cce	createJobs

操作名称	资源类型	事件名称
创建 namespace	cce	createNamespaces
创建 Node	cce	createNodes
创建 PersistentVolumeClaim	cce	createPersistentvolumeclaims
创建 Pod	cce	createPods
创建 ReplicaSet	cce	createReplicasets
创建 ResourceQuota	cce	createResourcequotas
创建密钥	cce	createSecrets
创建服务	cce	createServices
创建 StatefulSet	cce	createStatefulsets
创建卷	cce	createVolumes
删除 ConfigMap	cce	deleteConfigmaps
删除 DaemonSet	cce	deleteDaemonsets
删除 Deployment	cce	deleteDeployments
删除 Event	cce	deleteEvents
删除 Ingress	cce	deleteIngresses
删除 Job	cce	deleteJobs
删除 Namespace	cce	deleteNamespaces
删除 Node	cce	deleteNodes/claimOneHost
删除 Pod	cce	deletePods
删除 ReplicaSet	cce	deleteReplicasets
删除 ResourceQuota	cce	deleteResourcequotas
删除 Secret	cce	deleteSecrets
删除 Service	cce	deleteServices
删除 StatefulSet	cce	deleteStatefulsets
删除卷	cce	deleteVolumes
替换指定的 ConfigMap	cce	updateConfigmaps
替换指定的 DaemonSet	cce	updateDaemonsets

操作名称	资源类型	事件名称
替换指定的 Deployment	cce	updateDeployments
替换指定的 Event	cce	updateEvents
替换指定的 Ingress	cce	updateIngresses
替换指定的 Job	cce	updateJobs
替换指定的 Namespace	cce	updateNamespaces
替换指定的 Node	cce	updateNodes
替换指定的 PersistentVolumeClaim	cce	updatePersistentvolumeclaims
替换指定的 Pod	cce	updatePods
替换指定的 Replicaset	cce	updateReplicasets
替换指定的 ResourceQuota	cce	updateResourcequotas
替换指定的 Secret	cce	updateSecrets
替换指定的 Service	cce	updateServices
替换指定的 Statefulset	cce	updateStatefulsets
替换指定的状态	cce	updateStatus
上传组件模板	cce	uploadChart
更新组件模板	cce	updateChart
删除组件模板	cce	deleteChart
创建模板应用	cce	createRelease
更新模板应用	cce	updateRelease
删除模板应用	cce	deleteRelease

17.2 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 CCE 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 单击页面上方的“服务列表”，选择“管理与部署 > 云审计服务”，进入云审计服务信息页面。

步骤 3 单击左侧导航树的“事件列表”，进入事件列表信息页面。

步骤 4 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“CCE”。

  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。
- 时间范围：可选择查询最近七天内任意时间段的操作事件。

步骤 5 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 17-1 所示。

图17-1 展开记录

步骤 6 在需要查看的记录右侧，单击“查看事件”，弹出一个窗口，如图 17-2 所示，显示了该操作事件结构的详细信息。

图17-2 查看事件


```
{
    "service_type": "CCE",
    "user": {
        "domain": {
            "name": "[redacted]",
            "id": "cc9c0076188843ea938743dd166c7fef"
        },
        "id": "69d8a0dc65e2436eb973093b49bb5ecb",
        "name": "[redacted]"
    },
    "time": "2018/10/11 09:24:56 GMT+08:00",
    "code": 200,
    "resource_type": "clusters-action",
    "source_ip": "58.213.108.72",
    "trace_name": "operateCluster",
    "trace_type": "ConsoleAction",
    "message": "UserName: [redacted]; Operation : POST /api/v2/projects/6244f46518ab48d4ba84b3bcb93aba67/clusters/3cc
    "record_time": "2018/10/11 09:24:56 GMT+08:00",
    "trace_id": "7660c4e0-ccf4-11e8-9fe5-286ed488cbe3",
    "trace_status": "normal"
}
```


----结束

18 相关服务

18.1 容器镜像服务

镜像仓库是由容器镜像服务（Software Repository for Container，SWR）提供的用于存储、管理 docker 容器镜像的场所，可以让使用人员轻松存储、管理、部署 docker 容器镜像。

上传镜像

容器镜像服务上传镜像的详细操作请参见《SWR 用户指南》中的“客户端上传镜像”章节。

使用镜像

镜像上传成功后，在 CCE 中创建工作负载时可选择“我的镜像”，以 2048 游戏为例，具体操作如下：

步骤 1 登录 CCE 控制台，在左侧导航栏中选择“工作负载 > 无状态负载 Deployment”，单击“创建无状态工作负载”。

步骤 2 输入以下参数，其它保持默认。

- 工作负载名称：game。
- 集群名称：选择应用所要运行的集群。
- 实例数量：1。

步骤 3 单击“下一步：容器设置”，添加容器。

单击“添加容器”，在“我的镜像”页签下，选择已上传的镜像，单击“确定”。

步骤 4 更多创建工作负载的步骤请参见 6.2 创建无状态负载(Deployment)或 6.3 创建有状态负载(StatefulSet)。

----结束

18.2 应用运维管理

在 CCE 上创建工作负载后，可以使用 AOM 进行运维。

创建工作负载时，建议在 AOM 侧配置监控告警，以便异常时能及时收到告警。若未配置监控告警，工作负载异常时无法及时收到告警，需要人工巡检环境。

应用运维管理（Application Operations Management，简称 AOM）是云上应用的一站式立体化运维管理平台，实时监控您的应用及相关云资源，采集并关联资源的各项指标、日志及事件等数据共同分析应用健康状态，提供灵活的告警及丰富的数据可视化功能，帮助您及时发现故障，全面掌握应用、资源及业务的实时运行状况，快速锁定问题根源，保障业务顺畅运行。

常用操作

- 当您想要关注某些资源的变更信息时，可对这些资源的指标创建阈值规则。
- 在日常运维中，通过仪表盘可实时掌握全局。您可以创建并添加关注的内容到仪表盘。
- 日常巡检应用。