



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	6
1.1	产品定义	6
1.2	DOCKER 基础知识	6
1.2.1	<i>Docker</i> 基本概念	6
1.2.2	安装 <i>Docker</i>	7
1.2.3	制作 <i>Docker</i> 镜像.....	7
2	快速入门	8
2.1	创建集群	8
2.2	创建容器应用	10
2.3	配置弹性伸缩策略.....	12
2.4	应用运维监控	14
3	操作指南	19
3.1	控制台说明.....	19
3.2	使用流程	19
3.3	环境设置	20
3.3.1	创建虚拟私有云.....	20
3.3.2	创建密钥对.....	21
3.4	资源准备	21
3.4.1	集群管理.....	21

3.4.2	节点管理.....	24
3.4.3	存储管理.....	29
3.4.4	创建应用配置项.....	31
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	通过镜像创建有状态容器应用.....	35
3.5.3	设置应用生命周期.....	37
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	设置应用访问策略（外部访问）.....	42
3.5.7	为应用挂载数据卷.....	44
3.6	应用访问设置	47
3.6.1	概述.....	47
3.6.2	集群内访问.....	48
3.6.3	VPC 内网访问.....	54
3.6.4	公网访问-弹性 IP.....	61
3.6.5	公网访问-四层负载均衡.....	66
3.6.6	七层负载均衡-（Ingress）.....	71
3.7	应用管理	77
3.7.1	弹性伸缩.....	77
3.7.2	手动伸缩.....	79

3.7.3	优雅缩容策略	79
3.7.4	升级应用	79
3.7.5	应用监控	80
3.7.6	启停应用	80
3.7.7	删除应用	81
3.8	镜像仓库	81
3.8.1	基本概念	81
3.8.2	创建组织	82
3.8.3	上传私有镜像（外网上传）	82
3.8.4	上传私有镜像（内网上传）	84
3.8.5	删除镜像仓库中的镜像	86
3.8.6	查看镜像仓库地址	86
4	最佳实践	88
4.1	总体设计	88
4.1.1	硬件配置	88
4.1.2	软件环境	88
4.1.3	资源规划	89
4.1.4	部署流程	89
4.1.5	游戏架构改造	90
4.2	环境准备	91
4.3	安装客户端验证服务	104

4.4	应用访问方式	106
4.5	伸缩应用服务	108
4.6	升级应用服务	112
4.7	应用服务迁移调度及亲和、反亲和性	113
4.8	开发全流程对比分析	116
4.8.1	传统开发全流程	116
4.8.2	云容器开发全流程	117
4.9	实践总结	118
5	常见问题	120
5.1	云容器引擎常见的使用场景如下?	120
5.2	什么是环境变量?	120
5.3	集群删除之后相关数据能否再次找回?	120
5.4	节点状态与应用状态不一致?	120
5.5	在已有的节点创建应用时未重新拉取镜像?	120
5.6	如何绑定弹性 IP?	121
5.7	如何解决新增节点时提示弹性 IP 不足的问题?	121
5.8	创建容器应用有哪几种方式? 有什么区别?	121
5.9	镜像、容器、应用的关系是什么?	121
5.10	如何获取长期有效的 DOCKER LOGIN 指令?	122
5.11	如何通过 KUBECTL 命令行创建负载均衡?	125
6	参考知识	128

6.1	CONFIGMAP 配置项要求.....	128
7	修订记录.....	129

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:

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。

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。

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：

以 root 用户登录 Docker 所在的云主机；

创建一个名为 Dockerfile 的文件；

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

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；

构建 Docker 镜像

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

执行以下命令，可查看到已成功部署的 nginx 镜像，版本为 v3；

```
docker images
```


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，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。

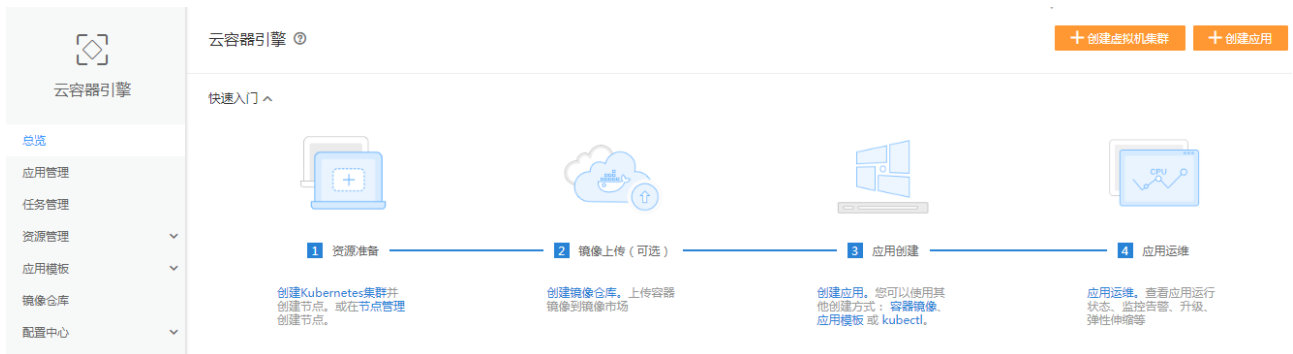
具体的操作步骤如下：

注册并登录天翼云 <http://www.ctyun.cn> ；

选择控制中心，切换到具体节点，如杭州节点；

单击服务列表 ，选择【计算】【云容器引擎】，进入容器服务界面；

在总览页面，单击【创建 Kubernetes 集群】按钮；



输入集群名称 **test001**，其余参数保持默认，单击【下一步】。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；



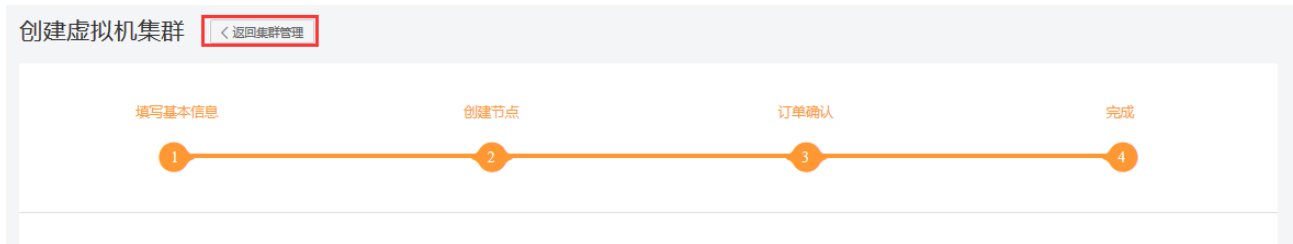
配置添加节点的参数。除网络和登录外，其余参数保持默认：

网络：弹性 IP 选择【现在购买】，将节点设为可被互联网访问；

登录：选择已创建的密钥对，用于登录节点的密钥。若未创建密钥对，请参见 3.3.2 创建密钥对创建；

单击【提交订单】。查看订单详情无误后，单击【创建】；

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，点击【返回集群管理】链接。



单击【资源管理】【节点管理】，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。创建节点时，选择购买弹性 IP，表示该节点已绑定弹性 IP；

操作步骤：

登录天翼云控制中心云容器引擎页面；


单击左侧导航栏的【应用管理】；



在【无状态应用】页签下，单击【创建应用】按钮；

入以下参数，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

单击【下一步】，进入添加容器页面。单击  **选择镜像**，在【开源镜像】页签中，搜索 nginx 的镜像，勾选后单击【确定】；

单击【下一步】；

单击【添加访问方式】，参照下表输入应用访问配置参数。本例中，将 nginx 应用设置为“通过弹性 IP 的方式”被外部互联网访问；

参数	说明
内部访问域名	输入应用发布的可被外部访问的名称，设置为：nginx
访问方式	外部访问
访问类型	弹性 IP
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	自动生成，系统会自动分配端口号
协议	TCP

添加服务 ✕

服务名称

访问方式 集群内访问 VPC内网访问 公网访问
表示应用可以被公网访问。

访问类型 负载均衡 弹性IP
弹性IP：可将互联网访问流量转发到指定节点上。


端口配置	协议	容器端口	访问端口	操作
	TCP	<input type="text" value="81"/>	自动生成	删除

[+ 添加Service端口配置](#)

确定
取消

单击【下一步】，高级设置不需要配置，单击【创建】。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；

获取新建的 nginx 应用后的外部访问地址；



云容器引擎

应用管理 + 创建应用

应用管理 ⊙

无状态应用 有状态应用

删除应用
集群: fgs-cn-j...
选择命名空间 (3)
全部状态
应用名称搜索

应用名称	状态	外部访问地址	实例个数(正常/全部)	命名空间	创建时间	操作
nginx	运行中	180.101.146.84:31699	2/2	default	2019-08-22 16:22:10 GMT...	伸缩 升级 监控 更多

在浏览器中输入“外部访问地址”，即可成功访问应用；

2.3 配置弹性伸缩策略

云容器引擎 CCE 支持“指标告警”、“定时”、“周期”的弹性伸缩策略，满足不同业务场景需求。本节以定时策略为例，指导用户如何为应用配置伸缩策略。

操作步骤：

登录天翼云控制中心云容器引擎页面；

单击左侧导航栏的【应用管理】；

单击新建的 nginx 应用，进入应用详情页面，选择【伸缩】页签；

云容器引擎

总览

应用管理

资源管理

镜像仓库

配置中心

应用名称	nginx	类型	无状态应用
实例个数(正常/全部)	1/1	外部访问地址	122.237.103.211:32188
创建时间	2018/01/27 19:03:04 GMT+08:00	标签	标签管理
描述	-- 编辑		

实例列表 访问方式 更新升级 伸缩 调度策略 事件 应用日志

您可以根据业务需求设置自动伸缩策略，以便在业务高峰时自动增加应用实例，并在低谷时回收相关资源，从容应对业务高峰压力，帮助您节约资源和人力成本。

手动伸缩

实例数: 1 个 [编辑](#)

优雅缩容策略

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。若未设置，则默认设置为30秒。

单击【添加伸缩策略】，此处以定时策略为例，输入策略名称、触发时间、执行动作及数量；

添加伸缩策略 ✕

策略名称

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

告警策略
定时策略
周期策略

触发时间 ✕ | 📅

定时策略的触发时间不能小于系统的当前时间

执行操作 增加 ▾ 个实例

确定
取消

单击【确定】后，可查看到该策略已启用；



The screenshot shows the '云容器引擎' (Cloud Container Engine) console. The left sidebar contains navigation options: 总览, 应用管理, 任务管理, 资源管理, 应用模板, 镜像仓库, and 配置中心. The main area is titled '描述' and has tabs for 监控, 实例列表, 访问方式, 更新升级, 伸缩, 迁移调度, 应用运维, and 事件. The '伸缩' (Scaling) tab is active, showing '手动伸缩' (Manual Scaling) with 2 instances and '弹性伸缩' (Elastic Scaling) with a '添加伸缩策略' (Add Scaling Policy) button. Below is a table of scaling policies:

策略名称	执行动作	状态	冷却时间 (s)	策略类型	创建时间	触发条件	操作
as-time	增加 2 个...	已...	60	定时策略	2019-08-22 16:33:...	2019-08-22 16:35:00 GMT+...	删除

在触发时间后，选择应用详情中的【实例列表】页签，可查看到应用已增加一个实例；

2.4 应用运维监控


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：

登录天翼云控制中心云容器引擎页面；

单击左侧导航栏的【应用管理】；

单击新建的 nginx 应用，进入应用详情页面；

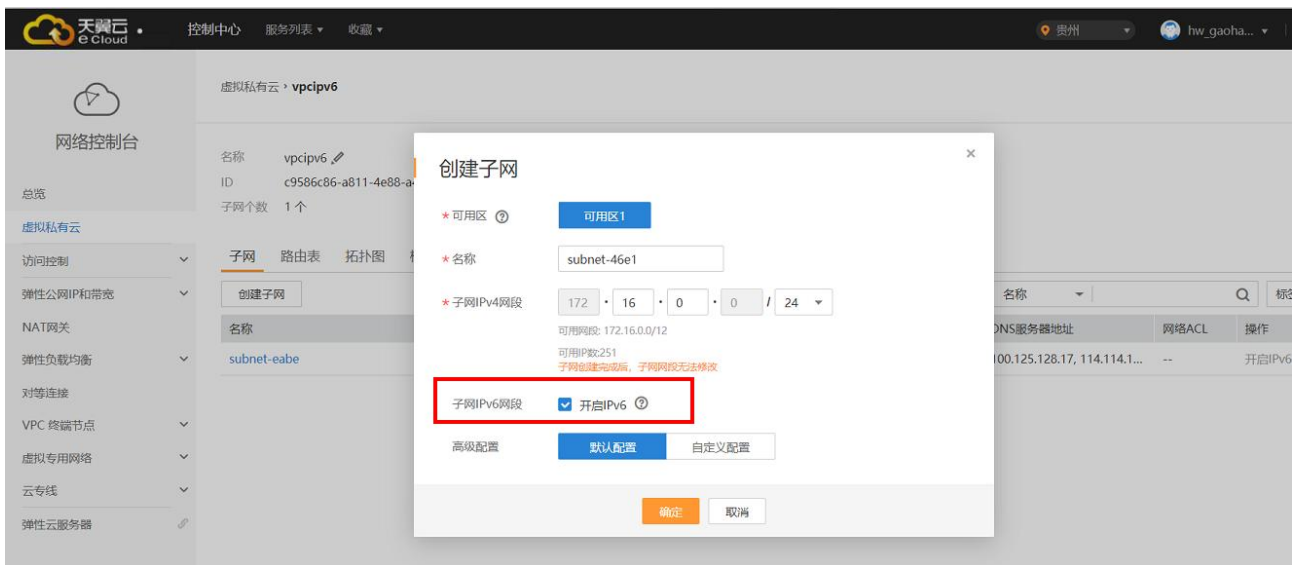
在实例列表页签中，单击实例名称前的 ，在监控页签下，可查看到应用实例的 CPU 使用率和内存使用率。只有实例处于运行状态时，才可以查看 CPU 使用率和内存使用率。

2.5 创建 IPv6 集群部署 IPv6 应用

以下示例步骤演示了通过创建一个 IPv6 的 CCE 集群后，在集群上部署 IPv6 应用并通过公网访问。

步骤 1 创建 IPv6 子网

在 VPC 服务中，创建并开启 IPv6 的子网，如下图

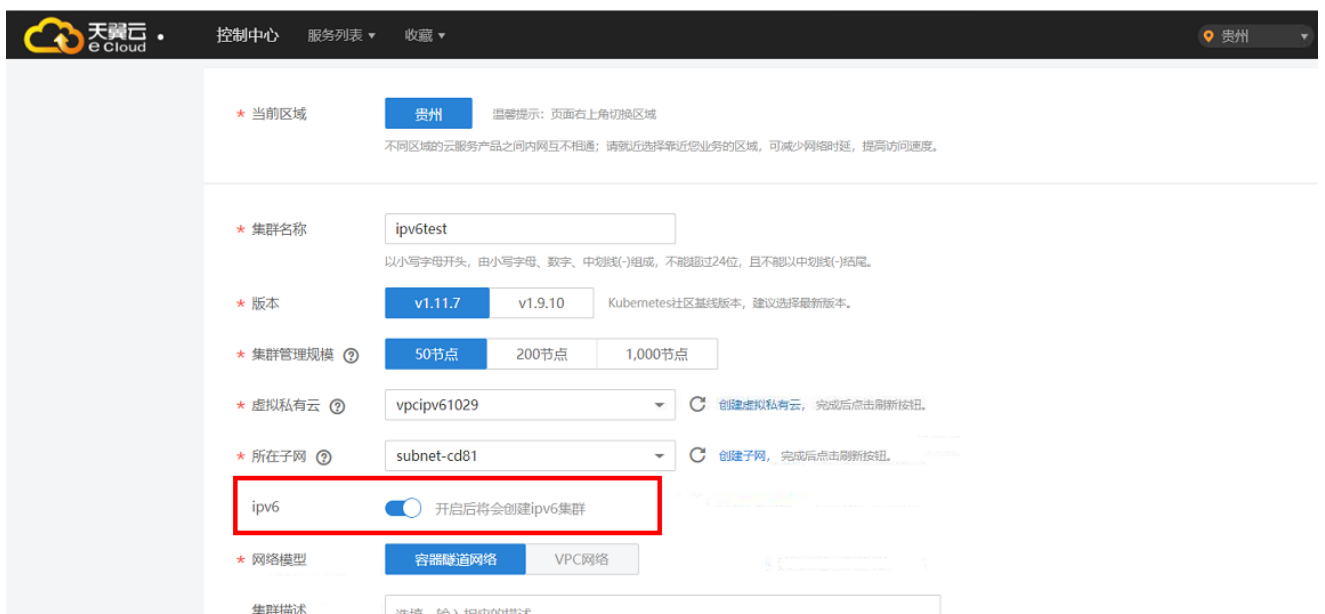


步骤 2 登录 CCE 控制台创建集群

单击左侧栏目树中的“资源管理 > 集群管理”，按步骤创建 IPv6 集群及节点，注意选择到步骤步骤 1 中创建的子网后开启 IPv6。

说明

- 1、容器引擎版本在 V1.11.7 开始支持 IPv6 网络能力，提供 IPv6 控制开关，开启后将会创建 ipv6 集群；
- 2、开启 IPv6 后，默认为普通集群管理模式，集群管理高可用选项暂不支持；
- 3、开启 IPv6 后，网络模型中默认为容器隧道网络，VPC 网络暂不支持；



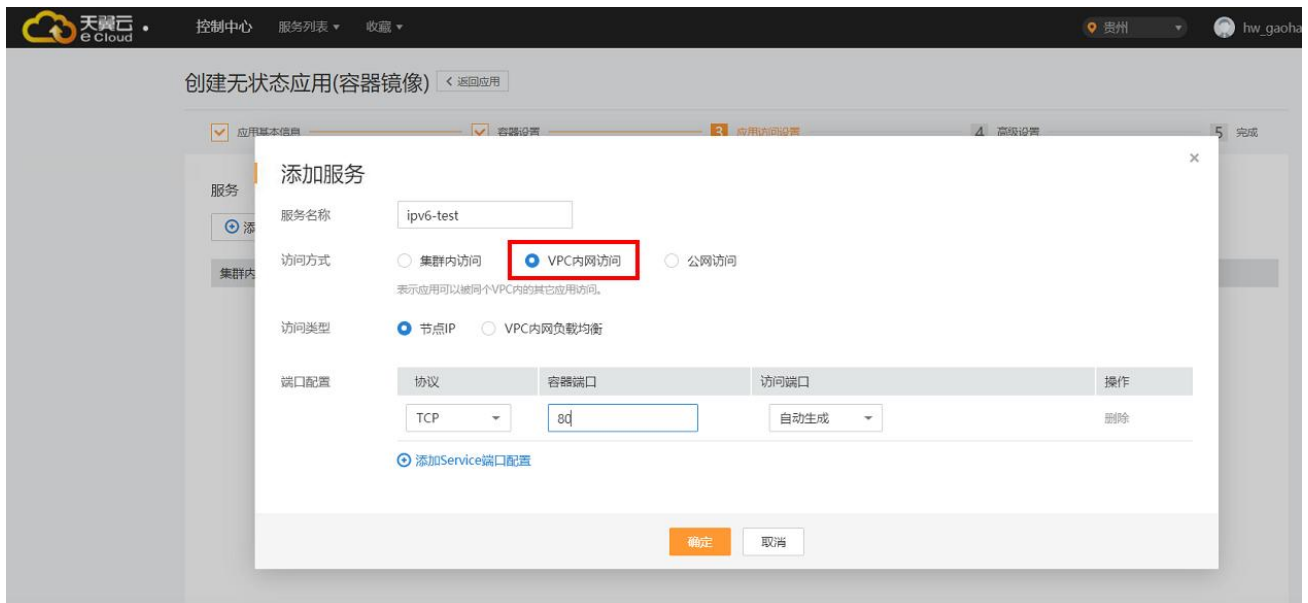
步骤 3 创建 IPv6 应用

单击左侧栏目树中的“应用管理”，单击并创建 IPv6 应用，并发布为“VPC 内网访问”服务。

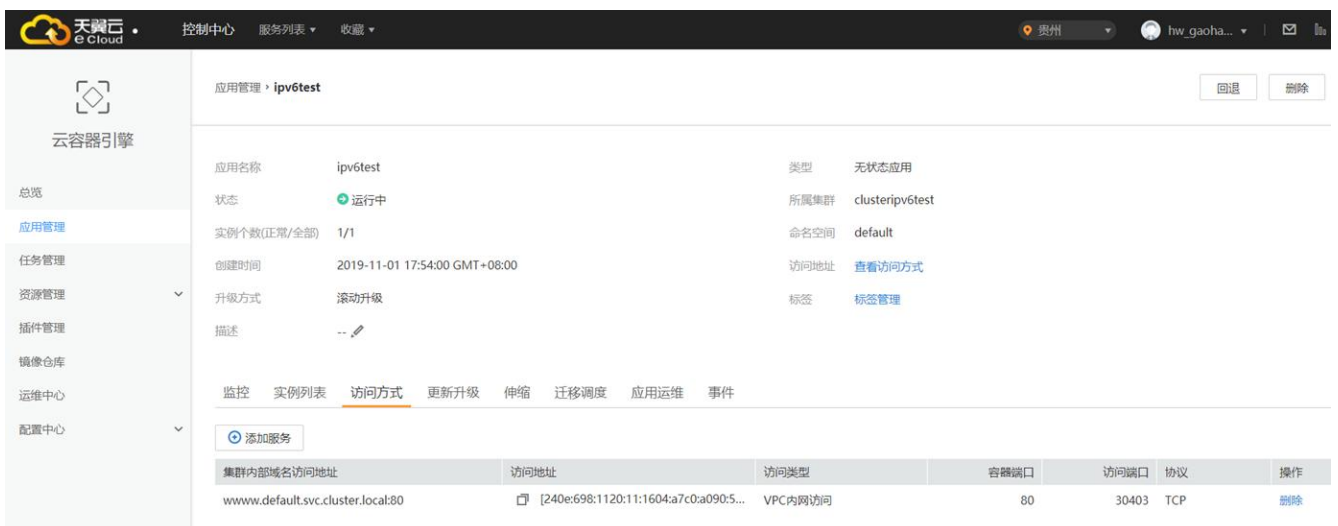
说明

如果该应用需要访问其他无状态应用或者有状态应用，需要该应用也支持 IPv6 的访问方式，并在本应用对应的配置中修改为 IPv6 的访问地址。

创建无状态应用

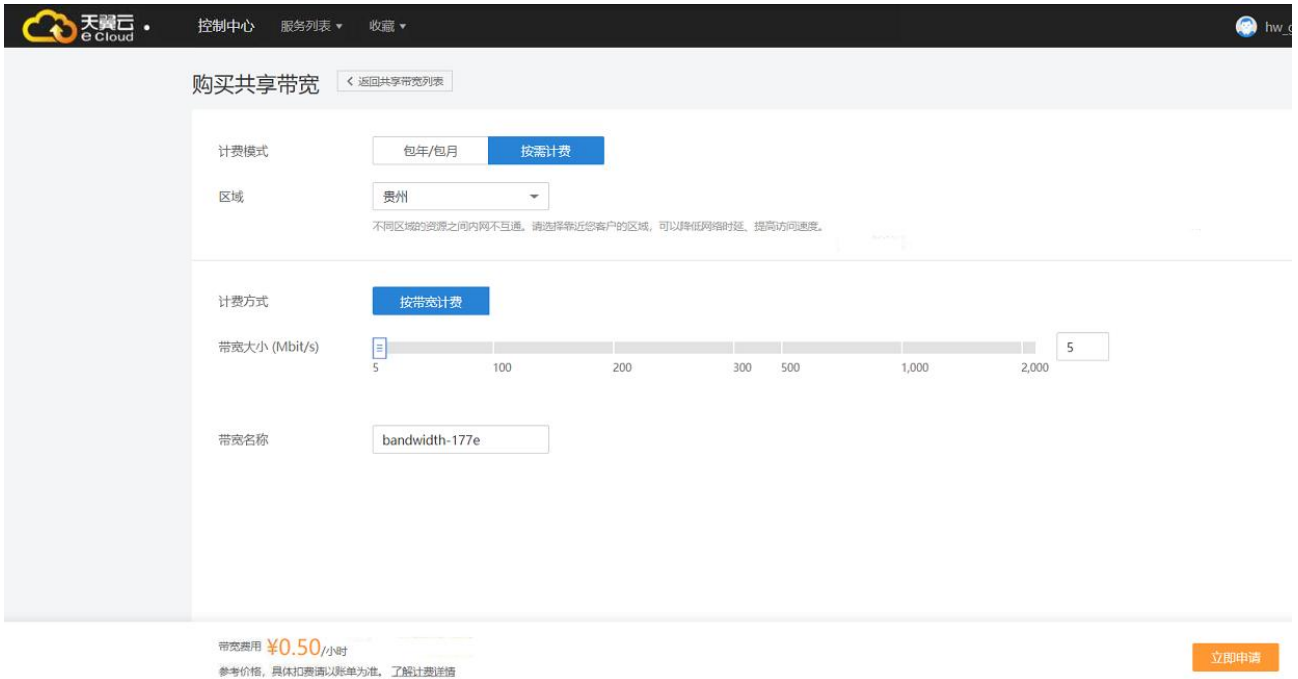


查看访问方式：



步骤 4 给 CCE 集群的 nod 节点的 IPv6 IP 绑定共享带宽，用于后续应用的公网访问。

购买共享带宽



天翼云 e Cloud 控制中心 服务列表 收藏

购买共享带宽

计费模式: 包年/包月 按需计费

区域:

计费方式: 按带宽计费

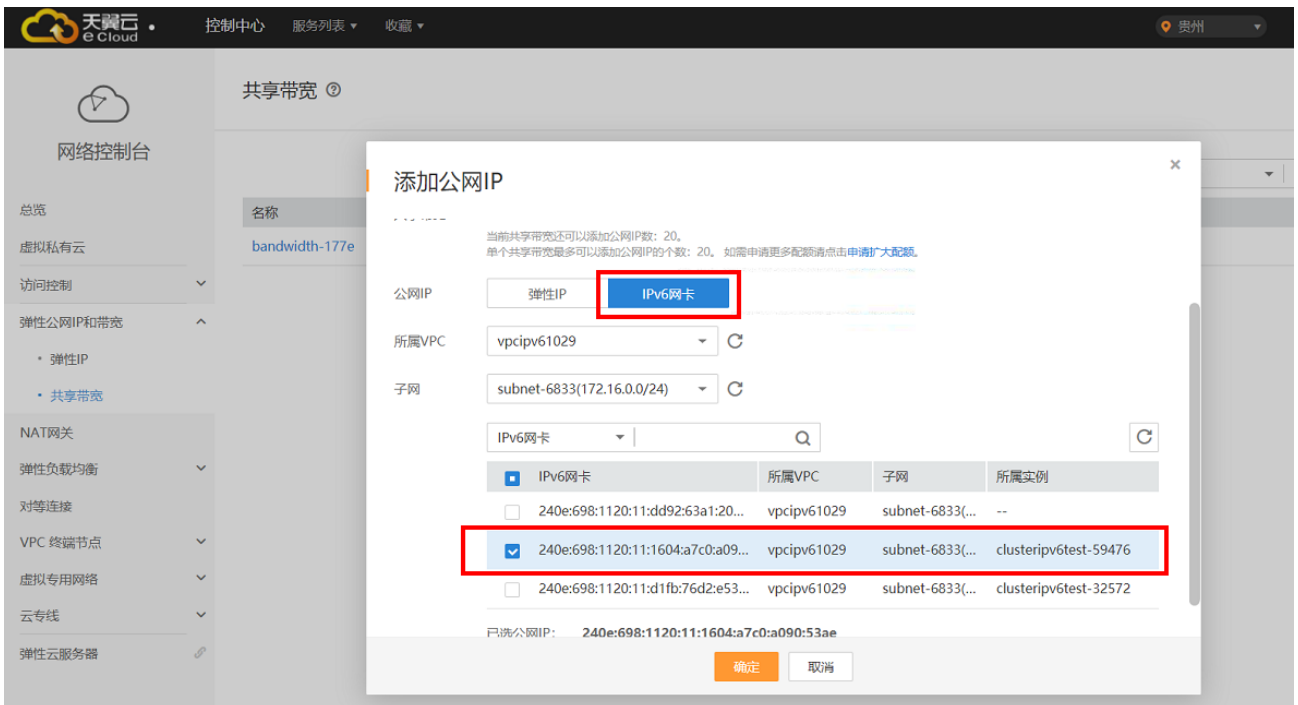
带宽大小 (Mbit/s): (Range: 5 to 2,000)

带宽名称:

带宽费用: **¥0.50/小时**

[了解计费逻辑](#) 立即申请

添加公网 IP



添加公网IP

当前共享带宽还可以添加公网IP数: 20, 单个共享带宽最多可以添加公网IP的个数: 20, 如需申请更多配额请点击[申请扩大配额](#).

公网IP: 弹性IP IPv6网卡

所属VPC:

子网:

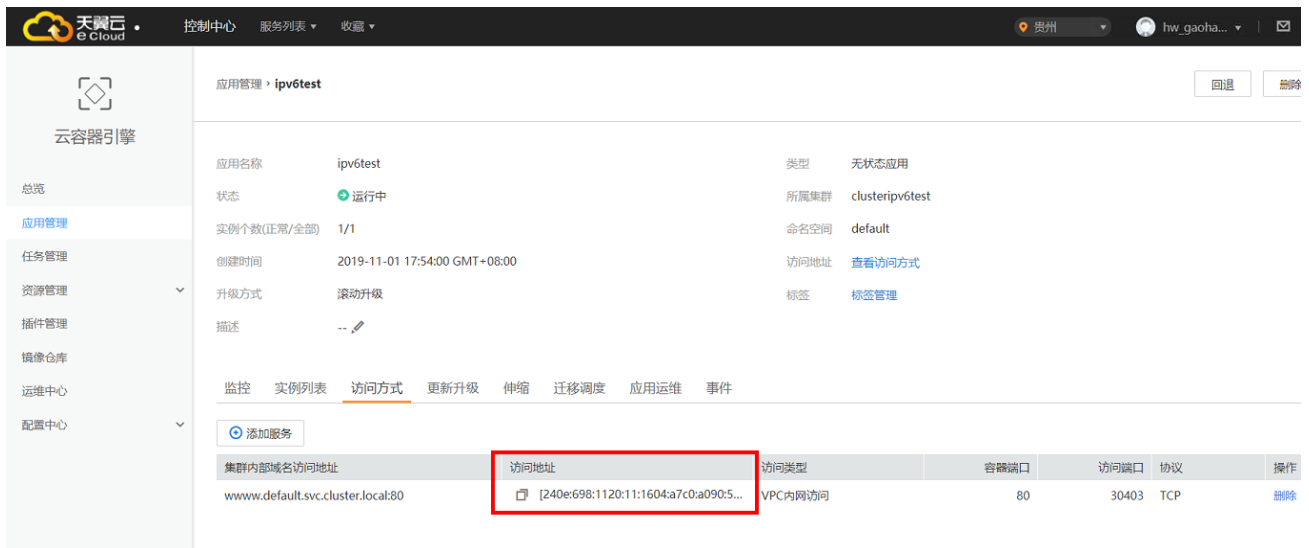
IPv6网卡:

<input type="checkbox"/>	IPv6网卡	所属VPC	子网	所属实例
<input type="checkbox"/>	240e:698:1120:11:dd92:63a1:20...	vpcipv61029	subnet-6833(...)	--
<input checked="" type="checkbox"/>	240e:698:1120:11:1604:a7c0:a09...	vpcipv61029	subnet-6833(...)	clusteripv6test-59476
<input type="checkbox"/>	240e:698:1120:11:d1fb:76d2:e53...	vpcipv61029	subnet-6833(...)	clusteripv6test-32572

已选公网IP: 240e:698:1120:11:1604:a7c0:a090:53ae

确定 取消

确认访问方式



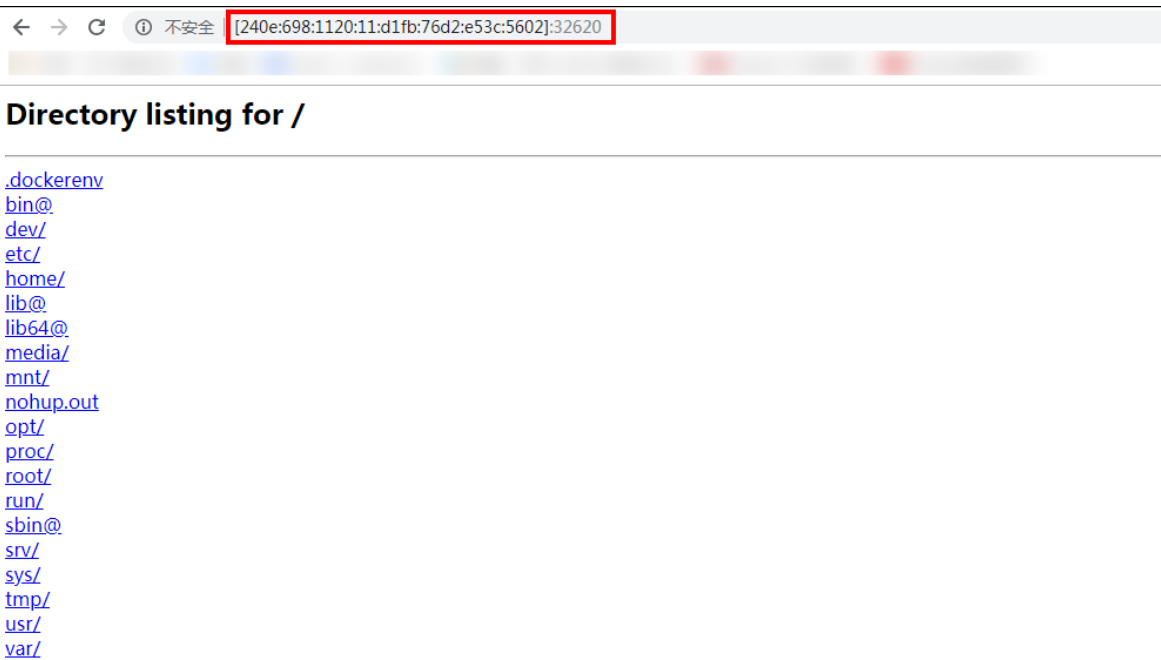
The screenshot shows the '应用管理' (Application Management) page for an application named 'ipv6test'. The application is in a '运行中' (Running) state. The '访问方式' (Access Method) tab is selected, showing a table of access configurations. A red box highlights the '访问地址' (Access Address) column, which contains the IPv6 address '[240e:698:1120:11:d1fb:76d2:e53c:5602]:32620'.

集群内部域名访问地址	访问地址	访问类型	容器端口	访问端口	协议	操作
www.default.svc.cluster.local:80	[240e:698:1120:11:d1fb:76d2:e53c:5602]:32620	VPC内网访问	80	30403	TCP	删除

步骤 5 通过支持 IPv6 的浏览器公网访问 IPv6 应用。

浏览器是否支持 IPv6 的测试：在浏览器地址栏输入网址 “<http://test-IPv6.com/>”，

在页面会给出您的 IPv6 网络测试结果。



The screenshot shows a browser window with the address bar containing the IPv6 address '[240e:698:1120:11:d1fb:76d2:e53c:5602]:32620'. The page content displays a directory listing for the root directory, including files like '.dockerenv', 'bin@', 'dev/', 'etc/', 'home/', 'lib@', 'lib64@', 'media/', 'mnt/', 'nohup.out', 'opt/', 'proc/', 'root/', 'run/', 'sbin@', 'srv/', 'sys/', 'tmp/', 'usr/', and 'var/'.

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、集群的数量、状态等信息。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、集群的数量、状态等信息；

应用管理：应用是可部署的软件实体，包含一个或一组容器。平台能够为各种类型的应用（有状态/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，如配置、更新、升级、卸载等；

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；

存储管理：存储管理即容器存储管理服务，主要负责不同后端存储服务的统一管理；

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。

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	创建集群前，您需要进行必要的环境设置。 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。 <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	CCE 支持两种部署方式，用户可基于自身需求选择。 <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。 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	CCE 支持无状态容器应用和有状态容器应用。 <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。 有状态应用：运行中不断产生数据，例如“Mysql”，它需要存储产生的新数据。CCE 利用云硬盘等的存储能力提供高可用存储卷，完美解决了传统中间件等“有状态”应用的容器化部署难题。
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。支持灵活的弹性伸缩策略，可按性能、时间、周期自由组合设置弹性伸缩策略，帮助客户轻松应对多场景业务突发浪涌。

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。

注册并登录天翼云 <https://www.ctyun.cn>；

选择控制中心，切换到具体节点，如杭州节点；

单击 ，选择【网络】【虚拟私有云】，进入虚拟私有云页面；

在【总览】界面，单击【申请虚拟私有云】，创建虚拟私有云；

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。如果用户已有密钥对，可重复使用，不需多次创建。

登录天翼云 <https://www.ctyun.cn>；

选择控制中心，切换到具体节点，如杭州节点；

单击 ，选择【计算】【弹性云主机】；

选择左侧导航中的【密钥对】，单击【创建密钥对】；

输入密钥对名称后，单击【确定】；

在弹出的对话框中，单击【确定】；

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：

登录天翼云 <https://www.ctyun.cn>；

选择控制中心，切换到具体节点，如杭州节点；

单击 ，选择【计算】【云容器引擎】，进入容器服务界面；

单击【资源管理】【虚拟机集群】，点击右侧界面的【创建 Kubernetes 集群】按钮；

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
*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“新建虚拟私有云”进行创建，操作步骤请参见 3.3.1 创建虚拟私有云。
*所在子网	节点虚拟机运行的子网环境。
*网络模型	<ul style="list-style-type: none"> 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。 VPC 网络：采用 VPC 路由的方式与底层网络深度整合，可以获得更高的网络性能，当前可以默认支持 50 节点集群规模。
集群描述	新建容器集群的描述信息。

配置完成后，单击【下一步】，添加节点；

配置添加节点的相关参数，如下表：

参数	参数说明
地域	
当前区域	展示集群部署的区域。区域指的是节点实例所在的物理位置。
可用分区	择集群部署的可用分区。指在同一地域下，电力、网络隔离的物理区域，可用分区之间内网互通，不同可用分区之间物理隔离。如果您需要提高应用的高可靠性，建议您将云服务器创建在不同的可用分区。
规格	
节点名称	请输入节点名称。
节点类型	<ul style="list-style-type: none"> 通用型：通用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应用等场景。
CPU/内存	新建节点的 CPU 配额和内存配额，请根据业务需求来选择。
购买数量	节点数量。
网络	

参数	参数说明
注意：若新增节点有互联网访问的需求，请选择“购买或使用已有弹性 IP”。若节点未绑定弹性 IP，在该节点上运行的应用将不能被外网访问。	
弹性 IP	弹性 IP 是独立申请的公网 IP 地址，如新增节点有互联网访问需求，请购买或使用已有弹性 IP。 <ul style="list-style-type: none"> 暂不使用：不使用弹性 IP 的节点不能与互联网互通，仅可作为私有网络中部署业务或者集群所需云主机进行使用。 现在购买：自动为每台云主机分配独享带宽的弹性 IP。创建弹性云主机过程中，请确保弹性 IP 配额充足。请根据界面要求，选择规格、计费模式、带宽。 使用已有：为当前节点分配已有弹性 IP，请选择已有的弹性 IP。
磁盘	分为系统盘和数据盘。 <ul style="list-style-type: none"> 系统盘的规格为 40GB，用户不可配置。 数据盘的规格为 [100, 32768]GB，用户可以配置，缺省值为 100GB。 提供超高 I/O，高 I/O，普通 I/O 三种性能规格的云硬盘。 普通 I/O：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 I/O：提供可达到 3000 的高 I/O 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 I/O：提供可达到 20000 的超高 I/O 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。
登录 > 密钥对	密钥对用于远程登录节点时的身份认证，请选择已有密钥对。若没有密钥对，可单击“查看秘钥”创建。

确认费用后，单击【提交订单】。查看订单详情无误后，单击【创建】。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

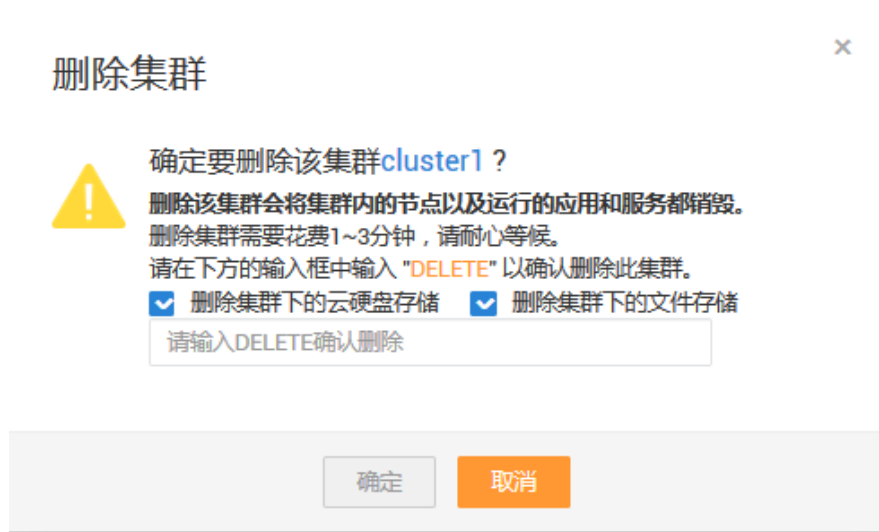
删除集群会将集群内的节点以及运行的应用和服务都销毁，请谨慎操作。

操作步骤：

控制中心中，单击 ，选择【计算】【云容器引擎】，进入容器服务界面；

单击【资源管理】【虚拟机集群】；

单击待删除集群后的【删除集群】。根据系统提示进行删除操作；



3.4.2 节点管理

3.4.2.1 新增节点（虚拟机）

节点是指接入到平台的计算资源，包括虚拟机，后续还将支持物理机。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。本节指导用户通过 CCE 新建弹性云主机节点。

操作步骤：

登录天翼云 <https://www.ctyun.cn> ；

选择控制中心，切换到具体节点，如杭州节点；

单击 ，选择【计算】【云容器引擎】，进入容器服务界面；

单击【资源管理】【虚拟机集群】，单击需要添加节点集群后的【创建节点】；



参照下表设置新增节点参数，其中带“*”标志的参数为必填参数；

参数	参数说明
地域	
当前区域	展示集群部署的区域。区域指的是节点实例所在的物理位置。
可用分区	选择集群部署的可用分区。指在同一地域下，电力、网络隔离的物理区域，可用分区之间内网互通，不同可用分区之间物理隔离。如果您需要提高应用的高可靠性，建议您将云服务器创建在不同的可用分区。
规格	
节点名称	请输入节点名称。
节点类型	<ul style="list-style-type: none"> 通用型：通用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应用等场景。
CPU/内存	新建节点的 CPU 配额和内存配额，请根据业务需求来选择。
购买数量	节点数量。
网络	
注意：若新增节点有互联网访问的需求，请选择“购买或使用已有弹性 IP”。若节点未绑定弹性 IP，在该节点上运行的应用将不能被外网访问。	
弹性 IP	弹性 IP 是独立申请的公网 IP 地址，如新增节点有互联网访问需求，请购买或使用已有弹性 IP。 <ul style="list-style-type: none"> 暂不使用：不使用弹性 IP 的节点不能与互联网互通，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。 现在购买：自动为每台云服务器分配独享带宽的弹性 IP。创建弹性云服务器过程中，请确保弹性 IP 配额充足。请根据界面要求，选择规格、计费模式、带宽。 使用已有：为当前节点分配已有弹性 IP，请选择已有的弹性 IP。

参数	参数说明
磁盘	分为系统盘和数据盘。 <ul style="list-style-type: none"> 系统盘的规格为 40GB，用户不可配置。 数据盘的规格为[100, 32768]GB，用户可以配置，缺省值为 100GB。 提供超高 I/O，高 I/O，普通 I/O 三种性能规格的云硬盘。 普通 I/O：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 I/O：提供可达到 3000 的高 I/O 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 I/O：提供可达到 20000 的超高 I/O 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。
登录 > 密钥对	密钥对用于远程登录节点时的身份认证，请选择已有密钥对。若没有密钥对，可单击“查看秘钥”创建。

设置完成后，单击【提交订单】。确认订单无误后，单击【创建】。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：

A、若创建节点时，提示弹性 IP 配额不足，请参照“4.2 如何解决新增节点时提示弹性 IP 不足的问题？”方法解决。

B、添加节点时会自动创建云主机。若创建失败，该失败回滚过程会根据公有云主机的统一收费策略来收取费用。若出现失败回滚类的收费，支持填写工单申请退款；

3.4.2.2 删除节点

删除该节点会将节点以及节点内运行的应用和服务都销毁，请谨慎操作

操作步骤：

登录天翼云 <https://www.ctyun.cn> ；

选择控制中心，切换到具体节点，如杭州节点；

单击 ，选择【计算】【云容器引擎】，进入容器服务界面；

单击【资源管理】【虚拟机集群】，单击需要【节点管理】；

单击节点后的【删除】;

根据系统提示进行删除操作;



3.4.2.3 为节点添加标签

标签是以 key/value 键值对的形式附加节点上的。添加标签后，可通过标签对节点进行管理和选择，主要用于设置亲和性与反亲和性调度。您可以给多个节点打标签，也可以给指定的某个节点打标签。

场景：


创建一个容器应用，该应用需要部署在 Node1 上。

方法为：先为 Node1 添加标签，例如标签为“apptype:web”。创建该应用时，通过亲和性调度，选择标签为“apptype:web”的节点。以下展示了此场景的操作方法。

操作步骤：

登录天翼云 <https://www.ctyun.cn> ;

选择控制中心，切换到具体节点，如杭州节点；

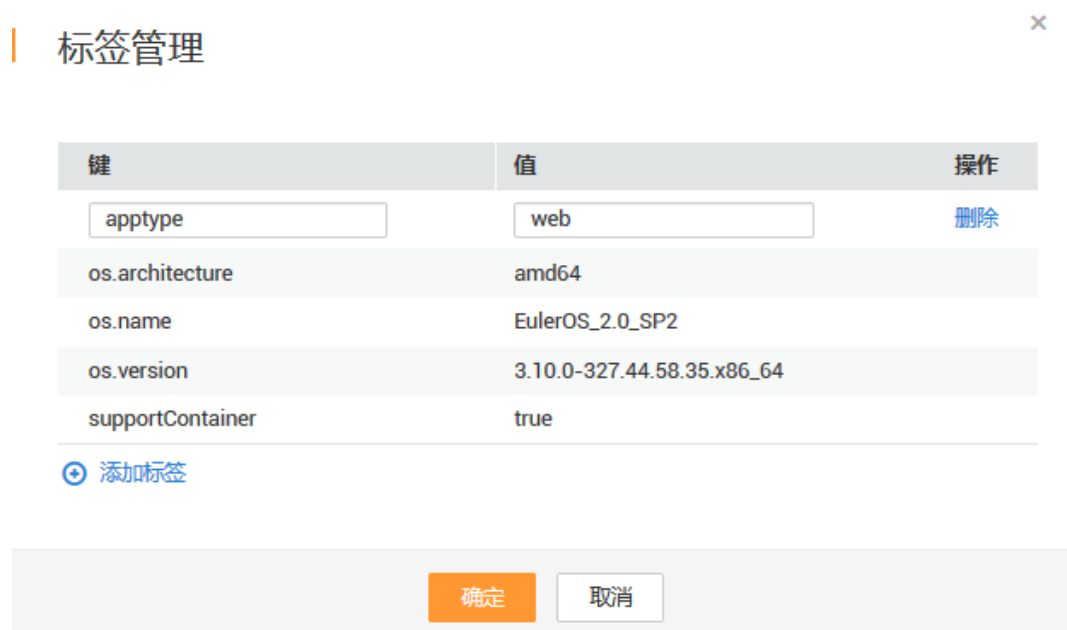
单击 ，选择【计算】【云容器引擎】，进入容器服务界面；

单击【资源管理】【节点管理】；

单击节点后的【标签管理】；



单击【添加标签】，输入键 **apptype** 和值 **web**，单击【确定】；



键	值	操作
apptype	web	删除
os.architecture	amd64	
os.name	EulerOS_2.0_SP2	
os.version	3.10.0-327.44.58.35.x86_64	
supportContainer	true	

创建应用时，通过亲和性调度选择部署在该节点中，亲和性调度说明请参见 3.5.4 设置应用调度策略；


3.4.2.4 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：

登录天翼云 <https://www.ctyun.cn> ；

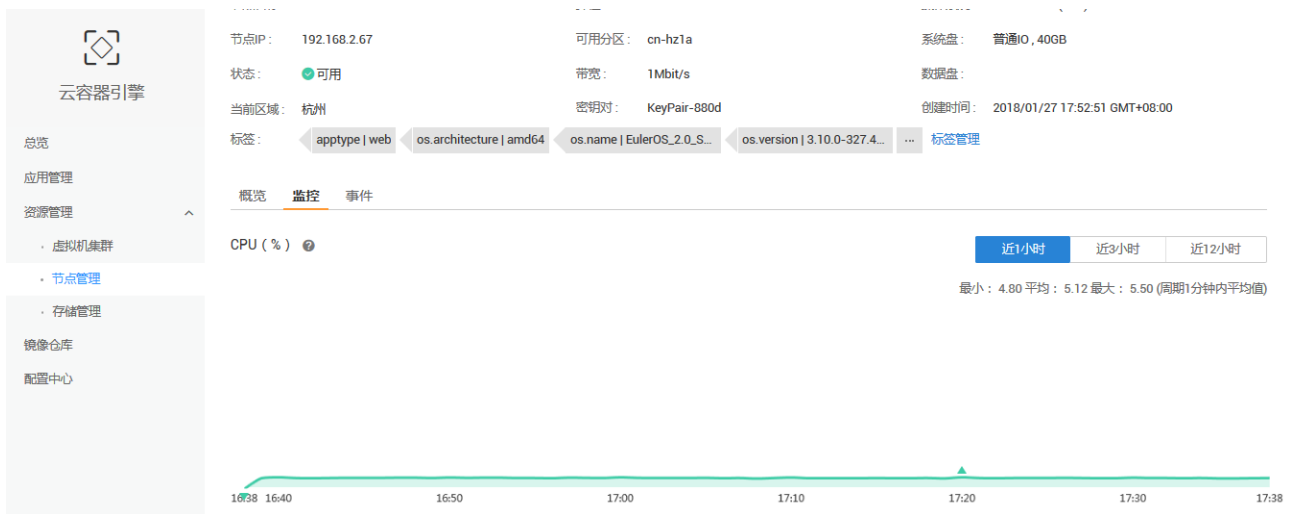
选择控制中心，切换到具体节点，如杭州节点；

单击 ，选择【计算】【云容器引擎】，进入容器服务界面；

单击【资源管理】【节点管理】；

单击待监控的节点名称，进入节点详情；

选择监控页签，可查看 CPU 和内存占用情况；



3.4.3 存储管理

创建云硬盘存储

为满足数据的持久化需求，CCE 支持云磁盘存储挂载到容器中。通过云磁盘，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，只是删除了挂载数据卷，数据卷中的数据依然保存在存储系统中。

操作步骤：

登录天翼云 <https://www.ctyun.cn> ；

选择控制中心，切换到具体节点，如杭州节点；

单击 ，选择【计算】【云容器引擎】，进入容器服务界面；

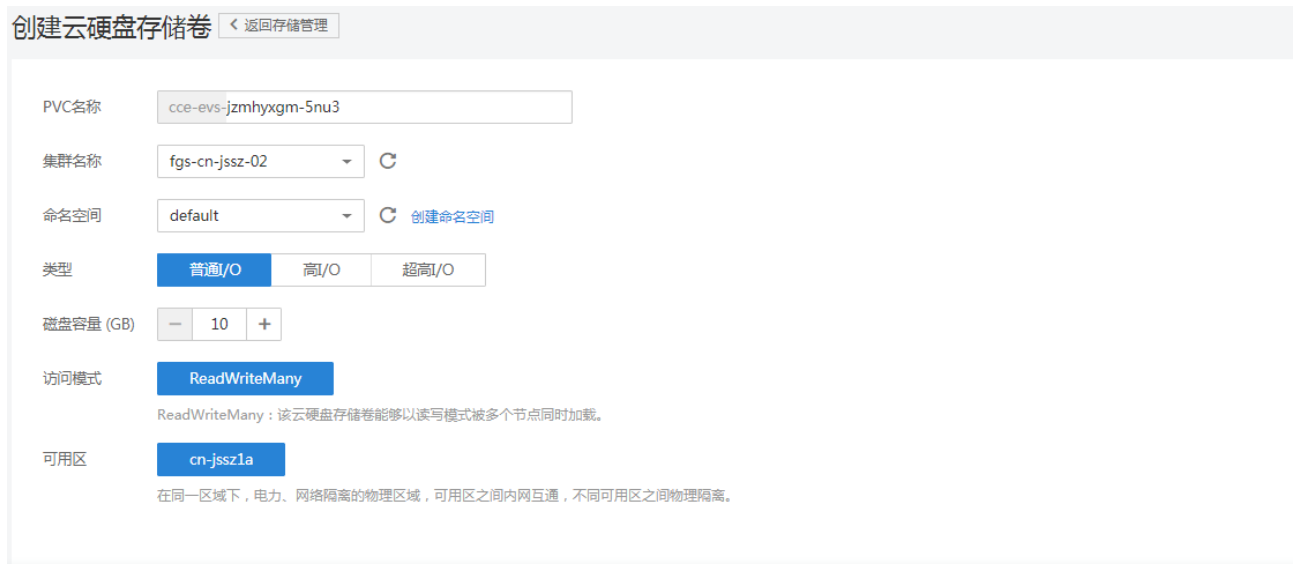
单击【资源管理】【存储管理】；

单击【创建云硬盘存储】；



配置磁盘基本信息：

参数	参数说明
名称	新建磁盘的名称。
集群	云硬盘部署所在的集群。
类型	新建云磁盘类型。 说明：通过快照创建云硬盘时，磁盘类型和快照源磁盘保持一致。
磁盘容量	新建存储的容量。



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，待状态为“可用”，表示存储已创建成功；

单击存储名称，进入存储详情页，可查看到存储的挂载信息、创建时间等信息；

选择需要删除的云硬盘，单击操作列的【删除】链接，可以删除云硬盘；

3.4.4 创建应用配置项

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：

在控制中心中，选择【计算】【云容器引擎】，进入容器服务界面；

单击左侧导航栏的【配置中心】，单击【创建配置项】；



CCE 支持“手动输入”和“上传文件”两种方式来创建配置项。

方式一：手动输入。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【添加配置】；

参数	参数说明
基本信息	
*配置名称	新建的配置名称，命名必须唯一。
*所属集群	新建配置所在的集群。
*集群命名空间	可以创建新的命名空间
描述	配置项的描述信息。
配置数据	应用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。

参数	参数说明
	<ul style="list-style-type: none"> 单击【添加更多配置数据】。 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(如应用、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。 <ul style="list-style-type: none"> 单击【添加配置标签】。 输入键、值。

方式二：上传文件。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 5.1 ConfigMap 配置项要求。

- 在创建配置项页面，选择【上传文件】页签；
- 在集群下拉框中，选择配置所属的集群；
- 在命名空间下拉框中，选择配置所属的命名空间。命名空间默认为 default；
- 单击【添加文件】，选择已创建的 ConfigMap 类型资源文件后，单击【打开】；

配置完成后，单击【添加配置】。应用配置列表中会出现新创建的应用配置；

选择需要更新的应用，单击【更新】，可以对配置项进行更新；

选择要删除的配置，单击【删除】，可以删除配置项；

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：

(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】【我的镜像】，将镜像上传

至容器镜像仓库，具体操作请参见镜像仓库；

在 GCE 左侧导航栏中选择【应用管理】，在右侧页面中选择【无状态应用】页签，单击【创建应用】；

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。
*部署集群	新建应用所在的集群。
应用组	您可以将某类应用放到同个应用组中，实现应用的批量管理（如批量启停、删除）。用户输入应用组名称后，会自动创建一个新的用户组。
*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。
应用描述	应用描述信息。

单击【下一步】，添加容器。单击 ，选择需要部署的镜像。

开源镜像：展示了所有用户上传到平台的公开镜像；

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	导入的镜像，您可单击“更换镜像”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	<p>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。详细步骤请参见 3.5.3 错误！未找到引用源。设置应用生命周期。</p> <ul style="list-style-type: none"> - 启动：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。 <p>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</p>

参数	说明
	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 3.5.7 为应用挂载数据卷。



（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 **添加容器**，再执行添加容器的操作；

单击【下一步】，设置应用访问；

A、单击【添加访问方式】，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），外部访问：参见 3.5.6 设置应用访问策略（外部访问）；

B、单击【下一步】，不设置应用访问（包括外网或内网访问）；

配置完成后，单击【下一步】；

（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；

配置完成后，单击【创建】，单击【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；

（可选）若应用需要外网访问，可通过如下方式访问应用：在应用详情页面，选择【访问方式】页签，拷贝【访问地址】一栏的【IP:端口】，在浏览器中粘贴该地址；

说明：若应用需要被外网访问，必须满足以下条件：

- A、应用所在集群已有至少 1 个节点绑定弹性 IP，即“添加节点”中“是否选择 EIP”参数选择“是”。若没有绑定 EIP，可以参照 4.6 如何绑定弹性 IP 绑定；
- B、应用访问方式需设置为“节点端口”或“负载均衡”。

3.5.2 通过镜像创建有状态容器应用

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：

（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】【我的镜像】，将镜像上传至容器镜像仓库，具体操作请参见镜像仓库；

在 CCE 左侧导航栏中选择【应用管理】，在右侧页面中选择【有状态应用】页签，单击【创建应用】；

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。
*部署集群	新建应用所在的集群。
应用组	您可以将某类应用放到同个应用组中，实现应用的批量管理（如批量启停、删除）。用户输入应用组名称后，会自动创建一个新的用户组。
*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。
应用描述	应用描述信息。

设置实例间的互相访问配置：

参数	参数说明
----	------

参数	参数说明
服务名称	输入应用所对应的服务名称，用于实例间互相访问。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。 例如一个应用有五个实例，例如应用名称为 test，此处服务名称为 service，五个实例的名称，系统会自动排序取名为：test-0、test-1、test-2、test-3、test-4。
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。

添加容器：单击  ，选择需要部署的镜像。

开源镜像：展示了所有用户上传到平台的公开镜像；

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	导入的镜像，您可单击“更换镜像”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。详细步骤请参见 3.5.3 错误！未找到引用源。设置应用生命周期。 <ul style="list-style-type: none"> - 启动：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"> 3. 在“环境变量”页签，单击“添加环境变量”。 4. 输入变量名称、变量/变量引用。 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 3.5.7 为应用挂载数据卷。



(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 **添加容器**，再执行添加容器的操作；

单击【下一步】，设置应用访问；

A、单击【添加访问方式】，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），外部访问：参见 3.5.6 设置应用访问策略（外部访问）；

B、单击【下一步】，不设置应用访问（包括外网或内网访问）；

配置完成后，单击【下一步】；

配置应用调度策略，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时时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；

配置完成后，单击【创建】，单击【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；

(可选) 若应用需要外网访问，可通过如下方式访问应用：在应用详情页面，选择【访问方式】页签，拷贝【访问地址】一栏的【IP:端口】，在浏览器中粘贴该地址；

说明：若应用需要被外网访问，必须满足以下条件：

A、应用所在集群已有至少 1 个节点绑定弹性 IP，即“添加节点”中“是否选择 EIP”参数选择“是”。若没有绑定 EIP，可以参照 4.6 如何绑定弹性 IP 绑定；

B、应用访问方式需设置为“节点端口”或“负载均衡”。

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：输入容器启动命令，容器启动后会立即执行；




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”的操作步骤：

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】页签；

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"> 单击  添加，在启动命令填框中输入可执行的程序，例如/run/server。 单击  添加，在启动命令填框中输入启动参数，例如--port=8080。 单击  添加，在启动命令填框中输入启动参数，例如--param2=value2。

“启动后处理”的操作步骤：

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；

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。如需要执行的命令如下：</p> <pre>exec: command: - /install.sh - install_agent</pre> <p>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。</p>
HttpGet 请求方式	<p>发起一个 HTTP 调用请求。配置参数如下：</p> <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：

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；

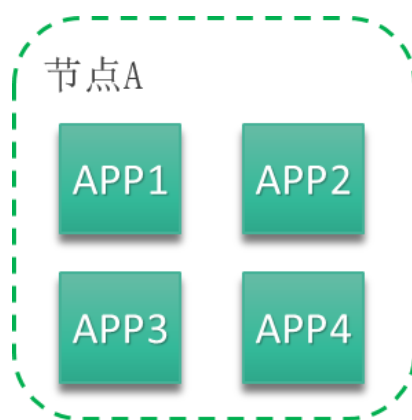
设置停止前处理的参数：

命令方式	操作步骤
可执行程序方式	<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。如需要执行的命令如下：</p> <pre>exec: command: - /install.sh - install_agent</pre> <p>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。</p>
HttpGet 请求方式	<p>发起一个 HTTP 调用请求。配置参数如下：</p> <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

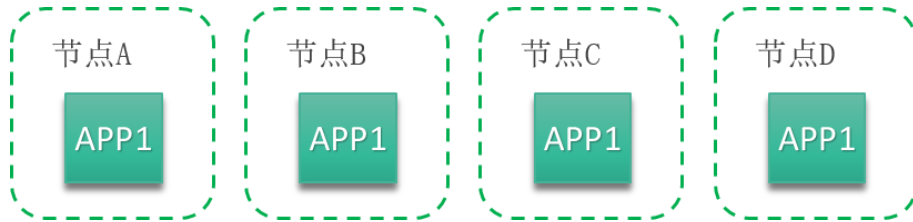
系统支持丰富的调度策略，包括静态的全局调度策略，以及动态的运行调度策略，用户可以根据需要自由组合使用这些策略来实现自己的需求。平台支持以下两种亲和性的调度方式。应用间的亲和性：决定应用部署在相同或不同节点中。

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	
与节点的亲和性	<ul style="list-style-type: none"> 在【调度策略】【应用和节点的亲和性】【与节点的亲和性】下，单击【添加】； 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【调度策略】【应用和节点的亲和性】【与节点的反亲和性】下，单击【添加】； 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【调度策略】【应用间的亲和性】【与应用的亲和性】下，单击【添加】； 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；

调度说明	操作步骤
与应用的反亲和	<ul style="list-style-type: none"> 在【调度策略】【应用间的亲和性】【与应用的反亲和性】下，单击【添加】； 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：

A、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；

B、节点私有 IP：选择此类型，系统将在每个节点上打开一个端口，可通过“节点 IP:节点端口”来访问服务；

“集群虚拟 IP”的操作步骤：

登录云容器引擎控制台，在应用访问设置中，单击【添加访问方式】；

配置服务参数：

参数	参数说明
内部访问域名	输入应用发布的可被外部访问的名称。
访问方式	内部访问。
访问类型	集群虚拟 IP。
容器端口	容器中应用启动监听的端口。
访问端口	Cluster IP 的服务端口，映射到容器的应用端口。
协议	TCP/UDP。

单击【确认】；

“节点私有 IP”的操作步骤：

登录云容器引擎控制台，在应用访问设置中，单击【添加访问方式】；

配置服务发布参数：

参数	参数说明
内部访问域名	输入应用发布的可被外部访问的名称。
访问方式	内部访问。
访问类型	节点私有 IP。
容器端口	容器中应用启动监听的端口。
访问端口	<p>容器或进程映射到节点上的端口。配置完成后，系统会在用户所在项目的所有节点上打开一个真实的端口号。</p> <ul style="list-style-type: none"> 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，默认为 30000-32767。 <p>说明：创建拥有多个 Docker 容器的应用时，请确保容器应用使用的端口不冲突，否则部署会失败。</p> <p>若选择指定端口，建议提前登录到任意一个纳管节点上（需确保已获取节点登录账号和密码），执行 <code>netstat -apn grep LISTEN</code> 命令，查询端口是否已被占用。若已被占用，需填写其他参数值。</p>
协议	TCP/UDP。

单击【确认】；

3.5.6 设置应用访问策略（外部访问）

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“外部访问”的类型：

A、负载均衡：弹性负载均衡（ELB）通过将访问流量自动分发到多台弹性云主机，扩展应用系统对外的服务能力，实现更高水平的应用程序容错性能。用户通过基于浏览器、统一化视图的云计算管理图形化界面，可以创建 ELB，为服务配置需要监听的端口，配置云主机。消除单点故障，提高整个系统的可用性；

B、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，实现弹性云主机访公网的目的；

“负载均衡”的操作步骤：

登录云容器引擎控制台，在应用访问方式设置中，单击【添加服务】；

- 访问类型：选择“负载均衡 (LoadBalancer)”。
- 服务名称：自定义服务名称，可与应用管理名称保持一致。
- 负载均衡：可以将互联网访问流量自动分发到应用管理所在的多个节点上。
- 公网：支持自动创建方式。可修改待创建增强型负载均衡实例的带宽。

 说明

- 增强型负载均衡配额不足时，不支持在云容器引擎中自动创建，请通过[新建增强型弹性负载均衡](#)创建。
- 负载均衡实例需与当前集群处于相同 VPC (如：my-vpc-9207ba6f)、相同子网 (如：my-subnet-9207ba6f)，且与选择的公网/私网类型保持一致。
- 端口配置：
 - 协议：请根据业务的协议类型选择。
 - 容器端口：容器镜像中应用管理实际监听端口，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应用管理时使用，端口范围为 1-65535，可任意指定。

- 1) 完成配置后，直接单击“确定”。
- 2) 单击“下一步”进入“高级设置”页面，直接单击“创建”。
- 3) 创建成功后，单击“应用管理 > 无状态(Deployment)”或“应用管理 > 有状态(StatefulSet)”，在应用管理列表页面，单击“应用管理名称”进入应用管理详情页，在“访问方式”页签下，获取方式地址，例如 10.4.10.230:2。
- 4) 单击访问地址，即可跳转到访问页面。

“弹性 IP”的操作步骤：

登录云容器引擎控制台，在应用访问方式设置中，单击【添加服务】；

配置服务发布参数：

参数	参数说明
服务名称	输入应用发布的可被外部访问的服务名称。
访问方式	外部访问。
访问类型	弹性 IP。
容器端口	容器中应用启动监听的端口。
访问端口	<p>容器或进程映射到节点上的端口。配置完成后，系统会在用户所在项目的所有节点上打开一个真实的端口号。</p> <ul style="list-style-type: none"> 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，默认为 30000-32767。 <p>说明：创建拥有多个 Docker 容器的应用时，请确保容器应用使用的端口不冲突，否则部署会失败。</p> <p>若选择指定端口，建议提前登录到任意一个纳管节点上（需确保已获取节点登录账号和密码），执行 <code>netstat -apn grep LISTEN</code> 命令，查询端口是否已被占用。若已被占用，需填写其他参数值。</p>
协议	仅支持 TCP。

单击【确认】；

3.5.7 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 支持挂载本地磁盘和云存储：

A、挂载本地磁盘：支持 hostPath、emptyDir、configMap 三种；

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。

B、挂载云存储：类似 Kubernetes 的第三方存储插件，指定一块云硬盘挂载到容器的某一路径下，创建数据卷时对应的云硬盘即可；

“挂载本地磁盘”的操作步骤：

登录云容器引擎控制台，在创建应用过程中，在【添加容器】【选择镜像】【数据存储】【本地磁盘】；

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【挂载】，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；

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【挂载】，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。

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 3.4.4 创建应用配置项创建 configMap。 <ul style="list-style-type: none"> 单击【挂载】，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。
权限	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。

“挂载云硬盘”的操作步骤：

登录云容器引擎控制台，在创建应用过程中，在【添加容器】【选择镜像】【数据存储】【云存储】页签中，单击【添加云存储】；

输入云存储参数配置：

参数	参数说明
云存储名称	输入存储名称。
类型	<p>当前仅支持云硬盘。</p> <p>云硬盘（EVS）的后端存储设备基于分布式架构，为云主机提供可弹性扩展的虚拟块存储设备。您通过管理控制台创建云硬盘并挂载给服务器，服务器可以为弹性云主机或者物理机，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应用。</p> <ul style="list-style-type: none"> • 使用已有存储：选择已创建的存储，创建存储步骤请参见 3.4.3 存储管理； • 自动分配存储：自动创建存储，需要输入存储的容量； <ul style="list-style-type: none"> - sas：高 IO，指由 SAS 存储介质构成的云硬盘。 - sata：普通 IO，指由 SATA 存储介质构成的云硬盘。 - ssd：超高 IO，指由 SSD 存储介质构成的云硬盘。
挂载路径	<p>单击【挂载】，输入数据卷挂载到容器上的路径。</p> <p>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。</p>
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。

3.6 应用访问设置

3.6.1 概述

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。

- 集群内访问

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。集群内部域名格式为“<自定义的访问方式名称>.<应用所在命名空间>.svc.cluster.local”，例如

“nginx.default.svc.cluster.local”。

- VPC 内访问

指应用可以让同一 VPC 内其他应用访问，通过“集群节点的 IP”或者“私网弹性负载均衡 ELB 的服务地址”访问，主要场景：云上同一 VPC 内其他应用需要访问 kubernetes 集群内部的应用。

- 公网访问-弹性 IP

通过弹性 IP 从公网访问应用，一般用于系统中需要暴露到公网的服务。该访问方式需要给集群内任一节点绑定弹性 IP，并设置一个映射在节点上的端口，其中节点端口的范围在 30000-32767 之间，例如访问地址为 10.0.0.0:30000。

- 公网访问-四层负载均衡

通过弹性负载均衡从公网访问应用，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”。

- 公网访问-七层负载均衡

与四层负载均衡的区别在于，七层负载均衡支持 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。

七层负载采用了增强型弹性负载均衡，该访问方式由公网弹性负载均衡 ELB 服务地址、设置的访问端口组成、定义的 URI 组成，例如：10.117.117.117:80/helloworld。

3.6.2 集群内访问

集群内访问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。集群

内部域名格式为“<自定义的访问方式名称>.<应用所在命名空间>.svc.cluster.local”，例如

“nginx.default.svc.cluster.local”。

添加方式

您可以在创建应用时设置访问方式，也可以应用创建完成后添加访问方式。

- 方式一：创建应用时配置；
- 方式二：应用创建完成后设置。

通过界面创建

步骤 1：创建无状态应用或创建有状态应用，在“应用访问设置”步骤，单击“添加服务”。

- 服务名称：自定义服务名称，可与应用名称保持一致。
- 访问方式：集群内访问。
- 协议：请根据应用的协议类型选择。
- 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。
- 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，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“查看应用详情”，在访问方式页签中获取访问地址，例如 10.247.74.100:2。

步骤 4：登录应用所在集群的任意节点，登录方法请参见 SSH 密钥登录方式。

步骤 5：使用 curl 命令访问应用验证应用是否可以正常访问。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 10.247.74.100:2
```

其中 10.247.74.100:2 为步骤 3 中获取的访问地址。

回显如下表示应用可正常访问。

```
<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.default.svc.cluster.local:2
```

其中 `nginx.default.svc.cluster.local` 为步骤 3 中获取的域名访问地址。

回显如下表示应用可正常访问。

```
<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>
```

应用创建完成后设置

步骤 1：登录 CCE 控制台，选择左侧导航栏的“资源管理 > 网络管理”，在 **Service** 页签下，单击“添加 Service”。选择类型为“集群内访问”。

步骤 2：设置集群内访问参数。

- 服务名称：自定义服务名称，可与应用名称保持一致。
- 集群名称：服务所在集群。
- 命名空间：服务所在命名空间。
- 关联应用：选择需要添加 Service 的应用。
- 端口配置：

- 协议：请根据业务的协议类型选择。
- 容器端口：容器镜像中应用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 访问应用时使用，端口范围为 1-65535，可任意指定。

步骤 3：单击“创建”。应用已添加“集群内访问”的服务。验证操作与步骤 4-步骤 5 相同。

通过 kubectl 命令行创建

本节以 nginx 应用为例，说明 kubectl 命令实现集群内访问的方法。

前提条件

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: 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
```

vi nginx-ClusterIp-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: nginx
    name: nginx-clusterip
spec:
  ports:
    - name: service0
      port: 2          #对应界面上的访问端口
      protocol: TCP
      targetPort: 80  #对应界面上的容器端口
  selector:
    app: nginx
  type: ClusterIP    #对应界面上的访问类型，ClusterIP 表示“集群虚拟 IP”
```

步骤 3：创建应用。

```
kubectl create -f nginx-deployment.yaml
```

回显如下，表示应用已开始创建。

```
deployment "nginx" created
```

```
kubectl get po
```

回显如下，应用状态为 Running，表示应用已处于运行中状态。

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 : 登录应用所在集群的任意节点 ;

步骤 6 : 采用 curl 命令访问应用验证应用是否可以正常访问。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 10.247.200.134:2

回显如下表示应用可正常访问。

```
<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:2

回显如下表示应用可正常访问。

```
<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>
```

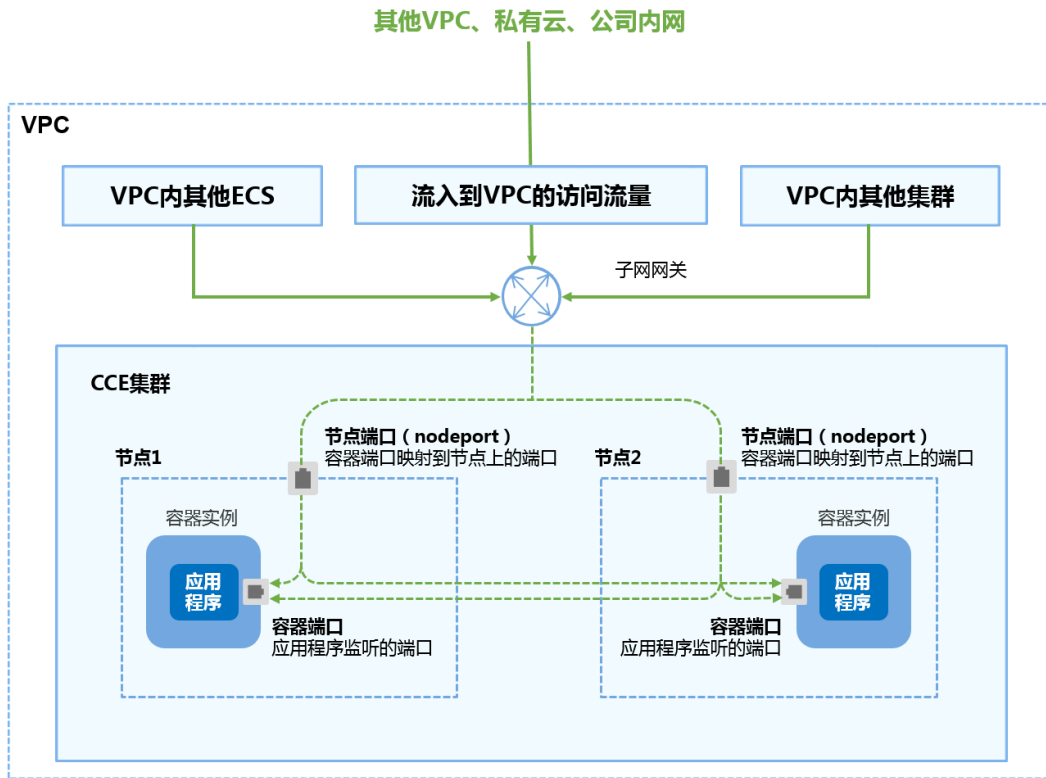
3.6.3 VPC 内网访问

VPC 内网访问是指应用可以让同一 VPC 内其他应用访问，通过“集群节点的 IP”或者“私网弹性负载均衡 ELB”的服务地址访问。

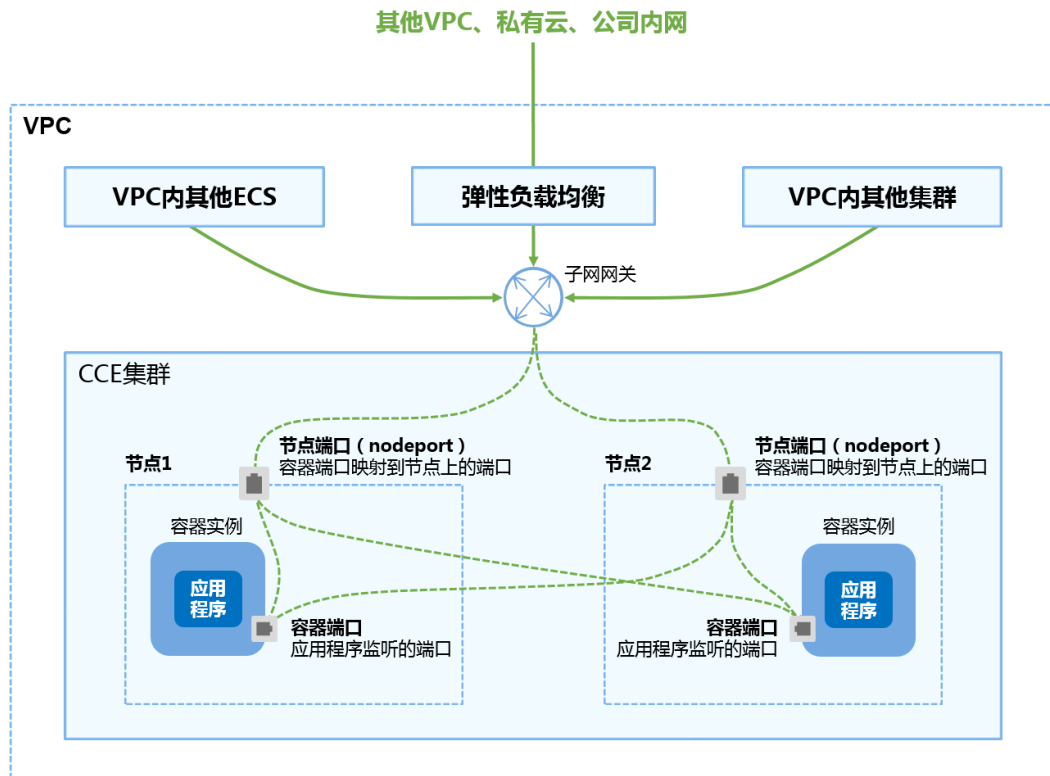
主要场景：同一 VPC 内其他应用需要访问 kubernetes 集群内部的应用。

VPC 内网访问有两种形式：

- 通过集群节点 IP 访问：如下图。



- 通过私网弹性负载均衡 ELB 访问：如下图。弹性负载均衡模式相比集群节点 IP 方式的访问，提供了高可靠的保障。



添加方式

您可以在创建应用时设置访问方式，也可以应用创建完成后添加访问方式。

- 方式一：创建应用时配置。
- 方式二：应用创建完成后设置。

通过界面创建

本节以 nginx 应用为例进行说明。

步骤 1：创建无状态应用或创建有状态应用，在“应用访问设置”步骤，单击“添加服务”。

- 服务名称：自定义服务名称，可与应用名称保持一致。
- 访问方式：VPC 内网访问。
 - 节点 IP：表示通过“集群节点 IP”来访问应用。
 - VPC 内网负载均衡：表示通过“弹性负载均衡”来访问应用。
- 协议：请根据业务的协议类型选择。
- 容器端口：容器镜像中应用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：
 - 集群节点 IP 访问：容器端口映射到节点私有 IP 上的端口，用私有 IP 访问应用时使用，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。
 - 通过弹性负载均衡器的私网 IP 访问集群节点：指定一个端口，将容器端口映射到负载均衡器的端口，使用私网负载均衡 IP 访问应用，端口范围为 1-65535。使用弹性负载均衡器的私有 IP 访问应用时，使用该端口。

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：单击查看应用详情，在访问方式页签中获取访问地址，例如“192.168.0.160:30358”。

步骤 4：在管理控制台首页，单击“计算 > 弹性云主机”。

步骤 5：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。

确认安全组开放



The screenshot shows the '安全组' (Security Group) tab in the console. It displays a table of rules for the security group 'testet-ccc-node-l33s' (ID: b2534734-efd3-4619-b1ca-4aac92407e43). The table lists rules for both outgoing and incoming traffic, with specific protocols and port ranges.

方向	类型	协议	端口范围/ICMP类型	远端
出方向	IPv4	Any	Any	Any
入方向	IPv4	TCP	22	Any
入方向	IPv4	TCP	3389	Any
入方向	IPv4	TCP	30000-32767	Any

步骤 6：单击“远程登录”，弹出登录页面，输入用户密码登录。

步骤 7：使用 curl 命令访问应用验证应用是否可以正常访问。

📖 说明

节点私有 IP 类型的访问方式在集群内也会分配一个集群虚拟 IP，即可以在集群内部通过集群虚拟 IP 的验证方式验证。其中，集群虚拟 IP 访问端口默认与容器端口一致，例如此示例的访问端口为 80 端口。

```
curl 192.168.0.160:30358
```

其中“192.168.0.160:30358”为步骤 3 中获取到的访问地址。

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>
```

应用创建完成后设置

步骤 1：登录 CCE 控制台，选择左侧导航栏的“资源管理 > 网络管理”，在 **Service** 页签下，单击“添加 Service”。选择类型为“VPC 内网访问”。

步骤 2：设置集群内访问参数。

- 服务名称：自定义服务名称，可与应用名称保持一致。
- 集群名称：服务所在集群。
- 命名空间：服务所在命名空间。
- 关联应用：选择需要添加 Service 的应用。
- 访问类型：
 - 节点 IP：表示通过“集群节点 IP”来访问集群内的节点。
 - VPC 内网负载均衡：表示选择通过“弹性负载均衡”来访问集群内的节点。
- 端口配置：
 - 协议：请根据业务的协议类型选择。

- 容器端口 容器镜像中应用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 :
 - 集群节点 IP 访问 :容器端口映射到节点私有 IP 上的端口 ,用私有 IP 访问应用时使用 ,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。
 - 通过私网弹性负载均衡 ELB 访问 :容器端口映射到负载均衡实例上的端口 ,用私网负载均衡 IP 访问应用时使用 ,端口范围为 1-65535。

步骤 3 :单击 “创建” 。应用已添加 “VPC 内网访问” 的服务。验证操作与步骤 4-步骤 7 相同。

kubectl 命令行创建

本节以 nginx 应用为例 ,说明 kubectl 命令实现 VPC 内访问的方法。

前提条件

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: 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
    
```

vi nginx-nodeport-svc.yaml

```

apiVersion: v1
kind: Service
metadata:
  labels:
    app: nginx
  name: nginx-nodeport
spec:
  ports:
    - name: service
      # nodePort: 30000      #对应界面上的访问端口，不填写表示自动生成
      port: 80             #集群虚拟 IP 的访问端口
      protocol: TCP
      targetPort: 80      #对应界面上的容器端口
  selector:
    app: nginx
  type: NodePort         #对应界面上的访问类型，NodePort 表示“节点私有 IP”
    
```

步骤 3：创建应用。

```
kubectl create -f nginx-deployment.yaml
```

回显如下，表示应用已开始创建。

```
deployment "nginx" created
```

```
kubectl get po
```

回显如下，应用状态为 Running，表示应用已处于运行状态。

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 命令访问应用验证应用是否可以正常访问。

```
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>

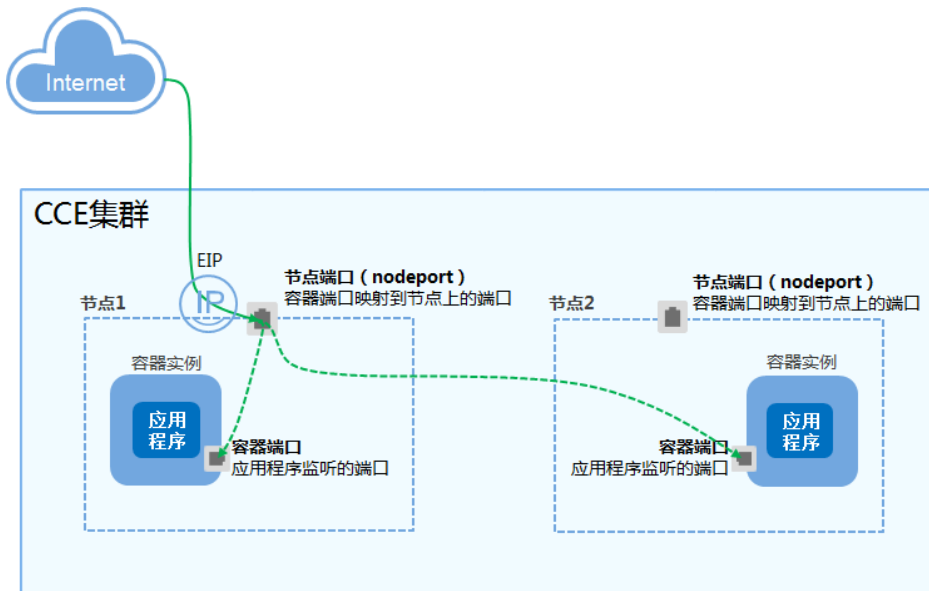
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

3.6.4 公网访问-弹性 IP

公网访问可以通过弹性 IP 从公网访问应用，一般用于系统中需要暴露到公网的服务。该访问方式需

要给集群内任一节点绑定弹性 IP，并设置一个映射在节点上的端口，其中节点端口的范围在 30000-32767 之间，例如访问地址为 10.117.117.117:30000。



添加方式

您可以在创建应用时设置访问方式，也可以应用创建完成后添加访问方式。

- 方式一：创建应用时配置。
- 方式二：应用创建完成后设置。

通过界面创建

本节以 nginx 应用为例进行说明。

步骤 1：创建无状态应用或创建有状态应用，在“应用访问设置”步骤，单击“添加服务”。

- 服务名称：自定义服务名称，可与应用名称保持一致。
- 访问方式：公网访问。
- 访问类型：弹性 IP，请确保当前集群内至少有一个节点已绑定弹性 IP。
- 协议：请根据业务的协议类型选择。
- 容器端口：容器镜像中应用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：容器端口映射到弹性 IP 上的端口，用私有 IP 访问应用时使用，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。

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“查看应用详情”，在“访问方式”页签下，获取访问地址，例如“10.78.27.59:30911”。

步骤 4：单击访问地址，即可跳转到访问页面。

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.

应用创建完成后设置

步骤 1：登录 CCE 控制台，选择左侧导航栏的“资源管理 > 网络管理”，在 **Service** 页签下，单击“添加 Service”。选择类型为“公网访问”。

步骤 2：设置集群内访问参数。

- 服务名称：自定义服务名称，可与应用名称保持一致。
- 集群名称：服务所在集群。
- 命名空间：服务所在命名空间。
- 关联应用：选择需要添加 Service 的应用。
- 访问类型：选择弹性 IP。
- 端口配置：
 - 协议：请根据业务的协议类型选择。
 - 容器端口：容器镜像中应用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，用私有 IP 访问应用时使用，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。

步骤 3：单击“创建”。应用已添加“公网访问-弹性 IP”的服务。

kubectl 命令行创建

本节以 nginx 应用为例，说明 kubectl 命令实现公网访问的方法。

前提条件

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-eip-svc.yaml 文件。

其中，nginx-deployment.yaml 和 nginx-eip-svc.yaml 为自定义名称，您可以随意命名。

vi nginx-deployment.yaml

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1cc
  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-eip-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  annotations:
```

```

    service.protal.kubernetes.io/access-ip: 10.78.44.60 #集群内部至少有一个节点绑定弹性 IP, 此处填写弹性 IP 地址
    service.protal.kubernetes.io/type: EIP #指定外部访问类型为弹性 IP
  labels:
    app: nginx
  name: nginx-eip
spec:
  ports:
  - name: service0
    nodePort: 30000 #对应界面上的访问端口, 不填写表示自动生成
    port: 80 #集群虚拟 IP 的访问端口
    protocol: TCP
    targetPort: 80 #对应界面上的容器端口
  selector:
    app: nginx
  type: NodePort #弹性 IP 需要基于 NodePort 类型的服务
    
```

步骤 3 : 创建应用。

```
kubectl create -f nginx-deployment.yaml
```

回显如下表示应用开始创建。

```
deployment "nginx" created
```

```
kubectl get po
```

回显如下, 应用状态为 Running, 表示应用已运行中。

NAME	READY	STATUS	RESTARTS	AGE
etcd-0	0/1	ImagePullBackOff	0	59m
icagent-m9dkt	0/0	Running	0	3d
nginx-2601814895-sf71t	1/1	Running	0	8s

步骤 4 : 创建服务。

```
kubectl create -f nginx-eip-svc.yaml
```

回显如下表示服务已创建成功。

```
service "nginx-eip" created
```

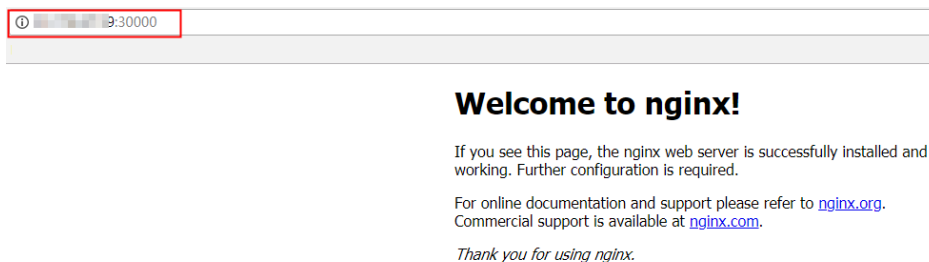
```
kubectl get svc
```

回显如下表示服务访问方式已设置成功。

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
etcd-svc	ClusterIP	None	<none>	3120/TCP	59m
kubernetes	ClusterIP	10.247.0.1	<none>	443/TCP	3d
nginx-eip	NodePort	10.247.120.135	<none>	80:30000/TCP	7s

步骤 5 : 在浏览器中输入访问地址, 例如为 10.78.44.60:30000 访问地址。

其中 10.78.44.60 为弹性 IP 地址，30000 为上一步中获取的节点端口号。

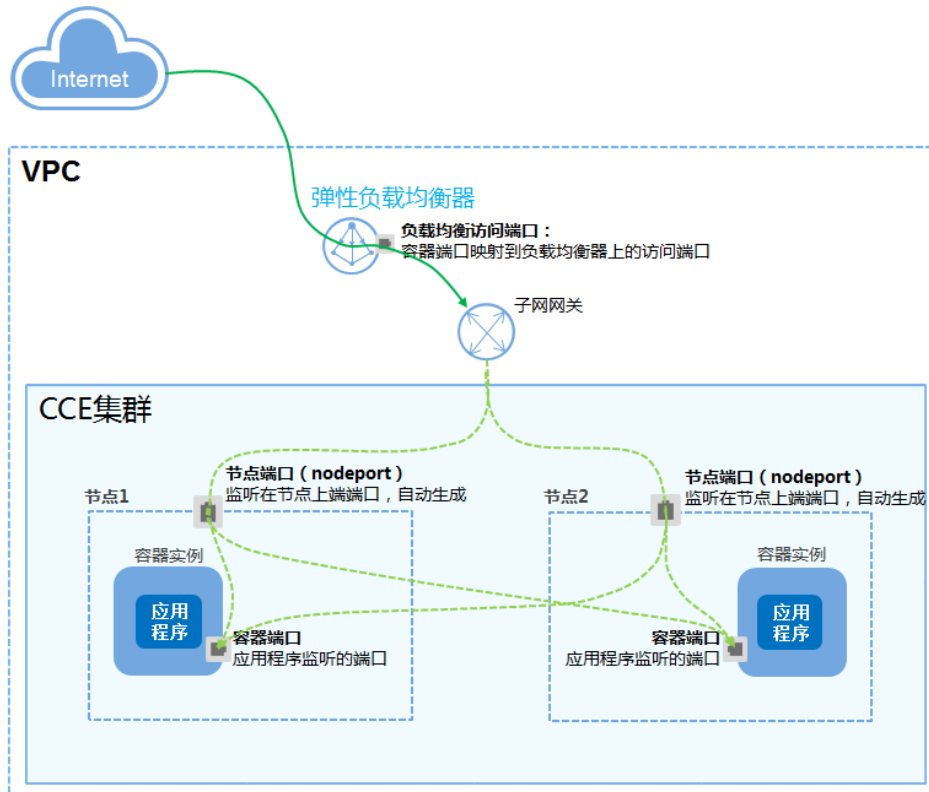


3.6.5 公网访问-四层负载均衡

公网访问-四层负载均衡可以通过弹性负载均衡从公网访问到应用，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。

四层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”。



添加方式

您可以在创建应用时设置访问方式，也可以应用创建完成后添加访问方式。

- 方式一：创建应用时配置。
- 方式二：应用创建完成后设置。

通过界面创建

本节以 nginx 为例进行说明。

步骤 1：创建无状态应用或创建有状态应用，在“应用访问设置”步骤，单击“添加服务”。

- 服务名称：自定义服务名称，可与应用名称保持一致。
- 访问方式：公网访问。
- 访问类型：负载均衡，支持使用已有负载均衡实例和自动创建两种方式。
- 协议：请根据业务的协议类型选择。
- 容器端口：容器镜像中应用实际监听端口，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应用时使用 ,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“查看应用详情”,在“访问方式”页签下,获取方式地址,例如 10.4.10.230:2。

步骤 4 :单击访问地址,即可跳转到访问页面。

应用创建完成后设置

步骤 1 :登录 CCE 控制台,选择左侧导航栏的“资源管理 > 网络管理”,在 **Service** 页签下,单击“添加 Service”。选择类型为“公网访问”。

步骤 2 :设置集群内访问参数。

- 服务名称 :自定义服务名称,可与应用名称保持一致。
- 集群名称 :服务所在集群。
- 命名空间 :服务所在命名空间。
- 关联应用 :选择需要添加 Service 的应用。
- 访问类型 :负载均衡,支持使用已有负载均衡实例和自动创建两种方式。
- 端口配置 :
 - 协议 :请根据业务的协议类型选择。
 - 容器端口 :容器镜像中应用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应用时使用,端口范围为 1-65535,可任意指定。

步骤 3 :单击“创建”,为应用添加“公网访问-负载均衡”的服务。

kubectl 命令行创建

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

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: 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
```

vi nginx-elb-svc.yaml

- 负载均衡实例已创建，请使用如下示例：

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    service.protal.kubernetes.io/type: LoadBalancer
    kubernetes.io/elb.class: union
    kubernetes.io/elb.id: a172d66c-e42f-4276-aa23-9258113478f6
  labels:
    app: nginx
  name: nginx
spec:
  loadBalancerIP: 10.78.42.242 # 配置为 ELB 的 IP 地址。
  ports:
    - name: service0
      nodePort: 31540 # 对应界面上的访问端口，不填写表示自动生成
      port: 80 # 集群虚拟 IP 的访问端口，也是注册到负载均衡上的端口
      protocol: TCP
      targetPort: 80 # 对应界面上的容器端口
  selector:
```

```
app: nginx
type: LoadBalancer
```

- 负载均衡实例未创建，在创建 service 时自动创建，请使用如下示例：

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
  annotations:
    service.protal.kubernetes.io/type: LoadBalancer
    kubernetes.io/elb.class: union
    kubernetes.io/elb.subnet-id: eff413e6-e6df-43e6-b586-ee36351d676d
    kubernetes.io/elb.autocreate:
'{"type":"public","bandwidth name":"cce-bandwidth-1548668872423","bandwidth chagemode":"traffic",
"bandwidth size":10,"bandwidth sharetype":"PER","eip type":"5 bgp"}'
spec:
  selector:
    app: nginx
  ports:
  - name: cce-service-0
    targetPort: 80
    nodePort: 0
    port: 80
    protocol: TCP
  type: LoadBalancer
  loadBalancerIP: ''
```

步骤 3：创建应用。

```
kubectl create -f nginx-deployment.yaml
```

回显如下，表示应用已开始创建。

```
deployment "nginx" created
```

```
kubectl get po
```

回显如下，应用状态为 Running 状态，表示应用已运行中。

NAME	READY	STATUS	RESTARTS	AGE
etcd-0	0/1	ImagePullBackOff	0	1h
icagent-m9dkt	0/0	Running	0	3d
nginx-2601814895-clxhw	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，表示应用访问方式已设置成功，应用可访问。

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	10.4.10.230	80:31540/TCP	51s

步骤 5：在浏览器中输入访问地址，即可成功访问 nginx。

例如，输入访问地址为 10.4.10.230:31540，其中 10.4.10.230 为负载均衡实例 IP 地址，31540 为对应界面上的访问端口。

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.

3.6.6 七层负载均衡-（Ingress）

七层负载均衡是采用了增强型弹性负载均衡，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。

添加方式

您可以在创建应用时设置访问方式，也可以应用创建完成后添加访问方式。

- 方式一：创建应用时配置。
- 方式二：应用创建完成后配置，此配置对工作应用无影响，且实时生效。

通过界面创建

本节以 ingress-test 应用为例进行说明。

步骤 1：创建应用；

- 若创建应用时，配置了应用访问方式，且设置为“VPC 内网访问”，请直接执行步骤 3。
- 若创建应用未设置访问方式，请先执行步骤 2。

步骤 2：（可选）若创建应用时，未配置“VPC 内网访问”，请执行如下操作。

1. 单击 CCE 左侧导航栏的“资源管理 > 网络管理”。
2. 在 Service 页签下，单击“添加 Service”。选择类型为“VPC 内网访问”。
 - 服务名称：自定义服务名称，可与应用名称保持一致。
 - 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。
 - 关联应用：单击“选择应用”，选择需要配置 VPC 内网访问的应用名称，单击“确定”。
 - 访问类型：选择节点 IP。
 - 端口配置：
 - 协议：请根据业务的协议类型选择。
 - 容器端口：容器镜像中应用实际监听的端口，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 访问应用时使用，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”，为应用添加 VPC 内网访问方式。

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。

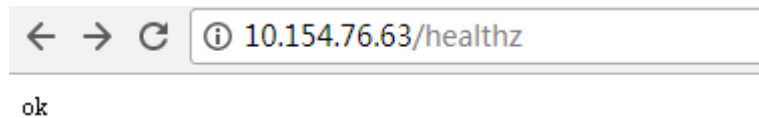
- 命名空间：选择需要添加 Ingress 的命名空间。
- 增强型负载均衡实例：支持使用已有负载均衡实例和自动创建两种方式。
- 对外端口：开放在负载均衡服务地址的端口，可任意指定。
- 对外协议：支持 HTTP 和 HTTPS。若选择 HTTPS，请选择密钥证书。密钥证书需提前创建，密钥类型为 IngressTLS。
- 域名：实际访问的域名地址，对应负载均衡服务域名地址，需用户购买备案自己的域名，可选填。一旦配置了域名规则，则必须使用域名访问。否则，可以使用负载均衡实例的 IP 地址访问。
- 路由配置：
 - 路由匹配规则：前缀匹配、精确匹配、正则匹配。
 - 前缀匹配：例如映射 URL 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。
 - 映射 URL：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，该服务访问类型为 VPC 内网服务。若服务不存在，请单击“点此创建服务”去创建一个服务。
 - 容器端口：容器镜像中容器实际监听端口，需用户确定。例如：defaultbackend 程序实际监听的端口为 8080。

步骤 4：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：访问应用（例如名称为 defaultbackend）的“/healthz”接口。

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：10.154.76.63:80/healthz。
2. 在浏览器中输入“/healthz”接口的访问地址，即可成功访问应用。



kubectl 命令行创建

本节以 nginx 为例，说明 kubectl 命令实现 ingress 访问的方法。

前提条件

配置 kubectl 命令，使弹性云主机连接集群。

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。



选择 HTTPS 协议时，才需要创建密钥证书 ingress-test-secret.yaml。

vi ingress-test-deployment.yaml

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: ingress-test
spec:
  replicas: 1
  selector:
    matchLabels:
      app: ingress-test
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: ingress-test
    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
```

vi ingress-test-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: ingress-test
    name: ingress-test
spec:
  ports:
    - name: service0
      port: 8080          #集群虚拟 IP 的访问端口
      protocol: TCP
      targetPort: 8080   #对应界面上的容器端口, 应用程序实际监听的端口
    #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
  type: NodePort        #采用 Nodeport 的访问类型连接负载均衡
```

vi ingress-test-ingress.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/elb.ip: 192.168.0.39      #必填, 为负载均衡增强型实例的服务地址, 公网 ELB 配置为公网 IP, 私
    网 ELB 配置为私网 IP
    kubernetes.io/elb.port: "80"          #必填, 界面上的对外端口, 为注册到负载均衡服务地址上的端口
  name: ingress-test
spec:
  tls:
    #可选, HTTPS 协议时, 需添加此参数
    - secretName: test-secret             #可选, HTTPS 协议时添加, 配置为创建的密钥证书名称
  rules:
    - http:
        paths:
          - backend:
              serviceName: ingress-test    #为 ingress-test-svc.yaml 的服务名称
              servicePort: 8080           #为 ingress-test-svc.yaml 的 targetPort, 即容器端口
            property:
              ingress.beta.kubernetes.io/url-match-mode: EQUAL_TO    #路由匹配策略, 可选值为 EQUAL_TO (精确匹
              配)、STARTS_WITH (前缀匹配)、REGEX (正则匹配)
              path: "/healthz"           #为路由, 用户自定义设置
```

vi ingress-test-secret.yaml

```
apiVersion: v1
data:
  tls.crt: LS0tLS1CRUdJ*****0tCg==
  tls.key: LS0tLS1*****ZLS0tLS0K
kind: Secret
```

```

metadata:
  annotations:
    description: test for ingressTLS secrets
  name: test-secret
  namespace: default
type: IngressTLS
    
```

步骤 3 : 创建应用。

```
kubectl create -f ingress-test-deployment.yaml
```

回显如下，表明应用已创建。

```
deployment "nginx" created
```

```
kubectl get po
```

回显如下，表明应用创建成功。

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" 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" created
```

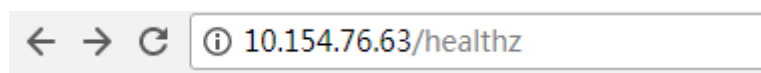
```
kubectl get ingress
```

回显如下，表示 ingress 服务创建成功，应用可访问。

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。



ok

3.7 应用管理

3.7.1 弹性伸缩

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。当前支持三种弹性伸缩策略：

- A、告警策略：支持根据 CPU/内存的设置，进行应用的自动伸缩；
- B、定时策略：支持在特定时间点进行应用的自动伸缩；
- C、周期策略：支持以天、周、月为周期的伸缩策略；

“告警策略”的使用步骤：

登录云容器引擎控制台，在左侧导航栏中选择【应用管理】，单击待设置伸缩策略的应用，进入应用详情页面，选择【伸缩】页签；

在弹性伸缩下，单击【添加伸缩策略】：

参数	参数说明
策略名称	请输入伸缩策略的名称。
策略类型	选择“告警策略”。
触发条件	支持“CPU使用率”和“内存使用率”。若输入“内存使用率”的“平均值>70%”，表示在该条件下触发伸缩策略。
周期时长	指标统计周期。单击下拉选项进行选择。若设置为20秒，表示每20秒统计一次。
连续出现次数	若设置为3，则表示若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。
执行动作	策略触发后执行的动作。增加或减少实例数。

单击【确定】

“定时策略”的使用步骤：

登录云容器引擎控制台，在左侧导航栏中选择【应用管理】，单击待设置伸缩策略的应用，进入应用详情页面，选择【伸缩】页签；

在弹性伸缩下，单击【添加伸缩策略】：

参数	参数说明
策略名称	请输入伸缩策略的名称。
策略类型	选择定时策略。
触发时间	策略触发时间。
执行动作	策略触发后执行的动作。增加或减少实例数。

单击【确定】

“周期策略”的使用步骤：

登录云容器引擎控制台，在左侧导航栏中选择【应用管理】，单击待设置伸缩策略的应用，进入应用详情页面，选择【伸缩】页签；

在弹性伸缩下，单击【添加伸缩策略】：


参数	参数说明
----	------

参数	参数说明
策略名称	请输入伸缩策略的名称。
策略类型	选择周期策略。
选择时间	选择策略触发的时间。
执行动作	策略触发后执行的动作。

单击【确定】

3.7.2 手动伸缩

登录云容器引擎控制台，在左侧导航栏中选择【应用管理】，单击待设置伸缩策略的应用，进入应用详情页面，选择【伸缩】页签；

在【手动伸缩】策略下，单击  修改实例数量，单击【保存】：

3.7.3 优雅缩容策略

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。若未设置，则默认设置为 30 秒。

登录云容器引擎控制台，在左侧导航栏中选择【应用管理】，单击待设置伸缩策略的应用，进入应用详情页面，选择【伸缩】页签；


在【优雅缩容策略】下，单击  修改时间窗，单击【保存】：

3.7.4 升级应用

登录云容器引擎控制台，在左侧导航栏中选择【应用管理】，单击待设置伸缩策略的应用，进入应用详情页面，选择【更新升级】页签；

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
升级策略	展示当前应用选择的升级方式。
容器配置	

参数	参数说明
更换镜像	在对应的容器下，单击【更换镜像】进行修改。
容器名称	单击  ，修改容器名称。
高级设置	
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	不支持修改。


更新完成后，单击【提交】；

3.7.5 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。

登录云容器引擎控制台，单击左侧导航栏的【应用管理】；

单击已创建应用的名，进入应用详情页面；

单击【实例列表】，单击某个实例名称前的 ，单击【监控】；

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。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。

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。实例处于非运行状态时，不能看到内存使用量；

3.7.6 启停应用

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

登录云容器引擎控制台，单击左侧导航栏的【应用管理】；

单击待停止应用后的【更多】【停止】，停止运行中的应用；

单击待启动的应用后的【启动】，“启动”已停止的应用；

3.7.7 删除应用

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

登录云容器引擎控制台，单击左侧导航栏的【应用管理】；

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；

单击【确定】；

3.8 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：

开源镜像：展示了所有用户上传到平台的公开镜像，您也可以基于开源镜像创建应用。

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：



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。

3.8.1 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。

组织管理：提供分组管理使用仓库的用户权限。

3.8.2 创建组织

同个仓库内可创建多个组织，形成逻辑上的不同分组，用来实现对用户的权限分组。本节指导用户创建组织，并且为组织设置用户权限。

登录云容器引擎控制台，单击左侧导航栏的【镜像仓库】->【组织管理】，单击【创建组织】；

输入组织名称，如 test，单击【确定】：

在【我的镜像】页面右侧【全部组织】的下拉框中，选择对应的组织，可出现该组织下创建的镜像；

3.8.3 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：

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；

请确保镜像的正确性，能够成功后台启动；

操作步骤：

创建组织：

A、登录云容器引擎控制台，单击左侧导航栏的【镜像仓库】->【组织管理】，单击【创建组织】；

B、输入命名空间名称，如 test，单击【确定】；

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

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-1.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-1.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-1.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 3.7.3 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
```

终端显示如下信息，表明 `push` 镜像成功。

```
6d6b9812c8ae: Pushed
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。

3.8.4 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：

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。

请确保镜像的正确性，能够成功后台启动；

操作步骤：


创建镜像命名空间：

- A、登录云容器引擎控制台，单击左侧导航栏的【镜像仓库】->【组织管理】，单击【创建组织】；
- B、输入命名空间名称，如 test，单击【确定】；
- C、单击【Pull/Push 指南】页签，可查看 Pull/Push 镜像的方法；

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 指令】，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:

```
6d6b9812c8ae: Pushed
695da0025de6: Pushed
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
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。

3.8.5 删除镜像仓库中的镜像

登录云容器引擎控制台, 单击左侧导航栏的【镜像仓库】, 单击【我的镜像】;

单击【自有镜像】页签, 进入镜像列表详情;

在【镜像名称】下, 勾选待删除镜像后的选项框。您还可以选择一个或多个镜像, 单击【删除】, 进行批量删除镜像操作;

3.8.6 查看镜像仓库地址

登录云容器引擎控制台, 单击左侧导航栏的【镜像仓库】, 单击【我的镜像】;

单击待删除的镜像仓库名称, 进入仓库镜像详情;

在【镜像版本】下查看镜像仓库地址：

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 最佳实践

4.1 总体设计

4.1.1 硬件配置

租户可以根据自己的业务需求选择相应的弹性云主机规格，本文以“s3.xlarge.2”规格为例介绍如何部署云容器节点。

“s3.xlarge.2”具有 4CPU，8GB 内存，40G SATA 作为系统盘，100G SATA 作为数据盘。

4.1.2 软件环境

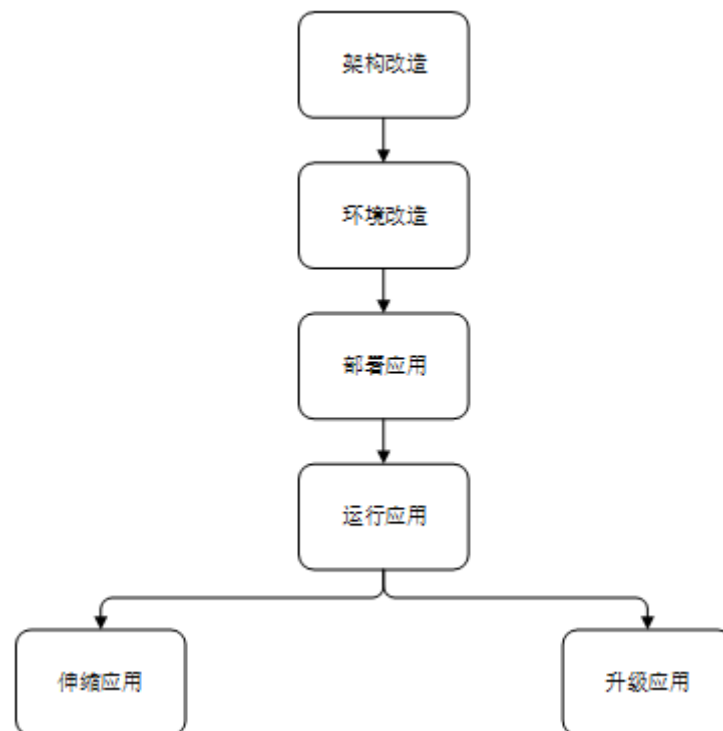
部件	版本	使用目的
Mysql-ndb	7.5.5	负责存储数据
Mysql-mysqld	7.5.5	用于从数据应用层访问 MySQL 应用数据
Mysql-mgmd	7.5.5	MySQL 的管理工作负载, 主要作用是管理 mysql 的其它组件
Docker 版本	Docker version 17.06	用于镜像制作

4.1.3 资源规划

虚拟私有云地址	集群管理规模	网络模型	云主机操作系统	云主机 EIP 地址	节点配置	节点数量	节点 EIP 地址
192.168.10.0/24	50 节点	VPC 网络	Centos7.5	36.112.134.172	8C12GB	1	27.128.232.97/
内部服务通讯		提供集群高性能	创建 docker 镜像	公网业务访问	业务提供	提供业务	公网业务访问

4.1.4 部署流程

本实践举例以某游戏 demo 为例,讲解如何部署游戏应用到 CCE 云容器引擎服务中,并演示了上线、升级扩容等全流程操作,配置概要如下:



4.1.5 游戏架构改造

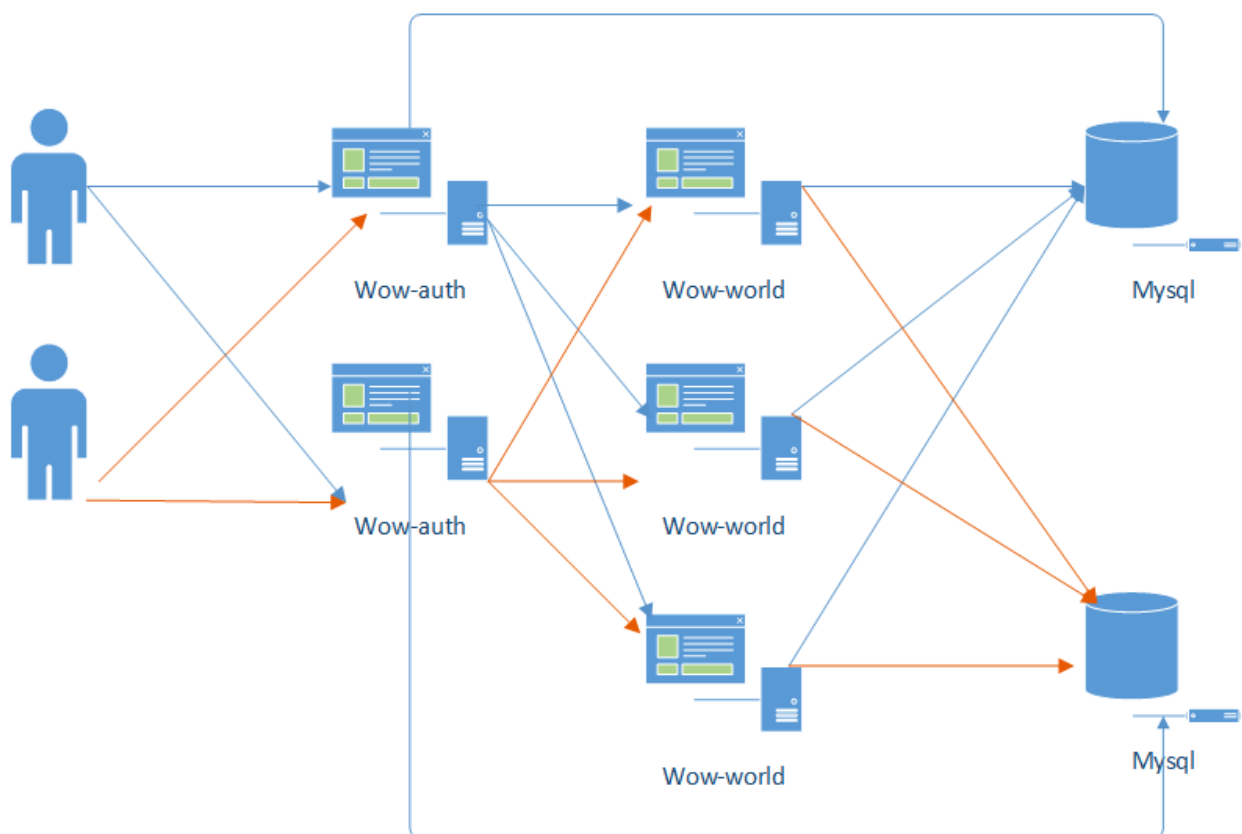
原始架构：

游戏包含三个组件：wow-auth 登录认证系统、wow-world 游戏服务器、mysql 数据库。

一台虚拟机安装了 wow-auth 登录认证系统及其依赖，两台或多台虚拟机安装 mysql 数据库及其依赖；三台或多台虚拟机安装 wow-world 游戏服务器及其依赖。在当前的架构下，如果有多个游戏服，需要多台虚拟机来安装。这就引发了这个架构带来的关键问题：部署重，需要准备多台虚拟机，并为每个虚拟机装上不同组件所需的依赖包。

扩展差，伸缩困难，维护成本高：新增服务器时，需增加新的虚拟机，再安装游戏服。且多台虚拟机维护十分复杂。

升级困难：每次升级需要多台虚拟机逐台配置升级，耗时久且容易出错。



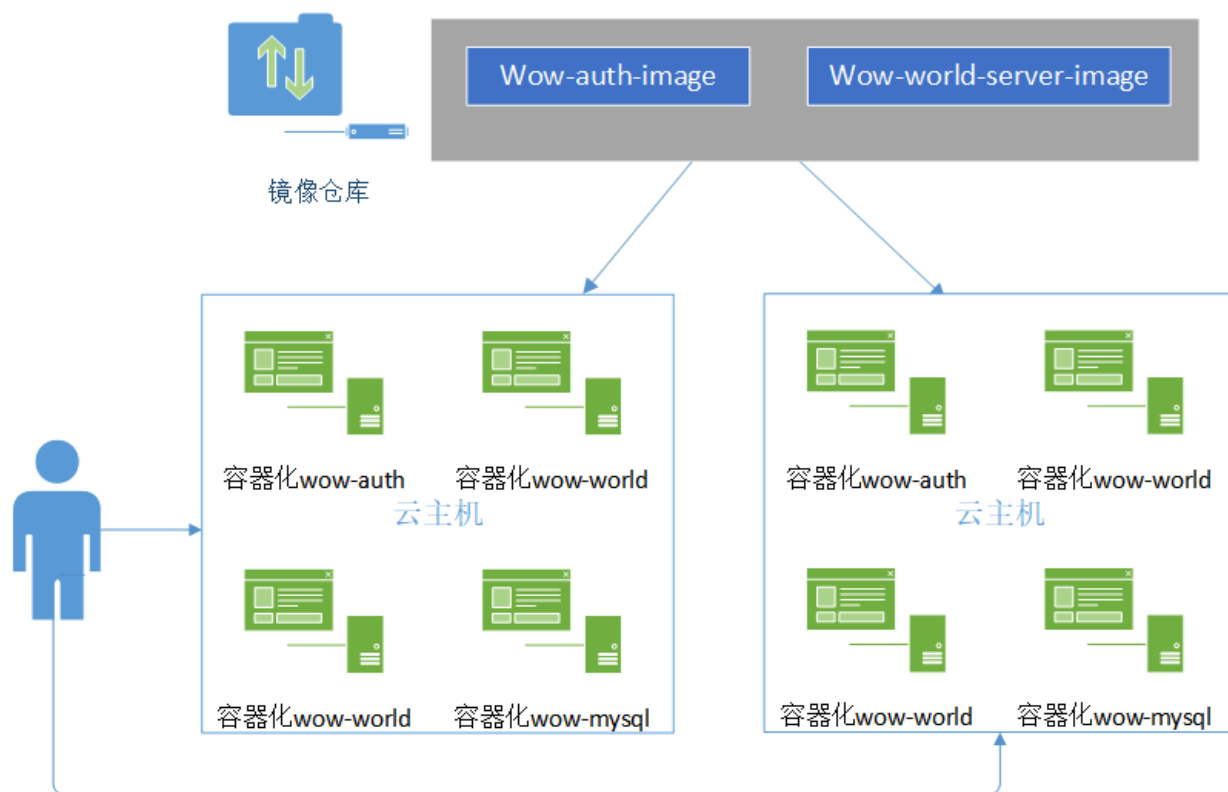
云上架构

将游戏的三个组件（wow-auth 登录认证系统、wow-world 游戏服务器、mysql 数据库）分别容器化，部署在虚拟机中。

部署轻便：将游戏的三个组件分别制作作为容器镜像，并将镜像上传到容器镜像仓库中。通过 CCE 云容器引擎可以直接基于该镜像部署容器应用。

扩展性好，伸缩迅速：当需要新增游戏服时，再启动一个容器即可。容器可实现秒级启动。

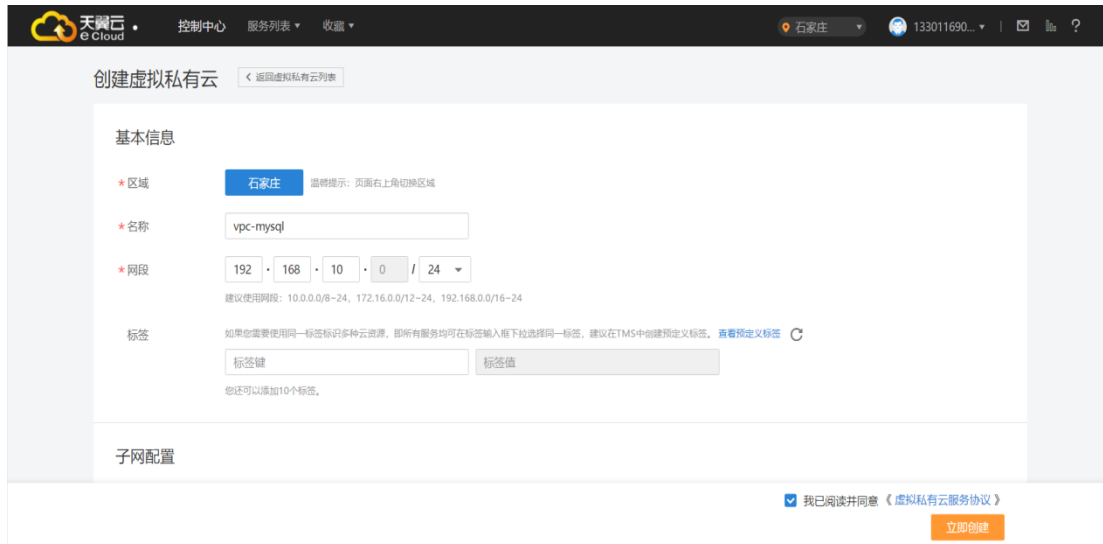
升级方便：升级时更换镜像即可快速升级。且 CCE 支持滚动升级，在升级时无需中断业务。



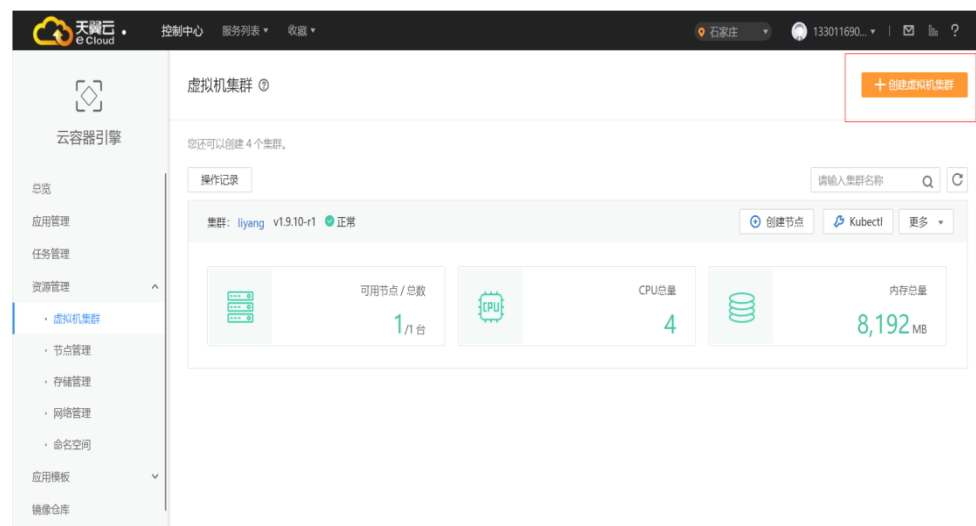
4.2 环境准备

创建 VPC

设置名称为 vpc-mysql，网段为 192.168.10.0/24



在云容器引擎中创建虚拟机集群



选择 VPC 虚拟私有云

* 集群名称
以小写字母开头，由小写字母、数字、中划线(-)组成，不能超过24位，且不能以中划线(-)结尾。

* 版本 **v1.9.10** Kubernetes社区基线版本，建议选择最新版本。

* 集群管理规模 50节点 200节点 1,000节点

* 高可用 是 否 集群创建完成后，高可用模式及普通模式之间不可变更，请按实际使用场景选择。

* 虚拟私有云 创建虚拟私有云，完成后点击刷新按钮。

* 所在子网 创建子网，完成后点击刷新按钮。

* 网络模型 容器隧道网络 VPC网络

集群描述

创建节点

选择节点规格为 4 核 8GB

节点名称
创建多个节点时，节点名称自动增加5位随机字符后缀。例如：输入cluster-node，将补充随机5位随机字符，也就是cluster-node-xxxxx，xxxxx是增加的5位随机字符。

节点规格 vCPUs 内存

通用型

规格	vCPUs 内存
<input type="radio"/> s3.large.2	2 核 4 GB
<input type="radio"/> s3.large.4	2 核 8 GB
<input checked="" type="radio"/> s3.xlarge.2	4 核 8 GB
<input type="radio"/> s3.xlarge.4	4 核 16 GB
<input type="radio"/> s3.2xlarge.2	8 核 16 GB
<input type="radio"/> s3.2xlarge.4	8 核 32 GB
<input type="radio"/> s3.4xlarge.2	16 核 32 GB
<input type="radio"/> s3.4xlarge.4	16 核 64 GB

当前规格 **通用型 | s3.xlarge.2 | 4 vCPUs | 8 GB**

操作系统 EulerOS 2.2

创建一台云主机，并安装 docker（准备镜像）

在系统上添加 yum 源：`/etc/yum.repos.d/docker-ce.repo`

```
# wget https://download.docker.com/linux/centos/docker-ce.repo
# mv docker-ce.repo /etc/yum.repos.d
# yum install -y docker-ce
```

或在 Linux 操作系统下，可以使用如下命令快速安装最新版本的 Docker。

```
curl -fsSL get.docker.com -o get-docker.sh
```

```
sh get-docker.sh
```

```
[root@ecs ~]# curl -fsSL get.docker.com -o get-docker.sh
[root@ecs ~]# vim get-docker.sh
[root@ecs ~]# sh get-docker.sh
```

在云主机中配置证书，以确保可以登录镜像仓库

```
vi /etc/sysconfig/docker
```

```
OPTIONS='--live-restore --log-opt max-size=50m --log-opt max-file=20 --log-driver=json-file --registry-mirror=
https://100.125.0.36:20202 --insecure-registry registry.cn-bj1.ctyun.cn'
```

或

Vim /usr/lib/systemd/system/docker.service 中添加

```
Execstart=/usr/bin/dockerd--registry-mirror=https://100.125.0.36:20202--insecure-registry=
registry.cn-hesj1.ctyun.cn
```

云主机登录镜像仓库

把生成的 docker login 指令复制到云主机中



```
[root@ecs ~]# docker login -u cn-hesjz1@f9XIphwodcHY7fleka8M -p 07a78a5e21edb484
6c172abda22c6d400033406c91d92294b6787b5d859b47c6 registry.cn-hesjz1.ctyun.cn
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

拉取镜像到本地云主机

```
docker pull swr.cn-north-1.myhuaweicloud.com/wow/wow:wowauth-5.0

docker pull swr.cn-north-1.myhuaweicloud.com/wow/wow:wowauth-5.1

docker pull swr.cn-north-1.myhuaweicloud.com/wow/wow:wowworld-5.0-withmap

docker pull swr.cn-north-1.myhuaweicloud.com/wow/wow:wowworld-5.1-withmap
```

```
[root@ecs ~]# docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
mysql/mysql-cluster latest       67fd04af4b8e     12 days ago     338MB
mysql               5.7         f6509bac4980     2 weeks ago     373MB
registry.cn-hesjz1.ctyun.cn/liyang/mysql 5.7         f6509bac4980     2 weeks ago     373MB
registry.cn-hesjz1.ctyun.cn/liyang/wow   wowworld-5.0-withmap 44342195d2d5     20 months ago   5.27GB
registry.cn-hesjz1.ctyun.cn/liyang/wow   wowworld-5.1-withmap 44342195d2d5     20 months ago   5.27GB
swr.cn-north-1.myhuaweicloud.com/wow/wow wowworld-5.0-withmap 44342195d2d5     20 months ago   5.27GB
swr.cn-north-1.myhuaweicloud.com/wow/wow wowworld-5.1-withmap 44342195d2d5     20 months ago   5.27GB
registry.cn-hesjz1.ctyun.cn/liyang/wow   wowauth-5.0       1a5af6744fba     20 months ago   166MB
registry.cn-hesjz1.ctyun.cn/liyang/wow   wowauth-5.1       1a5af6744fba     20 months ago   166MB
swr.cn-north-1.myhuaweicloud.com/wow/wow wowauth-5.0       1a5af6744fba     20 months ago   166MB
swr.cn-north-1.myhuaweicloud.com/wow/wow wowauth-5.1       1a5af6744fba     20 months ago   166MB
husseingalal/mysql-ndb                    7.5.5           4268bb9e8f51     2 years ago     3.22GB
[root@ecs ~]#
```

1. 上传镜像到河北石家庄节点的容器镜像仓库中

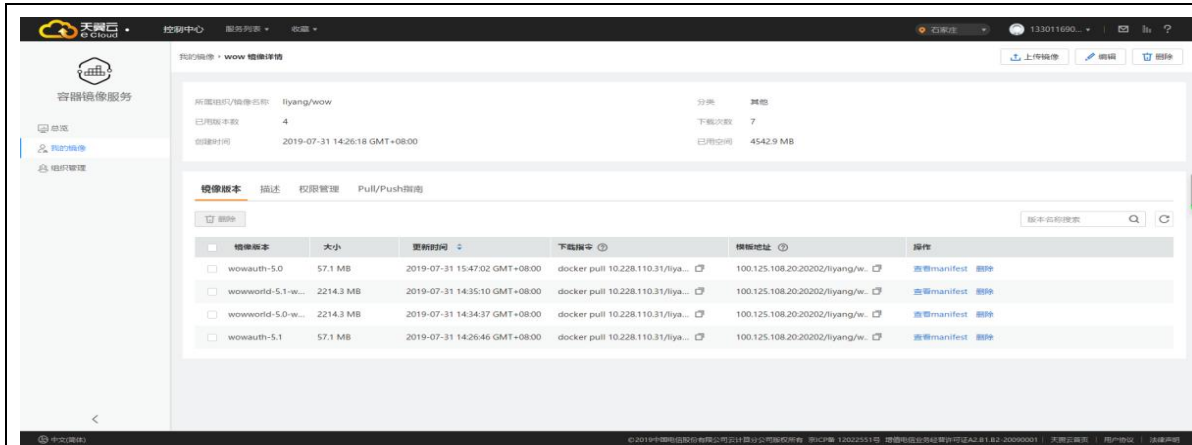
修改镜像名称为符合规定的名称

```
Docker tag swr.cn-north-1.myhuaweicloud.com/wow/wow:wowauth-5.0 registry.cn-
hesjz1.ctyun.cn/liyang/wow:wowauth-5.0
```

上传修改名称后的镜像到私有镜像仓库中

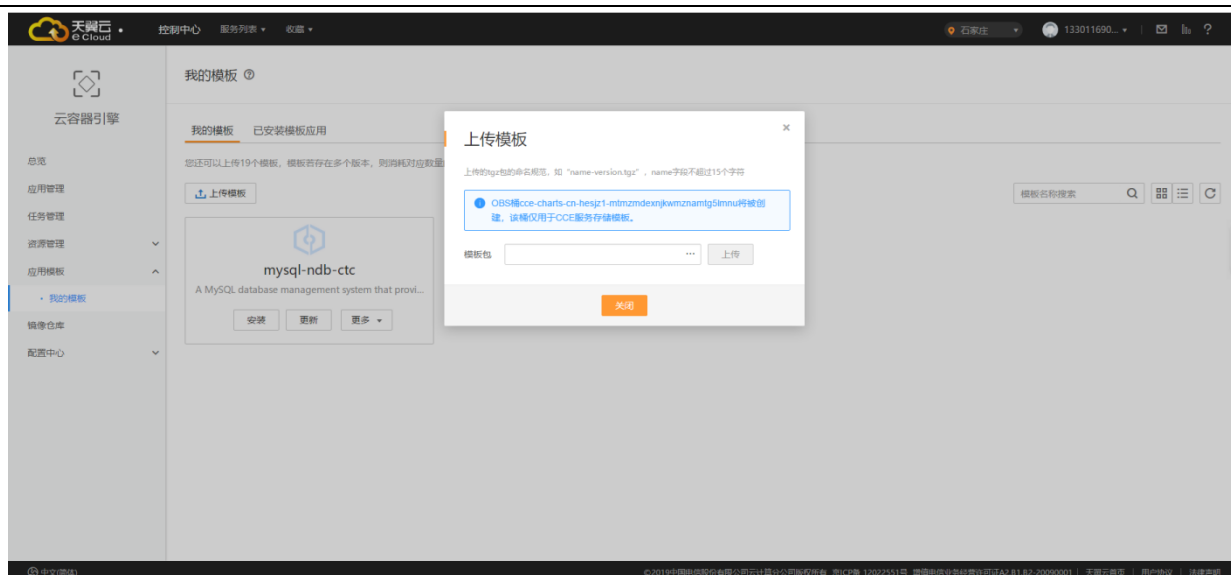
```
docker push registry.cn-bj1.ctyun.cn/liyang/wow:wowauth-5.0
```

请按相同的方法分别上传镜像：wowauth-5.1、wowworld-5.0-withmap、wowworld-5.1-withmap。



2. 上传本地应用模板

此模板为 mysql-ndb-ctc，是一个适用于分布式计算环境的高可用性、高冗余版本的 MySQL。



3. 安装上传好的模板

模板实例名称：自定义名称，例如为 mysql。

模板版本：1.0.0。

部署集群：选择创建集群中创建的集群。

部署工作负载规格：选择 1X。

描述：数据库描述信息，例如输入“数据库”。

数据库名称: clustertest。

数据库普通用户名称: trinity。此名称是在镜像中设定好的。

数据库普通用户登录密码: trinity。此密码是在镜像中设定好的。

管理员权限密码: 请保持默认 root。

单击“安装”，按在配置确认中，点击“确认”

单击“返回模板实例列表”，在“模板实例”中可查看到 mysql 应用已安装成功。

单击 wow-mysql 名称，可看到 mysql 的 3 个实例，待状态为“运行中”，表示 mysql 创建成功。

大约需要 5 分钟创建成功，请耐心等待。

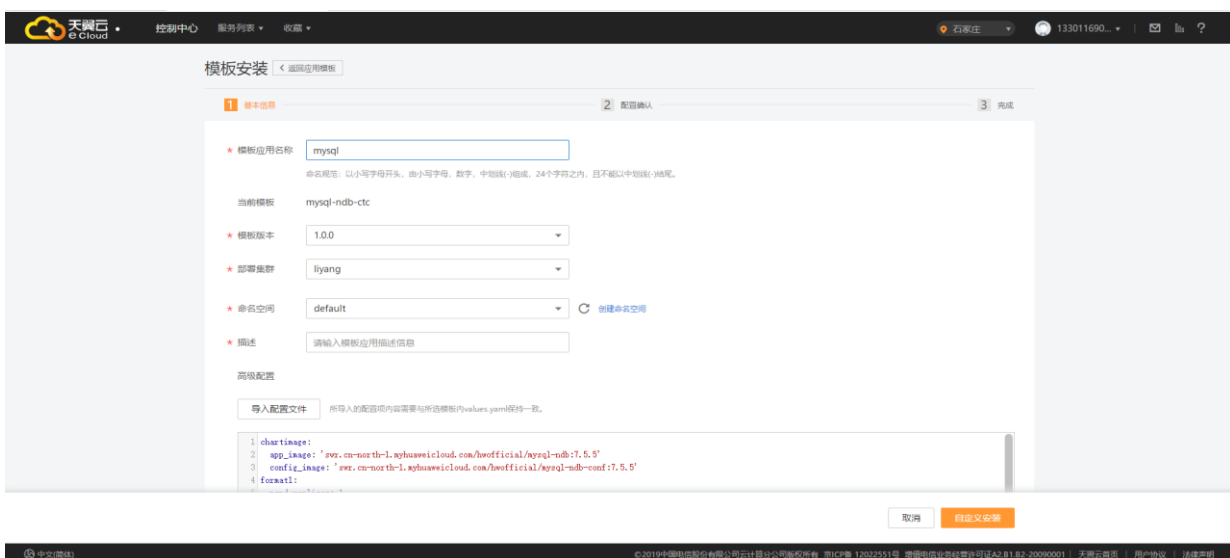
wow-mysql-mgmd: MySQL 的管理工作负载，主要作用是管理 mysql 的其它组件。

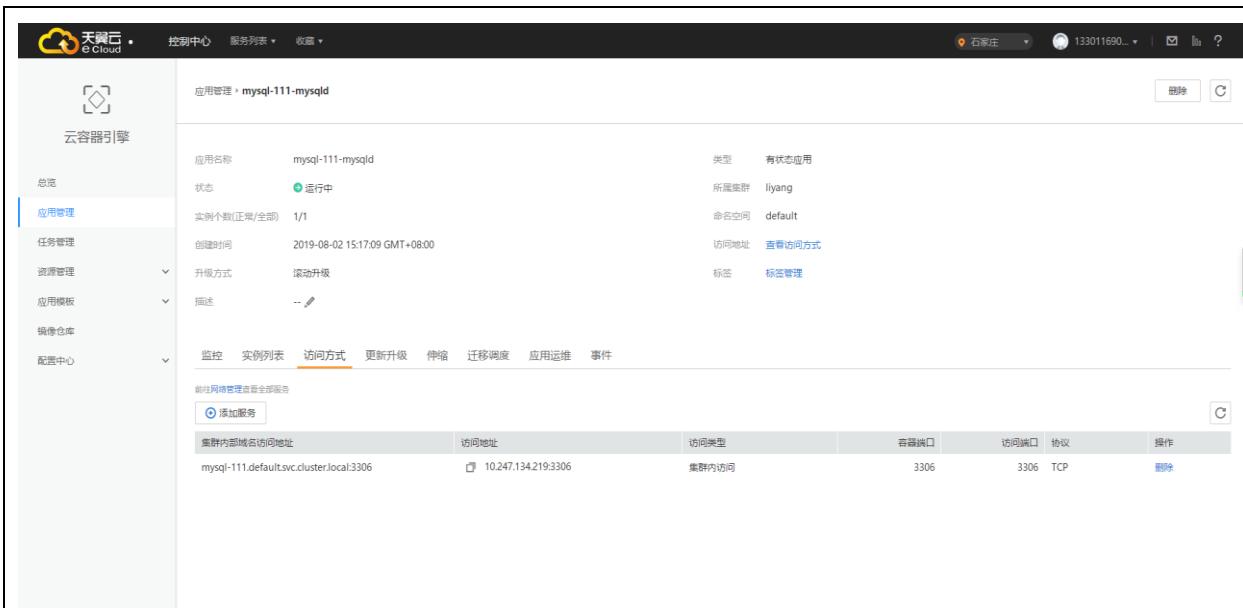
wow-mysql-ndbd: 负责存储数据。

wow-mysql-mysqld: 是使用 NDBCLUSTER 存储引擎的传统 MySQL 服务器。

该实例用于从数据应用层访问 MySQL 应用数据。

单击 wow-mysql-mysqld 名称，进入详情页面。单击“访问方式”页签，获取并记录“访问地址”。



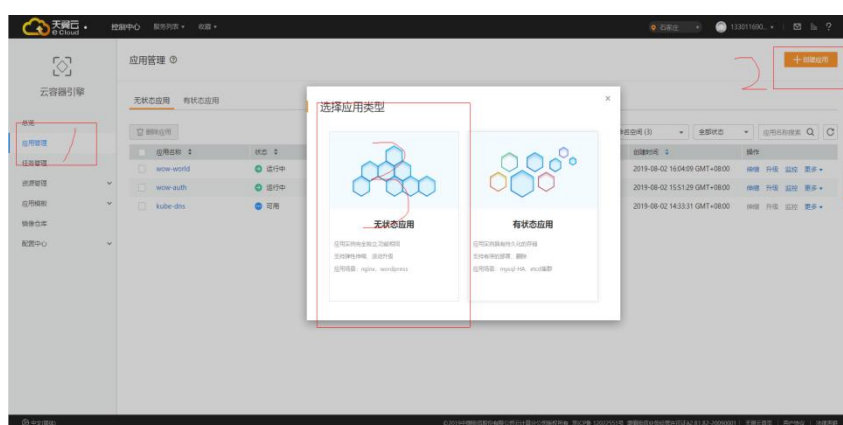


4. 点击应用管理，创建应用，选择无状态应用，部署 wow-auth 认证系统

无状态工作负载：在运行中始终不保存任何数据或状态的称为“无状态工作负载”，例如 nginx。

有状态工作负载：在应用运行过程中会保存数据或状态的称为“有状态工作负载”。

例如 Mysql，它需要存储产生的新数据。



5. 配置工作负载基本信息

工作负载名称：自定义名称，例如 wow-auth。

集群名称：选择创建集群中创建的集群。

命名空间：保持默认。

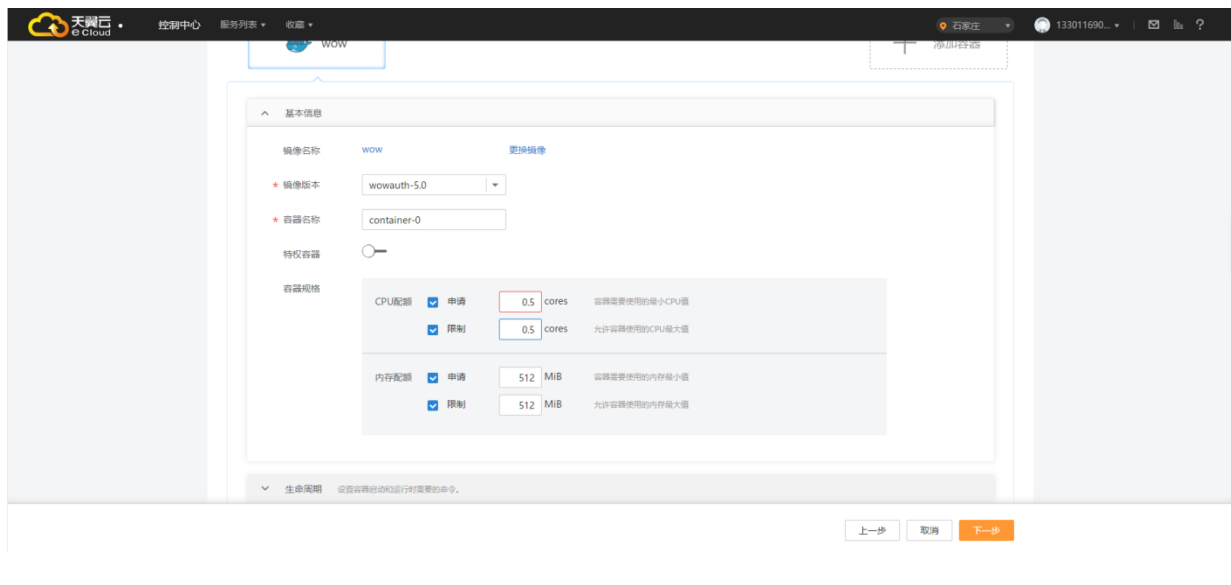
实例数量：建议配置为 1，否则资源会不够用。

工作负载描述：可不填写。

单击“下一步”，单击“添加容器”选择镜像，搜索 wow，选择 wowauth-5.0 版本的镜像。

其余参数保持默认，请单击容器规格下的“自定义配额”，设置该组件需要的规格。

设置完成后，选择 0.5X 的规格。



6. 设置环境变量

变量名称	变量说明	变量/变量引用（示例）
mysqlip	变量值请填写 6 中获取的数据库访问地址。	10.247.134.219
mysqlrootpasswd	数据库管理员权限密码，与 3 中的管理员权限密码保持一致。	root

biboaddress	游戏服务器外部访问地址，可以选择弹性 IP 方式，或者负载均衡。本例使用弹性 IP 方式。请选择 6 中的弹性 IP 地址。	27.128.232.97
biboport	游戏服务器外部访问端口，需要预先自定义定义设置一个值，范围为 30000-32767。 该值必须确保在当前集群中全局唯一。本例为了操作方便，指定固定值来连接游戏服务器。若部署真实业务，建议在制作镜像时明确依赖关系。	32500

7. 设置应用的外部访问方式。

访问类型：选择“公网访问”。

服务名称：可以和应用名称保持一致，例如设置为 wow-auth。

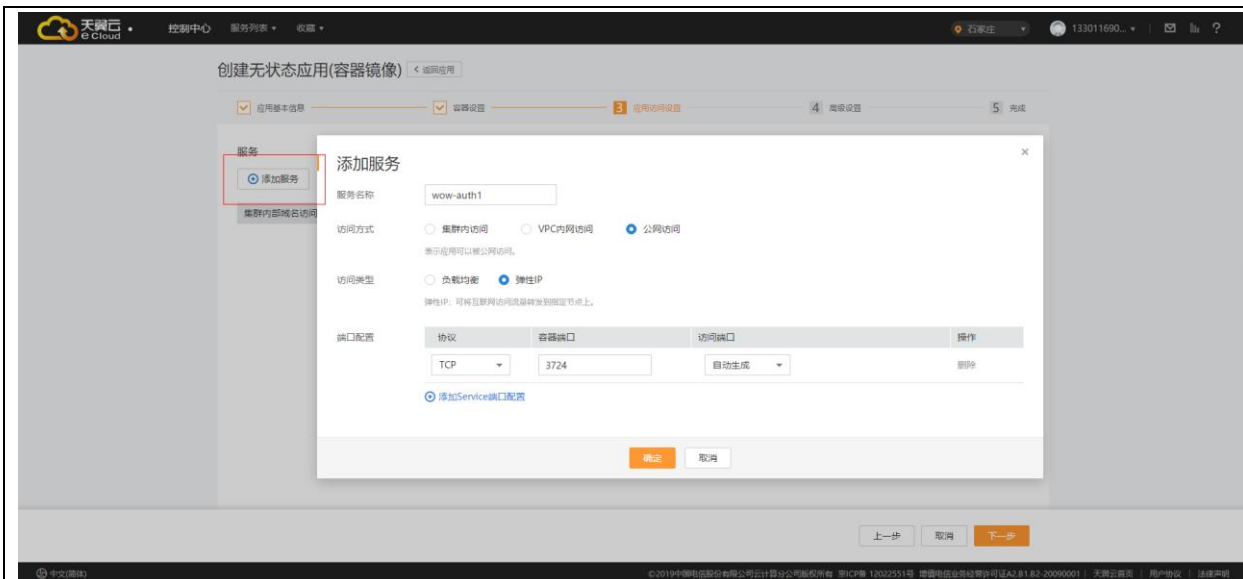
访问类型：本例选择“弹性 ip ”

端口配置：

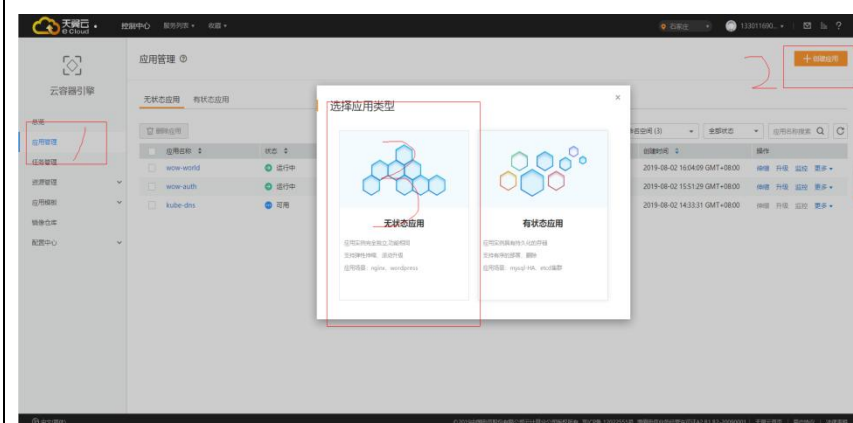
协议：设置为 TCP 协议。

容器端口：认证系统的监听端口，本例监听端口为 3724。镜像中已写 3724，此数据不可填写其它值。

访问端口：容器端口映射到节点私有 IP 上的端口，设置为自动生成。



8. 点击应用管理，创建应用，选择无状态应用，部署 wow-world 游戏服务器



9. 配置工作负载基本信息

工作负载名称：自定义名称，例如为 wow-world。

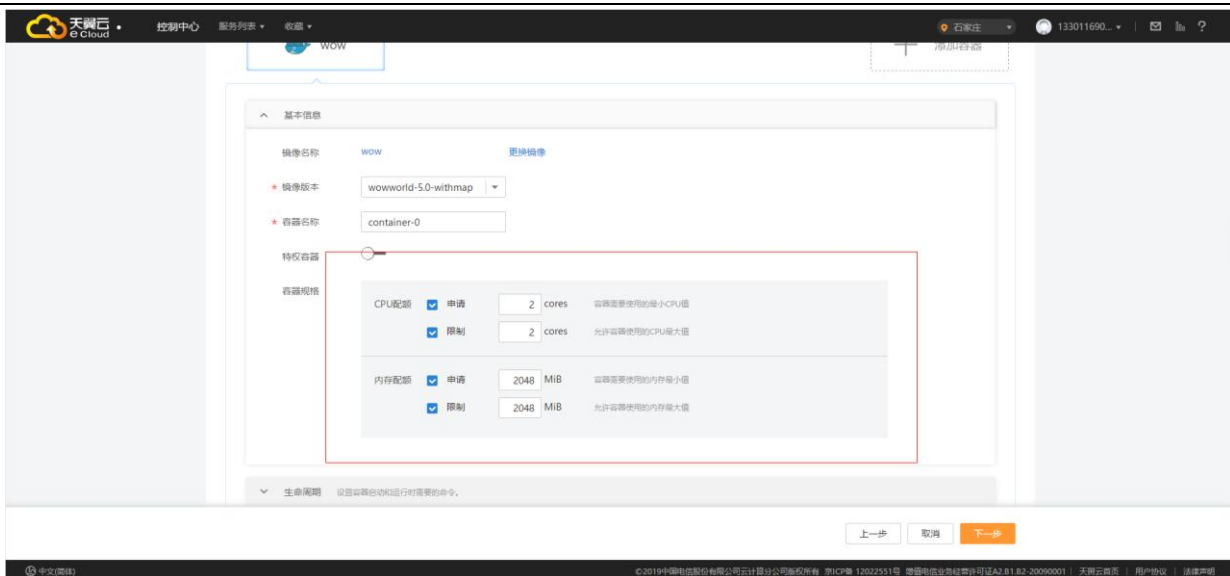
集群名称：选择创建集群中创建的集群。

命名空间：保持默认。

实例数量：建议配置为 1，否则资源会不够用。

工作负载描述：可不填写。

单击“下一步”，单击“添加容器”选择镜像，搜索 wow，选择 wowworld-5.0-withmap 版本的镜像。



10. 设置环境变量，用于连接数据库。

变量名称	变量说明	变量/变量引用（示例）
mysqlip	变量值请填写数据库访问地址。	10.247.134.219
mysqlrootpasswd	数据库管理员密码，与 3 中设置的管 理员权限密码一致。	root

11. 设置应用的外部访问方式。

访问类型：选择“公网访问”。

服务名称：可以和工作负载名称保持一致，例如设置为 wow-world。

访问类型：本例选择“弹性 ip”。

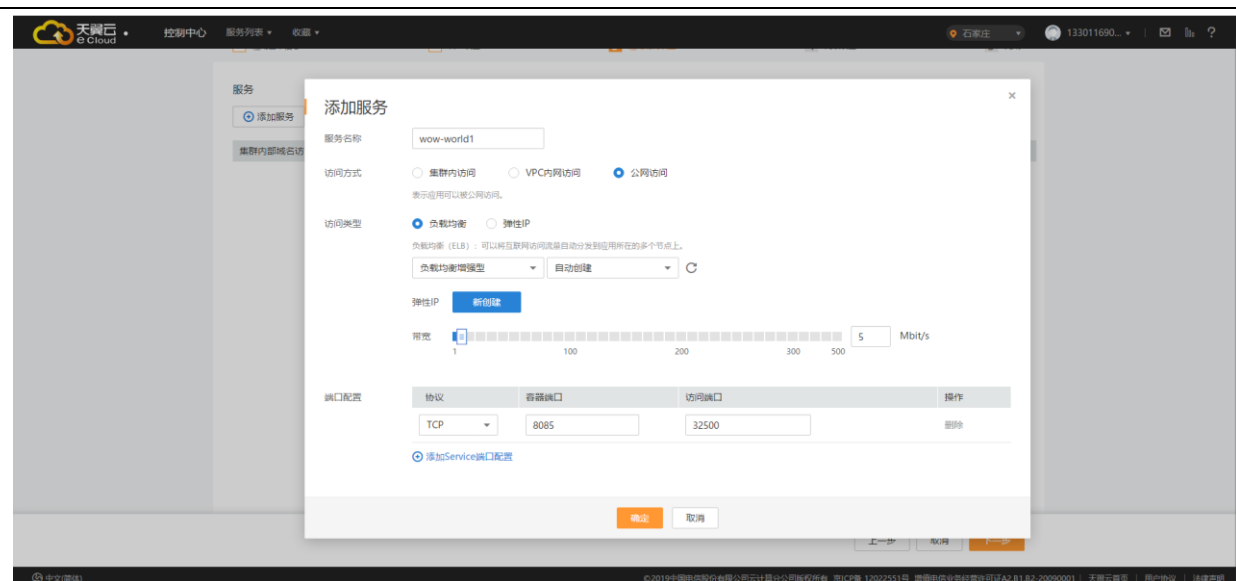
负载均衡（ELB）：可以将互联网访问流量自动分发到应用所在的多个节点上

弹性 IP：可将互联网访问流量转发到指定节点上。

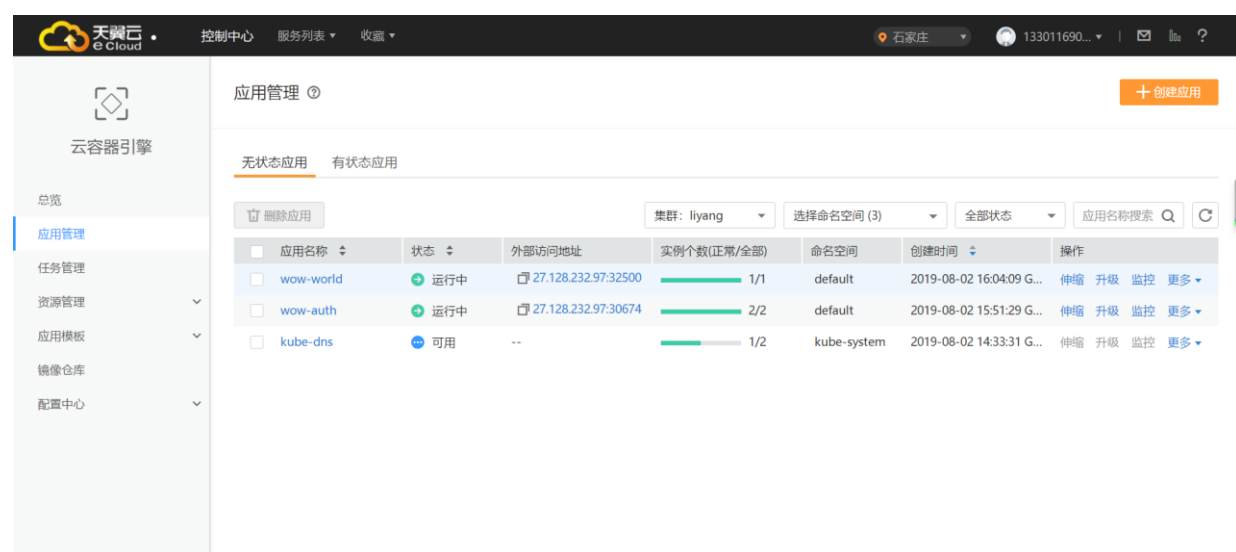
协议：设置为 TCP 协议。

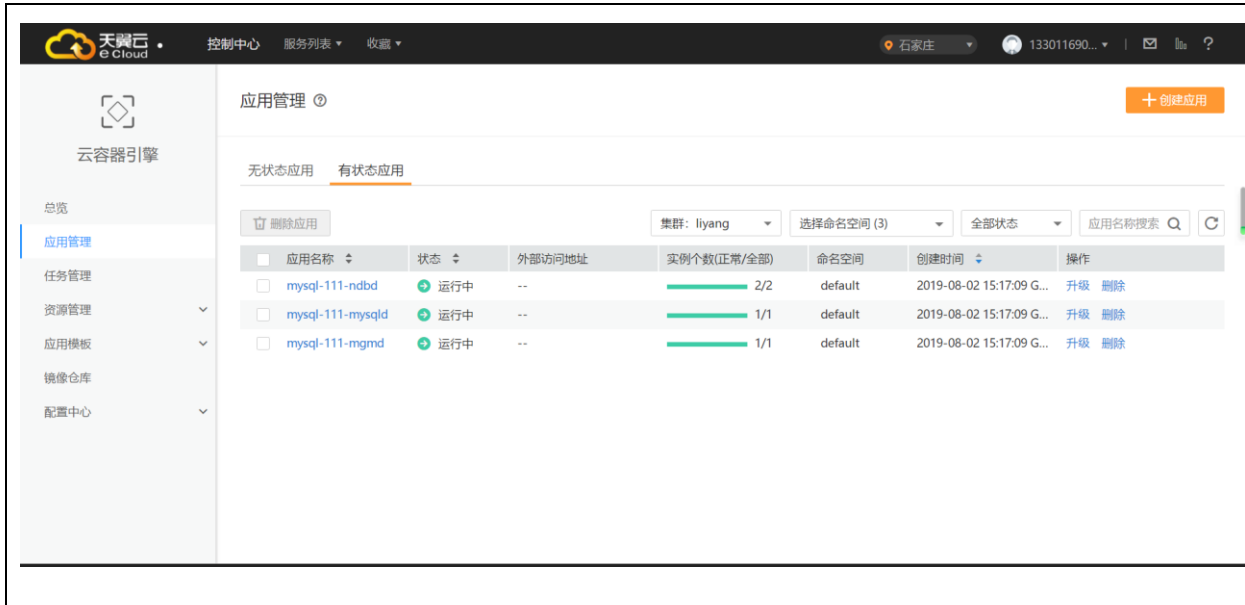
容器端口：游戏服务器的监听端口，配置为 8085。由于镜像中已写该端口，请不要修改。

访问端口：指定端口，需要与部署认证系统时环境变量中指定的 biboport 值保持一致。



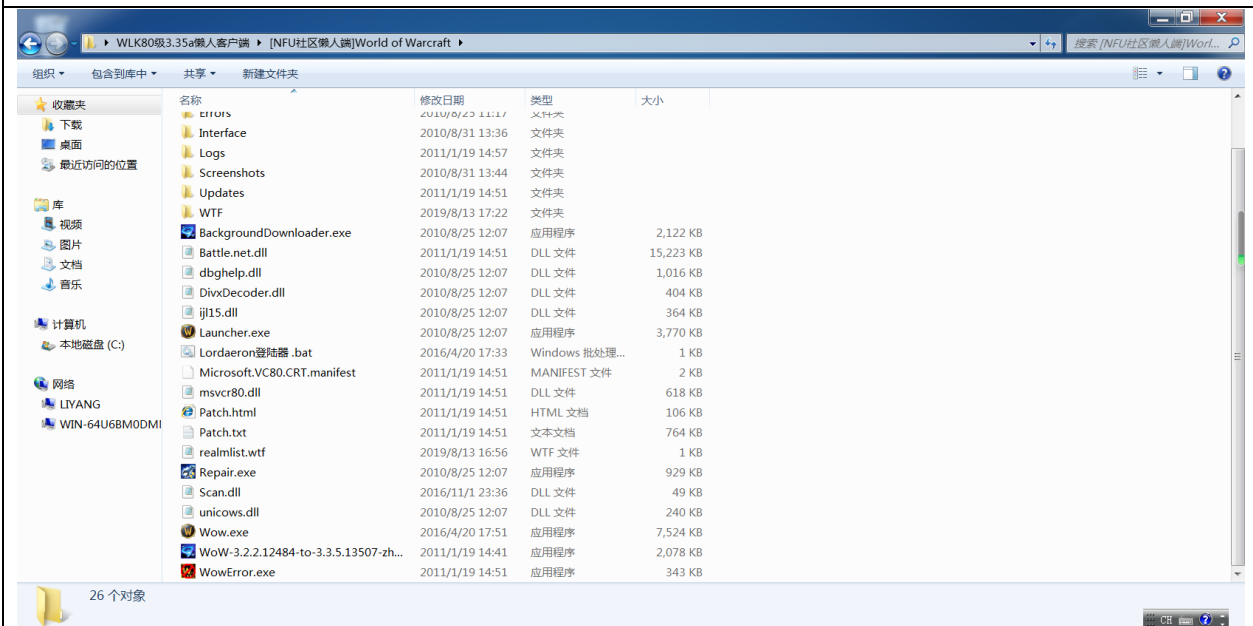
12. 至此样例部署已经完成





4.3 安装客户端验证服务

1. 准备 Windows7 服务器，下载安装 wow_cn_3.3.5.13930 客户端

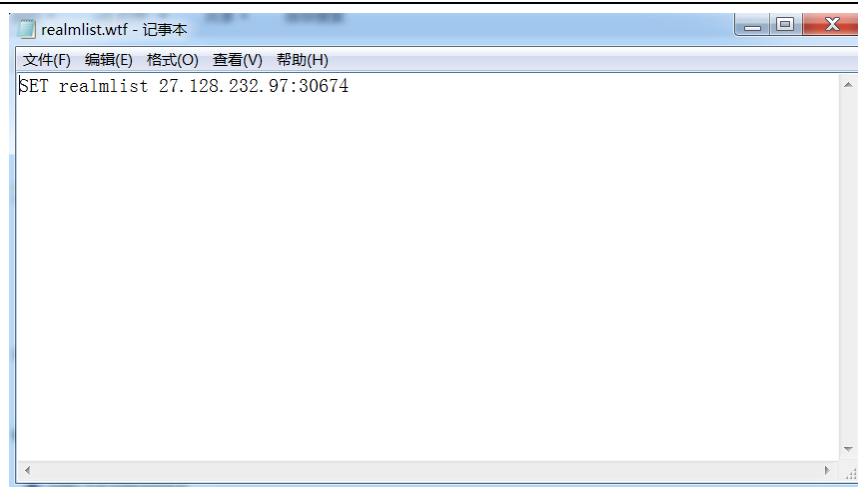


2. 获取服务器地址，以及 wow-auth 的服务端口

应用名称	状态	外部访问地址	实例个数(正常/全部)	命名空间	创建时间	操作
wow-world	运行中	27.128.232.97:32500	1/1	default	2019-08-02 16:04:09 G...	伸缩 升级 监控 更多
wow-auth	运行中	27.128.232.97:30674	2/2	default	2019-08-02 15:51:29 G...	伸缩 升级 监控 更多
kube-dns	可用	--	1/2	kube-system	2019-08-02 14:33:31 G...	伸缩 升级 监控 更多

3. 搜索文件夹下所有的 realmist.wtf 文件

设置为 SET realmist 服务器地址:wow-auth 的服务端口



4. 双击客户端程序 wow, 进入游戏。

本例为演示方便, 在镜像中内置了帐户, 用户真实部署游戏应用时, 会对接注册的接口。

用户名为 wangbo, 密码为 123456。



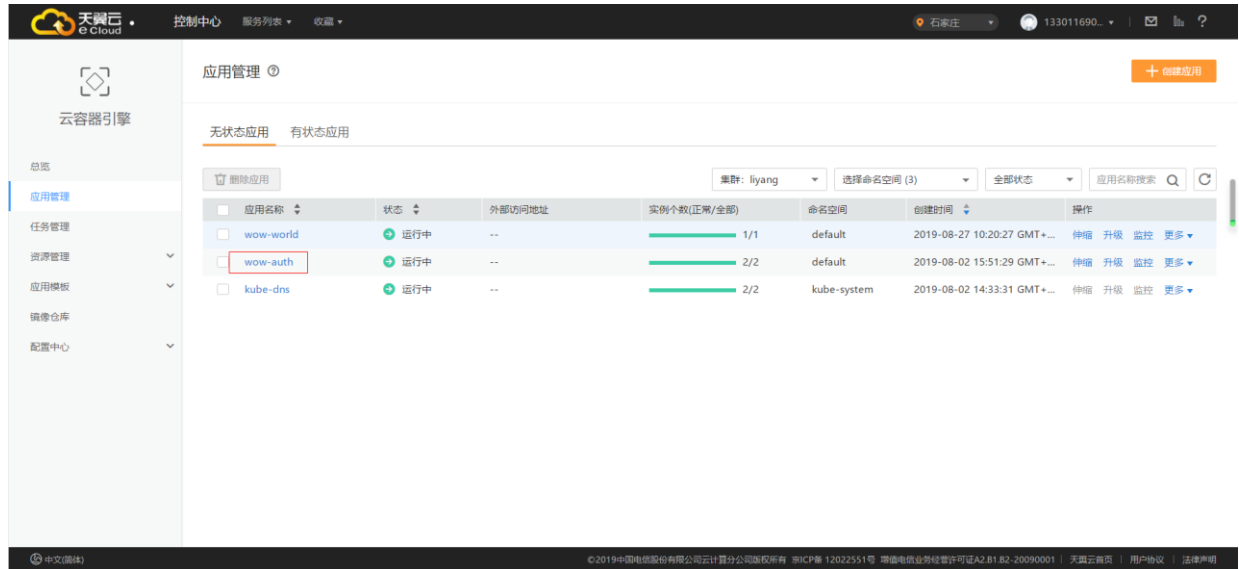
5. 登录成功



4.4 应用访问方式

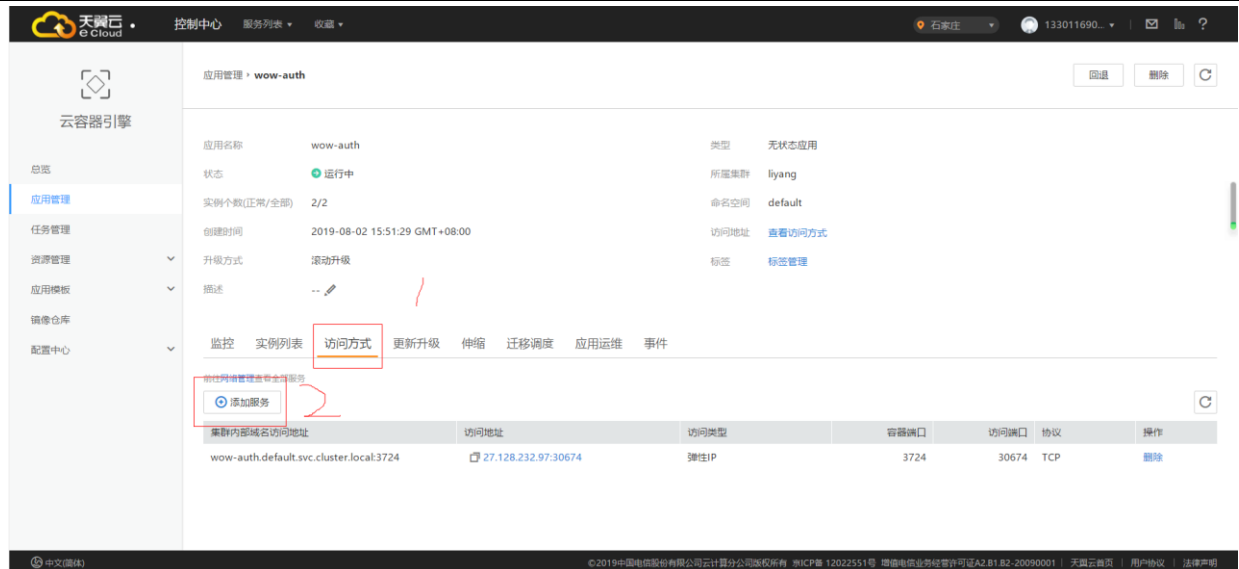
1. 选择应用

点击应用管理，选择一个应用。



2. 修改访问方式

点击访问方式选择添加服务



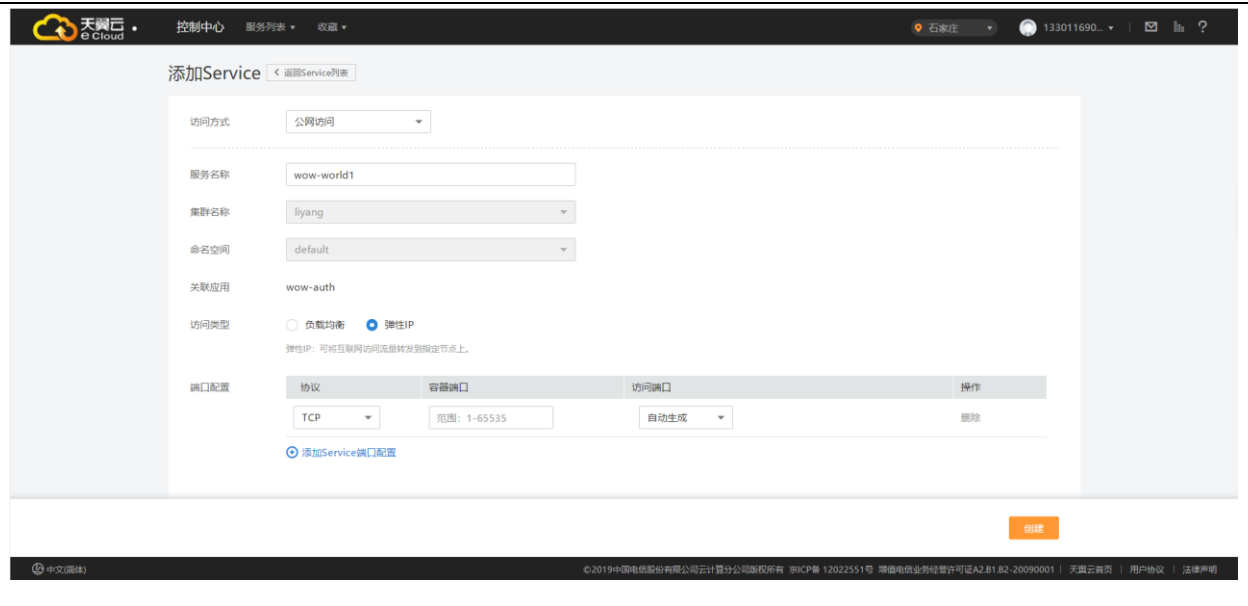
3. 选择访问方式

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

VPC 内网访问：是指应用可以让同一 VPC 内其他应用访问

公网访问：可以通过弹性 IP 从公网访问应用，一般用于系统中需要暴露到公网的服务。

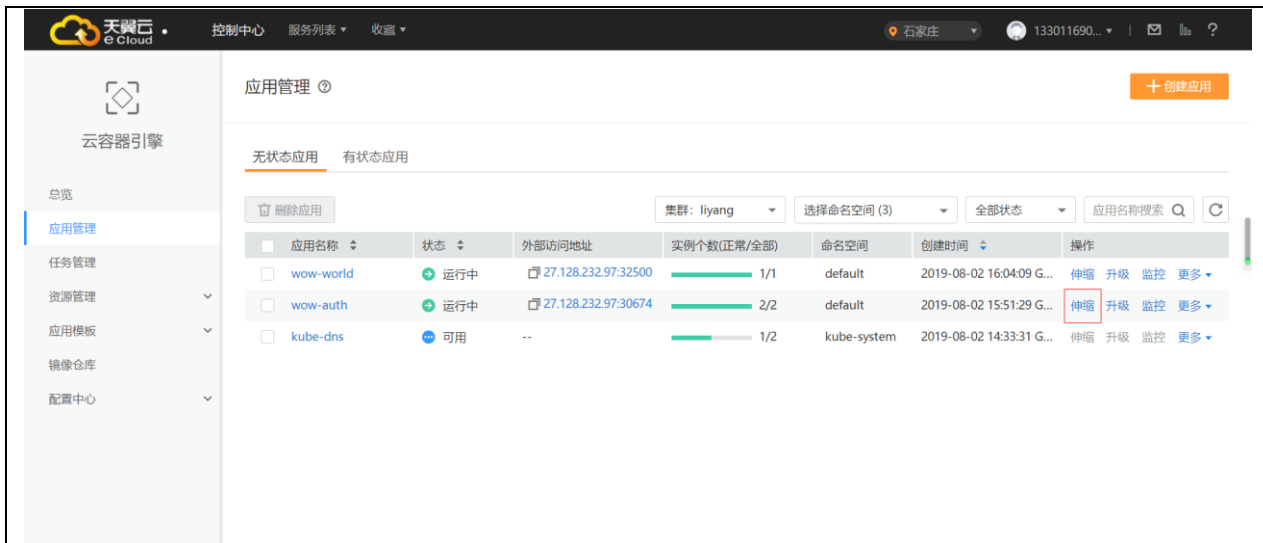
负载均衡(LoadBalancer)：可以通过弹性负载均衡从公网访问到工作负载，与弹性 IP 方式相比提供了高可靠的保障，一般用于系统中需要暴露到公网的服务。



4.5 伸缩应用服务

1. 手动伸缩

点击应用管理，选择伸缩选项。




应用管理

无状态应用 有状态应用

删除应用

集群: liyang 选择命名空间 (3) 全部状态 应用名称搜索

应用名称	状态	外部访问地址	实例个数(正常/全部)	命名空间	创建时间	操作
wow-world	运行中	27.128.232.97:32500	1/1	default	2019-08-02 16:04:09 G...	伸缩 升级 监控 更多
wow-auth	运行中	27.128.232.97:30674	2/2	default	2019-08-02 15:51:29 G...	伸缩 升级 监控 更多
kube-dns	可用	--	1/2	kube-system	2019-08-02 14:33:31 G...	伸缩 升级 监控 更多

2. 选择“伸缩”页签，单击手动伸缩下的 ，将实例数修改为2，单击“保存”。



云容器引擎

状态 运行中 所属集群 liyang

实例个数(正常/全部) 1/1 命名空间 default

创建时间 2019-08-02 16:04:09 GMT+08:00 访问地址 查看访问方式

升级方式 滚动升级 标签 标签管理

描述 --

监控 实例列表 访问方式 更新升级 伸缩 迁移调度 应用运维 事件

手动伸缩

实例数  2 保存 取消

缩容策略 应用停止前命令的执行时间窗 (0-9999秒)，默认30秒

时间窗 (s) 30

弹性伸缩

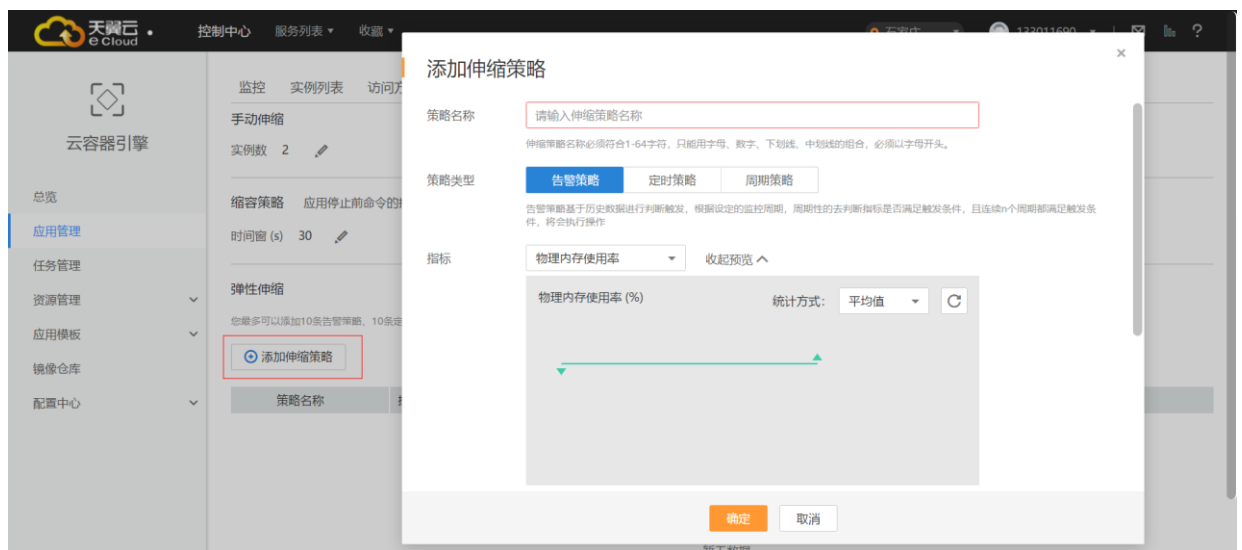
您最多可以添加10条告警策略、10条定时或周期策略，且策略总数小于等于20条

3. 实例秒级启动



4. 弹性伸缩

选择“伸缩”页签，单击“弹性伸缩”下的“添加伸缩策略”。



5. 伸缩策略建议


弹性伸缩策略建议	说明
周期策略	以某天某游戏的数据为例，午后和晚上达到峰值。

	<p>建议采用“周期策略”，比如在 2018 年 1 月 1 日到 2019 年 1 月 1 日，在每天 13:00，增加 100 个实例。</p>
告警策略	<p>一款新游戏上线，无法确定玩家数量，可能会出现大量玩家瞬间涌入，或无人问津。无法根据现有经验来预留适合数量的机器。</p> <p>建议采用“告警策略”，比如当 CPU/内存超过 70%时，增加 1 个实例；当 CPU/内存低于 40%时，减少 1 个实例。</p>
定时策略	<p>游戏中活动频繁，在活动之前配置好“定时策略”，比如 8 月 8 日的 12 点，增加 100 个实例。</p>

6. 添加告警策略

以 CPU 占用率为指标，添加伸缩策略，当 CPU 占用率超过 70%的时候，增加一个实例。

添加伸缩策略
×



触发条件 > %

监控周期 60秒 ?

连续周期 1 ?

执行操作 增加 1 个实例

确定
取消

4.6 升级应用服务

游戏特性需求快，版本变更频繁，升级效率以及升级过程中的用户体验至关重要。

CCE 提供滚动升级策略，对各个实例逐个进行单独更新，而非同时对所有实例进行全部更新的方式，确保升级过程中业务不中断。

本章节以 wow-auth 为例，演示应用的滚动升级。

请确保当前待升级的工作负载至少有 2 个实例。建议在手动伸缩到 2 个实例后，再进行升级操作。

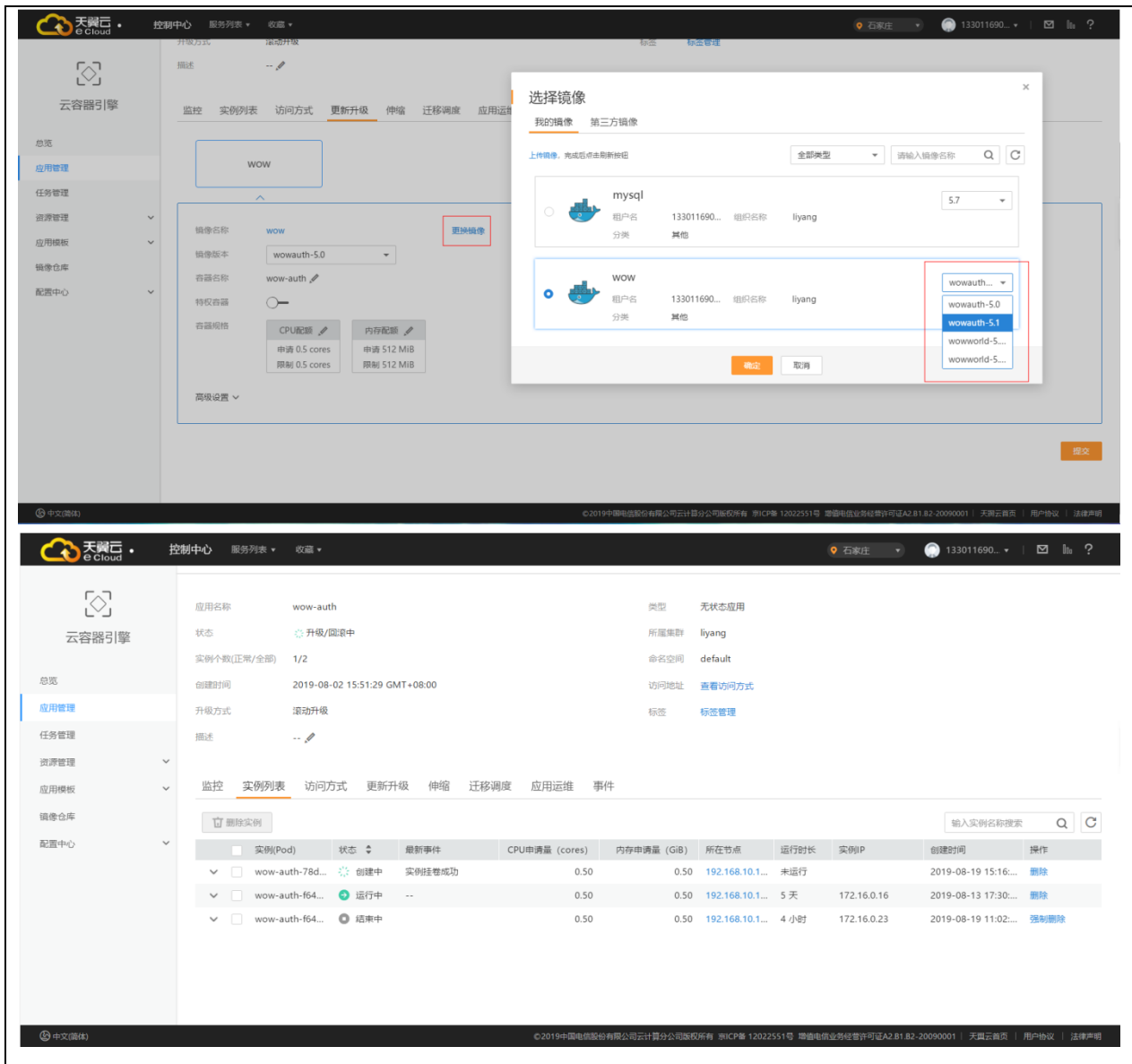
操作步骤：

步骤 1：登录 CCE 控制台。

步骤 2：在左侧导航栏中，单击“工作负载 > 无状态（Deployment）”，单击 wow-auth 名称，进入详情页面。

步骤 3：选择“更新升级”页签。工作负载在创建时，系统会默认为滚动升级。

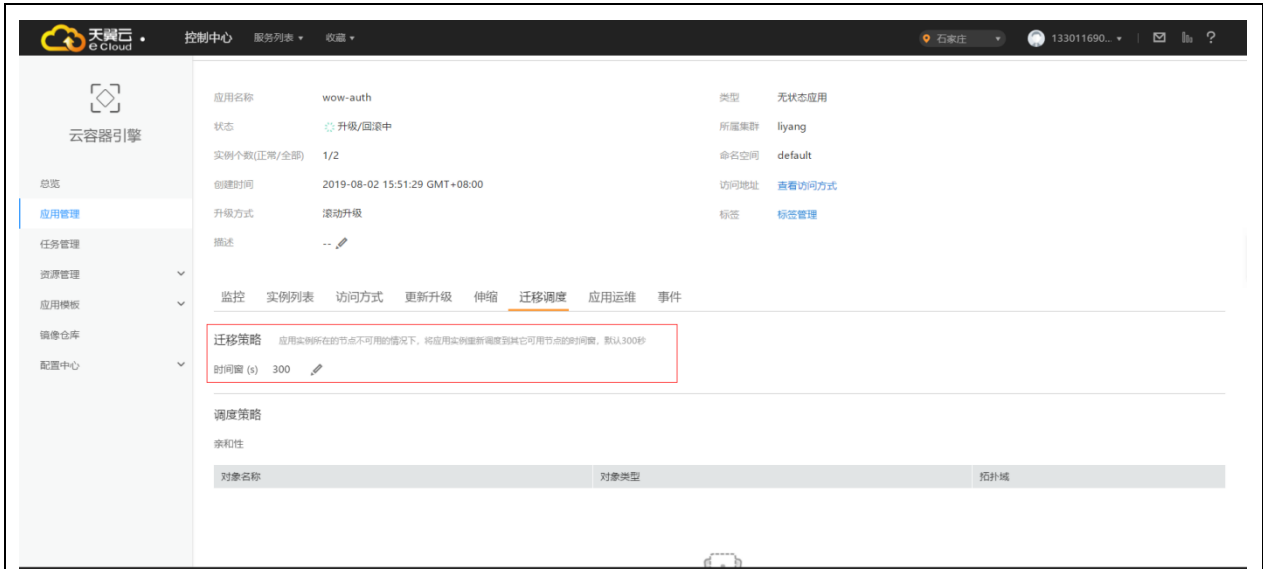
步骤 4：容器化之后，升级非常简单，只需要更换镜像即可。单击“更换镜像”，选择“wowauth-5.1”，单击“确定”。



4.7 应用服务迁移调度及亲和、反亲和性

1. 应用实例所在的节点不可用的情况下，将应用实例重新调度到其它可用节点的时间窗，默认 300 秒

点击可修改重新调度的时间



2. 点击企业中间件选项中的微服务云应用平台



3. 在应用上线中的应用管理选项中，选择调度策略



4. 添加亲和、反亲和规则

应用与可用区的亲和性：

亲和：决定应用部署在特定的可用区中。

反亲和：决定应用不能部署在特定的可用区中。

应用与节点间的亲和性：

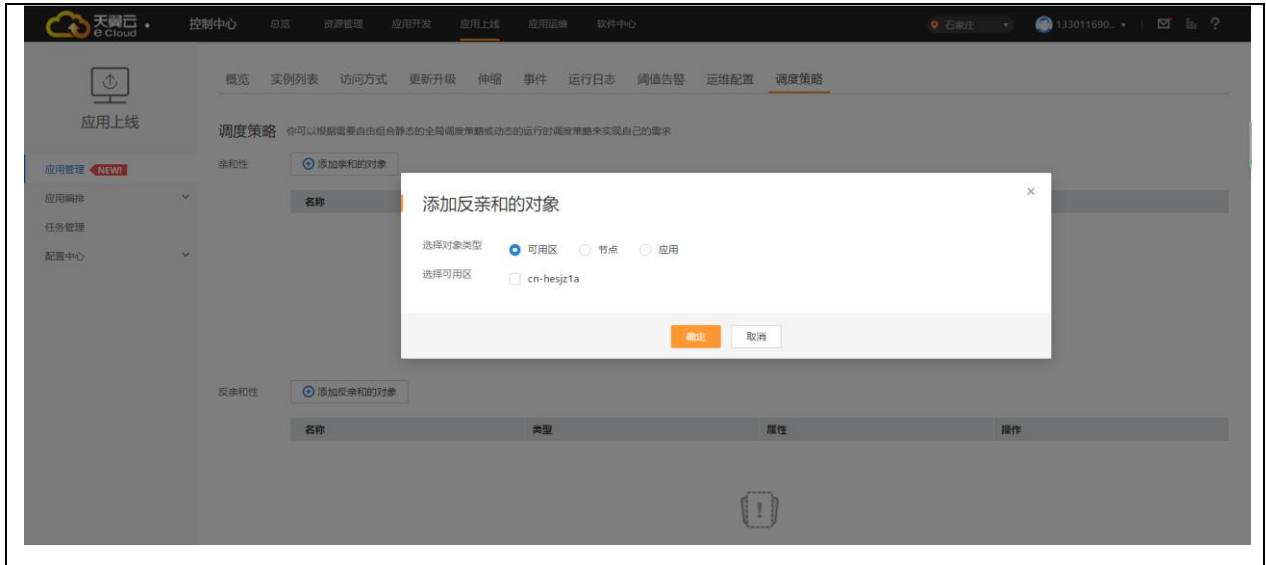
亲和：决定应用部署在某些特定的主机中。

反亲和：决定应用不能部署在某些特定的主机中。

应用间的亲和性：决定应用部署在相同或不同节点中。

亲和：用户可根据业务需求进行应用的就近部署，应用间通信就近路由，减少网络消耗。

反亲和：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。

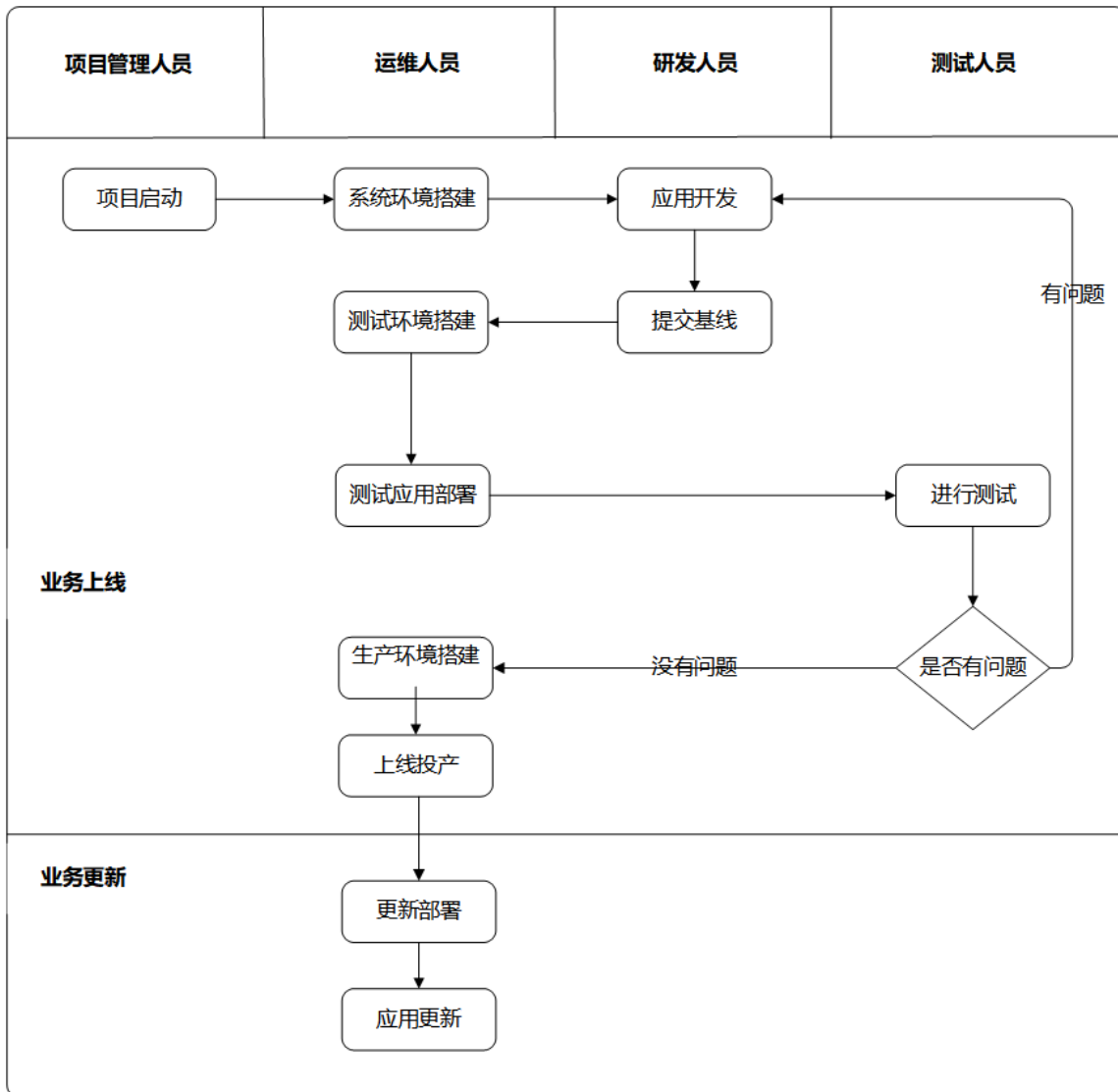


4.8 开发全流程对比分析

4.8.1 传统开发全流程

传统模式中，开发团队在开发环境中完成软件开发，本地完成单元测试，测试通过，则可提交到代码版本管理库；测试团队打包进行进一步测试。运维团队把应用部署到测试环境，开发团队或测试团队在册进行测试，没问题后通知布署人员发布到生产环境。

上述过程涉及到三个环境：开发、测试和生产，以及三个团队：开发、测试、运维。多个环境和多个团队之间的这种交互，很容易出现彼此环境不一致的情况，浪费不必要的人力物力。

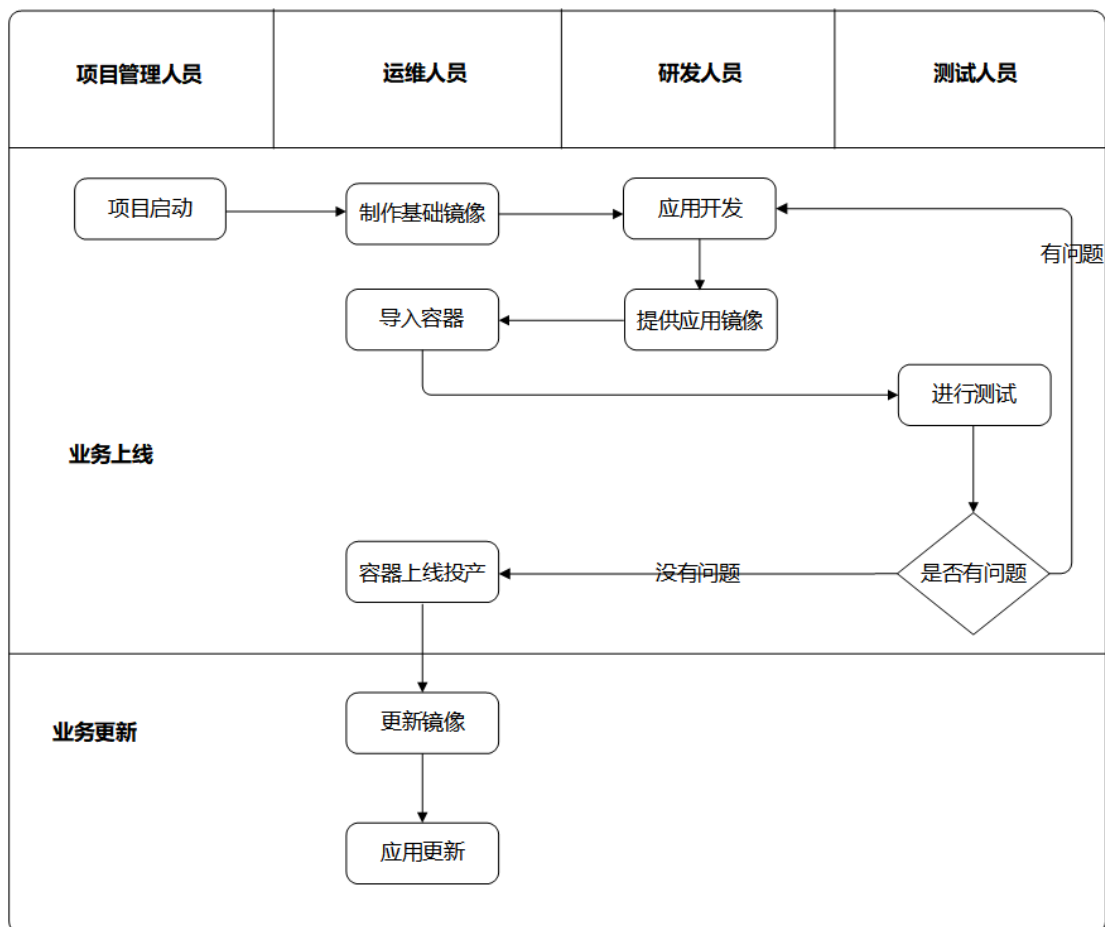


4.8.2 云容器开发全流程

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，您可以在天翼云上轻松部署、管理和扩展容器化应用程序。在容器模式中，应用以容器的形式存在，所有和该应用相关的依赖都会在容器中，因此移植非常方便，不会存在像传统模式中的环境不一致问题。项目开始，根据项目预期创建好需要的基础 base 镜像，nginx、tomcat、mysql 镜像或者将 Dockerfile 分发给所有开发人员，所有开发人员根据 Dockerfile 创建的容器

中进行开发达到开发环境的充分一致。若开发过程中需要添加新的软件，只需要申请修改基础 base 镜像的 Dockerfile 即可。

项目任务结束后，调整 Dockerfile 或者 Docker 镜像，然后分发给测试部门，测试部门马上就可以进行测试，消除了部署困难等纠缠的问题。



4.9 实践总结

云容器引擎对比传统虚拟化，如下：

特性	容器	虚拟化
启动	秒级	分钟级
硬盘使用	一般为 MB	一般为 GB
性能	相同规格配置, docker 性能约等于或 2 倍于 KVM	
Cpu	占用约 1.6%	占用约 14.6%
Memory	平均每个容器 46M	平均每个 VM 185M
基础环境部署时长	几个小时, 服务封装为统一镜像, 环境一致	几天到一个星期, 服务所需依赖复杂, 容易出错
环境部署时间占总上线时间比例	<10%	>50%
所需人员	少数人员	大量人员
自动化程度	高	低
安全性	高, 业务被隔离在不同容器内, 互相不受影响	低, 耦合性强, 一旦被攻击, 容易全部丢失
总体上线时间	与虚拟化相比可节约一半时长	环境部署时间+业务上线时间
成本	前期固定成本较高, 随之业务扩大, 全部运行在容器之中, 所需主机几乎不变	业务越多所需主机越多, 耗时越长

综上所述, 使用云容器引擎在启动速度、性能、时间、人员、成本等方面相比较于虚拟化有更大的优势。

5 常见问题

5.1 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。

5.2 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。

5.3 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。

5.4 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。

5.5 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。

5.6 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。

登录天翼云控制中心，点击产品列表中的【网络】【虚拟私有云】；

在左侧导航树，单击【弹性 IP】；

单击【申请弹性 IP】；

在申请弹性 IP 页面，选择计费方式、带宽值，点击【立即创建】；

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

绑定成功即可实现应用的外网访问；

5.7 如何解决新增节点时提示弹性 IP 不足的问题?

问题描述：新增节点时，弹性 IP 选择“现在使用”，创建节点失败，提示弹性 IP 不足。

解决方法：您可以有两种方法解决弹性 IP 不足的问题。

方法一：解绑已绑定弹性 IP 的云主机，再重新添加节点；

方法二：提高弹性 IP 的配额，您需要通过工单系统来提交申请。

5.8 创建容器应用有哪几种方式？有什么区别？

目前支持两种部署方式，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。

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。

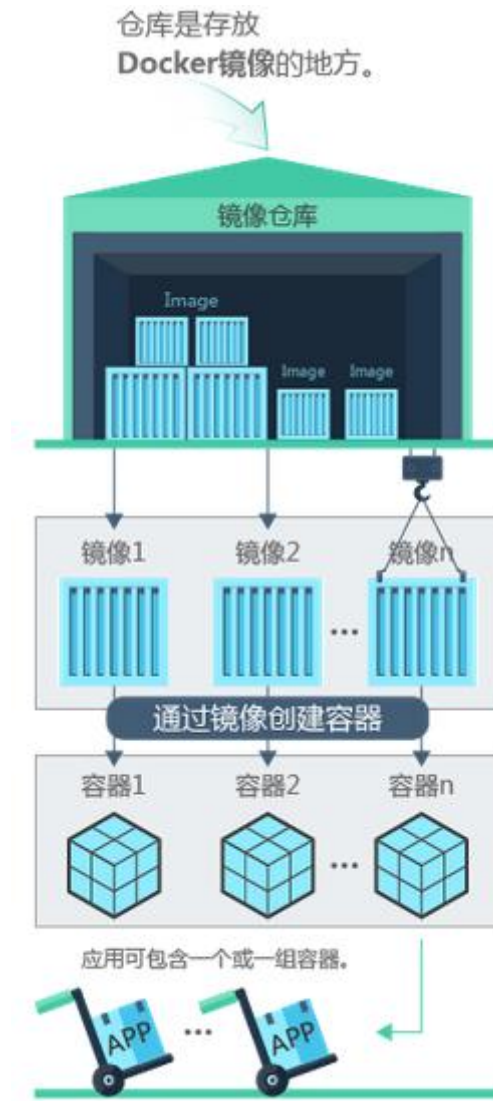
5.9 镜像、容器、应用的关系是什么？

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：



5.10 如何获取长期有效的 docker login 指令？

操作步骤

获取镜像仓库访问地址、区域项目名称。

3. 访问我的凭证：登录控制台，鼠标移动到右上角您的用户名处，单击“我的凭证”。

访问我的凭证



4. 在“项目列表”页签中查找当前区域对应的项目。

在容器镜像服务控制台中获取镜像仓库地址。

获取该地址的方式：单击“我的镜像”，单击镜像列表中的镜像名称，在“Pull/Push 指南”页签中的“**1. 本地镜像地址**”下可以看到镜像仓库地址。

获取镜像仓库地址

镜像版本 描述 权限管理 触发器 **Pull/Push指南**

前提条件:

准备一台计算机,要求安装的Docker版本必须为1.11.2及以上

1. 本镜像地址:

下载地址:100.193.72.84/test10/cse-mesher

2. 操作步骤:

Step 1. 以root用户登录Docker所在的虚拟机

Step 2. 获取登录Docker访问权限，并复制到节点上执行

请点击 [生成临时docker login指令](#) ([获取长期有效docker login指令](#))

获取 AK/SK 访问密钥。



如果已有 AK/SK，可以直接使用，无需再次获取。

1. 访问我的凭证

步骤：登录控制台，鼠标移动到右上角您的用户名处，单击“我的凭证”。

2. 在“管理访问密钥”页签，单击列表下侧的“新增访问密钥”，创建新的访问密钥。

新增访问密钥

项目列表 **管理访问密钥**

访问密钥对账号具有完全的访问权限，**如果访问密钥泄露，会带来数据泄露风险**，为了账号安全性，建议您定期更换并妥善保存访问密钥。

[+ 新增访问密钥](#) 您还可以添加0个访问密钥。

3. 输入当前用户的登录密码，并通过邮箱或者手机进行验证。



在统一身份认证服务中创建的用户，如果创建时未填写邮箱或者手机号，则只需校验登录密码。

4. 单击“确定”，下载访问密钥。

请妥善保存已下载的密钥，后续创建集群时需要上传该密钥，否则会无法创建集群。

 说明

为防止访问密钥泄露，建议您将其保存到安全的位置。

登录一台 linux 系统的计算机，执行如下命令获取登录密钥。

```
printf "$AK" | openssl dgst -binary -sha256 -hmac "$SK" | od -An -vtx1 | sed 's/[ \n]//g' | sed 'N;s/\n/'
```

其中\$AK 和\$SK 为[步骤 2](#) 获取的 AK/SK。

示例

```
[root@SZV1000258977 ~]# printf "DKAKX9J60BEVMARHLBQM" | openssl dgst -binary -sha256 -hmac "0uDrd9HcRhmngEhAXo6SQiflN1UqufLF531jiFkX" | od -An -vtx1 | sed 's/[ \n]//g' | sed 'N;s/\n/'  
7ca3582173f52caa98fcf87389e9cc26d007a2e4b2f6231006a301568f2e1ef8
```

使用如下的格式拼接 docker login 指令。

```
docker login -u [区域项目名]@[AK] -p [登录密钥] [镜像仓库地址]
```

其中，区域项目名和镜像仓库地址在[步骤 1](#) 中获取，AK 在[步骤 2](#) 中获取，登录密钥为[步骤 3](#) 的执行结果。

5.11 如何通过 kubectl 命令行创建负载均衡？

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[错误！未找到引用源](#)。ConfigMap 配置项要求配置 kubectl 命令，使弹性云服务器连接集群。

步骤 1：登录已配置好 kubectl 命令的弹性云服务器。登录方法请参见[登录 Linux 弹性云服务器](#)。

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: 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
```

vi nginx-elb-svc.yaml



说明

若需要开启会话保持，需要满足如下条件：

- 应用管理协议为 TCP。
- 应用管理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。详细请参见[指定应用管理部署在不同节点](#)。

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubernetes.io/elb.class: union #对接增强型负载均衡需要加此参数，若对接经典型负载均衡（已停
售），值为elasticity
    kubernetes.io/elb.id: 3c7caa5a-a641-4bff-801a-feace27424b6 #可选，为负载均衡增强型实例的 ID
    kubernetes.io/elb.subnet-id: 5083f225-9bf8-48fa-9c8b-67bd9693c4c0 #可选，当自动创建时必须，
1.11.7-r0 以上的版本可不填
    kubernetes.io/elb.autocreate: >-

{"type": "public", "bandwidth name": "cce-bandwidth-1558492655112", "bandwidth chagemode": "bandwidth
h", "bandwidth_size": 5, "bandwidth_sharetype": "PER", "eip_type": "5_sbgp"}
```

```
#必填
labels:
  app: nginx
  name: nginx
spec:
  loadBalancerIP: 10.78.42.242          #配置为 ELB 的 IP 地址
  ports:
  - name: service0
    port: 80          #集群虚拟 IP 的访问端口，也是注册到负载均衡上的端口
    protocol: TCP
    targetPort: 80    #对应界面上的容器端口
  selector:
    app: nginx
  type: LoadBalancer
```

步骤 3：创建应用管理。

```
kubectl create -f nginx-deployment.yaml
```

回显如下，表示应用管理已开始创建。

```
deployment "nginx" created
```

```
kubectl get po
```

回显如下，应用管理状态为 Running 状态，表示应用管理已运行中。

NAME	READY	STATUS	RESTARTS	AGE
etcd-0	0/1	ImagePullBackOff	0	1h
icagent-m9dkt	0/0	Running	0	3d
nginx-2601814895-clxhw	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，表示应用管理访问方式已设置成功，应用管理可访问。

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 10.78.42.242:31540。10.78.42.242 为负载均衡实例 IP 地址，31540 为对应界面上的访问端口。

6 参考知识

6.1 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：

7 修订记录

修订记录

发布日期	修改说明
2018-03	首次发布云容器引擎用户指南
2019-09	更新章节应用访问设置、容器镜像部分内容， 增加最佳实践章节
2019-12	补充云容器引擎 IPv6 能力