



天翼云 · 云容器引擎

用户使用指南

中国电信股份有限公司云计算分公司

目 录

1	产品介绍	4
1.1	产品定义	4
1.2	Docker 基础知识	4
1.2.1	Docker 基本概念	4
1.2.2	安装Docker	5
1.2.3	制作Docker 镜像	5
1.3	产品优势	5
1.4	产品架构	6
1.5	应用场景	7
2	快速入门	10
2.1	创建集群	10
2.2	创建容器应用	13
2.3	应用运维	15
3	操作指南	16
3.1	控制台说明	16
3.2	使用流程	17
3.3	环境设置	18
3.3.1	创建虚拟私有云	18
3.3.2	创建密钥对	18
3.4	资源准备	19
3.4.1	集群管理	19
3.4.2	节点管理	21
3.4.3	命名空间	21
3.4.4	网络管理	23
3.4.5	配置中心	26
3.5	创建应用	32
3.5.1	通过镜像创建无状态容器应用	32

3.5.2	容器组	35
3.5.3	设置应用生命周期	38
3.5.4	设置应用调度策略	39
3.5.5	设置应用访问策略（内部访问）	41
3.5.6	为应用挂载数据卷	42
3.6	应用管理运维	45
3.6.1	升级应用	45
3.6.2	应用监控	45
3.6.3	启停应用	46
3.6.4	删除应用	46
3.7	镜像仓库	46
3.7.1	基本概念	47
3.7.2	创建容器镜像仓库	47
3.7.3	上传私有镜像（外网上传）	48
3.7.4	上传私有镜像（内网上传）	50
3.7.5	删除镜像仓库	51
3.7.6	删除镜像仓库中的镜像	52
3.7.7	查看镜像仓库地址	52
4	常见问题	53
4.1	云容器引擎常见的使用场景如下？	53
4.2	什么是环境变量？	53
4.3	集群删除之后相关数据能否再次找回？	53
4.4	节点状态与应用状态不一致？	53
4.5	在已有的节点创建应用时未重新拉取镜像？	53
4.6	如何绑定弹性IP？	54
4.7	创建容器应用有哪几种方式？有什么区别？	54
4.8	镜像、容器、应用的关系是什么？	54
5	参考知识	56
5.1	ConfigMap配置项要求	56

1 产品介绍

1.1 产品定义

云容器引擎(CT-CCE, Cloud Container Engine)提供高性能可扩展的容器服务,基于云主机快速构建高可靠的容器集群,兼容 Kubernetes 及 Docker 容器生态,帮助用户轻松创建和管理多样化的容器应用,并提供容器故障自愈,监控日志采集,自动弹性扩容等高效运维能力。

1.2 Docker 基础知识

1.2.1 Docker 基本概念

Docker 是一个开源项目,诞生于 2013 年初,最初是 dotCloud 公司内部的一个业余项目。它基于 Google 公司推出的 Go 语言实现,基于 Linux 内核的 cgroup, namespace, 以及 AUFS 类的 Union FS 等技术,对进程进行封装隔离,属于操作系统层面的虚拟化技术。由于隔离的进程独立于宿主和其它的隔离的进程,因此也称其为容器。最初实现是基于 LXC,从 0.7 以后开始去除 LXC,转而使用自行开发的 libcontainer,从 1.11 开始,则进一步演进为使用 runC 和 containerd。

Docker 在容器的基础上,进行了进一步的封装,从文件系统、网络互联到进程隔离等等,极大的简化了容器的创建和维护。使得 Docker 技术比虚拟机技术更为轻便、快捷。

传统虚拟机技术是虚拟出一套硬件后,在其上运行一个完整操作系统,在该系统上再运行所需应用进程;而容器内的应用进程直接运行于宿主的内核,容器内没有自己的内核,而且也没有进行硬件虚拟。因此容器要比传统虚拟机更为轻便。

Docker 包括三个基本概念:

1) 镜像: Docker 镜像是一个特殊的文件系统,除了提供容器运行时所需的程序、库、资源、配置等文件外,还包含了一些为运行时准备的一些配置参数(如匿名卷、环境变量、用户等)。镜像不包含任何动态数据,其内容在构建之后也不会被改变。

2) 容器: 镜像(Image)和容器(Container)的关系,就像是面向对象程序设计中的类和实例一样,镜像是静态的定义,容器是镜像运行时的实体。容器可以被创建、启动、停止、删除、暂停等。

3) 仓库: 镜像构建完成后,可以很容易的在主机上运行。但是,如果需要在其它服务器上使用这个镜像,我们就需要一个集中的存储、分发镜像的服务,仓库就是这样的服务,一个仓库中可以包含多个镜像。

更多 Docker 基础知识请参见 <https://docs.docker.com/>。

1.2.2 安装 Docker

所有通过天翼云云容器引擎创建或添加的节点，系统都会自动为节点安装 1.11.2 版本的 Docker，无需用户手动安装。若需要了解更多 Docker 的安装信息，请参阅 [Docker 安装指南](#)。

1.2.3 制作 Docker 镜像

本节指导用户通过 Dockerfile 定制一个简单的 Web 应用程序的 Docker 镜像。

若使用官方的 Nginx 镜像来创建容器应用，在浏览器访问时则会看到默认的 Nginx 欢迎页面，本节以 Nginx 镜像为例，修改 Nginx 镜像的欢迎页面，定制一个新的镜像，将欢迎页面改为“Hello, CCE!”。具体的操作步骤如下：

- 1) 以 root 用户登录 Docker 所在的云主机；
- 2) 创建一个名为 Dockerfile 的文件；

```
mkdir mynginx
cd mynginx
touch Dockerfile
```

- 3) 编辑 Dockerfile；

```
vi Dockerfile
```

文件内容如下：

```
FROM nginx
RUN echo '<h1>Hello,CCE!</h1>' > /usr/share/nginx/html/index.html
```

其中：

FROM 语句：表示使用 nginx 镜像作为基础；

RUN 语句：表示执行 echo 命令，在显示器中显示一段 Hello, CCE 的文字；

- 4) 构建 Docker 镜像

```
docker build -t nginx:v3
```

- 5) 执行以下命令，可查看到已成功部署的 nginx 镜像，版本为 v3；

```
docker image
```

1.3 产品优势

云容器引擎是基于业界主流的 Docker 和 Kubernetes 开源技术构建的容器服务，提供众多契合企业大规模容器集群场景的功能，在系统可靠性、高性能、开源社区兼容性等多个方面具有独特的优势，满足企业在构建容器云方面的各种需求。

简单易用

- 通过WEB界面一键创建Kubernetes集群，无需再自行搭建Docker和Kubernetes。
- 一站式自动化部署和运维容器应用，整个生命周期都在容器服务内一站式完成。
- 通过Web界面轻松实现集群节点监控和容器应用的扩容和缩容，自由组合策略以应对多变的突发浪涌。

安全可靠

- 高可靠：集群控制面支持3 Master高可用，当其中某个或者两个控制节点故障时，集群依然可用，从而保障您的业务高可用。

开放兼容

- 云容器引擎在Docker技术的基础上，为容器化的应用提供部署运行、资源调度和动态伸缩等一系列完整功能，提高了大规模容器集群管理的便捷性。
- 云容器引擎基于业界主流的Kubernetes实现，完全兼容Kubernetes/Docker社区原生版本，与社区最新版本保持紧密同步，完全兼容Kubernetes API。

与天翼云其它资源整合

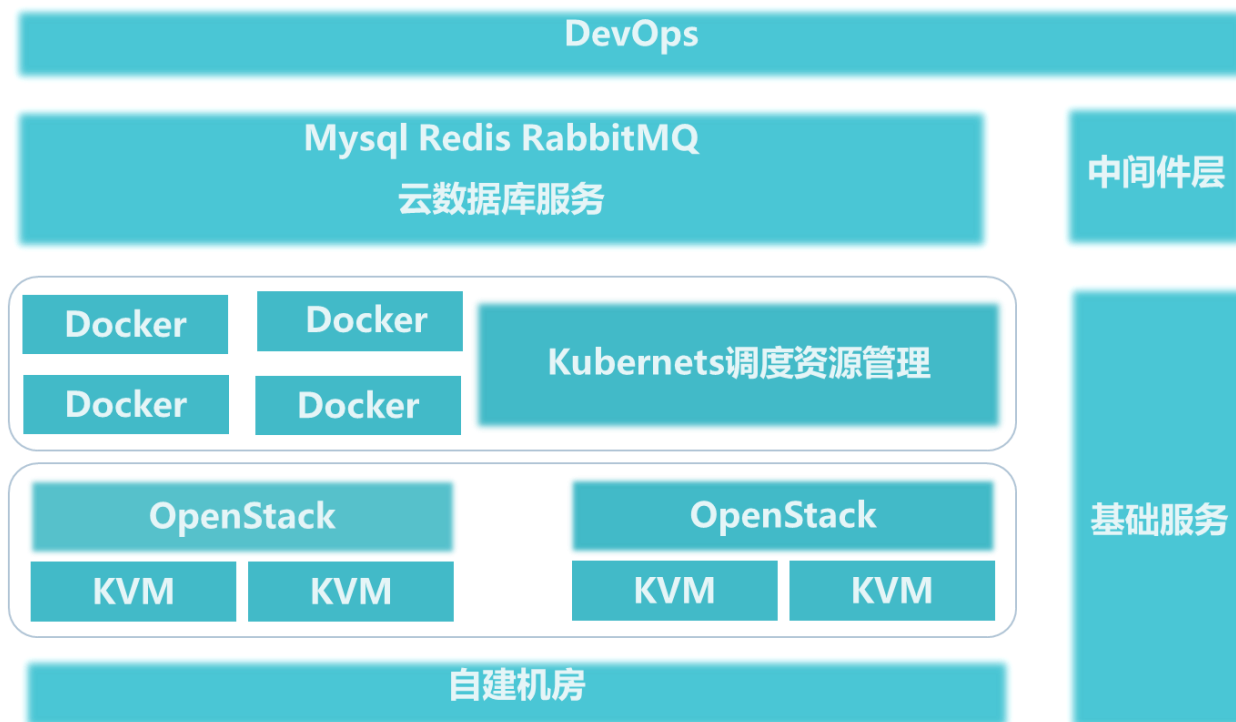
- 支持在容器中使用云主机、VPC、EIP、云硬盘等资源。

1.4 产品架构

天翼云容器平台，作为电信云平台未来发展的重要组成部分，旨在通过容器化先进可靠的技术方案，以平台化、产品化的思维方式，在网络、系统、通用解决方案、高可用、高性能等多方面对电信云平台进行全面增强，为上层应用业务提供更可靠的服务。云容器能够：

- 大幅提高线上业务的容灾、故障恢复、弹性伸缩能力。
- 大幅提高资源、设备及人力的利用率，降低成本。

- 大幅丰富了产品、提高了客户、运营部门满意度。
- 大幅提升产品、业务应用的研发效率



1.5 应用场景

弹性伸缩架构

应用场景：

- 电商客户遇到促销、限时秒杀等活动期间，访问量激增，需及时、自动扩展云计算资源。
- 视频直播客户业务负载变化难以预测，需要根据CPU/内存使用率进行实时扩缩容。
- 游戏客户每天中午12点及晚上18:00-23:00间需求增长，需要定时扩容。

价值：

云容器引擎可根据用户的业务需求预设策略自动调整计算资源，使云服务器或容器数量自动随业务负载增长而增加，随业务负载降低而减少，保证业务平稳健康运行，节省成本。

优势：

- 自由灵活

支持多种策略配置，业务流量达到扩容指标，秒级触发容器扩容操作。

- 高可用

自动检测伸缩组中实例运行状况，启用新实例替换不健康实例，保证业务健康可用。

- 低成本

只按照实际用量收取云服务器费用。

DevOps持续交付

应用场景：

当前IT行业发展日益快速，面对海量需求必须具备快速集成的能力。经过快速持续集成，才能保证不间断的补全用户体验，提升服务质量，为业务创新提供源源不断的动力。大量交付实践表明，不仅传统企业，甚至互联网企业都可能在持续集成方面存在研发效率低、工具落后、发布频率低等方面的问题，需要通过持续交付提高效率，降低发布风险。

价值：

云容器引擎搭配容器镜像服务提供DevOps持续交付能力，能够基于代码源自动完成代码编译、镜像构建、灰度发布、容器化部署，实现一站式容器化交付流程，并可对接已有CI/CD，完成传统应用的容器化改造和部署。

优势：

- 高效流程管理

优化流程交互设计，脚本编写量较传统CI/CD流水线减少80%以上，让CI/CD管理更高效。

- 高性能

全容器化架构设计，任务调度更灵活，执行效率更高。

微服务治理

应用场景：

伴随着互联网技术的不断发展，各大企业的系统越来越复杂，传统的系统架构越来越不能满足业务的需求，取而代之的是微服务架构。微服务是将复杂的应用切分为若干服务，每个服务均可以独立开发、部署和伸缩；微服务和容器组合使用，可进一步简化微服务的交付，提升应用的可靠性和可伸缩性。

随着微服务的大量应用，其构成的分布式应用架构在运维、调试、和安全管理等维度变得更加复杂，在管理微服务时，往往需要在业务代码中添加微服务治理相关的代码，导致开发人员不能专注于业务开发，还需要考虑微服务治理的解决方案，并且将解决方案融合到其业务系统中。

价值：

云容器引擎深度集成应用服务网格，提供开箱即用的应用服务网格流量治理能力，用户无需修改代码，即可实现灰度发布、流量治理和流量监控能力。

优势：

- 开箱即用

与云容器引擎无缝对接，一键开启后即可提供非侵入的智能流量治理解决方案。

- 策略化智能路由

无需修改代码，即可实现HTTP、TCP等服务连接策略和安全策略。

2 快速入门

创建一个完整的容器应用，可以通过“选择开源镜像”、“上传并选择私有镜像”两种方式实现，后续还将支持通过“应用模板”进行应用创建。

- 选择开源镜像：基于开源 docker 镜像创建容器应用，开源docker镜像由天翼云官方镜像和Dockerhub官方镜像来提供，无需上传私有镜像。
- 上传并选择私有镜像：您可基于业务需求制作私有 docker 镜像，上传到 CCE。基于该私有镜像创建容器应用。
- 选择应用模板：云容器引擎提供全容器化中间件的 Helm 模板，可一键完成中间件的部署。当前提供的官方模板有 redis、etcd、mysql-ndb。

本章节以“选择开源镜像”的方式创建应用为例，来创建一个 nginx 容器应用。nginx 是一款轻量级的 Web 服务器，您可通过云容器引擎 CCE 快速创建 nginx 容器应用，搭建 nginx web 服务器。

说明：节点和应用运行过程中会产生费用，建议您参照本章节创建应用后，删除应用和节点，避免费用产生。

本章节执行完成后，可成功访问 nginx 的网页，如下图：

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.


For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

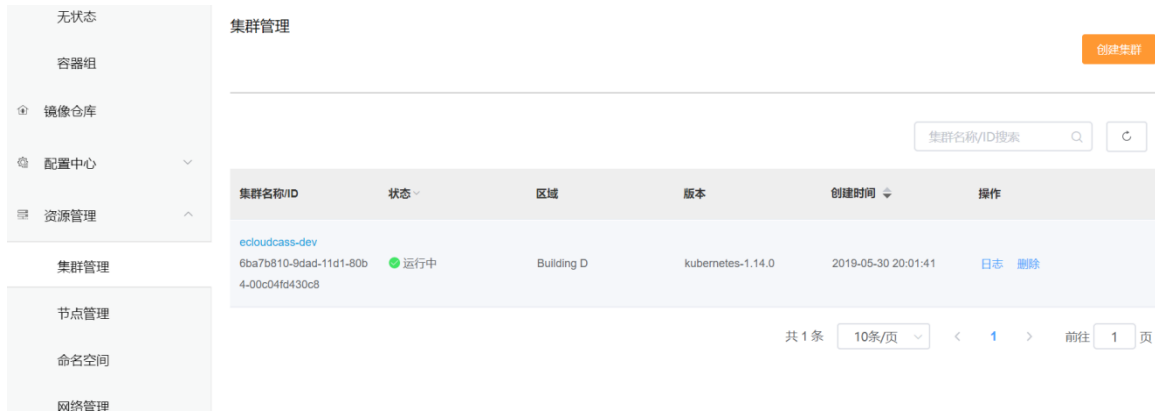
Thank you for using nginx.

2.1 创建集群

集群是运行应用的逻辑分组，包含一组云主机资源，每个节点对应一台云主机。首次使用 CCE 时，您需要创建一个初始集群，并添加一个节点。

具体的操作步骤如下：

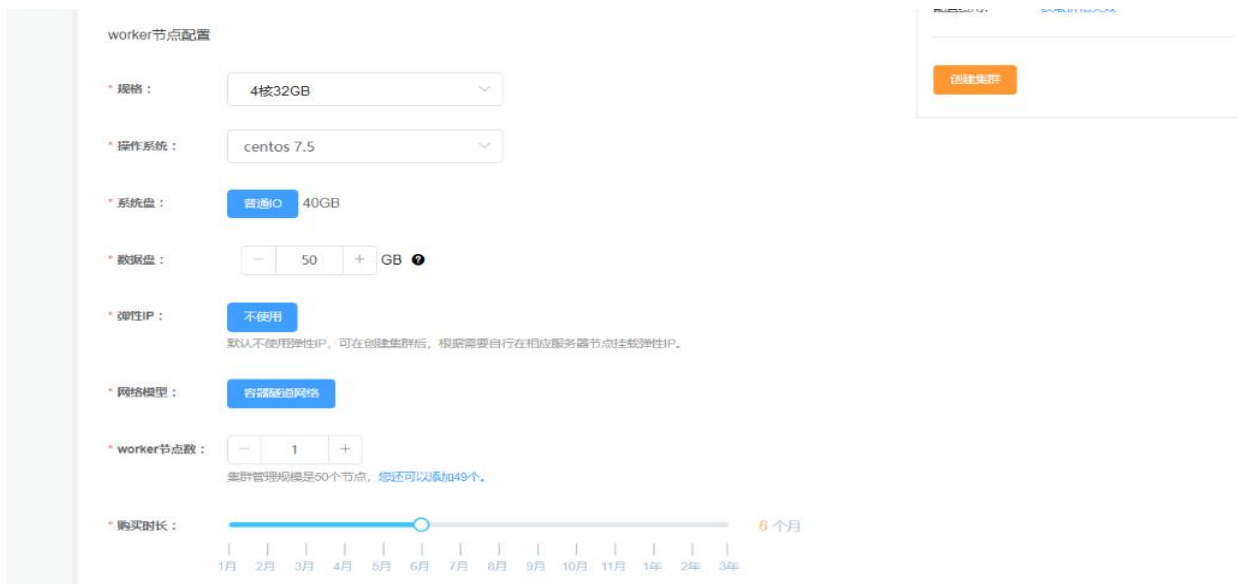
- 1) 注册并登录天翼云 <http://www.ctyun.cn> ;
- 2) 选择控制中心, 切换到具体节点, 如杭州节点;
- 3) 单击  , 选择【计算】【云容器引擎】, 进入容器服务界面;
- 4) 在资源管理中的集群管理页面, 单击【创建集群】按钮;



- 5) 输入集群名称 `cluster1`, 其余参数保持默认。若未创建虚拟私有云和子网, 请参见 3.3.1 创建虚拟私有云;



- 系列: 分为CAAS-基础版、CAAS-高级版。
基础版创建1个master节点和相应worker节点。高级版创建3个master节点和相应worker节点;
 - 容器网段: 默认是10.128.0.0/14;
 - 服务网段: 默认是172.30.0.0/14;
- 6) worker节点的配置。除规格和操作系统外, 其余参数保持默认:



- 规格：为2核4GB 、4核8GB 、8核32GB 、16核64GB，根据具体情况选择规格；
 - 操作系统：默认为centos7.5，根据具体情况选择规格；
 - 系统盘：类型默认为普通IO 容量为50GB；
 - 数据盘：类型可以为普通IO、高IO、超高IO。容量默认为50GB(最小容量)，最大容量为2048GB，步长10GB。
 - Worker节点数：默认为1个节点。最多可以设置50个节点。
 - 购买时长：最低为1个月，最高为3年。
- 7) 查看右侧订单详情无误后，单击【创建集群】。
 集群创建预计需要 6-10 分钟。请根据界面提示查看集群创建过程。创建完成后，单击【返回集群管理】链接。
- 8) 单击【资源管理】【节点管理】中的节点，进入节点详情页，获取并记录节点的弹性IP，用于应用访问；



2.2 创建容器应用

应用指运行在CCE上的一组实例。CCE提供第三方应用托管功能，提供从部署到运维全生命周期管理。本节指导用户通过容器镜像创建您的第一个容器应用。

前提条件：

- 已创建集群并添加节点；
- 节点已绑定弹性IP，并已获取节点的弹性IP地址。创建集群后，需要自行在相应的服务器节点挂载弹性IP，表示该节点已绑定弹性IP；

操作步骤：

- 1) 登录天翼云控制中心云容器引擎页面；
- 2) 单击左侧导航栏的【应用管理】【无状态】；



- 3) 在【无状态】页下，单击【创建应用】按钮；
- 4) 按照下表，输入相应参数，其它保持默认：

参数	说明
应用名称	nginx
部署集群	2.1.1 中创建的集群Cluster1
实例数量	1
命名空间	default

- 5) 单击【下一步】进入容器设置页面。单击【选择镜像】按钮，在【天翼云官方镜像】或【Dockerhub官方镜像】页签中，搜索nginx的镜像，勾选后单击【确定】；
- 6) 单击【下一步】；
- 7) 单击【添加服务】，参照下表，输入应用访问配置参数。本例中，将 nginx 应用设置为被集群内访问；

参数	说明
服务名称	输入应用发布的可被同个集群内的其他应用访问的名称，设置为：nginx
访问方式	集群内访问
容器端口	容器中应用启动监听的端口，nginx 镜像请设置为：81
访问端口	设置为8080
协议	TCP

添加服务
×

* 服务名称

访问方式 集群内访问
表示应用可以被同个集群内的其它应用访问

* 端口映射

协议	容器端口	访问端口	操作
TCP	81	8080	删除

[+ 添加端口配置](#)

确定
取消

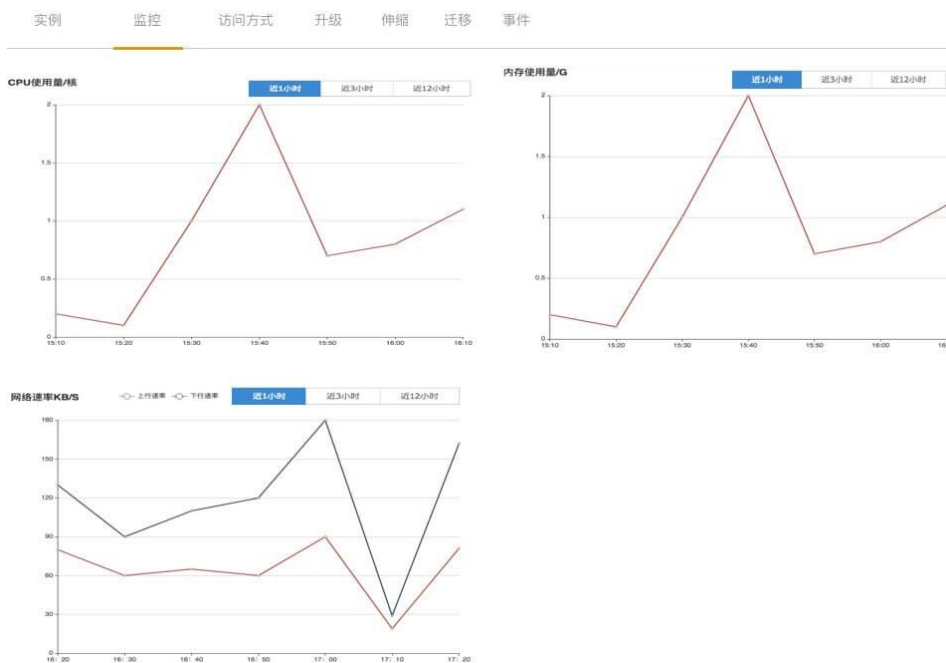
- 8) 单击【下一步】，高级设置不需要配置，单击【提交】。应用创建完成后，在应用管理列表中可查看到运行中的应用。

2.3 应用运维

应用创建完成后，您可以查看应用的运行状态、指标趋势等。

操作步骤：

- 1) 登录天翼云控制中心云容器引擎页面；
- 2) 单击左侧导航栏的【应用管理】【无状态】；
- 3) 单击新建的nginx应用，进入应用详情页面；
- 4) 在监控页签下，可查看到应用实例的CPU使用量、内存使用量、网络输入输出速率。只有实例处于运行状态时，才可以查看 CPU 使用量、内存使用量。



3 操作指南

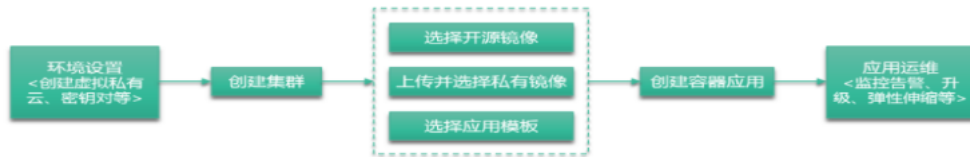
3.1 控制台说明

登录云容器引擎CCE后，可在总览界面查看使用CCE的相关信息，包括应用、节点的数量、状态等信息。页面左侧导航是CCE服务的目录，包括总览、应用管理、资源管理、镜像仓库和配置中心。

- 总览：总览界面查看使用CCE的相关信息，包括应用、节点的数量、状态等信息；
- 应用管理：
 - 无状态：应用是可部署的软件实体，包含一个或一组容器。平台能够为无状态容器应用选取合适的资源进行安装部署，并支持运行管理操作，如回退、伸缩、升级、删除等；
 - 容器组：是Kubernetes部署应用或服务的最小的基本单位。一个容器组封装多个应用容器、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。在容器组列表中可以对容器组进行监控、删除等。
- 资源管理：
 - 虚拟机集群：集群是计算资源的集合，包含一组节点资源，容器运行在节点上。在创建容器应用前，您需要存在一个可用集群；
 - 节点管理：节点是指接入到平台的计算资源，包括虚拟机等。用户需确保所在项目节点资源充足，若节点资源不足，会导致创建应用等操作失败；
 - 命名空间：创建多个虚拟的空间，在集群用户数量较多时，多个命名空间可以有效划分工作区间，将集群资源划分为多个用途；
 - 网络管理：包括服务和ingress。Kubernetes中的Service对象就是用来解决上述Pod访问问题的，Ingress就是为进入集群的请求提供路由规则的集合；
- 镜像仓库：镜像仓库是用于存储、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；
- 配置中心：
 - 配置项：是一种用于存储应用所需配置信息的资源类型，内容由用户决定。资源创建完成后，可在容器应用中加载使用。例如，在“数据卷”中加载资源文件，使其成为容器中的文件，或者在“环境变量”中加载，使其成为容器中的环境变量；
 - 私密凭据：是一种用于存储应用所需要认证信息、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，可在容器应用中作为文件或者环境变量使用。

3.2 使用流程

完整的容器服务使用流程包括以下步骤：




使用流程	说明
1. 环境设置	<p>创建集群前，您需要进行必要的环境设置。</p> <p>说明：如果用户已有“虚拟私有云”和“密钥对”，可重复使用，不需多次创建。</p> <ul style="list-style-type: none"> 创建虚拟私有云，提供一个隔离的、用户自主配置和管理的虚拟网络环境，提升公有云中资源的安全性，简化用户的网络部署； 创建密钥对，用于远程登录节点的身份认证；
2. 创建集群	CCE支持创建Kubernetes集群（即虚拟机集群），后续还将提供裸机集群。
3. 选择部署方式	<p>CCE支持两类部署方式，用户可基于自身需求选择。</p> <ul style="list-style-type: none"> 选择开源镜像：基于开源docker镜像创建容器应用，无需上传私有镜像。如：天翼云官方镜像、dockerhub官方镜像。 上传并选择私有镜像：您可基于业务需求制作私有docker镜像，上传到CCE。基于该私有镜像创建容器应用。
4. 创建容器应用	<p>CCE支持无状态容器应用。</p> <ul style="list-style-type: none"> 无状态应用：在运行中始终不保存任何数据或状态，例如nginx。
5. 应用运维	CCE支持容器应用监控、日志，提供全生命周期管理能力。

3.3 环境设置


3.3.1 创建虚拟私有云

您需要创建虚拟私有云(Virtual Private Cloud, 以下简称VPC), 提供一个隔离的、用户自主配置和管理的虚拟网络环境, 提升公有云中资源的安全性, 简化用户的网络部署。如果用户已有VPC, 可重复使用, 不需多次创建。

- 1) 注册并登录天翼云<http://www.ctyun.cn>;
- 2) 选择控制中心, 切换到具体节点, 如杭州节点;
- 3) 单击 , 选择【网络】【虚拟私有云】, 进入虚拟私有云页面;
- 4) 单击【创建虚拟私有云】, 进行虚拟私有云的创建;
- 5) 根据界面提示创建虚拟私有云。如无特殊需求, 界面参数均可保持默认; 关于创建VPC的详细信息可以参考《虚拟私有云用户使用指南》。

3.3.2 创建密钥对

为安全起见, 节点在登录时需使用密钥方式进行身份验证。因此, 您需要新建一个密钥对, 用于作为添加节点时的身份验证。如果用户已有密钥凭据, 可重复使用, 不需多次创建。

- 1) 登录天翼云 <http://www.ctyun.cn>;
- 2) 选择控制中心, 切换到具体节点, 如杭州节点;
- 3) 单击 , 选择【计算】【弹性云主机】;
- 4) 选择左侧导航中的【密钥对】, 单击【创建密钥对】;
- 5) 输入密钥对名称后, 单击【确定】;
- 6) 在弹出的对话框中, 单击【确定】;
- 7) 请根据提示信息, 查看并保存私钥。为保证安全, 私钥只能下载一次, 请妥善保管, 否则将无法登录节点。


3.4 资源准备

3.4.1 集群管理

3.4.1.1 创建虚拟机集群

集群是计算资源的集合，包含一组节点资源，容器运行在节点上。在创建容器应用前，您需要存在一个可用集群。目前最多可以创建5个集群，请确保至少有一个可用的集群。如集群数量不够，可申请增加配额。

操作步骤：

- 1) 登录天翼云 <http://www.ctyun.cn> ；
- 2) 选择控制中心，切换到具体节点，如杭州节点；
- 3) 单击 ，选择【计算】【云容器引擎】，进入容器服务界面；
- 4) 单击【资源管理】【集群管理】，点击右侧界面的【创建集群】按钮；
- 5) 参照下表设置新增集群参数，其中带“*”的参数为必填参数；

参数	参数说明
基础配置	
*集群名称	新建集群的名称。
*虚拟私有云	新建集群所在的虚拟私有云。若没有可选虚拟私有云，单击“创建虚拟私有云”进行创建，操作步骤请参见 3.3.1 创建虚拟私有云。
*子网	节点虚拟机运行的子网环境。
*系列	分为CAAS-基础版、CAAS-高级版（基础版集群将创建1个master节点和相应控制节点，worker节点配置由用户自由选择；高级版集群将创建3个master节点和相应控制节点，worker节点配置由用户自由选择。）
*容器CIDR	默认是10.128.0.0/14，确定容器网段后，容器实例将在规划的网段内分配IP。不能与集群所选的VPC重复，当vpc和容器网段或者服务网段冲突提示：默认网段与所选虚拟私有云网段有冲突，请重新选择vpc。
*服务CIDR	默认是172.30.0.0/14，服务网段为kubernetes service ip网段，请根据业务需求选择该网段。不能与集群所选的VPC重复。


参数	参数说明
worker节点配置	
*规格	请根据业务需求选择相应的节点规格。
*操作系统	请选择节点对应的操作系统。
*系统盘	默认为普通IO, 50GB。
*数据盘	普通IO、高IO、超高IO, 规格为[50, 2048]GB, 默认50GB; 范围50-2048GB; 整数; 填入小数自动四舍五入为整数, 输入为空自动设置为50; 步长为10。
*弹性IP	独立申请的公网IP地址。 暂不使用: 不使用弹性IP的云服务器不能和互联网互通, 可以到相应云服务器节点绑定弹性IP。
*网络模型	容器隧道网络: 基于底层 VPC 网络, 构建独立的容器 VXLAN 隧道化的容器网络, 与底层网络解耦, 支持高扩展性, 网络策略, 多容器网络平面, 可支持 500+节点集群规模。
*worker节点数	默认最小值是1, 只允许填入整数; 步长为1; 设置的节点数不能超过最大节点规模50; 输入为空自动设置为1。
*购买时长	根据需求, 设置购买时长

6) 确查看订单详情无误, 确认费用后, 单击【创建集群】。集群创建预计需要 6-10 分钟。请根据界面提示查看集群创建过程。

3.4.1.2 删除集群

删除集群会将集群内的节点以及运行的应用和服务都销毁, 请谨慎操作。

操作步骤:

- 1) 控制中心中, 单击 , 选择【计算】【云容器引擎】进入容器服务界面;
- 2) 单击【资源管理】【集群管理】;
- 3) 单击待删除集群后的【删除】。根据系统提示进行删除操作;




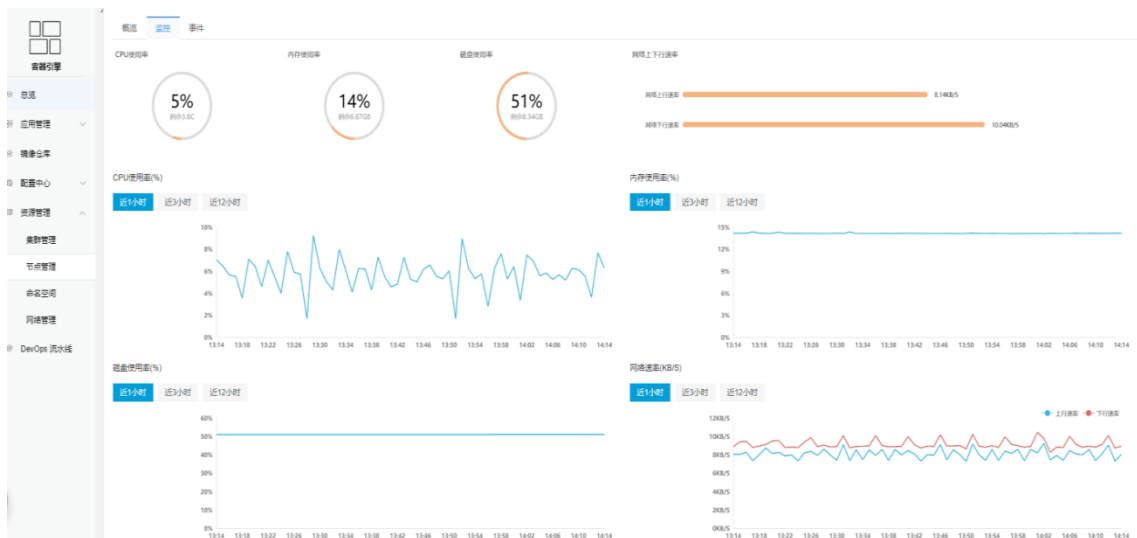
3.4.2 节点管理

3.4.2.1 节点监控

CCE 提供节点运行状态监控，包括所有节点的资源占用情况、每个节点的资源占用情况。

操作步骤：

- 1) 登录天翼云 <http://www.ctyun.cn>；
- 2) 选择控制中心，切换到具体节点，如杭州节点；
- 3) 单击 ，选择【计算】【云容器引擎】进入容器服务界面；
- 4) 单击【资源管理】【节点管理】；
- 5) 单击待监控的节点名称，进入节点详情；
- 6) 选择监控页签，可查看 CPU 使用率、内存使用率、磁盘使用率和网络速率的情况；




3.4.3 命名空间

3.4.3.1 创建命名空间

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项目，不同项目中的数据彼此隔离。使得它们既可以共享同一个集群的服务，也能够互不干扰。

命名空间按创建类型分为两大类：集群默认创建的、用户自己创建的。集群启动时默认创建default、kube-public、kube-system命名空间。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。

操作步骤：

- 1) 登录天翼云 <http://www.ctyun.cn>；
- 2) 选择控制中心，切换到具体节点，如杭州节点；
- 3) 单击 ，选择【计算】【云容器引擎】进入容器服务界面；
- 4) 单击【资源管理】【命名空间】；
- 5) 单击【创建云硬盘存储】；



- 6) 配置命名空间基本信息：

参数	参数说明
*名称	新建命名空间的名称。
*集群	命名空间所在的集群。
描述	新建命名空间描述信息。



- 7) 单击【确定】，等待创建成功；
- 8) 创建成功后，命名空间列表中会出现已创建的命名空间，状态为“可用”，表示命名空间已创建成功；
- 9) 在命名空间列表中选择需要删除的命名空间，单击操作列的【删除】按钮，可以删除命名空间。

3.4.4 网络管理

3.4.4.1 创建服务

云容器引擎为满足多种复杂场景下应用间的互相访问，提供了不同的访问方式，从而满足不同场景提供不同访问通道。本平台暂时支持的服务类型只包括：集群内访问（ClusterIP）。表示应用暴露给同一集群内其他应用访问的方式，可以通过“集群内部域名”访问。

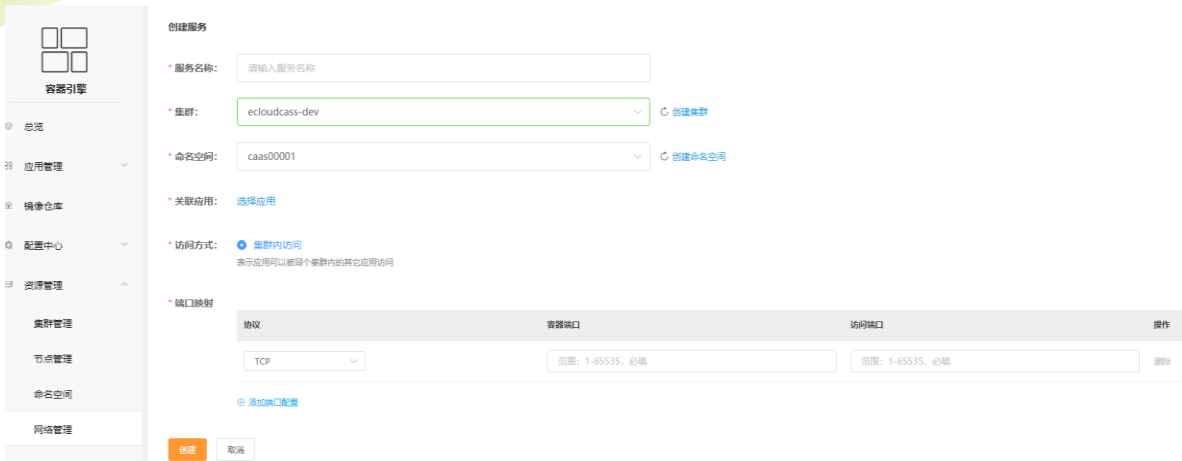
操作步骤：

- 1) 在控制中心中，选择【计算】【云容器引擎】，进入容器服务界面；
- 2) 单击左侧导航栏的【资源管理】【网络管理】，单击【创建服务】；



- 3) 参照下表设置新增服务参数，其中带“*”的参数为必填参数；

*服务名称	新建服务的名称。
*集群	服务所在集群。若没有可选集群，单击“创建集群”进行创建，操作步骤请参见3.4.1.1创建集群
*命名空间	服务所在命名空间。若没有可选命名空间，单击“创建命名空间”进行创建，操作步骤请参见3.4.3.1创建命名空间
*关联应用	选择需要添加Service的工作负载。
*访问方式	默认选择“集群内访问（ClusterIP）”。
*端口映射	协议：请根据业务的协议类型选择TCP、UDP。 容器端口：应用程序实际监听的端口。 访问端口：容器端口映射到集群虚拟IP上的端口，用虚拟IP访问应用时使用。 添加端口配置：点击添加端口配置新增一行端口映射。 删除：删除端口配置。一个端口配置不允许删除，删除后置灰。



创建服务

* 服务名称:

* 集群: 创建集群

* 命名空间: 创建命名空间

* 关联应用: [选择应用](#)

* 访问方式: 集群内访问
表示应用可以被同个集群内的其它应用访问

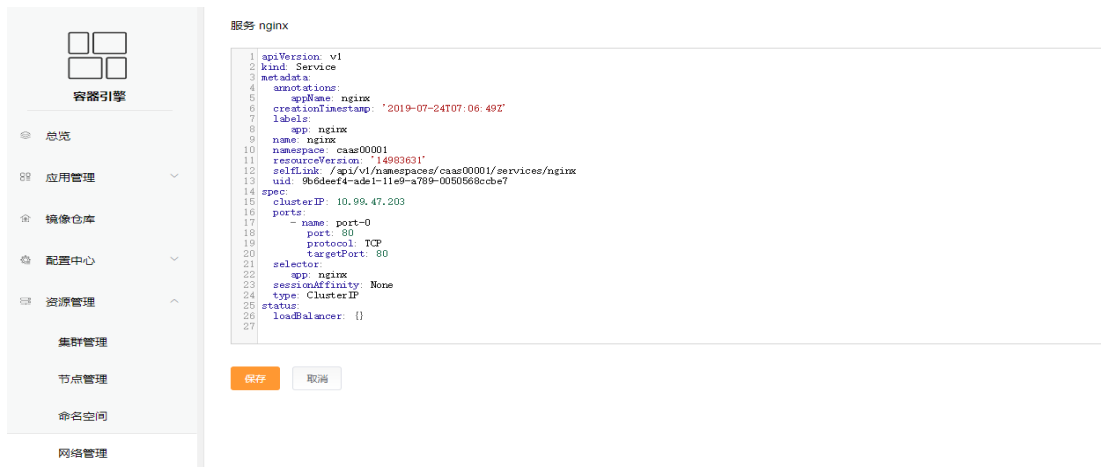
* 端口映射

协议	容器端口	访问端口	操作
TCP	<input type="text" value="范围: 1-65535, 必填"/>	<input type="text" value="范围: 1-65535, 必填"/>	<input type="button" value="删除"/>

[添加端口配置](#)

4) 单击【创建】。等待创建成功；

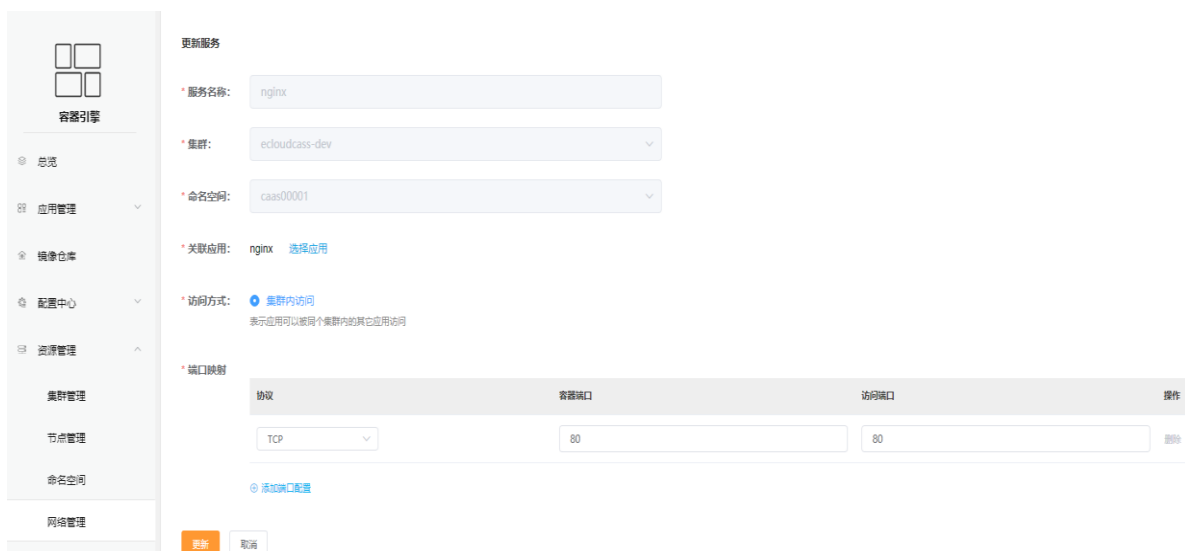
5) 创建成功后，服务列表中会出现已创建的服务，对已经创建的服务可以编辑YAML、更新、删除的操作。



服务 nginx

```

1 apiVersion: v1
2 kind: Service
3 metadata:
4   annotations:
5     appName: nginx
6     creationTimestamp: "2019-07-24T07:06:49Z"
7   labels:
8     app: nginx
9   name: nginx
10  namespace: caas00001
11  resourceVersion: "14983631"
12  selfLink: /api/v1/namespaces/caas00001/services/nginx
13  uid: 0b6deef4-ade1-11e9-a789-0050568cbe7
14 spec:
15   clusterIP: 10.99.47.203
16   ports:
17     - name: port-0
18       port: 80
19       protocol: TCP
20       targetPort: 80
21   selector:
22     app: nginx
23   sessionAffinity: None
24   type: ClusterIP
25 status:
26   loadBalancer: {}
27
    
```



更新服务

* 服务名称:

* 集群:

* 命名空间:

* 关联应用: [nginx 选择应用](#)

* 访问方式: 集群内访问
表示应用可以被同个集群内的其它应用访问

* 端口映射

协议	容器端口	访问端口	操作
TCP	<input type="text" value="80"/>	<input type="text" value="80"/>	<input type="button" value="删除"/>

[添加端口配置](#)

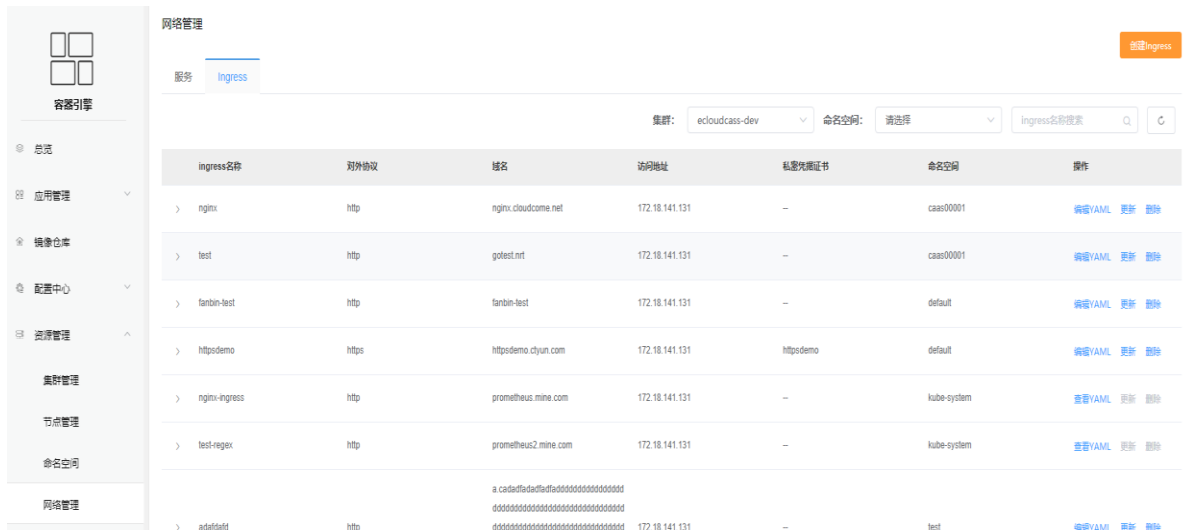
3.4.4.2 创建 Ingress

七层负载均衡支持URL配置，通过对应的URL将访问流量分发到对应的服务。同时，服务根据不同URL实现不同的功能。

Ingress七层负载由公网服务地址、访问端口、URL组成，例如：10.117.117.117:80/helloworld。

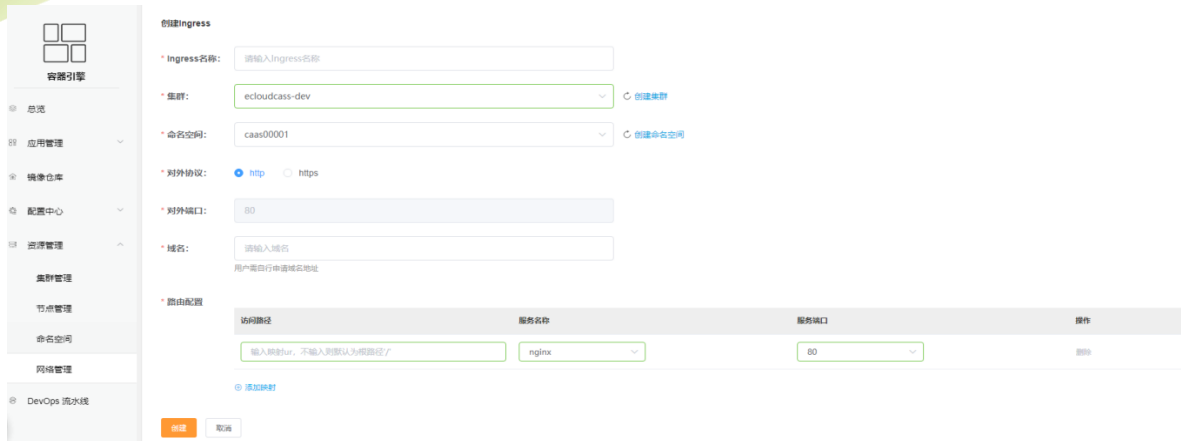
操作步骤：

- 1) 在控制中心中，选择【计算】【云容器引擎】，进入容器服务界面；
- 2) 单击左侧导航栏的【资源管理】【网络管理】，单击【创建Ingress】；



- 3) 参照下表设置新增Ingress参数，其中带“*”的参数为必填参数；

*Ingress名称	新建Ingress的名称。
*集群	Ingress所在集群。若没有可选集群，单击“创建集群”进行创建，操作步骤请参见3.4.1.1创建集群。
*命名空间	Ingress所在命名空间。若没有可选命名空间，单击“创建命名空间”进行创建，操作步骤请参见3.4.3.1创建命名空间。
*对外协议	支持HTTP和HTTPS。若选择HTTPS，SSL策略中选择SSL终端，请填写私密凭据，无合适的私密凭据，可点击创建IngressTLS类型的私密凭据，跳转到私密凭据创建页；SSL策略选择SSL穿透。
*对外端口	开放公网地址的端口，在对外协议中选择HTTP时，为80；HTTPS时，为443。
*域名	实际访问的域名地址，对应负载均衡服务域名地址，需用户购买备案自己的域名，可选填。一旦配置了域名规则，则必须使用域名访问。
*路由配置	访问路径：需要注册的访问路径，例如：/healthz。 服务名称：选择需要添加Ingress的服务。 容器端口：容器镜像中容器实际监听端口，需用户确定。 路由配置只有一项时不可删除。



创建Ingress

* Ingress名称:

* 集群: [创建集群](#)

* 命名空间: [创建命名空间](#)

* 对外协议: http https

* 对外端口:

* 域名:

用户需自行申请域名地址

* 路由配置


访问路径	服务名称	服务端口	操作
<input type="text" value="输入路径, 不输入则默认为根路径"/>	<input type="text" value="nginx"/>	<input type="text" value="80"/>	删除

[添加映射](#)

4) 单击【创建】。等待创建成功；

5) 创建成功后，服务列表中会出现已创建的Ingress，对已经创建的服务可以进行编辑YAML、更新、

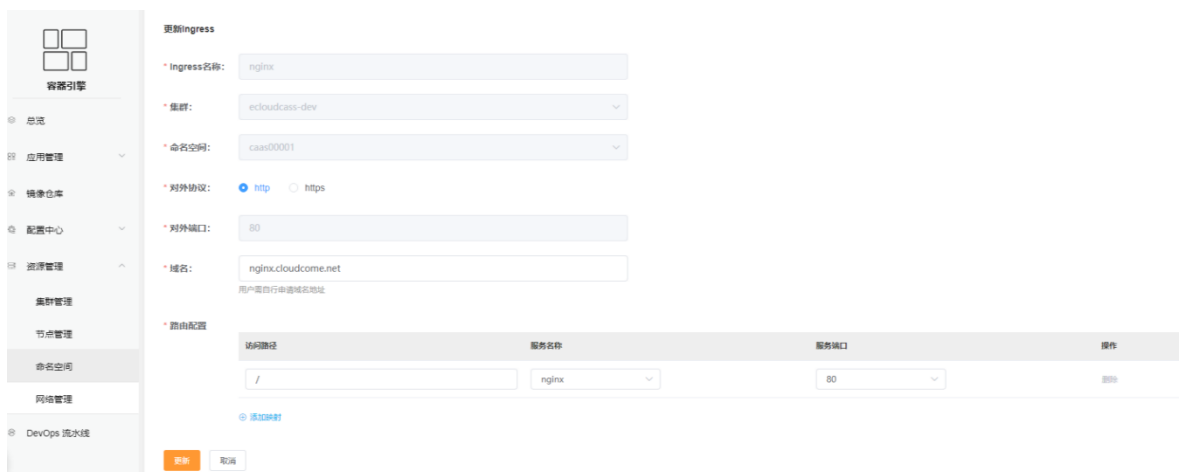
删除的操作。



ingresses nginx

```

1 apiVersion: extensions/v1beta1
2 kind: Ingress
3 metadata:
4   annotations:
5     ingress.kubernetes.io/ssl-redirect: 'false'
6   creationTimestamp: '2019-07-24T07:08:25Z'
7   generation: 0
8   name: nginx
9   namespace: caas00001
10  resourceVersion: '14991977'
11  selfLink: /apis/extensions/v1beta1/namespaces/caas00001/ingresses/nginx
12  uid: d49c4c61-ade1-11e9-a709-0050568cbe7
13 spec:
14   host: nginx.cloudcome.net
15   http:
16     paths:
17     - backend:
18         service: nginx
19         servicePort: 80
20       path: /
21   status:
22     loadBalancer: {}
23
24
    
```



更新Ingress

* Ingress名称:

* 集群:

* 命名空间:

* 对外协议: http https

* 对外端口:

* 域名:

用户需自行申请域名地址

* 路由配置

访问路径	服务名称	服务端口	操作
<input type="text" value="/"/>	<input type="text" value="nginx"/>	<input type="text" value="80"/>	删除

[添加映射](#)

3.4.5 配置中心

3.4.5.1 创建配置项

配置项是一种用于存储应用所需配置信息的资源类型，内容由用户决定。资源创建完成后，可在容器应用中加载使用。例如，在“数据卷”中加载资源文件，使其成为容器中的文件，或者在“环境变量”中加载，使其成为容器中的环境变量。

操作步骤：

- 1) 在控制中心中，选择【计算】【云容器引擎】，进入容器服务界面；
- 2) 单击左侧导航栏的【配置中心】【配置项】，单击【创建配置项】；



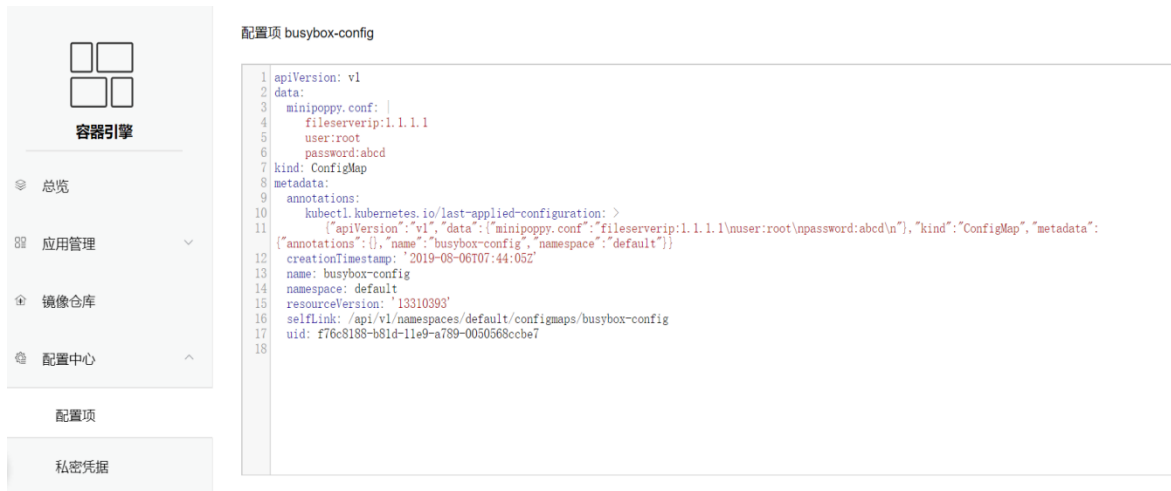
- 3) 参照下表设置新增配置参数，其中带“*”标志的参数为必填参数。输入完成后，单击【创建】；

参数	参数说明
*集群	配置项所在集群。若没有可选集群，单击“创建集群”进行创建，操作步骤请参见3.4.1.1创建集群。
*命名空间	配置项所在命名空间。若没有可选命名空间，单击“创建命名空间”进行创建，操作步骤请参见3.4.3.1创建命名空间。
*名称	新建配置项的名称，同一个命名空间里命名必须唯一。
添加项	增加一对键、配置项内容。
键	配置项的键值。键值只能由字母、数字、句点、连字符和下划线组成
配置项内容	配置项的内容，通过上传文件来表示。



- 3) 单击【创建】。等待创建成功；
- 4) 创建成功后，服务列表中会出现已创建的配置项，对创建的配置项可以编辑YAML、编辑、删除的操作。

编辑YAML：

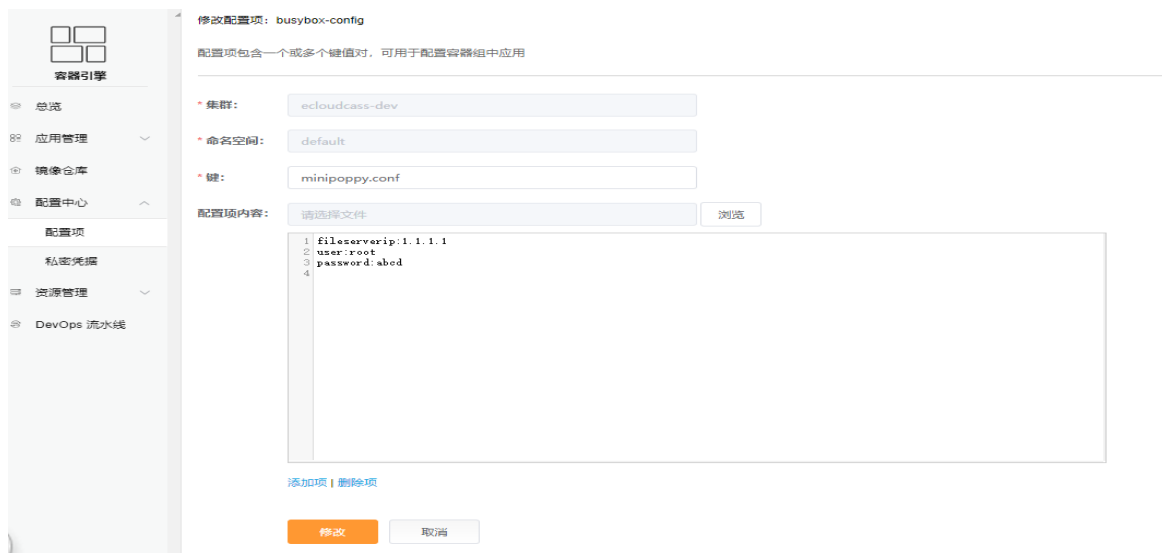


The screenshot shows the '配置项 busybox-config' (Configuration Item busybox-config) edit page. On the left is a navigation sidebar with '容器引擎' (Container Engine) selected, and sub-items like '总览', '应用管理', '镜像仓库', '配置中心', '配置项', and '私密凭据'. The main area displays the configuration item details and a YAML editor. The YAML content is as follows:

```

1 apiVersion: v1
2 data:
3   minipoppy.conf: |
4     fileserverip:1.1.1.1
5     user:root
6     password:abcd
7 kind: ConfigMap
8 metadata:
9   annotations:
10    kubernetes.io/last-applied-configuration: >
11    [{"apiVersion":"v1","data":{"minipoppy.conf":"fileserverip:1.1.1.1\nuser:root\npassword:abcd\n"},"kind":"ConfigMap","metadata":
12    {"annotations":{},"name":"busybox-config","namespace":"default"}}]
13 creationTimestamp: "2019-08-06T07:44:05Z"
14 name: busybox-config
15 namespace: default
16 resourceVersion: "13310393"
17 selfLink: /api/v1/namespaces/default/configmaps/busybox-config
18 uid: f76c8188-b81d-11e9-a789-0050568cbe7
    
```

配置项编辑：



The screenshot shows the '修改配置项: busybox-config' (Edit Configuration Item: busybox-config) form. The form includes the following fields and options:

- 集群:** ecloudcass-dev
- 命名空间:** default
- 键:** minipoppy.conf
- 配置项内容:** 请选择文件 (Please select a file) with a '浏览' (Browse) button.

The '配置项内容' (Configuration Item Content) field contains the following text:

```

1 fileserverip:1.1.1.1
2 user:root
3 password:abcd
4
    
```


At the bottom of the form, there are '添加项 | 删除项' (Add Item | Delete Item) links and '修改' (Modify) and '取消' (Cancel) buttons.

3.4.5.2 创建私密凭证

私密凭据是一种用于存储应用所需要认证信息、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，可在容器应用中作为文件或者环境变量使用。

操作步骤：

- 1) 在控制中心中，选择【计算】【云容器引擎】，进入容器服务界面；
- 2) 单击左侧导航栏的【配置中心】【私密凭据】，单击【创建私密凭据】；



容器引擎

- 总览
- 应用管理
- 镜像仓库
- 配置中心
- 配置项
- 私密凭据

私密凭据

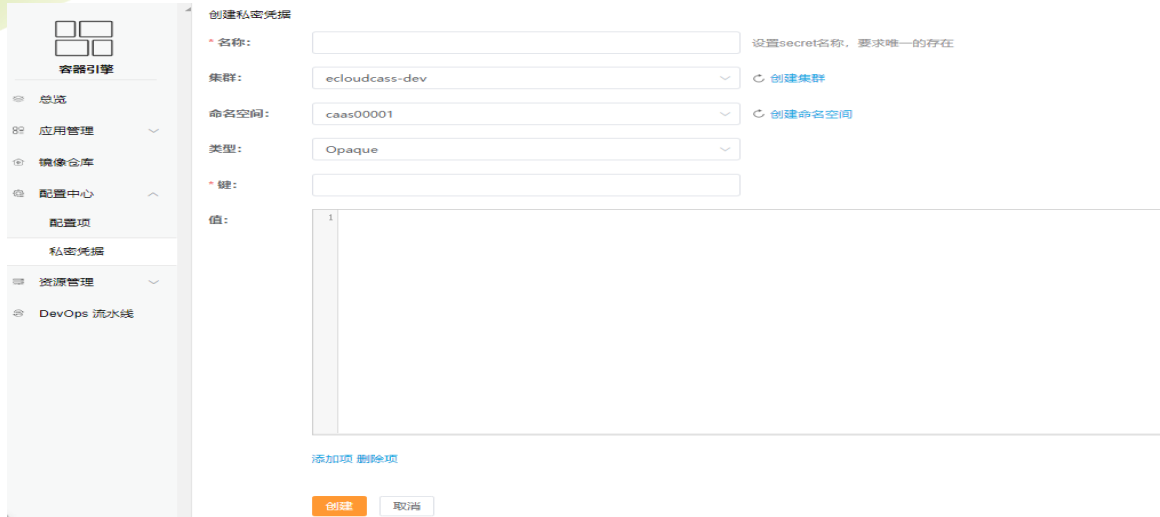
创建私密凭据

集群: ecloudcass-dev
命名空间: 请选择
私密凭据名称搜索

私密凭据名称	类型	命名空间	创建时间	操作
default-token-npqvk	kubernetes.io/service-account-token	test1	2019-08-16 15:14:02	查看YAML 删除
httpsdemo	kubernetes.io/tls	default	2019-08-15 15:37:57	查看YAML 删除
gohttps	kubernetes.io/tls	caas00001	2019-08-14 15:08:56	查看YAML 删除
cce81	kubernetes.io/dockerconfigjson	caas00001	2019-08-14 11:20:09	查看YAML 删除

3) 参照下表设置新增私密凭证参数，其中带“*”标志的参数为必填参数。输入完成后，单击【创建】；

参数	参数说明
*集群	配置项所在集群。若没有可选集群，单击“创建集群”进行创建，操作步骤请参见3.4.1.1创建集群。
*命名空间	配置项所在命名空间。若没有可选命名空间，单击“创建命名空间”进行创建，操作步骤请参见3.4.3.1创建命名空间。
*名称	新建配置项的名称，同一个命名空间里命名必须唯一。
*类型	Opaque：一般密钥类型。 Credentials：存放拉取私有仓库镜像所需的认证信息。 Dockercfg：存放拉取私有仓库镜像所需的配置文件。 IngressTLS：存放Ingress服务所需的证书。
类型为：Opaque	
添加项	增加一对键、配置项内容。
键	配置项的键值。键值只能由字母、数字、句点、连字符和下划线组成。
配置项内容	配置项的内容，通过上传文件来表示。



创建私密凭据

*名称: 设置secret名称, 要求唯一的存在

集群: [创建集群](#)

命名空间: [创建命名空间](#)

类型:

*键:

值:

1

[添加项](#) [删除项](#)

参数	参数说明
类型为: Credentials	
镜像仓库地址	输入镜像仓库的地址。
用户名	该镜像仓库的用户名称。
密码	用户名称所对应的的密码。
邮箱	该用户的邮箱地址, 必须为user@domain格式。



创建私密凭据

*名称: 设置secret名称, 要求唯一的存在

集群: [创建集群](#)

命名空间: [创建命名空间](#)

类型:

*镜像仓库地址:

*用户名:

*密码:

*邮箱:

参数	参数说明
类型为: Dockercfg	
配置文件	上传拉取私有仓库镜像所需的配置文件。

容器引擎

- 总览
- 应用管理
- 镜像仓库
- 配置中心
- 配置项
- 私密凭据
- 资源管理
- DevOps 流水线

创建私密凭据

*名称: 设置secret名称, 要求唯一的存在

集群: [创建集群](#)

命名空间: [创建命名空间](#)

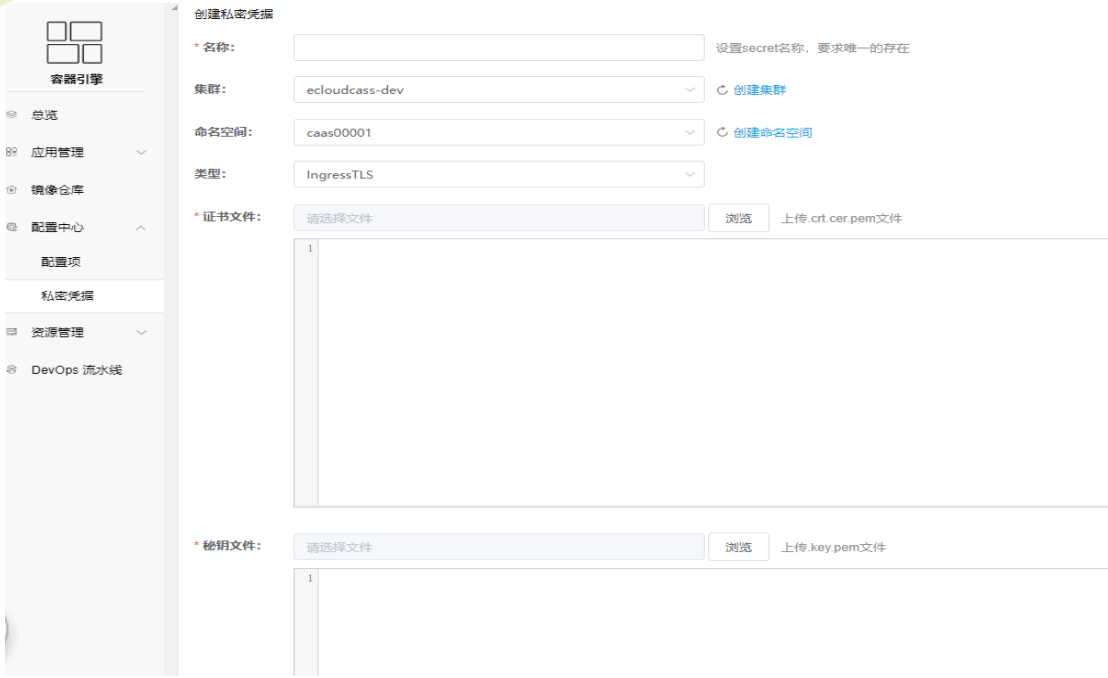
类型:

*配置文件: 上传 dockerconfigjson或

docker/config.json

1

参数	参数说明
类型为: IngressTLS	
证书文件	存放Ingress服务所需的证书文件。
秘钥文件	上传所需的秘钥文件。



4) 单击【创建】。等待创建成功；

5) 创建成功后，服务列表中会出现已创建的密钥凭证，对创建的密钥凭证可进行查看YAML、删除的操作。



3.5 创建应用

3.5.1 通过镜像创建无状态容器应用

若用户需要托管以docker容器打包的应用，请创建容器应用。无状态应用中各实例之间相互独立，互不依赖，任意一个Web请求完全与其他请求隔离。无状态容器应用更易实现可靠性和伸缩性。

说明：创建多个容器应用时，请确保容器应用使用的端口不冲突，否则部署会失败。

操作步骤：

- 1) (可选) 若基于私有镜像创建应用, 用户首先需要将镜像上传至镜像仓库。若基于平台提供的公有镜像创建应用, 则无需上传镜像;

登录控制中心, 进入云容器引擎界面, 单击左侧导航栏的【镜像仓库】, 在镜像仓库列表中点击仓库名称, 将镜像上传至容器镜像仓库, 具体操作请参见镜像仓库;

- 2) 在CCE左侧导航栏中选择【应用管理】【无状态应用】, 单击【创建应用】;
- 3) 参照下表设置基本信息, 其中带“*”标志的参数为必填参数:

参数	参数说明
*应用名称	新建容器应用的名称, 命名必须唯一。
*集群	应用所在集群。若没有可选集群, 单击“创建集群”进行创建, 操作步骤请参见 3.4.1.1 创建集群。
*命名空间	应用所在命名空间。若没有可选命名空间, 单击“创建命名空间”进行创建, 操作步骤请参见3.4.3.1 创建命名空间。
*实例数量	应用可以有一个或多个实例, 用户可以设置具体实例个数。每个应用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, 应用还能正常运行。

- 4) 添加容器: 单击【选择镜像】, 选择需要部署的镜像。

天翼云官方镜像: 展示了天翼云官方平台的公开镜像;

Dockerhub官方镜像: 展示了Dockerhub官方平台的公开镜像;

我的镜像: 展示了用户创建的所有镜像仓库。

配置镜像参数:

参数	说明
镜像名称	导入的镜像, 您可单击“选择镜像”进行更新。
容器名称	容器的名称, 可修改。
容器规格	可选择设定的配额, 或选择自定义配额。
高级设置	生命周期: 生命周期脚本定义, 主要针对容器类应用的生命周期事件应用采取的动作。步骤请参见 3.5.3 设置应用生命周期。

参数	说明
	<ul style="list-style-type: none"> - 启动命令：输入容器启动命令，容器启动后会立即执行。 - 启动后处理：应用启动后触发。 - 停止前处理：应用停止前触发。
	环境变量：容器运行环境中设定的一个变量。可以在应用部署后修改，为应用提供极大的灵活性。
	<ol style="list-style-type: none"> 1. 在“环境变量”页签，单击“添加环境变量”。 2. 变量类型分三种：手动添加、私密凭据导入、配置项导入。 3. 手动添加时，输入变量名称、变量/变量引用；私密凭据导入时，填写变量名称，并选择已经导入的变量/变量引用；配置项导入时，填写变量名称，并选择已经导入的变量/变量引用
	数据存储：支持挂载本地磁盘到容器中，以实现数据文件的持久化存储。详细步骤请参见 3.5.7 为应用挂载数据卷。
	健康检查：用于监测容器是否正常运行。设置了存活与业务两种探针。详细请参见应用健康检查。
	安全设置：请输入用户ID，对容器权限进行设置，保护系统和其他容器不受其影响。详细请参见安全设置。
	容器日志：设置日志采集策略、配置日志目录。用于收集容器日志以及日志防爆。详细请参见日志管理。

5) (可选) 一个应用实例包含1个或多个相关容器。若您的应用包含多个容器，请单击【添加】，进行容器的添加；

6) 单击【下一步】，添加服务；

A、单击【添加服务】，可将应用发布为可被访问的服务。内部访问参见 3.5.5 设置应用访问策略（内部访问）；

B、单击【下一步】，不设置应用访问（包括外网或内网访问）。

7) 配置完成后，单击【下一步】；

8) (可选) 为应用设置更多高级设置：

A、配置升级策略：

升级方式	说明
替换升级	先删除旧实例，再创建新实例。升级过程中业务会中断。
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。

B、配置缩容策略：输入时间，可设置范围(0-9999秒)，默认30秒；为应用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，该应用将被强制删除。

C、配置迁移策略：

(1) 选择是，输入时间，可设置范围(0-9000000000000000秒)，默认30秒，输入为有效整数。当应用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。

(2) 选择否，应用实例所在的节点不可用时，应用实例将不会调度到其它可用节点。

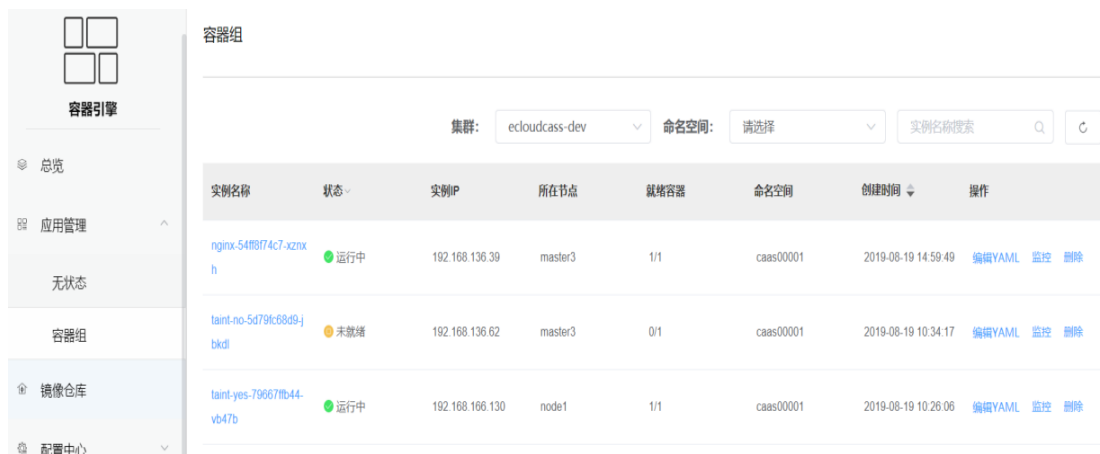
D、配置调度策略：你可以根据需要自由组合静态的全局调度策略或动态的运行调度策略来实现自己的需求。详情请参见 3.5.4 设置应用调度策略。

E、配置完成后，单击【提交】，创建完成后返回应用列表。在应用列表中，待应用状态为“运行中”，应用创建成功。应用状态不会实时更新，请按F5查看。

3.5.2 容器组

容器组是Kubernetes部署应用或服务的最小的基本单位。一个容器组可以封装多个应用容器(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。

用户创建完应用，查看【容器组列表】。



实例名称	状态	实例IP	所在节点	就绪容器	命名空间	创建时间	操作
nginx-54f874c7-xzrxh	运行中	192.168.136.39	master3	1/1	caas00001	2019-08-19 14:59:49	编辑YAML 监控 删除
taint-no-5d79fc68d9-jbkdl	未就绪	192.168.136.62	master3	0/1	caas00001	2019-08-19 10:34:17	编辑YAML 监控 删除
taint-yes-79d67db44-vb47b	运行中	192.168.166.130	node1	1/1	caas00001	2019-08-19 10:26:06	编辑YAML 监控 删除

容器组列表页中内容包括：实例名称、状态、实例IP、所在节点、就绪容器（已就绪/全部）、创建时间、操作（编辑YAML、监控、删除）。

点击【实例名称】页面可以跳转到【容器组详情页】。查看容器组基本信息、容器、监控、容器终端、日志、事件。

1、容器组详情_容器 界面样式：

容器引擎

总览

应用管理

无状态

容器组

镜像仓库

容器组详情

实例名称: nginx-54f8f74c7-xzndh

创建时间: 2019-08-19 14:59:49

状态: 运行中

原因: Running

所属集群: ecloudcass-dev

所属命名空间: caas00001

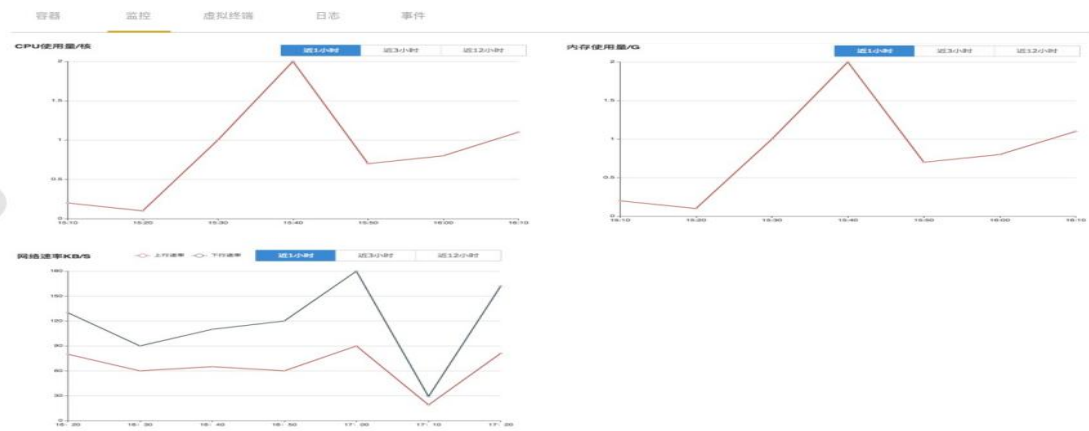
所属节点: master3

实例IP: 192.168.136.39

容器 监控 容器终端 日志 事件

容器名称	镜像	启停次数
nginx	172.18.141.128-443/library/nginx:latest	0

2、容器组详情_监控 界面样式



3、容器组详情_容器终端 界面样式

容器
监控
虚拟终端
日志
事件

ⓘ 当你离开这个容器组的页面时,所有已打开的终端连接都将被关闭,你通过终端启动的任何前台进程都将被终止

容器:

4、容器组详情_日志 界面样式

容器 监控 容器终端 **日志** 事件

* 容器名称: metrics-server

Go to End

1	E0728 21:43:21.011960	1 reststorage.go:147] unable to fetch pod metrics for pod default/testfanbin-6c6d9d4677-cqphf: no metrics known for pod
2	E0728 21:43:21.169738	1 reststorage.go:147] unable to fetch pod metrics for pod default/testfanbin-6c6d9d4677-cqphf: no metrics known for pod
3	E0728 21:43:36.034156	1 reststorage.go:147] unable to fetch pod metrics for pod default/testfanbin-6c6d9d4677-bmvx8: no metrics known for pod
4	E0728 21:43:36.095959	1 reststorage.go:147] unable to fetch pod metrics for pod default/testfanbin-6c6d9d4677-bmvx8: no metrics known for pod
5	E0728 21:43:36.235733	1 reststorage.go:147] unable to fetch pod metrics for pod default/testfanbin-6c6d9d4677-bmvx8: no metrics known for pod
6	E0728 21:43:36.242097	1 manager.go:111] unable to fully collect metrics: unable to fully scrape metrics from source kubelet_summary:node1: unable to get CPU for container "container-0" in pod default/testfanbin-6c6d9d4677-bmvx8 on node "172.18.141.132". discarding data: missing cpu usage metric

5、容器组详情_事件 界面样式

容器 监控 容器终端 日志 **事件**

消息内容搜索

发生次数	首次发生时间	最近发生时间	事件类型	消息内容
1	2019-08-02 15:01:39	2019-08-02 15:01:39	Normal	Started container gohttps
1	2019-08-02 15:01:39	2019-08-02 15:01:39	Normal	Created container gohttps
1	2019-08-02 15:01:39	2019-08-02 15:01:39	Normal	Successfully pulled image "172.18.141.128/library/gohttpsdemo:latest"
1	2019-08-02 15:01:39	2019-08-02 15:01:39	Normal	Pulling image "172.18.141.128/library/gohttpsdemo:latest"

容器组列表对实例可进行编辑YAML、监控、删除的操作。

编辑YAML界面：

容器引擎

- 总览
- 应用管理
- 无状态
- 容器组
- 镜像仓库
- 配置中心
- 资源管理

容器组 taint-yes-79667ffb44-vb47b

```

1 apiVersion: v1
2 kind: Pod
3 metadata:
4   annotations:
5     cnf.projectcalico.org/podIP: 192.168.166.130/32
6   creationTimestamp: '2019-08-19T02:26:06Z'
7   generateName: taint-yes-79667ffb44-
8   labels:
9     app: taint-yes
10    pod-template-hash: 79667ffb44
11    name: taint-yes-79667ffb44-vb47b
12    namespace: caas00001
13  ownerReferences:
14    - apiVersion: apps/v1
15      blockOwnerDeletion: true
16      controller: true
17      kind: ReplicaSet
18      name: taint-yes-79667ffb44
19      uid: f68cd96f-c227-11e9-b981-0030568cbe7
20  resourceVersion: '15799482'
21  selfLink: /api/v1/namespaces/caas00001/pods/taint-yes-79667ffb44-vb47b
22  uid: k2f36192-c228-11e9-b981-0030568cbe7
23  spec:
24    containers:
25      - image: '172.18.141.128:443/library/gohttpsdemo:latest'
26        imagePullPolicy: Always
27        name: container-1
          
```

3.5.3 设置应用生命周期

云容器引擎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：

- A、启动命令:输入容器启动命令，容器启动后立即执行；
- B、启动后处理：应用启动后触发；
- C、停止前处理：应用停止前触发；

“启动命令”的操作步骤：

- 1) 登录云容器引擎控制台，在创建应用配置生命周期过程中，选择【启动命令】页签；
- 2) 输入启动命令：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"...]。

命令方式	操作步骤
可执行程序方式	<ul style="list-style-type: none"> • 在启动命令填框中输入可执行的程序，例如/run/server。 • 在启动命令填框中输入启动参数，例如—port=8080。

“启动后处理”的操作步骤：

- 1) 登录云容器引擎控制台，在创建应用配置生命周期过程中，选择【启动后处理】页签；
- 2) 设置启动后处理的参数：

命令方式	操作步骤
可执行程序方式	<p>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需找可执行程序），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。</p> <p>如需要执行的命令如下：</p> <pre>exec:</pre>

命令方式	操作步骤
	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_agent 安装。
HttpGet请求方式	发起一个HTTP调用请求。配置参数如下: <ul style="list-style-type: none"> • 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。

“停止前处理”的操作步骤:

- 1) 登录云容器引擎控制台, 在创建应用配置生命周期过程中, 选择【停止前处理】页签;
- 2) 设置停止前处理的参数:

命令方式	操作步骤
可执行程序方式	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需找可执行程序), 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: <pre>exec: command: - /install.sh - install_agent</pre> 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_agent 安装。
HttpGet请求方式	发起一个HTTP调用请求。配置参数如下: <ul style="list-style-type: none"> • 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。

3.5.4 设置应用调度策略

系统支持丰富的调度策略, 包括静态的全局调度策略, 以及动态的运行时代调度策略, 用户可以根据需要自由组合使用这些策略来实现自己的需求。平台支持以下两种亲和性的调度方式。应用间的亲和性: 决定应用部署在相同或不同节点中。

设置应用间亲和：应用部署在相同“节点”中。用户可根据业务需求进行应用的就近部署，容器间通信就近路由，减少网络消耗。如下图，APP1、APP2、APP3 和 APP4 部署在相同节点上，为亲和性部署；



设置应用间反亲和：“不同应用”或“相同应用的多个实例”部署在不同主机中。同个应用的多个实例反亲和部署，减少宕机影响；互相干扰的应用反亲和部署，避免干扰。如 APP1、APP2、APP3 和 APP4 分别部署在不同节点上，这四个应用为反亲和性部署。



应用与节点间的亲和性：应用与节点亲和，决定应用部署在某些特定的主机中，应用与节点反亲和，决定应用不能部署在某些特定的主机中。



注意

在设置“应用间的亲和性”和“应用与节点间的亲和性”时，需确保不要出现互斥情况，否则应用会部署失败。例如如下互斥情况，应用将会部署失败：

1. APP1、APP2设置了应用间的反亲和，例如APP1部署在Node1，APP2部署在 Node2。
2. APP3部署上线时，既希望与 APP2 亲和，又希望可以部署在不同节点如Node3 上，这就造成了应用亲和和节点亲和间的互斥，导致最终应用部署失败。

操作步骤：

调度说明	操作步骤
应用和节点的亲和性	

调度说明	操作步骤
与节点的亲和性	<ul style="list-style-type: none"> 在【调度】【与节点的亲和性】下，单击【添加节点亲和】； 勾选需要亲和的节点，单击【确定】。该应用将部署在选择的节点中；
与节点的反亲和性	<ul style="list-style-type: none"> 在【调度】【与节点的反亲和性】下，单击【添加节点反亲和】； 勾选需要反亲和的节点，单击【确定】。该应用将不会部署在选择的节点中；
应用间的亲和性	
与应用的亲和	<ul style="list-style-type: none"> 在【调度】【与应用亲和性】下，单击【添加应用亲和】； 勾选需要亲和的应用，单击【确定】。亲和的应用将部署在相同节点中；
与应用的反亲和	<ul style="list-style-type: none"> 在【调度】【与应用反亲和性】下，单击【添加应用反亲和】； 勾选需要反亲和的应用，单击【确定】。反亲和的应用将部署在不同节点中；

3.5.5 设置应用访问策略（内部访问）

在实际环境中，对服务的访问会有两种来源：集群内部的程序、集群外部。本章节介绍集群“内部访问”的类型：

集群虚拟IP：选择此类型，系统将自动分配一个仅集群内部可以访问的虚拟IP，供集群内部的容器访问；

“集群虚拟IP”的操作步骤：

- 1) 登录云容器引擎控制台，在【网络管理】的服务页签下，单击【创建服务】；
- 2) 配置服务参数：

参数	参数说明
集群	服务所在集群。若没有可选集群，单击“创建集群”进行创建，操作步骤请参见 3.4.1.1创建集群。
命名空间	服务所在命名空间。若没有可选命名空间，单击“创建命名空间”进行创建，操作步骤请参见3.4.3.1创建命名空间。
关联应用	选择需要添加Service的工作负载。

参数	参数说明
服务名称	新建服务的名称。
访问方式	集群内访问
容器端口	容器中应用启动监听的端口。
访问端口	ClusterIP的服务端口，映射到容器的应用端口。
协议	TCP/UDP。

3) 单击【确认】；

3.5.6 为应用挂载数据卷

Docker镜像是由多个文件系统叠加而成，当启动一个容器的时候，Docker会加载只读镜像层并在上面添加一个读写层。当删除Docker容器并通过该镜像重新启动时，之前的更改将会丢失。为了能够保存数据以及共享容器间的数据，Docker提出了数据卷的概念。简单来说，数据卷就是目录或者文件，它可以绕过默认的联合文件系统，以正常的文件或者目录的形式存在于主机上。

在Docker中，数据卷只是磁盘或另一容器中的目录。其生命周期不受管理，且Docker现在提供的卷驱动程序功能非常有限。云容器引擎CCE采用的是Kubernetes的数据卷的概念，Kubernetes数据卷具有明确的生命周期管理，支持多种类型的数据卷，同时实例可以使用任意数量的数据卷。

更多Kubernetes数据卷信息可见[Volumes](#)。

CCE支持挂载本地磁盘：

挂载本地磁盘：支持hostPath、emptyDir、configMap、secret四种。

hostPath：指定主机中的文件或目录挂载到容器的某一路径中；

EmptyDir：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；

ConfigMap：将配置文件中的key映射到容器中，可以用于挂载配置文件到指定容器目录；

Secret：将密钥中的数据挂载到指定的容器路径。

“挂载本地磁盘”的操作步骤：

- 1) 登录云容器引擎控制台，在创建应用过程中，在【容器设置】【数据存储】【本地磁盘】【添加本地磁盘】；

- 2) 卷类型选择**hostPath**，表示在容器上挂载宿主机上的文件或目录。通常用于：“容器应用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应用”；

配置参数：

参数	参数说明
本地磁盘名称	输入存储名称。
卷类型	<ul style="list-style-type: none"> 选择卷类型为【hostPath】； 输入主机路径，如/tmp； 单击【确定】，挂载对应的容器路径；
挂载路径	数据卷挂载到容器上的路径。 注意：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，应用创建失败。
权限	<ul style="list-style-type: none"> 只读：只能读容器路径中的数据卷； 可写：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；

- 3) 卷类型选择**emptyDir**：容器分配到节点时系统将自动创建卷，初始内容为空。在同一个Pod中所有容器可以读写emptyDir中的相同文件。当Pod从节点上移除时，emptyDir中的数据也会永久删除。通常用于临时数据的高速存储。

配置参数：

参数	参数说明
本地磁盘名称	输入存储名称。
卷类型	<ul style="list-style-type: none"> 选择卷类型为【emptyDir】； 选择【存储介质】： <ul style="list-style-type: none"> 默认：存储在硬盘上，适用于数据量大，读写效率要求低的场景。 内存：存储在内存中，适用于数据量少，读写效率要求高的场景。 单击【确定】，挂载对应的容器路径；

参数	参数说明
挂载路径	数据卷挂载到容器上的路径。 注意：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，应用创建失败。
权限	<ul style="list-style-type: none"> • 只读：只能读容器路径中的数据卷。 • 可写：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。

- 4) 卷类型选择**configMap**：平台提供应用代码和配置文件的分离，configMap用于处理应用配置参数。用户需要提前创建应用配置，操作步骤请参见3.4.4创建应用配置项；

配置参数：

参数	参数说明
本地磁盘名称	输入存储名称。
卷类型	<ul style="list-style-type: none"> • 选择卷类型为【configMap】； • 选择对应的configMap名称； 说明：configMap需要提前创建，请参见 3.4.5.1创建配置项(configMap)。
挂载路径	数据卷挂载到容器上的路径。
权限	只读：只能读容器路径中的数据卷。

- 5) 卷类型选择 **secret**：用户需要提前创建私密凭据，操作步骤请参见 3.4.5.2创建私密凭据；

配置参数：

参数	参数说明
本地磁盘名称	输入存储名称。
卷类型	<ul style="list-style-type: none"> • 选择卷类型为【secret】； • 选择对应的 secret 名称； 说明：secret需要提前创建，请参见 3.4.5.2创建私密凭据(secret)。
挂载路径	数据卷挂载到容器上的路径。
权限	只读：只能读容器路径中的数据卷。

3.6 应用管理运维

3.6.1 升级应用

- 1) 登录云容器引擎控制台，在左侧导航栏中选择【应用管理】【无状态】，单击已创建的应用，进入应用详情页面，选择【升级】页签；
- 2) 请根据业务需求进行应用的升级，升级的参数说明请参见下表：

参数	参数说明
容器配置	
选择镜像	在对应的容器下，单击【选择镜像】进行修改。
容器名称	单击列表框，修改容器名称。
容器规格	对容器允许使用的cpu的最小值、最大值，内存的最小值、最大值进行修改。
高级设置	
生命周期	支持对生命周期进行修改，为应用提供极大的灵活性。 生命周期包括：启动命令、启动后处理、停止前处理。具体步骤如生命周期。
健康检查	支持对通过探针监测容器进行修改。具体步骤如健康检查中对探针的操作。
环境变量	容器运行环境中设定的一个变量。可以在应用部署后修改，为应用提供极大的灵活性。 1. 在【环境变量】页签，单击【添加环境变量】； 2. 输入变量名称、变量/变量引用；
数据存储	支持对数据存储进行修改，增加本地磁盘。 步骤同挂载本地磁盘。
安全设置	支持通过修改用户ID对容器权限进行修改。
容器设置	容器设置日志采集策略、配置日志目录。用于收集容器日志以及日志防爆。通过添加日志策略对容器设置进行修改。详细请参见日志管理。

- 3) 更新完成后，单击【提交】；

3.6.2 应用监控

应用创建成功后，可通过性能监控，来监控容器的CPU和内存使用情况。

- 1) 登录云容器引擎控制台，单击左侧导航栏的【应用管理】；
- 2) 单击已创建应用的名称，进入应用详情页面；
- 3) 单击【监控】页签；
- 4) 查看相应用的CPU使用率、内存使用率、网络速率：

CPU使用率：横坐标表示时间，纵坐标表示CPU使用率。蓝色线条表示CPU使用量，只有实例处于运行状态时，才能查看CPU使用量。

内存使用率：横坐标表示时间，纵坐标表示内存使用量。蓝色线条表示内存使用量，实例处于非运行状态时，不能看到内存使用量。

网络速率：横坐标表示时间，纵坐标表示网络速率。蓝色线条表示上行速率，红色线条表示下行速率。

3.6.3 启停应用

停止运行中的应用，应用将无法访问，状态显示为“已暂停”。应用停止后，可直接将其启动。

- 1) 登录云容器引擎控制台，单击左侧导航栏的【应用管理】【无状态】；
- 2) 单击运行中应用后的【更多】【暂停】，停止运行中的应用；
- 3) 单击已暂停应用后的【启动】，启动已停止的应用。

3.6.4 删除应用

任何状态的应用，都可直接将其删除。

- 1) 登录云容器引擎控制台，单击左侧导航栏的【应用管理】【无状态】；
- 2) 单击待删除应用后的【更多】【删除】，删除应用，请仔细阅读系统提示；
- 3) 单击【确定】；

3.7 镜像仓库

镜像仓库是用于存储、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。镜像仓库包括如下内容：

天翼云官方镜像：展示了天翼云平台上的公开镜像，您可以基于公开镜像创建应用。

Dockerhub官方镜像：展示了Dockerhub官方平台上的公开镜像，您可以基于公开镜像创建应用。

我的镜像：展示了用户创建的所有镜像仓库。

使用镜像仓库上传或下载镜像，您需要注意以下事项：



注意

- 镜像仓库不扫描用户上传的镜像，不负责对用户上传的镜像进行安全性验证。上传的镜像中请不要包含未加密的口令，密码等隐私信息，以避免隐私泄露。下载公有镜像时，应确定数据来自于可信的仓库源，以避免下载到恶意软件。
- 如果使用自定义镜像，请确保镜像来源可信，不在容器镜像内安装不必要的软件，在升级时使用安全补丁升级镜像。使用第三方镜像，造成的后果（例如，环境不可用）用户需自己承担。
- 磁盘满只会导致无法上传镜像到仓库，会产生异常提示信息，不会影响其他服务；为防止其它业务（例如日志）把磁盘占满，导致仓库无法上传，建议对仓库的存储独立挂盘。

3.7.1 基本概念

在使用之前，您需要了解以下基本概念：

镜像仓库：提供docker容器镜像管理功能，用户在创建容器应用前，需要将应用所需的镜像上传到镜像仓库。docker镜像是一个模板，用于创建docker容器。docker提供了一个简单的机制来创建新的镜像或更新已有镜像。

(仓库)属性：属性分为公有和私有两种。公有：任何租户、用户均可以下载。私有：仅当前租户或租户下的用户可用。

3.7.2 创建容器镜像仓库

容器镜像仓库是用于存储、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。

- 1) 登录云容器引擎控制台，单击左侧导航栏的【镜像仓库】，单击【创建镜像仓库】；
- 2) 参照下表设置镜像仓库信息，其中带“*”标志的参数为必填参数：

参数	参数说明
*仓库名称	输入容器镜像仓库名称，例如：nginx。
*(仓库)属性	容器镜像仓库类型，默认为公有。 <ul style="list-style-type: none"> • 公有：任何租户、用户均可以下载。 • 私有：仅当前租户或租户下的用户可用。

- 3) 配置完成后，单击【创建】；

3.7.3 上传私有镜像（外网上传）

通过外网上传私有镜像，指的是上传镜像的虚拟机(也可称为节点)不能联通云容器引擎的内部网络。建议使用内网上传镜像，网速会更快，内网上传请参见 3.7.4 上传私有镜像(内网上传)。

本章节以nginx:1.10 镜像为例，介绍如何上传本地镜像到CCE镜像仓库。上传成功后，会在镜像仓库显示已上传成功的镜像。

在使用Docker客户端上传镜像的过程中，如果出现网络异常断开或软件仓库被异常重启的情况，是由于Docker客户端，其不会继续向软件仓库发送上传镜像的请求，从而导致Docker客户端卡机无法退出。在此情况下可以通过重启docker daemon的方式解决。

前提条件：

- 1) 准备一台已安装Docker客户端的虚拟机，且Docker 版本为 1.11.2、1.12.0、1.12.1或1.12.6；
- 2) 请确保镜像的正确性，能够成功后台启动；
- 3) 已创建容器镜像仓库，请参见 3.7.2 创建容器镜像仓库。

操作步骤：

- 1) 创建镜像仓库，例如镜像仓库名称为nginx：
 - A、单击页面顶部的【创建镜像仓库】；
 - B、照下表设置镜像仓库信息，其中带“*”标志的参数为必填参数：

参数	参数说明
*仓库名称	输入容器镜像仓库名称，例如：nginx。
*(仓库)属性	容器镜像仓库类型，默认为公有。 <ul style="list-style-type: none"> • 公有：任何租户、用户均可以下载。 • 私有：仅当前租户或租户下的用户可用。

- C、配置完成后，单击【创建】。单击【查看仓库详情】；
 - D、单击【Pull/Push指南】页签，可查看Pull/Push镜像的方法；
- 2) 以下步骤讲解“通过外网云主机” Push镜像到CCE镜像仓库；
 - A、以root用户登录docker所在的虚拟机；

B、执行以下操作，允许docker访问我的镜像仓库；

1、配置docker参数：

a.Ubuntu系统下：

vi/etc/default/docker

在DOCKER_OPTS中，增加如下粗字体所示：

```
DOCKER_OPTS="-insecure-registry registry.cn-north-l.huaweicloud.com"
```

b. SuSE操作系统下：

vi/etc/sysconfig/docker

若配置文件中有DOCKER_OPTS字段，则在在DOCKER_OPTS中，增加如下粗字体所示：

```
DOCKER_OPTS="-insecure-registry registry.cn-north-l.huaweicloud.com"
```

若配置文件中没有DOCKER_OPTS字段，则在INSECURE_REGISTRY中，增加如下粗体所示：

```
INSECURE_REGISTRY="-insecure-registry registry.cn-north-l.huaweicloud.com"
```

2、保存文件并退出；

3、重启docker；


```
service docker restart
```

C、创建容器镜像仓库，参考3.7.2创建容器镜像仓库；

D、单击创建好的镜像仓库，进入仓库详情页面；

E、获取仓库访问权限文件：

1、在【Pull/Push指南】页面，单击【生成docker login指令】；

2、单击【生成docker login指令】，单击  复制docker login指令；

3、将docker login指令拷贝到虚拟机中，按Enter；

F、给nginx:1.10镜像打标签：

1、查看镜像名称

```
docker images
```

2、给nginx:1.10镜像打标签：docker tag[原镜像名:版本号] [该镜像的仓库外网地址]:[版本号]。样例如下：

```
docker tag nginx:1.10 192.168.0.0/namespace/nginx:1.10
```

其中，**192.168.0.0/namespace/nginx**为本镜像仓库外网地址，请查看“Pull/Push 指南”中的“1.本镜像仓库地址”中的“外网地址”，**1.10**为版本号。

G、Push镜像至镜像仓库：

docker push [该镜像的仓库外网地址]:[版本号]

样例如下：

docker push

192.168.0.0/namespace/nginx:1.10 终端显示如

```

信息，表明：push 镜像成功。
695da0025de6: Pushed
fe4c16cbf7a4: Pushed
1.10: digest: sha256:eb7e3bbd8e3040efa71d9c2cacfa12a8e39c6b2ccd15eac12bdc49e0b66cee63
size: 948
    
```

返回系统，在容器镜像仓库的“镜像详情”页面，执行刷新操作后可查看到对应的镜像信息。

3.7.4 上传私有镜像（内网上传）

通过内网上传私有镜像，指的是上传镜像的虚拟机（也可称为节点）能联通CCE的内部网络，在内网上传镜像时，网速会更快。建议使用在CCE创建的虚拟机，该虚拟机中已安装1.11.2版本的Docker。

本章节以nginx:1.10镜像为例，介绍如何上传本地镜像到CCE镜像仓库。上传成功后，会在镜像仓库显示已上传成功的镜像。

前提条件：

- 1) 准备一台已安装Docker客户端的虚拟机，且Docker版本为1.11.2、1.12.0、1.12.1或1.12.6。建议使用CCE创建的节点，该节点中已安装1.11.2版本的 Docker，并且与CCE内网联通，网速更快。
- 2) 请确保镜像的正确性，能够成功后台启动；

操作步骤：

- 1) 创建镜像仓库，例如镜像仓库名称为 nginx：

A、单击页面顶部的【创建镜像仓库】；

B、照下表设置镜像仓库信息，其中带“*”标志的参数为必填参数：


参数	参数说明
*仓库名称	输入容器镜像仓库名称，例如：nginx。
*(仓库)属性	容器镜像仓库类型，默认为公有。 <ul style="list-style-type: none"> • 公有：任何租户、用户均可以下载。 • 私有：仅当前租户或租户下的用户可用。

C、配置完成后，单击【创建】。单击【查看仓库详情】；

D、单击【Pull/Push指南】页签，可查看Pull/Push镜像的方法；

- 2) 以下步骤讲解“通过内网云主机”Push镜像到CCE镜像仓库；
 - A、以root用户登录docker所在的虚拟机；

B、获取仓库访问权限文件；

- 1、单击创建好的镜像仓库，进入仓库详情页面。单击【Pull/Push指南】页签，单击【2. 以外网操作为例的Step2】中的【生成docker login指令】；
- 2、单击【生成docker login指定】，单击  复制docker login指令，例如docker login指令为：

```
docker login -u southchina@MMZES9KTH43ELTVYTGXA -p
c4c8b79699abe427daf60b8035955230c146f744b990b03f7f30a5cb89fb382c registry.cn-north-1.huaweicloud.com
```

- 3、将docker login指令中的外网仓库地址“registry.cn-north-1.huaweicloud.com”替换为内网地址的IP:端口号。内网地址的 IP:端口号请在“1. 本镜像仓库地址”中查看，例如为10.175.11.161:20202，替换后的docker login指令为：

```
docker login -u southchina@MMZES9KTH43ELTVYTGXA -p
c4c8b79699abe427daf60b8035955230c146f744b990b03f7f30a5cb89fb382c 10.175.11.161:20202
```

- 4、将docker login指令拷贝到虚拟机中，按Enter；

C、给nginx:1.10镜像打标签：

docker tag[原镜像名:版本号] [该镜像的仓库外网地址]:[版本号]。样例如下：

样例如下：

docker tag nginx:1.10 10.175.11.161:20202/namespace/nginx:1.10

其中，10.175.11.161:20202/namespace/nginx 为本镜像仓库地址，请查看“Pull/Push指南”中的“1. 本镜像仓库地址”中的“内网地址”，1.10为版本号。

D、Push镜像至镜像仓库：

docker push 10.175.11.161:20202/namespace/nginx:1.10

终端显示如下信息，表明push镜像成功：

```
fe4c16cbf7a4: Pushed
1.10: digest: sha256:eb7e3bbd8e3040efa71d9c2cacfa12a8e39c6b2ccd15eac12bdc49e0b66cee63
size: 948
```

终端显示如下信息，表明push镜像成功。

```
6d6b9812c8ae:
Pushed
695da0025de6:
Pushed
fe4c16cbf7a4:
Pushed
```

返回系统，在容器镜像仓库的“镜像详情”页面，执行刷新操作后可查看到对应的镜像信息。

终

3.7.5 删除镜像仓库

- 1) 登录云容器引擎控制台，单击左侧导航栏的【镜像仓库】；
- 2) 在镜像仓库列表中，在待删除的镜像仓库名称后点击删除；在镜像仓库列表中，单击【删除】，删除已创建的容器镜像仓库，删除镜像仓库前，必须先删除容器镜像仓库中所包含的镜像。

3.7.6 删除镜像仓库中的镜像

- 1) 登录云容器引擎控制台，单击左侧导航栏的【镜像仓库】；
- 2) 单击待删除的镜像仓库名称，进入仓库中的镜像列表页；
- 3) 单击待删除镜像后的【删除】，删除镜像仓库中的镜像。

3.7.7 查看镜像仓库地址

- 1) 登录云容器引擎控制台，单击左侧导航栏的【镜像仓库】；
- 2) 单击镜像仓库名称，进入镜像列表页；
- 3) 查看镜像仓库地址：
内网拉取地址：同个 VPC 内的弹性云主机能访问的镜像地址；
外网拉取地址：互联网能访问的镜像地址；

4 常见问题

4.1 云容器引擎常见的使用场景如下？

- 适用于传统IT架构渐进式转型的场景，实现单一架构解耦拆分为多个容器，系统更灵活，轻松应对市场变化。
- 适用于业务上线效率低下的场景，实现容器镜像贯穿从开发到运维各环节，统一环境配置，业务快速上线。
- 适用于访问量有明显波峰、波谷的应用，实现自动弹性伸缩，系统秒级自动弹性扩容，快速响应并发高峰。
- 适用于系统IT资源浪费严重的场景，使平均负载维持较高水平，每分钱都真正支持业务。
- 适用于复杂系统运维压力大的场景，基于容器服务实现自动化运维，基于实时日志快速问题定位，界面化操作和短信通知实现24小时自动监控。

4.2 什么是环境变量？

环境变量是指容器运行环境中设定的一个变量，您可以在创建容器模板时设定不超过30个的环境变量；环境变量可以在应用部署后修改，为应用提供极大的灵活性。在CCE中设置环境变量与Dockerfile中的“ENV”效果相同。

4.3 集群删除之后相关数据能否再次找回？

集群删除之后，部署在集群上的应用也会同步删除，无法恢复，请慎重删除集群。

4.4 节点状态与应用状态不一致？

当节点的实际状态发生变化时，CCE界面上的节点状态可能不会立即更新(延迟大概2分钟)，而该节点上的应用状态能够实时更新，所以会出现节点状态与应用状态不一致的现象。

4.5 在已有的节点创建应用时未重新拉取镜像？

当容器镜像的tag不是latest时，用户更新该容器镜像但没有更改镜像的tag，则使用该镜像创建应用时，就不会重新拉取更新后的镜像，需要您在更新镜像的同时更改镜像的版本tag。

4.6 如何绑定弹性IP?

可以通过申请弹性IP并将弹性IP绑定到弹性云主机上，实现节点（云主机）访公网的目的。

- 1) 登录天翼云控制中心，点击产品列表中的【网络】【虚拟私有云】；
- 2) 在左侧导航树，单击【弹性公网IP】；
- 3) 单击【申请弹性公网IP】；
- 4) 在申请弹性公网IP页面，选择带宽时长、带宽值等，点击【下一步】【确认下单】；
- 5) 在成功申请的弹性IP申请后，点击【绑定】链接，选择创建的云主机(容器节点)，点击【确定】；

您还可以申请17个弹性IP。

弹性IP/ID	状态	已绑定私有IP	类型	带宽大小(Mbit/s)	计费模式	操作
183.134.74.142 888e4e10-cf72-4c79-8ff6-c5...	未绑定	--	电信	1	包年/包月	续费 绑定 更多 ▾

- 6) 绑定成功即可实现应用的外网访问；

4.7 创建容器应用有哪几种方式？有什么区别？

目前支持三种部署方式，用户可基于自身需求选择：

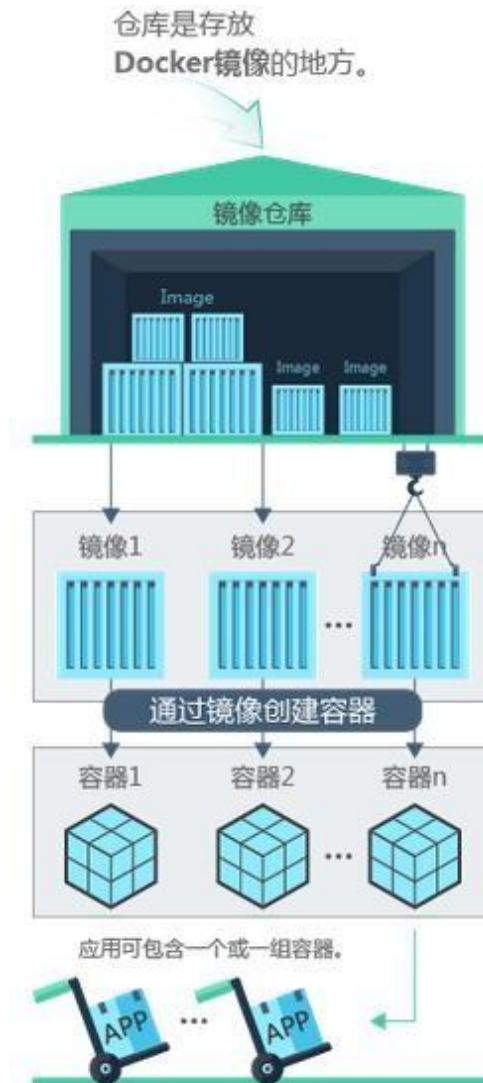
- 选择 Dockerhub 官方镜像：基于开源 docker 镜像创建容器应用，无需上传私有镜像。
- 选择天翼云官方镜像：基于天翼云官方平台上的镜像创建容器应用，无需上传私有镜像。
- 上传并选择私有镜像：您可基于业务需求制作私有 docker 镜像，上传到容器镜像服务。基于该私有镜像创建容器应用。

4.8 镜像、容器、应用的关系是什么？

镜像：Docker镜像是一个特殊的文件系统，除了提供容器运行时所需的程序、库、资源、配置等文件外，还包含了一些为运行时准备的配置参数（如匿名卷、环境变量、用户等）。镜像不包含任何动态数据，其内容在构建之后也不会被改变。

容器：镜像（Image）和容器（Container）的关系，就像是面向对象程序设计中的类和实例一样，镜像是静态的定义，容器是镜像运行时的实体。容器可以被创建、启动、停止、删除、暂停等。

镜像、容器、以及应用之间的关系请参见下图：



5 参考知识

5.1 ConfigMap配置项要求

configMap资源文件支持json和yaml两种格式，且文件大小不得超过2MB。

json格式：

文件名称为 configmap.json，配置示例如下：

```
{
  "kind": "ConfigMap",
  "apiVersion": "v1",
  "metadata": {
    "name": "paas-broker-app-017",
    "namespace": "lcqtest",
    "enable": true
  },
  "data": {
    "context":
    "{\"applicationComponent\":{\"properties\":{\"custom_spec\":{}}},\"node_name\": \"paas-broker-app\", \"stack_id\": \"0177eae1-89d3-cb8a-1f94-c0feb7e91d7b\"}, \"softwareComponents\": [{\"properties\":{\"custom_spec\":{}}}, \"node_name\": \"paas-broker\", \"stack_id\": \"0177eae1-89d3-cb8a-1f94-c0feb7e91d7b\"}]}"
  }
}
```

yaml格式：

文件名称为configmap.yaml，配置示例如下：

```
apiVersion: apps/v1beta2 # for versions before 1.8.0 use apps/v1beta1
kind: Deployment
metadata:
  name: nginx-deployment-basic
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      # nodeSelector:
      #   env: test-team
      containers:
        - name: nginx
          image: nginx:1.7.9 # replace it with your exactly <image_name:tags>
          ports:
            - containerPort: 80
```